Accelerating Optimal Power Flow: Condensed-Space Interior-Point Methods and Automatic Differentiation on GPUs

Sungho Shin*, François Pacaud†, Mihai Anitescu*,
*Mathematics and Computer Science Division, Argonne National Laboratory
†Centre Automatique et Systèmes, Mines Paris - PSL

While GPUs have demonstrated impressive capabilities in various computing domains, their effectiveness in large-scale, constrained nonlinear optimization regimes, such as AC OPFs, has been limited due to the challenges associated with parallel factorization of indefinite sparse matrices commonly encountered within constrained optimization algorithms [1]. Although GPU computation can accelerate various other parts of the optimization process in a straightforward manner, including automatic differentiation (AD) and sparse matrix-vector multiplications, the slow data transfer between host and device memory hinders the ad-hoc implementation of GPU accelerations. Thus, to fully leverage the capabilities of modern GPU hardware, it is essential to implement a comprehensive computational framework on the GPU, incorporating AD, linear algebra, and optimization, while minimizing data transfers to and from host memory.

We present a comprehensive computational framework and the associated software implementations—an optimization solver MadNLP.jl and an AD tool SIMDiff.jl—for solving AC OPF problems on GPUs [7], [6]. Our approach utilizes the following techniques: (i) condensed-space interior-point methods (IPMs), (ii) an inequality relaxation strategy, (iii) sparse matrix factorization with a fixed pivot sequence, and (iv) a single-instruction, multiple-data (SIMD) abstraction of nonlinear programs. Notably, our method relaxes power flow equality constraints by allowing small violations, which enables expressing the Karush-Kuhn-Tucker (KKT) system entirely in the primal space (referred to as the condensation strategy). Although this strategy is not new [4], it has traditionally been considered less efficient than the standard full-space method due to increased fill-in in the sparse factorization. However, when implemented on GPUs, it offers the key advantage of guaranteeing positive definiteness in the condensed KKT system through the application of standard regularization techniques. This allows for the utilization of linear solvers with a fixed numerical pivot sequence (an efficient implementation available as part of CUDA library), facilitating efficient KKT system solutions on GPUs. Although this method is susceptible to ill-conditioning, our results demonstrate that the solver is robust enough to solve problems with a relative accuracy of up to 10^{-6}.

Furthermore, by leveraging the SIMD abstraction of nonlinear programs, which preserves the parallelizable structure in the model, the model functions and derivative evaluations can be parallelized, facilitating evaluations on the GPU. We show that the AC power flow model is particularly well-suited for this abstraction, as it involves repetitive expressions for each component of the model (e.g., buses, lines, generators), and the number of computational patterns does not increase with the network's size. As observed by the impressive performance of Gravity [3], exploiting this structure significantly accelerates model function and derivative evaluations.

We will present comprehensive numerical benchmark results to demonstrate the efficiency of our proposed approach. The proposed method is implemented in our packages, MadNLP.jl and SIMDiff.jl, with the solution of the KKT system being carried out using the external CUSOLVER library. We compare our method against the standard CPU approach [8] and the recently developed reduced-space interior-point method on GPUs [5] using the standard benchmark library [2]. Our initial results suggest that the proposed framework has the potential to accelerate the solution of AC OPF problems by an order of magnitude compared to existing tools, such as Ipopt (interfaced with MATPOWER or PowerModels.jl), especially for large-scale instances, up to moderate tolerances (10^{-6}). We also discuss the future extensibility of our method for more complex optimization tasks, such as multi-period, security-constrained, and joint transmission-distribution optimization.

REFERENCES

[1] Mihai Anitescu, Kibaek Kim, Youngdae Kim, Adrian Maldonado, François Pacaud, Vishwas Rao, Michel Schanen, Sungho Shin, and Anirudh Subramanian. Targeting Exascale with Julia on GPUs for multiperiod optimization with scenario constraints. *SIAG/OPT Views and News*, 2021.

[2] Carleton Coffrin. pglib-opf. https://github.com/power-grid-lib/pglib-opf.

[3] Hassan Hijazi, Guanglei Wang, and Carleton Coffrin. Gravity: A mathematical modeling language for optimization and machine learning. 2018.

[4] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 2006.

[5] François Pacaud, Sungho Shin, Michel Schanen, Daniel Adrian Maldonado, and Mihai Anitescu. Condensed interior-point methods: porting reduced-space approaches on gpu hardware. *arXiv preprint arXiv:2203.11875*, 2022.

[6] Sungho Shin. SimDiff.jl. https://github.com/Simdiff/Simdiff.jl.

[7] Sungho Shin and Francois Pacaud. MadNLP.jl. https://github.com/MadNLP/MadNLP.jl.

[8] Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106:25–57, 2006.