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

Abstract—**This work presents an accelerated solution method for solving AC optimal power flow (OPF) problems on graphics processing units (GPUs). While GPUs have demonstrated impressive accelerated capabilities in various computing domains, thus far they have shown limited capabilities in large-scale, constrained nonlinear programming regime, due to the challenging nature of parallel factorization of indefinite sparse matrices commonly encountered within interior-point methods (IPMs) [1]. Although GPU computation can accelerate various other components of the optimization process, such as automatic differentiation and sparse matrix-vector multiplications, slow data transfer between host and device memory impedes 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 — including automatic differentiation, linear algebra, and optimization — minimizing data transfers from/to host memory.**

In this work, we present a computational framework with its associated software packages—MadNLP.jl[1] for the IPM solver and SIMDiff.jl[2] for the automatic differentiation—for solving AC OPF problems. Our approach leverages (i) condensed-space IPMs, (ii) inequality relaxation strategy, (iii) sparse matrix factorization with a fixed pivot sequence, and (iv) SIMD abstraction of nonlinear programs. Notably, our method relaxes power flow equality constraints by allowing small violations, which allows expressing the Karush-Kuhn-Tucker (KKT) system entirely in the primal space. While this condensation strategy is not a new concept [2], it has been considered less efficient compared to the standard full-space method, largely due to increased fill-in in the sparse factorization. When implemented on GPUs, however, it has the key advantage of guaranteeing a positive definite condensed KKT system upon the application of standard regularization techniques. This allows for the application of linear solvers with a fixed numerical pivot sequence (an efficient implementation available in CUDA), enabling efficient solution of KKT systems on GPUs. Although this method is prone to ill-conditioning, our results indicate that the solver is robust enough to solve problems up to a relative accuracy of 10^{-6}. Furthermore, the model functions and derivative evaluations can be parallelized by exploiting the single-instruction, multiple-data (SIMD) abstraction of nonlinear programs, which aims to preserve the parallelizable structure in the model to facilitate the evaluations on the GPU. We demonstrate that the AC power flow model is particularly suitable for such an abstraction, as it involves repetitive expressions for each component of the models (e.g., buses, lines, generators), and the number of computational patterns does not grow with the network's size. Exploiting this structure provides significant acceleration of model function and derivative evaluations.

We implement the entire process in our optimization solver, MadNLP.jl, and automatic differentiation tool, SIMDiff.jl. Only the solution of the condensed KKT system is carried out using the external CUSOLVER library. Our numerical experiments reveal 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. We will present comprehensive numerical results demonstrating the effectiveness of our accelerated OPF framework. Our method can be extended further and help solve more complex variants of OPF problems, such as multi-period, security-constrained, joint transmission-distribution problems.

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

[1] https://github.com/MadNLP/MadNLP.jl
[2] https://github.com/sshin23/SIMDiff.jl