

# Solving Radiance Transport as a Differential Equation

Derek Nowrouzezahrai      Chris Gonterman

Dynamic Graphics Project, University of Toronto

---

## Abstract

We introduce an alternative to Monte-Carlo techniques for solving radiance transport problems for participating media. We use a reformulation of the volume rendering equation from its standard integro-differential form to a purely differential form. We then leverage the large body of work in numerical methods for solving differential equations by framing and analyzing the problem as a differential equation. To our knowledge, this is the first application of such techniques in the area of photo-realistic rendering of volumes based on ray optics.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Color, shading, shadowing, and texture

---

## 1. Introduction

Accurately rendering scenes with various surface reflection properties and under complex lighting is a challenging problem in computer graphics. This problem is further complicated if a participating medium, such as smoke, is present since light rays from the surfaces and from within the medium itself interact to contribute to the measured radiance along these rays. Many techniques based on Monte-Carlo integration have been proposed for solving this problem [CPCP\*05]. We instead apply numerical methods for solving differential equations (NMDE) and validate the usefulness of such techniques for solving rendering problems. As a consequence of this new formulation we can also leverage the methodologies used in the NMDE literature for carefully analyzing and solving problems while providing guaranteed bounds on the error.

### 1.1. Contributions

Our main contribution is the application of NMDE analysis and algorithms to solving a reformulation of the volume rendering equation [Cha60]. To the best of our knowledge, NMDEs have never been applied to rendering volumetric effects using the *equations of ray optics*. NMDEs literature have previously been used to render refractive materials using the *equation of geometric optics* [IZT\*07, SZS\*08], however no analysis of the efficiency or error of these algorithms was conducted. We will compare the performance

and accuracy of several NMDEs to determine the properties of algorithms well-suited for solving this problem.

## 2. Previous Work

We will overview key contributions in accurate rendering techniques for participating media, as well as NMDEs.

**Rendering techniques for participating media** based on the radiative transfer equation have been studied extensively in computer graphics, starting with the seminal work for ray tracing of Kajiya and von Herzen [KH84], and Rushmeier and Torrance for radiosity [RT87]. Since then, many techniques have strived to increase the generality of produced results to include accurate single and multiple scattering effects for media consisting of both homogeneous and heterogeneous material combinations. Including the effects of isotropic and anisotropic phase distributions within media is also an area of current work. Cerezo et al. [CPCP\*05] provide a thorough survey of the area, and many recent works by Jarosz et al. [JDZJ08, JZJ08a, JZJ08b] have addressed integration, generalization and performance of offline rendering techniques for participating media.

**Numerical methods for solving differential equations** have been studied for nearly a century. Issues surrounding the design of robust single and multi-step algorithms with fixed or adaptive step sizes, the propagation of error in solutions for stiff and non-stiff problems, analyzing the order, performance and convergence of the proposed algorithms,



Figure 1: Results of solvers with guaranteed error bounds.

and trade-offs between these different conditions are among the many sub-areas of research in the field [AP98, Hai00]. While providing a complete literature review is beyond the scope of this paper, we will summarize the design and analysis issues related to our application of these techniques.

### 3. Terminology and Problem Definition

The radiance at a point in a participating medium,  $L(\mathbf{x}, \vec{w})$ , is described by the radiative transfer equation [Cha60]

$$(\vec{w} \cdot \nabla)L(\mathbf{x}, \vec{w}) + \sigma_t(\mathbf{x})L(\mathbf{x}, \vec{w}) = E(\mathbf{x}, \vec{w}) + \sigma_s(\mathbf{x}) \int_{S^2} p(\mathbf{x}, \vec{w}, \vec{w}_i)L(\mathbf{x}, \vec{w}_i)d\vec{w}_i \quad (1)$$

where  $\sigma_t$  and  $\sigma_s$  are the extinction and scattering coefficients,  $p$  is the (potentially anisotropic) phase function,  $E$  is the medium's self-emission, and  $S^2$  is the set of directions on the unit sphere. In order to solve this equation for a particular eye ray,  $\vec{w}_o$ , we parameterize points along the ray as

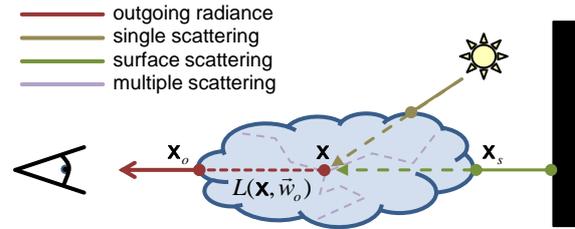


Figure 2: Radiance along rays in participating media.

$\mathbf{x} = \mathbf{x}_o + t \cdot \vec{w}_o$  (see Fig. 2) and equation 1 becomes

$$L'(t) = E(t) - \sigma_t(t)L(t) + \sigma_s(t) \int_{S^2} p(t, \vec{w}_i)L(t, \vec{w}_i)d\vec{w}_i \equiv F(L(t), t) . \quad (2)$$

As in previous works,  $t$  is integrated only along points within the volume, and includes the closest surface radiance in-

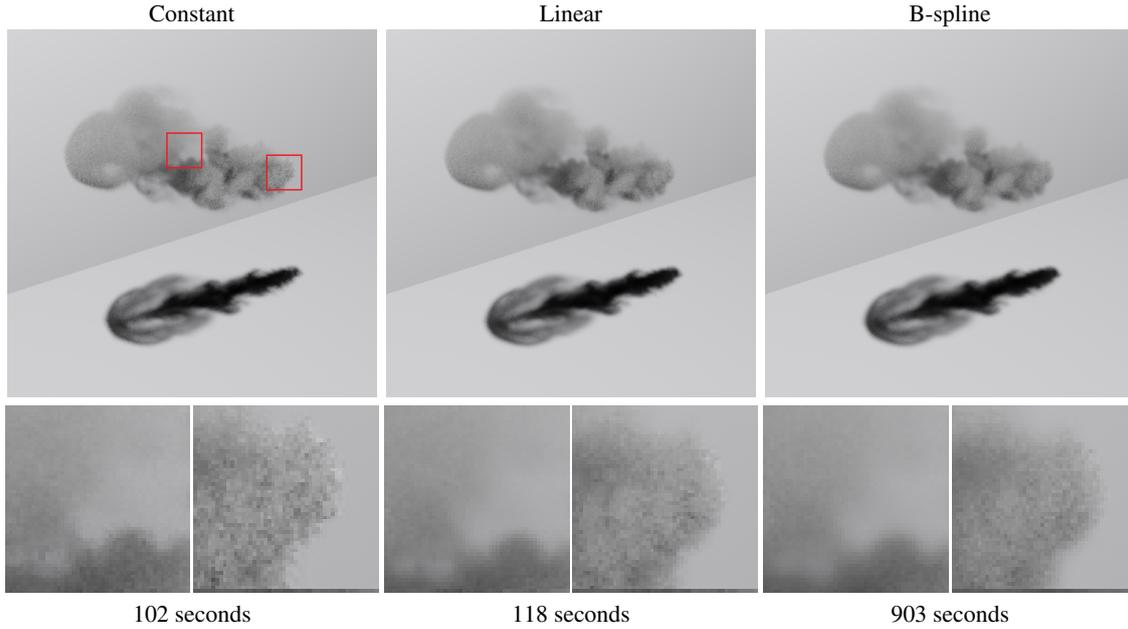


Figure 3: Effects of different interpolants during integration.

intersected within or outside the volume ( $\mathbf{x}_s$ .) At this point we break away from previous approaches which continue to manipulate equation 2 until the more common integro-differential form of the volume rendering equation is obtained and solved using Monte Carlo integration. In Section 4 we will overview the different classes of NMDE integrators which may be appropriate for solving this problem, and we will discuss the potential benefits and trade-offs of using these techniques as opposed to Monte Carlo techniques.

#### 4. Choosing an Appropriate Solver

Equation 2 defines a 1<sup>st</sup> order scalar initial value problem (IVP) where the solution  $L(t)$  is defined along a ray within the volume with an initial value of  $L(\mathbf{x}_s, \vec{w}_o)$ , and integration occurs towards the eye over the interval  $[\mathbf{x}_s, \mathbf{x}_o]$ .

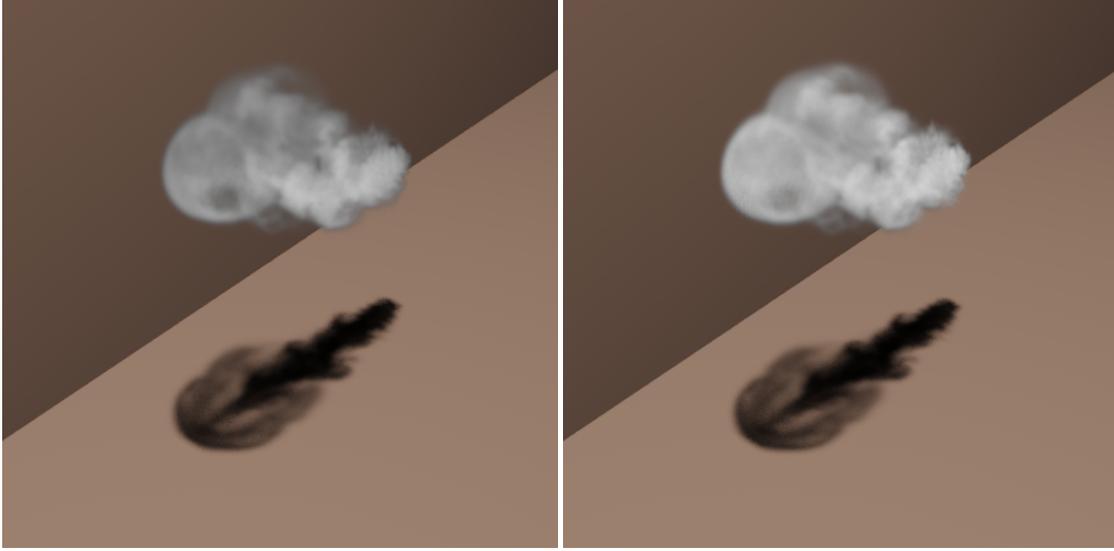
Furthermore, the solution  $L(t)$  does not typically exhibit *stiffness*. In other words, the radiance along the ray varies smoothly within the volume regardless of the radiance at the edge of the volume (e.g.  $L(\mathbf{x}_s, \vec{w}_o)$ .) Thus, the more expensive class of *implicit* integrators are not necessary and we instead use *explicit* integrators which do not need to compute the solution of an algebraic system at every step along the ray,  $L(t_{i+1})$ . We will now discuss the remaining factors necessary for picking a suitable solver: differences between one-step and multi-step methods, the order of the method, and fixed versus adaptive step sizes.

**One-step methods** do not use any of the previously calculated points along the ray when calculating the next

point, whereas **multi-step methods** do use a (sometimes variable) number of previously calculated points (e.g.  $\{L(t_{i-1}), \dots, L(t_{i-n})\}$ .) Choosing between these two types of solvers depends on the cost of evaluating the differential equation, the accuracy requirements, and the smoothness of the underlying problem, and there is no rigorous way to determine a-priori which technique is most suitable for a given problem. Therefore, we will experiment with both one-step and multi-step methods in Section 6 in order to determine which is more suitable in our case.

**Order of a solver** relates the total error and error introduced per step along the ray to the size of the step. Roughly speaking, if the solution has continuous high-order derivatives (see Section 4.1), then the error per-step of a  $p^{\text{th}}$ -order method will be  $\mathcal{O}(h^{p+1})$  and the total error is  $\mathcal{O}(h^p)$ . As with the choice between one-step and multi-step approaches, choosing the order of your solver depends on accuracy requirements, smoothness of the problem, and cost trade-offs. Section 6 will also compare solvers with various orders.

**Step sizes** can either be fixed or adaptive. Fixed size approaches are easier to implement and, with a small enough step size, can yield accurate results. However, adaptive step size approaches explicitly monitor the error introduced per step and use the data to increase the step size where suitable (e.g. where the solution varies slowly.) Typically, adaptive approaches are more efficient as the cost of performing the per-step error analysis and step size adjustment is justified, for both increasing the accuracy and performance.



**Figure 4:** A smoke dataset exhibiting both emission and scattering effects. (Left) Rendered with an emission-only integrator using Equation 3. (Right) Rendered with a single-scattering integrator using Equation 4.

Monte Carlo techniques in graphics with variable step-sizing heuristically divide the steps based on contrast or variance estimates between adjacent samples, whereas the adaptive Runge-Kutta solver we introduce in Section 4 divides the ray based on maintaining a guarantee on the error tolerance with dynamically computed error estimates (see the Appendix for the mathematical exposition.)

#### 4.1. Data Continuity Requirements

As mentioned earlier, many of the NMDE approaches make error and convergence guarantees based on assumptions on the behavior of the solution. Namely, the existence of continuous higher-order derivatives is necessary. We experimented with piece-wise constant, linear and b-spline interpolants of the spatially varying properties ( $\sigma_t(t)$ ,  $\sigma_s(t)$ ,  $p(t, \vec{w})$ , and  $E(t)$ ). Figure 3 illustrates the effects on the quality and performance of an adaptive solver with these interpolants. The effects of using these different interpolants on the behavior and performance of the different solvers will be discussed. Using the B-spline interpolant increased the number of steps required to converge using an adaptive Runge-Kutta solver, however it resulted in a much smoother result (with the same tolerance used across integrators:  $tol = 0.005$ .)

#### 4.2. Setting Tolerance

With NMDEs, we can set an error tolerance to be maintained throughout integration. We experimented with various tolerances and found that stringent settings only improve numerical convergence, not visual convergence. We use  $tol = 10^{-2}$

whereas other simulations, such as those of large physical bodies, require settings on the order of  $10^{-7}$  [AP98].

### 5. Solving the Radiance Transport Equation

We will discuss the details involved in applying different NMDE integrators to solving various forms of Equation 2.

**Emission Only:** The simplest form of participating media is one that only absorbs and emits light, but does not scatter any. In this case, Equation 2 can be simplified to

$$L'(t) = E(t) - \sigma_t(t)L(t) \quad (3)$$

and serves as a starting point for experimentation since robust algorithms exist in the NMDE and graphics literatures for solving this type of problem.

**Single Scattering:** In order to include scattering events, the integrated incident radiance at each point along the ray is simplified to only include single scattering events: light that has only scattered once within the volume at  $\mathbf{x}$ ,  $L_i(\mathbf{x}, \vec{w}_i)$ , replaces the  $L(\mathbf{x}, \vec{w}_i)$  term in the integrand of Equation 2.

$$L'(t) + \sigma_t(t)L(t) = E(t) + \sigma_s(t) \int_{\Omega^2} p(t, \vec{w}_i) L_i(t, \vec{w}_i) d\vec{w}_i. \quad (4)$$

We use an NMDE solver to integrate along each direction,  $\vec{w}_i$ , accounting for the light that bounces towards  $\mathbf{x}$  after a single scattering event. We experimented with different solvers and tolerance settings for the separate integrals along  $L$  and  $L_i$ , however we found it difficult to predict the effects on the results when multiple solvers or tolerance settings

were coupled. Moreover, we are able to obtain high-quality results using a single solver for both components of the solution.

Figure 4 compares and illustrates the effects of scattering and emission between both using our solvers and a reference path-tracing solution.

### 6. Experimental Results

We will compare the performance and behavior of six integrators on various scenes comprising participating media with different volumetric properties and surrounding geometry. Of these six integrators, three are constant step-sizing techniques (of which one is a multi-step approach), and three are adaptive step-sizing Runge-Kutta techniques. We will give a brief overview of the integrators below.

The **Euler** integrator is the simplest of the six and is very similar to the constant step-sizing algorithms typically used in Monte Carlo integration for participating media. This solver has order 1 and evaluates the differential equation  $F$  once per step along the ray:

$$L(t_{i+1}) = L(t_i) + hF(L(t_i), t_i), \quad (5)$$

where  $h$  is the step size. We will experimentally determine whether the low order of this solver affects the quality of the results it generates in the context of the accuracy requirements for the purposes of generating perceptually correct renderings (see Figure 5).

The second integrator is an order 4 **Runge-Kutta** method. This integrator performs four function evaluations per step:

$$L(t_{i+1}) = L(t_i) + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (6)$$

where

$$\begin{aligned} k_1 &= F(L(t_i), t_i) \\ k_2 &= F(L(t_i) + \frac{h}{2}k_1, t_i + \frac{h}{2}) \\ k_3 &= F(L(t_i) + \frac{h}{2}k_2, t_i + \frac{h}{2}) \\ k_4 &= F(L(t_i) + hk_3, t_i + h). \end{aligned} \quad (7)$$

Of the family of constant-step Runge-Kutta solvers, this solver is particularly attractive since Runge-Kutta formulas with order  $p$  higher than 4 require more than  $p$  function evaluations per step. The constant-step, one-step Euler and  $4^{th}$ -order Runge-Kutta solvers can be characterized in Runge-Kutta tableau form:

$\alpha_1$	$\beta_{11}$	$\dots$	$\beta_{1s}$	$-\frac{1}{2}$	$-\frac{1}{2}$	$-$	$-$	$0$	$1$
$\vdots$	$\vdots$	$\dots$	$\vdots$	$-\frac{1}{2}$	$0$	$-\frac{1}{2}$	$-$	$-$	$-$
$\alpha_s$	$\beta_{s1}$	$\dots$	$\beta_{ss}$	$1$	$0$	$0$	$1$	$-$	$-$
	$\omega_1$	$\dots$	$\omega_s$		$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$	

General Tableau
 $4^{th}$ -order Runge-Kutta Tableau
Euler Tableau

where  $s$  is the number of stages and

$$\begin{aligned} L(t_{i+1}) &= L(t_i) + h(\omega_1 k_1 + \dots + \omega_s k_s) \\ k_j &= F\left(L(t_i) + h \sum_{r=1}^s \beta_{jr} k_r, t_i + h\alpha_j\right). \end{aligned} \quad (8)$$

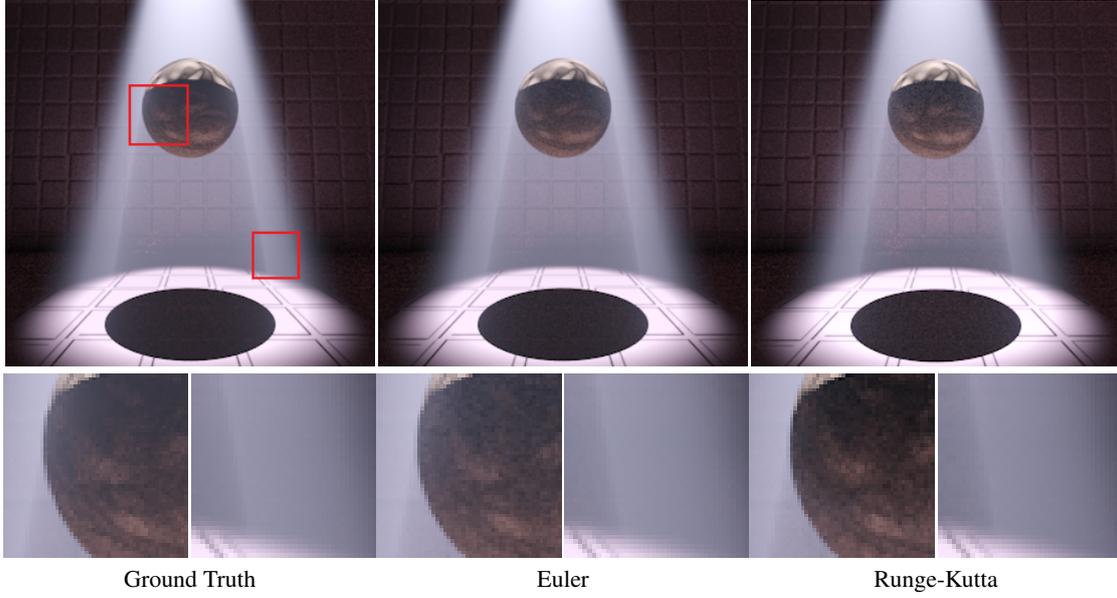
The third constant step-size integrator we implemented is the multi-step algorithm of **Adams-Bashforth**. This integrator has order equal to the number of stages used,  $s$ . The blending weights,  $\beta_j$ , can be obtained by solving a simple integral equation [Hai00] and the solver has the form

$$L(t_{i+1}) = L(t_i) + h \sum_{j=1}^k \beta_j L'(t_{i+1-j}). \quad (9)$$

Apart from requiring  $s - 1$  function evaluations to bootstrap the first iteration, this solver only evaluates  $F$  once per subsequent step and can therefore yield high-order results without incurring large computational penalties (assuming the integration is performed over a number of steps much larger than  $s$ ).

We have implemented three adaptive step-sizing Runge-Kutta solvers. The first is a  $5^{th}$ -order integrator based on the Verner equations [HEJ76] implemented by Hull et al. We also implemented two more  $4^{th}$ -order solvers based on the equation pairs introduced by Fehlberg [E.69] and Cash-Karp [CK90]. Unlike Monte Carlo adaptive step-sizing techniques, these integrators estimate the error at every integration point by taking the difference between the  $p^{th}$ -order estimator and its paired  $(p + 1)^{th}$ -order equation (see the Appendix for mathematical details.) This difference can be used to increase the order of the solver instead without changing the step-size, or to adaptively change the step size without using the higher-order accuracy. While these techniques require more function evaluations than the previous approaches, we will experimentally investigate whether using adaptive step-sizes yields an overall increase in performance. Like the constant step-size Runge-Kutta techniques introduced earlier,  $(p, p + 1)$  Runge-Kutta pair equations can be expressed in a very similar generalized form (see Table 1)

$\alpha_2$	$\beta_{21}$	$-$	$-$	$-$
$\vdots$	$\vdots$	$\dots$	$\vdots$	$\vdots$
$\alpha_s$	$\beta_{s1}$	$\dots$	$\beta_{s-1,s}$	$-$
	$\omega_1$	$\dots$	$\omega_s$	$-$
	$\hat{\omega}_1$	$\dots$	$\hat{\omega}_s$	$-$



**Figure 5:** (Left to right) Ground truth rendering, and equal-time renderings (120 seconds) with the 1<sup>st</sup>-order Euler and 4<sup>th</sup>-order Runge-Kutta constant step-size integrators. As we can see, banding artifacts from insufficient steps are significantly reduced with the Runge-Kutta solver.

$  \begin{array}{r cccccc}  - & - & & & & & \\  1 & 1 & & & & & \\  4 & 4 & - & & & & \\  3 & 3 & 9 & & & & \\  8 & 32 & 32 & - & & & \\  12 & 1932 & -7200 & 7296 & & & \\  13 & 2197 & 2197 & 2197 & - & & \\  & 439 & -8 & 3680 & -845 & & \\  1 & 216 & 513 & 4104 & & - & \\  1 & -8 & -3544 & 1839 & -11 & & \\  2 & 27 & 2 & 4104 & 40 & - & \\  \hline  & 25 & 0 & 1408 & 2197 & -1 & 0 \\  & 216 & 0 & 2565 & 4104 & 5 & 2 \\  & -16 & 0 & 6656 & 28561 & -9 & 2 \\  & 135 & & 12825 & 56430 & 50 & 55  \end{array}  $	$  \begin{array}{r cccccc}  - & - & & & & & \\  1 & 1 & & & & & \\  5 & 5 & - & & & & \\  3 & 3 & 9 & & & & \\  10 & 40 & 40 & - & & & \\  3 & 3 & -9 & 6 & & & \\  5 & 10 & 10 & 5 & - & & \\  1 & -11 & 2 & -70 & 35 & - & \\  7 & 55 & 27 & 27 & 44275 & 253 & \\  8 & 1631 & 175 & 575 & 110592 & 4096 & - \\  \hline  & 55296 & 512 & 13824 & 110592 & 4096 & \\  & 2825 & 0 & 18575 & 13525 & 277 & 1 \\  & 27648 & 0 & 48384 & 55296 & 14336 & 4 \\  & 37 & 0 & 250 & 125 & 0 & 512 \\  & 378 & & 621 & 594 & & 1771  \end{array}  $
Fehlberg Tableau	Cash-Karp Tableau

**Table 1:** Runge-Kutta paired equation tableaus.

where

$$\begin{aligned}
 L(t_{i+1}) &= L(t_i) + h(\omega_1 k_1 + \dots + \omega_s k_s) \\
 \hat{L}(t_{i+1}) &= L(t_i) + h(\hat{\omega}_1 k_1 + \dots + \hat{\omega}_s k_s). \quad (10)
 \end{aligned}$$

For all comparisons, timings only include volume integration calculations (not the time required to perform surface shading) and ground truth renderings are generated using a high-order solver with a small step size. Figure 5 compares equal-time renderings with 1<sup>st</sup>-order Euler and 4<sup>th</sup>-order Runge-Kutta solvers with step size equal to 1% of the minimum volume dimension to a ground truth rendering.

The cost of evaluating  $F$  four times per step in the higher-order one-step solver did not justify the added accuracy when compared to an equal-time rendering of a lower-order one-step technique with reduced step-size. Figure 6 illus-

trates renderings of the 4<sup>th</sup>-order Runge-Kutta and  $k^{\text{th}}$ -order Adams-Bashforth ( $k = 4$ ) solvers to compare one-step and multi-step solvers of equivalent order.

Adams-Bashforth generates better results in less time than a one-step method of equal order, combining the benefits of fewer function evaluations and increased accuracy. Our final experiment compares Adams-Bashforth and adaptive-step Runge-Kutta solvers in Figure 7. Adams-Bashforth outperforms the adaptive step-size solver significantly.

## 7. Conclusions and Future Work

In general, our solvers perform at least as well as standard Monte-Carlo techniques and often converge to less noisy results faster. Multi-step approaches out-perform one-step approaches as well as adaptive step-sizing algorithms of

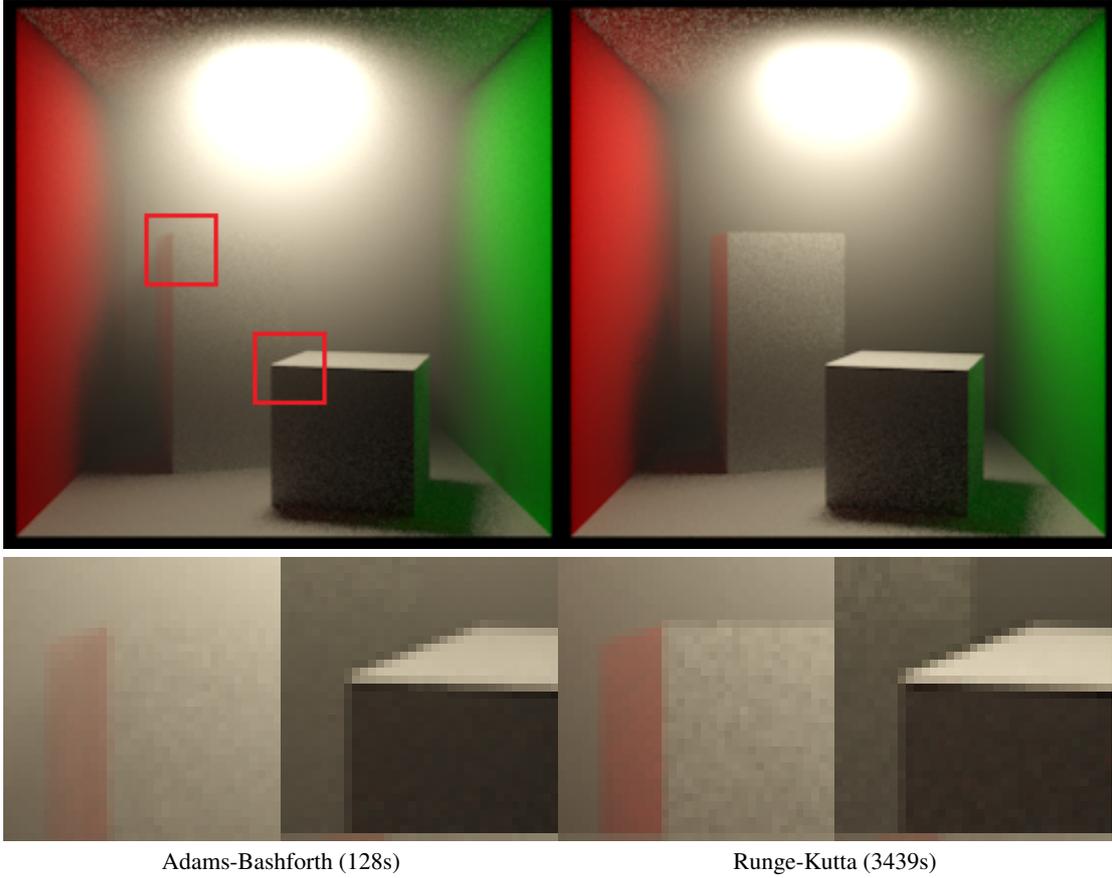


Figure 6: One and multi-step algorithms.

equal order. Adaptively changing the step-size improved results when compared to equivalent constant step-size algorithms. Given these results, we expect adaptive multi-step approaches to be even better suited for this volume rendering application. We are also investigating a hybrid NMDE-photon-mapping approach for simulating multiple-scattering effects.

#### Appendix - Adaptive Step-Size Tolerance Conditions

The adaptive Runge-Kutta solver attempts to adjust the step size as close to the largest value while still maintaining the tolerance criteria:  $\mathcal{O}(h^{p+1})$  error per-step and  $\mathcal{O}(h^p)$  total error. Given the formula pairs of our particular adaptive solver ( $p = 5$ ), we can estimate the current step to  $5^{th}$  and  $6^{th}$ -order accuracy,  $L(t_i)$  and  $\tilde{L}(t_i)$  respectively, as well as estimating the error at this step [HEJ76]

$$\begin{aligned} L(t_i) &= \hat{L}(t_i) + \mathcal{O}(h^{p+2}) \\ \tilde{L}(t_i) &= \hat{L}(t_i) + A(F) h^{p+1} + \mathcal{O}(h^{p+2}) \\ \mathbf{error}_i &= \tilde{L}(t_i) - L(t_i) = -A(F) h^{p+1} \end{aligned}$$

where  $\hat{L}$  is the estimated  $p^{th}$ -order function and  $A(F)$  are the  $(p + 1)^{th}$ -order terms of the Taylor expansion of  $L$ . A scale factor,  $\zeta$ , is required such that during the next step

$$\mathbf{error}_{i+1} = tol h'$$

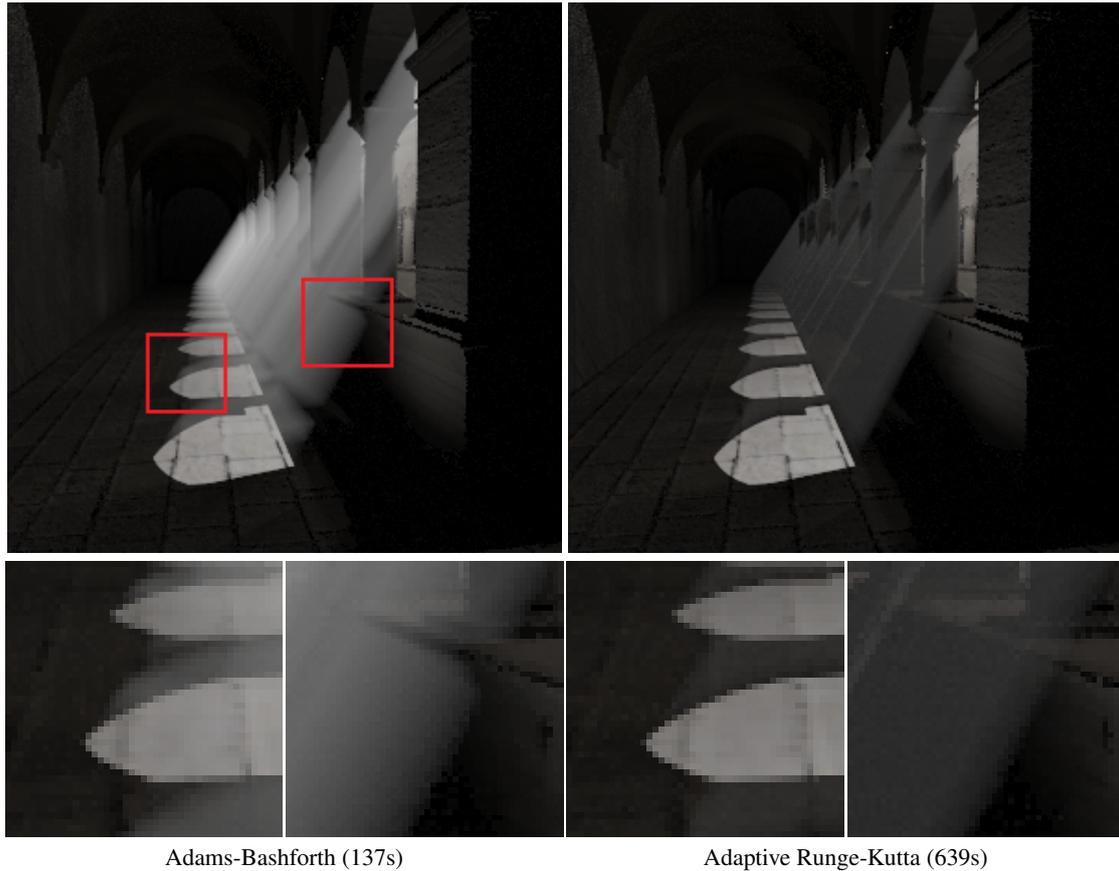
with  $h' = \zeta h$ . Since

$$\mathbf{error}_{i+1} = -A(F) h'^{p+1} = \zeta^{p+1} \mathbf{error}_i$$

then

$$h' = \alpha h \left( \frac{tol h}{|\mathbf{error}_i|} \right)^{1/p}$$

where  $\alpha < 1$  compensates for systematic errors. We use  $\alpha = 0.9$ .



**Figure 7:** Comparing adaptive step-size Runge-Kutta and constant step-size Adams-Bashforth solvers.

## References

- [AP98] ASCHER U. M., PETZOLD L. R.: *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. SIAM, Philadelphia, 1998.
- [Cha60] CHANDRASEKAR S.: *Radiate Transfer*. Dover Publications, NY, 1960.
- [CK90] CASH J. R., KARP A. H.: A variable order runge-kutta method for initial value problems with rapidly varying right-hand sides. *ACM Trans. Math. Softw.* 16, 3 (1990), 201–222.
- [CPCP\*05] CEREZO E., PEREZ-CAZORLA F., PUEYO X., SERON F., SILLION F.: A survey on participating media rendering techniques. *the Visual Computer* (2005).
- [E.69] E. F.: *Low-order classical Runge-Kutta formulas with step size control and their application to some heat transfer problems*. Tech. Rep. 315, NASA, 1969.
- [Hai00] HAIRER E. N. S. P. W. G.: *Solving Ordinary Differential Equations I, Nonstiff problems*. 2Ed. Springer-Verlag, 2000.
- [HEJ76] HULL T., ENRIGHT W., JACKSON K.: *Users guide for DVERK - a subroutine for solving non-stiff ODEs*. Tech. Rep. 100, University of Toronto, 1976.
- [IZT\*07] IHRKE I., ZIEGLER G., TEVS A., THEOBALT C., MAGNOR M., SEIDEL H.-P.: Eikonal rendering: Efficient light transport in refractive objects. In *SIGGRAPH* (NY, 2007), ACM.
- [JDZJ08] JAROSZ W., DONNER C., ZWICKER M., JENSEN H. W.: Radiance caching for participating media. *ACM Trans. Graph.* (2008).
- [JZJ08a] JAROSZ W., ZWICKER M., JENSEN H. W.: Irradiance Gradients in the Presence of Participating Media and Occlusions. *Computer Graphics Forum (Proceedings of EGSR 2008)* (2008).
- [JZJ08b] JAROSZ W., ZWICKER M., JENSEN H. W.: The Beam Radiance Estimate for Volumetric Photon Mapping. *Computer Graphics Forum (Proc. Eurographics EG'08)* (2008).
- [KH84] KAJIYA J. T., HERZEN B. P. V.: Ray tracing volume densities. *SIGGRAPH* (1984).

- [RT87] RUSHMEIER H. E., TORRANCE K. E.: The zonal method for calculating light intensities in the presence of a participating medium. *SIGGRAPH* (1987).
- [SZS\*08] SUN X., ZHOU K., STOLLNITZ E., SHI J., GUO B.: Interactive relighting of dynamic refractive objects. In *SIGGRAPH* (NY, 2008), ACM.