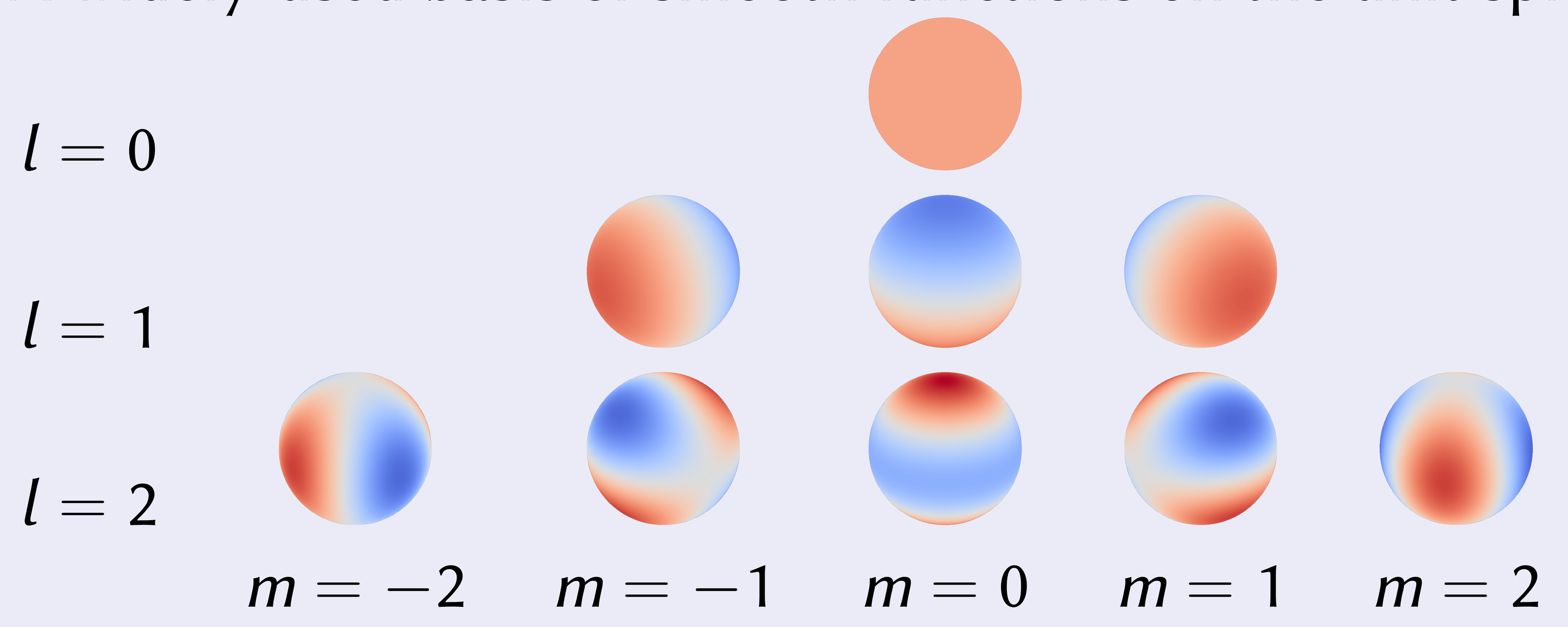
Derivatives of Spherical Harmonics

Christoph Peters, High-Performance Graphics 2025, June 24

TU Delft

Spherical Harmonics (SH)

A widely-used basis of smooth functions on the unit sphere.



Real SH is a product of a:

- Normalization constant K_l^m,
- Cosine or sine factor $c_m(x, y)$ or $s_m(x, y)$,
- Associated Legendre polynomial $Q_l^m(z)$.

$$Y_l^m(x, y, z) := K_l^{|m|} Q_l^{|m|}(z) \begin{cases} \sqrt{2} s_{|m|}(x, y) & \text{if } m < 0, \\ 1 & \text{if } m = 0, \\ \sqrt{2} c_m(x, y) & \text{if } m > 0. \end{cases}$$

Cosine/Sine Factors

Definition:

$$c_m := \cos(m\varphi), \quad s_m := \sin(m\varphi), \quad \text{where } \varphi := \arctan(y, x).$$

Recurrence formula (assuming $x^2 + y^2 = 1$):

$$c_0 := 1, \qquad s_0 := 0,$$
$$c_{m+1} := x c_m - y s_m, \qquad s_{m+1} := y c_m + x s_m.$$

The derivatives turn out to be

$$\frac{\partial c_m}{\partial x} = m c_{m-1}, \qquad \frac{\partial c_m}{\partial y} = -m s_{m-1},$$
$$\frac{\partial s_m}{\partial x} = m s_{m-1}, \qquad \frac{\partial s_m}{\partial y} = m c_{m-1}.$$

No additional computation at all!

Associated Legendre Polynomials

Recurrent definition of $Q_l^m(z)$ is

$$Q_0^0(z) = 1,\ Q_l^l(z) = (1 - 2l) Q_{l-1}^{l-1}(z),\ Q_{l+1}^l(z) = (2l + 1) z Q_l^l(z),$$

$$Q_l^m(z) = \frac{2l - 1}{l - m} z Q_{l-1}^m(z) - \frac{l + m - 1}{l - m} Q_{l-2}^m(z).$$

(Disregarding a factor of $\sqrt{1 - z^2}^m = \sqrt{x^2 + y^2}^m$, which cancels with the normalization of c_m, s_m.)

Associated Legendre polynomials satisfy

$$Q_l^m(z) := (-1)^m \frac{\partial^m Q_l^0(z)}{\partial z^m}.$$

Therefore, the derivatives are

$$\frac{\partial^k}{\partial z^k} Q_l^m(z) = (-1)^k Q_l^{m+k}(z).$$

No additional computation at all!

Derivatives of SH

Examples of partial derivatives for $m > 0$:

$$\frac{\partial Y_l^m}{\partial x}(x, y, z) = \sqrt{2} K_l^m m c_{m-1}(x, y) Q_l^m(z),$$
$$\frac{\partial Y_l^m}{\partial z}(x, y, z) = -\sqrt{2} K_l^m c_m(x, y) Q_l^{m+1}(z),$$
$$\frac{\partial^3 Y_l^m}{\partial y \partial z^2}(x, y, z) = -\sqrt{2} K_l^m m s_{m-1}(x, y) Q_l^{m+2}(z).$$

Code Generation for SH Evaluation

My open source Python script generates efficient code for:

- C, C++, GLSL, HLSL or Python,
- Arbitrarily many bands,
- Optionally only odd or even bands,
- Derivatives of arbitrary order (gradient, Hessian, ...).
- Tested using finite differences.

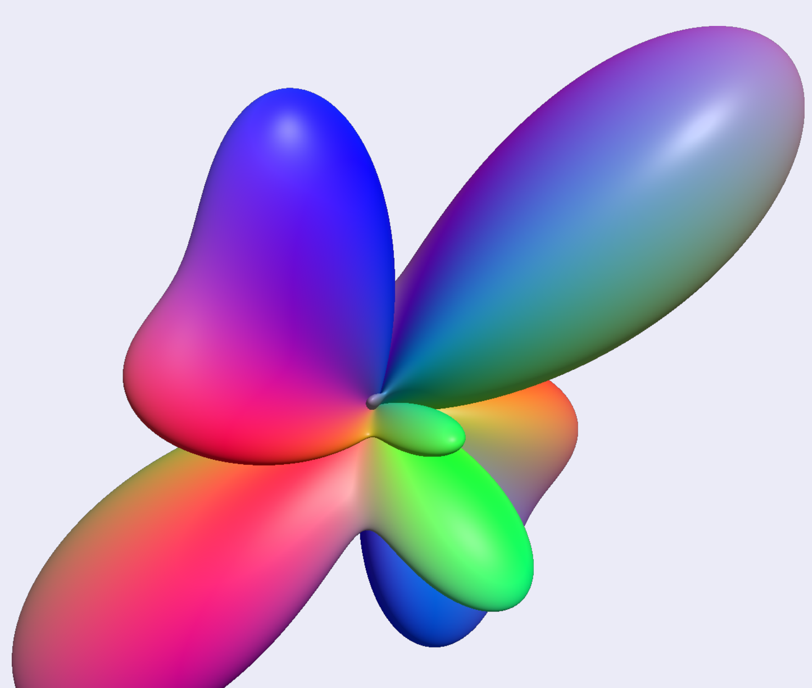
Example: GLSL code for SH band 0, 2 with gradient.

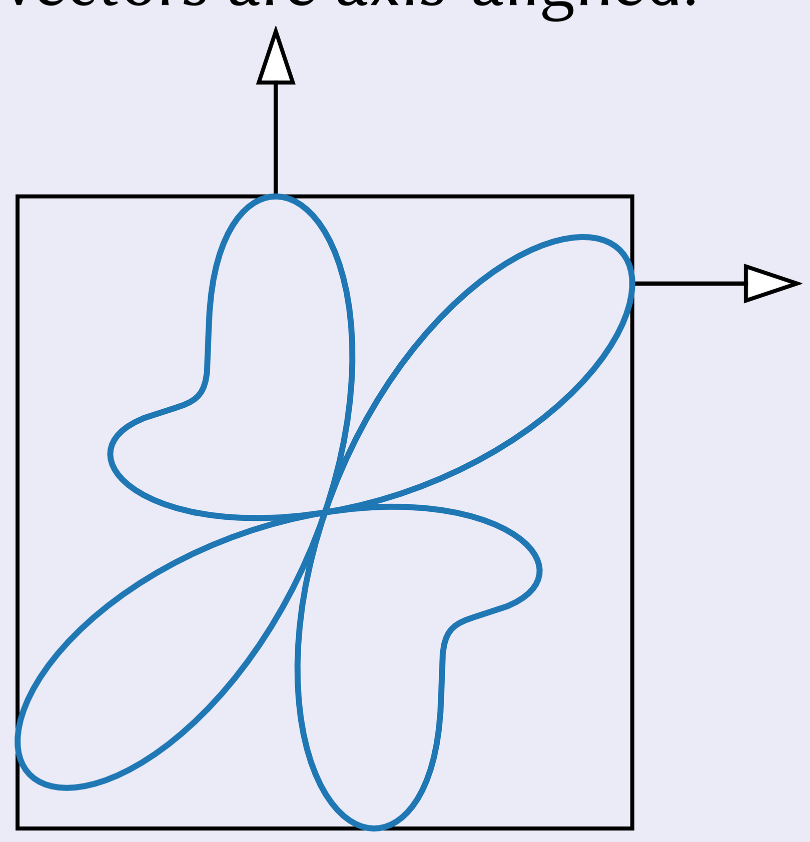
```
void eval_sh_0_2(out float out_shs[6], out vec3 out_grads[6], vec3 point) {
    float x, y, z, z2, c0, s0, c1, s1, d, a;
    x = point[0];  y = point[1];  z = point[2];  z2 = z * z;
    c0 = 1.0; s0 = 0.0;
    d = 0.282094792;
    out_shs[0] = d;
    a = z2 - 0.333333333;
    d = 0.946174696 * a;
    out_shs[3] = d;
    c1 = x; s1 = y;
    d = -1.092548431 * z;
    out_shs[2] = s1 * d;
    out_shs[4] = c1 * d;
    out_grads[2][0] = s0 * d;
    out_grads[4][0] = c0 * d;
    out_grads[2][1] = c0 * d;
    out_grads[4][1] = -s0 * d;
    d = 1.892349392 * z;
    out_grads[3][2] = d;
    c0 = x * c1 - y * s1;  s0 = y * c1 + x * s1;
    d = 0.546274215;
    out_shs[1] = s0 * d;
    out_shs[5] = c0 * d;
    d = 1.092548431;
    out_grads[1][0] = s1 * d;
    out_grads[5][0] = c1 * d;
    out_grads[1][1] = c1 * d;
    out_grads[5][1] = -s1 * d;
    d = -1.092548431;
    out_grads[2][2] = s1 * d;
    out_grads[4][2] = c1 * d;
    out_grads[0][0] = 0.0;  out_grads[0][1] = 0.0;  out_grads[0][2] = 0.0;
    out_grads[1][2] = 0.0;  out_grads[3][0] = 0.0;
    out_grads[3][1] = 0.0;  out_grads[5][2] = 0.0;
}
```

momentsingraphics.de/HPGPoster2025.html

Example Application: Computing AABBs for SH Glyphs [1]

SH glyph = points of a sphere, scaled by an SH polynomial.
At points on the AABB, normal vectors are axis-aligned.





Solution using Newton's method in 2D.
Requires gradient and Hessian of SH.

References

C. Peters, T. Patel, W. Usher, and C.R. Johnson.
Ray tracing spherical harmonics glyphs.
In *Vision, Modeling, and Visualization*. Eurographics, 2023.