Rendering Notes

Aaron Trent

April 15, 2017

Collection of equations and algorithms for use in the Combustion Engine, as well as general reference for developers wishing to understand the final equations and how I came to them.

This is a **work in progress** document. I repeat myself several times and the graphics are not entirely up to par with my standards yet.

Contents

Part I

Theory

1 Physically Based Rendering (PBR)

Physically Based Rendering (or Physically Based Shading) is simply the technique of using real-world-like parameters to represent a material, for example **Roughness**, **Anisotropy**, **Metalness**, **Index of Refraction** (η), and so forth.

1.1 Physical Correctness

By itself, PBR does not guarantee a realistic image. The equations and algorithms used to render the image must be physically correct. That is, they must obey the laws of physics to properly simulate light.

2 Microfacet Theory

Microfacet theory describes the surface of an object as being composed of incredibly tiny mirrors called microfacets. The microfacets are oriented in different directions based on surface roughness. The more smooth the surface, the more mirrors line up with the surface normal, forming a more mirrored macroscopic surface. Rougher surfaces are equivalent to the microfacets being oriented in ever more random directions, up until a point of entirely rough surfaces where the microfacets are totally randomly oriented and scatter light in all directions.

Since microfacets themselves are less intuitive, they are usually abstracted to become microgeometry, which is the microscopic ridges and crevices of a surface, where roughness and smoothness are as we know them traditionally.

At this abstraction level, it makes more sense that rougher surfaces are blurrier, while smoother surfaces are shinier.

3 Fresnel Effect

"Everything has Fresnel!" - Internet

If you haven't heard this before, you have now. Everything has Fresnel. However, what is Fresnel?

Put simply, the Fresnel effect determines how much light is reflected off of or transmitted into a surface. Reflected light forms specular highlights, while transmitted light becomes diffuse reflections or subsurface scattering.

For transparent materials, like glass, transmitted light is also refracted and passes all the way through it.

The Fresnel effect is governed by three primary variables (excluding any vectors):

1. The Internal Index of Refraction (η_i)
2. The Extinction Coefficient (k)
3. The External Index of Refraction (η_o)

where the internal IOR (η_i) is the index of refraction of the material itself, the extinction coefficient (k) is effectively how conductive the material is, and the external IOR (η_o) is the index of refraction of the material the object is contained in.

Normally, the external IOR is approximately 1.0, because the IOR of Air is approximately 1.0, so many forms of the Fresnel equations simply omit it.

However, if you wish to render an object as it appears underwater, the external IOR must be set to approximately 1.33 (the IOR of Water), or it will appear incorrect.

For metals, the interal IOR and extinction coefficient form a complex IOR in the form:

$$\eta + ik$$

where the color of the metal can be derived from it's wavelength-specific IORs.

4 Wavelength Response

4.1 Computer Color

Color in computer graphics is primarily RGB. Despite HSV/HSL/YUV/etc. color representations, RGB remains the primary way to represent color in computers. The reason for this is that the human eye only perceives Red, Green and Blue colors for its three cone type. The science of human color perception is quite in-depth and not necessary here, but the gist of it is that humans see a range of wavelengths for each primary color, with perception strength in roughly Gaussian curve shapes.

4.2 RGB Wavelengths and Human Eyesight

Here are the wavelengths used in Combustion:

$$\text{Red} = 620\text{nm} - 740\text{nm}$$
$$\text{Green} = 495\text{nm} - 570\text{nm}$$
$$\text{Blue} = 450\text{nm} - 495\text{nm}$$

where the human eye's response to individual wavelengths is this:
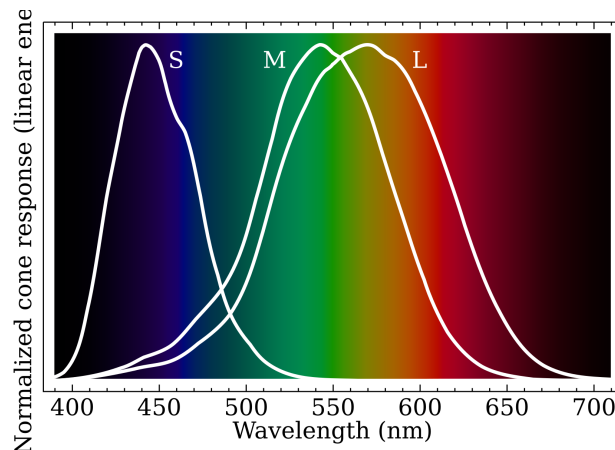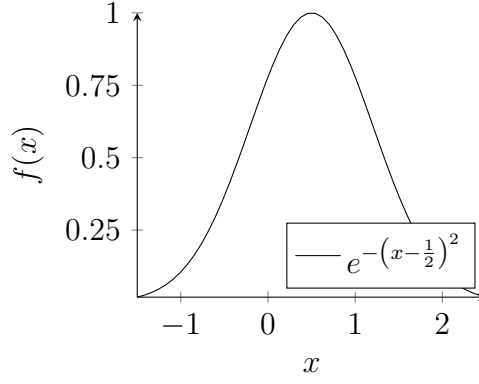

Figure 1: Human Eye Wavelength Response

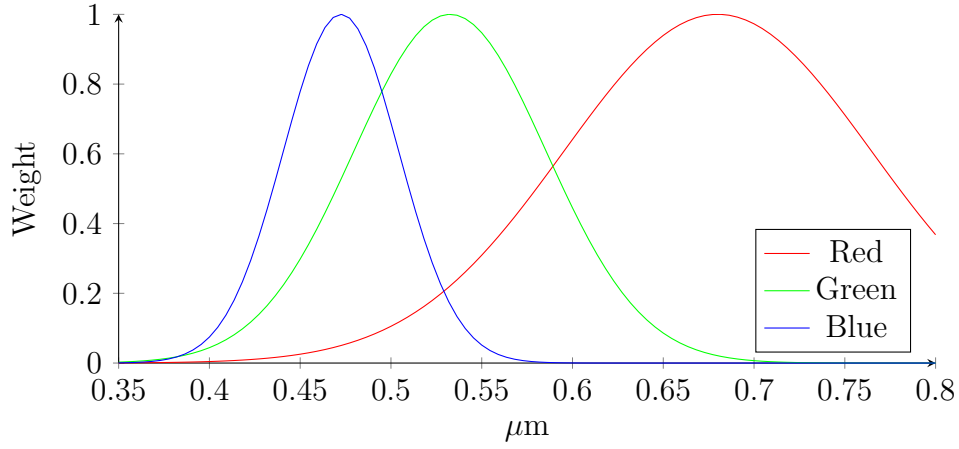which can be approximated using a modified Gaussian function like this:

to form:

Figure 2: Weighted RGB Wavelength Response

which is similar to our perception of pure Red, Green and Blue. Note that they don't line up exactly, because there is a difference between wavelength and RGB color.

So by using this simple form to our advantage, we can perform a weighted average of any wavelength dependent optical properties like so:

$$V = \frac{\int_{w_{min}}^{w_{max}} f(x) \, p(x) \, dx}{(w_{max} - w_{min}) \int_{w_{min}}^{w_{max}} p(x) \, dx}$$

where $p(x)$ is:

$$p(x) = e^{-\left(\frac{x - w_{min}}{w_{max} - w_{min}} - \frac{1}{2}\right)^2}$$

or the original distribution scaled to the wavelength range like so:

$$p(\frac{x - w_{min}}{w_{max} - w_{min}}) = e^{-\left(x - \frac{1}{2}\right)^2}$$

Part II

Light Equations

5   Specular Reflectance

5.1   Cook-Torrance

The Cook-Torrance equation is the core component for specular reflections today.

Variables for equations:

N = unit surface normal

V = view vector

L = light vector

$H = \dfrac{V + L}{\|V + L\|}$ = halfway vector and microfacet normal

η = Index of Refraction

α = surface roughness

It should be noted that these are ONLY specular reflections. That is, light which is reflected without diffusing into the material surface. For diffuse materials and how to mix diffuse and specular reflections, keep reading.

5.1.1 R. Cook and K. Torrance 1981

Original Definition:

$$f_{spec} = \frac{DFG}{\pi\,(\omega_o \cdot n)\,(\omega_i \cdot n)}$$

where ω_o is the view direction and ω_i is the incoming light direction.

G, the **Geometric Attenuation** function, which describes self-shadowing due to microfacet orientation and roughness, is defined as:

$$G_{Cook-Torrance} = min\left\{1, \frac{2\,(N \cdot H)\,(N \cdot V)}{(V \cdot H)}, \frac{2\,(N \cdot H)\,(N \cdot L)}{(V \cdot H)}\right\}$$

D, the **Microfacet Distribution** function, is the Beckmann distribution, which is defined as:

$$D_{Beckmann} = \frac{\exp\left(\frac{(n \cdot m)^2 - 1}{\alpha^2 (n \cdot m)^2}\right)}{\pi \alpha^2 (n \cdot m)^4}$$

where α is the surface roughness from $[0, 1]$, but it has been simplified to:

F is the **Fresnel** term, which was originally defined as:

$$F = \frac{1}{2}\frac{(g - c)^2}{(g + c)^2}\left\{1 + \frac{[c\,(g + c) - 1]^2}{[c\,(g - c) + 1]^2}\right\}$$

where

$$c = \cos\theta = V \cdot H$$
$$g^2 = \eta^2 + c^2 - 1$$
$$\eta = \frac{1 + \sqrt{F_0}}{1 - \sqrt{F_0}}$$

and F_0 is defined as:

$$F_0 = \left\{\frac{\eta - 1}{\eta + 1}\right\}^2$$

where η is the Index of Refraction of the material. Depending on how you do things, F_0 can be avoided entirely by just using the raw IOR η.

5.1.2 Walter 2007 (GGX)

Walter 2007 updates the Cook-Torrance model with a better set of equations that are commonly referred to as GGX.

GGX Cook-Torrance form:

$$f_{spec} = \frac{DFG}{4\,(\omega_o \cdot n)\,(\omega_i \cdot n)}$$

where G is split into two functions like so:

$$G = G_l(\omega_i)G_l(\omega_o)$$

Here, the Geometric Attenuation term is broken into two functions, collectively referred to as the Smith shadow-masking function. Each partial geometric attenuation function is the same, but is passed separate vector for light and view directions, then multiplied together.

Also note the replacement of π with just 4. I often see these used interchangeably around the internet, but they should remain distinct depending on how it is used. GGX works better with 4.

For the GGX equations, G_l is defined as:

$$G_l(v) = \chi^+ \left(\frac{\omega_v \cdot \omega_g}{\omega_v \cdot \omega_m} \right) \frac{2}{1 + \sqrt{1 + \alpha^2 \tan^2 \theta_v}}$$

where ω_v is the view direction, ω_g is the unit surface normal, ω_m is the halfway vector, and ω_v is the incoming or outgoing direction.

The GGX Distribution function (D) is defined as:

$$D = \chi^+ (n \cdot m) \frac{\alpha^2}{\pi \cos^4 \theta (\alpha^2 + \tan^2 \theta)^2}$$

or simplified to:

$$D = \chi^+ (n \cdot m) \frac{\alpha^2}{\pi \big((n \cdot m)^2\,(\alpha^2 - 1) + 1\big)^2}$$

with an anisotropic form defined as:

$$D = \chi^+ (n \cdot m) \frac{1}{\pi \alpha_x \alpha_y} \frac{1}{\left(\frac{(x \cdot m)^2}{\alpha_x^2} + \frac{(y \cdot m)^2}{\alpha_y^2} + (n \cdot m)^2 \right)^2}$$

as you can see, it takes two α (roughness) values and two anisotropy direction values (x and y) to control anisotropy.

6 Diffuse Reflectance

What light isn't directly reflected in the form of specular highlights is transmitted into the surface. However, most surfaces simply scatter the light below the surface and re-emit it in the form of colored diffuse light.

6.1 Lambertian Diffuse Model

For perfectly smooth surfaces, diffuse reflectance can be approximated by a single operation:

$$f_{diffuse} = N \cdot L = \|N\|\|L\| \cos \theta$$

where $\|N\|$ and $\|L\|$ are the magnitudes of the vectors. Consequently, because it is just $\cos \theta$ (when using normalized vectors), Lambertian diffuse has a reflectance curve like this:
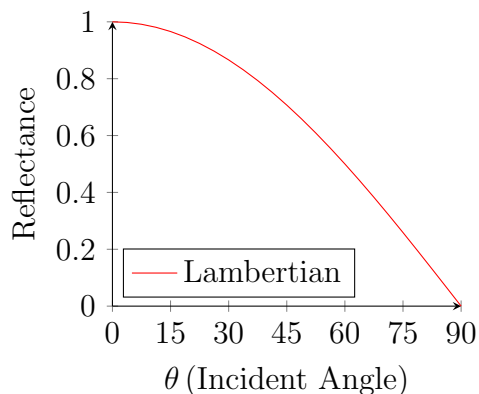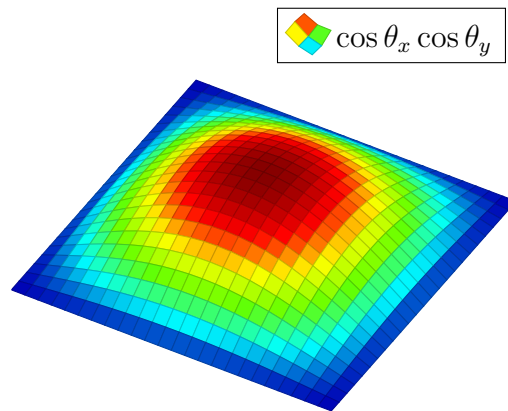

Figure 3: Lambertian reflectance

Figure 4: 3D plot

Lambertian is the most commonly used diffuse model mostly because of its simplicity. It's usually computed anyway for the specular component, so it's effectively free.

However, it does not account for surface roughness, so things get kind of weird looking in PBR heavy workflows.

You can see this effect more in Figure 5 on the next page.

6.2 Oren-Nayar Diffuse Model

Oren-Nayar is a "newer" (1994) set of equations based on empirical data that more accurately simulates surfaces of varying roughness.
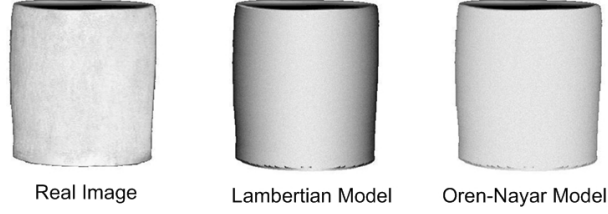

Figure 5: Comparison of Lambertian and Oren-Nayar models

As you can see, Oren-Nayar does a much better job than traditional Lambertian diffuse. It more accurately simulates rough surfaces, and is therefore much more useful in modern PBR workflows.

The full form of the Oren-Nayar equation is defined as:

$$L_r = \frac{\rho}{\pi} \cdot \cos\theta_i \cdot (A + (B \cdot \max[0, \cos(\phi_i - \phi_r)] \cdot \sin\alpha \cdot \tan\beta)) \cdot E_0$$

where

$$A = 1 - 0.5\frac{\sigma^2}{\sigma^2 + 0.57}$$
$$B = 0.45\frac{\sigma^2}{\sigma^2 + 0.09}$$
$$\alpha = \max(\theta_i, \theta_r)$$
$$\beta = \min(\theta_i, \theta_r)$$

and ρ is the surface albedo (diffuse absorption), σ is the surface roughness, and ϕ_i and ϕ_r are the azimuth angles
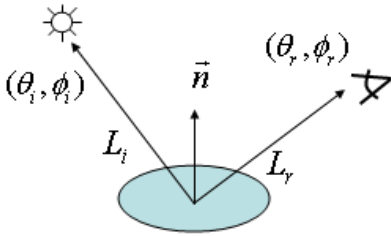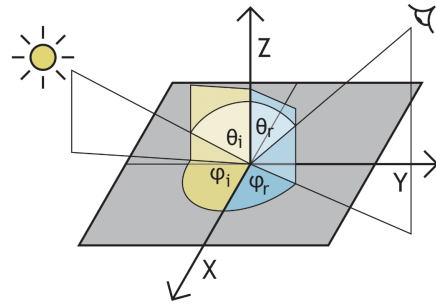

Figure 6: Reflectance Diagram

Figure 7: Angle Diagram

7 Fresnel Equations

The Fresnel equations determine how much light is reflected off of a surface, versus how much light is transmitted into it. Therefore, they are critical for correctly rendering any realistic material.

Most of these equations were taken from Memo on Fresnel equations by Sébastien Lagarde

The full Dielectric-Conductor Fresnel equations are as follows:

$$
a^2 = \frac{1}{2\eta_i^2} \left(\sqrt{\left(\eta_t^2 - k_t^2 - \eta_i^2 \sin^2 \theta \right)^2 + 4\eta_t^2 k_t^2} + \eta_t^2 - k_t^2 - \eta_i^2 \sin^2 \theta \right)
$$

$$
b^2 = \frac{1}{2\eta_i^2} \left(\sqrt{\left(\eta_t^2 - k_t^2 - \eta_i^2 \sin^2 \theta \right)^2 + 4\eta_t^2 k_t^2} - \eta_t^2 + k_t^2 + \eta_i^2 \sin^2 \theta \right)
$$

$$
R_s = \frac{a^2 + b^2 - 2a \cos \theta + \cos^2 \theta}{a^2 + b^2 + 2a \cos \theta + \cos^2 \theta}
$$

$$
R_p = R_s \frac{a^2 + b^2 - 2a \sin \theta \tan \theta + \sin^2 \theta \tan^2 \theta}{a^2 + b^2 + 2a \sin \theta \tan \theta + \sin^2 \theta \tan^2 \theta}
$$

where

$$
\eta_t = \text{Surface IOR } (t \text{ for transmitted})
$$

$$
k_t = \text{Surface Extinction Coefficient}
$$

$$
\eta_i = \text{External IOR } (i \text{ for incoming})
$$

Furthermore, this gives us two reflectance values, R_s and R_p, for parallel and perpendicular polarization of light. If light polarization is not important, these can be average together into a single reflectance value like so:

$$
R = \frac{R_s + R_p}{2}
$$

or if polarization is important, these values should be linearly interpolated like so:

$$
R = (1 - \omega_t) R_s + \omega_t R_p
$$

where ω_t is the polarity weight in the domain $[0, 1]$, so that 0 is fully perpendicular polarized light, and 1 is fully parallel polarized light. A weight of $\frac{1}{2}$ is equal to the first form where they are averaged together.

As mentioned in the Fresnel Effect theory section, the external IOR plays a very important role for scene that are not in air or vacuum. In air and/or vacuum, the external IOR is usually assumed to be 1 so the equation can be simplified, but for underwater scenes or something more exotic this parameter is absolutely necessary.

7.1 Metals

Metals are unique because they have no diffuse reflections, and therefore their entire color is derived from the Fresnel effect on varying wavelengths of light.

As you can see below in Figures 8 and 9, Gold behaves very differently depending on wavelength. It absorbs blue-ish light and reflects more red-ish light, producing its characteristic yellow color.
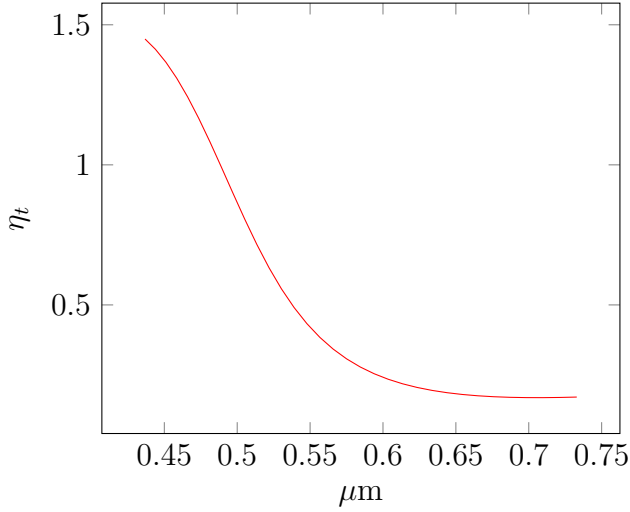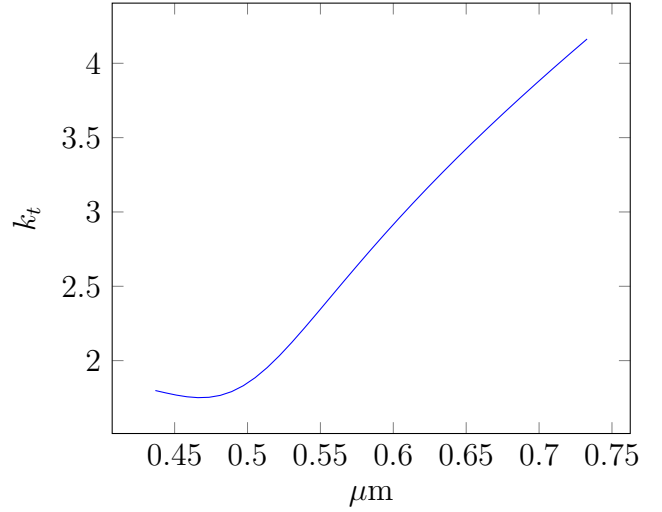

Figure 8: Gold η_t

Figure 9: Gold k_t

Consequently, because metals have different behaviors for different wavelengths of light, their color is created by the varying reflectance values at a given angle.

As you can see below in Figures 10 and 11, Gold and Copper are given their unique colors by only the Fresnel effect, and reflecting more white light at greater grazing angle.
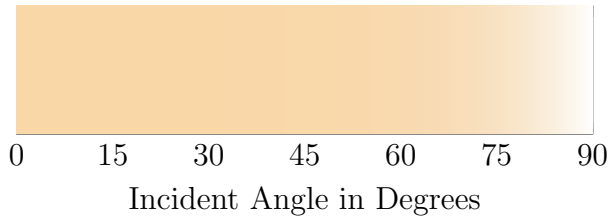

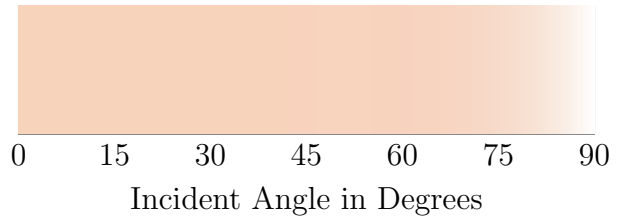Figure 10: Gold (Unweighted)

Figure 11: Copper (Unweighted)

You can further see how it affects individual spectra of light in Figure 12 on the next page.
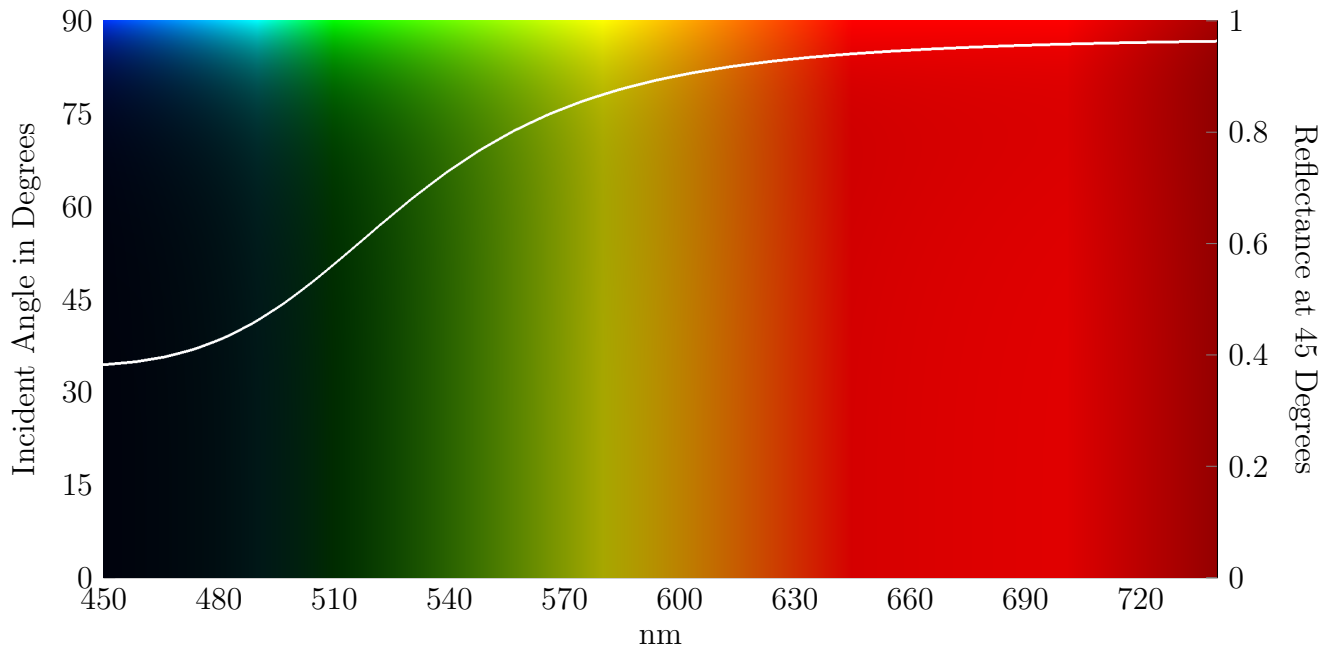
Figure 12: Gold Spectral Reflectance (R^3 for emphasis) at varying incident angles. The white line indicates reflectance at 45 Degrees.

It's important to understand the relationship all of these have together so you aren't tempted to ignore them for performance. For most of these, precomputed texture lookup tables can be used instead of computing the Fresnel effect for every fragment.

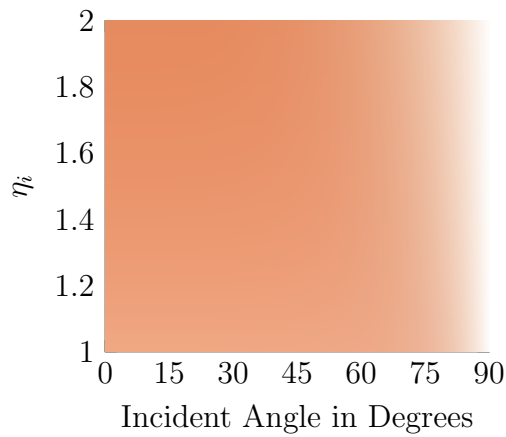An example of such a lookup table for Copper is below in Figure 13.

Figure 13: Copper Reflectance LUT with varying angles and external η_i

Where the LUT can be accessed with the following GLSL code:

```
vec3 R = texture(TODO);
```
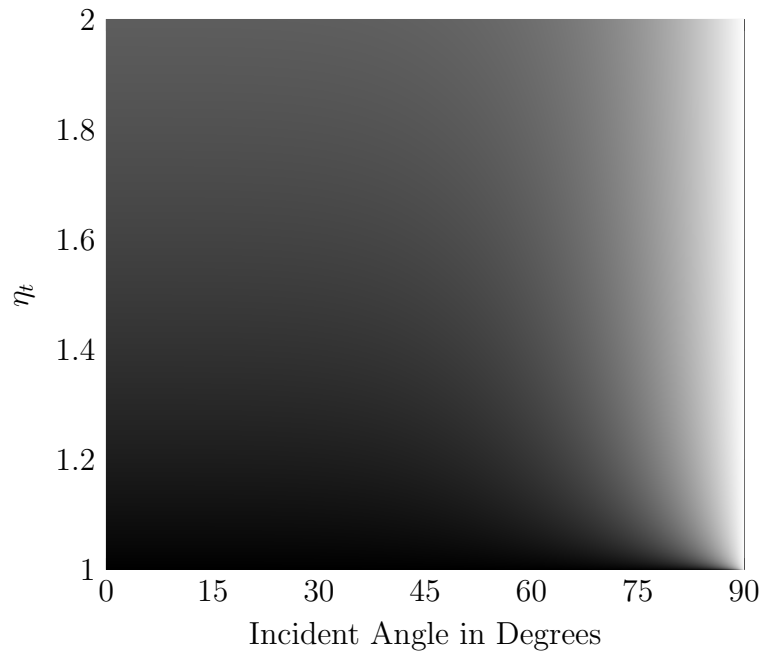
7.2 Dielectrics

Figure 14: Dielectric Reflectance for varying η_t and incident angles