Chord-Bisecting Trajectories and Arrival-Time Sweeps for Nadir-Camera Drone Swarms

Team Asclepius — NIDAR Rescueswarm

October 2026

Abstract

A swarm of N drones flies over a planar field as a single flexible line. Each drone carries a downward-looking (nadir) camera and sets its own altitude, which fixes the width of the ground strip it images. The goal is to image every point of an arbitrarily shaped field.

The problem began as a geometric question: find a smooth trajectory such that every chord perpendicular to it is bisected by it. A line of drones centred on such a trajectory would sweep the field symmetrically. We pose this as a constrained variational problem. We derive the first-hit derivatives, the Euler–Lagrange equations, a swept-area identity, and a no-fold condition. We show that the constraint reduces the problem to a line field, prove existence in convex fields, and give a desingularised boundary-pair ODE.

We then report what we used to solve these equations, and why each attempt failed: spline collocation with ray-cast residuals, boundary-pair collocation with shape continuation, and direct ODE shooting. The mathematics explains every failure. Finally, we describe the method now in use. The sweep is relaxed to an arrival-time field whose level sets are the formation: medial-axis chord time, principal-axis projection, a harmonic (Dirichlet-energy) field, and a geodesic (eikonal) field. A lane partition assigns ground to drones, and a monotone altitude repair guarantees 100% nadir coverage. Every equation is checked numerically.

Code: https://github.com/Avinashistoocool/asclepius
Demo: https://avinashistoocool.github.io/asclepius/
Reproduction: `paper/verify_derivations.py` (equations), `paper/figures.py` (figures), `paper/results.py` (tables).

Contents

1 Problem statement and notation **3**

2 The mathematics **4**
2.1 Derivatives of the first-hit distance . . . 4
2.2 Euler–Lagrange equations for $\mathcal{L} = f(p,\theta)\,|p'|$ . . . 4
2.3 The swept-area identity and the no-fold condition . . . 5
2.4 The constrained problem reduces to a line field . . . 5
2.5 Boundary-pair formulation . . . 6
2.6 Relaxation: arrival-time fields . . . 7
2.7 Nadir footprint and altitude . . . 7

3 What we used to solve the equations, and why it failed **8**
3.1 Attempt 1: spline collocation with ray-cast residuals . . . 8
3.2 Attempt 2: boundary-pair collocation with shape continuation . . . 9
3.3 Attempt 3: direct shooting on the pair ODE . . . 9
3.4 What did work for Problem 1: exact constructions . . . 10

4 What we use now **10**
4.1 Medial axis (the backbone) 10
4.2 The four arrival-time fields 11
4.3 Lanes, paths and altitudes 11

5 Results **12**

6 Numerical verification of the mathematics **13**

7 How to solve the equations further **13**

8 Limitations **14**

1 Problem statement and notation

Let $\Omega \subset \mathbb{R}^2$ be a bounded, simply connected field with piecewise-smooth boundary $\partial\Omega$. In the implementation, Ω is a simple polygon on a 600×600 canvas, and all lengths are in canvas units. This is a coverage path planning problem (Choset, 2001; Galceran and Carreras, 2013; Cabreira et al., 2019). It differs from boustrophedon decomposition (Choset and Pignon, 1998) and from Voronoi-partition coverage control (Cortés et al., 2004) in two ways: the swarm moves as one line, and every drone controls its own swath through its altitude.

Problem 1 (Bisecting trajectory)**.** Find a smooth curve $p : [0, L] \to \overline{\Omega}$, with both endpoints on $\partial\Omega$, whose perpendicular chord is bisected by p at every point. Optionally, among all such curves, minimise a cost $\int_0^L f \, \mathrm{d}s$.

Problem 2 (Swarm coverage)**.** N drones carry nadir cameras with fields of view (HFOV, VFOV). Find flight paths, headings and altitudes h_i such that every point of Ω lies inside at least one camera footprint, keeping $\max h_i$, the path lengths, and the workload imbalance small.

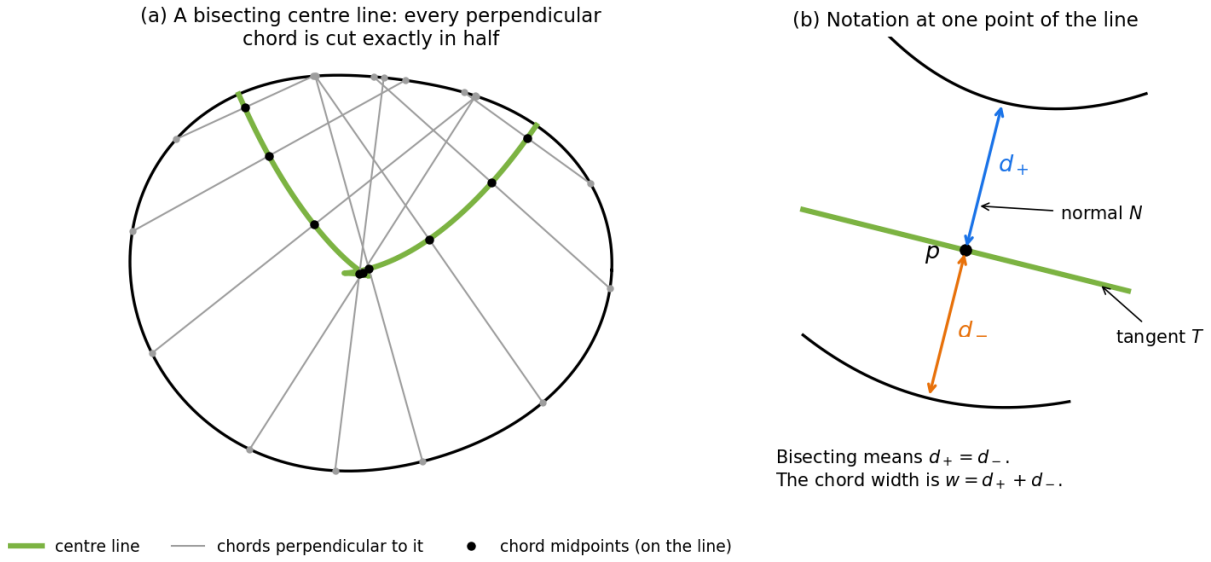


Figure 1: (a) A bisecting trajectory in a convex field: every perpendicular chord (grey) has its midpoint (black) on the curve (green). (b) Notation at one point.

Notation. Let $p(s)$ be parametrised by arc length s, and let J denote rotation by $+\pi/2$.

$T = p'$, $\quad N = JT$	unit tangent and unit normal of the trajectory
θ	angle of N, so $N = (\cos\theta, \sin\theta)$ and $\mathrm{d}N/\mathrm{d}\theta = JN = -T$
κ	signed curvature, with Frenet equations $T' = \kappa N$, $N' = -\kappa T$ (do Carmo, 1976); hence $\theta' = \kappa$
$d_\pm(p, \theta)$	distance from p to the *first* boundary point along $\pm N$
$x_\pm = p \pm d_\pm N$	the two chord endpoints
$\nu_\pm$	outward unit normal of $\partial\Omega$ at $x_\pm$
$w = d_+ + d_-$	chord width
$e = \frac{1}{2}(d_+ - d_-)$	signed offset of the chord midpoint from p (the *bisection error*)
$\tau_+ = \dfrac{\nu_+ \cdot T}{\nu_+ \cdot N}$, $\quad \tau_- = \dfrac{\nu_- \cdot T}{-\nu_- \cdot N}$	incidence slopes: tangents of the angle between the chord and the wall normal at each end
$\gamma(u)$	a parametrisation of $\partial\Omega$, used from Section 2.5 on

Problem 1 is the requirement

$$e(p(s), \theta(s)) = 0 \qquad \text{for all } s \in [0, L]. \tag{1.1}$$

Using *first* hits ensures that the chord through p stays within the connected part of $\Omega \cap (\text{line})$ that contains p. This matters when Ω is not convex.

2 The mathematics

2.1 Derivatives of the first-hit distance

Lemma 2.1 (Ray-hit derivatives)**.** *Let $x = p + d\,n$ be a transversal first hit ($\nu \cdot n \neq 0$) of the ray from p in the unit direction n, and let n be rotated by an angle α. Then*

$$\nabla_p d = -\frac{\nu}{\nu \cdot n}, \qquad \frac{\partial d}{\partial \alpha} = -\,d\,\frac{\nu \cdot Jn}{\nu \cdot n}. \tag{2.1}$$

Proof. Let ϕ be a local defining function of $\partial\Omega$ with $\nabla\phi \parallel \nu$. Differentiating $\phi(p + d\,n) = 0$ gives $\nu \cdot (\delta p + \delta d\,n + d\,\delta n) = 0$. Solve for δd, using $\delta n = Jn\,\delta\alpha$. □

Applying (2.1) with $n = \pm N$, decomposing $\nu_\pm$ in the frame (T, N), and using $\mathrm{d}N/\mathrm{d}\theta = -T$ gives

$$\begin{aligned} \nabla_p d_+ &= -N - \tau_+ T, & \nabla_p d_- &= N - \tau_- T, \\ \partial_\theta d_+ &= d_+ \tau_+, & \partial_\theta d_- &= -d_- \tau_-, \end{aligned} \tag{2.2}$$

$$\begin{aligned} \nabla_p w &= -(\tau_+ + \tau_-)\,T, & \nabla_p e &= -N - \tfrac{1}{2}(\tau_+ - \tau_-)\,T, \\ \partial_\theta e &= \tfrac{1}{2}(d_+ \tau_+ + d_- \tau_-). \end{aligned} \tag{2.3}$$

Two sanity checks follow. Sliding p along its own chord leaves the width unchanged and changes the offset at unit rate: $\partial_N w = 0$ and $\partial_N e = -1$. If the boundary is locally mirror-symmetric about the trajectory, then $\tau_+ = \tau_-$, so $\partial_T e = 0$ and a zero offset stays zero.

Remark 2.2 (Grazing singularity). Both expressions in (2.1) blow up as $\nu \cdot n \to 0$, that is, when the chord grazes the boundary. In a non-convex field the first hit then *jumps* to a different boundary piece, so $d_\pm$, w and e are only piecewise smooth in (p, θ). This is the root cause of the failures in Section 3.

2.2 Euler–Lagrange equations for $\mathcal{L} = f(p, \theta)\,|p'|$

Consider a cost depending on position and normal direction, written as a reparametrisation-invariant Lagrangian (Gelfand and Fomin, 1963):

$$\mathcal{L}(p, p') = f(p, N(p'))\,|p'|, \qquad N(p') = \frac{Jp'}{|p'|}. \tag{2.4}$$

Since $\partial N/\partial p' = J(I - TT^\top)/|p'|$, $J^\top = -J$, and $f_\theta = \nabla_N f \cdot \mathrm{d}N/\mathrm{d}\theta = -T \cdot \nabla_N f$,

$$\frac{\partial \mathcal{L}}{\partial p} = |p'|\,\nabla_p f, \qquad \frac{\partial \mathcal{L}}{\partial p'} = f\,T - (T \cdot \nabla_N f)\,N = f\,T + f_\theta\,N. \tag{2.5}$$

Substitute (2.5) into $\frac{\mathrm{d}}{\mathrm{d}s}\frac{\partial \mathcal{L}}{\partial p'} = \frac{\partial \mathcal{L}}{\partial p}$ and use $T' = \kappa N$, $N' = -\kappa T$, $\theta' = \kappa$. The result splits into two components.

Tangential component. This component holds identically. It is the Noether identity associated with reparametrisation invariance:

$$\frac{\mathrm{d}f}{\mathrm{d}s} + \kappa\,(T \cdot \nabla_N f) = T \cdot \nabla_p f. \tag{2.6}$$

Normal component. This is the curvature equation:

$$\boxed{\kappa\,(f + f_{\theta\theta}) = \partial_N f - \partial_T f_\theta} \tag{2.7}$$

where $\partial_N = N \cdot \nabla_p$ and $\partial_T = T \cdot \nabla_p$. For an isotropic cost ($f_\theta = 0$), (2.7) reduces to the weighted-geodesic (Snell) law $\kappa f = \partial_N f$. The factor $f + f_{\theta\theta}$ is the anisotropic *stiffness* known from crystalline and phase-boundary problems (Taylor, 1978; Gurtin, 1993). Equation (2.7) is a well-posed second-order ODE only where $f + f_{\theta\theta} > 0$.

Width cost. For $f = w$ we have $\partial_N w = 0$ by (2.3), so (2.7) becomes

$$\kappa\,(w + w_{\theta\theta}) = -\,\partial_T w_\theta, \tag{2.8}$$

where $w_\theta = d_+\tau_+ - d_-\tau_-$. The second derivative $w_{\theta\theta}$ involves the boundary curvature at $x_\pm$ through $\partial_\theta \nu_\pm$.

2.3 The swept-area identity and the no-fold condition

Parametrise the region swept by the chord as $x(s, r) = p(s) + r\,N(s)$ with $r \in [-d_-, d_+]$. Then $\partial_s x = (1 - \kappa r)\,T$ and $\partial_r x = N$, so the area element is $|1 - \kappa r|\,\mathrm{d}r\,\mathrm{d}s$.

Proposition 2.3 (Swept area)**.** *Wherever the chord does not fold,*

$$\frac{\mathrm{d}A}{\mathrm{d}s} = \int_{-d_-}^{d_+} (1 - \kappa r)\,\mathrm{d}r = w - \frac{\kappa}{2}(d_+^2 - d_-^2) = w\,(1 - \kappa e). \tag{2.9}$$

In particular, on a bisecting trajectory ($e \equiv 0$), $\int w\,\mathrm{d}s$ equals the swept area exactly*.*

So the bisection constraint is precisely the condition under which the line integral of chord width measures coverage. The chord sweeps monotonically (without folding) if and only if

$$\kappa\,d_+ < 1 \quad \text{and} \quad -\,\kappa\,d_- < 1, \tag{2.10}$$

that is, neither chord end passes the centre of curvature. A drone at offset r moves at speed $|1 - \kappa r|$ relative to the centre drone, so (2.10) also bounds the speed spread across the formation.

2.4 The constrained problem reduces to a line field

Adjoining (1.1) with a multiplier $\lambda(s)$ gives

$$\mathcal{L}_\lambda = (f + \lambda\,e)\,|p'|, \tag{2.11}$$

whose Euler–Lagrange equation is (2.7) with f replaced by $f + \lambda e$, together with $e = 0$. However, $e(p, \theta) = 0$ contains no derivative of θ, so it fixes the normal direction algebraically at every point. This is much stronger than an ordinary integral constraint.

Proposition 2.4 (Reduction)**.** *Bisecting trajectories are exactly the integral curves of the line field $\theta^\star(p)$ defined by $e(p, \theta^\star) = 0$. The only remaining freedom is the starting point $c \in \partial\Omega$, and the optimal-cost version of Problem 1 is the one-dimensional problem*

$$\min_{c \in \partial\Omega}\; J(c) = \int_{\Gamma_c} f\,\mathrm{d}s, \tag{2.12}$$

where Γ_c is the integral curve starting at c. The multiplier λ in (2.11) is the adjoint of the sensitivity $\partial\Gamma_c/\partial c$.

Proposition 2.5 (Existence of a bisecting direction)**.** *Let $p \in \Omega$, and suppose $d_\pm(p,\cdot)$ are continuous in θ. This holds, for example, if Ω is convex. Then some $\theta^\star \in [0,\pi)$ satisfies $e(p,\theta^\star) = 0$. If additionally $\partial_\theta e(p,\theta^\star) = \frac{1}{2}(d_+\tau_+ + d_-\tau_-) \neq 0$, then $\theta^\star$ is locally a C^1 function of p, and the integral curve through p is locally unique.*

Proof. Rotating by π swaps the two rays: $d_\pm(p,\theta+\pi) = d_\mp(p,\theta)$, so $e(p,\theta+\pi) = -e(p,\theta)$. A continuous function that changes sign on $[\theta_0,\theta_0+\pi]$ has a zero there. The second claim follows from the implicit function theorem and Picard–Lindelöf. □

In non-convex fields, the jumps of Remark 2.2 can skip over the root, so no bisecting direction exists. They can also create several roots, so the field branches.

2.5 Boundary-pair formulation

Instead of casting rays, parametrise the chord by its two endpoints on the boundary: $a = \gamma(u)$ and $b = \gamma(v)$. The chord vector is $q = b - a$ and the midpoint is

$$m = \tfrac{1}{2}(a+b), \qquad m' = \tfrac{1}{2}(\gamma'(u)\,u' + \gamma'(v)\,v'). \tag{2.13}$$

The chord is perpendicular to the midpoint path if and only if

$$q \cdot m' = 0 \iff (q\cdot\gamma'(u))u' + (q\cdot\gamma'(v))v' = 0, \tag{2.14}$$

so (u,v) follows the line field on the torus $\mathbb{T}^2 = (\partial\Omega)^2$:

$$\boxed{(u',v') \;\propto\; (q\cdot\gamma'(v),\; -q\cdot\gamma'(u))} \tag{2.15}$$

This requires no ray casting and is as smooth as γ. In convex fields, (2.15) is equivalent to Problem 1. In general, the first-hit property of a and b as seen from m must be checked separately. Related midpoint-of-boundary-pair constructions include smoothed local symmetries (Brady and Asada, 1984) and process-inferring symmetry axes (Leyton, 1987).

Equilibria. The point (u,v) is stationary for (2.15) if and only if $q\cdot\gamma'(u) = q\cdot\gamma'(v) = 0$. These are *double normals*: chords perpendicular to the boundary at both ends. Every smooth convex boundary has at least two, the diameter and the minimal-width chord, because each is critical for chord length.

Desingularisation at the boundary. The diagonal $u = v$ (a chord of zero length) is singular. Substitute $u = c - d$ and $v = c + d$. Expanding for small d gives $q\cdot\gamma'(c\pm d) = 2d|\gamma'|^2 \pm 2d^2\,\gamma'\cdot\gamma'' + O(d^3)$, so both components of (2.15) vanish to first order. Choosing the orientation in which d increases and dividing by $2d$, which is a time rescaling and leaves the curves unchanged, gives

$$c' = -\,d\,(\gamma'\cdot\gamma'') + O(d^2), \qquad d' = |\gamma'|^2 + O(d^2). \tag{2.16}$$

This field is regular and transversal to the diagonal.

Proposition 2.6 (Endpoints)**.** *Every boundary point c is the endpoint of exactly one bisecting curve of (2.15). The curve leaves $\partial\Omega$ orthogonally: $m \approx \gamma(c) + \frac{d^2}{2}\gamma''(c)$, which for an arc-length γ points along the curvature normal. Hence the two-point boundary-value problem for Problem 1 becomes an initial-value problem in c, consistent with (2.12).*

2.6 Relaxation: arrival-time fields

Replace the single trajectory by a scalar arrival-time field $T : \Omega \to [0, 1]$. The formation at time t is the front

$$\mathcal{F}_t = \{x \in \Omega : T(x) = t\}. \tag{2.17}$$

A point is swept once $t \geq T(x)$. Several connected components of $\mathcal{F}_t$ mean the swarm has split. With N drones spaced evenly along the front, the per-drone swath is $|\mathcal{F}_t|/N$. By the coarea formula (Evans, 2010),

$$\int_0^1 |\mathcal{F}_t| \, \mathrm{d}t = \int_\Omega |\nabla T| \, \mathrm{d}x, \tag{2.18}$$

so energies $\int_\Omega |\nabla T|^p$ control front lengths, and therefore altitudes.

Dirichlet principle (harmonic sweep). Impose $T = 0$ on a start cap Γ_0, $T = 1$ on an end cap Γ_1, and linear interpolation in arc length along the two side walls. Minimising $\int_\Omega \frac{1}{2}|\nabla T|^2$ yields the Euler–Lagrange equation

$$\Delta T = 0 \quad \text{in } \Omega. \tag{2.19}$$

More generally,

$$\min_T \int_\Omega |\nabla T|^p \implies \nabla \cdot (|\nabla T|^{p-2} \nabla T) = 0. \tag{2.20}$$

The case $p = 2$ is the harmonic sweep. As $p \to 1$, the energy tends to the total front length (2.18). As $p \to \infty$, the minimiser becomes the absolutely minimising Lipschitz extension, which minimises the steepest front (Aronsson, 1967). By the maximum principle, harmonic fronts have no interior extrema, so they never close into islands and never fold. Harmonic potentials are also used for navigation (Connolly et al., 1990).

Eikonal equation (geodesic sweep). If the front advances at unit normal speed, then $T(x)$ is the minimal travel time from Γ_0, i.e. the value function of the Lagrangian $|\dot{\xi}|$. It satisfies the Hamilton–Jacobi–Bellman (eikonal) equation (Bryson and Ho, 1975; Sethian, 1999)

$$|\nabla T| = \text{const}. \tag{2.21}$$

The shocks of the distance function to the *whole* boundary form the medial axis. This is the grassfire picture of the medial axis (Blum, 1967; Kimia et al., 1995).

2.7 Nadir footprint and altitude

A nadir camera at altitude h with heading $\hat{h}$ images the rectangle

$$\mathcal{R}(p, \hat{h}, h) = \Big\{x : \ |(x - p) \cdot \hat{h}| \leq h \tan \tfrac{\mathrm{VFOV}}{2}, \quad |(x - p) \cdot J\hat{h}| \leq h \tan \tfrac{\mathrm{HFOV}}{2}\Big\}. \tag{2.22}$$

Write a ground point in the body frame of the nearest sample p_k as (α, β) (along-track, across-track). Let Δ_k be the spacing between samples, and o the overlap margin. The minimal altitude at p_k is

$$h_k = (1 + o) \max_{x \mapsto k} \ \max\left(\frac{|\beta|}{\tan(\mathrm{HFOV}/2)}, \ \frac{(|\alpha| - \Delta_k/2)_+}{\tan(\mathrm{VFOV}/2)}\right). \tag{2.23}$$

Proposition 2.7 (Monotone coverage)**.** *For fixed paths and headings, the set of imaged points is non-decreasing in every h_k. Hence repairs that only raise altitudes never lose coverage. Moreover, full coverage is reached for a large enough uniform scale λh_k, and the minimal such λ can be found by bisection.*

Proof. $\mathcal{R}(p, \hat{h}, h)$ is nested increasing in h, and unions of increasing sets are increasing. □

3 What we used to solve the equations, and why it failed

All three attempts below target Problem 1 directly. The code is preserved in the repository history (commit `ae96e22`, folder `swarm_lab/legacy/`).

3.1 Attempt 1: spline collocation with ray-cast residuals

Formulation. Choose a chart direction D (a principal axis, a coordinate axis, or the axis between two long, facing boundary edges), with $D^\perp = JD$. Represent the trajectory as a graph over that chart,

$$p(t) = (x_0 + (x_1 - x_0)\,t)D + y(t)\,D^\perp, \qquad y(t) = \sum_{j=1}^{n} y_j\,B_{j,3}(t), \tag{3.1}$$

where $B_{j,3}$ are clamped cubic B-splines on uniform knots, so the curve is C^2, with $n \in \{20, 36, 56\}$. The unknowns are the control values y_j and the chart limits x_0, x_1.

Residuals. At $5n$ uniform points and $3n$ Chebyshev-clustered points t_i, compute the tangent from the analytic derivative $p'(t_i)$, cast rays along $\pm N$ to the first polygon edges, and impose

$$r_i = e(p(t_i), \theta(t_i)) = \tfrac{1}{2}(d_+ - d_-) = 0, \tag{3.2}$$

together with the endpoint residuals $20 \cdot \mathrm{dist}(p(0), \partial\Omega)$ and $20 \cdot \mathrm{dist}(p(1), \partial\Omega)$.

Solver. Bounded nonlinear least squares (trust-region reflective, `scipy.optimize.least_squares`, Virtanen et al., 2020) with an *analytic* Jacobian. The Jacobian comes from Lemma 2.1, applied edge by edge as ray–segment intersection derivatives. The solver was limited to 180 evaluations, with an 18 s budget over all charts and endpoint seeds.

Acceptance tests. The endpoint distance must be at most 10^{-5}. The maximum $|e|$ over 3074 off-grid samples must not exceed the tolerance (0.01 to 1 unit). Each Bézier piece must lie inside the polygon, which is checked by subdivision and convex-hull tests.

What happened. On most non-symmetric fields every chart failed. A typical message was:

```
No validated numerical trajectory was found ...
Endpoint residual 0.007304; midpoint residual 3.6480.
Endpoint residual 0.023611; midpoint residual 4.7275.
Endpoint residual 0.015916; midpoint residual 2.6701.
```

The chord midpoints remained 2.7 to 4.7 units from the curve.

Why it failed (explained by Section 2).

1. **Non-smooth residuals.** By Remark 2.2, e jumps whenever a ray switches polygon edges or boundary pieces. Gauss–Newton and trust-region methods assume C^1 residuals, so steps that cross a jump are rejected and the solver stalls.
2. **Over-determination.** By Proposition 2.4, bisecting curves form a one-parameter family fixed by the starting point. Pinning *both* endpoints while also imposing (1.1) at every collocation point leaves no solution, unless the two endpoints happen to lie on the same integral curve. Least squares then returns a compromise, hence the non-zero midpoint residual.
3. **Graph restriction.** The chart (3.1) forces p to be monotone in D. Bisecting curves that turn by more than 90° relative to D cannot be represented.

4. **Endpoint degeneracy.** As $p \to \partial\Omega$, the chord shrinks ($w \to 0$) and the rows of the Jacobian lose rank.

5. **Non-existence.** In strongly non-convex fields, a bisecting curve may not exist at all (Proposition 2.5 and Attempt 3).

3.2 Attempt 2: boundary-pair collocation with shape continuation

Formulation. To remove ray casting and the graph restriction, we used the pair form of Section 2.5. With $u = c - d$ and $v = c + d$, the centre phase $c(t)$ and the half-separation $d(t)$ were represented as cubic B-splines, with $d(0) = 0$ and $d(1) = \pi$, so that both chord ends start together and meet again on the far side. The residuals were collocated at 3 Gauss–Legendre nodes per knot interval:

$$r_i = \frac{q \cdot m'}{|q|\,\pi \sin(\pi t_i)} = 0, \qquad s_i = \frac{|m'|}{\pi \sin(\pi t_i)} - R = 0. \tag{3.3}$$

Here r_i is (2.14) normalised by the expected size near the ends, s_i is a regular-speed condition with a free radius R that prevents stationary points, and penalties enforce $0 < d < \pi$.

Solver. Least squares with an analytic Jacobian, combined with *shape continuation*: the boundary was deformed as $\gamma_\lambda = (1 - \lambda)\,\gamma_{\text{circle}} + \lambda\,\gamma$ from a circle ($\lambda = 0$, where every diameter is a solution) to the target field ($\lambda = 1$), with adaptive steps in λ. The seeds were the critical phases of the half-perimeter chord length, i.e. double normals (Section 2.5). The spline resolution was then refined through 48, 72, 104 and 144 controls. Finally, an interval-arithmetic enclosure verified $|e| \leq$ tolerance over every parameter subinterval. The budget was 80 s in total and 24 s per seed.

What happened. The method worked on convex and mildly non-convex smooth fields. On harder fields it ended with "No interval-verified trajectory was found in this bounded continuation search", after continuation stalls, time-outs, or failed verification. Each attempt took up to 80 s.

Why it failed.

1. **Wrong boundary condition.** Imposing $d(1) = \pi$ forces the two chord ends to meet exactly halfway round the boundary. By Proposition 2.6, the far endpoint is *determined* by the start c, and is generally not halfway round. The collocation problem is therefore again over-determined.

2. **Bifurcations along the continuation.** As λ increases, double normals are created and annihilated in pairs. At these points the Jacobian of (3.3) becomes singular, and the continuation branch turns back or ends (a fold bifurcation; Allgower and Georg, 2003).

3. **Recurrent orbits.** When the line field (2.15) has no orbit joining the diagonal to itself, no solution exists at any resolution (Attempt 3).

4. **First hits are not guaranteed.** In non-convex fields, a pair solution of (2.14) can pass behind a bump, so the first-hit condition fails and verification rejects an otherwise converged curve.

3.3 Attempt 3: direct shooting on the pair ODE

Method. Following Proposition 2.6, integrate (2.16) and then (2.15) as an initial-value problem from $(c, d) = (c_0, 2 \cdot 10^{-3})$. We used classical RK4 with step 0.004 (Hairer et al., 1993). Because (2.15) defines only a line field, each step chooses the sign that continues the previous direction. Integration stops when d returns to 0 (the far endpoint) or after 6000 steps. Adaptive integrators (`solve_ivp`) were tried first, but the unoriented field made them flip direction and stall.

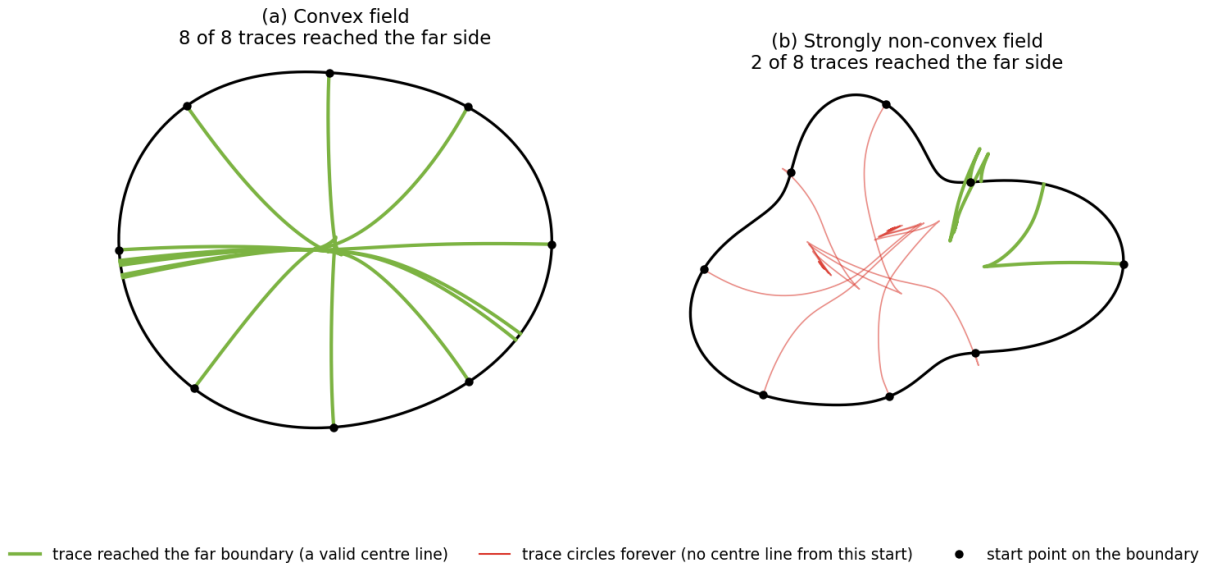


Figure 2: Shooting on (2.15) from 8 boundary points (black dots). (a) Convex field: 8 of 8 traces reach the far boundary and are valid bisecting curves. (b) Strongly non-convex field $r(t) = 150 + 30\cos 2t + 25\cos 3t + 18\cos 4t + 10\sin 2t - 15\sin 3t + 12\sin 4t$: only 2 of 8 reach the far side. The other 6 (red) are recurrent orbits on $\mathbb{T}^2$; after 6000 steps their path lengths are 1200–1400 units, against a field diameter of about 400.

What happened. This is the most faithful method, and it confirmed the theory (Figure 2). In convex fields every start gives a bisecting curve. In strongly non-convex fields most starts lead to recurrent orbits, on which the chord keeps rotating and never closes.

Conclusion. Bisecting trajectories through the whole field exist reliably only in convex or near-convex fields. This does not depend on the solver: it is a property of the line field. A practical planner for arbitrary fields must therefore relax Problem 1.

3.4 What did work for Problem 1: exact constructions

For fields with known symmetry, bisecting trajectories are available in closed form, and these were kept as exact presets:

- a diameter of a circle;
- the major axis of an ellipse, since reflection swaps the chord ends;
- any reflection axis, verified exactly with rational arithmetic on the polygon vertices;
- the mid-radius arc of an annular sector, where normal chords are radial.

4 What we use now

The planner solves Problem 2 with the relaxation of Section 2.6: it computes four arrival-time fields, splits the field among the drones, and plans altitudes.

4.1 Medial axis (the backbone)

The medial axis is the closure of the set of centres of maximal inscribed disks (Blum, 1967; Choi et al., 1997). We sample the boundary at about 1 unit (400 to 6000 samples) and compute the Voronoi diagram of the samples. Interior Voronoi edges converge to the medial axis as the sampling density increases (Brandt and Algazi, 1992). A ridge generated by samples g_1, g_2 is kept

only if the boundary arc between them exceeds 0.05× the perimeter (potential-residual pruning, Ogniewicz and Kübler, 1995). The *trunk* is the longest path, found with two Dijkstra passes (Dijkstra, 1959), in the minimum spanning tree of the largest component. It is smoothed by a cubic spline kept inside Ω and extended tangentially to $\partial\Omega$. Its two ends define the start cap Γ_0 and the end cap Γ_1.

4.2 The four arrival-time fields

All fields are computed on a 200×200 grid, and fronts are extracted at 90 levels by marching squares (Lorensen and Cline, 1987; van der Walt et al., 2014).

1. **Medial-axis chords.** $T(x) = s/L$ for x on the perpendicular chord of the trunk at arc length s. The medial axis is equidistant from the boundary, which is a *different* condition from bisection. The two coincide only where $\tau_+ = \tau_-$ in (2.3). The measured mean $|e|$ along the trunk is 0.0 (ellipse), 0.6 (C-bend), 10.4 (star), 13.1 (narrow neck) and 18.1 (random blob). The chords also fold wherever (2.10) fails.
2. **Principal-axis (lawnmower).** $T(x) = \langle x, a\rangle$, rescaled to $[0, 1]$, where a is the leading principal axis of the boundary samples. Fronts are straight lines, the continuous analogue of boustrophedon coverage (Choset and Pignon, 1998).
3. **Harmonic.** Solve (2.19) with the 5-point Laplacian and a sparse direct solver (Virtanen et al., 2020).
4. **Geodesic.** Solve (2.21) from Γ_0 by fast marching (Sethian, 1996).

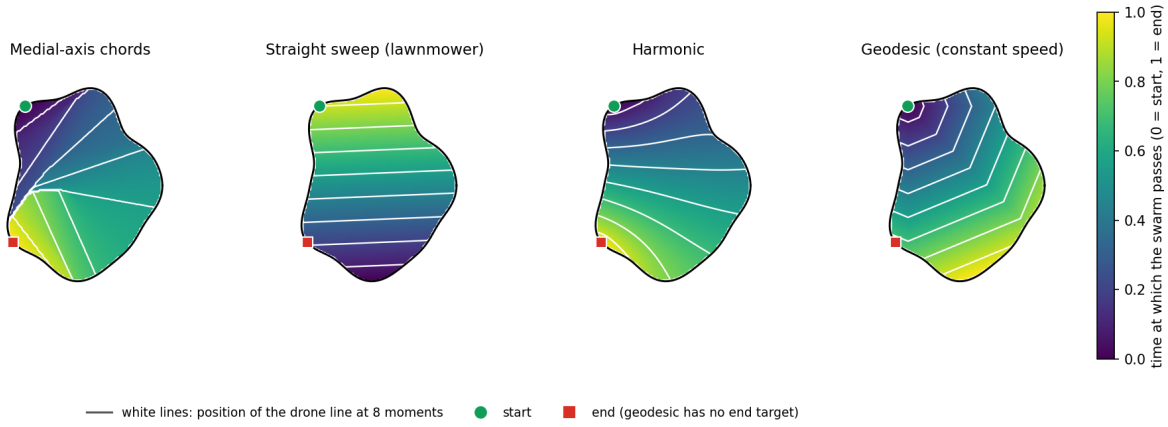


Figure 3: The four arrival-time fields on one field. Colour shows $T(x)$; white lines are the fronts $\mathcal{F}_t$ at 8 times. Green: start cap. Red: end cap.

4.3 Lanes, paths and altitudes

Lane partition. Each front is oriented consistently in time: its pieces are reordered and reversed so that their nearest points on the previous front have increasing lane coordinate. The front is then parametrised by normalised arc length $\sigma \in [0, 1]$. Drone i owns

$$\mathcal{A}_i = \Big\{x : \ \sigma(\pi(x)) \in \big[\tfrac{i}{N}, \tfrac{i+1}{N}\big)\Big\}, \tag{4.1}$$

where $\pi(x)$ is the nearest point on the nearest-in-time front. Drone i flies at $\sigma = (i + \frac{1}{2})/N$. A step longer than 25 units counts as a *transit jump* between pieces of a split front.

Altitudes and repair. Initial altitudes come from (2.23), followed by a 3-sample max filter; samples with no assigned cells are interpolated from their neighbours. Footprints (2.22) are then stamped at every sample, and densely between samples using the lower of the two altitudes. Any field cell left unimaged raises its owner's altitude near the nearest sample by $\times 1.15$ plus one cell. This is repeated for at most 8 rounds. By Proposition 2.7 the loop never loses coverage. In every experiment it ended with zero gaps within two rounds.

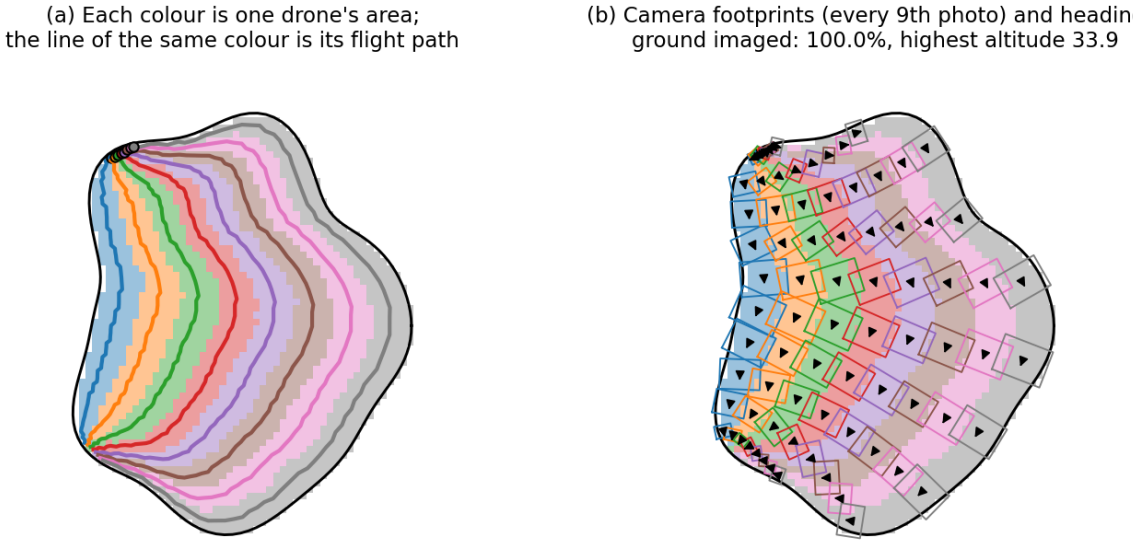


Figure 4: Harmonic plan with 8 drones. (a) Lane partition (4.1) with flight paths. (b) Camera footprints (2.22) for every 9th photo, with headings.

5 Results

The setup is 12 drones, HFOV $= 70°$, VFOV $= 55°$ and 10% overlap. All four planners reached **100%** nadir coverage with zero unimaged cells on every field. Altitudes are in canvas units; multiply by (field width in metres)/600 to convert to metres.

Table 1: Highest altitude needed for full coverage (lower is better; bold marks the best in each row).

Field	Medial-axis	Lawnmower	Harmonic	Geodesic
Ellipse	21.4	**21.3**	**21.3**	24.0
Star	140.7	58.4	**31.0**	77.0
C-bend	31.8	55.9	**15.0**	23.8
Narrow neck	93.5	84.1	**21.1**	159.2
Blob	27.9	28.1	**24.9**	79.1

Discussion. The harmonic field needs the lowest altitude on every field. This is consistent with (2.18)–(2.20): it minimises a front-length energy, and by the maximum principle its fronts never fold. The price is unequal drone areas, because fronts bunch up in narrow regions. The lawnmower and geodesic sweeps balance work best but split, and the geodesic front needs very high altitudes at cusps. The medial-axis chord sweep is reliable only on near-convex or tube-like fields, where (2.10) holds and $\tau_+ \approx \tau_-$. Computing all four plans takes 3 to 6 s in Python and 0.8 to 2.3 s in the browser port.

Table 2: Area balance (smallest $|\mathcal{A}_i|$ divided by largest; 1 is ideal), and the share of medial-axis chord steps that fold, i.e. violate (2.10).

Field	Medial-axis	Lawnmower	Harmonic	Geodesic	Medial-axis folds
Ellipse	**0.94**	0.81	0.79	0.86	0%
Star	0.12	0.80	0.41	**0.81**	23%
C-bend	0.64	**0.73**	0.48	**0.73**	0%
Narrow neck	0.16	**0.85**	0.32	**0.85**	42%
Blob	0.16	**0.87**	0.39	0.84	52%

6 Numerical verification of the mathematics

All checks are in `paper/verify_derivations.py`.

Check	Equation	Result
Ray-hit derivative in position, 200 random rays in an ellipse	(2.1)	4.2×10^{-8}
Ray-hit derivative in direction	(2.1)	7.9×10^{-8}
Tangential Euler–Lagrange identity, anisotropic f	(2.6)	3.4×10^{-9}
Pair ODE: chord $\perp$ midpoint velocity (cosine)	(2.14)	8.1×10^{-15}
Integrated midpoint path $\perp$ chord (finite RK4 steps)	(2.14)	3.8×10^{-4}
Discrete Laplace residual of the harmonic field	(2.19)	2.7×10^{-15}
Harmonic field minimises the discrete Dirichlet energy	(2.19)	no decrease
Geodesic field: $\lvert\nabla T\rvert$ constant (relative IQR)	(2.21)	0

The unit tests `swarm_lab/test_medial.py` also assert that the lanes partition the field among all N drones, and that nadir coverage is 100% (Proposition 2.7).

7 How to solve the equations further

1. **Solve Problem 1 by shooting in c with first-hit checks.** Integrate (2.16)→(2.15) from many c, using an oriented adaptive integrator. Stop at $d \to 0$ (success) or on Poincaré recurrence (failure). Verify first hits along each accepted orbit, and minimise $J(c)$ in (2.12) by golden-section search. The gradient $J'(c)$ follows from the variational equation of (2.15), or equivalently from λ in (2.11). Seed the search on both sides of each double normal.

2. **Solve (2.7) as a two-point BVP** when a cost-optimal curve is wanted without exact bisection. Use the state (p, θ) with $p' = T(\theta)$ and $\theta' = \kappa$. Suitable solvers are collocation with residual control (Kierzenka and Shampine, 2001) (`scipy.integrate.solve_bvp`) or multiple shooting (Keller, 1968; Ascher et al., 1995). Continue from a symmetric shape with pseudo-arclength continuation (Allgower and Georg, 2003), monitoring $f + f_{\theta\theta} > 0$.

3. **Regularise the non-smoothness.** Replace the first hit $d = \min_j r_j$ by a soft minimum $-\varepsilon \log \sum_j e^{-r_j/\varepsilon}$, and let $\varepsilon \to 0$ along the continuation. Alternatively, pose the problem as an NLP with complementarity constraints for chord switches and solve it with an interior-point method (Wächter and Biegler, 2006) through CasADi (Andersson et al., 2019). Pontryagin's principle (Pontryagin et al., 1962; Bryson and Ho, 1975) reproduces (2.11) and adds turn-rate limits.

4. **Minimise the worst altitude.** The objective $\max_t |\mathcal{F}_t|/N$ is a homotopy height (Chambers and Letscher, 2009). Approximate it by (2.20) with increasing p, solved by Newton

iteration under continuation in p.

5. **Balance harmonic workloads** by solving $\nabla \cdot (a(x)\nabla T) = 0$ with a chosen to equalise $|\nabla T|$. This interpolates between (2.19) and (2.21).
6. **Minimal altitude.** Bisect on a uniform scale (Proposition 2.7), or solve an LP / set-cover relaxation over the footprint–cell incidence matrix.
7. **Real flight.** Add turn-rate and climb-rate limits, terrain following from a DEM, and georeferencing.

8 Limitations

- Whole-field bisecting trajectories exist only under the hypotheses of Proposition 2.5. On strongly non-convex fields they are generically absent (Figure 2).
- The medial axis is sampled, so the pruning threshold and the choice of trunk affect the start and end caps.
- Coverage is guaranteed on a 200×200 grid; features smaller than one cell (3 units) are not resolved.
- Battery, wind and flight dynamics are not modelled.
- The browser port smooths the trunk with a Gaussian filter and uses 8-connected Dijkstra instead of fast marching; its statistics agree with the Python version to within a factor of 3.

References

E. L. Allgower and K. Georg. *Introduction to Numerical Continuation Methods*. SIAM, 2003.

J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl. CasADi: a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36, 2019.

G. Aronsson. Extension of functions satisfying Lipschitz conditions. *Arkiv för Matematik*, 6:551–561, 1967.

U. M. Ascher, R. M. M. Mattheij, and R. D. Russell. *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*. SIAM, 1995.

H. Blum. A transformation for extracting new descriptors of shape. In W. Wathen-Dunn, editor, *Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press, 1967.

M. Brady and H. Asada. Smoothed local symmetries and their implementation. *The International Journal of Robotics Research*, 3(3):36–61, 1984.

J. W. Brandt and V. R. Algazi. Continuous skeleton computation by Voronoi diagram. *CVGIP: Image Understanding*, 55(3):329–338, 1992.

A. E. Bryson and Y.-C. Ho. *Applied Optimal Control*. Hemisphere, 1975.

T. M. Cabreira, L. B. Brisolara, and P. R. Ferreira Jr. Survey on coverage path planning with unmanned aerial vehicles. *Drones*, 3(1):4, 2019.

E. W. Chambers and D. Letscher. On the height of a homotopy. In *Proceedings of the 21st Canadian Conference on Computational Geometry*, pages 103–106, 2009.

H. I. Choi, S. W. Choi, and H. P. Moon. Mathematical theory of medial axis transform. *Pacific Journal of Mathematics*, 181(1):57–88, 1997.

H. Choset. Coverage for robotics: a survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31:113–126, 2001.

H. Choset and P. Pignon. Coverage path planning: the boustrophedon cellular decomposition. In *Field and Service Robotics*, pages 203–209. Springer, 1998.

C. I. Connolly, J. B. Burns, and R. Weiss. Path planning using Laplace's equation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2102–2106, 1990.

J. Cortés, S. Martínez, T. Karataş, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255, 2004.

E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

M. P. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, 1976.

L. C. Evans. *Partial Differential Equations*. American Mathematical Society, 2nd edition, 2010.

E. Galceran and M. Carreras. A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, 61(12):1258–1276, 2013.

I. M. Gelfand and S. V. Fomin. *Calculus of Variations*. Prentice-Hall, 1963.

M. E. Gurtin. *Thermomechanics of Evolving Phase Boundaries in the Plane*. Oxford University Press, 1993.

E. Hairer, S. P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer, 2nd edition, 1993.

H. B. Keller. *Numerical Methods for Two-Point Boundary-Value Problems*. Blaisdell, 1968.

J. Kierzenka and L. F. Shampine. A BVP solver based on residual control and the MATLAB PSE. *ACM Transactions on Mathematical Software*, 27(3):299–316, 2001.

B. B. Kimia, A. R. Tannenbaum, and S. W. Zucker. Shapes, shocks, and deformations I: the components of two-dimensional shape and the reaction-diffusion space. *International Journal of Computer Vision*, 15:189–224, 1995.

M. Leyton. Symmetry-curvature duality. *Computer Vision, Graphics, and Image Processing*, 38(3):327–341, 1987.

W. E. Lorensen and H. E. Cline. Marching cubes: a high resolution 3D surface construction algorithm. *ACM SIGGRAPH Computer Graphics*, 21(4):163–169, 1987.

R. L. Ogniewicz and O. Kübler. Hierarchic Voronoi skeletons. *Pattern Recognition*, 28(3):343–359, 1995.

L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko. *The Mathematical Theory of Optimal Processes*. Interscience, 1962.

J. A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.

J. A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 2nd edition, 1999.

J. E. Taylor. Crystalline variational problems. *Bulletin of the American Mathematical Society*, 84(4):568–588, 1978.

S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, and the scikit-image contributors. scikit-image: image processing in Python. *PeerJ*, 2:e453, 2014.

P. Virtanen et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, 17:261–272, 2020.

A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.