

# Test of the Multiple k Eigenfunction Convergence Acceleration Method in MCNP

Thomas E. Booth

LA-UR 11-02222 April 13, 2011

## Abstract

A simple method for Monte Carlo source convergence acceleration was demonstrated on a simple one-speed one-dimensional nuclear criticality problem. The method used multiple eigenvalue estimates to achieve the acceleration. This report demonstrates that the same multiple k eigenfunction method also accelerates convergence on a continuous energy heterogeneous three-dimensional problem in MCNP.

## 1 Introduction

In the past few years a number of papers [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] have exploited the fact that the eigenvalue of a system can be computed at any point, or set of points, one wishes. It is common to compute the eigenvalue from global quantities, but the most basic definition of an eigenfunction and eigenvalue for a linear operator  $A$  is:

$$A\psi(s) = k\psi(s) \quad (1)$$

Note that this eigenvalue/eigenfunction relation is a pointwise relation at every  $s$ , rather than a global relation. Of course, one can *derive* a global relation for the fundamental eigenvalue by integrating to obtain:

$$\int A\psi_1(s)ds = k \int \psi_1(s)ds \quad (2)$$

This paper uses the basic definition Eq. 1 of the eigenvalue as a pointwise relation rather than the derived expression of Eq. 2. Because Eq. 1 can be integrated over any region  $R_i$ ,

$$\int_{R_i} A\psi_1(s)ds = k \int_{R_i} \psi_1(s)ds \quad (3)$$

This allows one to compute  $k$  using any number of regions. That is,

$$k_{R_i} = k = \frac{\int_{R_i} A\psi_1(s)ds}{\int_{R_i} \psi_1(s)ds} \quad (4)$$

The multiple  $k$  source convergence acceleration method showed promising results [10] on a simple problem in a toy transport code. In particular, the method was demonstrated on a problem so simple that an analytic solution was available.

Sometimes methods that work quite well on homogeneous one-speed and one-dimensional problems do not work well when the problems get more complicated. This paper shows that the multiple  $k$  method can also work for some more complicated problems. Whether the method is currently practical enough to implement for general MCNP eigenvalue calculations remains to be seen, but this paper shows that the method is worth investigating further for potential implementation.

The method is a modification of the power iteration method in which multiple eigenvalue estimates in multiple regions are used instead of one eigenvalue estimate over the entire system. The eigenvalue estimates involve an unknown weight multiplier in each region. These unknown weight multipliers are then determined by requiring the eigenvalue estimates in all regions to be the same because at any point  $s$  for which  $\psi(s) \neq 0$ , the definition of the eigenfunction means that  $k_1 = (A\psi)/\psi$ .

Note that no claim is made that the method herein is better than any existing method.

## 2 Requiring that $k$ be constant - the Weight Multiplier Equation

This section uses the fact that  $k$  must be the same in all regions. In order to enforce the requirement, the particles in  $R_i$  have an unknown weight multiplier  $z_i$ . The  $k_j$  are then functions of the weight multipliers  $z_i$ . One then solves for a set of  $z_i$  that produces  $k_1 = k_2 = k_3 = \dots = k_n$ . (Because the eigenvalue relationship does not depend on the magnitude of the eigenvector, if  $z_i$  produces  $k_1 = k_2 = k_3 = \dots = k_n$ , then so will  $Z_j = \text{constant} \times z_j$ .)

Before solving for the weight multipliers that equalize the  $k_j$ , a few terms need to be defined

1.  $M_j$  is the (unweighted) number of particles started in state  $j$ .
2.  $M_{ij}$  is the (unweighted) number of fission particles produced in state  $i$  by the  $M_j$  particles.
3.  $x_i = z_i M_i$  is the weighted number of particles started in state  $j$ .
4.  $y_i = \sum_j z_j M_{ij}$  is the weighted number of fission particles produced in state  $i$ .

The eigenvalue estimate in region  $i$  is

$$q_i = \frac{\sum_j z_j M_{ij}}{z_i M_i} = \frac{y_i}{x_i} \tag{5}$$

Requiring equality of the eigenvalues

$$q = \frac{\sum_j z_j M_{ij}}{z_i M_i} = \frac{y_i}{x_i} \quad i = 1, \dots, n \quad (6)$$

This is a set of  $n$  equations for the  $n + 1$  unknowns  $z_i$  and  $q$ . Normalizing the total resulting fission weight to some convenient value

$$W_0 = \sum_j z_j M_{ij} \quad (7)$$

provides the final equation necessary to solve for the weight multipliers  $z_i$ .

After the  $z_j$  are obtained, the Monte Carlo fission particle weights are updated by multiplying the weight of each fission particle produced by  $z_i$  corresponding to the region  $R_i$  where the particle started. A new fission source can then be produced for the next cycle (e.g. by combing) with the desired number of source particles.

### 3 The Tenth Density Fuel Vault Problem

The standard benchmark fuel vault problem was too difficult to run on desktop type computers and obtain statistically accurate weight multipliers, so I reduced all densities to one tenth of their original densities so that the problem was doable on a desktop computer. Appendix A gives the input file for the problem.

The problem was divided into 36 regions specified by

That is, there are 12  $x$  regions, 1  $y$  region, and 3  $z$  regions. Figs. 1-4 show some MCNP plots of the geometry. The boundaries of the 12 regions in  $x$  are

$$\begin{aligned} x_0 &= -13.5 \\ x_1 &= 40.5 \\ x_2 &= 94.5 \\ x_3 &= 148.5 \\ x_4 &= 202.5 \\ x_5 &= 256.5 \\ x_6 &= 310.5 \\ x_7 &= 364.5 \\ x_8 &= 418.5 \\ x_9 &= 472.5 \\ x_{10} &= 526.5 \\ x_{11} &= 580.5 \end{aligned}$$

$$x_{12} = 634.5$$

The boundaries of the single region in  $y$  are

$$y_0 = -13.5$$

$$y_1 = 67.5$$

The boundaries of the 2 regions in  $z$  are

$$z_0 = -180$$

$$z_1 = -60$$

$$z_2 = 60$$

$$z_3 = 180$$

## 4 Convergence Results for the Tenth Density Fuel Vault Problem

At convergence, all the  $z_i$  (weight multipliers) have to (within statistical errors) be equal to one. Regions 1 to 12 are the twelve  $x$  regions for the bottom of the fuel vault, regions 13 to 24 are the twelve  $x$  regions for the middle of the fuel vault, and regions 25 to 36 are the twelve  $x$  regions for the top of the fuel vault. A total of 10 million particles were used on each iteration. The initial distribution (entering iteration 1) put 5 million particles at the single point (625, 27.5, -179) in region 12 and the other 5 million particles were started uniformly over all regions.

Fig. 5 shows the iteration by iteration change for standard MCNP in the weight multipliers in the 36 regions. Note that the required weight multipliers do seem to be getting closer to 1.0 with each iteration. Fig. 6 shows the change every 10 iterations. Again, the weight multipliers are getting closer to 1. (Note the change in scale between Figs. 5 and 6.)

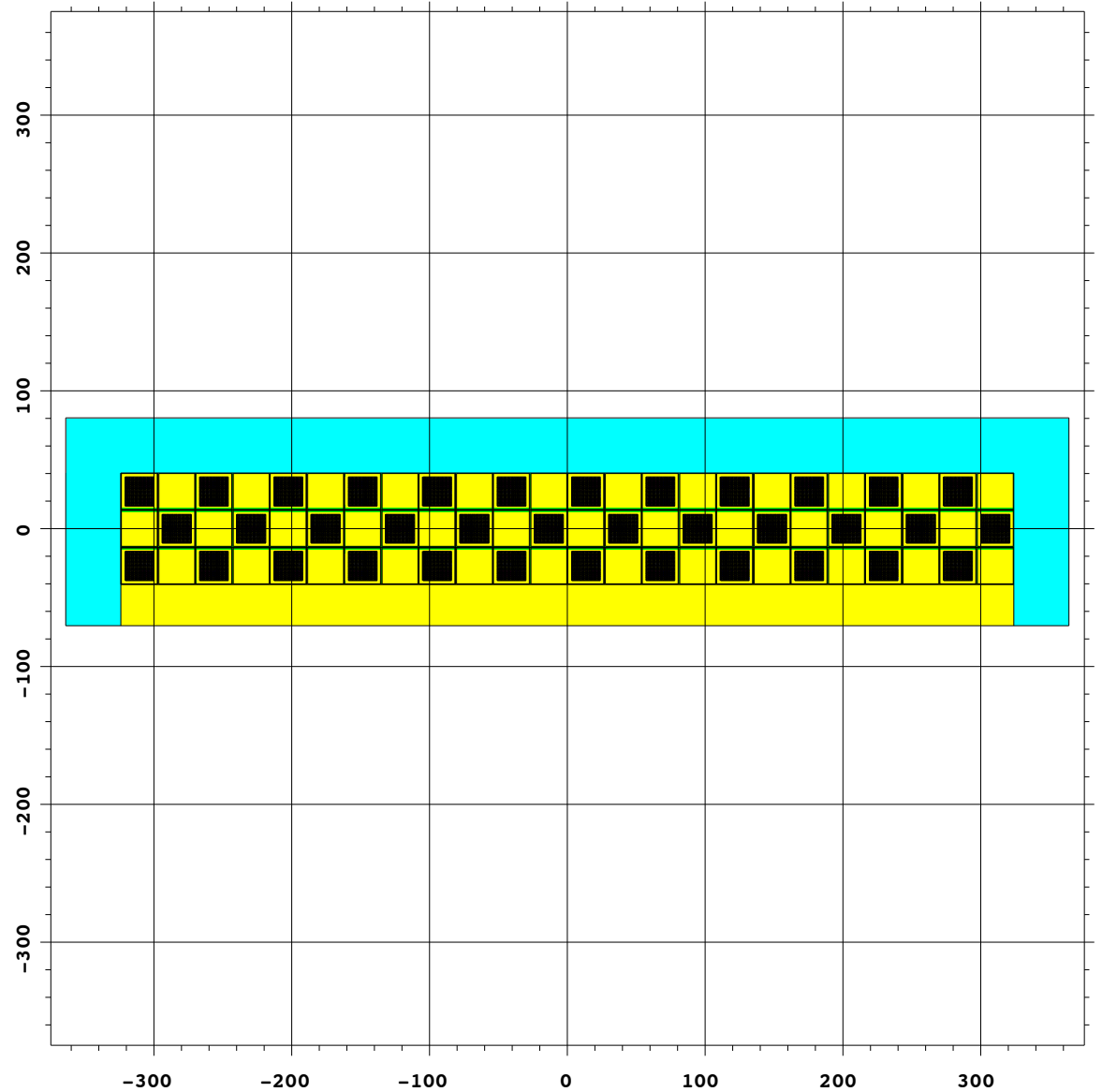
Fig. 7 shows the iteration by iteration change for the multiple k method in the weight multipliers in the 36 regions. Note that the required weight multipliers approach 1.0 far faster than in Fig. 5. Fig. 8 regraphs the same data in Fig. 7, but on the same scale as Fig. 6. Note that just a few iterations with the multiple k method achieve the results similar to a few *tens* of iterations with the standard MCNP method. Fig. 9 emphasizes this point showing the weight multipliers for the fourth iteration on the multiple k method against the 49-th and 50-th iteration for standard MCNP.

It is also interesting to look at the source points in each region by iteration. Although, unlike the weight multipliers, one does not know the correct value for the converged source points, one does know that the source distribution should become constant with iteration at convergence. Figs. 10 and 11 show the source convergence for standard MCNP. Figs. 12 and 13 show the source convergence

04/05/11 14:55:56  
Problem fvf - Fuel storage  
vault

probid = 04/05/11 14:54:55  
basis: XY  
( 1.000000, 0.000000, 0.000000)  
( 0.000000, 1.000000, 0.000000)  
origin:  
( 310.50, 27.00, 0.00)  
extent = ( 375.00, 375.00)

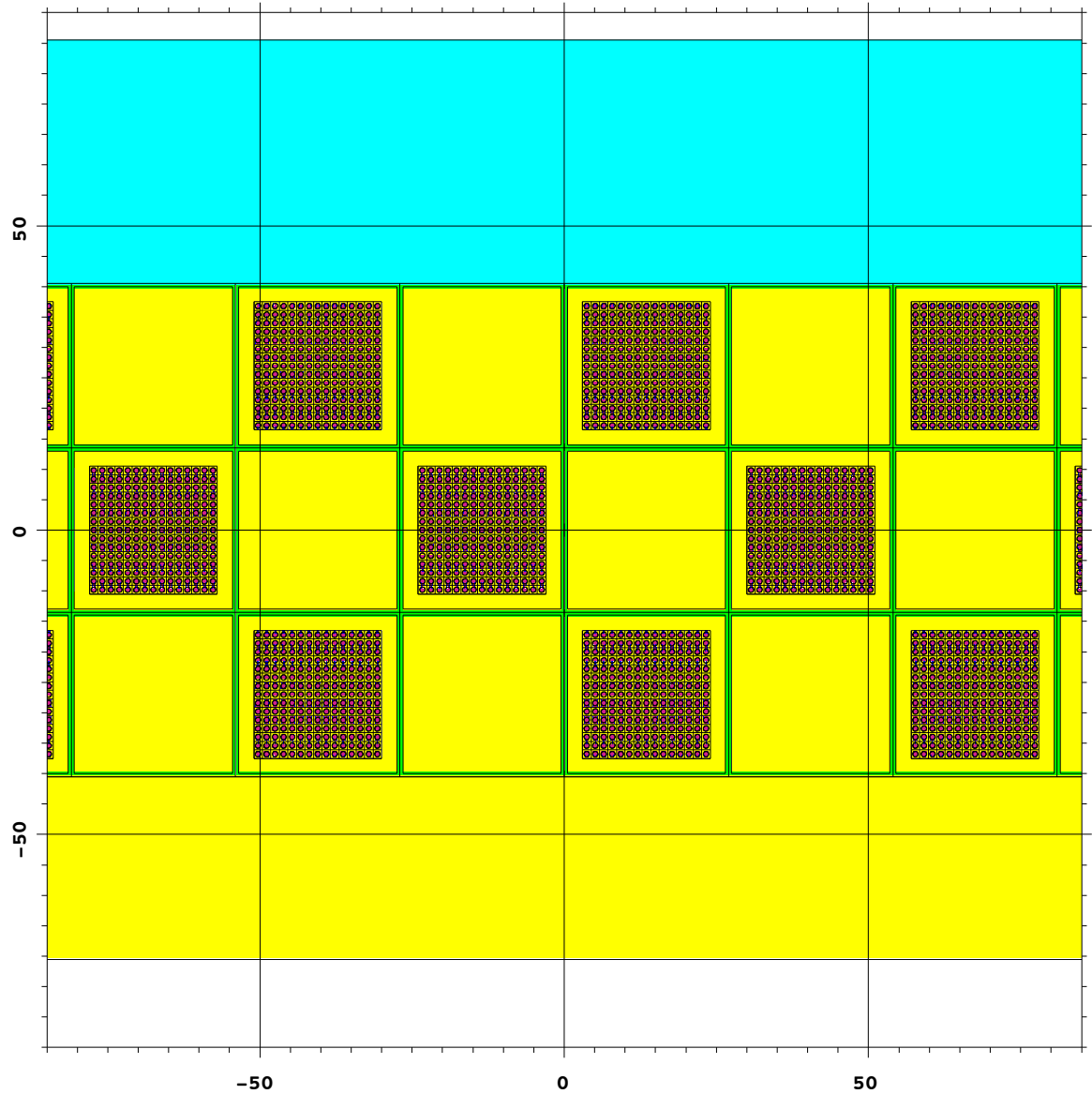
Figure 1: Fuel vault geometry xy plot large scale



04/05/11 15:04:19  
Problem fvf - Fuel storage  
vault

probid = 04/05/11 15:03:18  
basis: XY  
( 1.000000, 0.000000, 0.000000)  
( 0.000000, 1.000000, 0.000000)  
origin:  
( 310.50, 27.00, 0.00)  
extent = ( 85.00, 85.00)

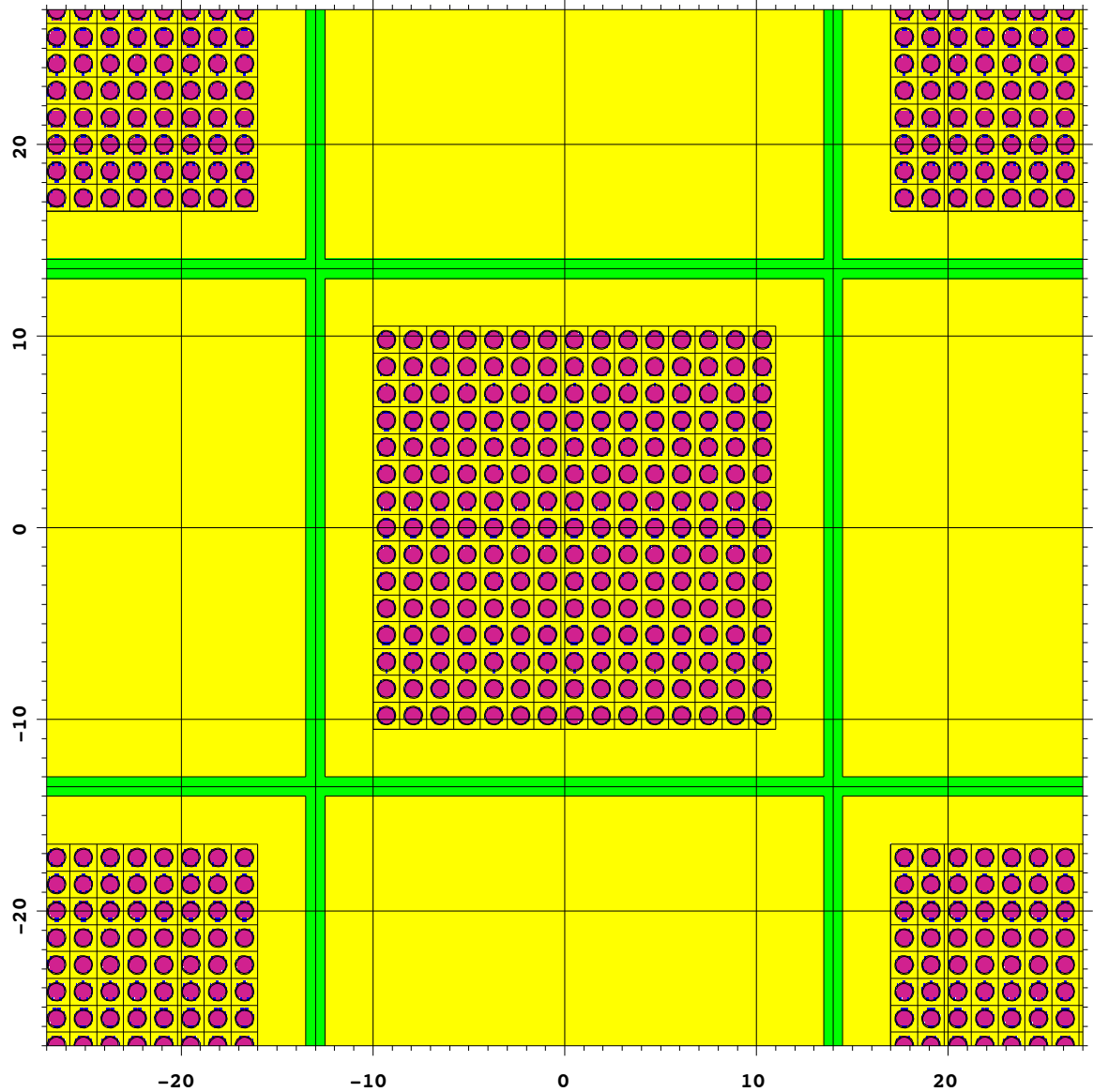
Figure 2: Fuel vault geometry xy plot medium scale



04/05/11 14:52:04  
Problem fvf - Fuel storage  
vault

probid = 04/05/11 14:50:53  
basis: XY  
( 1.000000, 0.000000, 0.000000)  
( 0.000000, 1.000000, 0.000000)  
origin:  
( 296.50, 27.00, 0.00)  
extent = ( 27.00, 27.00)

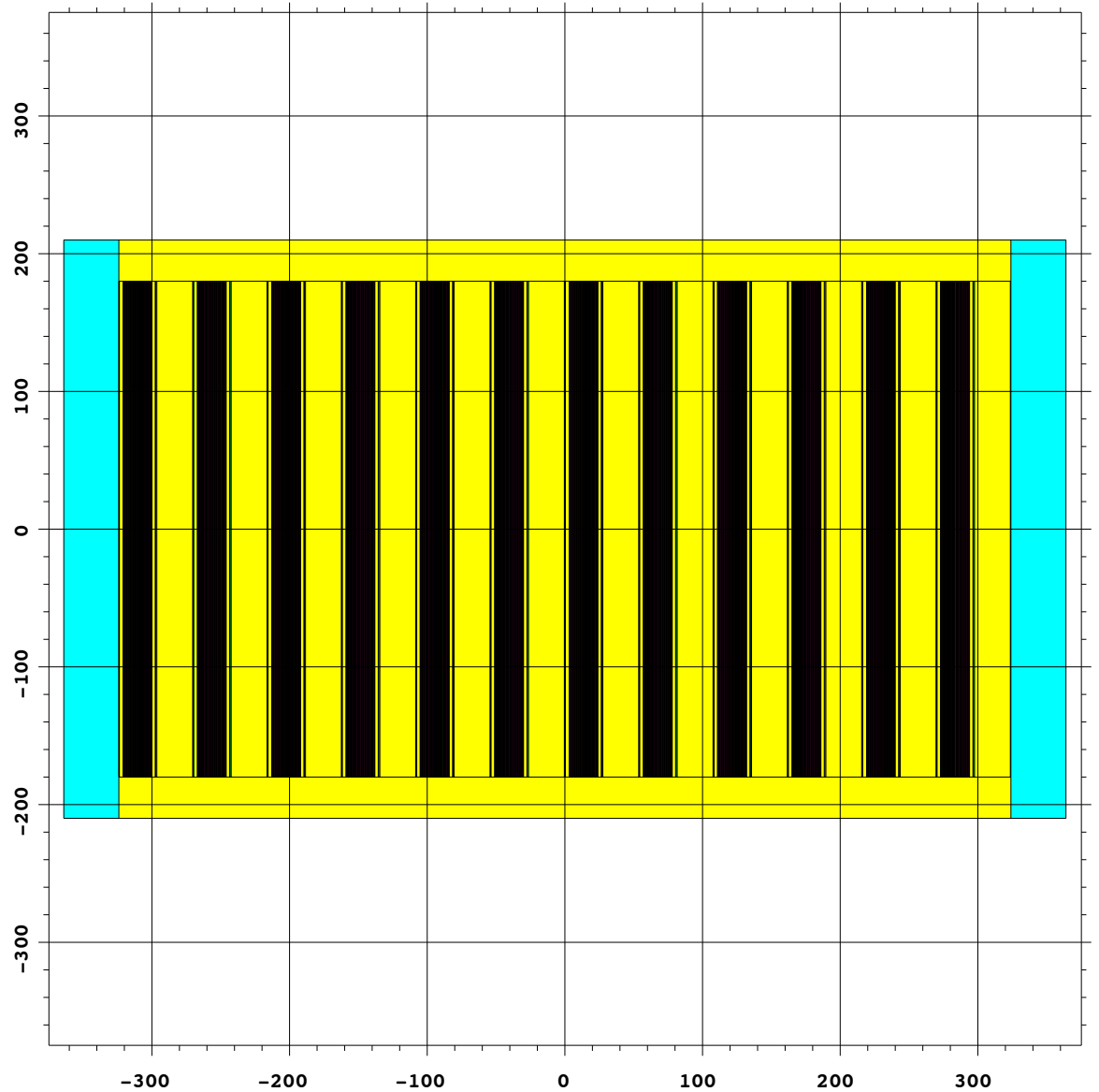
Figure 3: Fuel vault geometry xy plot small scale



04/05/11 15:12:04  
Problem fvf - Fuel storage  
vault

probid = 04/05/11 15:09:56  
basis: XZ  
( 1.000000, 0.000000, 0.000000)  
( 0.000000, 0.000000, 1.000000)  
origin:  
( 310.50, 0.00, 0.00)  
extent = ( 375.00, 375.00)

Figure 4: Fuel vault geometry xz plot



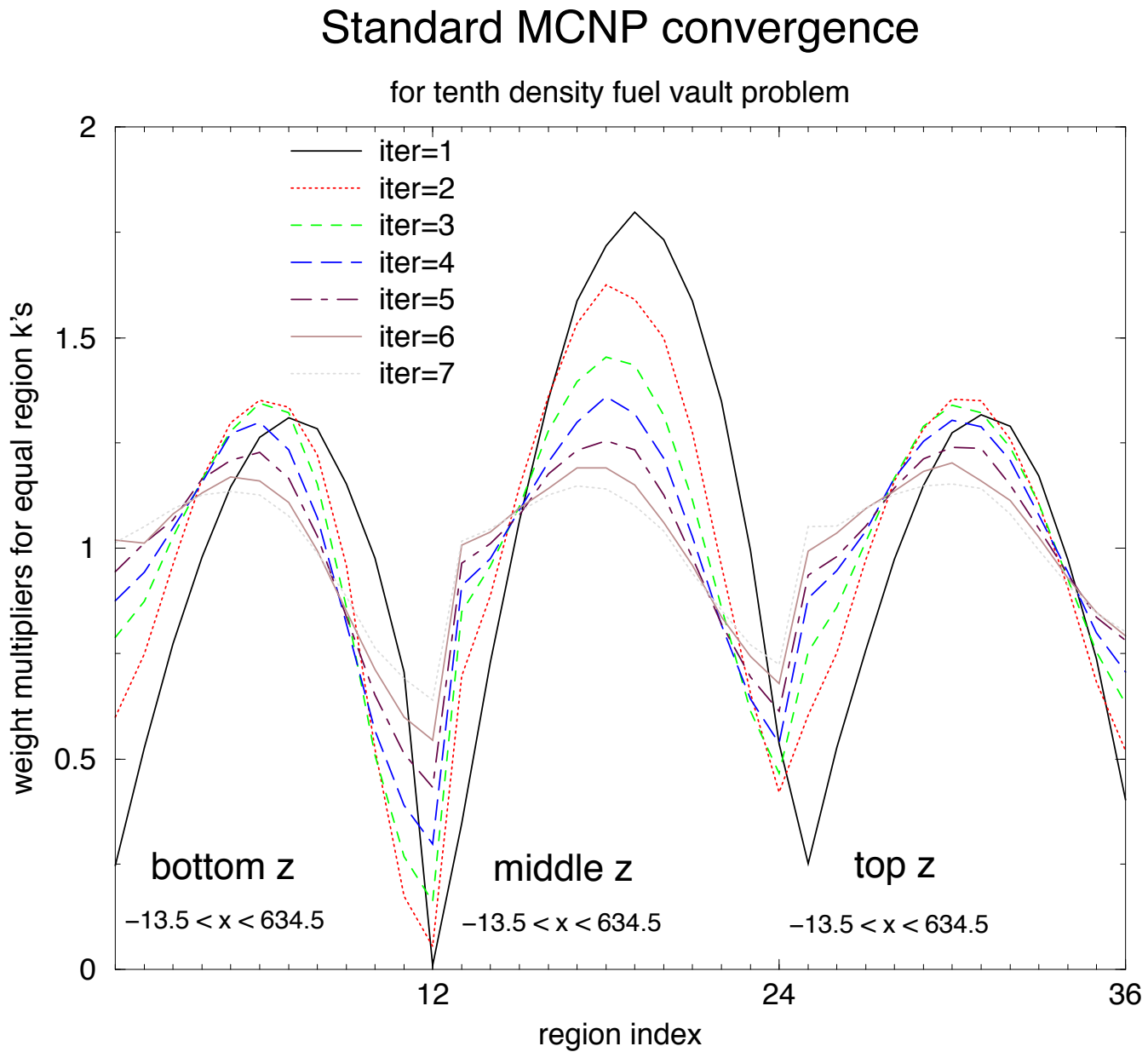


for the multiple k method. The spike in region 12 at the initial iteration is due to the 5 million initial source particles started at the point (625, 27.5, -179) in region 12. To see the accelerated convergence in the source distribution, compare standard MCNP in Fig. 11 with the multiple k method in Fig. 13.

## 5 Summary and Future Work

The results presented here demonstrate that the multiple k source acceleration method works on more than just toy problems. My intention is to try the method on the full density fuel vault problem once I figure out why I am having trouble with my patched version of MCNP on the high performance computers.

Figure 5: Standard MCNP convergence weight multipliers for iterations 1-7



# Standard MCNP convergence

for tenth density fuel vault problem

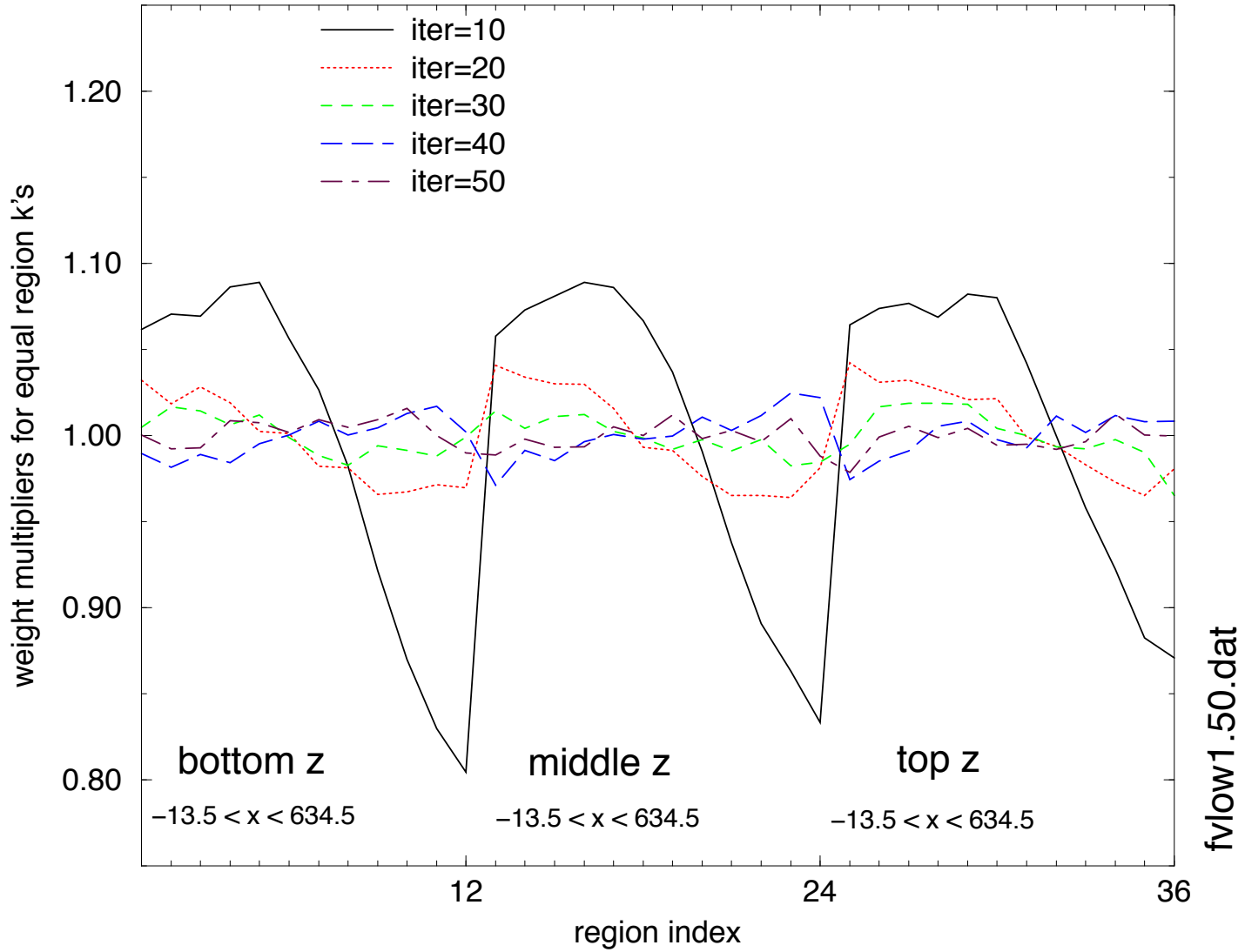


Figure 6: Standard MCNP convergence weight multipliers for iterations 10, 20, 30, 40, and 50

# Multiple k MCNP convergence

for tenth density fuel vault problem

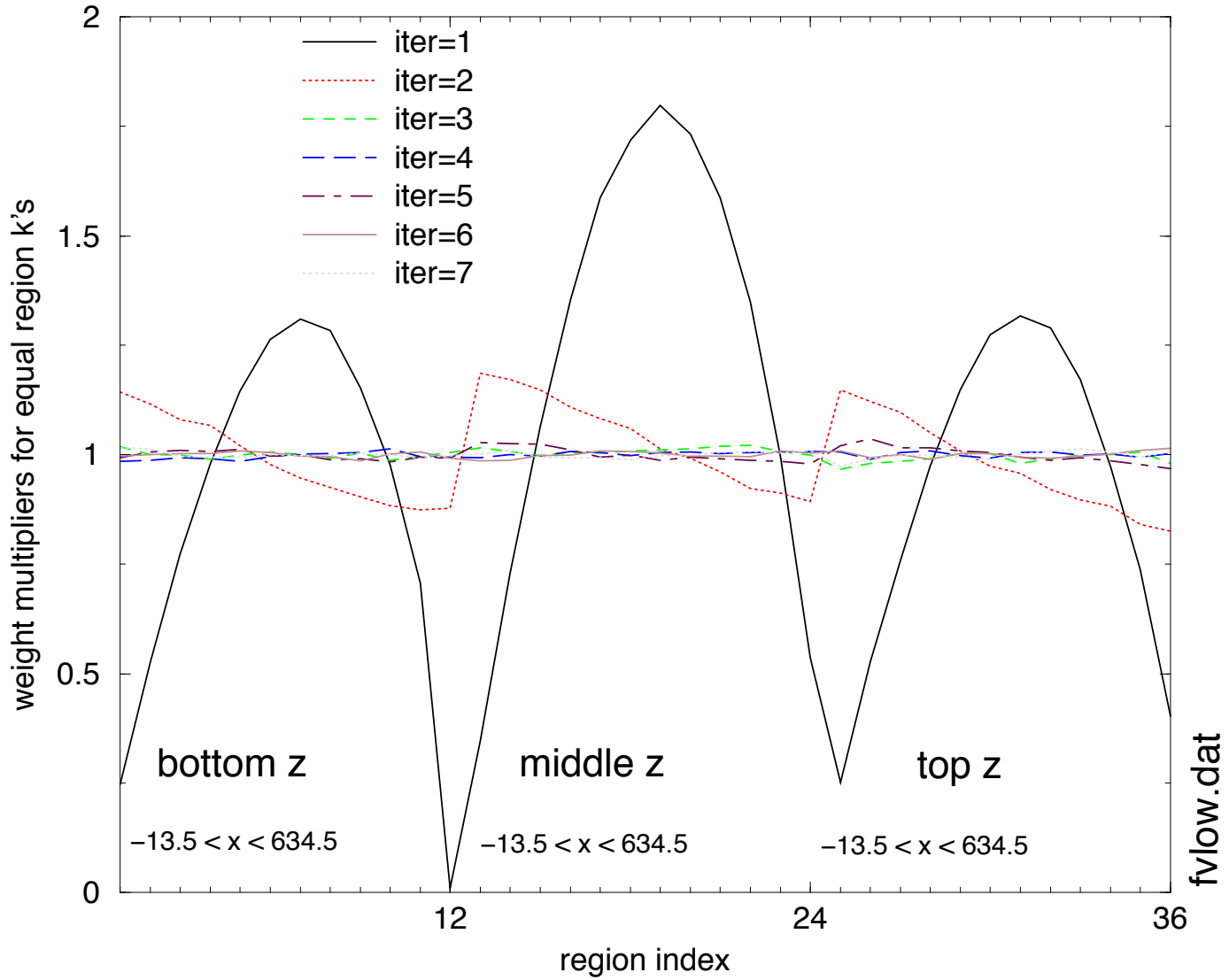


Figure 7: Multiple k convergence weight multipliers for iterations 1-7

# Multiple k MCNP convergence

for tenth density fuel vault problem

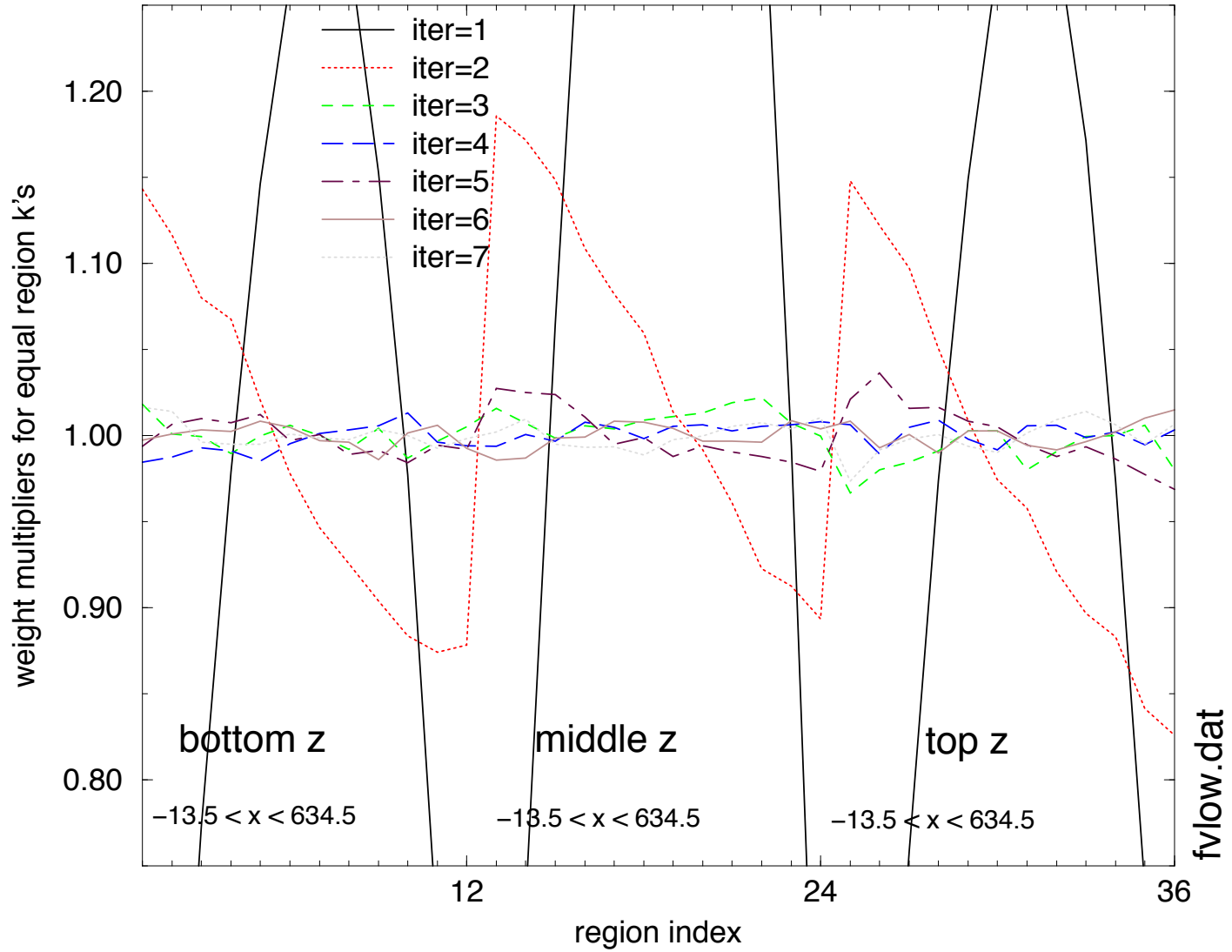


Figure 8: Multiple k convergence weight multipliers for iterations 1-7, matching scale to Fig. 6

# Standard vs Multiple k MCNP convergence

for tenth density fuel vault problem

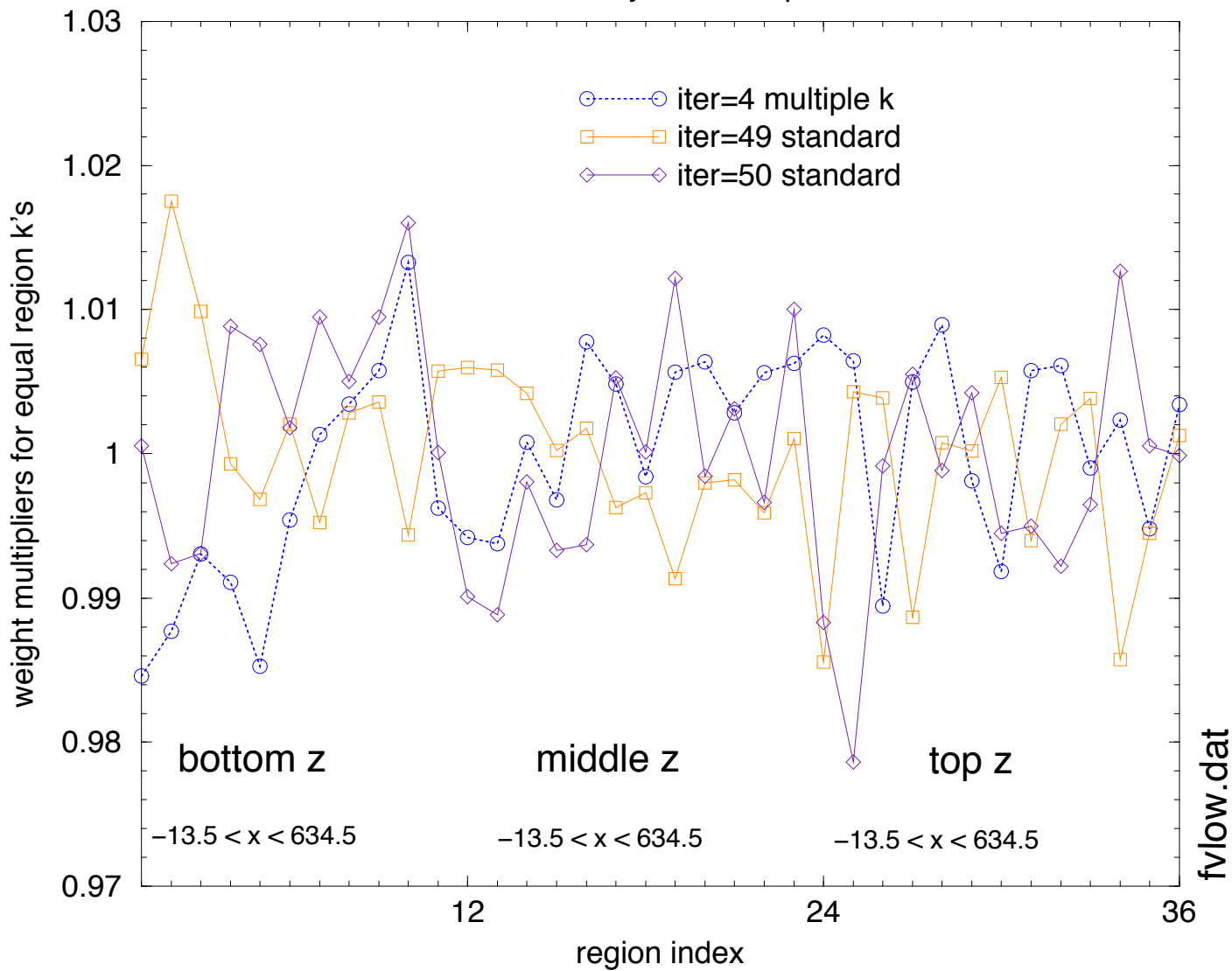


Figure 9: Standard MCNP vs multiple k converged weight multipliers

# Standard MCNP : Source points by cycle

tenth density fuel vault (10 million source particles)

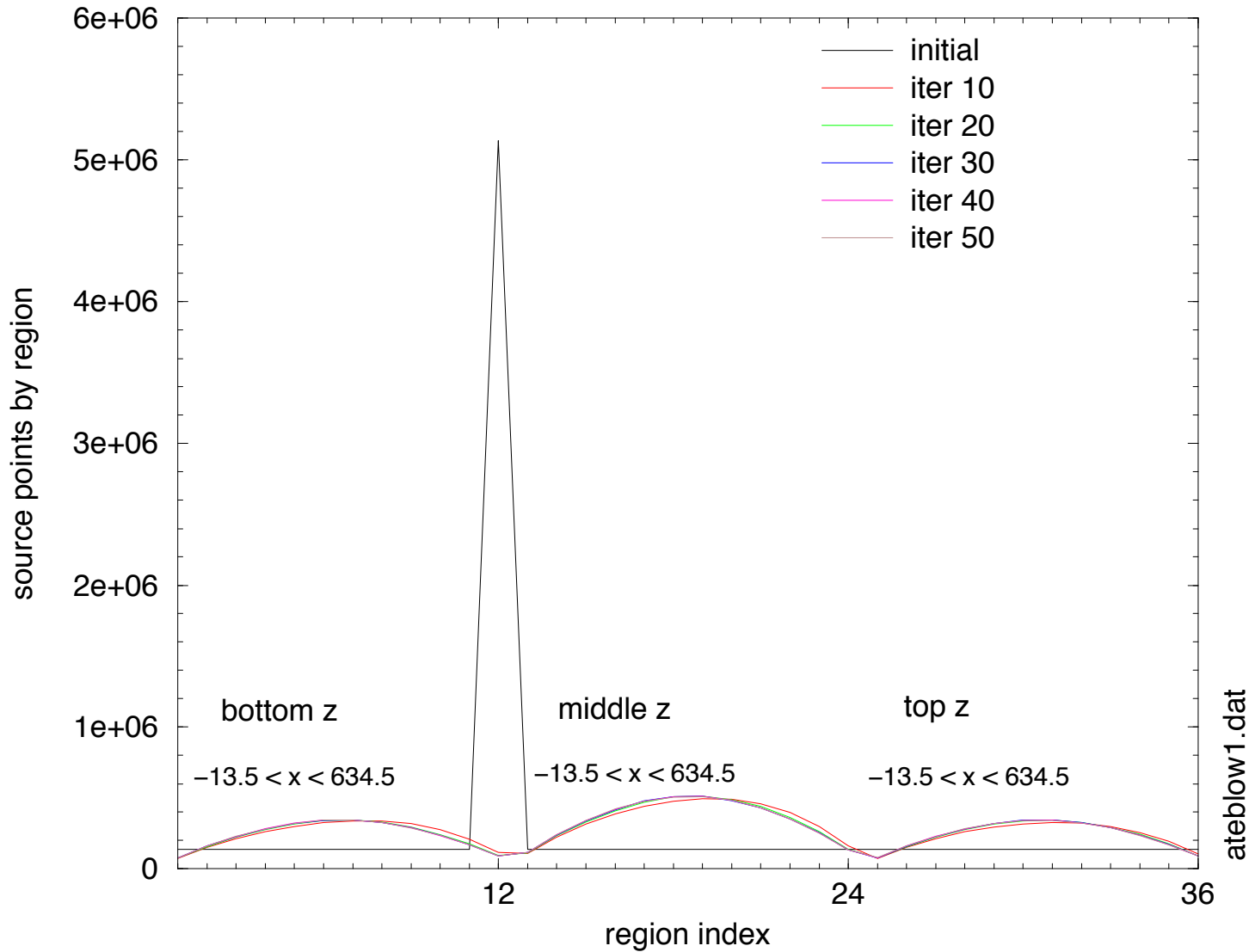


Figure 10: Standard MCNP: source points by cycle  
15

ateblow1.dat

# Standard MCNP : Source points by cycle

tenth density fuel vault (10 million source particles)

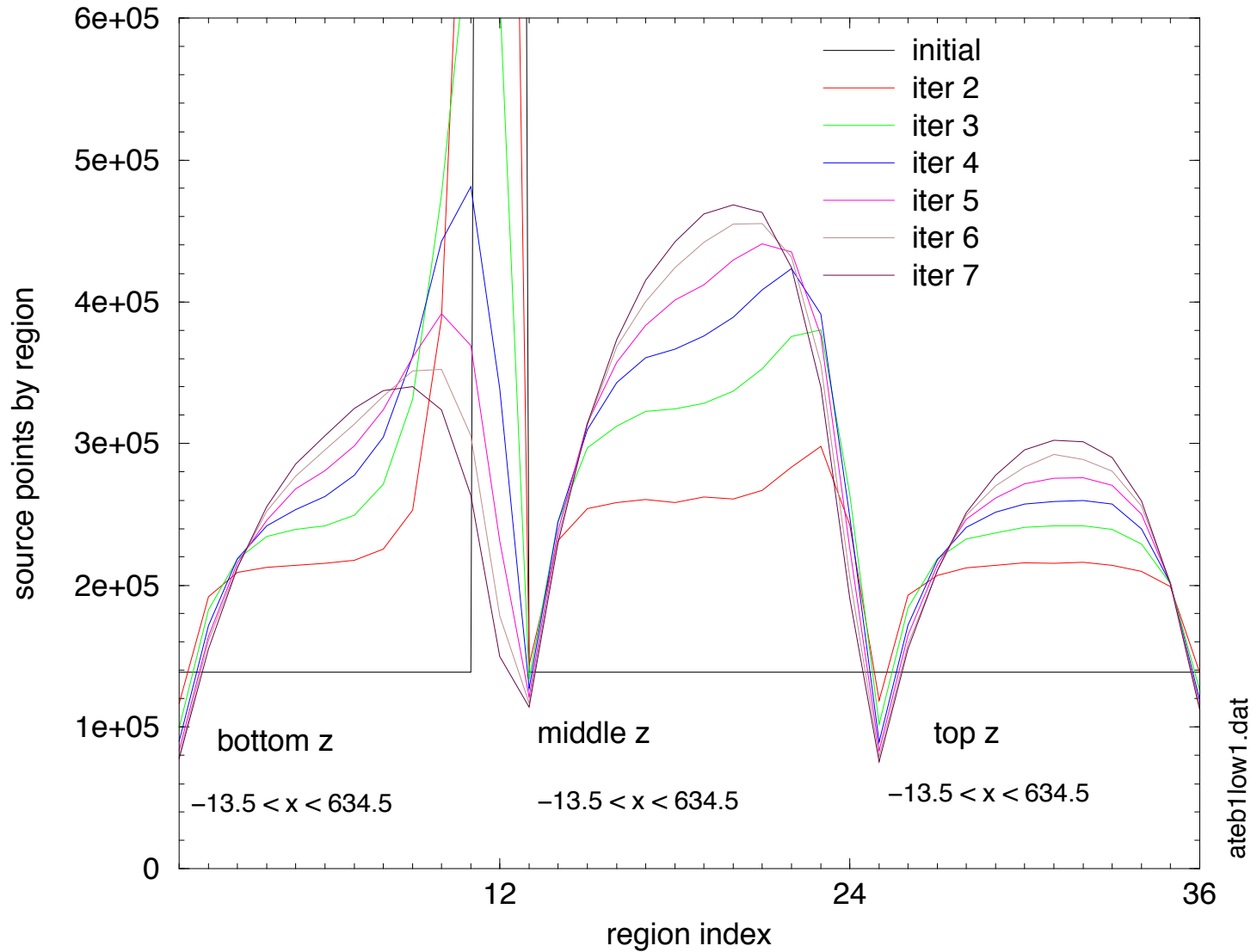


Figure 11: Standard MCNP: source points on cycles 1-7



# Multiple k : Source points by cycle

tenth density fuel vault (10 million source particles)

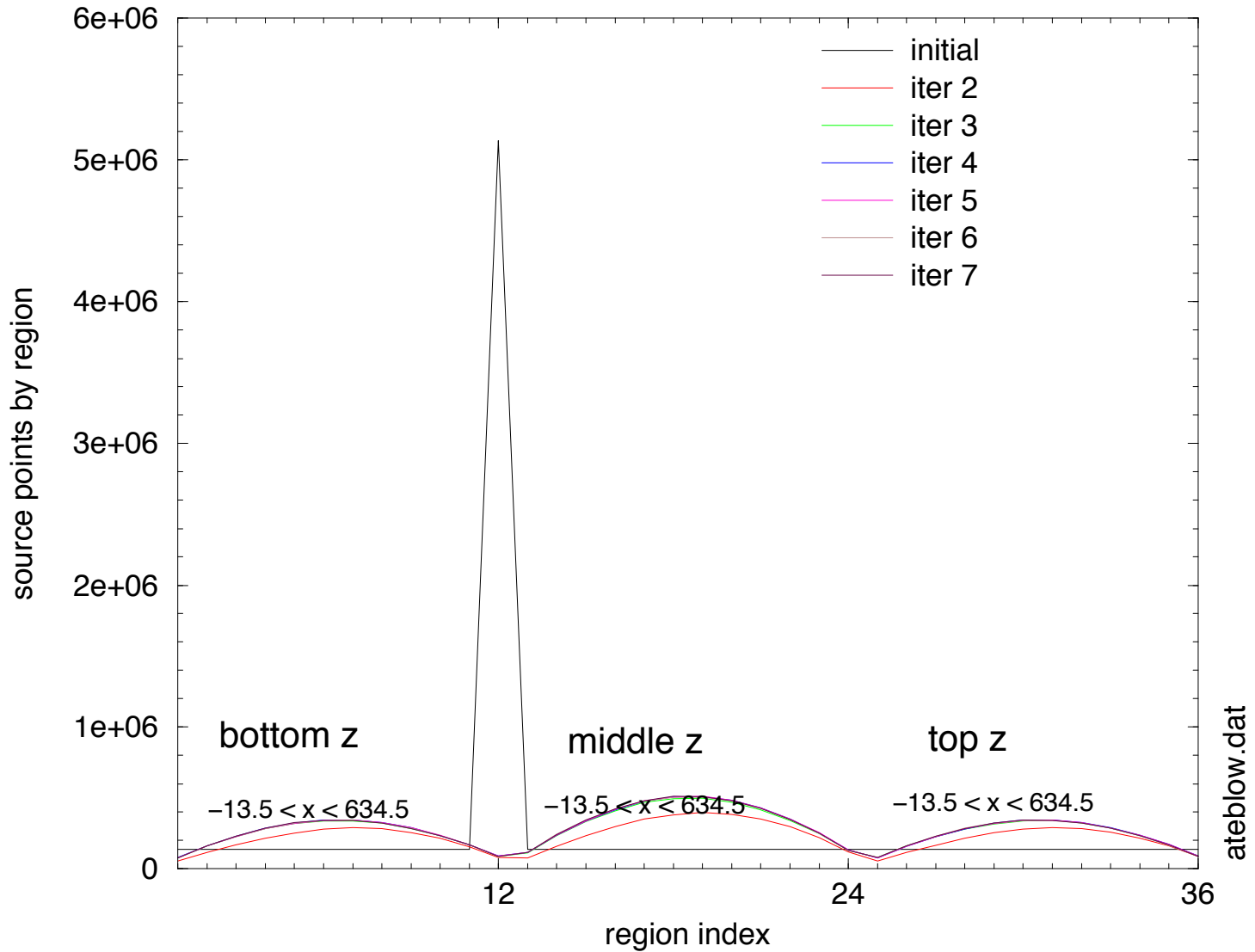


Figure 12: Multiple k method: source points on cycles 1-7  
17

# Multiple k : Source points by cycle

tenth density fuel vault (10 million source particles)

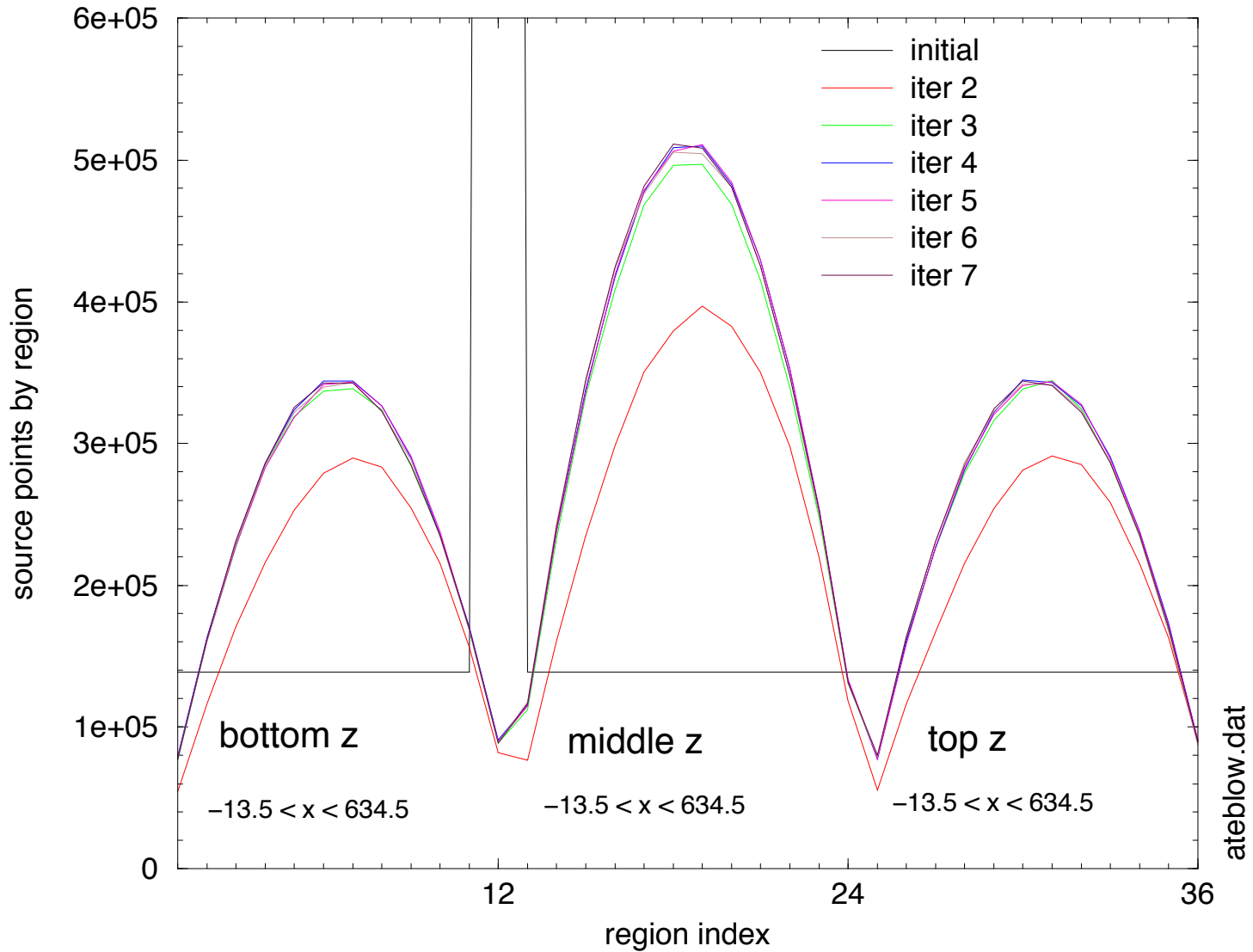


Figure 13: Multiple k method: source points on cycles 1-7, changed scale

## References

- [1] T. E. Booth, "Computing the Higher k-Eigenfunctions by Monte Carlo Power Iteration: A Conjecture," *Nucl. Sci. Eng.*, **143**, pp. 291-300 (2003).
- [2] T. E. Booth, "Power Iteration Method for the Several Largest Eigenvalues and Eigenfunctions," *Nucl. Sci. Eng.*, **154**, pp. 48-62 (2006).
- [3] J.E. Gubernatis and T.E. Booth, "Multiple Extremal Eigenpairs by the Power Method," *Journal of Computational Physics*, **227**, pp 8508-8522 (2008)
- [4] T.E. Booth and J.E. Gubernatis, "Multiple Extremal Eigenpairs of Very Large Matrices by Monte Carlo Simulation," Los Alamos Report LA-UR-08-0043, arXiv:0807.1273 (July 2008)
- [5] T.E. Booth and J.E. Gubernatis, "Monte Carlo Determination of Multiple Extremal Eigenpairs," *Physical Review E*, **80**, pp 046704-046713 (2009).
- [6] T. Yamamoto, "Convergence of the Second Eigenfunction in Monte Carlo Power Iteration", *Annals of Nuclear Energy*, **36**, pp. 7-14 (2009)
- [7] T. E. Booth and J. E. Gubernatis, "Improved Criticality Convergence Via a Modified Monte Carlo Power Iteration," International Conference on Mathematics, Computational Methods & Reactor Physics (M&C 2009), Saratoga Springs, New York, May 3-7, 2009, on CD-ROM. Los Alamos Report LA-UR-08-06461.
- [8] T. E. Booth and J. E. Gubernatis, "Exact Regional Monte Carlo Weight Cancellation for Second Eigenfunction Calculations," *Nuclear Science and Engineering*, **165**, pp. 283-291 (2010).
- [9] Bo Shi and Bojan Petrovic, "Implementation of the modified power iteration method to two-group Monte Carlo eigenvalue problems," *Annals of Nuclear Energy*, Volume 38, Issue 4, April 2011, Pages 781-787
- [10] "A Simple Eigenfunction Convergence Acceleration Method for Monte Carlo", Thomas E. Booth. *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2011)* Rio de Janeiro, RJ, Brazil, May 8-12, 2011, on CD-ROM, Latin American Section (LAS) / American Nuclear Society (ANS) ISBN 978-85-63688-00-2 (also LA-UR 10-07763)
- [11] X-5 Monte Carlo Team, "MCNP-A General Monte Carlo N-Particle Transport Code, Version 5," Los Alamos National Laboratory Report LA-UR-03-1987, April 24, 2003

## 6 Appendix: Tenth Density Fuel Vault MCNP Input File

```

Problem fvf - Fuel storage vault
c
c CELLS
10 100 0.06925613e-1 -1 u=1 $ fuel
20 200 0.042910e-1 1 -2 u=1 $ clad
30 300 0.100059e-1 2 u=1 $ water
c =====> fuel lattice, infinite array of pins in water
40 0 -3 fill=1 lat=1 u=2
c =====> fuel element
50 0 -4 fill=2 u=3 $ fuel lattice
60 300 0.100059e-1 4 -5 u=3 $ water gap
70 400 0.083770e-1 5 u=3 $ Fe
c =====> water element
80 300 0.100059e-1 -5 u=4 $ water
90 400 0.083770e-1 5 u=4 $ Fe
c =====> element lattice, infinite
100 0 -6 u=5 lat=1 fill= 0:23 0:2 0:0
    3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4
    4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3
    3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4
c =====> full model
110 0 -7 fill=5 $ lattice of elements
120 300 0.100059e-1 -8 7 $ outside water
130 500 0.0725757e-1 -9 8 $ outside concrete
140 0 9

c SURFACES
1 RCC 0. 0. -180. 0. 0. 360. 0.44
2 RCC 0. 0. -180. 0. 0. 360. 0.49
3 RPP -.7 .7 -.7 .7 -180. 180.
4 RPP -10.5 10.5 -10.5 10.5 -210. 210.
5 RPP -13.0 13.0 -13.0 13.0 -210. 210.
6 RPP -13.5 13.5 -13.5 13.5 -210. 210.
7 RPP -13.5 634.5 -13.5 67.5 -180. 180.
8 RPP -13.5 634.5 -43.5 67.5 -210. 210.
9 RPP -53.5 674.5 -43.5 107.5 -210. 210.

c DATA
imp:n 1 12r 0
c
kcode 10000000 .3 50 100 20000000
ksrc 0 0 0 0 0 0 0 0 0 0 0 0 295 0 0 301 0 0

```

```

c
c =====> material cards
m100  92238 2.2380e-2  92235 8.2213e-4  8016 4.6054e-2  $ fuel
m200  40000 4.2910e-2                                     $ clad
m300   1001 6.6706e-2    8016 3.3353e-2                                     $ water
mt300  lwtr
m400  26000 8.3770e-2                                     $ Fe
m500   1001 5.5437e-3  6000 6.9793e-3 14000 7.7106e-3  $ concrete
      20000 8.9591e-3  8016 4.3383e-2
mt500  lwtr
prdmp  1 1 1 1 j

```