Document Number: P1891R0
Date: 2019-10-01
Reply-to: guy@hatcat.com
Authors: Guy Davidson (guy@hatcat.com); Mark Hoemmen (mhoemmen@sandia.gov); David Hollman (dshollm@sandia.gov); Bob Steagall (bob.steagall.cpp@gmail.com); Christian Trott (crtrott@sandia.gov)
Audience: SG6, SG14, SG19, LEWG

Two papers related to linear algebra are being presented to LEWG for consideration at Belfast. The purpose of this brief overview paper is to present a context for both papers, and offer guidance to LEWG participants about reading the papers.

The linear algebra effort has been underway for about 18 months, with two different foci. The first group is presenting what we shall call the identifier paper, P1385. The second group is presenting what we shall call the BLAS paper, P1673. We should also mention the mdspan paper, P0009, and the later mdarray paper, P1684. Although the mdspan and mdarray papers are relevant to the implementation of linear algebra, they have many applications outside of this domain and do not form part of this consideration.

P1385 seeks to introduce new types to the standard library, matrix and vector, with associated non-member operators to enable a clear syntax for carrying out linear algebra mathematics as represented in textbooks devoted to the subject. It does NOT seek to mandate an implementation, nor a quality of implementation: it only seeks to enable the syntax and require an implementation from standard library writers which, by the very design of the library, can be substituted by the user in edge cases where specialised performance gains may be available.

P1673 seeks to introduce BLAS to the standard library. The BLAS is a (non-ISO) standard C and Fortran interface to many low-level linear algebra algorithms, including vector and matrix multiplication. BLAS implementations have a long history of optimisation by several hardware platform vendors and third parties. The paper offers identifiers similar to those defined by the BLAS report, but with a C++ API resembling existing standard algorithms. It uses basic_mdspan (P0009) and basic_mdarray (P1684) to represent vectors and matrices.

The papers are orthogonal. They are not competing papers; the latter paper is not a counter-proposal to the former paper; there is no overlap of functionality. If either paper fails to make the standard, this is not a disaster. If only P1385 succeeds, then the C++ community will have linear algebra types and a default implementation, and will be able to specialise them privately in terms of BLAS if they choose. It's likely that a BLAS-based specialization would build a subset of P1673's functionality, but it need not expose that subset to users. If only P1673 succeeds, then the C++ community will continue to implement their own private linear algebra types, but they will have ready access to a generic C++ BLAS interface to

assist them. Ideally, both papers will make their way into the library before C++23, giving vendors the opportunity to add textbook-friendly linear algebra implemented in terms of BLAS.

Extensive development has gone into these papers. There have been monthly telecons under SG14 chaired by Michael Wong, regularly attended by upwards of a dozen experts in the field, devoted solely to advancing these papers. We look forward to LEWG reviewing the API design and naming of identifiers and hope that the work to date has proved satisfactory for this group.