

MAGMA

MAGMA (Matrix Algebra on GPU and Multicore Architectures) 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- and communication-avoiding algorithms, as well as dynamic runtime systems. Interfaces for the current LAPACK and BLAS standards are supported to allow 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 multi-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 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.

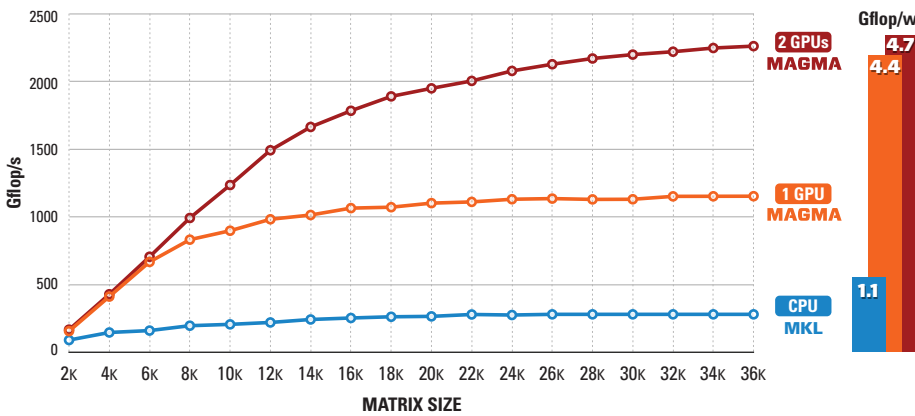
PERFORMANCE & ENERGY EFFICIENCY

MAGMA on Kepler K40

LU factorization in double precision arithmetic

GPU NVIDIA K40 (Atlas)
15 MP x 192 @ 0.88 GHz

CPU Intel Xeon ES-2670 (Sandy Bridge)
2 x 8 cores @ 2.60 GHz



FEATURES AND SUPPORT

- ▶ **MAGMA 1.6** FOR **CUDA**
- ▶ **clMAGMA 1.3** FOR **OpenCL**
- ▶ **MAGMA MIC 1.3** FOR **Intel Xeon Phi**

CUDA	OpenCL	Intel Xeon Phi	Feature
●	●	●	Linear system solvers
●	●	●	Eigenvalue problem solvers
●	●	●	Auxiliary BLAS
●	●	●	Batched LA
●	●	●	Sparse LA
●	●	●	CPU Interface
●	●	●	GPU Interface
●	●	●	Multiple precision support
●	●	●	Non-GPU-resident factorizations
●	●	●	Multicore and multi-GPU support
●	●	●	LAPACK testing
●	●	●	Linux
●	●	●	Windows
●	●	●	Mac OS

INDUSTRY COLLABORATION



NVIDIA's CUDA Center of Excellence Program recognizes universities expanding the frontier of massively parallel computing using CUDA.



Intel Parallel Computing Center

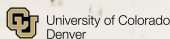
The objective of the Innovative Computing Laboratory's IPCC is the development and optimization of numerical linear algebra libraries and technologies for applications, while tackling current challenges in heterogeneous Intel® Xeon Phi™ coprocessor-based High Performance Computing.



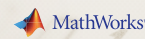
Long-term collaboration and support on the development of clMAGMA, the OpenCL™ port of MAGMA.



IN COLLABORATION WITH



WITH SUPPORT FROM



SPONSORED BY



U.S. Department of Defense



U.S. DEPARTMENT OF ENERGY



National Science Foundation
SILAS Award



MAGMA

MAGMA BATCHED

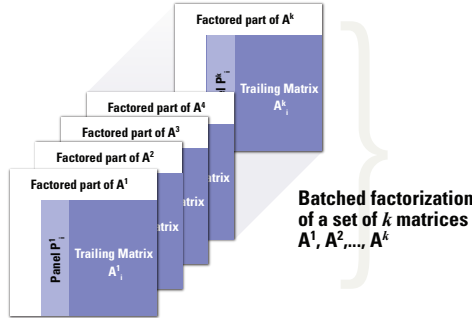
BATCHED FACTORIZATION OF A SET OF SMALL MATRICES IN PARALLEL

Numerous applications require factorization of many small matrices

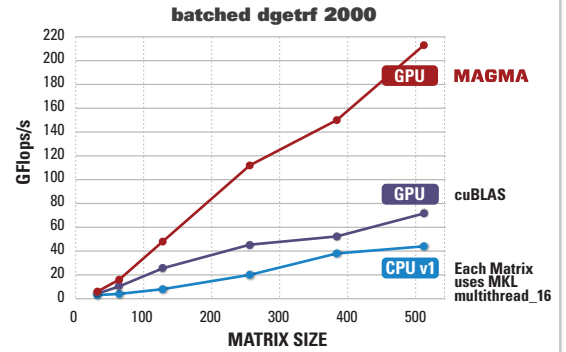
- Structural mechanics
- Astrophysics
- Sparse direct solvers
- High-order FEM simulations

ROUTINES

- LU, QR, and Cholesky ✓
- Solvers and matrix inversion ✓
- BLAS: GEMM, TRSM, GEMV, SYRK ✓



PERFORMANCE



MAGMA 1.6 DRIVER ROUTINES

	MATRIX	OPERATION	ROUTINE	INTERFACES	
				CPU	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, optionally e-vectors	{sdcz}geev	✓	
		Computes all e-values, optionally e-vectors	{sd}syevd	✓	✓
	SY/HE	Range (D&C)	{cz}heevdx		✓
		Range (B&I It.)	{cz}heevx	✓	✓
		Range (MRRR)	{cz}heevr	✓	✓
STAND. SVP	GE	Compute SVD, optionally s-vectors	{sdcz}gesvd	✓	
			{sdcz}gesdd	✓	
GENERALIZED EVP		Compute all e-values, optionally e-vectors	{sd}sygvd	✓	
			{cz}hegvd	✓	
	SPD/HPD	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

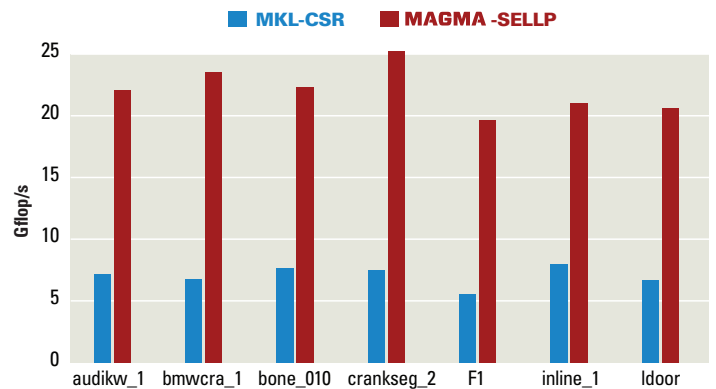
magma_{routine name}_{gpu}

MAGMA SPARSE

ROUTINES	CG, GMRES, BiCGSTAB, LOBPCG, Iterative refinement
PRECONDITIONERS	ILU / IC, Jacobi
KERNELS	SpMV, SpMM
DATA FORMATS	CSR, ELL, SELL-P

PERFORMANCE

Sparse matrix - vector product (SpMV) in double precision arithmetic



MATRICES
The University of Florida Sparse Matrix Collection <http://www.cise.ufl.edu/research/sparse/matrices/>

GPU NVIDIA K40 (Atlas)
15 MP x 192 @ 0.88 GHz

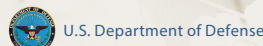
CPU Intel Xeon ES-2670 (Sandy Bridge)
2 x 8 cores @ 2.60 GHz



IN COLLABORATION WITH



SPONSORED BY



WITH SUPPORT FROM

