MAGMA

LEARN MORE AT https://icl.utk.edu/magma

MATRIX ALGEBRA ON GPU AND MULTICORE ARCHITECTURES (MAGMA) is a collection of next-generation linear algebra libraries for heterogeneous architectures. MAGMA is designed and implemented by the team that developed LAPACK and ScaLAPACK, incorporating the latest developments in hybrid synchronization-avoiding and communication-avoiding algorithms, as well as dynamic runtime systems. Interfaces for the current LAPACK and BLAS standards are supported to enable computational scientists to seamlessly port any linear algebra-reliant software components to heterogeneous architectures. MAGMA allows applications to fully exploit the power of current heterogeneous systems of multi/many-core CPUs and multiple GPUs to deliver the fastest possible time to accurate solution within given energy constraints.

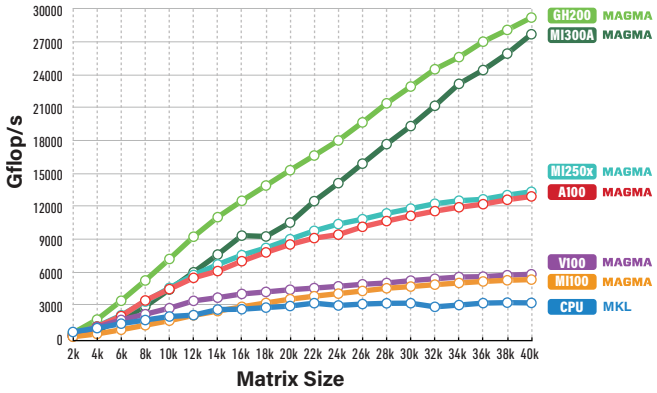
HYBRID ALGORITHMS

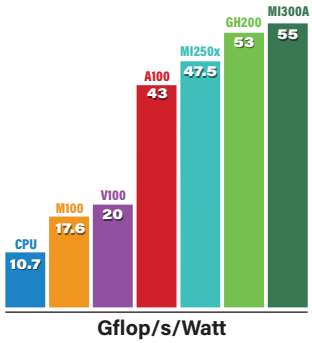
MAGMA uses a hybridization methodology, where algorithms of interest are split into tasks of varying granularity, and their execution is scheduled over the available hardware components. Scheduling can be static or dynamic. In either case, small non-parallelizable tasks, often on the critical path, are scheduled on the CPU, and larger more parallelizable ones, often Level-3 BLAS, are scheduled on the GPU. When CPU-GPU communication overhead becomes significant, the entire computation might run exclusively on the GPU, leading to what is termed as "GPU-native algorithms".

PERFORMANCE & ENERGY EFFICIENCY

MAGMA LU factorization in double-precision arithmetic

CPU	MI100	V100	A100	MI250x	GH200	MI300A
Intel Xeon Platinum 8460Y+ (Sapphire Rapids)	AMD Instinct MI100 GPU	NVIDIA Volta V100 GPU	NVIDIA Ampere GPU	AMD Instinct MI250x GCD	NVIDIA Grace-Hopper GH200 GPU	AMD Instinct MI300A APU
2 x 40 cores @3.7GHz	120 CUs @ 1.5GHz	80 SMs x 64 @ 1.37 GHz	108 SMs x 64 @ 1.41 GHz	110 CUs @ 1.7GHz	132 SMs @ 1.98GHz	228 CUs @ 2.1GHz





FEATURES AND SUPPORT

- **MAGMA 2.9** for CUDA or HIP
- **MAGMA Pre-release** for SYCL/DPC++

CUDA	HIP	SYCL Pre-release	
✓	✓	✓	Linear system solvers
✓	✓	✓	Eigenvalue problem solvers
✓	✓	✓	Auxiliary BLAS
✓	✓	✓	Batched LA
✓	✓	✓	CPU/GPU Interface
✓	✓	✓	Multiple precision support
✓	✓		Mixed precision (including FP16)
✓	✓	✓	Non-GPU-resident factorizations
✓	✓	✓	GPU-only factorizations
✓	✓		Multicore and multi-GPU support
✓	✓	✓	LAPACK testing
✓	✓	✓	Linux
✓	✓		Windows
✓	✓		macOS

INDUSTRY COLLABORATION

NVIDIA

Long-term collaboration and support on the development of MAGMA.

intel **Intel Parallel Computing Center**

The objective of the Innovative Computing Laboratory's IPCC is the development and optimization of numerical linear algebra libraries and technologies for applications, while tackling current challenges in heterogeneous Intel® Xeon Phi™ coprocessor-based High Performance Computing.

AMD

Long-term collaboration and support on the development of clMAGMA, the OpenCL™ port of MAGMA, and hipMAGMA.

IN COLLABORATION WITH

Berkeley UNIVERSITY OF CALIFORNIA

University of Colorado Denver

Inria informatics mathematics

KAUST

SPONSORED BY

U.S. Department of Defense

ECP

U.S. DEPARTMENT OF ENERGY

NSF National Science Foundation

WITH SUPPORT FROM

AMD

intel

MathWorks™

NVIDIA

ICL INNOVATIVE COMPUTING LABORATORY

THE UNIVERSITY OF TENNESSEE KNOXVILLE

MAGMA

LEARN MORE AT https://icl.utk.edu/magma

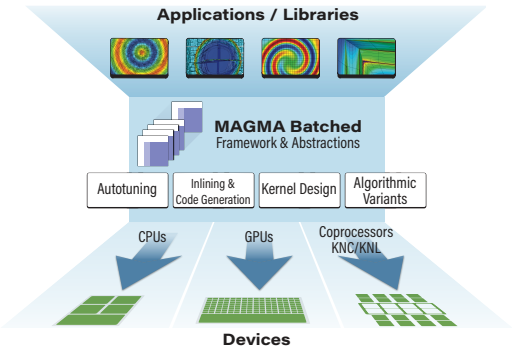
MAGMA BATCHED — Batched factorization of a set of small matrices in parallel

Numerous applications require factorization of many small matrices:

- Deep learning
- Sparse direct solvers
- Structural mechanics
- High-order FEM simulations
- Astrophysics

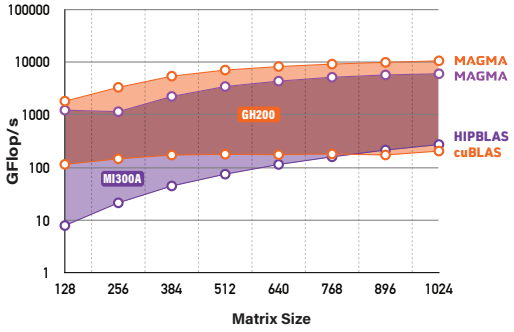
ROUTINES

- ✓ LU, QR, and Cholesky
- ✓ Solvers and matrix inversion
- ✓ All BLAS 3 (fixed + variable)
- ✓ SYMV, GEMV (fixed + variable)



Performance of Batch QR

double-precision arithmetic on 1 thousand matrices



MAGMA 2.9 DRIVER ROUTINES

	MATRIX	OPERATION	ROUTINE	INTERFACES CPU	INTERFACES GPU
LINEAR EQUATIONS	GE	Solve using LU	{sdcz}gesv	✓	✓
		Solve using MP	{zc,ds}gesv		✓
	SPD/HPD	Solve using Cholesky	{sdcz}posv	✓	✓
		Solve using MP	{zc,ds}posv		✓
LEAST SQUARES	GE	Solve LLS using QR	{sdcz}gels		✓
		Solve using MP	{zc,ds}geqrsv		✓
STANDARD EVP	GE	Compute e-values,	{sdcz}geev	✓	
		optionally e-vectors		✓	
	SY/HE	Computes all e-values,	{sd}syevd	✓	✓
		optionally e-vectors	{cz}heevd	✓	✓
		Range (D&C)	{cz}heevdx		✓
		Range (B&I It.)	{cz}heevx	✓	✓
		Range (MRRR)	{cz}heevr	✓	✓
STAND. SVP	GE	Compute SVD,	{sdcz}gesvd	✓	
		optionally s-vectors	{sdcz}gesdd	✓	
GENERALIZED EVP	SPD/HPD	Compute all e-values,	{sd}sygvd	✓	
		optionally e-vectors	{cz}hegvd	✓	
		Range (D&C)	{cz}hegvdx	✓	
		Range (B&I It.)	{cz}hegvx	✓	
		Range (MRRR)	{cz}hegvr	✓	

Abbreviations

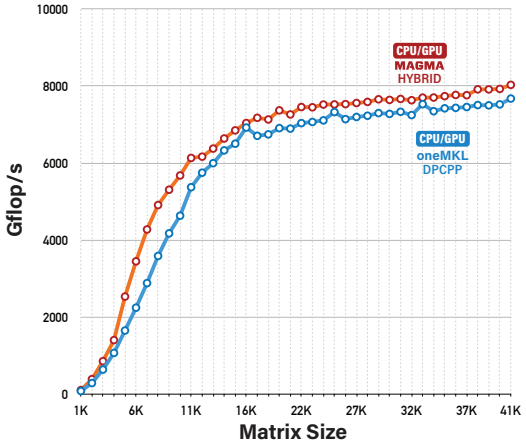
GE	General
SPD/HPD	Symmetric/Hermitian Positive Definite
TR	Triangular
D&C	Divide & Conquer
B&I It	Bisection & Inverse Iteration
MP	Mixed-Precision Iterative Refinement

Naming Convention

magma_{routine name}[_gpu]

Intel GPU results on Aurora

Intel Xeon Max Series CPU + Intel Data Center Max Series GPU (codename PVC)



Results using Intel oneAPI release 2025.0.5. Aurora is a resource of the Argonne Leadership Computing Facility at Argonne National Laboratory

UPCOMING IN MAGMA 2.10

- New functionality: Batch SVD
- New functionality: Variable-size batch non-pivoting LU factorization
- Performance Improvements for Batch Cholesky factorization and solve
- Support for CUDA-13 and ROCM-7

IN COLLABORATION WITH

Berkeley UNIVERSITY OF CALIFORNIA · University of Colorado Denver · Inria · KAUST

SPONSORED BY

U.S. Department of Defense · ECP · U.S. DEPARTMENT OF ENERGY · National Science Foundation

WITH SUPPORT FROM

AMD · intel · MathWorks · NVIDIA