Partial Dual Method for Solving QCQPs

Justin Cardona
Engineering Physics Department, Polytechnique Montréal

This work implements a computationally inexpensive method to solve indefinite quadratically constrained quadratic programs (QCQPs). Using the Lagrange dual perspective, the need to use a barrier method to check for dual feasibility is circumvented using a *Partial Dual* that leverages one of the positive definite constraints. This method trades the tens of thousands of matrix-vector products needed in trace estimation solutions for order 10 inverse solves.

I. INTRODUCTION

Inverse design in photonics relies heavily on being able to investigate large numbers of unintuitive geometries. However, because the design parameter space is typically extremely large local methods are typically employed. As such the designs created by current programs provide little certainty on providing globally optimal results. There have been recent efforts to to establish bounds on the performance of these methods by using Lagrange duality. However, currently available optimization methods do not allow for computationally efficient means to perform the dual optimization[1]. This typically involves doing positive-definite checks using eigenvalue decomposition, which is not practical for extremely large systems. There is a promising method [2] that employs a *Partial Dual* perspective to alleviate some of the complexity. This approach considers another dual program to the primal by leveraging one of its positive definite constraints to avoid the need to do positive definite checks on the program as a whole. However, for large system sizes current methods for forming this problem are still infeasible. Thus the focus of this paper is to present a computationally inexpensive algorithm that can implement the partial dual perspective. The general idea is to use Padé approximants to fit to a positive definite constraint function using very few sample points. While generating the sample points is computationally expensive, working with the approximation is cheap. In section II the optimization problem is introduced along with an overview of why current methods are not computationally pratical. In section III, the workaround to this problem is presented.

II. BACKGROUND

In this work a numerical scheme is considered for performing inverse design. Firstly the volume where the device must lie is discretized (into a cartesian grid for example). A field in this perspective is represented by a vector with entries that correspond to the polarization current of each cell. For example, if a volume is divided into an $n \times n \times n$ grid then a polarization current can be represented using a vector in $\mathbb{C}^{3n^3}$. Additionally, they can be considered vectors in a Hilbert space with the following inner product:

$$\langle F|G\rangle = \int_{\mathbb{R}^3} \mathrm{d}^3x\, F^*(x)\cdot G(x) \tag{1}$$

Strictly speaking, the Hilbert space being discussed has the above product where the vectors are all elements of $\mathbb{C}^{3n^3}$ that correspond to a physical current/field. In any case, discussing this directly is not particularly important for the rest of this discussion. What is important however, is that physical fields respect the symmetry, linearity, and positive definiteness requirements of a Hilbert space. This can be quickly verified using equation 1.

A. Problem Formulation

With this in mind, consider the following quadratic program:

$$\max_{|T\rangle\in\mathbb{C}^{3n^3}} \mathrm{Im}\,\langle S|T\rangle - \langle T|O|T\rangle \tag{2}$$

In general, $|S\rangle$ can be any element in $\mathbb{C}^{3n^3}$ and O is some general objective linear operator on the space. In the case of Purcell enhancement for example, $O = \mathrm{Asym}\big[V^{-\dagger}\big]$ is an appropriate choice. Without loss of generality, it is convenient in the photonics context to have it represent the source field of the problem (which must be scattered to acheive the desired objective) explicitly. Given a source field $|S\rangle$ and a scattering potential V, the total field $|T\rangle$ produced is given by

$$|T\rangle = \left(V^{-1} - G_0\right)^{-1}|S\rangle\,, \tag{3}$$

where G_0 is the Green's function of Maxwell's equations for free space. To this objective function constraints are added to enforce power conservation using a hierarchical mean field approach[2]. This is done by considering connected clusters ($\mathbf{\Omega} = \{\Omega_k\}_{k\in K}$) within the design domain and imposing constraints in one of the following forms $\forall \Omega_k \in \mathbf{\Omega}$:

$$\begin{aligned}
\langle S|\mathbb{I}_{\Omega_k}|T\rangle &= \langle T|U\mathbb{I}_{\Omega_k}|T\rangle \\
\langle S|\mathbb{I}_{\Omega_k}|R\rangle &= \langle R|\mathbb{I}_{\Omega_k}|R\rangle \\
\langle S|\mathbb{I}_{\Omega_k}|T\rangle &= \langle R|\mathbb{I}_{\Omega_k}|T\rangle \\
\langle S|\mathbb{I}_{\Omega_k}|R\rangle &= \langle T|U\mathbb{I}_{\Omega_k}|R\rangle
\end{aligned} \tag{4}$$

Here $\mathbb{I}$ is an indicator function over its subscript and $U = V^{-\dagger} - G_o^{\dagger}$. In order to obtain bounds on this program P, the Lagrange dual program D(P) is considered, the corresponding Lagrangian is

$$\mathcal{L} = \begin{bmatrix}\langle T| & \langle S|\end{bmatrix}\begin{bmatrix}-Z^{TT} & Z^{TS} \\ Z^{ST} & 0\end{bmatrix}\begin{bmatrix}|T\rangle \\ |S\rangle\end{bmatrix} \tag{5}$$

where

$$Z^{TT} = O + \mathrm{Sym}[U\Phi_1] + \mathrm{Asym}[U\Phi_2] \tag{6}$$
$$Z^{TS} = Z^{ST*} = \frac{1}{2}(\Phi_1 + i\Phi_2)I.$$

In this notation $\Phi_1 \in \mathbb{R}^{n_1}_{\geq 0}$ and $\Phi_2 \in \mathbb{R}^{n_2}_{\geq 0}$ contain the Lagrange multipliers associated with the n_1 symmetric and n_2 antisymmetric cluster constraints. The dual function is then $\mathcal{G} = \max_{|T\rangle} \mathcal{L}(|T\rangle, \Phi)$ for the dual program:

$$\mathrm{D}(P) = \inf_{\Phi} \mathcal{G}(\Phi) \tag{7}$$

This problem is typically solved using local methods (newton-like gradient descents for example).

B. Issues with Standard Dual Solutions

A notable feature of this problem is that Z^{TT} is indefinite in general, so it may not have a global extremum and diverge in certain regions. In order to obtain a bound on the primal, these regions must be avoided so it must be verfied whether $Z^{TT} \succcurlyeq 0$ for each value of Φ tested. To do this explicitly requires finding the eigenvalues of the matrix, typically done with a Cholesky-like decoposition. The discretizations typically require n in the hundreds, make doing this check at each point in the optimization far to costly.

A common attempt to remedy this is to use the $\log\det Z^{TT}$ in order to ascertain when the boundary of the positive-definiteness is nearby. Assuming that Z^{TT} is positive definite to begin with, when a value of Φ is chosen such that Z^{TT} is close to being indefinite there will be eigenvalues that start to approach zero. Since the determinant is the product of the eigenvalues, it is naively expected that it also become very small. Therefore, the log det is expected to become very large around the the boundary. The reason that this method is advantageous over eigenvalue solvers is that a Hutchinson trace estimator can be used ($\log\det Z^{TT} = \mathrm{Tr}\log Z^{TT}$). Briefly, for any square matrix $A \in \mathbb{C}^{m\times m}$ the Hutchinson trace estimator does probabilistic sampling over vector-matrix-vector products to estimate the trace:

$$\mathrm{Tr}_A \approx \frac{1}{N}\sum_{i=1}^{N} x^{\dagger}Ax,\ x \sim \{-1,1\}^m \tag{8}$$

This method, however is still too slow. In order to obtain relative error ϵ to probability $1-\delta$ the number of sample vectors needed is [4]

$$N = \frac{2}{\epsilon^2}\left(2 + \frac{8\sqrt{2}}{3}\epsilon\right)\log\frac{2}{\delta}. \tag{9}$$

For reference a 99.9% chance to have 0.1% error needs 30 460 939 matrix vector products, and a 75% chance to have 1% error needs 84 746 matrix vector products.

III. PADÉ ALGORITHM

In order to avoid this expensive step another method must be employed to guarantee the points tested are within the positive definite domain. Therefore a modification of the dual function is considered (the *Partial Dual* $\mathrm{D}_\partial(\mathrm{P})$), singling out one of the constraints,

$$\mathcal{G}_\partial = \max_{|T\rangle} \mathcal{L}_\partial(|T\rangle, \Phi) \tag{10}$$
$$\text{such that } \mathrm{Im}\langle S|T\rangle - \langle T|E|T\rangle \geq 0$$

such that

$$\mathcal{L}_\partial = \begin{bmatrix}\langle T| & \langle S|\end{bmatrix}\begin{bmatrix}-Z_\partial^{TT} & Z_\partial^{TS} \\ Z_\partial^{ST} & 0\end{bmatrix}\begin{bmatrix}|T\rangle \\ |S\rangle\end{bmatrix}. \tag{11}$$

Here the ∂ subscript for the partial dual problem denotes that the quantity is the same as in the ordinary case, except for the fact that terms containing the dual constraint have been removed. If $Z_\partial^T \succcurlyeq 0$ then the problem is convex and no checking is required. In this case, $\mathrm{D}_\partial(\mathrm{P}) = \langle S|Z_\partial^{TS}Z_\partial^{TT-1}Z_\partial^{TS}|S\rangle$ with the corresponding current $|\mathbf{T}\rangle = Z_\partial^{TT-1}Z_\partial^{TS}|S\rangle$.

Therefore, the idea of this method is that a feasible ζ (the dual constraint's multiplier) must be chosen such that Z_∂^{TT} is positive definite. Note that $Z_\partial^{TT} = Z^{TT} + \zeta E$ and $Z_\partial^{TS} = Z^{TS} + \frac{1}{2}i\zeta I$ so the constraint function is implicitly dependant on ζ, so it will hereafter be abbreviated to C_ζ. With this notation, the goal is to have ζ such that

$$Z^{TT} + \zeta E \succcurlyeq 0 \tag{12}$$
$$C_\zeta \geq 0$$

Since $E \succcurlyeq 0$, $\exists\zeta \geq 0$ such that $Z_\partial^{TT} \succcurlyeq 0$. Next note that the derivative of the constraint areas follows:

$$\frac{dC_\zeta}{d\zeta} = 2\left(\frac{1}{2}\langle S| - i\langle T|E\right)Z_\partial^{TT-1}\left(\frac{1}{2}|S\rangle + iE|T\rangle\right)$$
$$\frac{d^2C_\zeta}{d\zeta} = -6\left(\frac{1}{2}\langle S| - i\langle T|E\right)Z_\partial^{TT-1}EZ_\partial^{TT-1}\left(\frac{1}{2}|S\rangle + iE|T\rangle\right)$$

Therefore if $Z_\partial^{TT} \succcurlyeq 0$, then C_ζ must be increasing and concave. Furthermore, in the limit of large ζ:

$$\begin{aligned} \lim_{\zeta\to\infty} Z_\partial^{TT-1} &= (\zeta E)^{-1} \\ \lim_{\zeta\to\infty} Z_\partial^{TS} &= \frac{1}{2} i\zeta I \\ \implies \lim_{\zeta\to\infty} C_\zeta &= \langle S|E^{-1}|S\rangle \end{aligned} \tag{13}$$

Since $E \succcurlyeq 0$, C_ζ asymptotes to a positive number. The combination of these properties of the constraint mean that if a zeta is chosen past the last crossing, it is feasible and makes the problem convex.

The next logical step is to find an efficient way to find the last zero of this constraint function. The issue is that evaluating C_ζ is very expensive to evaluate since it involves doing an inverse solve for a very large system, so it must be done sparingly. For this reason, the strategy here is to create a series approximation of C_ζ. The last root of the series ought to be computationally cheap to find and should be able to capture the beviour of poles, of which the constraint function has many. Therefore, a Padé approximant is well suited for the task.

A. The AAA Algorithm

The Padé approximant is constructed according to the AAA algorithm [3], this is done as follows: Consider a function $f : \mathbb{C} \to \mathbb{C}$, the goal is to find another function $r : \mathbb{C} \to \mathbb{C}$ to approximate it. Given a finite ordered set $Z \subset \mathbb{C}$ and its corresponding f values $F \subset \mathbb{C}$. The AAA algorithm will split Z and F, each into 2 partitions. The first partitions (z^m and f^m) will be used to make the m support point in the Padé series:

$$r(z) = \frac{n(z)}{d(z)} = \sum_{j=1}^{m} \frac{w_j f_j}{z - z_j} \Bigg/ \sum_{j=1}^{m} \frac{w_j}{z - z_j} \tag{14}$$

The second partitions ($Z^m = Z \setminus z^m$ and $F^m = F \setminus f^m$) will be used as sample points to do a least squares fit the series, thus determining the weights. This is accomplished in the following way:

1. $f^1 = \left\{\text{argmin}_{f\in F}(f - \langle F\rangle)^2\right\}$ and z^1 is the corresponding singleton. This defines F^1 and Z^1.

2. Obtain w^1 by performing a least squares fit to F^1 and Z^1 using f^1 and z^1 as the support point.

3. Given F^m, Z^m, f^m, z^m, and w^m calculate the square residuals for all $z \in Z^m$. Select the $(z, f) \in Z^m \times F^m$ with the least residual and add it to the support points.

4. Using the newly obtained f^{m+1} and z^{m+1}, obtain w^{m+1} by least squares fitting to F^{m+1} and Z^{m+1}.

5. Repeat steps 3 and 4 until the desired error tolerance is reached.

B. Finding the Last C_ζ Root

Now for how this is applied to the partial dual:

1. Sample on a uniform distribution centered on an initial guess to form Z and use C_ζ to form F.

2. Form the AAA approximant of C_ζ using Z and F and find the zero of the approximant ζ_{guess}

3. If $C_{\zeta_{\text{guess}}}$ is not within tolerance, add $(\zeta_{\text{guess}}, C_{\zeta_{\text{guess}}})$ to (Z, F) and return to step 2.

While finding the initial guess might be costly, this must only be done once in the entire inverse design. In subsequent steps in a search in the Lagrange multiplier space, the Z matrices are perturbed according to their dependance on Φ. Since their dependance is continuous, it is expected that small changes in Φ will result in small changes in the last zero crossing.

IV. RESULTS

In order to test the efiiciency of the root finding algorithm, statistical test is done to find the typical behaviour. This is accomplished by randomly sampling the variables that appear in equation 6. The distributions are uniform over the domains seen in Table I.

Variable	Sampling Domain
O	$[0, 1]$
S	$[-0.5, 0.5] \times [-0.5i, 0.5i]$
V	$[] \times [0, 10^{-3}i]$
Φ	[0, 1]

TABLE I. Each element of the discretized quantities is sampled from the distribution. The interval notation is taken to represent the line segment in the complex plane bounded by the two points.

The root finding method on average takes 4.65 ± 2.89 inverse solves to terminate.

[1] Sean Molesky, Pengning Chao, Weiliang Jin, and Alejandro W Rodriguez. Global t operator bounds on electromagnetic scattering: Upper bounds on far-field cross sec-

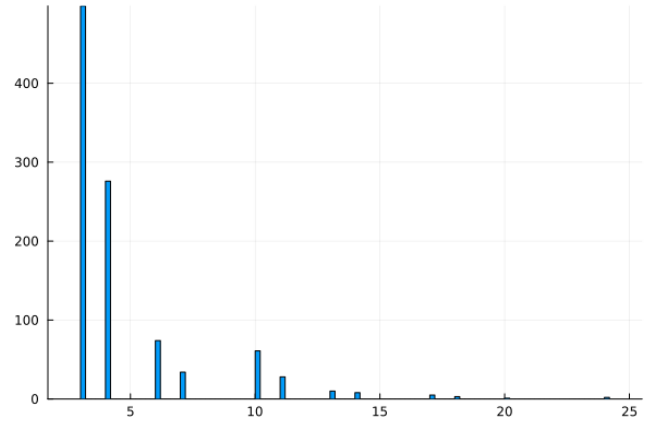

FIG. 1. Histogram of the number of inverse solves it takes for the algorithm to terminate. Note the behaviour where there is a peridodic appearance of peaks, this is due to the fact that the for testing there is a maximum iteration imposed, after which resampling occurs.

tions. *Physical Review Research*, 2(3):033172, 2020.

[2] Sean Molesky, Pengning Chao, and Alejandro W Rodriguez. Hierarchical mean-field t operator bounds on electromagnetic scattering: Upper bounds on near-field radiative purcell enhancement. *Physical Review Research*, 2(4):043398, 2020.

[3] Yuji Nakatsukasa, Olivier Sète, and Lloyd N Trefethen. The aaa algorithm for rational approximation. *SIAM Journal on Scientific Computing*, 40(3):A1494–A1522, 2018.

[4] Maciej Skorski. Modern analysis of hutchinson's trace estimator. In *2021 55th Annual Conference on Information Sciences and Systems (CISS)*, pages 1–5. IEEE, 2021.