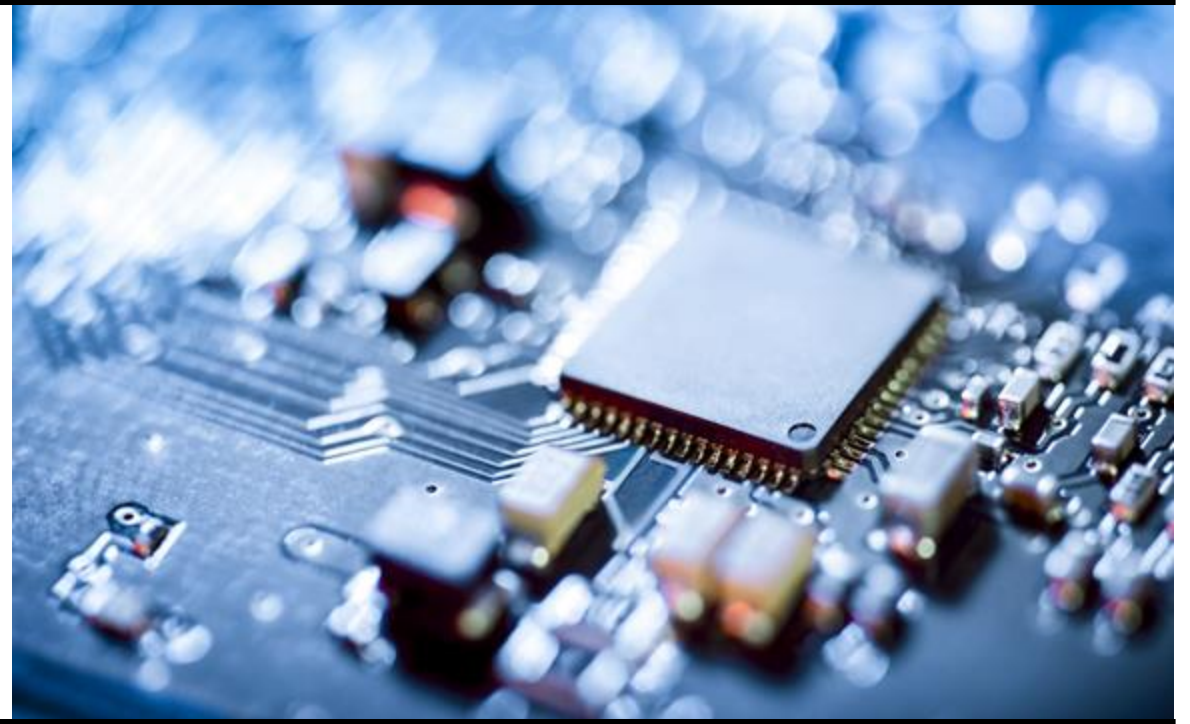


# Parallel Methods in Linear Algebra

Rob Pellegrin



# Background & Motivation

## Importance of Linear Algebra

- Foundation of scientific computing, simulations, machine learning

## Need for Performance

- Large matrices require efficient algorithms to reduce computation time

## Project Goal

- Compare serial vs. parallel performance—How much faster are parallel implementations?
- Compare C++ and Rust implementations—How much does the language matter?

# Methods & Tools

## **Operations Implemented**

- Matrix Multiplication
- LU Decomposition

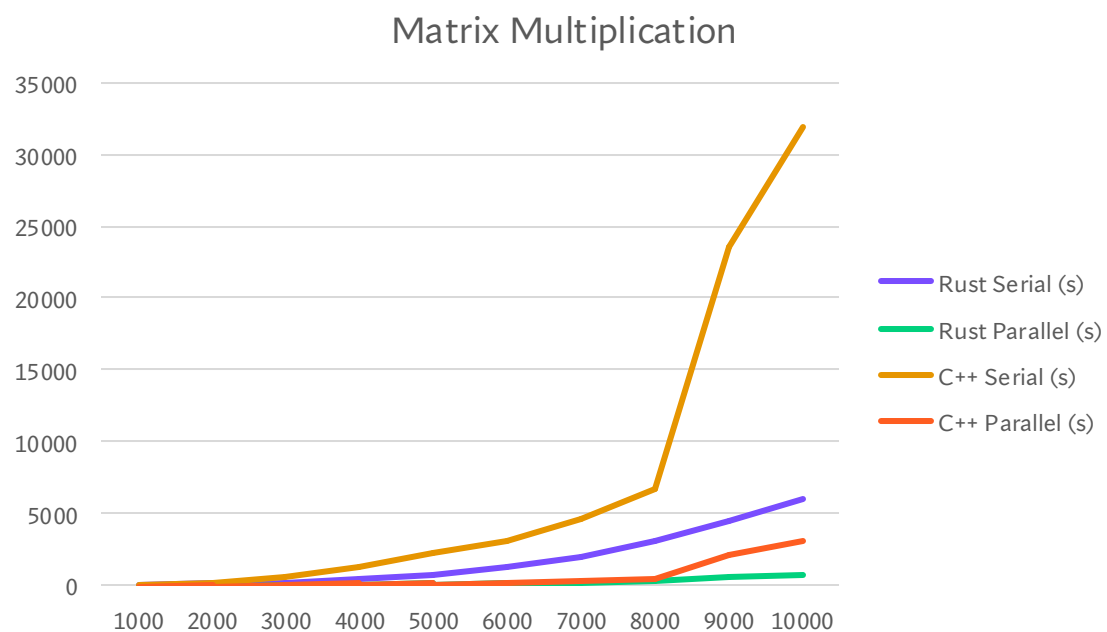
## **Languages & Libraries**

- C++ (parallelized with OpenMP)
- Rust (parallelized with Rayon)
- GNU Parallel

## **Testing Conditions**

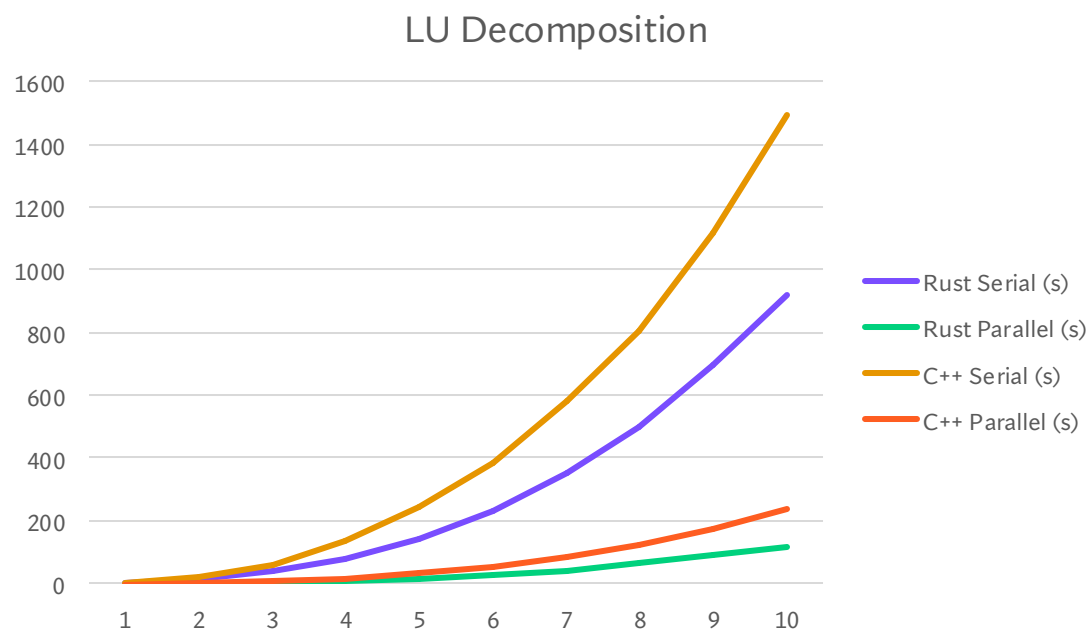
- Same machine and compilers
- Compiler optimizations enabled
- Standard high-resolution timers used for performance measurement

# Matrix Multiplication Results



Matrix Size	Rust Serial (s)	Rust Parallel (s)	C++ Serial (s)	C++ Parallel (s)
1000	2.36	0.19	4	0
2000	43.37	1.86	161	2
3000	144.72	6.67	574	9
4000	346.12	16.92	1160	29
5000	619.36	35.46	2205	65
6000	1144.6	67.48	3068	128
7000	1953.34	126.3	4614	222
8000	2961.97	243.83	6708	350
9000	4400.2	476.71	23492	2110
10000	6016.93	692.93	31935	3096

# LU Decomposition Results



Matrix Size	Rust Serial (s)	Rust Parallel (s)	C++ Serial (s)	C++ Parallel (s)
1000	1.33	0.25	2	0
2000	11.39	1.01	18	1
3000	39.2	3.45	60	6
4000	79.38	8.07	132	15
5000	143.09	15.63	241	30
6000	232.67	26.62	385	52
7000	353.85	41.75	580	82
8000	499.4	61.86	804	122
9000	693.2	87.32	1120	174
10000	917.97	118.78	1492	239

---

# Summary & Future Work

---

---

## **Key Takeaways**

Parallelization significantly improves computation time

Rust is superior to C++ in performance—choice in language matters!

---

## **Future Directions**

Implement GPU acceleration (CUDA or OpenCL)

Investigate other matrix decompositions (QR, Cholesky)

---

# Bibliography

Ole Tange, *GNU Parallel: The Command-Line Power Tool*, ;login: The USENIX Magazine, February 2011. Available at: <https://www.gnu.org/software/parallel/>

OpenMP Architecture Review Board, *OpenMP Application Programming Interface Version 5.1*, 2020. Available at: <https://www.openmp.org/specifications/>

Niko Matsakis and Rayon Contributors, *Rayon: A data parallelism library for Rust*. Available at: <https://github.com/rayon-rs/rayon> (Accessed 2025).

Rayon Contributors, *Rayon Crate Documentation*. Available at: <https://docs.rs/rayon/latest/rayon/> (Accessed 2025).