Supplementary material for: Integrated photonic multigrid solver for partial differential equations

Timoteo Lee[†1,2], F. Brückerhoff-Plückelmann[†1], Jelle Dijkstra[1], W. Pernice[1], and Jan M. Pawlowski[2]

[1] *Kirchhoff-Institute for Physics, University of Heidelberg; Heidelberg, 69120, Germany.*
[2] *Institute for Theoretical Physics, University of Heidelberg; Heidelberg, 69120, Germany*

Contents

[†]: These authors contributed equally.

Materials and Methods

S1. Photonic matrix-vector multiplications

In a specialized solver within the mixed-precision photonic multigrid (MPPMG) framework, the weights are ideally sent to the photonic processors once at the beginning, and only the inputs are sent to the processors during the calculations. Currently, our photonic processor is optimized for fixed two-dimensional kernel operations. Hence, we map the sparse matrix-vector multiplications (MVMs) into a few kernel operations. The inherent equivalent bit-resolution of the photonic processor is approximately 4 bits. Thus, we round the matrix elements to 4-bits discretization in the QQAO problem, which reduces the number of different kernel operations. The diagonal terms in this matrix have the shape

$$\mathrm{Diag}_j = \frac{1}{2}\left(\frac{2}{h^2} + y_j^2 + 2\mu y_j^4\right), \tag{1}$$

and for the chosen parameters, the 4-bit discretization leads to the matrix diagonal elements shown in Figure S1. A similar pattern emerges for the matrix of the second level of the multigrid algorithm. Hence, the MVM on the photonic processor is a chain of kernel operations. For the PC problem, we do not round the matrix elements, because the matrix only has the element values -4, 1, and -1. For the internal points, we only need one 2D next-neighbour kernel, and at the boundary conditions, the identity kernel. Hence, we have two kernel operations required for one MVM. Finally, we map the input to integer values in the range $[-m, m]$, where $m := 2^{8-1} - 1$ and rescale the output accordingly after the MVM.

S2. Details on the emulation

We describe one emulated photonic MVM with 8-bit digital-to-analog and analog-to-digital converters (DACs and ADCs). The steps are

1. The matrix and input are rescaled into the range [-m, m] by dividing them by their respective maximum absolute value, and multiplying them by m. Then, we round the values into integer values to emulate the behaviour of the 8-bit DACs.

2. Noise is sampled from a normal distribution with standard deviation $0.045 \times m$, and it is added to the matrix and input to emulate the analog noise.

3. The MVM between the noisy, discretized matrix and input is performed.

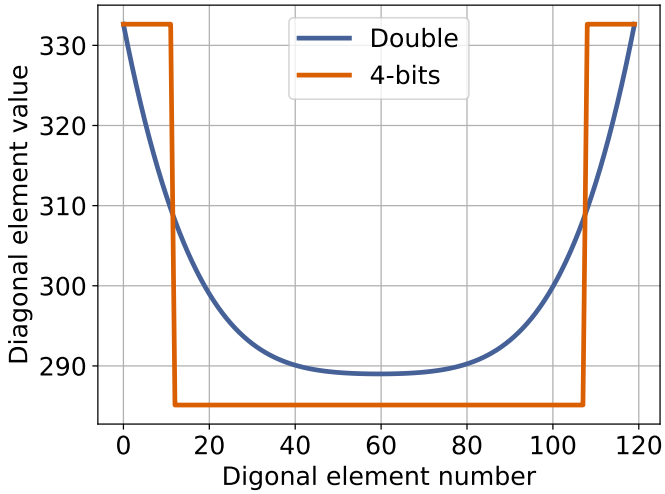


Figure S1. Rounding scheme for the QQAO matrix. We plot the values of the diagonal elements of the first-level QQAO matrix with double-precision and 4-bit fixed-point representation. We note that we apply the 4-bit discretization to the whole matrix, where the off-diagonal elements are always negative for the first level.

4. At this point, most output values are in the range [-m^2, m^2]. We divide the output by m, we clip all values outside of the range [-m, m], and round the values into integer values to emulate the 8-bit ADCs.

5. The result is rescaled back to its correct scale by dividing this output by m and multiplying it by the original maximum absolute values of the matrix and input.

Here, we call the calculations done on the digital twin (Emulated) "Noisy 8-bits". For the fixed-point precision emulations, we perform the same procedure without noise.

S3. Multigrid implementation

We use algebraic multigrid (AMG) methods, where no knowledge of the grid is required, and only the mathematical properties of the matrix are used to construct the multigrid solver. AMG solvers are, hence, better "black-box" solvers compared to geometric multigrid solvers. As mentioned in the main text, we use the library PyAMG to construct the smoothed aggregation algebraic multigrid solvers. We run the functions from this library with only the matrix as input to construct the multigrid solvers. The resulting multigrid object contains the transfer operators, namely, the restrictor and prolongator, which are related by

$$R = P^\dagger, \tag{2}$$

and the coarse matrix has the shape

$$A_c = RAP. \tag{3}$$

For the LQCD problem, we use the function "fit_candidates" and "richardson_prolongation_smoother" to construct the transfer operators. The multigrid iteration illustrated in Figure 1C follows the steps:

1. The Richardson smoother is applied to the current solution, resulting in a new current solution estimate x with a smooth error.

2. The residual $r = b - Ax$ is calculated.

3. The residual is transferred to the coarser grid through $r_c = Rr$, where the subindex c stands for coarse.

4. The coarse problem $A_c e_c = r_c$ is solved, potentially, by using an even coarser representation of the system.

5. The solution e_c is transferred to the finer grid through $e = Pe_c$.

6. The fine solution is updated $x \leftarrow x + e$.

7. (For double-precision solvers:) The Richardson smoother is applied to the updated solution.

We call the combination of steps 2 and 3, and the combination of steps 5 and 6 "transfer".

Supplementary Text

S4. Measurements

While photonic processors can achieve ultra-low latency, they are affected by analog noise. To visualize the quality of the MVMs, we plot the measured values of the dot products within the MVMs on the photonic processor and on the emulator as a function of the respective expected values in Figure S2. For the QQAO matrix, we observe that the error is not fully Gaussian and get an L_2 relative error

$$\epsilon_{L_2} = \frac{\sqrt{\sum_i^{100} |y_{i,expected} - y_{i,measured}|^2}}{\sqrt{\sum |y_{i,expected}|^2}} \tag{4}$$

of 7.7%. For the PC matrix, we get an L_2 relative error of 17.8%, where there is a systematic error, which is due to a problem with the calibration of the map between the electrical output and the output results of the dot products. This issue explains the difference in performance between the photonic and the emulated MPPMG solver for this problem. Nevertheless, the MPPMG solver remains robust and converges to the solution with the same accuracy.

S5. Approximating the spectral radius

The Richardson's smoother has the tunable parameter ω. The smoother affects the error according to

$$e_{k+1} = (I_N - \omega A)e_k =: Me_k. \tag{5}$$

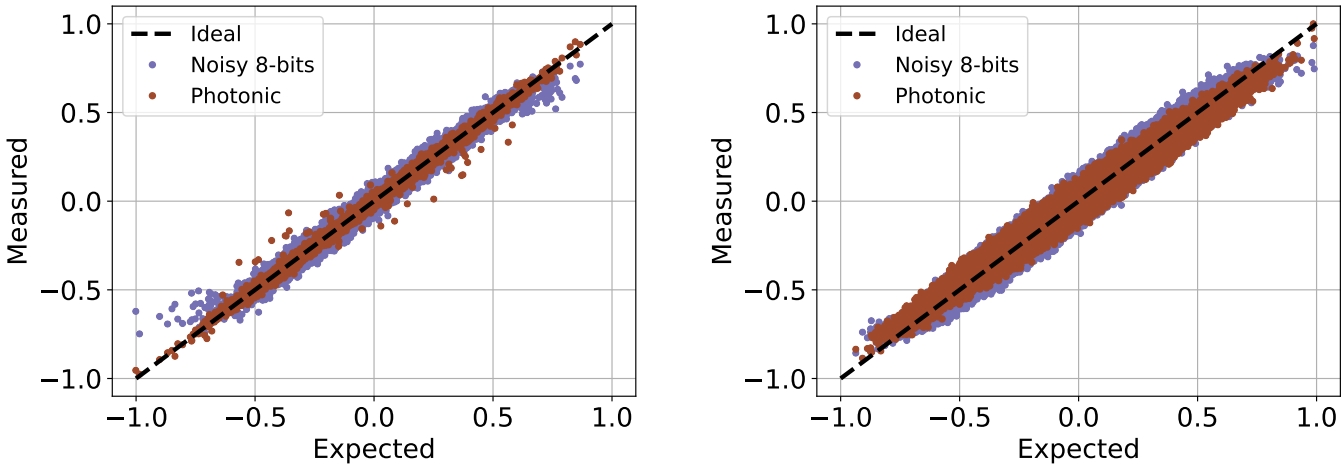
The decomposition of the initial error in the eigenbasis of A shows that

$$e_{k+1} = \sum_i (1 - \omega\lambda_i)^{k+1} c_i v_i, \tag{6}$$

where (λ_i, v_i) are the eigenpairs of A, and

$$e_0 = \sum_i c_i v_i. \tag{7}$$

Hence, $\omega = 1/\rho$, where $\rho = max(|\lambda_i|)$ is the spectral radius, is a common choice. With this, the high-modes are quickly eliminated, while the low-modes are slowly dampened.



(a) QQAO (b) PC

Figure S2. Dot product measurements. For each problem, we first perform the same random 100 MVMs with the digital processor, photonic processor, and the emulator. We take the maximum absolute value between all these 300 MVMs to rescale the result into the range [-1,1], and we plot the measured values as a function of the expected values.

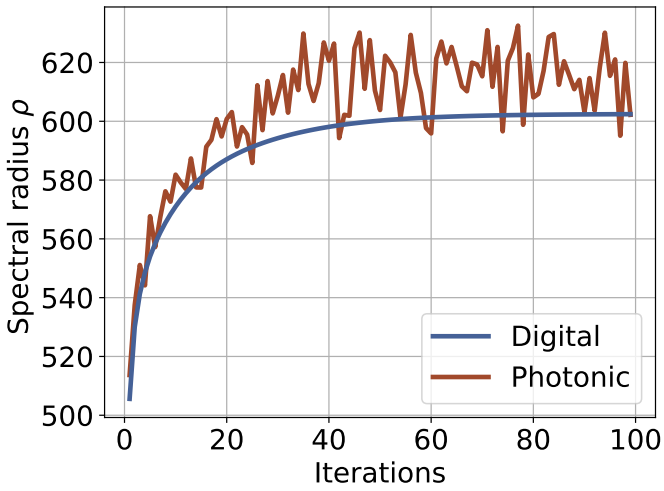


Figure S3. Spectral radius of the QQAO matrix. We approximate the spectral radius with MVMs in the digital and the optical domain using the power iteration method. We observe that after 80-100 iterations, the digital result converges to the spectral radius.

We approximate the spectral radius by using two power iterations, and we use the same value for all the solvers to allow a fair comparison. We offload this procedure to the optical domain, and we do not consider these two MVMs in the results. In Figure S3, we plot the spectral radius as function of the number of power iterations. We observe that the optical result fluctuates around the result of the digital calculations. Nevertheless, two iterations in the optical domain lead to a good approximation for the problems in this paper.

S6. One multigrid iteration

We present further results, which illustrate the performance of the MPPMG solver by looking at the error and its spectral decomposition at different stages in the multigrid iteration. We use here the quantum harmonic oscillator (QHO) ($\mu = 0$, $h = 0.13$, $y \in [-8, 8]$), and a simple geometric two-grid solver for illustrative reasons. The first grid has 120 points, and the second grid has 60 points. We use a full-weighting restriction operator, and a linear interpolation operator to transfer the information between the grids. We damp the high-frequency modes by applying the photonic smoother, which allows the transfer of information between the grids. After a full multigrid iteration, all frequency components are dampened, including the numerically challenging low-frequency modes.

For the results in Figure S4, we manually choose an initial guess such that its error equally contains all frequency modes, and we find the coefficients of the spectral decomposition through a fit, where we order the eigenpairs from the lowest eigenvalue to the highest one.

S7. Total number of operations

We reiterate that we make the approximation that MVMs are the only non-negligible computations in terms of computation time and energy consumption. We count the number of operations according to

$$\#Ops = \#MVMs \times (2 \times nnz - V)\,, \tag{8}$$

where nnz is the number of non-zero entries, and V is the dimension of the matrix A. During one iteration of the multigrid preconditioned solver, we perform one MVM and one multigrid iteration. As explained in Section S3 and in the main text, the multigrid iteration consists of 4 smoothing operations and one residual calculation at each level, except at the coarsest level. Furthermore, for the PC and LQCD problem, we have an outer iterative refinement (IR) loop, which requires one digital MVM. For the QQAO and the PC problem, we offload the MVM from the residual calculation to the optical domain, which increases the performance of the MPPMG solvers. Hence, for each iteration of the solver, we count 5 optical MVMs per level and one digital MVM. For the LQCD problem, due to numerical stability, we choose to calculate the residual in the digital domain, which leads to 4 optical MVMs and 2 digital MVMs at each iteration of the solver. In Figure S5, we plot the total number of operations(OPs) required to converge to the same desired high accuracy for the different solvers applied to the different problems. In Figure S6, we observe

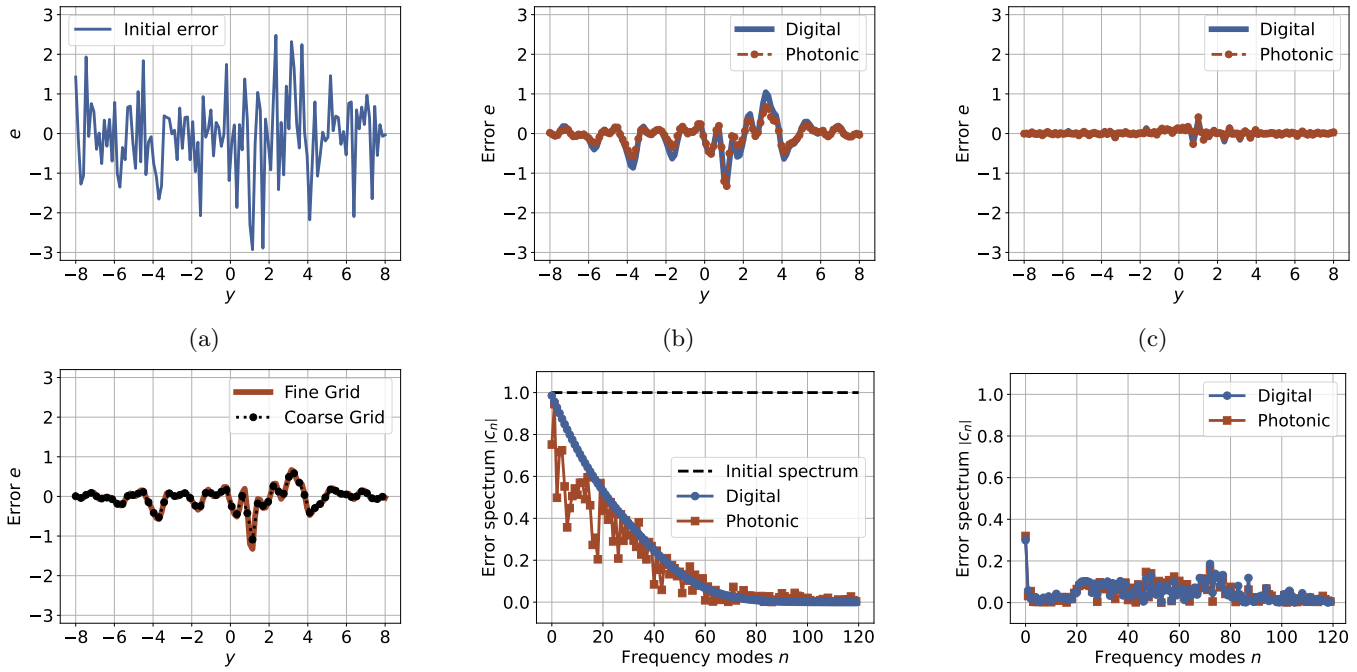


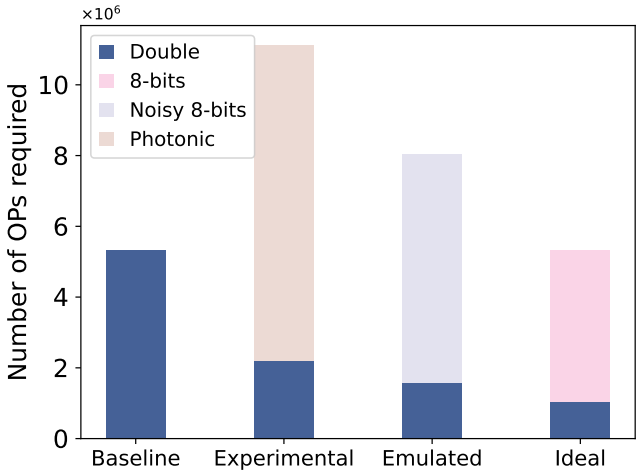
Figure S4. Visualization of one multigrid iteration. We apply the smoother 4 times to an initial guess with an error shown in a), and we get the result in b) with its spectral decomposition in e). The result in the photonic domain is smooth enough, and we transfer the information to a coarser grid without considerable loss, as seen in d). Then, we solve the coarse problem, and we use the result to improve the solution in the fine grid. This is the last step of one full asymmetrical two-grid iteration leading to an error seen in c) with its spectral decomposition in f).

that offloading the MVM in the residual calculation to the optical domain only slightly increases the total number of operations for the QQAO problem. Hence, in this case, the use of a photonic transfer reduces the number of double-precision operations. However, for the LQCD problem, the use of photonic transfer considerably increases the total number of iterations required for convergence, and it is not fully numerically stable. In the end, ideal algorithmic choices will depend on the design of the specialized hybrid hardware.

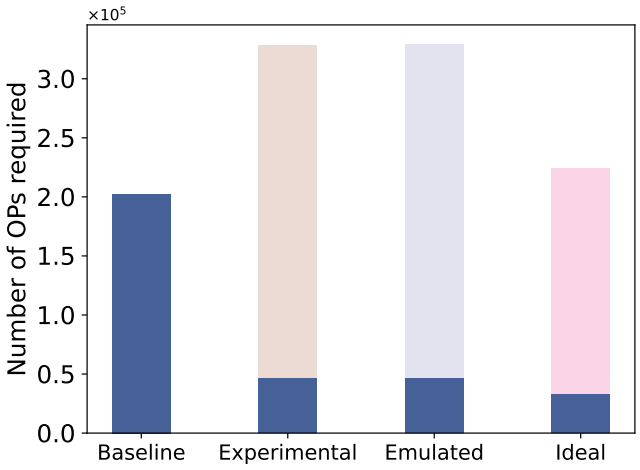
S8. Robustness of the MPPMG

The proposed MPPMG solvers achieve high-accuracy solutions for the three problems, despite most calculations being performed in the low-accuracy analog domain. To further examine the robustness of the MPPMG solver, we benchmark emulated solvers on the QQAO problem using matrix-vector multipliers with different fixed-point precisions and different levels of noise. We plot the respective relative errors in Figure S7a for the different matrix-vector multipliers. The noisy operations are 8-bit operations with noise applied to both the matrix and the input, as described in Section S2. Then, we plot the behaviour of the MPPMG eigensolvers in Figure S7b. Here, we observe that the noisy MPPMG solvers behave similarly to the fixed-point ones for this problem, with a considerable difference for the 3-bit and its equivalent noisy solver.

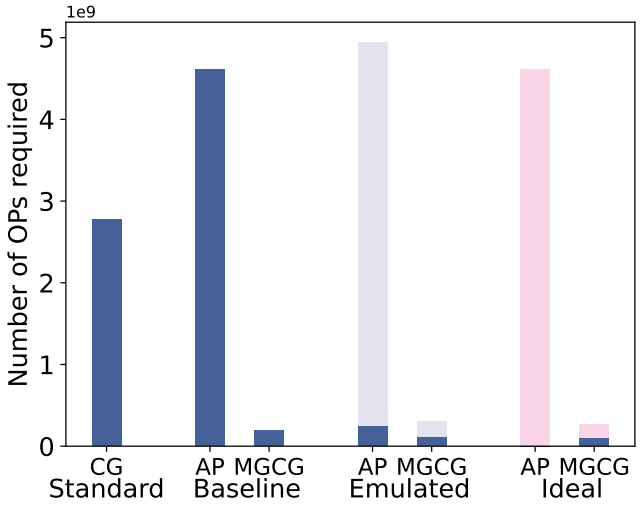
In Figure S8a, we plot the reduction of double-precision operations for different fixed-point matrix-vector multipliers. The photonic processor behaves similarly to an emulated 4-bit MPPMG solver, and higher reductions in double-precision operations can be achieved by lowering noise levels. Hence, the better the analog bit resolution, the more numerically stable the MPPMG solvers are. However, for specific problems, we can trade off MVM accuracy for overall speedup and energy-efficiency gains. Additionally, we plot the same for the PC problem in Figure S8b. Here, we observe that the photonic processor performs similarly to a 3-bit solver, which is explained by the systematic error seen in Figure S2b.



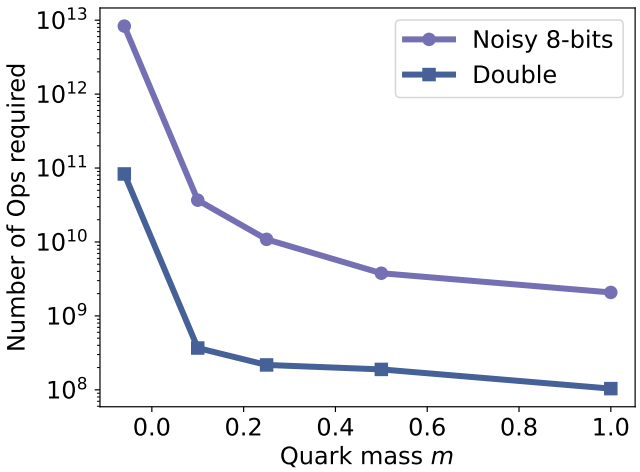
(a) PC



(b) QQAO



(c) LQCD



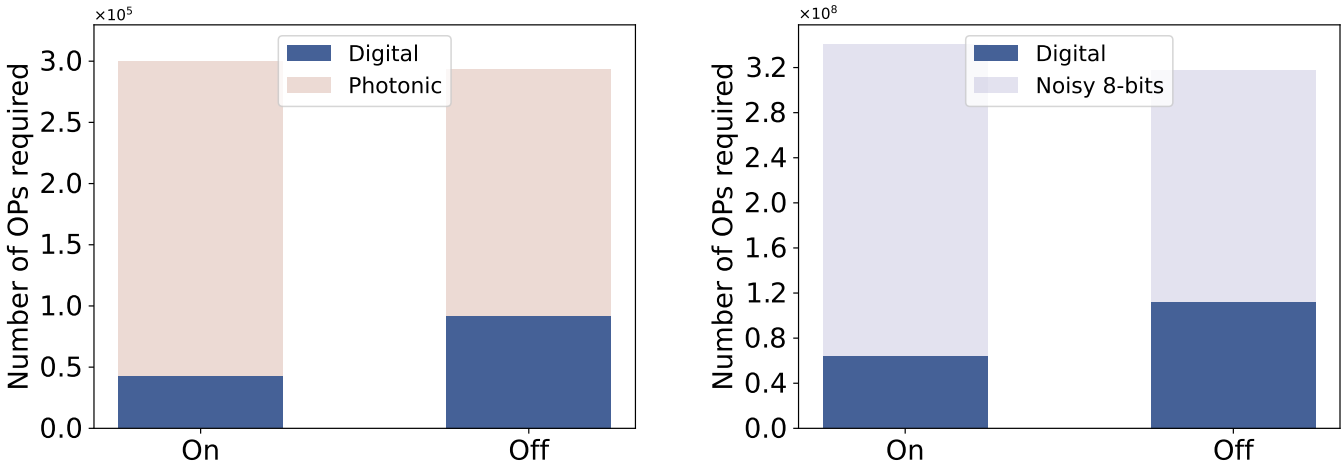
(d) Hybrid IR for LQCD

Figure S5. Complementary results to Figures 3 and 4: Total number of operations. In the main text, we only consider the number of digital operations required for convergence for simplicity. Here, we plot the total number of operations, which includes the analog operations. While the total number of operations increases, the number of digital operations is reduced, leading to a performance boost thanks to the low-latency and energy efficiency of photonic processors. c) Here, we can observe more details behind the MPPMG solver with the adaptive scheme. During the adaptive part (AP), we use the photonic smoother and IR steps. The adaptive scheme constructs a multigrid preconditioner that reduces the number of iterations by two orders of magnitude. However, the cost of the adaptive part hinders its application to different problems. This is especially true for problems with only one right-hand side, which is relevant during the sampling process in LQCD calculations. d) Here, we dissect the result in Fig. 4B. There, we plot the sum of both digital and analog OPs. The result of the MPPMG solver for the different masses is qualitatively identical to the result in c), which is for $m = -0.063$.

S9. Details on the iterative refinement loops

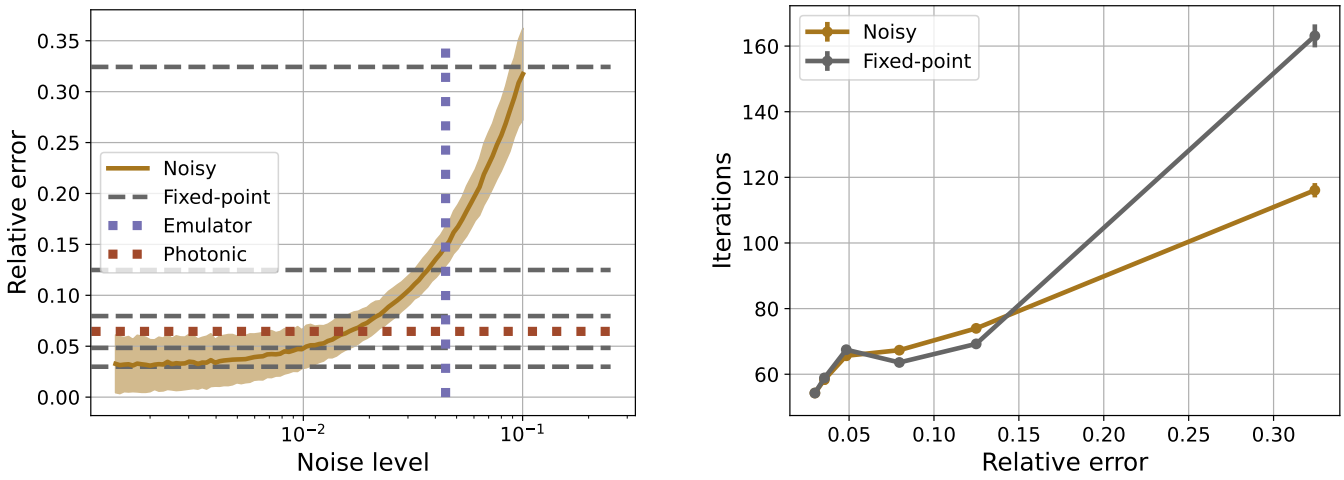
The iterative refinement (IR) method resets the accumulation of rounding errors and serves as a mixed-precision method by itself. This method requires empirically choosing the parameters of the inner solver. We illustrate this using an example shown in Figure S9a. In this example, we use the Richardson smoother as the inner solver to solve the QHO problem, and we offload the MVMs to the optical domain. Without the IR method, the progress of the solver stagnates at some value of the residual norm. Therefore, we apply an IR strategy, where we run the optical solver 100 times, apply one outer IR step, and then alternate between 10 solver steps and one outer IR step.

When using the MPPMG solver as the inner solver, the two parameters of the main solver, namely conjugate gradient (CG), are the target residual norm and the max number of iterations, which are defined as rtol and maxiter in SciPy, respectively. In this paper, we aim to reduce the number of digital MVMs while keeping these two parameters



(a) QQAO (b) LQCD

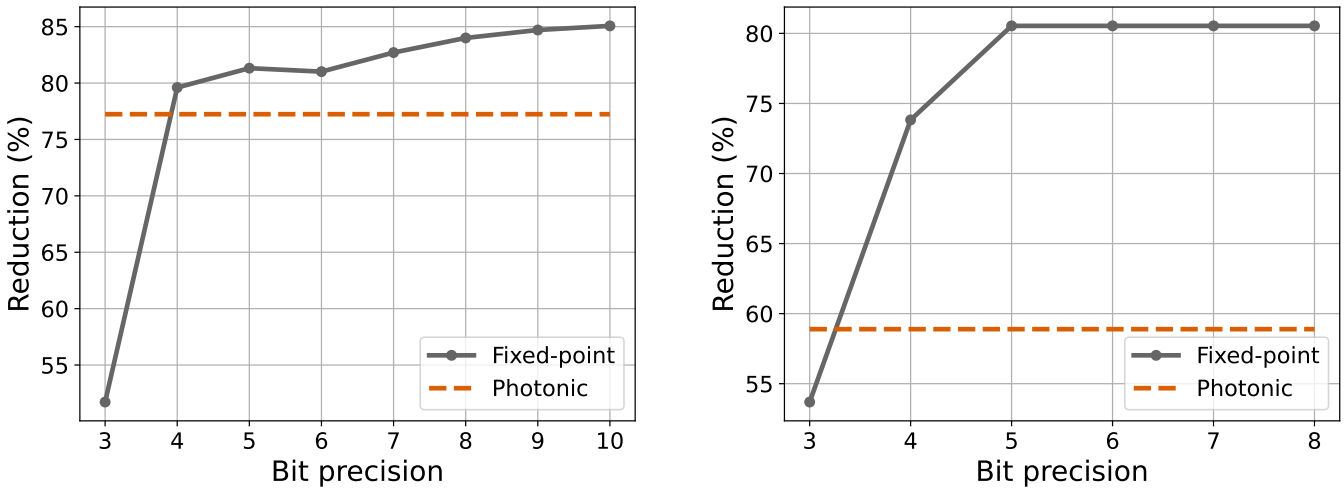
Figure S6. Number of OPs required with and without photonic transfer. a) We take an average of 10 calculations with different random initial guesses. b) We take an average of 100 calculations, where we use the double-precision adaptive scheme used in Figure 4C. We note that the calculations for one gauge links configuration are not numerically stable with the use of photonic transfer. For this configuration, the convergence within the maximum number of iterations depends on the result of the adaptive scheme. Here, we run the adaptive scheme until the resulting multigrid preconditioner leads to convergence within a reasonable number of iterations.



(a) Relative error for different matrix-vector multipliers (b) Iterations required for different emulated MPPMG

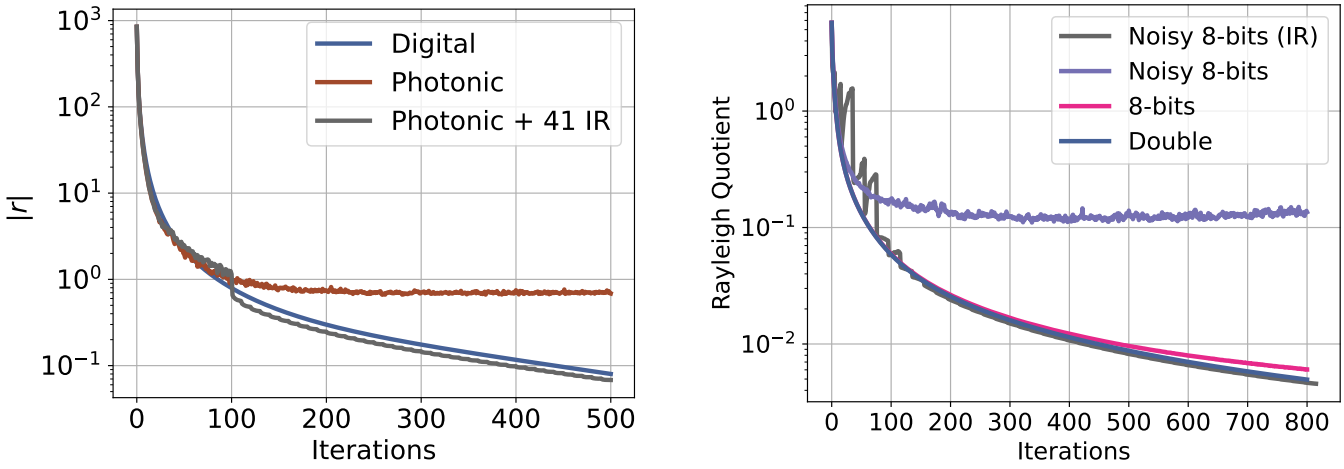
Figure S7. Robustness test of MPPMG eigensolver on the QQAO. a) We perform 1000 random MVMs, and calculate the average relative error for the different matrix-vector multipliers, where the relative error is the norm of the difference between the measured and expected vector, divided by the norm of the expected vector. The fixed-point lines are 3 to 7 bits from top to bottom. For this plot, we rescale the noise level such that it represents the noise strength for MVMs with values in the range [-1,1]. b) We solve the QQAO problem with 100 different random initial guesses and plot the average number of iterations as a function of the relative error of the respective matrix-vector multipliers in Table S1.

fixed during the solving process. The sources of digital MVMs are the MVMs required for the IR outer loops and the MVMs required at each iteration of the solver. Increasing the target residual norm reduces the number of outer IR loops, at the potential cost of increasing the number of inner-solver iterations. We control this increase through the max number of iterations. For the PC problem, we use rtol = 1e-3 and maxiter = 4, and for the LQCD problem, rtol = 1e-1 and maxiter = 8.



(a) QQAO (b) PC

Figure S8. Reduction of double-precision operations for different emulated MPPMG solvers on the QQAO and the PC compared to the experimental MPPMG solvers. a) We take the average of 100 calculations with different initial guesses. b) We take 0 as the standard initial guess and perform the calculations once. For the result of the photonic processor, we take the average of 100 calculations with the same initial guess.



(a) (b)

Figure S9. The iterative refinement method. We plot the evolution of two different calculations, where we use the IR method. a) We solve a linear equation with the QHO matrix and a random right-hand side (RHS) b and plot the residual norm as a function of the number of iterations. b) We plot the Rayleigh quotient for one vector as a function of the number of smoothing steps for the different setups in Figure 4C.

S10. Details on the adaptive multigrid method

We build the multigrid preconditioner for the LQCD problem using an adaptive scheme. In this procedure, we approximately compute a space spanned by the eigenvectors with the smallest eigenvalues. For this, we compute 8 vectors rich in low-frequency modes as explained in the Methods Section. Here, we explain how the IR method is applied to this procedure for the emulated solver. Due to empirical observations, we choose to perform 5 smoothing steps, an IR step, 10 smoothing steps, an IR step, and then 40 loops of 20 smoothing steps followed by an IR step. Hence, we perform 815 smoothing steps and 42 double-precision correction steps in total. We plot the evolution of the Rayleigh quotient, which is a quality metric for these vectors, in Figure S9b. We observe that performing no high-precision correction steps is possible for the 8-bit smoother with some loss in quality, but it is not possible for the noisy 8-bit smoother. Hence, we observe that deploying low-accuracy smoothers with double-precision correction steps considerably reduces the number of digital OPs required for this computationally expensive adaptive procedure.

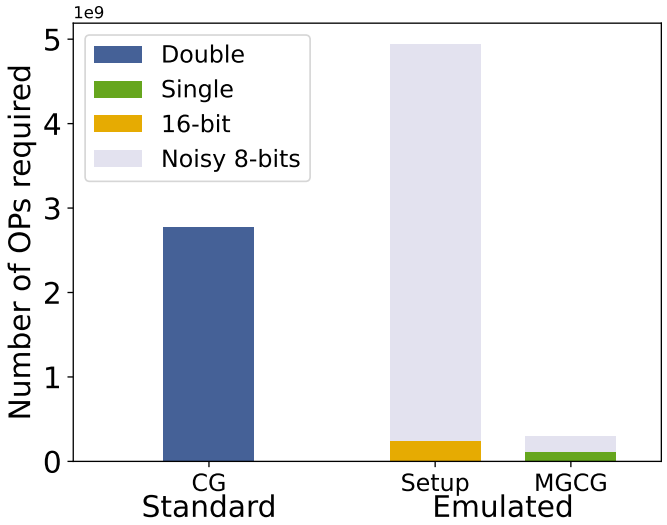


Figure S10. MPPMG solver with multiple precisions on the LQCD problem. We plot the number of OPs in different precisions required for convergence with the same accuracy. In this plot, we emulate the 16-bit fixed-point precision. We note that we perform a small amount of double-precision MVM during the MGCG run, which is not visible in this plot.

S11. Hybrid approach based on the iterative refinement method

To highlight the robustness of the MPPMG approach to high condition numbers, we compare it with the IR-based hybrid approach, which is widely used in the literature. We use a Richardson's solver as the inner solver with a fixed number of iterations. Hence, there are two tunable parameters to optimize this hybrid solver, namely, the number of inner iterations and the parameters of the solver ω. Here, we choose $\omega = 0.25/(\lambda_{max} + \lambda_{min})$ and empirically adjust the number of inner iterations for the different linear systems to decrease the number of digital operations. In Table S2, we state the bare fermionic masses m used for Figure 4B and the respective chosen number of inner iterations.

S12. Usage of multiple precisions

Mixed-precision methods with multiple precisions increase the performance of algorithms when applied to suitable hardware. In the main text, we introduce the hybrid MPPMG solver and restrict ourselves to only double-precision digital operations for the sake of simplicity. In Figure S10, we show the results using multiple precisions for the LQCD problem, where we use the same solver parameters as in Figure 4C. We observe that we require a negligible amount of double-precision OPs to solve the problem with the desired accuracy. Additionally, decreasing the precision of the digital calculations opens the possibility of implementing efficient application-specific integrated circuits (ASICs) to supplement the photonic processors. Efficient use of digital precision and analog computing is a key component for co-optimizing hardware and algorithms. We hope that the advancement of specialized hybrid hardware pushes researchers to further develop algorithms that benefit from calculations in the efficient, low-accuracy analog domain.

Table S1. Equivalent fixed-point and 8-bit noisy emulators. We choose the noise levels such that the mean relative errors match the ones of the respective fixed-point emulators. This is the intersection between the horizontal lines with the noisy curve in Figure S7a.

Bit precision	Noise	Relative error
3	0.100	0.205
4	0.037	0.064
5	0.021	0.054
6	0.010	0.027
7	0.004	0.016
8	0.000	0.013

Table S2. Problem and solver parameters in Figure 4D. We tune the bare fermionic mass to adjust the condition number of the linear system. We empirically choose the number of inner Richardson's iterations to decrease the number of digital IR steps.

Bare fermionic mass	Condition number	Number of inner iterations
1	22.61	20
0.5	65.95	20
0.25	190.76	50
0.1	637.70	100
-0.06	140311.92	100