Solving Large-scale Optimal Power Flow Problems With GPU Accelerators

INFORMS 2021

François Pacaud, Sungho Shin, Adrian Maldonado, Michel Schanen
PI: Mihai Anitescu

Argonne National Laboratory
Mathematics and Computer Science Division

October, 2021

Who are we?

We are a team of enthusiastic computational mathematicians at Argonne National Lab

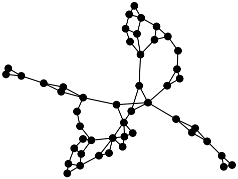
EXASCALE COMPUTING PROJECT

ExaSGD project

- Optimizing Stochastic Grid Dynamics at ExaScale
- Leverage new GPU-centric HPC architectures

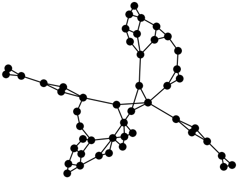
Question: How to solve optimal power flow problems at exascale?

Solving Optimal Power Flow on GPU is easy, huh?



- Graphs are the natural abstraction for power networks, but come with *unstructured sparsity*
- OPF formulate as large-scale nonlinear nonconvex optimization problems

Solving Optimal Power Flow on GPU is easy, huh?



- Graphs are the natural abstraction for power networks, but come with *unstructured sparsity*
- OPF formulate as large-scale nonlinear nonconvex optimization problems

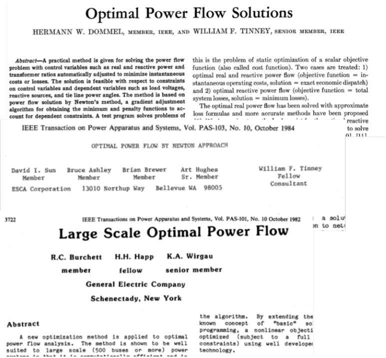
But ... Large-scale optimization solvers rely on sparse solvers!

State-of-the-art for OPF: *Interior-Point Method* (IPM)

- Newton method with very ill-conditioned linear systems
- Efficient IPM requires indefinite sparse direct inertia revealing solvers (HSL, Pardiso)...
- Sparse linear libraries on GPU are not mature (yet!) (Tasseff et al., 2019)

Back to the future: Revisiting reduced space method for OPF

A brief (and partial) history of the resolution of OPF (nonlinear optimization only)



Optimal Power Flow Solutions

HERMANN W. DOMMEL, MEMBER, IEEE, AND WILLIAM F. TINNEY, SENIOR MEMBER, IEEE

Abstract—A practical method is given for solving the power flow problem with control variables such as real and reactive power and transformer ratios automatically adjusted to minimize instantaneous costs or losses. The solution is feasible with respect to constraints on control variables and dependent variables such as load voltages, reactive sources, and tie line power angles. The method is based on power flow solution by Newton's method, a gradient adjustment algorithm for obtaining the minimum and penalty functions to account for dependent constraints. A test program solves problems of

this is the problem of static optimization of a scalar objective function (also called cost function). Two cases are treated: 1) optimal real and reactive power flow (objective function = instantaneous operating costs, solution = exact economic dispatch) and 2) optimal reactive power flow (objective function = total system losses, solution = minimum losses).

The optimal real power flow has been solved with approximate loss formulas and more accurate methods have been proposed

IEEE Transaction on Power Apparatus and Systems, Vol. PAS-103, No. 10, October 1984

OPTIMAL POWER FLOW BY NEWTON APPROACH

David I. Sun, Member; Bruce Ashley, Member; Brian Brewer, Member; Art Hughes, Sr. Member; William F. Tinney, Fellow, Consultant

ESCA Corporation 13010 Northup Way Bellevue WA 98005

3722 IEEE Transactions on Power Apparatus and Systems, Vol. PAS-101, No. 10 October 1982

Large Scale Optimal Power Flow

R.C. Burchett, member; H.H. Happ, fellow; K.A. Wirgau, senior member

General Electric Company

Schenectady, New York

Abstract

A new optimization method is applied to optimal power flow analysis. The method is shown to be well suited to large scale (500 buses or more) power

the algorithm. By extending the known concept of "basic" so programming, a nonlinear objecti optimized (subject to a full constraints) using well develope technology.

- 1962: introduction of the OPF problem by Carpentier
- 1968: Reduced Gradient method Dommel and Tinney (1968)
- 1972: Generalized Reduced Gradient Peschon et al. (1972)
- 1982: SQP method for OPF Burchett et al. (1982)
- 1984: OPF by Newton approach Sun et al. (1984)
- 1994: Primal-Dual Interior-Point Granville (1994)

Formulating the OPF problem

We adopt the *polar formulation*

- **Variables**
  $\boldsymbol{z} = (\boldsymbol{v}, \boldsymbol{\theta}, \boldsymbol{p}_g, \boldsymbol{q}_g) \in \mathbb{R}^{2\times(n_b+n_g)}$
  - Voltage magnitudes $\boldsymbol{v} \in \mathbb{R}^{n_b}$
  - Voltage angles $\boldsymbol{\theta} \in \mathbb{R}^{n_b}$
  - Active power generations $\boldsymbol{p}_g \in \mathbb{R}^{n_g}$
  - Reactive power generations $\boldsymbol{q}_g \in \mathbb{R}^{n_g}$
- **Objective**
  - Minimize costs of power generations

$$F(\boldsymbol{z}) = \sum_{g=1}^{n_g} c_2^g p_g^2 + c_1^g p_g$$

- **Constraints**
  - Bounds $\boldsymbol{z}^\flat \leq \boldsymbol{z} \leq \boldsymbol{z}^\sharp$

$$\boldsymbol{z} \in \mathcal{Z}$$

  - Power-flow equality constraints

$$G(\boldsymbol{z}) = 0$$

  - Line-flow inequality constraints

$$H(\boldsymbol{z}) \leq 0$$

Original OPF

$$\begin{aligned} \min_{\boldsymbol{z}\in\mathcal{Z}} \quad & F(\boldsymbol{z}) \\ \textbf{subject to} \quad & G(\boldsymbol{z}) = 0 \\ & H(\boldsymbol{z}) \leq 0 \end{aligned} \tag{OPF}$$

Physically-constrained OPF

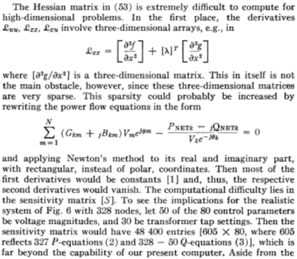
We reorder $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{u})$ with

- a state $\boldsymbol{x} = (\boldsymbol{\theta}^{pv}, \boldsymbol{\theta}^{pq}, \boldsymbol{v}^{pq})$
- a control $\boldsymbol{u} = (\boldsymbol{v}^{ref}, \boldsymbol{v}^{pv}, \boldsymbol{p}_g^{pv})$

and consider the equivalent formulation

$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{u}} \quad & F(\boldsymbol{x}, \boldsymbol{u}) \\ \textbf{subject to} \quad & \boldsymbol{x} \in \mathcal{X}, \ \boldsymbol{u} \in \mathcal{U} \\ & G(\boldsymbol{x}, \boldsymbol{u}) = 0 \\ & H(\boldsymbol{x}, \boldsymbol{u}) \leq 0 \end{aligned} \tag{1}$$

Our plan of action



The Hessian matrix in (53) is extremely difficult to compute for high-dimensional problems. In the first place, the derivatives $\mathfrak{L}_{uu}$, $\mathfrak{L}_{xx}$, $\mathfrak{L}_{xu}$ involve three-dimensional arrays, e.g., in

$$\mathfrak{L}_{xx} = \left[\frac{\partial^2 f}{\partial x^2}\right] + [\lambda]^T \left[\frac{\partial^2 g}{\partial x^2}\right]$$

where $[\partial^2 g/\partial x^2]$ is a three-dimensional matrix. This in itself is not the main obstacle, however, since these three-dimensional matrices are very sparse. This sparsity could probably be increased by rewriting the power flow equations in the form

$$\sum_{m=1}^{N} (G_{km} + {}_jB_{km}) V_m e^{j\theta m} - \frac{P_{\mathrm{NET}k} - {}_jQ_{\mathrm{NET}k}}{V_k e^{-j\theta_k}} = 0$$

and applying Newton's method to its real and imaginary part, with rectangular, instead of polar, coordinates. Then most of the first derivatives would be constants [1] and, thus, the respective second derivatives would vanish. The computational difficulty lies in the sensitivity matrix $[S]$. To see the implications for the realistic system of Fig. 6 with 328 nodes, let 50 of the 80 control parameters be voltage magnitudes, and 30 be transformer tap settings. Then the sensitivity matrix would have 48 400 entries [605×80, where 605 reflects 327 P-equations (2) and $328 - 50$ Q-equations (3)], which is far beyond the capability of our present computer. Aside from the

Figure: Dommel and Tinney (1968)

We port on GPUs the reduced space method of Dommel and Tinney (1968) (revisited recently in Kardos et al. (2020))

Two steps:

1. Projection on the reduced space
   - Implementation of a differentiable power flow solver on GPU
   - Evaluation of the reduced Hessian in batch
2. Resolution in the reduced space
   - Penalty methods of Dommel and Tinney (1968), revisited with an Augmented Lagrangian algorithm
   - Dense KKT system solved directly on the GPU, using a Schur complement approach

Outline

Projection

Resolution

Projecting the problem into the power flow manifold

- Remember that the power flow equality constraints write $G(\boldsymbol{x}, \boldsymbol{u}) = 0$
- If $\nabla_x G$ is non-singular, then Implicit Function theorem applies:
  For each u, there exists a local function $\underline{x}(u)$ such that

$$G(\underline{x}(u), u) = 0$$

- Numerically, the nonlinear equation is inverted with Newton-Raphson

Reduced problem

Let $f(\boldsymbol{u}) := F(\underline{x}(\boldsymbol{u}), \boldsymbol{u})$ and $h(\boldsymbol{u}) := H(\underline{x}(\boldsymbol{u}), \boldsymbol{u})$. Problem (OPF) is equivalent to

$$\min_{\boldsymbol{u}} \; f(\boldsymbol{u}) \quad \text{s.t.} \quad h(\boldsymbol{u}) \leq 0 \,, \quad x(\boldsymbol{u}) \in \mathcal{X}, \quad \boldsymbol{u} \in \mathcal{U}$$

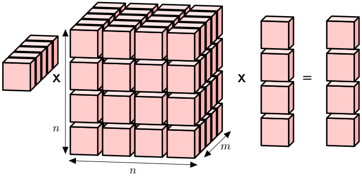
Reduced gradient

If $F : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$ is a differentiable function,
then the function $f(\boldsymbol{u}) := F(\underline{x}(\boldsymbol{u}), \boldsymbol{u})$ is *differentiable*, with

$$\nabla f(\boldsymbol{u}) = \underbrace{\nabla_u F}_{n_u} + (\underbrace{\nabla_u G}_{n_x \times n_u})^\top \underbrace{\boldsymbol{\lambda}}_{n_x} \quad \text{with} \quad (\underbrace{\nabla_x G}_{n_x \times n_x})^\top \boldsymbol{\lambda} = -\underbrace{\nabla_x F}_{n_x}$$

The vector $\boldsymbol{\lambda} \in \mathbb{R}^{n_x}$ is the first-order adjoint

Reduced Hessian: dense, dense, dense!



Can we extract second-order information as well? Yes!

- We derive two first-order adjoints ψ and $\boldsymbol{z}$, using the *adjoint-adjoint* method (Wang et al., 1992)
- Involve only *Hessian-vector products*!
- Reduced Hessian $\nabla^2 f$ is *dense* (dimension $n_u \times n_u$)

Reduced Hessian

Let $\boldsymbol{w} \in \mathbb{R}^{n_u}$ be a vector and $\widehat{G}$ the first-order residual:

$$\widehat{G}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\lambda}) := \nabla_x F(\boldsymbol{x}, \boldsymbol{u}) + \nabla_x G(\boldsymbol{x}, \boldsymbol{u})^\top \boldsymbol{\lambda} \qquad (\approx 0)$$

The Hessian-vector product $(\nabla^2 f)\boldsymbol{w}$ is equal to

$$(\nabla^2 f)\boldsymbol{w} = (\nabla^2_{\boldsymbol{uu}} F)\, \boldsymbol{w} + \boldsymbol{\lambda}^\top (\nabla^2_{\boldsymbol{uu}} G)\, \boldsymbol{w} + (\nabla_{\boldsymbol{u}} G)^\top \psi + (\nabla^2_{\boldsymbol{ux}} F)^\top \boldsymbol{z} + \boldsymbol{\lambda}^\top (\nabla^2_{\boldsymbol{ux}} G)^\top \boldsymbol{z}$$

with the second-order adjoints $(\boldsymbol{z}, \psi)$ solutions of the two sparse linear systems

$$\begin{cases} (\nabla_{\boldsymbol{x}} G) \quad \boldsymbol{z} = -(\nabla_{u} G)\ \boldsymbol{w} \\ (\nabla_{\boldsymbol{x}} G)^\top \psi = -(\nabla_{u} \widehat{G})\boldsymbol{w} - (\nabla_{\boldsymbol{x}} \widehat{G})\boldsymbol{z} \ , \end{cases}$$

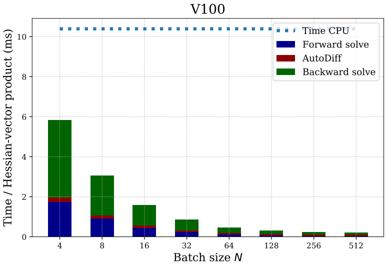
Computing the reduced Hessian on the GPU

Parallel computation

- ✓ We evaluate the Hessian-vector products $(\nabla^2 f)\boldsymbol{w}$ in batch
- ✓ Callbacks for $\nabla^2 F$ and $\nabla^2 G$ evaluated using Forward-over-Reverse Autodiff, (batch automatic differentiation implemented on GPU)
- ✓ Sparse linear systems solved in batch with `cusolverRF`

Results on case9241pegase:

i) Reduced space: CPU versus GPU



ii) Reduced space versus full space

lib	device	space	time
AMPL	CPU	full space	130ms
ExaPF	GPU	reduced space	350ms

Table: Time to evaluate the Hessian of the Lagrangian

Outline

Projection

Resolution

Augmented Lagrangian formulation

Where are we?

In the reduced space, the OPF writes as a nonlinear problem

$$\min_{\boldsymbol{u}\geq 0} \; f(\boldsymbol{u}) \quad \text{s.t.} \quad c(\boldsymbol{u}) \leq 0$$

with

- Bound constraints $\boldsymbol{u} \geq 0$
- Inequality constraints $c(\boldsymbol{u}) \leq 0$
  (the functional $c : \mathbb{R}^{n_u} \to \mathbb{R}^m$ concatenates the line constraints $h(\boldsymbol{u}) \leq 0$ and the state constraints $x(\boldsymbol{u}) \in \mathcal{X}$ to get a problem in standard form)

- Amenable to resolution with interior-point? But...
  Jacobian $\nabla c(\boldsymbol{u}) \to m$ linear systems ; Hessian $\nabla^2 c(u) \to 2m \times n$ linear systems...
- Dommel and Tinney (1968) used quadratic penalties in their resolution algorithm!

Smooth Augmented Lagrangian formulation

Let $\boldsymbol{s} \in \mathbb{R}^m$ a slack variable, $\rho^k > 0$ a penalty, and a multiplier $\boldsymbol{y}^k \in \mathbb{R}^m$.

$$\min_{\boldsymbol{u}\geq 0, \boldsymbol{s}\geq 0} \; L_\rho(\boldsymbol{u}, \boldsymbol{s}; \boldsymbol{y}^k) := f(\boldsymbol{u}) + \langle \boldsymbol{y}^k, c(\boldsymbol{u}) - \boldsymbol{s} \rangle + \frac{\rho^k}{2} \| c(\boldsymbol{u}) - \boldsymbol{s} \|^2$$

Resolution of the Augmented Lagrangian subproblems

- × Active set methods not amenable to GPUs (expensive reordering)
- ✓ Use Interior-point method (IPM) instead! (even if poor warm-starting...)

IPM-Augmented Lagrangian formulation

$$\min_{\boldsymbol{u},\boldsymbol{s}} \; \psi_\mu(\boldsymbol{u},\boldsymbol{s},\boldsymbol{y}^k) := L_\rho(\boldsymbol{u},\boldsymbol{s};\boldsymbol{y}^k) - \mu \sum_{i=1}^{n_u} \log(u_i) - \mu \sum_{i=1}^{m} \log(s_i) \quad \text{(IPM-EqAugLag)}$$

Denote by $\boldsymbol{v} := (\boldsymbol{u},\boldsymbol{s})$ the primal variable,
and $\boldsymbol{z}$ the dual variable associated to bound-constraints $\boldsymbol{v} \geq 0$
We get the primal-dual equations (see Nocedal and Wright (2006)):

$$\begin{cases} \nabla L_\rho(\boldsymbol{v};\boldsymbol{y}^k) - \boldsymbol{z} = 0 \\ VZ\boldsymbol{e} - \mu\boldsymbol{e} = 0 \end{cases} \implies \begin{bmatrix} \nabla^2 L_\rho & -I \\ Z & V \end{bmatrix} \begin{bmatrix} \boldsymbol{d}_v \\ \boldsymbol{d}_z \end{bmatrix} = - \begin{bmatrix} \nabla L_\rho(\boldsymbol{v};\boldsymbol{y}^k) - \boldsymbol{z} \\ VZ\boldsymbol{e} - \mu\boldsymbol{e} \end{bmatrix}$$

simplifies as $\left[\nabla^2 L_\rho + \Sigma\right] \boldsymbol{d}_v = -\nabla_v \psi_\mu(\boldsymbol{v};\boldsymbol{y}^k)$ with $\Sigma = V^{-1}Z$ diagonal matrix

But...still, matrix $\nabla^2 L_\rho$ has size $(n_u + m) \times (n_u + m)$...

Solving the KKT system with a Schur complement approach

Looking more closely at the Hessian $\nabla^2 L_\rho$

$$\nabla^2 L_\rho = \begin{bmatrix} H_{uu} + \rho A_u^\top A_u & -\rho A_u^\top \\ -\rho A_u & \rho I \end{bmatrix} \in \mathbb{R}^{(n_u+m)\times(n_u+m)}$$

- reduced Hessian (dense) $H_{uu} = \nabla^2 f(\boldsymbol{u}) + \sum_{i=1}^m y_i \nabla^2 h(\boldsymbol{u})$
- and reduced Jacobian (dense) $A_u = \nabla h(\boldsymbol{u})$

Theorem

Let $\boldsymbol{d}_v = (\boldsymbol{d}_u, \boldsymbol{d}_s)$ and $\boldsymbol{g}_v = (\boldsymbol{g}_u, \boldsymbol{g}_s)$.
The Newton-step $\left[\nabla^2 L_\rho + \Sigma\right] \boldsymbol{d}_v = -\boldsymbol{g}_v$ is equivalent to

$$\begin{bmatrix} S_{uu} & 0 \\ -\rho A_u & \Sigma_s + \rho I \end{bmatrix} \begin{bmatrix} \boldsymbol{d}_u \\ \boldsymbol{d}_s \end{bmatrix} = \begin{bmatrix} -\boldsymbol{g}_u + \rho A_u^\top [\Sigma_s + \rho I]^{-1} \boldsymbol{g}_s \\ -\boldsymbol{g}_s \end{bmatrix}$$

with S_{uu} the Schur-complement matrix of $\left[\nabla^2 L_\rho + \Sigma\right]$:

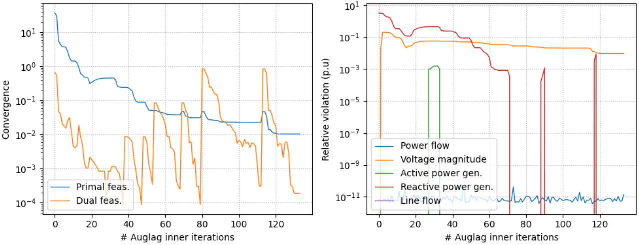
$$S_{uu} = H_{uu} + \Sigma_u + A_u^\top \left[\rho - \rho^2 [\Sigma_s + \rho I]^{-1}\right] A_u$$

Now, it remains just to factorize S_{uu} (with size $n_u \times n_u$)!

Numerical results

Numerical settings

- Algorithm implemented inside the MadNLP solver (Shin et al., 2020)
- IPM warmstarted following (Ma et al., 2021)
- Reduced Hessian evaluated using the projection algorithm we presented before
- In practice, dense matrix S_{uu} is factorized on the GPU with a Bunch-Kaufman factorization (as implemented in cuSOLVER)



- Total running time: 160s

Conclusion

Thanks for listening!

- **Achievements**
  - We have revisited the reduced gradient method of Dommel and Tinney, with second-order information
  - We have developed a custom Augmented Lagrangian algorithm, and exploited the structure of the KKT system

- **Perspectives**
  - Prove formally the convergence of the algorithm
  - Adapt the algorithm to a real-time optimization setting

Slides available at: https://frapac.github.io/pdf/INFORMS_2021.pdf

References I

Burchett, R., Happ, H., and Wirgau, K. (1982). Large scale optimal power flow. IEEE Transactions on Power Apparatus and Systems, (10):3722–3732.

Dommel, H. and Tinney, W. (1968). Optimal Power Flow Solutions. IEEE Transactions on Power Apparatus and Systems, PAS-87(10):1866–1876.

Granville, S. (1994). Optimal reactive dispatch through interior point methods. IEEE Transactions on power systems, 9(1):136–146.

Kardos, J., Kourounis, D., and Schenk, O. (2020). Reduced-space interior point methods in power grid problems. arXiv preprint arXiv:2001.10815.

Ma, D., Orban, D., and Saunders, M. A. (2021). A julia implementation of algorithm ncl for constrained optimization. arXiv preprint arXiv:2101.02164.

Nocedal, J. and Wright, S. J. (2006). Numerical optimization. Springer series in operations research. Springer, New York, 2nd ed edition. OCLC: ocm68629100.

Peschon, J., Bree, D. W., and Hajdu, L. P. (1972). Optimal power-flow solutions for power system planning. Proceedings of the IEEE, 60(1):64–70.

Shin, S., Coffrin, C., Sundar, K., and Zavala, V. M. (2020). Graph-based modeling and decomposition of energy infrastructures. arXiv preprint arXiv:2010.02404.

Sun, D. I., Ashley, B., Brewer, B., Hughes, A., and Tinney, W. F. (1984). Optimal power flow by newton approach. IEEE Transactions on Power Apparatus and systems, (10):2864–2880.

Tasseff, B., Coffrin, C., Wächter, A., and Laird, C. (2019). Exploring benefits of linear solver parallelism on modern nonlinear optimization applications. arXiv preprint arXiv:1909.08104.

Wang, Z., Navon, I. M., Le Dimet, F.-X., and Zou, X. (1992). The second order adjoint analysis: theory and applications. Meteorology and atmospheric physics, 50(1):3–20.