LA-UR-19-28572
Approved for public release; distribution is unlimited.

Title: Predicting Monte Carlo Tally Variance and Calculation Time when Using Forced-flight Variance Reduction—Verification

Author(s): Kulesza, Joel A.
Solomon, Clell Jeffrey Jr.
Kiedrowski, Brian C.

Intended for: MC2019, 2019-08-25/2019-08-29 (Portland, Oregon, United States)

Issued: 2019-09-12 (rev.1)

Disclaimer:
Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by Triad National Security, LLC for the National Nuclear Security Administration of U.S. Department of Energy under contract 89233218CNA000001. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

Predicting Monte Carlo Tally Variance and Calculation Time when Using Forced-flight Variance Reduction—Verification

Joel A. Kulesza[1,2], Clell J. Solomon, Jr.[1], and Brian C. Kiedrowski[2]

[1]Los Alamos National Laboratory, Computational Physics Division
[2]University of Michigan, Dept. of Nuclear Engineering & Radiological Sciences

American Nuclear Society
International Conference on Mathematics and Computational Methods applied to Nuclear Science and Engineering (M&C 2019)
August 25–29, 2019

Acknowledgements

This work is supported by the Department of Energy National Nuclear Security Administration (NNSA) Advanced Simulation and Computing (ASC) Program. It is also supported by the NNSA under Award Number(s) DE-NA0002576 and in part by the NNSA Office of Defense Nuclear Nonproliferation R&D through the Consortium for Nonproliferation Enabling Capabilities.

Outline

Introduction

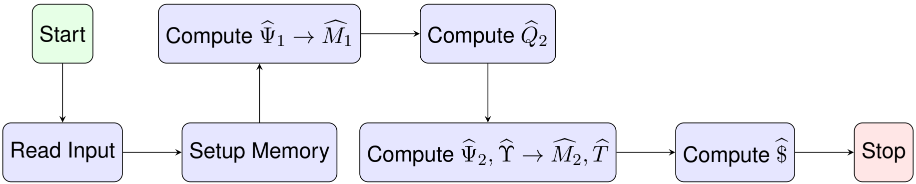
Objective: Numerically solve the forced-flight history-score moment and future-time equations

- Previously derived the forced-flight HSMEs and FTE
- This work builds upon the COVRT software (Solomon, 2010)
- COVRT originally solved:
  - Cell-based importance splitting and rouletting
  - Implicit capture & weight cutoff (rouletting)
  - Weight windows
- New work:
  - Modified transport sweep to account for forced-flight particle creation
  - Python-based interface to optimization (generally out of scope)

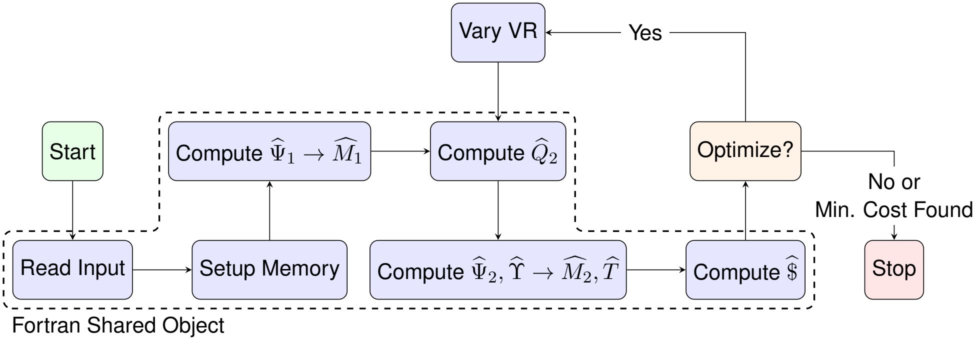
Implementation Overview

- Forced-flight HSMEs and FTE added to COVRT
  - COVRT: **C**ost-**O**ptimized **V**ariance **R**eduction **T**echnique
  - Cartesian multi-group discrete ordinates HSME & FTE solver
- Two approaches to solving the HSMEs and FTE
  - Explicitly separated and truncated moment
  - Combined moments and implicitly truncated moments
    - Eases combination with other variance reduction techniques
- COVRT is also bound to a Python driver via `ctypes`
  - Permits Python-based optimization
  - This work uses MADS-based optimization from NOMAD
    - (Audet et al., 2009; Le Digabel, 2011)
  - Tested other optimization algorithms
    - SLSQP, COBLYA, Basinhopping, Differential Evolution
    - Particle Swarm
    - Markov-chain Monte Carlo (MCMC)

Once-through COVRT Calculation Process

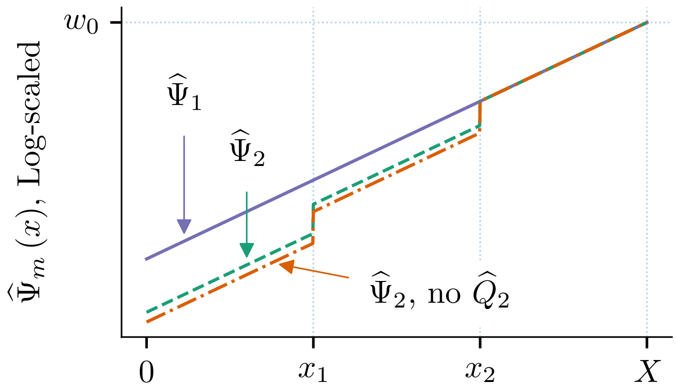


Python-driven Optimization Calculation Process



Solving for Moments

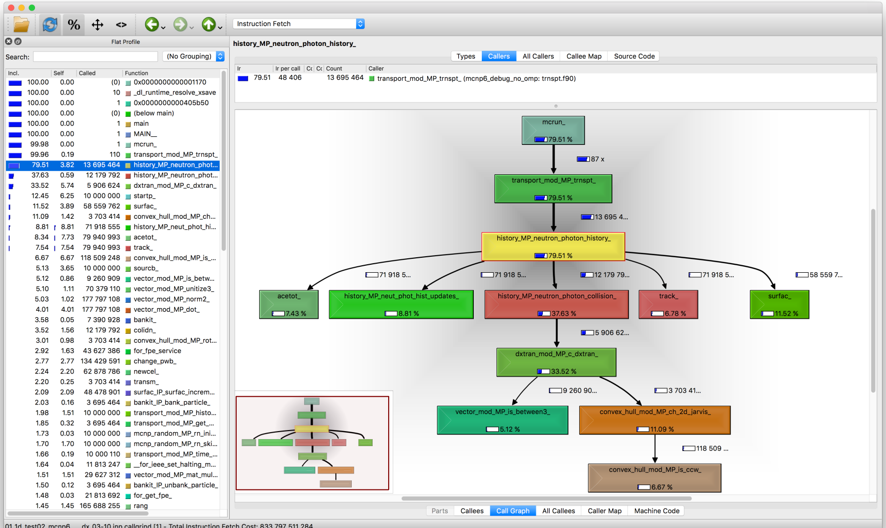
- The integral equations are recast into integro-differential form
- Solver for the HSMEs and FTE uses unaccelerated source iteration
- Complications can arise when computing $\widehat{Q}_2$: the $\widehat{\Psi}_2$ source term
  - Detecting incorrect $\widehat{Q}_2$ can be difficult; differences can be subtle



Monte Carlo Time Profiling

- Future-time Equations require computational event times τ_{xs}, τ_{ff}, etc.
  - Specific to Monte Carlo code and problem definition
- Overall timing function somewhat insensitive to times
  - Relative trends are important to capture, accurate values less so
- The MCNP code is used for this work
  - Representative problems are used to compute event times
- Times are obtained with Valgrind's Callgrind (Weidendorfer et al., 2004; Nethercote and Seward, 2007)
  - The MCNP code is compiled with debugging symbols
  - Obtain: the total time to run the calculation and the number of instructions for each computational event
  - Assume: each instruction takes the same amount of time
  - Calculate: time per instruction for each computational event
  - Validate: compare Monte Carlo and deterministic cost-surfaces

Example Profiling Graphical Output



Example Profiling Tabular Output

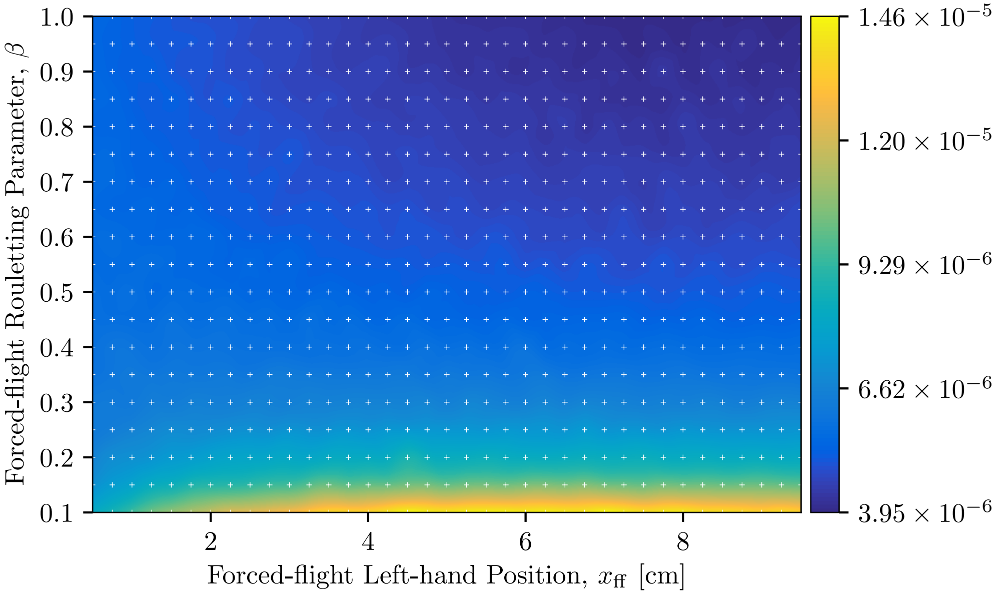
MCNP Routine	# Instructions	Time In Routine	Calls	Time Per Call
surfac	87353373653	5.59	58559762	9.54×10^{-8}
bankit	29833481748	1.91	7390928	2.58×10^{-7}
acetot	69569700257	4.45	79940993	5.57×10^{-8}
transm	18370101431	1.18	3703414	3.17×10^{-7}
track	62894992419	4.02	79940993	5.03×10^{-8}
startp	103766122730	6.64	10000000	6.64×10^{-7}
tally	8718657535	5.58×10^{-1}	2850356	1.96×10^{-7}
colidn	29370590911	1.88	12179792	1.54×10^{-7}
mcnp_random_mp_rn_init_particle	14418536758	9.23×10^{-1}	10000000	9.23×10^{-8}
dxtran_mod_MP_c_dxtran	261080123457	1.67×10^{1}	5906624	2.83×10^{-6}
cell_properties_MP_acetot_embed	8264243712	5.29×10^{-1}	8022438	6.59×10^{-8}

Obtained from a total of 833797511284 instructions in a 53.35 minute run.

Time Profiling Values Used

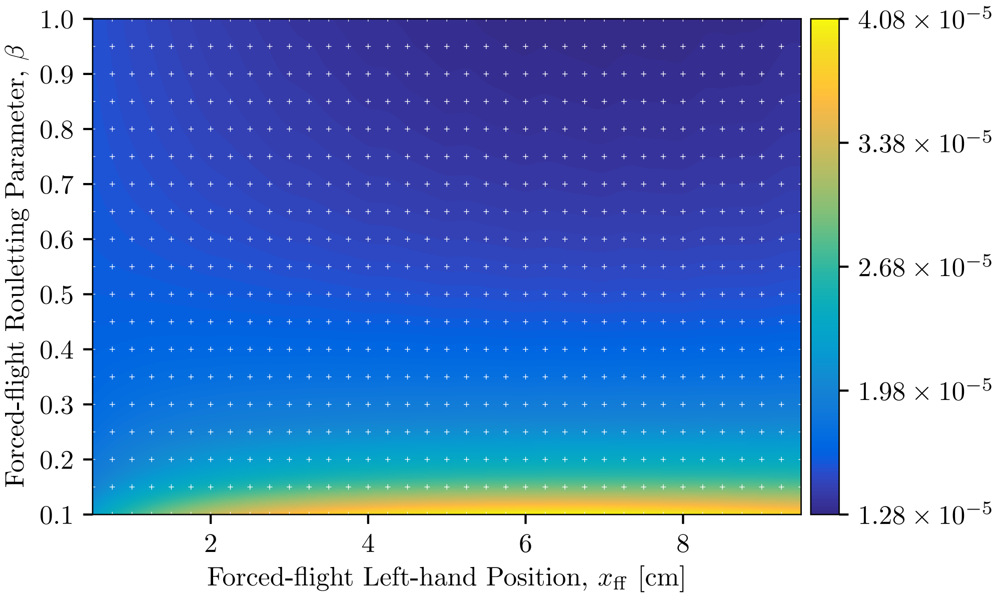
FTE Term	COVRT Input Variable	Value [min]	Note
τ_{tally}	`tally`	4.31506×10^{-7}	Reused from (Solomon, 2010).
τ_{col}	`colidn`	1.52900×10^{-7}	Reused from (Solomon, 2010).
τ_{xs}	`acetot`	8.45004×10^{-8}	—
τ_{geom}	`track`	1.05709×10^{-7}	—
τ_{rt}	`transm`	9.55000×10^{-8}	—
τ_{ff}	`dxtran`	5.88000×10^{-7}	—
τ_{surf}	`surfac`	4.39085×10^{-7}	—
τ_{bank}	`bankit`	2.24612×10^{-7}	—
τ_{src}	`startp`	1.34064×10^{-6}	—

Time Profile Validation: MCNP Cost Surface

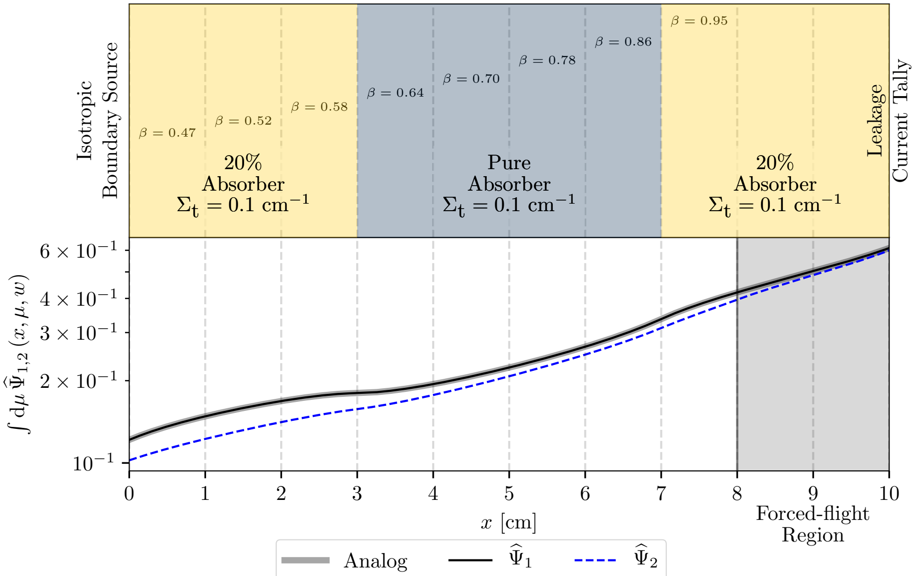


Time Profile Validation: COVRT Cost Surface

Validate



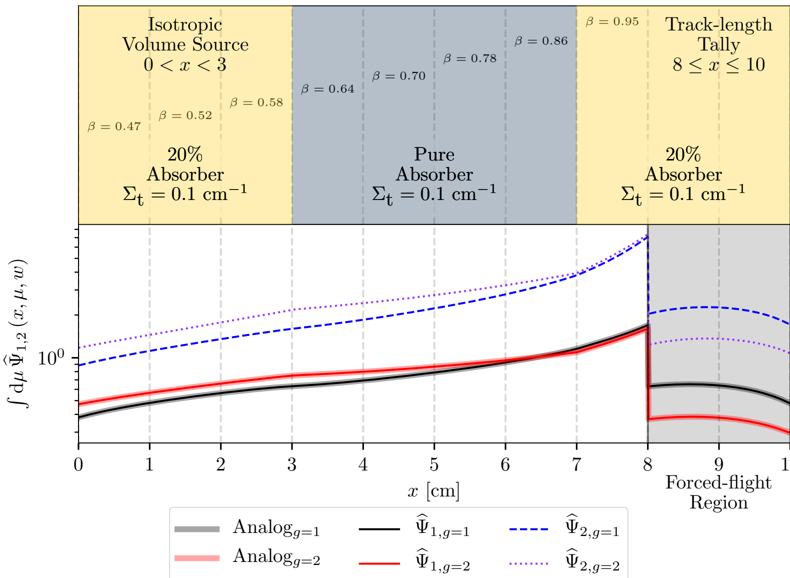
1-D, Surface Source, Leakage-current Tally



Various Forced-flight Locations

x_{ff} [cm]		MCNP6		COVRT		MCNP6/COVRT	
min.	max.	Mean	Variance	Mean	Variance	Mean	Variance
0	10	1.216×10^{-1}	1.068×10^{-1}	1.215×10^{-1}	1.067×10^{-1}	1.00	1.00
1	10	1.216×10^{-1}	9.892×10^{-2}	1.215×10^{-1}	9.885×10^{-2}	1.00	1.00
2	10	1.216×10^{-1}	9.471×10^{-2}	1.215×10^{-1}	9.463×10^{-2}	1.00	1.00
3	10	1.216×10^{-1}	9.211×10^{-2}	1.215×10^{-1}	9.205×10^{-2}	1.00	1.00
4	10	1.216×10^{-1}	9.139×10^{-2}	1.215×10^{-1}	9.135×10^{-2}	1.00	1.00
5	10	1.216×10^{-1}	9.071×10^{-2}	1.215×10^{-1}	9.067×10^{-2}	1.00	1.00
6	10	1.216×10^{-1}	9.013×10^{-2}	1.215×10^{-1}	9.007×10^{-2}	1.00	1.00
7	10	1.215×10^{-1}	8.958×10^{-2}	1.215×10^{-1}	8.954×10^{-2}	1.00	1.00
8	10	1.216×10^{-1}	8.757×10^{-2}	1.215×10^{-1}	8.750×10^{-2}	1.00	1.00
9	10	1.216×10^{-1}	8.658×10^{-2}	1.215×10^{-1}	8.653×10^{-2}	1.00	1.00
10	10	1.216×10^{-1}	8.651×10^{-2}	1.215×10^{-1}	8.648×10^{-2}	1.00	1.00

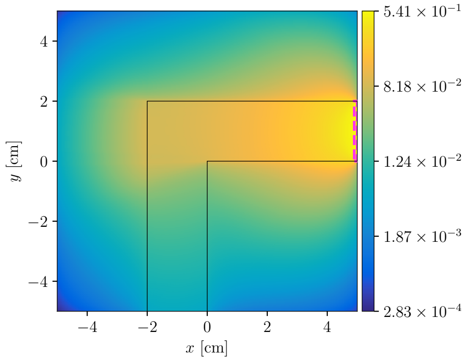
1-D, Distributed Source, Track-length Tally



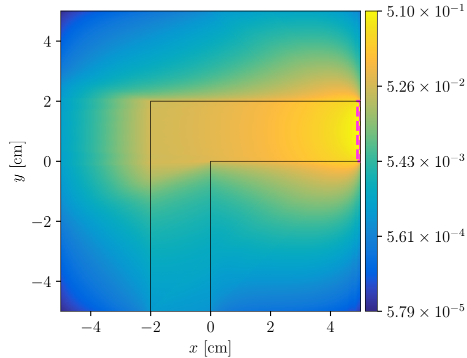
Various Forced-flight Locations

x_{ff} [cm]		MCNP6		COVRT		MCNP6/COVRT	
min.	max.	Mean	Variance	Mean	Variance	Mean	Variance
0	10	5.210×10^{-1}	1.400×10^{0}	5.198×10^{-1}	1.382×10^{0}	1.00	1.01
1	10	5.210×10^{-1}	1.316×10^{0}	5.198×10^{-1}	1.300×10^{0}	1.00	1.01
2	10	5.211×10^{-1}	1.223×10^{0}	5.198×10^{-1}	1.209×10^{0}	1.00	1.01
3	10	5.212×10^{-1}	1.138×10^{0}	5.198×10^{-1}	1.126×10^{0}	1.00	1.01
4	10	5.211×10^{-1}	1.114×10^{0}	5.198×10^{-1}	1.104×10^{0}	1.00	1.01
5	10	5.210×10^{-1}	1.092×10^{0}	5.198×10^{-1}	1.084×10^{0}	1.00	1.01
6	10	5.210×10^{-1}	1.073×10^{0}	5.198×10^{-1}	1.066×10^{0}	1.00	1.01
7	10	5.209×10^{-1}	1.052×10^{0}	5.198×10^{-1}	1.046×10^{0}	1.00	1.01
8	10	5.212×10^{-1}	9.798×10^{-1}	5.198×10^{-1}	9.688×10^{-1}	1.00	1.01

2-D, Right-angle Duct, Configuration A



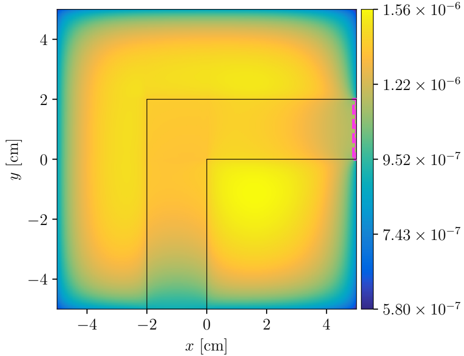
First HSME, $\widehat{\Psi}_1$



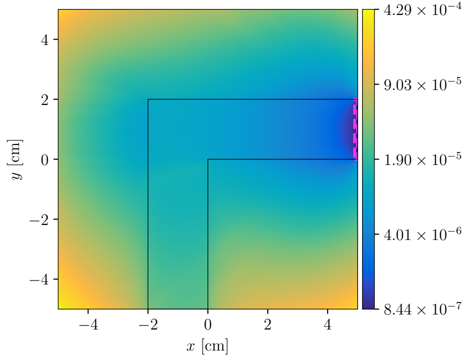
Second HSME, $\widehat{\Psi}_2$

MCNP6		COVRT		MCNP6/COVRT	
Mean	Variance	Mean	Variance	Mean	Variance
5.418×10^{-3}	5.388×10^{-3}	5.583×10^{-3}	5.552×10^{-3}	0.97	0.97
5.450×10^{-3}	1.062×10^{-3}	5.583×10^{-3}	1.052×10^{-3}	0.98	1.01

2-D, Right-angle Duct, Configuration A, cont.



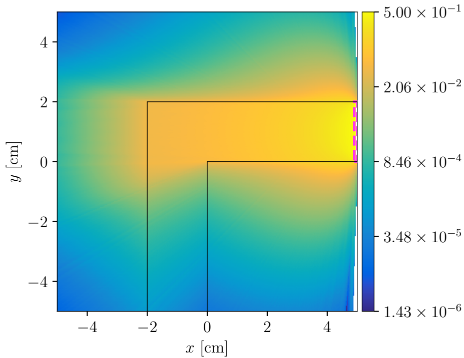
FTE, $\widehat{\Upsilon}$



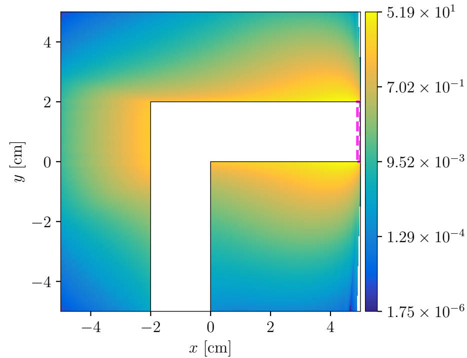
Cost, $\widehat{\$}$

- Optimization predicted a 1% improvement; 2% degradation realized

2-D, Right-angle Duct, Configuration A, cont.

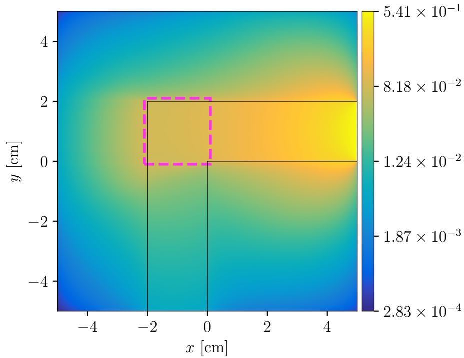


Free-flight Probability

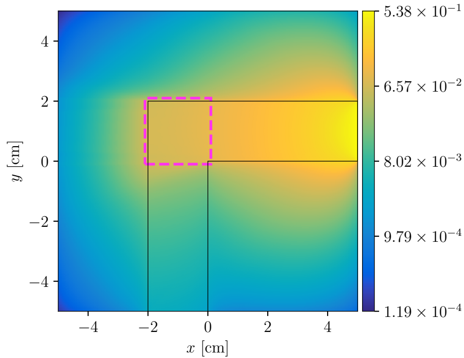


Source, $\widehat{Q}_2$

2-D, Right-angle Duct, Configuration B



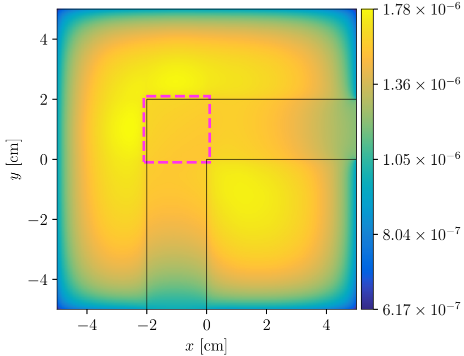
First HSME, $\widehat{\Psi}_1$



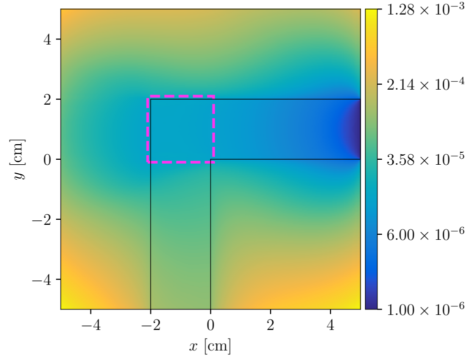
Second HSME, $\widehat{\Psi}_2$

MCNP6		COVRT		MCNP6/COVRT	
Mean	Variance	Mean	Variance	Mean	Variance
5.418×10^{-3}	5.388×10^{-3}	5.583×10^{-3}	5.552×10^{-3}	0.97	0.97
5.413×10^{-3}	3.262×10^{-3}	5.583×10^{-3}	3.377×10^{-3}	0.97	0.97

2-D, Right-angle Duct, Configuration B, cont.



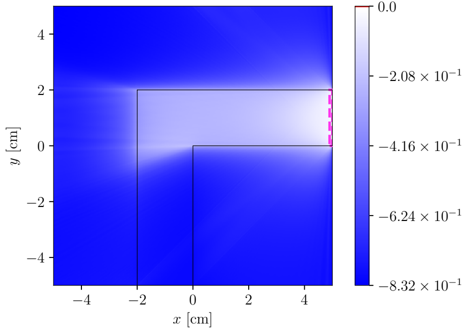
FTE, $\widehat{\Upsilon}$



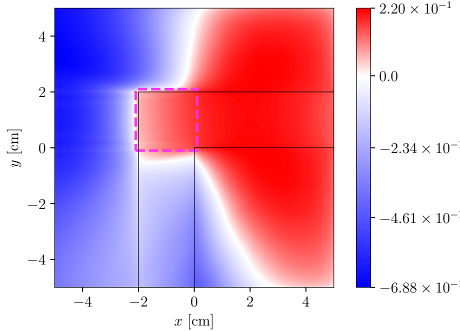
Cost, $\widehat{\$}$

- Optimization predicted a 64% improvement; 954% improvement realized
  - Optimizer moves the forced-flight region to be around the tally
  - The β_c values for this case are varied across the permissible range

Biased Computational Cost Ratio (Lower is Better)



Configuration A



Configuration B

- Ratio is

$$\frac{\$_{\text{forced flight}} - \$_{\text{analog}}}{\$_{\text{analog}}}$$

- Ratio less than one provides benefit if the forward source is present there

Summary & Future Work

- History-score moment and future-time equations
  - Evidence given of correct derivation and implementation
  - Can yield effective, optimized, biasing parameters
    - Demonstrated here; shown elsewhere in detail
  - Cost field can provide useful intuition regarding biasing properties

- Future work:
  - Increase size of calculations to "engineering scale"
  - Exercise 3-D capability

Questions?

Contact Information

Joel A. Kulesza
Office: +1 (505) 667-5467
Email: jkulesza@lanl.gov

Clell J. Solomon, Jr.
Office: +1 (505) 665-5720
Email: csolomon@lanl.gov

Brian C. Kiedrowski
Office: +1 (734) 615-5978
Email: bckiedro@umich.edu

Backup Slides

References

C. J. Solomon, “Discrete-Ordinates Cost Optimization Of Weight-Dependent Variance Reduction Techniques For Monte Carlo Neutral Particle Transport,” Ph.D. dissertation, Kansas State University, Manhatten, KS, USA, 2010. [Online]. Available: http://hdl.handle.net/2097/7014

C. Audet, S. Le Digabel, and C. Tribes, “NOMAD User Guide,” Les Cahiers du GERAD, Tech. Rep. G-2009-37, 2009.

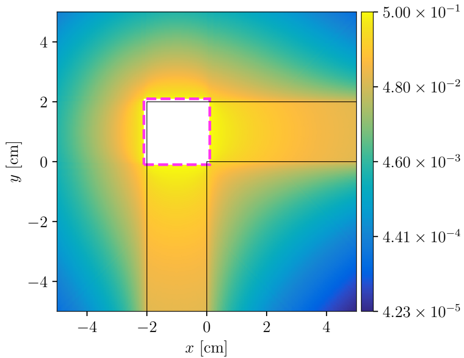
S. Le Digabel, “Algorithm 909: NOMAD: Nonlinear Optimization with the MADS Algorithm,” *ACM Transactions on Mathematical Software*, vol. 37, no. 4, Feb. 2011.

J. Weidendorfer, M. Kowarschik, and C. Trinitis, “A Tool Suite for Simulation Based Analysis of Memory Access Behavior,” in *Proceedings of the 4th International Conference on Computational Science (ICCS 2004)*, Krakow, Poland, Jun. 2004.

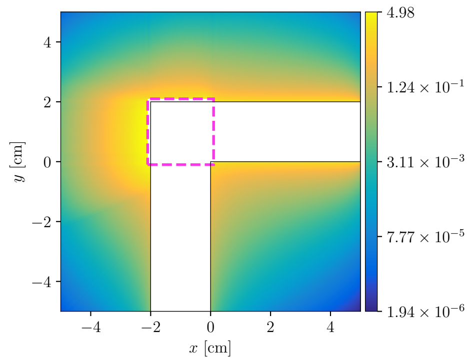
References

N. Nethercote and J. Seward, “Valgrind: A Framework for Heavyweight Dynamic Binary Instrumentation,” in *Proceedings of ACM SIGPLAN 2007 Conference on Programming Language Design and Implementation (PLDI 2007)*, San Diego, CA, USA, Jun. 2007.

2-D, Right-angle Duct, Configuration B, cont.



Free-flight Probability



Source, $\widehat{Q}_2$