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

Sungho Shin and Mihai Anitescu
Mathematics and Computer Science Division
Argonne National Laboratory
Lemont, IL, USA
sshin@anl.gov, anitescu@mcs.anl.gov

François Pacaud
Centre Automatique et Systèmes
Mines Paris - PSL
Paris, France
francois.pacaud@minesparis.psl.eu

***Abstract*—This paper presents a computational framework for solving AC optimal power flow (OPF) problems using GPUs. Despite the impressive capabilities of GPUs in various computing domains, their application in AC OPF has been limited by various challenges, including sparse matrix factorization and automatic differentiation. To address these issues, we propose two strategies: (i) a condensed-space interior-point method (IPM) with inequality relaxation, and (ii) a single-instruction, multiple-data (SIMD) abstraction of nonlinear programs. By relaxing equality constraints and condensing the Karush-Kuhn-Tucker system to the primal space, we enable efficient sparse matrix factorization on GPUs without numerical pivoting. Further, we introduce the SIMD abstraction for parallelized derivative evaluations on GPUs. The paper provides numerical benchmark results, comparing our approaches with state-of-the-art tools such as JuMP, AMPL, and Ipopt, demonstrating an order of magnitude speedup. The paper concludes by discussing the future extensibility.**

I. Introduction

While graphics processing units (GPUs) have showcased impressive capabilities in various computing domains, their utilization in large-scale constrained nonlinear optimization regimes, such as alternating current (AC) optimal power flow (OPF) problems, has been somewhat limited. This limitation stems from 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, 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. To fully leverage the capabilities of modern GPU hardware, it is essential to implement a comprehensive computational framework on the GPU that incorporates AD, linear algebra, and optimization while minimizing data transfers to and from host memory.

This paper presents a comprehensive computational framework and the associated software implementations for solving AC OPF problems on GPUs. Our approach utilizes the following techniques: (i) condensed-space interior-point methods (IPMs) with an inequality relaxation strategy, (ii) sparse matrix factorization with a fixed pivot sequence, and (iii) a single-instruction, multiple-data (SIMD) abstraction of nonlinear programs. Specifically, 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 through the condensation procedure. Although this strategy is not new (see Nocedal and Wright, 2006), 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, which in turn, allows for the utilization of linear solvers with a fixed numerical pivot sequence (so-called refactorization). An efficient implementation of the sparse refactorization is available as part of the CUDA library, facilitating the implementation of 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 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, thereby facilitating evaluations on the GPU. We demonstrate that the AC power flow model is particularly well-suited for this abstraction as it involves repetitive expressions for each type of component in the model (e.g., buses, lines, generators), and the number of computational patterns does not increase with the network's size. This structure has been effectively utilized by Gravity (Hijazi et. al. 2018), demonstrating significant acceleration in model evaluations.

We present comprehensive numerical benchmark results to demonstrate the efficiency of our proposed approach. Our method is implemented in our packages, a nonlinear optimization solver MadNLP.jl and an automatic differentiation tool 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 (Wächter and Beigler, 2006) and the recently developed reduced-space interior-point method on GPUs (Pacaud et. al., 2022) using the standard data available in pglib-opf. Preliminary results indicate that our proposed framework has the potential to accelerate the solution of AC OPF problems up to moderate tolerances (10^{-6}) 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 conclude with a discussion on the future extensibility of our method for more complex optimization tasks, such as multi-period, security-constrained, and joint transmission-distribution optimization.

Contributions: The contribution of this paper is three-fold. First, we present a condensed-space interior point method with inequality relaxation strategy is a general framework for solving sparse, large-scale nonlinar optimization problems on GPUs. Second, we present a SIMD abstraction of nonlinear program, which facilitates the implementation of parallel automatic differentiation on GPUs. Third, we present a comprehensive numerical benchmark result for the proposed two strategy using the standard test cases, thereby demonstrating the effectiveness of the two proposed methods. To the best of our knowledge, our approach is the fastest algorithm to solve AC OPF problems to 10^{-5} accuracy.

Organization: The paper is organized as follows. In the remainder of the current section, we introduce the mathematical notation. In Section II, we provide general preliminary knowledge on numerical optimization and GPU computing. In Section III, we present the SIMD abstraction strategy for large-scale nonlinear programs and their advantages in terms of implementing parallel automatic differentiation. Section IV presents the optimization algorithm under study, the condensed-space interior point method wiht inequality relaxation strategy. Section V presents the numerical results, comparing our approach with other state-of-the-art solution methods on GPUs. Finally, conclusions and future outlooks are given in Section VI.

Notation: We denote the set of real numbers and the set of integers by $\mathbb{R}$ and $\mathbb{I}$. We let $[M] := \{1, 2, \cdots, M\}$.

II. Preliminaries

This section introduces two key background information: GPU computing and numerical optimization for large-scale nonlinear optimzation problems.

A. GPU Computing

GPUs are particularly effective in performing the following operations:

$$y \leftarrow [g(x; q_j)]_{j\in[J]} \tag{1a}$$

$$y \leftarrow y + \sum_{k\in[K]} h(x; s_k) \tag{1b}$$

$$o \leftarrow \sum_{i\in[I]} f(x; p_i), \tag{1c}$$

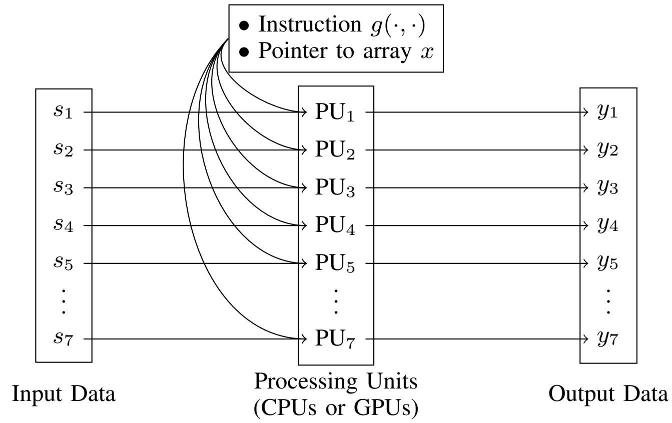

Fig. 1. A schematic description of SIMD parallelism

B. Numerical Optimization

The state-of-the-art solution strategies for AC OPF problems rely on numerical optimziation software, either general-purpose [?] or specialized [?]. Three main pillars of modern nonlinear optimization software include not only optimization solver, but also automatic differentiation tool (often provided by or interfaced with algebraic modeling language) and linear solvers. In fact, most of the expensive computational efforts are delegated to the external linear solver and automatic differentiation library. Optimization solver orchestrates the operation of these tools to drive the solution iterate to the stationary point of the optimization problem.

a) Optimization Solvers: Optimization solvers

b) Automatic Differentation: For the efficient solution of the optimization problems, it is crucial to be have the ability to

c) Linear Solvers:

III. SIMD Abstraction of Nonlinear Programs

$$\min_{x^L \le x \le x^U} \sum_{\ell\in[L]} \sum_{i\in[I_k]} f^{(\ell)}(x; p_i^{(\ell)}) \tag{2a}$$

$$\text{s.t. } \forall m \in [M]: \tag{2b}$$

$$g_\flat^{(m)} \le \left[g^{(m)}(x; q_j)\right]_{j\in[J]} + \sum_{n\in[N_m]} \sum_{k\in[K_n]} h^{(n)}(x; s_k^{(n)}) \le g_\sharp^{(m)}, \tag{2c}$$

IV. Condensed-Space Interior Point Methods

V. Numerical Results

VI. Conclusions and Future Outlook

References

[1] M. Anitescu, K. Kim, Y. Kim, A. Maldonado, F. Pacaud, V. Rao, M. Schanen, S. Shin, and A. Subramanian, "Targeting Exascale with Julia on GPUs for multiperiod optimization with scenario constraints," *SIAG/OPT Views and News*, 2021. [Online]. Available: http://wiki.siam.org/siag-op/images/siag-op/e/e8/ViewsAndNews-29-1.pdf