Vol. 15, No. 3: 183–198

On the Evaluation of the Complex-Valued Exponential Integral

Vincent Pegoraro and Philipp Slusallek
Saarland University

Abstract. Although its applications span a broad scope of scientific fields ranging from applied physics to computer graphics, the exponential integral is a nonelementary special function available in specialized software packages but not in standard libraries, consequently requiring custom implementations on most platforms. In this paper, we provide a concise and comprehensive description of how to evaluate the complex-valued exponential integral. We first introduce some theoretical background on the main characteristics of the function, and outline available third-party proprietary implementations. We then provide an analysis of the various known representations of the function and present an effective algorithm allowing the computation of results within a desired accuracy, together with the corresponding pseudocode in order to facilitate portability onto various systems. An application to the calculation of the closed-form solution to single light scattering in homogeneous participating media illustrates the practical benefits of the provided implementation with the hope that, in the long term, the latter will contribute to standardizing the availability of the complex-valued exponential integral on graphics platforms.
CCS: G.1.2 [Numerical Analysis]: Approximation – Special function approximations

1. Introduction

Introduced by A. M. Legendre in 1811 and later coined with the Ei notation by J. W. L. Glaisher in 1870, the exponential integral appears in the solution of various physical problems in applied sciences, ranging from thermal radiation

1086-7651/06 $0.50 per page

transfer to transient groundwater flow [Lebedev and Silverman 72]. Applications of the complex-valued function also exist within the field of computer graphics where it has been shown to be central to the analytic solutions to the single-scattering air-light integral with general angular distributions of phase functions and light sources [Pegoraro 09], which characterizes the fundamental behavior of light propagation in homogeneous participating media.

Although a standard mathematical entity, the exponential integral belongs to the class of nonelementary special functions, inducing available implementations to be, unfortunately, restricted to specialized mathematics software packages or libraries. Because the function is currently neither part of most standard programming libraries nor available on graphics hardware, custom implementations are consequently required on such platforms. To this end, sample code exists for the real-valued case [Press et al. 93] based on the various representations of the function [Tseng and Lee 98]. Alas, to the best of our knowledge, there seems to be no such reference document for the complex-valued case, despite the wealth of knowledge available on the characteristics of this well-studied function [Wolfram Research 11].

In this paper, we provide a concise and comprehensive description of how to evaluate the complex-valued exponential integral. We first introduce some theoretical background on the main characteristics of the function, followed by an outline of available third-party proprietary implementations. We then provide a mathematical analysis synthesizing the strengths and weaknesses of the various known representations of the function and present a robust and efficient algorithm allowing the computation of results to within a desired accuracy, together with the corresponding pseudocode in order to facilitate portability onto various systems. The latter is then applied to solving the air-light integral in closed form, illustrating the practical benefits over an existing preliminary implementation while maintaining competitive performance characteristics. Over the long term, we hope this paper will contribute to promoting the availability of the complex-valued exponential integral on graphics platforms as a standard tool, providing a real-time, high-quality alternative to the simplistic exponential-based OpenGL fog model.

2. Function Characteristics

Noting the imaginary unit $\imath^2 = -1$, let us recall that a complex number $z = x + \imath y \in \mathbb{C}$ is defined in Cartesian coordinates by its real $x = \Re(z) \in \mathbb{R}$ and imaginary $y = \Im(z) \in \mathbb{R}$ parts. Alternatively, its polar form $z = re^{\imath\varphi}$ is formulated in terms of its modulus $r = |z| = \sqrt{x^2 + y^2} \in [0, \infty)$ and the principal value of its argument $\varphi = \arg(z) = \operatorname{atan2}(y, x) \in (-\pi, \pi]$. Its complex conjugate then reads $\bar{z} = x - \imath y = re^{-\imath\varphi}$, where the complex exponential is defined as $e^z = e^{\Re(z)}\big(\cos(\Im(z)) + \imath\sin(\Im(z))\big)$ while the definition of the complex logarithm directly follows from the polar form $\ln(z) = \ln(|z|) + \imath\arg(z)$.

Using the above notation, the complex-valued exponential integral $\mathrm{Ei}(z) : \mathbb{C} \to \mathbb{C}$ [Abramowitz and Stegun 72] is defined as

$$\mathrm{Ei}(z) = -\int_{-z}^{\infty} \frac{e^{-t}}{t}\mathrm{d}t, \qquad |\arg(z)| < \pi \tag{1}$$

such that

$$\frac{\mathrm{d}\,\mathrm{Ei}(z)}{\mathrm{d}z} = \frac{e^z}{z}. \tag{2}$$

In the real-valued case, the function $\mathrm{Ei}(x) : \mathbb{R} \to \mathbb{R}$ may be formulated as

$$\mathrm{Ei}(x) = ⨍_{-\infty}^{x} \frac{e^t}{t}\mathrm{d}t, \tag{3}$$

where $⨍$ denotes the Cauchy principal value of the singular integral at the origin, more generally defined for a singularity at s as

$$⨍_a^b \frac{f(t)}{t-s}\mathrm{d}t = \lim_{\varepsilon\to+0}\left(\int_a^{s-\varepsilon} \frac{f(t)}{t-s}\mathrm{d}t + \int_{s+\varepsilon}^{b} \frac{f(t)}{t-s}\mathrm{d}t\right), \qquad a < s < b. \tag{4}$$

As illustrated in Figure 1, the function satisfies the identity $\mathrm{Ei}(\bar{z}) = \overline{\mathrm{Ei}(z)}$ and exhibits a branch cut discontinuity in the complex plane on both sides of the negative real axis such that

$$\lim_{y\to 0} \mathrm{Ei}(x + \imath y) = \mathrm{Ei}(x) + \mathrm{sgn}(y)\imath\pi H(-x), \tag{5}$$

with the Heaviside step function defined as

$$H(x) = \frac{1 + \mathrm{sgn}(x)}{2} = \begin{cases} 0 & \text{if } x < 0 \\ \frac{1}{2} & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases}, \tag{6}$$

where the signum function reads

$$\mathrm{sgn}(x) = \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases}. \tag{7}$$

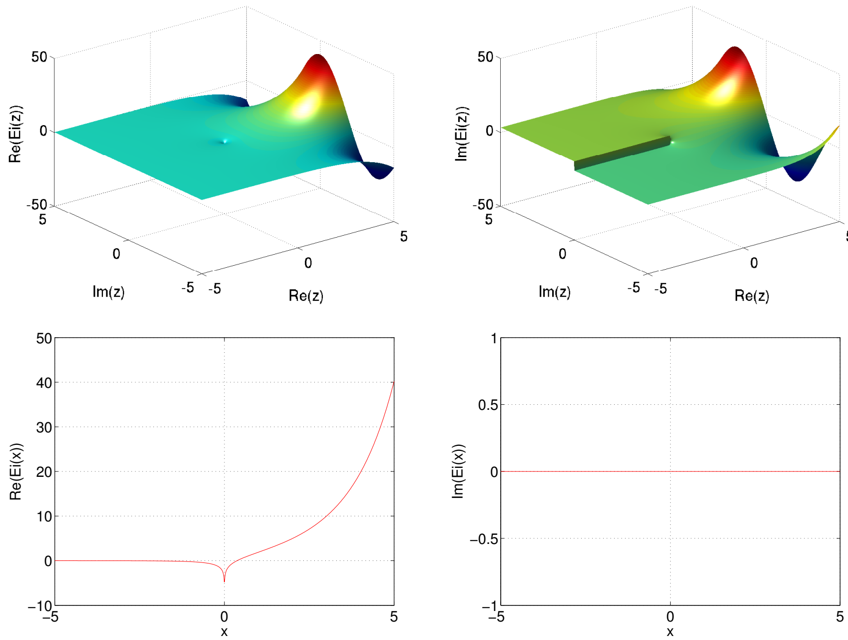


Figure 1. Plots of the real (left-hand column) and imaginary (right-hand column) parts of the exponential integral function in the complex plane (top row) and along the real axis (bottom row).

3. Third-Party Implementations

The Boost C++ libraries provide an implementation of the exponential integral as `expint()`, while the function is available in Maple as `Ei()` and in Mathematica as `ExpIntegralEi[]`. Although the exponential integral per se isn't currently available in MATLAB, the language does provide a closely related entity, namely the Theis well function E_1 (sometimes also called exponential integral) as `expint()`. The identity between the two functions [Wolfram Research 11],

$$\mathrm{Ei}(z) = -\mathrm{E}_1(-z) + \frac{1}{2}\left(\ln(z) - \ln\left(\frac{1}{z}\right)\right) - \ln(-z), \tag{8}$$

may then be simplified by analyzing the properties of the complex-valued logarithm, resulting in the following formulation:

$$\mathrm{Ei}(z) = -\mathrm{E}_1(-z) - \operatorname{sgn}\left(\arg(-z)\right)\imath\pi, \tag{9}$$

which yields the implementation provided in Figure 2. As an alternative, the Symbolic Math Toolbox may also be used to numerically evaluate the function via `mfun('Ei', z)`. However, we have found this approach to be approximately an order of magnitude slower than the implementation introduced above, relying on standard MATLAB routines.

Ei(z)

```
1. if     (imag(z) > 0) return - E1(- z) + i * PI;
2. elseif (imag(z) < 0) return - E1(- z) - i * PI;
3. elseif (real(z) > 0) return - E1(- z) - i * PI;
4. else                 return - E1(- z);
```

Figure 2. Pseudocode for evaluating the complex-valued exponential integral using the Theis well function.

4. Implementing the Complex-Valued Exponential Integral

This section first provides an analysis of the various known representations of the complex-valued exponential integral before outlining the limit behavior of the function. The latter are then synthesized in order to present a robust and efficient algorithm, allowing the computation of results within a desired accuracy, together with pseudocode facilitating portability onto various systems.

4.1. Power Series Representation

Expanding the integrand in a Taylor series about the origin, and integrating term by term yields the convergent power series representation [Abramowitz and Stegun 72], which reads

$$\mathrm{Ei}(z) = \gamma + \ln(z) - \imath\pi \left\lfloor \frac{\arg(z) + \pi}{2\pi} \right\rfloor + \sum_{k=1}^{\infty} \frac{z^k}{k\, k!}, \tag{10}$$

with the Euler-Mascheroni constant [Havil 10] (also referred to as the Euler gamma constant or Euler's constant) defined as

$$\gamma = 0.57721566490153286060651209008240243104215933593992359\ldots \tag{11}$$

The above infinite series typically yields an initially diverging result as the magnitude of the summands accrues during the first stage of the summation process. As the value of the iterator grows, compared with the modulus of the complex input, though, the magnitude of the summands then decreases such that the series is guaranteed to converge as more terms are added. This implies that the result may theoretically be evaluated to an arbitrary accuracy by taking sufficiently many terms in the truncation.

In practice, given that the number of terms required for the result to reach a given precision is directly correlated to the modulus of the complex input z, the power series representation is well behaved and converges rapidly whenever $|z| < 1$. However, the computational cost of the evaluation scheme provided in Figure 3 will increase as $|z| \gg 1$, ultimately leading to numerical instabilities.

EiPowerSeries(z)

```
ei = GAMMA + log(mod(z)) + sign(imag(z)) * i * abs(arg(z));
tmp = 1;
for(k=1; k<=MAX_ITERATIONS; k++)
    tmp *= z / k;
    old = ei;
    ei += tmp / k;
    if (converged(ei, old)) break;
return ei;
```

Figure 3. Pseudocode for evaluating the complex-valued exponential integral using its power series representation.

4.2. Asymptotic Series Representation

Successively integrating by parts the definition of the exponential integral yields the divergent asymptotic series [Bleistein and Handelsman 86]:

$$\mathrm{Ei}(z) = \mathrm{sgn}\left(\Im(z)\right)\imath\pi + e^z \sum_{k=1}^{K} \frac{k!}{k\, z^k} + R_K(z), \tag{12}$$

with the remainder given as

$$R_K(z) = -(-1)^K K! \int_{-z}^{\infty} \frac{e^{-t}}{t^{K+1}} \mathrm{d}t. \tag{13}$$

The asymptotic series representation converges more and more rapidly the larger the modulus of the complex input z is. The number of terms required for the result to reach a desired precision consequently decreases as $|z| \gg 1$, inducing the computational cost of the evaluation scheme provided in Figure 4 to diminish accordingly.

Conversely to the power series, the asymptotic series typically yields an initially converging result as the magnitude of the summands shrinks during the first stage of the summation process. As the value of the iterator increases compared to the modulus of the complex input, though, the magnitude of the summands then accrues causing the series to ultimately diverge as more terms are added. In practice, the maximum number of terms that may be considered in the truncation is consequently defined as the greatest integer guaranteeing a unit bound on the ratio of the magnitude of the k^{th} and $k-1^{\text{th}}$ successive terms in the series $\frac{k-1}{|z|} \leq 1$, yielding $K = \lfloor |z| \rfloor + 1$.

This, in turn, implies a limitation on the expected precision of the result for a given complex input. Unfortunately, we are currently not aware of any reliable indicator of the achievable relative accuracy allowing to predict beforehand whether a result can be computed to within machine precision (although a study on the absolute error did appear elsewhere [Pegoraro 09]), whose potential must then be determined by trial and error. Nevertheless, we have experimentally observed that the domain of the complex plane for which the asymptotic series does converge appears to lie approximately beyond a circle centered at the origin, whose radius behaves logarithmically with the tolerable relative error ε (which correlates with the real-valued case [Press et al. 93]). However, further empirical analysis revealed the radius to more closely match the adjusted formula $2 - 1.035\ln(\varepsilon)$, which may then be used as a heuristic to preliminary discard trial-and-error attempts with complex inputs for which the asymptotic series is likely to fail.

EiAsymptoticSeries(z)

```
ei = sign(imag(z)) * i * PI;
tmp = exp(z) / z;
for(k=1; k<=floor(mod(z))+1; k++)
    old = ei;
    ei += tmp;
    if (converged(ei, old)) return ei;
    tmp *= k / z;
return CONVERGENCE_FAILURE;
```

Figure 4. Pseudocode for evaluating the complex-valued exponential integral using its asymptotic series representation.

4.3. Continued Fraction Representation

Defining the Kettenbruch operator in terms of its partial numerators a_k and denominators b_k with $k \in \mathbb{N}$ as

$$\operatorname*{K}_{k=1}^{n} \frac{a_k}{b_k} = \cfrac{a_1}{b_1 + \cfrac{a_2}{b_2 + \cfrac{\ddots}{\ddots + \cfrac{a_n}{b_n}}}}, \tag{14}$$

the exponential integral may also be represented, whenever $z \notin [0, \infty)$, as a continued fraction whose extended form reads [Cuyt et al. 08]

$$\begin{aligned} \mathrm{Ei}(z) &= \operatorname{sgn}\big(\Im(z)\big)\imath\pi - \frac{e^z}{-z + \operatorname{K}_{k=1}^{\infty} \frac{\lfloor \frac{k+1}{2} \rfloor}{(-z)^{(k+1) \bmod 2}}} & (15) \\ &= \operatorname{sgn}\big(\Im(z)\big)\imath\pi + \cfrac{-\,e^z}{-z + \cfrac{1}{1 + \cfrac{1}{-z + \cfrac{2}{1 + \cfrac{2}{-z + \cfrac{3}{1 + \cfrac{3}{-z + \ddots}}}}}}}, & (16) \end{aligned}$$

while its twice-as-rapidly converging even contraction is given by

$$\begin{aligned} \mathrm{Ei}(z) &= \operatorname{sgn}\big(\Im(z)\big)\imath\pi - \frac{e^z}{1 - z + \operatorname{K}_{k=1}^{\infty} \frac{-k^2}{2k+1-z}} & (17) \\ &= \operatorname{sgn}\big(\Im(z)\big)\imath\pi + \cfrac{-\,e^z}{1 - z + \cfrac{-\,1}{3 - z + \cfrac{-\,4}{5 - z + \cfrac{-\,9}{7 - z + \cfrac{-\,16}{9 - z + \ddots}}}}}. & (18) \end{aligned}$$

The above infinite continued fractions converge more and more rapidly as the complex input z becomes distant from the positive real axis, consequently reducing the number of terms required for the result to achieve a given precision. Although the result could be evaluated to an arbitrary accuracy by taking sufficiently many terms in the approximant, the computational cost increases accordingly, ultimately making the representation impractical in the vicinity of the positive reals.

A finite or truncated infinite continued fraction might be most efficiently computed via backward recursion, iteratively calculating the terms in a bottom-up fashion. This approach leads to the evaluation scheme provided in Figure 5, which, however, requires knowing in advance how many terms are needed in order for the result to reach a given accuracy.

EiContinuedFractionBackward(z)

```
ei = 0;
for(k=MAX_ITERATIONS; k>=1; k--)
    ei = - k * k / (2 * k + 1 - z + ei);
ei = sign(imag(z)) * i * PI - exp(z) / (1 - z + ei);
return ei;
```

Figure 5. Pseudocode for the backward evaluation of the complex-valued exponential integral (excluding along the positive real axis), using the even contraction of its continued fraction representation.

As an alternative to determining this number by trial and error if it is a priori unknown, the continued fraction may be computed in a top-down fashion by noting that the n^{th} convergent of a generalized continued fraction,

$$K_n = b_0 + \operatorname*{K}_{k=1}^{n} \frac{a_k}{b_k}, \tag{19}$$

represents a rational expression $K_n = \frac{A_n}{B_n}$, whose n^{th} continuants A_n and B_n satisfy, as may be easily proven by induction, the recurrence relations defined by the following second-order linear difference equations [Khrushchev 08]:

$$A_{-1} = 1; \qquad A_0 = b_0; \qquad A_n = b_n A_{n-1} + a_n A_{n-2}, \ \forall n \in \mathbb{N}^* \tag{20}$$

$$B_{-1} = 0; \qquad B_0 = 1; \qquad B_n = b_n B_{n-1} + a_n B_{n-2}, \ \forall n \in \mathbb{N}^*. \tag{21}$$

Because the canonical numerator A_n and denominator B_n typically tend to grow or shrink rapidly and would require rescaling in order to avoid numerical overflow or underflow, the result can be more robustly computed using Lentz's algorithm [Press et al. 93], which defines $C_n = \frac{A_n}{A_{n-1}}$ and $D_n = \frac{B_n}{B_{n-1}}$ such that $K_n = \frac{C_n}{D_n} K_{n-1}, \forall n \in \mathbb{N}^*$, yielding the first-order nonlinear recursions:

$$C_0 = b_0; \qquad C_n = b_n + \frac{a_n}{C_{n-1}}, \ \forall n \in \mathbb{N}^* \tag{22}$$

$$D_0 = \infty; \qquad D_n = b_n + \frac{a_n}{D_{n-1}}, \ \forall n \in \mathbb{N}^*. \tag{23}$$

In the particular case where the complex input to the exponential integral is along the (negative) real axis, special care must additionally be taken because $b_0 = 0$, leading to an undefined NaN floating-point value of K_1. A modified version of Lentz's algorithm addresses this general issue by shifting the offending terms by a small amount [Press et al. 93]. In order to avoid resorting to ad hoc parameters, though, we instead consider the limit behavior for such zero-valued initial term and explicitly encode the result. Special handling of terms with infinite modulus may then be avoided by manipulating the inverse of the C_n and D_n coefficients rather than the variables themselves, yielding the evaluation scheme provided in Figure 6.

4.4. Limits

The limit behavior of the exponential integral near the origin is given by

$$\lim_{|z|\to 0} \mathrm{Ei}(z) = \gamma + \ln(z) - \imath\pi \left\lfloor \frac{\arg(z) + \pi}{2\pi} \right\rfloor, \tag{24}$$

such that the function exhibits a singularity at zero

$$\mathrm{Ei}(0) = -\infty. \tag{25}$$

Conversely, its asymptotic limits are given by

$$\lim_{|z|\to\infty} \mathrm{Ei}(z) = \mathrm{sgn}\left(\Im(z)\right)\imath\pi + \frac{e^z}{z}, \tag{26}$$

from which it follows that

$$\mathrm{Ei}(z) = \mathrm{sgn}\left(\Im(z)\right)\imath\pi, \quad \forall z \in \mathbb{C} : |z| = \infty \wedge |\arg(z)| \geq \frac{\pi}{2} \tag{27}$$

$$\mathrm{Ei}(+\infty) = +\infty. \tag{28}$$

4.5. Evaluation Algorithm

Each representation can finally be exploited where it performs best by partitioning the complex plane accordingly. That is, if the complex input z fulfills

EiContinuedFractionForward(z)

```
ei = sign(imag(z)) * i * PI;
if (ei != 0)
    c = 1 / ei;
    d = 0;
    c = 1 / (1 - z - exp(z) * c);
    d = 1 / (1 - z - exp(z) * d);
    ei *= d / c;
else
    c = INF;
    d = 0;
    c = 0;
    d = 1 / (1 - z - exp(z) * d);
    ei = d * (- exp(z));
for(k=1; k<=MAX_ITERATIONS; k++)
    c = 1 / (2 * k + 1 - z - k * k * c);
    d = 1 / (2 * k + 1 - z - k * k * d);
    old = ei;
    ei *= d / c;
    if (converged(ei, old)) break;
return ei;
```

Figure 6. Pseudocode for the forward evaluation of the complex-valued exponential integral (excluding along the positive real axis), using the even contraction of its continued fraction representation.

the radial heuristic described in Section 4.2, the exponential integral may be tentatively evaluated using its computationally efficient asymptotic series representation. To partition the rest of the domain, we then rely on a simple heuristic stemming from the observations made in Sections 4.1 and 4.3, such that the result is evaluated using the continued fraction representation whenever z is farther than a fixed axial distance away from the positive reals, whereas the power series is used otherwise. Defining the convergence test as an assertion on the relative error of both the real and imaginary parts of the result, as illustrated in Figure 7, we then present two variants of the algorithm provided in Figure 8, which additionally handles limit parameters.

The first variant is meant to be a generic scheme allowing the exponential integral to be evaluated to an arbitrary accuracy over the complex plane by setting the maximum tolerable relative error `MAX_ERROR` accordingly. As such, the maximum number of iterations is set to infinity `MAX_ITERATIONS = INF`,

converged(ei, old)

```
return abs(real(ei - old)) <= MAX_ERROR * abs(real(ei))
    && abs(imag(ei - old)) <= MAX_ERROR * abs(imag(ei));
```

Figure 7. Pseudocode for the convergence test defined as an assertion on the relative error of both the real and imaginary parts of the result.

Ei(z)

```
if (mod(z) == +INF)
    return sign(imag(z)) * i * PI + exp(z) / z;
if (mod(z) > 2 - 1.035 * log(MAX_ERROR))
    ei = EiAsymptoticSeries(z);
    if (ei != CONVERGENCE_FAILURE)
        return ei;
if (mod(z) > DIST && (real(z) < 0 || abs(imag(z)) > DIST))
    return EiContinuedFraction(z);
if (mod(z) > 0)
    return EiPowerSeries(z);
if (mod(z) == 0)
    return -INF;
```

Figure 8. Pseudocode for the algorithm evaluating the complex-valued exponential integral, using both its limits and its various representations.

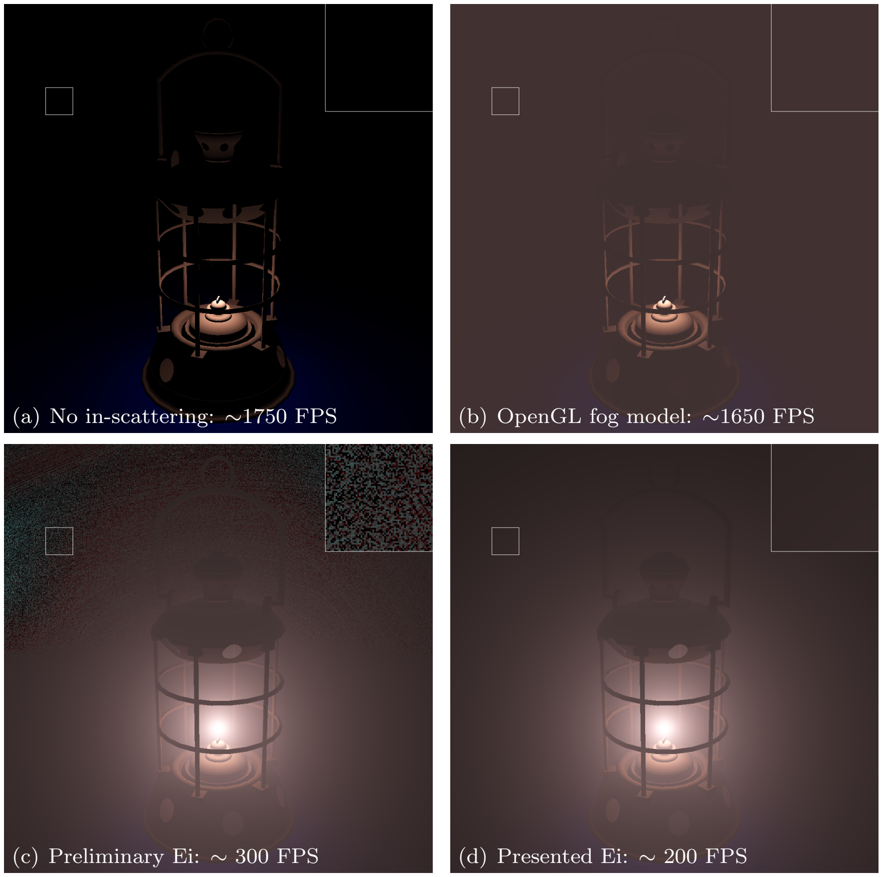
and the continued fraction representation is evaluated in a forward fashion according to `EiContinuedFractionForward()`. Following the observations made in Sections 4.1, the axial distance is then conservatively defined as `DIST = 1`.

Given a predefined precision requirement `MAX_ERROR`, the second variant is an optimized alternative dedicated to the complex half-plane $\Re(z) \leq 0$ of sole interest when solving the air-light integral. Assuming the maximum number of terms needed for the result to reach the required accuracy over the corresponding region of the domain has been a priori determined [Zhang and Jin 96], the scheme evaluates the continued fraction representation using its substantially more efficient backward recurrence form `EiContinuedFractionBackward()`. To balance the computation between the two representations and to minimize the worst case scenario, the constraint previously defined conservatively on the axial distance is relaxed in order to make the maximum numbers of iterations required by the power series and the continued fraction match, under the reasonable assumption that both entail similar computational costs. To this end, we have experimentally determined that results exact up to machine precision could be computed within `MAX_ITERATIONS = 24` by both representations with an axial distance of `DIST = 3.46` for a single-precision floating-point format in the order of 10^{-8}, and within `MAX_ITERATIONS = 47` with an axial distance of `DIST = 6.6` for a double-precision floating-point format in the order of 10^{-16}, respectively, although the specific values might be architecture dependent.

5. Example Application

The algorithm can be easily implemented in high-level object-oriented programming languages with the help of a class handling basic complex-number arithmetic operations, or by exploiting the intrinsic functionalities of modern shading languages such as Cg whose `float2` type readily provides support for multiplication and division of complex numbers by scalars, as well as for addition, subtraction, and even comparison between complex numbers themselves. Additionally, the `length()` function can be used to compute the modulus of a complex input `z` and the square of the latter readily given by `dot(z, z)`. Custom helper routines may then be defined to provide support for missing functionalities, such as multiplication or division (using the inverse identity $\frac{1}{z} = \frac{\bar{z}}{|z|^2}, \forall z \neq 0$) between complex numbers.

Figure 9 illustrates an application within the field of computer graphics where the exponential integral appears in the analytic solutions to the air-light integral. The single-scattering contributions were computed independently for each color channel (i.e., three times per fragment), at a resolution of 768×768, using a Cg-based implementation running on an NVIDIA GeForce GTX 280. Compared with the absence of in-scattering contributions, the simplistic exponential-based OpenGL fog model is adequate to efficiently render diffusely lit fog in outdoor scenes, however, it fails to simulate the propagation of light emanating from punctual light sources. In contrast, evaluating the air-light contributions in closed form, using the exponential integral, faithfully reproduces the interaction of light with the participating medium, consequently allowing the generation of more realistic images. To this end, the preliminary implementation of the function that appeared elsewhere [Pegoraro 09] may occasionally lead to numerical instabilities because it relies solely on the power series representation whenever convergence might not be reached using the asymptotic series. Alternatively, the results can be more robustly evaluated using the implementation presented in this paper, while maintaining competitive performance characteristics.



(a) No in-scattering: ∼1750 FPS

(b) OpenGL fog model: ∼1650 FPS

(c) Preliminary Ei: ∼ 300 FPS

(d) Presented Ei: ∼ 200 FPS

Figure 9. Synthesis of the image of a lantern as a sample application, rendered (a) without in-scattering contributions, (b) with the exponential-based OpenGL fog model, (c) with a preliminary implementation of the exponential integral, and (d) with the implementation of the exponential integral presented in this paper.

6. Discussion

Regarding the implementation, the evaluation schemes presented in this paper assume the availability of support for complex numbers, which we have found to be less prone to numerical errors than using a polar formulation as preliminary proposed [Pegoraro 09]. The approach is also generally more efficient because it avoids the repeated evaluation of computationally costly transcendental functions, although hardware support for trigonometric functions might invalidate this statement. Performance is, overall, most critical in the vicinity of the positive real axis that is away from the origin and not handled by the asymptotic series, where none of the various representations is really satisfactory. Additionally, in our experience, evaluating the continued fraction in a forward fashion might not be numerically insensitive, and it therefore appears to be more adequate for use with a double-precision rather than single-precision floating-point format.

From a more theoretical standpoint, and to the best of our knowledge, there seems to be very little information available in the literature on how to determine which representation is optimal for a given complex input, despite the fact that each has been individually well studied. Thus, the heuristics presented in this paper mostly stem from experimentation with the function in the course of our work on deriving analytic solutions to the air-light integral. Although they appear to be reasonable in practice, the availability of analytically derived transition rules should presumably allow for a more effective use of the various representations. Ultimately, gaining insights from leading experts in the field and/or the publication of proprietary implementations could substantially advance the process of standardizing the availability of the complex-valued exponential integral, including on graphics platforms.

Acknowledgments. This work was supported by the German Research Foundation (DFG) via the Cluster of Excellence on Multimodal Computing and Interaction, funded within the Excellence Initiative. Lantern model courtesy of Scott Graham.

References

[Abramowitz and Stegun 72] Milton Abramowitz and Irene A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables.* Washington, DC, United States of America: U.S. Department of Commerce, 1972.

[Bleistein and Handelsman 86] Norman Bleistein and Richard A. Handelsman. *Asymptotic Expansions of Integrals.* Mineola, NY, United States of America: Dover, 1986.

[Cuyt et al. 08] Annie Cuyt, Vigdis Brevik Petersen, Brigitte Verdonk, Haakon Waadeland, and William B. Jones. *Handbook of Continued Fractions for Special Functions.* Heidelberg, Germany: Springer, 2008.

[Havil 10] Julian Havil. *Gamma: Exploring Euler's Constant.* Princeton, NJ, United States of America: Princeton University Press, 2010.

[Khrushchev 08] Sergey Khrushchev. *Orthogonal Polynomials and Continued Fractions: From Euler's Point of View.* Cambridge, United Kingdom: Cambridge University Press, 2008.

[Lebedev and Silverman 72] Nikolaĭ Nikolaevich Lebedev and Richard A. Silverman. *Special Functions and their Applications*. Mineola, NY, United States of America: Courier Dover Publications, 1972.

[Pegoraro 09] Vincent Pegoraro. "Efficient Physically-Based Simulation of Light Transport in Participating Media." Ph.D. thesis, School of Computing, University of Utah, Salt Lake City, UT, United States of America, 2009.

[Press et al. 93] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C, The Art of Scientific Computing*, Second edition. Cambridge, United Kingdom: Cambridge University Press, 1993.

[Tseng and Lee 98] Peng-Hsiang Tseng and Tien-Chang Lee. "Numerical Evaluation of Exponential Integral: Theis Well Function Approximation." *Journal of Hydrology* 205:1–2 (1998), 38–51.

[Wolfram Research 11] Wolfram Research. "Exponential Integral Ei." http://functions.wolfram.com/GammaBetaErf/ExpIntegralEi/, 2011.

[Zhang and Jin 96] Shanjie Zhang and Jianming Jin. *Computation of Special Functions*. Chichester, United Kingdom: John Wiley and Sons, 1996.

Web Information:

http://jgt.akpeters.com/

Vincent Pegoraro, Saarland University & M2CI, Computer Graphics Laboratory, Campus E1 1, 66123 Saarbrücken, Germany (<lastname>@cs.uni-saarland.de)

Philipp Slusallek, Saarland University & DFKI, Computer Graphics Laboratory, Campus E1 1, 66123 Saarbrücken, Germany (<lastname>@cs.uni-saarland.de)

Received June 21, 2011; accepted July 24, 2011.