

Rational Jacobi Rotations and the Complexity of Approximating Mixed Integer Quadratic Programming

Alberto Del Pia *

July 31, 2026

Abstract

We present an algorithm that finds an ϵ -approximate solution to a mixed integer quadratic programming (MIQP) problem, and that runs on a Turing machine in time polynomial in the size of the instance and in $1/\epsilon$, provided that the number of integer variables and the number of negative eigenvalues of the Hessian of the objective function are fixed. Unless $P = NP$, both restrictions are necessary, so this completes the characterization of the complexity of approximating MIQP in terms of the number of integer variables and the inertia of the Hessian; the result is new already in the purely continuous case. The main ingredient is a polynomial-time simultaneous diagonalization algorithm: it computes a rational change of basis that maps a given ellipsoid, presented in factored form, exactly to a ball, while making the objective function separable up to an arbitrarily small perturbation and preserving the inertia of its Hessian. The classical construction, in which the objective function is made exactly separable, requires a change of basis that is in general irrational, and cannot be carried out on a Turing machine; ours rests instead on rational Jacobi rotations, which we believe to be of independent interest.

Key words: Mixed integer quadratic programming; Approximation algorithm; Simultaneous diagonalization; Jacobi rotations; Computational complexity

1 Introduction

A mixed integer quadratic programming problem is an optimization problem of the form

$$\begin{aligned} \min \quad & x^\top Hx + h^\top x \\ \text{s. t.} \quad & Wx \leq w \\ & x \in \mathbb{Z}^p \times \mathbb{R}^{n-p}. \end{aligned} \tag{MIQP}$$

In this formulation, $x \in \mathbb{R}^n$ is the vector of variables and the input data consists of a symmetric $H \in \mathbb{Q}^{n \times n}$, which we refer to as the Hessian of the objective function, $h \in \mathbb{Q}^n$, $W \in \mathbb{Q}^{m \times n}$, $w \in \mathbb{Q}^m$, and $p \in \{0, 1, \dots, n\}$, the number of integer variables.

Already deciding whether Problem MIQP is feasible is NP-hard, since it contains integer linear feasibility as a special case. Lenstra [12] showed that this can be done in polynomial time provided that p is fixed, and we place ourselves in this setting throughout.

The difficulty of Problem MIQP is governed, besides p , by the spectrum of the matrix H . The *inertia* of a symmetric matrix is an ordered triple (k_-, k_0, k_+) , where k_- is the number of negative eigenvalues, k_0 the number of zero eigenvalues, and k_+ the number of positive eigenvalues. If H is positive semidefinite, that is, if $k_- = 0$, then Problem MIQP can be solved in polynomial time on a Turing machine, for fixed p [8]. As soon as H has a single negative eigenvalue, however, the problem becomes NP-hard, already for

*Department of Industrial and Systems Engineering & Wisconsin Institute for Discovery, University of Wisconsin-Madison, Madison, WI, USA. E-mail: delpia@wisc.edu.

$p = 0$ [15]. In the non-convex case one can therefore only aim at approximation algorithms, and this is the object of this paper.

To state our main result, we first define the concepts of ϵ -approximate solution and of size. Consider an instance of a general optimization problem $\inf \{f(x) : x \in \mathcal{S}\}$, with $\mathcal{S} \neq \emptyset$. For $\epsilon \in [0, 1]$, we say that $x^\diamond \in \mathcal{S}$ is an ϵ -approximate solution if

$$f(x^\diamond) - f_{\inf} \leq \epsilon \cdot (f_{\sup} - f_{\inf}),$$

where $f_{\inf} := \inf_{x \in \mathcal{S}} f(x)$ and $f_{\sup} := \sup_{x \in \mathcal{S}} f(x)$. Note that 0-approximate solutions coincide with optimal solutions and 1-approximate solutions coincide with feasible solutions. For any $\epsilon > 0$, if $f_{\inf} = -\infty$, or $f_{\sup} = +\infty$, then our definition loses its value because any feasible point is an ϵ -approximate solution. This definition of approximation has some well-known invariance properties which make it a natural choice for the problems considered in this paper. For instance, it is unchanged when the objective function is multiplied by a positive scalar or shifted by a constant, the feasible region being left untouched, and it is also unchanged when an affine transformation is applied simultaneously to the objective function and to the feasible region, like for example a change of basis. This definition has been used in our earlier work [5, 6, 7], and in numerous other places, including [14, 18, 17, 1, 4].

We define the *sizes* of a rational number p/q (where p and q are relatively prime integers), of a rational vector $v \in \mathbb{Q}^n$ and of a rational matrix $A \in \mathbb{Q}^{m \times n}$ as in [16]:

$$\begin{aligned} \text{size}(p/q) &:= 1 + \lceil \log(|p| + 1) \rceil + \lceil \log(|q| + 1) \rceil, \\ \text{size}(v) &:= n + \sum_{i=1}^n \text{size}(v_i), \\ \text{size}(A) &:= mn + \sum_{i=1}^m \sum_{j=1}^n \text{size}(a_{ij}). \end{aligned}$$

Throughout the paper, $\log$ denotes the base-2 logarithm, and $\ln$ denotes the natural logarithm. Given an instance of Problem MIQP (i.e., specific data H, h, W, w), we define its *size* as $\text{size}(H) + \text{size}(h) + \text{size}(W) + \text{size}(w)$. We are now ready to state the main result of this paper.

Theorem 1 (Approximation algorithm for Problem MIQP). *Assume that the objective function of Problem MIQP is bounded below on the feasible region, and let $\epsilon \in (0, 1]$ be rational. There is an algorithm that either detects that Problem MIQP is infeasible, or finds an ϵ -approximate solution to Problem MIQP. The algorithm runs on a Turing machine in time polynomial in the size of the instance, in $\text{size}(\epsilon)$, and in $1/\epsilon$, provided that p and the number k_- of negative eigenvalues of H are fixed.*

The two parameters that Theorem 1 keeps fixed are precisely the two that must be fixed, unless $P = NP$. If the assumption that p is fixed is dropped, the algorithm would decide in polynomial time whether the feasible region of Problem MIQP is empty, which is NP-hard. If the assumption that k_- is fixed is dropped, the algorithm could be used to solve 3SAT in polynomial time, already for $p = 0$ [18]. In this sense Theorem 1 completes the characterization of the complexity of approximating Problem MIQP in terms of p and of the inertia of H . The dependence of the running time on $1/\epsilon$ cannot be dropped either: an algorithm running in time polynomial in the size of the instance and in $\text{size}(\epsilon)$ alone could be used to solve quadratic programming with one negative eigenvalue in polynomial time [18], and the latter problem is NP-hard [15].

The assumption that the objective function is bounded below on the feasible region is imposed rather than checked, and necessarily so: deciding whether it holds is NP-hard, even if $p = 0$ and H has only two negative eigenvalues. This follows from Theorem 2 in [7], which is stated in terms of the rank of H ; the instances constructed in its proof have a Hessian with exactly two negative eigenvalues and one positive eigenvalue.

Related work. Theorem 1 can be seen as the conclusion of a line of approximation algorithms with theoretical guarantees for quadratic programming (Problem MIQP with $p = 0$) and for mixed integer

quadratic programming. For quadratic programming, Vavasis gave an approximation algorithm for the concave case ($k_+ = 0$) with k_- fixed [18], and then for the more general case with only k_- fixed [17]. In the mixed integer setting (with p fixed), the concave case ($k_+ = 0$) with k_- fixed was settled in [5, 6], and the fixed rank case, with both k_- and k_+ fixed, in [7]. Theorem 1 subsumes all of these: it requires only that k_- is fixed, and in particular it places no restriction whatsoever on the number k_+ of positive eigenvalues, hence none on the rank of H . All the algorithms mentioned above run in polynomial time on a Turing machine, with one exception: the algorithm of [17] relies on a simultaneous diagonalization of two matrices which, as we explain below, cannot be carried out on a Turing machine, since the matrix it produces is in general irrational. We stress that our result is thus new already for $p = 0$: for quadratic programming with a fixed number of negative eigenvalues, no polynomial-time approximation algorithm in the Turing model of computation was known.

All the algorithms above, including ours, are based on the classical technique of mesh partitioning and linear underestimators, in which some of the quadratic terms of the objective function are replaced, on each piece of a partition of the feasible region, by linear underestimators; this idea has been used in optimization since at least the 1950s. For this technique to yield an ϵ -approximate solution, one first needs a change of variables that makes the objective function *separable*, that is, that replaces H by a diagonal matrix. In the concave case, this change of variables is the only preprocessing that is needed, and it can be achieved on a Turing machine in polynomial time by the symmetric decomposition algorithm of Dax and Kaniel [3], whose running time was analyzed in [7]: that algorithm produces a rational nonsingular L with $L^\top H L$ diagonal, and, by Sylvester's law of inertia, the inertia is automatically preserved.

In the indefinite case the situation is considerably more delicate, because a separable objective function is no longer enough: one also needs the polyhedron described by the linear constraints, which we may assume to be bounded, to be of *spherical form*, that is, sandwiched between two concentric balls whose radii differ by a controlled factor. The reason is that the analysis needs to know that the polytope is neither too large nor too small in any direction, and this is exactly what the two balls guarantee. If the rank of H is fixed, which is the setting of [7], separability and spherical form are only needed on a subspace of fixed dimension, and the decomposition of Dax and Kaniel essentially suffices. If the rank of H is not fixed, as in [17] and in the present paper, this simplification is not available, and both properties are needed in full. The two requirements pull in opposite directions: a general nonsingular change of variables can diagonalize H , but it distorts the feasible region and destroys sphericity, whereas an orthogonal change of variables preserves balls but is, in general, irrational.

The main obstacle. In [17] both requirements are met at once by constructing, given the symmetric matrix H and a symmetric positive definite matrix M describing an ellipsoid that approximates the feasible region, a nonsingular matrix L such that

$$L^\top H L \text{ is diagonal} \quad \text{and} \quad L^\top M L = I, \quad (1)$$

where I denotes the identity matrix; the second condition is what turns the ellipsoid into a ball. Such a matrix L always exists over the reals, and is classically obtained from a Cholesky factorization $M = C^\top C$ followed by a spectral decomposition (see, e.g., Algorithm 8.7.1 in [11]), but both steps are in general irrational, and therefore this construction cannot be carried out on a Turing machine. Throughout, we assume that M is given already in factored form, that is, that a rational nonsingular C with $M = C^\top C$ is part of the input; this is how M arises in our application, where C is produced by Lenstra's ellipsoid rounding algorithm, and it is a genuine restriction, since such a C need not exist (Example 3). The irrationality of the spectral decomposition, however, remains, and is already apparent in dimension two.

Example 1. Let $n := 2$, let $M := I$, which is in factored form with $C := I$, and let $H := \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$. Suppose that L satisfies (1). Since $M = I$, the second condition reads $L^\top L = I$, so L is orthogonal and $L^\top H L = L^{-1} H L$ is similar to H ; being diagonal, it has the eigenvalues $\frac{1 \pm \sqrt{5}}{2}$ of H on its diagonal, and these are irrational. Hence $L^\top H L$, and therefore L , cannot be rational, even though M is given in factored form.

The obstruction is not an artifact of this small example: the columns of L are generalized eigenvectors of H with respect to M , that is, nonzero vectors v with $Hv = \lambda Mv$ for some $\lambda \in \mathbb{R}$, so computing them exactly amounts to computing eigenvalues exactly, which for $n \geq 5$ is not in general possible by radicals.

Our approach. Our approach keeps the second condition in (1) and relaxes the first. We show (Theorem 4) that a rational nonsingular L satisfying $L^\top M L = I$ *exactly*, and for which $L^\top H L$ is diagonal *up to a perturbation E of prescribed Frobenius norm at most δ* , can be computed in polynomial time; moreover the diagonal part D has the same inertia as H , which is what allows the number k_- of negative eigenvalues to be tracked through the reduction. Relaxing exact diagonality is precisely what makes L rational. The transformed problem is obtained by applying the change of variables $x = Ly$ (up to a translation), which turns the quadratic part of the objective function into $y^\top (D + E)y$, and then discarding E so as to be left with a separable objective function. The price to pay is that this problem is no longer equivalent to the original one, since discarding E perturbs its objective function. We show that this is harmless, in the sense that an approximate solution to the transformed problem yields an approximate solution to Problem MIQP, provided that the objective function varies enough on the feasible region. The transformed problem is in spherical form, so mesh partitioning and linear underestimators apply, and the resulting subproblems are mixed integer convex quadratic programs, which we solve with the polynomial-time algorithm of [8]. A theorem of the alternative, based on a flatness theorem for mixed integer lattices, removes the proviso: it either produces an approximate solution, certifying that the objective function does vary enough, or splits the instance into sub-instances with one fewer integer variable, on which we recurse.

The engine behind Theorem 4 is a rational analogue of the classical Jacobi eigenvalue algorithm, which we develop in Section 2 and which we believe is of independent interest. Its goal is Theorem 2: given a symmetric matrix H and a tolerance $\delta \in (0, 1]$, compute in polynomial time a rational *orthogonal* change of basis after which the off-diagonal part of H has Frobenius norm at most δ . A Jacobi rotation acts on a symmetric matrix by an orthogonal similarity that annihilates one off-diagonal entry; iterating on the largest off-diagonal entry drives the matrix towards diagonal form at a geometric rate. The rotation that annihilates a given entry involves square roots and is in general irrational, so the classical algorithm lives in the real RAM model. We observe that the rational points are dense on the unit circle, via the Weierstrass substitution

$$c = \frac{1 - u^2}{1 + u^2}, \quad s = \frac{2u}{1 + u^2},$$

and we use it to construct, in polynomial time, *rational* Jacobi rotations: matrices that are exactly orthogonal, have rational entries of controlled size, and annihilate the chosen off-diagonal entry not exactly but up to any prescribed tolerance. Exact orthogonality is what makes this work, and it is what distinguishes our rotations from simply rounding the entries of a real one: it keeps the Frobenius norm constant along the iterations, so that annihilating an off-diagonal entry decreases the off-diagonal norm, and it makes the accumulated change of basis orthogonal.

Two further ingredients turn near-diagonalization into diagonalization with a guarantee on the inertia. First, a classical bound of Cauchy on the roots of a polynomial shows that the nonzero eigenvalues of an integer symmetric matrix are bounded away from zero by a quantity that depends only on the size of the matrix. Second, a perturbation argument then shows that, once the off-diagonal part has been driven below that threshold, the small entries can safely be rounded to zero without changing the inertia. Together with the rational Jacobi rotations, this yields Theorem 3, which, by a rational orthogonal change of basis, makes H diagonal up to a perturbation of Frobenius norm at most δ , and guarantees that the diagonal part has the same inertia as H . Applying it to $C^{-\top} H C^{-1}$, and composing the two changes of basis, yields Theorem 4. Finally, if in place of the matrix M we are given a full-dimensional rational polytope, and C is obtained by applying the ellipsoid rounding algorithm of Lenstra [12] to it, we obtain Theorem 5, where the change of basis in addition maps the polytope to one sandwiched between two concentric balls, as required for spherical form.

Organization of the paper. Section 2 develops exact and rational Jacobi rotations, and culminates in the near-diagonalization algorithm of Theorem 2. Section 3 turns it into the inertia-preserving diagonalization algorithm of Theorem 3, and then into the simultaneous diagonalization algorithms of

Theorems 4 and 5. Section 4 introduces spherical form MIQP, shows how to reduce Problem MIQP to it, solves it by mesh partitioning and linear underestimators, proves two theorems of the alternative, and assembles the proof of Theorem 1.

2 Jacobi rotations

We say that $(c, s) \in \mathbb{R}^2$ is a *cosine-sine pair* associated with some angle $\theta \in \mathbb{R}$ if $c = \cos(\theta)$ and $s = \sin(\theta)$. Note that (c, s) is a cosine-sine pair if and only if $c^2 + s^2 = 1$. Throughout this section, n denotes an integer with $n \geq 2$. Given an index pair (p, q) that satisfies $1 \leq p < q \leq n$ and a cosine-sine pair (c, s) associated with some angle θ , a *Jacobi rotation*, denoted by $J(p, q; c, s)$, is an $n \times n$ orthogonal matrix that differs from the $n \times n$ identity matrix only in the 2×2 principal submatrix corresponding to row/columns p, q , which has the form

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix}.$$

Jacobi rotations are also known as *Givens rotations*. Multiplying a vector in $\mathbb{R}^n$ on the left by $J(p, q; c, s)$ rotates the vector through θ radians clockwise in the (p, q) plane.

Before proceeding, we first recall some standard matrix norms. Given a matrix $A \in \mathbb{R}^{m \times n}$, its *spectral norm* and *Frobenius norm* are defined by

$$\|A\|_2 := \sup_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2}, \quad \|A\|_F := \sqrt{\sum_{i=1}^m \sum_{j=1}^n a_{ij}^2}.$$

It is well known that $\|A\|_2 \leq \|A\|_F$ for every $A \in \mathbb{R}^{m \times n}$. Furthermore, viewing a vector $v \in \mathbb{R}^n$ as an $n \times 1$ matrix, its spectral norm $\|v\|_2$ coincides with its Euclidean norm. For a matrix $A \in \mathbb{R}^{n \times n}$ we use the notation

$$\text{off}(A) := \sqrt{\sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij}^2}$$

for the off-diagonal analogue of the Frobenius norm. Note that $\text{off}(A) \leq \|A\|_F$ for every $A \in \mathbb{R}^{n \times n}$, since $\|A\|_F^2 = \text{off}(A)^2 + \sum_{i=1}^n a_{ii}^2$.

2.1 The effect of a single rotation

Both the exact rotations of Section 2.2 and the rational ones of Section 2.3 act on $\text{off}(\cdot)$ through the same identity, which we record here once and for all. It holds for an arbitrary cosine-sine pair: in the exact case the rotation annihilates the (p, q) entry of the transformed matrix, while in the rational case that entry is merely small. In the former case the identity is well known.

Note that, given a symmetric matrix $A \in \mathbb{R}^{n \times n}$, an index pair (p, q) with $1 \leq p < q \leq n$, a cosine-sine pair (c, s) , and $J := J(p, q; c, s)$, the entries in rows and columns p, q of $B := J^T A J$ satisfy

$$\begin{bmatrix} b_{pp} & b_{pq} \\ b_{qp} & b_{qq} \end{bmatrix} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}^T \begin{bmatrix} a_{pp} & a_{pq} \\ a_{qp} & a_{qq} \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix}. \quad (2)$$

Lemma 1. *Let $A \in \mathbb{R}^{n \times n}$ be symmetric and let (p, q) be an index pair that satisfies $1 \leq p < q \leq n$. Let (c, s) be a cosine-sine pair, let $J := J(p, q; c, s)$, and let $B := J^T A J$. Then*

$$\text{off}(B)^2 = \text{off}(A)^2 - 2a_{pq}^2 + 2b_{pq}^2.$$

Proof. Since A is symmetric, so is $B = J^T A J$; and since J is orthogonal and the Frobenius norm is preserved by orthogonal transformations, $\|B\|_F^2 = \|A\|_F^2$. Since also the 2×2 block of J given by

$\begin{bmatrix} c & s \\ -s & c \end{bmatrix}$ is orthogonal, from (2) we obtain

$$a_{pp}^2 + a_{qq}^2 + 2a_{pq}^2 = b_{pp}^2 + b_{qq}^2 + 2b_{pq}^2.$$

Since B agrees with A except in rows and columns p and q (so that $b_{ii} = a_{ii}$ for every $i \neq p, q$), we have

$$\sum_{i=1}^n b_{ii}^2 = \sum_{i \neq p, q} b_{ii}^2 + b_{pp}^2 + b_{qq}^2 = \sum_{i \neq p, q} a_{ii}^2 + b_{pp}^2 + b_{qq}^2 = \sum_{i=1}^n a_{ii}^2 + 2a_{pq}^2 - 2b_{pq}^2.$$

Hence,

$$\text{off}(B)^2 = \|B\|_F^2 - \sum_{i=1}^n b_{ii}^2 = \|A\|_F^2 - \sum_{i=1}^n a_{ii}^2 - 2a_{pq}^2 + 2b_{pq}^2 = \text{off}(A)^2 - 2a_{pq}^2 + 2b_{pq}^2.$$

□

2.2 Exact Jacobi rotations

We first treat the classical case, in which each rotation zeroes out the chosen off-diagonal entry exactly. The results of this subsection are well known; we include their proofs for completeness, and because the rational versions of Section 2.3 follow the same pattern.

2.2.1 Construction

Off-diagonal entries of a symmetric matrix $A \in \mathbb{R}^{n \times n}$ can be zeroed out by replacing A with $J^T A J$, for some suitable Jacobi rotation J . The following lemma makes this construction explicit, giving closed-form formulas for a Jacobi rotation that zeroes out a prescribed off-diagonal entry.

Lemma 2. *Let $A \in \mathbb{R}^{n \times n}$ be symmetric and let (p, q) be an index pair that satisfies $1 \leq p < q \leq n$. If $a_{pq} = 0$, let $t := 0$; otherwise, let*

$$\tau := \frac{a_{qq} - a_{pp}}{2a_{pq}}, \quad t := \begin{cases} \sqrt{1 + \tau^2} - \tau \in (0, 1] & \text{if } \tau \geq 0 \\ -\tau - \sqrt{1 + \tau^2} \in (-1, 0) & \text{if } \tau < 0. \end{cases}$$

In both cases, let

$$\theta := \arctan(t), \quad c := \frac{1}{\sqrt{1 + t^2}}, \quad s := \frac{t}{\sqrt{1 + t^2}}.$$

Then (c, s) is a cosine-sine pair associated with angle $\theta \in (-\pi/4, \pi/4]$. Furthermore, if we let $B := J^T A J$, where J denotes the Jacobi rotation $J(p, q; c, s)$, then $b_{pq} = 0$.

Proof. In all cases $t \in (-1, 1]$: this is clear if $a_{pq} = 0$, and otherwise it follows from $|\tau| < \sqrt{1 + \tau^2} \leq 1 + |\tau|$, the second inequality being strict for $\tau \neq 0$, which also gives the ranges for t stated in the lemma. Since $\theta = \arctan(t) \in (-\pi/2, \pi/2)$, we have $\cos(\theta) > 0$. Since $\tan(\theta) = t$, we have

$$\cos(\theta) = \frac{1}{\sqrt{1 + t^2}} = c, \quad \sin(\theta) = \frac{t}{\sqrt{1 + t^2}} = s,$$

so (c, s) is the cosine-sine pair associated with θ . Since $t \in (-1, 1]$, and since $\arctan$ is increasing with $\arctan(-1) = -\pi/4$ and $\arctan(1) = \pi/4$, we get $\theta \in (-\pi/4, \pi/4]$.

It remains to show $b_{pq} = 0$. If $a_{pq} = 0$, then $t = 0$, hence $c = 1$ and $s = 0$, and (2) gives $b_{pq} = a_{pq}(c^2 - s^2) + (a_{pp} - a_{qq})cs = 0$. Assume now $a_{pq} \neq 0$. From (2),

$$b_{pq} = a_{pq}(c^2 - s^2) + (a_{pp} - a_{qq})cs = a_{pq} \cdot \frac{1 - t^2}{1 + t^2} + (a_{pp} - a_{qq}) \cdot \frac{t}{1 + t^2} = \frac{a_{pq}((1 - t^2) - 2\tau t)}{1 + t^2},$$

where we used $a_{pp} - a_{qq} = -2a_{pq}\tau$ (from the definition of τ , valid since $a_{pq} \neq 0$). It then suffices to show $t^2 + 2\tau t - 1 = 0$.

If $\tau \geq 0$, then $t = \sqrt{1 + \tau^2} - \tau$, so

$$t^2 + 2\tau t = \left(1 + 2\tau^2 - 2\tau\sqrt{1 + \tau^2}\right) + \left(2\tau\sqrt{1 + \tau^2} - 2\tau^2\right) = 1.$$

If $\tau < 0$, then $t = -\tau - \sqrt{1 + \tau^2}$, so

$$t^2 + 2\tau t = \left(1 + 2\tau^2 + 2\tau\sqrt{1 + \tau^2}\right) + \left(-2\tau^2 - 2\tau\sqrt{1 + \tau^2}\right) = 1.$$

In either case $t^2 + 2\tau t = 1$, hence $b_{pq} = 0$, as desired. $\square$

2.2.2 Repeated updates

Combining Lemmas 1 and 2, repeatedly zeroing out the largest-magnitude off-diagonal entry, one Jacobi rotation at a time, drives $\text{off}(\cdot)$ toward zero at a geometric rate; this is a classical fact, which we state below in the form convenient for our later use.

Let $A \in \mathbb{R}^{n \times n}$ be symmetric, and let (p, q) be an index pair that satisfies $1 \leq p < q \leq n$ and $|a_{pq}| = \max_{i \neq j} |a_{ij}|$. We say that $B \in \mathbb{R}^{n \times n}$ is obtained by applying a *Jacobi update* to A if $B = J^\top A J$ for some cosine-sine pair (c, s) , where $J := J(p, q; c, s)$, such that $b_{pq} = 0$. Note that such a pair (c, s) always exists, and can be constructed via the closed-form formulas in Lemma 2.

Lemma 3. *Let $A \in \mathbb{R}^{n \times n}$ be symmetric and let $\delta \in (0, 1]$. For every nonnegative integer k , let $A^{(k)}$ denote a matrix obtained from A by k successive Jacobi updates (with $A^{(0)} := A$). Then, there exists*

$$k \leq \max \left(0, \left\lceil \frac{n(n-1)}{2} \ln \left(\frac{\text{off}(A)^2}{\delta^2} \right) \right\rceil \right)$$

such that $\text{off}(A^{(k)}) \leq \delta$.

Proof. Let $N := \frac{n(n-1)}{2}$ and $\xi := \ln(\text{off}(A)^2/\delta^2)$. If $\text{off}(A)^2 < \delta^2$, then $\text{off}(A) \leq \delta$, and $k = 0$ satisfies the claim. We may therefore assume $\text{off}(A)^2 \geq \delta^2$, so that $\xi \geq 0$. Let $K := \lceil N\xi \rceil$, a nonnegative integer. We show that $\text{off}(A^{(K)}) \leq \delta$. Let a_{pq} be the off-diagonal entry of A with largest absolute value. Then

$$\text{off}(A)^2 \leq n(n-1)a_{pq}^2 = 2Na_{pq}^2.$$

Since a Jacobi update satisfies $b_{pq} = 0$, Lemma 1 gives

$$\text{off}(A^{(1)})^2 = \text{off}(A)^2 - 2a_{pq}^2 \leq \left(1 - \frac{1}{N}\right) \text{off}(A)^2.$$

Since the same argument applies to each $A^{(k)}$, whose pivot is again an off-diagonal entry of largest absolute value, by induction

$$\text{off}(A^{(K)})^2 \leq \left(1 - \frac{1}{N}\right)^K \text{off}(A)^2.$$

Since $K \geq N\xi$ and $0 \leq 1 - \frac{1}{N} < 1$, we have $\left(1 - \frac{1}{N}\right)^K \leq \left(1 - \frac{1}{N}\right)^{N\xi}$. Since $1 - \frac{1}{N} \leq e^{-1/N}$, we also have $\left(1 - \frac{1}{N}\right)^{N\xi} \leq e^{-\xi}$. Thus,

$$\text{off}(A^{(K)})^2 \leq \left(1 - \frac{1}{N}\right)^K \text{off}(A)^2 \leq e^{-\xi} \text{off}(A)^2 = \frac{\delta^2}{\text{off}(A)^2} \text{off}(A)^2 = \delta^2,$$

so $\text{off}(A^{(K)}) \leq \delta$, and $k = K$ satisfies the claim. $\square$

2.2.3 Near-diagonalization via exact rotations

Combining Lemmas 2 and 3 yields a real-RAM algorithm for driving a symmetric matrix arbitrarily close to diagonal form; we record this immediate consequence below before turning, in the next section, to a version of this result suitable for computation on a Turing machine. In the real RAM model we assume, as is standard, that square roots can be computed at unit cost. Throughout, we state our bounds in terms of $\log(1/\delta)$ when they depend only on the precision requested, as for the number of arithmetic operations in the real RAM model, and in terms of $\text{size}(\delta)$ when they depend on the encoding of δ , as for running times on a Turing machine. The two are not interchangeable: for every rational $\delta \in (0, 1]$ we have $\lceil \log(1/\delta) \rceil \leq \text{size}(\delta)$, but $\text{size}(\delta)$ also accounts for the number of bits needed to encode δ , which may be much larger than $\log(1/\delta)$.

Proposition 1 (Near-diagonalization via exact Jacobi rotations). *Let $A \in \mathbb{R}^{n \times n}$ be symmetric and let $\delta \in (0, 1]$. There is an algorithm that computes an orthogonal matrix $L \in \mathbb{R}^{n \times n}$ such that $\text{off}(L^\top A L) \leq \delta$. The algorithm performs $O(n^4(1 + \log(1/\delta) + \log(\max(1, \|A\|_F))))$ arithmetic operations in the real RAM model.*

Proof. From Lemma 3, there exists

$$k \leq \max\left(0, \left\lceil \frac{n(n-1)}{2} \ln\left(\frac{\text{off}(A)^2}{\delta^2}\right) \right\rceil\right)$$

such that after k Jacobi updates, the obtained matrix $A^{(k)}$ satisfies $\text{off}(A^{(k)}) \leq \delta$. Let $J_1, \dots, J_k$ be the corresponding Jacobi rotations, each constructed via Lemma 2 to zero out the largest-magnitude off-diagonal entry, and let $L := J_1 \cdots J_k$, so that $L^\top A L = A^{(k)}$. As a product of orthogonal matrices, L is orthogonal.

Running time. Since $\text{off}(A) \leq \|A\|_F$, Lemma 3 bounds the number of updates by

$$k = O(n^2(\log(1/\delta) + \log(\max(1, \|A\|_F)))) .$$

Each update costs $O(n^2)$ arithmetic operations: $O(n^2)$ comparisons to find the pivot (p, q) , $O(1)$ to construct J_{t+1} from the closed-form formulas of Lemma 2, and $O(n^2)$ to form $A^{(t+1)} = J_{t+1}^\top A^{(t)} J_{t+1}$, since only rows and columns p, q change. Taking into account the $O(n^2)$ arithmetic operations needed to form the output L , which are performed even when $k = 0$, the total is therefore

$$O(n^2(1 + k)) = O(n^4(1 + \log(1/\delta) + \log(\max(1, \|A\|_F)))) .$$

□

2.3 Rational Jacobi rotations

Proposition 1 drives $\text{off}(\cdot)$ below any prescribed tolerance using a number of arithmetic operations that is polynomial in n and in $\log(1/\delta)$, but it does so in the real RAM model, and it cannot be implemented on a Turing machine: both the Jacobi rotation J constructed in Lemma 2 and the resulting matrix $B := J^\top A J$ can have irrational entries, as suggested by the presence of square roots in the formulas defining c and s .

Example 2. Let $n := 2$, let $A := \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$, and let $(p, q) := (1, 2)$. We have $\tau = 1/2$ and $t = (\sqrt{5} - 1)/2$ in Lemma 2, so that

$$c = \sqrt{\frac{5 + \sqrt{5}}{10}}, \quad s = \sqrt{\frac{5 - \sqrt{5}}{10}}$$

are both irrational (since $c^2 = \frac{5 + \sqrt{5}}{10}$ and $s^2 = \frac{5 - \sqrt{5}}{10}$ are themselves irrational, as $\sqrt{5} \notin \mathbb{Q}$). Thus, even though A has integer entries, the Jacobi rotation $J(p, q; c, s)$ diagonalizing A has irrational entries.

Moreover, $B := J^T A J$ has irrational entries as well:

$$B = J^T A J = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}^T \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix} = \begin{bmatrix} \frac{1-\sqrt{5}}{2} & 0 \\ 0 & \frac{1+\sqrt{5}}{2} \end{bmatrix}.$$

In practice, c and s are often rounded, but this in general destroys orthogonality, so the resulting matrix is not a Jacobi rotation. Our next goal is instead to approximate a target Jacobi rotation by another Jacobi rotation that is rational and can be computed in polynomial time. This is possible because rational points are dense on the two-dimensional unit circle, as can be seen via the Weierstrass substitution

$$c = \frac{1-u^2}{1+u^2}, \quad s = \frac{2u}{1+u^2},$$

also known as the $\tan(\theta/2)$ substitution: as u ranges over the rationals, this parametrizes a dense set of rational points on the circle. We will use this substitution throughout our proofs, together with the following simple lemma.

Lemma 4. *Let $\delta \geq 0$ and let $a, \hat{a}, b, \hat{b} \in \mathbb{R}$ be such that $|a - \hat{a}| \leq \delta$ and $|b - \hat{b}| \leq \delta$.*

1. *If $|a|, |\hat{a}| \leq 1$, then $|a^2 - \hat{a}^2| \leq 2\delta$.*
2. *If $|a|, |\hat{b}| \leq 1$, then $|ab - \hat{a}\hat{b}| \leq 2\delta$.*
3. *If $|b|, |\hat{b}| \geq 1$, then*

$$\left| \frac{a}{b} - \frac{\hat{a}}{\hat{b}} \right| \leq (|a| + |b|)\delta.$$

Proof. 1. We have $|a^2 - \hat{a}^2| = |a + \hat{a}| \cdot |a - \hat{a}| \leq 2\delta$.

2. We have $|ab - \hat{a}\hat{b}| = |a(b - \hat{b}) + \hat{b}(a - \hat{a})| \leq |a| \cdot |b - \hat{b}| + |\hat{b}| \cdot |a - \hat{a}| \leq 2\delta$.

3. We have

$$\begin{aligned} \left| \frac{a}{b} - \frac{\hat{a}}{\hat{b}} \right| &= \left| \frac{a\hat{b} - \hat{a}b}{b\hat{b}} \right| \leq |a\hat{b} - \hat{a}b| = |a(\hat{b} - b) + b(a - \hat{a})| \\ &\leq |a| \cdot |\hat{b} - b| + |b| \cdot |a - \hat{a}| \leq (|a| + |b|)\delta, \end{aligned}$$

where the first inequality uses $|b\hat{b}| \geq 1$. □

2.3.1 Construction

The next lemma is a rational, algorithmic version of Lemma 2: rather than the exact (generally irrational) cosine-sine pair (c, s) produced by that lemma, it efficiently computes a nearby rational cosine-sine pair $(\hat{c}, \hat{s})$.

Lemma 5. *Let $A \in \mathbb{R}^{n \times n}$ be symmetric and let (p, q) be an index pair that satisfies $1 \leq p < q \leq n$. Let c, s be defined as in Lemma 2, and let $\delta \in (0, 1]$. There is an algorithm that, on input a_{pp}, a_{qq}, a_{pq} and δ , finds integers P_1, P_2, Q with $Q > 0$, $P_1^2 + P_2^2 = Q^2$, and $\text{size}(Q) \leq 2\lceil \log(1/\delta) \rceil + 10$, such that, letting $\hat{c} := P_1/Q$ and $\hat{s} := P_2/Q$,*

$$|c - \hat{c}| \leq \delta, \quad |s - \hat{s}| \leq \delta.$$

The algorithm performs $O(1 + \log(1/\delta))$ arithmetic operations in the real RAM model. Moreover, if $a_{pp}, a_{qq}, a_{pq}, \delta \in \mathbb{Q}$, the algorithm runs on a Turing machine in time polynomial in $\text{size}(a_{pp}) + \text{size}(a_{qq}) + \text{size}(a_{pq}) + \text{size}(\delta)$.

Proof. If $a_{pq} = 0$, then $t = 0$, $c = 1$, and $s = 0$, and the algorithm returns $(P_1, P_2, Q) := (1, 0, 1)$: then $P_1^2 + P_2^2 = Q^2$, the bound on $\text{size}(Q)$ holds trivially, and $\hat{c} = c$, $\hat{s} = s$. We may therefore assume $a_{pq} \neq 0$. We compute $\tau \in \mathbb{R}$ as defined in Lemma 2, and we define $d := \lceil \log(8/\delta) \rceil = 3 + \lceil \log(1/\delta) \rceil$, which is a positive integer since $\delta \leq 1$. Let θ, t be as in Lemma 2 and let

$$u := \tan(\theta/2) = \frac{\sqrt{1+t^2}-1}{t}.$$

Construction of $\hat{u}$ with $|u - \hat{u}| \leq 2^{-d}$. We give an algorithm that takes in input τ , and returns $\hat{u} \in \mathbb{Q} \cap [-1, 1]$ with $\text{size}(\hat{u}) \leq 2d + 3$ so that

$$|u - \hat{u}| \leq 2^{-d}.$$

First, we consider the case $\tau \geq 0$. The algorithm is based on a simple binary search. Since $t \in (0, 1]$, we have $\theta \in (0, \pi/4]$ and hence $u = \tan(\theta/2) \in (0, 1)$, thus we start from the interval $[0, 1]$. At each iteration we compare the midpoint $m > 0$ of the interval with u using the conditions on $u \lesseqgtr m$ derived below:

$$\begin{aligned} u \lesseqgtr m &\Leftrightarrow \frac{\sqrt{1 + (\sqrt{1 + \tau^2} - \tau)^2} - 1}{\sqrt{1 + \tau^2} - \tau} \lesseqgtr m \\ &\Leftrightarrow \sqrt{1 + (\sqrt{1 + \tau^2} - \tau)^2} \lesseqgtr m\sqrt{1 + \tau^2} - m\tau + 1 \\ &\Leftrightarrow 1 + (\sqrt{1 + \tau^2} - \tau)^2 \lesseqgtr (m\sqrt{1 + \tau^2} - m\tau + 1)^2 \\ &\Leftrightarrow 2 + 2\tau^2 - 2\tau\sqrt{1 + \tau^2} \lesseqgtr 2m^2\tau^2 + m^2 - 2m\tau + 1 - 2m(m\tau - 1)\sqrt{1 + \tau^2} \\ &\Leftrightarrow \underbrace{(2m^2\tau - 2m - 2\tau)}_{=: \alpha} \sqrt{1 + \tau^2} \lesseqgtr \underbrace{(2m^2\tau^2 + m^2 - 2m\tau - 2\tau^2 - 1)}_{=: \gamma} \\ &\Leftrightarrow \begin{cases} -1 \lesseqgtr 1 & \text{if } \alpha < 0, \gamma \geq 0 \\ 1 \lesseqgtr -1 & \text{if } \alpha \geq 0, \gamma < 0 \\ \alpha^2(1 + \tau^2) \lesseqgtr \gamma^2 & \text{if } \alpha, \gamma \geq 0 \\ \gamma^2 \lesseqgtr \alpha^2(1 + \tau^2) & \text{if } \alpha, \gamma < 0. \end{cases} \end{aligned}$$

At the end of iteration d , we have an interval of length 2^{-d} , with rational extreme points in $[0, 1]$ of the form $p/2^d$ with $p \in \{0, 1, \dots, 2^d\}$. Since this interval contains u , either extreme point is at distance at most 2^{-d} from u . We then set $\hat{u}$ to be any such extreme point.

Next, we consider the case $\tau < 0$. Again, our algorithm is based on a simple binary search. Since $t \in (-1, 0)$, we have $\theta \in (-\pi/4, 0)$ and hence $u = \tan(\theta/2) \in (-1, 0)$, thus we start from the interval $[-1, 0]$. At each iteration we compare the midpoint $m < 0$ of the interval with u using the conditions on

$u \lesseqgtr m$ derived below:

$$\begin{aligned}
u \lesseqgtr m &\Leftrightarrow \frac{\sqrt{1 + (-\tau - \sqrt{1 + \tau^2})^2} - 1}{-\tau - \sqrt{1 + \tau^2}} \lesseqgtr m \\
&\Leftrightarrow -m\sqrt{1 + \tau^2} - m\tau + 1 \lesseqgtr \sqrt{1 + (-\tau - \sqrt{1 + \tau^2})^2} \\
&\Leftrightarrow \left(-m\sqrt{1 + \tau^2} - m\tau + 1\right)^2 \lesseqgtr 1 + (-\tau - \sqrt{1 + \tau^2})^2 \\
&\Leftrightarrow 2m^2\tau^2 + m^2 - 2m\tau + 1 - 2m(1 - m\tau)\sqrt{1 + \tau^2} \lesseqgtr 2 + 2\tau^2 + 2\tau\sqrt{1 + \tau^2} \\
&\Leftrightarrow \underbrace{2m^2\tau^2 + m^2 - 2m\tau - 2\tau^2 - 1}_{=: \gamma} \lesseqgtr \underbrace{(2m + 2\tau - 2m^2\tau)}_{=: \beta} \sqrt{1 + \tau^2} \\
&\Leftrightarrow \begin{cases} 1 \lesseqgtr -1 & \text{if } \gamma \geq 0, \beta < 0 \\ -1 \lesseqgtr 1 & \text{if } \gamma < 0, \beta \geq 0 \\ \gamma^2 \lesseqgtr \beta^2 (1 + \tau^2) & \text{if } \gamma, \beta \geq 0 \\ \beta^2 (1 + \tau^2) \lesseqgtr \gamma^2 & \text{if } \gamma, \beta < 0. \end{cases}
\end{aligned}$$

At the end of iteration d , we have an interval of length 2^{-d} , with rational extreme points in $[-1, 0]$ of the form $p/2^d$ with $p \in \{-2^d, -2^d + 1, \dots, 0\}$. Since this interval contains u , either extreme point is at distance at most 2^{-d} from u . We then set $\hat{u}$ to be any such extreme point.

In both cases, since $\hat{u} = p/2^d$ for some integer p with $|p| \leq 2^d$, we have

$$\text{size}(\hat{u}) \leq 1 + \lceil \log(2^d + 1) \rceil + \lceil \log(2^d + 1) \rceil = 2d + 3.$$

We remark that the above comparisons only involve the rational quantities $\alpha, \beta, \gamma, \tau, m$, and never require evaluating the irrational quantity $\sqrt{1 + \tau^2}$ itself; this is what allows the algorithm to run in polynomial time on a Turing machine.

Construction of $(\hat{c}, \hat{s})$. In the remainder of the proof, we denote $\hat{\theta} := 2 \arctan(\hat{u})$ so that $\hat{u} = \tan(\hat{\theta}/2)$. Since $\hat{u} = p/2^d$ with $|p| \leq 2^d$, define

$$P_1 := 2^{2d} - p^2, \quad P_2 := 2p \cdot 2^d, \quad Q := 2^{2d} + p^2,$$

and let $\hat{c} := P_1/Q$, $\hat{s} := P_2/Q$. Note that

$$\hat{c} = \cos(\hat{\theta}) = \frac{1 - \hat{u}^2}{1 + \hat{u}^2}, \quad \hat{s} = \sin(\hat{\theta}) = \frac{2\hat{u}}{1 + \hat{u}^2},$$

so $(\hat{c}, \hat{s})$ is the cosine-sine pair associated with $\hat{\theta}$, approximating

$$c = \cos(\theta) = \frac{1 - u^2}{1 + u^2} \in \mathbb{R}, \quad s = \sin(\theta) = \frac{2u}{1 + u^2} \in \mathbb{R}.$$

Also, $Q > 0$, and

$$P_1^2 + P_2^2 = (2^{2d})^2 - 2 \cdot 2^{2d} p^2 + p^4 + 4p^2 \cdot 2^{2d} = (2^{2d})^2 + 2 \cdot 2^{2d} p^2 + p^4 = (2^{2d} + p^2)^2 = Q^2,$$

implying $\hat{c}^2 + \hat{s}^2 = 1$. Since $0 \leq p^2 \leq 2^{2d}$, we have $2^{2d} \leq Q \leq 2 \cdot 2^{2d} = 2^{2d+1}$, so

$$\text{size}(Q) \leq 1 + \lceil \log(2^{2d+1} + 1) \rceil + \lceil \log 2 \rceil = 1 + (2d + 2) + 1 = 2d + 4 = 2\lceil \log(1/\delta) \rceil + 10.$$

Bounding $|c - \hat{c}|$ and $|s - \hat{s}|$. From Lemma 4.1 (with $a = u, \hat{a} = \hat{u}$), since $u, \hat{u} \in [-1, 1]$ and $|u - \hat{u}| \leq 2^{-d}$, we obtain

$$|u^2 - \hat{u}^2| \leq 2 \cdot 2^{-d} = 2^{-d+1}.$$

From Lemma 4.3 (with $a = 1 - u^2$, $\hat{a} = 1 - \hat{u}^2$, $b = 1 + u^2$, $\hat{b} = 1 + \hat{u}^2$), we obtain

$$|c - \hat{c}| \leq 3 \cdot 2^{-d+1} \leq 2^{-d+3} \leq \delta,$$

where we used bounds $|1 - u^2| \leq 1$, $|1 + u^2| \leq 2$ and $|1 + u^2|, |1 + \hat{u}^2| \geq 1$. Similarly, from Lemma 4.3 (with $a = 2u$, $\hat{a} = 2\hat{u}$, $b = 1 + u^2$, $\hat{b} = 1 + \hat{u}^2$), we obtain

$$|s - \hat{s}| \leq 4 \cdot 2^{-d+1} = 2^{-d+3} \leq \delta,$$

where we used bounds $|2u| \leq 2$, $|1 + u^2| \leq 2$ and $|1 + u^2|, |1 + \hat{u}^2| \geq 1$.

Running time. Computing d from δ , performing the d binary-search iterations that construct $\hat{u}$, and constructing $(\hat{c}, \hat{s})$ with a further $O(1)$ arithmetic operations amount to $O(d) = O(1 + \log(1/\delta))$ arithmetic operations in total; this establishes the real RAM bound. Now suppose $a_{pp}, a_{qq}, a_{pq}, \delta \in \mathbb{Q}$. Then τ has size $O(\text{size}(a_{pp}) + \text{size}(a_{qq}) + \text{size}(a_{pq}))$, and the midpoint m at iteration i has size $O(i) = O(d)$, so the quantities α, β, γ , obtained from τ and m by $O(1)$ arithmetic operations, have size $O(\text{size}(a_{pp}) + \text{size}(a_{qq}) + \text{size}(a_{pq}) + \text{size}(\delta))$, where we used $d = O(\text{size}(\delta))$. Since each arithmetic operation on rationals of size L can be performed in time polynomial in L , and d is computable from δ in polynomial time, each of the $O(d)$ iterations and the $O(1)$ arithmetic operations forming $(\hat{c}, \hat{s})$ run in polynomial time; hence the entire algorithm runs on a Turing machine in time polynomial in $\text{size}(a_{pp}) + \text{size}(a_{qq}) + \text{size}(a_{pq}) + \text{size}(\delta)$. $\square$

The next lemma builds on Lemma 5: for a given tolerance δ , we show that the exact condition $b_{pq} = 0$ from Lemma 2 can be relaxed to $|\hat{b}_{pq}| \leq \delta$, for a suitable rotation $\hat{J}$. This is achieved by invoking Lemma 5 with a tolerance δ' obtained from δ by scaling it down by the magnitude of a_{pp}, a_{qq}, a_{pq} , which is how accurately $\hat{c}$ and $\hat{s}$ must approximate c and s in order for $\hat{b}_{pq}$ to be at most δ .

Lemma 6. *Let $A \in \mathbb{R}^{n \times n}$ be symmetric, let (p, q) be an index pair that satisfies $1 \leq p < q \leq n$, and let $\delta \in (0, 1]$. Let $S := \max \{0, \lceil \log(4(|a_{pq}| + |a_{pp}| + |a_{qq}|)) \rceil\}$. There is an algorithm that, on input a_{pp}, a_{qq}, a_{pq} and δ , finds integers P_1, P_2, Q with $Q > 0$, $P_1^2 + P_2^2 = Q^2$, and $\text{size}(Q) \leq 2\lceil \log(1/\delta) \rceil + 2S + 10$, such that, letting $\hat{c} := P_1/Q$, $\hat{s} := P_2/Q$, $\hat{J} := J(p, q; \hat{c}, \hat{s})$, and $\hat{B} := \hat{J}^\top A \hat{J}$, we have $|\hat{b}_{pq}| \leq \delta$. The algorithm performs $O(1 + \log(1/\delta) + S)$ arithmetic operations in the real RAM model. Moreover, if $a_{pp}, a_{qq}, a_{pq}, \delta \in \mathbb{Q}$, the algorithm runs on a Turing machine in time polynomial in $\text{size}(a_{pp}) + \text{size}(a_{qq}) + \text{size}(a_{pq}) + \text{size}(\delta)$.*

Proof. If $a_{pq} = 0$, the algorithm returns $(P_1, P_2, Q) := (1, 0, 1)$: then $P_1^2 + P_2^2 = Q^2$, the bound on $\text{size}(Q)$ holds trivially, and $\hat{J}$ is the identity, so $\hat{b}_{pq} = a_{pq} = 0 \leq \delta$. We may therefore assume $a_{pq} \neq 0$. Let c, s be as defined in Lemma 2, let $J := J(p, q; c, s)$, and $B := J^\top A J$. Define

$$\delta' := \min \left\{ 1, \frac{\delta}{4(|a_{pq}| + |a_{pp}| + |a_{qq}|)} \right\} \in (0, 1]$$

(note the denominator is positive, since $a_{pq} \neq 0$). We apply Lemma 5 with δ' in place of δ , and obtain integers P_1, P_2, Q with $Q > 0$, $P_1^2 + P_2^2 = Q^2$, and $\text{size}(Q) \leq 2\lceil \log(1/\delta') \rceil + 10$, such that, letting $\hat{c} := P_1/Q$, $\hat{s} := P_2/Q$,

$$|c - \hat{c}| \leq \delta', \quad |s - \hat{s}| \leq \delta'.$$

Let $\hat{J} := J(p, q; \hat{c}, \hat{s})$.

Consider the entries in rows and columns p, q in the matrix products $B = J^\top A J$ and $\hat{B} = \hat{J}^\top A \hat{J}$:

$$\begin{aligned} \begin{bmatrix} b_{pp} & b_{pq} \\ b_{qp} & b_{qq} \end{bmatrix} &= \begin{bmatrix} c & s \\ -s & c \end{bmatrix}^\top \begin{bmatrix} a_{pp} & a_{pq} \\ a_{qp} & a_{qq} \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix}, \\ \begin{bmatrix} \hat{b}_{pp} & \hat{b}_{pq} \\ \hat{b}_{qp} & \hat{b}_{qq} \end{bmatrix} &= \begin{bmatrix} \hat{c} & \hat{s} \\ -\hat{s} & \hat{c} \end{bmatrix}^\top \begin{bmatrix} a_{pp} & a_{pq} \\ a_{qp} & a_{qq} \end{bmatrix} \begin{bmatrix} \hat{c} & \hat{s} \\ -\hat{s} & \hat{c} \end{bmatrix}. \end{aligned}$$

Hence,

$$\begin{aligned} b_{pq} &= 0 = a_{pq}(c^2 - s^2) + (a_{pp} - a_{qq})cs, \\ \hat{b}_{pq} &= a_{pq}(\hat{c}^2 - \hat{s}^2) + (a_{pp} - a_{qq})\hat{c}\hat{s}. \end{aligned}$$

Taking the difference of the two equations,

$$\hat{b}_{pq} = a_{pq}((\hat{c}^2 - c^2) - (\hat{s}^2 - s^2)) + (a_{pp} - a_{qq})(\hat{c}\hat{s} - cs).$$

Since $c, s, \hat{c}, \hat{s} \in [-1, 1]$, we can apply Lemma 4.1 and Lemma 4.2 (with $\delta := \delta'$) to obtain

$$\begin{aligned} |\hat{b}_{pq}| &\leq |a_{pq}|(|\hat{c}^2 - c^2| + |\hat{s}^2 - s^2|) + |a_{pp} - a_{qq}||\hat{c}\hat{s} - cs| \\ &\leq 4\delta'|a_{pq}| + 2\delta'|a_{pp} - a_{qq}| \\ &\leq \delta'(4|a_{pq}| + 2|a_{pp}| + 2|a_{qq}|) \\ &\leq \delta' \cdot 4(|a_{pq}| + |a_{pp}| + |a_{qq}|). \end{aligned}$$

By definition of δ' , we have $\delta' \cdot 4(|a_{pq}| + |a_{pp}| + |a_{qq}|) \leq \delta$: this is an equality if $\delta' < 1$, and otherwise $\delta' = 1$ and $4(|a_{pq}| + |a_{pp}| + |a_{qq}|) \leq \delta$. Therefore

$$|\hat{b}_{pq}| \leq \delta' \cdot 4(|a_{pq}| + |a_{pp}| + |a_{qq}|) \leq \delta.$$

Furthermore, since $1/\delta' = \max\{1, 4(|a_{pq}| + |a_{pp}| + |a_{qq}|)/\delta\}$ and $S \geq \log(4(|a_{pq}| + |a_{pp}| + |a_{qq}|))$, we have $\log(1/\delta') \leq S + \log(1/\delta)$, and hence, S being an integer, $\lceil \log(1/\delta') \rceil \leq S + \lceil \log(1/\delta) \rceil$; the bound $\text{size}(Q) \leq 2\lceil \log(1/\delta) \rceil + 2S + 10$ follows.

Running time. Besides invoking Lemma 5 with δ' , the algorithm only computes δ' from a_{pp} , a_{qq} , a_{pq} , and δ , which takes $O(1)$ arithmetic operations; since $\log(1/\delta') \leq S + \log(1/\delta)$, the total is $O(1 + \log(1/\delta')) = O(1 + \log(1/\delta) + S)$, establishing the real RAM bound. Now suppose $a_{pp}, a_{qq}, a_{pq}, \delta \in \mathbb{Q}$. Then δ' is rational of size polynomial in $\text{size}(a_{pp}) + \text{size}(a_{qq}) + \text{size}(a_{pq}) + \text{size}(\delta)$, and by Lemma 5 the entire algorithm runs on a Turing machine in time polynomial in $\text{size}(a_{pp}) + \text{size}(a_{qq}) + \text{size}(a_{pq}) + \text{size}(\delta')$, hence in $\text{size}(a_{pp}) + \text{size}(a_{qq}) + \text{size}(a_{pq}) + \text{size}(\delta)$. $\square$

2.3.2 Repeated approximate updates

Let $A \in \mathbb{R}^{n \times n}$ be symmetric, let (p, q) be an index pair that satisfies $1 \leq p < q \leq n$ and $|a_{pq}| = \max_{i \neq j} |a_{ij}|$, and let $\delta \in (0, 1]$. We say that $\hat{B} \in \mathbb{R}^{n \times n}$ is obtained by applying a δ -approximate Jacobi update to A if $\hat{B} = \hat{J}^\top A \hat{J}$ for some cosine-sine pair $(\hat{c}, \hat{s})$, where $\hat{J} := J(p, q; \hat{c}, \hat{s})$, such that $|\hat{b}_{pq}| \leq \delta$. A δ -approximate Jacobi update is an approximate version of a Jacobi update, where the exact condition $b_{pq} = 0$ is relaxed to $|\hat{b}_{pq}| \leq \delta$. Such a pair $(\hat{c}, \hat{s})$ can be computed via Lemma 6.

The next two lemmas give approximate versions of Lemma 3: exact Jacobi updates are replaced with approximate ones, and correspondingly the constant N in the contraction factor $1 - 1/N$ changes from $\frac{n(n-1)}{2}$ to $\frac{2}{3}n(n-1)$, to compensate for the weaker per-step contraction that Lemma 1 yields when $\hat{b}_{pq}$ is only small rather than zero. Unlike Lemma 3, whose conclusion is stated in terms of $\text{off}(A^{(k)})$, we state the first lemma in terms of individual off-diagonal entries, since this is the natural form here: $(\delta/2)$ -approximate Jacobi updates are precisely what is needed to obtain an entrywise bound of δ . The subsequent lemma then recovers a conclusion in terms of $\text{off}(\hat{A}^{(k)})$, the form needed to match Proposition 1 and to serve as input to Theorem 2.

Lemma 7. *Let $A \in \mathbb{R}^{n \times n}$ be symmetric and let $\delta \in (0, 1]$. For every nonnegative integer k , let $\hat{A}^{(k)}$ denote a matrix obtained from A by k successive $(\delta/2)$ -approximate Jacobi updates (with $\hat{A}^{(0)} := A$). Then, there exists*

$$k \leq \max \left(0, \left\lceil \frac{2}{3}n(n-1) \ln \left(\frac{\text{off}(A)^2}{\delta^2} \right) \right\rceil \right)$$

such that all off-diagonal entries of $\hat{A}^{(k)}$ are at most δ in absolute value.

Proof. Let $N := \frac{2}{3}n(n-1)$ and $\xi := \ln(\text{off}(A)^2/\delta^2)$. If $\text{off}(A)^2 < \delta^2$, then every off-diagonal entry of A has absolute value at most δ , and $k = 0$ satisfies the claim. We may therefore assume $\text{off}(A)^2 \geq \delta^2$, so that $\xi \geq 0$. Let $K := \lceil N\xi \rceil$, a nonnegative integer. If there exists $k \leq K-1$ such that all off-diagonal entries of $\hat{A}^{(k)}$ are at most δ in absolute value we are done. We may therefore assume that for each $k \leq K-1$ there is at least one off-diagonal entry of $\hat{A}^{(k)}$ that is larger than δ in absolute value. We show that all off-diagonal entries of $\hat{A}^{(K)}$ are at most δ in absolute value. Let a_{pq} be the off-diagonal entry of A with largest absolute value. Then

$$\text{off}(A)^2 \leq n(n-1)a_{pq}^2 = \frac{3}{2}Na_{pq}^2.$$

Since a $(\delta/2)$ -approximate Jacobi update satisfies $|\hat{b}_{pq}| \leq \delta/2$, Lemma 1 gives

$$\text{off}(\hat{A}^{(1)})^2 = \text{off}(A)^2 - 2a_{pq}^2 + 2\hat{b}_{pq}^2 \leq \text{off}(A)^2 - 2a_{pq}^2 + \frac{\delta^2}{2} \leq \text{off}(A)^2 - \frac{3}{2}a_{pq}^2 \leq \left(1 - \frac{1}{N}\right) \text{off}(A)^2,$$

where the second inequality uses $2\hat{b}_{pq}^2 \leq 2(\delta/2)^2 = \delta^2/2$, the third uses $|a_{pq}| > \delta$, which gives $\delta^2/2 \leq a_{pq}^2/2$, and the last uses $\text{off}(A)^2 \leq \frac{3}{2}Na_{pq}^2$. Since for every $k \leq K-1$ the off-diagonal entry of $\hat{A}^{(k)}$ with largest absolute value has absolute value larger than δ , by induction,

$$\text{off}(\hat{A}^{(K)})^2 \leq \left(1 - \frac{1}{N}\right)^K \text{off}(A)^2.$$

Since $K \geq N\xi$ and $0 < 1 - \frac{1}{N} < 1$, we have $(1 - \frac{1}{N})^K \leq (1 - \frac{1}{N})^{N\xi}$. Since $1 - \frac{1}{N} \leq e^{-1/N}$, we also have $(1 - \frac{1}{N})^{N\xi} \leq e^{-\xi}$. Thus,

$$\text{off}(\hat{A}^{(K)})^2 \leq \left(1 - \frac{1}{N}\right)^K \text{off}(A)^2 \leq e^{-\xi} \text{off}(A)^2 = \frac{\delta^2}{\text{off}(A)^2} \text{off}(A)^2 = \delta^2.$$

Therefore, each off-diagonal entry of $\hat{A}^{(K)}$ is, in absolute value, at most δ , and $k = K$ satisfies the claim. $\square$

Lemma 8. Let $A \in \mathbb{R}^{n \times n}$ be symmetric and let $\delta \in (0, 1]$. For every nonnegative integer k , let $\hat{A}^{(k)}$ denote a matrix obtained from A by k successive $(\delta/(2n))$ -approximate Jacobi updates (with $\hat{A}^{(0)} := A$). Then, there exists

$$k \leq \max \left(0, \left\lceil \frac{2}{3}n(n-1) \ln \left(\frac{n^2 \text{off}(A)^2}{\delta^2} \right) \right\rceil \right)$$

such that $\text{off}(\hat{A}^{(k)}) \leq \delta$.

Proof. Note that $\delta/n \in (0, 1]$, since $n \geq 1$. By Lemma 7 (applied with δ/n in place of δ , so that its updates are exactly the $(\delta/(2n))$ -approximate ones considered here), there exists

$$k \leq \max \left(0, \left\lceil \frac{2}{3}n(n-1) \ln \left(\frac{n^2 \text{off}(A)^2}{\delta^2} \right) \right\rceil \right)$$

such that all off-diagonal entries of $\hat{A}^{(k)}$ are at most δ/n in absolute value. Since $\text{off}(\hat{A}^{(k)})^2$ sums the squares of at most $n(n-1)$ off-diagonal entries, each at most δ^2/n^2 ,

$$\text{off}(\hat{A}^{(k)})^2 \leq n(n-1) \frac{\delta^2}{n^2} = \frac{n-1}{n} \delta^2 \leq \delta^2.$$

Hence $\text{off}(\hat{A}^{(k)}) \leq \delta$, as claimed. $\square$

2.3.3 Near-diagonalization via rational rotations

Unlike Proposition 1, which relies on exact Jacobi rotations and hence cannot be implemented on a Turing machine, the following theorem assembles Lemmas 6 and 8 into an algorithm that drives a symmetric matrix arbitrarily close to diagonal form via rational Jacobi rotations. Unlike Proposition 1, the output L is rational even when A is only real; when A is rational, the algorithm runs in time polynomial on a Turing machine. This result is the main algorithmic building block for the remainder of the paper.

Theorem 2 (Near-diagonalization via rational Jacobi rotations). *Let $A \in \mathbb{R}^{n \times n}$ be symmetric and let $\delta \in (0, 1]$. There is an algorithm that computes an orthogonal matrix $L \in \mathbb{Q}^{n \times n}$ such that $\text{off}(L^\top A L) \leq \delta$. The algorithm performs $O\left(n^4 (\log(1/\delta) + \log n + \log(\max(1, \|A\|_F)))^2\right)$ arithmetic operations in the real RAM model. Moreover, if $A \in \mathbb{Q}^{n \times n}$ and δ is rational, the algorithm runs on a Turing machine in time polynomial in $\text{size}(A) + \text{size}(\delta)$.*

Proof. From Lemma 8, there exists

$$k \leq \max\left(0, \left\lceil \frac{2}{3} n(n-1) \ln \left(\frac{n^2 \text{off}(A)^2}{\delta^2} \right) \right\rceil\right),$$

such that after k successive $(\delta/(2n))$ -approximate Jacobi updates, the obtained matrix $\hat{A}^{(k)}$ satisfies $\text{off}(\hat{A}^{(k)}) \leq \delta$. Let $\hat{J}_1, \dots, \hat{J}_k$ be the corresponding Jacobi rotations, each computed via Lemma 6 with tolerance $\delta/(2n)$, and define $L := \hat{J}_1 \cdots \hat{J}_k$ (with $L := I$ if $k = 0$), so that $L^\top A L = \hat{A}^{(k)}$. As a product of orthogonal matrices, L is orthogonal; moreover $L \in \mathbb{Q}^{n \times n}$, since each $\hat{J}_i$ has rational entries (Lemma 6), regardless of whether A is rational.

Running time. We first bound the entries encountered along the sequence. For each $t \leq k-1$, since $\hat{J}_1, \dots, \hat{J}_t$ are orthogonal we have $\|\hat{A}^{(t)}\|_F = \|A\|_F$; as the squares $a_{pp}^2, a_{qq}^2, a_{pq}^2$ of the entries of $\hat{A}^{(t)}$ used to construct $\hat{J}_{t+1}$ are terms of $\|\hat{A}^{(t)}\|_F^2$, each of $|a_{pp}|, |a_{qq}|, |a_{pq}|$ is at most $\|A\|_F$. Hence the quantity S of Lemma 6 at step $t+1$, namely $S_{t+1} := \max\{0, \lceil \log(4(|a_{pp}| + |a_{qq}| + |a_{pq}|)) \rceil\}$, satisfies $S_{t+1} \leq S^* := \max\{0, \lceil \log(12\|A\|_F) \rceil\} = O(1 + \log(\max(1, \|A\|_F)))$, using $|a_{pp}| + |a_{qq}| + |a_{pq}| \leq 3\|A\|_F$. Write $\sigma := \log(2n/\delta) + \log(\max(1, \|A\|_F))$. By Lemma 8 and $\text{off}(A) \leq \|A\|_F$,

$$k = O(n^2 \sigma).$$

Note that $S^* = O(\sigma)$. At each step t , finding the pivot takes $O(n^2)$ comparisons, constructing $\hat{J}_{t+1}$ takes $O(\log(2n/\delta) + S_{t+1}) = O(\sigma)$ arithmetic operations (Lemma 6 with tolerance $\delta/(2n)$), and forming $\hat{A}^{(t+1)} = \hat{J}_{t+1}^\top \hat{A}^{(t)} \hat{J}_{t+1}$ takes $O(n^2)$ more, since only rows and columns p, q change, for $O(n^2 + \sigma)$ per step. Since $\sigma \geq 1$, the total is

$$O(k(n^2 + \sigma)) = O(n^2 \sigma(n^2 + \sigma)) = O(n^4 \sigma^2);$$

substituting the definition of σ gives the claimed bound

$$O\left(n^4 (\log(1/\delta) + \log n + \log(\max(1, \|A\|_F)))^2\right)$$

and establishes the real RAM bound.

Now suppose $A \in \mathbb{Q}^{n \times n}$ and δ is rational. Let r_0 be the product of the denominators of the entries of A and $\tilde{A}_0 := r_0 A \in \mathbb{Z}^{n \times n}$, so $\text{size}(r_0), \max_{i,j} \text{size}((\tilde{A}_0)_{ij}) = O(\text{size}(A))$. By Lemma 6 (with tolerance $\delta/(2n)$), we may write $\hat{J}_t = \tilde{J}_t / Q_t$ with $Q_t \in \mathbb{Z}_{>0}$, $\text{size}(Q_t) = O(\sigma)$, and $\tilde{J}_t \in \mathbb{Z}^{n \times n}$ satisfying $|(\tilde{J}_t)_{ij}| \leq Q_t$. Setting $\tilde{A}_t := \tilde{J}_t^\top \tilde{A}_{t-1} \tilde{J}_t \in \mathbb{Z}^{n \times n}$ and $r_t := Q_t^2 r_{t-1}$, we have $\hat{A}^{(t)} = \tilde{A}_t / r_t$ for every t . Since $\|A\|_F = 2^{O(\text{size}(A))}$, $n^2 = O(\text{size}(A))$, and $\log(2n/\delta) = O(\text{size}(A) + \text{size}(\delta))$, both σ and $k = O(n^2 \sigma)$ are polynomial in $\text{size}(A) + \text{size}(\delta)$. Since the size of a product of positive integers is at most the sum of their sizes up to an additive $O(1)$, and $\tilde{A}_t$ is obtained from $\tilde{A}_{t-1}$ by $O(1)$ integer additions and multiplications involving entries of $\tilde{J}_t$ (each of size $O(\sigma)$), induction on $t \leq k$ gives

$$\text{size}(r_t), \max_{i,j} \text{size}((\tilde{A}_t)_{ij}) = O(\text{size}(A) + k\sigma),$$

which is polynomial in $\text{size}(A) + \text{size}(\delta)$. Hence every entry of $\hat{A}^{(t)} = \tilde{A}_t/r_t$ has size polynomial in $\text{size}(A) + \text{size}(\delta)$, so each of the k steps (finding the pivot, constructing $\hat{J}_{t+1}$, and forming $\tilde{A}_{t+1}$ and r_{t+1}) runs on a Turing machine in time polynomial in $\text{size}(A) + \text{size}(\delta)$. As k is also polynomial in $\text{size}(A) + \text{size}(\delta)$, the entire algorithm runs on a Turing machine in time polynomial in $\text{size}(A) + \text{size}(\delta)$. $\square$

3 Polynomial-time diagonalization

The goal of this section is to compute, in polynomial time, an orthogonal matrix that diagonalizes a symmetric rational matrix up to a prescribed precision, while preserving its inertia (Theorem 3); we then extend this to the simultaneous diagonalization of a symmetric matrix and a positive definite matrix (Theorem 4), and to simultaneous diagonalization with respect to a rational polytope (Theorem 5). The main obstacle is that a symmetric matrix may have eigenvalues arbitrarily close to 0, which could cause a small perturbation to change the inertia; we first show (Lemmas 9–13) that this cannot happen for rational matrices, since their nonzero eigenvalues are bounded away from 0 by an amount that depends only on the size of the matrix.

3.1 Eigenvalue gaps

Consider an $n \times n$ matrix A . The *characteristic polynomial* of A , denoted by $p_A(t)$, is the polynomial defined by

$$p_A(t) = \det(tI - A).$$

It is well known that the roots of $p_A(t)$ are exactly the eigenvalues of A , and that, if A is symmetric, all eigenvalues of A are real. Throughout the paper, for a symmetric matrix $A \in \mathbb{R}^{n \times n}$, we denote by $\lambda_k(A)$ its k -th largest eigenvalue, so that $\lambda_1(A) \geq \lambda_2(A) \geq \dots \geq \lambda_n(A)$ (counted with multiplicity); when the matrix is clear from the context, we simply write $\lambda_1, \dots, \lambda_n$. In particular, if A is symmetric,

$$p_A(t) = (t - \lambda_1)(t - \lambda_2) \cdots (t - \lambda_n).$$

For a polynomial $p(t)$, we denote by $\|p\|$ the maximum absolute value of the coefficients of p . We first bound the coefficients of the characteristic polynomial of a symmetric matrix in terms of n and its Frobenius norm.

Lemma 9. *Let $A \in \mathbb{R}^{n \times n}$ be symmetric and let $p_A(t)$ be its characteristic polynomial. Then,*

$$\|p_A\| \leq 2^n \max(1, \|A\|_F)^n.$$

Furthermore, if $A \in \mathbb{Z}^{n \times n}$, then the coefficients of p_A are integer.

Proof. It follows directly from the definition of the characteristic polynomial that, if $A \in \mathbb{Z}^{n \times n}$, then the coefficients of p_A are integer. Let $\lambda_1, \dots, \lambda_n \in \mathbb{R}$ be the eigenvalues of A . As noted above,

$$p_A(t) = t^n + a_{n-1}t^{n-1} + \dots + a_1t + a_0 = (t - \lambda_1)(t - \lambda_2) \cdots (t - \lambda_n).$$

For $i = 0, \dots, n-1$, the number a_i is then

$$a_i = \sum_{\mathcal{I} \subseteq \{1, \dots, n\}: |\mathcal{I}|=i} \prod_{j \notin \mathcal{I}} (-\lambda_j).$$

Since $|\lambda_j| \leq \|A\|_F \leq \max(1, \|A\|_F)$ for every j (as $\|A\|_F^2 = \sum_i \lambda_i^2$, so each λ_j^2 is one term in this sum), each of the $\binom{n}{i}$ terms in the sum satisfies

$$|\prod_{j \notin \mathcal{I}} (-\lambda_j)| = \prod_{j \notin \mathcal{I}} |\lambda_j| \leq \max(1, \|A\|_F)^{n-i} \leq \max(1, \|A\|_F)^n,$$

where the last inequality holds since $\max(1, \|A\|_F) \geq 1$ and $n - i \leq n$. Hence,

$$|a_i| \leq \binom{n}{i} \max(1, \|A\|_F)^n \leq 2^n \max(1, \|A\|_F)^n.$$

The leading coefficient of p_A is 1, which also satisfies this bound. $\square$

Using the bound in Lemma 9, we now show that every nonzero eigenvalue of an integer symmetric matrix is bounded away from 0, by an amount that depends only on n and the Frobenius norm of the matrix. We will be using the following lemma, due to Cauchy [2], and taken from Mignotte [13].

Lemma 10 ([2, 13]). *Let $q(t) \in \mathbb{R}[t]$ be the nonzero polynomial*

$$q(t) = b_d t^d + \cdots + b_1 t + b_0.$$

Let λ be a complex root of q . Then

$$|\lambda| \geq \frac{|b_0|}{|b_0| + \max(|b_1|, \dots, |b_d|)}.$$

Lemma 11. *Let $A \in \mathbb{Z}^{n \times n}$ be symmetric. Then, for every nonzero eigenvalue λ of A , we have*

$$|\lambda| \geq \frac{1}{1 + 2^n \max(1, \|A\|_F)^n}.$$

Proof. Let the characteristic polynomial of A be

$$p_A(t) = t^n + a_{n-1}t^{n-1} + \cdots + a_1t + a_0, \quad a_n := 1.$$

If $A = 0$, the result is trivial (there is no nonzero eigenvalue), so we assume $A \neq 0$.

Let $k \in \{0, 1, \dots, n-1\}$ be the multiplicity of 0 as a root of $p_A(t)$, equivalently, as an eigenvalue of A (we have $k \leq n-1$ since A is symmetric and $A \neq 0$, so not all eigenvalues of A are zero). Write $p_A(t) = t^k q(t)$, where

$$q(t) = t^{n-k} + b_{n-k-1}t^{n-k-1} + \cdots + b_1t + b_0, \quad b_i := a_{i+k} \text{ for } i = 0, \dots, n-k.$$

Since the eigenvalues of A are the roots of $p_A(t) = t^k q(t)$, with exactly k of them equal to 0, the roots of $q(t)$ are exactly the nonzero eigenvalues of A ; in particular, $q(0) = b_0 \neq 0$. Since $A \in \mathbb{Z}^{n \times n}$, the coefficients of p_A are integer by Lemma 9; in particular a_k (hence b_0) is a nonzero integer, so $|b_0| \geq 1$. From Lemma 10 (applied to $q(t)$), we have that each root λ of $q(t)$ satisfies

$$|\lambda| \geq \frac{|b_0|}{|b_0| + \max\{|b_1|, \dots, |b_{n-k}|\}}.$$

Bounding each $|b_i|$ through Lemma 9, and using $|b_0| \geq 1$, we obtain

$$\begin{aligned} |\lambda| &\geq \frac{|b_0|}{|b_0| + \max\{|b_1|, \dots, |b_{n-k}|\}} \geq \frac{|b_0|}{|b_0| + 2^n \max(1, \|A\|_F)^n} \\ &\geq \frac{1}{1 + 2^n \max(1, \|A\|_F)^n}, \end{aligned}$$

where the second inequality uses $b_i = a_{i+k}$ for $i = 1, \dots, n-k$ together with Lemma 9. Since every nonzero eigenvalue λ of A is a root of $q(t)$, this proves the claim. $\square$

3.2 Rounding near-diagonal matrices

The next result shows that, given an integer symmetric matrix A and an orthogonal L such that $L^\top AL$ is already nearly diagonal, we can safely round to zero its off-diagonal entries and its small diagonal entries, without changing the inertia of A or moving its eigenvalues by more than a controlled amount; the proof relies on the eigenvalue gap established in Lemma 11. We will also be using the following classical eigenvalue perturbation result.

Lemma 12 (Corollary 8.1.6 in [11]). *If $B, C \in \mathbb{R}^{n \times n}$ are symmetric, then, for $i = 1, \dots, n$,*

$$|\lambda_i(B) - \lambda_i(C)| \leq \|B - C\|_2.$$

Lemma 13. *Let $A \in \mathbb{Z}^{n \times n}$ be symmetric, let $L \in \mathbb{R}^{n \times n}$ be orthogonal, and let $\delta \geq 0$ be such that $\text{off}(L^\top AL) \leq \delta$ and*

$$\delta \leq \frac{1}{4(1 + 2^n \max(1, \|A\|_F)^n)}.$$

Let $D \in \mathbb{R}^{n \times n}$ be the diagonal matrix obtained from $L^\top AL$ by replacing all off-diagonal entries with zero and all diagonal entries with absolute value at most δ with zero. Then, A and D have the same inertia. Furthermore, for $i = 1, \dots, n$, we have $|\lambda_i(A) - \lambda_i(D)| \leq \delta$.

Proof. Since L is orthogonal, $L^\top = L^{-1}$; in particular L is nonsingular, so the eigenvalues of A coincide with those of $L^{-1}AL = L^\top AL$. From Lemma 11, every nonzero eigenvalue λ of A satisfies

$$|\lambda| \geq \frac{1}{1 + 2^n \max(1, \|A\|_F)^n} = 4 \cdot \left(\frac{1}{4(1 + 2^n \max(1, \|A\|_F)^n)} \right) \geq 4\delta.$$

Let $D_0 \in \mathbb{R}^{n \times n}$ be the diagonal matrix obtained from $L^\top AL$ by replacing all off-diagonal entries with zero. Note that D_0 and $L^\top AL$ are both symmetric, since D_0 is diagonal, and A is symmetric. Since $L^\top AL - D_0$ has zero diagonal and off-diagonal entries matching those of $L^\top AL$, we have $\|L^\top AL - D_0\|_F = \text{off}(L^\top AL) \leq \delta$. Hence, from Lemma 12 (with $B := L^\top AL$ and $C := D_0$), we have, for every $i = 1, \dots, n$,

$$|\lambda_i(A) - \lambda_i(D_0)| = |\lambda_i(L^\top AL) - \lambda_i(D_0)| \leq \|L^\top AL - D_0\|_2 \leq \|L^\top AL - D_0\|_F \leq \delta.$$

Thus: $\lambda_i(A) > 0$ implies $\lambda_i(A) \geq 4\delta$ and $\lambda_i(D_0) \geq 3\delta$, $\lambda_i(A) < 0$ implies $\lambda_i(A) \leq -4\delta$ and $\lambda_i(D_0) \leq -3\delta$, and $\lambda_i(A) = 0$ implies $|\lambda_i(D_0)| \leq \delta$. Let $D \in \mathbb{R}^{n \times n}$ be the diagonal matrix obtained from D_0 by replacing all diagonal entries with absolute value at most δ with zero. Since D_0 is diagonal, its eigenvalues are its diagonal entries, and by the above each of them has absolute value either at least 3δ or at most δ . Replacing those of the second kind by 0 therefore does not change the relative order of the diagonal entries, so $\lambda_i(D)$ is obtained from $\lambda_i(D_0)$ by the same replacement: $|\lambda_i(D_0)| \geq 3\delta$ implies $\lambda_i(D) = \lambda_i(D_0)$, and $|\lambda_i(D_0)| \leq \delta$ implies $\lambda_i(D) = 0$. Therefore, for every i , $\lambda_i(A)$ and $\lambda_i(D)$ are both positive, both zero, or both negative; since these cases are exhaustive, A and D have the same inertia. Finally, if $\lambda_i(D) = \lambda_i(D_0)$, then $|\lambda_i(A) - \lambda_i(D)| = |\lambda_i(A) - \lambda_i(D_0)| \leq \delta$; if $\lambda_i(D) = 0$, then $|\lambda_i(D_0)| \leq \delta$ implies $\lambda_i(A) = 0$, so $|\lambda_i(A) - \lambda_i(D)| = |0 - 0| = 0 \leq \delta$. In either case, $|\lambda_i(A) - \lambda_i(D)| \leq \delta$, as claimed. $\square$

3.3 Diagonalization

We now combine Theorem 2 with Lemmas 11 and 13 to obtain our main diagonalization result: given a symmetric rational matrix A and a tolerance δ , we compute in polynomial time a rational orthogonal L and a rational diagonal D with the same inertia as A , such that $L^\top AL$ differs from D by a symmetric matrix of Frobenius norm at most δ .

Theorem 3 (Diagonalization). *Let $A \in \mathbb{Q}^{n \times n}$ be symmetric and let $\delta \in (0, 1]$ be rational. There is an algorithm that computes an orthogonal matrix $L \in \mathbb{Q}^{n \times n}$, a diagonal matrix $D \in \mathbb{Q}^{n \times n}$, and a symmetric matrix $E \in \mathbb{Q}^{n \times n}$ such that (i) $L^\top AL = D + E$ with $\|E\|_F \leq \delta$, and (ii) A and D have the same inertia. The algorithm runs on a Turing machine in time polynomial in $\text{size}(A) + \text{size}(\delta)$.*

Furthermore, there exists $\zeta \in \mathbb{Z}_{>0}$, depending only on A (not on δ), with $\log(\zeta)$ polynomial in $\text{size}(A)$, such that every nonzero entry of D is at least $1/\zeta$ in absolute value; moreover, ζ can be computed on a Turing machine in time polynomial in $\text{size}(A)$.

Proof. Let $r \in \mathbb{Z}_{>0}$ be the product of the denominators of the entries in A and define $\tilde{A} := rA \in \mathbb{Z}^{n \times n}$. The size of each entry of $\tilde{A}$ is $O(\text{size}(A))$, thus $\text{size}(\tilde{A}) = O(n^2 \text{size}(A))$. By Lemma 11, every nonzero eigenvalue of $\tilde{A}$ has absolute value at least

$$\beta(\tilde{A}) := \frac{1}{1 + 2^n \max(1, \|\tilde{A}\|_F)^n} \geq \frac{1}{\omega}, \quad \omega := 1 + 2^n \max(1, \|\tilde{A}\|_F^2)^n,$$

where the inequality holds since $\max(1, \|\tilde{A}\|_F) \leq \max(1, \|\tilde{A}\|_F^2)$, and $\omega \in \mathbb{Z}_{>0}$ because $\|\tilde{A}\|_F^2 = \sum_{i,j} \tilde{a}_{ij}^2$ is a nonnegative integer. Let

$$\delta' := \min \left\{ \frac{\delta}{2n}, \frac{1}{4\omega} \right\} \in (0, 1],$$

so that $2n\delta' \leq \delta$ and $\delta' \leq 1/(4\omega) \leq \beta(\tilde{A})/4$; note that δ' is rational, of size polynomial in $\text{size}(A) + \text{size}(\delta)$. We employ Theorem 2 (with $A := \tilde{A}$ and $\delta := \delta'$) and obtain an orthogonal matrix $L \in \mathbb{Q}^{n \times n}$ such that $\text{off}(L^\top \tilde{A} L) \leq \delta'$. Since $\delta' \leq \beta(\tilde{A})/4 = \frac{1}{4(1+2^n \max(1, \|\tilde{A}\|_F)^n)}$, we may apply Lemma 13 (with $A := \tilde{A}$, $\delta := \delta'$). Let $\tilde{D} \in \mathbb{Q}^{n \times n}$ be the diagonal matrix obtained from $L^\top \tilde{A} L$ by replacing all off-diagonal entries with zero, and all diagonal entries with absolute value at most δ' with zero. By that lemma, $\tilde{A}$ and $\tilde{D}$ have the same inertia and, for $i = 1, \dots, n$,

$$|\lambda_i(\tilde{A}) - \lambda_i(\tilde{D})| \leq \delta'. \quad (3)$$

Let $\tilde{E} := L^\top \tilde{A} L - \tilde{D}$. The off-diagonal entries of $\tilde{E}$ coincide with those of $L^\top \tilde{A} L$, so the sum of their squares equals $\text{off}(L^\top \tilde{A} L)^2 \leq (\delta')^2$; by the definition of $\tilde{D}$, the diagonal entries of $\tilde{E}$ have absolute value at most δ' . Thus,

$$\|\tilde{E}\|_F^2 \leq (\delta')^2 + n(\delta')^2 = (1+n)(\delta')^2 \leq 2n(\delta')^2,$$

so $\|\tilde{E}\|_F \leq \sqrt{2n} \delta' \leq 2n\delta' \leq \delta$, where we used $\sqrt{2n} \leq 2n$. Now let $D := \tilde{D}/r \in \mathbb{Q}^{n \times n}$ and $E := \tilde{E}/r \in \mathbb{Q}^{n \times n}$. Clearly D is diagonal, E is symmetric, and

$$L^\top A L = \frac{1}{r} L^\top \tilde{A} L = \frac{1}{r} (\tilde{D} + \tilde{E}) = D + E.$$

Since r is positive, A and $\tilde{A}$ have the same inertia, and so do $\tilde{D}$ and D ; since $\tilde{A}$ and $\tilde{D}$ have the same inertia, this proves (ii). Moreover, $r \geq 1$ implies

$$\|E\|_F = \frac{1}{r} \|\tilde{E}\|_F \leq \|\tilde{E}\|_F \leq \delta.$$

This proves (i). **An a priori lower bound on the nonzero diagonal entries of D .** Let $\zeta := 2r\omega \in \mathbb{Z}_{>0}$. Since $\|\tilde{A}\|_F^2$ is a nonnegative integer of size polynomial in $\text{size}(A)$ (as $\text{size}(\tilde{A}) = O(n^2 \text{size}(A))$) and $\text{size}(r)$ is polynomial in $\text{size}(A)$, $\log(\zeta)$ is polynomial in $\text{size}(A)$ and ζ is computable in time polynomial in $\text{size}(A)$; note also that ζ depends only on A , not on δ . For every i with $\lambda_i(A) \neq 0$, we have $\lambda_i(\tilde{A}) = r\lambda_i(A) \neq 0$, so $|\lambda_i(\tilde{A})| \geq \beta(\tilde{A})$ by Lemma 11. Hence, using $|\lambda_i(\tilde{A}) - \lambda_i(\tilde{D})| \leq \delta'$ (shown in (3)),

$$|\lambda_i(\tilde{D})| \geq |\lambda_i(\tilde{A})| - |\lambda_i(\tilde{A}) - \lambda_i(\tilde{D})| \geq \beta(\tilde{A}) - \delta'.$$

Since $\delta' \leq \beta(\tilde{A})/4$ (shown above), this gives

$$|\lambda_i(\tilde{D})| \geq \beta(\tilde{A}) - \frac{\beta(\tilde{A})}{4} = \frac{3}{4}\beta(\tilde{A}).$$

Dividing by $r \geq 1$ and using $\beta(\tilde{A}) \geq 1/\omega$,

$$|\lambda_i(D)| = \frac{|\lambda_i(\tilde{D})|}{r} \geq \frac{3\beta(\tilde{A})}{4r} \geq \frac{3}{4r\omega} = \frac{3}{2} \cdot \frac{1}{\zeta} \geq \frac{1}{\zeta}.$$

Thus, every nonzero eigenvalue $\lambda_i(D)$ satisfies $|\lambda_i(D)| \geq 1/\zeta$. Since D is diagonal, its diagonal entries coincide, as a multiset, with its eigenvalues $\lambda_1(D), \dots, \lambda_n(D)$; hence every nonzero diagonal entry of D satisfies the same bound, establishing the claim.

Running time. Every quantity other than L is obtained by $O(n^2)$ arithmetic operations on integers or rationals of size polynomial in $\text{size}(A) + \text{size}(\delta)$: r and $\tilde{A} = rA$ have size polynomial in $\text{size}(A)$, as do $\|\tilde{A}\|_F^2$ and ω , while $\text{size}(\delta')$ is polynomial in $\text{size}(A) + \text{size}(\delta)$. By Theorem 2 (applied to $\tilde{A}$ with $\delta := \delta'$), computing L takes time polynomial in $\text{size}(\tilde{A}) + \text{size}(\delta')$, hence in $\text{size}(A) + \text{size}(\delta)$, since $\text{size}(\tilde{A}) = O(n^2 \text{size}(A))$. Forming $\tilde{D}, \tilde{E}$ from $L^\top \tilde{A} L$ and rescaling by r to obtain D, E take a further $O(n^2)$ arithmetic operations on rationals of size polynomial in $\text{size}(A) + \text{size}(\delta)$. Hence the entire algorithm runs on a Turing machine in time polynomial in $\text{size}(A) + \text{size}(\delta)$. $\square$

Note that Theorem 3 recovers the conclusion of Theorem 2, for the same δ : since D is diagonal, the off-diagonal entries of $L^\top A L$ coincide with those of E , so $\text{off}(L^\top A L) = \text{off}(E) \leq \|E\|_F \leq \delta$.

Theorem 3 also yields an approximate spectral decomposition of A .

Corollary 1 (Approximate spectral decomposition). *Let $A \in \mathbb{Q}^{n \times n}$ be symmetric, let $\delta \in (0, 1]$ be rational, and let L, D, E be as in Theorem 3. Denote by v^i the i th column of L and by d_i the i th diagonal entry of D . Then, for every $i = 1, \dots, n$, (i) $|\lambda_i(A) - \lambda_i(D)| \leq \delta$, and (ii) $\|Av^i - d_i v^i\|_2 \leq \delta$.*

Proof. Since L is orthogonal, A and $L^\top A L = D + E$ are similar, hence $\lambda_i(A) = \lambda_i(D + E)$ for every i . By Lemma 12, applied to $D + E$ and D , we obtain

$$|\lambda_i(A) - \lambda_i(D)| = |\lambda_i(D + E) - \lambda_i(D)| \leq \|E\|_2 \leq \|E\|_F \leq \delta,$$

which proves (i). For (ii), let e^i denote the i th vector of the standard basis of $\mathbb{R}^n$, so that $v^i = L e^i$ and $L^{-1} v^i = e^i$, since L is orthogonal and thus $L^{-1} = L^\top$. From $L^\top A L = D + E$ we get $A = L(D + E)L^{-1}$, so

$$Av^i = L(D + E)L^{-1}v^i = L(D + E)e^i = d_i v^i + L E e^i,$$

and therefore $\|Av^i - d_i v^i\|_2 = \|L E e^i\|_2 = \|E e^i\|_2 \leq \|E\|_F \leq \delta$. $\square$

By (ii), each column v^i of L behaves as an approximate eigenvector of A , with corresponding approximate eigenvalue d_i , while by (i) the diagonal entries of D approximate the eigenvalues of A . Since the columns $v^1, \dots, v^n$ of the orthogonal matrix L form an orthonormal basis of $\mathbb{R}^n$, this yields a full set of n simultaneous approximate eigenvectors, each with residual controlled by the same tolerance δ . In particular, denoting by $L_\delta, D_\delta, E_\delta$ the matrices obtained for the tolerance δ , we have $\|E_\delta\|_F \rightarrow 0$ as $\delta \rightarrow 0$ by Theorem 3(i); since the set of orthogonal matrices is compact, every limit point of L_δ as $\delta \rightarrow 0$ is an exact eigenvector matrix of A , and the corresponding limit point of D_δ is the diagonal matrix of the eigenvalues of A , consistently with the classical spectral theorem for symmetric matrices.

3.4 Simultaneous diagonalization of two matrices

We now consider the simultaneous diagonalization of a symmetric matrix A and a positive definite matrix M : classically, one seeks a nonsingular matrix L such that $L^\top A L$ is diagonal and $L^\top M L = I$. As shown by Example 1, such an L is in general irrational, hence it cannot be computed on a Turing machine. The next theorem shows that, if the requirement that $L^\top A L$ be diagonal is relaxed to being diagonal up to a perturbation of Frobenius norm at most δ , then L can be taken rational and can be computed in polynomial time, while the condition $L^\top M L = I$ is still satisfied exactly and the inertia of A is still preserved. We assume that M is given in the factored form $M = C^\top C$ for a rational matrix C ; the remark following the theorem shows that this assumption cannot be dropped.

Theorem 4 (Simultaneous diagonalization of two matrices). *Let $A \in \mathbb{Q}^{n \times n}$ be symmetric, let $C \in \mathbb{Q}^{n \times n}$ be nonsingular, let $M := C^\top C$, and let $\delta \in (0, 1]$ be rational. There is an algorithm that computes a nonsingular matrix $L \in \mathbb{Q}^{n \times n}$, a diagonal matrix $D \in \mathbb{Q}^{n \times n}$, and a symmetric matrix $E \in \mathbb{Q}^{n \times n}$ such that (i) $L^\top A L = D + E$ with $\|E\|_F \leq \delta$, (ii) A and D have the same inertia, and (iii) $L^\top M L = I$. The algorithm runs on a Turing machine in time polynomial in $\text{size}(A) + \text{size}(C) + \text{size}(\delta)$.*

Furthermore, there exists $\zeta \in \mathbb{Z}_{>0}$, depending only on A and C (not on δ), with $\log(\zeta)$ polynomial in $\text{size}(A) + \text{size}(C)$, such that every nonzero entry of D is at least $1/\zeta$ in absolute value; moreover, ζ can be computed on a Turing machine in time polynomial in $\text{size}(A) + \text{size}(C)$.

Proof. Let $\tilde{A} := C^{-\top}AC^{-1}$, which is symmetric and belongs to $\mathbb{Q}^{n \times n}$, and which has the same inertia as A by Sylvester's law of inertia. Since $\tilde{A}$ is obtained from A and C by a constant number of matrix inversions and multiplications, $\text{size}(\tilde{A})$ is polynomial in $\text{size}(A) + \text{size}(C)$. We employ Theorem 3 (with $A := \tilde{A}$ and the same δ) and obtain an orthogonal matrix $\tilde{L} \in \mathbb{Q}^{n \times n}$, a diagonal matrix $D \in \mathbb{Q}^{n \times n}$, and a symmetric matrix $E \in \mathbb{Q}^{n \times n}$ such that $\tilde{L}^\top \tilde{A} \tilde{L} = D + E$, $\|E\|_F \leq \delta$, and A and D have the same inertia. Define $L := C^{-1} \tilde{L} \in \mathbb{Q}^{n \times n}$, which is nonsingular. Then $L^\top AL = \tilde{L}^\top C^{-\top} AC^{-1} \tilde{L} = \tilde{L}^\top \tilde{A} \tilde{L} = D + E$, which proves (i), and (ii) holds since A , $\tilde{A}$, and D all have the same inertia. Moreover

$$L^\top ML = \tilde{L}^\top C^{-\top} (C^\top C) C^{-1} \tilde{L} = \tilde{L}^\top \tilde{L} = I,$$

since $\tilde{L}$ is orthogonal, which proves (iii). The “Furthermore” clause follows from the corresponding clause of Theorem 3, applied with $A := \tilde{A}$, together with the fact that $\text{size}(\tilde{A})$ is polynomial in $\text{size}(A) + \text{size}(C)$.

Running time. Forming $\tilde{A}$ and L takes a constant number of matrix operations on rationals of size polynomial in $\text{size}(A) + \text{size}(C)$, and, by Theorem 3, computing $\tilde{L}, D, E$ takes time polynomial in $\text{size}(\tilde{A}) + \text{size}(\delta)$. Hence the entire algorithm runs on a Turing machine in time polynomial in $\text{size}(A) + \text{size}(C) + \text{size}(\delta)$. $\square$

Both departures of Theorem 4 from the classical statement (1) are necessary. First, by Example 1, the first requirement in (1) cannot be met by a rational L , even when $M = I$; this is why condition (i) is a relaxation. Second, the hypothesis that M be given in the factored form $M = C^\top C$, with C rational, cannot be dropped, as the next example shows.

Example 3. Let $n := 2$ and $M := 3I$, and suppose that $L^\top ML = I$ for some rational nonsingular L . Then $L^\top L = \frac{1}{3}I$, so the first column (x, y) of L satisfies $x^2 + y^2 = \frac{1}{3}$, and thus $(3x)^2 + (3y)^2 = 3$, exhibiting 3 as a sum of two rational squares. Writing $3 = (p/r)^2 + (q/r)^2$ with p, q, r integers such that $\gcd(p, q, r) = 1$, we obtain $3r^2 = p^2 + q^2$; since every square is congruent to 0 or 1 modulo 3, it follows that $p \equiv q \equiv 0 \pmod{3}$, hence 9 divides $3r^2$, so that 3 divides r , contradicting $\gcd(p, q, r) = 1$. Therefore no rational nonsingular L satisfies $L^\top ML = I$; equivalently, there is no nonsingular rational C with $M = C^\top C$, since $L := C^{-1}$ would then satisfy $L^\top ML = I$.

Example 3 exhibits a further obstruction, beyond the irrationality of Example 1, to performing the classical simultaneous reduction on a Turing machine.

3.5 Simultaneous diagonalization and rounding

We now extend Theorem 4 to the setting where, in place of the matrix M , we are given a full-dimensional rational polytope, and we wish the matrix L to transform this polytope into one sandwiched between two concentric balls (an ellipsoid rounding condition). This will later allow us to reduce a general mixed integer quadratic program to one with a separable objective function and posed over a polytope with a spherical form.

We will be using the following algorithmic result, due to Lenstra [12] (see also Corollary 15.6a in Schrijver [16]). Given a nonsingular matrix $C \in \mathbb{R}^{n \times n}$ and a vector $a \in \mathbb{R}^n$, we define the *ellipsoid*

$$\mathcal{E}(a, C) := \{x \in \mathbb{R}^n : \|C(x - a)\|_2 \leq 1\}.$$

Lemma 14 ([12], Corollary 15.6a in [16]). *There is an algorithm that, given a rational system $Wx \leq w$ such that $\mathcal{P} := \{x \in \mathbb{R}^n : Wx \leq w\}$ is full-dimensional and bounded, computes a nonsingular matrix $C \in \mathbb{Q}^{n \times n}$ and a vector $a \in \mathbb{Q}^n$ such that*

$$\mathcal{E}(a, C) \subseteq \mathcal{P} \subseteq \mathcal{E}\left(a, C/n^{3/2}\right),$$

in time polynomially bounded by $\text{size}(W) + \text{size}(w)$.

Theorem 5 (Simultaneous diagonalization and rounding). *Let $A \in \mathbb{Q}^{n \times n}$ be symmetric, let $\delta \in (0, 1]$ be rational, and let $Wx \leq w$ be a rational system such that $\mathcal{P} := \{x \in \mathbb{R}^n : Wx \leq w\}$ is full-dimensional and bounded. There is an algorithm that computes a nonsingular matrix $L \in \mathbb{Q}^{n \times n}$, a diagonal matrix $D \in \mathbb{Q}^{n \times n}$, a symmetric matrix $E \in \mathbb{Q}^{n \times n}$, and $a \in \mathbb{Q}^n$ such that (i) $L^T A L = D + E$ with $\|E\|_F \leq \delta$, (ii) A and D have the same inertia, and (iii) $\mathcal{E}(a, L^{-1}) \subseteq \mathcal{P} \subseteq \mathcal{E}(a, L^{-1}/n^{3/2})$. The algorithm runs on a Turing machine in time polynomial in $\text{size}(A) + \text{size}(W) + \text{size}(w) + \text{size}(\delta)$.*

Furthermore, there exists $\zeta \in \mathbb{Z}_{>0}$, depending only on A, W, w (not on δ), with $\log(\zeta)$ polynomial in $\text{size}(A) + \text{size}(W) + \text{size}(w)$, such that every nonzero entry of D is at least $1/\zeta$ in absolute value; moreover, ζ can be computed on a Turing machine in time polynomial in $\text{size}(A) + \text{size}(W) + \text{size}(w)$.

Proof. By Lemma 14, there is an algorithm, running on a Turing machine in time polynomial in $\text{size}(W) + \text{size}(w)$, that computes a nonsingular matrix $C \in \mathbb{Q}^{n \times n}$ and $a \in \mathbb{Q}^n$ such that

$$\mathcal{E}(a, C) \subseteq \mathcal{P} \subseteq \mathcal{E}(a, C/n^{3/2}); \quad (4)$$

in particular, $\text{size}(C)$ and $\text{size}(a)$ are polynomial in $\text{size}(W) + \text{size}(w)$. We employ Theorem 4 (with this A and C , the same δ , and $M := C^T C$) and obtain a nonsingular matrix $L \in \mathbb{Q}^{n \times n}$, a diagonal matrix $D \in \mathbb{Q}^{n \times n}$, and a symmetric matrix $E \in \mathbb{Q}^{n \times n}$ such that $L^T M L = I$, $L^T A L = D + E$ with $\|E\|_F \leq \delta$, and A and D have the same inertia. This establishes (i) and (ii).

It remains to prove (iii). From $L^T M L = I$ we obtain $M = L^{-T} L^{-1}$, and therefore, for every $x \in \mathbb{R}^n$,

$$\|L^{-1}(x - a)\|_2^2 = (x - a)^T M (x - a) = \|C(x - a)\|_2^2,$$

from which we derive $\mathcal{E}(a, L^{-1}) = \mathcal{E}(a, C)$ and $\mathcal{E}(a, L^{-1}/n^{3/2}) = \mathcal{E}(a, C/n^{3/2})$. Condition (iii) then follows from (4).

An a priori lower bound on the nonzero diagonal entries of D . By the “Furthermore” clause of Theorem 4, there exists $\zeta \in \mathbb{Z}_{>0}$, depending only on A and C (not on δ), with $\log(\zeta)$ polynomial in $\text{size}(A) + \text{size}(C)$, such that every nonzero entry of D is at least $1/\zeta$ in absolute value, and ζ can be computed on a Turing machine in time polynomial in $\text{size}(A) + \text{size}(C)$. Since $\text{size}(C)$ is polynomial in $\text{size}(W) + \text{size}(w)$, both $\log(\zeta)$ and the time to compute ζ are polynomial in $\text{size}(A) + \text{size}(W) + \text{size}(w)$; note also that ζ depends only on A, W, w (via C), not on δ .

Running time. Computing C and a takes time polynomial in $\text{size}(W) + \text{size}(w)$ (Lemma 14), and forming $M = C^T C$ takes a constant number of matrix operations. By Theorem 4, computing L, D, E takes time polynomial in $\text{size}(A) + \text{size}(C) + \text{size}(\delta)$, hence in $\text{size}(A) + \text{size}(W) + \text{size}(w) + \text{size}(\delta)$. Hence the entire algorithm runs on a Turing machine in time polynomial in $\text{size}(A) + \text{size}(W) + \text{size}(w) + \text{size}(\delta)$. $\square$

4 Approximating mixed integer quadratic programming

In this section we prove Theorem 1. We first introduce *spherical form* MIQP, a special form of Problem MIQP in which the objective function is separable and the polyhedron described by the linear constraints is sandwiched between two concentric balls. Theorem 5 allows us to reduce Problem MIQP to this form, and we give an approximation algorithm for spherical form MIQP, based on mesh partitioning and linear underestimators. Both the reduction and the algorithm are subject to the same proviso: the objective function must vary enough on the feasible region. Two theorems of the alternative then remove this proviso: applied to an arbitrary instance, they either produce an approximate solution, certifying in the process that the objective function does vary enough, or split the instance into sub-instances with one fewer integer variable. A recursion on these two outcomes yields Theorem 1.

4.1 Spherical form MIQP

To define a spherical form MIQP, we first introduce mixed integer lattices and balls. Given linearly independent vectors $b^1, \dots, b^n \in \mathbb{R}^n$ and an integer $p \in \{0, \dots, n\}$, we define the *mixed integer lattice*

$$\Lambda_p(b^1, \dots, b^n) := \left\{ \sum_{i=1}^n \mu_i b^i : \mu_i \in \mathbb{Z} \ \forall i = 1, \dots, p, \ \mu_i \in \mathbb{R} \ \forall i = p+1, \dots, n \right\}.$$

A *ball* with center a and radius r is a set of the form

$$\mathcal{B}(a, r) := \{y \in \mathbb{R}^n : \|y - a\|_2 \leq r\},$$

where $a \in \mathbb{R}^n$ and $r > 0$. Note that $\mathcal{B}(a, r) = \mathcal{E}(a, I/r)$.

A *spherical form MIQP* is an optimization problem of the form

$$\begin{aligned} \min \quad & y^\top D y + h^\top y \\ \text{s. t.} \quad & W y \leq w \\ & y \in \Lambda_p(b^1, \dots, b^n) + \{c\}, \end{aligned} \tag{S-MIQP}$$

where D is diagonal and the polyhedron $\{y \in \mathbb{R}^n : W y \leq w\}$ has a *spherical form*:

$$\mathcal{B}(0, 1) \subseteq \{y \in \mathbb{R}^n : W y \leq w\} \subseteq \mathcal{B}(0, n^{3/2}).$$

In Problem S-MIQP, $y \in \mathbb{R}^n$ is the vector of variables and the input data consists of a diagonal matrix $D \in \mathbb{Q}^{n \times n}$, vectors $h, c \in \mathbb{Q}^n$, a matrix $W \in \mathbb{Q}^{m \times n}$, a vector $w \in \mathbb{Q}^m$, an integer $p \in \{0, 1, \dots, n\}$, and linearly independent vectors $b^1, \dots, b^n \in \mathbb{Q}^n$. Given an instance of Problem S-MIQP (i.e., specific data $D, h, W, w, b^1, \dots, b^n, c$), we define its *size* as $\text{size}(D) + \text{size}(h) + \text{size}(W) + \text{size}(w) + \sum_{i=1}^n \text{size}(b^i) + \text{size}(c)$. Note that the feasible region of Problem S-MIQP is compact: it is contained in $\mathcal{B}(0, n^{3/2})$, and it is closed because $\Lambda_p(b^1, \dots, b^n) + \{c\}$ is closed. In particular, if it is nonempty, the objective function is bounded on it and attains its minimum and its maximum. The two key properties of a spherical form MIQP are thus that its objective function is separable, and that its linear inequalities describe a polytope with a spherical form. The price paid for these properties is that the constraint $x \in \mathbb{Z}^p \times \mathbb{R}^{n-p}$ of Problem MIQP is replaced by the less elegant constraint $y \in \Lambda_p(b^1, \dots, b^n) + \{c\}$. We remark that, due to the different type of lattice constraint, Problem S-MIQP is technically not of the form (MIQP), however it is simple to check that the change of variables $y = Bx + c$, where B is the matrix with columns $b^1, \dots, b^n$, maps Problem S-MIQP into a problem of the form (MIQP).

Our definition differs substantially from the spherical form MIQP considered in [7], whose shape is dictated by the fixed rank assumption made there: the variables are split into a first block, of fixed dimension, which carries the quadratic part of the objective function and all the integrality constraints, and a second block, on which the objective function is linear and the variables are continuous; the sphericity condition is then imposed not on the polytope itself, but on its orthogonal projection onto the space of the first block, with a constant ratio between the radii of the two balls. Problem S-MIQP requires no such splitting: the objective function is separable in all n variables, and the entire polyhedron has a spherical form. The price is that the ratio between the two radii is $n^{3/2}$ rather than a constant. In this respect our spherical form is closer to the one used by Vavasis in the continuous setting [17]; the difference is that here the feasible region also carries a lattice constraint.

4.2 Reduction to spherical form

We need a lemma that quantifies how sensitive approximate solutions are to perturbations of the objective function.

Lemma 15 (Approximation of perturbed optimization problems). *Let $\mathcal{S}$ be a nonempty set and let $f, g : \mathcal{S} \rightarrow \mathbb{R}$ be such that $f_{\inf} := \inf_{y \in \mathcal{S}} f(y)$ and $f_{\sup} := \sup_{y \in \mathcal{S}} f(y)$ are finite, and define*

$$\min \{f(y) : y \in \mathcal{S}\}, \tag{5}$$

$$\min \{f(y) + g(y) : y \in \mathcal{S}\}. \tag{6}$$

Let $y^\diamond$ be an $\epsilon/2$ -approximate solution to Problem (5), for $\epsilon \in [0, 1]$, and assume $|g(y)| \leq \epsilon(f_{\sup} - f_{\inf})/8$ for every $y \in \mathcal{S}$. Then $y^\diamond$ is an ϵ -approximate solution to Problem (6).

Proof. Let $\hat{f}(y) := f(y) + g(y)$ and $\Delta := \epsilon(f_{\sup} - f_{\inf})/8$. Let $\hat{f}_{\inf} := \inf_{y \in \mathcal{S}} \hat{f}(y)$ and $\hat{f}_{\sup} := \sup_{y \in \mathcal{S}} \hat{f}(y)$, which are both finite since $|g(y)| \leq \Delta$ for every $y \in \mathcal{S}$. First, note that

$$\begin{aligned}\hat{f}_{\sup} - \hat{f}_{\inf} &\geq f_{\sup} - f_{\inf} - 2\Delta \\ &= (1 - \epsilon/4)(f_{\sup} - f_{\inf}),\end{aligned}$$

where we used $\hat{f}_{\sup} \geq f_{\sup} - \Delta$ and $\hat{f}_{\inf} \leq f_{\inf} + \Delta$. We obtain

$$\begin{aligned}\hat{f}(y^\diamond) - \hat{f}_{\inf} &\leq f(y^\diamond) - f_{\inf} + 2\Delta \\ &\leq \epsilon(f_{\sup} - f_{\inf})/2 + \epsilon(f_{\sup} - f_{\inf})/4 \\ &= (1/2 + 1/4)\epsilon(f_{\sup} - f_{\inf}) \\ &\leq \frac{1/2 + 1/4}{1 - \epsilon/4} \epsilon(\hat{f}_{\sup} - \hat{f}_{\inf}) \\ &\leq \epsilon(\hat{f}_{\sup} - \hat{f}_{\inf}),\end{aligned}$$

where the first inequality uses $\hat{f}(y^\diamond) \leq f(y^\diamond) + \Delta$ and $\hat{f}_{\inf} \geq f_{\inf} - \Delta$, the second uses that $y^\diamond$ is an $\epsilon/2$ -approximate solution to Problem (5) together with $2\Delta = \epsilon(f_{\sup} - f_{\inf})/4$, the fourth uses the inequality above, and the last holds for $\epsilon \in [0, 1]$. Thus, $y^\diamond$ is an ϵ -approximate solution to Problem (6). $\square$

We are now ready to show that Problem MIQP can be reduced to a spherical form MIQP. Unlike in the corresponding reduction in [7], the obtained spherical form MIQP is not equivalent to the original problem; instead, approximate solutions to the spherical form MIQP yield approximate solutions to the original problem, provided that the objective function varies enough over the feasible region.

Proposition 2 (Reduction to spherical form MIQP). *Consider an instance of Problem MIQP such that $\mathcal{P} := \{x \in \mathbb{R}^n : Wx \leq w\}$ is full-dimensional and bounded and H is nonzero, let $\epsilon \in (0, 1]$ be rational, and let κ be a positive rational absolute constant. There is an algorithm that computes a nonsingular matrix $L \in \mathbb{Q}^{n \times n}$, a diagonal matrix $D \in \mathbb{Q}^{n \times n}$, and a vector $a \in \mathbb{Q}^n$ such that: (i) H and D have the same inertia. (ii) The following problem is a spherical form MIQP, where $b^1, \dots, b^n$ denote the columns of L^{-1} :*

$$\begin{aligned}\min \quad & y^\top D y + (h^\top L + 2a^\top H L) y \\ \text{s. t.} \quad & W L y \leq w - W a \\ & y \in \Lambda_p(b^1, \dots, b^n) - \{L^{-1}a\}.\end{aligned}\tag{7}$$

(iii) Let f denote the objective function of Problem (7), let $f_{\inf}$ and $f_{\sup}$ be its infimum and supremum on the feasible region of that problem, and let $\gamma_{\min}$ be the smallest absolute value among the nonzero entries of D , which exists since H is nonzero and by (i). If $y^\diamond$ is an $\epsilon/2$ -approximate solution to Problem (7) and $f_{\sup} - f_{\inf} \geq \kappa \gamma_{\min}$, then $x^\diamond := L y^\diamond + a$ is an ϵ -approximate solution to Problem MIQP. The algorithm runs on a Turing machine in time polynomial in the size of the instance of Problem MIQP and in $\text{size}(\epsilon)$.

Proof. By the ‘‘Furthermore’’ clause of Theorem 5 (applied with $A := H$, the given W, w , and any rational $\delta \in (0, 1]$), there exists $\zeta \in \mathbb{Z}_{>0}$, depending only on H, W, w (not on δ), with $\log(\zeta)$ polynomial in the size of the instance, such that, for every choice of δ , every nonzero entry of the resulting diagonal matrix D is at least $1/\zeta$ in absolute value; moreover, ζ can be computed on a Turing machine in time polynomial in $\text{size}(H) + \text{size}(W) + \text{size}(w)$.

We set

$$\delta := \min \left\{ 1, \frac{\epsilon \kappa}{8n^3 \zeta} \right\} \in (0, 1],$$

and employ Theorem 5 with $A := H$, the given W, w , and this δ , obtaining a nonsingular matrix $L \in \mathbb{Q}^{n \times n}$, a diagonal matrix $D \in \mathbb{Q}^{n \times n}$, a symmetric matrix $E \in \mathbb{Q}^{n \times n}$, and $a \in \mathbb{Q}^n$ such that $L^\top H L = D + E$, H and D have the same inertia, $\|E\|_F \leq \delta$, and $\mathcal{E}(a, L^{-1}) \subseteq \mathcal{P} \subseteq \mathcal{E}(a, L^{-1}/n^{3/2})$. Since H is nonzero

and symmetric, it has a nonzero eigenvalue; since H and D have the same inertia, D also has a nonzero diagonal entry, so $\gamma_{\min}$ is well-defined, and, by the choice of ζ above, $\gamma_{\min} \geq 1/\zeta$.

With the change of variables $y = L^{-1}(x - a)$, we obtain

$$\mathcal{B}(0, 1) \subseteq \{y \in \mathbb{R}^n : WLy \leq w - Wa\} \subseteq \mathcal{B}(0, n^{3/2}).$$

Since the objective function of Problem (7) is separable (as D is diagonal) and the polyhedron described by its linear constraints has this spherical form, Problem (7) is a spherical form MIQP, establishing (ii). At this point, we have proven (i) and (ii) and have obtained all matrices and vectors that appear as data in Problem (7). The remainder of the proof is devoted to proving (iii).

Let $y^\diamond$ be an $\epsilon/2$ -approximate solution to Problem (7). Our goal is to show that $x^\diamond := Ly^\diamond + a$ is an ϵ -approximate solution to Problem MIQP. Since the definition of ϵ -approximate solution is preserved under changes of variables and translations of the objective function, it suffices to show that $y^\diamond$ is an ϵ -approximate solution to the following problem, obtained from Problem MIQP by first performing the change of variables $y = L^{-1}(x - a)$, and then by dropping the constant $a^\top Ha + h^\top a$ in the objective function:

$$\begin{aligned} \min \quad & y^\top (D + E)y + (h^\top L + 2a^\top HL)y \\ \text{s. t.} \quad & WLy \leq w - Wa \\ & y \in \Lambda_p(b^1, \dots, b^n) - \{L^{-1}a\}. \end{aligned} \tag{8}$$

To obtain the last constraint, note that $Ly + a \in \mathbb{Z}^p \times \mathbb{R}^{n-p}$ if and only if $y \in L^{-1}(\mathbb{Z}^p \times \mathbb{R}^{n-p}) - \{L^{-1}a\}$ and

$$\begin{aligned} L^{-1}(\mathbb{Z}^p \times \mathbb{R}^{n-p}) &= L^{-1} \left\{ \sum_{i=1}^n \mu_i e^i : \mu_i \in \mathbb{Z} \ \forall i = 1, \dots, p, \ \mu_i \in \mathbb{R} \ \forall i = p+1, \dots, n \right\} \\ &= \left\{ \sum_{i=1}^n \mu_i L^{-1} e^i : \mu_i \in \mathbb{Z} \ \forall i = 1, \dots, p, \ \mu_i \in \mathbb{R} \ \forall i = p+1, \dots, n \right\} \\ &= \left\{ \sum_{i=1}^n \mu_i b^i : \mu_i \in \mathbb{Z} \ \forall i = 1, \dots, p, \ \mu_i \in \mathbb{R} \ \forall i = p+1, \dots, n \right\} \\ &= \Lambda_p(b^1, \dots, b^n). \end{aligned}$$

The only difference between Problem (7) and Problem (8) is the presence of the quadratic function $y^\top Ey$ in the objective function of Problem (8). We now bound the absolute value of this quadratic function, for all $y \in \mathcal{B}(0, n^{3/2})$, as follows:

$$\begin{aligned} |y^\top Ey| &\leq \|y\|_2^2 \|E\|_2 \leq n^3 \|E\|_F \leq n^3 \delta \\ &\leq \epsilon \kappa / (8\zeta) \leq \epsilon \kappa \gamma_{\min} / 8 \leq \epsilon (f_{\sup} - f_{\inf}) / 8, \end{aligned}$$

where the third inequality holds by definition of δ (it is an equality if $\delta = \epsilon \kappa / (8n^3 \zeta)$, and otherwise $\delta = 1 \leq \epsilon \kappa / (8n^3 \zeta)$), and the second-to-last inequality uses $\gamma_{\min} \geq 1/\zeta$, established above.

The feasible region of Problem (7) is nonempty, since it contains $y^\diamond$, and compact, since Problem (7) is a spherical form MIQP; hence $f_{\inf}$ and $f_{\sup}$ are finite. We now apply Lemma 15 with (5) := (7) and (6) := (8). We obtain that $y^\diamond$ is an ϵ -approximate solution to Problem (8), which, as observed above, implies that $x^\diamond$ is an ϵ -approximate solution to Problem MIQP. This proves (iii).

Running time. As noted above, ζ is computable in time polynomial in the size of the instance, and $\log(\zeta)$ is polynomial in the size of the instance; hence $\delta = \min\{1, \epsilon \kappa / (8n^3 \zeta)\}$ is computable in polynomial time and is a rational of size polynomial in the size of the instance and in $\text{size}(\epsilon)$ (as κ is a fixed constant). By Theorem 5, computing L, D, E, a takes time polynomial in the size of the instance and in $\text{size}(\delta)$, hence in the size of the instance and in $\text{size}(\epsilon)$; computing the columns $b^1, \dots, b^n$ of L^{-1} takes a constant number of further matrix operations. Hence the entire algorithm runs on a Turing machine in time polynomial in the size of the instance and in $\text{size}(\epsilon)$. $\square$

4.3 Approximation algorithm for Problem S-MIQP

In this section we show that we can find an ϵ -approximate solution to Problem S-MIQP, provided that the objective function varies enough on the feasible region. The algorithm and its analysis are not new: they follow closely a well-established line of work on mesh partitioning and linear underestimators, and we present them here in the precise form needed later. The technique was applied in the continuous case by Vavasis [18, 17], and in the mixed integer case in [5, 6, 7]; our treatment borrows from both, with only minor modifications. As in the mixed integer papers, the feasible region carries a lattice constraint, which is absent in the continuous setting. As in [17], we linearize the objective function only in the concave directions, leaving the remaining, already convex, quadratic terms unchanged; as a result, the subproblems are mixed integer convex quadratic programs, rather than the mixed integer linear programs obtained in [5, 6, 7], and we solve them using the polynomial-time algorithm established in [8].

Formally, a *mixed integer convex quadratic program (MICQP)* is a problem of the form (MIQP) whose objective function is convex, or, equivalently, whose matrix H is positive semidefinite. We will make use of the following consequence of [8].

Lemma 16 ([8]). *Assume that, in Problem MIQP, the matrix H is positive semidefinite and $\{x \in \mathbb{R}^n : Wx \leq w\}$ is bounded. There is an algorithm that decides whether Problem MIQP is infeasible, or has an optimal solution, and in the latter case finds an optimal solution. The algorithm runs on a Turing machine in time polynomial in the size of the instance, provided that the number p of integer variables is fixed.*

We will also use the following standard lemma, whose short proof we include for completeness.

Lemma 17. *Let $q(\lambda) = a\lambda^2 + b\lambda + c$ be a univariate quadratic function with $a \leq 0$, and let $\ell, u \in \mathbb{R}$ with $\ell \leq u$. Let $\tilde{q}(\lambda)$ be the affine univariate function that attains the same values as q at ℓ, u . Then, for every $\lambda \in [\ell, u]$,*

$$\tilde{q}(\lambda) \leq q(\lambda) \leq \tilde{q}(\lambda) + |a|(u - \ell)^2/4.$$

Proof. The function $q - \tilde{q}$ is a quadratic with leading coefficient a that vanishes at ℓ and at u , hence $q(\lambda) - \tilde{q}(\lambda) = a(\lambda - \ell)(\lambda - u) = |a|(\lambda - \ell)(u - \lambda)$. For $\lambda \in [\ell, u]$ both factors are nonnegative, so this quantity is at least 0, and their product is at most $((u - \ell)/2)^2$, so it is at most $|a|(u - \ell)^2/4$. $\square$

To simplify the notation, in the remainder of the paper we denote the objective function of Problem S-MIQP by

$$f^S(y) := y^T D y + h^T y.$$

We also denote by $f_{\inf}^S, f_{\sup}^S$ the infimum and supremum of f^S on the feasible region of Problem S-MIQP.

Proposition 3 (Approximation algorithm for Problem S-MIQP). *Let κ be a positive rational absolute constant and let $\epsilon \in (0, 1]$ be rational. Assume that Problem S-MIQP is feasible, that D has at least one negative entry, and that $f_{\sup}^S - f_{\inf}^S \geq \kappa \gamma_{\text{neg}}$, where γ_{neg} is the largest absolute value among the negative entries of D . There is an algorithm that finds an ϵ -approximate solution to Problem S-MIQP. The algorithm runs on a Turing machine in time polynomial in the size of the instance, in $\text{size}(\epsilon)$, and in $1/\epsilon$, provided that p and the number k of negative entries of D are fixed.*

Proof. In this proof, we denote by d_i the i th diagonal entry of D . After possibly renaming the variables, we assume without loss of generality that $d_1 \leq d_2 \leq \dots \leq d_n$. We denote by $k \geq 1$ the number of negative entries of D , thus we have $d_1 \leq \dots \leq d_k < 0 \leq d_{k+1} \leq \dots \leq d_n$. We then have $\gamma_{\text{neg}} = \max_{i=1, \dots, k} |d_i|$.

The algorithm. We start by describing the approximation algorithm. Let $r := n^2$ (so that $\mathcal{B}(0, n^{3/2}) \subseteq \mathcal{B}(0, r)$, since $n^{3/2} \leq n^2$ for $n \geq 1$). Next, we define φ^k boxes in the first k coordinates of $\mathbb{R}^n$, where $\varphi := \lceil r\sqrt{k/(\kappa\epsilon)} \rceil$. Namely, for each $j_1, \dots, j_k \in \{1, \dots, \varphi\}$, we define the box

$$\mathcal{C}_{j_1, \dots, j_k} := \left\{ y \in \mathbb{R}^n : -r + \frac{2r}{\varphi}(j_i - 1) \leq y_i \leq -r + \frac{2r}{\varphi}j_i, \forall i = 1, \dots, k \right\}. \quad (9)$$

Note that the union of these φ^k boxes is the polyhedron

$$\{y \in \mathbb{R}^n : -r \leq y_i \leq r, \forall i = 1, \dots, k\},$$

which contains $\mathcal{B}(0, r)$.

We now fix one box $\mathcal{C} = \prod_{i=1}^k [\ell_i, u_i] \times \mathbb{R}^{n-k}$ among those defined in (9). For the box $\mathcal{C}$, we construct the affine functions $g_i : \mathbb{R} \rightarrow \mathbb{R}$ that attain the same values as $d_i y_i^2$ at ℓ_i, u_i , for $i = 1, \dots, k$:

$$g_i(y_i) := d_i(\ell_i + u_i)y_i - d_i\ell_i u_i \quad \forall i = 1, \dots, k.$$

Then we define the affine function $g : \mathbb{R}^k \rightarrow \mathbb{R}$ given by

$$g(y_1, \dots, y_k) := \sum_{i=1}^k g_i(y_i), \quad (10)$$

and we formulate the optimization problem obtained from Problem S-MIQP by substituting, in the objective function, $\sum_{i=1}^k d_i y_i^2$ with $g(y_1, \dots, y_k)$ and adding the $2k$ linear inequalities in (9) that enforce $y \in \mathcal{C}$:

$$\begin{aligned} \min \quad & \sum_{i=1}^k (d_i(\ell_i + u_i)y_i - d_i\ell_i u_i) + \sum_{i=k+1}^n d_i y_i^2 + h^\top y \\ \text{s. t.} \quad & Wy \leq w \\ & \ell_i \leq y_i \leq u_i \quad \forall i = 1, \dots, k \\ & y \in \Lambda_p(b^1, \dots, b^n) + \{c\}. \end{aligned} \quad (11)$$

We show that the change of variables $y = Bx + c$, where B is the matrix with columns $b^1, \dots, b^n$, maps Problem (11) to a MICQP defined on a bounded polyhedron, so that we can apply Lemma 16. First, this change of variables maps the constraint $y \in \Lambda_p(b^1, \dots, b^n) + \{c\}$ to the constraint $x \in \mathbb{Z}^p \times \mathbb{R}^{n-p}$. Second, since B is nonsingular, the map $y = Bx + c$ is an invertible affine transformation, and so sends the polyhedron $\{y \in \mathbb{R}^n : Wy \leq w, \ell_i \leq y_i \leq u_i \forall i = 1, \dots, k\}$ to a polyhedron in x , namely $\{x \in \mathbb{R}^n : WBx \leq w - Wc, \ell_i \leq (Bx + c)_i \leq u_i \forall i = 1, \dots, k\}$. Note that the latter polyhedron is bounded, since the polyhedron $\{y \in \mathbb{R}^n : Wy \leq w\}$ is already bounded, by the spherical form of Problem S-MIQP. Finally, the Hessian of the objective function of Problem (11), in the variable y , is the diagonal matrix with 0 in each of the first k diagonal entries (since these terms have been linearized) and $d_i \geq 0$ in each of the remaining diagonal entries, for $i = k+1, \dots, n$; this matrix is positive semidefinite. Since B is nonsingular, the Hessian of the objective function of Problem (11) in the variable x is obtained from this matrix by a congruence transformation, and hence, by Sylvester's law of inertia, is positive semidefinite as well. This shows that Problem (11), expressed in the variable x , is indeed a MICQP.

We now apply Lemma 16 to each Problem (11). Since Problem S-MIQP is feasible and the boxes constructed in (9) cover $\mathcal{B}(0, r)$, and hence the entire feasible region of Problem S-MIQP, at least one of the MICQPs (11) is feasible, and therefore has an optimal solution by Lemma 16. Among the MICQPs (11) that are feasible, the approximation algorithm selects one whose optimal value is smallest, and returns an optimal solution $y^\diamond$ of that MICQP. This concludes the description of the algorithm.

Approximation guarantee. Next, we show that $y^\diamond$ is an ϵ -approximate solution to Problem S-MIQP.

We first derive an upper bound on $f^S(y^\diamond) - f_{\inf}^S$ that depends linearly on γ_{neg} . Let $\mathcal{C} \subseteq \mathbb{R}^n$ be a box constructed in (9), say $\mathcal{C} = \prod_{i=1}^k [\ell_i, u_i] \times \mathbb{R}^{n-k}$. For each $i = 1, \dots, k$, we apply Lemma 17 to each univariate quadratic function $d_i y_i^2$ (using $d_i \leq 0$ for $i = 1, \dots, k$) and points ℓ_i, u_i . Since $u_i - \ell_i = 2r/\varphi$ and $|d_i| \leq \gamma_{\text{neg}}$ for $i = 1, \dots, k$, we obtain that, for every $y \in \mathcal{C}$,

$$g_i(y_i) \leq d_i y_i^2 \leq g_i(y_i) + \gamma_{\text{neg}} r^2 / \varphi^2.$$

We sum up all these inequalities for $i = 1, \dots, k$ and obtain that for every $y \in \mathcal{C}$,

$$g(y_1, \dots, y_k) \leq \sum_{i=1}^k d_i y_i^2 \leq g(y_1, \dots, y_k) + \gamma_{\text{neg}} k r^2 / \varphi^2. \quad (12)$$

Let $\mathcal{C}^\diamond$ be the box among (9) that yields the solution $y^\diamond$ and let $g^\diamond$ be the corresponding affine function defined in (10). Let y^* be an optimal solution to Problem S-MIQP, so that $f^S(y^*) = f_{\inf}^S$, let $\mathcal{C}^*$ be a box among (9) such that $y^* \in \mathcal{C}^*$ and let g^* be the corresponding affine function. We have

$$\begin{aligned}
f^S(y^\diamond) &= \sum_{i=1}^k d_i(y_i^\diamond)^2 + \sum_{i=k+1}^n d_i(y_i^\diamond)^2 + h^\top y^\diamond \\
&\leq g^\diamond(y_1^\diamond, \dots, y_k^\diamond) + \sum_{i=k+1}^n d_i(y_i^\diamond)^2 + h^\top y^\diamond + \gamma_{\text{neg}} k r^2 / \varphi^2 \\
&\leq g^*(y_1^*, \dots, y_k^*) + \sum_{i=k+1}^n d_i(y_i^*)^2 + h^\top y^* + \gamma_{\text{neg}} k r^2 / \varphi^2 \\
&\leq f^S(y^*) + \gamma_{\text{neg}} k r^2 / \varphi^2.
\end{aligned} \tag{13}$$

The first inequality follows by applying the right inequality in (12) to $\mathcal{C}^\diamond$ and $y^\diamond$. The second inequality holds by definition of $y^\diamond$: its left-hand side is the optimal value of the MICQP on $\mathcal{C}^\diamond$, which is smallest among all boxes, whereas its right-hand side is the value of the objective function of the MICQP on $\mathcal{C}^*$ at y^* , which is feasible for that MICQP since y^* is feasible for Problem S-MIQP and $y^* \in \mathcal{C}^*$. The third inequality follows by applying the left inequality in (12) to $\mathcal{C}^*$ and y^* .

We are now ready to show that $y^\diamond$ is an ϵ -approximate solution to Problem S-MIQP. We have

$$\frac{f^S(y^\diamond) - f_{\inf}^S}{f_{\sup}^S - f_{\inf}^S} \leq \frac{\gamma_{\text{neg}} k r^2}{\varphi^2} \cdot \frac{1}{\kappa \gamma_{\text{neg}}} = \frac{k r^2}{\kappa \varphi^2} \leq \epsilon.$$

In the first inequality we used (13) and the assumption $f_{\sup}^S - f_{\inf}^S \geq \kappa \gamma_{\text{neg}}$, and the last inequality holds by definition of φ .

Running time. The algorithm solves $\varphi^k = \left\lceil r \sqrt{k/(\kappa \epsilon)} \right\rceil^k$ MICQPs, one per box. Since $r = n^2$ and k is fixed, this number is polynomial in n and in $1/\epsilon$. Each of these MICQPs has size polynomial in the size of the instance and in $\text{size}(\epsilon)$: φ and the box endpoints ℓ_i, u_i have size $O(\log n + \text{size}(\epsilon))$, and φ is computable in polynomial time, being the ceiling of the square root of a rational number of size $O(\log n + \text{size}(\epsilon))$; the remaining coefficients of Problem (11) are obtained from these and from the original data by $O(1)$ arithmetic operations. Each such MICQP has p integer variables, and, by Lemma 16, is solved in polynomial time since p is fixed. Hence the entire algorithm runs on a Turing machine in time polynomial in the size of the instance, in $\text{size}(\epsilon)$, and in $1/\epsilon$, provided that p and k are fixed. $\square$

4.4 Theorems of the alternative

Our next goal is to present two theorems of the alternative, one for Problem S-MIQP and one for Problem MIQP. In these results, we show that there is an algorithm that either finds an ϵ -approximate solution, or finds a direction along which the instance splits into a number of sub-instances with one fewer integer variable. Let $\mathcal{S} \subseteq \mathbb{R}^n$ be a bounded closed set. Given a vector $v \in \mathbb{R}^n$, we define the *width of $\mathcal{S}$ along v* to be

$$\text{width}_v(\mathcal{S}) := \max \{v^\top x : x \in \mathcal{S}\} - \min \{v^\top x : x \in \mathcal{S}\}.$$

Now let $b^1, \dots, b^n$ be linearly independent vectors in $\mathbb{R}^n$ and consider a nonzero vector $v \in \mathbb{R}^n$ with $v^\top b^1, \dots, v^\top b^p$ integer and $v^\top b^{p+1} = \dots = v^\top b^n = 0$. Then $v^\top x$ is an integer for every $x \in \Lambda_p(b^1, \dots, b^n)$. Since $v^\top x$ takes values in an interval of length $\text{width}_v(\mathcal{S})$ as x ranges over $\mathcal{S}$, and takes only integer values on $\mathcal{S} \cap \Lambda_p(b^1, \dots, b^n)$, it follows that the number of hyperplanes orthogonal to v that contain points of $\mathcal{S} \cap \Lambda_p(b^1, \dots, b^n)$ is at most $\lfloor \text{width}_v(\mathcal{S}) \rfloor + 1$. We will use the following three lemmas, all of which have already been used, at least implicitly, in previous algorithms of this type. The first is a flatness-type result: it either finds a mixed integer point in a ball, or a direction along which the ball is flat.

Lemma 18 (Lemma 5.2 in [9]). *Let $a, c \in \mathbb{Q}^n$, let $r \in \mathbb{Q}$ with $r \geq 0$, let $b^1, \dots, b^n$ be linearly independent vectors in $\mathbb{Q}^n$, and let $p \in \{0, \dots, n\}$. There is an algorithm that either finds a vector in $\mathcal{B}(a, r) \cap (\Lambda_p(b^1, \dots, b^n) + \{c\})$, or finds a nonzero vector $v \in \mathbb{Q}^n$ with $v^\top b^1, \dots, v^\top b^p$ integer and $v^\top b^{p+1} = \dots = v^\top b^n = 0$ such that $\text{width}_v(\mathcal{B}(a, r)) \leq p2^{p(p-1)/4}$. The algorithm runs on a Turing machine in time polynomial in $\text{size}(a) + \text{size}(c) + \text{size}(r) + \sum_{i=1}^n \text{size}(b^i)$.*

Lemma 19 (Lemma 4 in [7] with $d = n$). *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a quadratic function of the form $f(y) = y^\top D y + h^\top y$, where D is diagonal, and the entry of D with the largest absolute value is the 1st diagonal entry d_1 . Let $y^+, y^- \in \mathbb{R}^n$ be such that $y_1^+ - y_1^- \geq 1$ and $\sum_{i=2}^n (y_i^+ - y_i^-)^2 \leq 1/4$. Let $\underline{f}$ and $\bar{f}$ be the minimum and maximum values attained by f on the three vectors $y^+, y^-, (y^+ + y^-)/2$. Then $\bar{f} - \underline{f} \geq \frac{3}{16}|d_1|$.*

Lemma 20. *Let $b^1, \dots, b^n$ be linearly independent vectors in $\mathbb{R}^n$, let $p \in \{0, \dots, n\}$, and let $c \in \mathbb{R}^n$. Let $y^+, y^- \in \Lambda_p(2b^1, \dots, 2b^n) + \{c\}$. Then*

$$\frac{y^+ + y^-}{2} \in \Lambda_p(b^1, \dots, b^n) + \{c\}.$$

Proof. Since $y^+, y^- \in \Lambda_p(2b^1, \dots, 2b^n) + \{c\}$, we can write

$$y^+ = \sum_{i=1}^n \mu_i (2b^i) + c, \quad y^- = \sum_{i=1}^n \nu_i (2b^i) + c,$$

with $\mu_i, \nu_i \in \mathbb{Z}$ for $i = 1, \dots, p$ and $\mu_i, \nu_i \in \mathbb{R}$ for $i = p+1, \dots, n$. Averaging the two expressions, we obtain

$$\frac{y^+ + y^-}{2} = \sum_{i=1}^n (\mu_i + \nu_i) b^i + c.$$

Since $\mu_i + \nu_i \in \mathbb{Z}$ for $i = 1, \dots, p$, the right-hand side is in $\Lambda_p(b^1, \dots, b^n) + \{c\}$. $\square$

We are now ready to give our first theorem of the alternative.

Proposition 4 (Theorem of the alternative for Problem S-MIQP). *Assume that, in Problem S-MIQP, D has at least one negative entry, and let $\epsilon \in (0, 1]$ be rational. There is an algorithm that either finds an ϵ -approximate solution to Problem S-MIQP and certifies that $f_{\sup}^S - f_{\inf}^S \geq 3\gamma_{\max}/16$, where $\gamma_{\max}$ is the largest absolute value of an entry of D , or finds a nonzero vector $v \in \mathbb{Q}^n$ with $v^\top b^1, \dots, v^\top b^p$ integer and $v^\top b^{p+1} = \dots = v^\top b^n = 0$ such that $\text{width}_v(\mathcal{B}(0, n^{3/2})) \leq 8p2^{p(p-1)/4}n^{3/2}$. The algorithm runs on a Turing machine in time polynomial in the size of the instance, in $\text{size}(\epsilon)$, and in $1/\epsilon$, provided that p and the number k of negative entries of D are fixed.*

Proof. In this proof, we denote by d_i the i th diagonal entry of D . After possibly renaming the variables, we assume without loss of generality that the entry of D with the largest absolute value is d_1 , so that $\gamma_{\max} = |d_1|$.

We denote by e^1 the first vector of the standard basis of $\mathbb{R}^n$, and we apply Lemma 18 to $\mathcal{B}(3e^1/4, 1/4)$ and the mixed integer lattice $\Lambda_p(2b^1, \dots, 2b^n) + \{c\}$, which is contained in $\Lambda_p(b^1, \dots, b^n) + \{c\}$ since $2\mu \in \mathbb{Z}$ whenever $\mu \in \mathbb{Z}$. Consider first the case where Lemma 18 finds a nonzero vector $v' \in \mathbb{Q}^n$ with $v'^\top (2b^1), \dots, v'^\top (2b^p)$ integer and $v'^\top (2b^{p+1}) = \dots = v'^\top (2b^n) = 0$ such that $\text{width}_{v'}(\mathcal{B}(3e^1/4, 1/4)) \leq p2^{p(p-1)/4}$. We then set $v := 2v' \in \mathbb{Q}^n$ and we have $v^\top b^1, \dots, v^\top b^p$ integer and $v^\top b^{p+1} = \dots = v^\top b^n = 0$. Furthermore, we have

$$\begin{aligned} \text{width}_v(\mathcal{B}(0, n^{3/2})) &= 2 \text{width}_{v'}(\mathcal{B}(0, n^{3/2})) = 8n^{3/2} \text{width}_{v'}(\mathcal{B}(0, 1/4)) \\ &= 8n^{3/2} \text{width}_{v'}(\mathcal{B}(3e^1/4, 1/4)) \leq 8p2^{p(p-1)/4}n^{3/2}. \end{aligned}$$

Hence the statement of the proposition holds. Therefore, we now assume that Lemma 18 finds a vector

$$y^+ \in \mathcal{B}(3e^1/4, 1/4) \cap (\Lambda_p(2b^1, \dots, 2b^n) + \{c\}).$$

Clearly, we also have $y^+ \in \mathcal{B}(0, 1)$, which is contained in the polyhedron $\{y \in \mathbb{R}^n : Wy \leq w\}$ by the spherical form; since y^+ also lies in $\Lambda_p(b^1, \dots, b^n) + \{c\}$, as observed above, it is feasible to Problem S-MIQP.

Next, we apply Lemma 18 to $\mathcal{B}(-3e^1/4, 1/4)$ and the mixed integer lattice $\Lambda_p(2b^1, \dots, 2b^n) + \{c\}$. Symmetrically, we can assume that Lemma 18 finds a vector

$$y^- \in \mathcal{B}(-3e^1/4, 1/4) \cap (\Lambda_p(2b^1, \dots, 2b^n) + \{c\}),$$

which in particular implies that y^- is in $\mathcal{B}(0, 1)$, and so it is feasible to Problem S-MIQP.

To conclude the proof, it suffices to show that $f_{\sup}^S - f_{\inf}^S \geq 3\gamma_{\max}/16$. Indeed, we can then use Proposition 3 (with $\kappa := 3/16$, the constant of Lemma 19) to find an ϵ -approximate solution, since $\gamma_{\max} \geq \gamma_{\text{neg}}$, where γ_{neg} is the largest absolute value among the negative entries of D , as in the statement of Proposition 3.

Since $y^+ \in \mathcal{B}(3e^1/4, 1/4)$ and $y^- \in \mathcal{B}(-3e^1/4, 1/4)$, we obtain $y_1^+ - y_1^- \geq 1$. The vectors $(y_2^+, \dots, y_n^+)$ and $(y_2^-, \dots, y_n^-)$ are both contained in the $(n-1)$ -dimensional ball $\mathcal{B}(0, 1/4)$, thus their distance is at most $1/2$. In other words,

$$\sum_{i=2}^n (y_i^+ - y_i^-)^2 \leq 1/4.$$

We define the midpoint of the segment joining y^+ and y^- as $y^\circ := (y^+ + y^-)/2$. Note that also the vector y° is feasible to Problem S-MIQP. Indeed, it satisfies the linear inequalities by convexity, and it is in $\Lambda_p(b^1, \dots, b^n) + \{c\}$ by Lemma 20, since $y^+, y^- \in \Lambda_p(2b^1, \dots, 2b^n) + \{c\}$.

By Lemma 19 (with $f := f^S$), the minimum and maximum values attained by f^S on y^+, y^-, y° differ by at least $\frac{3}{16}|d_1| = \frac{3}{16}\gamma_{\max}$. Since all three vectors are feasible to Problem S-MIQP, we conclude that $f_{\sup}^S - f_{\inf}^S \geq 3\gamma_{\max}/16$.

Running time. Each application of Lemma 18 runs in time polynomial in the size of the instance, since the balls $\mathcal{B}(\pm 3e^1/4, 1/4)$ have constant size and $b^1, \dots, b^n, c$ are part of the input. In the partition case, forming $v = 2v'$ takes $O(n)$ further arithmetic operations. Otherwise, forming y° takes $O(n)$ arithmetic operations, and, by Proposition 3, finding the ϵ -approximate solution takes time polynomial in the size of the instance, in $\text{size}(\epsilon)$, and in $1/\epsilon$ when p and k are fixed. Hence the entire algorithm runs on a Turing machine in time polynomial in the size of the instance, in $\text{size}(\epsilon)$, and in $1/\epsilon$, provided that p and k are fixed. $\square$

Next, we present our theorem of the alternative for Problem MIQP.

Proposition 5 (Theorem of the alternative for Problem MIQP). *Assume that, in Problem MIQP, the matrix H has at least one negative eigenvalue and $\{x \in \mathbb{R}^n : Wx \leq w\}$ is full-dimensional and bounded, and let $\epsilon \in (0, 1]$ be rational. There is an algorithm that either finds an ϵ -approximate solution to Problem MIQP, or finds a nonzero vector $v \in \mathbb{Z}^n$ with $v_{p+1} = \dots = v_n = 0$ and a scalar $\rho \in \mathbb{Q}$ such that*

$$\{v^\top x : Wx \leq w\} \subseteq [\rho, \rho + 8p2^{p(p-1)/4}n^{3/2}].$$

The algorithm runs on a Turing machine in time polynomial in the size of the instance, in $\text{size}(\epsilon)$, and in $1/\epsilon$, provided that p and the number k of negative eigenvalues of H are fixed.

Proof. We employ Proposition 2 with $\kappa := 3/16$, and we find a nonsingular matrix $L \in \mathbb{Q}^{n \times n}$, a diagonal matrix $D \in \mathbb{Q}^{n \times n}$, and a vector $a \in \mathbb{Q}^n$ such that: (i) H and D have the same inertia, thus the number of negative eigenvalues of H coincides with the number of negative entries of D . (ii) The following problem is a spherical form MIQP, where $b^1, \dots, b^n$ denote the columns of L^{-1} :

$$\begin{aligned} \min \quad & y^\top Dy + (h^\top L + 2a^\top HL)y \\ \text{s. t.} \quad & WLy \leq w - Wa \\ & y \in \Lambda_p(b^1, \dots, b^n) - \{L^{-1}a\}. \end{aligned} \tag{14}$$

(iii) If $y^\diamond$ is an $\epsilon/2$ -approximate solution to Problem (14) and $f_{\sup} - f_{\inf} \geq 3\gamma_{\min}/16$ (where f is the objective function of Problem (14), $f_{\inf}, f_{\sup}$ are the infimum and supremum of f on the feasible region of Problem (14), and $\gamma_{\min}$ is the smallest absolute value among the nonzero entries of D), then $x^\diamond := Ly^\diamond + a$ is an ϵ -approximate solution to Problem MIQP.

We use Proposition 4 (with $\epsilon := \epsilon/2$) and we either find an $\epsilon/2$ -approximate solution $y^\diamond$ to Problem (14) and we know that $f_{\sup} - f_{\inf} \geq 3\gamma_{\max}/16$, where $\gamma_{\max}$ is the largest absolute value of an entry of D , or we find a nonzero vector $v' \in \mathbb{Q}^n$ with $v'^T b^1, \dots, v'^T b^p$ integer and $v'^T b^{p+1} = \dots = v'^T b^n = 0$ such that $\text{width}_{v'}(\mathcal{B}(0, n^{3/2})) \leq 8p2^{p(p-1)/4}n^{3/2}$.

In the first case, since $\gamma_{\max} \geq \gamma_{\min}$, we have $f_{\sup} - f_{\inf} \geq 3\gamma_{\min}/16$, and hence $x^\diamond := Ly^\diamond + a$ is an ϵ -approximate solution to Problem MIQP by (iii), and we are done. Consider now the second case. Let $v := L^{-T}v' \in \mathbb{Q}^n$. Since L is nonsingular, v is nonzero. For $j = 1, \dots, n$, we have $v_j = b^j T v'$. Thus, $v'^T b^1, \dots, v'^T b^p$ integer implies $v_1, \dots, v_p$ integer, and $v'^T b^{p+1} = \dots = v'^T b^n = 0$ implies $v_{p+1} = \dots = v_n = 0$. Furthermore,

$$\begin{aligned} \text{width}_v \{x \in \mathbb{R}^n : Wx \leq w\} &= \max \{v^T x : Wx \leq w\} - \min \{v^T x : Wx \leq w\} \\ &= \max \{v'^T L^{-1}x : Wx \leq w\} - \min \{v'^T L^{-1}x : Wx \leq w\} \\ &= \max \{v'^T (y + L^{-1}a) : WLy \leq w - Wa\} - \min \{v'^T (y + L^{-1}a) : WLy \leq w - Wa\} \\ &= \max \{v'^T y : WLy \leq w - Wa\} - \min \{v'^T y : WLy \leq w - Wa\} \\ &= \text{width}_{v'} \{y \in \mathbb{R}^n : WLy \leq w - Wa\} \\ &\leq \text{width}_{v'} \mathcal{B}(0, n^{3/2}) \\ &\leq 8p2^{p(p-1)/4}n^{3/2}. \end{aligned}$$

Let $\rho := \min \{v^T x : Wx \leq w\}$, and note that we can calculate it in polynomial time by solving a linear programming problem. We obtain

$$\{v^T x : Wx \leq w\} \subseteq [\rho, \rho + 8p2^{p(p-1)/4}n^{3/2}].$$

This concludes the second case.

Running time. By Proposition 2, computing L, D, a takes time polynomial in the size of the instance and in $\text{size}(\epsilon)$; by (i), D has k negative entries, so Proposition 4 (applied with $\epsilon/2$) runs in time polynomial in the size of the instance, in $\text{size}(\epsilon)$, and in $1/\epsilon$ when p and k are fixed. In the first case, forming $x^\diamond = Ly^\diamond + a$ takes $O(n^2)$ further arithmetic operations; in the second, forming $v = L^{-T}v'$ takes $O(n^2)$ arithmetic operations and computing ρ takes polynomial time via a linear program. Hence the entire algorithm runs on a Turing machine in time polynomial in the size of the instance, in $\text{size}(\epsilon)$, and in $1/\epsilon$, provided that p and k are fixed. $\square$

4.5 Auxiliary results

The proof of Theorem 1 relies on a recursive application of Proposition 5. Before turning to it, we state a few additional results that we will use.

Lemma 21 (Theorem 4 in [10]). *Assume that Problem MIQP is feasible and the objective function is bounded below on the feasible region. There is an optimal solution to Problem MIQP of size bounded by an integer ψ , which is polynomial in the size of the instance.*

We remark that, even though Theorem 4 in [10] does not give ψ explicitly, a formula for ψ , as a function of the size of the instance, can be derived from its proof.

Lemma 22 (Theorem 1.1 in [8]). *Let $W \in \mathbb{Q}^{m \times n}$, $w \in \mathbb{Q}^m$, $p \in \{0, 1, \dots, n\}$, and consider the sets*

$$\mathcal{P} := \{x \in \mathbb{R}^n : Wx \leq w\}, \quad \mathcal{S} := \mathcal{P} \cap (\mathbb{Z}^p \times \mathbb{R}^{n-p}).$$

There is a polynomial-time algorithm that either returns that $\mathcal{S}$ is empty, or finds $p' \in \{0, 1, \dots, p\}$, $n' \in \{p', p'+1, \dots, p'+n-p\}$, $\bar{x} \in \mathbb{Z}^p \times \mathbb{Q}^{n-p}$, and $T \in \mathbb{Q}^{n \times n'}$ of full rank such that, defining $\tau : \mathbb{R}^{n'} \rightarrow \mathbb{R}^n$ by $\tau(x') := \bar{x} + Tx'$, the polyhedron

$$\mathcal{P}' := \{x' \in \mathbb{R}^{n'} : WT x' \leq w - W\bar{x}\}$$

is full-dimensional, and

$$\mathcal{P} = \tau(\mathcal{P}'), \quad \mathcal{S} = \tau\left(\mathcal{P}' \cap (\mathbb{Z}^{p'} \times \mathbb{R}^{n'-p'})\right).$$

Furthermore, if an equality $v^\top x = \beta$ with $v_{p+1} = \dots = v_n = 0$ is valid for $\mathcal{P}$, then $p' \leq p-1$.

Lemma 23 (Theorem 3 in [7]). *Let $H \in \mathbb{Q}^{n \times n}$ be symmetric. There is an algorithm that computes a nonsingular matrix $L \in \mathbb{Q}^{n \times n}$ and a diagonal matrix $D \in \mathbb{Q}^{n \times n}$ such that $L^\top H L = D$. The algorithm is strongly polynomial: it performs a number of arithmetic operations polynomial in n , and runs on a Turing machine in time polynomial in $\text{size}(H)$.*

Lemma 24 (Lemma 7.3 in [9]). *Let $\mathcal{S}^1, \dots, \mathcal{S}^t \subseteq \mathbb{R}^n$. Let $f : \bigcup_{i=1}^t \mathcal{S}^i \rightarrow \mathbb{R}$ and assume that f has a minimum on $\bigcup_{i=1}^t \mathcal{S}^i$. Let $\epsilon \in [0, 1]$. For $i = 1, \dots, t$, let x^i be an ϵ -approximate solution to the optimization problem $\inf\{f(x) : x \in \mathcal{S}^i\}$. Then, each optimal solution to $\min\{f(x) : x \in \{x^1, \dots, x^t\}\}$ is an ϵ -approximate solution to the optimization problem $\inf\{f(x) : x \in \bigcup_{i=1}^t \mathcal{S}^i\}$.*

4.6 Proof of Theorem 1

Proof. We now present our approximation algorithm for Problem MIQP. The input to the algorithm is an instance of Problem MIQP, that is, specific data H, h, W, w , such that the objective function is bounded below on the feasible region. We denote by k the number of negative eigenvalues of the matrix H of the input instance, that is, $k = k_-$ in the statement of Theorem 1.

Initialization step 1: Bounded polyhedron. By Lemma 21 (together with the remark following it), there is a formula, depending only on the size of the instance, for an integer ψ such that: if the instance is feasible, then it has an optimal solution of size at most ψ . We compute ψ from this formula, regardless of whether the instance is feasible, and construct a new instance, obtained from the original one by adding the linear constraints $-2^\psi \leq x_i \leq 2^\psi$, for $i = 1, \dots, n$. The polyhedron corresponding to this new instance is $\{x \in \mathbb{R}^n : Wx \leq w, -2^\psi \leq x_i \leq 2^\psi, \text{ for } i = 1, \dots, n\}$, which is bounded. Furthermore, the size of the new instance is polynomial in the size of the former. If the original instance is infeasible, then the new instance remains infeasible, since its feasible region is contained in the one of the original instance. If the original instance is feasible, then, since some optimal solution has size at most ψ and hence satisfies the added inequalities, the feasible region remains nonempty and the infimum of the objective function is unchanged; since the supremum can only decrease, an ϵ -approximate solution to the new instance is also an ϵ -approximate solution to the original one. Therefore, we may replace the original instance by this new instance, whose corresponding polyhedron is bounded, without affecting feasibility, and so that finding an ϵ -approximate solution to the new instance (or detecting its infeasibility) suffices. For notational simplicity, we now use H, h, W, w to denote the data of the new instance, so we now assume that the system $Wx \leq w$ contains the linear inequalities $-2^\psi \leq x_i \leq 2^\psi$, for $i = 1, \dots, n$.

Initialization step 2: Full dimensional polyhedron. We now apply Lemma 22 to the feasible region of the instance. If we find out that the feasible region is empty, then the instance is infeasible, and we are done. Otherwise, we obtain $p' \in \{0, 1, \dots, p\}$, $n' \in \{p', p'+1, \dots, p'+n-p\}$, $\bar{x} \in \mathbb{Z}^p \times \mathbb{Q}^{n-p}$, and $T \in \mathbb{Q}^{n \times n'}$ of full rank, as in the statement of Lemma 22. We perform the change of variables $x = \bar{x} + Tx'$ to the instance and, after dropping the constant $\bar{x}^\top H \bar{x} + h^\top \bar{x}$ in the objective function, we obtain the instance

$$\begin{aligned} \min \quad & x'^\top H' x' + h'^\top x' \\ \text{s. t.} \quad & W' x' \leq w' \\ & x' \in \mathbb{Z}^{p'} \times \mathbb{R}^{n'-p'}, \end{aligned} \tag{15}$$

where $H' := T^\top HT$, $h'^\top := h^\top T + 2\bar{x}^\top HT$, $W' := WT$, and $w' := w - W\bar{x}$. By Lemma 22, the polyhedron $\{x' \in \mathbb{R}^{n'} : W'x' \leq w'\}$ is full-dimensional; furthermore, since $\{x \in \mathbb{R}^n : Wx \leq w\}$ is bounded and $\tau(x') := \bar{x} + Tx'$ maps this polyhedron bijectively onto $\{x \in \mathbb{R}^n : Wx \leq w\}$, the polyhedron $\{x' \in \mathbb{R}^{n'} : W'x' \leq w'\}$ is bounded as well. The number of negative eigenvalues of H' is at most the number of negative eigenvalues of H . Indeed, for a symmetric matrix, the number of negative eigenvalues equals the largest dimension of a subspace on which the associated quadratic form is negative definite; and if $x'^\top H'x'$ is negative definite on a subspace $V' \subseteq \mathbb{R}^{n'}$, then, since T has full column rank, $x^\top Hx$ is negative definite on the subspace $TV' \subseteq \mathbb{R}^n$, which has the same dimension as V' . Since the definition of ϵ -approximate solution is preserved under changes of variables and translations of the objective function, and since the instance is feasible if and only if Instance (15) is, it then suffices to find an ϵ -approximate solution to Instance (15), or to detect that it is infeasible.

Recursive steps. The remainder of the algorithm will be applied recursively to several instances of Problem MIQP whose corresponding polyhedron is full-dimensional and bounded. In the first iteration, the instance considered is Instance (15). For notational simplicity, we assume that the instance considered has again data H, h, W, w . Therefore, we assume that the polyhedron $\{x \in \mathbb{R}^n : Wx \leq w\}$ is full-dimensional and bounded.

Recursive step 1: Convex case. By Lemma 23, there is a strongly polynomial algorithm that finds a nonsingular matrix $L \in \mathbb{Q}^{n \times n}$ and a diagonal matrix $D \in \mathbb{Q}^{n \times n}$ such that $L^\top HL = D$; we use this only to check whether H is positive semidefinite, and L, D play no further role below. By Sylvester's law of inertia, H and D have the same inertia; in particular, H is positive semidefinite if and only if every diagonal entry of D is nonnegative, which can be checked in polynomial time. If H is positive semidefinite, then, since $\{x \in \mathbb{R}^n : Wx \leq w\}$ is bounded, we apply Lemma 16 to our instance. If Lemma 16 finds that the instance is infeasible, we are done with this instance. Otherwise, Lemma 16 finds an optimal solution x^* to our instance in polynomial time, which is also an ϵ -approximate solution. Thus, in the remainder of the proof, we assume that H is not positive semidefinite; in particular, since H and D have the same inertia, the matrix H of the instance considered has at least one negative eigenvalue.

Recursive step 2: Approximation or partition. We now apply the algorithm in Proposition 5. If the algorithm finds an ϵ -approximate solution to our instance, then we are done with this instance. Otherwise, the algorithm finds a nonzero vector $v \in \mathbb{Z}^n$ with $v_{p+1} = \dots = v_n = 0$ and a scalar $\rho \in \mathbb{Q}$ such that

$$\{v^\top x : Wx \leq w\} \subseteq \left[\rho, \rho + 8p2^{p(p-1)/4}n^{3/2} \right].$$

Note that $p \geq 1$: indeed, v is nonzero and satisfies $v_{p+1} = \dots = v_n = 0$, which is impossible if $p = 0$. Thus the partition case can occur only when $p \geq 1$, so that each partition strictly decreases the number of integer variables; this is what guarantees that the recursion terminates. Note also that $v^\top x$ is an integer for every $x \in \mathbb{Z}^p \times \mathbb{R}^{n-p}$, since $v \in \mathbb{Z}^n$ and $v_{p+1} = \dots = v_n = 0$. Therefore, every feasible point of our instance is contained in one of the hyperplanes $\{x \in \mathbb{R}^n : v^\top x = \beta\}$, where β ranges over the integers with $\lceil \rho \rceil \leq \beta \leq \lceil \rho + 8p2^{p(p-1)/4}n^{3/2} \rceil$. For each such β , we define the instance

$$\begin{aligned} \min \quad & x^\top Hx + h^\top x \\ \text{s. t.} \quad & Wx \leq w \\ & v^\top x = \beta \\ & x \in \mathbb{Z}^p \times \mathbb{R}^{n-p}. \end{aligned} \tag{16}$$

For each β , we apply Lemma 22 to the feasible region of Instance (16), as in Step 2. If we find out that the feasible region is empty, we do not need to consider further Instance (16) for such value of β . Otherwise, we transform, with a change of variables, Instance (16) into a new instance defined over a full-dimensional polyhedron, which we denote by Instance (16)'. This polyhedron is also bounded: the feasible region of Instance (16) is contained in the bounded set $\{x \in \mathbb{R}^n : Wx \leq w\}$, and, by Lemma 22, it is the image of the polyhedron of Instance (16)' under an injective affine map. Since $v_{p+1} = \dots = v_n = 0$, Instance (16)' has at most $p - 1$ integer variables and at most $n - p$ continuous variables. Moreover,

exactly as in Initialization step 2, the number of negative eigenvalues of the Hessian of Instance (16)' is at most that of H .

Recursive step 3: Recursion. We apply the recursive steps of the algorithm to each obtained Instance (16)'. Since each partition in Recursive step 2 decreases the number of integer variables by at least one, the recursion terminates.

Termination. Each ϵ -approximate solution of a sub-instance found by the algorithm is mapped back to the corresponding feasible solution to the original instance by inverting the affine transformations performed by the algorithm (via Lemma 22) to construct such sub-instance. The algorithm then returns one of these feasible solutions with minimum objective value. Lemma 24 implies that the returned solution is an ϵ -approximate solution to the original instance. In case the algorithm never found any ϵ -approximate solutions of any sub-instance, the algorithm returns that the original instance is infeasible.

Running time. The initialization steps run in polynomial time: computing ψ and adding the box constraints (Initialization step 1), and applying Lemma 22 (Initialization step 2). For the recursion, since each partition decreases the number of integer variables by at least one, its depth is at most p ; and since each partition creates at most $8p2^{p(p-1)/4}n^{3/2} + 1$ sub-instances, the total number of iterations is at most $(p + 1)(8p2^{p(p-1)/4}n^{3/2} + 1)^p$, which is polynomial in n because p is fixed. As this depth is the constant p , every instance considered has size polynomial in the size of the original instance, and, by Recursive step 2, its number of negative eigenvalues never exceeds k . Each iteration checks positive semidefiniteness via Lemma 23, and then either solves a MICQP via Lemma 16 or invokes Proposition 5 and applies Lemma 22 to each sub-instance; when p and k are fixed, each of these runs in time polynomial in the size of the instance, in $\text{size}(\epsilon)$, and in $1/\epsilon$. Hence the entire algorithm runs on a Turing machine in time polynomial in the size of the instance, in $\text{size}(\epsilon)$, and in $1/\epsilon$, provided that p and k are fixed. $\square$

Funding: A. Del Pia is partially supported by AFOSR Grant FA9550-23-1-0433 and ONR Grant N00014-25-1-2490. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Air Force Office of Scientific Research or the Office of Naval Research.

References

- [1] Mihir Bellare and Phillip Rogaway. The complexity of approximating a nonlinear program. *Mathematical Programming*, 69:429–441, 1995.
- [2] Augustin Cauchy. *Exercices de Mathématiques, Quatrième Année*. Bibliothèque Nationale, Paris, 1829.
- [3] A. Dax and S. Kaniel. Pivoting techniques for symmetric Gaussian elimination. *Numerische Mathematik*, 28:221–241, 1977.
- [4] Etienne de Klerk, Monique Laurent, and Pablo A. Parrilo. A PTAS for the minimization of polynomials of fixed degree over the simplex. *Theoretical Computer Science*, 361:210–225, 2006.
- [5] Alberto Del Pia. On approximation algorithms for concave mixed-integer quadratic programming. In *Proceedings of IPCO 2016*, volume 9682 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 2016.
- [6] Alberto Del Pia. On approximation algorithms for concave mixed-integer quadratic programming. *Mathematical Programming, Series B*, 172(1–2):3–16, 2018.
- [7] Alberto Del Pia. An approximation algorithm for indefinite mixed integer quadratic programming. *Mathematical Programming, Series A*, 201:263–293, 2023.
- [8] Alberto Del Pia. Convex quadratic sets and the complexity of mixed integer convex quadratic programming. *SIAM Journal on Optimization*, 35(3):1822–1845, 2025.

- [9] Alberto Del Pia. The mixed integer trust region problem. *Mathematical Programming, Series A*, 213:699–736, 2025.
- [10] Alberto Del Pia, Santanu S. Dey, and Marco Molinaro. Mixed-integer quadratic programming is in NP. *Mathematical Programming, Series A*, 162(1):225–240, 2017.
- [11] G.H. Golub and C.F. Van Loan. *Matrix Computations, 4th edition*. Johns Hopkins University Press, Baltimore, MD, USA, 2013.
- [12] Hendrik W. Jr. Lenstra. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4):538–548, 1983.
- [13] M. Mignotte. Some useful bounds. In B. Buchberger, G.E. Collins, and R. Loos, editors, *Computer Algebra: Symbolic and Algebraic Computation*, pages 259–263. Springer, Verlag, Wien, 1982.
- [14] Arkadi S. Nemirovski and David Berkovich Yudin. *Problem Complexity and Method Efficiency in Optimization*. Wiley, Chichester, 1983. Translated by E.R. Dawson from Slozhnost’ Zadach i Effektivnost’ Metodov Optimizatsii (1979).
- [15] Panos M. Pardalos and Stephen A. Vavasis. Quadratic programming with one negative eigenvalue is NP-hard. *Journal of Global Optimization*, 1(1):15–22, 1991.
- [16] Alexander Schrijver. *Theory of Linear and Integer Programming*. Wiley, Chichester, 1986.
- [17] Stephen A. Vavasis. Approximation algorithms for indefinite quadratic programming. *Mathematical Programming*, 57:279–311, 1992.
- [18] Stephen A. Vavasis. On approximation algorithms for concave quadratic programming. In C.A. Floudas and Panos M. Pardalos, editors, *Recent Advances in Global Optimization*, pages 3–18. Princeton University Press, Princeton, NJ, 1992.