LA-UR-97-0919

Title:

AVATAR - Automatic Variance Reduction in Monte Carlo Calculations

Author(s):

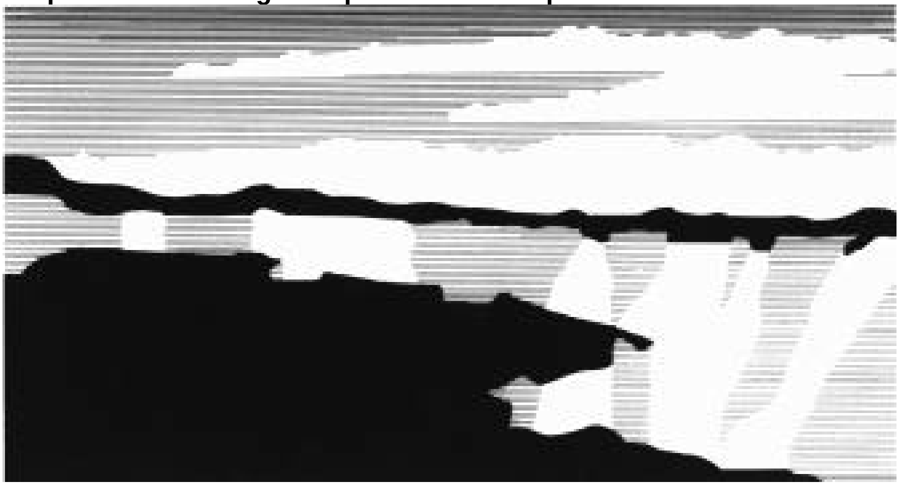
Kenneth A. Van Riper, White Rock Science

Todd J. Urbatsch, Patrick D. Soran, Kent Parsons, Jim E. Morel, Gregg W. McKinney, Stephen R. Lee, Laura A. Crotzer, Raymond Alcouffe, Los Alamos National Laboratory

Forest W. Brinkley, Thomas E. Booth, Jacob Anderson, BBN Systems & Technologies

Submitted to:

http://lib-www.lanl.gov/la-pubs/00412792.pdf



AVATAR – AUTOMATIC VARIANCE REDUCTION IN MONTE CARLO CALCULATIONS

Kenneth A. Van Riper[1], Todd J. Urbatsch, Patrick D. Soran, D. Kent Parsons[2],
Jim E. Morel[3], Gregg W. McKinney, Stephen R. Lee, Laura A. Crotzer[4],
Forrest W. Brinkley, Thomas E. Booth, Jacob W. Anderson[5], and Raymond E. Alcouffe

XTM, MS B226
University of California
Los Alamos National Laboratory
Los Alamos, New Mexico 87544
xtm@lanl.gov

1. White Rock Science, P.O. Box 4729, Los Alamos, NM 87544
2. ESH-6, MS F691, Los Alamos National Laboratory
3. CIC-19, MS B265, Los Alamos National Laboratory
4. XHM, MS F663, Los Alamos National Laboratory
5. BBN Systems & Technologies, 9655 Granite Ridge Dr., Suite 245, San Diego, CA 92123

ABSTRACT

AVATAR$^{\mathrm{TM}}$ (Automatic Variance And Time of Analysis Reduction), accessed through the graphical user interface application Justine$^{\mathrm{TM}}$, is a superset of MCNP$^{\mathrm{TM}}$ that automatically invokes THREEDANT$^{\mathrm{TM}}$ for a three-dimensional deterministic adjoint calculation on a mesh independent of the Monte Carlo geometry, calculates weight windows, and runs MCNP. Computational efficiency increases by a factor of 2 to 5 for a three-detector oil well logging tool model. User efficiency increases dramatically, since AVATAR eliminates the need for deep intuition and hours of tedious data manipulation.

I. INTRODUCTION

Monte Carlo is frequently the method of choice in transport calculations because it handles complex geometries particularly well. Unfortunately, these complex geometries are often computationally difficult. Two types of difficult calculations are oil well logging and radiation shielding simulations. Oil well logging tools unobtrusively probe the earth surrounding a borehole by sending neutrons or photons into the surrounding formation and measuring the return signal. Radiation shields minimize the flow of radiation through them. The desired quantity, whether the logging tool's return signal or the radiation flowing through a shield, is orders of magnitude smaller than the source. Since the number of Monte Carlo particles simulated is also many orders of magnitude smaller than the actual number of particles, the tally of the particles contributing to the return signal or escaping radiation is small and, thus, unreliable.

Reducing the variance on these desired tallies is vital for a reliable result. One common variance reduction technique is *weight windows.* Weight windows depend on importance functions and allow the Monte Carlo to concentrate on "important" particles, particles that eventually contribute to the tally. Although variance reduction techniques are sometimes necessary and may be quite successful, they may also be difficult to properly employ. For instance, the user must often further subdivide the cells for sufficient resolution of the importance function. This additional refinement requires extra time, experience, and intuition to properly implement.

For many years, the solution of an adjoint calculation has been recognized as an importance function for the regular forward calculation.[1] If the exact adjoint solution is known, the forward solution is known. Of course, an adjoint solution of the same precision as a forward solution would take just as long to compute as the forward solution. However, the adjoint solution from a less precise calculation provides an importance function that may sufficiently accelerate the forward Monte Carlo calculation. Thus, one performs a deterministic adjoint calculation with relatively few energy groups and low angular resolution, converts by hand the adjoint solution to weight windows, and uses the weight windows in a subsequent Monte Carlo calculation. This practice of obtaining weight windows from an adjoint calculation has two disadvantages: the user must be proficient in running both deterministic and Monte Carlo codes (typically not the case), and the user must handle a copious amount of weight window data by hand.

AVATAR is a software package for automatically accelerating Monte Carlo calculations with weight windows. AVATAR's weight windows are fully three dimensional, separate from the Monte Carlo cells, and dependent upon space, energy, and angle. In practice, the user sets up the problem with the graphical user interface application Justine, and Justine runs AVATAR. AVATAR superimposes a weight window mesh on the Monte Carlo cells, runs the three-dimensional deterministic code THREEDANT in adjoint mode, constructs the weight windows from the adjoint solution, and runs the three-dimensional Monte Carlo transport code MCNP.[2, 3, 4] If the user simply defines the problem and asks for the AVATAR option, Justine will select default parameters and execute the process in a fully automatic manner. However, the user has the option to intervene and set some problem-specific parameters if the defaults do not suffice.

II. THE ADJOINT FLUX

The adjoint flux is a particle's importance to the response of a detector.[5] In an adjoint calculation, the detector and source from the regular forward calculation reverse roles. The particles emanate from the forward detectors—now the adjoint sources—and transport backwards. The adjoint calculation may be performed stochastically with Monte Carlo or deterministically with discrete-ordinates transport or diffusion. For an adjoint solution intended for importances in a forward calculation, a Monte Carlo adjoint calculation may provide too much detail and cost just as much as a forward calculation. A deterministic adjoint transport calculation with few energy groups and relatively low angular resolution gives inexpensive, yet sufficient, importances dependent upon space, energy, and, possibly, angle. If transport effects are not prominent in the calculation, a diffusion adjoint calculation may be adequate.

III. CONVENTIONAL WEIGHT WINDOWS IN MCNP

The conventional weight windows in MCNP are either input by the user or constructed by the user with MCNP's Weight Window Generator.

In each Monte Carlo cell, the lower conventional weight window boundary is inversely proportional to the importance. The upper boundary is a multiple of the lower boundary, and both boundaries are normalized so source particles are emitted within the window. During transport, if a particle's weight is above the weight window, it is split into enough particles, each with an equal fraction of the original weight, such that the new weights are within the weight window. Splitting is unbiased since it conserves total weight. If a particle's weight is below the weight window, it is Russian rouletted. With appropriate probability, the particle is either killed, or its weight is increased to a weight within the weight window. Russian roulette is a statistically unbiased way to reduce low importance particles. Conventional weight windows spatially depend on the Monte Carlo cells and are intimately involved in MCNP's tracking. The weight windows are applied at collisions and cell boundary crossings.

With weight windows, more particles are put in locations and energies that are important for eventual contribution to the tally, and fewer particles are considered in unimportant locations and energies. Thus, weight windows reduce the tally variance.

The MCNP Weight Window Generator[6] calculates, during a forward run, a set of weight windows for a subsequent run. For each phase space (position and energy) bin, the importance is determined as the ratio of the total tally contribution due to particles and their progeny entering the bin to the total weight entering the bin. In typical usage, the user supplies an initial set of weight windows; multiple runs of MCNP are made, with the output of the weight window generator used as the weight windows for the next run.

The generator, while useful, is cumbersome and cannot overcome a bad set of initial weight windows. Unreliable weight windows may result if a region is sampled poorly (or not at all). Because the weight windows are generated statistically, smooth variation from cell to cell is not guaranteed and the user must scrutinize and, possibly, smooth the weight windows by hand. There is no guarantee that a given set of weight windows is useful, let alone optimum.

IV. OTHER METHODS AND OTHER CODES

We briefly describe other methods and other codes that employ the adjoint solution in some form of variance reduction.

The TRIPOLI–4 Monte Carlo code[7] uses several weighting schemes (exponential biasing, quota sampling, and collision biasing), which require an importance function defined on a mesh. The mesh can be refined based on a spatial window or specified volumes. The importance function may be constructed using graph theory applied to the mesh[8] or from a collision probability approximation to the adjoint Boltzmann equation. Both methods are based on cross sections collapsed to a coarse energy grid specified by the user. The collapsed cross sections are transport corrected to account for deviations from diffusive flow.

The MCBEND Monte Carlo code[9] makes use of geometric importance splitting and Russian roulette based on energy-dependent importances defined on a user-defined orthogonal XYZ or $RZ\Theta$ mesh overlying the geometry. (Geometric splitting and Russian roulette are similar to weight window. It relies on regional importances and is applied when crossing region boundaries.) The importances are calculated through the use of an adjoint diffusion calculation, which is usually executed as an automatic part of a MCBEND run.[10] The diffusion coefficients are modified to correct for known shortcomings of the diffusion calculation in deep penetration problems.[11]

The multigroup Monte Carlo code MORSE may be run in adjoint mode, and that solution may then be used to accelerate forward runs. A version for a class of shielding calculations, MORSE–SGC/S, makes use of an automatic one-dimensional adjoint discrete ordinates calculation.[12]

Mickael has developed an an automatic method for generating weight windows for MCNP based on the solution of a one- or three-dimensional adjoint diffusion calculation.[13] For both cases, the mesh is tuned to oil well logging applications.

Wagner and Haghighat have a modified version of MCNP that reads the two-dimensional adjoint scalar fluxes from the transport code DORT and converts them to weight windows.[14] Three-dimensionality comes from an assumed analytic shape function. The importance function has no angular dependence and the process is not automated.

Liu and Gardner have modified MCNP to run its weight window generator on a fine mesh that is independent of the Monte Carlo geometric cells.[17]

Dubi and Burn's "direct statistical approach" (DSA) method[18] is implemented in a special version of MCNP4A. In DSA, the user divides the problem into space-energy regions. A spatial portion of the region is defined as the union of one or more MCNP geometry cells. DSA explicitly attempts to minimize the cost,

$\sigma^2 T$, of the calculation by considering the computer time, T, as well as the variance, σ^2. AVATAR and the other methods here seek to minimize only the variance. Like the weight window generator in MCNP, DSA requires user intervention to hand-adjust importances (or windows) in regions that were poorly sampled. However, checks on the statistical quality of the information make such intervention infrequent.

V. AVATAR

The AVATAR package is a superset of MCNP embodied in the graphical user interface, Justine. Upon problem specification, AVATAR automatically generates a mesh for the adjoint calculation. With no further user intervention, Justine performs an adjoint THREEDANT calculation with this mesh, constructs three-dimensional energy-dependent weight windows from the adjoint solution, and runs MCNP. Prior to running AVATAR, the user may choose to have the weight windows depend upon angle as well. Also, the user may intervene and edit the weight window mesh after Justine generates it.

A. Mesh Description

The mesh for the adjoint calculation with THREEDANT is not connected to the MCNP cells, so its refinement does not directly affect the Monte Carlo tracking. The mesh is orthogonal in either a rectangular (XYZ) or cylindrical ($RZ\Theta$) coordinate system and consists of coarse mesh intervals containing fine mesh intervals. When used in AVATAR, MCNP obtains weight window data from this orthogonal mesh rather that its own cells. The material of the mesh element is the material at the center of the mesh element; a more sophisticated approach is planned for the future.

B. Automatic Mesh Generation

Justine automatically generates the mesh for THREEDANT's adjoint calculation. It begins with a bounding box that contains the entire system. In each direction, the bounding box is divided into coarse mesh elements. The uniform coarse mesh is generated by dividing the bounding box uniformly in each direction. The non-uniform, or "smart," mesh is generated by subdividing the bounding box into smaller bounding boxes that bound each MCNP cell in each direction. Although the "smart" mesh is not directly connected to the MCNP cells, it is designed to resolve the geometry appropriately. A representative mesh is shown in Figure 1 for the "smart" mesh generation. Justine defaults to the "smart" mesh, although the user may intervene to select uniform coarse meshes.

Once the coarse meshes are calculated, they may be either uniformly or logarithmically subdivided into fine meshes. AVATAR defaults to uniform fine meshes, but the user may intervene and select logarithmic fine meshes. We use uniform instead of logarithmic fine mesh spacing because it is slightly faster when MCNP is looking up weight window information.

Without intervention, AVATAR continues after either the uniform or "smart" mesh has been generated. However, the user may choose to intervene and add or delete mesh lines as needed. Justine provides a user interface to handle these mesh edits.

C. Deterministic Adjoint Calculation

THREEDANT performs the deterministic discrete-ordinates transport calculation. AVATAR runs THREEDANT with a lower quadrature order and fewer energy groups than a typical stand-alone THREEDANT calculation. A low resolution deterministic adjoint calculation executes quickly and still accelerates the Monte Carlo calculation, resulting in increased overall efficiency. Thus, we used a quadrature of S_4, a P_2 Legendre scattering moment expansion, and three energy groups for our calculations. THREEDANT produces a scalar adjoint flux and an average current for each mesh element and each energy group. The user may intervene and have AVATAR run THREEDANT as an adjoint diffusion calculation if the diffusion approximation is adequate (e.g., no streaming). In that case, THREEDANT produces no average currents, only the scalar adjoint fluxes.

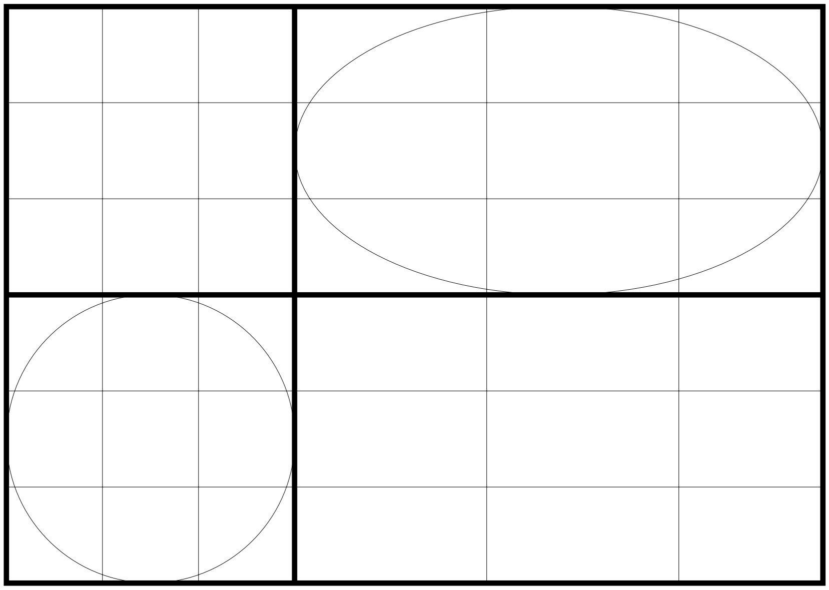


Figure 1: A representative "smart" adjoint/weight window mesh overlaid on MCNP cells. The coarse mesh is shown in thick lines, the fine mesh in thin straight lines.

D. Weight Windows

AVATAR inverts the scalar adjoint fluxes to get the lower weight window boundary in each mesh element and each group. The weight window is then normalized so the source particles are born with weights inside the weight window. Regions of phase space with high scalar adjoint fluxes relate to a high importance, which in turn relate to a low weight window. Low weight windows tend to split more particles, so more lower weight particles occupy those regions, thus sampling the region better. The local particle weight variance and, ultimately, the tally variance decrease.

E. Angle-Dependent Weight Windows

Using the full angular adjoint flux as the importance function produces weight windows dependent upon space, energy, and each discrete angle in the S_N adjoint calculation. Since this level of detail requires an inordinate amount of storage, we approximate the adjoint angular flux using information theory.[15, 16] First, we assume that the adjoint angular flux, ψ^*, is symmetric about the average current vector, $\mathbf{J}^*$,

$$\psi^*(\mathbf{\Omega}) = \psi^*(\mathbf{\Omega} \cdot \mathbf{n}) \quad , \tag{1}$$

where $\mathbf{n}$ is the normalized current vector,

$$\mathbf{n} = \frac{\mathbf{J}^*}{||\mathbf{J}^*||} \quad . \tag{2}$$

Physically, this assumption implies that the adjoint angular flux is locally one-dimensional. Information theory tells us that we may approximate $\psi^*(\mathbf{\Omega} \cdot \mathbf{n})$ by the maximum entropy distribution,

$$\psi^*(\mathbf{\Omega} \cdot \mathbf{n}) = \phi^* \mathbf{f}(\mathbf{\Omega} \cdot \mathbf{n}) \quad , \tag{3}$$

where ϕ^* is the adjoint scalar flux, and f is given by

$$f(\mathbf{\Omega} \cdot \mathbf{n}) = \frac{\lambda \mathrm{e}^{(\mathbf{\Omega} \cdot \mathbf{n})\lambda}}{\mathbf{2} \sinh(\lambda)} \quad . \tag{4}$$

The parameter λ is a function of the average cosine, $\bar{\mu}$, and is approximated as

$$\lambda = \frac{2.99821\bar{\mu} - 2.2669248\bar{\mu}^2}{1 - 0.769332\bar{\mu} - 0.519928\bar{\mu}^2 + 0.2691594\bar{\mu}^3} \tag{5}$$

for $0 \leq \bar{\mu} < 0.8001$, and

$$\lambda = \frac{1}{1-\bar{\mu}} \tag{6}$$

for $0.8001 \leq \bar{\mu} < 1.0$, where the average cosine is given by

$$\bar{\mu} = \frac{||\mathbf{J}^*||}{\phi^*} \quad . \tag{7}$$

With angle-dependent weight windows, preference is given to particles not only in important locations and energies, but also going in important directions. Those particles heading in unimportant directions are more likely killed.

Our approximation of ψ^* is exact in both the isotropic and the streaming limits. If the adjoint angular flux is isotropic, the weight windows, in turn, have no angular dependence. If the adjoint angular flux is a delta function in angle, the corresponding weight window would try to kill every particle not going in the exact desired direction. We limit the average cosine to smooth out the delta function and avoid numerical difficulty.

Our angular weight window approach is not as efficient as using the angular information to directly bias the source particles. We are working on this direct biasing.

F. Mean Free Path Sampling

Since AVATAR's weight window mesh is not involved in the Monte Carlo tracking, we only apply the weight windows at absolute particle positions, not weight window boundary crossings. Particles with long paths may travel over many weight window meshes without being checked. Therefore, even with spatially resolved weight windows, particles may arrive in a location where their weights are significantly outside the local weight window. In this case, variance in particle weights increases. Whenever the variance increases along the path of particles, subsequent weight window applications cannot correct the increase, so the tally variance increases as well. We protect against this pitfall by applying the weight windows at least every mean free path that the particles travel.

VI. TEST PROBLEM

We applied AVATAR to a three-detector oil well logging tool model. The tool consists of a 0.662 MeV gamma-ray source and three gadolinium (Gd) detectors, labeled *near*, *middle*, and *far* with respect to their distance from the source. The system is a deep-penetration problem since only a small fraction of the gammas emitted into the surrounding formation reach the detectors. The tool also has angular dependence due to a collimator on one of the detectors. A qualitative diagram of the tool is shown in Figure 2.

A. Figure of Merit

We measure the efficiency of a Monte Carlo calculation by the figure of merit (FOM),

$$\mathrm{FOM} = \frac{1}{R^2 T} \quad , \tag{8}$$

where R is the estimated relative error for a tally and T is the computer time. Since computer time increases with the number of particles, and, by the central limit theorem, the relative error decreases with the square root of the number of particles, the FOM remains constant with the number of particles. Thus the FOM allows comparison of calculations with different numbers of particles (but same computer speed).

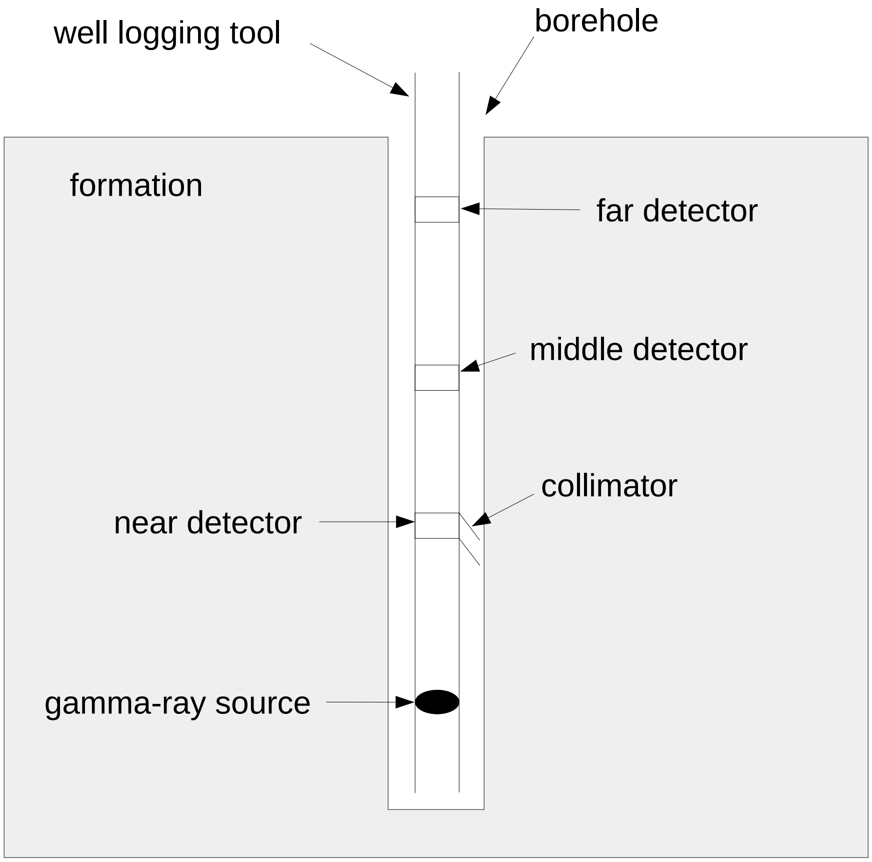


Figure 2: Idealized three-detector oil well logging tool.

B. Adjoint Source Spectrum Weighting

We ran THREEDANT with three energy groups with boundaries of 1.0, 0.5, 0.1, and 0.0 MeV. The adjoint source is the forward detector response, so to get an idea of adjoint source spectrum weighting, we need the forward detector response. An estimate of the forward detector response is the energy deposition in the detector,

$$E_D = \Sigma_t \phi \Sigma_h \quad , \tag{9}$$

where Σ_t is the total cross section, ϕ is the scalar flux, $\Sigma_t \phi$ is the collision rate, and Σ_h is the heating cross section in MeV per collision. We evaluated the cross sections of the forward detector material, gadolinium, at the energy of the forward source, 0.662 MeV. We evaluated Equation 9 by replacing the unknown flux with a flat weighting function. From highest to lowest energy group, the energy deposition rates of 8, 20, and 70 suggested an adjoint source spectrum weighting of 1, 2, and 10.

VII. RESULTS AND DISCUSSION

We first present results for the full three-detector problem. An additional degree of freedom exists in the three-detector problem: relative adjoint source strengths between the three adjoint sources (the three forward detectors). Optimization of relative source strengths requires user intervention; further work is required to automate the process. Next, we present results for single detector calculations. That is, only one adjoint source (detector in the forward problem) is used in the adjoint calculation, and, hence, only one detector is optimized in the forward problem. With AVATAR's decreased running times and reduced manual labor, the user has the option of running three problems, each one optimizing a different detector, and superimposing the results.

Table I shows the full three-detector FOM's for MCNP's Weight Window Generator, and three AVATAR options. "AVATAR diffusion" refers to a multigroup adjoint diffusion calculation by THREEDANT producing weight windows dependent only upon space and energy and not capturing any transport effects. "AVATAR transport" refers to a multigroup S_N calculation by THREEDANT using only the scalar adjoint flux, so the weight windows are still only dependent upon space and energy, but transport effects are captured in an integral sense. "AVATAR angle transport" includes the average adjoint currents, so the weight windows are dependent upon angle as well.

Table I: Figure of Merits for the three-detector problem.

	Detector FOM			Minimum
Method	Near	Middle	Far	FOM
Weight Window Generator	0.2	5	3	0.2
AVATAR diffusion	0.5	1	0.5	0.5
AVATAR transport	0.5	5	0.5	0.5
AVATAR angle transport	1	9	2	1

For the Weight Window Generator (WWG) cases, the MCNP geometry needed to be refined by hand in order to have an adequate importance map. Then a crude estimate of the importance function was made to put on the refined geometry. After running for an hour (on a Sun SPARC2), the weight windows from the WWG were inspected and hand-smoothed to get rid of suspected noise. Because the WWG only allows optimization of one detector, here, the far detector, the weight windows were also hand-modified to coerce particles into the near and middle detectors.

The calculation must proceed until all three statistical errors are acceptable; the calculation is only as good as the lowest FOM, which is indicated in Table I. We see that AVATAR without angular dependence in the weight window had a minimum FOM more than two times the minimum WWG FOM. With angular-dependent weight windows, the AVATAR FOM was five times the WWG FOM. Without angular-dependent weight windows, the adjoint method, whether diffusion or transport, appeared to have little impact on the FOM's. Apparently, the S_N scalar adjoint fluxes were similar enough to the diffusion that, without including angle dependence, the calculational behavior did not improve much.

Table II shows FOM's for optimization of a single detector. For the far detector, AVATAR was about 2.5 times more efficient than MCNP's WWG. For the near detector, AVATAR was about 3.5 times more efficient than MCNP's WWG. Again, the single-detector results are more straightforward than the three-detector results since they do not require selection of relative adjoint source strengths.

Table II: Figure of Merits with only one forward detector optimized.

Method	Detector	FOM
Weight Window Generator	Far	9.0
AVATAR angle transport	Far	22
Weight Window Generator	Near	0.5
AVATAR angle transport	Near	1.8

Other variance reduction techniques may be used with AVATAR or with MCNP's WWG, such as DXTRAN spheres and DXTRAN's auxiliary techniques. These techniques are not automatic, but they may increase the FOM if properly used. However, the relative gain of AVATAR over WWG remains about the same. Even though additional variance reduction techniques may not have a place in the comparison of AVATAR and WWG, the user should not rule them out.

VIII. SUMMARY

AVATAR is a superset of the Monte Carlo code MCNP that, once certain parameters are set, automatically accelerates deep penetration Monte Carlo problems, such as oil well logging tools and radiation shielding. AVATAR is unique because it automatically solves a full three-dimensional deterministic adjoint calculation with THREEDANT and uses the scalar adjoint fluxes and average adjoint currents to produce weight windows dependent upon space, energy, and angle.

Multiple forward detectors do the most to diminish the automation in AVATAR, since the user must determine relative adjoint source strengths. However the user may choose to run each detector separately and superimpose the results. The user must also select from various AVATAR options and determine computational parameters, such as the adjoint quadrature order and the adjoint Legendre scattering moments. Once these parameters are appropriately selected, they should be valid for other similar problems.

AVATAR, on a model of a three-detector oil well logging tool, ran about 2 to 5 times faster than a run with MCNP's Weight Window Generator. More importantly, user efficiency increased dramatically. Apart from selecting adjoint source spectrums and relative strengths, AVATAR required no extra work. MCNP's Weight Window Generator required many additional hours and expertise to set up the refined geometry and hand manipulate the weight windows.

ACKNOWLEDGEMENTS

The oil well logging tool model and a great deal of assistance in testing and evaluation of AVATAR were provided by Art Becker, Charlie Case, and Joe Chiarmonte of Schlumberger-Doll Research. Other members of the Radiation Transport Group at Los Alamos provided assistance, advice, and support. We thank Forrest Brown, Jabo Tang, and Peggy Emmett for sharing their insights on the MORSE code, and Roger Blomquist for information on VIM.

REFERENCES

[1] G. Goertzel and M. H. Kalos, "Monte Carlo Methods in Transport Problems," Progr. Nucl. Energy Ser. I, Vol. 11, Physics and Mathematics, Pergamon Press, New York (1958).

[2] R. A. Forster, R. C. Little, J. F. Briesmeister, and J. S. Hendricks, "MCNP Capabilities for Nuclear Well Logging Calculations," *IEEE Transactions on Nuc. Sci.*, **37** (3) 1378 (1990).

[3] J.F. Briesmeister, Editor; "MCNP^{TM}-A General Monte Carlo N-Particle Transport Code, Version 4B," LA-12625-M, Los Alamos National Laboratory (March 1997).

[4] John S. Hendricks, "New Methods for Neutron Response Calculations with MCNP," *Joint International Conference on Mathematical Methods and Supercomputing for Nuclear Applications*, American Nuclear Society, Saratoga Springs, NY, October 5-10 (1997).

[5] E. E Lewis and W. F. Miller, *Computational Methods of Neutron Transport*, John Wiley & Sons, Inc., New York (1984).

[6] T. E. Booth and J. S. Hendricks, "Importance Estimation in Forward Monte Carlo Calculations", Nuclear Technology/Fusion, **5**, 90 (1984).

[7] J. P. Both, H. Derriennic, B. Morillon, and J. C. Nimal, "A survey of TRIPOLI–4," *Proceedings of the 8th International Conference on Radiation Shielding*, Arlington, TX USA, April 24-28 (1994).

[8] J. P. Both, J. C. Nimal, and T. Vergnaud, "Automated Importance Generation and Biasing Techniques in TRIPOLI–3," Prog. in Nucl. Energy, **24** (1–3) (1990).

[9] S. Chucas, I. Curl, T. Shuttleworth, and G. Morrell, "Preparing the Monte Carlo Code MCBEND for the 21st Century," *Proceedings of the 8th International Conference on Radiation Shielding*, Arlington, TX USA, April 24-28 (1994).

[10] P. C. Miller, et al, "The Use of an Inbuilt Importance Generator for Acceleration of the Monte Carlo Code MCBEND," *Proc. of the Int. Conf. on the Physics of Fast Reactors: Operation, Design and Computation*, Marseille, France (1990).

[11] E. Shuttleworth and S. J. Chucas, "Linked Monte Carlo and Finite Element Diffusion Methods for Reactor Shield Design," *Proceedings of the Sixth Int. Conf. on Radiation Shielding*, Tokyo, Japan (1983).

[12] J. S. Tang and T. J. Hoffman, "Monte Carlo Shielding Analyses Using an Automated Biasing Procedure," Nuc. Sci. and Eng., **99**, 329 (1988).

[13] M. W. Mickael, "A Fast, Automated, Semideterministic Weight Windows Generator for MCNP," Nuc. Sci. and Eng., **119**, 34 (1995).

[14] J. C. Wagner and A. Haghighat, "Application of the Discrete Ordinates Adjoint Function to Accelerate Monte Carlo Reactor Cavity Dosimetry Calculation," *Proceedings of the 1996 Topical Meeting on Radiation Protection and Shielding*, Vol. 1, 345-352, American Nuclear Society, Inc., No. Falmouth, Massachusetts (1996).

[15] E. T. Jaynes, "Information Theory and Statistical Mechanics," Phys. Rev. **106**, No. 4 (1957).

[16] W. F. Walters and T. A. Wareing, "A Nonlinear Positive Method for Solving the Transport Equation on Coarse Meshes," *Proceedings of the 8th International Conference on Radiation Shielding*, Arlington, TX USA, April 24-28 (1994).

[17] Lianyan Liu and Robin P. Gardner, "A Geometry-Independent Fine-Mesh-Based Monte Carlo Importance Generator," Nuc. Sci. and Eng., **125**, 188-195 (1997).

[18] K. W. Burn, "A New Weight-Dependent Direct Statistical Approach Model," Nuc. Sci. and Eng., **125**, 128-170 (1997).