SNO-STR-90-7

Advantages of Weight Splitting in Deep Penetration γ-Ray Calculations

E.T.H. CLIFFORD

January 9, 1990

1 Introduction

Calculations of γ-ray shielding which employ the Monte Carlo method run into severe limitations when the attenuation reaches many orders of magnitude. In order to achieve any useful statistical accuracy a large number of histories must be followed in order that a reasonable number of γ-rays survive. As an example, if an attenuation of 10^{-6} is to be calculated one needs to follow 10^{8} histories in order to expect a statistical accuracy of 10%. This is difficult to achieve in a reasonable CPU time, and an attenuation of 10^{-9} is prohibitive.

The method of weight splitting has been proposed to circumvent this problem and for the simple shielding calculation required for the SNO detector it is admirably suited. This method avoids the expenditure of a large amount of CPU time on following uninteresting histories – those many many histories which simply terminate by the absorbtion of the photon or its scattering to low energy.

Weight splitting is a technique which must be used with some care. It is possible to design problems where misleading results may be obtained. However, for problems where the material is uniform, such as the SNO detector, there should not be any difficulty. The technique of weight splitting does not introduce bias into the results: for a sufficiently large number of histories the weight-split problem converges to the same answer as the normal calculation.

2 CPU Time Considerations

The standard monte carlo method works by comparing the number of photons started with the number which survive transport through the absorbing material. For severe attenuations it is clear that most photons will not make it through the material. And yet it is precisely these photons which do not make it that take up most of the CPU time.

The probability of survival may be described as a weighted integral over the entire phase space of photon trajectories which begin at the starting point and end up at the other side of the radiation shield. This integral depends on the exponential nature of survival as well as the geometry of phase space. The monte carlo problem evaluates this integral by the method of random sampling in phase space. This is particularly inefficient with respect to CPU time, at least in the standard monte carlo, because the sampling scheme used emphasizes an uninteresting region of phase space–namely the part that leads to photons which are absorbed. The efficient use of CPU time requires that the sampling be concentrated in the region where photons have a reasonable chance to survive.

As an example consider trajectories through a one-dimensional space where a pure exponential law applies with a known absorbtion coefficient, k. The monte carlo problem could be set up with a number of equally spaced intervals say of length q. This would lead to a probability of survival through each interval of $\exp(-kq)$ and the calculation could be set up to sample this probability as the photon passes each interval. The number of photons surviving the entire thickness, say x, then gives the desired probability (with some statistical error). However this is CPU time wasteful for the simple reason that the answer to the problem is known *exactly*, namely, $\exp(-kx)$. So why do a monte carlo calculation for this problem? In this case a monte carlo calculation is pointless. However, when the number of spatial (and other) dimensions increase, the problem no longer yields a simple analytical solution. Here the method of sampling the space can be helpful but the cost is to introduce statistical uncertainty into the result.

It is already known that the radiation follows an exponential or near exponential law. The object of the calculation is to investigate the more subtle effects caused by the many possible trajectories in phase space that can be followed by the photons that do survive. Therefore it seems reasonable to emphasize these kinds of histories in the calulation.

3 Weight Splitting

The weight splitting method works as follows. The space that a photon traverses is divided into zones with weights to be set appropriate to the problem. A photon which survives a zone and passes to the next zone is allowed to generate two (or more) duplicate photons each of which have half (or the appropriate fraction of) the weight of the original. This leads to an exponential like decrease in the statistical weight of the surviving photons. At the same time, if these zones and weights are set up correctly, the probability of a photon surviving the absorber is very good. This allows the possibility of sampling many trajectories which survive , but now the exponential aspect of the problem is built into the scheme of weights. In effect the monte carlo calculation determines the *deviation* from the "exponential" law built into the weights. If the weights are near to the actual exponential behaviour then the calculation will be efficient in its use of CPU time.

4 Estimation of the Variance

In this section the variance is estimated under the assumption that the attenuation is purely exponential. One layer is considered first; then several layers, and finally, several layers with weight splitting.

4.1 One Layer

Let N γ-rays start from A and see how many reach B. Let the probability for a single γ-ray to get through the layer of material to B be α. Then the probability of n γ-rays reaching A is given by the binomial distribution:

$$P(n) = \binom{N}{n} \alpha^n (1-\alpha)^{N-n} \tag{1}$$

The mean number of photons reaching b is

$$< n > = \sum P(n)\, n = \alpha N \tag{2}$$

The variance in $< n >$ is

$$\sigma^2 = < n^2 > - < n >^2 = N\alpha(1-\alpha) \tag{3}$$

These last two equations are well known properties of the binomial distribution.

Now consider the same layer but this time let α be unknown. If we want to determine α by measurement, we could start N γ-rays at A and see how many reach B, calling this number n. The from equation (2), a measure of α is

$$\alpha = \frac{n}{N} \tag{4}$$

There will be an uncertainty $\delta\alpha$ in the measurement because of the variance in n. Thus, using equation (3),

$$\frac{\delta\alpha}{\alpha} = \frac{\delta n}{n} = \frac{\sigma}{n} = \frac{\sqrt{N\alpha(1-\alpha)}}{n} = \frac{\sqrt{N\alpha(1-\alpha)}}{N\alpha} = \sqrt{\frac{1-\alpha}{N\alpha}} \tag{5}$$

4.2 Several Layers

Now consider several layers, first without weight splitting: assume the attenuation in each layer is α; the overall attenuation is $\alpha^n = a$. To determine the variance in the overall attenuation consider two methods:

i) layer by layer:

1^{st} layer as above:

$$\alpha_1 = \frac{n_1}{N} \quad \frac{\delta\alpha_1}{\alpha_1} = \sqrt{\frac{1-\alpha_1}{N\alpha_1}}$$

2^{nd} layer:

$$\alpha_2 = \frac{n_2}{n_1} \quad \frac{\delta\alpha_2}{\alpha_2} = \sqrt{\frac{1-\alpha_2}{n_1\alpha_2}} = \sqrt{\frac{1-\alpha_2}{N\alpha_1\alpha_2}} \tag{6}$$

$$\alpha_3 = \frac{n_3}{n_2} \quad \frac{\delta\alpha_3}{\alpha_3} = \sqrt{\frac{1-\alpha_3}{N\alpha_1\alpha_2\alpha_3}}$$

etc.

Then a is determined by $a = \prod \alpha_i$. If all layers are the same thickness $\alpha_i = \alpha\ \forall i$ and $a = \alpha^k$, and

$$
\begin{aligned}
(\frac{\delta a}{a})^2 &= (\frac{\delta \alpha_1}{\alpha_1})^2 + (\frac{\delta \alpha_2}{\alpha_2})^2 + \cdots + (\frac{\delta \alpha_k}{\alpha_k})^2 \\
&= \frac{1-\alpha_1}{N\alpha_1} + \frac{1-\alpha_2}{N\alpha_1\alpha_2} + \cdots + \frac{1-\alpha_k}{N\alpha_1\alpha_2\ldots\alpha_k} \\
&= \frac{1-\alpha}{N\alpha} + \frac{1-\alpha}{N\alpha^2} + \cdots + \frac{1-\alpha}{N\alpha^k} \\
&= \frac{1-\alpha}{N\alpha}(1 + \frac{1}{\alpha} + \frac{1}{\alpha^2} + \ldots + \frac{1}{\alpha^{k-1}}) \\
&= \frac{1-\alpha}{N\alpha}\frac{(\frac{1}{\alpha^k} - 1)}{(\frac{1}{\alpha} - 1)} \\
&= \frac{1-\alpha^k}{N\alpha^k} \\
&= \frac{1-a}{Na} \qquad (7) \\
\frac{\delta a}{a} &= \sqrt{\frac{1-a}{Na}} \qquad (8)
\end{aligned}
$$

ii) consider one thick layer with attenuation, a, then,

$$\frac{\delta a}{a} = \sqrt{\frac{1-a}{Na}}$$

which gives the same result as before.

4.3 With Weight Splitting

Now consider weight splitting: at each boundary, if n photons arrive, then send gn photons to the next layer. Analyse this layer by layer:

1^{st} layer as above:

$$\alpha_1 = \frac{n_1}{N} \qquad \frac{\delta\alpha_1}{\alpha_1} = \sqrt{\frac{1-\alpha_1}{N\alpha_1}}$$

2^{nd} layer:

$$
\begin{aligned}
\alpha_2 &= \frac{n_2}{gn_1} = \frac{n_2}{N\alpha_1 g} & \frac{\delta\alpha_2}{\alpha_2} &= \sqrt{\frac{1-\alpha_2}{gn_1\alpha_2}} = \sqrt{\frac{1-\alpha_2}{Ng\alpha_1\alpha_2}} \\
\alpha_3 &= \frac{n_3}{gn_2} = \frac{n_3}{N\alpha_1\alpha_2 g^2} & \frac{\delta\alpha_3}{\alpha_3} &= \sqrt{\frac{1-\alpha_3}{gn_2\alpha_3}} = \sqrt{\frac{1-\alpha_3}{N\alpha_1\alpha_2\alpha_3 g^2}} \\
& & \frac{\delta\alpha_k}{\alpha_k} &= \sqrt{\frac{1-\alpha_k}{N\alpha_1\alpha_2\ldots\alpha_k g^{k-1}}}
\end{aligned} \qquad (9)
$$

If all the α_i's are equal then

$$\frac{\delta\alpha_k}{\alpha_k} = \sqrt{\frac{1-\alpha}{N\alpha(g\alpha)^{k-1}}} \qquad (10)$$

The overall attenuation is $a = \alpha_1\alpha_2\alpha_3\ldots\alpha_k$, so

$$(\frac{\delta a}{a})^2 = (\frac{\delta\alpha_1}{\alpha_1})^2 + (\frac{\delta\alpha_2}{\alpha_2})^2 + \cdots + (\frac{\delta\alpha_k}{\alpha_k})^2 = \frac{1-\alpha}{N\alpha} + \frac{1-\alpha}{N\alpha g\alpha} + \frac{1-\alpha}{N\alpha(g\alpha)^2} + \cdots + \frac{1-\alpha}{N\alpha(g\alpha)^{k-1}}$$

$$= \frac{1-\alpha}{N\alpha}\left(1 + \frac{1}{g\alpha} + \frac{1}{(g\alpha)^2} + \cdots + \frac{1}{(g\alpha)^{k-1}}\right)$$

$$= \begin{cases} \dfrac{1-\alpha}{N\alpha} \dfrac{\frac{1}{(g\alpha)^k} - 1}{\frac{1}{g\alpha} - 1} & \text{for } g\alpha \neq 1 \\ \dfrac{k(1-\alpha)}{N\alpha} & \text{for } g\alpha = 1 \end{cases} \tag{11}$$

With $\alpha^k = a$,

$$\left(\frac{\delta a}{a}\right)^2_w = \begin{cases} \dfrac{1-a^{1/k}}{N_w a^{1/k}} \dfrac{\frac{1}{g^k a} - 1}{\frac{1}{g a^{1/k}} - 1} & \text{for } g a^{1/k} \neq 1 \\ \dfrac{k(1-a^{1/k})}{N_w a^{1/k}} & \text{for } g a^{1/k} - 1 \end{cases} \tag{12}$$

(The subscript, w, has been introduced to indicate that this is specifically the weight splitting case.)

5 Minimizing the Variance by Maintaining the Flux

It is possible, using the scheme outlined in the previous section, to imagine a number of strategies for setting the weights. If all the weights are fixed equal (to one) then we have the standard monte carlo with its limitations for deep penetration calculations. Another possibility is to set the weights so that the number of photons followed increases with distance through the attenuator. This strategy suffers from the flaw that all the CPU time is merely shifted from the incident side of attenuator to the exit side and therefore still ignores the important region of the phase space.

The best strategy is to choose weights such that the flux is maintained approximately constant throughout the attenuator and this can be proved as follows.

5.1 Proof that the Variance is Minimized by Maintaining the Flux

Let the attenuator be divided into equal length zones and let the weights be set so that the ratio of weights between adjacent zones be constant. This corresponds to an exponential growth in the number of photons followed if there is no attenuation. If the multiplication factor associated with each zone change (determined by the ratio of weights) is g and the attenuation for the zone is α, then after n zones the probable number of photons followed is $(g\alpha)^n$ for each photon started. This number can grow, attenuate or be constant depending on whether $g\alpha$ is respectively greater than one, less than one, or equal to one.

5.1.1 Estimate of CPU Time

When weight splitting is used, the CPU time required for each started photon is greater than when weight splitting is not used. This is simply because more photons must be followed. However, since weight splitting improves the variance on the measured transmittance, it seems reasonable to imagine that the variance might be reduced even when the CPU time is not increased, and this conjecture is correct.

Were it not, then weight splitting would have no advantage over simply increasing the number of started photons.

The CPU time required for a problem is proportional to the number of photons transported and to some approximation it is also proportional to the mean path length of the photons:

$$t = N_0 \frac{\int_0^x s e^{-\lambda s}\, ds}{\int_0^x e^{-\lambda s}\, ds} = \frac{N_0}{\lambda}\{1 - \lambda x \frac{e^{-\lambda x}}{1 - e^{-\lambda x}}\} \tag{13}$$

for a layer of thickness x.

Since the attenuation in this layer is $\alpha = e^{-\lambda x}$,

$$t = \frac{N_0}{\lambda}\{1 + \frac{\alpha}{1-\alpha} \ln \alpha\} \tag{14}$$

If there are k layers then the overall attenuation is $a = \alpha^k$ and the time for the first layer is

$$t = \frac{N_0}{\lambda}\{1 + \frac{a^{1/k}}{1 - a^{1/k}} \frac{1}{k} \ln a\}. \tag{15}$$

5.1.2 Time with Weight Splitting

With weight splitting, as each boundary is crossed g photons are generated. The time for the 1^{st} layer is t, given in equation 15, above. The time for the second layer would be αt since only αN_0 photons survive the first layer, but with weight splitting this becomes $(\alpha g)t$; the 3^{rd} layer requires $(\alpha g)^2 t$, *etc.* The total time is

$$T = t + \alpha g t + (\alpha g)^2 t + \cdots + (\alpha g)^{k-1} t = \begin{cases} t\,\dfrac{1-(\alpha g)^k}{1-\alpha g} & \text{for } g\alpha \neq 1 \\ tk\alpha g & \text{for } g\alpha = 1 \end{cases} \tag{16}$$

Substituting $a = \alpha^k$ and equation 15 gives,

$$T = \begin{cases} \dfrac{N_0}{\lambda} f(k) \dfrac{1 - ag^k}{1 - ga^{1/k}} & \text{for } ga^{1/k} \neq 1 \\ \dfrac{N_0}{\lambda} k f(k) & \text{for } ga^{1/k} = 1 \end{cases} \tag{17}$$

where

$$f(k) = 1 + \frac{1}{k} \frac{a^{1/k}}{1 - a^{1/k}} \ln a \tag{18}$$

From this can be deduced the number of photons that can be transported for a given CPU time, T:

$$N_w = \begin{cases} \dfrac{\lambda T}{f(k) \frac{1 - ag^k}{1 - ga^{1/k}}} & \text{for } ga^{1/k} \neq 1 \\ \dfrac{\lambda T}{k f(k)} & \text{for } ga^{1/k} = 1 \end{cases} \tag{19}$$

The number of photons transported, N, in the same time, T, when no weight splitting is used can be found from equation 18 by putting $g = 1$ and thus the ratio of photons transported in time,T, is

$$\frac{N_W}{N} = \begin{cases} \frac{1 - ga^{1/k}}{1 - ag^k} \frac{1 - a}{1 - a^{1/k}} & \text{for } ga^{1/k} \neq 1 \\ \frac{1}{k} \frac{1 - a}{1 - a^{1/k}} & \text{for } ga^{1/k} = 1 \end{cases} \quad (20)$$

The variance with weight splitting is given in equation 12 where N_w photons were started. This can be written in terms of N, the number of photons for the no weight splitting case, by substituting equation 19 into equation 12. Thus for the same CPU time which would give a variance given by equation 10 (no weight splitting), the variance obtained with weight splitting is

$$(\frac{\delta a}{a})^2_w = \begin{cases} \frac{1 - a}{Na} \{ \frac{1 - a^{1/k}}{1 - a} \frac{1 - ag^k}{1 - ga^{1/k}} \}^2 \frac{1}{g^{k-1}} & \text{for } ga^{1/k} \neq 1 \\ \frac{1 - a}{Na} \{ \frac{k(1 - a^{1/k})}{1 - a} \}^2 \frac{a}{a^{1/k}} & \text{for } ga^{1/k} = 1 \end{cases} \quad (21)$$

The ratio of errors with and without weight splitting for the same CPU time is obtained from equations 20 and 10:

$$\frac{(\frac{\delta a}{a})_W}{(\frac{\delta a}{a})} = \begin{cases} \frac{1 - a^{1/k}}{1 - a} \frac{1 - ag^k}{1 - ga^{1/k}} \frac{1}{\sqrt{g^{k-1}}} & \text{for } ga^{1/k} \neq 1 \\ \frac{k(1 - a^{1/k})}{1 - a} \sqrt{\frac{a}{a^{1/k}}} & \text{for } ga^{1/k} = 1 \end{cases} \quad (22)$$

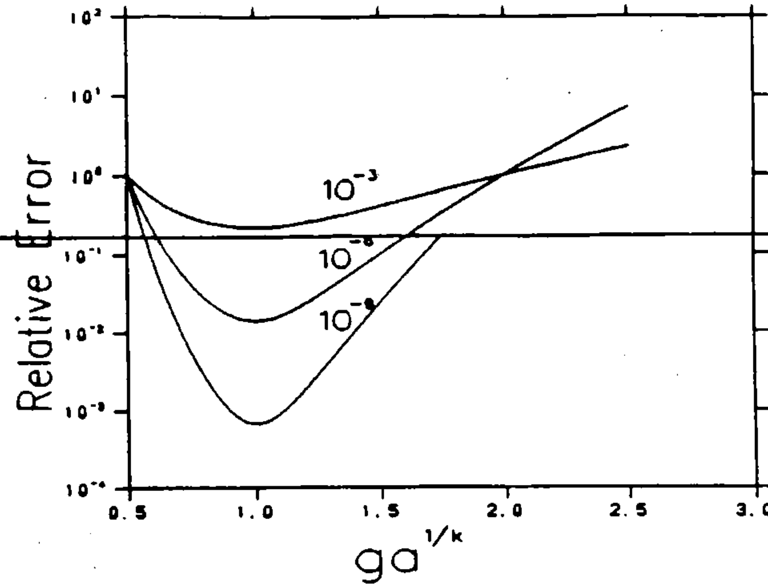
5.1.3 Minimum Variance

The minimum variance for a given CPU time can be found by taking the derivative of equation 22 with respect to g and setting it to zero. $ga^{1/k} = 1$ is a solution. This is shown graphically in Figure 1. The ratio of errors given by equation 22 is plotted against $ga^{1/k}$ for three values of a: 10^-3, 10^-6, and 10^-9. The minimum is at $ga^{1/k} = 1$. With this value of $ga^{1/k}$ the gain in photon number exactly compensates the exponential decay so that the flux is maintained constant.

It can be seen from the figure that the improvement in the errors can be substantial, particularly when the attenuation is many orders of magnitude. It should be noted that for a given error the improvement in CPU time is the square of the quantity plotted in Figure 1.

The number of layers into which the problem is divided also influences the variance reduction. A good method is to chosse k such that each layer attenuates by a factor of two, and then set $g = 2$. In this case Equation 22 becomes

$$\frac{(\frac{\delta a}{a})_W}{(\frac{\delta a}{a})} \approx 0.7k\sqrt{a} \quad (23)$$



6 Conclusions

The method of weight splitting provides an enormous gain in the amount of CPU time required to solve γ-ray attenuation problems. The optimum scheme for setting the weights is to set them so that the splitting compensates for the exponential attenuation, thus maintaining a constant flux of photons. Of course, weight splitting reduces only the *statistical* error in the calculation; it does not improve systematic errors.