THE UNIVERSITY OF TENNESSEE KNOXVILLE

# BATCHED BLAS

## PREMISE

In a growing number of computational science disciplines, multidimensional non-linear equations are approximated as large batches of rudimentary linear algebra computations. Basic Linear Algebra Subprograms (Batched BLAS) aims to standardize the interface to these routines through a community-driven process. This enables the users to efficiently perform thousands of small-size BLAS operations on massively parallel hardware, be it traditional multi-core CPUs or a variety of computational hardware accelerators.

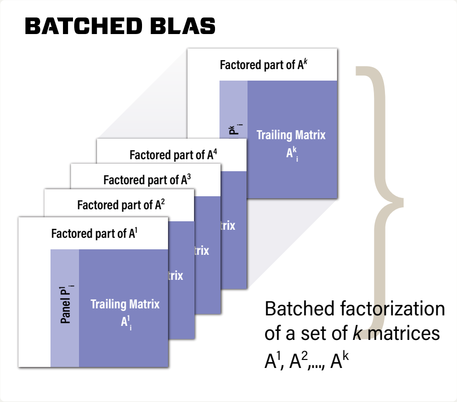
## DEFINITION

Batched BLAS computes multiple and independent BLAS operations on small-sized matrices and/or vectors in a single routine invocation.

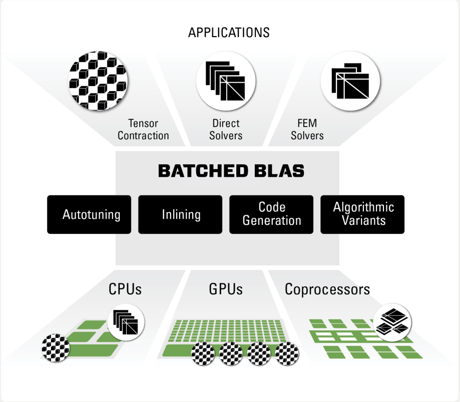
## APPLICATIONS

Batched BLAS benefits multiple computational fields:

- Structural mechanics
- Astrophysics
- Direct sparse solvers
- High-order FEM simulations



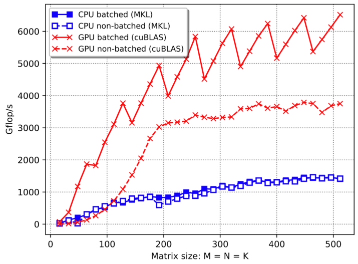
Batched factorization of a set of *k* matrices $A^1, A^2, \ldots, A^k$



## PERFORMANCE

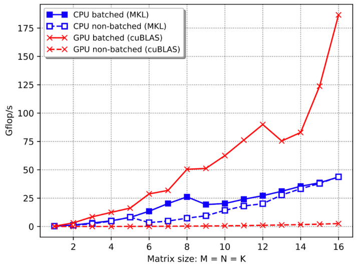
### BATCHED LEVEL 3 BLAS DGEMM EXAMPLE

DGEMM (N,N) batch_count = 500
CPU: Intel Xeon Gold 6140, 18 cores, 2 sockets
GPU: Tesla Titan V, 80 SMs



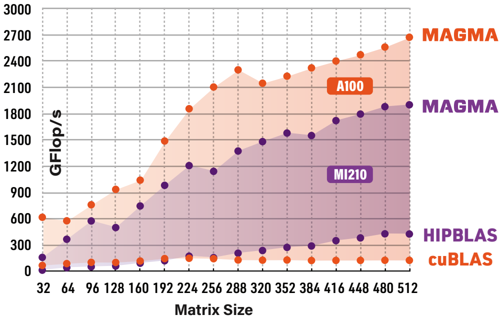
### BATCHED LEVEL 2 BLAS DGEMV EXAMPLE

DGEMM (N,N) batch_count = 100000
CPU: Intel Xeon Gold 6140, 18 cores, 2 sockets
GPU: Tesla Titan V, 80 SMs



### PERFORMANCE OF BATCH QR

double-precision arithmetic on 1 thousand matrices



## TECHNOLOGIES

OpenMP
- Multicore
- Accelerators

NVIDIA CUDA
- Fused Kernels
- Multiple Streams

intel oneAPI HPC TOOLKIT

ROCm ON

## ADVANTAGES

More efficient and portable implementations

HPC numerical library for modern architectures

Better hardware utilization and energy efficiency

Encourages and simplifies community efforts to build higher-level algorithms on top of Batched BLAS

Multiple precisions: 16, 32, and 64 bits in real and complex domains

IN COLLABORATION WITH

UMEÅ UNIVERSITET

Inria

Science & Technology Facilities Council
Rutherford Appleton Laboratory

MANCHESTER 1824
The University of Manchester

WITH SUPPORT FROM

European Commission

U.S. DEPARTMENT OF ENERGY

NSF

SPONSORED BY

AMD

intel

NVIDIA

ICL INNOVATIVE COMPUTING LABORATORY

FIND OUT MORE AT https://icl.utk.edu/bblas

THE UNIVERSITY OF TENNESSEE KNOXVILLE

# BATCHED BLAS

## WORKSHOPS

**Sparse BLAS Workshop 2023**
Workshop on the Design and Standardization of Basic and Advanced Sparse Linear Algebra Routines
Knoxville, TN | November 7-9, 2023

**https://icl.utk.edu/workshops/sparseblas2023**

**Workshop on Batched, Reproducible, and Reduced Precision BLAS 2017**
Atlanta, GA

**http://bit.ly/Batch-BLAS-2017**

**Workshop on Batched, Reproducible, and Reduced Precision BLAS 2016**
Knoxville, TN

**http://bit.ly/Batch-BLAS-2016**

## PAPERS AND RELATED MATERIAL

Abdelfattah, A., S. Tomov, and J. Dongarra, **"Optimizing Batch HGEMM on Small Sizes Using Tensor Cores,"** San Jose, CA, *GPU Technology Conference (GTC)*, March 2019.

Dongarra, J., S. Hammarling, N. J. Higham, S. Relton, P. Valero-Lara, and M. Zounon, **"The Design and Performance of Batched BLAS on Modern High-Performance Computing Systems,"** *International Conference on Computational Science (ICCS 2017)*, Zürich, Switzerland, Elsevier, June 2017. DOI: DOI:10.1016/j.procs.2017.05.138

Ahmad Abdelfattah, Timothy Costa, Jack Dongarra, Mark Gates, Azzam Haidar, Sven Hammarling, Nicholas J. Higham, Jakub Kurzak, Piotr Luszczek, Stanimire Tomov, Mawussi Zounon, **"A Set of Batched Basic Linear Algebra Subprograms and LAPACK Routines,"** *ACM TOMS*, 47(3):1–23, June, 2020. DOI: 10.1145/3431921

Peter Ahrens, Hong Diep Nguyen, and James Demmel, **"Efficient Reproducible Floating Point Summation and BLAS,"** *Electrical Engineering and Computer Sciences University of California at Berkeley Technical Report* no. UCB/EECS-2015-229, December 2015.

Jack Dongarra, Iain Duff, Mark Gates, Azzam Haidar, Sven Hammarling, Nicholas J. Higham, Jonathan Hogg, Pedro Valero Lara, Mawussi Zounon, Samuel D. Relton, and Stanimire Tomov, **"A Proposed API for Batched Basic Linear Algebra Subprograms,"** *Draft Report*, May 2016.

**ReproBLAS**
**http://bebop.cs.berkeley.edu/reproblas/**

Samuel D. Relton, Pedro Valero-Lara, and Mawussi Zounon, **"A Comparison of Potential Interfaces for Batched BLAS Computations,"** *NLAFET Working Note 5*, August 2016.

**Compact Batched API Document**
Intel MKL Team
**https://www.dropbox.com/s/gplop3sxhg8le3r/MKL_COMPACT_v4.docx?dl=0**

Batched Sparse Linear Algebra functionality and implementation was under development for DOE's Exascale Computing Project since 2021. The current interface design spans banded, direct, and iterative methods and integration in the following libraries: Ginkgo, hypre, Kokkos Kernels, MAGMA, SuperLU.

Ginkgo hypre kokkos MAGMA SuperLU

IN COLLABORATION WITH

UMEÅ UNIVERSITET · Inria · Science & Technology Facilities Council Rutherford Appleton Laboratory · MANCHESTER 1824 The University of Manchester

WITH SUPPORT FROM

European Commission · U.S. DEPARTMENT OF ENERGY · NSF

SPONSORED BY

AMD · intel · NVIDIA

ICL INNOVATIVE COMPUTING LABORATORY

FIND OUT MORE AT **https://icl.utk.edu/bblas**