

Laser Processing Unit™ Accuracy Explained

How an 8-Bit Analog Core Delivers 64-Bit Precision at High Speed

In short: The Laser Processing Unit (LPU) is wrapped by a digital controller that allows delivering high precision results while maintaining the LPU's speed advantage. The LPU acts as a fast analog linear solver, while the surrounding digital controller utilizes an iterative refinement algorithm (Wilkinson). The refinement process compensates for low precision of the computing core as well as systematic inaccuracies. Thus, the complete LPU is a fast, reliable, high-precision linear solver.

The core principle: correct the error, not the answer

The LPU is an accelerator that efficiently solves a particular subclass of linear systems ($Ax = b$)¹. Every LPU solve begins with a fast, approximate answer at native precision which is currently 8 bits. Rather than treating this as the final result, a digital controller computes exactly how far it deviates from the true solution and returns this deviation to the optical core as a new problem to solve. The resulting correction is added to the running solution. After a small number of repetitions, the error converges toward zero.

This is a well-established technique in numerical computing known as **iterative refinement (Wilkinson, 1963)**², and it is the same principle underlying modern mixed-precision HPC libraries, which combine a fast low-precision step with high-precision residual correction³.

Keeping up to speed

In the LPU context, the iterative refinement approach is extremely powerful due to its multiplicative accuracy gain. Each refinement step reduces the current error by the accuracy factor of the LPU, roughly allowing an exponential improvement with the number of calls. The digital controller steps are computationally cheap, as they are implemented on the digital hardware that controls the optical core, so the iteration time is mostly governed by the LPU solve time. In contrast, the conventional approach to a noisy analog measurement is to average over many readings which increases precision only with the square root of the number of measurements.

In practice, four LPU calls reach FP32 precision, and eight calls reach FP64⁴. Given that a single LPU solve runs hundreds to thousands of times faster than the equivalent GPU solve, even under the additional cost of refinement, the LPU remains tens to hundreds of times faster than a high-precision digital solver — while matching its precision exactly (see Figure 1 as an example).

¹ Dan Gluck, Yotam Mimran, Andrey Karenskih, Talya Vaknin, Omri Wolf, Ruti Ben-Shlomi, and Johannes Gebert. (2026). *Accelerating Sparse Linear Solvers with an Optical Laser Processing Unit: Invited Paper*. DOI: [10.1145/3801488.3808041](https://doi.org/10.1145/3801488.3808041).

² https://en.wikipedia.org/wiki/Iterative_refinement

³ Erin Carson, and Nicholas J. Higham. (2018). *Accelerating the Solution of Linear Systems by Iterative Refinement in Three Precisions*. SIAM Journal on Scientific Computing, 40(2), A817–A847. DOI: [10.1137/17M1140819](https://doi.org/10.1137/17M1140819)

⁴ These numbers assume typical use-cases, extreme cases may exist, e.g. ill-conditioned matrices, where the number of iterations may be slightly higher, this limitation isn't unique to the LPU and plagues all Krylov solvers.

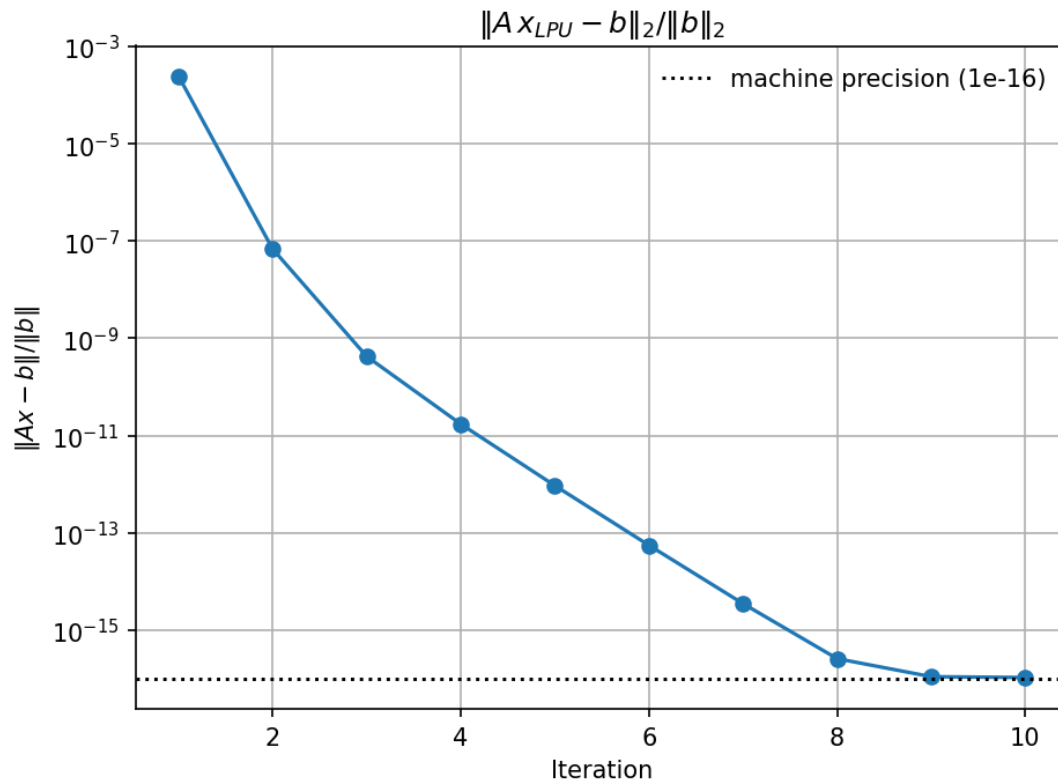


Figure 1: Error reduction upon LPU iterations, for an electrophysiology benchmark problem

Correcting errors specific to analog hardware

A related and distinct concern merits attention: Analog systems are subject not only to noise but to systematic error — manufacturing tolerances, calibration imperfections, thermal drift, and unit-to-unit variance. These are errors that averaging cannot address, as they are not random in nature.

Iterative refinement addresses them by the same mechanism as any other error. Any physical bias in the hardware surfaces directly as residual error and is corrected in the subsequent call. The loop requires no knowledge of *why* the analog core deviated — only the magnitude and direction of the deviation. This allows the system to remain robust to drift and manufacturing variance without recalibration.

What the end user sees

This process is entirely abstracted from the user. A call to the solver specifies a set of linear equations ($Ax = b$), and a precision tolerance and returns a result x at that precision. The refinement loop, and the analog core underlying it, remain invisible unless specifically examined. Users may also trade precision for additional speed by specifying a looser tolerance.

Conclusion

The LPU enjoys the advantages that come from in-memory computing at light speed. The analog physics-based computation is nested in a Wilkinson-style iterative refinement algorithm. Together, the optical processor and digital controller become a very fast, high precision linear solver.

Technical appendix (Wilkinson's iterative refinement):

1. The optical core produces an initial approximate solution x_0 to $Ax = b$, at native (8-bit) precision.
2. For the k th iterations, the residual is computed digitally, $r_k = b - Ax_k$ at the precision of A and b .
3. The optical core solves the correction problem: $Ae_k = r_k$
4. The correction is accumulated digitally: $x_{k+1} = x_k + e_k$
5. Steps 2–4 are repeated until $|r_k|$ falls below the target tolerance.

Copyright/Disclaimer

All information contained in this document, including but not limited to concepts, designs, methodologies, and business strategies, is confidential and proprietary to LightSolver. No part of this document may be reproduced, disclosed, or used without the express written permission of LightSolver. By reviewing this document, the recipient agrees to maintain the confidentiality of the information and acknowledges that any unauthorized use or disclosure may result in legal consequences. All intellectual property rights pertaining to the contents of this document are reserved by LightSolver.