CGI2020 manuscript No.
(will be inserted by the editor)

Constant-Time Energy-Normalization for the Phong Specular BRDFs

Ian Mallett and Cem Yuksel

Abstract The Phong and Modified Phong specular BRDFs, although of limited physical basis, are nevertheless some of the simplest BRDFs exhibiting glossy and specular qualities to understand and to implement, making them useful for validation and teaching. Unfortunately, although it is well-known how to make these BRDFs *conserve* energy (that is, never gain energy), making them energy-*normalized* (that is, never lose nor gain energy) is far more difficult. Lesser-known algorithms exist, but require the specular exponent n to be integer-valued, and have $O(n)$ runtime cost. We express these algorithms as mathematical formulae and generalize to the real-valued specular exponent case. We then simplify and optimize to finally attain an algorithm that is $O(1)$. Energy normalization makes the Phong BRDFs more physically plausible and therefore both more practically and theoretically useful—and our improvements allow for this energy normalization to be done efficiently and without arbitrary limitations.

1 Introduction

The importance of a reflectance model, i.e. the bidirectional reflectance distribution function (BRDF), to be *energy-conserving* is well-understood in computer graphics. An energy-conserving BRDF ensures that the material does not reflect more light than it receives—that is, the material is permitted to absorb some of the light it receives, but it cannot add light.

Ian Mallett
University of Utah School of Computing
imallett@cs.utah.edu

Cem Yuksel
University of Utah School of Computing
cem@cemyuksel.com

In the absence of any absorption, real materials still exhibit some darkening in their appearance based on the view angle. When light hits a surface, the Fresnel equations define the portion that is reflected or transmitted. Collectively, the Fresnel coefficient, multiplied by a *distribution function* that describes the distribution of reflectance, gives the BRDF. For a material to be physically realistic, the distribution function must integrate to one—that is, any darkening in the BRDF should be due to the Fresnel term, not due to artifacts of the distribution; when there is no absorption, the distribution function must scatter *all* light it receives. We refer to this criterion as *energy normalization*. Note that energy normalization is a stronger criterion than energy conservation, and is arguably part of the definition of the BRDF itself [12].

Energy normalization has received less attention in computer graphics than energy conservation, as its solution is usually more difficult. In production (such as AAA video games, special effects, and feature animation), microfacet BRDFs [2,15] are common. Energy normalization for a microfacet BRDF can be physically interpreted, mostly, as accounting for multiple scattering off of secondary microfacets, and convenient solutions have only become available recently [5,7,9,16].

In this paper, we concern ourselves with calculating the correct energy normalization terms for the specular component BRDFs of the Phong [14] and Modified Phong [10] reflectance models. These BRDFs are still widely implemented and used and, even if microfacet models are more popular, Phong specular BRDFs remain valuable for their simplicity and intuition. These BRDFs are not physically based, but energy normalization makes them behave in a physically sensible way, which may also make them more-appealing in a contest with more-modern BRDF models. Nonetheless, we view the contribution of our work as more-theoretical

in nature, rather than necessarily having practical importance for realistic graphics productions.

Arvo [1] introduced a powerful framework which can be applied for expressing BRDFs, including (approximately) modern BRDFs [8]. However, it provides an *exact* solution for the Phong specular BRDFs. Unfortunately, this algorithm only works for integer values of the specular exponent n and it has linear computational complexity $O(n)$. Considering that the normalization factor must be computed every time the BRDF is evaluated and n can be arbitrarily (even infinitely) large, this algorithm quickly becomes impractical for shiny materials.

We introduce an algorithm for exact energy normalization of the Phong specular BRDFs, which can instead be computed in constant time and space. Our formulation also supports non-integral specular exponent values. To achieve this, we simply convert Arvo's method (as it is expressed in [4]) into a mathematical language, apply continuation and subsequently various simplifications to the mathematical form, and then interpret the result back as an algorithm. The factors introduced in this paper can be considered a practical "fix" for the Phong specular BRDFs.

2 Background

When light hits a surface point $\mathbf{x}$, a part of it is transmitted (absorbed, for "opaque" materials) and the rest is reflected. Let ϵ be the ratio of reflectance (i.e. $1-\epsilon$ accounts for transmission) and L_i be the incoming radiance from a particular direction ω_i. Thus, ϵ portion of the light received by the surface $(\mathbf{N}\cdot\omega_i)\,L_i$ is reflected toward outgoing directions ω_o, where $\mathbf{N}$ is the surface normal. The incoming light multiplied by ϵ must be equal to the integral over the whole hemisphere for outgoing illumination, such that

$$(\mathbf{N}\cdot\omega_i)\,L_i\,\epsilon = \int_\Omega (\mathbf{N}\cdot\omega_i)\,L_i\,f_r(\omega_i,\mathbf{x},\omega_o)\,(\mathbf{N}\cdot\omega_o)\,d\omega_o\,.$$

By canceling out the terms $(\mathbf{N}\cdot\omega_i)\,L_i$ on either side, we can write:

$$\epsilon = \int_\Omega f_r(\omega_i,\mathbf{x},\omega_o)\,(\mathbf{N}\cdot\omega_o)\,d\omega_o\,. \tag{1}$$

A realistic material cannot reflect more light than it receives, meaning it conserves energy, such that $\epsilon \le 1$. If a material has no transmission/absorption, we must get exactly $\epsilon = 1$.

2.1 The Phong Specular Component BRDFs

The Phong reflectance model [14] proposes a reflection model that is a combination of a diffuse and a specular term, controlled by albedo parameters ρ_d and ρ_s, respectively. Later, the rendering equation [6] formalized the geometry term $\mathbf{N}\cdot\omega_i$. We recognize Phong's diffuse term as a Lambertian diffuse BRDF times the geometry term from the rendering equation, but his specular term lacks the geometry term, and thereby would implicitly have a divided term in the BRDF. Dropping this detail gives the Modified Phong BRDF [10]. In modern conventions, the Phong and Modified Phong BRDFs therefore respectively take the forms:

$$f_{r;P}(\omega_i,\mathbf{x},\omega_o) = \frac{\rho_d}{\pi} + \frac{\rho_s}{I_P}\,\frac{\max(0,\mathbf{R}(\omega_i)\cdot\omega_o)^n}{(\mathbf{N}\cdot\omega_i)}$$
$$f_{r;M}(\omega_i,\mathbf{x},\omega_o) = \frac{\rho_d}{\pi} + \frac{\rho_s}{I_M}\,\max(0,\mathbf{R}(\omega_i)\cdot\omega_o)^n\,,$$

where ω_i and ω_o are the incoming and outgoing light directions, $\mathbf{N}$ is the surface normal, I_P and I_M are the energy normalization terms for the two BRDFs, and $\mathbf{R}(\mathbf{u}) := -\mathbf{u} + 2\,(\mathbf{N}\cdot\mathbf{u})\,\mathbf{N}$ is the perfect reflection direction. For these BRDFs to be sensible, the diffuse and specular albedos must sum to no more than 1, i.e. $\rho_d + \rho_s \le 1$. The specular exponent n is an extended real number whose sensible values range in $[0,\infty]$. In particular, infinite values are allowed, and correspond to perfect specular (i.e. mirror) reflection.

The Phong BRDFs bundle diffuse and specular into what we would now call a multi-layer model[1]. The diffuse term is already normalized as a simple Lambertian BRDF; however, the specular component is not. Without loss of generality, we need normalize only the Phong and Modified Phong *specular component* BRDFs, and the distinction will be hereafter dropped:

$$f^{spec}_{r;P}(\omega_i,\mathbf{x},\omega_o) = \frac{1}{I_P}\,\frac{\max(0,\mathbf{R}(\omega_i)\cdot\omega_o)^n}{(\mathbf{N}\cdot\omega_i)} \tag{2}$$
$$f^{spec}_{r;M}(\omega_i,\mathbf{x},\omega_o) = \frac{1}{I_M}\,\max(0,\mathbf{R}(\omega_i)\cdot\omega_o)^n \tag{3}$$

Substituting these BRDFs into Equation 1 with $\epsilon = 1$ while applying reciprocity tells us that the values of I_P and I_M must be:

$$I_P = \int_\Omega \max(0,\mathbf{R}(\omega_o)\cdot\omega_i)^n\,d\omega_o \tag{4}$$
$$I_M = \int_\Omega \max(0,\mathbf{R}(\omega_o)\cdot\omega_i)^n\,(\mathbf{N}\cdot\omega_o)\,d\omega_o \tag{5}$$

It has been widely observed that when $\omega_i = \mathbf{N}$, the integrals are maximal. The equations then simplify easily (see Appendix B), giving energy conservation (but not normalization) terms:

$$I_P = \frac{2\pi}{n+1}, \qquad I_M = \frac{2\pi}{n+2} \tag{6}$$

Unfortunately, being only conservation terms, these

[1] As in other multi-layer composite-BRDF models, one would typically weight the specular term by the Fresnel reflection and the diffuse term by the Fresnel transmission.

Algorithm 1 Variant of Arvo's method, used for the Phong BRDF.

```
procedure ArvoIntegratePhong(N, ! i, n)
    c    (N    ! i)
    s    p 1    c^2
    if n even then
        A    π    arccos(c)
        k    1
        T_k    s
    else
        A    π/2
        k    0
        T_k    π/2
    end if
    t    0
    while k    n    1 do
        t    t + T_k
        k    k + 2
        T_k    s^2 ((k    1)/k) T_k
    end while
    I_P    2 (A + c t) / (n + 1)
    return I_P
end procedure
```

Algorithm 2 Arvo's method, used for the Modified Phong BRDF.

```
procedure ArvoIntegrateModifiedPhong(N, ! i, n)
    c    (N    ! i)
    s    p 1    c^2
    if n even then
        A    π/2
        k    0
        T_k    π/2
    else
        A    π    arccos(c)
        k    1
        T_k    s
    end if
    t    0
    while k    n    2 do
        t    t + T_k
        k    k + 2
        T_k    s^2 ((k    1)/k) T_k
    end while
    I_M    2 (T_k + A c + c^2 t) / (n + 2)
    return I_M
end procedure
```

only normalize the Phong BRDFs at normal incidence: even with $\rho_s = 1$, energy is lost for any $\omega_i \neq \mathbf{N}$, particularly near grazing angles. We must turn to more-advanced techniques to truly normalize Phong for all directions ω_i and ω_o.

2.2 Energy Normalization

Unfortunately, computing the correct energy normalization factors in Equations 4 and 5 is difficult. Intuitively, the specular lobe intersects the ecliptic plane in a complex way when the outgoing direction is not $\mathbf{N}$. Moreover, when $\omega_o \neq \mathbf{N}$, the geometry term no-longer aligns with the specular lobe, and so must be treated separately.

Arvo [1] developed algorithms that can be applied to this problem. Specifically, I_M can be computed using Algorithm 2 (slightly modified; see Arvo's §6.4 and Integral 31a in [4]). Similarly, I_P can be computed using a result derived from the same paper (see [4] Integral 31b), which we rework into a similar form in Algorithm 1. These algorithms are not widely implemented or known, but were the first, and to our knowledge heretofore-only, way to get the correct normalization terms for the Phong BRDFs. Notice that apart from the direction of the branch, the loop bound, and the last line, the algorithms are identical.

Unfortunately, these algorithms have two major limitations. The first is that they have $O(n)$ time complexity. This is unacceptable considering that BRDFs are evaluated millions or billions of times in a typical render. The specular exponent n is unbounded, and is in practice quite large (thousands or at-least hundreds would be expected for a shinier material).

The second limitation is that both algorithms are limited to values of n that are nonnegative integers. This adds implementation complexity, but is mainly objectionable because discretization artifacts are likely to be produced if n varies (such as when roughness is textured), especially at the lower values.

Precomputing a 1D table for a particular specular exponent n and a range of cosine values $\mathbf{N} \quad \omega_i$ allows the BRDF to be evaluated fairly efficiently, albeit due to interpolation, only approximately. For acceptable results at glancing angles, the table must be quite large, incurring excessive precomputation and caching cost. A 2D table, which allows the specular exponent to vary, is even larger; the precomputation cost can be seconds or minutes.

3 Energy Normalization of the Phong BRDFs

Our goal is to convert Arvo's algorithms into a constant-time method. In this section, we achieve this by first formulating the algorithms mathematically. Various sums and products of this can then be evaluated either in closed-form or in terms of various easy-to-work-with functions, allowing further simplifications and evaluation at non-integral specular exponents. Interpreting the result back as an algorithm yields a constant-time computation.

We begin by providing the following reformulation of both Algorithms 1 and 2 into mathematical terms:

$$c = (\mathbf{N} \quad \omega_i), \qquad s = \sqrt{1 \quad c^2}, \qquad \text{specular exponent } n$$

$$A = \begin{cases} \pi \quad \arccos(c) & \text{if } n \text{ even (Phong) or} \\ & \quad n \text{ odd (Modified Phong)} \\ \pi/2 & \text{if } n \text{ odd (Phong) or} \\ & \quad n \text{ even (Modified Phong)} \end{cases}$$

$$T_k = \begin{cases} (\pi/2)\ s^k \prod_{j=1}^{k=2} \frac{2j-1}{2j} & \text{if } k \text{ even} \\ s^k \prod_{j=1}^{(k-1)=2} \frac{2j}{2j+1} & \text{if } k \text{ odd} \end{cases}$$

$$S_k = \begin{cases} \sum_{j=0}^{k=2} T_{2j} & \text{if } k \text{ even} \\ \sum_{j=1}^{(k+1)=2} T_{2j-1} & \text{if } k \text{ odd} \end{cases}$$

$$I_P = 2\ (A + c\ S_{n-1})\ /\ (n+1)$$

$$I_M = 2\ (T_n + A\ c + c^2\ S_{n-2})\ /\ (n+2)$$

It is worth noting that when written out mathematically, the algorithm has quadratic complexity: computing T_k is $O(k)$ and so computing S_k is $O(k^2)$.

We now turn to writing these expressions in terms of functions without summations or products. In the following, let n be odd for the Phong BRDF and n be even for the Modified Phong BRDF so that we only need to compute T_k and S_k for k even (it turns out we will get the same result without this assumption; this is just to simplify the presentation).

T_k (again, for k even) can be computed as [2] [3]:

$$\begin{aligned} T_k &= \frac{\pi}{2}\ s^k \prod_{j=1}^{k=2} \frac{2j-1}{2j} = \frac{\pi}{2}\ s^k \frac{(k-1)!!}{2^{k=2}\ (k/2)!} \\ &= \frac{\pi}{2}\ s^k\ \frac{2^{k=2}\ ((k+1)/2)\ /\ \sqrt{\pi}}{2^{k=2}\ (k/2+1)} \\ &= \frac{\sqrt{\pi}}{2}\ s^k\ \frac{((k+1)/2)}{(k/2+1)} \end{aligned}$$

This result is resistant to further simplification, but the ratio of gamma functions can be computed efficiently and robustly by taking the exponential of the difference of the logs of the numerator and denominator; the log-of-gamma function is widely available and computationally efficient in most programming environments, both on the CPU and GPU (e.g. `lgamma(...)` in C/C++, CUDA, and OpenCL).

To solve the series S_k, after substituting in the appropriate T term, we break the summation into two infinite series:

$$S_k = \sum_{j=0}^{k=2} T_{2j} = \frac{\sqrt{\pi}}{2} \sum_{j=0}^{k=2} s^{2j}\ \frac{j+\frac{1}{2}}{(j+1)} = \frac{\sqrt{\pi}}{2}\ S_{k;1}\quad S_{k;2}$$

where $S_{k;1}$ and $S_{k;2}$ are defined as:

$$S_{k;1} := \sum_{j=0}^{} s^{2j}\ \frac{j+\frac{1}{2}}{(j+1)} \quad \text{and} \quad S_{k;2} := \sum_{j=\frac{k}{2}+1}^{} s^{2j}\ \frac{j+\frac{1}{2}}{(j+1)}\ .$$

The first infinite subseries $S_{k;1}$ can be rearranged by applying the double-factorial identity:

$$\begin{aligned} S_{k;1} &= \frac{1}{2} \sum_{j=0}^{} s^{2j}\ \frac{2^j\ \ j+\frac{1}{2}\ /\ \frac{1}{2}}{2^j\ j!} \\ &= \frac{\sqrt{\pi}}{2} \sum_{j=0}^{} s^{2j}\ \frac{(2j-1)!!}{(2j)!!} \end{aligned}$$

This is just a binomial series:

$$\begin{aligned} S_{k;1} &= \frac{\sqrt{\pi}}{2} \sum_{j=0}^{} (-1)^j \binom{\frac{1}{2}}{j} (s^2)^j \\ &= \sqrt{\frac{\pi}{1-s^2}} = \frac{\sqrt{\pi}}{c} \end{aligned}$$

The second infinite subseries $S_{k;2}$ can be solved by first shifting it back to $j=0$. Letting $a := k/2+1$:

$$S_{k;2} = \sum_{j=a}^{} s^{2j}\ \frac{j+\frac{1}{2}}{(j+1)} = \sum_{j=0}^{} (s^2)^{j+a}\ \frac{j+a+\frac{1}{2}}{(j+a+1)}$$

and then rearranging in terms of the Pochhammer symbol $(x)_n := \Gamma(x+n)\ /\ \Gamma(x)$:

$$\begin{aligned} S_{k;2} &= (s^2)^a \sum_{j=0}^{} (s^2)^j\ \frac{j+a+\frac{1}{2}}{(j+a+1)} \\ &= (s^2)^a\ \frac{a+\frac{1}{2}}{(a+1)} \sum_{j=0}^{} \frac{(s^2)^j\ \ (1+j)\ \ a+\frac{1}{2}+j\ \ (a+1)}{(1+j)\ \ (1)\ \ a+\frac{1}{2}\ \ (a+1+j)} \\ &= (s^2)^a\ \frac{a+\frac{1}{2}}{(a+1)} \sum_{j=0}^{} \frac{(1)_j\ \ a+\frac{1}{2}\ _j}{(a+1)_j}\ \frac{(s^2)^j}{j!} \end{aligned}$$

we get the form of a hypergeometric series. Applying the Gaussian hypergeometric function ${}_2\mathrm{F}_1(\ \ )$:

$$S_{k;2} = (s^2)^a\ \frac{a+\frac{1}{2}}{(a+1)}\ {}_2\mathrm{F}_1\ 1, a+\tfrac{1}{2}; a+1; s^2$$

Then, because the hypergeometric function takes a special form (the first argument is 1), on to the incomplete beta function $\mathrm{B}(x;a,b)$:

$$\begin{aligned} S_{k;2} &= (s^2)^a\ \frac{a+\frac{1}{2}}{(a+1)}\ a\ (s^2)^{-a}\ (1-s^2)^{1=2}\ \mathrm{B}\ s^2; a, \tfrac{1}{2} \\ &= \frac{a+\frac{1}{2}}{c\ \ (a)}\ \mathrm{B}\ s^2; a, \tfrac{1}{2} \end{aligned}$$

Substituting the $S_{k;1}$ and $S_{k;2}$ subseries back in, we find S_k is:

$$\begin{aligned} S_k &= \frac{\sqrt{\pi}}{2}\ \frac{\sqrt{\pi}}{c}\ \frac{a+\frac{1}{2}}{c\ \ (a)}\ \mathrm{B}\ s^2; a, \tfrac{1}{2} \\ &= \frac{\sqrt{\pi}}{2\ c}\ \sqrt{\pi}\ \frac{((k+3)/2)}{(k/2+1)}\ \mathrm{B}\ s^2; \tfrac{k}{2}+1, \tfrac{1}{2} \end{aligned}$$

[2] The double-factorial function $n!! = n(n-2)(n-4)\ \ (1)$ must not be confused with the factorial function $n! = n(n-1)(n-2)\ \ (1)$ applied twice.

[3] The gamma function $\Gamma(\ \ )$ generalizes the factorial function from the non-negative integers to most complex numbers. For any positive integer n, we have $\Gamma(n) = (n-1)!$, but in particular $\Gamma(\ \ )$ also works for floating-point values.

Evaluating A and $S_{n\ 1}$ for Phong, we find that the normalization factor I_P is:

$$I_P = \frac{2}{n+1}\ \frac{\pi}{2} + c\ \frac{\sqrt{\pi}}{2\ c}\ \sqrt{\pi}\ \frac{(n/2+1)}{((n+1)/2)}\ \mathrm{B}\ s^2; \tfrac{n+1}{2}, \tfrac{1}{2}$$

Similarly, evaluating T_n, A, and $S_{n\ 2}$ for Modified Phong, we find that the normalization factor I_M is:

$$I_M = \frac{2}{n+2}\ \frac{\sqrt{\pi}}{2}\ s^n\ \frac{((n+1)/2)}{(n/2+1)} + \frac{\pi}{2}\ c + c^2\ \frac{\sqrt{\pi}}{2\ c}\ \sqrt{\pi}\ \frac{((n+1)/2)}{(n/2)}\ \mathrm{B}\ s^2; \tfrac{n}{2}, \tfrac{1}{2}$$

Simplification yields our final two normalization terms:

$$I_P = \frac{1}{n+1}\ 2\pi\ \sqrt{\pi}\ \frac{(n/2+1)}{((n+1)/2)}\ \mathrm{B}\ s^2; \tfrac{n+1}{2}, \tfrac{1}{2} \tag{7}$$

$$I_M = \frac{1}{n+2}\ 2\pi c + \sqrt{\pi}\ \frac{((n+1)/2)}{(n/2+1)}\ s^n\ \frac{n}{2}\ c\ \mathrm{B}\ s^2; \tfrac{n}{2}, \tfrac{1}{2} \tag{8}$$

For Modified Phong, there is a complicating detail: when $n = 0$, the incomplete beta function becomes indeterminate. In this case, we must replace that term by its limit. To compute the limit, we use the integral definition of the incomplete beta function and evaluate the term:

$$\begin{aligned}\lim_{n\ !\ 0}\ \frac{n}{2}\ c\ \mathrm{B}\ s^2; \tfrac{n}{2}, \tfrac{1}{2} &= c \lim_{n\ !\ 0} \frac{n}{2} \int_0^{s^2} t^{n=2\ \ 1} (1\ \ t)^{1=2\ \ 1}\ dt \\ &\qquad u := \sqrt{1\ \ t} \\ &= c \lim_{n\ !\ 0} n \int_c^1 (1\ \ u^2)^{n=2\ \ 1}\ du\end{aligned}$$

The integrand does not satisfy the requirements for bringing the limit into the integral, so we instead proceed by series expansion. We factor the integrand and take the Taylor series at $x = 1$, which converges everywhere and will simplify nicely:

$$\begin{aligned}(1\ \ x^2)^p &= (1\ \ x)^p\ (1+x)^p \\ &= (1\ \ x)^p \sum_{j=0}^{\infty} (1+1)^p\ \ ^j\ \binom{p}{j}\ (x\ \ 1)^j \\ &= 2^p \sum_{j=0}^{\infty} \frac{p}{j}^{!}\ \ \frac{1}{2}^{\ j}\ (1\ \ x)^{p+j}\end{aligned}$$

We substitute the series into the integral:

$$\begin{aligned}&c \lim_{n\ !\ 0} n \int_c^1 (1\ \ u^2)^{n=2\ \ 1}\ du \\ &= c \lim_{n\ !\ 0} n \int_c^1 4 2^{n=2\ \ 1} \sum_{j=0}^{\infty} \frac{n}{2}\ \ \frac{1}{j}^{!}\ \ \frac{1}{2}^{\ j}\ (1\ \ u)^{n=2\ \ 1+j} 5\ du\end{aligned}$$

From here, we simplify and rearrange:

$$\begin{aligned}&= c \lim_{n\ !\ 0} 2^{n=2} \sum_{j=0}^{\infty} \frac{n}{2}\ \ \frac{\frac{n}{2}}{j}^{!}\ \ 1\ \ \frac{1}{2}^{\ j} \int_c^1 (1\ \ u)^{n=2\ \ 1+j}\ du \\ &= c \lim_{n\ !\ 0} 2^{n=2} \sum_{j=0}^{\infty} \frac{n}{2}\ \ \frac{\frac{n}{2}}{j}^{!}\ \ 1\ \ \frac{1}{2}^{\ j}\ \ \frac{(1\ \ c)^{n=2+j}}{\frac{n}{2}+j}^{!}\end{aligned}$$

When $j = 0$, the leading $n/2$ cancels with the $n/2 + j$ in the denominator, leaving the term nonzero. This cancellation does not happen in subsequent terms, so the leading $n/2$ causes them all to disappear when we take the limit $n\ !\ 0$. Therefore, we have but to pull the first term out of the summation to attain the limiting value:

$$\begin{aligned}&= c \lim_{n\ !\ 0} 2^{n=2}\ \frac{n}{2}\ \ \frac{\frac{n}{2}}{0}^{!}\ \ 1\ \ \frac{1}{2}^{\ 0}\ \ \frac{(1\ \ c)^{n=2+0}}{\frac{n}{2}+0}^{!} \\ &= c \lim_{n\ !\ 0} 2^{n=2}\ (1\ \ c)^{n=2} \\ &= c\end{aligned}$$

Therefore, if the specular exponent $n = 0$, the incomplete beta term $\frac{n}{2}\ c\ \mathrm{B}\ s^2; \frac{n}{2}, \frac{1}{2}$ for computing I_M in Equation 8 should be replaced by c.

Compare Equations 7 and 8 to the energy conservation terms in Equation 6. We can see that when $c = 1$ (i.e. at normal incidence), the former reduces to the latter (since $c = 1$ means $s = 0$, and $\mathrm{B}(0;\ \ ,\ \ ) = 0$). However, these normalization formulae are correct for other incidence also. No series or products remain, and in practice computing the (ratio of) gamma function(s) and the incomplete beta function are constant-time operations (see next section), so our approach has $O(1)$ complexity.

Although we derived this result using n even, the same result is obtained when n is odd, and so this formula works for all integer-valued $n\ \ 0$. In-fact, empirically, the algorithm produces correct results for *real-valued* $n\ \ 0$.

4 Results

We evaluate these normalization methods with a GPU pathtracer based on OptiX [13] running on an NVIDIA GTX 1080, using standard importance sampling [4].

4.1 Numerical Validation

The top graph of Figure 1 shows different values of I_M computed for the full range of incident light angles and a range of exponent values n. Arvo's method only works with integer values of n, so its results are drawn as curves (in red). Our results match Arvo's method (down to numerical precision), as expected. In addition, our method also smoothly interpolates between them for non-integer values.

The bottom graph of Figure 1 shows the difference between our method and a numerically computed integral. Notice that our results closely match the

Numerical Validation of Integration

Fig. 1 Validating the Modified Phong BRDF (the plots for Phong are qualitatively similar). The interesting range of n is shown; at larger values the graph tapers smoothly as one expects. Top: Arvo's algorithm can normalize at integer specular exponents. Our method works for any specular exponent. Bottom: Our algorithm differs from a numerical validation by no more than 2.54×10^{-7}, which is comparable to the worst-case convergence of the numerical procedure itself.

Reciprocal Integrals of Phong and Modified Phong BRDFs

Fig. 2 Log-log plots of the reciprocal normalization terms $1/I_P$ and $1/I_M$ over a wide range of specular exponent n values. Notice that as n increases, I_P and I_M get closer to zero and their reciprocals continue to increase.

numerically computed results to 2.54×10^{-7}, which is roughly the numerical integrator's worst-case accuracy. This validates that we can compute the correct normalization terms, including particularly for floating-point specular exponents where Arvo's method cannot provide a ground truth.

To visualize the shape of the integral far from $n = 0$, we present a visualization of the (reciprocal of) the integrals of both BRDF variants in Figure 2. With increasing n, the reciprocals $1/I_P$ and $1/I_M$ (used in Equations 2 and 3) continue to grow, as expected: since the areas of the BRDFs become progressively smaller, their values approach the delta function of a perfect specular BRDF. While we did not encounter numerical problems in this range, they are inevitable for any specular BRDF that becomes sufficiently close to a delta but remains finite.

4.2 White Furnace Test

Figure 3 shows a standard white furnace test. A sphere is placed into a uniform environment, and the specular albedo is set to $_s = 1$ (i.e. $_d = 0$). Under these circumstances, every point on the object must have the same radiance as the environment. The object should therefore disappear into the background. The specular exponent for both BRDFs is arbitrarily set to $n = 5$ (note that the qualitative behavior in all four images is not affected by the exponent).

On the left, we see the effect of the typical energy conservation terms used today (i.e., Equation 6). As the specular lobe tilts over at glancing angles, it intersects progressively more with the ecliptic plane. This energy is lost, causing erroneous darkening of the edges.

On the right, we see the effect of energy normalization (computed with our algorithm, although Arvo's algorithms would give the same result for integer-valued specular exponents). The sphere cannot be readily distinguished from the background (though due to quantization and sampling issues, there is variation of about half a gray level, which may be visible).

4.3 Performance Comparison

Figure 4 shows the performance scaling of various methods, as measured for the white furnace test.

Only doing energy conservation (pale orange) requires negligible computation, and so should be viewed