

MOHAMMED ALFARHAN

farhan@icl.utk.edu | <https://farhanma.github.io>

EDUCATION	King Abdullah University of Science and Technology 2014-2019 PhD, Computer Science Topic: Unstructured Computations on Emerging Architectures Advisor: David E. Keyes
	King Abdullah University of Science and Technology 2012-2013 MSc, Computer Science
	King Faisal University 2007-2012 BSc, Computer Science
RESEARCH EXPERIENCE	Postdoctoral Researcher, The University of Tennessee, Knoxville 2019-Present Developing software libraries for solving linear algebra problems at scale [with Jack Dongarra] <ul style="list-style-type: none">• SLATE: Software for Linear Algebra Targeting Exascale• MAGMA: Matrix Algebra on GPU and Multicore Architectures PhD Student, King Abdullah University of Science and Technology 2013-2019 Developed performance-centric algorithms for high performance computing [with David E. Keyes] Directed Research, King Abdullah University of Science and Technology 2012-2013 Developed combinatorial algorithms for multi-pruning of decision trees [with Mikhail Moshkov]
INDUSTRIAL EXPERIENCE	Co-founder and Software Consultant, RoboCrop Systems 2017-2019 Worked as a robotics software engineer and full stack developer for a robotics solutions startup Software Engineer, Saudi Electricity Company 2012 Developed distributed systems to monitor and detect anomalies in the reading meters Software Engineer Intern, Saudi Aramco Summer 2011 Developed a distributed key-value store system to monitor IT incidents and changes requests Software Engineer Intern, Saudi Aramco Summer 2010 Developed a database system to store reports on IT problems for further processing with ease
TEACHING EXPERIENCE	Teaching Assistant, King Abdullah University of Science and Technology 2014-2018 Worked as a TA for High Performance Computing (AMCS 312) course [with David E. Keyes] <ul style="list-style-type: none">• KAUST: Fall 2014, Fall 2015, Fall 2016, Fall 2017, Fall 2018• Saudi Aramco: Fall 2018• Blue Waters Online Courses: Fall 2016 (Introduction to High Performance Computing)
PUBLICATIONS	Journal Articles <ul style="list-style-type: none">• M. Al Farhan, A. Charara, M. Gates, A. YarKhan, D. Sukkari, J. Kurzak, and J. Dongarra. <i>GPU-Optimized Banded Dense Matrix Computations at Scale</i>. Under review, SISC 2020• A. Charara, M. Al Farhan, M. Gates, A. YarKhan, D. Sukkari, J. Kurzak, and J. Dongarra. <i>Object-Oriented Framework for Scaling Up Dense Matrix Computations on Distributed Heterogeneous Hardware</i>. Under review, IEEE TPDS 2020• M. Al Farhan, A. Abdelfattah, S. Tomov, M. Gates, D. Sukkari, A. Haidar, R. Rosenberg, and J. Dongarra. <i>MAGMA Templates for Scalable Linear Algebra on Emerging Architectures</i>, IJHPCA 2020• M. Abduljabbar, M. Al Farhan, N. Al-Harthi, R. Chen, R. Yokota, H. Bagci, and D. Keyes. <i>Extreme Scale FMM-Accelerated Boundary Integral Equation Solver for Wave Scattering</i>, SISC 2019 [Code: https://ecrc.github.io/bemfmm/]• M. Al Farhan and D. Keyes. <i>Optimizations of Unstructured Aerodynamics Computations for Many-core Architectures</i>, IEEE TPDS 2018 [Code: https://ecrc.github.io/kfun3d/]• M. Al Farhan, D. Kaushik, and D. Keyes. <i>Unstructured Computational Aerodynamics on Many Integrated Core Architecture</i>, Parallel Computing 2016

Conference Papers

- **M. Al Farhan**, M. Abduljabbar, H. Bagci, and D. Keyes. *ExaBEM: Exascale Boundary Element Method Solver for Acoustic Simulation*. To be submitted, ACM PASC 2021
- **M. Al Farhan** and D. Keyes. *GPU-Optimized Unstructured Computations*. To be submitted, ACM PPOPP 2021
- M. Abduljabbar, **M. Al Farhan**, R. Yokota, and D. Keyes. *Performance Evaluation of Computation and Communication Kernels of the Fast Multipole Method on Intel Manycore Architecture*, Euro-Par 2017
- H. AbouEisha, **M. Al Farhan**, I. Chikalov, and M. Moshkov. *An Algorithm for Reduct Cardinality Minimization*, IEEE GrC 2013 [Code: <https://farhanma.github.io/MinReduct/>]

Technical Reports

- M. Gates, **M. Al Farhan**, D. Sukkari, A. YarKhan, and J. Dongarra. *SLATE Working Note 15: Implementing Generalized Hermitian Eigenvalue Solvers*, Innovative Computing Laboratory Technical Report ICL-UT-20-02, 2020
- M. Gates, A. Charara, A. YarKhan, D. Sukkari, **M. Al Farhan**, and J. Dongarra. *SLATE Working Note 14: Performance Tuning SLATE*, Innovative Computing Laboratory Technical Report ICL-UT-20-01, 2020
- A. Charara, M. Gates, J. Kurzak, A. YarKhan, **M. Al Farhan**, and J. Dongarra. *SLATE Working Note 11: SLATE Developers' Guide*, Innovative Computing Laboratory Technical Report ICL-UT-19-02, 2020

ORAL/POSTER PRESENTATIONS

- Unstructured Computations on Emerging Architectures
 - SIAM CSE 2019, Spokane, Washington
- BEMFMM: An Extreme Scale FMM-Accelerated BIE Solver for Wave Scattering
 - SIAM CSE 2019, Spokane, Washington
 - Intel IXPUG 2018, KAUST
 - SIAM PP 2018, Tokyo, Japan
- Optimizations of Unstructured Aerodynamics Computations for Intel KNL Hardware
 - Intel IXPUG 2018, KAUST
 - SIAM PP 2018, Tokyo, Japan
 - Intel HPC Developer Conference 2017, Denver, Colorado
 - PCCFD Workshop 2017, KAUST
 - HPC Saudi Conference 2017, KAUST [best poster award]
 - SIAM CSE 2017, Atlanta, Georgia
 - SHAXC-3 Workshop 2017, KAUST
- Performance Evaluation of Fast Multipole Method on Intel Manycore Architecture
 - Euro-Par 2017, Santiago de Compostela, Spain
 - ISC 2017, Frankfurt, Germany
- Implicit Unstructured Computational Aerodynamics on MIC Architecture
 - ParCFD 2014, Trondheim, Norway
 - SHAXC-2 Workshop 2014, KAUST

SERVICE

- **Program Committee:** ICCS 2020
- **Reviewer:** ACM/IEEE SC 2015, Euro-Par 2016, IEEE Cluster 2016, PLOS One 2018, IJHPCA 2018, IEEE IPDSPS 2019, ACM TOPC 2019, Parallel Computing 2019, and ACM PASC 2020
- **Artifact Evaluator:** ACM PPOPP 2016
- **Member:** KAUST IEEE Student Chapter (2012-2013), KAUST Graduate Council (2012-2014), KAUST ACM/SIAM Student Chapter (2012-2019), and KAUST Code Clinic (2014-2019)
- **Lecturer:** Gave several tutorials on: Python Programming (Spring 2014 and 2015), Fundamentals of High Performance Computing (Summer 2014 and 2015), PETSc: Portable, Extensible Toolkit for Scientific Computation (Summer 2016), and Version Control using Git (Fall 2020)

PROGRAMMING

- **Languages:** C/C++, Python, Java, Unix Shell, MATLAB, L^AT_EX
- **Models:** MPI, OpenMP, CUDA, pThreads, TBB, Intrinsics
- **Tools:** Make, CMake, Autotools, perf, Valgrind, gdb, git