The MMatrix, TEnsoR, and Deep-learning Optimized Routines (MATEDOR) project seeks to develop software technologies and standard APIs, along with a sustainable and portable library for large-scale computations, the individual parts of which are very small matrix or tensor computations. The main target is the acceleration of science and engineering applications that fit this profile, including deep learning, data mining, astrophysics, image and signal processing, hydrodynamics, and more.

**Standard Interface for Batched Routines**

Working closely with affected application communities, we will define modular, language-agnostic interfaces that can be implemented to work seamlessly with the compiler and optimized using techniques like code replacement and pinning. This will provide the developers of applications, compilers, and runtime systems with the option of expressing as a single call to a routine from the new batch operation standard and would allow the entire linear algebra (LA) community to collectively attack a wide range of small matrix or tensor problems. Success in such an effort will require innovations in interface design, computational and numerical optimization, as well as packaging and deployment at the user site to trigger final stages of tuning at the moment of execution.

**Sustainable and Performance-Portable Software Library**

We will demonstrate the power of the MATEDOR interface by delivering a high-performance numerical library for batched LA subroutines autotuned for the modern processor architecture and system designs. The MATEDOR library will include LAPACK routine equivalents for many small dense problems, tensor, and application-specific operations, e.g., for deep learning; these routines will be constructed as much as possible out of calls to batched BLAS routines and their look-alikes required in sparse computation.

**Enabling Technologies**

MATEDOR will develop enabling technologies for very small matrix and tensor computations, including: (1) autotuning, (2) inlining, (3) code generation, and (4) algorithmic variants. We define the success of the research conducted and the software developed under the MATEDOR project as being able to automate these four aspects to allow for both flexibility and close-to-optimal performance of the final code used by the domain scientist.

**Standard APIs (for Batched BLAS and LAPACK)**

Proposed API is very similar to the standard BLAS/LAPACK API

```c
void dgemm_batched(
    batched_trans_t transA, batched_trans_t transB, batched_int_t m, batched_int_t n, batched_int_t k,
    double alpha ,
    double * const * const * da_array , batched_int_t ldda ,
    double * const * const * db_array , batched_int_t ldbb ,
    double beta ,
    double ** dc_array , batched_int_t ldc ,
    batched_int_t batchCount , batched_queue_t queue
    batched_int_t *info);
```

**PUBLICATIONS**

Abdelfattah, A., S. Tomov, and J. Dongarra

*Matrix Multiplication on Batches of Small Matrices in Half and Half-Complex Precisions*


D. Nichols, N.-S. Tomov, F. Betancourt, S. Tomov, K. Wong, and J. Dongarra

*MagmaGPU: Towards High-Performance Data Analytics and Machine Learning for Data-Driven Scientific Computing*

ISC High Performance, Frankfurt, Germany, Springer International Publishing, June 2019

Masliah, I., A. Abdelfattah, A. Haidar, S. Tomov, M. Baboulin, J. Falcou, and J. Dongarra

*Algorithms and Optimization Techniques for High-Performance Matrix-Matrix Multiplications of Very Small Matrices*

Parallel Computing, vol. 81, pp. 1–21, January 2019

Lu, Y., I. Yamazaki, F. Ino, Y. Matsushima, S. Tomov, and J. Dongarra

*Reducing the Amount of out-of-core Data Access for GPU-Accelerated Randomized SVD*

Concurrency and Computation: Practice and Experience, April 2020


*Integrating Deep Learning in Domain Sciences at Exascale*

Proc. of Smoky Mountains Computational Sciences & Engineering Conference (SMC2020), August 2020


*Performance of Hierarchical-Matrix BiCGStab Solver on GPU clusters*

2018 IEEE International Parallel & Distributed Processing Symposium

---

**FIND OUT MORE AT**

https://icl.utk.edu/matedor/

---

**WITH SUPPORT FROM**

ICL INNOVATIVE COMPUTING LABORATORY

THE UNIVERSITY OF TENNESSEE KNOXVILLE
Breadth of MATEDOR’s Impact on Application Domains

**PERFORMANCE OF BATCHED LU**
in double precision arithmetic on 1 million matrices

**Tensor Contractions in**
High Order FEM & Applications
Tensor Contractions: computing $B^T \cdot (B \cdot (A.B^T)) \cdot B$, double precision, Tesla V100 GPU

**Sparse/Dense Solvers & Preconditioners**
Batch Matrix Factorization, 100k matrices, double precision, Tesla V100 GPU

**Hierarchical Linear Solvers on GPU clusters**
DMRG++ Acceleration using MATEDOR Batched computations

**Density Matrix Renormalization Group DMRG++**
Hierarchical Linear Solver, 2 P100 GPUs per node

**Deep Neural Networks and Data Analytics**
Batched DGEMM acceleration on V100 GPU

WITH SUPPORT FROM

This work is partially supported by NSF Grant No. OAC 1740250 and CSR 1514288, NVIDIA, and the Department of Energy under the Exascale Computing Project (17-SC-20-SC and LLNL subcontract under DOE contract DE-AC52-07NA27344).