SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

February 2025, Vol. 68, Iss. 2, 122402:1–122402:14
https://doi.org/10.1007/s11432-023-3990-4

Smaller, faster, lower-power analog RRAM matrix computing circuits without performance compromise

Yubiao LUO[1,2], Pushen ZUO[1,2], Shiqing WANG[1,2], Zhong SUN[1,2,3*] & Ru HUANG[1,2,3]

[1]*School of Integrated Circuits, Peking University, Beijing 100871, China*
[2]*Institute for Artificial Intelligence, Peking University, Beijing 100871, China*
[3]*Beijing Advanced Innovation Center for Integrated Circuits, Beijing 100871, China*

Received 3 August 2023/Revised 14 November 2023/Accepted 25 March 2024/Published online 14 January 2025

Abstract Recently, the analog matrix computing (AMC) concept has been proposed for fast, efficient matrix operations, by configuring global feedback loops with crosspoint resistive memory arrays and operational amplifiers (OAs). The implementation of a general real-valued matrix (containing both positive and negative elements) is enabled by using a set of analog inverters, which, however, is considered inefficient regarding circuit compactness, power consumption, and temporal response. Here, with the assistance of the conductance compensation (CC) strategy to take full advantage of the inherent differential inputs of OAs, new AMC circuits without analog inverters are designed. Such a design saves the area occupation and power dissipation of analog inverters, and thus turns to be smaller and lower-power. Simulation results reveal that the new circuit also shows a faster response towards the steady state, thanks to the reduction of poles in the circuit, which, again, is contributed by the elimination of analog inverters. Along with all of these benefits, extensive simulations demonstrate that the CC-AMC circuits do not compromise the computing performance in terms of relative error caused by various non-ideal factors in the circuit.

Keywords analog computing, in-memory computing, matrix, resistive memory, conductance compensation

Citation Luo Y B, Zuo P S, Wang S Q, et al. Smaller, faster, lower-power analog RRAM matrix computing circuits without performance compromise. Sci China Inf Sci, 2025, 68(2): 122402, https://doi.org/10.1007/s11432-023-3990-4

1 Introduction

Matrix problems pose a significant challenge for traditional digital computers, which rely on complex algorithms such as lower-upper factorization and QR decomposition [1, 2]. These methods exhibit high computational complexities, primarily owing to the sequential nature of arithmetic operations, the reliance on Boolean logic gates, and constraints imposed by the von Neumann architecture [3]. This issue has become more pronounced with the exponential increase in data volumes associated with matrix-related applications, such as artificial intelligence models [4] and the deployment of 5G/6G massive multiple-input multiple-output systems [5]. Furthermore, the gradual diminishing returns of Moore's law exacerbate the need for more efficient computational approaches [6]. In response to these challenges, analog matrix computing (AMC) has arisen as a promising solution, offering fast response and massive parallelism. These advantages significantly reduce computational complexity [7–10].

AMC leverages crosspoint resistive memory arrays to solve matrix problems in a single step, including tasks such as matrix inversion, generalized inverse calculations, and eigenvector determination [11–19]. The circuit functionalities rely on the global feedback loops configured with traditional analog components, e.g., operational amplifiers (OA) and transimpedance amplifiers (TIA). Among the various resistive memory technologies suitable for AMC, resistive random-access memory (RRAM) stands out for its excellent analog conductance capability, simple structure, and compatibility with existing contemporary complementary metal-oxide-semiconductor (CMOS) technology [20, 21]. In AMC, the conductance values of RRAM devices represent matrix elements. However, this approach initially supports the mapping of positive values. To accommodate matrices with both positive and negative elements, analog inverters are employed to implement column-wise splitting (CS) of each matrix column into two columns of RRAM devices [22–29]. However, it results in a significant overhead of circuit area and energy consumption. Furthermore, the limited open-loop gain of OAs can lead to accuracy loss. The inclusion of analog inverters

* Corresponding author (email: zhong.sun@pku.edu.cn)

raises an extra source of accuracy loss and complicates the circuit dynamics by introducing additional poles, which can increase circuit sensitivity and delay the transient response.

In this paper, we present new designs for AMC circuits that eliminate the need for analog inverters in processing real-valued matrices. These new designs utilize the inherent differential inputs of a global feedback OA, achieving the conductance difference necessary for mapping arbitrary real-valued elements. This is made possible through a conductance compensation (CC) strategy [30], which was originally developed to enhance the efficiency of real-valued matrix-vector multiplication (MVM). Extending the CC concept to AMC circuits for solving matrix equations demands a comprehensive investigation owing to the fundamentally different principles these circuits operate on. Unlike MVM circuits that perform forward multiplication, equation-solving circuits tackle inverse problems and are constructed around global feedback loops. When redesigning these circuits with a fresh principle, multiple aspects should be addressed. These include addressing circuit stability and response time, examining the intrinsic condition number issue and conducting an accuracy analysis. For circuits designed to compute generalized inverses, which utilize two arrays, the CC method should be applied to both row and column sums, unlike single-array circuits where only row sum is considered. The removal of analog inverters from CC-AMC circuits results in designs that are more compact, consume less power, and operate faster compared to their CS-AMC counterparts. Extensive simulations have been performed on the matrix inversion circuits across various matrix dimensions and condition numbers. The results show that the CC-AMC circuits maintain accuracy performance regarding device and circuit non-idealities, including conductance variations, parasitic resistances and capacitances, as well as source and sink resistances.

The rest of this paper is organized as follows. Section 2 outlines the basic structure of three types of CS-based AMC circuits, including those for matrix inversion, eigenvector calculation, and generalized inverse computation. Section 3 presents the newly developed CC-AMC circuits for these same applications, detailing their underlying principles and evaluating the benefits in terms of reduced circuit area and lower-power consumption. In Section 4, we compare the simulation results of two matrix inversion circuit designs, highlighting that the new design does not compromise performance in the face of device and circuit non-idealities. Section 5 draws a conclusion to this work.

2 CS-based AMC circuits

In Figure 1, the CS-AMC circuits for solving matrix problems are illustrated. The matrix inversion circuit in Figure 1(a) solves a system of linear equations in the matrix form, namely

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}, \tag{1}$$

where $\boldsymbol{A}$ is a square $n \times n$ matrix, $\boldsymbol{y}$ is a known $n \times 1$ vector, and $\boldsymbol{x}$ is the unknown $n \times 1$ vector to be solved. By mapping $\boldsymbol{A}$ as the equivalent conductances $\boldsymbol{G_A}$ in a crosspoint RRAM array, and upon the injection of currents $-\boldsymbol{I_y}$ that represent vector $\boldsymbol{y}$ to the rows, the steady-state output voltages $\boldsymbol{V_x}$ give the solution of $\boldsymbol{x}$, since they have to satisfy $\boldsymbol{G_A V_x} - \boldsymbol{I_y} = \boldsymbol{0}$, which is a precise mapping of (1) in the circuit.

The eigenvector circuit in Figure 1(b) solves the eigenvector $\boldsymbol{x}$ of a given eigenvalue λ, which is described by the following matrix equation:

$$\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x}. \tag{2}$$

The negation of $\boldsymbol{A}$ is mapped as equivalent conductances $-\boldsymbol{G_A}$ of the RRAM array, and λ is mapped as the feedback conductance of all TIAs. There is no external input in the eigenvector circuit, rather it works in a self-sustained manner. Upon the enablement of OAs, the circuit dynamics involves until the output voltages $\boldsymbol{V_x}$ of TIAs stabilize, satisfying $-\boldsymbol{G_A V_x} + g_\lambda \boldsymbol{V_x} = \boldsymbol{0}$ and thus providing the eigenvector solution.

The generalized inverse circuit in Figure 1(c) solves (1) as well. However, it addresses situations where $\boldsymbol{A}$ is a non-square $n \times m$ matrix. If $\boldsymbol{A}$ is a tall matrix (more rows than columns, $n > m$), it is mapped as $\boldsymbol{G_A}$ in two RRAM arrays in the circuit, and the input vector $-\boldsymbol{I_y}$ is fed to the rows of the left array. Consequently, the steady-state output voltages $\boldsymbol{V_x}$ on the left columns are subject to $\boldsymbol{G_A^{\mathrm{T}}}\left(\boldsymbol{I_y} - \boldsymbol{G_A V_x}\right) = \boldsymbol{0}$, resulting in the left inverse solution, namely $\boldsymbol{V_x} = \left(\boldsymbol{G_A^{\mathrm{T}}} \cdot \boldsymbol{G_A}\right)^{-1} \cdot \boldsymbol{G_A^{\mathrm{T}}} \boldsymbol{I_y}$. If $\boldsymbol{A}$ is a broad matrix (more columns than rows, $n < m$), by contrast, the transpose of $\boldsymbol{A}$ is mapped as $\boldsymbol{G_A^{\mathrm{T}}}$ in the two RRAM arrays, $-\boldsymbol{I_y}$ is fed to the right columns, the steady-state voltages $\boldsymbol{V_z}$ on the left columns

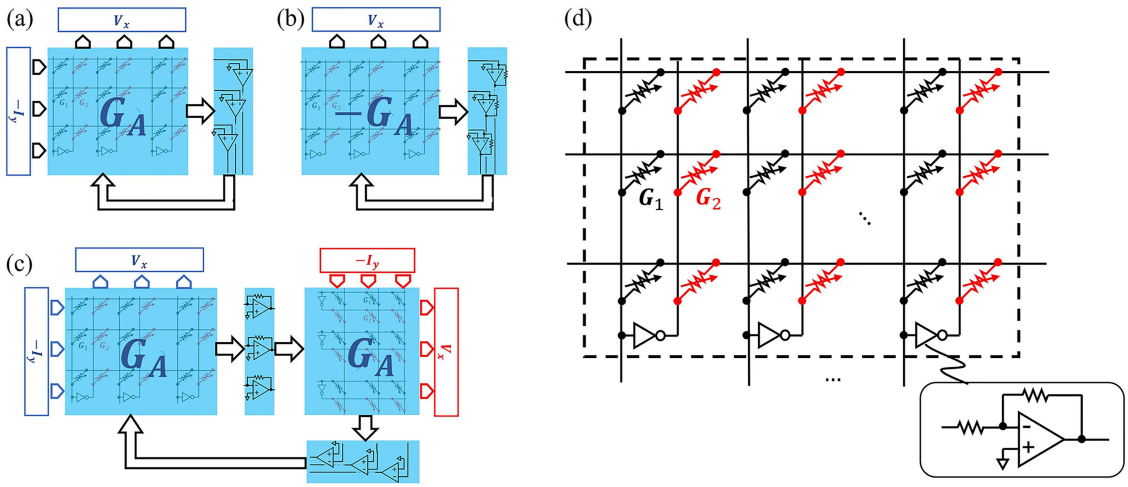

Figure 1 (Color online) CS-AMC circuits for matrices with negative entries, and the structure of the crosspoint array. (a) CS-AMC circuit for solving the matrix inversion problem of (1). (b) CS-AMC circuit for solving the eigenvector problem of (2). (c) CS-AMC circuit for solving (1) of a non-square matrix $\boldsymbol{A}$. It has two input/output interfaces, for solving a left inverse or right inverse problem corresponding to a tall or broad matrix. (d) Matrix $\boldsymbol{A}$ is represented by the difference between two conductance matrices $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ with analog inverters connecting two adjacent columns.

and $\boldsymbol{V_x}$ on the right rows are described by a combination of two equations, i.e., $\frac{1}{g_0}\boldsymbol{G_A}\boldsymbol{G_A}^{\mathrm{T}}\boldsymbol{V_z}+\boldsymbol{I_y}=0$ and $\boldsymbol{V_x}=-\frac{1}{g_0}\boldsymbol{G_A}^{\mathrm{T}}\boldsymbol{V_z}$. The result of $\boldsymbol{V_x}$ is given by $\boldsymbol{V_x}=\boldsymbol{G_A}^{\mathrm{T}}\cdot\left(\boldsymbol{G_A}\cdot\boldsymbol{G_A}^{\mathrm{T}}\right)^{-1}\boldsymbol{I_y}$, which represent the right inverse solution to (1).

3 New CC-based AMC circuits

3.1 CC-based matrix inversion circuit

In Figure 2(a), we present a new matrix inversion circuit based on the CC strategy. It eliminates the use of analog inverters, by connecting the RRAM arrays to the differential input terminals that are inherent of OAs. In contrast to the CS-AMC circuits in Figure 1, the new circuit adopts the row-wise splitting (RS) scheme, and the RRAM rows connected to the inverting and non-inverting terminals of OAs constitute the conductance matrices $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$, respectively. An extra column of RRAM devices is designed for CC. In the circuit, the outputs of OAs are fed back to the columns in order as usual.

According to the Kirchhoff's current law (KCL), the input voltages of the k-th row OA, v_k^- and v_k^+, are given by

$$v_k^-=\frac{\sum_j G_{1,kj}V_{x,j}}{\sum_j G_{1,kj}+g_{1k}},\ v_k^+=\frac{\sum_j G_{2,kj}V_{x,j}+g_0V_{y,k}}{\sum_j G_{2,kj}+g_{2k}+g_0},\tag{3}$$

where $G_{1,kj}$ and $G_{2,kj}$ are the conductance of the j-th device in k-th row. Considering the virtual short characteristics of OA, namely $v_k^-=v_k^+$, there is

$$\frac{\sum_j G_{1,kj}V_{x,j}}{\sum_j G_{1,kj}+g_{1k}}=\frac{\sum_j G_{2,kj}V_{x,j}+g_0V_{y,k}}{\sum_j G_{2,kj}+g_{2k}+g_0},\tag{4}$$

$$\text{i.e.,}\ U_{1,k}^{-1}\sum_j G_{1,kj}V_{x,j}=U_{2,k}^{-1}\left(\sum_j G_{2,kj}V_{x,j}+g_0V_{y,k}\right),$$

where $U_{1,k}=\sum_j G_{1,kj}+g_{1k}$ and $U_{2,k}=\sum_j G_{2,kj}+g_{2k}+g_0$. Therefore, considering (4) for all rows, it forms a matrix equation, that is

$$\boldsymbol{U}_1^{-1}\boldsymbol{G}_1\boldsymbol{V_x}=\boldsymbol{U}_2^{-1}\left(\boldsymbol{G}_2\boldsymbol{V_x}+g_0\boldsymbol{V_y}\right),\tag{5}$$

where $\boldsymbol{V_y}$ and $\boldsymbol{V_x}$ are input and output voltages defined as $\boldsymbol{V_y}=[V_{y,1},V_{y,2},\ldots,V_{y,n}]^{\mathrm{T}}$ and $\boldsymbol{V_x}=[V_{x,1},V_{x,2},\ldots,V_{x,n}]^{\mathrm{T}}$, respectively. g_0 is the reference conductance for mapping the matrix $\boldsymbol{A}$, namely $\boldsymbol{G_A}=$

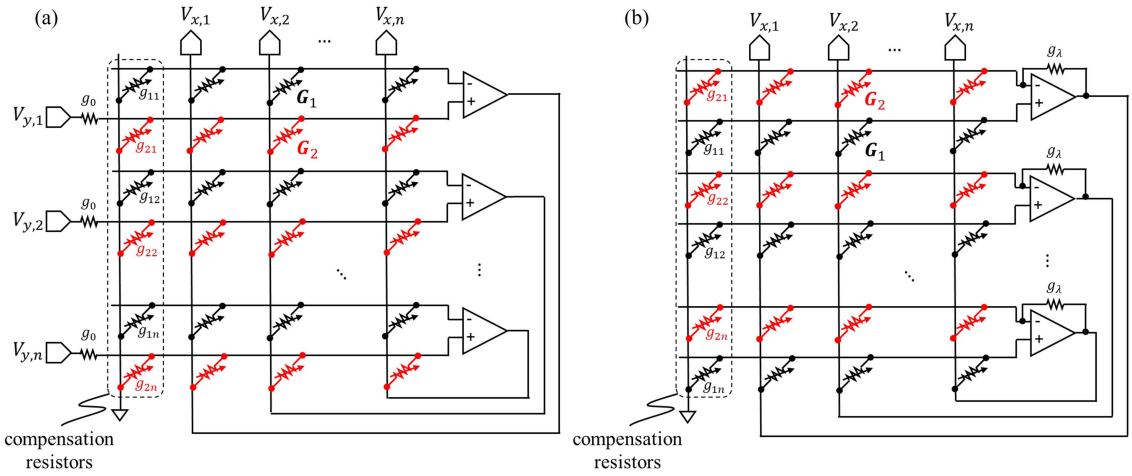

Figure 2 (Color online) (a) CC-AMC matrix inversion circuit. The leftmost column of devices in the crosspoint array is used for conductance compensation. This ensures that the total conductance values of every pair of rows connected to a single OA are identical. (b) CC-AMC eigenvector circuit. Again, the leftmost column of devices is used for conductance compensation.

$g_0\boldsymbol{A}$. g_{1k} and g_{2k} are the conductances of the compensation resistors connected to the inverting and non-inverting input terminals of the k-th OA, respectively. $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$ are diagonal matrices defined as $\boldsymbol{U}_1 = \mathrm{diag}(\sum_j \boldsymbol{G}_{1,kj} + g_{1k}, k = 1, 2, \ldots, n)$, $\boldsymbol{U}_2 = \mathrm{diag}(\sum_j \boldsymbol{G}_{2,kj} + g_{2k} + g_0, k = 1, 2, \ldots, n)$.

To enable the matrix inversion computation in the circuit, compensation conductances are selected to render the equality of the conductance summations of every two rows of resistive devices, i.e., $\sum_j \boldsymbol{G}_{1,kj} + g_{1k} = \sum_j \boldsymbol{G}_{2,kj} + g_{2k} + g_0, k = 1, 2, \ldots, n$, which translates to

$$g_{2k} - g_{1k} = \sum_j \boldsymbol{G}_{kj} - g_0 = \left(\sum_j \boldsymbol{A}_{kj} - 1 \right) g_0. \tag{6}$$

Based on the prior knowledge of row sums $\sum_j \boldsymbol{A}_{kj}$ of the matrix, every pair of compensation conductances g_{1k} and g_{2k} can be determined. Specifically, according to the sign of the result in (6), g_{1k} or g_{2k} could be set to 0 and the other one is given the absolute value of the result.

The assumption of (6) implies the equality $\boldsymbol{U}_1 = \boldsymbol{U}_2$. As a result, Eq. (5) is simplified as

$$(\boldsymbol{G}_1 - \boldsymbol{G}_2)\, \boldsymbol{V}_{\boldsymbol{x}} = g_0 \boldsymbol{V}_{\boldsymbol{y}}, \tag{7}$$

$$\text{i.e.,} \frac{\boldsymbol{G}_1 - \boldsymbol{G}_2}{g_0} \boldsymbol{V}_{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{V}_{\boldsymbol{x}} = \boldsymbol{V}_{\boldsymbol{y}},$$

which is exactly the analog of the matrix inversion problem of (1). Upon the stimuli of the input voltages $\boldsymbol{V}_{\boldsymbol{y}}$, the circuit quickly responds and stabilizes, with the output voltages $\boldsymbol{V}_{\boldsymbol{x}}$ of OAs representing the solution.

3.2 CC-based eigenvector circuit

The CC-based matrix eigenvector circuit is shown in Figure 2(b). Likewise, the crosspoint RRAM array includes a column of CC devices to facilitate the mapping of (2) in the circuit without analog inverters. Since it is the negation of $\boldsymbol{A}$ (instead of $\boldsymbol{A}$) to be mapped, the positions of the conductance matrices $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ are interchanged, in contrast to the CC-based matrix inversion circuit. The concerned eigenvalue λ is represented by the conductance g_λ of the feedback resistors, which connect the output and the inverting input terminals of OAs. The circuit is described by the matrix equation:

$$\boldsymbol{U}_{\lambda 1}^{-1} \boldsymbol{G}_1 \boldsymbol{V}_{\boldsymbol{x}} = \boldsymbol{U}_{\lambda 2}^{-1} \left(\boldsymbol{G}_2 \boldsymbol{V}_{\boldsymbol{x}} + g_\lambda \boldsymbol{V}_{\boldsymbol{x}} \right), \tag{8}$$

where, this time, the diagonal matrices $\boldsymbol{U}_{\lambda 1}$ and $\boldsymbol{U}_{\lambda 2}$ are defined as $\boldsymbol{U}_{\lambda 1} = \mathrm{diag}(\sum_j \boldsymbol{G}_{1,kj} + g_{1k}, k = 1, 2, \ldots, n)$, $\boldsymbol{U}_{\lambda 2} = \mathrm{diag}(\sum_j \boldsymbol{G}_{2,kj} + g_{2k} + g_\lambda, k = 1, 2, \ldots, n)$. Similar to (6), by selecting compensation

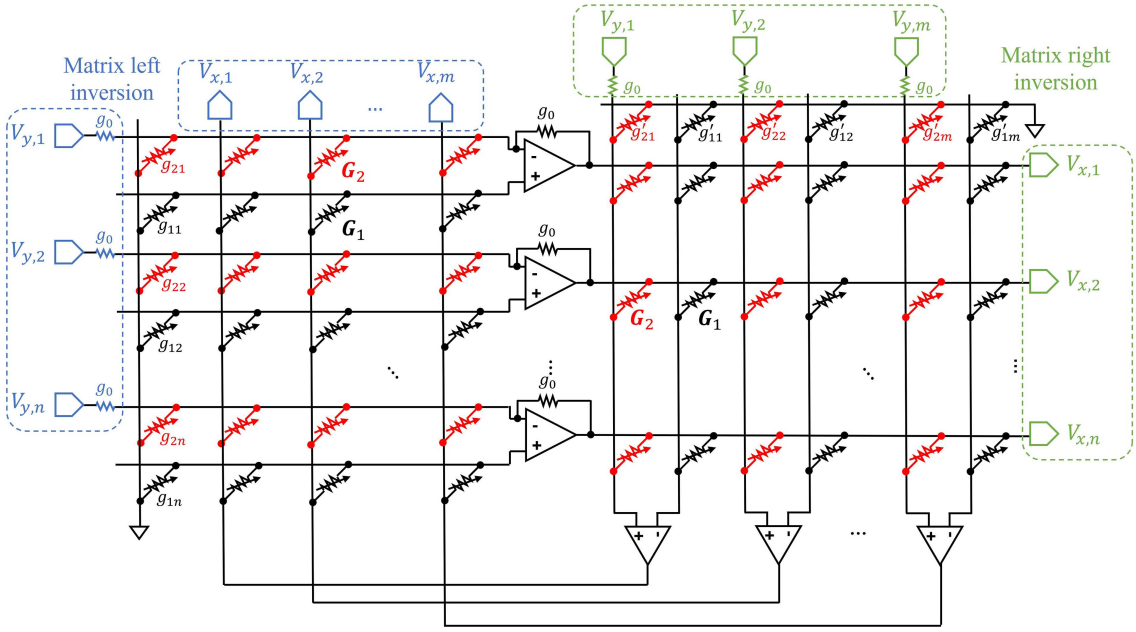

Figure 3 (Color online) CC-AMC generalized inverse circuit. For left (right) inverse operations, the left (right) input/output interfaces are utilized to connect to the left (right) crosspoint RRAM array. The leftmost column in the left RRAM array, and the upmost row in the right RRAM array are for conductance compensation. Furthermore, the conductance matrices $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ in the RRAM arrays represent matrix $\boldsymbol{A}$ for left inverse computations, while their transposed versions, $\boldsymbol{G}_1^{\mathrm{T}}$ and $\boldsymbol{G}_2^{\mathrm{T}}$, represent $\boldsymbol{A}^{\mathrm{T}}$ for right inverse computations.

conductances that are subject to $g_{2k} - g_{1k} = \sum_j \boldsymbol{G}_{kj} - g_\lambda$ for each pair of rows of resistive devices, Eq. (8) turns out to be

$$(\boldsymbol{G}_1 - \boldsymbol{G}_2)\,\boldsymbol{V}_{\boldsymbol{x}} = g_\lambda \boldsymbol{V}_{\boldsymbol{x}}, \tag{9}$$

$$\text{i.e.,}\ \boldsymbol{A}\boldsymbol{V}_{\boldsymbol{x}} = \lambda \boldsymbol{V}_{\boldsymbol{x}},$$

which is analog to (2). Since there is no external input, the circuit works by amplifying the signal from the noise level according to the eigenvector solution, and eventually limited by the maximal available voltage in the circuit.

3.3 CC-based generalized inverse circuit

Figure 3 illustrates the CC-based generalized inverse circuit, which consists of two crosspoint RRAM arrays, one set of TIAs, and one set of OAs that form global feedback loops. It is equipped with two sets of interfaces for the left inverse and right inverse operations, respectively. In the former case, the input voltages are applied to the rows of the left crosspoint RRAM array, the output voltages are read from the left columns, and the interfaces to the right array stay floating. In the latter case, by contrast, the right interfaces are used for input and output, and the left ones are floating.

Consider the matrix mapping for the left inverse operation. The original $n \times m$ matrix $\boldsymbol{A}$ is mapped as $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ of the two crosspoint RRAM arrays. While the left array is row-wise split as in the former two CC-AMC circuits, the right array is column-wise split due to the feedback loop configuration. Accordingly, a column of RRAM devices for CC is included in the left array, while a row of CC devices is designed in the right array. The latter requires additional calculations of column sums of matrix $\boldsymbol{A}$ to determine the compensation conductances. In the left array, rows of $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ are connected to the non-inverting and inverting input terminals of OAs, respectively. In the right array, the opposite is the case. With this design, the global negative feedback of the circuit is achieved to ensure the stability [13]. The feedback resistors with conductance g_0 connect the inverting inputs and the outputs of OAs to convert the resulting currents in the left array into voltages applied to the right array.

The equality of voltages of two sets of rows in the left array leads to the following matrix equation:

$$\boldsymbol{U}_{L1}^{-1}\boldsymbol{G}_1\boldsymbol{V}_{\boldsymbol{x}} = \boldsymbol{U}_{L2}^{-1}\left(\boldsymbol{G}_2\boldsymbol{V}_{\boldsymbol{x}} + g_0\boldsymbol{V}_{\boldsymbol{y}} + g_0\boldsymbol{V}_{\boldsymbol{r}}\right), \tag{10}$$

where $\boldsymbol{V_r} = [V_{r,1}, V_{r,2}, \ldots, V_{r,n}]^{\mathrm{T}}$ is the row voltages vector of right array, $\boldsymbol{U}_{L1}$ and $\boldsymbol{U}_{L2}$ are defined as $\boldsymbol{U}_{L1} = \mathrm{diag}(\sum_j \boldsymbol{G}_{1,kj} + g_{1k}, k = 1, 2, \ldots, n)$, $\boldsymbol{U}_{L2} = \mathrm{diag}(\sum_j \boldsymbol{G}_{2,kj} + g_{2k} + 2g_0, k = 1, 2, \ldots, n)$.

Similarly, the equality of voltages on the columns in the right array is described by

$$\boldsymbol{U}_{R1}^{-1}\boldsymbol{G}_1^{\mathrm{T}}\boldsymbol{V_r} = \boldsymbol{U}_{R2}^{-1}\boldsymbol{G}_2^{\mathrm{T}}\boldsymbol{V_r}, \tag{11}$$

where $\boldsymbol{U}_{R1}$ and $\boldsymbol{U}_{R2}$ are defined as $\boldsymbol{U}_{R1} = \mathrm{diag}(\sum_j \boldsymbol{G}_{1,kj}^{\mathrm{T}} + g'_{1k}, k = 1, 2, \ldots, m)$, $\boldsymbol{U}_{R2} = \mathrm{diag}(\sum_j \boldsymbol{G}_{2,kj}^{\mathrm{T}} + g'_{2k}, k = 1, 2, \ldots, m)$. By selecting compensation conductances according to $g_{2k} - g_{1k} = \sum_j \boldsymbol{G}_{kj} - 2g_0$ and $g'_{2k} - g'_{1k} = \boldsymbol{G}_{kj}^{\mathrm{T}}$, Eqs. (10) and (11) are simplified respectively as follows:

$$\boldsymbol{G}_1\boldsymbol{V_x} = \boldsymbol{G}_2\boldsymbol{V_x} + g_0\boldsymbol{V_y} + g_0\boldsymbol{V_r}, \tag{12}$$

$$\boldsymbol{G}_1^{\mathrm{T}}\boldsymbol{V_r} = \boldsymbol{G}_2^{\mathrm{T}}\boldsymbol{V_r}. \tag{13}$$

By combining (12) and (13), the circuit turns out to be solving the following equation:

$$\left(\boldsymbol{G}_1^{\mathrm{T}} - \boldsymbol{G}_2^{\mathrm{T}}\right) \cdot \left(\frac{\boldsymbol{G}_1 - \boldsymbol{G}_2}{g_0}\boldsymbol{V_x} - \boldsymbol{V_y}\right) = \boldsymbol{0}, \tag{14}$$

$$\text{i.e., } \boldsymbol{A}^{\mathrm{T}} \cdot (\boldsymbol{A}\boldsymbol{V_x} - \boldsymbol{V_y}) = \boldsymbol{0},$$

whose solution is exactly the left inverse result, namely $\boldsymbol{V_x} = \left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\right)^{-1} \cdot \boldsymbol{A}^{\mathrm{T}}\boldsymbol{V_y}$.

For the right inverse operation of a broad matrix, rather the transpose of $\boldsymbol{A}$ is mapped in the two crosspoint arrays, resulting in the same situation as the left inverse, i.e., more rows than columns. The calculations of row sums and column sums are demanded as well, but are applied to the right array and left array, respectively. By analyzing the equalities of voltages on the rows in the left array and those on the columns in the right array, two matrix equations are obtained, namely,

$$g_0\boldsymbol{V_x} + \boldsymbol{G}_2^{\mathrm{T}}\boldsymbol{V_r} = \boldsymbol{G}_1^{\mathrm{T}}\boldsymbol{V_r}, \tag{15}$$

$$g_0\boldsymbol{V_y} + \boldsymbol{G}_2\boldsymbol{V_x} = \boldsymbol{G}_1\boldsymbol{V_x}, \tag{16}$$

whose solution corresponds to the right inverse result faithfully,

$$\boldsymbol{V_x} = \left(\boldsymbol{G}_1^{\mathrm{T}} - \boldsymbol{G}_2^{\mathrm{T}}\right)\boldsymbol{V_r} = \left(\boldsymbol{G}_1^{\mathrm{T}} - \boldsymbol{G}_2^{\mathrm{T}}\right)\left[(\boldsymbol{G}_1 - \boldsymbol{G}_2) \cdot \left(\boldsymbol{G}_1^{\mathrm{T}} - \boldsymbol{G}_2^{\mathrm{T}}\right)\right]^{-1}\boldsymbol{V_y}, \tag{17}$$

$$\text{i.e., } \boldsymbol{V_x} = \boldsymbol{A}^{\mathrm{T}} \cdot \left(\boldsymbol{A} \cdot \boldsymbol{A}^{\mathrm{T}}\right)^{-1}\boldsymbol{V_y}.$$

By using the CC strategy, the generalized inverse operations are enabled with only $n + m$ OAs. Otherwise, the count of OAs would be doubled in the CS-AMC circuits, causing prohibitively high circuit area and power consumption. The CC-AMC circuits overcome this issue by taking full advantage of the differential inputs that is inherent of OAs. Also, since there is no virtual ground in the circuit, rather the voltages of all rows and columns are averaged non-zero results according to the KCL, the currents through the RRAM devices in the circuit are reduced, contributing to the enhanced energy efficiency of AMC. Notably, compared to the CS-AMC circuits and the positive-matrix AMC circuits, the CC-based differential input approach offers one more set of connection ports for feeding the input vector, thus eliminating the preliminary sign change of it.

3.4 OA design and circuit demonstrations

For CC-AMC circuit demonstration, an OA has been designed with a commercial 40 nm CMOS technology process. An RRAM device occupies a square of 130 nm×130 nm. To adapt to the RRAM array structure with small row-to-row spaces, the OA layout is designed with a high aspect ratio, as shown in Figure 4(a). To achieve a sufficient open-loop gain, which would help improve the computing accuracy of the circuit, the two-stage telescopic OA structure is adopted. Between the two stages, a pair of Miller resistance and capacitance is added to eliminate the secondary pole, which would make the OA faster and more stable. According to the Bode plot results in Figure 4(b), the open-loop gain of the OA is 65.26 dB, the phase margin is 87.4°, and the limiting frequency, f_p, is 708 MHz.

Simulation results of CC-AMC circuits in HSPICE D-2010.03-SP1 are shown in Figures 5(a)–(d), based on ideal situations with precise matrix mapping and without circuit non-idealities. In this case, all the

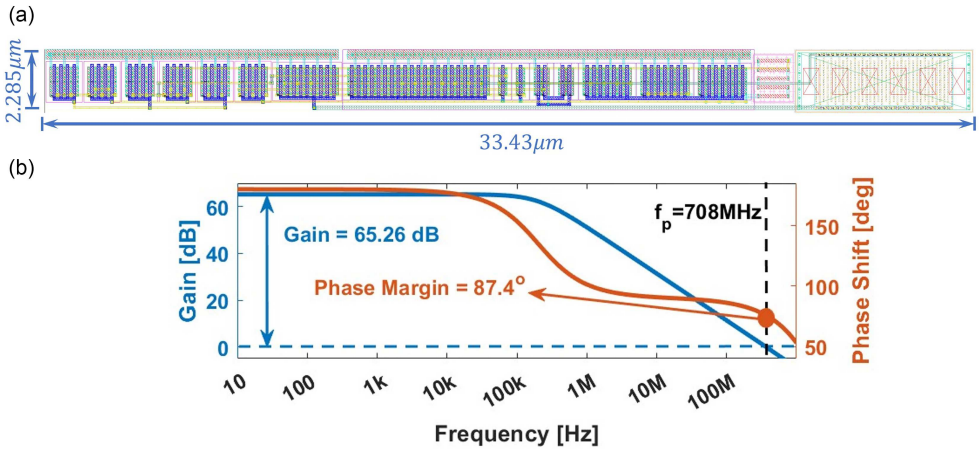

Figure 4 (Color online) (a) Layout of the OA used in this work; (b) bode plot of the OA.

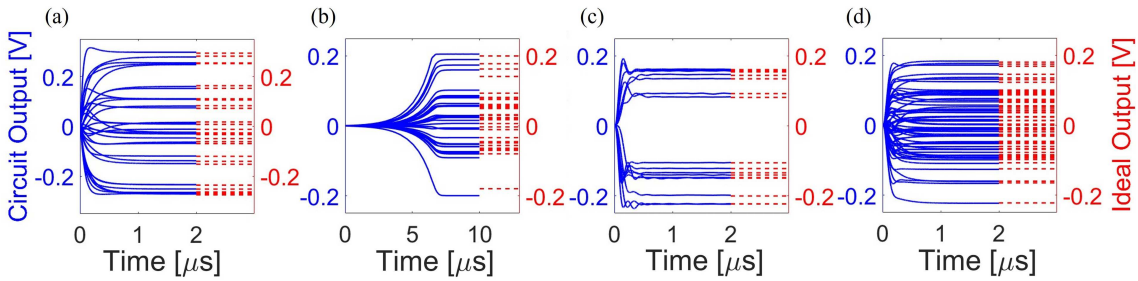

Figure 5 (Color online) Comparisons between CC-AMC outputs and ground-truth results, for (a) matrix inversion circuit of a 32×32 matrix, (b) eigenvector circuit of a 32×32 matrix, (c) left inverse circuit of a 64×16 matrix, and (d) right inverse circuit of a 16×64 matrix.

four CC-AMC circuits show high computing accuracy in comparison with the ground-truth solutions. Notably, although the matrix in simulation is considerably large, e.g., 32×32, all the circuits stabilize within only a few microseconds, demonstrating AMC again as a highly promising approach for matrix accelerations. In spite of their distinct computational functions, the CC-AMC circuits share a considerable degree of similarity. In the remainder of this work, the performance analysis will be concentrated on the matrix inversion circuit, and similar conclusion should be drawn for other ones.

3.5 Area and power consumption

To evaluate the area and power performances of the matrix inversion circuit, matrices with various sizes, ranging from 16×16 to 256×256, have been generated and mapped in the simulations. During matrix mapping, the maximal and the minimal conductances ($G_{\max}$ and $G_{\min}$) are reasonably assumed as 100 and 1 μS, respectively [31]. For the evaluation, the layout areas and power consumptions of both RRAM devices and OAs are included. The results of CS-AMC and CC-AMC circuits in comparison are summarized in Figures 6(a) and (b). For both metrics, the performances in CC-AMC are significantly improved. Specifically, the area of the CC-AMC circuit is reduced by 8.9%–37.9% compared to the CS-based counterpart. Since the power dissipated in RRAM devices is several orders of magnitude lower than that in OAs, the CC-AMC circuit consumes nearly half of the total power of the CS-AMC circuit, thanks to the removal of analog inverters in the new design.

3.6 Stability issue

To access the stability issue of the CC-AMC circuit, its transfer function is analyzed. In the matrix inversion circuit, the input-output relationship of the OAs in terms of the Laplace transform is characterized as

$$\left[\boldsymbol{U}_2^{-1}\left(\boldsymbol{G}_2\boldsymbol{V}_{\boldsymbol{x}}\left(s\right)+\boldsymbol{V}_{\boldsymbol{y}}\left(s\right)\right)-\boldsymbol{U}_1^{-1}\boldsymbol{G}_1\boldsymbol{V}_{\boldsymbol{x}}\left(s\right)\right]L\left(s\right)=\boldsymbol{V}_{\boldsymbol{x}}\left(s\right), \tag{18}$$

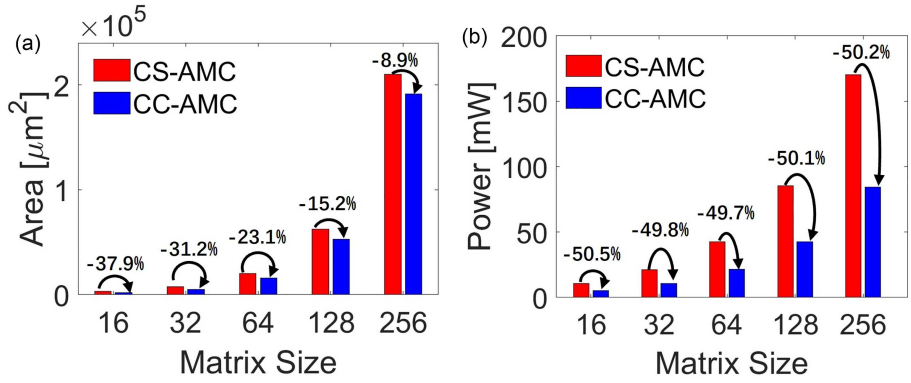

Figure 6 (Color online) (a) Areas and (b) power consumptions of CS-AMC and CC-AMC matrix inversion circuits of different sizes.

where s is the complex frequency in Laplace transform and $L(s)$ is the open-loop gain of OA. Considering a single-pole OA model, namely $L(s) = L_0/(1 + (s/\omega_0))$, where L_0 is the DC open-loop gain and ω_0 is the 3-dB bandwidth, Eq. (18) results in

$$s\boldsymbol{V_x}(s) = L_0\omega_0\left[\boldsymbol{U_2^{-1}V_y}(s) - \left(\boldsymbol{U_1^{-1}G_1} - \boldsymbol{U_2^{-1}G_2} + \frac{\boldsymbol{I}}{L_0}\right)\boldsymbol{V_x}(s)\right], \tag{19}$$

where $\boldsymbol{I}$ is the $n \times n$ identity matrix. Since L_0 is usually much larger than 1, the small term in $\boldsymbol{I}/L_0$ could be omitted. Consequently, Eq. (19) is converted into time domain inverse Laplace transform, obtaining

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{V_x}(t) = L_0\omega_0\left[\boldsymbol{U_2^{-1}V_y}(t) - \left(\boldsymbol{U_1^{-1}G_1} - \boldsymbol{U_2^{-1}G_2}\right)\boldsymbol{V_x}(t)\right]. \tag{20}$$

In the circuit, $\boldsymbol{V_y}$ is a constant input vector that is initiated by a step function. By using the finite difference method, Eq. (20) could be approximated by an iterative process, namely,

$$\boldsymbol{V_x}(t + \Delta t) = \alpha\boldsymbol{U_2^{-1}V_y} + (\boldsymbol{I} - \alpha\boldsymbol{M})\,\boldsymbol{V_x}(t), \tag{21}$$

where matrix $\boldsymbol{M}$ is defined as $\boldsymbol{M} = \boldsymbol{U_1^{-1}G_1} - \boldsymbol{U_2^{-1}G_2}$, $\alpha = L_0\omega_0\Delta t$ is a small dimensionless coefficient. In (21), if the spectral radius of $\boldsymbol{I} - \alpha\boldsymbol{M}$ is less than 1, the convergence of the circuit response is guaranteed. Such a condition can be conveniently met by assuming the positive definiteness of matrix $\boldsymbol{A}$. Therefore, for a positive definite linear system, which is a common case in real-world applications, the CC-AMC matrix inversion circuit should be stable, which is similar to the positive-matrix AMC circuit [32]. Particularly, the minimal eigenvalue of $\boldsymbol{M}$, which corresponds to the dominant pole of the circuit, determines the elapsed time towards the steady state. In fact, Eq. (20) is formally the same as its counterpart in the positive-matrix case, both featured by a simple first-order matrix differential equation. By contrast, in the CS-AMC matrix inversion circuit, the inclusion of analog inverters results in a second-order matrix differential equation, which inevitably causes a shift of the dominant pole, prolonging the response time and increasing the circuit sensitivity to noises [15]. Figures 7(a) and (b) compare the transient behaviors of the CS-AMC and CC-AMC matrix inversion circuits, demonstrating a faster response of the latter than the former. The elapsed time is evaluated according to when the output voltages are within the 1 mV margin of the steady states. Note that the elapsed time of the two AMC circuits is closely related to the gain bandwidth product of OA, the optimization of which would result in a faster speed of about 20 ns [33]. However, it might trade off the computing accuracy resulting from the limited open-loop gain of OA.

The benefits of the CC-AMC circuit come at the cost of the calculation and implementation of the compensation conductances, which demand an elaborated controller module for carrying out these operations. Notably, calculations of row sums and column sums of the split matrices can also leverage the natural summation capability of the crosspoint RRAM array. By simply applying all identical input voltages to the rows or columns, the collected currents on the columns or rows would then represent the summation results, according to which the compensation conductances can be determined, as illustrated in Figure 8. In some applications, due to the special matrix structure, e.g., DFT/IDFT matrix, the prior knowledge of row/column sums may be easily known, which leaves out the requirement of pre-calculations and relieves the system design [30].

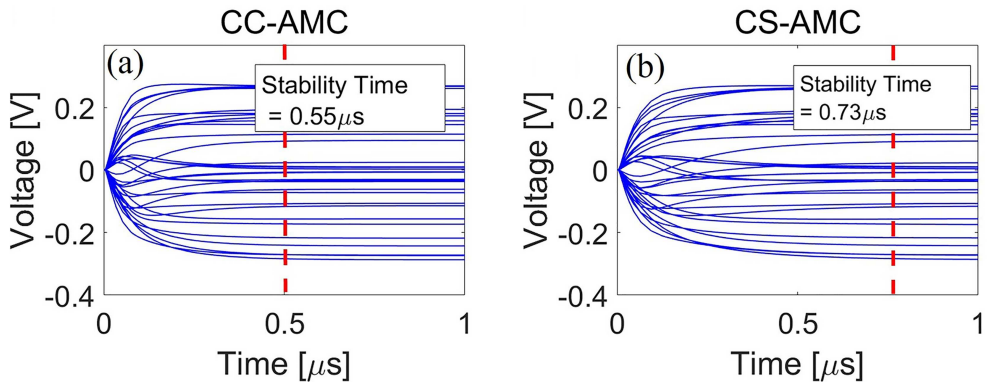

Figure 7 (Color online) Transient simulation results of the two circuits for solving (1) of a 32×32 matrix. (a) CC-AMC; (b) CS-AMC.

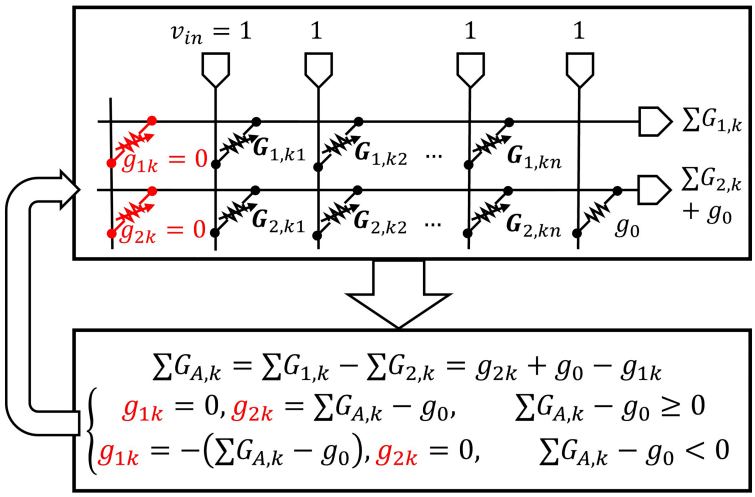

Figure 8 (Color online) CC estimation strategy with the RRAM array.

4 Performance analysis

In AMC, a critical concern lies in the limited computing accuracy caused by the non-ideal factors related to the RRAM devices or the circuit interconnects. Specifically, RRAM devices always show conductance variations resulting from the initial imprecise programming and the temporal state relaxation [34–43]. The interconnect non-idealities include the parasitic resistors and capacitors along crosspoint rows and columns, and source and sink resistors at the interfaces. Extracted from the circuit layout, the parasitic resistance of an interconnect segment between every two rows or columns is about 1 Ω, the parasitic capacitance is 0.1 fF, and the interconnect resistances at the interfaces are around 50 Ω. To assess the computing accuracy performance of the CS-AMC and CC-AMC circuits, we have carried out extensive pre-layout simulations of the matrix inversion circuits including all non-idealities.

Since the condition number κ plays a crucial role in affecting the computing accuracy of matrix inversion [44], a range of matrices with various κ values should be covered in the simulations. Also, as the circuits are committed to solving positive definite linear systems [32], we proposed to generate model Gram matrices with different sizes, based on which the circuit simulations were performed. Specifically, an $m \times n$ $(m > n)$ matrix $\boldsymbol{H}$ is randomly generated, with entries following Gaussian distribution with zero mean and unit variance. Then, an $n \times n$ Gram matrix $\boldsymbol{A}$ is obtained by $\boldsymbol{A} = \boldsymbol{H}^{\mathrm{T}} \cdot \boldsymbol{H}$. Theory reveals that a larger number m of matrix $\boldsymbol{H}$ generally makes matrix $\boldsymbol{A}$ better conditioned (κ is reduced). Five different sizes of matrix $\boldsymbol{A}$ are considered, ranging from 16×16, 32×32, 64×64, 128×128 to 256×256. For each matrix scale, three sets of matrices with different ranges of condition numbers, namely κ_l, κ_m, and κ_s (representing large, middle, and small κ values), are generated, by assuming $m = 2n$, $m = 4n$, and $m = 8n$ for matrix $\boldsymbol{H}$, respectively. 500 matrices were randomly generated for each set. Figure 9 shows the case of 32×32 Gram matrix, including one representative matrix $\boldsymbol{H}$ and the resulting $\boldsymbol{A}$ for

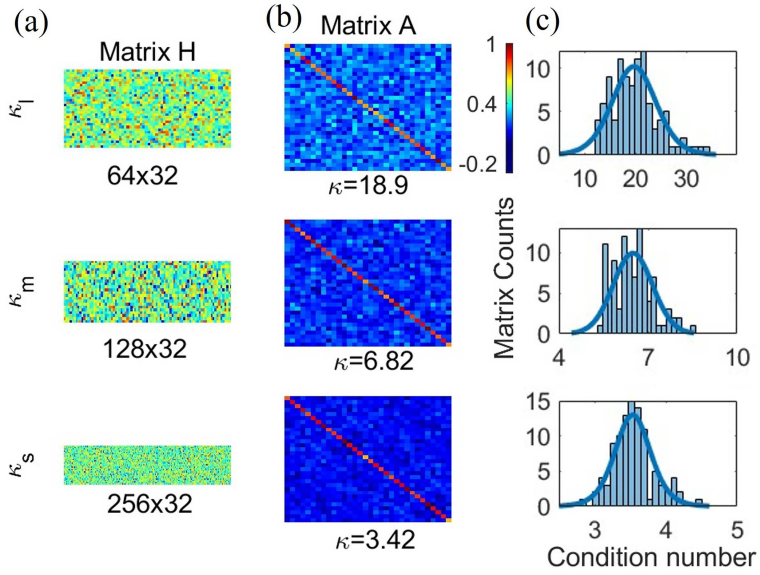

Figure 9 (Color online) Randomly generated 32×32 matrices with different condition numbers for AMC matrix inversion circuits simulation. (a) Matrix $\boldsymbol{H}$ examples featuring $m = 2n$, $m = 4n$, and $m = 8n$ in size; (b) the resulting Gram matrices from (a), featuring a large, middle, or small condition number, respectively; (c) condition number distributions of the three sets of 32×32 matrix.

Table 1 Average condition number for different sets.

Size	Average κ value		
	κ_l	κ_m	κ_s
16×16	14.81	5.53	3.11
32×32	19.92	6.49	3.56
64×64	23.16	7.42	3.85
128×128	27.15	7.88	4.03
256×256	29.23	8.28	4.16

each range of κ, as well as the distributions of κ_l, κ_m, and κ_s. The average κ values of different sets for matrices with different sizes are shown in Table 1.

4.1 Device variation effect analysis

To evaluate the impact of device variation, each matrix element is mapped to a device conductance that follows Gaussian distribution with a standard deviation of $2\%G_{\max}$, which is achievable by using a write&verify algorithm [45,46]. Figure 10(a) shows the statistics of the mapping results for a 32×32 real-valued matrix. Based on the distorted conductances, the CS-AMC and CC-AMC matrix inversion circuits were simulated. The results are presented in Figure 10(b), both in comparison with the ideal results. The relative error ε_r is defined as $\varepsilon_r = |\boldsymbol{v}_{\text{out}} - \boldsymbol{v}_{\text{out,ideal}}| \,/\, |\boldsymbol{v}_{\text{out,ideal}}|$, where $\boldsymbol{v}_{\text{out}}$ is the output voltage vector detected in the simulation circuit, $\boldsymbol{v}_{\text{out,ideal}}$ is the ideal result from the ideal matrix, $|\cdot|$ represents the Euclidean norm of a vector. As a result, CS-AMC and CC-AMC circuits show an ε_r of 0.23778 and 0.23264, respectively, which are fairly close. Alternatively, the simulated output vector $\boldsymbol{v}_{\text{out}}$ can be fitted using the linear model $\boldsymbol{v}_{\text{out}} = b + k \cdot \boldsymbol{v}_{\text{out,ideal}}$. The closer b is to 0 and k is to 1, the more accurate the result is. The correlation coefficient r in this model represents the confidence of the analysis. In this case, the k, b, and r of the CS-AMC circuit are 0.991, -0.008, and 0.9435, respectively. Those of the CC-AMC circuit are 0.995, -0.009, and 0.9466, respectively.

We have conducted systematic simulations of the two circuits with the matrices with different sizes and condition numbers, where each matrix is accompanied by a randomly-generated input vector for the solution of (1). The statistical results are summarized in Figure 11(a). It is observed that, both circuits show nearly the same accuracy behaviors in terms of relative error in the relatively small κ range, namely κ_m and κ_s. Although the extra column of CC brings about more device variation in CC-AMC circuits, the absence of analog inverters also reduces the source of computing error, thus resulting in similar relative

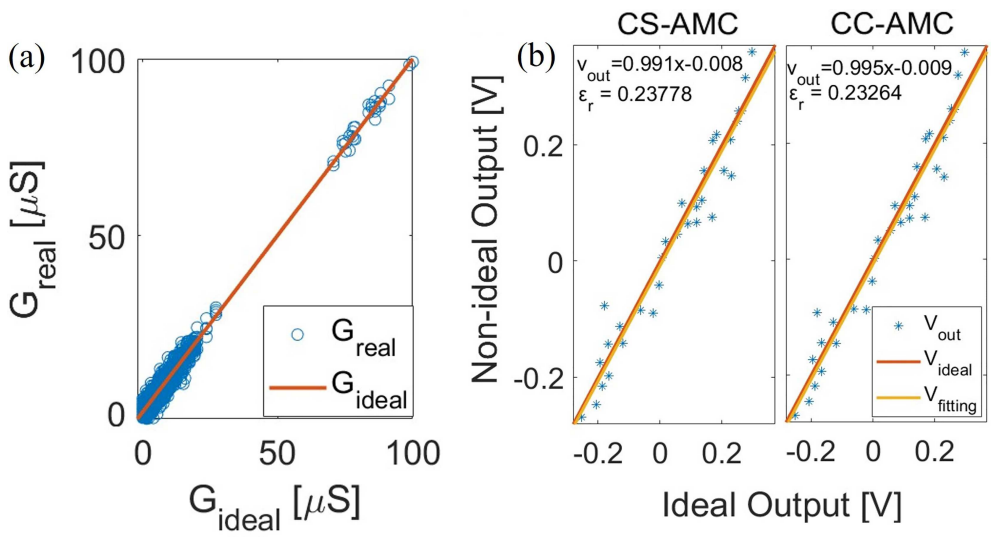

Figure 10 (Color online) (a) Mapping result of a 32×32 matrix with conductance variations, whose standard deviation is $2\%G_{\max}$; (b) simulation results of CS-AMC and CC-AMC matrix inversion circuits with the 32×32 matrix, in comparison with the ideal output voltages. The data fitting results are also included.

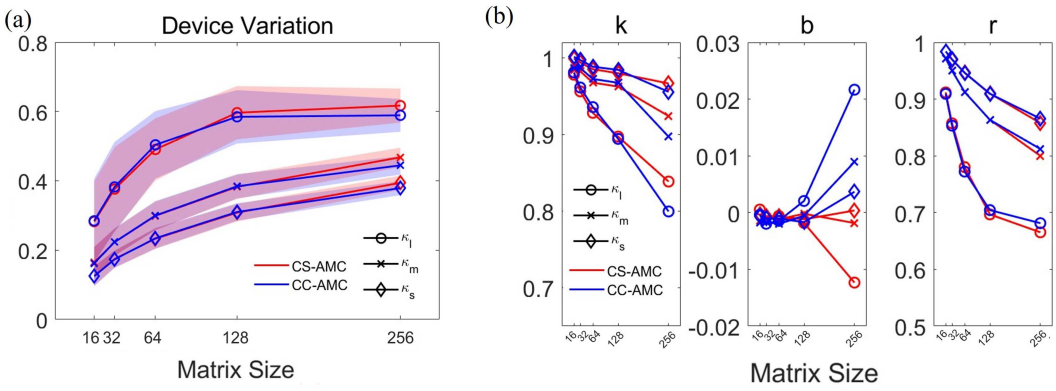

Figure 11 (Color online) (a) Computing accuracy comparison between CS-AMC and CC-AMC circuits with device variations. For each matrix condition (n and κ), 100 simulations were conducted, and the relative error distribution is shown. (b) Data fitting analysis of the two circuits. k, b, and r are the intercept, slope, and correlation coefficient in the linear regression model.

error. In the cases of three sets of 256×256 matrix, the CC-AMC circuit even demonstrates a better accuracy than the CS-AMC circuit. Only in the large κ range of 32×32 and 64×64 matrix sets, the CC-AMC shows a slight accuracy degradation, as the inequality of conductance summation with CC contributes another error source for such relatively ill-conditioned matrices.

In Figure 11(b), the data fitting results are summarized, to evaluate the error model of the two circuits. Regarding the fitting slope parameter k, the CC-AMC circuit shows values closer to 1, except for the 256×256 matrix sets. Regarding the fitting offset parameter b, for small condition number κ_s, both circuits are close to 0 regardless of matrix size. As for larger condition numbers, κ_m and κ_l, the CC-AMC circuit shows positive values relative to the CS-AMC circuit, and the trend is enhanced as the matrix scale increases. However, the correlation coefficient r of the CC-AMC circuit shows an improvement for the 256×256 matrix sets, and the combination of all three parameters results in the accuracy performances of two circuits in Figure 11(a).

4.2 Interconnect resistor issue analysis

In AMC circuits, interconnect resistances within the arrays and interfaces pose a significant challenge. Owing to the distinct topologies of the CS-AMC and CC-AMC circuits, the distributions of interconnect resistors differ in the arrays. In the CS-AMC matrix inversion circuit, there is one segment of interconnect resistance lying between every two rows along one column. Additionally, two such segments are

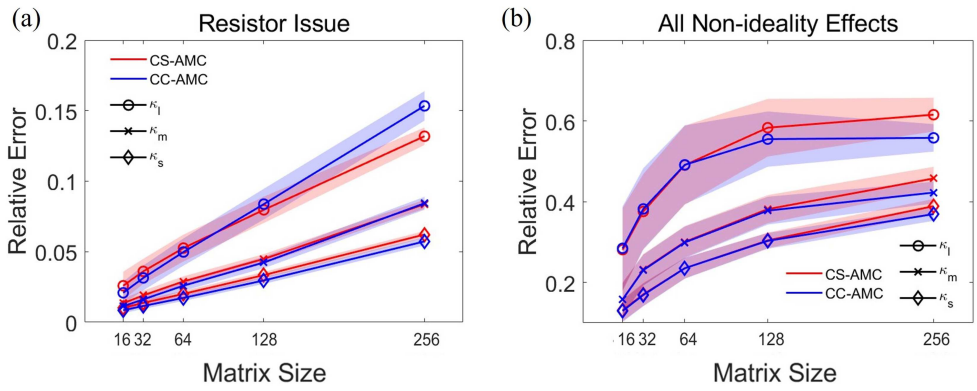

Figure 12 (Color online) Computing accuracy comparison between CS-AMC and CC-AMC circuits (a) with interconnect resistances and (b) with all non-ideal factors.

found between every two columns along one row, suggesting that the interconnect resistors along rows have a more significant impact. Conversely, the CC-AMC circuit, which is based on row-wise splitting, experiences the opposite effect: the interconnect resistors along columns are more impactful. To assess the performance degradation caused by these interconnect resistances as well as the source resistors at the row input interface and the feedback connection resistors to the column output in the two circuits, we carried out extensive circuit simulations, considering device variation analysis. The simulation results, as presented in Figure 12(a), reveal that the CC-AMC circuit exhibits a lower relative error compared to the CS-AMC circuit across various condition numbers κ_s, κ_m, and κ_l for small matrix sizes. However, for larger matrices, specifically those sized 128×128 and 256×256, the CC-AMC circuit shows accuracy degradation. This phenomenon can be attributed to the more pronounced role of row interconnect resistances in affecting computing accuracy within the CS-AMC circuit, as they disrupt the virtual ground condition on the rows. Conversely, the larger column interconnect resistances in the CC-AMC circuit cause a smaller significant error. Another factor contributing to the lower relative error in the CC-AMC circuit is the reduced currents flowing through the crosspoint RRAM devices. According to (5), the voltages at the rows connected to the inverting/non-inverting input terminals of an OA represent average voltages. These averages are weighted by the device conductance values through KCL, rather than being zero (virtual ground) as seen in the CS-AMC circuit. This average effect decreases the voltage drops across the devices and consequently reduces the currents, mitigating the IR drop caused by the interconnect resistances. Such a factor also contributes to the reduction in power consumption of the CC-AMC circuit.

To assess the overall performance of the two circuits, we incorporated all relevant device and circuit non-idealities into our simulations. These simulations were based on the previously mentioned matrix sets, as shown in Figure 12(b). Overall, the CC-AMC circuit exhibited nearly the same accuracy compared with the CS-AMC circuit owing to the dominant role of conductance variation affecting the relative error. This observation underscores the importance of focusing on improvements in RRAM device performance and programming strategies to enhance the accuracy of AMC circuits. The better performance of the CC-AMC circuit with 256×256 matrix sets highlights the potential for large-scale applications. While it is evident that the AMC results may not always provide a solution with the desired level of precision, they serve an invaluable role as an initial seed. By offering a starting point for high-precision solutions in the digital paradigm, these AMC results significantly expedite the iterative convergence of the algorithms.

4.3 Time complexity

Given that parasitic capacitors lying on the interconnect wires and RRAM devices influence the elapsed time, they have also been included in the simulation for a comprehensive evaluation. The result is shown in Figure 13. As anticipated, the CC-AMC circuit, which operates without the use of analog inverters, showcases a notable improvement in elapsed time when compared to the CS-AMC circuit. This comparison is based on the evaluation method in Figures 7(a) and (b). Therefore, these results provide compelling evidence that the CC-AMC circuit does not compromise on accuracy performance while offering significant advantages. Specifically, this circuit design benefits from a smaller area, lower-power consumption, and faster response times, all by eliminating the need for analog inverters.

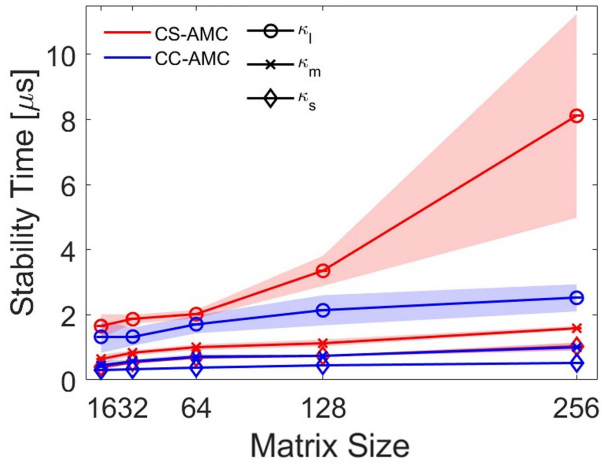

Figure 13 (Color online) Elapsed time comparison between CS-AMC and CC-AMC circuits with all non-ideal resistive and capacitive factors.

5 Conclusion

In this work, we have presented the design and operational principles of CC-AMC circuits for processing real-valued matrices. This includes operations such as matrix inversion, eigenvector calculation, and computing generalized inverses. A key innovation in our approach is the elimination of analog inverters, which traditionally consume significant space and power. Instead, we leverage the intrinsic differential properties of OAs, enhanced by a CC strategy. This results in more compact, efficient, and faster circuits. Based on the specially designed OA, the CC-AMC circuits achieve significant savings in circuit area, power dissipation, and elapsed time compared to the CS-AMC circuits. Our simulations, which involved randomly-generated matrices of varying sizes and condition numbers, demonstrate that the CC-AMC circuits can achieve comparable and in some cases superior-computing accuracy. This is impressive given the presence of various non-ideal factors, such as device conductance variations, parasitic resistances, capacitances in the array, and source and sink resistances at circuit interfaces. The introduction of the CC-based concept paves the way toward compact and efficient AMC circuit designs, especially for solving a broad spectrum of problems involving both positive and arbitrarily real-valued matrices.

Acknowledgements This work was supported by National Key R&D Program of China (Grant No. 2020YFB2206001), National Natural Science Foundation of China (Grant Nos. 62004002, 92064004, 61927901), and 111 Project (Grant No. B18001).

References

1 Saad Y. Iterative Methods for Sparse Linear Systems. 2nd ed. Philadelphia: Society for Industrial and Applied Mathematics, 2003. 4: 105–132
2 Golub G H, Loan C F V. Matrix Computations. 4th ed. Baltimore City: Johns Hopkins University Press, 2013. 3: 105–151
3 Zou X Q, Xu S, Chen X M, et al. Breaking the von Neumann bottleneck: architecture-level processing-in-memory technology. Sci China Inf Sci, 2021, 64: 160404
4 Dhar P. The carbon impact of artificial intelligence. Nat Mach Intell, 2020, 2: 423–425
5 Rusek F, Persson D, Lau B K, et al. Scaling up MIMO: opportunities and challenges with very large arrays. IEEE Signal Process Mag, 2013, 30: 40–60
6 Waldrop M M. The chips are down for Moore's law. Nature, 2016, 530: 144–147
7 Ielmini D, Wong H S P. In-memory computing with resistive switching devices. Nat Electron, 2018, 1: 333–343
8 Hu M, Strachan J P, Li Z, et al. Dot-product engine for neuromorphic computing: programming 1T1M crossbar to accelerate matrix-vector multiplication. In: Proceedings of the 53rd ACM/EDAC/IEEE Design Automation Conference (DAC), Austin, 2016. 1–6
9 Jiang Y N, Huang P, Zhou Z, et al. Circuit design of RRAM-based neuromorphic hardware systems for classification and modified Hebbian learning. Sci China Inf Sci, 2019, 62: 062408
10 Hasler J, Natarajan A. Continuous-time, configurable analog linear system solutions with transconductance amplifiers. IEEE Trans Circuits Syst I, 2021, 68: 765–775
11 Sun Z, Pedretti G, Ambrosi E, et al. Solving matrix equations in one step with cross-point resistive arrays. Proc Natl Acad Sci USA, 2019, 116: 4123–4128
12 Sun Z, Pedretti G, Bricalli A, et al. One-step regression and classification with cross-point resistive memory arrays. Sci Adv, 2020, 6: 2378
13 Sun Z, Ambrosi E, Pedretti G, et al. In-memory pagerank accelerator with a cross-point array of resistive memories. IEEE Trans Electron Devices, 2020, 67: 1466–1470
14 Sun Z, Pedretti G, Ambrosi E, et al. In-memory eigenvector computation in time $O(1)$. Adv Intelligent Syst, 2020, 2: 2000042
15 Wang S, Sun Z, Liu Y, et al. Optimization schemes for in-memory linear regression circuit with memristor arrays. IEEE Trans Circ Syst I, 2021, 68: 4900–4909
16 Li J C, Zhou H J, Li Y, et al. A memristive neural network based matrix equation solver with high versatility and high energy efficiency. Sci China Inf Sci, 2023, 66: 122402

17 Song T, Chen X, Han Y. Eliminating iterations of iterative methods: solving large-scale sparse linear system in $O(1)$ with RRAM-based in-memory accelerator. In: Proceedings of the Great Lakes Symposium on VLSI (GLSVLSI), New York, 2021. 71–76
18 Feinberg B, Wong R M, Xiao T P, et al. An analog preconditioner for solving linear systems. In: Proceedings of IEEE International Symposium on High-Performance Computer Architecture (HPCA), Seoul, 2021. 761–774
19 Zhou H J, Li Y, Miao X S. Low-time-complexity document clustering using memristive dot product engine. Sci China Inf Sci, 2022, 65: 122410
20 Sebastian A, Le Gallo M, Khaddam-Aljameh R, et al. Memory devices and applications for in-memory computing. Nat Nanotechnol, 2022, 15: 529–544
21 Ielmini D. Resistive switching memories based on metal oxides: mechanisms, reliability and scaling. Semicond Sci Technol, 2016, 31: 063002
22 Sun Z, Ielmini D. Invited tutorial: analog matrix computing with crosspoint resistive memory arrays. IEEE Trans Circ Syst II, 2022, 69: 3024–3029
23 Li C, Belkin D, Li Y, et al. Efficient and self-adaptive in-situ learning in multilayer memristor neural networks. Nat Commun, 2018, 9: 2385
24 Li C, Belkin D, Li Y, et al. In-memory computing with memristor arrays. In: Proceedings of IEEE International Memory Workshop (IMW), Kyoto, 2018. 1–4
25 Uysal N, Zhang B, Jha S K, et al. DP-MAP: towards resistive dot-product engines with improved precision. In: Proceedings of the 39th International Conference on Computer-Aided Design (ICCAD), San Diego, 2020. 1–9
26 Zou M, Zhu Z, Cai Y, et al. Security enhancement for RRAM computing system through obfuscating crossbar row connections. In: Proceedings of Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, 2020. 466–471
27 Kingra S K, Parmar V, Negi S, et al. Methodology for realizing VMM with binary RRAM arrays: experimental demonstration of binarized-adaline using OxRAM crossbar. In: Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS), Seville, 2020. 1–5
28 Chen X, Song T, Han Y. RRAM-based analog in-memory computing: invited paper. In: Proceedings of IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH), Alberta, 2021. 1–6
29 Ma W, Li L, Li Z, et al. RRAM-based precision-scaleable computing-in-memory scheme and its error correction approach. In: Proceedings of IEEE 5th International Conference on Artificial Intelligence Circuits and Systems (AICAS), Hangzhou, 2023. 1–5
30 Luo Y, Wang S, Zuo P, et al. Modeling and mitigating the interconnect resistance issue in analog RRAM matrix computing circuits. IEEE Trans Circ Syst I, 2022, 69: 4367–4380
31 Azzaz M, Vianello E, Sklenard B, et al. Endurance/retention trade off in HfO_x and TaO_x based RRAM. In: Proceedings of IEEE 8th International Memory Workshop (IMW), Paris, 2016. 1–4
32 Sun Z, Pedretti G, Mannocci P, et al. Time complexity of in-memory solution of linear systems. IEEE Trans Electron Devices, 2020, 67: 2945–2951
33 Zuo P, Sun Z, Huang R. Extremely-fast, energy-efficient massive MIMO precoding with analog RRAM matrix computing. IEEE Trans Circ Syst II, 2023, 70: 2335–2339
34 Han R Z, Huang P, Zhao Y D, et al. Efficient evaluation model including interconnect resistance effect for large scale RRAM crossbar array matrix computing. Sci China Inf Sci, 2019, 62: 022401
35 Wang C, Wu H, Gao B, et al. Relaxation effect in RRAM arrays: demonstration and characteristics. IEEE Electron Device Lett, 2016, 37: 182–185
36 Veksler D, Bersuker G, Vandelli L, et al. Random telegraph noise (RTN) in scaled RRAM devices. In: Proceedings of IEEE International Reliability Physics Symposium (IRPS), Monterey, 2013. 14–18
37 Chen L, Li J, Chen Y, et al. Accelerator-friendly neural-network training: learning variations and defects in RRAM crossbar. In: Proceedings of Design, Automation & Test in Europe Conference & Exhibition (DATE), Lausanne, 2017. 19–24
38 Chen P Y, Lin B, Wang I T, et al. Mitigating effects of non-ideal synaptic device characteristics for on-chip learning. In: Proceedings of IEEE/ACM International Conference on Computer-Aided Design (ICCAD), Austin, 2015. 194–199
39 Liu B, Li H, Chen Y, et al. Reduction and IR-drop compensations techniques for reliable neuromorphic computing systems. In: Proceedings of IEEE/ACM International Conference on Computer-Aided Design (ICCAD), San Jose, 2014. 63–70
40 Gao S, Yang F, Lu C, et al. Fast algorithms for exact IR drop de-embedding in analog multiply-accumulate computing. IEEE Trans Electron Devices, 2022, 69: 6376–6383
41 Hu M, Li H, Wu Q, et al. Hardware realization of BSB recall function using memristor crossbar arrays. In: Proceedings of DAC Design Automation Conference, San Francisco, 2012. 498–503
42 Liang J, Wong H S P. Cross-point memory array without cell selectors-device characteristics and data storage pattern dependencies. IEEE Trans Electron Devices, 2010, 57: 2531–2538
43 Li T, Bi X, Jing N, et al. Sneak-path based test and diagnosis for 1R RRAM crossbar using voltage bias technique. In: Proceedings of the 54th ACM/EDAC/IEEE Design Automation Conference (DAC), Austin, 2017. 1–6
44 Higham N J. Accuracy and Stability of Numerical Algorithms. 2nd ed. Philadelphia: Society for Industrial and Applied Mathematics, 2002. 15: 287–304
45 Wan W, Kubendran R, Schaefer C, et al. A compute-in-memory chip based on resistive random-access memory. Nature, 2022, 608: 504–512
46 Wang Z, Li C, Song W, et al. Reinforcement learning with analogue memristor arrays. Nat Electron, 2019, 2: 115–124