Seven directions illuminate every complex convex body in $\mathbb{C}^2$

hexagon:2611.01355v1

A complex convex body is a compact convex set with nonempty interior invariant under multiplication by every scalar of modulus one. A direction d illuminates $x \in \partial K$ if $x + td \in \operatorname{int} K$ for some $t > 0$. Let $H_2^{\mathbb{C}}$ be the supremum over these bodies of the minimum number of illuminating directions. Here and below B denotes the complex Euclidean unit ball. The real John ellipsoid of K is invariant under scalar rotations, by uniqueness. Consequently a complex-linear change of variables makes it B. At a unit contact u, K satisfies $|\langle u, x\rangle| \le 1$, since its real supporting inequality holds after every scalar rotation. Averaging the real John decomposition under scalar rotations gives

$$I_n = \sum_j c_j u_j u_j^*, \qquad c_j > 0, \qquad \sum_j c_j = n.$$

In particular $B \subset K \subset \sqrt{n}\, B$. To see the averaging factor, on the real plane $\mathbb{C}u$ the average of the real matrices of projection onto $\mathbb{R}e^{it}u$ is one half of the identity on that plane. We use Hermitian products linear in the second variable. For reference, $-d$ illuminates x precisely when $\operatorname{Re} f(d) > 0$ for every complex linear f norming x, i.e. $f(x) = 1$ and $|f| \le 1$ on K. The real parts of these f's are exactly the normalized real supports at x, by balancedness. The criterion follows from separation, or from the one-sided derivative of the norm.

1 The sharp bound in complex dimension two

Theorem 1. *Every complex convex body in $\mathbb{C}^2$ can be illuminated by seven directions. In particular $H_2^{\mathbb{C}} = 7$.*

The proof uses John's contact directions to select between two explicit sets of seven vectors. One finite covering calculation is certified by an exhaustive integer-arithmetic program. The theorem concerns the maximum value of the illumination number. The classification of the bodies attaining it is a further question.

A sufficient condition for illumination. Assume the John ellipsoid is the Euclidean ball. Write a boundary point as $x = ru$, where $\|u\|_2 = 1$ and $r \ge 1$. A norming complex functional at x has the form $f(y) = \langle u/r + w, y\rangle$, where $w \perp_{\mathbb{C}} u$ and $\|w\|_2 \le \sqrt{1 - r^{-2}}$. Indeed $f(ru) = 1$ and $\|f\|_2 \le 1$ since $B \subset K$. It follows that $-v$ illuminates x whenever

$$\operatorname{Re}\langle u, v\rangle > \sqrt{r^2 - 1}\, |\det(u, v)|. \tag{1}$$

Here $|\det(u, v)|$ equals the length of the component of v orthogonal to the complex line $\mathbb{C}u$.

The contact directions on the two-sphere. For a unit vector $u = (u_1, u_2)$ put

$$p(u) = (2\operatorname{Re}(\overline{u}_1 u_2), 2\operatorname{Im}(\overline{u}_1 u_2), |u_1|^2 - |u_2|^2) \in S^2.$$

The rank-one projector identity gives $2|\langle u, v\rangle|^2 = 1 + p(u) \cdot p(v)$ for unit vectors. The John decomposition above gives a probability distribution on contact directions with average projector $I_2/2$. Thus zero belongs to their convex hull on the two-sphere. By Carathéodory's theorem we can select at most four such sphere points v_i with $0 \in T = \operatorname{conv}\{v_i\}$. Use a minimal choice, so its barycentric weights are positive and the vertices are affinely independent. Each contact supplies $r^2(1 + p(u) \cdot v_i) \le 2$. Consequently, with h_T the support function of T,

$$\sqrt{r^2 - 1} \le \sqrt{\frac{1 - h_T(p(u))}{1 + h_T(p(u))}}. \tag{2}$$

Choosing a complex orthonormal basis allows any prescribed sphere axis to be the third coordinate: take a unit vector on its projective line and an orthogonal complement.

Lemma 2. *The convex hull of three points on the unit circle that contains its center contains a segment $[-e/\sqrt{3}, e/\sqrt{3}]$ for a unit vector e.*

Proof. If two points are antipodal their diameter suffices. Otherwise write $aw_1 + bw_2 + cw_3 = 0$, where $a + b + c = 1$ and $a \ge b \ge c > 0$. Put $q = aw_1 - bw_2 + (b - a)w_3$. These three coefficients sum to zero and their absolute values are at most a, b, c respectively: $a \le b + c$ by the triangle inequality. Adding or subtracting them from the original weights shows that both q and $-q$ belong to the triangle. If γ is the angle between w_1 and w_2, then $\cos\gamma = (c^2 - a^2 - b^2)/(2ab) \le -1/2$. Also $w_3 \cdot (w_1 - w_2) = -(a - b)(1 - \cos\gamma)/c \le 0$. Taking the projection of q onto the unit vector parallel to $w_1 - w_2$ gives $\|q\| \ge (a + b)\sin(\gamma/2) \ge 1/\sqrt{3}$. □

Two lower bounds on the support function. Let $D = 3/100$, $c = 577/1000$ and $d = \min_{p \in S^2} h_T(p)$. If $d \le D$, we claim that an axis can be chosen so that, with z the third coordinate of p,

$$h_T(p) \;\ge\; L_0(z) := \max\left\{\frac{c|z|}{1 + \sqrt{1 - z^2}}, c|z| - D\sqrt{1 - z^2}\right\}. \tag{3}$$

For a tetrahedron, the ball of radius d about zero touches a facet with unit outer normal m at dm. This follows by representing the tetrahedron as the intersection of its facet halfspaces and minimizing their distances from zero. The vertices of that facet lie on a circle of radius $\sqrt{1 - d^2}$ centered at dm, and the triangle contains dm. Lemma 2 supplies a unit vector $e \perp m$ such that $dm \pm e\sqrt{(1 - d^2)/3}$ both belong to T. In particular $dm \pm ce \in T$, since $3c^2 < 1 - D^2$. For $z = p \cdot e$, this gives

$$h_T(p) \ge \max\{d, c|z| - d\sqrt{1 - z^2}\}.$$

The maximum is at least $c|z|/(1 + \sqrt{1 - z^2})$, and the second term is at least $c|z| - D\sqrt{1 - z^2}$. For a triangle T, its affine plane contains zero, and the same argument uses its actual circumcircle with $d = 0$. For a segment T, its unit endpoints are antipodal and its own axis suffices.

If $d \ge D$, take as axis a contact $e = v_j$ of largest barycentric weight $a \ge 1/4$. The convex combination of the remaining contacts with their weights renormalized is $-ae/(1 - a)$. Thus both e and $-e/3$ belong to T, giving

$$h_T(p) \;\ge\; L_1(z) := \max\{D, z, -z/3\}, \qquad z = p \cdot e. \tag{4}$$

The finite covering calculation. Use the associated complex orthonormal coordinates in each case. The following lemma is the computational step. Its exact certificate is the accompanying file `verify_two.py`. Its directions have integer real and imaginary parts, obtained by rounding each

coordinate of $10^6(\zeta^{2j}, s\zeta^{3j})$, $0 \le j < 7$, with $\zeta = \exp(2\pi i/7)$. The integers, given literally in the certificate, specify the direction sets.

Lemma 3. *For $k = 0, 1$, use the corresponding set of seven directions with $s = 1$ or $s = 121/100$, respectively. For every complex Euclidean unit vector u, at least one vector v in that set satisfies*

$$\operatorname{Re}\langle u, v\rangle > \sqrt{\frac{1 - L_k(z)}{1 + L_k(z)}}\,|\det(u, v)|, \qquad z = |u_1|^2 - |u_2|^2.$$

Exact finite verification of Lemma 3. The certificate covers the projective coordinate by the two charts $(1, w)$ and $(w, 1)$, $|w| \le 1$. It covers the scalar phase by $\pm(1 + it)^2/(1 + t^2)$, $-1 \le t \le 1$. After homogeneity cancels the normalization, this leaves the three real parameters $\operatorname{Re} w, \operatorname{Im} w, t$. The program subdivides a cube into eight children until it either lies outside the projective chart or a single direction is certified by a strict inequality throughout the cube. The certificate evaluates rational bounds with integer arithmetic. Its full chart domains retain all seven directions, including the small changes in the coordinates arising from rounding.

For completeness, write the center of a dyadic cube as $(X, Y, T)/N$ and its half-side as $1/N$. Its coordinates represent w and t. Interchanging the components of each direction handles the second chart. Set

$$M = \min\{N^2, (|X|+1)^2 + (|Y|+1)^2\}, \quad B_0 = N^2 + M, \quad Z = N^2 - M, \quad S = \min\{B_0, 2N\lceil\sqrt{M}\rceil\}.$$

On the part of the cube in the chart, $Z/B_0 \le |z|$ and $S/B_0 \ge \sqrt{1 - z^2}$, by monotonicity of $2\sqrt{m}/(1 + m)$ on $[0, 1]$. In the small regime a rational lower bound U/V for L_0 is the larger of $577Z/[1000(B_0 + S)]$ and $(577Z - 30S)/(1000B_0)$. In the large regime take the larger of $3/100$ and Z/B_0 (or $Z/(3B_0)$ in the interchanged chart). Thus the coefficient of the determinant is at most $k = \sqrt{(V - U)/(V + U)} \le 1$.

For $v = (a + ib, c' + ib')$, the expression to bound is

$$\operatorname{Re}\big(\overline{\rho(t)}(v_1 + \overline{w}v_2)\big) - k|v_2 - wv_1|, \qquad \rho(t) = \pm(1 + it)^2/(1 + t^2).$$

Here k is held fixed on the cube. Its loss relative to the center is at most $(2\sqrt{2} + 2\sqrt{3})\|v\|_2/N < 11{,}000{,}000/N$: the two occurrences of w contribute at most $2\sqrt{2}\|v\|_2/N$, $|\rho'| \le 2$, and the center satisfies $|v_1 + \overline{w}v_2| \le \sqrt{3}\|v\|_2$. The last strict numerical inequality follows for every listed direction from $49\|v\|_2^2 < 11{,}000{,}000^2$. The script accepts a box if the center's real part, after subtracting that loss, is positive and its square strictly exceeds the squared determinant term. Clearing positive denominators makes this test an integer comparison. Starting from $[-1, 1]^3$, the eight runs terminate at maximum depths $16, 15, 15, 15$ in the small regime and $12, 12, 13, 13$ in the large regime. Within each group the direct chart precedes the interchanged chart and the positive phase sign precedes the negative one. □

The same certificate for exact seventh roots. The unrounded vectors $(\zeta^{2j}, s\zeta^{3j})$ also illuminate after the same choice of coordinates. Indeed the center test reserves a loss $11 \cdot 10^6/N$, whereas the variation on a box is less than $10^7/N$ since $2\sqrt{2} + 2\sqrt{3} < 6.3$ and $\|v\|_2 < 11 \cdot 10^6/7$. Every accepted box thus retains a margin greater than $10^6/N$ in the unnormalized inequality for its integer vector. The verifier enforces $N \le 2^{17}$, so this margin exceeds 7. For the vector with exact seventh-root coordinates scaled by 10^6, the error in each real coordinate is less than one, hence the Euclidean error is less than two. The additional check in the program verifies the coordinate bounds by degree-20 Taylor polynomials on $[0, 6\pi/7]$, with remainder at most $3^{21}/21!$. On $|w| \le 1$

both the scalar product and the determinant are Lipschitz in the direction vector with constant at most $\sqrt{2}$. Replacing the integer vector by the exact one therefore costs less than $4\sqrt{2} < 7$. Positive rescaling gives the asserted seventh-root directions.

Proof of Theorem 1. Choose the axis and directions corresponding to the applicable bound on h_T. The right-hand side of (2) decreases with h_T. Lemma 3 now implies (1) at each boundary point for one of the seven directions. The lower bound follows below. □

Sharpness for the bidisc. At (z, w) with $|z| = |w| = 1$, a direction (a, b) illuminates precisely when $\mathrm{Re}(\bar{z}a) < 0$ and $\mathrm{Re}(\bar{w}b) < 0$. Given six directions, choose z perpendicular to a nonzero first component; if none exists, all six already fail. Across the signs $z, -z$ at most five first inequalities succeed in total. Choose the sign for which at most two succeed. Each of those directions covers at most an open semicircle of possible w. Two open semicircles cannot cover the circle, since one of the two antipodal endpoints of the first lies outside the second. Some (z, w) is therefore missed.

A Exact verification program

Run with Python 3.8 or later, with assertions enabled.

```
"""Verify the two seven-direction coverings in complex dimension two.

Run with Python 3.8 or later. All verification uses exact integer arithmetic.
The four coordinates of a direction are Re u1, Im u1, Re u2, Im u2.
"""

from fractions import Fraction
from math import factorial, isqrt


SMALL = (
    (1000000, 0, 1000000, 0),
    (-222521, 974928, -900969, 433884),
    (-900969, -433884, 623490, -781831),
    (623490, -781831, -222521, 974928),
    (623490, 781831, -222521, -974928),
    (-900969, 433884, 623490, 781831),
    (-222521, -974928, -900969, -433884),
)

LARGE = (
    (1000000, 0, 1210000, 0),
    (-222521, 974928, -1090172, 524999),
    (-900969, -433884, 754423, -946016),
    (623490, -781831, -269250, 1179663),
    (623490, 781831, -269250, -1179663),
    (-900969, 433884, 754423, 946016),
    (-222521, -974928, -1090172, -524999),
)


def support_lower_bound(m, n, swapped, small):
    """Return numerator and positive denominator of a lower bound for h."""
    n2 = n*n
    b = n2+m
    z = n2-m  # z/b <= absolute Bloch height on the entire clipped box
    if not small:
        divisor = 3*b if swapped else b
        return (z, divisor) if 100*z > 3*divisor else (3, 100)

    # The transverse Bloch radius is at most s/b. On 0 <= m/n^2 <= 1,
    # the function 2 sqrt(r)/(1+r) is increasing.
    sqrt_up = isqrt(m)
    if sqrt_up*sqrt_up < m:
        sqrt_up += 1
    s = min(b, 2*n*sqrt_up)
    lower1, denom1 = 577*z, 1000*(b+s)
    lower2, denom2 = 577*z - 30*s, 1000*b
    if lower1*denom2 >= lower2*denom1:
        return lower1, denom1
    return lower2, denom2


def verify_chart(swapped, sign, small):
    directions = [u[2:] + u[:2] if swapped else u
                  for u in (SMALL if small else LARGE)]
    # The bound (2 sqrt(2) + 2 sqrt(3)) ||u|| < 11,000,000
    # follows from 2 sqrt(2) + 2 sqrt(3) < 7 and this check.
    assert all(49 * sum(t*t for t in u) < 11000000**2 for u in directions)
    counts = []

    def visit(x, y, t, n, depth):
        if depth == len(counts):
            counts.append(0)
```

```
        counts[depth] += 1
        n2 = n*n
        # A projective chart uses |z| <= 1. Exclude boxes disjoint from it.
        if max(abs(x)-1, 0)**2 + max(abs(y)-1, 0)**2 > n2:
            return
        # m/n2 bounds |z|^2 on the part of the box inside the chart.
        m = min(n2, (abs(x)+1)**2 + (abs(y)+1)**2)
        u, v = support_lower_bound(m, n, swapped, small)
        numerator, denominator = v-u, v+u
        r = n2 + t*t
        for a, b, c, d in directions:
            real = n*a + x*c + y*d
            imag = n*b + x*d - y*c
            det_real = n*c - x*a + y*b
            det_imag = n*d - x*b - y*a
            h = sign*((n2-t*t)*real + 2*n*t*imag) - 11000000*r
            if h > 0 and h*h*denominator > r*r*(det_real**2+det_imag**2)*numerator:
                return
        if depth >= 17:
            raise RuntimeError(f"Uncertified box: {(x,y,t,n)}")
        for i in (-1, 1):
            for j in (-1, 1):
                for k in (-1, 1):
                    visit(2*x+i, 2*y+j, 2*t+k, 2*n, depth+1)

    visit(0, 0, 0, 1, 0)
    return counts


def check_phase_approximation():
    """Verify a coordinate error of at most one in the scaled phase vectors."""
    pi_lo = Fraction(314159265358, 100000000000)
    pi_hi = Fraction(314159265360, 100000000000)
    error = Fraction(3**21, factorial(21))

    def enclosure(j, imaginary):
        if j > 3:
            lo, hi = enclosure(7-j, imaginary)
            return (-hi, -lo) if imaginary else (lo, hi)
        lo_arg, hi_arg = 2*j*pi_lo/7, 2*j*pi_hi/7
        low, high = -error, error
        for k in range(int(imaginary), 21, 2):
            a = lo_arg**k / factorial(k)
            b = hi_arg**k / factorial(k)
            if ((k-int(imaginary))//2) % 2:
                low -= b
                high -= a
            else:
                low += a
                high += b
        return low, high

    for rows, scale in ((SMALL, 1000000), (LARGE, 1210000)):
        for j, row in enumerate(rows):
            for i, rounded in enumerate(row):
                phase = (2 if i < 2 else 3)*j % 7
                factor = 1000000 if i < 2 else scale
                lo, hi = enclosure(phase, bool(i % 2))
                assert rounded-1 < factor*lo <= factor*hi < rounded+1


if __name__ == "__main__":
    check_phase_approximation()
    for small in (True, False):
        for swapped in (False, True):
            for sign in (1, -1):
                counts = verify_chart(swapped, sign, small)
                print("small =", small, "swapped =", swapped, "phase sign =", sign,
```

```
"maximum depth =", len(counts)-1, "boxes =", sum(counts))
```