

# Armadillo: a template-based C++ library for linear algebra

Conrad Sanderson<sup>1</sup> and Ryan Curtin<sup>2</sup>

DOI: [10.21105/joss.00026](https://doi.org/10.21105/joss.00026)

<sup>1</sup> CSIRO, Australia, and NICTA, Australia <sup>2</sup> Symantec Corporation, USA

## Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

## Licence

Authors of JOSS papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC-BY](#)).

## Summary

The C++ language is often used for implementing functionality that is performance and/or resource sensitive. While the standard C++ library provides many useful algorithms (such as sorting), in its current form it does not provide direct handling of linear algebra (matrix maths).

Armadillo is an open source linear algebra library for the C++ language, aiming towards a good balance between speed and ease of use. Its high-level Application Programming Interface (API) is deliberately similar to the widely Matlab and Octave languages (Eaton et al. 2015), so that mathematical operations can be expressed in a familiar and natural manner. The library is useful for algorithm development directly in C++, or relatively quick conversion of research code into production environments.

Armadillo provides efficient objects for vectors, matrices and cubes (third order tensors), as well as over 200 associated functions for manipulating data stored in the objects. Integer, floating point and complex numbers are supported, as well as dense and sparse storage formats. Various matrix factorisations are provided through integration with LAPACK (Demmel 1997), or one of its high performance drop-in replacements such as Intel MKL (Intel 2016) or OpenBLAS (Xianyi, Qian, and Saar 2016). It is also possible to use Armadillo in conjunction with NVBLAS to obtain GPU-accelerated matrix multiplication (NVIDIA 2015).

Armadillo is used as a base for other open source projects, such as MLPACK, a C++ library for machine learning and pattern recognition (Curtin et al. 2013), and RcppArmadillo, a bridge between the R language and C++ in order to speed up computations (Eddelbuettel and Sanderson 2014).

Armadillo internally employs an expression evaluator based on template metaprogramming techniques (Abrahams and Gurtovoy 2004), to automatically combine several operations in order to increase speed and efficiency. An overview of the internal architecture is given in (Sanderson 2010).

## References

- Abrahams, David, and Aleksey Gurtovoy. 2004. *C++ Template Metaprogramming: Concepts, Tools, and Techniques from Boost and Beyond*. Addison-Wesley Professional.
- Curtin, Ryan R., James R. Cline, N. P. Slagle, William B. March, Parikshit Ram, Nishant A. Mehta, and Alexander G. Gray. 2013. “MLPACK: A Scalable C++ Machine Learning Library.” *Journal of Machine Learning Research* 14 (Mar): 801–5. <http://jmlr.org/>

[papers/v14/curtin13a.html](#).

Demmel, James W. 1997. *Applied Numerical Linear Algebra*. SIAM.

Eaton, John W., David Bateman, Søren Hauberg, and Rik Wehbring. 2015. *GNU Octave Version 4.0.0 Manual: A High-Level Interactive Language for Numerical Computations*. <http://www.gnu.org/software/octave/>.

Eddelbuettel, Dirk, and Conrad Sanderson. 2014. “RcppArmadillo: Accelerating R with High-Performance C++ Linear Algebra.” *Computational Statistics & Data Analysis* 71: 1054–63. doi:10.1016/j.csda.2013.02.005.

Intel. 2016. “Math Kernel Library (MKL).” <http://software.intel.com/en-us/intel-mkl/>.

NVIDIA. 2015. “NVBLAS Library.” <http://docs.nvidia.com/cuda/nvblas/>.

Sanderson, Conrad. 2010. “Armadillo: An Open Source C++ Linear Algebra Library for Fast Prototyping and Computationally Intensive Experiments.” NICTA.

Xianyi, Zhang, Wang Qian, and Werner Saar. 2016. “OpenBLAS: An Optimized BLAS Library.” <http://www.openblas.net/>.