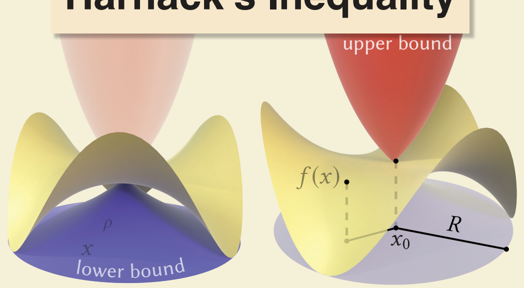
Ray Tracing Harmonic Functions

Mark Gillespie Carnegie Mellon University
Denise Yang Carnegie Mellon University, and Pixar Animation Studios
Mario Botsch TU Dortmund University
Keenan Crane Carnegie Mellon University

Harnack's Inequality



Let f be a positive harmonic function on a ball of radius R:

$$\underbrace{\frac{1-\rho/R}{(1+\rho/R)^{n-1}} f(x_0)}_{\text{lower bound}} \le f(x) \le \underbrace{\frac{1+\rho/R}{(1-\rho/R)^{n-1}} f(x_0)}_{\text{upper bound}}$$

Safe Step Size (in 3D)

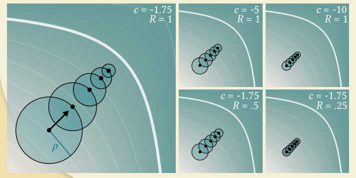
If we take a step of size ρ starting from x_0 we will never step past the f^* level set:

$$\rho := \frac{R}{2}\left|a + 2 - \sqrt{a^2 + 8a}\right|, \quad \text{where} \quad a = \frac{f(x_0)}{f^*}$$

Algorithm

Algorithm 1 HarnackTrace($\mathbf{r}_0, \mathbf{v}, f^*, f(\mathbf{x}), R(\mathbf{x}), c(\mathbf{x}), \varepsilon, t_{\max}$)

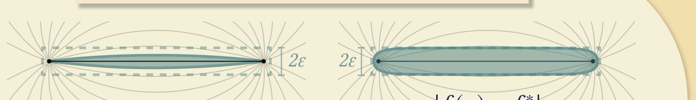
```
t ← 0
while t < t_max do
    r_t ← r_0 + t v                              ▹ current point along ray
    if |f(r_t) − f*| ≤ ε‖∇f(r_t)‖ then           ▹ stopping condition
        return t
    if f* ≤ c(r_t) then                          ▹ if f* lies below the lower bound...
        ρ ← R(r_t)                               ▹ ...we can safely take the maximum step of R
    else
        a ← (f(r_t) − c(r_t))/(f* − c(r_t))      ▹ otherwise, shift and...
        ρ ← ½ R(r_t) |a + 2 − √(a² + 8a)|        ▹ ...compute safe step size
    t ← t + ρ                                    ▹ take step
return −1                                        ▹ ray does not hit surface
```



We use the formulas above to find a safe step size ρ. Since these inequalities apply only to positive functions, we "shift" f within a local ball to get a safe step size.

The only challenge is determining ball radii R and lower bounds c. The largest step size will be achieved by using (i) the tightest lower bound c on f and (ii) the largest radius R: the step size ρ approaches the maximum step size R as c approaches $f(\mathbf{r}_t)$, and simultaneously, the step size grows in proportion to the ball radius R. However, for harmonic functions (which are saddle-like everywhere) larger balls inevitably contain more negative values. To achieve efficient computation, one must hence balance the choice of R and c.

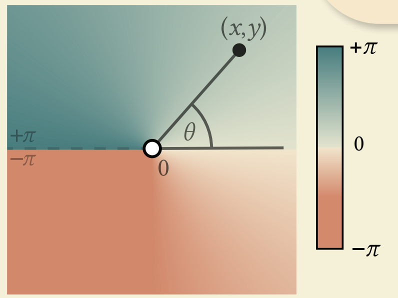
Termination Condition



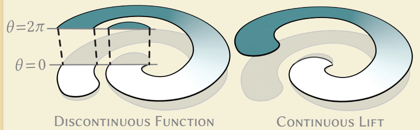
$$|f(\mathbf{x}) - f^*| < \varepsilon \qquad \frac{|f(\mathbf{x}) - f^*|}{|\nabla f(\mathbf{x})|} < \varepsilon$$

If $f(x)$ is a signed distance function, then terminating intersection queries when $|f(x) - f^*| < \varepsilon$ ensures that x is within ε of the chosen level set. But, when $f(x)$ is a general function, this condition loses its geometric meaning and produces an uneven profile along the target surface *(left)*. We can obtain a more meaningful stopping condition using the gradient to relate changes in function value to changes in position *(right)*.

Angle-Valued Functions



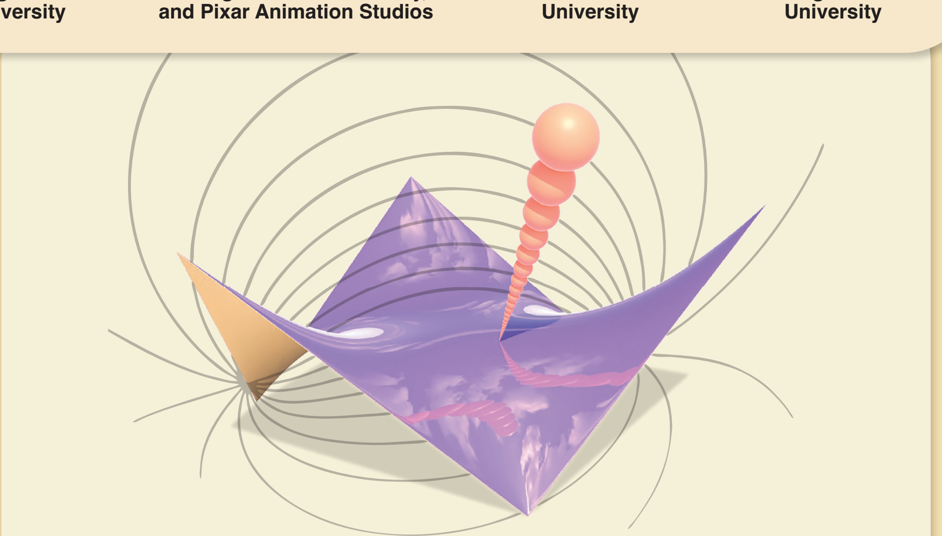
Harmonic functions can be *angle-valued* and exhibit singularities. The function $\theta(x,y) = \text{atan2}(y, x)$ is a model example: it jumps by 2π at $y=0$, and is singular at $x=y=0$. Using angle-valued functions allows us to represent implicit surfaces with boundary



Although angle-valued functions appear discontinuous when plotted in the range $[0, 2\pi)$, they can always be lifted to a continuous harmonic function on any simply-connected domain. Importantly, we never find this lift explicitly: its mere existence ensures that the bound holds.

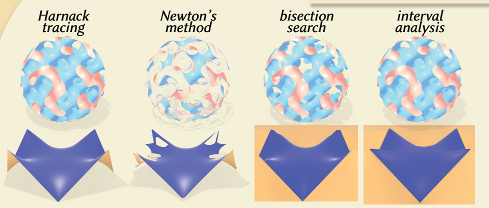
Algorithm 2 TraceAngleValued($\mathbf{r}_0, \mathbf{v}, f^*, f(\mathbf{x}), R(\mathbf{x}), c(\mathbf{x}), \varepsilon, t_{\max}$)

```
t ← 0
while t < t_max do
    r_t ← r_0 + t v                                  ▹ current point along ray
▹ Find the two level set values bracketing the current value of f
    f_0 ← (f(r_t) − f*)/(2π)
    f_− ← 2π⌊f_0⌋ + f*
    f_+ ← 2π⌈f_0⌉ + f*
▹ Stop if close to either surface (§3.1.2)
    if min(f(r_t) − f_−, f_+ − f(r_t)) ≤ ε‖∇f(r_t)‖ then
        return t
▹ Compute step size bound for each of the two closest level sets
    a_− ← (f(r_t) − c(r_t))/(f_− − c(r_t))
    a_+ ← (f(r_t) − c(r_t))/(f_+ − c(r_t))
    ρ_− ← ½ R(r_t) |a_− + 2 − √(a_−² + 8a_−)|
    ρ_+ ← ½ R(r_t) |a_+ + 2 − √(a_+² + 8a_+)|
    ρ ← min(ρ_−, ρ_+)                                ▹ Take the smaller of the two steps
    t ← t + ρ
return −1                                            ▹ ray does not hit surface
```



We introduce a ray tracing algorithm for a novel class of surfaces defined by level sets of harmonic functions. Here we directly visualize a nonplanar polygon which has no well-defined inside or outside—and hence cannot be represented by an ordinary implicit function or SDF. Isolines depict a 2D slice of the harmonic function; spheres show conservative *Harnack bounds* along a ray. Note the smooth reflection lines, even near edges where the function is highly singular.

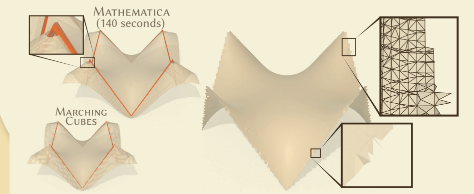
Comparisons



General-purpose root finders may fail to find the *closest* intersection, yielding incorrect occlusions or even gaps if they converge to intersections outside of the visible area.

Ray marching is a simple ray tracing technique, but the algorithm can leave gaps by "tunneling" through the surface, especially near singularities.

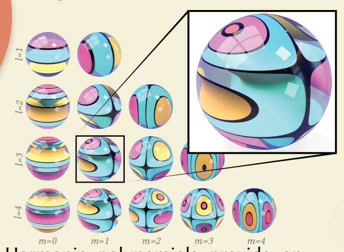
We can also run sphere tracing using any purported Lipschitz bound. However, when no valid Lipschitz bound exists, we obtain artifacts near singularities



Running marching cubes "out of the box" on many of our example problems yields unsightly artifacts, especially around singularities *(bottom left)*. More sophisticated adaptive methods, like Mathematica's ContourPlot3D, surrer from similar artifacts *(top left)*. One can attempt to filter out the extraneous faces, e.g. by evaluating f at the barycenter of each face, but doing so still leaves behind a noisy surface, especially near singularities *(right)*.

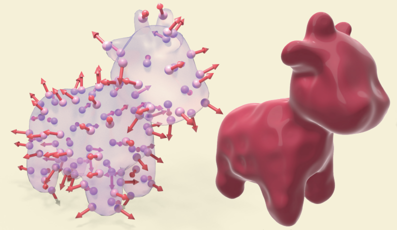
Examples

Spherical Harmonics



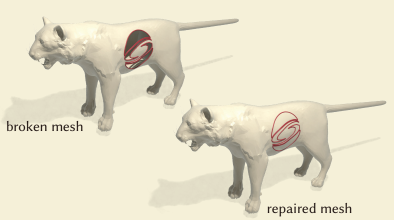
Harmonic polynomials provide an elementary example of harmonic functions. When restricted to the sphere, these polynomials describe the *spherical harmonics*. We visualize each spherical harmonic by restricting the level sets of the associated polynomial to the unit ball.

Point Clouds (Poisson Reconstruction)



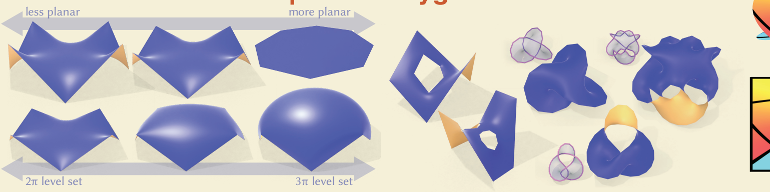
Given an oriented point cloud *(left)*, we can directly visualize an interpolating surface *(right)*. This procedure effectively shows the result of running the Poisson surface reconstruction algorithm of Kazhdan et al. [2006], without requiring any volumetric meshing or linear solves.

Mesh Repair (Generalized Winding Number)



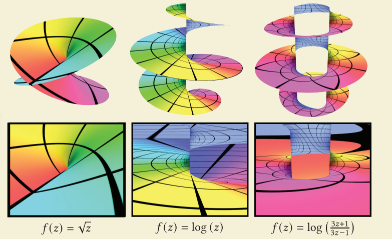
Given a surface mesh with imperfections such as holes *(left)*, we can directly visualize the repaired surface defined via the generalized winding number *(right)*, reproducing the example from Jacobson et al. [2013, Figure 1].

Nonplanar Polygons



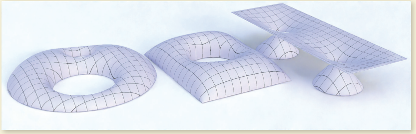
One can define the geometry of a nonplanar polygon to be a level set of its solid angle function, which is harmonic. This definition provides well-behaved geometry even when the boundary is highly nonplanar, and varies smoothly as the boundary changes *(top left)*. By taking different level sets, one can adjust the convexity/concavity of the interpolating geometry *(bottom left)*. The definition even automatically applies to difficult cases like polygons with holes or knotted boundaries *(right)*.

Riemann Surfaces



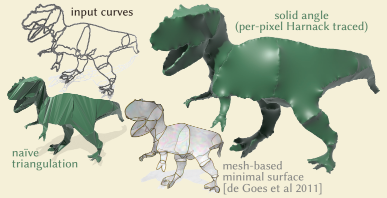
Riemann surfaces are central objects of study in complex analysis. We can use Harnack tracing to render the surfaces associated to several standard complex functions, showing both the intersection with the unit ball *(top)*, and a camera view of the surface *(bottom)*.

Architectural Geometry



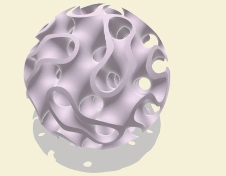
Here we use Harnack tracing to directly visualize a special class of grid shells used in architecture, reproducing examples from Adiels et al. [2022] figures 11, 12, and 22 (resp.).

Surface Exoskeletons



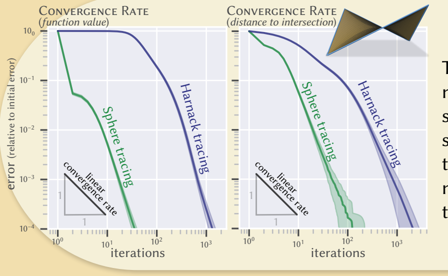
Given a sparse "exoskeleton" approximating a surface *(top left)* we can directly ray trace an interpolating surface *(top right)*. The result is both higher-quality than the naïve triangulation used by most mesh viewers *(bottom left)*, and simpler to compute than optimizing a mesh-based minimal surface, as done by de Goes et al. [2011].

Beyond Harmonic Functions



The gyroid surface, commonly used in 3D manufacturing, is neither a signed distance function nor a harmonic function. However, it can easily be extended to a harmonic function in 4D. By ray tracing a "3D slice" of this function, we can directly visualize it via Harnack tracing.

Convergence



To compare the convergence rates of Harnack tracing and sphere tracing, we measured the cost of sphere tracing a mesh of the 2π level set of solid angle for a sample curve (top right), as well as the cost of Harnack tracing the level set. The plots show the rate at which the function value approaches the target value *(left)*, as well as the rate at which the time approaches the optimal time *(right)*. Empirically, both methods appear to converge faster than the theoretically-guaranteed linear rate, though sphere tracing still requires significantly fewer iterations than Harnack