DISSERTATION

PERFORMANCE AND ACCURACY ENHANCEMENTS
OF RADIATIVE HEAT TRANSFER MODELING
VIA MONTE CARLO

Submitted by
Charles Nelson Zeeb
Department of Mechanical Engineering

In partial fulfillment of the requirements
for the Degree of Doctor of Philosophy
Colorado State University
Fort Collins, Colorado
Fall 2002

COLORADO STATE UNIVERSITY

October 10th, 2002

WE HEREBY RECOMMEND THAT THE DISSERTATION PREPARED UNDER OUR SUPERVISION BY CHARLES NELSON ZEEB ENTITLED, "PERFORMANCE AND ACCURACY ENHANCEMENTS OF RADIATIVE HEAT TRANSFER MODELING VIA MONTE CARLO," BE ACCEPTED AS FULFILLING IN PART REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY.

Committee on Graduate Work

Advisor

Department Head

ABSTRACT OF DISSERTATION

PERFORMANCE AND ACCURACY ENHANCEMENTS OF RADIATIVE HEAT TRANSFER MODELING VIA MONTE CARLO

Two ways to reduce the computational requirements of radiative heat transfer Monte Carlo simulation are explored. First, an efficient algorithm for tracing particles in large, arbitrarily complex, planar geometries containing nonparticipating media is presented. For arbitrary triangles and / or convex planar quadrilaterals, an efficient intersection algorithm is discussed in detail. After surveying several techniques used in ray tracing to limit the number of surfaces tested, the method of Uniform Spatial Division (USD) is implemented. The efficiency of the intersection algorithm and USD are demonstrated by timing results.

Second, improving the accuracy of the Monte Carlo results by applying reciprocity and closure is explored. Statistical theory is applied to the reciprocity estimation smoothing (RES) technique which combines reciprocity enforcement through estimation and closure enforcement through the technique of least-squares smoothing. By examining a large number of runs of two large geometries, several RES methods are compared to find the best method. The effects of the RES method on surfaces and individual results between surfaces are also explored. Estimates of the improvements caused by the RES method that can be calculated from the results of a single run are also derived.

Charles Nelson Zeeb
Mechanical Engineering Department
Colorado State University
Fort Collins, Colorado 80523
Fall 2002

ACKNOWLEDGMENTS

First, I am grateful to my family, particularly my father; my stepmother, Susan; and my sisters, Laurie and Kathy, for their help, support, and understanding over the last few years. Second, I would like to thank my advisor, Prof. Patrick Burns, for all the time and guidance he has given me on this project. From the time I started my Master's program, he and I have been on a long and interesting journey. I would also like to thank Prof. Charles Mitchell, Prof. Richard Loehrke, and Prof. David Zachmann for serving on my committee. Furthermore, I would like to acknowledge the support given to me by the Department of Energy's Computational Science Fellowship program. For his friendship, support, and help well above and beyond the call of duty, I am greatly indebted to John Dolaghan. He more than any one else kept prodding me along and made me see this through to completion.

This dissertation is dedicated to:

Margot Harper Zeeb

1935-2001

Only because of her support and belief in me have I been able to make it this far.

TABLE OF CONTENTS

Section **Page**

LIST OF TABLES

Table **Page**

LIST OF FIGURES

NOMENCLATURE

Note: Bold symbols and symbols marked by $\vec{\ }$ denote vector quantities.

English Letters

A	Area (m^2).
A, B, C	Cartesian coordinates of a plane's surface normal (m).
A^2	Anderson-Darling statistic defined in eqn. (C.4).
a, b	Curve fit parameters defined in eqn. (2.7).
a, b, c	Quadratic formula coefficients.
a_i	BLUE weight defined in eqn. (C.6).
BLUE	Best linear unbiased estimate.
BML	Binomial maximum likelihood estimate defined in eqn. (3.27).
C	Confidence interval.
CDF	Cumulative distribution function.
D	Distance from the plane to the origin of the system (m).
E	Energy (J).
E	Expected value (various).
$\boldsymbol{E}$	Unit direction vector (m).
E	Error (various).
EDF	Empirical distribution function defined in eqn. (C.1).
erf	Error function.
F	Exchange fraction defined in eqn. (1.3).
F	Cumulative distribution function.
F	Fraction of blackbody energy.
F_n	Empirical distribution function defined in eqn. (C.1).
FCI	Fractional confidence interval weighted estimate defined in eqn. (3.10).

FV	Fractional variance weighted estimate whose weight is defined in eqn. (3.14).
g_k	Row-sum constraints for the LSS Lagrangian defined in eqn. (3.34) (m^2).
g_k^*	Column-sum constraints for the LSS Lagrangian defined in eqn. (3.34) (m^2).
H	LSS objective function defined in eqn. (3.32).
h	Halfwidth of a confidence interval (various).
i, j, k, l	Specific members of a series.
L	Likelihood function defined in (3.20).
L	LSS Lagrangian defined in eqn. (3.33).
LSS	Least-squares smoothing method.
M	Number of values.
MV	Minimum variance estimate defined in eqns. (3.15) and (3.18).
m	Integer variable used in Table 2.1.
m	Modulus.
N	Number of photons.
$\boldsymbol{N}$	Node point in $\boldsymbol{R}^3$ (m).
$\boldsymbol{N_p}$	Unit surface normal for a plane (m).
N_{runs}	Number of Monte Carlo simulation runs.
N_{surf}	Number of surfaces.
$N_{V,k}$	Number of voxels along the kth Cartesian axis.
n	Index of refraction.
n	Number of random samples.
P	Percentage value.
$P_{k,l}$	Coordinate of the lth voxel plane along the kth Cartesian axis.
p	Probability.
Q	Energy rate (W).
$\boldsymbol{R}$	Position of the photon (m).
$\boldsymbol{R}^{\boldsymbol{M}}$	M-dimensional space.
R	Random number uniformly distributed between zero and one.
RES	Reciprocity estimation smoothing.

RP	Reciprocity pair.
$\boldsymbol{r}$	Position on a surface (m).
S	Surface.
$\overline{S_k S_l}$	Alternate symbol for the total exchange area, η (m^2).
$\mathbf{s}$	Vector of values.
T	Temperature (K).
t	Distance the photon has travelled (m).
t	Time (s).
t_p	Time per photon (s).
$\boldsymbol{U}$	LSS matrix defined in eqns. (3.39) and (3.40) (m^2).
U^2	Watson statistic defined in eqn. (C.3).
USD	Uniform spatial division method.
u, v	Addition lagged-Fibonacci pseudo-random generator lags.
V	Voxel.
W	Shapiro-Wilk statistic defined in eqn. (C.5).
W^2	Cramér-von Mises statistic defined in eqn. (C.2).
w	LSS weight defined in eqn. (3.32).
X, Y, Z	Cartesian coordinates (m).
X	Pseudo-random integer.
x	Random variable.

Greek Letters

$\vec{\beta}$	LSS solution vector defined in eqn. (3.41) (m^2).
γ	Percent closure error for a surface defined in eqn. (3.54).
Δ	Normalized matrix error defined in eqn. (3.52) (various).
Δ^k	Normalized surface error defined in eqn. (3.53) (various).
Δ_N	Quantity defined in Section C.2 and listed in Table C.1.
δ	Halfwidth of the fractional confidence interval defined in eqn. (3.4).
ε	Emissivity.
η	Total exchange area defined in eqn. (1.7) (m^2).

$\underset{\sim}{\eta}$	η matrix maximum likelihood estimate defined in Section 3.3 (m^2).
θ	Statistical estimator (various).
θ	Cone angle (rad).
κ	Estimation weights defined in eqns. (3.11) and (3.14).
λ	Wavelength of the radiation (μm).
$\vec{\lambda}$	LSS Lagrange multiplier vector defined in eqn. (3.33) ($1/m^2$).
μ	Expected value of a random variable (various).
ν_0	Quantity defined in eqn. (2.3) (m).
ν_E	Quantity defined in eqn. (2.3) (m).
ρ_λ''	Spectral, bidirectional reflection function.
σ	Standard deviation (various).
σ	Stephen-Boltzman constant ($W/[m^2 \cdot K^4]$).
Ψ	$\underset{\sim}{\eta}$ matrix probability constant defined in eqn. (3.45) (various).
ψ	Azimuthal angle (rad).
Ω	Area-emissivity product, εA (m^2).
Ω^*	Dimensionless Ω, Ω divided by the smallest Ω in the geometry.
ω	Confidence interval constant defined in eqn. (3.1).

Subscripts

$+, -, \pm$	Plus and / or minus roots of a quadratic equation.
0	Origin of the photon.
$1, 2, 3, 4$	Numbered series.
A, B, C	Lettered series.
abs	Absorbed.
bin	Binomial distribution.
B	Bias.
$BLUE$	Best linear unbiased estimate.
BML	Binomial maximum likelihood estimate.
em	Emitted.
F	Exchange fraction.

FCI	Fractional confidence interval weighted estimate.
FV	Fractional variance weighted estimate.
H	Half-plane test “bounding” plane.
i, j, k, l	Specific members of a series.
i	Intersection.
i	Incoming.
$k \to l$	From surface k to surface l.
ML	Maximum likelihood estimate.
MV	Minimum variance estimate.
nor	Normal distribution.
o	Optimal.
r	Reflected.
s	Shortest distance.
V	Voxel.
VB	Voxel boundary.
$\mathrm{X, Y, Z}$	Quantity in the given Cartesian direction.
η	Total exchange area.
ηM	Total exchange area matrix.
θ	Statistical estimator (various).
σ	Standard deviation (various).

Superscripts

$\vec{\ }$	Vector quantity.
$\tilde{\ }$	“Exact” or “true” value of a statistical quantity.
$\bar{\ }$	The mean value of a quantity.
$\hat{\ }$	Estimated value that applies reciprocity such as those estimators in Section 3.1.
$\ddot{\ }$	Least-squares smoothed value of a quantity.
∞	“Ground truth” result defined in Section 3.4.4.
i	Wavelength band.
α	Curve fit parameter defined in eqn. (2.7).

CHAPTER 1 INTRODUCTION

1.1 Background

The modeling of radiative heat transfer is a particularly challenging subject. Since radiation involves “action at a distance,” every surface and volume of media in a geometry can affect every other surface and volume. Geometries can be very complex. Furthermore, surface properties are often a function of angle, and media properties often are a function of wavelength. Few methods are versatile enough to handle problems this complex. One proven method is the Monte Carlo method [Howell, 1968; Haji-Sheikh, 1988; Siegel and Howell, 1992; Modest, 1993; Howell, 1998; Burns and Pryor, 1999].

In radiative heat transfer Monte Carlo, results are obtained by tracing a statistically significant number of “bundles” or “photons” of energy. Statistical relationships are used to model the emission, absorption, reflection, transmission, and scattering of these bundles. Although the method is very versatile, it can also be computationally very intensive, since most simulations involve tracing millions or even billions of photons.

While Monte Carlo methods are becoming more feasible with today’s more powerful computers, it is still very important that Monte Carlo programs be as efficient as possible. While previous work on a parallel Monte Carlo code [Zeeb et al., 1999] shows that it is easy to create a program with a very high level of parallel efficiency, implementing the program in parallel only reduces the “wall clock” execution time, not the computational resources required. Although computational power is readily available, the trade off in enhanced physical fidelity versus compositional resources should be evaluated when considering implementation of the Monte Carlo method.

The purpose of this dissertation is to explore two ways to reduce the computational requirements of radiative heat transfer Monte Carlo simulation. First, an efficient Monte Carlo tracing algorithm for large, arbitrary, planar geometries in nonparticipating media is presented. Second, while the convergence of Monte Carlo methods is slow, inversely proportional to the square root of the number of emissions, there is redundant information that can be used to improve accuracy. Therefore, the application of reciprocity estimation and Larson and Howell's [1986] method of least-squares smoothing to use reciprocity and closure relations to improve the accuracy of Monte Carlo results is explored.

1.2 The Radiative Heat Transfer Monte Carlo Solution Method

The results of a Monte Carlo simulation from programs such as LSMONTE [Zeeb and Burns, 2000] and MONT3D [Zeeb et al., 1999] are used to determine the unknown temperatures and energy emission rates for a geometry with nonparticipating media for which the properties of each surface are known. To aid in understanding the issues involved, the general solution method for determining the unknowns is presented.

The geometry is divided into N_{surf} surfaces of uniform temperature and emissivity which emit and absorb photons. The total rate of energy, Q_{em}, emitted from a surface of area, A, at temperature, T, is [Siegel and Howell, 1992; Modest, 1993]:

$$Q_{em} = n^2 \varepsilon A \sigma T^4 = n^2 \Omega \sigma T^4 \tag{1.1}$$

where ε is the total hemispherical emissivity for the surface, which is assumed to be independent of temperature; σ is the Stephen-Boltzman constant; Ω is the area-emissivity product for the surface; and n is the index of refraction of the medium into which the radiation is emitted, which is one for nonparticipating media. For this reason, from this point on, n is assumed to be unity and omitted.

The estimated amount of power emitted by surface k that is absorbed by surface l, $Q_{abs,kl}$, is:

$$Q_{abs,\,kl} = F_{kl} Q_{em,\,k} \tag{1.2}$$

F_{kl} is the Monte Carlo estimate of the exchange fraction from surface k to surface l:

$$F_{kl} = \frac{E_{kl}}{E_k} \tag{1.3}$$

where E_k is the total energy of “bundles” emitted by surface k, and E_{kl} is the total energy of “bundles” emitted by surface k which are absorbed by surface l, no matter what the path. E_{kl} results directly from the Monte Carlo “photon” (or “bundle”) tracing. It should be noted that if all properties are independent of temperature, then all F_{kl} are also independent of temperature. Enough “bundles” must be emitted in the Monte Carlo simulation to reduce the statistical scatter in F_{kl} to an acceptable level. More on this will be presented later.

The net outgoing radiative energy rate for surface k, Q_k, is:

$$Q_k = Q_{em,\,k} - \sum_{l=1}^{N_{surf}} F_{lk} Q_{em,\,l} = \Omega_k \sigma T_k^4 - \sum_{l=1}^{N_{surf}} F_{lk} \Omega_l \sigma T_l^4 \tag{1.4}$$

If heat transfer is only by radiation and the surface is at equilibrium, Q_k is zero. If the surface is not at equilibrium and / or there are other significant sources of heat such as conduction, convection, or chemical reaction, Q_k is generally not zero. Particularly noteworthy is that the summation in eqn. (1.4) includes the surface *k* itself. This is why eqn. (1.4) does not explicitly include self-absorption, as it is accounted for in the F_{kk} term.

While ε is assumed to be independent of temperature, it not necessarily independent of wavelength. Since the emissivity of surfaces is usually constant over large wavelength bands, it is possible to perform separate Monte Carlo simulations for each wavelength band. For wavelength band *i*, the total energy emission rate for the band, Q_k^i, is:

$$Q_k^i = \varepsilon_k^i A_k F_k^i \sigma T_k^4 - \sum_{l=1}^{N_{surf}} F_{lk}^i \varepsilon_l^i A_l F_l^i \sigma T_l^4 = \Omega_k^i F_k^i \sigma T_k^4 - \sum_{l=1}^{N_{surf}} F_{lk}^i \Omega_l^i F_l^i \sigma T_l^4 \qquad (1.5)$$

where the total hemispherical emissivities are now for the band and the *F*'s are the fraction of blackbody energy for the given surface in the band. More detail about using wavelength bands can be found in Maltby and Burns, 1991; Burns et al., 1992; and Burns and Pryor, 1999. In the work in this dissertation, since only one wavelength band is used, the *F* factor is always one and the *i* superscript is dropped henceforth.

If the proper boundary conditions are supplied, that is either the temperature or energy rate for each surface, eqn. (1.4) (or eqn. (1.5) for each wavelength band) forms a system of N_{surf} equations and N_{surf} unknowns. Solving for the unknowns is a two step process. First, since eqn. (1.4) is linear in T^4, the surfaces with unknown temperatures form a system of linear equations that can be solved. Once all the temperatures are known, eqn. (1.4) can be used to solve for the unknown *Q*'s.

1.3 Improving the Efficiency of the Monte Carlo Photon Tracing Algorithm

As stated above, one of the goals of this dissertation is the efficient modeling using the Monte Carlo method of complex geometries with nonparticipating media. Since the first work in radiative heat transfer Monte Carlo by Howell and Perlmutter [Howell and Perlmutter, 1964a; Howell and Perlmutter, 1964b; Perlmutter and Howell, 1964], there have been many papers on the subject. Good reviews are given by Howell [1968; 1998], Haji-Sheikh [1988] and Burns and Pryor [1999]. Unfortunately, most published radiative Monte Carlo work is for simple geometries such as slabs and cubes; there are few published routines to handle arbitrary, complex geometries. Corlett [1966] uses a geometry composed of several types of objects and Modest [1978] creates geometries that include curved surfaces, but neither explains their photon tracing algorithm in detail. Chin et al. [1989; 1992] have covered several Monte Carlo issues, particularly applying Monte Carlo to finite element meshes. Farmer [1995] has also discussed using Monte Carlo to simulate arbitrary geometries constructed using finite element meshes. In addition, Henson et al. [1996] discuss some techniques for improving the speed of Monte Carlo routines. Several generalized radiative heat transfer Monte Carlo programs do exist, for example, TSS [Panczak, 1989] and MATRAD [Koeck, 1988]. Still, except for the above references, nothing else has been published in detail about these and other such codes' photon tracing algorithms.

One generalized photon tracing algorithm discussed in detail has been implemented in the codes MONT3D [Maltby, 1987; Burns et al., 1990; Maltby and Burns, 1991; Burns and Pryor, 1999; Zeeb et al., 1999] and LSMONTE [Zeeb and Burns, 2000], which simulate radiative transfer in geometries with nonparticipating media. The codes have been used extensively for more than a decade by Lawrence Livermore National Laboratory (LLNL) and other sites. The output of the codes is a radiative exchange fraction matrix which is used as input to thermal analysis codes, particularly TOPAZ3D [Shapiro, 1985]. Geometries with more than 14,000 surfaces have been modeled. The codes have been independently validated theoretically and verified experimentally.

This dissertation presents improvements made in the photon tracing algorithm. The present work differs from most previous works in that techniques from the computer graphics field of ray tracing are incorporated. As pointed out by Rushmeier [1993], while much of the early work in computer graphics simply borrowed from heat transfer, computer graphics has matured sufficiently that some of the techniques can be used to improve heat transfer calculations - especially true for ray tracing. While Henson et al. [1996] and particularly Panczak [1989] have also addressed this topic, this dissertation covers many aspects not touched in those papers. Since Monte Carlo calculations are computationally intensive, the emphasis here is to present an algorithm that allows complex geometries but is computationally efficient.

1.4 Collision Based Versus Pathlength Based Radiative Monte Carlo

One point that is often overlooked is that radiative Monte Carlo is, in the words of Howell [1968], a “semimacroscopic” approach where small packets of the total energy emitted by a surface are traced. While these energy packets are often referred to as photons, they are actually “bundles” of energy or photons. This is obvious when one thinks about it because processes such as emission and reflection are modeled as macroscopic processes instead of quantum processes. While these “bundles” may contain differing amounts of energy [Haji-Sheikh, 1988], calculations are usually simplified by making all “bundles” of equal energy.

Using the terminology of Farmer [1995], there are two ways to tally these “bundles” in what he calls forward approaches to radiative heat transfer Monte Carlo. In collision based radiative Monte Carlo, each “bundle” is absorbed or reflected probabilistically as a whole. In pathlength based Monte Carlo, each “bundle,” while still reflected probabilistically, is partially absorbed deterministically by each surface it hits. In other words, the "bundle" becomes successively smaller (lower in energy) as it strikes each absorbing surface, by virtue of a continuous amount of energy absorbed in proportion to the absorptivity of each surface. Tracing terminates when the "bundle" strikes a black surface or leaves the geometry. Since neither condition is found in many enclosures, some sort of termination criteria, such as to stop tracing when a certain energy threshold is reached, or Russian roulette [Lewis and Miller, 1984; Lux and Koblinger, 1991] must be implemented.

As shown by Sobol' [1974], the advantage of pathlength based Monte Carlo is that for the same number of emissions, pathlength based Monte Carlo results have lower variances and uncertainties than collision based results. On the other hand, collision based emissions take less calculation and CPU time than pathlength based emissions. Which method produces the least uncertainty in the same amount of CPU time is unclear and depends heavily on the material properties and the geometry of the problem.

The reciprocity estimation smoothing method described below which is used to improve the accuracy of the Monte Carlo results depends heavily on the statistics involved which in turn depend on the photon absorption method used. Past work with MONT3D [Zeeb et al., 1999] and LSMONTE [Zeeb and Burns, 2000] has provided experience with collision based Monte Carlo. Furthermore, the statistics of collision based Monte Carlo are much simpler than the statistics of pathlength based Monte Carlo. For these reasons, collision based Monte Carlo is used in this work.

In this work, since results are temperature independent, all "bundles" emitted from a given surface are of equal although unspecified size. Since these "bundles" are absorbed probabilistically, it is common to refer to them as "photons" which is how they will be referred to in the rest of this work. For the collision based radiative Monte Carlo method used in this work, the exchange fraction calculation simplifies to:

$$F_{kl} = \frac{N_{kl}}{N_k} \tag{1.6}$$

where N_k is the total number of photons emitted by surface k and N_{kl} is the total number of photons emitted by surface k that are absorbed by surface l, no matter what the path. The statistics of this collision based Monte Carlo method are the basic Bernoulli trial, binomial, and multinomial distributions. Details on these distributions can be found in most statistics books, including Martin [1971], Mood et al. [1974], Ross [1988], and Evans et al. [2000].

1.5 Total Exchange Areas, Closure, and Reciprocity

A useful quantity in radiative heat transfer Monte Carlo is the total exchange area, η_{kl}, which is defined in this dissertation as:

$$\eta_{kl} = \varepsilon_k A_k F_{kl} = \Omega_k F_{kl} \tag{1.7}$$

Using total exchange areas, the power transferred from surface k to surface l, Q_{kl} is:

$$Q_{kl} = Q_{k \to l} - Q_{l \to k} = \eta_{kl}\sigma T_k^4 - \eta_{lk}\sigma T_l^4 \tag{1.8}$$

Since all the exchange fractions for a surface sum to unity, the conservation of energy (“photons” or “bundles”), the first law of thermodynamics, leads to the constraint of closure for η’s:

$$\sum_{l=1}^{N_{surf}} \eta_{kl} = \Omega_k \sum_{l=1}^{N_{surf}} F_{kl} = \Omega_k(1) = \Omega_k \tag{1.9}$$

Another constraint on the η's is shown in eqn. (1.10) below. This constraint is sometimes called total exchange area reciprocity but will usually be referred to simply as reciprocity from this point forward:

$$\eta_{kl} = \eta_{lk} \tag{1.10}$$

This property is directly related to the reciprocity of the spectral, bidirectional reflection function (also known as the bidirectional spectral reflectivity) [Siegel and Howell, 1992; Modest, 1993]:

$$\rho_\lambda''(\boldsymbol{r}, \lambda, \theta_i, \psi_i, \theta_r, \psi_r) = \rho_\lambda''(\boldsymbol{r}, \lambda, \theta_r, \psi_r, \theta_i, \psi_i) \tag{1.11}$$

where the spectral, bidirectional reflection function, ρ_λ'', is a measure of the reflectivity of a surface at a point, **r**, as a function of not only the wavelength, λ, but also the cone angles, θ, and azimuthal angles, ψ, of both the incoming (subscript i) and reflected (subscript r) radiation. While both Siegel and Howell [1992] and Modest [1993] state that the reciprocity of ρ_λ'' applies to all radiative materials, some argue that it does not apply in some very special cases [Snyder et al., 1998]. Still it is found to apply to almost all if not all radiative materials that have been tested experimentally. Eqn. (1.11) is essentially a statement of the Helmholtz optical reciprocity theorem [Hottel and Sarofim, 1967; Snyder et al., 1998] which states that for geometries for which eqn. (1.11) applies, the intensity registered by a detector from a source elsewhere in a geometry is the same as would be measured if the source and detector positions were switched.

Due to the complexity of eqn. (1.11), most total exchange area radiative work is based on diffuse emission and two simple reflectivity models for which the reciprocity of ρ_λ'' holds, diffuse reflection and specular reflection. Furthermore (total exchange area) reciprocity has only been proven to apply in two cases: black surfaces and specularly reflecting surfaces [Siegel and Howell, 1992; Modest, 1993]. These two proven cases of reciprocity allow the solution of problems where the radiosity is constant across each surface and the material properties are a mixture of diffuse and specular reflection which are both not a function of angle. The versatility of the Monte Carlo method has allowed the modeling of more complex cases in which both reciprocity and ρ_λ'' reciprocity holds. In particular, it can model systems where the radiosity varies across surfaces and the material properties are a mixture of constant diffuse reflection and specular reflection that varies as a function of angle [Zeeb et al., 1999; Branner, 2000].

It should be noted that, while Monte Carlo results automatically observe closure (if no photons are "lost" during tracing; a condition that can be enforced algorithmically), due to statistical scatter, reciprocity is observed only to within statistical convergence tolerance.

The term total exchange area is usually connected with the zone (also called zonal) method [Hottel and Sarofim, 1967; Larson and Howell, 1986; Siegel and Howell, 1992; Modest, 1993]. A different symbol, $\overline{S_k S_l}$, is usually used to represent the total exchange area between surfaces (usually in participating media). While the name and notation are used in Hottel and Sarofim [1967] where the zone method is covered extensively, their definition of total exchange areas in that book is somewhat broader. Although Hottel and Sarofim do not cover the Monte Carlo method, their definition of the total exchange area is very similar to eqn. (1.7). The term "total exchange area" is actually well suited to the Monte Carlo method. In fact, Maltby [1994] uses Hottel and Sarofim's definition and symbols for total exchange area in his description of his zonal participating media Monte Carlo code, TRIM3D.

While the total exchange area in the zone method is defined using an equation very similar to eqn. (1.8), it is not exactly the same as the Monte Carlo definition. In particular, while equations essentially the same as eqns. (1.8) to (1.10) apply to the zone method, and both the current Monte Carlo method and the zone method require uniform temperature and emissivity across a surface, the zone method also requires uniform radiosity across surfaces. To stress that the concepts are similar but not the same, this dissertation uses the same name as Hottel and Sarofim, the total exchange area, but a different symbol, η.

1.6 Improving Monte Carlo Results by Estimation

Due to the reciprocity relationship mentioned above, it is obvious that almost half the information in the η matrix is redundant. While most methods for obtaining η use this redundancy to decrease the number of η that are calculated, this is not possible with Monte Carlo simulations. Since emissions from any surface automatically calculate all η for that surface, the most work that can be avoided in a Monte Carlo simulation is photon emissions for one surface.

What at first appears to make this generation of additional information even worse is that, due to statistical scatter, the reciprocity relation does not actually apply to Monte Carlo results except in the limit of an infinite number of emissions. This supposed deficit may actually offer an advantage because the reciprocity relationship may still be applied to Monte Carlo results by estimation. From statistics, it is known that proper application of estimation decreases the error for most of the η and the uncertainty for almost all if not all of the η. Thus, one of the goals of this dissertation is to apply statistical theory to obtain the best possible estimator for improving the accuracy of the reciprocity pairs.

1.7 Re-enforcing Closure on Estimated Results

Once Monte Carlo results have had reciprocity applied by estimation, the first law of thermodynamics, closure, no longer applies. This is a common problem that often occurs when other methods, such as view factor methods, are used to calculate η and the reciprocity relation is used to reduce the number of η to be calculated. Therefore, there have been several schemes designed to rectify the results generated by Monte Carlo, zone, and other such methods so that closure applies. Most of these schemes either lump the corrections into N_{surf} of the η, such as Sowell and O'Brien's [1972] scheme and Taylor and Luck's [1995] naive rectification scheme, or spread the correction over all the η's by some simple formula which avoids generating negative η's, such as Van Leersum, 1989 and Tsuyuki, 1992. While rectifying the η's, these schemes may produce large errors for some of the η's. However, Larson and Howell's [1986] least-squares smoothing (LSS) method, used in this work, is different in that it uses Lagrange multipliers to minimize the changes in the η's caused by re-enforcing closure on the η's. It should be noted that a similar method has been proposed by Vercammen and Froment [1980], but their method is less versatile in that it requires that zero and repeating η's be identified beforehand. Taylor and Luck [1995] actually extend the LSS technique by using nonlinear programming to ensure that all η's are non-negative. While it is possible to apply nonlinear programming to the solution method devised in this dissertation, it has not been done in this work for several reasons. First, for properly defined geometries, there do not seem to be many, if any, negative η's for most smoothing techniques. Second, for the large geometries for which Monte Carlo is particularly well suited as a method, nonlinear programming is almost certainly impractical. Third, Taylor and Luck found that there was little or no difference between using nonlinear programming and recalculating the problem after setting all η that became negative to zero.

The combination of reciprocity estimation with the LSS method is referred to as reciprocity estimation smoothing (RES) in this work. The present work extends that implemented in SMOOTH, a program to apply RES to the results of MONT3D [Burns et al., 1992; Dolaghan et al., 1992; Loehrke et al., 1995]. The RES technique has also been used by

Chen et al. [2000] to improve the view factors used to simulate direct-injection diesel engines and by Liu et al. [2001] to improve η's used to simulate oil-fired furnaces.

The purpose of the RES work in this dissertation is to apply detailed statistical theory to improve the accuracy of the estimation and LSS and also to determine metrics that quantify the decrease in the uncertainty of the RES results. This work is the first study to apply RES to large Monte Carlo geometries. Geometries with 144 and 1,182 emitting surfaces are used. In addition, results from many numerical experiments ("runs") are obtained to provide an in-depth view of the distribution of the effects of RES.

1.8 The Importance of Enforcing Closure and Reciprocity

A review of the literature suggests that there appear to be several potential benefits to applying reciprocity and closure. First, the acts of enforcing reciprocity and closure themselves appear to increase the accuracy of the final Q's and T's. This is not well studied, but Taylor and his coworkers [1993; 1995] have found a simple heat transfer problem where the results are very sensitive to small errors in the view-factor matrix. This sensitivity is greatly reduced by enforcing reciprocity and closure in the problem. Clarksean and Solbrig [1994] have also found that satisfying closure and particularly satisfying reciprocity is more important to the accuracy of the final results than the accuracy of the view factors themselves. Murty and Murty [1991], on the other hand, have found that enforcing closure by Larson and Howell's least-squares smoothing method has little effect on their zone-radiation calculations. This work will attempt to explain why these seemingly disparate conclusions exist.

Enforcement may be required for stable calculations. As pointed out by Brewster [1992], if reciprocity and closure are not enforced, it is possible to obtain a singular or poorly conditioned η matrix that makes the solution of the linear equations eqn. (1.4) for the unknown T^4's, impossible. One example of when this can occur is that one η of a reciprocity pair may be zero while the other is not.

Furthermore, it should be realized that thermal balance codes that use the results of the Monte Carlo simulation are formulated based on the principle of energy conservation. Some such codes can exhibit convergence difficulties if both reciprocity and closure are not obeyed to within stringent tolerances. In addition, post-processing often involves examining energy flows and energy balances, and the analyst could be led to misinterpret the accuracy and reliability of the thermal balance code unless both conditions are observed to within very small tolerances for all modes of energy transport. Furthermore, if closure is observed and reciprocity is not, then although the energies will balance, the distribution of energy will be in error. Also, thermal analysis codes such as TOPAZ3D [Shapiro, 1985] may expect reciprocity to apply and, for example, arbitrarily use only the upper triangle or lower triangle of the η matrix.

1.9 Overview of Dissertation

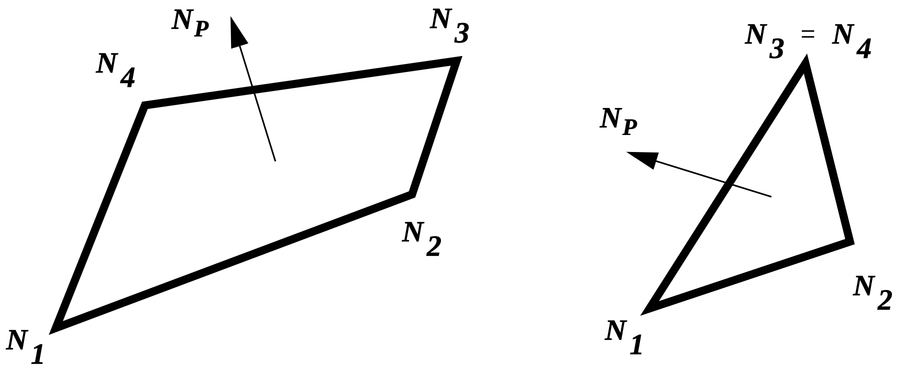
In chapter two, an efficient Monte Carlo particle tracing algorithm for large, arbitrary, planar geometries is developed. The third chapter discusses increasing the accuracy of Monte Carlo results using the RES method. The final chapter gives conclusions and recommendations for the work done in the previous two chapters. This work also includes appendices describing the CCS (Center for Computing Science) addition lagged-Fibonacci pseudo-random number generator and a random emission routine used in this dissertation. Furthermore, there is an appendix presenting the results of some tests for normality on total exchange areas before and after the RES method has been applied.

CHAPTER 2 DETERMINING AN EFFICIENT MONTE CARLO PARTICLE TRACING ALGORITHM

Drawing on techniques used in the computer graphics field of ray tracing, an efficient algorithm for tracing particles (photons) in large, arbitrarily complex, planar geometries containing nonparticipating media is presented. An efficient intersection algorithm for arbitrary triangles and / or convex planar quadrilaterals is discussed in detail. Several techniques used in ray tracing to limit the number of surfaces tested are discussed and the method of Uniform Spatial Division (USD) is implemented. The “mailbox” technique, saving some calculated results for possible use in the future, is also discussed. To determine the efficiency of the intersection algorithm and the effectiveness of USD, timing results are presented for a number of different spatial divisions for four geometries.

2.1 Surface Geometries

For the algorithms described below, it is assumed the geometry is defined in a global Cartesian coordinate system from node points, **N**, that are input by the user or generated using a grid generation program such as TrueGrid [XYZ, 1997]. Surfaces are defined by specifying four node points and an unique surface number. The types of surfaces modeled are either triangles or convex quadrilaterals as shown in Fig. 2.1. Surfaces must be planar. If the four nodes of a quadrilateral are not coplanar to within a small tolerance, the quadrilateral is divided into two planar triangles. The orientation of the surfaces follows the right-hand rule, so the surface normal, which determines the direction of emis-



(a) Quadrilateral **(b) Triangle**

Figure 2.1 Radiating Surface Geometries

sion, always points outward as the surface is traversed in the direction of increasing node number. Surfaces are one-sided; the “back” side, which is opposite the surface normal, does not emit or interact with photons.

2.2 Calculating the Intersection Distance to a Plane

In most ray tracing simulations, most of the work is spent determining which surface a photon hits. This calculation is the starting point of our discussion due to its importance and also because it is the basis for all intersection routines. For this algorithm, there are two parts to the calculation: 1) finding the distance to the plane that contains the polygon, and 2) determining if the intersection point with the plane is inside the polygon. The first part is covered in this section; the second part is covered in the next section. The previous version of the tracing algorithm [Maltby, 1987; Burns et al., 1990; Maltby and Burns, 1991; Burns and Pryor, 1999] used an intersection routine with many similarities to this one. The major difference is that while the current algorithm focuses on calculating the distance to intersection, t_i, the previous algorithm focuses on the point of intersection, $\boldsymbol{R_i}$,which requires more calculation.

Intersections with planar surfaces are common in ray tracing, so the formula for the intersection with a plane is covered in many ray tracing tutorials. A particularly good discussion is given by Haines [1989]. Defining the origin of the photon as $\boldsymbol{R_0} = (X_0, Y_0, Z_0)$ and its unit direction vector as $\boldsymbol{E} = \{E_X, E_Y, E_Z\}$, the equation for the position of the photon, $\boldsymbol{R}$, is given by:

$$\boldsymbol{R} = \boldsymbol{R_0} + \boldsymbol{E}t \tag{2.1}$$

where t is the distance the photon has travelled. The equation for the plane which contains the polygon is given by:

$$AX + BY + CZ - D = 0 \qquad \text{where } A^2 + B^2 + C^2 = 1 \tag{2.2}$$

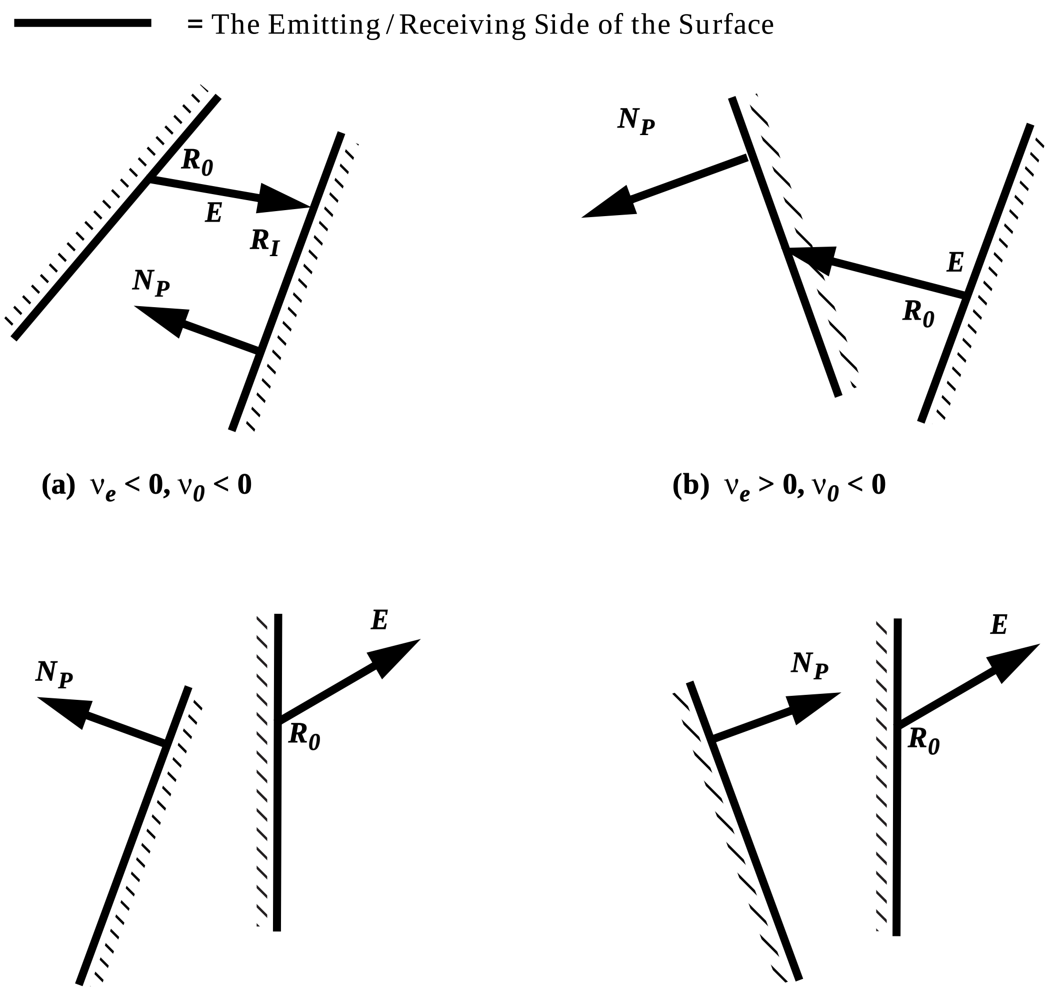
The unit surface normal for the plane (and the polygon in the plane), $\boldsymbol{N_p}$, is equal to $\{A,B,C\}$ and D is the distance from the origin of the system, (0,0,0), to the plane. Inserting eqn. (2.1) into eqn. (2.2) an equation for t_i, the distance to the intersection with the plane, is obtained.

$$t_i = \frac{-(AX_0 + BY_0 + CZ_0 - D)}{AE_X + BE_Y + CE_Z} = \frac{D - (\boldsymbol{N_p} \bullet \boldsymbol{R_0})}{\boldsymbol{N_p} \bullet \boldsymbol{E}} = \frac{\nu_0}{\nu_E} \tag{2.3}$$

$\boldsymbol{N_p}$ and D are stored for each surface during preprocessing, and $\boldsymbol{R_0}$ and $\boldsymbol{E}$ are calculated every time a photon is emitted or reflected.

As Fig. 2.2 (a) shows, two conditions must be met for a valid intersection. First, ν_E must be less than zero, which means the plane's surface normal is pointing toward the direction vector. Second, ν_0 must be less than zero, which means $\boldsymbol{R}_0$ is in “front” of the plane, i.e. on the same side of the plane as the surface normal [Jeger and Eckmann, 1967].

The calculation above can be done most efficiently in steps.



(a) $\nu_e < 0, \nu_0 < 0$ **(b)** $\nu_e > 0, \nu_0 < 0$

(c) $\nu_e < 0, \nu_0 > 0$ **(d)** $\nu_e > 0, \nu_0 > 0$

Figure 2.2 Intersection Types

1. Calculate ν_E.
2. If ν_E is greater than or equal to zero, then the cases in either Fig. 2.2 (b) or Fig. 2.2 (d) apply, the intersection point is rejected and no further calculation for this surface is needed.
3. Calculate ν_0.

4. If v_0 is greater than or equal to zero, the case in Fig. 2.2 (c) applies. Therefore, the intersection is rejected and no further calculation for this surface is needed.
5. Calculate t_i using eqn. (2.3).

Step 2 is referred to as “backface culling.” As mentioned in Section 2.1 and shown in Fig. 2.2, only the “front” of a surface emits or intersects photons. In a properly defined geometry, no photons intersect the back side, so all interactions with that side are ignored. Steps 1 and 2 only require three multiplies, two additions, and one compare, and, on average, cause about half the surfaces tested to be rejected.

While it is true that steps 3 and 4 should also reject about half the surfaces in the geometry, there are two reasons to do backface culling first. First, backface culling is slightly more efficient. Second, when the USD algorithm described later is used, very few surfaces, particularly after the first voxel is traversed, are “behind” the plane.

2.3 Point-in-Polygon Test

Once a valid distance, t_i, is found from step 5, a check must be made to ensure that the point is within the polygon. A rather complete study of point-in-polygon strategies has been done by Haines [1994]. Since the polygons described in Section 2.1 are convex and have only three or four sides, the exterior-edges algorithm has been chosen for the point-in-polygon test. Within the uncertainty in Haines’ test results, this test is as fast as any other, and was chosen over the others he discussed because it requires less storage and is easier to implement.

In the exterior-edges test, the half-plane test is performed on all edges of the polygon. The half-plane test requires a “bounding” plane for each edge that is perpendicular to the polygon’s surface and includes that edge. From eqn. (2.2), the “bounding” plane is defined by the normal $\boldsymbol{N_{p,H}} = \{A_H, B_H, C_H\}$ which points into the surface, and the distance, D_H. For the intersection point, $\boldsymbol{R_i}$, if [Jeger and Eckmann, 1967]:

$$A_H X_i + B_H Y_i + C_H Z_i > D_H \tag{2.4}$$

then the point is outside that edge of the polygon and is rejected. The test is performed for each edge of the polygon. As soon as the point fails any test, it is rejected. The constants A_H, B_H, C_H, and D_H are calculated for each surface in the preprocessing (input) stage.

Haines [1989, 1994] suggests projecting the polygon and the test point into two dimensions. This saves a floating point add and multiply for each half-plane test. The simplest way to project the problem into two dimensions is to discard one of the X, Y, or Z coordinates. The area of the polygon is not preserved, but the topology is. The best coordinate to throw away is the one whose magnitude in the polygon's surface normal, $\boldsymbol{N_p}$, is the greatest. $\boldsymbol{N_{p,H}}$ and D_H must be calculated in this two-dimensional plane using the new two-dimensional coordinates. While the current version of the intersection algorithm implements this two-dimensional test, the previous version uses the three-dimensional form.

2.4 Limiting the Search

Every time a photon is emitted or reflected, the next surface it strikes must be found. Since the time to trace each photon increases linearly with the number of surfaces searched for intersections, it is obvious that great improvements in efficiency can be gained by reducing the number of surfaces that need to be tested. Reducing the number of surface interaction calculations has been a topic of extensive study in the field of ray tracing. Good overviews of the general techniques applied to this problem are given by

Arvo and Kirk [1989] and Watt and Watt [1992]. Of all the techniques reviewed, two general techniques are the most promising: bounding volumes and spatial subdivision. A discussion with a different perspective on applying ray tracing techniques to radiative Monte Carlo can be found in Panczak [1989].

The bounding volume technique reduces the number of intersection calculations by surrounding all the objects (surfaces) in the scene (geometry) with bounding volumes. The bounding volumes are chosen to be simple objects such as spheres and cubes so that intersection calculations with them are swift. The object or objects inside the bounding volume need only be checked if the bounding volume is intersected. According to Arvo and Kirk [1989], when a hierarchy of bounding volumes is used, the complexity of the intersection calculation is proportional to the logarithm of the number of objects. If the bounding volumes are not used in a hierarchy, the time for the intersection calculations is reduced but is still linear with the number of objects. Since the polygon intersection calculations described above are so simple, bounding volumes should contain collections of polygons. Devising a good way to define bounding volumes for groups of arbitrary surfaces would be difficult and probably not add much to the efficiency of the program.

The other promising technique is three-dimensional spatial subdivision, depicted in two dimensions in Fig. 2.3. For this technique, instead of placing bounding volumes around objects, the volume bounding the geometry is partitioned. Space is usually divided into axis-aligned rectangular prisms which are referred to as voxels (a three-dimensional version of a pixel). By aligning the voxel planes with the axes, computations are simplified. The photon is then traced from voxel to voxel. A check is made only inside each voxel to determine if any of the surfaces inside it are intersected. The search stops once the closest intersection within the current voxel is found. This reduces the number of intersection calculations in two ways. First, only surfaces in voxels along the photon

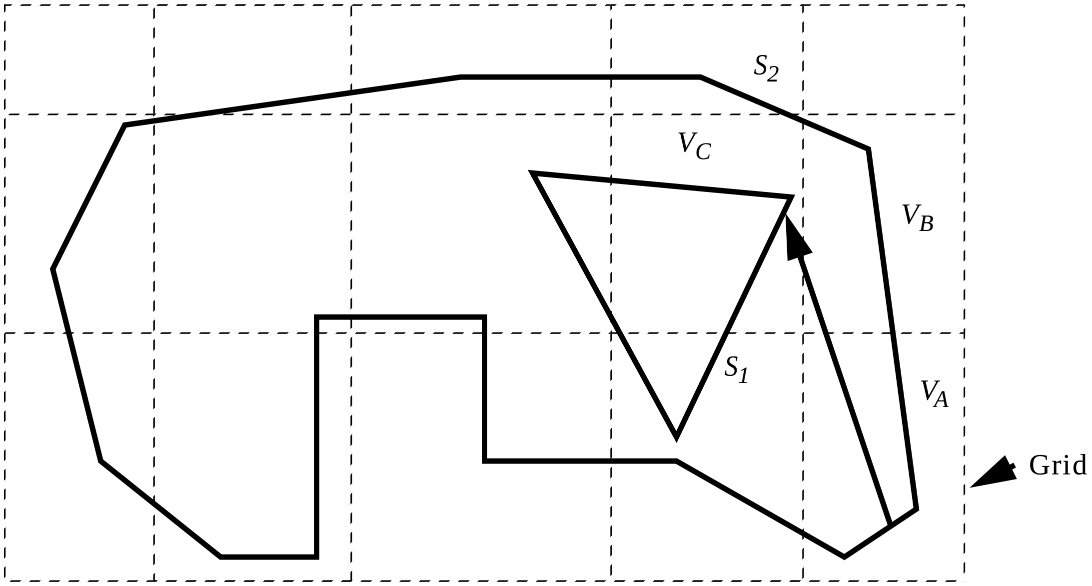


Figure 2.3 Example of a Non-uniform Grid

path are checked. Second, since voxels are traversed in order, surfaces in voxels further from the origin of the photon are checked only if an intersection is not found in an earlier voxel. For large geometries, the reduction in search time can be very significant.

It should be noted that if an intersection is found outside the current voxel, it must be rejected, as it is possible that the photon might strike another surface which, while not in the current voxel, contains an intersection point between the current voxel and the rejected intersection point. For example, in Fig. 2.3, the photon is traced from its point of emission in voxel V_A through voxel V_B to its intersection point in voxel V_C. In voxel V_B, the potential intersection point on surface S_2 is found, and must be rejected because the intersection point is outside the voxel. The true intersection point on surface S_1 is found only after the photon enters voxel V_C.

Arvo and Kirk [1989] classify spatial subdivision into two general categories: non-uniform and uniform. In non-uniform subdivision, space is discretized into regions of various sizes to allow the voxel density to be greater where the surface density is greater. Two of the most popular examples of this technique are the octree [Glassner, 1984] and the binary space partition (BSP) tree [Kaplan, 1987; Sung and Shirley, 1992]. Octrees are created by recursively dividing a rectangular volume around the geometry into eight subordinate octants until the resulting “leaf” voxels meet some criterion for stopping, such as a certain maximum number of surfaces per voxel. A BSP tree, on the other hand, is formed by partitioning space at each level of the tree into two pieces using a separating plane. While the planes are often aligned with the coordinate axes, they do not have to be [Chin, 1995].

While both the octree and the BSP tree store the geometry information efficiently, moving from voxel to voxel within a tree requires some calculation. Uniform spatial division (USD) [Fujimoto et al., 1986; Amanatides and Woo, 1987; Cleary and Wyvill, 1988] constitutes another approach. In USD, a regular three-dimensional grid of voxels of uniform size is placed over the geometry. While not dividing the geometry as efficiently as the above two methods, the “next” voxel can be found by fast incremental calculation. More voxels are usually traversed in USD but the cost of traversing the voxels is less.

Other methods do exist, for example, see Samet [1989] and Sung [1991]. These methods and most others are just combinations and variations of the octree, BSP tree, and USD methods mentioned above.

Using too many voxels can create inefficiencies. To search the surfaces inside each voxel quickly, a list must be made for each voxel of surfaces completely or partially inside it. As the number of voxels increases, the memory required for these lists and the time required to generate these lists increases. Also, it must be remembered that surface intersections outside the voxel must be rejected. If voxels are too small, surfaces will span several voxels and several intersection calculations for a surface will have to be rejected until the voxel with the intersection is entered. One way to avoid these repetitive calculations is to keep track of past intersection calculations. This is termed the “mailbox” technique and will be discussed below. It should also be noted that if the grid is too fine, time spent traversing the voxels will be significant. This can be a particular problem when using USD because the number of empty voxels generally increases with the number of voxels.

Although some comparisons of the three spatial division methods mentioned have been done, it is not clear which of the methods is the most efficient; different geometries have been used for each test. Since geometries can vary widely, each method does better on some geometries than others. Sung and Shirley [1992] have found that axis-aligned octrees and BSP trees give similar performance. Fujimoto et al. [1986] have found that, for the sample problems they did, USD is an order of magnitude faster than octrees. Both the octree [Panczak, 1989; Chin et al., 1992] and USD [Koeck, 1988] methods have been implemented in radiative heat transfer Monte Carlo codes, but no comparison of the benefits of the two methods has been made.

The general consensus about USD is that it works well but requires a lot of memory. In fact, this is one of the major complaints against the method [Sung and Shirley, 1992; Watt and Watt, 1992]. Sung [1991] has compared USD to various octree methods. In the one case where there was enough memory for the number of voxels USD required, it outperformed all other methods.

In this work, USD is chosen for two reasons. First, a version of the USD called the Margolies algorithm [Maltby, 1987; Burns et al., 1990; Maltby and Burns, 1991] has already been implemented in MONT3D. This algorithm has been used for years, has worked well, and has proven very robust. Still, there is room for improvement. As pointed out above, photons under USD usually traverse more voxels than in the octree and BSP methods. Therefore, for the USD method to be more efficient than the octree and BSP tree method, the algorithm that determines the next voxel to enter must be very efficient. The next section describes an improved algorithm that is quite different from the one used earlier. Again, the major difference is that the previous algorithm stressed calculating the intersection point with the voxel boundaries, while the new algorithm only calculates the intersection distance to the voxel boundaries.

Secondly, as long as enough voxels are used to keep the average and maximum number of surfaces per voxel low, USD appears to be the best algorithm. As shown in the results below, with more memory capacity available today, USD is definitely feasible.

The Margolies algorithm is actually a variant of USD because it does not require uniform voxels; it also allows the use of a non-uniform grid in which the grid is still divided into rows and columns, but the spacing between X, Y, and Z divisions is variable. An example is shown in Fig. 2.3. More about the advantages and disadvantages of non-uniform grids will be discussed below.

2.5 The Voxel Tracing Algorithm

This section outlines the algorithm for tracing using uniform or non-uniform grids. A less detailed description of the use of USD in radiative Monte Carlo is given by Koeck [1988]. As shown in Fig. 2.3, whether the grid is uniform or non-uniform, the grid has $N_{V,k}$ voxels along each axis and $N_{V,k} + 1$ grid planes along each axis, where k equals {1, 2, 3} for the {X, Y, Z} axes, respectively.

The key to the speed of the algorithm is that since the voxel boundaries are aligned with the axes, it is very easy to determine the next voxel to be entered. Although there are six sides to a voxel, only three sides, indicated by photon direction, have to be checked. Also, since all the voxel sides intersect, the side the shortest distance from the photon's origin is guaranteed to be the side that is intersected. Furthermore, due to the fact that the voxel sides are aligned with the axes, $t_{V,k}$, the distance to intersect a voxel side along each axis can be derived from the “distance to a plane” equation in Section 2.2, and has the following form:

$$t_{V,k} = (P_{k,l} - R_{0,k}) / E_k \tag{2.5}$$

where l is the index of the next voxel plane the photon will cross along the k^{th} axis, and P is the coordinate along the k^{th} axis for that plane. If the grid is uniform, then $t_{VB,k}$, the distance between voxel boundaries along an axis, is constant. Therefore, for uniform grids, while the first value of $t_{V,k}$ must be calculated using eqn. (2.5), for subsequent voxels $t_{V,i}$ can be updated by using the equation ($t_{VB,k}$ may be negative):

$$t_{V,k} = t_{V,k} + t_{VB,k} \tag{2.6}$$

The basic algorithm is simple.

1. Determine the emitting voxel cell and first $P_{k,l}$ values. For non-uniform grid divisions, these are found by bisection and for uniform grid divisions by direct interpolation.
2. Calculate $t_{V,k}$ for each axis at the photon emission point.
3. Determine the minimum $\boldsymbol{t_V}$ value.
4. Search for the shortest distance to intersection within the voxel. All surfaces even partially inside the voxel must be checked. To increase the efficiency of this search, the search is done over a precomputed list which specifies the surfaces in each voxel.
5. If no intersection is found, determine the next voxel to enter by the minimum $\boldsymbol{t_V}$ value and the direction of $\boldsymbol{E}$, update the value of $t_{V,\mathrm{k}}$ along the axis traversed, and go back to step 3.

When testing for intersection points inside a voxel, once a valid intersection point is found, any intersection points further from the emission point than the current one are automatically rejected. If $t_{i,s}$, the current shortest distance to intersection, is stored, the point-in-polygon test may be skipped by any surface with a t_i value greater than $t_{i,s}$. When entering a voxel, $t_{i,s}$ should be initialized to the minimum value of $\boldsymbol{t_V}$ required to exit the voxel (at step 3 above). This will automatically reject any intersection points outside the voxel.

Since there are often many empty voxels, it is important that both the intersection calculations and the voxel traversal algorithm be coded as efficiently as possible. Several authors suggest coding the grid traversal so that it only uses integer arithmetic [Fujimoto et al., 1986; Amanatides and Woo, 1987; Cleary and Wyvill, 1988; Cohen, 1994]. However, not only will this give a minimal speed improvement, if any, it also increases the possibility of precision errors. It should be noted that even for moderately sized geometries, photon tracing must be coded in double precision or round-off errors become significant.

As noted above, the use of non-uniform spaced voxels makes the calculation of $t_{V,k}$ less efficient. This may make one wonder why non-uniformly spaced grids are desirable. The reason is that if enough is known about the geometry, voxels can be enlarged where there are few surfaces and shrunk where there are many, improving the efficiency of the algorithm. However, the usefulness of this type of grid resizing is limited. Enlarging or shrinking one voxel will affect all voxels along one or more coordinate directions. Furthermore, there is no easy way to choose a non-uniform grid *a priori*. In general, one would probably get more benefit from an optimized uniform grid code than one that allows non-uniform divisions.

2.6 Mailboxes

One of the problems with spatial subdivision is that no record is kept of past calculations. If a surface exists in several voxels, the same intersection is calculated repeatedly. To prevent this, “mailboxes” [Amanatides and Woo, 1987; Arnaldi et al., 1987; Cleary and Wyvill, 1988; Arvo and Kirk, 1989; Sung, 1991; Sung and Shirley, 1992] are used to store past calculations. Each surface has a “mailbox” that holds the results of the calculations and a photon counter value that indicates the last time those calculations were done. The photon counter is incremented by one each time a photon is emitted or reflected. When the intersection calculation is performed for a surface, the first step is to compare the value of the counter in the mailbox to the current counter. If the two are equal, then the intersection in the mailbox is used, instead of being recalculated.

It should be noted that Sung [1991] has found several cases in which mailboxes increase execution time. He notes that mailboxes are efficient only if most objects span more than one voxel. In particular, he suggests that, for the mailbox algorithm to be efficient, the objects in the scene must be larger than the voxels.

The mailbox algorithm is implemented here using two one-dimensional arrays of length number of surfaces, one for the distance to the intersection and the other for the last value of the photon counter. To implement the mailbox algorithm, only minor changes are required in the loop over surfaces in a voxel. The first change is that, for each surface, the first step is to compare the value of the photon counter stored for the surface to the current value. If the counter values do not match, then the algorithm is the same as before, except that the two one-dimensional arrays are updated for each intersection calculation. If the values do match, then the distance to surface has already been calculated. In this case, the distance is compared to $t_{i,s}$. If it is less than $t_{i,s}$, the point-in-polygon test is performed. Otherwise, the surface is rejected. It should be noted that if a surface fails any of the tests except the comparison to $t_{i,s}$, then the distance to intersection is set to a very large number, insuring that it fails the comparison to $t_{i,s}$ the next time it is encountered.

The mailbox algorithm has been tested on a number of large geometries (1,000 to 5,000 surfaces) and grid resolutions. In all cases, the mailbox technique is found to increase the run time. The reason the run times increase is believed to be that the extra arrays used by the mailbox technique overload or further overload the memory cache on the CPU. For the types of simplified surfaces and complex geometries usually modeled by this algorithm, the mailbox technique is not effective.

2.7 Grid Tracing Results

To assess the improvements mentioned above, four geometries of varying complexity are tested. Several different discretizations are tested, in hopes of determining some guidelines in selecting the optimal grid for a geometry. Besides showing the overall efficiency of the current algorithm compared to the previous one, these large geometries demonstrate the power of USD. Earlier tests of USD with small geometries (143 surfaces or less) found that USD yields reductions in run time by 50% to 80% [Burns et al., 1990; Maltby and Burns, 1991]. As will be shown below, the reductions in run time are much more significant for larger geometries.

2.7.1 Geometries

Four geometries are used: Cham, AmpA, AmpB, and Gun. They are described below.

Cham is a model of the National Ignition Facility (NIF) target chamber containing 1,382 surfaces. It is a medium-sized geometry, shown in Fig. 2.4. The front section of the sphere is removed to show the inner detail. NIF <http://lasers.llnl.gov/lasers/nif.html> is currently under construction and will contain 192 extremely powerful lasers allowing

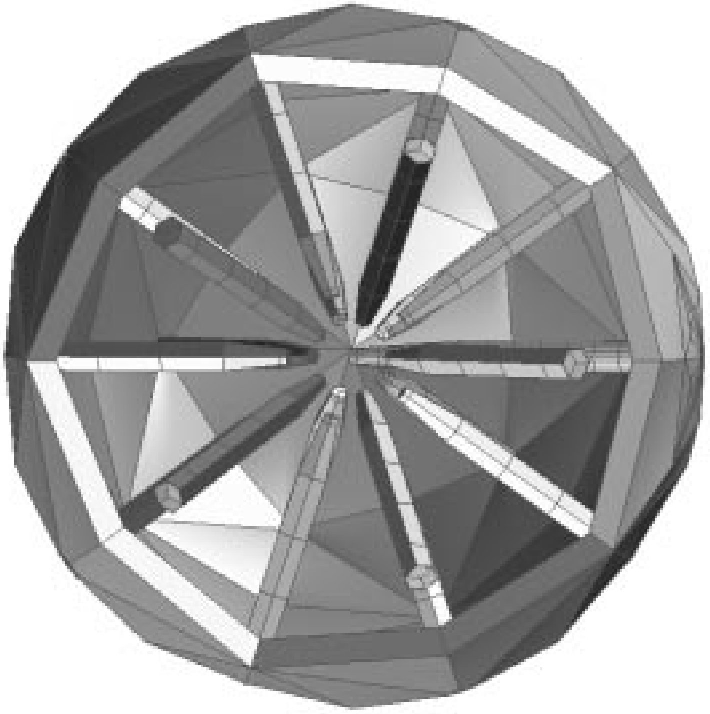


Figure 2.4 Cut-Away View of the Chamber Geometry

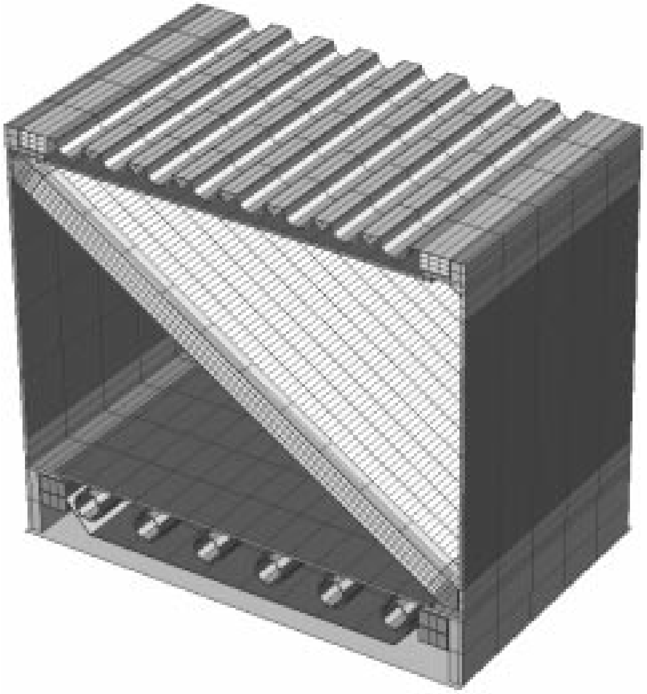


Figure 2.5 View of the Amplifier Geometry

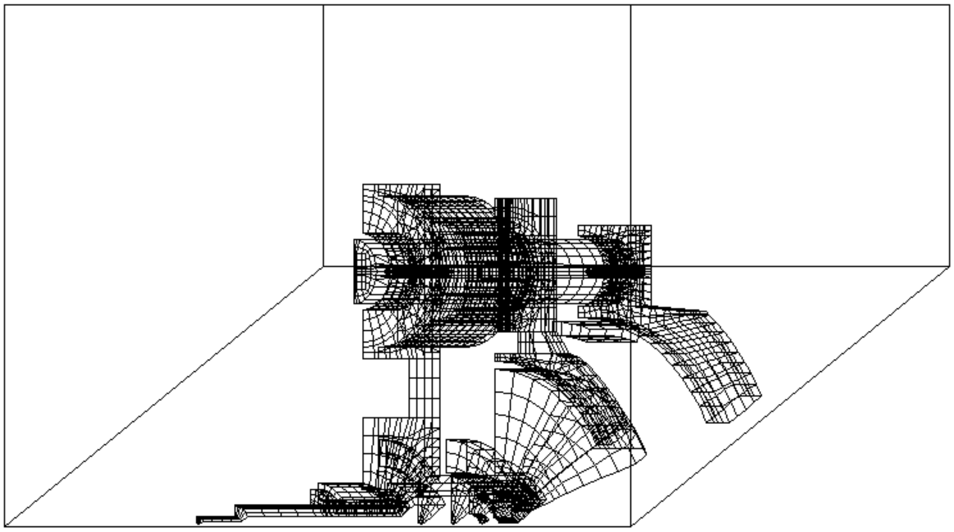


Figure 2.6 View of the Gun Geometry

research in inertial confinement fusion and other related topics. A discussion of the modeling of this chamber by TOPAZ3D is given by Raboin [1998].

AmpA and AmpB are geometries representing the NIF laser amplifier assembly as shown in Fig. 2.5. The front wall and symmetry plane across the top of the geometry have been removed to show the inner detail of assembly. AmpB, with 4,581 surfaces, is a more detailed representation than AmpA, with 3,381 surfaces. More details about the modeling of this geometry are given by Sutton et al. [1998].

Gun, shown in Fig. 2.6, models the radiation coupling between the outer parts of an electron gun. The gun is enclosed to capture escaping photons. The gun is axially symmetric and is modeled as a wedge from 0° to 60°. The edges of the wedge are modeled as specular symmetry planes. This geometry has 4, 580 surfaces; one less than AmpA.

Table 2.1: Grid Information

Geometry	Geometry Dimensions $\{X, Y, Z\}$	Grid Divisions
Cham	{9.91, 10, 10}	$m\{1,1,1\}$
AmpA	{774, 665.6, 480}	$m\{3,3,2\}$
AmpB	{774, 665.6, 480}	$m\{3,3,2\}$
Gun	{15.24, 13.20, 25.91}	$m\{8,7,14\}$; $m\{8,7,14\} + \{4,4,7\}$

Table 2.2: Photon Statistics

Geometry	Number of Original Surfaces	Number of Split Surfaces	Total Number of Surfaces	Photons Emitted per Original Surface	Total Number of Photons Emitted (Millions)
Cham	1,182	144	1,326	20,000	23.64
AmpA	3,381	0	3,381	10,000	33.81
AmpB	4,581	0	4,581	10,000	45.81
Gun	4,297	283	4,580	10,000	42.95

2.7.2 Test Description

In all tests below, surfaces are black, as using reflecting surfaces would just obscure observations about the photon intersection algorithm. Unless otherwise specified, all times given are solution times.

There are, of course, an infinite number of ways these geometries could have been “gridded.” For testing, the geometries are gridded using approximately cubical voxels. Table 2.1 gives the X, Y, Z extents of each geometry and the formula used to specify the grid divisions. m can take on any integer value. It should be noted that to better populate the graphs below, the Gun geometry uses two different grid formulas.

A different number of photons are emitted for each geometry, depending on the number of original surfaces and the number of photons per original surface as shown in Table 2.2. The Cham and Gun geometries have non-planar surfaces that are split into two triangles as described in Section 2.1. Each triangle is treated as a separate surface by the intersection routine, so the efficiency of the intersection routine depends on the total number of surfaces.

All tests are performed on a 233 MHz 604e PowerPC chip running Macintosh OS 8.0. The Absoft f77 compiler is used. Timings among repeated runs differ by 5% at most, and typically by less than 1%. To obtain accurate timing for the photon tracing only, a modified version of the code, which does not write the large output files, is used for all results.

2.7.3 Determining the Optimal Grid

While USD has been used in the past, there has been no in-depth study of its application to large-scale geometries. The purpose of this section is determine the efficiency of USD for large geometries and determine guidelines in selecting grids.

Execution times versus numbers of voxels for the current algorithm for the four geometries are given in Figs. 2.7 and 2.8. The curves marked “solution” are the solution times for the algorithm and those marked “input” are the input times for the algorithm. The curves are very flat around the optimal grid. In fact, it is hard to specify one grid as “optimal.” As stated above, the uncertainty in the results is about 5%. For each geometry, there are several points that are within 5% of the minimum value, yielding a range. Several statistics for the optimal grid range are given in Table 2.3. The results “min” and “max” are for the minimum and maximum size grid in the optimal range. The optimal grid range is quite large; the maximum grid size is around three to four times the minimum grid size. While the optimal grid varies with geometry, these results suggest that for geometries of 1,000 to 5,000 surfaces, a good first estimate of the optimal grid is 15,000 voxels.

Since determining the next voxel to enter requires less calculation than the surface intersection calculations, the optimal grid favors fewer surfaces per voxel over fewer

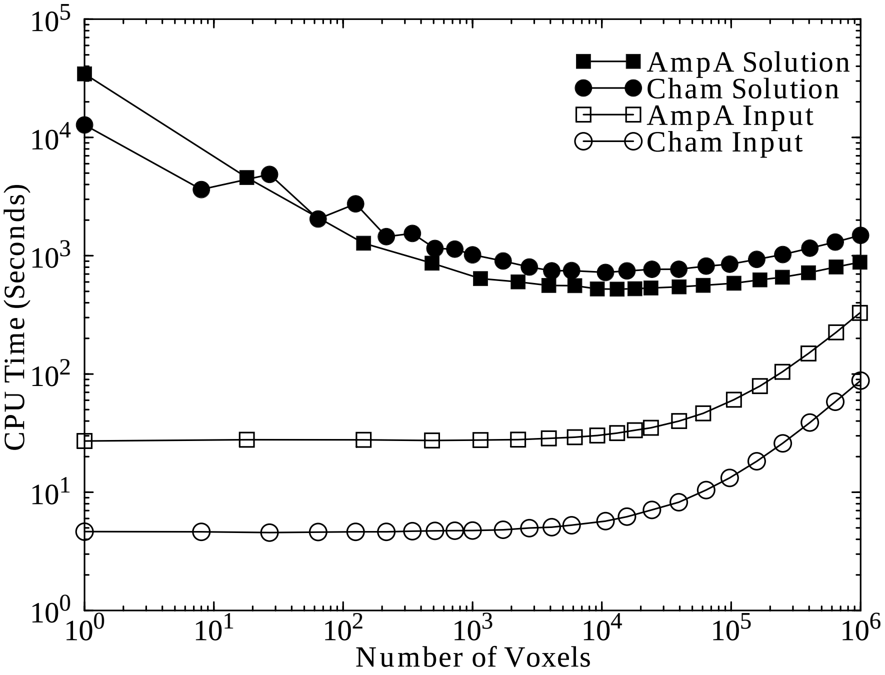


Figure 2.7 Results for the AmpA and Cham Geometries

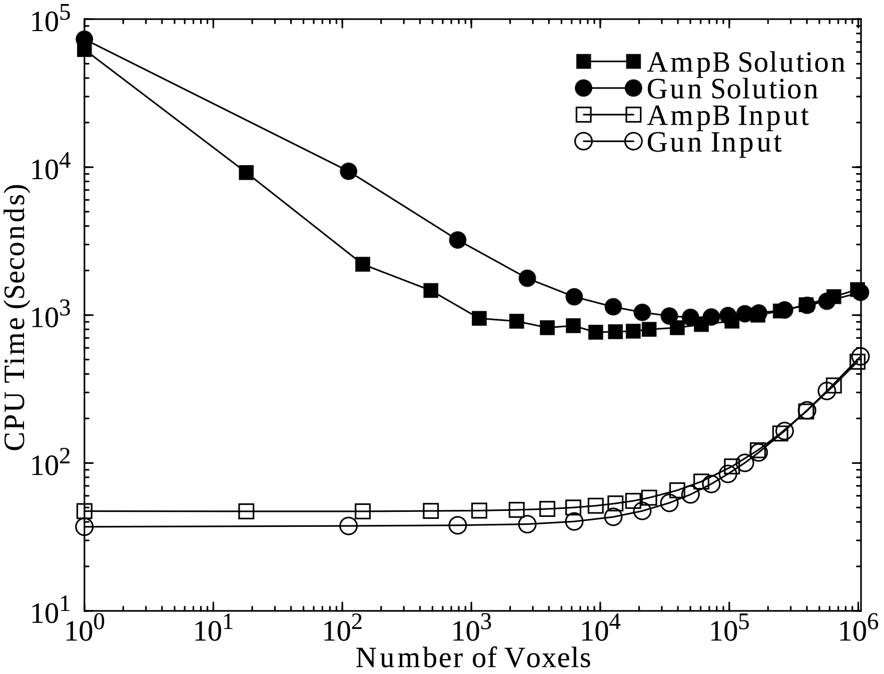


Figure 2.8 Results for the AmpB and Gun Geometries

Table 2.3: Grid Statistics

Geometry	Number of Voxels	Surfaces per Voxel	Number of Empty Voxels	Surfaces per Non-empty Voxel
Cham min	4,096	2.40	1,521	3.82
Cham max	15,625	1.52	7,192	2.82
AmpA min	9,216	2.44	3,942	4.26
AmpA max	39,546	1.10	23,578	2.73
AmpB min	9,216	2.66	4,114	4.80
AmpB max	23,958	1.53	13,476	3.50
Gun min	15,360	1.52	5,880	2.46
Gun max	65,910	0.887	30,084	1.63

empty voxels. For the geometries tested, 37% to 60% of the voxels are empty when execution time is within 5% of the optimal time.

The input times for the code are shown in Figs. 2.7 and 2.8. The input time grows quickly as the number of voxels becomes large. For the geometries tested this does not present a problem, as in the optimal grid range, the input time is insignificant relative to the solution time. Furthermore, production runs are much longer than the ones for these tests. Usually, more photons are emitted per surface, particularly as the number of surfaces increases. Also, tracing times are longer since reflection exists. At even larger scales, this may become a problem, but only future testing can determine this.

As noted in Table 2.2, each geometry emits a different number of photons, so comparing the results can be difficult. For this reason, Table 2.4 provides results in time per photon. The “optimal grid” solution time is the shortest execution time for that geometry. The “no grid” solution time is the solution time when no grid is used. The “no grid” case is the same as specifying only one grid cell or voxel, and includes the insignificant overhead of initializing the voxel tracing routine once per photon. The other columns in the table are described below.

Table 2.4: Current Algorithm Statistics

Geometry	CPU Time Per Photon (μs)			Speedup Ratios (No Grid / Optimal)		α_o
	“Optimal Grid” Solution Time	“No Grid” Solution Time	Time Not Tracing	Solution Time	Tracing Time	
Cham	30.6	541	5.51	17.7	21.4	0.574
AmpA	15.4	1,020	5.68	66.1	104	0.429
AmpB	16.7	1,360	5.72	81.4	123	0.429
Gun	18.6	1,370	5.72	73.3	106	0.447

The “time not tracing” is the time spent by the solution phase in activities other than tracing, including specifying the photon emission point and direction, determining if the photon is absorbed or reflected, and the overhead of the algorithm. If the surfaces are not all black, this would also include determining the photon’s new direction after being reflected.

The “time not tracing” was obtained by running the codes with the intersection routines commented out. While it is difficult to assess how accurately this measures the time not tracing, similar estimates were obtained by profiling the code on an RS / 6000 workstation.

The speedup ratios listed in the table are the solution or time spent tracing for the “no grid” case divided by the same result for the optimal grid case. Comparing the “time not tracing” to the optimal solution time, it can be seen that only 63% to 82% of the optimal solution time is spent in photon tracing. This represents quite an improvement over the “no grid” solution where 99% or more of the time is spent in photon tracing. The “time not tracing” appears to be a weak function of the number of surfaces.

Table 2.4 indicates that the speedup can exceed a factor of 80 for the entire solution phase and over a factor of 120 for the intersection calculations. However, it would be helpful to quantify the speedup as a function of the number of surfaces. Cleary and Wyvill [1988] have done an in-depth theoretical analysis of a USD algorithm using a mailbox scheme. They have found that the run time is a complex function that depends on the number of surfaces, the average times for four different parts of the algorithm, the mean area of the objects, and the mean circumference of the objects.

A simplified model that is often used in these cases is the time per photon, t_p, given by:

$$t_p = aN_{surf}^{\alpha} + b \tag{2.7}$$

where N_{surf} is the number of surfaces. For the “no grid” case, α is one since all N_{surf} surfaces have to be checked. If it is assumed that a is constant for a geometry and that b is equivalent to the “time not tracing” per photon, then α_o, the value of α for the optimal grid can be determined. As the table shows, α_o lies between 0.43 and 0.57. This is consistent with estimates of the dependence of optimal grid tracing time on N_{surf} derived from an analytical perspective [Burns and Pryor, 1999].

2.7.4 Memory Requirements

Although photon tracing is often performed on large geometries with thousands of surfaces, its memory requirements are usually not prohibitive. Ignoring the memory required for the Margolies grid algorithm which will be discussed below, the storage required is on the order of the number of surfaces. While it is true that the final output, the exchange number matrix, is number of surfaces squared in size, only a small block of rows of that matrix are stored in memory at any one time. For the geometries used in this study of 14,080 nodes, 4,581 surfaces and six surface materials types, excluding the storage for the exchange matrix and the uniform grid, only 1.7 megabytes of storage are required for the entire Monte Carlo code. Storing 200 rows of the exchange matrix in memory at a time requires about another 3.5 megabytes. If the block size is decreased, the code must write its results to disk more frequently, thereby increasing I/O time. For the Margolies grid algorithm, the extra storage required is around 0.1 megabytes for 10,000 voxels and around 0.7 megabytes for 100,000 voxels, an insignificant increase.

2.7.5 Previous Algorithm Results

To gauge the effectiveness of the algorithm discussed in this chapter, comparisons are made to the previous version of the algorithm [Maltby, 1987; Burns et al., 1990; Maltby and Burns, 1991; Burns and Pryor, 1999] which is publicly available [Maltby et al., 1994; <http://www.colostate.edu/~pburns/monte/documents.html>]. Several runs are performed and some of the results for the older version of the algorithm are given in Table 2.5. The speedup ratios listed in Table 2.5 are the solution time or time spent tracing for the old algorithm optimal grid case divided by that for the new algorithm optimal grid case. Gathering data from all the runs, the new algorithm is 33% to 45% faster than the old algorithm. Interestingly, the time not tracing is slightly better for the old algorithm, as new features added to the code slow it down slightly. The improvements to the voxel tracing algorithm seem to be just as important as the improvements to intersection calcu-

Table 2.5: Previous Algorithm Statistics

Geometry	CPU Time Per Photon (μs)		Optimal Algorithm Ratios (Old/New Algorithm)	
	"Optimal Grid" Solution Time	Time Not Tracing	Solution Time	Tracing Time
Cham	53.0	5.31	1.73	1.90
AmpA	22.9	5.38	1.48	1.79
AmpB	25.7	5.43	1.54	1.84
Gun	29.5	5.43	1.59	1.87

lations. The timing curves for the old algorithm show similar tends compared to the curves for the new algorithm results and exhibit very similar optimal grid ranges. As mentioned in Section 2.2, the main difference between the algorithms is that for both surfaces and voxels the current algorithm focuses calculating the distance to intersection, t_i, instead of the point of intersection, $\boldsymbol{R_i}$.

2.8 Conclusions

An in-depth study has been completed of a Monte Carlo particle (photon) tracing algorithm for large geometries with arbitrary planar surfaces in nonparticipating media. Methods from the computer graphics field of ray tracing have been reviewed and implemented. An efficient algorithm for determining intersections has been presented. Furthermore, an assessment of ways to further increase the efficiency of the algorithm has been conducted. Uniform spatial division (USD) is chosen as the most promising technique, based upon its simplicity and effectiveness. The mailbox technique is found in all cases to increase execution times, and is therefore not recommended. Four geometries containing between 1,000 and 5,000 surfaces each were chosen for testing. For these geometries, USD yields speedups in run time of factors as great as 81. While the optimal subdivision varies with geometry, execution time varies slowly with number of voxels (grid cells). Good results are obtained with 15,000 voxels. The memory requirements for USD are found to be slight; less than a megabyte of memory is required to store the grid variables for the optimal grids for all geometries tested. The memory requirements for the rest of the Monte Carlo code itself are also found to be slight, between 1.7 and 5.2 megabytes of additional memory are required for the largest geometry tested. The current photon tracing algorithm discussed in this chapter is found to be 33% to 45% faster than the previous algorithm.

CHAPTER 3 IMPROVING MONTE CARLO RESULTS BY APPLYING RECIPROCITY AND CLOSURE

In this chapter, the improvement of Monte Carlo results by enforcing reciprocity and closure is investigated. Several different statistical estimation techniques, particularly those based on the reciprocity estimation smoothing (RES) method discussed in Chapter 1, are derived and tested to determine the best method. The effects of the best method on individual surfaces and individual η are determined by examining a large number of test runs of two large geometries. Furthermore, single run error metrics are derived so that the general accuracy improvement can be assessed quantitatively.

3.1 Applying Reciprocity by Statistical Estimation

The first step in attempting to improve the accuracy of the Monte Carlo results in the RES method is to use an estimator to enforce reciprocity using η_{kl} and η_{lk}. Such estimated values are denoted with a $\hat{}$ in this work. From statistics, proper estimators of this type are expected to improve the accuracy and to decrease the uncertainty in the results. Later in this work numerical experiments are performed to assess various estimators. Below, in the first subsection, the concept of the confidence interval is introduced. In the next section, some observations are made on reciprocity effects on the standard deviation. In the last three subsections, four different estimation techniques are presented, each based on a different statistical theory.

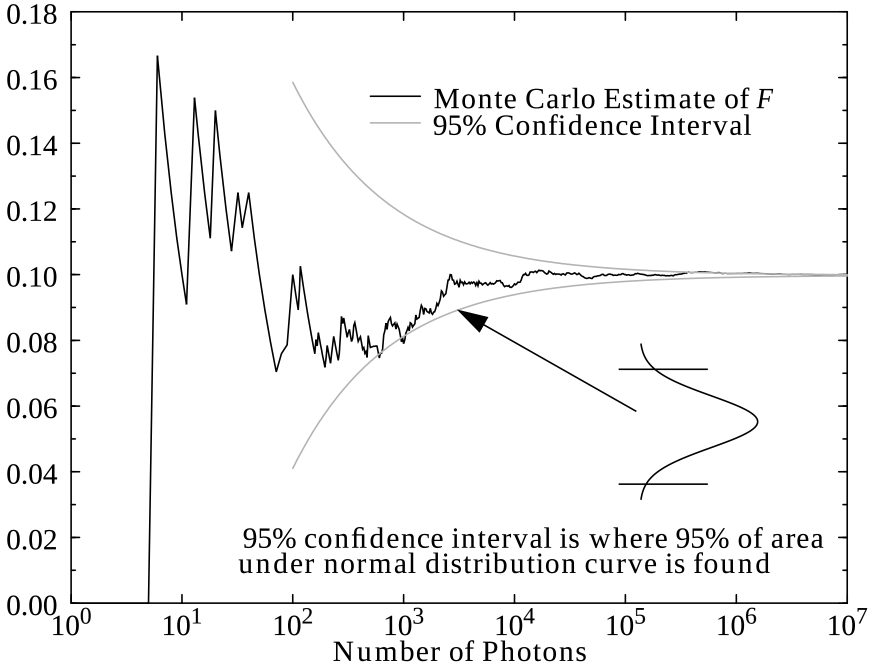


Figure 3.1 Monte Carlo Results and Confidence Interval for $\tilde{F}_{kl}$ = 0.0998

3.1.1 Confidence Intervals

One advantage of Monte Carlo methods is that the uncertainty in the results is generally known. The uncertainty is often measured by the *P*% confidence interval which when centered around the observed result has the property that *P*% of such intervals computed from different random data contain the true value of the parameter.

An example of a 95% confidence interval is shown in Fig. 3.1 which graphs the Monte Carlo calculation of the exchange fraction, F_{kl}, that has a true value, $\tilde{F}_{kl}$, of 0.0998. The “envelope” shown in the figure is the 95% confidence interval centered around the true value.

The halfwidth of *P*% confidence interval for a normal distribution, $h_{nor}(p)$, is related to the error function, erf:

$$h_{nor}(p) = \sqrt{2}\mathrm{erf}^{-1}(p)\sigma = \omega(p)\sigma \tag{3.1}$$

where p is the fractional probability of the interval, $P/100$; erf^{-1} is the inverse error function; and ω is the confidence interval constant. For 95% confidence, ω is 1.96. It should be noted that the confidence interval itself, $C_{nor}(p)$, is the range of $\pm h_{nor}(p)$ centered around either the true value of μ or its estimate.

$$C_{nor}(p) = [\mu - h_{nor}(p), \mu + h_{nor}(p)] \tag{3.2}$$

For the binomial distribution that governs F_{kl}'s obtained using the collision based Monte Carlo method of this work, F_{kl} is said to be well approximated by a normal distribution when about nine [Martin, 1971], 11 [Ross, 1988], or more photons are absorbed by a surface. The fact that the binomial distribution is usually well modeled by the normal distribution is not surprising, since the normal distribution was first derived by the French mathematician de Moivre in 1733 as an approximation of the binomial distribution when the number of trials is large and p (F_{kl}) is not close to zero or one [Evans et al., 2000]. From the variance for a binomial distribution [Ross, 1988] and eqn. (3.1), the half-width of the confidence interval for F_{kl}, $h_{F_{kl}}(p)$ is:

$$h_{F_{kl}}(p) = \omega(p)\sqrt{\frac{F_{kl}(1 - F_{kl})}{N_k}} \tag{3.3}$$

where N_k is the number of photons emitted by surface k. Eqn. (3.3) is used to construct Fig. 3.1. The confidence interval is calculated using $\tilde{F}_{kl}$, the known “true” value of F_{kl} which is 0.0998. As the number of emissions increases, the standard deviation and the uncertainty decrease. The 95% confidence interval marks where 95% of the area under each normal distribution curve is found.

While Fig. 3.1 shows that the Monte Carlo results are generally within a confidence interval calculated around and using $\tilde{F}_{kl}$ (i.e. the mean of the distribution), $\tilde{F}_{kl}$ is not usually known. Fortunately, as long as N_k is large enough, then the calculated halfwidth of the confidence interval is essentially the same as the true halfwidth. That being the case, there is a P% chance that $\tilde{F}_{kl}$ is in a P% calculated confidence interval centered around the calculated value of the mean (F_{kl}). More generally, calculated confidence intervals are a good measurement of the range in which the correct answer is expected to be found any time the standard deviation of the results can be calculated fairly accurately.

Early research on applying the confidence interval to Monte Carlo methods was done by Maltby. He derived a quantity, δ, he termed a confidence interval or confidence level by applying the Central Limit Theorem to F_{kl} derived from the Bernoulli trial distribution [Maltby, 1990; Burns et al., 1990; Maltby and Burns, 1991; Burns et al., 1992; Dolaghan et al., 1992; Loehrke et al., 1995; Burns and Pryor, 1999; Zeeb et al., 1999; Zeeb and Burns, 2000]. In this work, δ is called the halfwidth of the fractional confidence interval. It is related to halfwidth of the confidence interval, $h_{F_{kl}}(p)$, defined above as follows:

$$\delta(p, F_{kl}) = \frac{h_{F_{kl}}(p)}{F_{kl}} \tag{3.4}$$

3.1.2 Some Reciprocity Effects on the Standard Deviation

Let it be assumed that the exact values of the exchange fractions, $\tilde{F}_{kl}$ and $\tilde{F}_{lk}$, are known. Then, due to reciprocity, the exact total exchange areas are equal.

$$\tilde{\eta}_{kl} = \Omega_k \tilde{F}_{kl} = \Omega_l \tilde{F}_{lk} = \tilde{\eta}_{lk} \tag{3.5}$$

In general, the uncertainty (standard deviations) of the original Monte Carlo results, F_{kl} and F_{lk}, differ. Since the distribution is binomial [Ross, 1988]:

$$\sigma_{\tilde{\eta}_{kl,\,\mathrm{bin}}} = \Omega_k\sqrt{\frac{\tilde{F}_{kl}(1-\tilde{F}_{kl})}{N_k}} = \tilde{\eta}_{kl}\sqrt{\frac{(1-\tilde{F}_{kl})}{N_k\tilde{F}_{kl}}} = \frac{\tilde{\eta}_{kl}}{\omega(p)}\delta(p,\tilde{F}_{kl}) \tag{3.6}$$

where N_k is the number of photons emitted by surface k and $\omega(p)$ and δ are the confidence interval constant and the halfwidth of the fractional confidence interval defined in Section 3.1.1. Similarly,

$$\sigma_{\tilde{\eta}_{lk,\,\mathrm{bin}}} = \Omega_l\sqrt{\frac{\tilde{F}_{lk}(1-\tilde{F}_{lk})}{N_l}} = \tilde{\eta}_{kl}\sqrt{\frac{(1-\tilde{F}_{lk})}{N_l\tilde{F}_{lk}}} = \frac{\tilde{\eta}_{kl}}{\omega(p)}\delta(p,\tilde{F}_{lk}) \tag{3.7}$$

Using eqn. (3.5) to define $\tilde{F}_{lk}$ as a function of $\tilde{F}_{kl}$, the ratio of the two σ's is:

$$\frac{\sigma_{\tilde{\eta}_{lk,\,\mathrm{bin}}}}{\sigma_{\tilde{\eta}_{kl,\,\mathrm{bin}}}} = \sqrt{\frac{N_k(1-[\Omega_k/\Omega_l]\tilde{F}_{kl})}{N_l(\Omega_k/\Omega_l)(1-\tilde{F}_{kl})}} \tag{3.8}$$

This is a monotonically increasing function of $\tilde{F}_{kl}$ with lower bound:

$$\lim_{\tilde{F}_{kl}\to 0}\frac{\sigma_{\tilde{\eta}_{lk,\,\mathrm{bin}}}}{\sigma_{\tilde{\eta}_{kl,\,\mathrm{bin}}}} = \sqrt{\frac{N_k}{N_l}\frac{\Omega_l}{\Omega_k}} \tag{3.9}$$

The ratio is roughly proportional to the square root of the ratio of the Ω's and inversely proportional to the square roots of the number of photons emitted by each surface. In other words, lowest uncertainty (i. e. the smallest σ) in an η_{kl} pair is obtained from the surface which has the smallest ratio of Ω / N, e.g. from surfaces with small Ω values from which many photons are emitted. This makes sense intuitively.

3.1.3 Fractional Confidence Interval and Fractional Variance Estimation

As stated previously, the program SMOOTH [Burns et al., 1992; Dolaghan et al., 1992] was derived from the work of Maltby [1990]. The estimator used in that program is the fractional confidence interval (FCI) estimator which is a weighted average calculated using Maltby's halfwidth of the fractional confidence interval, δ, as defined in eqn. (3.4):

$$\hat{\eta}_{kl,\,FCI} = \kappa_{FCI}\eta_{kl} + (1-\kappa_{FCI})\eta_{lk} \tag{3.10}$$

where:

$$\kappa_{FCI} = \frac{\delta(p, F_{lk})}{\delta(p, F_{kl}) + \delta(p, F_{lk})} \tag{3.11}$$

It should be noted that the ω(*p*) terms in eqn. (3.11) cancel out, which removes the dependence of κ_{FCI} on *p*. From eqns. (3.6) and (3.7), it is possible to view κ_{FCI} as:

$$\kappa_{FCI} = \frac{\sigma_{\eta_{lk,\,\mathrm{bin}}}}{\sigma_{\eta_{kl,\,\mathrm{bin}}} + \sigma_{\eta_{lk,\,\mathrm{bin}}}} \tag{3.12}$$

Therefore κ_{FCI} takes into account the differences in the Ω's, *F*'s, and *N*'s.

For constant weighting factors, areas, and emissivities in eqn. (3.10), the variance of $\hat{\eta}_{kl,\,FCI}$ is:

$$\sigma^2_{\hat{\eta}_{kl},\,FCI} = \kappa_{FCI}{}^2\Omega_k^2\frac{F_{kl}(1-F_{kl})}{N_k} + (1-\kappa_{FCI})^2\Omega_l^2\frac{F_{lk}(1-F_{lk})}{N_l} \tag{3.13}$$

Loehrke et al. [1995] uses a somewhat different estimator, termed the fractional variance (FV) estimator in this work. The FV estimator is similar to the FCI estimator except that the weight, $\kappa_{FV,}$ depends on the variance (the square of δ):

$$\kappa_{FV} = \frac{\delta^2(p, F_{lk})}{\delta^2(p, F_{kl}) + \delta^2(p, F_{lk})} \tag{3.14}$$

The value and variance for FV estimation are calculated using equations identical to eqns. (3.10) and (3.13) with κ_{FCI} replaced by κ_{FV}. As explained by Loehrke et al. [1995], this estimator can be seen as a variant of the minimum variance estimator described in the next section. The FV estimator is also used by Chen et al. [2000] and Liu et al. [2001], who applied it to exchange factors.

As for the binomial confidence interval from which δ is derived, δ and eqns. (3.10) to (3.14) are technically only valid when N_{kl} and N_{lk} are sufficiently large so that δ represents a normally distributed variable. Still, it is standard practice to use eqns. (3.10) and (3.13) and either eqn. (3.11) or (3.14) for all values of N_{kl} and N_{lk}. However, if F_{kl} is zero, then $\delta(p, F_{kl})$ is infinite, i.e., a singularity exists and requires accommodation. When one F is zero, the value produced by either estimator is taken as the non-zero η. Alternatively, if N_{kl} and N_{lk} are both large enough, then $\hat{\eta}_{kl}$ by either method is normally distributed with confidence interval defined by eqn. (3.2) using the proper value of either $\sigma_{\hat{\eta}_{kl}, FCI}$ or $\sigma_{\hat{\eta}_{kl}, FV}$.

3.1.4 Minimum Variance Estimation

As shown by the confidence interval for a normally distributed variable described in Section 3.1.1, the uncertainty of a random variable is generally some function of its variance. Therefore, one type of estimation is to minimize the variance. Martin [1971] derives a minimum variance estimator for n populations that have the same "true" mean, $\tilde{\mu}$, but differing "true" variances, $\tilde{\sigma}_i^2$. The n populations can even have differing distributions. The minimum variance (MV) estimator and associated variance are given by:

$$\hat{\mu}_{MV} = \sum_{i=1}^{n} \frac{\mu_i}{\sigma_i^2} \Big/ \left(\sum_{i=1}^{n} \frac{1}{\sigma_i^2} \right) \tag{3.15}$$

$$\hat{\sigma}_{MV}{}^2 = \sum_{i=1}^{n} 1 \Big/ \left(\sum_{i=1}^{n} \frac{1}{\sigma_i^2} \right) \tag{3.16}$$

Both of the above equations rely on the assumption that the sample sizes of the n populations are large enough such that σ_i can be used in place of $\tilde{\sigma}_i$ with little loss of accuracy.

$\hat{\eta}_{kl,\ MV}$ is calculated as follows. First, it should be noted that due to the reciprocity condition, eqn. (1.7), the means, $\tilde{\eta}_{kl}$ and $\tilde{\eta}_{lk}$, are equal. Furthermore, if the variables are considered binomially distributed, then [Ross, 1988]:

$$\sigma^2_{\eta_{kl,\,\mathrm{bin}}} = \frac{\Omega_k^2 F_{kl}(1 - F_{kl})}{N_k} \tag{3.17}$$

Inserting η_{kl}, η_{lk}, $\sigma^2_{\eta_{kl}}$, and $\sigma^2_{\eta_{lk}}$ into eqns. (3.15) and (3.16) and simplifying:

$$\hat{\eta}_{kl,\ MV} = \left(\frac{F_{kl}N_k}{\Omega_k(F_{kl} - F_{kl}^2)} + \frac{F_{lk}N_l}{\Omega_l(F_{lk} - F_{lk}^2)} \right) \Big/ \left(\frac{N_k}{\Omega_k^2(F_{kl} - F_{kl}^2)} + \frac{N_l}{\Omega_l^2(F_{lk} - F_{lk}^2)} \right) \tag{3.18}$$

$$\sigma^2_{\hat{\eta}_{kl},\ MV} = 1 \Big/ \left(\frac{N_k}{\Omega_k^2(F_{kl} - F_{kl}^2)} + \frac{N_l}{\Omega_l^2(F_{lk} - F_{lk}^2)} \right) \tag{3.19}$$

This estimator becomes ill-defined when either F is equal to zero or one. If one (or both) of the F's is (are) zero, $\hat{\eta}_{kl,\ MV}$ and $\sigma^2_{\hat{\eta}_{kl},\ MV}$ are both equal to zero. If both F's equal one, then $\hat{\eta}_{kl,\ MV} = \Omega_k = \Omega_l$ and $\sigma^2_{\hat{\eta}_{kl},\ MV}$ equals zero. If one of the F's is equal to one, then $\hat{\eta}_{kl,\ MV}$ is equal to the Ω of the emitting side for that F and $\sigma^2_{\hat{\eta}_{kl},\ MV}$ equals zero. As with the FCI and FV estimators above, if the original F's are normally distributed then $\hat{\eta}_{kl,\ MV}$

should be normally distributed with confidence interval defined by eqn. (3.2) using $\sigma_{\hat{\eta}_{kl},\ MV}$. One advantage of this particular estimator is that as long as both distributions have the same mean, $\tilde{\mu}$, the statistics are valid no matter what the distributions are. A disadvantage of this estimator is that if one of either of the η's is zero, then $\hat{\eta}_{kl,\ MV}$ is also zero which is obviously incorrect.

3.1.5 Binomial Maximum Likelihood Estimation

A common method of estimation is maximum likelihood estimation where the probability for a series of independent variables or observations are used to determine an unknown estimator, θ, common to the probability functions of the variables. The probability of all independent observations occurring together is given by the likelihood function, $L(\theta)$, which is the product of the probability functions for each variable [Martin, 1971]:

$$L(\theta) = \prod_{i=1}^{n} p(x_i;\theta) \tag{3.20}$$

By choosing θ to maximize L, the most probable value of θ can be determined. To understand the reasoning behind this, recall that in statistical thermodynamics values are determined by finding the state with the highest probability.

The θ that maximizes L fulfills the following conditions:

$$\frac{\partial}{\partial\theta}L(\theta) = 0 \qquad \frac{\partial}{\partial\theta^2}^2 L(\theta) < 0 \tag{3.21}$$

Since L is greater than zero, the first part of eqn. (3.21) is equivalent to:

$$\frac{1}{L}\frac{\partial}{\partial\theta}L(\theta) = \frac{\partial}{\partial\theta}\ln(L(\theta)) = 0 \tag{3.22}$$

One unique feature of the radiative heat transfer collision based Monte Carlo method used in this work is that the reciprocity relation, eqn. (1.10), can be used to obtain a special binomial maximum likelihood estimation. The reciprocity condition allows the following definitions to be used:

$$F_{kl} = \frac{\hat{\eta}_{kl}}{\Omega_k} \qquad F_{lk} = \frac{\hat{\eta}_{kl}}{\Omega_l} \tag{3.23}$$

where $\hat{\eta}_{kl}$ is some potential estimation of η_{kl}. From the probability mass function of a binomial variable [Ross, 1988], L is:

$$L(\hat{\eta}_{kl}) = p_{bin}(N_{kl}, N_k, \hat{\eta}_{kl})p_{bin}(N_{lk}, N_l, \hat{\eta}_{kl}) = \\ \frac{N_k!N_l!}{(N_k - N_{kl})!N_{kl}!(N_l - N_{lk})!N_{lk}!}\left(\frac{\hat{\eta}_{kl}}{\Omega_k}\right)^{N_{kl}}\left(\frac{\Omega_k - \hat{\eta}_{kl}}{\Omega_k}\right)^{N_k - N_{kl}}\left(\frac{\hat{\eta}_{kl}}{\Omega_l}\right)^{N_{lk}}\left(\frac{\Omega_l - \hat{\eta}_{kl}}{\Omega_l}\right)^{N_l - N_{lk}} \tag{3.24}$$

To determine the binomial maximum likelihood (BML) estimator, $\hat{\eta}_{kl,\,BML}$, the derivative of the natural log of eqn. (3.24) with respect to $\hat{\eta}_{kl,\,BML}$ is calculated and set to zero.

$$\left.\frac{\partial}{\partial\hat{\eta}_{kl}}\ln L(\hat{\eta}_{kl})\right|_{\hat{\eta}_{kl} = \hat{\eta}_{kl,\,BML}} = \frac{N_{kl}}{\hat{\eta}_{kl,\,BML}} - \frac{N_k - N_{kl}}{\Omega_k - \hat{\eta}_{kl,\,BML}} + \frac{N_{lk}}{\hat{\eta}_{kl,\,BML}} - \frac{N_l - N_{lk}}{\Omega_l - \hat{\eta}_{kl,\,BML}} = 0 \tag{3.25}$$

or

$$(N_k + N_l)\hat{\eta}^2_{kl,\,BML} - (\Omega_k(N_{kl} + N_l) + \Omega_l(N_{lk} + N_k))\hat{\eta}_{kl,\,BML} + \Omega_k\Omega_l(N_{kl} + N_{lk}) = 0 \tag{3.26}$$

This is a quadratic equation with solution:

$$\hat{\eta}_{kl,\,BML\pm} = \frac{b \pm \sqrt{b^2 - 4ac}}{2a} \qquad a = N_k + N_l$$
$$b = \Omega_k(N_{kl} + N_l) + \Omega_l(N_{lk} + N_k) \qquad c = \Omega_k\Omega_l(N_{kl} + N_{lk}) \tag{3.27}$$

While not able to be proven analytically from the equation itself, tests of this formula suggest that the roots are always real. Furthermore, testing shows that $\hat{\eta}_{kl,\,BML+}$ always appears to be invalid because it is too large (i.e. $F > 1$). The easiest way to discern why this is so to realize that:

$$\frac{b}{2a} = \frac{\Omega_k(N_{kl} + N_l) + \Omega_l(N_{lk} + N_k)}{2(N_k + N_l)} = \frac{1}{2}\left(\frac{N_k\eta_{kl} + N_l\eta_{lk}}{N_k + N_l} + \frac{N_k\Omega_k + N_l\Omega_l}{N_k + N_l}\right) \tag{3.28}$$

Therefore, $b/2a$ is a simple average of two other averages weighted by the number of photons emitted: one involving the η_{kl}'s and the other involving the Ω's. The effect of the averaged Ω value is to make $b/2a$ itself have a positive bias, usually very large, as an estimate of η_{kl} suggesting that the proper root for the estimator is always $\hat{\eta}_{kl,\,BML-}$, which will be referred to simply as $\hat{\eta}_{kl,\,BML}$ from this point forward.

Calculating the variance for $\hat{\eta}_{kl,\,BML}$ is difficult. In general, the variance for a maximum likelihood estimate is [Martin, 1971]:

$$\sigma^2_{\theta,\,ML} = \left(\int_{-\infty}^{\infty}(\theta - \bar{\theta}_{ML})^2 L(\theta)d\theta\right) \Big/ \left(\int_{-\infty}^{\infty} L(\theta)d\theta\right) \tag{3.29}$$

For the current case, this equation is too difficult to solve analytically or numerically. A simpler formula that applies when the maximum likelihood variable is normally distributed, for example when the sample it is obtained from is large, is [Martin, 1971]:

$$\sigma^2_{\hat{\theta},\,ML} = \left[-\frac{\partial}{\partial\theta^2}\ln(L)\right]^{-1}\Bigg|_{\theta = \hat{\theta}_{ML}} \tag{3.30}$$

Assuming eqn. (3.30) is valid for BML, then,

$$\begin{gathered}\sigma^2_{\eta_{kl,\,BML}} = (\hat{\eta}^2_{kl,\,BML}(\Omega_k - \hat{\eta}_{kl,\,BML})^2(\Omega_l - \hat{\eta}_{kl,\,BML})^2) \div \\ [(N_{kl} + N_{lk})(\Omega_k - \hat{\eta}_{kl,\,BML})^2(\Omega_l - \hat{\eta}_{kl,\,BML})^2 + \\ (N_k - N_{kl})\hat{\eta}^2_{kl,\,BML}(\Omega_l - \hat{\eta}_{kl,\,BML})^2 + (N_l - N_{lk})\hat{\eta}^2_{kl,\,BML}(\Omega_k - \hat{\eta}_{kl,\,BML})^2]\end{gathered} \tag{3.31}$$

The validity of eqn. (3.31) will be assessed numerically later. It should be noted that when $F_{kl} = F_{lk} = 1$, then $\hat{\eta}_{kl,\,BML} = \Omega_k = \Omega_l$ and $\sigma^2_{\hat{\eta}_{kl,\,BML}} = 0$.

3.2 Method of Least-Squares Smoothing

3.2.1 Overview of the Method of Least-Squares Smoothing

While the estimators mentioned above enforce reciprocity, they also cause closure to be obeyed no longer. Due to the importance of enforcing closure as discussed in Section 1.8, the least-squares smoothing (LSS) method of Larson and Howell [1986] is used to re-establish closure while attempting to minimizing the changes in $\hat{\eta}$.

In LSS, the changes to the $\hat{\eta}$ are represented by the objective function, *H*, which is defined as:

$$H = \sum_{k=1}^{N_{surf}}\sum_{l=1}^{N_{surf}}\frac{(\hat{\eta}_{kl} - \ddot{\eta}_{kl})^2}{2w_{kl}} \tag{3.32}$$

where $\ddot{\eta}_{kl}$ are the smoothed values that obey closure (i.e. the “answers”) and w_{kl} are weights allowing different relative penalties to be assigned to the adjustment of each η. It should be noted that after applying reciprocity estimation, the N^2_{surf} values of η have been reduced to $N_{surf}(N_{surf} + 1)/2$ unique values of η, the $N_{surf}(N_{surf} - 1)/2$ estimated values of $\hat{\eta}_{kl}$ and the N_{surf} values of η_{kk} along the diagonal of the η matrix. To simplify the notation below, $\hat{\eta}_{kk}$ and η_{kk} are identical. Furthermore, it should be noted that due to the symmetry of the $\hat{\eta}_{kl}$ and the $\ddot{\eta}_{kl}$, the w_{kl} must also be symmetric.

To apply the constraint of closure, the method of Lagrange multipliers is used with the Lagrangian, L, defined as:

$$L = H + \sum_{k=1}^{N_{surf}} \lambda_k (g_k + g_k^*) \tag{3.33}$$

where λ_k are the Lagrange multipliers. The row-sum constraints and the column-sum constraints, g_k and g_k^*, are defined as:

$$g_k = \Omega_k - \sum_{l=1}^{N_{surf}} \ddot{\eta}_{kl} \qquad g_k^* = \Omega_k - \sum_{l=1}^{N_{surf}} \ddot{\eta}_{lk} \tag{3.34}$$

While the Lagrangian has $2N_{surf}$ constraints, due to symmetry, there are only N_{surf} independent constraints. Therefore, there are N_{surf} Lagrange multipliers. The values of $\ddot{\eta}_{kl}$ which minimize the Lagrangian are found by differentiating L with respect to $\ddot{\eta}_{kl}$ and equating the derivative to zero:

$$\frac{\partial L}{\partial \ddot{\eta}_{kl}} = \frac{(\hat{\eta}_{kl} - \ddot{\eta}_{kl})}{w_{kl}} - \lambda_k - \lambda_l = 0 \tag{3.35}$$

Solving for $\ddot{\eta}_{kl}$:

$$\ddot{\eta}_{kl} = \hat{\eta}_{kl} + w_{kl}(\lambda_k + \lambda_l) \tag{3.36}$$

If the $\ddot{\eta}_{kl}$ from eqn. (3.36) are substituted into the closure equation, eqn. (1.9), and a bit of rearranging is done, then:

$$\Omega_k - \sum_{l=1}^{N_{surf}} \hat{\eta}_{kl} = \lambda_k \sum_{l=1}^{N_{surf}} w_{kl} + \sum_{l=1}^{N_{surf}} \lambda_l w_{kl} \tag{3.37}$$

Eqn. (3.37) defines a system of N_{surf} equations that may be represented as:

$$[\boldsymbol{U}]\vec{\lambda} = \vec{\beta} \tag{3.38}$$

where:

$$U_{kl} = w_{kl} \qquad k \neq l \tag{3.39}$$

$$U_{kk} = w_{kk} + \sum_{l=1}^{N_{surf}} w_{kl} \tag{3.40}$$

$$\beta_k = \Omega_k - \sum_{l=1}^{N_{surf}} \hat{\eta}_{kl} \tag{3.41}$$

After solving the system of equations, eqn. (3.38), for the Lagrange multipliers, $\vec{\lambda}$, the smoothed η that obey closure, $\ddot{\eta}_{kl}$, can be obtained using eqn. (3.36).

The N_{surf} system of linear equations used in smoothing, eqn. (3.38), is solved iteratively due to the potentially large number of equations. The method used in this work is the Successive Over-Relaxation (SOR) method [Burden and Faires, 1993]. Successive iterations are repeated until a convergence criterion is met, in this case when the maximum change between iterations in λ over all values drops below a certain threshold. Past experience with SMOOTH [Dolaghan et al., 1992], indicates that an appropriate value for this threshold is 1×10^{-5}. The only calculable measure of smoothing convergence appears to be g, the row-sum constraint, or g^*, the column-sum constraint. For a convergence criterion of 1×10^{-5}, the g for each row divided by the Ω for that row (surface) is always less than 3×10^{-8} for the geometries tested, suggesting that 1×10^{-5} is adequate convergence criterion. Indeed, changing the threshold yields no significant effect on the smoothed results.

3.2.2 Choice of Smoothing Weights

In this section, choices for the smoothing weights, w_{kl}, are discussed. Due to the quadratic form of H, w_{kl} is usually a squared quantity. Two values for the weights, $\hat{\eta}_{kl}^2$ and $\sigma_{\hat{\eta}_{kl}}^2$, are investigated in this work.

An *ad hoc* assumption used previously [Larson and Howell, 1986; Burns et al., 1992; Dolaghan et al., 1992; Loehrke et al., 1995; Chen et al., 2000; Liu et al., 2001] is that the error in each η is proportional to size of the η itself. Due to the form of the objective function, H, in eqn. (3.32), this assumption suggests that the weight, w_{kl}, should be proportional to the square of the estimated η, $\hat{\eta}_{kl}^2$, leading to what is termed "η^2 smoothing" in this work.

A potentially better approach is to use statistical theory to formulate the weights. Indeed, an advantage of using statistics is that a general formula is known for the uncertainty that can be used to formulate the weights. As discussed in Section 3.1.1, a common measure of the uncertainty of a result in statistics is the confidence interval which is a function of the standard deviation, σ. This suggests a better value for ω_{kl} may be the variance of the result, $\sigma_{\hat{\eta}_{kl}}^2$, which is termed "$\sigma^2$ smoothing" in this work. Taylor and Luck [1995] point out that σ^2 is a good weight for Monte Carlo results and use only that weight in their work on LSS and its extensions. Vercammen and Froment [1980] use the binomial variance as the weight in their method equivalent to LSS, although their F's are obtained by Monte Carlo integration for which the use of that variance is questionable.

3.3 Total Exchange Area Matrix Maximum Likelihood Estimation

A statistical method that employs all of the constraints at once may result in better results than the RES method. One idea is to obtain a maximum likelihood estimate of the η matrix, $\underset{\sim}{\eta}$, while incorporating all the constraints which is called total exchange area matrix maximum likelihood estimation in this work. To illustrate the concept of this estimation technique, an example is given below. Although it is impossible for a three-dimensional geometry to have less than four planar surfaces, a three-surface geometry is used in the example to illustrate the method. Applying all constraints, the $\underset{\sim}{\eta}$ matrix is:

$$\begin{bmatrix} \Omega_1 - \underset{\sim}{\eta}_{12} - \underset{\sim}{\eta}_{13} & \underset{\sim}{\eta}_{12} & \underset{\sim}{\eta}_{13} \\ \underset{\sim}{\eta}_{12} & \Omega_2 - \underset{\sim}{\eta}_{12} - \underset{\sim}{\eta}_{23} & \underset{\sim}{\eta}_{23} \\ \underset{\sim}{\eta}_{13} & \underset{\sim}{\eta}_{23} & \Omega_3 - \underset{\sim}{\eta}_{13} - \underset{\sim}{\eta}_{23} \end{bmatrix} \tag{3.42}$$

From the probability mass function of a multinomial variable [Martin, 1971; Mood et al., 1974; Ross, 1988],the probability of an $\underset{\sim}{\eta}$ matrix obtained from a specific Monte Carlo run, $p_{\eta M}$, is:

$$p_{\eta M} = \frac{\prod_{k=1}^{N_{surf}} N_k!}{\prod_{k=1}^{N_{surf}} \prod_{l=1}^{N_{surf}} N_{kl}!} \prod_{k=1}^{N_{surf}} \prod_{l=1}^{N_{surf}} \frac{\underset{\sim}{\eta}_{kl}^{N_{kl}}}{\Omega_k} = \tag{3.43}$$

$$\frac{N_1!N_2!N_3!}{N_{11}!N_{12}!N_{13}!N_{21}!N_{22}!N_{23}!N_{31}!N_{32}!N_{33}!}\left(\frac{1}{\Omega_1\Omega_2\Omega_3}\right)^3 (\Omega_1 - \underset{\sim}{\eta}_{12} - \underset{\sim}{\eta}_{13})^{N_{11}} \underset{\sim}{\eta}_{12}^{N_{12}} \underset{\sim}{\eta}_{13}^{N_{13}} X$$

$$\underset{\sim}{\eta}_{12}^{N_{21}} (\Omega_2 - \underset{\sim}{\eta}_{12} - \underset{\sim}{\eta}_{23})^{N_{22}} \underset{\sim}{\eta}_{23}^{N_{23}} (\underset{\sim}{\eta}_{13})^{N_{31}} (\underset{\sim}{\eta}_{23})^{N_{32}} (\Omega_3 - \underset{\sim}{\eta}_{13} - \underset{\sim}{\eta}_{23})^{N_{33}}$$

or:

$$p_{\eta M} = \Psi(\Omega_1 - \underset{\sim}{\eta}_{12} - \underset{\sim}{\eta}_{13})^{N_{11}}(\Omega_2 - \underset{\sim}{\eta}_{12} - \underset{\sim}{\eta}_{23})^{N_{22}}(\Omega_3 - \underset{\sim}{\eta}_{13} - \underset{\sim}{\eta}_{23})^{N_{33}} X \\ \underset{\sim}{\eta}_{12}^{(N_{12} + N_{21})}\,\underset{\sim}{\eta}_{13}^{(N_{13} + N_{31})}\,\underset{\sim}{\eta}_{23}^{(N_{23} + N_{32})} \tag{3.44}$$

where the N's are the photon counts obtained from the Monte Carlo run and Ψ, the $\underset{\sim}{\eta}$ matrix probability constant, is:

$$\Psi = \frac{\prod_{k=1}^{N_{surf}} N_k!}{\prod_{k=1}^{N_{surf}}\prod_{l=1}^{N_{surf}} N_{kl}!} \prod_{k=1}^{N_{surf}} \left(\frac{1}{\Omega_k}\right)^{N_{surf}} = \\ \frac{N_1!N_2!N_3!}{N_{11}!N_{12}!N_{13}!N_{21}!N_{22}!N_{23}!N_{31}!N_{32}!N_{33}!}\left(\frac{1}{\Omega_1\Omega_2\Omega_3}\right)^3 \tag{3.45}$$

Note that although η_{kk} is not part of eqn. (3.44), N_{kk} is.

Now as eqn. (3.42) shows, there are only three independent full matrix maximum likelihood estimates: $\underset{\sim}{\eta}_{12}$, $\underset{\sim}{\eta}_{13}$, and $\underset{\sim}{\eta}_{23}$. As discussed in Section 3.1.5, the maximum likelihood estimate for $p_{\eta M}$ is determined by finding the values of $\underset{\sim}{\eta}$ that maximize $p_{\eta M}$. The maximum value of $p_{\eta M}$ can be found by minimizing $ln(p_{\eta M})$.

Several methods were attempted to solve for the maximum likelihood values, particularly using methods of maximizing functions. It is found that the maximum likelihood values do not differ appreciably from results of the best RES method. While the LSS method used in RES requires finding N_{surf} unknowns, the η matrix maximum likelihood estimation requires finding $N_{surf}(N_{surf} - 1)/2$ unknowns or $O(N_{surf}^2)$ unknowns. Therefore, the computing time and memory required for the maximum likelihood results is much greater than that for the RES method. In fact, the memory requirements are often too large to be feasible. For these reasons, the maximum likelihood η matrix estimate was found not to be practical and the RES method is the best practical rectification scheme.

3.4 Numerical Approach

This work is the first in-depth study of RES methods on large, “real-world” geometries. Increases in computational power allow enough runs to be done on these geometries to show a number of statistical effects of RES for the first time. In this section, the numerical procedures used to explore RES are described. Exposition begins with a description of the two geometries used. Next, efforts to ensure the accuracy of the Monte Carlo results are discussed. This is followed by an enumeration of the ensembles of runs done for these two geometries.The final two sections describe the error measurements used in this work including error metrics.

3.4.1 Test Geometries

The majority of runs done in this work are for the 145 surface geometry representing the Electron Test Facility (ETF) from the LLNL Laser Isotope Separation Project which is shown in side elevation view in Fig. 3.2. This geometry, having only a modest number of surfaces, allows many runs to be accomplished in a reasonable time within the limits of available computer resources. The ETF geometry is symmetric about the Y-axis. A plane of symmetry, modeled by a perfectly reflecting specular surface, is used to reduce in half the amount of Monte Carlo photon tracing required. The rectangular symmetry plane is shown at the back of Fig. 3.2. For the tests done, all surfaces except the symmetry plane are modeled as a mixture of 25% diffusely reflecting and 25% specularly reflecting. The symmetry plane is modeled as a non-emitting, perfectly reflecting, specular surface which reduces the number of emitting surfaces in the geometry to 144.

As stated previously, the uncertainty in the exchange fraction depends, in part, upon the area-emissivity product, Ω, which is a surface property. For the ETF geometry, the 144 emitting surfaces have a large range of area values which, since all surfaces have the

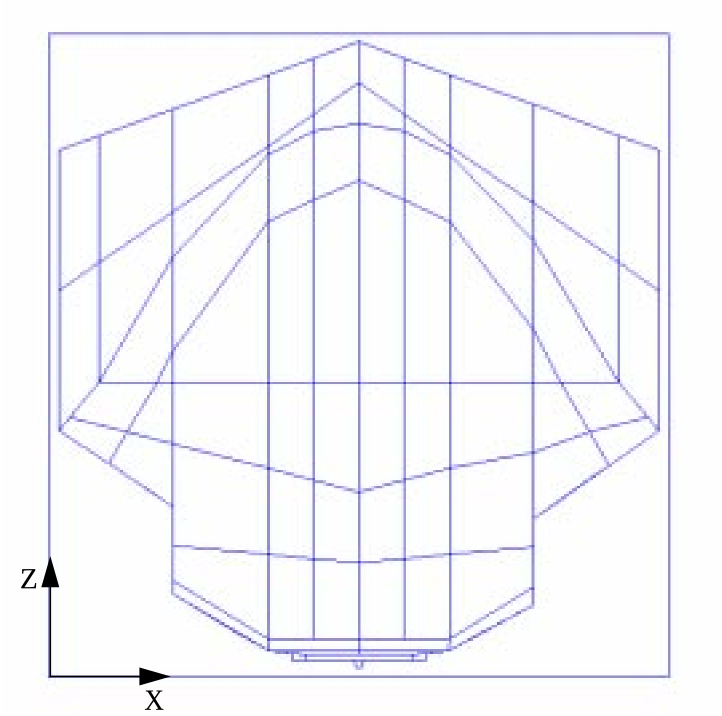


Figure 3.2 View of the ETF Geometry

same emissivity, is equivalent to a wide range of Ω as shown in Fig. 3.3. The abscissa in Fig. 3.3 measures the surfaces in Ω*, the dimensionless area-emissivity product, defined as the Ω for the surface divided by the minimum Ω for the geometry. While the minimum value of Ω* is always one, its maximum is determined by how the geometry is discretized, here, Ω^*_{max} = 466 for the ETF geometry.

The second geometry used in this work is the 1,182 surface Cham geometry that is discussed in Section 2.7.1. The properties used in this chapter are the same as those used originally to model the chamber, some surfaces are 25% diffusely reflecting while others are 95% diffusely reflecting. In addition to this geometry being much more sophisticated than the ETF geometry, its surfaces also exhibit a greater range in Ω* as shown in Fig. 3.4. The maximum value of Ω* for the Cham geometry is 81,200. Large values such as this

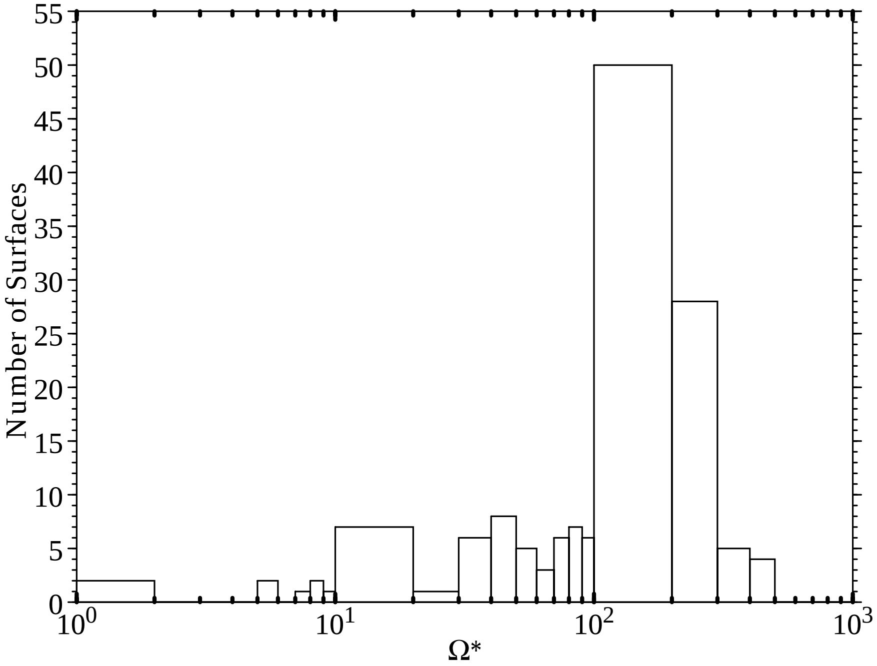


Figure 3.3 Distribution of Ω* for the ETF Geometry

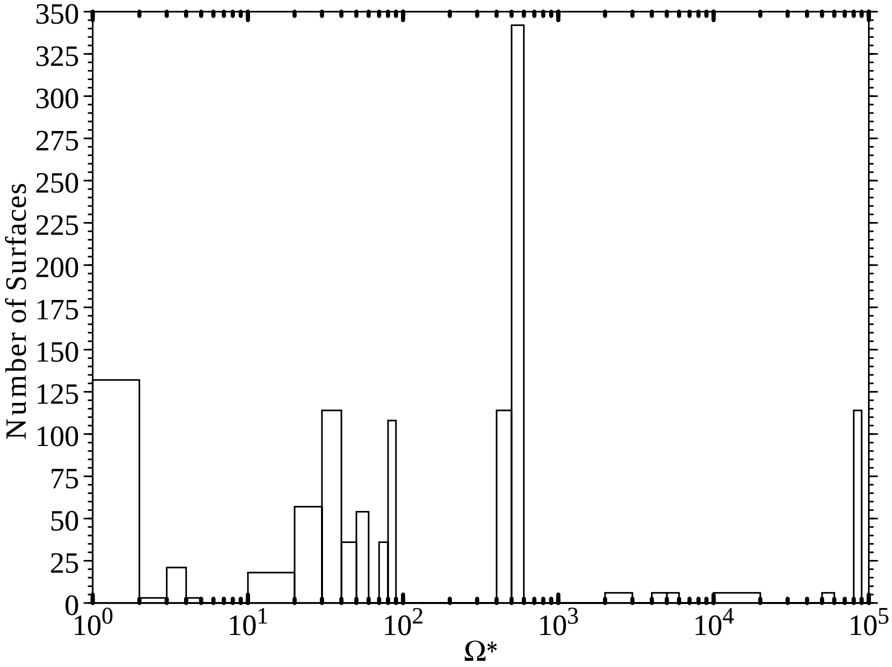


Figure 3.4 Distribution of Ω* for the Cham Geometry

appear to be common; for the geometries in Chapter 2, the maximum Ω^* values are 330,000 for the Gun geometry and 27,900 for the Amp geometries.

3.4.2 Ensuring the Accuracy of the Monte Carlo Simulation

Several measures are taken to ensure that the numerics do not affect the Monte Carlo simulation results. Double precision is used for all calculations and a particularly robust random number generator, the CCS addition lagged-Fibonacci pseudo-random number generator, described in Appendix B, is used. Each run uses a different random number seed derived from the time. Also, all quadrilateral surfaces are split into two triangles because non-planar surfaces can lead to small errors in the intersection calculations. Random surface emission is used in all Monte Carlo runs because, as discussed in Appendix A, unlike the fixed point emission routine used in MONT3D [Maltby, 1987; Zeeb et al., 1999], random emission causes no bias in reciprocity. Furthermore, since material properties are constant as a function of angle, the angular distributions for emission and diffuse reflection (re-emission) are calculated directly using the relationship:

$$\theta = \sin^{-1}(\sqrt{R}) \tag{3.46}$$

where R is a uniformly distributed random number between zero and one. MONT3D [Maltby, 1987; Zeeb et al., 1999] and LSMONTE [Zeeb and Burns, 2000] obtain θ values by tabular interpolation which has exhibited small reciprocity errors in off-line simulations for simple geometries.

3.4.3 Ensembles of Runs

It is only from the distribution of results obtained from performing a sufficient number of runs to “fill out” the distribution, that definitive conclusions can be made. Therefore, three ensembles of runs, described in Table 3.1, are done in this work to obtain these dis-

Table 3.1: Ensembles of Runs

Ensemble	Geometry	Number of Photons Emitted	Number of Runs
A	ETF	10,000	10,000
B	ETF	100,000	1,000
C	Cham	100,000	100

tributions. The ensembles are done varying the number of photon emissions, and for both geometries, ETF and Cham, to elucidate both convergence due to increasing photon emissions and geometrical effects.

Two ensembles of runs, A and B, are done for the ETF geometry with 10,000 and 100,000 photons emitted per surface. As shown in Table 3.1, for ensemble A at 10,000 photon emissions per surface, 10,000 runs are done, and for ensemble B at 100,000 photon emissions per surface, 1,000 runs are done. Experience has shown for the ETF geometry, that runs with 10,000 photons per surface yield reasonably converged results, and runs with 100,000 photons per surface are “converged” to engineering precision. Comparing results from the two ensembles of runs allows the nature of convergence to be observed.

For the Cham geometry, resource limitations prevented the same degree of scrutiny. Indeed, ensemble C of only 100 runs with 100,000 photons per surface is done for the Cham geometry. Still, the ensemble is invaluable for allowing the geometrical effects of a large range in Ω^* to be assessed.

3.4.4 “Ground Truth” Results and Individual Error Measurement

In this work, “ground truth” results are required for the calculation of “errors.” “Ground truth” results are obtained by emitting, for a single run, so many photons that, compared to the other runs in this work, the variance in the answers (exchange fractions) is so small that the results are considered “exact.” For the ETF geometry, 300 million photons per surface are emitted for the “ground truth” results. Due to limited computational resources, for the Cham geometry, only 15 million photons per surface are emitted for the "ground truth" results. The uncertainty of the "ground truth" results for the Cham geometry is less than 10% of that of the 100,000 photon per surface results. It should be noted that the "ground truth" results for the Cham geometry include 10 million photons per surface obtained by coalescing the runs from ensemble C.

In this work, three types of error are defined. The two most basic are E_η, the total exchange area error, and E_F, the exchange fraction error:

$$E_{\eta_{kl}} = \eta_{kl} - \eta_{kl}^{\infty} \tag{3.47}$$

$$E_{F_{kl}} = F_{kl} - F_{kl}^{\infty} \tag{3.48}$$

The quantities above denoted by the ∞ superscript are the "ground truth" results.

Another way to express E_F is:

$$E_{F_{kl}} = (F_{kl} - \bar{F}_{kl}) - (\bar{F}_{kl} - F_{kl}^{\infty}) \tag{3.49}$$

where F_{kl} is current estimate of the result and $\bar{F}_{kl}$ is the mean value of that current estimate. In the equation above, the first term measures the deviation of the estimate about its mean (in aggregate, over many runs, this yields the uncertainty) and the second term is a measure of the bias of the estimate, i.e. how far the mean value of the estimate is from the "ground truth" value. A desirable trait for an estimator is for it to be unbiased. To measure bias, this work defines the individual bias error, E_B, by calculating the mean value over N_{run} runs:

$$E_{B,F_{kl}} = \frac{1}{N_{runs}} \sum_{i=1}^{N_{runs}} F_{kl,i} - F_{kl}^{\infty} = \bar{F}_{kl} - F_{kl}^{\infty} \tag{3.50}$$

Using a value of 100 for N_{runs} has been observed empirically to yield acceptable fidelity for both the bias error and the uncertainty or variance.

3.4.5 Measuring Multidimensional Error, the l_2*-Norm*

While this work explores quantifying the error and uncertainty in the individual η's, global measurements of error are also useful. One quantity that can be used for measuring the error for the matrix as a whole or for an individual surface is the l_2-norm also known as the Euclidean norm or Euclidean length [Anton, 1987; Kreyszig, 1993]. In general, M different quantities can be plotted in M-dimensional space, $\mathbf{R}^M$, as a vector, **s**. The l_2-norm of **s**, $\|\boldsymbol{s}\|_2$, is defined as:

$$\|\boldsymbol{s}\|_2 = \sqrt{\sum_{i=1}^{M} s_i^2} \tag{3.51}$$

Two different error l_2-norms, normalized by the number of elements in each, are used in the work. The first, the normalized matrix error, Δ, is defined as:

$$\Delta = \sqrt{\frac{1}{N_{surf}^2} \sum_{k=1}^{N_{surf}} \sum_{l=1}^{N_{surf}} E_{kl}^2} \tag{3.52}$$

This “matrix error” is a global measure of error over all elements and will be used in a relative evaluation of the RES methods.

It is also helpful to consider error metrics of finer granularity. Therefore, the normalized surface error, Δ^k is also defined:

$$\Delta^k = \sqrt{\frac{1}{N_{surf}} \sum_{l=1}^{N_{surf}} E_{kl}^2} \tag{3.53}$$

Δ^k will be used subsequently to demonstrate several effects of the recommended RES method.

3.5 Results and Discussion

In this section, all three granularities of error are used. In the first subsection, the RES methods are assessed relative to one another by examining the matrix errors. In the next subsection, the surface errors are presented. The third subsection presents an analysis at the local level of the distribution of individual RES smoothed answers. The fourth subsection discusses the CPU and memory requirements of the RES method.The final subsection discusses concepts useful for the implementation of a Monte Carlo convergence criterion that takes into account the effects of the RES method.

3.5.1 Determining the Best RES Method

3.5.1.1 The Total Exchange Area Error Matrix

Since the purpose of the RES method is to improve the estimates of η, an obvious first choice to measure the effectiveness of the method is the E_η matrix error (i.e. the global uncertainty.) The mean values of Δ_η, averaged over all runs in an ensemble, are presented in Tables 3.2 to 3.4. The tables include: 1) the original Monte Carlo results, 2) the results after reciprocity estimation but before smoothing (labelled “no smoothing”), and 3) the results after smoothing using both η^2 and σ^2 weighting. The most striking observation from the tables below is that, for a given ensemble, there is little difference between the Δ_η’s for different RES methods, including the results that are not smoothed! While the BML / σ^2 method yields slightly better results than all the other methods, no results differ

Table 3.2: Mean Δ_η's for Ensemble A

ETF Geometry; 10,000 Photons per Surface					
Estimate		FCI	FV	MV	BML
Smoothing	None	2.11x10^{-2}	2.09x10^{-2}	2.10x10^{-2}	2.08x10^{-2}
	η^2	2.13x10^{-2}	2.13x10^{-2}	2.18x10^{-2}	2.09x10^{-2}
	σ^2	2.08x10^{-2}	2.06x10^{-2}	2.08x10^{-2}	2.06x10^{-2}
Original Monte Carlo Results					3.12x10^{-2}

Table 3.3: Mean Δ_η's for Ensemble B

ETF Geometry; 100,000 Photons per Surface					
Estimate		FCI	FV	MV	BML
Smoothing	None	6.67x10^{-3}	6.58x10^{-3}	6.58x10^{-3}	6.58x10^{-3}
	η^2	6.71x10^{-3}	6.62x10^{-3}	6.63x10^{-3}	6.61x10^{-3}
	σ^2	6.59x10^{-3}	6.51x10^{-3}	6.51x10^{-3}	6.51x10^{-3}
Original Monte Carlo Results					9.88x10^{-3}

Table 3.4: Mean Δ_η's for Ensemble C

Cham Geometry; 100,000 Photons per Surface					
Estimate		FCI	FV	MV	BML
Smoothing	None	3.75x10^{-5}	3.75x10^{-5}	3.76x10^{-5}	3.74x10^{-5}
	η^2	3.74x10^{-5}	3.73x10^{-5}	3.84x10^{-5}	3.73x10^{-5}
	σ^2	3.73x10^{-5}	3.73x10^{-5}	3.76x10^{-5}	3.73x10^{-5}
Original Monte Carlo Results					5.32x10^{-5}

from the BML / σ^2 results by more than 3.4% except for the MV / η^2 method for the ETF geometry with 10,000 photons per surface which differs only by 6%. Clearly if there are significant differences among the RES methods, it is not apparent in Δ_η. It should be noted that Δ_η emphasizes the largest errors for the surfaces with the largest Ω_k values.

Relative to their original Monte Carlo results, the improvement in Δ_η for the three ensembles ranges from 28% to 34%. This, on average, is a little better than the 29%

improvement that would occur from doubling the emissions for the original Monte Carlo results. Thus, the RES method appears, in general, to be effective in using the “redundant” information in the E_η matrix.

3.5.1.2 The Exchange Fraction Error Matrix

As discussed in Section 1.2, the radiative heat transfer rate depends not only on the values of η but also on T^4. Thus, terms with small values of η can be more important in determining the radiative transport than terms with large values of η. In fact, experience indicates that smaller surfaces are often used to capture greater spatial variation where higher temperatures exist, so this is often the case. Therefore, another measure of error that may be pertinent is Δ_F. Indeed, Δ_F is a most stringent measure of error because it emphasizes the largest F deviations in the matrix no matter which surface they represent.

Tables 3.5 to 3.7 present Δ_F results corresponding to the Δ_η results in Tables 3.2 to 3.4. The Δ_F values exhibit greater differentiation among the different RES methods than do the Δ_η values. Comparing the various RES methods, those utilizing MV estimation are the worst. In the best case, for ensemble B, the Δ_F’s for MV estimation are less (albeit slightly) than that for the original Monte Carlo results. However, the opposite is true for the other two ensembles.The MV Δ_F’s are particularly bad for the Cham ensemble, which includes a Δ_F over 90 times worse than that for the original Monte Carlo results. The reason for this is a peculiarity of the MV estimation technique that was noted before. If one of the η’s in a reciprocity pair is zero, then the MV estimate for the pair is also zero. From the reciprocity relation, it is easy to see how this might lead to large errors in a geometry such as Cham where there is a large Ω range in the geometry and not enough photons have been emitted, causing several of the η’s for the surfaces with large Ω values to be zero.

Table 3.5: Mean Δ_F's for Ensemble A

ETF Geometry; 10,000 Photons per Surface					
Estimate		FCI	FV	MV	BML
Smoothing	None	6.93x10^{-4}	6.53x10^{-4}	9.76x10^{-4}	6.45x10^{-4}
	η^2	8.03x10^{-4}	7.06x10^{-4}	22.2x10^{-4}	6.51x10^{-4}
	σ^2	6.72x10^{-4}	6.46x10^{-4}	14.5x10^{-4}	6.42x10^{-4}
Original Monte Carlo Results					8.24x10^{-4}

Table 3.6: Mean Δ_F's for Ensemble B

ETF Geometry; 100,000 Photons per Surface					
Estimate		FCI	FV	MV	BML
Smoothing	None	2.15x10^{-4}	2.04x10^{-4}	2.22x10^{-4}	2.04x10^{-4}
	η^2	2.23x10^{-4}	2.08x10^{-4}	2.34x10^{-4}	2.06x10^{-4}
	σ^2	2.13x10^{-4}	2.03x10^{-4}	2.22x10^{-4}	2.03x10^{-4}
Original Monte Carlo Results					2.61x10^{-4}

Table 3.7: Mean Δ_F's for Ensemble C

Cham Geometry; 100,000 Photons per Surface					
Estimate		FCI	FV	MV	BML
Smoothing	None	30.4x10^{-5}	9.41x10^{-5}	165x10^{-5}	8.76x10^{-5}
	η^2	28.4x10^{-5}	10.1x10^{-5}	851x10^{-5}	8.76x10^{-5}
	σ^2	14.0x10^{-5}	9.04x10^{-5}	655x10^{-5}	8.76x10^{-5}
Original Monte Carlo Results					9.05x10^{-5}

The RES methods that utilize FCI estimation, while better than those utilizing MV estimation, are also clearly not good. In the best cases, for the ETF ensembles, the Δ_F's for FCI estimation are smaller than those for the original Monte Carlo results. However, for the Cham geometry, the Δ_F's for FCI estimation are all more than 50% greater than for the original Monte Carlo results. The results for FV estimation are, in all cases, better than those for FCI estimation, indicating that the variance is a much better weight for estimation than the standard deviation.

The BML / σ^2 RES method produces Δ_F values that are lower than or equal to those for all other methods. Still, the differences compared to the Δ_F*'s* for BML / η^2 RES and FV / σ^2 RES are slight. While it is fairly conclusive that BML / σ^2 RES is the best method, further analysis is presented in the next section that verifies this conclusion.

The effects of LSS are observed to depend heavily on the reciprocity estimation method used. Still, the results indicate that σ^2 smoothing is superior to η^2 smoothing for all reciprocity estimation methods used. In addition, η^2 smoothing always increases Δ_F values compared to the unsmoothed results, often by significant amounts. On the other hand, σ^2 smoothing decreases Δ_F compared to the unsmoothed results in all cases except those involving MV estimation, a method not recommended herein. It should be noted that the decrease in the Δ_F caused by σ^2 smoothing is usually slight.

These results offer a possible explanation why Murty and Murty [1991] found that η^2 smoothing had little effect on their zone radiation calculations. Any benefits obtained from the enforcement of closure were probably offset by the increased error caused by η^2 smoothing.

In summary, the Δ_F is a very stringent measurement emphasizing the largest E_F values in the matrix. Not unexpectedly, the relative improvement in the Δ_F compared to the Δ_F of the original Monte Carlo results is less then the relative improvement found from Δ_η. The decrease in the Δ_F from that of the original Monte Carlo results is only about 22% at best for the ETF geometries and around 3% for the Cham geometry.

3.5.1.3 The Bias Error Matrix for the ETF Geometry

While an estimator with a small bias and small variance may be preferred to an unbiased estimator with a large variance, all else being equal, an unbiased estimator is pre-

Table 3.8: Mean Δ_B's for the ETF Geometry

		100 Values Derived from Ensemble A			
Estimate		FCI	FV	MV	BML
Smoothing	None	$10.5\text{x}10^{-5}$	$8.58\text{x}10^{-5}$	$51.5\text{x}10^{-5}$	$6.47\text{x}10^{-5}$
	η^2	$41.0\text{x}10^{-5}$	$28.5\text{x}10^{-5}$	$184\text{x}10^{-5}$	$6.53\text{x}10^{-5}$
	σ^2	$10.8\text{x}10^{-5}$	$9.62\text{x}10^{-5}$	$85.4\text{x}10^{-5}$	$6.44\text{x}10^{-5}$
Original Monte Carlo Results					$8.26\text{x}10^{-5}$

ferred. Furthermore, a large bias invalidates any benefits of a small variance estimator. For these reasons, the normalized matrix bias error, Δ_B, is examined.

Bias error results in this work are calculated only for ensemble A, the ETF geometry with 10,000 photons emitted per surface. The reason for this is as follows. To fill out the bias error distribution, each point in the distribution is derived from the mean over 100 runs, i.e. a subset of the ensemble A runs. For the other two ensembles, averaging over 100 runs yields bias error results that are statistically almost indistinguishable from the "ground truth" results.

The 10,000 runs in ensemble A yield a distribution of 100 Δ_B's. The mean values of Δ_B for these 100 subsets is presented in Table 3.8. The table clearly shows that all methods except those using the BML method increase the bias error, i.e. have Δ_B values greater than the Δ_B for the original Monte Carlo results. Indeed, for all methods except those based on BML estimation, the biases are large enough to negate any benefit of reduced variance relative to the original Monte Carlo results. In addition, the values in the table enforce the previous conclusion that the BML / σ^2 method is superior to the FV / σ^2 method.

It is noteworthy to consider the sources of the biases for the MV, FCI, and FV estimators. For the MV estimator, as discussed previously, when one of the η's used to calculate the estimate is zero, then the estimate is zero. This leads to frequent underestimation (i.e.

a negative bias) when there is significant probability for a nonzero η being estimated as zero (i.e. too few photons have been emitted per surface).

To understand the source of the biases for the FCI and FV estimators, further analysis is required. For this purpose, the percent closure error for a surface k, γ_k, is defined as:

$$\gamma_k = \frac{100}{\Omega_k}\left(\sum_{l=1}^{N_{surf}} \hat{\eta}_{kl} - \Omega_k\right)\% \tag{3.54}$$

It should be noted that γ only pertains to reciprocity estimated results. For the original Monte Carlo results, γ's are zero to machine precision. For smoothed results, γ's are zero to the level of convergence of the results. Figs. 3.5 and 3.6 show the γ's for FCI and FV estimation for twenty runs of the Cham geometry with 100,000 photons emitted per surface. For every surface, the range of γ's, from minimum to maximum, is shown as a vertical line. It should be noted that, frequently, more than one surface exists at the same value of Ω^*.

From these graphs and other results, three observations about the biases of the FCI and FV estimators are evident. From the graphs, it is obvious that the bias for both estimators is positive. From the graphs and Table 3.8, it can be determined that FCI estimation exhibits much greater bias than FV estimation. Furthermore, reviewing the data from the individual runs shows that FV estimation results in more surfaces exhibiting positive γ's per run than for FCI estimation.

The reason for these positive biases is that both estimators are based on δ, a monotonically decreasing function of F. Thus, the larger the F value of the original Monte Carlo results, the smaller the influence of the other Monte Carlo F on the final estimated value. Therefore, the more F is overestimated, the greater effect F has on the final estimated value. The graphs clearly show that the largest positive biases occur for the surfaces with

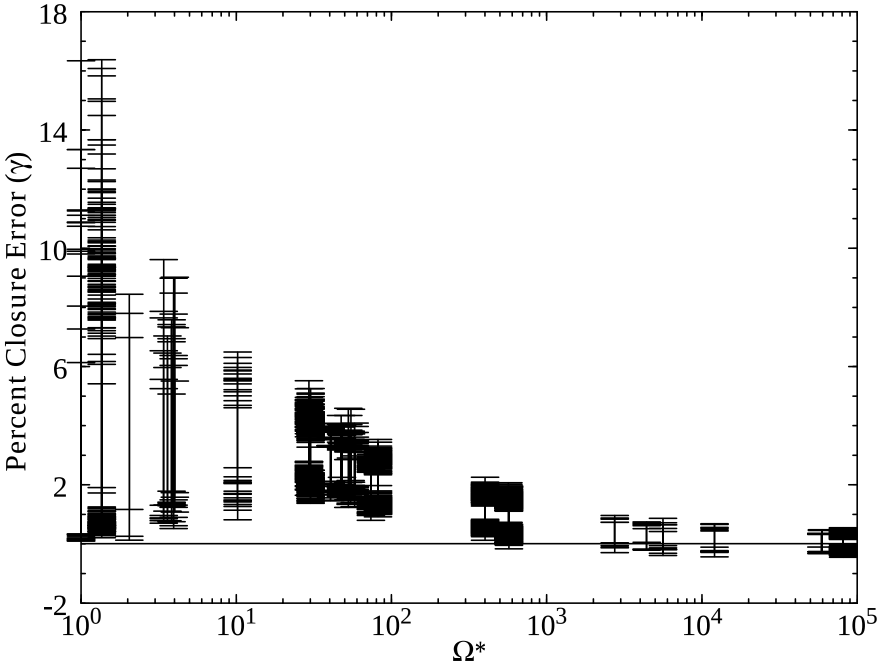


Figure 3.5 Percent Closure Errors for 20 Runs of the Cham Geometry, FCI Estimation

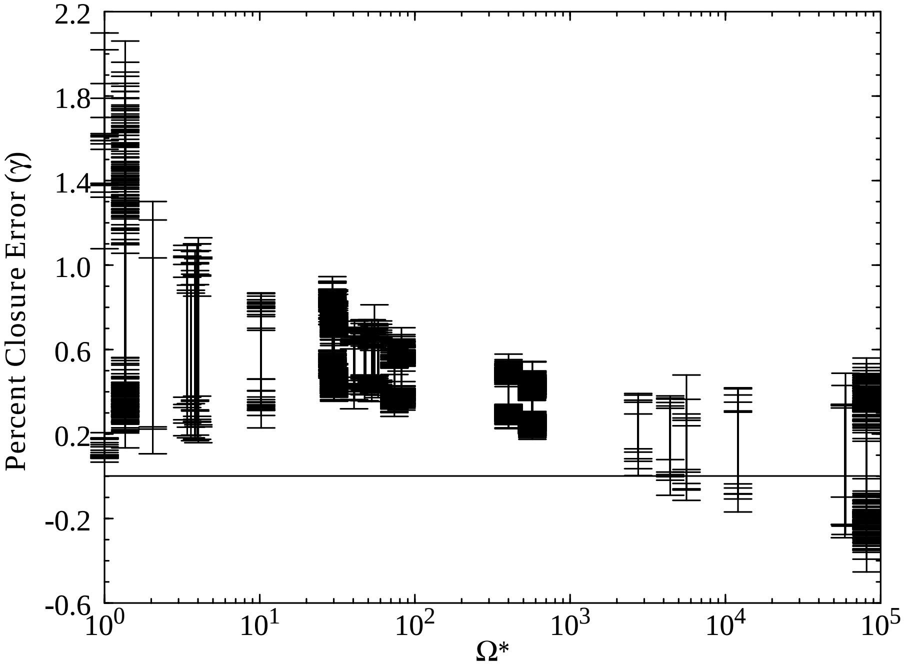


Figure 3.6 Percent Closure Errors for 20 Runs of the Cham Geometry, FV Estimation

the smallest Ω^* values. Therefore, it is the overestimation of the smaller F's in the reciprocity pair that cause the positive bias. FV estimation, which squares the δ's, reduces the effect of the smaller F in a reciprocity pair relative to the larger F compared to FCI estimation which leads to smaller biases overall.

Table 3.8 above shows that the effect on bias of smoothing varies significantly depending on the reciprocity estimator used. Still, one trend is clear. For all estimators except BML, smoothing increases bias. Later, it will be shown that all the results with BML estimation are unbiased to statistical accuracy. Therefore, it is concluded that if the reciprocity estimator is biased, then smoothing increases that bias, at least slightly. Furthermore, it can also be concluded that η^2 smoothing significantly increases the bias for a biased estimator.

While the Δ_B results clearly exhibit bias, or at least increased error in general for MV, FCI, and FV estimation, it is unclear from the results whether the BML estimator results and the original Monte Carlo results are biased. To assess this, Fig. 3.7 plots the mean Δ_F values and mean Δ_B values for the ETF geometry ensembles for the original Monte Carlo and BML / σ^2 RES results. Since E_B contains a mean value from 100 runs, Δ_B is plotted as emitting one million photons per surface. The bars on the results show the range of the Δ_F and Δ_B values over all runs.

The plot emphasizes two aspects of the convergence of Δ_F and Δ_B. First, while it is well known that the errors in the individual original Monte Carlo results decrease as the square root of the number of emissions, the graph shows that this is also true for Δ_F's and Δ_B's of the original Monte Carlo and the BML / σ^2 RES results. A second aspect is that the bias in the BML / σ^2 RES and original Monte Carlo results, if it exists, is negligible. This conclusion results from the fact that Δ_B lies, to within statistical error, on the same line as do the results for ensembles A and B (the results differ by only 0.3%.) Similar results are

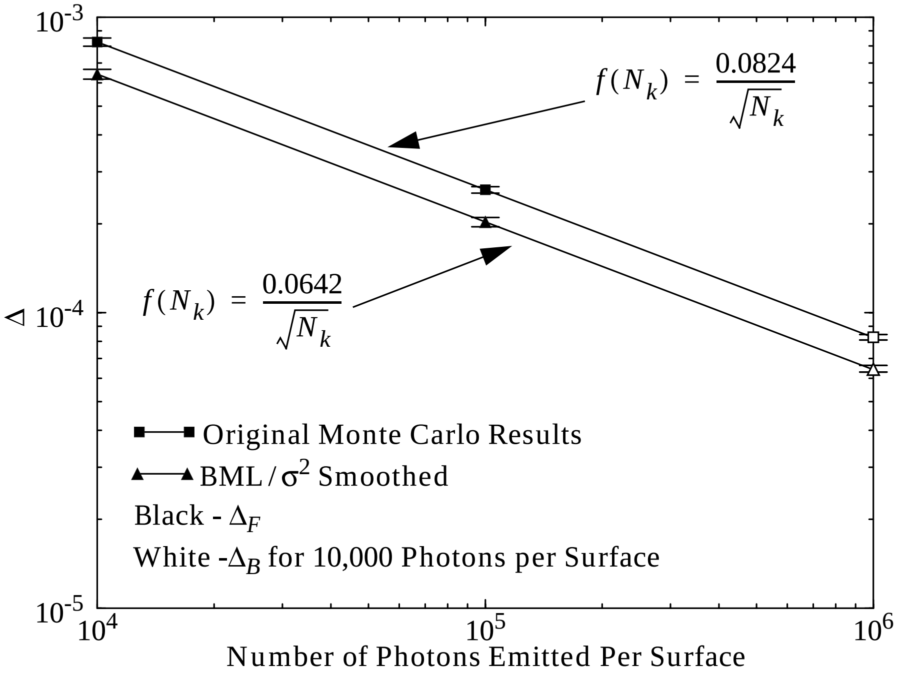


Figure 3.7 Mean Δ Values for the ETF Geometry

found for Δ_F and Δ_B values for the unsmoothed BML and the BML / η^2 RES results. From this, it can be concluded that all the differences in error for all the BML methods is caused by differences in the variances of the methods, not bias.

3.5.1.4 Summary of Results and Recommended RES Method

To determine the best RES method, Δ values for E_η, E_F, and E_B were calculated. Δ_F is found to be a much better indicator of difference between the RES methods than Δ_η. The E_F matrix shows that MV and FCI estimation often lead to larger errors than the other methods. In addition, all methods except those based on BML estimation exhibit significant bias error. While BML / η^2 is found to be almost as good as BML / σ^2, BML / σ^2 is found to give the best results in all cases and is the method of choice.

3.5.2 Surface Errors

As shown in Section 3.5.1.2, the improvement of the Δ_F values by BML / σ^2 smoothing over the original Monte Carlo results was rather slight for the Cham geometry. To understand why this is so, this section presents error at the next level of granularity, that of surfaces. The thrust of this section is the improvement in E_F for each surface caused by BML / σ^2 smoothing relative to the original Monte Carlo results. Therefore, the quantity studied in this section is the Δ_F^k ratio for a single run which is the ratio of Δ_F^k after BML / σ^2 smoothing divided by Δ_F^k for the original Monte Carlo results. Thus, values < 1 represent an improvement, while values > 1 indicate that error increases. The median and absolute ranges of Δ_F^k ratios for the ETF geometry and Cham geometry, for 100,000 photons per surface are shown in Figs. 3.8 and 3.9. The results shown are only for 100 runs of each case. This illustrates the variance of the Δ_F^k ratios without overemphasizing outlier values.

The most striking feature of Figs. 3.8 and 3.9 is that the improvement caused by BML / σ^2 RES is clearly an increasing function of Ω^*. The improvement ranges from negligible for the surfaces with the lowest Ω^* values to, at the highest values, 47% for the ETF geometry and 31% for the Cham geometry. This is to be expected as, from Section 3.1.2, the ratio of the variances of η's in a reciprocity pair is proportional to the ratios of their Ω^*'s, and greater weight is given to the reciprocity surface with the lower variance. Ergo, the larger surface is “corrected” the most, and exhibits the most error reduction. This explains why the improvement in Δ_F by BML / σ^2 smoothing for the Cham geometry is so slight. As shown by Fig. 3.4, most surfaces in the Cham geometry have small Ω^* values which are in the range where Fig. 3.9 shows minimal improvement.

Also apparent from the figures is that while error decreases for the majority of surfaces (i.e. a ratio < 1), error increases for some surfaces for some runs. Figs. 3.10 to 3.13 depict

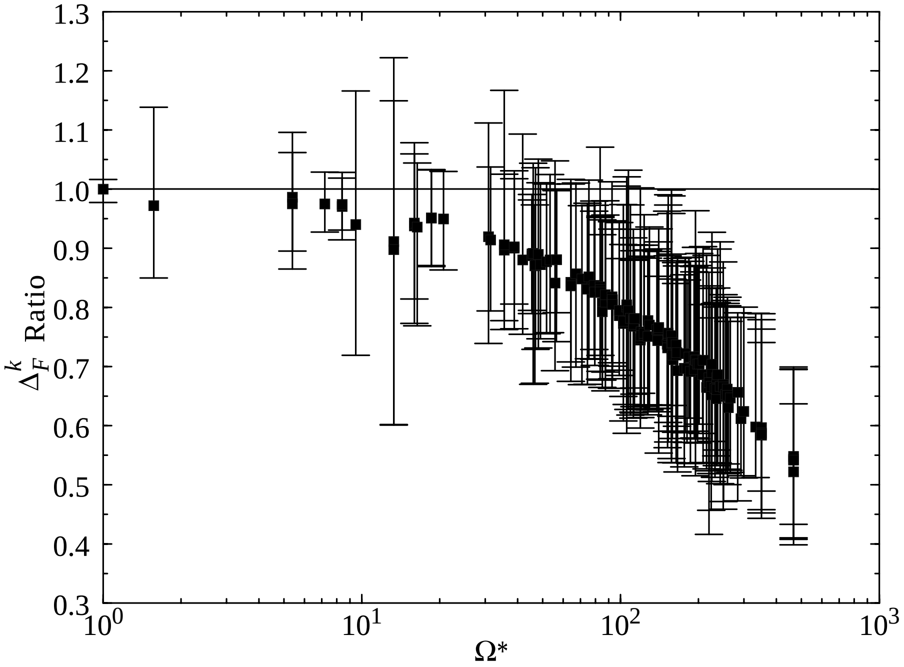


Figure 3.8 Δ_F^k Ratios for 100 Runs of the ETF Geometry

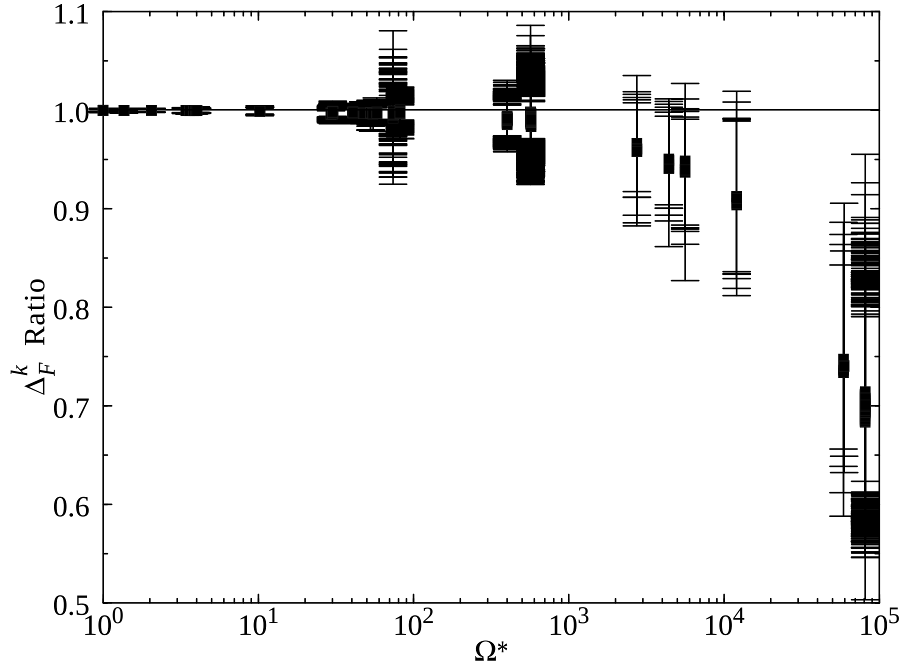


Figure 3.9 Δ_F^k Ratios for 100 Runs of the Cham Geometry

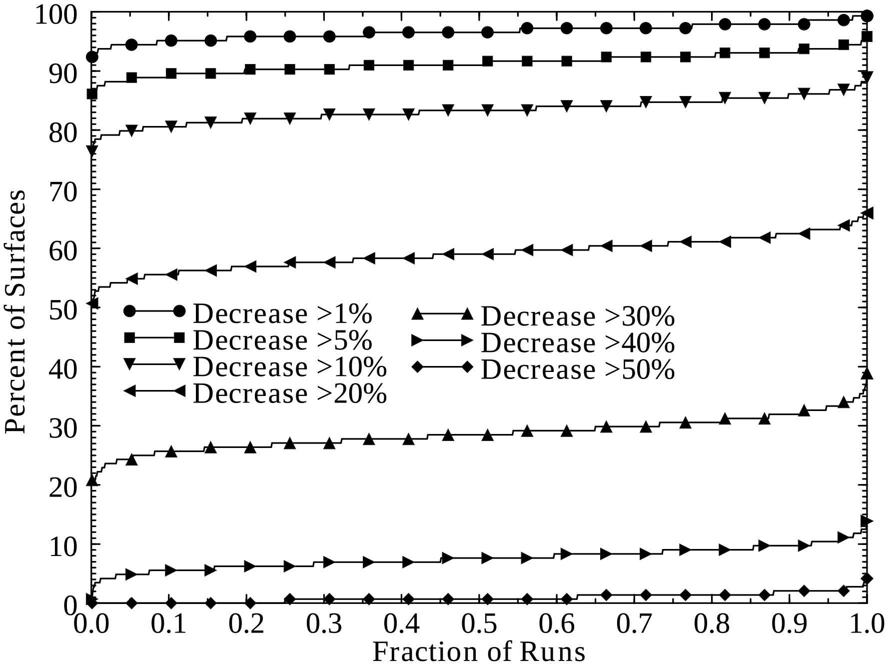


Figure 3.10 Decreases in Δ_F^k for the ETF Geometry with BML/σ^2 Smoothing

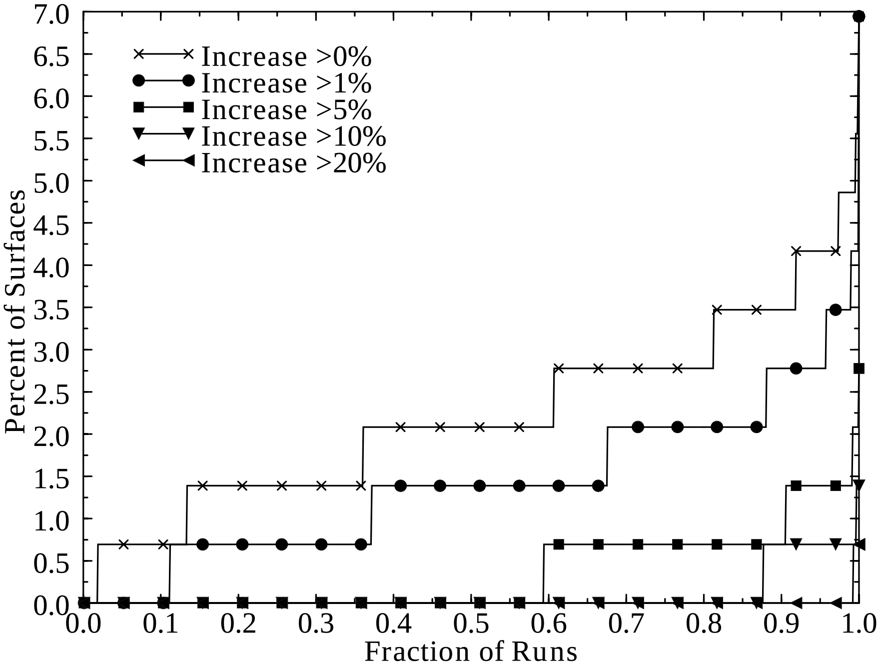


Figure 3.11 Increases in Δ_F^k for the ETF Geometry with BML/σ^2 Smoothing

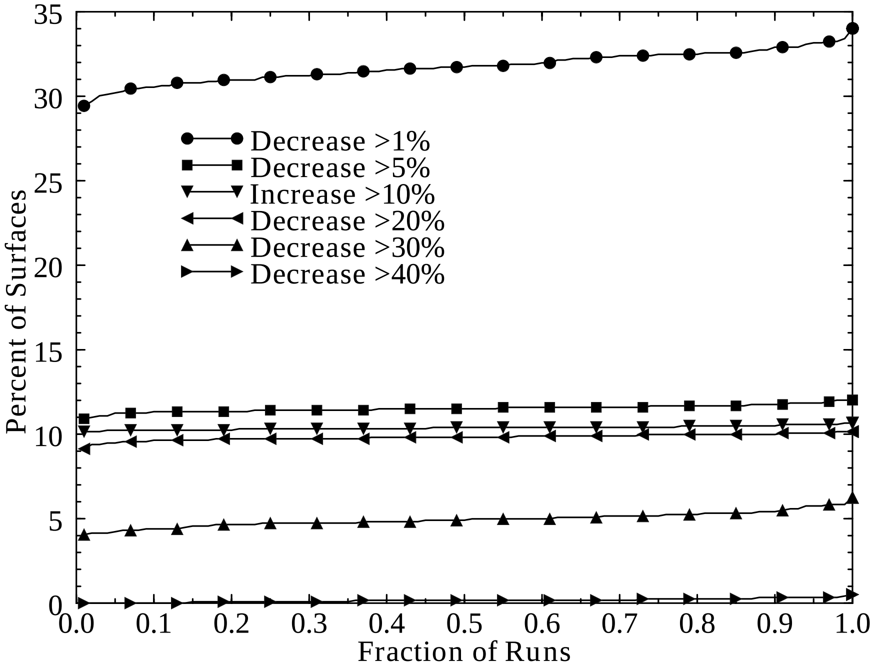


Figure 3.12 Decreases in Δ_F^k for the Cham Geometry with BML/σ^2 Smoothing

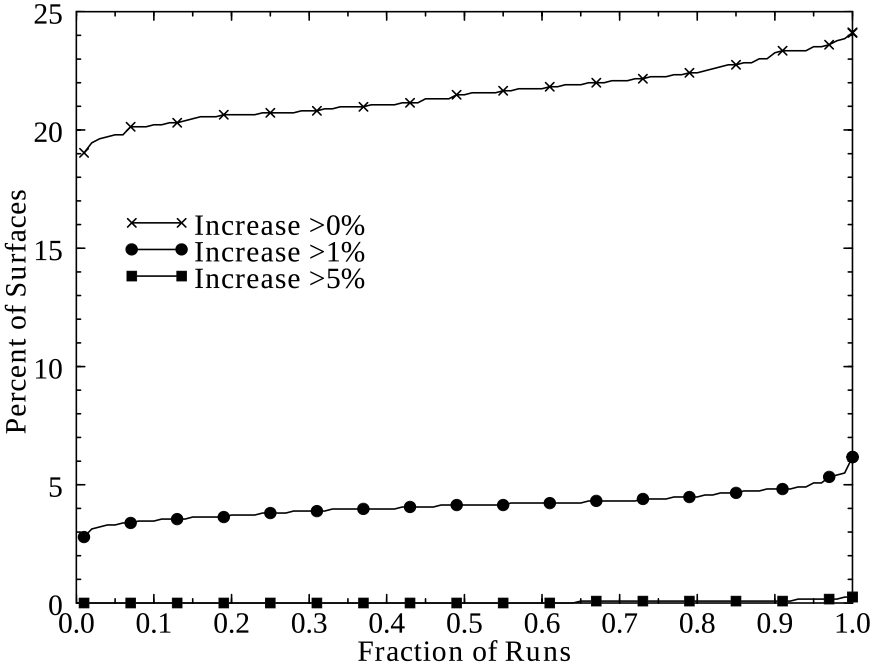


Figure 3.13 Increases in Δ_F^k for the Cham Geometry with BML/σ^2 Smoothing

the percent of surfaces that have their errors reduced or increased, with percent change as the parameter. For the ETF geometry, it is shown that few surfaces (always fewer than 7%) experience increases in error. While increases in error of over 20% are possible, the odds of this occurring are so small that it is hard to read it from the graph. Furthermore, around 80% of the surfaces have their results improved by 10% or more, over 50% have their results improved by 20% or more, and there are a significant number of surfaces with a 30% or greater improvement. The Cham geometry, on the other hand, while having up to 25% of its surfaces showing some degradation at times, usually has fewer then 5% degrade more than 1% and almost never has a surface degrade more than 5%. While fewer than 35% of the surfaces in the Cham geometry show even a 1% improvement, almost 10% of the surfaces have a 20% or greater improvement and around 5% show improvement of 30% or more.

Although the results above show that improvement of Monte Carlo results by BML/σ^2 smoothing is very geometry dependent and is probably less effective for geometries with large ranges in Ω^*, the other benefits of the method should not be overlooked. As mentioned in Section 1.8, there may be serious repercussions when reciprocity is not obeyed, particularly if the thermal balance code expects it to be. While surfaces with small values of Ω^* may not exhibit much improvement from BML/σ^2 smoothing, extremely large reciprocity errors between these surfaces and other surfaces with much larger Ω^* are avoided. Since the largest interactions for the surfaces with the smallest Ω^* values usually occur with the surfaces with the largest Ω^* values, the effect of this improvement can be very important.

3.5.3 Confidence Intervals for the Individual η

Due to the usefulness of confidence intervals as a measure of the error in individual η's, this section addresses deriving them for individual BML/σ^2 smoothed results from a single run. As discussed in Section 3.1.1, the uncertainty of the individual η can be expressed as confidence intervals if the η can be considered normally distributed. While the normality of the original Monte Carlo results is well known, the normality of the estimated and estimated / smoothed η is unproven. Therefore, the first subsection presents an analysis of the normality of the BML / σ^2 smoothed results. Even if the η are normally distributed, their standard deviation must be known to calculate the confidence interval. For this reason, the second subsection examines whether the formula given for $\sigma_{\eta_{kl},\,BML}$ in eqn. (3.31) is a suitable estimate of the BML / σ^2 smoothed σ.

3.5.3.1 Tests of Normality

Since the BML / σ^2 smoothed η give the best results, that case along with the original Monte Carlo results and the unsmoothed BML estimated η are tested. The results are summarized in this section, but more details, including many results, are provided in Appendix C.

When considering the original Monte Carlo results and other estimates based on them, there are two obvious deviations from the normal distribution to consider. First, while the normal distribution is continuous, the original Monte Carlo results and the reciprocity estimated results are discrete. While this discreteness is often overlooked, it can have an effect. In fact, it is found in this work that many times the same value is simulated for an unsmoothed result five times in a row. This can even happen when over 100 photons are absorbed by a surface. Still, it should be mentioned that the combination of results in reciprocity estimation and the changes caused by smoothing tend to moderate the effects of this discreteness.

Another difference is that while any estimate for an exchange fraction is bound by the values zero and one, the normal distribution ranges over all real values. While this means that no estimate of an exchange factor can be technically normal, in most cases the normal probability of an exchange fraction outside the range of zero to one is so small that the distribution is essentially normal.

The degree to which a distribution can be considered normal is arbitrary. Normality tests function by sampling a number of points in the distribution. The more points that are sampled, the greater chance that the behavior in the tails of the distribution or the effects of discrete values will be noticed and the distribution declared non-normal. For example, in this work, normality tests are done sampling 10 and 100 values. For the tests using 10 values, the original Monte Carlo results are found to be normal when somewhere around 20 to 100 photons are absorbed by a surface, while the BML/σ^2 smoothed results appear normal when 10 or more photons are absorbed per surface. On the other hand, the tests using 100 values to sample the distribution suggest 200 or more absorbed photons are required for the normality of the original Monte Carlo results while 50 or more absorbed photons are required for the normality of the BML/σ^2 smoothed results.

These tests indicate that if enough photons are absorbed then BML/σ^2 smoothed results can be considered normally distributed and that the number of photons required for normality of the BML/σ^2 smoothed results is less than the number required for the original Monte Carlo results. Furthermore, the numbers of photons that must be absorbed to achieve normality that are listed above are so low that it can be assumed that in most cases the results can be considered normally distributed.

3.5.3.2 Calculation of the Standard Deviation

In order to apply confidence intervals to the individual η, not only must the results be normally distributed but a reasonably accurate estimate of the standard deviation is required. The variance of an η estimate is easily calculated from multiple runs:

$$\sigma^2_{\eta_{kl}} = \frac{1}{N_{runs} - 1} \sum_{i=1}^{N_{runs}} (\eta_{kl,\,i} - \bar{\eta}_{kl})^2 \tag{3.55}$$

where $\bar{\eta}_{kl}$ is the mean value of η_{kl} over the $N_{runs.}$ Still, Monte Carlo work typically involves the results from only a single simulation, so a single run estimate of σ is desired. For the BML / σ^2 smoothed η, an obvious choice for the estimate of σ is the single run formula for σ of the unsmoothed BML estimated η, $\sigma_{\eta_{kl},\,BML}$, defined in eqn. (3.31).

Yet the validity of eqn. (3.31) for even unsmoothed BML estimated η has not been proven. While eqn. (3.31) assumes a normal distribution, it relies on only two samples. Still, it is found to be very accurate for the unsmoothed BML estimated η. For the 10,000 runs of the ETF geometry with 10,000 photons emitted per surface, i.e. ensemble A, the mean values of $\sigma_{\eta_{kl},\,BML}$ for all the individual η are found to differ from the σ's calculated from the 10,000 unsmoothed BML estimated runs using eqn. (3.55) by 3% at most which is roughly the uncertainty in the σ’s calculated from the 10,000 runs. Furthermore, for 100 or more photons absorbed per surface, most $\sigma_{\eta_{kl},\,BML}$ values usually differ from their mean value by 5% or less.

The next question is how good an estimate is the $\sigma_{\eta_{kl},\,BML}$ for the BML / σ^2 smoothed results; the answer is surprisingly good. For BML / σ^2 smoothing, the ratio of the mean value of $\sigma_{\eta_{kl},\,BML}$ to σ for the smoothed results calculated over all runs ranges from 0.97 to 1.09. Very few of the mean $\sigma_{\eta_{kl},\,BML}$’s differ very much from the σ’s calculated from the 10,000 runs for the smoothed results. Out of the 7,990 reciprocity pairs of η with both surfaces absorbing 10 or more photons, only 430 of the mean $\sigma_{\eta_{kl},\,BML}$’s are not within 1% of the σ’s calculated for the BML / σ^2 smoothed results. Furthermore, only seven of the mean $\sigma_{\eta_{kl},\,BML}$ ‘s differ by more than 5% from the σ’s for the smoothed results. All seven cases involve reciprocity pairs where one or both surfaces absorb 1,000 or more photons.

Therefore, the BML σ formula, eqn. (3.31), is valid for the BML / σ^2 smoothed η. It occasionally slightly overestimates σ, particularly when a large number of photons are absorbed by at least one of the two surfaces.

3.5.4 CPU and Memory Requirements

CPU requirements of the smoothing code are insignificant. Smoothing of the Cham geometries requires about 12 seconds per run on one processor of a 296 MHz ultraSPARC system. This includes a significant amount of overhead calculating many of the statistics used in this work. One MONT3D run for the Cham geometry with one million photons per surface takes over 14 hours on the same system.

The memory requirements of smoothing, on the order of N_{surf}^2, can be large. On the other hand, any thermal analysis code using the smoothed results will almost certainly require at least the same amount of memory. Furthermore, the memory requirements might be reduced by dividing the results of large geometries into non-interacting subsets. If this can not be done, the η matrix may be “blocked,” and successively accessed from the disk. This of course results in a significant penalty in disk usage and CPU time.

3.5.5 Convergence Criteria for Monte Carlo Simulations

From the Δ_F^k ratio results in Section 3.5.2, it is clear that the total emissions required for a Monte Carlo simulation can be reduced significantly if the effects of BML / σ^2 smoothing are taken into account. While deriving a method for determining the number of photons to be emitted by surface in the Monte Carlo simulation is beyond the scope of this work, it is possible to make several observations that could assist in the creation of such a method.

It is of course possible to apply a convergence criterion to the smoothed results to determine how many photons emissions should be added to the current total when restarting the Monte Carlo simulation. In fact, if the material properties are independent of temperature, the criterion could be applied after the thermal balance code has been run and the calculated temperatures could be used as weights in the criterion. Still, there are advantages to having the convergence criterion self-contained in the Monte Carlo code itself. In particular, it removes the complication of having several codes communicate with each other and the need to create scripts to keep running all the codes in sequence until convergence.

As noted before, the computational requirements of BML / σ^2 smoothing are insignificant but the memory requirements for the actual smoothing are large. Since Monte Carlo simulations are often done in parallel with limited memory resources, this suggests that convergence criteria used during the Monte Carlo simulation could benefit from BML estimation calculations but should probably not use the LSS technique. Considering that the Δ results and the derivation of the BML / σ^2 smoothed confidence intervals above indicate that σ^2 smoothing usually changes the BML estimated results only slightly, this constraint should not be too limiting on a convergence technique. The fact that smoothing changes the BML estimated results little is not surprising since the purpose of the LSS method is to enforce closure while minimizing the changes in the estimated η.

The problem with creating a convergence criterion for reciprocity estimated results, even when unsmoothed, is that almost every result depends on the emissions from two surfaces. Determining the minimum number of emissions for the surfaces as a whole is probably impossible. Still, the Δ_F^k ratio defined above has great potential in cases such as these to be a “potentiometer” in determining how many photons to emit from a surface. When a surface has a low Δ_F^k ratio, the number of photons it needs to emit to obtain a given level of accuracy is diminished. While the Δ_F^k ratio is usually not known, it is easily estimated as described below.

The first step in the estimation is to assume the results are normally distributed, which means the E_F themselves are normally distributed with the same standard deviations as the F's, σ_F, and an average value of zero. Then the expected value for E_F^2, $\mathrm{E}[E_F^2]$, is:

$$\mathrm{E}[E_F^2] = 2\int_0^{\infty} E_F{}^2\left[\frac{1}{\sqrt{2\pi}\sigma_F}\exp\left(\frac{E_F^2}{2\sigma_F^2}\right)\right]dE_F = \sigma_F^2 \tag{3.56}$$

where the quantity in brackets in the integral is the probability distribution function for the normally distributed errors. Therefore, if the E_F's are normally distributed, then the expected value of the Δ_F^k is the normalized surface σ_F l_2-norm, $\Delta_{\sigma_F}^k$, which is defined:

$$\Delta_{\sigma_F}^k = \sqrt{\frac{1}{N_{surf}}\sum_{l=1}^{N_{surf}}\sigma_{kl,F}^2} \tag{3.57}$$

$\Delta_{\sigma_F}^k$ is easily calculated for the original Monte Carlo results using eqn. (3.17) and for the BML estimated results using eqn. (3.31), but it is not easily calculated for the BML$/\sigma^2$ smoothed results which are not assumed to be known during the Monte Carlo simulation anyway.

Since the effects of smoothing are found to be small, an estimate for the Δ_F^k ratio, the $\Delta_{\sigma_F}^k$ ratio, is defined as $\Delta_{\sigma_F}^k$ for the unsmoothed BML estimated results divided by $\Delta_{\sigma_F}^k$ for the original Monte Carlo results. As a measure of expected error instead of actual error, the $\Delta_{\sigma_F}^k$ ratio is found to be surprisingly constant. The most an individual $\Delta_{\sigma_F}^k$ ratio value differs from its mean value is 1.5% for the ETF geometry with 10,000 photons emitted per surface and by 0.4% for the other two ensembles. The $\Delta_{\sigma_F}^k$ ratio is also found to be a good estimate of the Δ_F^k ratio, differing by 2.3% at most from the mean of the Δ_F^k ratio.

The $\Delta_{\sigma_F}^{k}$ ratio is more likely to be an overestimate of the mean Δ_F^k ratio then an underestimate.

The best way to apply this potentiometer, the $\Delta_{\sigma_F}^{k}$ ratio, is not clear. One use is as a part of a function used to reduce an estimate of the number of photons required obtained from another source such as the maximum confidence interval for a given surface. Realizing that the smaller Ω value for a surface or the closer $\Delta_{\sigma_F}^{k}$ ratio is to one the less benefit it gets from other surfaces, the $\Delta_{\sigma_F}^{k}$ ratio can be used to determine the order in which the surfaces are handled in the search for convergence. Surfaces with smaller Ω values and higher $\Delta_{\sigma_F}^{k}$ ratios should have their emissions increased first. It can only decrease the number of emissions required for later surfaces.

3.6 Conclusions

An in-depth study has been completed of the effects of applying the reciprocity and closure constraints on Monte Carlo radiative transport simulation results using the methods of reciprocity estimation and least-squares smoothing (LSS). Of the reciprocity estimation smoothing (RES) techniques investigated, binomial maximum likelihood (BML) estimation with σ^2 weighted LSS is found to be an unbiased estimator that gives the greatest improvements in accuracy (lowest error) compared to the original Monte Carlo results. For least-squares smoothing, σ^2 weighting is preferred over η^2 weighting because it produces smaller errors in every case. The uncertainties (errors) of the matrix as a whole for the BML / σ^2 RES method and the original Monte Carlo results are found to decrease as the square root of the number of emissions. For the BML / σ^2 RES method, the improvement in error by surface compared to the original Monte Carlo results is found to increase as the area-emissivity product, Ω, for the surface increases. While detailed analysis indicates that a small number of surfaces for a single Monte Carlo run have their errors increased by BML/σ^2 smoothing, the size of these increases is small. Over several runs, all surfaces experience a mean decrease in error; quite a number of the mean decreases are significant. Furthermore, the BML/σ^2 smoothed results for the individual η are found to be normally distributed when even relatively few photons are absorbed by the surfaces. A good estimate of the confidence interval for these BML/σ^2 smoothed results can be obtained using the BML σ formula, eqn. (3.31). The $\Delta_{\sigma_F}^{k}$ ratio, an estimate of the effects of BML/σ^2 smoothing on surface from the results of a single run is shown to be accurate and potentially useful as a potentiometer in Monte Carlo convergence criteria calculations. The BML/σ^2 RES method is found to take insignificant calculation time. Its memory requirements, while large, should be no larger than the thermal analysis code that uses its results.

CHAPTER 4 CONCLUSIONS AND RECOMMENDATIONS

4.1 Summary of Work

In this work, two ways to reduce the computational requirements of radiative heat transfer Monte Carlo simulation are explored. Drawing on techniques used in the computer graphics field of ray tracing, an efficient algorithm for tracing particles in large, arbitrary, planar geometries containing nonparticipating media is presented. An efficient intersection algorithm for arbitrary triangles and/or convex planar quadrilaterals is derived in detail by drawing on techniques used in the computer graphics field of ray tracing. To determine the efficiency of the intersection algorithm, timing results are presented for a number of different spatial divisions for four geometries.

Second, the reciprocity estimation smoothing (RES) method which takes advantage of the redundant information in the Monte Carlo total exchange area matrix by the application of reciprocity and closure to potentially improve the accuracy of the total exchange area estimates is investigated. Statistical theory is applied to obtain the best RES method. An in-depth view of the distribution of the effects of RES is obtained by performing many numerical experiments (“runs”) on two large geometries.

4.2 Conclusions

A summary of the major conclusions of this work follows.

- By applying methods from the computer graphics field of ray tracing, an efficient Monte Carlo particle (photon) tracing algorithm for large geometries with arbitrary, planar surfaces in nonparticipating media is presented. It is found to be 35% to 45% faster than the previous MONT3D photon tracing algorithm [Maltby, 1987; Burns et al., 1990; Maltby and Burns, 1991; Burns and Pryor, 1999; Maltby et al., 1994].

- The uniform spacial division (USD) method used in the tracing algorithm yields speedups in run time as great as 81. Execution time is found to vary slowly with the number of voxels (grid cells). For geometries with 1,000 to 5,000 surfaces, good speedups are found when around 15,000 voxels are used.
- The mailbox ray tracing technique is found to increase execution times in all cases and is not recommended for radiative heat transfer Monte Carlo.
- The memory requirements of the Monte Carlo photon tracing algorithm are found to be slight. For geometries tested, the USD method requires less than one megabyte of memory and the rest of the Monte Carlo code as a whole requires between 1.7 and 5.2 megabytes of additional memory.
- An in-depth study of the reciprocity estimation smoothing (RES) method for collision based radiative heat transfer Monte Carlo demonstrates that binomial maximum likelihood (BML) estimation combined with σ^2 weighted least-squares smoothing (LSS) gives the best results (lowest error). Furthermore, BML / σ^2 smoothing is found to be an unbiased estimator.
- For least-squares smoothing, σ^2 weighting is found in all cases to give better overall results than η^2 weighting. For the matrix as a whole, σ^2 weighting usually reduces the error slightly while η^2 weighting increases the error in every case.
- The uncertainties (errors) of the matrix as a whole for the BML / σ^2 smoothing and the original Monte Carlo results are found to decrease as the square root of the number of emissions.

- Over several runs, BML / σ^2 smoothing causes all surfaces to experience a mean decrease in error compared to the original Monte Carlo results. While the mean decrease in error is small for the surfaces with the smallest values of Ω, the area-emissivity product, the mean decrease in error increases with increasing Ω values and is often quite significant
- While a small number of surfaces for a single Monte Carlo run have their errors increased by BML / σ^2 smoothing, the size of those increases is small.
- Even when relatively few photons are absorbed by surfaces, BML / σ^2 smoothed results for the individual η are found to be normally distributed. A good estimate of the confidence interval for these BML / σ^2 smoothed results can be obtained using the BML σ formula, eqn. (3.31).
- BML / σ^2 smoothing is shown to take insignificant calculation time and is expected to require at least no more memory then the thermal analysis code that uses its results.
- The $\Delta^k_{\sigma_F}$ ratio is found to be an accurate measure of the improvement in results by BML / σ^2 smoothing over the original Monte Carlo results for a surface that can be calculated from the results of a single run.

4.3 Recommendations

Recommendations for future work follows.

- Work should be done on deriving an efficient Monte Carlo particle tracing algorithm for participating media.
- Monte Carlo simulation convergence criteria that take into account the effects of BML / σ^2 results should be investigated. The $\Delta^k_{\sigma_F}$ ratio derived in this work would be a good starting point for that investigation.

- Studies should be done to determine the effect of the reciprocity improvements from RES on the accuracy of the final thermal analysis results as suggested by the work of Taylor et al. [1993; 1995] and Clarksean and Solbrig [1994].
- RES should be extended to pathlength based radiative heat transfer Monte Carlo. The minimum variance estimation discussed in this work should be well suited to pathlength based Monte Carlo RES. In pathlength based Monte Carlo, zero η values are almost always truly zero, so the problems minimum variance estimation exhibits with collision based Monte Carlo should not occur.
- Work should also be done on obtaining an RES technique for the zone method. The results of this dissertation suggest that the LSS methods used on the zone method may cause significant errors.
- More complex radiative material models that obey reciprocity should be determined. The only radiative heat transfer material models that are currently known to obey reciprocity are combinations of diffuse reflection as a constant function of angle and specular reflection as a varying function of angle [Zeeb et al., 1999; Branner, 2000].

REFERENCES

Altman, N. S., 1988. “Bit-wise Behavior of Random Number Generators,” *SIAM Journal on Scientific and Statistical Computing*, Vol. 9 No. 5, pp. 941-949.

Amanatides, J., and Woo, A., 1987. “A Fast Voxel Traversal Algorithm for Ray Tracing,” In *Proceedings of EUROGRAPHICS ‘87*, G. Marechal (ed.), Elsevier Science Publishers B. V., North-Holland, pp. 3-9.

Anderson, S. L., 1990. “Random Number Generators on Vector Supercomputers and Other Advanced Architectures,” *SIAM Review*, Vol. 32 No. 2, pp. 221-251.

Anton, H., 1987. *Elementary Linear Algebra, 5th Ed.*, John Wiley & Sons, New York.

Arnaldi, B., Priol, T., and Bouatouch, K. 1987. “A New Space Subdivision Method for Ray Tracing CSG Modelled Scenes,” *The Visual Computer*, Vol. 3, Springer-Verlag, pp. 98-108.

Arvo, J., and Kirk, D., 1989. “A Survey of Ray Tracing Acceleration Techniques,” In *An Introduction to Ray Tracing*, A. Glassner (ed.), Academic Press, San Diego, CA, pp. 201-262.

Branner, K., 2000. *An Enhanced Material Model for Radiative Heat Transfer via Monte Carlo*, M.S. Thesis, Colorado State University, Fort Collins, CO.

Brent, R. P., 1992. “Uniform Random Number Generators for Supercomputers, “*Proceedings Fifth Australian Supercomputing Conference*, SASC Organizing Committee, pp. 95-104. (unpublished).

Brent, R. P., 1994. “On the Periods of Generalized Fibonacci Recurrences,” *Mathematics of Computation*, Vol. 63, pp. 389-401.

Brewster, M.Q., 1992. *Thermal Radiative Transfer and Properties*, John Wiley & Sons, New York.

Burden, R., L., and Faires, J. D., 1993. *Numerical Analysis, 5th Ed.*, PWS-Kent Publishing, Boston.

Burns, P. J., Maltby, J. D., and Christon, M. A., 1990. “Large-Scale Surface to Surface Transport for Photons and Electrons via Monte Carlo,” *Computing Systems in Engineering*, Vol. 1 No. 1, pp. 75-99.

Burns, P. J., Loehrke, R. I., Dolaghan, J. S., and Maltby, J. D., 1992. “Photon Tracing in Axisymmetric Enclosures,” *Developments in Radiative Heat Transfer*, American Society of Mechanical Engineers, HTD-Vol. 203, pp. 93-100.

Burns, P. J., and Pryor, D. V., 1999. “Surface Radiative Transport at Large Scale via Monte Carlo,” In Vol. 9 of *Annual Review of Heat Transfer*, C. L. Tien (ed.), Begell House, New York, pp. 79-158.

Chen, S., Chan, T. L., Leung, C. W., Liu, M. A., Pan, K. Y., and Zhou, L. B., 2000. "Multidimensional Numerical Simulation of Heat Radiation in Direct Injection Diesel Engines," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, Vol. 214 No. 4, pp. 453-466.

Chin, J. H., Panczak, T., and Fried, L., 1989. "Finite Element and Raytracing in Coupled Thermal Problems," In *Numerical Methods in Thermal Problems, Vol. VI, Proceedings of the Sixth International Conference*, R. W. Lewis, and K. Morgan (eds.), Pineridge Press, Swansea, U.K., pp. 683-701.

Chin, J. H., Panczak, T., and Fried, L., 1992. "Spacecraft Thermal Modelling," *International Journal for Numerical Methods in Engineering*, Vol. 35, pp. 641-653.

Chin, N., 1995. "A Walk through BSP Trees," In *Graphic Gems V*, A. W. Paeth (ed.), AP Professional, San Diego, CA, pp. 121-138.

Clarksean, R. and Solbrig, C., 1994. "Minimization of the Effect of Errors in Approximate Radiation View Factors," *Nuclear Engineering and Design*, Vol. 149, pp. 431-440.

Cleary, J. G., and Wyvill, G., 1988. "Analysis of an Algorithm for Fast Ray Tracing Using Uniform Space Subdivision," *The Visual Computer*, Vol. 4, Springer-Verlag, pp. 65-83.

Cohen, D., 1994. "Voxel Traversal along a 3D Line" In *Graphic Gems IV*, P. S. Heckbert (ed.), AP Professional, San Diego, CA, pp. 366-369.

Corlett, R. C., 1966. "Direct Monte Carlo Calculation of Radiative Heat transfer in a Vacuum," *Journal of Heat Transfer*, Vol. 88, pp. 376-382.

D'Agostino, R. B., and Stephens, M. A. (eds.), 1986. *Goodness-of-Fit Techniques*, Marcel Dekker, Inc., New York.

Davis, C. S., and Stephens, M. A., 1989. "Algorithm AS 248: Empirical Distribution Function Goodness-of-Fit Tests," *Applied Statistics*, Vol. 38 No. 3, pp.535-543.

Dolaghan, J. S., Loehrke, R. I., and Burns, P. J., 1992. "User's Manual for SMOOTH," Department of Mechanical Engineering, Colorado State University, Fort Collins, CO 80523.

Evans, M., Hastings, N., and Peacock, B., 2000. *Statistical Distributions, 3rd Ed.*, John Wiley and Sons, New York.

Farmer, J. T., 1995. *Improved Algorithms for Monte Carlo Analysis of Radiative Heat Transfer in Complex Participating Media*, Ph.D. Dissertation, University of Texas at Austin, Austin, TX.

Fujimoto, A., Tanaka, T., and Iwata, K., 1986. "ARTS: Accelerated Ray Tracing System," *IEEE Computer Graphics and Applications*, Vol. 6 No. 4, pp. 16-26.

Glassner, A. S., 1984. "Space Subdivision for Fast Ray Tracing," *IEEE Computer Graphics and Applications*, Vol. 4 No. 10, pp. 15-22.

Haines, E., 1989. "Essential Ray Tracing Algorithms," In *An Introduction to Ray Tracing*, A. Glassner (ed.), Academic Press, San Diego, CA, pp. 33-77.

Haines, E., 1994. “Point in Polygon Strategies,” In *Graphic Gems IV*, P. S. Heckbert (ed.), AP Professional, San Diego, CA, pp. 24-46.

Haji-Sheikh, A., 1988. “Monte Carlo Methods,” Chapter 16 in *Handbook of Numerical Heat Transfer*, Minkowycz et al. (eds.), John Wiley & Sons, New York, NY, pp. 673-722.

Henson, J. C., Malalasekera, M. G., and Dent, J. C., 1996. “Heat Transfer in Three-Dimensional, Nonhomogeneous Participating Media,” *ASME Proceedings of the 31st National Heat Transfer Conference - Vol. 3 - Solution Methods for Radiative Heat Transfer in Participating Media*, HTD-Vol. 276, ASME, New York, pp. 25-34.

Hottel, H. C., and Sarofim, A. F., 1967. *Radiative Transfer*, McGraw-Hill, St. Louis.

Howell, J. R., and Perlmutter, M., 1964a. “Monte Carlo Solution of Thermal Transfer Through Radiant Media Between Gray Walls,” *ASME Journal of Heat Transfer*, Vol. 86, pp. 116-122.

Howell, J. R., and Perlmutter, M., 1964b. “Monte Carlo Solution of Radiant Heat Transfer in a Nongray Nonisothermal Gas with Temperature Dependent Properties,” *American Institute of Chemical Engineers Journal*, Vol. 10, No. 4, pp. 562-567.

Howell, J. R., 1968. “Application of Monte Carlo to Heat Transfer Problems,” In Vol. 5 of *Advances in Heat Transfer*, T. F. Irvine and J. P. Hartnett (eds.), Academic Press, New York, pp. 1-54.

Howell, J. R., 1998. “The Monte Carlo Method in Radiative Heat Transfer,” *ASME Journal of Heat Transfer*, Vol. 120, pp. 547-560.

Jeger, M., and Eckmann, B., 1967. *Vector Geometry and Linear Algebra (For Engineers and Scientists)*, Interscience Publishers, New York.

Kaplan, M. R., 1987. “The Use of Spatial Coherence in Ray Tracing,” In *Techniques for Computer Graphics*, D. F. Rogers, and R. A. Earnshaw (eds.), Springer-Verlag, New York, NY, pp. 174-193.

Koeck, C., 1988. “Improved Ray Tracing Technique for Radiation Heat Transfer Modelling,” *Proceedings of the 3rd European Symposium on Space Thermal Control & Life Support Systems*, Noordwijk, The Netherlands, (3-6 Oct. 1988) ESA SP-288, pp. 255-260.

Kreyszig, E., 1993. *Advanced Engineering Mathematics, 7th Ed.*, John Wiley & Sons, New York.

Larsen, M. E., and Howell, J. R., 1986. “Least-Squares Smoothing of Direct-Exchange Areas in Zonal Analysis,” *Journal of Heat Transfer*, Vol. 108, pp. 239-242.

Lewis, E. E., and Miller, W. F. Jr. , 1984. *Computational Methods of Neutron Transport*, John Wiley & Sons, New York, NY pp. 296-360.

Liu, M. S., Choi, C. K., and Leung, C. W., 2001. “Startup Analysis of Oil-Fired Furnace - the Smoothing Monte Carlo Model Approach,” *Heat and Mass Transfer*, Vol. 37, pp. 449-457.

Loehrke, R. I., Dolaghan, J. S., and Burns, P. J., 1995. “Smoothing Monte Carlo Exhange Factors,” *ASME Journal of Heat Transfer*, Vol. 117, pp. 524-526.

Lux, I., and Koblinger, L., 1991. *Monte Carlo Particle Transport Methods: Neutron and Photon Calculations*, CRC Press, Ann Arbor, MI

Maltby, J. D., 1987. *Three-Dimensional Simulation of Radiative Heat Transfer by the Monte Carlo Method*, M.S. Thesis, Colorado State University, Fort Collins, CO.

Maltby, J. D., 1990. *Analysis of Electron Heat Transfer via Monte Carlo Simulation*, Ph.D. Dissertation, Colorado State University, Fort Collins, CO.

Maltby, J. D., and Burns, P. J., 1991. “Performance, Accuracy and Convergence in a Three-Dimensional Monte Carlo Radiative Heat Transfer Simulation,” *Numerical Heat Transfer, Part B: Fundamentals*, Vol. 16, pp. 191-209.

Maltby, J. D., 1994. “Evaluation of Property-induced Uncertainty in a Monte Carlo Simulation of Radiative Heat Transfer in a Participating Medium,”*Radiative Heat Transfer: Current Research*, American Society of Mechanical Engineers, HTD-Vol. 276, pp. 161-170.

Maltby, J. D., Zeeb, C. N., Dolaghan, J., and Burns, P. J., 1994. *User’s Manual for MONT2D - Version 2.6 and MONT3D - Version 2.3*, Department of Mechanical Engineering, Colorado State University, Fort Collins, CO.

Marsaglia, G., 1985. “A Current View of Random Number Generators”, In *Computing Science and Statistics: Proceedings of the XVIth Symposium on the Interface*, L. Billard (ed.), Elsevier Science Publishers B. V., North-Holland, pp. 3-10.

Marsaglia, G. and Tsay, L.-H., 1985. “Matrices and the Structure of Random Number Sequences”, *Linear Algerbra and Its Applications.*, Vol. 67 pp. 147-156.

Martin, B. R., 1971. *Statistics for Physicists*, Academic Press, New York, NY.

Mascagni, M., Cuccaro, S., Pryor, D., and Robinson, M., 1995a. “A Fast, High Quality, and Reproducible Parallel Lagged-Fibonacci Pseudorandom Number Generator,” *Journal of Computational Physics*, Vol. 119, pp. 211-219.

Mascagni, M., Robinson, M., Pryor, D., and Cuccaro, S., 1995b. “Parallel Pseudorandom Number Generation Using Additive Lagged-Fibonacci Recursions,” *Springer-Verlag Lecture Notes in Statistics*, Vol. 106 pp. 263-277.

Modest, M. F., 1978. “Three-dimensional Radiative Exchange Factors for Nongray, Diffuse Surfaces,” *Numerical Heat Transfer*, Vol. 1, pp. 403-416.

Modest, M. F., 1993. *Radiative Heat Transfer*, McGraw-Hill, St. Louis, MO.

Mood, A. M., Graybill, F. A., and Boes, D. C., 1974. *Introduction to the Theory of Statisitics, 3rd Ed.*, McGraw-Hill, St. Louis, MO.

Murty, C. V. S., and Murty, B. S. N., 1991. “Significance of Exchange Area Adjustment in Zone Modelling,” *International Journal of Heat and Mass Transfer*, Vol. 34, No. 2, pp. 499-503.

Panczak, T. D., 1989. “A Fast, Linear Time, Monte Carlo Radiation Interchange Program Utilizing Adaptive Spatially Coherent Subdivision,” In *Numerical Methods in Thermal*

Problems, Vol. VI, Proceedings of the Sixth International Conference, R. W. Lewis and K. Morgan (eds.), Pineridge Press, Swansea, U.K., pp. 701-712.

Perlmutter, M., and Howell, J. R. 1964. "Radiative Transfer Through a Gray Gas Between Concentric Cylinders Using Monte Carlo," *ASME Journal of Heat Transfer*, Vol. 86, pp. 169-179.

Pryor, D. V., Cuccaro, S. A., Mascagni, M., and Robinson, M. L., 1994. "Implementation of a Portable and Reproducible Parallel Pseudorandom Number Generator," *Supercomputing '94 Proceedings*, IEEE Computer Society Press, Los Alamitos, CA, pp. 311-319.

Raboin, P. J., 1998. "Computational Mechanics Moves Ahead," *Science and Technology Review*, May, Lawrence Livermore National Laboratory, UCRL-52000-98-5, pp. 12-19.

Ross, S., 1988. *A First Course in Probability, 3rd Ed,* Macmillian, New York, NY.

Royston, P., 1993. "A Toolkit for Testing for Non-normality in Complete and Censored Samples," *Statistician*, Vol. 42 No. 1, pp. 37-43.

Royston, P., 1995. "Remark AS R94: A Remark on Algorithm 181: The *W*-test for Normality," *Applied Statistics*, Vol. 44 No. 4, pp. 547-551.

Rushmeier, H. E., 1993. "Computer Graphics Techniques for Computing Radiation Heat Transfer," *Proceedings of the NFS Joint Workshop on Radiative Heat Transfer in Highly Coupled Physical Systems*, University of Texas at Austin, Austin, TX, October 4-8.

Samet, H., 1989. "Implementing Ray Tracing with Octrees and Neighbor Finding," *Comput. & Graphics*, Vol. 13 No. 4, pp. 445-460.

Shapiro, A. B., 1985. "TOPAZ3D - A Three-Dimensional Finite Element Heat Transfer Code," Lawrence Livermore National Laboratory, UCID-20484.

Shapiro, S. S. and Wilk, M. B., 1965. "An Analysis of Variance Test for Normality (Complete Samples)," *Biometrika*, Vol. 52 No. 3/4, pp. 591-611.

Shapiro, S. S., 1986. "How to Test Normality and Other Distributional Assumptions," Vol. 3 of *The ASQC Basic References in Quality Control: Statisitical Techniques*, E. J. Dudewicz (ed.), American Society for Quality Control, Milwaukee, WI.

Siegel, R., and Howell, J. R., 1992. *Thermal Radiation Heat Transfer, 3rd Ed.*, Hemisphere Publishing, Washington.

Snyder, W. C., Wan, Z., and Li, X., 1998. "Thermodynamic Constraints on Reflectance Reciprocity and Kirchhoff's Law," *Applied Optics*, Vol. 37 No. 16, pp. 3,464-3,470.

Sobel', I. M., 1974. *The Monte Carlo Method*, University of Chicago Press, Chicago, IL.

Sowell, E. F., and O'Brien, R. F., 1972. "Efficient Computation of Radiant-interchange Configuration Factors Within an Enclosure," *ASME Journal of Heat Transfer*, Vol. 94, pp. 326-328.

Stephens, M. A., 1986. "Tests Based on EDF Statistics," In *Goodness-of-Fit Techniques*, R. B. D'Agostino, and M. A. Stephens, (eds.), Marcel Dekker, Inc., New York. pp. 97-193.

Sung, K., 1991. “A DDA Octree Traversal Algorithm for Ray Tracing,” In *Proceedings of EUROGRAPHICS ‘91*, F. H. Post and W. Barth (eds.), Elsevier Science Publishers B. V., North-Holland, pp. 73-85.

Sung, K., and Shirley, P., 1992. “Ray Tracing with a BSP Tree,” In *Graphic Gems III*, D. Kirk (ed.), AP Professional, San Diego, CA, pp. 271-274.

Sutton, S., Erlandson, A., London, R., Manes, K., Marshall, C., Petty, C., Pierce, R., Smith, L., Zapata, L., Beullier, J., and Bicrel, B., 1998. *Thermal Recovery of the NIF Amplifiers*, Lawrence Livermore National Laboratory, UCRL-JC-124528 Rev 1.

Taylor, R. P., Luck, R., Hodge, B. K., and Steele, W. G., 1993. ”Uncertainty Analysis of Diffuse-Gray Radiation Enclosure Problems - A Hypersensitive Case Study,” *5th Annual Thermal and Fluids Analysis Workshop,* NASA CP 10122, pp. 27-40.

Taylor, R. P., and Luck, R., 1995. ”Comparison of Reciprocity and Closure Enforcement Methods for Radiation View Factors,” *Journal of Thermophysics and Heat Transfer,* Vol. 9, No. 4, pp. 660-666.

Taylor, R. P., Luck, R., Hodge, B. K., and Steele, W. G., 1995. ”Uncertainty Analysis of Diffuse-Gray Radiation Enclosure Problems,” *Journal of Thermophysics and Heat Transfer,* Vol. 9, No. 1, pp. 63-69.

Tsuyuki, G. T., 1992. “TRASYS Form Factor Matrix Normalization,” *4th Annual Thermal and Fluids Analysis Workshop,* NASA CP 10106, pp. 71-82.

Turk, G., 1990. “Generating Random Points in Triangles,” In *Graphic Gems*, A. S. Glassner (ed.), AP Professional, San Diego, CA, pp. 24-28.

Van Leersum, J., 1989. “A Method for Determining a Consistant Set of Radiation View Factors from a Set Generated by a Non-Exact Method,” *International Journal of Heat and Fluid Flow,* Vol. 10, No. 1, pp. 83-85.

Vercammen, H. A. J., and Froment, G. F., 1980. “An Improved Zone Method Using Monte Carlo Techniques for the Simulation of Radiation in Industrial Furnaces,” *International Journal of Heat and Mass Transfer,* Vol. 23, No. 3, pp. 329-336.

Watt, A., and Watt, M. 1992. *Advanced Animation and Rendering Techniques: Theory and Practice*, ACM Press, New York, NY.

XYZ Scientific Applications, Inc., 1997. *TrueGrid Manual - Version 1.4.0,* XYZ Scientific Applications, Inc., Livermore, CA.

Zeeb, C. N., and Burns, P. J., 1997. *Random Number Generator Recommendation,* Report prepared for Sandia National Laboratories, Albuquerque, NM. Updated version available as a WWW document, URL =http://www.colostate.edu/~pburns/monte/documents.html.

Zeeb, C. N., Burns, P. J., Branner, K., and Dolaghan, J., 1999. *User’s Manual for MONT3D - Version 2.4*, Department of Mechanical Engineering, Colorado State University, Fort Collins, CO.

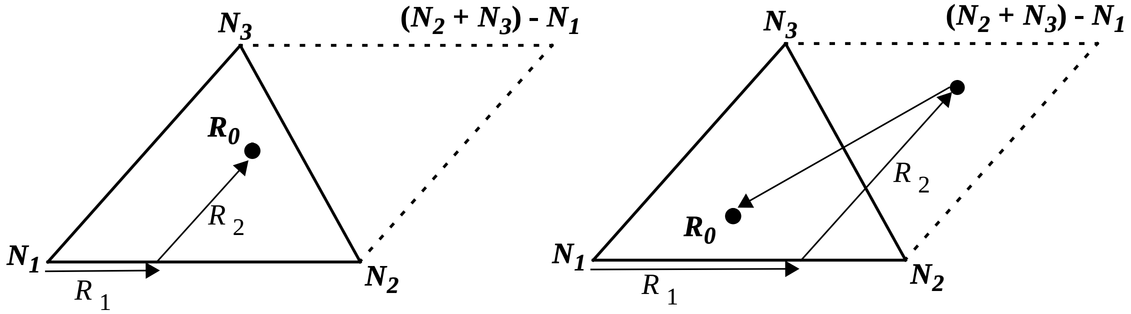
Zeeb, C. N., and Burns, P. J., 2000. *User's Manual for LSMONTE: A Three-dimensional, Radiative Heat Transfer Analysis Computer Code*, Department of Mechanical Engineering, Colorado State University, Fort Collins, CO

APPENDIX A RANDOM EMISSION

This work and LSMONTE [Zeeb and Burns, 2000] implement random emission from each surface. Since temperature is constant for each surface, the emission routine is derived from the routine by Turk, 1990 for uniformly distributed random points in a triangle. It depends on two uniformly distributed random numbers between zero and one, R_1 and R_2, and the three node points of the triangle, $\boldsymbol{N_1}$, $\boldsymbol{N_2}$, and $\boldsymbol{N_3}$. Two different formulas are used to calculate the origin of the photon, $\boldsymbol{R_0}$, depending on the sum of R_1 and R_2.

$$\boldsymbol{R_0} = (1 - R_1 - R_2)\boldsymbol{N_1} + R_1\boldsymbol{N_2} + R_2\boldsymbol{N_3} \qquad R_1 + R_2 \le 1 \tag{A.1}$$

$$\boldsymbol{R_0} = (R_1 + R_2 - 1)\boldsymbol{N_1} + (1 - R_1)\boldsymbol{N_2} + (1 - R_2)\boldsymbol{N_3} \qquad R_1 + R_2 > 1 \tag{A.2}$$



(a) $R_1 + R_2 \le 1$ ***(b)*** $R_1 + R_2 > 1$

Figure A.1 Examples of the Random Emission Algorithm

As shown in Fig. A.1, eqn. (A.1) defines a random point in the parallelogram with vertices, $\boldsymbol{N_1}$, $\boldsymbol{N_2}$, $\boldsymbol{N_3}$, and ($\boldsymbol{N_2}$ + $\boldsymbol{N_3}$) - $\boldsymbol{N_1}$. A point that lands in the triangle $\boldsymbol{N_2}$, $\boldsymbol{N_1}$, ($\boldsymbol{N_2}$ + $\boldsymbol{N_3}$) - $\boldsymbol{N_1}$ is moved into the triangle $\boldsymbol{N_1}$, $\boldsymbol{N_2}$, $\boldsymbol{N_3}$ by reflecting it about the center of the parallelogram using eqn. (A.2). While it is possible to generalize the above equations to any convex polygon [Turk, 1990], the code reduces the overhead of calculation by dividing each quadrilateral into two triangles and emitting from each separately.

MONT3D [Zeeb et al., 1999; Maltby, 1987] uses a fixed point emission routine which divides each surface into equally spaced regions and emits from the centroid of each of the subsurfaces. The reason this method was chosen is that all previously known random emission routines rely on an accept-reject method that can be very inefficient for surfaces where most of the generated points are rejected. To handle surfaces that are not well approximated by squares, the fixed point emission routine requires many subsurfaces and many photons emitted per subsurface, which in turn requires a large number of photons to be emitted per surface. Since the number of photons emitted per surface is often limited, this can lead to small reciprocity errors in the results. This random emission routine does not suffer from that problem.

APPENDIX B THE CENTER FOR COMPUTING SCIENCE PSEUDO-RANDOM NUMBER GENERATOR

The random number generator used in this work, the latest version of MONT3D [Zeeb et al., 1999], and MONT3D's public domain counterpart, LSMONTE, [Zeeb and Burns, 2000] is the 127-seed Center for Computing Science addition lagged-Fibonacci pseudo-random number generator also known as simply the CCS RNG. This generator has been discussed by Burns and Pryor [1999] and further studied and actually named by Zeeb and Burns [1997]. An implementation is available at <http://www.colostate.edu/~pburns/monte/code.html>. This generator is chosen because it is found to be computationally efficient, and to have good statistical properties and a long period.

An addition lagged-Fibonacci generator with u integer seeds has the form [Anderson, 1990]:

$$X_i = (X_{i-u} + X_{i-v}) \bmod m \qquad u > v > 0 \tag{B.1}$$

where the X's are integer values, mod is the integer remainder function, and u and v are the lags, which are limited to certain values. The modulus, m, is usually a power of two. For portability to almost any computer architecture, m is 2^{31} for the CCS generator. Uniform random numbers, R_i, are generated from the above integers as follows:

$$R_i = \frac{X_i}{m} \qquad 0 \le R_i < 1 \tag{B.2}$$

While addition lagged-Fibonacci generators were originally avoided due to their lack of firm theoretical basis, this is unjustified today due to the large volume of work and theory published about them [Marsaglia and Tsay, 1985; Brent, 1992, 1994; Pryor et al., 1994; Mascagni et al., 1995a, 1995b]. The 127-seed generator is chosen because, when properly seeded, it is the is Fibonacci generator with the smallest value of u for the CCS generator that passes all of Marsaglia's [1985] stringent DIEHARD random number generator tests including the birthday spacings test [Brent, 1992].

This generator is called CCS because it implements a seeding scheme developed there [Pryor et al., 1994; Mascagni et al., 1995a, 1995b] to produce one of the $2^{u-2}m$ (m is a power of two) independent random sequences of maximum length. The maximum length of a cycle is $(2^{u-1} - 1)m$ although no more than $2^{u} - 1$ of the numbers should be used in a series because of bit correlations [Brent, 1992]. Therefore, for 127 seeds, the period is at least $1.7\text{X}10^{38}$. The independence of each sequence of random numbers makes the CCS generator ideal for parallel programming. One potential problem with the CCS seeding algorithm is that the last bit of each random integer is not random [Zeeb and Burns, 1997; Burns and Pryor, 1999]. Since the last bit is not significant for the random numbers generated from the random integers, this is usually not a problem.

What differentiates the CCS generator from other generators based on the work of Mascagni and his coworker is that the CCS generator creates its initial set of seeds from a single 32-bit seed using a binary shift register [Zeeb and Burns, 1997; Burns and Pryor, 1999]. While limiting the possible number of random sequences to $2^{31} - 1$, extensive testing has shown this method gives excellent randomness properties across parallel sequences, even when the initial seeds for each sequence are very similar [Zeeb et al., 1999]. In particular, this initialization scheme avoids sequences with bad bitwise behavior [Altman, 1988].

APPENDIX C TESTS OF NORMALITY

The purpose of this appendix is to present the results of the normality tests described in the first section of this appendix for BML / σ^2 smoothed results. The results of the tests are summarized in Section 3.5.3.1.

C.1 Tests for Normality

While the normal distribution is known to be applicable in many circumstances [Evens et al., 2000], its applicability to RES results is not known. Fortunately, there is a large body of literature of tests to determine the normality of a given distribution [D'Agostino and Stephens, 1986; Shapiro, 1986]. To test for normality in this work, four tests of normality are chosen: three empirical distribution function (EDF) tests; A^2, U^2, and W^2; and the Shapiro-Wilk statistic, W. All are described below.

The empirical distribution function (EDF), $F_n(x)$ is created from a series of n ordered random samples from a distribution, and is defined as [Stephens, 1986]:

$$F_n(x) = \frac{\text{number of observations} \le x}{n} \tag{C.1}$$

An example of $F_n(\eta)$ for 100 samples of a specific η with value 2.87 is shown in Fig. C.1. As the figure shows, $F_n(x)$ is a step function that smooths out as the number of samples increases. Statistics based on $F_n(x)$ are often called EDF or distance statistics.

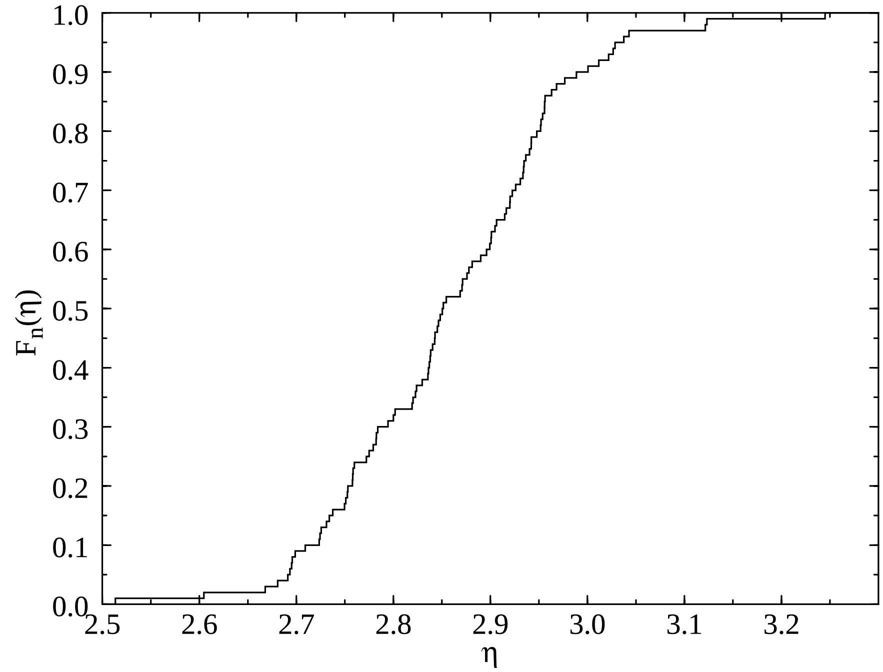


Figure C.1 Sample $F_n(\eta)$ Function

The three EDF tests used in this work are part of the quadratic or Cramér-von Mises family of statistics [Stephens, 1986] which determine the agreement of $F_n(x)$ with a given cumulative distribution function (CDF), $F(x)$, by integrated the weighted square of the difference of the two. The three statistics are: W^2, the Cramér-von Mises statistic:

$$W^2 = n \int_{-\infty}^{\infty} [F_n(x) - F(x)]^2 dF(x) \tag{C.2}$$

U^2, the Watson statistic:

$$U^2 = n \int_{-\infty}^{\infty} \left(F_n(x) - F(x) - \int_{-\infty}^{\infty} [F_n(x) - F(x)] dF(x) \right)^2 dF(x) \tag{C.3}$$

and A^2, the Anderson-Darling statistic:

$$A^2 = n \int_{-\infty}^{\infty} \frac{[F_n(x) - F(x)]^2}{\{F(x)[1 - F(x)]\}} dF(x) \tag{C.4}$$

Since $F(x)$ is the normal CDF and $F_n(x)$ is a step function, the calculation of these integrals is rather straight forward. Larger values of the statistics are indicative of sample distributions that are not normal because these are the situations where the difference between $F(x)$ and $F_n(x)$ is the greatest. A^2 is known to be the strongest statistic of the three in determining non-normality.

W, the Shapiro-Wilk statistic, is defined as [Royston, 1993]:

$$W = \left(\sum_{i=1}^{n} a_i x_i \right)^2 \Big/ \sum_{i=1}^{n} (x_i - \bar{x})^2 \tag{C.5}$$

which is essentially the ratio of the square of the BLUE (best linear unbiased estimate) of the standard deviation of x_i assuming normality, $\sigma_{\mathrm{BLUE,\ nor}}$:

$$\sigma_{\mathrm{BLUE,nor}} = \frac{1}{\sqrt{n-1}} \sum_{i=1}^{n} a_i x_i \tag{C.6}$$

where a_i are the weights, and the variance of the sample, σ^2:

$$\sigma^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2 \tag{C.7}$$

The W-statistic ranges from zero to one and small values of W indicate non-normality [Shapiro and Wilk, 1965]. It is known as a powerful test of normality.

Due to the large number of η to be tested in this work, these tests of normality were chosen for two reasons. First, codes to calculate the A^2, U^2, W^2, and W statistics with mean and variance unknown are available at <http://www.statlib.org> [Davis and Stephens, 1989; Royston, 1995]. Second, all four tests give a cumulative probability, *p*, that a normal distribution would generate such a statistic. Small values of *p* indicate non-normal behavior. This is equivalent to small values of the W statistic and large values of the A^2, U^2, and W^2 statistics. This simplifies comparisons of normality.

C.2 Description of How the Normality Tests Were Used

The first problem with analysis is the wealth of results to test. For the ETF geometry used in these tests which has 144 surfaces, there are 20,736 single η and 10,440 η that are either a reciprocity pair (RP) or a diagonal value. It was decided that the best use of this large amount of information is to determine rough general rules about how many photons must be emitted for the original Monte Carlo results and BML/σ^2 smoothed results to be normally distributed.

Due to the variance of the results between runs, the results are divided into different categories depending on the “ground truth” results. The “ground truth” results are used to determine which original Monte Carlo result or smoothed diagonal is expected to have an average number of photons absorbed equal to or greater than a given N and which smoothed reciprocity pair is expected to have both of its original Monte Carlo values equal to or above a certain N. For ETF geometry with 10,000 photons emitted per surface, the number of single values and pairs fitting into various categories is shown in Table C.1. The full number of individual values, RP’s, and diagonal values are given as results for N equal to zero. The columns marked Δ_N have the difference between that row and the row below it for the previous column. Since the results for each N are a subset of all lower value N’s, Δ_N is a measure of how many single values or pairs are within the current N range but below the next N range.

Table C.1: Numbers of Each Type of Single η and η Reciprocity Pair

N	Original Single: Value ≥ N	Δ_N	Smoothed Pair: Both ≥ N	Δ_N	Smoothed Diagonal: Value ≥ N	Δ_N
0	20,736	2,706	10,296	2,440	144	4
10	18,030	1,933	7,850	1,380	140	7
20	16,097	6,133	6,470	3,590	133	17
50	9,964	5,633	2,880	2,023	116	46
100	4,331	3,456	857	714	70	58
200	875	749	143	114	12	10
500	126	84	29	24	2	2
1,000	42	-	5	-	0	-

For these tests, only the runs for the ETF geometry with 10,000 photons per surface are used. For the ETF geometry, the main reason for this is that the results from runs with 10,000 photons per surface are mainly in the N equal 50 or less range, while most of the results from runs with 100,000 photons per surface are in the N equal 200 and above range. The crossover point for normal behavior for smoothed results is found at the lower N’s represented by the results from runs with 10,000 photons per surface. The Cham geometry is found to have too few runs to give meaningful results for these tests.

If only a few variables are being tested, then what is usually done is to repeat the test until it is clear whether or not the variable is normally distributed. For this work, to achieve this effect, the 10,000 runs with 10,000 photons per surface are divided into three different ensemble groupings described in Table C.2 so that normality can be tested at several granularities. With around 10,000 to 20,000 different η to analyze for normality, some way of easily checking each η has to be determined. This is done by realizing that the test results for each η are uniformly distributed between zero and one. Therefore, a rough measure of normality is how close the means of the values for a test are to 0.5. More on this will be presented below.

Table C.2: Normality Test Ensembles

Ensemble	Number of Values per Test	Number of Test Values
α	100	100
β	10	1,000
γ	5	2,000

C.3 Test Results

Of the three test ensembles, ensemble γ with only five values per test is found to be too course a granularity for accurate testing. Many times all five values used for the normality tests are the exact same value, invalidating the test. While this problem mainly occurs for the original Monte Carlo results, it has also occurred for some of the reciprocity estimated results. The problem arises for a wide range of number of absorbed photons, N. For the original Monte Carlo results, this is a problem even for $N \geq 100$. Clearly, the fact that the original Monte Carlo results and reciprocity estimated results are discrete instead of continuous has quite an effect on the normality of the results.

The two remaining ensembles using 10 or 100 values for their tests do not suffer from the repeating value problem. The minimum mean values for all the tests for both of these ensembles are given in Tables C.3 to C.8. The tables divide the results into three categories: 1) the original individual results, 2) the unsmoothed and smoothed reciprocity estimated results for the reciprocity pairs, and 3) the original and smoothed results for the diagonals which are unaffected by reciprocity estimation. The reason the tables list the minimum mean values is that non-normal behavior is found to lead to mean values less than 0.5. As mentioned above, small values of p for any of the normality tests is indicative of non-normal behavior so this is to be expected.

Of course, due to statistical scatter, some means below 0.5 are expected of even normally distributed results. In fact, applying the Central Limit Theorem [Ross, 1988] to a uniform distribution between zero and one suggests that the lower bound of the 99%

Table C.3: Minimum Mean Values for Ensemble α for the Monte Carlo Results

	Normality Test (100 Values per Test)			
N	W	A^2	U^2	W^2
10	0.08	0.04	0.03	0.03
20	0.18	0.13	0.11	0.11
50	0.30	0.27	0.26	0.26
100	0.35	0.32	0.31	0.32
200	0.40	0.38	0.38	0.38
500	0.41	0.41	0.41	0.42

Table C.4: Minimum Mean Values for Ensemble β for the Monte Carlo Results

	Normality Test (10 Values per Test)			
N	W	A^2	U^2	W^2
10	0.41	0.37	0.36	0.38
20	0.44	0.41	0.41	0.42
50	0.45	0.43	0.43	0.44
100	0.46	0.44	0.44	0.45
200	0.47	0.44	0.44	0.45
500	0.47	0.45	0.45	0.46

confidence interval for the mean of the 100 test values from ensemble α is 0.40 while the lower bound for the mean of the 1,000 test values from ensemble β is 0.47. Due to the uncertainty of the normality tests and the variable number of results at each N level, this work arbitrarily defines as normal any N level that is within 0.03 of the 99% confidence interval values calculated above. Where the various types of estimates meet this criterion is given in Table C.9.

The tables clearly demonstrate several trends. The first trend is that the more values a normality test uses, the more sensitive it is to deviations from the normal. Therefore, more results from ensemble β are considered normal than from ensemble α. Still, consid-

Table C.5: Minimum Mean Values for Ensemble α for Both of a Pair $\geq N$

N	(100 Values for Each Normality Test)							
	BML Estimation Only				BML / σ^2 Smoothing			
	W	A^2	U^2	W^2	W	A^2	U^2	W^2
10	0.24	0.16	0.14	015	0.30	0.25	0.24	0.25
20	0.29	0.26	0.24	0.25	0.36	0.36	0.36	0.36
50	0.36	0.34	0.33	0.34	0.38	0.38	0.38	0.38
100	0.40	0.38	0.38	0.39	0.41	0.40	0.40	0.40
200	0.44	0.41	0.39	0.40	0.43	0.43	0.42	0.43
500	0.47	0.45	0.44	0.44	0.45	0.45	0.43	0.44

Table C.6: Minimum Mean Values for Ensemble β for Both of a Pair $\geq N$

N	(10 Values for Each Normality Test)							
	BML Estimation Only				BML / σ^2 Smoothing			
	W	A^2	U^2	W^2	W	A^2	U^2	W^2
10	0.45	0.43	0.42	0.44	0.46	0.44	0.44	0.45
20	0.46	0.44	0.44	0.45	0.46	0.44	0.44	0.45
50	0.47	0.44	0.44	0.45	0.46	0.44	0.44	0.45
100	0.47	0.45	0.45	0.46	0.47	0.45	0.45	0.46
200	0.47	0.46	0.45	0.46	0.48	0.46	0.46	0.47
500	0.47	0.46	0.46	0.47	0.48	0.46	0.46	0.47

ering the problems that occur with ensemble γ, there is definitely a lower bound on the number of values that can be used in the normality tests. This suggests that the versions that use ten values are the least stringent forms of the normality tests used in this work. Looking at the tests themselves, W clearly defines more of the distributions in this work as normal then any of the EDF tests. Of the EDF tests, W^2 defines the most distributions as normal while U^2 defines the least.

Part of the deviation of the original Monte Carlo results from the normal distribution is their discreteness. As shown by the tables, the combination of the two results in reciproc-

Table C.7: Minimum Mean Values for Ensemble α for Diagonal $\geq N$

N	(100 Values for Each Normality Test)							
	Original Monte Carlo				BML / σ^2 Smoothing			
	W	A^2	U^2	W^2	W	A^2	U^2	W^2
10	0.12	0.06	0.05	0.06	0.25	0.21	0.20	0.21
20	0.28	0.21	0.20	0.21	0.39	0.41	0.42	0.42
50	0.37	0.31	0.29	0.29	0.42	0.42	0.42	0.43
100	0.40	0.36	0.33	0.34	0.42	0.42	0.42	0.43
200	0.42	0.39	0.39	0.40	0.44	0.44	0.46	0.45
500	0.52	0.51	0.51	0.51	0.52	0.51	0.51	0.51

Table C.8: Minimum Mean Values for Ensemble β for Diagonal $\geq N$

N	(100 Values for Each Normality Test)							
	Original Monte Carlo				BML / σ^2 Smoothing			
	W	A^2	U^2	W^2	W	A^2	U^2	W^2
10	0.43	0.41	0.40	0.41	0.46	0.44	0.43	0.44
20	0.46	0.44	0.43	0.44	0.48	0.46	0.45	0.46
50	0.47	0.44	0.44	0.45	0.48	0.46	0.45	0.46
100	0.47	0.45	0.44	0.46	0.48	0.46	0.45	0.46
200	0.48	0.46	0.46	0.47	0.50	0.47	0.47	0.48
500	0.50	0.47	0.47	0.48	0.51	0.49	0.48	0.49

Table C.9: Estimated N Values Required for Normality

Number of Values Used in Normality Tests	10	100
Original Monte Carlo	20 to 100	200
BML Estimation	10 to 20	100
BML / σ^2 Smoothed	10	50

ity estimation, tends to “fill in” the distribution of a result causing the distribution to appear closer to normal. The changes in η caused by smoothing have the same effect on the reciprocity estimated results.