Report: LLNL TR-847814

Monte Carlo
Statistical Convergence

by
Dermott E. Cullen
1466 Hudson Way
Livermore, CA 94550
Tele: 925-321-4177
E.Mail: RedCullen1@comcast.net
Website: http://redcullen1.net/homepage.new

April 2023

U.S. Department of Energy
Lawrence
Livermore
National
Laboratory

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Report: LLNL-847814

Monte Carlo
Statistical Convergence

**by
Dermott E. Cullen
1466 Hudson Way
Livermore, CA 94550
Tele: 925-321-4177
E.Mail: RedCullen1@comcast.net
Website: http://redcullen1.net/homepage.new**

April 2023

Abstract

This document reviews Monte Carlo Statistical Convergence over the last 60 years, based on the real World state of computers over this time. Stress is placed on providing examples of what was and was not practically possible over this period of time. I present a few examples to illustrate convergence based not only on sample size, but also how cross sections can be modelled.

Overview

In principle Monte Carlo has always been the most accurate method to model nuclear and atomic systems; most accurate to model in the sense of geometry and physical data. In practice it has always been limited by the time required to accurately calculate results on the computer facilities that we have had at any given time. Here I present a few real World example results based on using the TART Monte Carlo code [1].

TART

The TART Monte Carlo code is designed both as a production code, and as a teaching tool. TART has many input options, and these have been used over many years to optimize results and simplify use of the code today. All input options have default values, and to use the code for production users need not change/define additional options. Here I will use TART as a teaching tool, to illustrate how sample size and nuclear data models affect running time and calculated results. I would encourage users/developers of all Monte Carlo codes to present similar results for the benefit of our entire community.

Acknowledgments

I thank Dave Heinrichs (LLNL) for supporting the publication of this report and agreeing to distribute TART 2022 within LLNL. Next I acknowledge the cooperation with other neutron transport code designer/users, in particular, Bob MacFarlane (LANL, MCNP), Rich Procassini (LLNL, MERCURY), Ed Lent (LLNL, COG) and Maurice Greene (ORNL); coordinated code comparisons have led to improvements in TART as well as ALL of these other codes. I cannot stress enough that modern transport codes are far too complex to allow true code verification without detailed comparisons to other similar codes. Only in our dreams may we be perfect.

I also thank the many users of earlier versions of TART who have supplied extremely useful feedback to me. Since the general release of TART2005, TART2012 and TART2016, the response from users in terms of feedback has been extremely useful in improving the code. These improvements have been in terms of correcting problems in the earlier releases of TART, and in terms of users proposing new or improved options to meet their needs, now incorporated in TART 2022. Today virtually ALL improvements to TART are based on feedback to me from TART users. Therefore, I highly encourage all users to supply their feedback to me, so that ALL TART users can benefit from your experience, including yourself by including your ideas in future versions if TART.

Convergence

Let me first define what I will use as my definition of convergence in this paper. Many publications claim that with today's codes and nuclear data we can calculate K-eff for critical systems to within three digits, e.g., K-eff ~ 1.000 +/- 0.001; 0.1 %. Personally, I think this is far too optimist, but in this paper I will use it as a measure of convergence. I will FLAG any differences in K-eff that we calculate using the various approximations used in this paper, as a simple means of indicating a result we should be concerned by.

In this paper I have presented a few example results to illustrate both the advantages and disadvantages of Monte Carlo. One obvious advantage is its ability to model geometry more precisely, in more detail. A second advantage is its ability to define nuclear data cross sections more precisely, on a continuous energy basis. On the other hand, an obvious disadvantage is that the estimated solutions slowly converge to accurate answers. For example, convergence can vary as 1/sqrt(samples), so that to improve accuracy by a factor of 10 requires an increase in samples, and therefore running time, by a factor of 100. Even with the computer power that we have today, it may not be practical to increase sample size, and therefore running time, by a factor of 100. More to the point, it certainly wasn't practical many years ago with the computers we then had; I will briefly address this point.

A controversial point is that convergence speed depends on the number of degrees of freedom in each problem. Using continuous energy cross sections admittedly improves the detail to which nuclear data can be represented, but it can vastly increase the number of degrees of freedom in a problem. For example, in a multi-group calculation, the number of total cross sections to sample is equal to the number of groups, whereas using continuous energy cross sections there can be many thousands of tabulated cross sections, and using even simple linearly interpolation between tabulated values introduces vastly more degrees of freedom.

Here I present example results comparing continuous energy, multi-band, and multi-group results (both multi-band and multi-group use the same TART 616 energy groups). These results illustrate that even with 616 groups and a million neutron criticality samples, multi-group results differ significantly from results based on using continuous energy cross sections. We see this in particular for metal reflected critical assemblies; presumably due to neutron leakage through minima in the continuous energy cross sections, that are not included in the multi-group cross sections. However, even simple multi-band (2-band) results significantly improve agreement. This latter point should be of interest to users/developers of multi-group transport codes, since multi-band cross sections can be used in multi-group codes, but multi-group codes cannot easily be converted to use continuous energy data; the energy-to-energy transfer matrices become singular, and do not converge, as the number of groups are increased.

History 101

I wrote my first FORTRAN (FORmula TRANslation) code in 1962. At the time FORTRAN was in its infancy, having been created only slightly earlier in 1957. There was no number after the word FORTRAN (e.g., FORTRAN-77), and there were no subroutines, and therefore no common [“no common” what?]; still, this was a great leap forward in making computers accessible to anyone.

At this time not only was the language new/primitive, so were the computers. The first computer I used was I think/remember as the largest generally available computer at the time, namely the IBM-7090/94, which had 32,000, 36 bit words of memory, and a cycle time of 2 microseconds. This computer cost a minimum of $2.9 million. Storage was also extremely limited, with a 20 megabyte disk being the size of a washing machine; our main storage device was magnetic tape.

At the time our transport codes limited were to a few group diffusion calculations, and we saw the beginnings of discrete ordinates (Sn) at major laboratories. This was a primitive beginning, but the situation changed very rapidly, both due to advances in computers and the codes available to use them and I must mention nuclear data (through the ENDF effort).

By the time I started my first post-graduate position in 1967 at Brookhaven National Laboratory (BNL) an effort was just beginning to improve our nuclear data (ENDF) (without accurate nuclear data we were in a “garbage in = garbage out” situation) , and by the time I started by second position in 1972 at Lawrence Livermore National Laboratory work had begun on Monte Carlo codes, MCNP at Los Alamos, and TART at Livermore, two codes that over the years have maintained close cooperation and development (fortunately, early on it was recognized that our particle transport codes are far too complicated to believe/rely on results without code comparisons).

In the relatively short period of time between 1962 and 1972 computers had advanced from the IBM-7090/94 to CRAY computers; improving both speed and memory/storage, allowing us to consider ever more complicated problems and more detail. Since then, we have undergone a revolution in our computer capabilities., as illustrated by the below tables.

Running Time

The below table presents results obtained using a collection of 68 TART benchmark criticality problems. All 68 problems were run on each computer using only one processor, all with exactly the same run parameters, such as number of samples, and nuclear data models. This table summarizes timing results for the older TARTND code that only runs on CRAY computers, as well as all released versions of TART, on a variety of computers. The original TARTND running times from many years ago are included to illustrate how far we have come; ratios are normalized to these original CRAY-YMP running times.

Code	Computer	Running Time (Seconds)	Ratio to TARTNP CRAY-YMP	
TARTNP	**CRAY-YMP**	**5396**	**1.0**	**1995 Beginning**
TARTNP	**CRAY-J90**	**7727**	**1.43**	
TART2022	**XPS 15 7590/64bits/Windows10**	**14**	**0.0025**	**Today**
TART2016	**XPS 9100/64bits/Windows10**	**19**	**0.0035**	**2016**
TART2016	**Inspiron 5759/64bits/Windows10**	**23**	**0.0043**	
TART2005	AMD 3500+	47	0.0087	**2005**
TART2005	AMD 3400+	48	0.0089	
TART2005	IBM-PC Pentium IV/3600	58	0.0107	
TART 2002	Athlon XP1800/1520	89	0.0165	**2002**
TART 2002	IBM-PC Pentium IV/2000	132	0.025	
TART 2002	IBM-PC Lap Top III/1200	133	0.025	
TART 2002	IBM-PC Pentium III/1000	170	0.031	
TART 2002	IBM-PC Pentium III/500	500	0.09	
TART 2002	DEC-Alpha Model 5/625	516	0.10	
TART 2002	IBM-PC Pentium II/400	579	0.11	
TART 2002	PowerMAC/LapTop/500	683	0.126	
TART 2002	IBM-PC Pentium II/333	697	0.13	
TART 2002	DEC-Alpha Model 5/300	712	0.13	
TART 2002	IBM-PC Pentium II/266	855	0.16	
TART 2002	IBM-PC Pentium Pro/200	1185	0.22	
TART 2002	IBM-PC Lap Top/233	1301	0.24	
TART 2002	Power-MAC 7500/275	1350	0.25	
TART 2002	iMAC	1664	0.31	
TART 2002	HP-735/125	1834	0.34	
TART 2002	SUN E3000/166	2107	0.39	
TART 2002	IBM-PC LapTop/133	2990	0.58	
TART 2002	CRAY-YMP	4262	0.79	
TART 2002	IBM-RISC RS-6000	5739	1.06	
TART 2002	CRAY-J90	6095	1.13	
TART 2002	Meiko CS-2/66	6225	1.15	
TART95	CRAY-YMP	4912	0.91	**1995**
TART95	HP-350	4322	0.80	
TART95	DEC-Alpha	6130	1.14	
TART95	SUN	9673	1.79	

The latest TART2022 results shown above illustrate that today my humble $2000 laptop can run the 68 criticality problems in 14 seconds; an incredible **400 times faster** than TARTNP could run these same problems on a multi-million dollar CRAY-YMP. For the most up-to-date list of running times see, http://redcullen1.net/homepage.new/speed.htm

It is worth noting that the above results are only for the last 28 years since TART's first public release outside LLNL. I cannot reliably guess how these times would compare to trying to run a code like TART 50 years ago on an IBM-7090/94 – but a wild guess would be that today we can run the same problem 10,000 times faster.

Monte Carlo's Statistical Problems

Even with this incredible increase in computer power coupled with improved Monte Carlo codes and nuclear data, Mone Carlo with all of its advantages has one obvious problem: the results depend on how many histories are run, and the fact that convergence to the ultimate answer varies very slowly with the number of histories run. Basically, convergence varies as 1/sqrt(samples), e.g., to increase the accuracy by a factor of ten requires that one hundred times as many samples be run. The below table uses the TART 68 fast critical assemblies to illustrate this point using 4 results, for 10^8, 10^7, 10^6, 10^5 neutron convergence samples.

For a criticality calculation TART uses two distinct steps. The first step is **settle cycles**, where we initially "guess" the answer by defining a starting neutron flux distribution in space, energy, time. TART starts from this initial guess and iterates neutron generation by generation assuming that highly level modes will dampen out and our "guess" will evolve into the correct final lowest mode flux distribution. The second step is the actual **criticality calculation**, where we start all over from our last generation of our settled flux and iterate generation by generation, defining the effective multiple (K-eff) as the ratio of neutrons produced in one generation compared to the number in the preceding generation, i.e., how fast the flux is multiplying per generation. With TART input the users controls,

Batch Size (sentl 3) = the number of neutron samples run in each batch (generation).
Settle Cycles (critcalc) = the number of **settle** cycles (generations) to run.
Critical Cycles (sentl 2) = the number of **criticality** cycles (generations) to run.

Settle Cycles X Batch Size = number of neutron samples in **settle** calculation.
Critical Cycles X Batch Size = number of neutron samples in **criticality** calculation.

For the TART 68 fast critical assemblies the below table defines how each TART parameter was defined. Here is a very brief explanation. Experience has shown that the settle cycles are just as important as the criticality cycles, and it is recommended that both be 100 or more. Batch size is then set to define the final total samples used in the criticality calculation. Note, that in the first case the settle cycles take 10% as much time (samples) as the criticality calculation, and in the other 3 cases settle cycles take 100% as much time (samples). This settle cycle calculation is overhead required to ensure that the flux distribution has settled sufficiently to allow averaging of the following criticality cycles to define the correct average K-eff.

CAVEAT EMPTOR: Always use Howerton's first law: "**We are in no rush for the wrong answer"** [2]. When using TART or any other Monte Carlo code do not make the mistake of trying to save computer time if it means sacrificing accuracy. If you do sacrifice accuracy you are not saving time, you are wasting it. This is particularly true of settle cycles; if you do any use enough settle generations to relax close to right answer, you will end up averaging incorrect generations to define your final estimate of the answer. The can greatly – make that GREATLY – bias the final average result.

The below table summarizes the results shown in the two tables that follow. First results are presented using **continuous energy cross section** for four criticality sample sizes varying from 100,000,000 down to 100,000, in step sizes of 10. Next results are presented **varying the cross section model**, comparing results using continuous energy cross sections, to multi-band [3, 4, 5], and multi-group cross sections.

It is worth noting how far we have come: today's time to sample (settle + criticality) 1.1 x (problems) 68 X 10^8 samples per problem is 11,153 seconds. In this case TART is calculating over **670,000 samples per second**. 11,153 seconds is 186 minutes, or about 3 hours. The above table of running times shows today's TART is above 400 times faster than 30 years ago. In other words, in 1995 it would have taken TART about 1,200 hours or 50 days to do this calculation. Or if you want to go to the extreme of using my guess of 10,000 faster than a 1962 IBM-7090/94, back then it would have taken 1,250 days. Please remember that this set of 68 fast critical assemblies is only a small part of the 1,173 critical assemblies today routinely calculated by TART [1]. In our dreams if I had started these calculations in 1962 on a IBM-7090/94 by today they might not yet be finished; we could still be running today (my sense of humor).

Continuous Energy Cross Sections

Total Samples	Settle Cycles	Criticality Cycles	Batch Size	Time (Seconds) 68 Problems
10^8	100	1,000	100,000	11153.88
10^7	100	100	100,000	2161.40
10^6	100	100	10,000	136.57
10^5	100	100	1,000	1.68
Multi-Band or Multi-Group Cross Sections				
10^6 M-Band	100	100	10,000	119.29
10^6 M-Group	100	100	10,000	63.43

Returning to how I will define **convergence** in this paper, I will assume any K-eff that differs from the 10^8 results by more than 0.1 % (3 digits accuracy) is questionable. In the below tables I have marked all such K-eff estimates in YELLOW. When we compare results to the 10^8 sample size results,

What we see from the first below table based on **sample size**,

1) 10^7: ALL 68 results agree – to 0.1% 10^7 is as good as 10^8 = can save 10 times.
2) 10^6: About 25% of the results differ = marginally o.k.
3) 10^5: Most of the results differ = TOO FEW SAMPLES = CAVEAT EMPTOR

What we see from the second table based on **nuclear data models** and 10^6 samples,

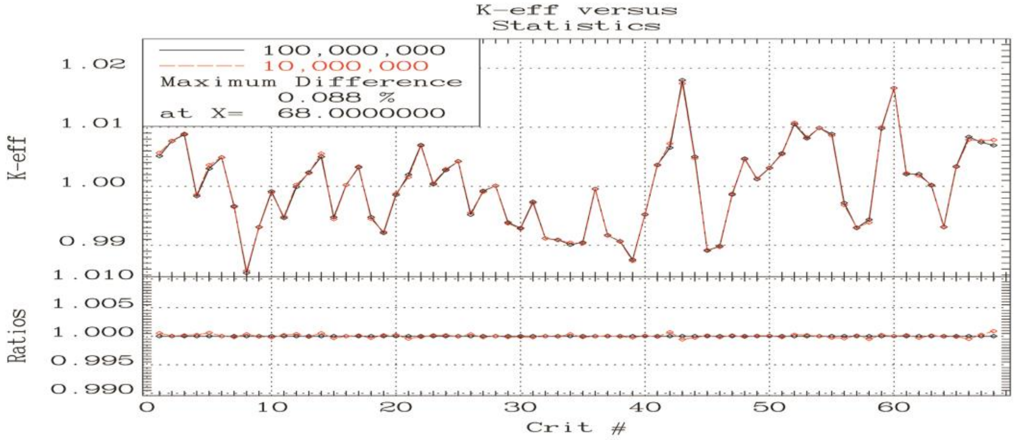
1) Continuous results are the same of the first table = marginally o.k.
2) Multi-band results are statistically better than continuous
3) Multi-Group is terrible - even the 68 case average exceeding 0.1%.

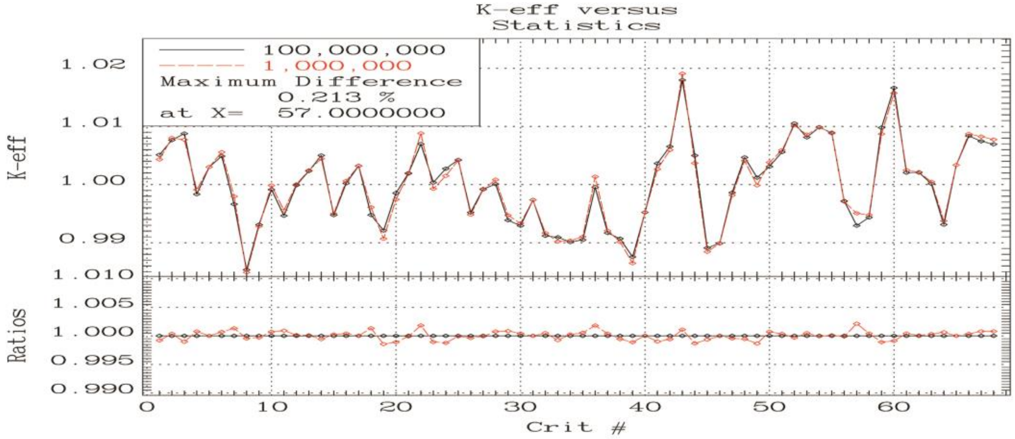
Hopefully you can take away from these results that 10^5 is far too few samples to use and expect accurate answers. At least 10^6 samples are needed to obtain even marginally acceptable answers. Note, that statistically the 10^6 multi-band results are better than the continuous (Average difference from 10^8: Multi-Band 0.002% vs. Continuous 0.005%); which is what we expect since multi-band has far fewer degrees of freedom.

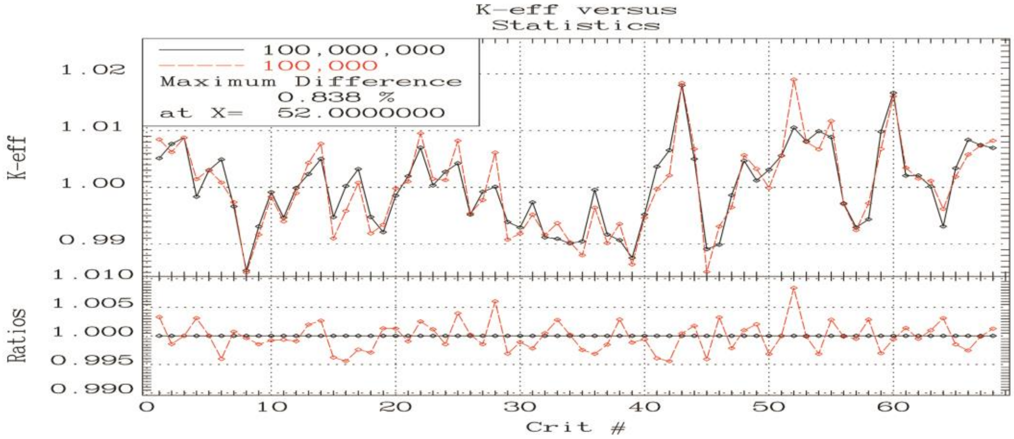
10^8, 10^7, 10^6 , 10^5 Convergence Samples Comparison

Crit.	Fuel	Reflect		K-expect	K-expect	%	K-expect	%	K-expect	%
ID.				100 Mill	10 Mill	Diff.	1 Mill	Diff.	0.1 Mill	Diff.
c10100	pu-a	be	5.222	1.005130	1.005650	-0.052	1.004330	0.080	1.008450	-0.332
c20100	pu-a	be	8.170	1.007660	1.007700	-0.004	1.008020	-0.036	1.006220	0.144
c30100	pu-a	be	13.00	1.008770	1.008920	-0.015	1.007730	0.104	1.008760	0.001
c40100	pu-d			0.998348	0.998578	-0.023	0.999115	-0.077	1.001450	-0.310
c50100	pu-d	be	3.690	1.003010	1.003600	-0.059	1.003020	-0.001	1.003060	-0.005
c60100	pu-d	be	5.250	1.004900	1.004920	-0.002	1.005590	-0.069	1.000850	0.405
c70100	pu-d	c	3.830	0.996636	0.996487	0.015	0.997957	-0.132	0.997371	-0.074
c80100	pu-d	ti	8.000	0.985341	0.985670	-0.033	0.984891	0.045	0.984978	0.036
c90100	pu-d	w	4.700	0.993105	0.993045	0.006	0.992816	0.029	0.991646	0.146
c10010	pu-d	u-235	0.660	0.999140	0.998936	0.020	0.999839	-0.070	0.998351	0.079
c11010	pu-d	u-238	1.930	0.994653	0.994847	-0.019	0.995566	-0.091	0.993983	0.067
c12010	pu-d	u-238	6.740	0.999892	1.000260	-0.037	1.000040	-0.015	0.998975	0.092
c13010	pu-d	u	4.130	1.002340	1.002230	0.011	1.002440	-0.010	1.004320	-0.198
c14010	pu-d	u	19.60	1.005010	1.005530	-0.052	1.004450	0.056	1.007680	-0.267
c10100	u-233			0.994754	0.994437	0.032	0.994997	-0.024	0.991002	0.375
c20100	u-233	be	2.050	1.000220	1.000210	0.001	1.000630	-0.041	0.995827	0.439
c30100	u-233	be	4.200	1.003220	1.003350	-0.013	1.003250	-0.003	1.000810	0.241
c40100	u-233	w	2.440	0.994778	0.994475	0.030	0.996098	-0.132	0.991873	0.291
c50100	u-233	w	5.790	0.992107	0.992280	-0.017	0.990685	0.142	0.993404	-0.130
c60100	u-233	u-235	1.210	0.998533	0.998745	-0.021	0.997441	0.109	0.999813	-0.128
c70100	u-233	u-235	1.980	1.001970	1.001540	0.043	1.001870	0.010	1.001030	0.094
c80100	u-233	u-235	4.820	1.006990	1.006860	0.013	1.008820	-0.183	1.009520	-0.253
c90100	u-233	u	2.300	1.000350	1.000510	-0.016	0.999280	0.107	1.001490	-0.114
c10010	u-233	u	5.310	1.002740	1.002910	-0.017	1.001510	0.123	1.001270	0.147
c11010	u-233	u	19.91	1.004240	1.004190	0.005	1.004160	0.008	1.008210	-0.397
c001	u-235	be	1.27	0.995201	0.995523	-0.032	0.994833	0.037	0.995395	-0.019
c002	u-235	be	2.54	0.999217	0.999044	0.017	0.999133	0.008	0.997761	0.146
c003	u-235	c	1.27	1.000080	1.000110	-0.003	1.000840	-0.076	1.006120	-0.604
c004	u-235	c	2.54	0.993876	0.993722	0.015	0.994717	-0.084	0.990783	0.309
c005	u-235	mg	1.27	0.992930	0.992806	0.012	0.993336	-0.041	0.991841	0.109
c006	u-235	mg	2.54	0.997364	0.997178	0.019	0.997353	0.001	0.995186	0.218
c007	u-235	al	1.27	0.991186	0.991167	0.002	0.991661	-0.048	0.991646	-0.046
c008	u-235	al	2.54	0.990938	0.990874	0.006	0.990207	0.073	0.993689	-0.275
c009	u-235	ti	1.27	0.990121	0.990455	-0.033	0.990370	-0.025	0.990326	-0.021
c010	u-235	ti	2.54	0.990491	0.990325	0.017	0.991024	-0.053	0.988037	0.245
c011	u-235	fe	1.27	0.999562	0.999522	0.004	1.001360	-0.180	0.996415	0.315
c012	u-235	fe	2.54	0.991684	0.991752	-0.007	0.992060	-0.038	0.990170	0.151
c013	u-235	ni	1.27	0.990690	0.990622	0.007	0.990139	0.055	0.993540	-0.285
c014	u-235	ni	2.54	0.987563	0.987346	0.022	0.986477	0.109	0.986415	0.115
c015	u-235	cu	1.27	0.995208	0.995286	-0.008	0.995245	-0.004	0.994625	0.058
c016	u-235	cu	2.54	1.003640	1.003550	0.009	1.002650	0.099	0.999706	0.393
c017	u-235	mo	1.27	1.006540	1.007240	-0.070	1.005950	0.059	1.002090	0.445
c018	u-235	mo	2.54	1.017970	1.017400	0.057	1.019080	-0.111	1.018400	-0.043
c019	u-235	mo-allo		1.005000	1.004740	0.026	1.003680	0.132	1.006760	-0.176
c020	u-235	w	1.27	0.989099	0.989213	-0.011	0.988451	0.065	0.985083	0.402
c021	u-235	w	2.54	0.989894	0.989719	0.018	0.989853	0.004	0.993111	-0.322
c10100	u-235			0.998608	0.998732	-0.012	0.998168	0.044	0.996463	0.215
c20100	u-235			1.004680	1.004570	0.011	1.004130	0.055	1.005670	-0.099
c30100	u-235			1.001210	1.001290	-0.008	0.999867	0.134	1.003250	-0.204
c40100	u-235	be	2.222	1.003100	1.003170	-0.007	1.003820	-0.072	0.999906	0.319
c50100	u-235	be	3.260	1.005590	1.005430	0.016	1.005940	-0.035	1.005540	0.005
c60100	u-235	be	4.710	1.010520	1.010780	-0.026	1.010210	0.031	1.018990	-0.847
c70100	u-235	be	5.440	1.008130	1.008310	-0.018	1.008600	-0.047	1.008000	0.013
c80100	u-235	be	9.270	1.009880	1.009850	0.003	1.009810	0.007	1.006690	0.319
c90100	u-235	be	11.79	1.008860	1.008600	0.026	1.008960	-0.010	1.011700	-0.284
c10010	u-235	c	10.16	0.997181	0.996818	0.036	0.997079	0.010	0.997041	0.014
c11010	u-235	c	15.24	0.992946	0.993044	-0.010	0.995065	-0.212	0.992447	0.050
c12010	u-235	ni	4.940	0.994357	0.993871	0.049	0.994772	-0.041	0.997189	-0.283
c13010	u-235	cu	5.030	1.009800	1.010020	-0.022	1.008710	0.109	1.006740	0.306
c14010	u-235	cu	10.56	1.016640	1.016620	0.002	1.015750	0.089	1.016030	0.061
c15010	u-235	w	5.080	1.002060	1.002240	-0.018	1.002470	-0.041	1.003440	-0.138
c16010	u-235	w	10.16	1.002080	1.001800	0.028	1.002150	-0.007	1.001570	0.051
c17010	u-235	pb	8.990	1.000150	1.000260	-0.011	1.000470	-0.032	1.001130	-0.098
c18010	u-235	pb	17.22	0.993122	0.993054	0.007	0.993752	-0.063	0.996199	-0.308
c19010	u-235	u	1.760	1.003370	1.003270	0.010	1.003370	0.000	1.001870	0.150
c20010	u-235	u	4.470	1.008370	1.007890	0.048	1.008720	-0.035	1.005790	0.258
c21010	u-235	u	9.960	1.007460	1.007710	-0.025	1.008260	-0.080	1.007330	0.013
c22010	u-235	u	18.01	1.006950	1.007840	-0.089	1.007750	-0.080	1.008220	-0.127
			Average	1.000105	1.000127	-0.002	1.000158	-0.005	0.999980	0.013
			Minimum	0.985341	0.985670		0.984891		0.984978	
			Maximum	1.017970	1.017400		1.019080		1.018990	

10^8, 10^7, 10^6 , 10^5 Convergence Samples Comparison







10^8, 10^6, 10^6 Multi-Band, 10^6 Multi-Group Convergence Samples Comparison

```
-===============================================================================
Crit.  Fuel   Reflect       K-expect K-expect  %     K-expect  %     K-expect  %
ID.                         100 Mill 1 Mill   Diff.  Bands    Diff   Groups
Groups=====================================================================================
c10100 pub-a   be      5.222 1.005130 1.004330  0.080 1.003970  0.116 1.005570 -0.044
c20100 pu-a   be      8.170 1.007660 1.008020 -0.036 1.008580 -0.092 1.005190  0.247
c30100 pu-a   be      13.00 1.008770 1.007730  0.104 1.010050 -0.128 1.006290  0.248
c40100 pu-d                 0.998348 0.999115 -0.077 0.999187 -0.084 0.997640  0.071
c50100 pu-d   be      3.690 1.003010 1.003020 -0.001 1.002620  0.039 1.003130 -0.012
c60100 pu-d   be      5.250 1.004900 1.005590 -0.069 1.003520  0.138 1.004090  0.081
c70100 pu-d   c       3.830 0.996636 0.997957 -0.132 0.996820 -0.018 0.998055 -0.142
c80100 pu-d   ti      8.000 0.985341 0.984891  0.045 0.985318  0.002 0.989815 -0.447
c90100 pu-d   w       4.700 0.993105 0.992816  0.029 0.995053 -0.195 0.994216 -0.111
c10010 pu-d   u-235   0.660 0.999140 0.999839 -0.070 0.999111  0.003 0.999096  0.004
c11010 pu-d   u-238   1.930 0.994653 0.995566 -0.091 0.994328  0.033 0.993655  0.100
c12010 pu-d   u-238   6.740 0.999892 1.000040 -0.015 1.000150 -0.026 1.000570 -0.068
c13010 pu-d   u       4.130 1.002340 1.002440 -0.010 1.002160  0.018 1.003400 -0.106
c14010 pu-d   u       19.60 1.005010 1.004450  0.056 1.004420  0.059 1.008050 -0.304
c10100 u-233                0.994754 0.994997 -0.024 0.994519  0.023 0.994887 -0.013
c20100 u-233  be      2.050 1.000220 1.000630 -0.041 0.998618  0.160 1.000270 -0.005
c30100 u-233  be      4.200 1.003220 1.003250 -0.003 1.002670  0.055 1.004020 -0.080
c40100 u-233  w       2.440 0.994778 0.996098 -0.132 0.994078  0.070 0.994385  0.039
c50100 u-233  w       5.790 0.992107 0.990685  0.142 0.992892 -0.079 0.990730  0.138
c60100 u-233  u-235   1.210 0.998533 0.997441  0.109 0.998305  0.023 0.998610 -0.008
c70100 u-233  u-235   1.980 1.001970 1.001870  0.010 1.001650  0.032 1.000780  0.119
c80100 u-233  u-235   4.820 1.006990 1.008820 -0.183 1.006000  0.099 1.006670  0.032
c90100 u-233  u       2.300 1.000350 0.999280  0.107 1.000830 -0.048 1.000170  0.018
c10010 u-233  u       5.310 1.002740 1.001510  0.123 1.003570 -0.083 1.002580  0.016
c11010 u-233  u       19.91 1.004240 1.004160  0.008 1.003330  0.091 1.004970 -0.073
c001   u-235  be      1.27  0.995201 0.994833  0.037 0.994637  0.056 0.995590 -0.039
c002   u-235  be      2.54  0.999217 0.999133  0.008 0.999669 -0.045 0.997613  0.160
c003   u-235  c       1.27  1.000080 1.000840 -0.076 0.998850  0.123 1.000430 -0.035
c004   u-235  c       2.54  0.993876 0.994717 -0.084 0.994266 -0.039 0.993075  0.080
c005   u-235  mg      1.27  0.992930 0.993336 -0.041 0.991861  0.107 0.993096 -0.017
c006   u-235  mg      2.54  0.997364 0.997353  0.001 0.998272 -0.091 0.998617 -0.125
c007   u-235  al      1.27  0.991186 0.991661 -0.048 0.990965  0.022 0.991389 -0.020
c008   u-235  al      2.54  0.990938 0.990207  0.073 0.991841 -0.090 0.990189  0.075
c009   u-235  ti      1.27  0.990121 0.990370 -0.025 0.990711 -0.059 0.990715 -0.059
c010   u-235  ti      2.54  0.990491 0.991024 -0.053 0.990928 -0.044 0.993513 -0.302
c011   u-235  fe      1.27  0.999562 1.001360 -0.180 0.998895  0.067 1.004400 -0.484
c012   u-235  fe      2.54  0.991684 0.992060 -0.038 0.991828 -0.014 1.000280 -0.860
c013   u-235  ni      1.27  0.990690 0.990139  0.055 0.990452  0.024 0.996786 -0.610
c014   u-235  ni      2.54  0.987563 0.986477  0.109 0.986948  0.061 1.003390 -1.583
c015   u-235  cu      1.27  0.995208 0.995245 -0.004 0.994339  0.087 0.996295 -0.109
c016   u-235  cu      2.54  1.003640 1.002650  0.099 1.001680  0.196 1.006860 -0.322
c017   u-235  mo      1.27  1.006540 1.005950  0.059 1.006520  0.002 1.008700 -0.216
c018   u-235  mo      2.54  1.017970 1.019080 -0.111 1.017800  0.017 1.018680 -0.071
c019   u-235  mo-allo       1.005000 1.003680  0.132 1.005010 -0.001 1.003840  0.116
c020   u-235  w       1.27  0.989099 0.988451  0.065 0.988973  0.013 0.990684 -0.159
c021   u-235  w       2.54  0.989894 0.989853  0.004 0.989500  0.039 0.990881 -0.099
c10100 u-235                0.998608 0.998168  0.044 0.999182 -0.057 0.998641 -0.003
c20100 u-235                1.004680 1.004130  0.055 1.003660  0.102 1.005610 -0.093
c30100 u-235                1.001210 0.999867  0.134 1.000430  0.078 1.001630 -0.042
c40100 u-235  be      2.222 1.003100 1.003820 -0.072 1.003510 -0.041 1.003750 -0.065
c50100 u-235  be      3.260 1.005590 1.005940 -0.035 1.005880 -0.029 1.005170  0.042
c60100 u-235  be      4.710 1.010520 1.010210  0.031 1.012290 -0.177 1.008970  0.155
c70100 u-235  be      5.440 1.008130 1.008600 -0.047 1.008470 -0.034 1.009000 -0.087
c80100 u-235  be      9.270 1.009880 1.009810  0.007 1.010810 -0.093 1.008240  0.164
c90100 u-235  be      11.79 1.008860 1.008960 -0.010 1.009800 -0.094 1.007650  0.121
c10010 u-235  c       10.16 0.997181 0.997079  0.010 0.997536 -0.035 0.996982  0.020
c11010 u-235  c       15.24 0.992946 0.995065 -0.212 0.994051 -0.111 0.991671  0.128
c12010 u-235  ni      4.940 0.994357 0.994772 -0.041 0.993762  0.060 1.010100 -1.574
c13010 u-235  cu      5.030 1.009800 1.008710  0.109 1.008550  0.125 1.014710 -0.491
c14010 u-235  cu      10.56 1.016640 1.015750  0.089 1.015500  0.114 1.027350 -1.071
c15010 u-235  w       5.080 1.002060 1.002470 -0.041 1.002530 -0.047 1.002860 -0.080
c16010 u-235  w       10.16 1.002080 1.002150 -0.007 1.002290 -0.021 1.004920 -0.284
c17010 u-235  pb      8.990 1.000150 1.000470 -0.032 1.000180 -0.003 1.002090 -0.194
c18010 u-235  pb      17.22 0.993122 0.993752 -0.063 0.993551 -0.043 0.997023 -0.390
c19010 u-235  u       1.760 1.003370 1.003370  0.000 1.004350 -0.098 1.003200  0.017
c20010 u-235  u       4.470 1.008370 1.008720 -0.035 1.008770 -0.040 1.009070 -0.070
c21010 u-235  u       9.960 1.007460 1.008260 -0.080 1.006740  0.072 1.006060  0.140
c22010 u-235  u       18.01 1.006950 1.007750 -0.080 1.008300 -0.135 1.006760  0.019
================================================================================
                     Average 1.000105 1.000158 -0.005 1.000085  0.002 1.001372 -0.127
                     Minimum 0.985341 0.984891        0.985318        0.989815
                     Maximum 1.017970 1.019080        1.017800        1.027350
================================================================================
```

Effect of Cross Sections: Measured vs. Theory

The below table illustrates the effect of two different sets of nuclear data. In this case both were run with exactly the same converged sample size, 10^8 neutron samples, and continuous cross sections; they differ only in the nuclear data used by each. Here I will not name the two sets of data, I will only mention that they differ in among other ways by the fact that one contains much more elemental data, and the other isotopic data.

The elemental data has the advantage of being better known from experiments, while the more theoretical isotopic data has the advantage of allowing each isotopic to be defined in more detail, such as different resolved and unresolved resonance regions. Which is better? Which produces more accurate results? Or does it matter?

In an attempt to answer these questions, I again return to the definition of **convergence** that I am using throughout this paper: that differences of more than 0.1% in K-eff is significant.

The below table illustrates that K-eff for all 68 cases differs by well in excess of 0.1%; indeed, even the average of all 68 cases is 0.5% different, with maximum differences approaching 1.0%. Note that the difference is not statistical, with some higher and some lower. Here in ALL 68 cases one set has a higher K-eff. Which one is better? Only time and testing will answer that question.

The point that I hope to make here is that even with the great strides we have made over the last more than 50 years to improve our nuclear data, it still remains a MAJOR factor and concern in our nuclear applications. Please always remember that even with a perfect code and methods, without good data we are in a "**Garbage In = Garbage Out**" situation.

10^8 Samples Comparison of two Different Sets of Nuclear Data

```
=====================================================================
Crit.     Fuel      Reflector              Expected K          %
ID.                                      Data A    Data B     Diff
=====================================================================
c10100    pu-a      be          5.222    1.0051300 0.9987940  0.634 %
c20100    pu-a      be          8.170    1.0076600 1.0014900  0.617 %
c30100    pu-a      be          13.000   1.0087700 1.0026600  0.611 %
c40100    pu-d                           0.9983480 0.9915910  0.676 %
c50100    pu-d      be          3.690    1.0030100 0.9964720  0.654 %
c60100    pu-d      be          5.250    1.0049000 0.9985310  0.637 %
c70100    pu-d      c           3.830    0.9966360 0.9896390  0.700 %
c80100    pu-d      ti          8.000    0.9853410 0.9801100  0.523 %
c90100    pu-d      w           4.700    0.9931050 0.9862920  0.681 %
c10010    pu-d      u-235       0.660    0.9991400 0.9925030  0.664 %
c11010    pu-d      u-238       1.930    0.9946530 0.9870730  0.758 %
c12010    pu-d      u-238       6.740    0.9998920 0.9914460  0.845 %
c13010    pu-d      u           4.130    1.0023400 0.9943420  0.800 %
c14010    pu-d      u           19.600   1.0050100 0.9957360  0.927 %
c10100    u-233                          0.9947540 0.9905680  0.419 %
c20100    u-233     be          2.050    1.0002200 0.9966000  0.362 %
c30100    u-233     be          4.200    1.0032200 1.0000200  0.320 %
c40100    u-233     w           2.440    0.9947780 0.9911950  0.358 %
c50100    u-233     w           5.790    0.9921070 0.9887550  0.335 %
c60100    u-233     u-235       1.210    0.9985330 0.9946780  0.386 %
c70100    u-233     u-235       1.980    1.0019700 0.9979380  0.403 %
c80100    u-233     u-235       4.820    1.0069900 1.0027800  0.421 %
c90100    u-233     u           2.300    1.0003500 0.9961960  0.415 %
c10010    u-233     u           5.310    1.0027400 0.9984810  0.426 %
c11010    u-233     u           19.910   1.0042400 0.9990690  0.517 %
c001      u-235     be          1.27     0.9952010 0.9906170  0.458 %
c002      u-235     be          2.54     0.9992170 0.9946110  0.461 %
c003      u-235     c           1.27     1.0000800 0.9952350  0.485 %
c004      u-235     c           2.54     0.9938760 0.9891320  0.474 %
c005      u-235     mg          1.27     0.9929300 0.9881520  0.478 %
c006      u-235     mg          2.54     0.9973640 0.9927210  0.464 %
c007      u-235     al          1.27     0.9911860 0.9860450  0.514 %
c008      u-235     al          2.54     0.9909380 0.9865740  0.436 %
c009      u-235     ti          1.27     0.9901210 0.9854960  0.462 %
c010      u-235     ti          2.54     0.9904910 0.9862920  0.420 %
c011      u-235     fe          1.27     0.9995620 0.9946920  0.487 %
c012      u-235     fe          2.54     0.9916840 0.9867870  0.490 %
c013      u-235     ni          1.27     0.9906900 0.9856940  0.500 %
c014      u-235     ni          2.54     0.9875630 0.9826570  0.491 %
c015      u-235     cu          1.27     0.9952080 0.9906600  0.455 %
c016      u-235     cu          2.54     1.0036400 0.9986830  0.496 %
c017      u-235     mo          1.27     1.0065400 1.0018700  0.467 %
c018      u-235     mo          2.54     1.0179700 1.0134100  0.456 %
c019      u-235     mo-alloy             1.0050000 1.0001800  0.482 %
c020      u-235     w           1.27     0.9890990 0.9846640  0.443 %
c021      u-235     w           2.54     0.9898940 0.9854630  0.443 %
c10100    u-235                          0.9986080 0.9939070  0.470 %
c20100    u-235                          1.0046800 1.0006900  0.399 %
c30100    u-235                          1.0012100 0.9965570  0.465 %
c40100    u-235     be          2.222    1.0031000 0.9985040  0.460 %
c50100    u-235     be          3.260    1.0055900 1.0009900  0.460 %
c60100    u-235     be          4.710    1.0105200 1.0059100  0.461 %
c70100    u-235     be          5.440    1.0081300 1.0035700  0.456 %
c80100    u-235     be          9.270    1.0098800 1.0053600  0.452 %
c90100    u-235     be          11.790   1.0088600 1.0043400  0.452 %
c10010    u-235     c           10.160   0.9971810 0.9931480  0.403 %
c11010    u-235     c           15.240   0.9929460 0.9888790  0.407 %
c12010    u-235     ni          4.940    0.9943570 0.9898860  0.447 %
c13010    u-235     cu          5.030    1.0098000 1.0048500  0.495 %
c14010    u-235     cu          10.560   1.0166400 1.0121700  0.447 %
c15010    u-235     w           5.080    1.0020600 0.9976830  0.438 %
c16010    u-235     w           10.160   1.0020800 0.9977310  0.435 %
c17010    u-235     pb          8.990    1.0001500 0.9957320  0.442 %
c18010    u-235     pb          17.220   0.9931220 0.9886030  0.452 %
c19010    u-235     u           1.760    1.0033700 0.9984420  0.493 %
c20010    u-235     u           4.470    1.0083700 1.0028700  0.550 %
c21010    u-235     u           9.960    1.0074600 1.0008100  0.665 %
c22010    u-235     u           18.010   1.0069500 0.9998690  0.708 %
=====================================================================
                                Averages 1.0001052 0.9950455  0.506 %
                                Minimum  0.9853410 0.9801100
                                Maximum  1.0179700 1.0134100
=====================================================================
```

Conclusions

Convergence

I first defined what I will use as my definition of convergence in this paper. Many publications claim that with today's codes and nuclear data we can calculate K-eff for critical systems to within three digits, e.g., K-eff ~ 1.000 +/- 0.001; 0.1 %. Personally, I think this is far too optimist, but in this paper I will use it as a measure of convergence. I will FLAG any differences in K-eff that we calculate using the various approximations used in this paper, as a simple means of indicating a result we should be concerned by.

In this paper I have presented a few example results to illustrate both the advantages and disadvantages of Monte Carlo. One obvious advantage is its ability to model geometry more precisely, in more detail. A second advantage is its ability to define nuclear data cross sections more precisely, on a continuous energy basis. On the other hand, an obvious disadvantage is that the estimated solutions slowly converge to accurate answers. For example, convergence can vary as 1/sqrt(samples), so that to improve accuracy by a factor of 10 requires an increase in samples, and therefore running time, by a factor of 100. Even with the computer power that we have today, it may not be practical to increase sample size, and therefore running time, by a factor of 100. More to the point, it certainly wasn't practical many years ago with the computers we then had; I will briefly address this point.

A controversial point is that convergence speed depends on the number of degrees of freedom in each problem. Using continuous energy cross sections admittedly improves the detail to which nuclear data can be represented, but it can vastly increase the number of degrees of freedom in a problem. For example. in a multi-group calculation, the number of total cross sections to sample is equal to the number of groups, whereas using continuous energy cross sections there can be many thousands of tabulated cross section, and using even simple linearly interpolation between tabulated values introduces vastly more degrees of freedom.

Here I present example results comparing continuous energy, multi-band, and multi-group results (both multi-band and multi-group use the same TART 616 energy groups). These results illustrate that even with 616 groups and a million neutron criticality samples, multi-group results differ significantly from results based on using continuous energy cross sections. We see this in particular for metal reflected critical assemblies; presumably due to neutron leakage through minima in the continuous energy cross sections, that are not included in the multi-group cross sections. However, even simple multi-band (2-band) results significantly improve agreement. This latter point should be of interest to users/developers of multi-group transport codes, since multi-band cross sections can be used in multi-group codes, but multi-group codes cannot easily be converted to use continuous energy data; the energy-to-energy transfer matrices become singular, and do not converge, as the number of groups are increased.

References

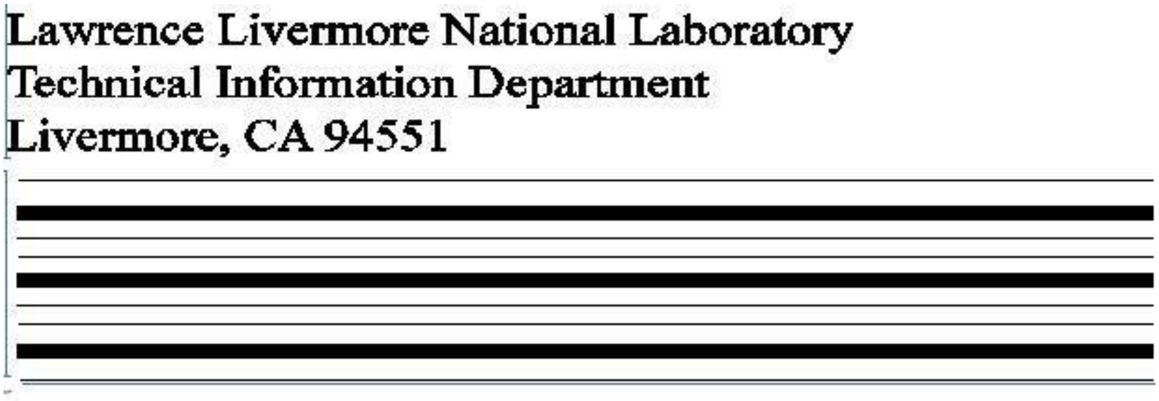
[1] "**TART 2022**: An Overview of A Coupled Neutron-Photon 3-D, Combinatorial Geometry Time Dependent Monte Carlo Transport Code", Report: LLNL-SM-835527, Code Release: LLNL-CODE-836321, May 2022.

[2] **Private Communication**, R.J.Howerton (1972)

[3] "Application of the Probability Table Method to Multi-Group Calculations," BNL-50387 (ENDF-187), Brookhaven, New York (1973).

[4] "Nuclear Cross Section Processing", Handbook of Nuclear Reactor Calculation, vol. I, Yigal Ronen, Editor, CRC Press, inc., Boca Raton, Florida (1986).

[5] "**PREPRO 2021**: 2021 ENDF/B Pre-processing Codes (ENDF/B-VIII0.0 Improved Precision)", IAEA-NDS-0238, Nuclear Data Section (NDS), IAEA, Vienna, Austria, July 14, 2021.



Lawrence Livermore National Laboratory
Technical Information Department
Livermore, CA 94551