Tzarouchis et al. *Light: Science & Applications* (2022)11:263
https://doi.org/10.1038/s41377-022-00950-1

Official journal of the CIOMP 2047-7538
www.nature.com/lsa

ARTICLE Open Access

Mathematical operations and equation solving with reconfigurable metadevices

Dimitrios C. Tzarouchis[1], Mario Junior Mencagli[2], Brian Edwards[1] and Nader Engheta[1]✉

Abstract

Performing analog computations with metastructures is an emerging wave-based paradigm for solving mathematical problems. For such devices, one major challenge is their reconfigurability, especially without the need for a priori mathematical computations or computationally-intensive optimization. Their equation-solving capabilities are applied only to matrices with special spectral (eigenvalue) distribution. Here we report the theory and design of wave-based metastructures using tunable elements capable of solving integral/differential equations in a fully-reconfigurable fashion. We consider two architectures: the Miller architecture, which requires the singular-value decomposition, and an alternative intuitive direct-complex-matrix (DCM) architecture introduced here, which does not require a priori mathematical decomposition. As examples, we demonstrate, using system-level simulation tools, the solutions of integral and differential equations. We then expand the matrix inverting capabilities of both architectures toward evaluating the generalized Moore–Penrose matrix inversion. Therefore, we provide evidence that metadevices can implement generalized matrix inversions and act as the basis for the gradient descent method for solutions to a wide variety of problems. Finally, a general upper bound of the solution convergence time reveals the rich potential that such metadevices can offer for stationary iterative schemes.

Introduction

Naturally occurring or artificially designed systems are often modeled by linear equations, which can be expressed through simple algebraic entities, e.g., vectors and matrices. Building on these entities, numerical linear algebra describes the methods for performing a large variety of operations, such as vector-matrix multiplications and matrix inversions, which enable the extraction of valuable information regarding the system under study[1]. The most successful platform for numerically studying these systems is through the digital electronic computer. However, as these devices reach their limits with regards to speed and power consumption considerations[2], the idea of designing more efficient, application-specific, analog platforms has gained significant momentum in recent years[3]. To this end, one can find two major areas of hardware-enhanced analog computational approaches: electronic[4] and wave-enabled (free-space[5–7] and waveguided[8,9]) systems.

In the very mature electronic domain, there are examples of reconfigurable memristive devices that implement non-von Neumann/neuromorphic[10,11] crossbar architectures that perform enhanced analog real-valued matrix-vector operations[10] and solve real-valued mathematical equations[11–13]. In the free-space optical domain, analog vector-matrix multiplication for image processing and deep-learning is performed through a cascade of lenses and volumetric scatterers[7,14,15]. However, given that conventional optical systems are inherently bulky, the flat-optics/metasurfaces demonstrate mathematical functionalities, e.g, computational imaging[16], mathematical operations (integration/differentiation[5,6,17,18]), and discrete Fourier transformations[19], in a much more compact manner.

In parallel, waveguided systems, such as integrated photonics, are proving to be a suitable platform for analog —classical[20] and quantum[8]—computations and machine

Correspondence: Nader Engheta (engheta@ee.upenn.edu)
[1]University of Pennsylvania, Department of Electrical and Systems Engineering, Philadelphia, PA 19104, USA
[2]University of North Carolina at Charlotte, Department of Electrical and Computer Engineering, Charlotte, NC 28223, USA

© The Author(s) 2022
Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit http://creativecommons.org/licenses/by/4.0/.

learning tasks[21,22]. These systems are designed using a combination of tunable and reconfigurable directional couplers, e.g., Mach–Zehnder interferometers (MZIs)[23], or tunable photonic waveguides[24–27] and other wave-based tunable technologies, e.g., graphene and semiconducting nanopillars [28–30]. The majority of the MZI-based systems utilize the network designs introduced by: (a) Reck et al.[31] (first of its kind at the optical/quantum domain), (b) Miller[32,33] (with *self-configuring* characteristics), and (c) Clements et al.[34] (balanced path lengths). All three architectures require the a priori mathematical decomposition of the main operator. Like optics, photonic devices are orders of magnitude larger than the operation wavelength. Hence, inverse-designed metastructures are emerging examples for compact, ultrafast, low-power, and parallel analog computations[9,35].

The majority of the aforementioned wave-based approaches, including the neural network approaches[21], deal with the matrix-vector product problem (wave interference). For the matrix inversion problem, one can find works based on optical systems[24,36–38] and metastructures[9]; the latter exhibiting limited reconfigurability. All the inversion systems[9,36–38] (including the ones at the electronic domain[12,13]), implement a simple Jacobi iterative scheme[39], by a properly designed feedback loop. The scheme converges only for square matrices with eigenvalues inside the unit circle[40]. Inevitably, such devices cannot be readily used for the inversion of general matrices, e.g., rectangular (minimization/regression problems), singular (inverse scattering/imaging problems), or simple square matrices with complex eigenvalues outside the unit circle. Therefore, the need for a wave-based system that (1) intuitively offers full reconfigurability without any a priori mathematical decomposition of the operator and (2) can perform matrix inversion for general matrices is apparent. Here we introduce such a system.

This work is structured and presented in two parts. The first part deals with the metadevice architecture, while the second part addresses the extension of its capability to invert general matrices. Specifically, first, we theoretically demonstrate (via system simulations) the matrix inversion capabilities of a structure that uses the Miller architecture (requires singular-value decomposition (SVD)) for solving integral equation (IE) problems in the complex domain. Next, we introduce an alternative direct-complex-matrix (DCM) architecture that requires no a priori mathematical calculations and can be tuned intuitively. The DCM architecture can be regarded either as a generalized form of a tunable beamforming/beamsteering network or the wave-analog of a crossbar architecture with complex algebra abilities. The DCM architecture is then used to find the solutions of a set of differential equations (DE). Finally, we use both architectures to perform the generalized inversion of a complex-valued singular and a binary-valued rectangular matrix, i.e., implementing the generalized Moore–Penrose inverse.

The second part of this work relates to the choice of the DE problems, highlighting that even well-studied DEs may yield matrices that cannot be inverted with a naively applied Jacobi iteration method. For these purposes, we rethink the algorithm, introducing a mathematical modification inspired by the definition of the pseudoinverse[41], which is essentially equivalent to the modified Richardson[42,43] iteration method, an equivalent form of gradient descent method (GDM)[44]. In other words, this modification enables the generalized inversion of any matrix, including singular and rectangular cases. We demonstrate this method utilizing both architectures, extending the previously reported results[9], and we extract theoretical upper bounds that quantify the possible hardware-induced enhancement to the convergence of GDM. Both architectures demonstrate that metadevices may enable compact, parallel, low-power, ultrafast platforms for general numerical linear algebraic problems (matrix inversion and the GDM) and open avenues for metadevice-powered generic algebraic explorations. The proposed metadevice can be suitably aligned with open quantum algorithmic[45] and quantum computing[26] forefronts.

Results

Solving integral equations with the Miller architecture

To begin with, we consider the following integral equation:

$$x(u) = c(u) + \int_a^b K(u,v)x(v)dv \tag{1}$$

which is known as the Fredholm integral equation of the second kind[43]. Based on a feedback network consisting of m input and m output waveguides, Eq. (1) can be represented in matrix-vector form as $\mathbf{x} = \mathbf{c} + \mathbf{K}\mathbf{x}$, where $\mathbf{x} \in \mathbb{C}^{m\times 1}$ is the solution vector, $\mathbf{c} \in \mathbb{C}^{m\times 1}$ denotes the input vector, and $\mathbf{K} \in \mathbb{C}^{m\times m}$ is the kernel matrix obtained by sampling $K(u,v)$ in $m \times m$ points taking into account the appropriate $\Delta x = (b-a)/m$ factor. (Following the numerical algebra nomenclature, boldfaced capital and lowercase Latin variables indicate matrices and vectors, respectively, while Greek and unboldfaced Latin letters indicate scalar quantities.) Although the previous vector equation is identical to the common feedback equation in electronics, its variable's physical meaning is fundamentally different. In electronics, the signals are represented as functions of time, and the processing occurs in the temporal domain; here, they represent the complex amplitude of monochromatic electromagnetic wave modes, and the processing occurs in the spatial domain[9,46]. The matrix $\mathbf{K}$ can then be interpreted as a

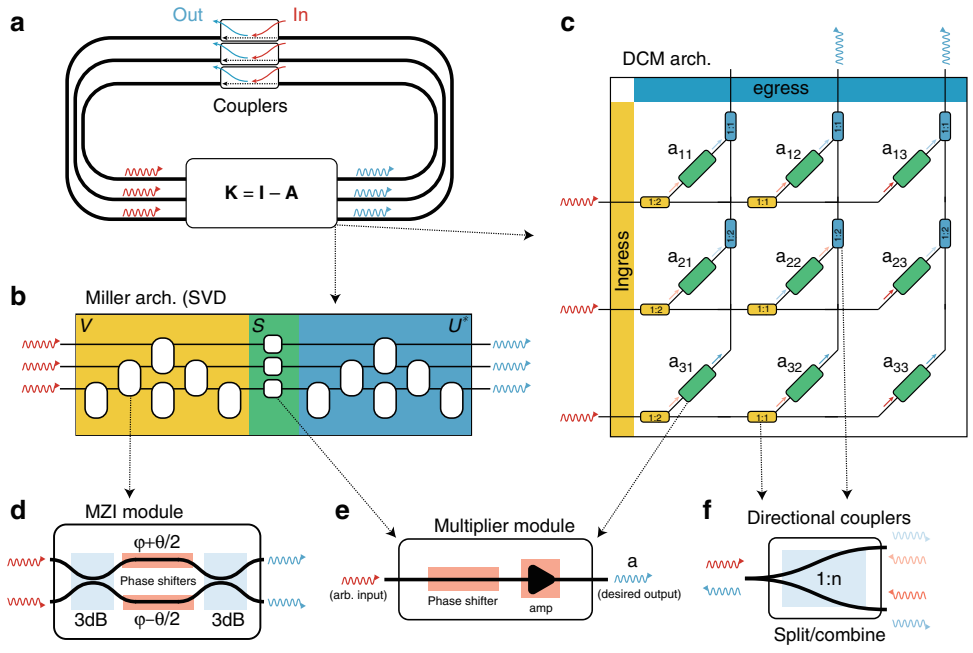

Fig. 1 Reconfigurable equation solvers. Solving mathematical equations (or equivalently calculating the inversion of a matrix **A**) requires the implementation of an operator (i.e., $\mathbf{K} = \mathbf{I} - \mathbf{A}$) within a looped system, schematically shown in **a**. Here we explore two different reconfigurable architectures: **b** The Miller architecture consists of three stages, each representing the three matrices of the singular matrix decomposition. Stages **V** and $\mathbf{U}^*$ can be implemented as a cascade of Mach–Zehnder interferometers (MZI, **d**). In contrast, the **S** stage can be implemented either with MZI or with a multiplier module (**e**), where a given input signal can be modified accordingly via a phase shifter and an amplification/attenuation unit. **c** A more straightforward and more intuitive implementation of the same operator can be given in the introduced DCM architecture. Here, the ingress stage consists of directional couplers **f**) with a fixed split ratio. Each of the outputs is driven to a dedicated multiplier (**e**) where the desired matrix value is assigned (here is an example of a 3×3 matrix). Finally, the signals are recombined at the egress stage using a set of directional couplers (**f**). Note that an MZI module can be reconstructed as two 3dB couplers and two multipliers, making the multiplier module the main constituent module for both architectures

transmission matrix relating a set of input waves to a set of output waves, thus describing the matrix-vector multiplication **Kx**. This linear operation can be performed in a variety of ways. Within this section, we employ an architecture based on the MZI mesh proposed by Miller[32,47] (see Fig. 1). Such a mesh configuration, exploiting the SVD, enables us to implement any $m \times m$ transmission matrix. In addition to the advantages of the previous metamaterial-based equation solver[9] (i.e., phase stability, low-power consumption, compatibility with silicon photonics technology[8], and implementation of more general, translational (shift) variant kernels), this approach presents two other features. First, this platform is fully programmable. With high reconfigurability, it can perform multiple functions such as solving linear integral and differential equations, and performing matrix inversion. Second, contrary to inhomogeneous materials, an MZI mesh that implements a kernel operator can be trained by simple progressive algorithms[32] without involving global optimization techniques.

To test the performances of the proposed MZI-based computing platform, as an example here we employ it to solve Eq. (1) characterized by the following kernel (Fig. 2a)

$$K(u,v) = 1.2H_0^{(2)}\left(8\sqrt{(u+1)^2+(v+1)^2}\right) + 1.7iH_0^{(2)}\left(8\sqrt{u^2+(v-1)^2}\right) \quad (2)$$

where $H_0^{(2)}(.)$ is the Hankel function of the second kind and order 0, defined over $[a,b] = [-1,1]$. Considering a feedback network composed of $m = 7$ waveguides, the 7×7 transmission matrix **K** is obtained based on the kernel distribution (Fig. 2b). Using the SVD technique, **K** is then written as the product of three transmission matrices: $\mathbf{K} = \mathbf{VSU}^*$, where * denotes the complex conjugate transpose operation. **U** and **V** are unitary transmission matrices and are respectively implemented by the left and right triangular-shaped MZI sets of Fig. 2e. These MZI sets have been configured by a compiled programming protocol described in ref. [32], whose phase shifters settings for each MZI result from a progressive algorithm based on unitary operator factorization. **S** is a 7×7 diagonal transmission matrix of non-negative real numbers smaller than one satisfying the

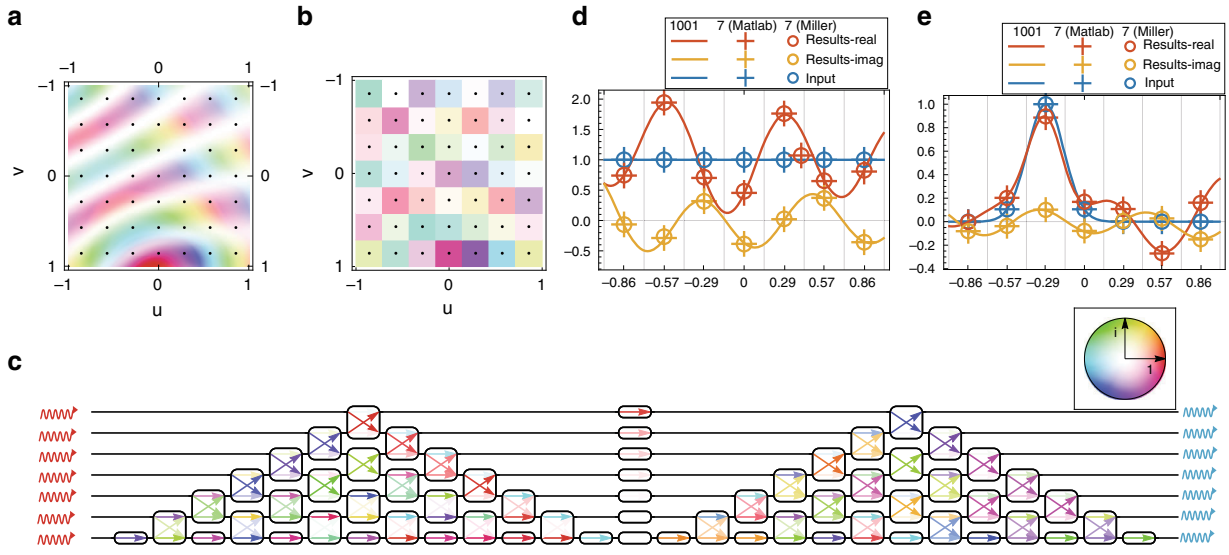

Fig. 2 Solving integral equations with the Miller architecture. a Continuous distribution of the kernel $K(u, v)$. The black dots denote the spatial points where the kernel is sampled to derive its 7×7 discrete version **K** reported in the color plot (**b**). The maximum amplitude of the complex value in the color circle insets is set to 1.15 and (2/7)(1.15) in **a** and **b**, respectively. **c** Schematic of flow through the MZI network wherein the color of the arrows denotes the complex transmission between the MZI ports with a maximum value of 1. **d**, **e** The extracted solution (real-red and imaginary-yellow) of the integral equation (1) with the proposed kernel under two different real inputs (blue). **d** The input is a constant value 1, while in **e**, it is an appropriately sampled Gaussian distribution centered on $u = -2/7$. This solution is evaluated in three different ways: the highly sampled ($n = 1001$) *true* solution (continuous line) evaluated using linear algebra (i.e., $\mathbf{x} = (\mathbf{I} - \mathbf{K})^{-1}\mathbf{c}$) method, the solution (crosses) with $n = 7$ also calculated directly using linear algebra, and the solution (circles) calculated via the simulation of our MZI mesh. All three show excellent agreement

convergence condition[9] and has been implemented by 7 multipliers connecting the two triangular-shaped MZI sets (see Fig. 2b). These multipliers have been configured to match the diagonal elements of **S**.

It is worth noticing that the proposed computing platform can also be applied to kernels that do not satisfy the convergence condition. As indicated in literature[9], in such cases, the kernel and the input signal can be properly scaled down to meet the convergence condition and ensure that integral equation recursive and exact solution remains proportional. Satisfying the convergence condition in a wave-based recursive computing platform implies that any time the waves circulate through the structure representing the kernel, they lose part of its energy. Unlike the approach utilized in[9], where such energy gets dissipated by absorbing materials, in the platform here, that energy is not lost and is still available at non-connected ports of MZIs implementing **S**. As a result, it can be potentially used for other purposes.

The MZI structure, the waveguide feedback network, and in/out couplers with 1% coupling coefficient were numerically simulated with a system-level software[48] (see supporting information). Figure 2c, d present the solutions (real and imaginary part) of Eq. (1) with the kernel in (2) for two different inputs, obtained using (i) the proposed approach using MZIs (shown as dots) in which the discrete solution **x** has been extracted from the couplers in the feedback network, (ii) a standard numerical inversion of the corresponding 7×7 linear system (shown as crosses), and (iii) theoretical approach, i.e., the linear system resulting from the oversampling of the kernel approaching its continuous variation over the entire chosen spatial domain (shown as a solid line). As can be seen, the MZI-based structure's solutions agree well with the corresponding linear equations' exact solutions and the exact almost continuous solutions derived by oversampling the kernel (see supporting information).

Solving differential equations with the DCM architecture

The adaptation of the Miller architecture for a metastructure that is able to solve integral equations introduces a compact, elegant and self-configuring realization for any given operator, provided a preliminary SVD decomposition of the operator itself is performed beforehand. However, its elegance comes at a cost, since performing the SVD for a matrix is as computationally involved as inverting the same matrix. In the context of solving a set of linear equations (i.e. matrix inversion), preferably one might desire to implement and solve any kernel without incurring this cost. This section describes the DCM network architecture that allows the realization of any matrix in a simple, intuitive, and direct manner, without the need for any a priori calculations.

As seen in Fig. 1 the alternative architecture consists of three stages: the *ingress* (yellow), *middle* (green), and *egress* (blue) stages. The input signal vector is directed at

the ingress stage, where each one of the m inputs is split into m outputs to yield m^2 distinct values (m lines, split m times), each of them with relative amplitude $1/\sqrt{m}$.

The ingress stage can be implemented in a multitude of ways. For example, one could use either m signal-splitters (using inverse-designed waveguides or RF lumped elements, etc.) or m cascades of $m-1$ properly connected directional couplers or MZIs. In either case, these devices are fixed and are not dependent on the choice of the kernel. Note that any amplitude or phase change that the ingress stage introduces can be compensated by the proper adjustment of the components at the middle stage.

The m^2 outputs of the ingress stage are directed into m^2 dedicated *complex multipliers*, i.e., the middle stage. An individual complex multiplier changes the amplitude and phase of its input signal, and is a direct implementation of an element of the kernel matrix. A possible implementation of a multiplier module can be achieved either by using a phase-shifting component followed by an amplifier (that offers a dynamical range of amplification or attenuation), or a tunable directional-coupling module, or an MZI. The first choice might be particularly interesting and feasible in the RF-electronics platforms while the MZI approach can be particularly attractive for integrated photonics platforms.

Finally, the m^2 outputs of multipliers are directed to the final egress stage. This stage operates similarly to the ingress but in the reverse direction. For example, it combines m^2 input signals into m output signals (as the output vector), which are then directed to the feedback-coupling modules. The m outputs are consequently directed to the feedback loop stage, that essentially enables the extraction of the main matrix inversion. (It is important to note the egress stages is not a power combiner. The output amplitude of each of the m egress parts is proportional to the fraction (i.e., $1/\sqrt{m}$) of the algebraic sum of complex-valued amplitudes of m input signals to it, and consequently the output power is usually less than the sum of the input power. As a result, the remaining power goes out of the system in unused channels in the MZI mesh in egress. After the egress stage we need to have amplifiers to bring the output signal level up to compensate for the $1/\sqrt{m}$ of the ingress and the $1/\sqrt{m}$ of the egress (i.e., therefore compensate for $1/m$).)

The DCM architecture functionally shares a few similarities with the electronic crossbar architectures, e.g., as in refs. [12,13], only this time using waveguiding systems. The electronic crossbar (or crosspoint) architecture utilizes Ohm's and Kirchhoff's law combined with analog and reconfigurable resistive memories to perform real-valued calculations. In our case, the reconfigurable resistive memories are the multiplier modules that enable full-complex arithmetic manipulation. At the same time, the ingress/egress stages utilize the wave signal splitting and recombining to deliver the signal towards all the multiplier modules and the output. In contrast, the electronic counterparts utilize Ohm's and Kirchhoff's law (voltage division and current-voltage proportionality) for the same purpose.

Although the proposed alternative architecture can be applied to solve IEs, here we apply it to solve another major class of problems, ordinary differential equations, even with non-constant coefficients. As we mentioned above, the existence of a feedback loop with a certain initial excitation $\mathbf{c}$ along the main kernel $\mathbf{K}=\mathbf{I}-\mathbf{A}$ yields to the matrix inversion of the following problem $\mathbf{A}=\mathbf{I}-\mathbf{K}$, hence the solution of any linear set of equations can be obtained. The key for solving differential equations (DE) is the proper discretization and the formulation of the equivalent linear problem, akin to the case of integral equations.

For reference, the 1D (in terms of x) DE problem of 2nd order can be generally expressed as

$$\mathcal{D}: \begin{cases} r(x)\frac{d^2f(x)}{dx^2}+p(x)\frac{df(x)}{dx}+q(x)f(x)=g(x), x\in[x_\mathrm{d},x_\mathrm{u}] \\ f(x_\mathrm{d})=a, f(x_\mathrm{u})=b \end{cases} \tag{3}$$

where the functions $r(x)$, $p(x)$, $q(x)$, and $g(x)$ are general complex functions while the values a, b are the boundary conditions at the extreme points of a given interval $[x_\mathrm{d}, x_\mathrm{u}]$ for which the solution of the differential equation is desired. While the above expression implements a Dirichlet-type boundary conditions, other types of boundary conditions can also be implemented in a similar manner.

Following a standard Finite Differences (FD) method (see Methods section) one can obtain an equivalent difference equation (ΔE), i.e., a discrete version of DE, whose solution (inversion) yields the continuous solution of the DE providing the mesh is sufficiently dense. The corresponding discrete version of the problem reads

$$\mathcal{D}_\Delta: \begin{cases} r(x_j)\left(f_{j+1}-2f_j+f_{j-1}\right)+p(x_j)\frac{f_{j+1}-f_{j-1}}{2}(\Delta x) \\ \quad +q(x_j)f_j(\Delta x)^2=g(x_j)(\Delta x)^2, j\in\{1,\ldots,m\} \\ f_0=a, f_{m+1}=b \end{cases} \tag{4}$$

where x is the unknown variable and $\Delta x=\frac{|x_\mathrm{d}-x_\mathrm{u}|}{m}$ is the difference constant of the interval $x\in[x_\mathrm{d},x_\mathrm{u}]$ discretized in m pieces. In this way it is possible to express the problem as $\mathbf{Ax}=\mathbf{c}$, where $\mathbf{A}\in\mathbb{C}^{m\times m}$ is the difference matrix, $\mathbf{c}\in\mathbb{C}^{m\times 1}$ is the input vector (including the boundary conditions) and $\mathbf{x}\in\mathbb{C}^{m\times 1}$ is the unknown solution vector. As before, the solution vector $\mathbf{x}$ can be found once the kernel of the metadevice reads $\mathbf{K}=\mathbf{I}-\mathbf{A}$. It is worth noticing that the matrix inversion is affected by the eigenvalues of the kernel, i.e., the eigenvalues should be strictly within the unit circle in order for the iterative method to converge. The behavior of the kernel follows the spectral behavior of $\mathbf{A}$. In general (see Methods and

supporting information), the combination of the arbitrary complex functions $r(x)$, $p(x)$, $q(x)$ with $\mathbf{A}_{\Delta x}$ (diagonally dominant (This holds for strictly diagonal matrices in our case, however, the introduction of a small positive perturbation allows us to expand this argument.)), $\mathbf{P}_{\Delta x}$ (skew-symmetric) and $\mathbf{I}$ (unit matrix) can lead to $\mathbf{A}$ matrices with arbitrarily distributed complex eigenvalues. The above point can affect the convergence of the implemented iterative method. Previous reports on solutions of integral and differential equations with hardware-accelerated approaches[9,12,13] demonstrated the implementation of a matrix kernel with eigenvalues within the unit circle, i.e., a condition that guarantees the convergence of the iteration method. It may appear that this feature limits the applicability of these devices for solving more general problems. Here, however, we address this problem by carefully preconditioning the kernel and modifying the input, which eventually can lead to the realization that the proposed hardware-accelerated or hardware-implemented solvers, including our proposed architectures, can expand the range of their applicability towards a larger problem set.

Let us first extract and examine the peculiarities of certain well-known DE equations. For these examples we considered an 11×11 network consisting of ideal directional couplers and MZIs using the AWR Microwave Office®[48]. Similar to the previous section, here the results are obtained using 1% (−20 dB) couplers for the feedback loop (see supporting info). Additionally, all the results are compared against the exact solution and a FD code that implements the same method in MATLAB (see Methods). As the first example, we start with the Hermite equation that reads

$$\frac{d^2}{dx^2}f(x) - 2x\frac{d}{dx}f(x) + kf(x) = 0 \tag{5}$$

This particular equation is useful in a multitude of fields since it is the basis of the Hermite polynomials[49]. The general well-known solution reads

$$f(x) = c_1 \mathrm{H}_{\frac{k}{2}}(x) + c_2 {}_1\mathrm{F}_1\left(-\frac{k}{2}, \frac{1}{2}, x^2\right) \tag{6}$$

where $\mathrm{H}_n(x)$ is the Hermite polynomial and ${}_1\mathrm{F}_1(x)(a, b, x)$ is the confluent hypergeometric function of the first kind[50]. In our case we seek for a solution on the interval $x \in [-1, 1]$ with conditions $f(-1) = f(1) = \frac{1}{2}$. Following the above analysis we have $r(x) = 1, p(x) = -2x$, $q(x) = -10$, and $g(x) = 0$. The solution of the above is $f(x) = \frac{3}{38e}e^{x^2}\left(1 + 4x^2 + \frac{4}{3}x^4\right)$. For this particular case the matrix $\mathbf{A}$ has eigenvalues with a negative real part. The "modified" kernel matrix $\mathbf{K} = \mathbf{I} - \alpha_\lambda \mathbf{A}$, where the additional scale factor is $\alpha_\lambda = -0.2$ such that all eigenvalues of the modified kernel lie within the unit circle. Eigenvalues bound theorems, such as Gershgorin's theorem[50], can provide a good estimation of how large the scaling factor should be. As we can see in Fig. 3a.1, a comparison between the analytical solution, the simulated results from the MZI implementation and the corresponding MATLAB code reveals an excellent agreement. A denser discrete mesh of the interval (larger matrix implementation) can further improve the results' accuracy. An interesting feature here is that for these types of problems where the inverse of sparse matrices is required, one could avoid the implementation of a full $m \times m$ network, but rather a much smaller subset (see for example Fig. 3a.2).

The second example examines the solution of Helmholtz's equation, i.e.,

$$\frac{d}{dx^2}f(x) + k^2 f(x) = 0 \tag{7}$$

on $x \in [0, 1]$ with $f(0) = 1$ and $f(1) = 0$. For this particular example we assume $k = 5$. In this case the solution reads $f(x) = c_1 e^{ikx} + c_2 e^{-ikx}$ with $c_1 = \frac{1}{1-e^{10i}}$ and $c_2 = \frac{-e^{10i}}{1-e^{10i}}$.

Although this is a seemingly simple equation we observe that with a proper choice of k and Δx the difference matrix $\mathbf{A}$ can have eigenvalues with real parts both positive and negative (inset plot at Fig. 3b.1). For this case the direct implementation of such kernel will not converge. We can overcome this obstacle by implementing the following modified kernel instead

$$\mathbf{K} = \mathbf{I} - \alpha_\lambda \mathbf{A}^* \mathbf{A} \tag{8}$$

where $\alpha_\lambda = 0.1415$ is a properly chosen scaling factor. In this case we can extract the solution once we apply the following input vector, $\mathbf{c}_{\mathrm{new}} = \alpha_\lambda \mathbf{A}^* \mathbf{b}$. As we can see in Fig. 3 (b.1) the results yield a very good agreement with the analytical curve, while the implemented matrix (Fig. 3b.1) shows that we have the implementation of an almost pentadiagonal matrix. Note that the zero-valued components could be as well omitted, allowing the implementation of such a system with fewer MZIs/multipliers.

Finally, we examine the solution of the Airy equation, i.e.,

$$\frac{d^2}{dx^2}f(x) + k^2 x f(x) = 0 \tag{9}$$

This is a function with non-constant coefficients with an analytical solution that is expressed in terms of the Airy functions of the 1st and 2nd order, i.e.

$$f(x) = c_1 \mathrm{Ai}\left(-\frac{k^2}{k^{4/3}}x\right) + c_2 \mathrm{Bi}\left(-\frac{k^2}{k^{4/3}}x\right) \tag{10}$$

where c_1 and c_2 are arbitrary constants[49]. Here we examine the solution for $k = 2\pi$ in the range $x \in [0, 1]$ with $f(0) = 1$ and $f(1) = 0$ ($c_1 = 1.16 + i0.42$ and $c_2 = 0.95 - i0.24$). As before, the difference matrix exhibits some eigenvalues with negative and positive real parts

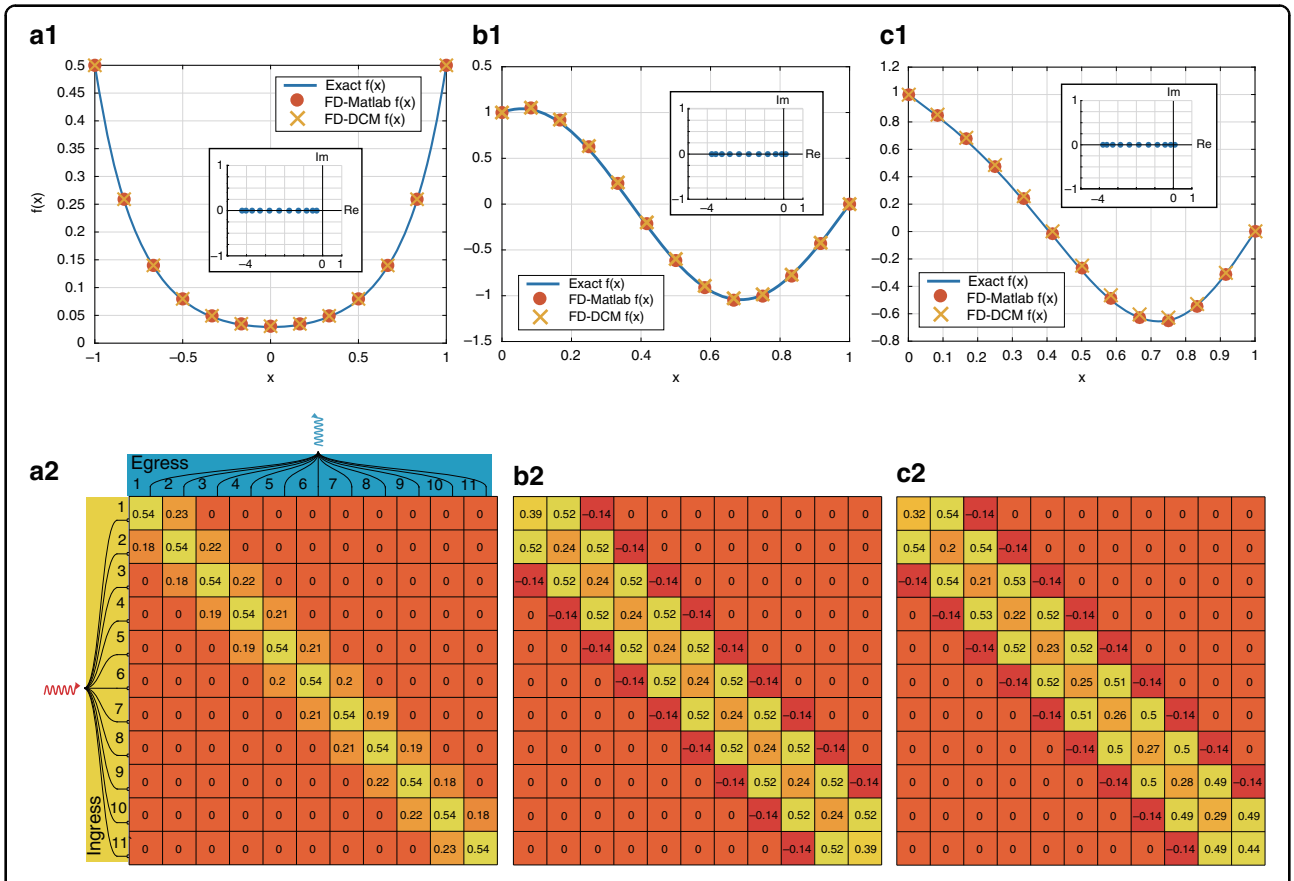

Fig. 3 Solving differential equations with the DCM architecture. A schematic representation of an 11×11 network is depicted, with the corresponding element numbering. (a.1–a.2) The result for the case of a Hermite equation in the interval $x \in [-1, 1]$ with boundary conditions $f(-1) = f(1) = 1/2$. The inset figure depicts the distribution of the eigenvalues of the corresponding difference matrix **A** in the complex plane. For this case, all eigenvalues are negative (reside in the left-half plane). The corresponding color-coded matrix in (a.2) represents the values for each of the multipliers of the alternative network; (b.1–b.2) Similarly, the solution of a Helmholtz equation in the range $x \in [0, 1]$ with $f(0) = 1$ and $f(1) = 0$. For this case, one of the eigenvalues of the difference matrix resides in the right-half plane; (c.1–c.2) Finally, the solution of an Airy equation for $x \in [0, 1]$, $f(0) = 1$ and $f(1) = 0$. For case **a** the kernel matrix is derived using the following formula $\mathbf{K} = \mathbf{I} - \alpha_\lambda \mathbf{A}$, where α_λ is just a scaling factor, while for the cases (**b**) and (**c**) the implemented kernel was derived by $\mathbf{K} = \mathbf{I} - \alpha_\lambda \mathbf{A}^* \mathbf{A}$

(inset plot at Fig. 3 (c.1)), hence the utilized preconditioned is with the scale factor $\alpha_\lambda = 0.138$. As we can observe in Fig. 3 (c.1) the results have an excellent agreement. Notice that both Fig. 3 (b.2) and Fig. 3 (c.2) cases implement a quite similar (not identical) kernel.

The results for all the above cases demonstrate several notable features. First, the kernel has been intuitively implemented via the DCM architecture, yielding results with excellent agreement compared to the analytical solutions. Second, the proposed spatially discretized approach allows for the solution of general differential equations, especially those with non-constant coefficients, such as the Hermite and Airy equations. This is a feature that clearly outperforms other conventional hardware-based approaches that use a Fourier transformation approach in the time domain[51] - in these cases the solution of non-constant differential equations is complicated. Third, some of the available solutions are expressed in terms of highly specialized forms, such as the Airy, Hermite, and Hypergeometric functions. In all these cases the MZI network actually generates the expected results, demonstrating that the proposed machine can be used as an alternative way for the evaluation of cumbersome/generalized functions. As a reminder, the used methodology is subject to all mathematical restrictions of the FD methodology[52].

Wave-enhanced GDM and its convergence bounds

Let us first examine some of the details and implications of the preconditioning that we described above. In all the works where the solution of equations is desired, essentially the following basic iterative scheme

$$\mathbf{x}_{n+1} = \mathbf{c} + (\mathbf{I} - \mathbf{A})\mathbf{x}_n \tag{11}$$

is utilized, where **A** is the desired square operator/matrix to be inverted and n is the iteration number. This scheme

can indeed lead to a solution for the linear problem when the unknown vector $\mathbf{x}$ has sufficiently close values for two iterations, i.e., $||\mathbf{x}_{n+1} - \mathbf{x}_n|| < \epsilon$, where ϵ is the desired accuracy error. Once this criterion is achieved, then we may conclude that $\mathbf{x}_{\mathrm{n}+1} \approx \mathbf{x}_{\mathrm{n}} \approx \mathbf{x} = \mathbf{A}^{-1}\mathbf{c}$. Despite the conceptual simplicity of Eq. (11), the convergence of the method depends entirely on the distribution of the (complex) eigenvalues of the kernel $\mathbf{K} = \mathbf{I} - \mathbf{A}$, i.e., the eigenvalues should reside within the complex unit circle.[38,40]. Moreover, all the recent examples, e.g., with inverse-designed metastructures[9] or with electronic memristor array technology[12,13], demonstrate the inversion of well-behaved (non-singular) square matrices that satisfy the aforementioned convergence criterion.

From the solution of DE it became apparent that even well-defined problems can yield matrices with eigenvalues beyond the unit circle, making the convergence of the implemented feedback scheme challenging. While for some cases a simple scaling of the kernel suffices, we examined some examples where the solution is possible only after the proper preconditioning of both the kernel and the input of the system. Here we examine that the $\mathbf{A}^*\mathbf{A}$ modification, introduced in for optical systems in[40], can have much wider implications that enable the utilization of electronic or wave-based systems for inverting a much larger set of matrices. Actually, the aforementioned preconditioning allows us to evaluate the inverse of square matrices with arbitrary eigenvalues, but also to implement the notion of the generalized inverse (a.k.a. pseudoinverse) for rectangular and singular matrices, effectively introducing the GDM for the proposed metadevice architecture.

In full form, this simple modification reads

$$\mathbf{x}_{n+1} = \alpha_\lambda \mathbf{A}^*\mathbf{c} + (\mathbf{I} - \alpha_\lambda \mathbf{A}^*\mathbf{A})\mathbf{x}_n \tag{12}$$

which is nothing but a version of the well-known *Richardson iteration*[42], implementing essentially the geometric series version for operators (Neumann series). This iteration has been rediscovered in several contexts and forms, such as the *Landweber* iteration[43], and it represents a special form of the GDM[39].

At this point, a couple of observations can be made for the above expression. First, the quantity $\mathbf{A}^*\mathbf{A}$ forms a positive semidefinite matrix for any rectangular matrix $\mathbf{A}$, hence all of its eigenvalues reside in the positive real plane. Moreover, the factor α_λ can scale all the positive eigenvalues of $\mathbf{A}^*\mathbf{A}$, hence the new modified kernel $\mathbf{K} = \mathbf{I} - \alpha_\lambda \mathbf{A}^*\mathbf{A}$ always satisfies the convergence condition. As before, when $\mathbf{x}_{n+1} \approx \mathbf{x}_n$ we have that the iteration yields to the solution $\mathbf{x} = (\mathbf{A}^*\mathbf{A})^{-1}\mathbf{A}^*\mathbf{b}$. This approach coincides with the definition of the generalized inverse or pseudoinverse of a matrix[41]. Therefore, this simple modification enables the inversion of any given matrix. Additionally, GDM is a simple yet robust method for the evaluation of linearized problems, least-square problems, pseudoinverses, and unconstrained minimization problems[1]. Therefore, the proposed wave-based metadevice can in theory perform calculations and explorations for a much wider numerical linear algebra domain. However, GDM is a stationary method since both the step (scaling) factor α_λ and the kernel remain constant (stationary) at every iteration[39]. The possibility of expanding the proposed devices toward non-stationary numerical methods is an obvious next step directly affected by the reconfigurability and the output readout speed.

In support of the above theoretical observations, in Fig. 4 we demonstrate the pseudoinverse of a singular $\mathbf{A} \in \mathbb{C}^{5\times 5}$ and a rectangular binary $\mathbf{A} \in \mathbb{Z}^{3\times 5}$ matrix, utilizing both the Miller and the DCM architecture. For the first case (Fig. 4 (I)) the pseudoinverse of a singular matrix (one eigenvalue is zero—see inset Fig. 4a) using the preconditioning technique is presented for both architectures. We observe an excellent agreement with the *exact* solution (via MATLAB) for both platforms. For the second case (Fig. 4 (II)), we observe that by applying the same preconditioning one can get an excellent estimation of the required generalized inverse of a binary rectangular matrix. Overall, the simulated results reveal that both architectures can successfully perform the required pseudoinverse calculation.

Finally, the implementation of GDM (even in its simplest form) allows the extraction of a convergence bound that depends on both the properties of the matrix (eigenvalues) and the iteration timing (hardware). Assuming the case of a positive semidefinite matrix $\mathbf{A}$, the convergence of iteration in Eq. (11) is of the order

$$\mathcal{O}\left(t\kappa(\mathbf{A})\log\left(\frac{1}{\epsilon}\right)\right) \tag{13}$$

where $\kappa(\mathbf{A})$ is the condition number of the matrix and ϵ is the desired approximation error (see Methods section).

For either of the suggested wave-operated architectures, each iteration requires a total of $t = L\frac{\lambda}{c}$ time to complete, where c is the speed of light, λ the operating wavelength, and L is the overall network length (dimensionless quantity, i.e., expressed in wavelengths). The overall network length varies depending the waveguiding systems that are used for either the Miller or the DCM. Assuming photonic implementations the network length can be 2 or 3 orders of magnitude larger than the wavelength (a single MZI operating at 1550 nm has a typical physical length at several μm scale). A possible reduction of this length can be achieved with emerging ultracompact dielectric metamaterial approaches[53–55], while novel inverse-designed microwave/photonic implementations can reduce the required length to a few wavelengths[9]. Moreover, the proposed metadevice can be implemented with lumped RF elements/technology, hence pushing the

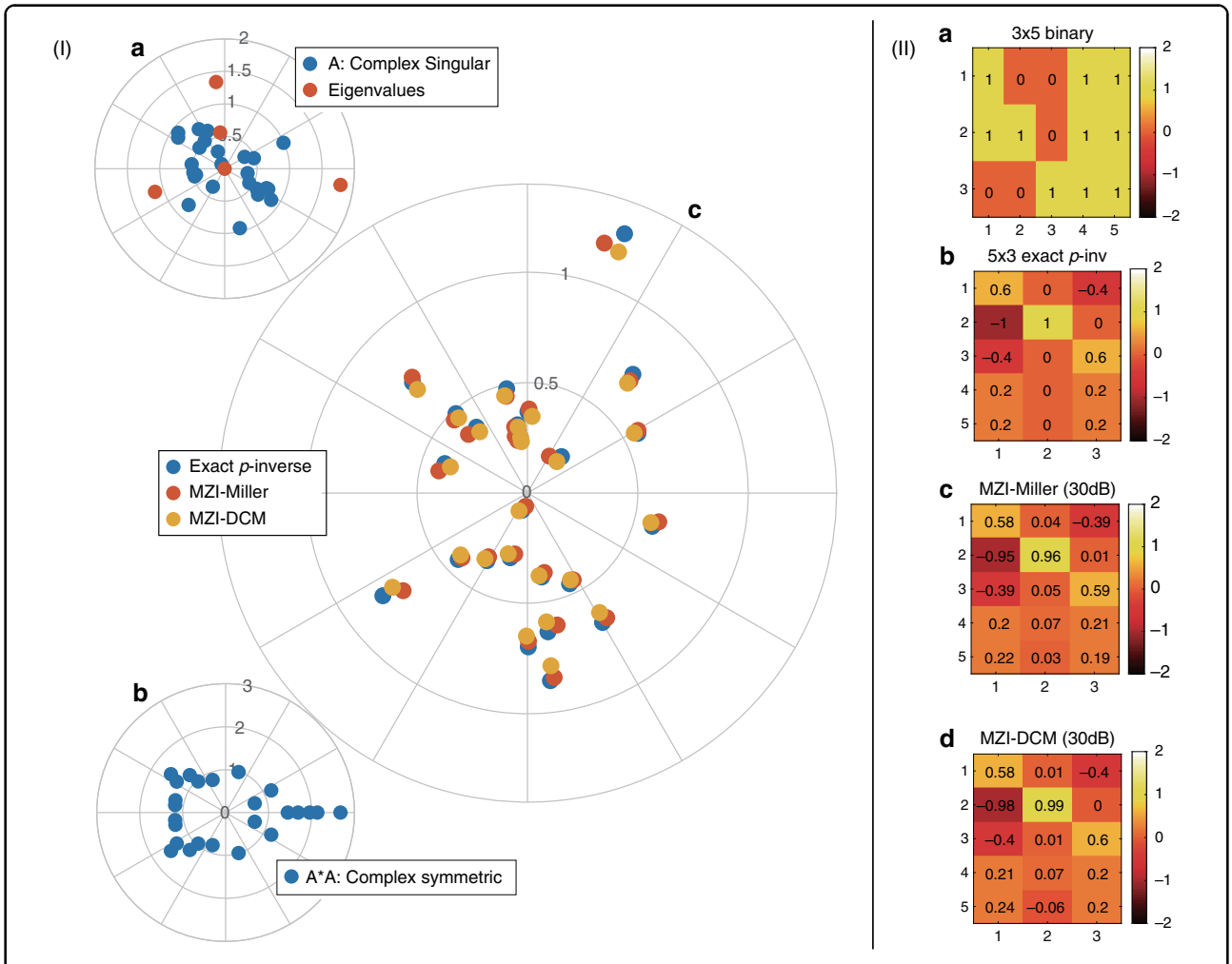

Fig. 4 Generalized inversion (pseudo-inversion) of singular and rectangular matrices using the Miller and DCM architectures. (Column I) The entries (blue dots) and the eigenvalues (red dots) of a singular $\mathbf{A} \in \mathbb{C}^{5\times 5}$ matrix (**a** polar plot) and the entries of the preconditioned complex symmetric matrix $\mathbf{A}^{*}\mathbf{A}$ (**b** polar plot). The central polar plot **c** depicts the entries of the pseudoinverse $\mathbf{A}^{+}$ (blue dots) compared against the results obtained via the Miller (red dots) and the DCM network architecture, both utilizing the optimum scaling factor $\alpha_\lambda = \frac{2}{\lambda_1+\lambda_5} \approx 0.316$. (Column II) The entries of a binary $\mathbf{A} \in \mathbb{Z}^{3\times 5}$ matrix **a** and the corresponding pseudoinverse **b**. The bottom figures depict the estimated pseudoinverses via the Miller (**c**) and the DCM network architecture (**d**), for $\alpha_\lambda = 0.125$ and a feedback coupler with a 30 dB coupling ratio

overall size to subwavelength regimes. Such size reduction could potentially lead to iteration timings at and below the nano/pico seconds range ($t \leq 10^{-9} - 10^{-12}$s). Therefore, the proposed metadevice holds the promise of an excellent playground for ultrafast numerical linear algebra techniques, within a given technological domain, such as optics, RF-circuits, photonics, and electronics.

Discussion

In conclusion, in this work, we theoretically demonstrated the matrix inversion capabilities of a structure that utilizes the Miller architecture and the introduced DCM architecture for solving IE and DE problems in the complex domain. In particular, the DCM architecture (the wave-analogous of the crossbar architecture in electronics) can intuitively implement any required matrix, without the need for the a priori calculations required by the Miller architecture.

We then performed the inversion (i.e., Moore–Penrose inverse) of complex-valued singular and binary-valued rectangular matrices using both architectures. The introduced mathematical modifications required for performing the above tasks revealed that the proposed device could perform the equivalent form of the gradient descent method. Finally, where possible, we extracted the upper convergence bounds of the inversion method. Those bounds depend on both the iteration time (platform dependent), the condition number of the matrix, and the required numerical accuracy of the computation. This bound revealed that matrix inversion could have the potential of ultrafast enhancement with the proper platform choice. However, the proposed metadevice

and the presented methodology are platform-independent, offering a fertile path for RF and photonic experimental implementation. Therefore, the ideas presented here can further investigate compact, parallel, ultrafast platforms for solving general numerical linear algebraic problems and other hardware-enhanced generic numerical linear algebraic methods with metadevices. Currently, we are in the midst of an experimental investigation of our DCM architecture for matrix inversions and other advanced mathematical functionalities using an RF implementation.

Materials and methods

System simulations

All presented results were obtained using AWR ®[48] Microwave Office, a commercially available RF circuital program able to perform system-level simulations. For both architectures an MZI equivalent module was used as the main module. In the first case a 7×7 Miller matrix was constructed, while for the second case an 11×11 DCM architecture. The final results regarding the inversion of singular and rectangular cases were simulated in two 5×5 system for both Miller and DCM architecture (see supporting info).

As for the number of simulated elements, the Miller architecture requires a total of $m^2 + 2m$ MZIs[8], while the proposed DCM require a total of m^2 for the middle stage and $2m(m-1)$ for both the ingress/egress stage MZIs, assuming an MZI-only implementation. However, for DCM both the ingress and egress stages could as well be designed with fixed couplers/power splitting waveguiding systems, therefore the actual number of "active" (i.e., reconfigurable) elements is $\approx m^2$, comparable with the Miller approach.

Note that in our implementation each MZI requires the existence of 2 "active" tunable components (one for each waveguiding "arm" or one that controls tha amplitude and one for the phase of the output signal). In this case the Miller architecture requires $2m^2 + 4m$ active components. However, with the DCM with the multiplier requires two active elements (one phase shifter and one amplifier/attenuator unit), therefore it could be implemented with $2m^2$ active components. Therefore, any scalablity/insertion loss or other implementation related issues are practically dictated and limited by the platform used.

Concerning the number of waveguides m used for representing the problem, it highly depends on mainly two factors: (a) the choice of the technological platform and the related experimental challenges and (b) the choice of the problem, e.g., kernels in solving integral/differential equations, and the required samples that are needed for achieving a certain numerical accuracy. For the first one, assuming a certain platform, one should consider the impact on the signal integrity/energy, i.e., the noise floor that the experimental measuring devices can achieve, and potential path length errors that might occur. For the second factor, if both our kernel and input are band-limited, one can possibly define a lower bound for the number of the waveguides. However, in principle, the number of waveguides depends very much on the nature of the mathematical problem to be solved.

Numerical methods

Rectangular integration

The rectangular integration technique, also known as the midpoint rule, was used to evaluate the integral of Eq. (1). This technique approximates the area under a curve as the sum of rectangles of different widths and heights. The integration domain $[a, b]$ was divided into m subintervals of equal width $\Delta_v = (b-a)/m$. The midpoint was extracted from each subinterval to build the vector $\mathbf{v} = [v_1, v_2, \ldots, v_m]$. This represents the discretized version of the integration variable v. It was assumed that the input electromagnetic signal, $c(u)$, and the solution electromagnetic wave, $x(u)$, were also sampled at these same points so that $\mathbf{u} = \mathbf{v}$. Using this numerical approximation, Eq. (1) can be discretized by replacing the continuous variables u and v with the discrete variables $\mathbf{u}$ and $\mathbf{v}$, respectively, can be rewritten as

$$x(u_i) = c(u_i) + \Delta_v \sum_{j=1}^{m} K(u_i, v_j) x(v_j) \qquad i = 1, \ldots, m \tag{14}$$

where the integral has been replaced by a sum of areas of rectangles with width Δ_v and heights equal to the integrand value at midpoints. Eq. (14) can then be expressed in matrix-vector form as $\mathbf{x} = \mathbf{c} + \mathbf{K}\mathbf{x}$ where the term Δ_v has been incorporated in the kernel matrix $\mathbf{K}$. The latter has been interpreted as a transmission matrix and implemented through an MZI mesh that evaluates numerically the integral of Eq. (1). Other numerical integration techniques such as trapezoidal rule can be used to solve Eq. (1) with the proposed platform.

Finite-difference method

The finite-difference method is a widely used method for the solution of differential equations[52]. The main feature is that a continuous differential equation of the form

$$\mathcal{D}\begin{cases} \frac{d^2 f(x)}{dx^2} + p(x)\frac{df(x)}{dx} + q(x)f(x) = g(x), x \in [x_d, x_u] \\ f(x_d) = a, f(x_u) = b \end{cases} \tag{15}$$

result in the following difference equation

$$\mathcal{D}_\Delta : \begin{cases} r(x_j)\left(f_{j+1} - 2f_j + f_{j-1}\right) + p(x_j)\frac{f_{j+1} - f_{j-1}}{2}(\Delta x) \\ + q(x_j)f_j(\Delta x)^2 = g(x_j)(\Delta x)^2, j \in \{1, \ldots, m\} \\ f_0 = a, f_{m+1} = b \end{cases} \tag{16}$$

The discretization of the differential of first and second order can be either by applying central (CD), forward

(FoD), and backward (BD) differences. As shown in the previous section the resulted difference matrix reads

$$\mathbf{A} = \mathbf{R}(x)\mathbf{A}_{\Delta x} + \mathbf{P}(x)\mathbf{P}_{\Delta x}\,\Delta x + \mathbf{Q}(x)\,(\Delta x)^2 \tag{17}$$

In particular, the second difference matrix reads

$$\mathbf{A}_{\Delta x} = \begin{pmatrix} -2 & 1 & 0 & \dots & 0 \\ 1 & -2 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & 1 & -2 & 1 \\ 0 & \dots & 0 & 1 & -2 \end{pmatrix} \tag{18}$$

where the first and last row implement the FoD and the BD scheme, respectively. This matrix is almost diagonally dominant with real and positive diagonal entries, hence all (or the majority) of the eigenvalues are positive. The matrix $\mathbf{R}(x) = \mathrm{diag}\,(r(x_1), r(x_2), \dots, r(x_m))$ is a diagonal matrix derived by the values of $r(x)$. Similarly, the first difference matrix reads

$$\mathbf{P}_{\Delta x} = \frac{1}{2}\begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ -1 & 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & -1 & 0 & 1 \\ 0 & \dots & 0 & -1 & 0 \end{pmatrix} \tag{19}$$

where the same FoD/BD scheme is used and $\mathbf{P}(x) = \mathrm{diag}\,(p(x_1), p(x_2), \dots, p(x_m))$. The first difference matrix is skew-symmetric with real entries, implying that its eigenvalues are imaginary. The last matrix is a diagonal matrix $\mathbf{Q}(x) = \mathrm{diag}\,(0, q(x_2), \dots, 0)$

The input vector $\mathbf{b}$ for this particular Dirichlet-type boundary conditions reads

$$\mathbf{c} = \begin{pmatrix} g(x_1)(\Delta x)^2 - (1 - p(x_1)\Delta x)a \\ g(x_2)(\Delta x)^2 \\ \vdots \\ g(x_{m-1})(\Delta x)^2 \\ g(x_m) - (1 + p(x_m)\Delta x)b \end{pmatrix} \tag{20}$$

Note that the input vector $\mathbf{c}$ changes depending on the boundary-type problem that is used. In our case the FoD/BD scheme was used as a way to increase the accuracy of the results at the boundaries. Assuming a much denser discretization (larger than the 11 points), then a uniform CD scheme could also be used without affecting the accuracy of the results.

On the convergence of the iteration and GDM

The optimal choice of the scaling factor α_λ is when it is proportional to the minimum and maximum eigenvalues of the operator, i.e., $\alpha_\lambda = \frac{2}{\lambda_1 + \lambda_n}$, where λ_1 and λ_n are the largest and smallest eigenvalues of the matrix. Knowledge of the operator's larger and smallest eigenvalues requires certainly some calculations. In order to avoid these calculations, one can utilize some of the eigenvalue inclusion theorems that are available, such as Schur's or Gershgorin's theorems[50]. In the case where a different scaling factor is used the convergence will not be optimum, hence cannot be described by Eq.(13). However, the convergence of the method to the desired level of accuracy can happen well before the bound is met even for optimum scaling factor.

Finally, the iteration used above have been invented multiple times carrying a multitude of names such as Richardson's iteration[42], Landweber iteration[43], steepest or GDM[44]. An interesting feature with regards to its convergence should be mentioned here. Briefly, for a given positive semidefinite matrix $\mathbf{A} \in \mathbb{C}^{m\times m}$ we have that the iteration $x_{n+1} = \alpha\mathbf{c} + (\mathbf{I} - \alpha\mathbf{A})x_n$ converges as long as $\mathbf{I} - \alpha\mathbf{A}$ has norm <1 and its actual convergence depends on how much this norm is <1. Let $\lambda_1 \le \lambda_2 \le \dots \le \lambda_m$ are the eigenvalues of $\mathbf{A}$, then the norm of the kernel minimizes when $\alpha = \alpha_\lambda = \frac{2}{\lambda_1 + \lambda_m}$.

For this type of scaling value the $\mathbf{x}_n$ converges to the desired solution $\mathbf{x}$ when the norm is less than the desired accuracy level ε, i.e, after n iterations we have that the value is bounded by

$$\begin{aligned} ||\mathbf{x} - \mathbf{x}^n|| &= ||(\mathbf{I} - \alpha_\lambda\mathbf{A})^n\mathbf{x}|| \le ||(\mathbf{I} - \alpha_\lambda\mathbf{A})^n||\,||\mathbf{x}|| \\ &= (1 - \alpha_\lambda)^n||\mathbf{x}|| \le e^{-\alpha_\lambda n}||\mathbf{x}|| \end{aligned}$$

and for $\frac{||\mathbf{x}-\mathbf{x}^n||}{||\mathbf{x}||} \le \varepsilon$ we finally have that the iteration will converge after

$$n = \frac{1}{\alpha_\lambda}\ln(1/\varepsilon) = \frac{1}{2}(\kappa(\mathbf{A}) + 1)\ln(1/\varepsilon) \tag{21}$$

iterations. Assuming that each iteration requires time t then we reach Eq. (13). For the case of GDM where we use the normal matrix $\mathbf{A}^*\mathbf{A}$ then the condition number is modified accordingly.

As a side note, in realistic implementations (assuming noise and other intrinsic system imperfections) the kernel matrix is followed by $(\mathbf{I} - \mathbf{A}^*\mathbf{A} + \delta\mathbf{N})$, where $\mathbf{N}$ is a random noise matrix while δ is a small perturbation. In the ideal case we have $\delta \to 0$. Assuming a certain stochastic dependence of the noise matrix the above generic quantity could be simplified to be $(\mathbf{I} - \mathbf{A}^*\mathbf{A} + \delta\mathbf{I})$, implying that the Kernel represents a Tikhonov regularization scheme[56,57], hence the quantity $(\mathbf{A}^*\mathbf{A})^{-1}$ has always an answer.

Acknowledgements
This work was supported in part by the US Air Force Office of Scientific Research (AFOSR) Multidisciplinary University Research Initiative (MURI) grant number FA9550-17-1-0002. The authors also thank Mark Saffian for the valuable discussions on the AWR® Microwave Office simulations.

Author contributions
N.E. initiated the research, acquired the funds, and presented the idea for solving mathematical equations with the Miller architecture. The DCM architecture was proposed by N.E., developed by D.C.T., and further refined by D.C.T. and B.E. D.C.T. performed all system-level simulations. M.J.M. implemented the Miller algorithm and M.J.M. and B.E. developed the code for the IE solution. D.C.T. developed the code for the DE problems and extended the usage of metadevices for inverting general rectangular matrices. All authors discussed and interpreted all the results reported in this manuscript. D.C.T. wrote the first draft of the manuscript and all authors discussed, commented, and edited the manuscript.

Data availability
All data (simulation files, raw data, figures, etc.) are available by the corresponding authors upon reasonable request.

Code availability
All codes (Matlab scripts etc.) are available by the corresponding authors upon reasonable request.

Competing interests
A non-provisional patent application has been filed on some of the ideas reported here, and more patent applications on other aspects of this work may be filed in the near future. N.E. is a strategic scientific advisor/consultant to Meta Materials, Inc. No other conflict of interest is present.

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41377-022-00950-1.

Received: 19 November 2021 Revised: 15 July 2022 Accepted: 2 August 2022
Published online: 07 September 2022

References

1. Trefethen, L. & Bau, D. III. *Numerical Linear Algebra* (SIAM, Philadelphia, PA, 1997).
2. Caulfield, H. J. & Dolev, S. Why future supercomputing requires optics. *Nat. Photonics* **4**, 261–263, https://doi.org/10.1038/nphoton.2010.94 (2010).
3. Solli, D. R. & Jalali, B. Analog optical computing. *Nat. Photonics* **9**, 704–706, https://doi.org/10.1038/nphoton.2015.208 (2015).
4. Wang, Z. et al. Resistive switching materials for information processing. *Nat. Rev. Mater.* **5**, 173–195, https://doi.org/10.1038/s41578-019-0159-3 (2020).
5. Silva, A. et al. Performing mathematical operations with metamaterials. *Science* **343**, 160–163, https://doi.org/10.1126/science.1242818 (2014).
6. Pors, A., Nielsen, M. G. & Bozhevolnyi, S. I. Analog computing using reflective plasmonic metasurfaces. *Nano Lett.* **15**, 791–797, https://doi.org/10.1021/nl5047297 (2015).
7. Wetzstein, G. et al. Inference in artificial intelligence with deep optics and photonics. *Nature* **588**, 39–47 (2020).
8. Harris, N. C. et al. Linear programmable nanophotonic processors. *Optica* **5**, 1623–1631, https://doi.org/10.1038/s41586-020-2973-6 (2018).
9. Estakhri, N. M., Edwards, B. & Engheta, N. Inverse-designed metastructures that solve equations. *Science* **363**, 1333–1338, https://doi.org/10.1126/science.aaw2498 (2019).
10. Le Gallo, M. et al. Mixed-precision in-memory computing. *Nat. Electronics* **1**, 246–253, https://doi.org/10.1038/s41928-018-0054-8 (2018).
11. Mead, C. How we created neuromorphic engineering. *Nat. Electronics* **3**, 434–435 (2020).
12. Zidan, M. A. et al. A general memristor-based partial differential equation solver. *Nat. Electronics* **1**, 411–420, https://doi.org/10.1038/s41928-020-0448-2 (2018).
13. Sun, Z. et al. Solving matrix equations in one step with cross-point resistive arrays. *Proc. Natl. Acad. Sci. USA* **116**, 4123–4128, https://doi.org/10.1073/pnas.1815682116 (2019).
14. Vellekoop, I. M. & Mosk, A. P. Focusing coherent light through opaque strongly scattering media. *Opt. Lett.* **32**, 2309–2311 (2007).
15. Zhou, Y., Zheng, H., Kravchenko, I. I. & Valentine, J. Flat optics for image differentiation. *Nat. Photon.* **14**, 316–323, https://doi.org/10.1364/OL.32.002309 (2020).
16. Imani, M. F. et al. Review of metasurface antennas for computational microwave imaging. *IEEE Trans. Antennas Propag.* **68**, 1860–1875, https://doi.org/10.1109/TAP.2020.2968795 (2020).
17. del Hougne, P. & Lerosey, G. Leveraging chaos for wave-based analog computation: demonstration with indoor wireless communication signals. *Phys. Rev. X* **8**, 041037, https://doi.org/10.1103/PhysRevX.8.041037 (2018).
18. Cordaro, A. et al. High-index dielectric metasurfaces performing mathematical operations. *Nano Lett.* **19**, 8418–8423, https://doi.org/10.1021/acs.nanolett.9b02477 (2019).
19. Matthès, M. W., del Hougne, P., de Rosny, J., Lerosey, G. & Popoff, S. M. Optical complex media as universal reconfigurable linear operators. *Optica* **6**, 465–472, https://doi.org/10.1364/OPTICA.6.000465 (2019).
20. Feldmann, J. et al. Parallel convolutional processing using an integrated photonic tensor core. *Nature* **589**, 52–58, https://doi.org/10.1038/s41586-020-03070-1 (2021).
21. Shen, Y. et al. Deep learning with coherent nanophotonic circuits. *Nat. Photon.* **11**, 441 (2017).
22. Hughes, T. W., England, R. J. & Fan, S. Reconfigurable photonic circuit for controlled power delivery to laser-driven accelerators on a chip. *Phys. Rev. Appl.* **11**, 64014, https://doi.org/10.1103/PhysRevApplied.11.064014 (2019).
23. Haffner, C. et al. All-plasmonic Mach-Zehnder modulator enabling optical high-speed communication at the microscale. *Nat. Photon.* **9**, 525–528, https://doi.org/10.1038/nphoton.2015.127 (2015).
24. Wu, J. et al. Compact tunable silicon photonic differential-equation solver for general linear time-invariant systems. *Opt. Express* **22**, 26254–26264, https://doi.org/10.1364/OE.22.026254 (2014).
25. Wu, K., Soci, C., Shum, P. P. & Zheludev, N. I. Computing matrix inversion with optical networks. *Opt. Express* **22**, 295–304, https://doi.org/10.1364/OE.22.000295 (2014).
26. Taballione, C. et al. 8×8 reconfigurable quantum photonic processor based on silicon nitride waveguides. *Opt. Express* **27**, 26842, https://doi.org/10.1364/OE.27.026842 (2019).
27. Bogaerts, W. et al. Programmable photonic circuits. *Nature* **586**, 207–216, https://doi.org/10.1038/s41586-020-2764-0 (2020).
28. Low, T. & Avouris, P. Graphene plasmonics for terahertz to mid-infrared applications. *ACS Nano* **8**, 1086–1101, https://doi.org/10.1021/nn406627u (2014).
29. Morea, M., Zang, K., Kamins, T. I., Brongersma, M. L. & Harris, J. S. Electrically tunable, CMOS-compatible metamaterial based on semiconductor nanopillars. *ACS Photon.* **5**, 4702–4709, https://doi.org/10.1021/acsphotonics.8b01383 (2018).
30. Fan, Y. et al. Graphene plasmonics: a platform for 2D optics. *Adv. Opt. Mater.* **7**, 1–14, https://doi.org/10.1002/adom.201800537 (2019).
31. Reck, M., Zeilinger, A., Bernstein, H. J. & Bertani, P. Experimental realization of any discrete unitary operator. *Phys. Rev. Lett.* **73**, 58–61, https://doi.org/10.1103/PhysRevLett.73.58 (1994).
32. Miller, D. A. B. Self-configuring universal linear optical component [Invited]. *Photon. Res.* **1**, 1–15, https://doi.org/10.1364/PRJ.1.000001 (2013).
33. Miller, D. A. B. Sorting out light. *Science* **347**, 1423–1424, https://doi.org/10.1126/science.aaa6801 (2015).
34. Clements, W. R., Humphreys, P. C., Metcalf, B. J., Kolthammer, W. S. & Walmsley, I. A. Optimal design for universal multiport interferometers. *Optica* **3**, 1460–1465, https://doi.org/10.1364/OPTICA.3.001460 (2016).
35. Zangeneh-Nejad, F., Sounas, D. L., Alù, A. & Fleury, R. Analogue computing with metamaterials. *Nat. Rev. Mater.* **6**, 207–225, https://doi.org/10.1038/s41578-020-00243-2 (2021).
36. Psaltis, D., Casasent, D. & Carlotto, M. Iterative color-multiplexed, electro-optical processor. *Opt Lett.* **4**, 348–350, https://doi.org/10.1364/OL.4.000348 (1979).
37. Cederquist, J. & Lee, S. H. The use of feedback in optical information processing: a review. *Appl. Phys.* **18**, 311–319, https://doi.org/10.1007/BF00899683 (1979).
38. Akins, R. P., Athale, R. A. & Lee, S. H. Feedback in analog and digital optical image processing: a review. *Opt. Eng.* **19**, 193347 (1980).

39. Barrett, R. et al. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods* (SIAM, 1994).
40. Goodman, J. W. & Song, M. S. Performance limitations of an analog method for solving simultaneous linear equations. *Appl. Opt.* **21**, 502–506, https://doi.org/10.1364/AO.21.000502 (1982).
41. Penrose, R. A generalized inverse for matrices. *Math. Proc. Camb. Philos. Soc.* **51**, 406–413, https://doi.org/10.1017/S0305004100030401 (1955).
42. Richardson, L. F. The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam. *Philos. Trans. Royal Soc. A* **210**, 307–357, https://doi.org/10.1098/rsta.1911.0009 (1911).
43. Landweber, L. An iteration formula for fredholm integral equations of the first kind. *Am. J. Math.* **73**, 615–624 (1951). 10.2307/2372313.
44. Shewchuk, Jonathan, R. *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*. Tech. Rep., Carnegie Mellon University (1994).
45. Harrow, A. W., Hassidim, A. & Lloyd, S. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.* **103**, 150502, https://doi.org/10.1103/PhysRevLett.103.150502 (2009).
46. Pozar, D. M. *Microwave engineering; 3rd edn*. (Wiley, Hoboken, NJ, 2005).
47. Miller, D. A. B. Setting up meshes of interferometers - reversed local light interference method. *Opt. Express* **25**, 29233–29248, https://doi.org/10.1364/OE.25.029233 (2017).
48. Cadence. AWR Microwave Office (2020).
49. Abramowitz, M. & Stegun, I. A. *Handbook of mathematical functions with formulas, graphs, and mathematical tables* (Washington: U.S. Government Printing Office, 1964).
50. Kreyszig, E. *Advanced Engineering Mathematics* (Hoboken: John Wiley & Sons, Inc., 1999), 8th edn.
51. Zangeneh-Nejad, F. & Fleury, R. Topological analog signal processing. *Nat. Commun.* **10**, 2058, https://doi.org/10.1038/s41467-019-10086-3 (2019).
52. LeVeque, R. *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems* (Philadelphia: SIAM, USA, 2007).
53. Jahani, S. et al. Controlling evanescent waves using silicon photonic all-dielectric metamaterials for dense integration. *Nat. Commun.* **9**, 1893, https://doi.org/10.1038/s41467-018-04276-8 (2018).
54. Mia, M. B. et al. Exceptional coupling in photonic anisotropic metamaterials for extremely low waveguide crosstalk. *Optica* **7**, 881–887, https://doi.org/10.1364/OPTICA.394987 (2020).
55. Song, T. et al. Ultracompact photonic circuits without cladding layers. *Phys. Rev. X* **12**, 011053, https://doi.org/10.1103/PhysRevX.12.011053 (2022).
56. Kleinman, R. E. & Van den Berg, P. M. A modified gradient method for two-dimensional problems in tomography. *J. Comput. Appl. Math.* **42**, 17–35, https://doi.org/10.1016/0377-0427(92)90160-Y (1992).
57. Jensen, J. S. & Sigmund, O. Topology optimization for nano-photonics. *Laser Photon. Rev.* **5**, 308–321, https://doi.org/10.1002/lpor.201000014 (2011).