22

Quantum interior point methods

The authors are grateful to Sander Gribling for reviewing this chapter.

Rough overview (in words)

Interior point methods (IPMs) are a type of efficient classical algorithm for solving convex optimization problems such as linear programs (LPs), second-order cone programs (SOCPs), and semidefinite programs (SDPs). IPMs are the basis for effective optimization software tools (e.g., [355, 38]), which are widely used for solving convex optimization problems that arise in industry. They are called *interior* point methods because, in contrast to the simplex method, they iteratively generate a sequence of points that lie in the interior of the convex region; this sequence of points is guaranteed to rapidly approach the optimal point (which, when it exists and the objective function is convex, is guaranteed to lie at the boundary of the convex region). At each iteration, the next point is produced by solving a system of linear equations. See, for example, [1053, 1052, 797, 438] for context on how IPMs fit into the history of methods for optimization.

Quantum interior point methods (QIPMs) are quantum algorithms that leverage a similar approach as classical IPMs, but perform certain aspects of the algorithm in a quantum manner. For example, QIPMs were first introduced in [610], where the quantum algorithm is identical to classical IPMs, except that it determines the next point using a quantum linear system solver (QLSS) combined with quantum state tomography, rather than a classical linear system solver. Subsequent work has explored other forms of quantizing classical IPMs that do not rely on the QLSS [51, 69].

Classical IPMs are generally efficient in the sense that they can solve convex optimization problems in time scaling as a polynomial in the number of vari-

ables. The exact degree of the polynomial depends on which kind of convex optimization problem is being solved, as well as certain choices about the IPM. Since QIPMs often rely on state tomography, they are generally expected to require time that scales at least linearly in the number of variables, and lower bounds along these lines are known [48]; thus, *the best one can hope for is a polynomial speedup* over classical IPMs. The exact runtime of the quantum algorithm depends on instance-specific parameters, such as the condition number of matrices that appear during the course of the algorithm, which makes it difficult to determine whether a speedup exists in practice.

Rough overview (in math)

For simplicity, we focus on LPs, the simplest kind of optimization problem where QIPMs can be applied. An LP is specified by an $m \times n$ matrix A, an n-dimensional vector c, and an m-dimensional vector b, and it is given by

$$\begin{aligned} &\min_{x \in \mathbb{R}^n} \langle c, x \rangle \\ &\text{subject to } Ax = b \\ &\qquad x_i \geq 0 \text{ for } i = 1, \ldots, n \end{aligned} \quad , \tag{22.1}$$

where $\langle u, v \rangle$ denotes the standard dot product between vectors u and v.

The function $\langle c, x \rangle$ is called the objective function, and a point x is called feasible if it satisfies $Ax = b$ and $x_i \geq 0$ for all i. Inequality constraints of the form $Ax \leq b$ can be handled by introducing slack variables. We denote the feasible point that optimizes the objective function by x^*.

An important concept in mathematical optimization is duality, where given one optimization problem, an equivalent "dual" optimization problem can be generated through the method of Lagrange multipliers (see [180, Section 5]). The dual of the LP in Eq. (22.1) is given by

$$\begin{aligned} &\max_{y \in \mathbb{R}^m} \langle b, y \rangle \\ &\text{subject to } A^\intercal y + s = c \\ &\qquad s_i \geq 0 \text{ for } i = 1, \ldots, n \end{aligned} \quad . \tag{22.2}$$

Alternatively, one can drop the s variable and constraints that s_i are positive, and simply write $A^\intercal y \leq c$. Denote the optimal feasible points for the dual by (y^*, s^*).

It can be shown that the optimal point lies at the boundary of the feasible region and satisfies the relationship $x_i s_i = 0$ for all i. A key concept in IPMs is the *central path*, a set of points parameterized by $\mu > 0$. The central point with parameter μ is the feasible point for which $x_i s_i = \mu$ for all i. In general,

this point will be in the interior of the feasible region, but as $\mu \to 0$, the central path approaches the optimal point on the boundary.

The most effective classical IPMs are "primal-dual path-following methods," which generate a length-T sequence of primal-dual point pairs $(x^{(t)}, y^{(t)}, s^{(t)}) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$ for $t = 0, \ldots, T-1$ that approximately follows the central path toward the optimum. Given $(x^{(t)}, y^{(t)}, s^{(t)})$, the point $(x^{(t+1)}, y^{(t+1)}, s^{(t+1)}) = (x^{(t)} + \Delta x, y^{(t)} + \Delta y, s^{(t)} + \Delta s)$ is formed by solving the following linear system of equations, which is called the *Newton system*, as it corresponds to one iteration of Newton's method.

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^\intercal & I \\ S & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta s \end{pmatrix} = \begin{pmatrix} b - Ax^{(t)} \\ c - A^\intercal y^{(t)} - s^{(t)} \\ \sigma \frac{x^{(t)\intercal} s^{(t)}}{n} \mathbf{1} - X s^{(t)} \end{pmatrix}, \tag{22.3}$$

where $\sigma < 1$, $\mathbf{1}$ denotes the all 1s vector, and $S = \mathrm{diag}(s^{(t)}), X = \mathrm{diag}(x^{(t)})$ are diagonal $n \times n$ matrices formed from the entries of $s^{(t)}$ and $x^{(t)}$. Note that there are alternative ways to formulate the Newton system (see, e.g., [70, 68]). To understand Eq. (22.3), note that if the point $(x^{(t)}, y^{(t)}, s^{(t)})$ is feasible, then the first two entries on the right-hand side are zero. Furthermore, if it is on the central path, then $X s^{(t)} = \frac{x^{(t)\intercal} s^{(t)}}{n} \mathbf{1}$, so if we were to choose $\sigma = 1$, then the entire right-hand side would be zero, and the solution to the system would be $\Delta x = \Delta y = \Delta s = 0$. If instead we set $\sigma = 1 - \delta$ for sufficiently small δ, the solution will correspond to taking a small step along the central path in the direction of decreasing μ. Technically, we do not exactly follow the central path, but it can be guaranteed that the sequence of points stays within a small neighborhood of it. As $\mu \to 0$, the central path approaches the optimal point (x^*, y^*, s^*), so by following the path toward $\mu = 0$, a classical or quantum IPM can guarantee success.

The classical IPM can solve the Newton system exactly using Gaussian elimination in $O(n^3)$ operations, or it can solve the system approximately using a variety of iterative solvers such as the conjugate gradient method. In contrast, the standard approach for a QIPM is to solve the Newton system by using a QLSS to repeatedly prepare the $O(\log(n))$-qubit state $|\Delta x, \Delta y, \Delta s\rangle$ whose amplitudes encode the solution to the Newton system. By preparing many copies, the algorithm can perform (pure state) quantum state tomography to yield an estimate $(\overline{\Delta x}, \overline{\Delta y}, \overline{\Delta s})$ for the amplitudes $(\Delta x, \Delta y, \Delta s)$ to some desired precision ξ (in 2-norm), that is,

$$\|(\overline{\Delta x}, \overline{\Delta y}, \overline{\Delta s}) - (\Delta x, \Delta y, \Delta s)\| \leq \xi \|(\Delta x, \Delta y, \Delta s)\| .$$

Due to the tomography step, the QIPM is only able to generate solutions to the Newton system that are *inexact*. There has been some question in the liter-

ature whether the (classical or quantum) IPMs with the fastest guaranteed convergence rate (i.e., the number of iterations needed to reduce μ to ϵ) are applicable even when inexact solutions are used, as this causes intermediate points to be (slightly) infeasible [70]. However, if ξ is sufficiently small, the method appears to work empirically even using the inexact solutions that would be output by a quantum solver [328]. Alternatively, there exist workarounds [70] that ensure feasibility is maintained even when linear systems are solved inexactly, at the expense of some additional classical cost.

The IPMs and QIPMs for SOCPs [612, 68] are quite similar to those for LPs described above: the main difference is that the matrices X and S are no longer strictly diagonal matrices. QIPMs have also been proposed for SDPs [610, 70, 537], which are more complex but have more expressive power; here, additional considerations must be taken to guarantee that the intermediate solutions continue to be symmetric even after experiencing errors due to tomography.

The above exposition represents the original approach to quantizing the classical IPM, which has so far garnered the most study. An alternative to this approach was proposed in [51], which focuses on the case that the LP constraint matrix A is "tall," that is, $m \gg n$. As above, they follow the central path to the optimal point; however, they adopt a primal-only approach, where the Newton linear system takes on the form $(B^\intercal B)g = h$, with g and h length-n vectors and B an $m \times n$ matrix. Rather than using the QLSS and quantum state tomography, their quantum algorithm performs a Grover search–like step to identify the "important" rows of B and thus produce an $O(n) \times n$ matrix $\tilde{B}$ for which $\tilde{B}^\intercal \tilde{B} \approx B^\intercal B$. This enables a quadratic speedup in the large parameter m. To obtain the right-hand side vector h, which is the gradient of the objective function, they require the multivariate mean-estimation algorithm of [310], which is related to the quantum gradient estimation primitive developed in [587, 430]—this is key for avoiding a dependence on the condition number of $B^\intercal B$. Matrix inversion is then performed classically at cost polynomial in n, independent of m and not depending on the condition number of any matrix.

Meanwhile, another quantum algorithm inspired by IPMs was proposed in [69]. Where the standard QIPM encodes the variable x into the amplitudes of the quantum state, requiring readout with quantum state tomography, the method of [70] encodes the components of x into separate binary registers, truncated to some finite number of bits of precision. It constructs a Hamiltonian, parameterized by μ, whose ground state is a wavefunction localized near the associated point on the central path. By slowly decreasing μ and invoking the adiabatic theorem, the wavepacket follows the central path to $\mu = 0$, where the optimal point can be recovered by a measurement. Thus, the main primitive required is time-dependent Hamiltonian simulation.

Dominant resource cost (gates/qubits)

The outer loop of QLSS-based QIPMs is purely classical; at each iteration a small step is taken to form the next point in the sequence. For LPs, SOCPs, and SDPs, the number of iterations T required to yield a point for which the objective function is within ϵ of optimal is $O(\sqrt{n}\log(1/\epsilon))$. The main cost of each iteration is solving the Newton system. In the complexity statements that follow, we assume the number of constraints m is on the order of the number of degrees of freedom (i.e., $m = O(n)$ in the case of LPs and SOCPs, and $m = O(n^2)$ in the case of SDPs).

The QIPM solves the Newton system by preparing many copies of the state corresponding to the solution to the linear system. This state can be prepared in time $\text{polylog}(n)\cdot\zeta\kappa$, where κ is the condition number of the matrix in Eq. (22.3) and ζ is the ratio $\|\cdot\|_F/\|\cdot\|$ of the Frobenius and spectral norms of the matrix, assuming that one can perform a block-encoding of the Newton matrix in $\text{polylog}(n)$ time, a task that requires access to large-scale quantum random access memory (QRAM).[1] For LPs and SOCPs, the number of copies that must be prepared scales as $O(n/\xi^2)$ when using the basic version (see [610, Section 4] and [328, Section IVD]) of pure state tomography that simply measures each copy in the computational basis. A more recent and complex version of tomography [49] can achieve this task using $O(n/\xi)$ copies along with additional gates. For SDPs, since the variables are matrices rather than vectors, the number of copies is $O(n^2/\xi^2)$ or $O(n^2/\xi)$. Overall, using the more efficient version of tomography and ignoring the additional gates, the runtime of the QIPM is expected to scale as

$$\begin{aligned} \text{LP, SOCP:} \quad & \widetilde{O}\left(\frac{n^{1.5}\zeta\kappa}{\xi}\log(1/\epsilon)\right) \\ \text{SDP:} \quad & \widetilde{O}\left(\frac{n^{2.5}\zeta\kappa}{\xi}\log(1/\epsilon)\right), \end{aligned} \tag{22.4}$$

where κ denotes the maximum condition number, ζ the maximum ratio of Frobenius to spectral norm, and ξ the minimum tomographic precision required across all iterations. There may be an additional purely classical cost of $O(n^{2.5})$ for LPs/SOCPs and $O(n^{4.5})$ for SDPs, deriving from classical matrix-vector multiplications necessary for setting up the Newton system at each iteration.

[1] It is worth emphasizing that the origin of the dependence on the Frobenius norm of the Newton matrix here is the normalization factor that arises when block-encoding a dense classical matrix. If the matrix were sparse or had some compact representation, this normalization factor could potentially be improved—but for Newton matrices in QIPMs we do not expect this to be the case.

In the worst case, it may be necessary to take ξ as small as $O(1/\kappa)$, and ζ can be as large as $\sqrt{n}$ (SOCP/LP) or n (SDP)—complexity statements in the literature, such as [70], often assume these worst-case values for those parameters, but we refrain from doing so as these worst-case values may be overly pessimistic in practice. The hidden constant prefactors are dependent primarily on the implementation of the QLSS and tomography. It is clear that the viability of the QIPM is highly dependent on the value and scaling of the parameters κ and ξ. Unfortunately, it is believed that for some LP/SOCP/SDP instances, the value of κ will diverge as the target precision ϵ is made smaller, perhaps as $O(1/\epsilon)$ [612, 70], although this may not be the case in every instance (see, e.g., the numerical results of [328]).

The QIPM only requires a register of $O(\log(n))$ qubits to hold the solution of the linear system; however, achieving the runtimes quoted requires queries to QRAM. In this case, the explicit QRAM circuits that achieve shallow depths of $O(\log(n))$ necessarily require $O(n^2)$ total gates across $O(n^2)$ total qubits.

The alternative approach of [51] is best suited for the case where $m \gg n$, and requires $\sqrt{m} \cdot \text{poly}(n, \log(1/\epsilon))$ queries to the entries of the matrix A, where the $\sqrt{m}$-dependence fundamentally comes from Grover-like primitives with quadratic speedup. Like the standard QIPM formulation, this approach requires a QRAM to implement the queries in polylog(m) time. However, since it does not use QLSS or tomography, it avoids polynomial dependence on the instance-specific parameters κ and $1/\xi$.

Caveats

There are several important caveats that must be considered when evaluating a speedup claimed by QIPM.

- Even in a best case scenario, the quantum speedup is at most polynomial (and even subquadratic). Since quantum computation requires significant constant-factor overheads due to slower clock speeds and error correction, the value of n for which a QIPM would be faster than a classical IPM on actual hardware is likely to be large (see [328] for further discussion).
- Since n must be large for a quantum speedup to be obtained, a very large QRAM, corresponding to millions or billions of (logical) qubits, would be needed for any speedup to be realized.
- QIPMs are most effective when the matrices that need to be inverted over the course of the algorithm are well conditioned, due to their reliance on the QLSS. However, when the condition number κ is small, iterative classical methods may also be effective, limiting the advantage of the quantum algorithm. In particular, a linear system with $O(n)$ dense constraints on n

variables can be solved to error ξ in time $O(n\zeta^2\kappa^2 \log(1/\xi))$ using the randomized Kaczmarz method [959]. In comparison, the QIPM utilizes QLSS and tomography to solve the same task (once per iteration) in time $O(n\zeta\kappa/\xi)$. Even if $\xi = \Omega(1)$, this limits the magnitude of the quantum speedup to a factor of $O(\zeta\kappa)$. Thus, for the quantum speedup to be maximized, κ can be neither too small nor too large. While we are not aware of any IPM implementations based on the Kaczmarz method, its complexity allows for clean comparison with quantum algorithms involving the QLSS for dense matrices, since both depend directly on the quantity $\zeta\kappa$. Here it is also worth mentioning that there exist other approximate classical linear system solvers for which the complexity depends on κ, but not on ζ. One example is the conjugate gradient method [713]. Another straightforward example is to solve the system $Gu = v$ by finding a degree-$O(\kappa \log(1/\xi))$ polynomial approximation $p(x) \approx 1/x$, and then computing $p(G)v \approx G^{-1}v = u$ via a sequence of $O(\kappa \log(1/\xi))$ matrix-vector products—this is a classical analog of the quantum approach based on the quantum singular value transformation [431]. Classically, each matrix-vector product costs $O(n^2)$ when G is dense and $O(ns)$ for when G is s-sparse.

- If the matrices that define the convex problem have a certain structure (e.g., sparsity), this could be exploited to potentially reduce the overhead from block-encoding—in particular, the value of ζ and the size of the QRAM required. However, this can help the quantum algorithm only to a limited extent, as the vectors $(\Delta x, \Delta y, \Delta s)$ will still be dense and reading out estimates for all $O(n)$ amplitudes with quantum tomography will be necessary.

Example use cases

- Portfolio optimization, the canonical optimization problem that appears in finance, can be formulated as an SOCP and solved with a QIPM; a study of the condition number of the matrices that appear in this application was consistent with a small quantum speedup [611]; however, a follow-up study did not replicate this finding [328] and also pointed out that in any case large constant-factor overheads would make achieving practical advantage challenging.
- Support vector machines, a common task in machine learning, can be reduced to SOCPs and solved with a QIPM; a study of the condition number of the matrices that appear in this application was consistent with a small quantum speedup [612].
- Sample-efficient protocols for mixed-state tomography reduce the problem of reconstructing an estimate of the quantum state to solving an SDP. This

SDP could be solved with a QIPM (note that the tomography needed within the QIPM is always on *pure states* and does not require solving an SDP, thus avoiding an issue of circular logic).

- Nonconvex optimization is often solved approximately by relaxing the problem into a convex problem like an SDP. For example, the MAX-CUT problem is a combinatorial optimization problem over the nonconvex space $\{+1, -1\}^n$, but by solving the associated SDP relaxation and rounding, an approximate solution can be obtained.

Further reading

- See Boyd and Vandenberghe [180] for an accessible book on convex optimization including (classical) IPMs.
- QIPMs are an active area of research. A QIPM for LPs and SDPs was originally proposed by Kerenidis and Prakash in [610]. This was followed up by a QIPM for SOCPs in [612], along with numerical simulations for specific applications [612, 611]. Later, [70] pointed out a potential error in the convergence analysis of previous works, and they presented two possible workarounds called the "inexact-infeasible" and "inexact-feasible" IPMs. Note also the work in [537] for another way to avoid this issue, giving a QIPM for SDP.
- See [51, 69] for quantum methods related to IPMs that do not rely on the QLSS.