2

II

An Approximation to the Chapman Grazing-Incidence Function for Atmospheric Scattering

Christian Schüler

2.1 Introduction

Atmospheric scattering for computer graphics is the treatment of the atmosphere as a participating medium, essentially "calculating the color of the sky." This is interesting for any application where the time of day, the season, or the properties of the atmosphere are not known in advance, or the viewpoint may not be restricted to a point on the earth's surface. It is a historically difficult effect to render, especially at planetary scale.

Early attempts at atmospheric scattering can be found in [Klassen 87] and [Nishita et al. 93]. Recent implementations with an emphasis on real time are [Hoffmann and Preetham 02], [O'Neil 05], and [Bruneton and Neyret 08]. A common theme of all these approaches is finding ways to efficiently evaluate or precompute the Chapman function, $\mathrm{Ch}(x, \chi)$. This is the density integral for a ray in a spherically symmetric, exponentially decreasing atmosphere.

The Chapman function has been subject to extensive treatment in the physics literature. Approximations and tabulations have been published, most of it with a focus on precision. This article explores a different direction for its evaluation: an approximation so cheap that $\mathrm{Ch}(x, \chi)$ can be considered a commodity, while still being accurate enough for our graphics needs.

2.2 Atmospheric Scattering

This section is a brief review of atmospheric scattering and a definition of terms.

When light travels through air, it will be partly absorbed and partly scattered into other directions. This gives rise to the phenomenon of *aerial perspective*. The

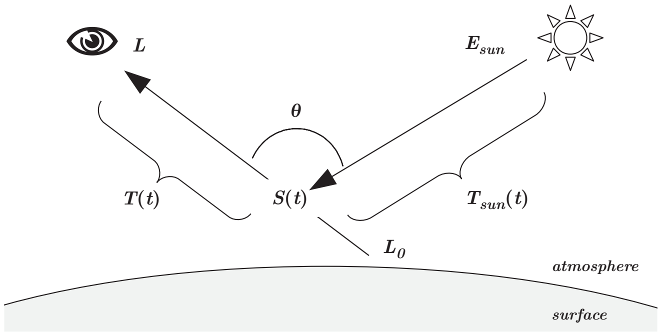

Figure 2.1. Atmospheric scattering 101. See the text for an explanation of the symbols.

fraction of light that is unimpeded along a path is the transmittance T, and the amount that is added into the path due to scattering is the in-scatter S (see Figure 2.1). Thus, the aerial perspective of a distant source of radiance L_0 is seen by an observer as the radiance L:

$$L = L_0 T + S.$$

To arrive at the total in-scatter S, in general one would have to integrate it along the path. Then, the in-scatter at a particular point $S(t)$ would have to be calculated from the local irradiance field E over the entire sphere of directions Ω with an atmosphere-dependent phase function $f(\theta)$:

$$\begin{aligned} S &= \int S(t)\, T(t)\, dt, \\ S(t) &= \int_{\Omega} E(t)\, f(\theta)\, d\Omega. \end{aligned}$$

The irradiance is usually discretized as a sum of individual contributions. Especially during the day, the single most important contributor is the sun, which can be simplified to a directional point source E_{sun}, for the irradiance arriving at the outer atmosphere boundary; E_{sun} is attenuated by the transmittance T_{sun} for the path from the atmosphere boundary towards point $S(t)$:

$$S(t) = E_{\text{sun}}\, T_{\text{sun}}(t)\, f(\theta).$$

The transmittance itself is an exponentially decreasing function of the airmass m times an extinction coefficient β. The latter is a property of the scattering medium, possibly wavelength dependent. The airmass is an integral of the air density $\rho(t)$ along the path:

$$\begin{aligned} T &= \exp\left(-\beta m\right), \\ m &= \int \rho(t)dt. \end{aligned}$$

To complete the calculation, we need a physical model for β and f. There exists Rayleigh theory and Mie theory, which have been discussed in depth in previous publications, e.g., [Nishita et al. 93] and [Hoffmann and Preetham 02]. It is beyond the scope of this article to provide more detail here.

2.3 The Chapman Function

In order to reduce the algorithmic complexity, it would be nice to have an efficient way to calculate transmittances along rays. It turns out that this reduces to an evaluation of the Chapman function.

Without loss of generality, let's start a ray at an observer inside the atmosphere and extend it to infinity (see Figure 2.2). The ray can be traced back to a point of lowest altitude r_0. We take the liberty to call this point the *periapsis* even though the path is not an orbit. Here we define $t = 0$, and the altitude for any point along the ray as follows:

$$r(t) = \sqrt{r_0^2 + t^2}.$$

Let's further assume a spherically symmetric atmosphere with an exponentially decreasing density, characterized by a scale height H. We normalize the

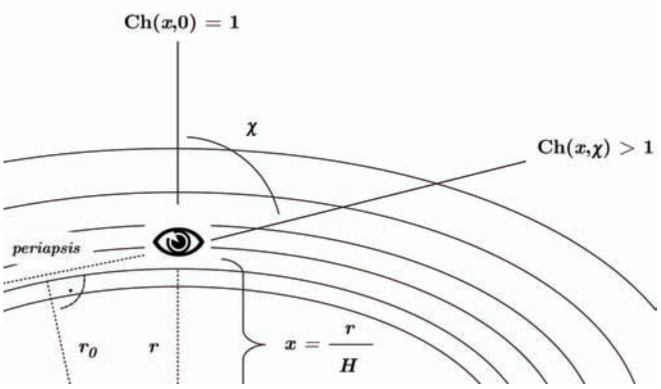

Figure 2.2. The Chapman function. Relative airmass in an exponentially decreasing atmosphere with scale height H, normalized observer altitude x and incidence angle χ. The lowest altitude is at r_0.

density to a reference density ρ_0 at reference altitude R. The density at any altitude is then

$$\rho(r) \quad = \quad \rho_0 \exp\left(\frac{R-r}{H}\right).$$

Combining these expressions yields an integral for the airmass along the entire ray. The integration boundary is trigonometrically related to the observer altitude r and the incidence angle χ:

$$\begin{aligned} m &= \rho_0 \int_{r\cos\chi}^{\infty} \exp\left(\frac{R-\sqrt{r_0^2+t^2}}{H}\right) dt, \\ r_0 &= r\sin\chi. \end{aligned}$$

This integral does not have a simple solution, except when looking straight upwards into the zenith ($\chi = 0$). In this case, the mass along the ray is just the mass of the air column above the observer. This is, by the definition of the density distribution, one scale height times the density at the observer. Let's call that mass $m_\perp$:

$$m_\perp \quad = \quad \rho_0\, H \exp\left(\frac{R-r}{H}\right).$$

Can we possibly have a function that relates $m_\perp$ to m? We can, for this is the Chapman function $\mathrm{Ch}(x,\chi)$, named after the physicist who was the first to formulate this problem [Chapman 31]. We write the Chapman function as follows:

$$\begin{aligned} m &= m_\perp \, \mathrm{Ch}(x,\chi), \\ \text{with} & \\ x &= \frac{r}{H}. \end{aligned}$$

The arguments are historically named x for normalized altitude and the Greek letter chi for incidence angle (don't blame me for that). The function is independent of scale and is usually tabulated or approximated numerically. For convenience, an analytic expression is given below. This has been stripped down from a more general solution found in [Kocifaj 96]:

$$\begin{aligned} \mathrm{Ch}(x,\chi) = \frac{1}{2}\Bigg[& \cos(\chi) + \\ & + \exp\left(\frac{x\cos^2\chi}{2}\right) \mathrm{erfc}\left(\sqrt{\frac{x\cos^2\chi}{2}}\right)\left(\frac{1}{x}+2-\cos^2\chi\right)\sqrt{\frac{\pi x}{2}}\Bigg]. \end{aligned}$$

The above expression is not practical for real-time evaluation, for several reasons: it contains the complementary error function, erfc, which needs a numerical approximation for itself. It has bad numerical behavior for large x and small χ, where the exp-term becomes very large and the erfc-term virtually zero. We use this expression, however, as our ground-truth standard, evaluated with arbitrary precision math software.

2.4 Towards a Real-Time Approximation

To better understand the Chapman function, we will plot a graph of it (see Figure 2.3) and observe a number of important properties:

The function is even wrt. χ; $\mathrm{Ch}(x,\chi) = \mathrm{Ch}(x,-\chi)$.

There is unity at $\chi = 0$; $\mathrm{Ch}(x,0) = 1$.

There is a limit for large x; $\lim\limits_{x\to\infty} \mathrm{Ch}(x,\chi) = \dfrac{1}{\cos\chi}$.

These properties are easily explained. The even symmetry follows from the problem specification. Only the cosine of the incidence angle appears in the expression. This allows us to reduce the covered range to $0 < \chi < 180°$. Second, since $\mathrm{Ch}(x,\chi)$ relates $m_\perp$ to m, its value must approach unity for small incidence angles. And finally, in the limit of a flat earth, the Chapman function must approach the secant function.

These properties can be used to engineer our approximation, Ch'. The limit for large x suggests a rational approximation, as we are going to cope with a pole. Runtime efficiency demands the lowest possible order. So we are going to

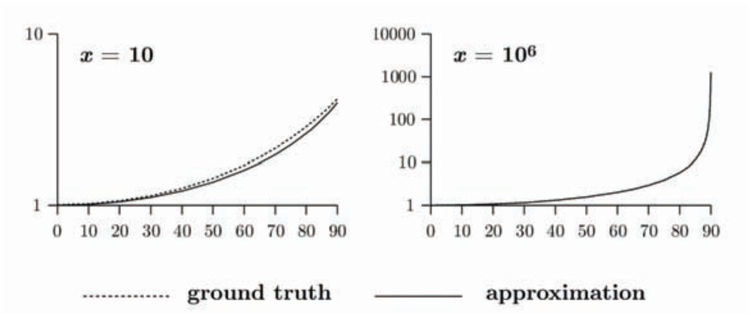

Figure 2.3. Graph of the Chapman function. $\mathrm{Ch}(x,\chi)$ is plotted on a logarithmic scale as a function of incidence angle χ, for two extreme cases of x. An observer on the earth's surface with an atmosphere scale height of 8.4 km would correspond to an x of $r_\oplus/8.4 \simeq 760$.

look for a first-order rational function of $\cos\chi$, approaching unity on the left and the value of $\mathrm{Ch}_{\parallel}(x)$ on the right, where $\mathrm{Ch}_{\parallel}(x)$ is used as a shorthand for the Chapman function evaluated at $\chi = 90°$. There is only one possibility for such a rational function:

$$\begin{aligned} \mathrm{Ch}'(c,\chi) &= \frac{c}{(c-1)\cos(\chi)+1} \qquad \Big| \; |\chi| < 90°, \\ \text{with} & \\ c &= \mathrm{Ch}_{\parallel}(x). \end{aligned}$$

As it turns out, this low-order function is a pretty good approximation. The useful range for χ is, however, limited to below $90°$. Beyond that angle, the approximation grows hyperbolically to a pole at infinity, while the exact Chapman function grows exponentially, and always stays finite. We will look for ways to handle $\chi > 90°$, but we must first turn our attention to the coefficient c.

2.4.1 At the Horizon

If the observer altitude is fixed, we could precalculate a value for c. However, for a moving observer, and in the absence of a Chapman function to fall back on, we need an approximation for c itself. Let's take a look at $\mathrm{Ch}_{\parallel}(x)$:

$$\mathrm{Ch}_{\parallel}(x) = \left(\frac{1}{2x}+1\right)\sqrt{\frac{\pi x}{2}}.$$

This is already a lot simpler than the full Chapman function, but still requires a square root and a division. To simplify it further, we assume that x is usually large and neglect the term $^1/_{2x}$ to get a function that is purely proportional to $\sqrt{x}$. Using the value $\sqrt{^\pi/_2} \simeq 1.2533$ as a coefficient results in

$$\mathrm{Ch}_{\parallel}(x) \simeq 1.2533\sqrt{x} \qquad \big| \; x > 10.$$

2.4.2 Beyond the Horizon

Consider Figure 2.4. The airmass m_L along an entire line is the airmass along the forward ray plus the airmass along the backward ray:

$$m_L = \rho\Big[\mathrm{Ch}(x,\chi) + \mathrm{Ch}(x, 180° - \chi)\Big].$$

The above equation must be true for any point along the line. We can move the observer to the periapsis, where both the forward and the backward ray are horizontal. Using trigonometry, the altitude at the periapsis is $x_0 = x\sin\chi$ and density follows as $\rho\exp(x - x_0)$. Another way of expressing m_L is therefore

$$m_L = 2\,\rho\exp(x - x_0)\,\mathrm{Ch}_{\parallel}(x_0).$$

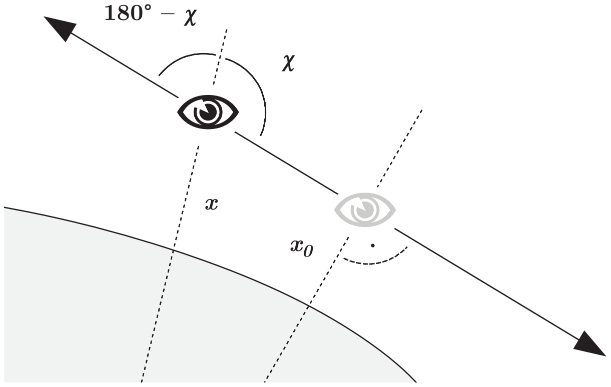

Figure 2.4. Airmass along an entire line. A virtual observer is placed at the periapsis.

Combining the above equations, it is possible to arrive at an identity that expresses the Chapman function in terms of itself with a reflected incidence angle:

$$\mathrm{Ch}(x,\chi) \quad = \quad 2\,\exp(x - x\sin\chi)\,\mathrm{Ch}_{\|}(x\sin\chi) - \mathrm{Ch}(x, 180^\circ - \chi).$$

If the Chapman function is known for $0 < \chi < 90^\circ$, it is therefore known for all χ.

2.5 Implementation

See Listing 2.1 for an implementation of the approximate Chapman function in C. It consists of a branch on the value of χ, either applying the identity or not. The code differs from the mathematical formulae in three aspects, which are discussed in Sections 2.5.1–2.5.3.

2.5.1 Numeric Range

First, a numeric range problem must be resolved, which happens when x is large and x_0 is small. The identity formula contains the exponential $\exp(x - x_0)$, which overflows. To remedy this situation, we introduce a modified function $\mathrm{Ch_h}(X,h,\chi)$:

$$\mathrm{Ch_h}(X,h,\chi) \quad = \quad \mathrm{Ch}(X+h,\chi)\exp(-h).$$

This function takes a reference altitude X and the corresponding observer height h, and includes the factor $\exp(-h)$. This factor would have to be applied anyway to calculate the airmass. By including it in the function, the range problem cancels out, since $\exp(x - x_0)\exp(-h) = \exp(X - x_0)$.

```
float chapman_h( float X, float h, float coschi )
{
    // The approximate Chapman function
    // Ch(X+h,chi) times exp2(-h)
    // X - altitude of unit density
    // h - observer altitude relative to X
    // coschi - cosine of incidence angle chi
    // X and h are given units of the 50%-height

    float c = sqrt( X + h );
    if( coschi >= 0. )
    {
        // chi above horizon
        return c / ( c * coschi + 1. ) * exp2( -h );
    }
    else
    {
        // chi below horizon, must use identity
        float x0 = sqrt( 1. - coschi * coschi ) * ( X + h );
        float c0 = sqrt( x0 );
        return
            2. * c0 * exp2( X - x0 ) -
            c / ( 1. - c * coschi ) * exp2( -h );
    }
}
```

Listing 2.1. The approximate Chapman function in C, with modifications discussed in the text.

2.5.2 Distance Differences

The second modification relates to the ability of the approximation to stay faithful to sums and differences of path lengths. Special attention is drawn to the case of an observer who is close above a reflective surface. Consider Figure 2.5. Assuming near-horizontal angles, the airmasses m_A, m_B, and m_C for the three segments can be simplified to

$$\begin{aligned} m_A &\sim \mathrm{Ch}(x,\chi), \\ m_B &\sim \mathrm{Ch}(x,180^\circ-\chi)-\mathrm{Ch}(x',180^\circ-\chi), \\ m_C &\sim \mathrm{Ch}(x',\chi). \end{aligned}$$

The question is now, does the approximation satisfy $m_A = m_B + m_C$? This would be required for an exact color match at the horizon above a water surface. The short answer is that the original approximation does not hold for this property. The form of the denominator in the rational function, as well as the factor 1.2533, stand in the way. The approximation must be rigged to enable this

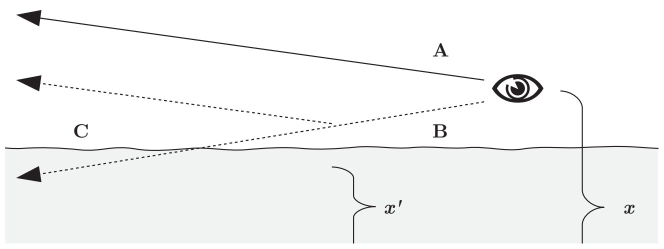

Figure 2.5. The Chapman function on a reflected ray. Is the approximation going to stay true to sums and differences of path lengths?

property; this new approximation is even simpler:

$$\begin{aligned}\mathrm{Ch}'(c,\chi) &= \frac{c}{c\cos(\chi)+1},\\ \text{with}\\ c &= \sqrt{x}.\end{aligned}$$

Dropping the factor of 1.2533 hardly makes a difference in the visual result. The change in the denominator is more severe, causing the approximation to fall below unity for small incidence angles. However, for typical uses with x well in the hundreds, the visual impact is again negligible. If needed, the loss can be compensated by an increase of the β-coefficients.

2.5.3 Using exp2

The code will make exclusive use of the dual exponential 2^x instead of the natural exponential e^x. We will therefore need all scales converted to the dual logarithm. We need a 50%-height H_{50} instead of the scale height H; we need 50%-extinction coefficients, and so on:

$$\begin{aligned}H_{50} &= H\ln 2,\\ \beta_{50} &= \beta\ln 2,\\ &\ldots .\end{aligned}$$

The reason for this change is that `exp2` is usually the more efficient function to compute. To optimize even more, an implementation can employ a fast `exp2` function for all scattering calculations. An example of such a fast `exp2` function is presented in the appendix (Section 2.8). It is a definitive win when the calculation is CPU-based, especially if it is vectorized with SIMD, calculating four values at once. On the GPU, there have been assembly instructions for a fast version (called `exp2pp` for “partial precision”) or the shader compiler takes it as a hint if a call to `exp2` has both a low-precision argument and a low-precision result.

```
vec3 transmittance( vec3 r, vec3 viewdir )
{
    // calculate the transmittance along a ray
    // from point r towards infinity

    float rsq = dot(r,r);
    float invrl = inversesqrt( rsq );
    float len = rsq * invrl;
    float x = len * invH50;
    float h = x - X50;
    float coschi = dot( r, viewdir ) * invrl;
    return exp2( -beta50 * H50 * chapman_h( X50, h, coschi ) );
}
```

Listing 2.2. Function for the transmittance along a ray.

2.6 Putting the Chapman Function to Use

Finally, in this section we are going to explore the ways in which we can use our shiny new tool. You should consult the example code on the website since not all listings are shown here.

2.6.1 Airmass and Transmittance

The airmass is calculated easily with the modified Chapman function. You need to know the observer height against some reference altitude (conveniently, this is the planet radius, or mean sea level), and the scale height of the atmosphere:

$$m \quad = \quad H\,Ch_h(X, h, \chi).$$

It is a small step from the airmass to the transmittance, since $T = \exp(-\beta m)$. See Listing 2.2 for a function to calculate the transmittance along a straight line through the atmosphere. The scale height and the extinction coefficients must be available globally. In the complete fragment program, this function is used to calculate the local sun color for surface shading.

2.6.2 Aerial Perspective

The full aerial perspective function has two colors as a result: the transmittance and the in-scatter. The function is too long to be listed here, but is included in the fragment program on the website. The signature is as follows:

```
void aerial_perspective( out vec3 T, out vec3 S,
    in vec3 r0, in vec3 r1, in bool infinite );
```

The function calculates the aerial perspective from point `r0` to point `r1`, or alternatively, from point `r0` along the ray through `r1` to infinity. The resulting transmittance is written to argument `T` and the in-scatter is written to argument `S`.

Here is a short explanation of the algorithm: In a first step, the function intersects the ray with the atmosphere boundary to get the integration interval, which is subdivided into a fixed number of segments. It then iterates over all segments in reverse order (back to front). For each segment, the Chapman function is called to calculate airmass, transmittance, and in-scatter. The in-scatter is then propagated along the ray in a way similar to alpha-blending.

2.6.3 The Example Raytracer

The accompanying online material for this article contains a shader (fragment- and vertex program) that implements a fully ray-traced, atmospheric single-scattering solution in real time. See the color images in Figure 2.6. You should be able to load the shader into a shader authoring tool (RenderMonkey, FX-Composer) and apply it on a unit sphere. If you are on the Mac, there is a ShaderBuilder project that you can simply double click.

The code assumes a planet centered at the origin of variable radius, with a Rayleigh atmosphere. For each pixel that hits the atmosphere, **`aerial_perspective`** is called once to get the in-scatter against the black background of space. The transmittance is not needed. For each pixel that hits the surface, the **`aerial_perspective`** function is called twice, one time for the view ray, and a second time for the ocean reflection.

To give a realistic appearance, the β-values were calculated for a situation similar to Earth. A little bit of orange absorption was added to account for the effect of atmospheric ozone, which is significant at twilight (see [Nielsen 03]). The landmass is shaded according to the Lommel-Seeliger law, which is a good model for rough surfaces, avoiding the typical Lambertian limb darkening. The specular reflection of the sun in the ocean is shaded with an approximate microfacet model (see [Schüler 09]). The final color is then tone mapped with an exponential soft saturation and 2.2 display gamma.

2.7 Conclusion

This article shows a way to accelerate atmospheric scattering calculations, by making the Chapman function an efficient commodity. This allows a strength reduction of the numerical integration, up to the point where the full single-scattering solution can be run in real time in a shader program.

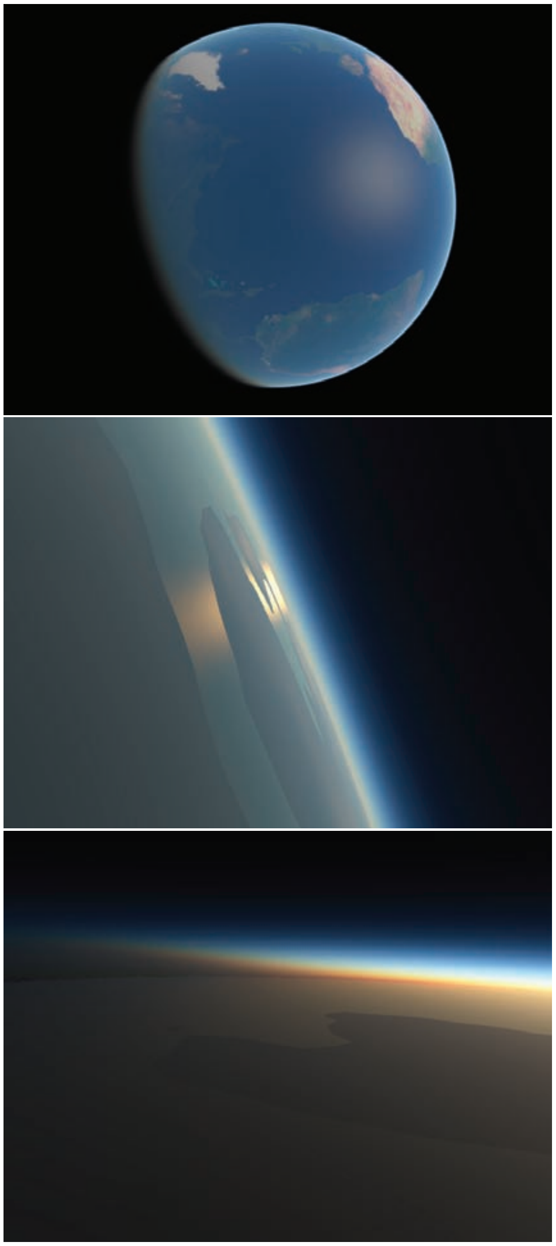

Figure 2.6. Color images from the example ray tracer. View of the Atlantic Ocean (top). Sun reflection, which is yellow due to transmittance (middle). Earth shadow, which is implicit in the solution (bottom).

```
float exp2pp( float x )
{
    // partial precision exp2, accurate to 12 bits

    const float c[3] = { 5.79525, 12.52461, -2.88611 };
    int e = round(x);
    float t = x - e;
    float m = ( t*t + c[0]*t + c[1] ) / ( c[2]*t + c[1] );
    return ldexp( m, e );
}
```

Listing 2.3. An example of a fast `exp2` function.

2.8 Appendix

2.8.1 A fast exp2 function

Listing 2.3 shows an example for a fast `exp2` function for use in atmospheric-scattering calculations. It is a low-order rational function in the range $-1/2$ to $+1/2$ together with the usual range reduction and scaling. Although it is an approximation, it does preserve the property that $1 - \exp(-x)$ is strictly positive if $x > 0$, which is important for the scattering calculations. The function is dependent on an efficient way to assemble a floating point number from a mantissa and an exponent. For the sake of brevity, this part is expressed here in terms of the standard C-function `ldexp`. The optimal method would be a direct bit-manipulation, using language-specific constructs.

Bibliography

[Bruneton and Neyret 08] Éric Bruneton and Fabrice Neyret. "Precomputed Atmospheric Scattering." *Comput. Graph. Forum. Proceedings of the 19th Eurographics Symposium on Rendering 2008* 27:4 (2008), 1079–1086. Special Issue.

[Chapman 31] Sydney Chapman. "The Absorption and Dissociative or Ionizing Effect of Monochromatic Radiation in an Atmosphere on a Rotating Earth." *Proceedings of the Physical Society* 43:1 (1931), 26. Available online (http://stacks.iop.org/0959-5309/43/i=1/a=305).

[Hoffmann and Preetham 02] Naty Hoffmann and Arcot J. Preetham. "Rendering outdoor light scattering in real time." Technical report, ATI Technologies, Inc., 2002. Available online (http://www.ati.com/developer).

[Klassen 87] R. Victor Klassen. "Modeling the Effect of the Atmosphere on Light." *ACM Trans. Graph.* 6 (1987), 215–237. Available online (http://doi.acm.org/10.1145/35068.35071).

[Kocifaj 96] M. Kocifaj. "Optical Air Mass and Refraction in a Rayleigh Atmosphere." *Contributions of the Astronomical Observatory Skalnate Pleso* 26:1 (1996), 23–30. Available online (http://www.ta3.sk/caosp/Eedition/Abstracts/1996/Vol_26/No_1/pp23-30_abstract.html).

[Nielsen 03] R. S. Nielsen. "Real Time Rendering of Atmospheric Scattering Effects for Flight Simulators." Master's thesis, Informatics and Mathematical Modelling, Technical University of Denmark, 2003. Available online (http://www2.imm.dtu.dk/pubdb/p.php?2554).

[Nishita et al. 93] Tomoyuki Nishita, Takao Sirai, Katsumi Tadamura, and Eihachiro Nakamae. "Display of the Earth Taking into Account Atmospheric Scattering." In *Proceedings of SIGGRAPH '93, Proceedings, Annual Conference Series*, edited by James T. Kajiya, pp. 175–182. New York: ACM Press, 1993. Available online (http://doi.acm.org/10.1145/166117.166140).

[O'Neil 05] Sean O'Neil. "Accurate Atmospheric Scattering." In *GPU Gems 2*, edited by Matt Pharr and Randima Fernando, pp. 253–268. Reading, MA: Addison-Wesley, 2005.

[Schüler 09] Christian Schüler. "An Efficient and Physically Plausible Real-Time Shading Model." In *ShaderX^7: Advanced Rendering Techniques*, edited by Wolfgang Engel, Chapter 2.5, pp. 175–187. Hingham, MA: Charles River Media, 2009.