A New Monte Carlo Alpha-Eigenvalue Estimator with Delayed Neutrons

Colin Josey, Forrest B. Brown

Los Alamos National Laboratory, P.O. Box 1663 MS K784, Los Alamos, NM 87545, cjosey@lanl.gov

INTRODUCTION

The α-eigenvalue is one of the possible eigenvalue solutions to the neutron transport equation. The value α corresponds to the asymptotic logarithmic time derivative of the neutron flux, assuming the materials and geometry are constant, and is computed by a separation of variables. The α-eigenvalue has extensive utility, from determining if a pulsed reactor experiment will be safe to analyzing the decay of flux during a reactor shutdown.

One issue with the α-eigenvalue is that it is most often computed by performing what is known as a $k-\alpha$ iteration [1]. A k eigenvalue solve is run in which α is adjusted until $k \to 1$. This stochastic root finding approach does not incorporate a complete picture of the physics. For example, with delayed neutrons in which the minimum decay group constant is λ_m, the fundamental α-eigenvalue must be greater than $-\lambda_m$. This work attempts to alleviate this by integrating the transport equation over all phase space and directly solving for α, eliminating the $k-\alpha$ search. This algorithm is then implemented and compared against a deterministic reference solution.

THEORY

Converting the neutron transport equation into an α-eigenvalue equation is relatively simple. First, a separation of variables is performed on the flux (ψ) and the delayed neutron precursor distribution (C) as shown in Eq. (1).

$$\begin{aligned}\psi(\boldsymbol{r}, E, \hat{\boldsymbol{\Omega}}, t) &= \psi(\boldsymbol{r}, E, \hat{\boldsymbol{\Omega}})e^{\alpha t} \\ C(\boldsymbol{r}, t) &= C(\boldsymbol{r})e^{\alpha t}\end{aligned} \tag{1}$$

Upon substitution into the transport equation, $e^{\alpha t}$, which is nonzero everywhere, can be divided out. The resulting transport equation is given by Eq. (2).

$$\begin{aligned}&\left(\frac{\alpha}{v(E)} + \hat{\boldsymbol{\Omega}}\cdot\nabla + \Sigma_t(\mathbf{r}, E)\right)\psi(\mathbf{r}, E, \hat{\boldsymbol{\Omega}}) \\ &= \frac{\chi_p(E)}{4\pi}\int_0^\infty dE'\nu_p(E')\Sigma_f(\mathbf{r}, E')\phi(\mathbf{r}, E') + \sum_{i=1}^N \frac{\chi_{di}(E)}{4\pi}\lambda_i C_i(\mathbf{r}) \\ &+ \int_{4\pi} d\Omega' \int_0^\infty dE'\nu_s(E')\Sigma_s(\mathbf{r}, E'\to E, \hat{\boldsymbol{\Omega}}'\to\hat{\boldsymbol{\Omega}})\psi(\mathbf{r}, E', \hat{\boldsymbol{\Omega}}') \\ \alpha C_i(\mathbf{r}) &= \int_0^\infty dE'\nu_{di}(E')\Sigma_f(\mathbf{r}, E')\phi(\mathbf{r}, E') - \lambda_i C_i(\mathbf{r})\end{aligned} \tag{2}$$

One can then solve the second equation for $\lambda_i C_i(\boldsymbol{r})$ for substitution into the first equation. This new equation requires two modifications from the classic k eigenvalue. The first is that ν is now a function of α, as given by Eq. (3), and sampling of χ is performed using these scaled ν values.

$$\nu(E', \alpha) = \nu_p(E') + \sum_{i=1}^N \frac{\lambda_i}{\alpha+\lambda_i}\nu_{di}(E') \tag{3}$$

The second is that α adds either a fictitious absorber or a fictitious source depending on if α is positive or negative, respectively. The source can easily be implemented via an (n, 2n) reaction with a macroscopic cross-section of α/v.

With the basics introduced, the goal now is to find a way to update α as the Monte Carlo simulation iterates. The easiest way is to simply integrate Eq. (2) (with $\lambda_i C_i(\mathbf{r})$ already solved out) over all phase space. This results in Eq. (4).

$$\begin{aligned}&\alpha\int_0^\infty dE\int_V dV\frac{1}{v(E)}\phi(\mathbf{r}, E) + \int_0^\infty dE\int_V dV\Sigma_t(\mathbf{r}, E)\phi(\mathbf{r}, E) \\ &+ \int_0^\infty dE\int_{4\pi} d\Omega\oint_S dS\,\psi(\mathbf{r}, E, \hat{\boldsymbol{\Omega}})\hat{\boldsymbol{\Omega}}\cdot\mathbf{n} \\ &= \int_V dV\int_0^\infty dE'\nu_p(E')\Sigma_f(\mathbf{r}, E')\phi(\mathbf{r}, E') \\ &+ \sum_{i=1}^N \frac{\lambda_i}{\alpha+\lambda_i}\int_V dV\int_0^\infty dE'\nu_{di}(E')\Sigma_f(\mathbf{r}, E')\phi(\mathbf{r}, E') \\ &+ \int_V dV\int_0^\infty dE'\nu_s(E')\Sigma_s(\mathbf{r}, E')\phi(\mathbf{r}, E')\end{aligned} \tag{4}$$

All of the integrals in Eq. (4) can be tallied in a Monte Carlo simulation. The resulting $N+1$ unknown equation can then be solved using simple means. One possible way is to perform a bisection search over the span $[-(1-\epsilon)\min\lambda_i, \alpha_{\max}]$, where ϵ is a small value (10^{-14}) and $\alpha_{\max}$ is the largest anticipated α. In addition, a general improvement can be made by performing the tallies both with a collision-based and a tracklength-based estimator (with the exception of leakage, which will still be treated analog). The resulting αs for each can be combined in much the same way as the k-eigenvalue. The process of combining covarying estimators is described in [2].

REFERENCE SOLUTION

In order to verify an implementation, a simple benchmark was created. An infinite region is filled with a material with cross-sections as listed in Table I and Table II. These cross-sections, as well as $\chi(E)$, are constant from 10^{-5} eV to 20 MeV and zero otherwise. All scattering is isotropic elastic scattering with the target at rest. The target mass is given by A in units of neutron mass. f is a scaling parameter for ν to allow for testing in subcritical and supercritical regimes. When $f = 1.0$, the geometry is slightly subcritical due to neutron leakage below 10^{-5} eV. Constants are provided in order to compute relativistic velocity. These constants match those used in MCNP6 [3].

In order to compute the reference α, a deterministic solver was written in which energy was discretized into N equal

TABLE I. Neutronics Parameters for the α-Eigenvalue Benchmark

Parameter	Value
Σ_t	1.0 cm^{-1}
Σ_s	0.3 cm^{-1}
Σ_c	0.45 cm^{-1}
Σ_f	0.25 cm^{-1}
ν_p	$2.6f$ cm^{-1}
ν_d	$0.2f$ cm^{-1}
A	10
Speed of Light	2.997925×10^8 m/s
Mass of Neutron	939.58×10^6 MeV/c^2

TABLE II. Delayed Precursor Parameters for the α-Eigenvalue Benchmark

Group	λ, s^{-1}	p, Probability
1	0.0133	0.0350
2	0.0327	0.1807
3	0.1208	0.1725
4	0.3028	0.3868
5	0.8495	0.1586
6	2.8530	0.0664

logarithmically-spaced groups. Within each group, the flux is treated as constant. The resulting multigroup equations take the form of Eq. (5).

$$\begin{aligned}\frac{1}{\delta_j}\int_{E_j}^{E_{j+1}} dE\frac{1}{v(E)}\alpha\phi_j + \Sigma_t\phi_j &= \sum_i \chi_j\nu_p\Sigma_f\phi_i + \sum_i \chi_j\lambda_i C_i \\ &+ \sum_i \int_{E_j}^{E_{j+1}} dE' \int_{E_i}^{E_{i+1}} dE\frac{\Sigma_s f(\alpha_s E, E; E')}{\delta_i}\phi_i \quad (5)\\ \alpha C_j &= \sum_i p_j\nu_d\Sigma_f\phi_i - \lambda_j C_j\end{aligned}$$

Where:

$$\begin{aligned}\alpha_s &= \frac{(A-1)^2}{(A+1)^2}\\ \delta_i &= E_{i+1} - E_i\\ \chi_i &= \frac{\delta_i}{E_{\max} - E_{\min}}\end{aligned}$$

and $f(\alpha_s E, E; E')$ is the uniform distribution spanning $\alpha_s E$ to E. This equation can be transformed into a matrix form:

$$\begin{aligned}\alpha\begin{pmatrix}\boldsymbol{\phi}\\ \boldsymbol{C}\end{pmatrix} &= \begin{pmatrix}V_I^{-1}(F+S-T) & D_p\\ D_f & -\lambda\end{pmatrix}\begin{pmatrix}\boldsymbol{\phi}\\ \boldsymbol{C}\end{pmatrix}\\ \alpha\boldsymbol{\Psi} &= L\boldsymbol{\Psi}\end{aligned}$$

Where:

$$\begin{aligned}V_I[j,j] &= \frac{1}{\delta_j}\int_{E_j}^{E_{j+1}} dE\frac{1}{v(E)}\\ F[j,i] &= \chi_j\nu_p\Sigma_f\\ S[j,i] &= \int_{E_j}^{E_{j+1}} dE' \int_{E_i}^{E_{i+1}} dE\frac{\Sigma_s f(\alpha_s E, E; E')}{\delta_i}\\ D_p[j,i] &= \chi_j\lambda_i\\ D_f[j,i] &= p_j\nu_d\Sigma_f\\ \lambda[j,j] &= \lambda_j\end{aligned}$$

The eigenvalues of the resulting matrix L correspond to the α eigenvalues. The eigenvalue with the largest real component is then the primary α-eigenvalue.

Computing the eigenvalues of L requires a numerical solution. A solver was written in Julia [4]. The inner integral of the scatter matrix can be performed analytically, but the outer integral and the integral of inverse relativistic velocity must be performed numerically. The function `quadgk` was used to do these integrals. Discontinuities are explicitly avoided and integrals were converged to a relative tolerance of 10^3 `eps`. The primary eigenvalue was computed via a shift-invert scheme [5], with the shifted eigenvalue corresponding to the maximum real eigenvalue from running `eig` on a double precision $N = 64$ matrix.

Of course, this is still an inexact process. First, numeric error can appear in the solution. To test for this, a $N = 512$ solution was run in both double precision and using 256-bit `BigFloats`. Results correspond to 11 digits for all but the $f = 1.0$ case. Reported values for $f \leq 1.05$ are all from this 256-bit simulation. The second issue is that this is only exact at the limit of $N \to \infty$. The number of groups was repeatedly doubled until the relative difference between pairs of solutions was below 10^{-10}. The number of groups is reported in Table IV.

For $f = 1.1$, however, the error was sufficiently large that repeatedly doubling the number of groups quickly exceeded memory limitations. To improve these results, a series of double precision simulations were run with $N = 2048, 4096, 8192, 16384, 32768$. The resulting α_N form a sequence. This sequence can be accelerated using a technique such as Wynn's epsilon method [6]. The resulting sequence accelerated value had a relative difference of 7.1×10^{-9} as compared to α_{32768}. The sequence accelerated value is the one reported.

RESULTS

A local version of MCNP6 [3] which already included α support without delayed neutrons was modified to add delayed neutrons. This required four modifications. The first was that ν was computed using the scaled value in Eq. (3). The second was that the χ spectrum was sampled using the α-scaled ν values. Third, all of the integrals in Eq. (4) (without the leading α or $\lambda_i/(\lambda_i + \alpha)$ terms) were individually tallied using both a tracklength and a collision estimator. Finally, α was computed using these integrals with bisection. The resulting $\alpha_{\text{tracklength}}$ and $\alpha_{\text{collision}}$ are combined statistically. The original

TABLE III. MCNP Simulation Parameters

Parameter	Value
Neutrons / Batch	100000
Batches	200
Batches Before α Updated	50
Batches Before Active	100
Initial α	0.0
Minimum α	$-1.33 \times 10^{-2} + 10^{-13}$ s^{-1}

code already implemented the α source (for negative α using a fictitious (n, 2n) reaction) and the α absorber (for positive α) from Eq. (2).

Then, the cross-section data from Table I and Table II were converted into ACE format using a modified version of the `simple_ace.pl` script described in [7]. The temperature of both the material and the library is set to 10^{-30} MeV/k to ensure that MCNP both does not modify the scattering cross-section and that the target velocity is always effectively zero. The result is a continuous energy problem in which only elastic scattering with a static target, absorption, and fission occur. This model is directly comparable to the reference problem described in the previous section.

The MCNP simulations were run using the parameters listed in Table III. Each simulation was repeated 50 times with different random sequences to compute true uncertainty. α is then computed for several f values. The results are then compared against the deterministic values in Table IV. All deterministic values are either rounded at 16 digits or at the last digit that corresponds between the N and $N/2$ group results. For $f \geq 1.05$, the values match within statistics. Below that, they do not. However, the maximum absolute error is below 10^{-12} for all remaining values.

In addition, the flux for $f = 0.9$ and $f = 1.1$ for a deterministic and an MCNP run are plotted in Fig. 1. Here, a single MCNP run was performed with 100 times as many neutrons for each f value. The flux was tallied in 512 equal logarithmically-spaced groups. Some groups are missing as no tally of any flux occurred. Error bars are omitted for visual clarity.

There are two possible explanations for the minute discrepancies between solutions. The first is undersampling. A good example is the comparison for $f = 1.0$. This configuration is only subcritical due to loss of neutrons below 10^{-5} eV. However, Fig. 1 shows that effectively no particles would be sampled below 10^{-3} eV. As such, to MCNP, the reactor is critical (with α much closer to 0). Second, there is also the possibility of numeric error, particularly when merging tallies.

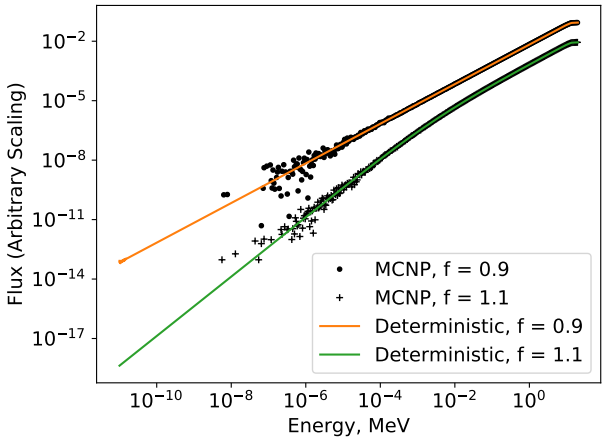

Fig. 1. Flux comparison between deterministic and MCNP for both $f = 0.9$ and $f = 1.1$.

CONCLUSIONS

Overall, for this simple benchmark problem, the new α-eigenvalue estimator works extremely well. Errors are either within statistical certainty or are extremely small. The likely source of these errors is either numerical error or undersampling. This indicates that the correct equations are being solved and that the implementation is correct. Based on these results, further investigation is warranted. Further testing will quanitfy the relative peformance of the estimator as compared to current $k-\alpha$ iteration techniques as well as test more complicated problems.

TABLE IV. Results Comparing the Deterministic Solution to the Modified MCNP Solution

f	Deterministic α	Mean α 50 MCNP Runs	α Std. Dev of Mean 50 MCNP Runs	Det. Groups
0.9	$-1.297\,467\,136\,352\,483 \times 10^{-2}$	$-1.297\,467\,136\,352\,466 \times 10^{-2}$	2.2×10^{-18}	512
0.925	$-1.284\,043\,823\,409\,392 \times 10^{-2}$	$-1.284\,043\,823\,409\,358 \times 10^{-2}$	5.0×10^{-18}	512
0.95	$-1.254\,909\,434\,037\,18 \times 10^{-2}$	$-1.254\,909\,434\,037\,101 \times 10^{-2}$	1.2×10^{-17}	512
0.975	$-1.150\,917\,974\,504\,57 \times 10^{-2}$	$-1.150\,917\,974\,504\,120 \times 10^{-2}$	5.3×10^{-17}	512
1.0	$-6.385\,261\,566\,394 \times 10^{-14}$	-4.066×10^{-16}	2.9×10^{-18}	512
1.025	$8.941\,945\,357\,03 \times 10^{-2}$	$8.941\,945\,357\,076\,1 \times 10^{-2}$	3.2×10^{-14}	512
1.05	$5.435\,470\,141\,3 \times 10^{-1}$	$5.435\,470\,141\,350 \times 10^{-1}$	1.2×10^{-12}	512
1.075	$1.873\,125\,125 \times 10^{1}$	$1.873\,125\,124\,4 \times 10^{1}$	1.6×10^{-8}	2048
1.1	$4.560\,412\,921 \times 10^{7}$	$4.560\,69 \times 10^{7}$	2.9×10^{3}	seq. accel

ACKNOWLEDGMENTS

This work was supported by the US-DOE-NNSA programs for Nuclear Criticality Safety and Advanced Simulation & Computing.

REFERENCES

1. A. ZOIA, E. BRUN, and F. MALVAGI, "Alpha eigenvalue calculations with TRIPOLI-4®," *Annals of Nuclear Energy*, **63**, 276–284 (2014).
2. T. J. URBATSCH, R. A. FORSTER, R. E. PRAEL, and R. J. BECKMAN, "Estimation and Interpretation of k_{eff} Confidence Intervals in MCNP," *Nuclear technology*, **111**, *2*, 169–182 (1995).
3. C. J. WERNER (EDITOR), "MCNP User's Manual – Code Version 6.2," Tech. Rep. LA-UR-17-29981, Los Alamos National Laboratory (2017).
4. J. BEZANSON, A. EDELMAN, S. KARPINSKI, and V. B. SHAH, "Julia: A fresh approach to numerical computing," *SIAM Review*, **59**, *1*, 65–98 (2017).
5. Y. SAAD, *Numerical Methods for Large Eigenvalue Problems: Revised Edition*, SIAM (2011).
6. P. WYNN, "On a device for computing the $e_m(S_n)$ transformation," *Mathematical Tables and Other Aids to Computation*, pp. 91–96 (1956).
7. F. B. BROWN, "New Tools to Prepare ACE Cross-section Files for MCNP Analytic Test Problems," Tech. Rep. LA-UR-16-24290, Los Alamos National Laboratory (2016).