A Multiple Eigenvalue Power Iteration Convergence Metric

Thomas E. Booth

LA-UR 11-02223 April 13, 2011

Abstract

This report suggests a possible source convergence assessment method based on multiple eigenvalue estimates. The method is illustrated on a sample fission source convergence 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. This opens up the possibility of using a *difference* in eigenvalue estimates in different regions as a measure of how far the source distribution is from convergence. At convergence, all the eigenvalue estimates should be equal. A natural metric is to ask how much the source distribution would have to change in order that all eigenvalue estimates would be equal. (One can also use the knowledge of how far the system is from convergence to accelerate the convergence, but that is a subject already discussed in [10] and is not the focus of this note.)

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:

$$A\psi(s) = k\psi(s) \tag{1}$$

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:

$$\int A\psi_1(s)ds = k\int \psi_1(s)ds \tag{2}$$

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,

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

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

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

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 $k_1 = k_2 = k_3 = \cdots = k_n$. (Because the eigenvalue relationship does not depend on the magnitude of the eigenvector, if z_i produces $k_1 = k_2 = k_3 = \cdots = k_n$, then so will $Z_j = constant \times z_j$.)

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

$$q = \frac{\sum_j z_j M_{ij}}{z_i M_i} = \frac{y_i}{x_i} \qquad i = 1, \ldots, n \tag{6}$$

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

$$W_0 = \sum_j z_j M_{ij} \tag{7}$$

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

As mentioned earlier, when convergence is achieved, no weight multipliers are needed, i.e., $z_i = 1$. The more the z_i differ from one, the farther the system is from convergence. This property is demonstrated on the test problem in the next section. Other than calculating the z_i, the test problem is run in standard fashion with MCNP; the z_i are not used in the transport process.

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

$$x_0 = -13.5$$

$$x_1 = 40.5$$

$$x_2 = 94.5$$

$$x_3 = 148.5$$

$$x_4 = 202.5$$

$$x_5 = 256.5$$

$$x_6 = 310.5$$

$$x_7 = 364.5$$

$$x_8 = 418.5$$

$$x_9 = 472.5$$

$$x_{10} = 526.5$$

$$x_{11} = 580.5$$

$$x_{12} = 634.5$$

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

$$z_0 = -180$$

$$z_1 = -60$$

$$z_2 = 60$$

$$z_3 = 180$$

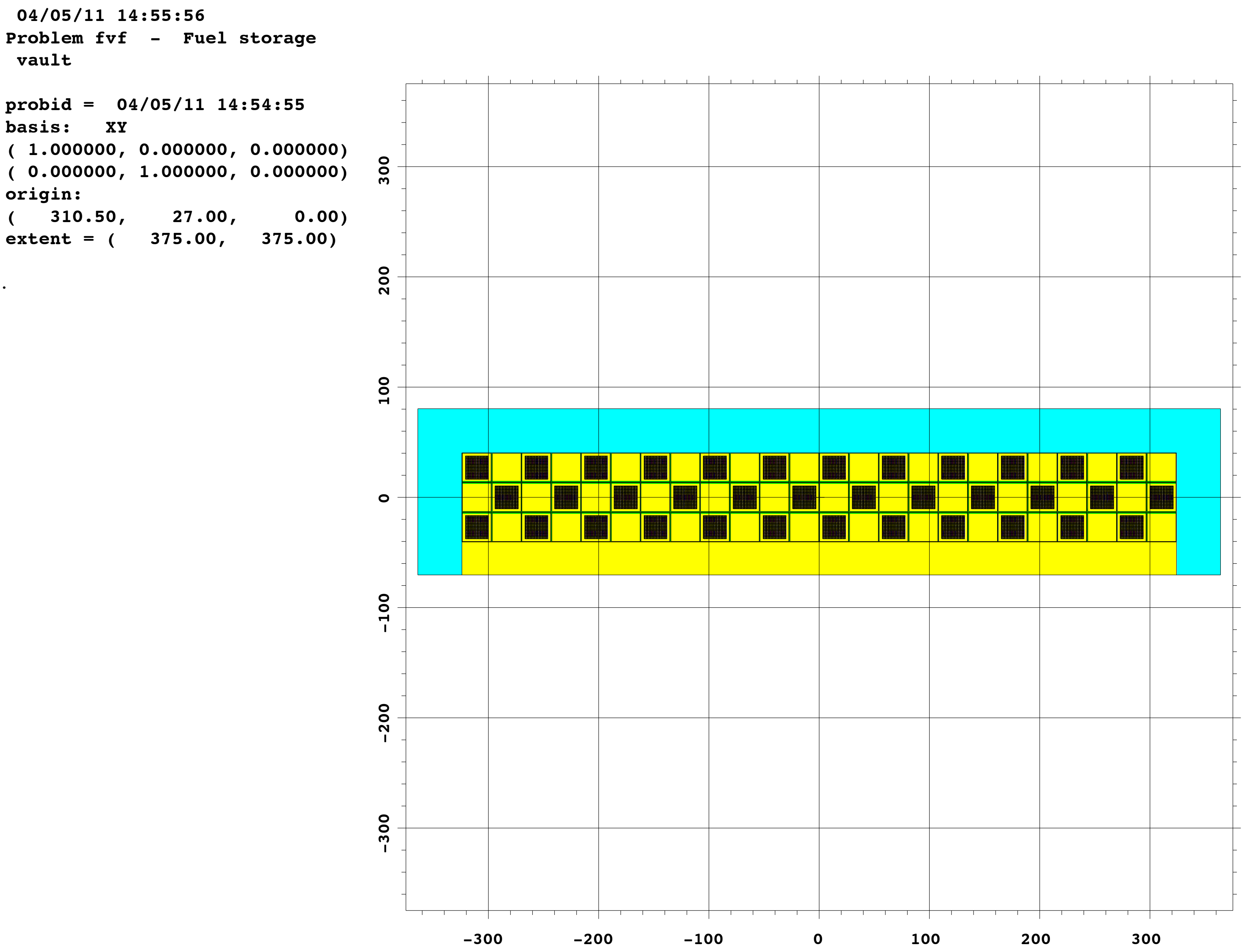

Figure 1: Fuel vault geometry xy plot large scale

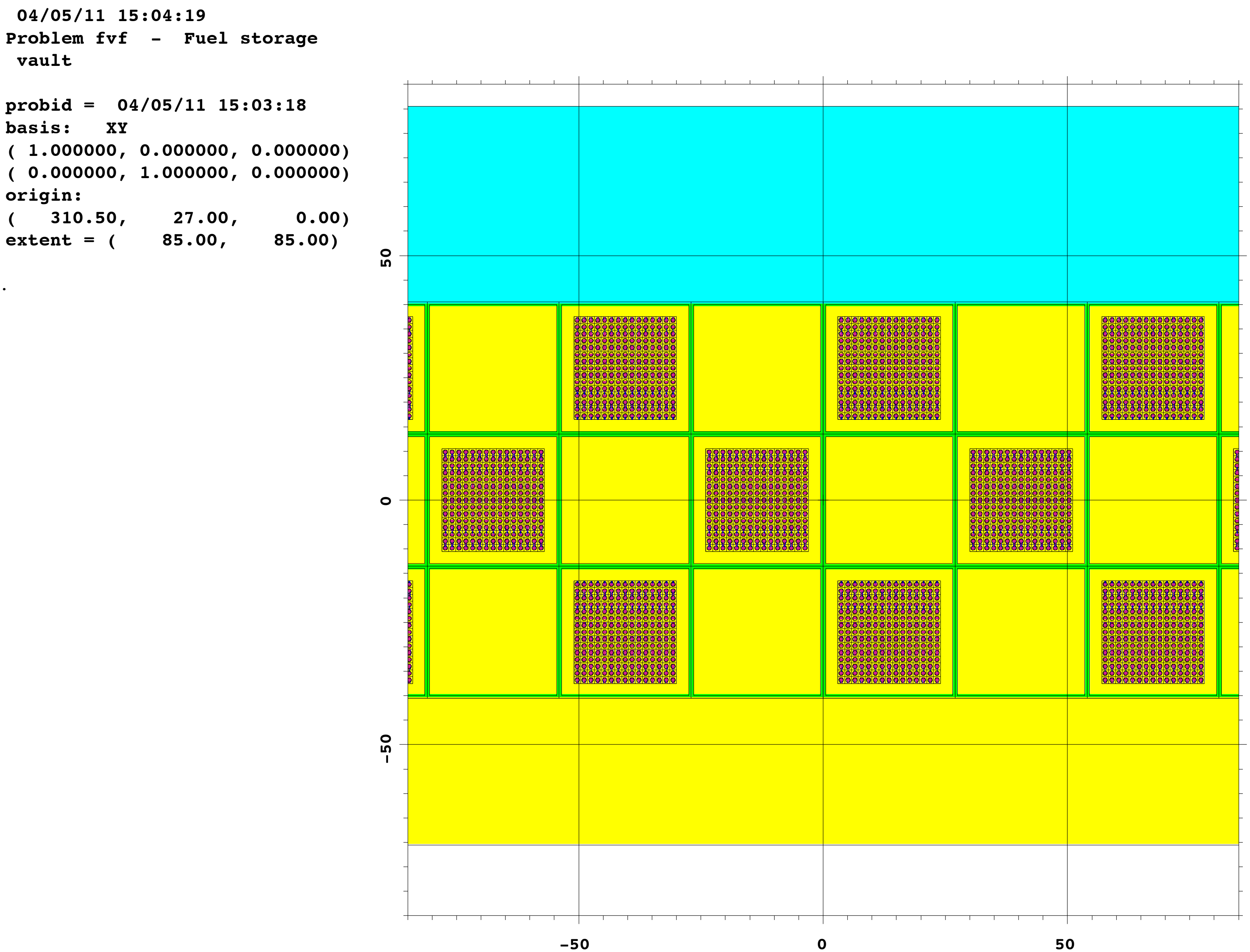

Figure 2: Fuel vault geometry xy plot medium scale

.

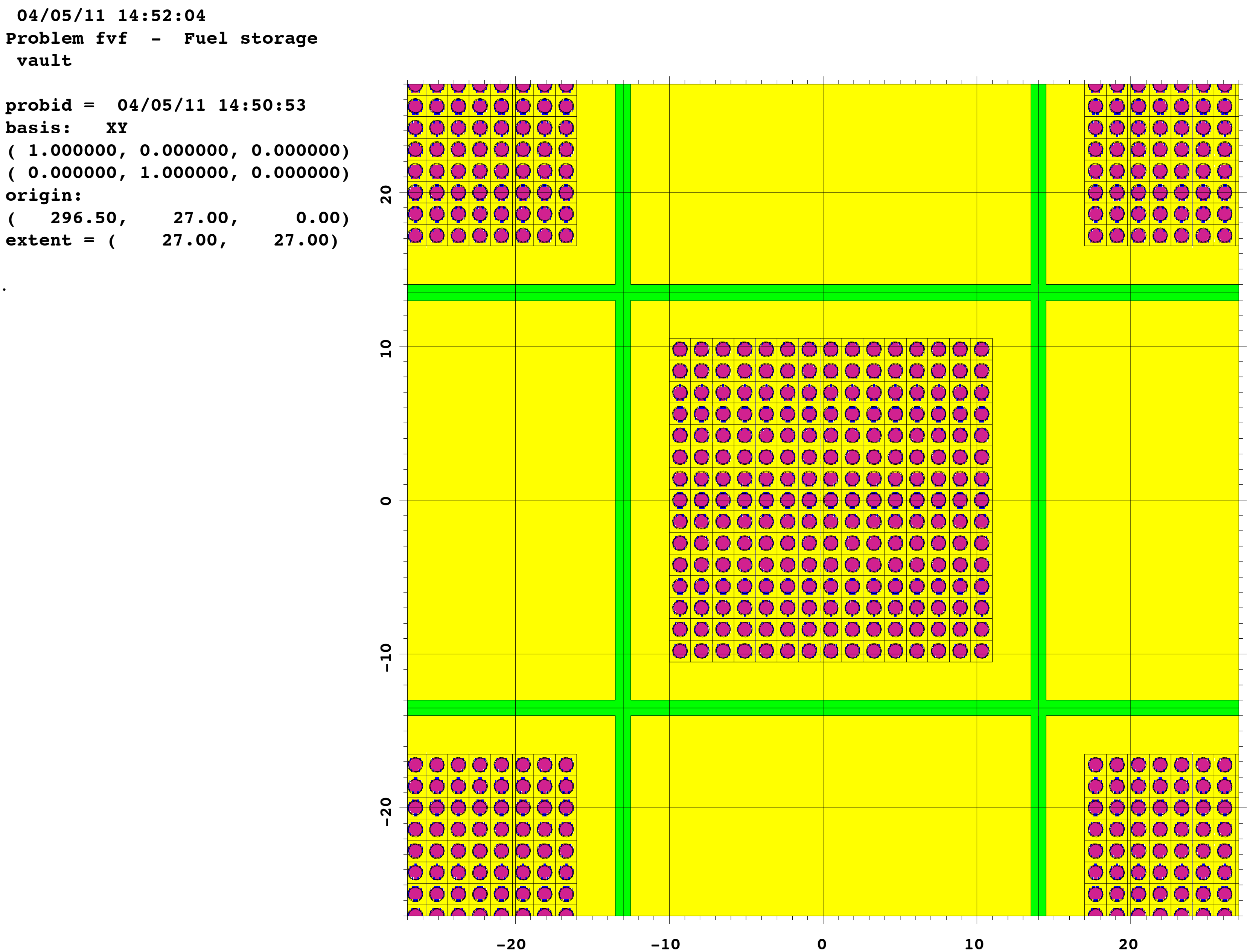

Figure 3: Fuel vault geometry xy plot small scale

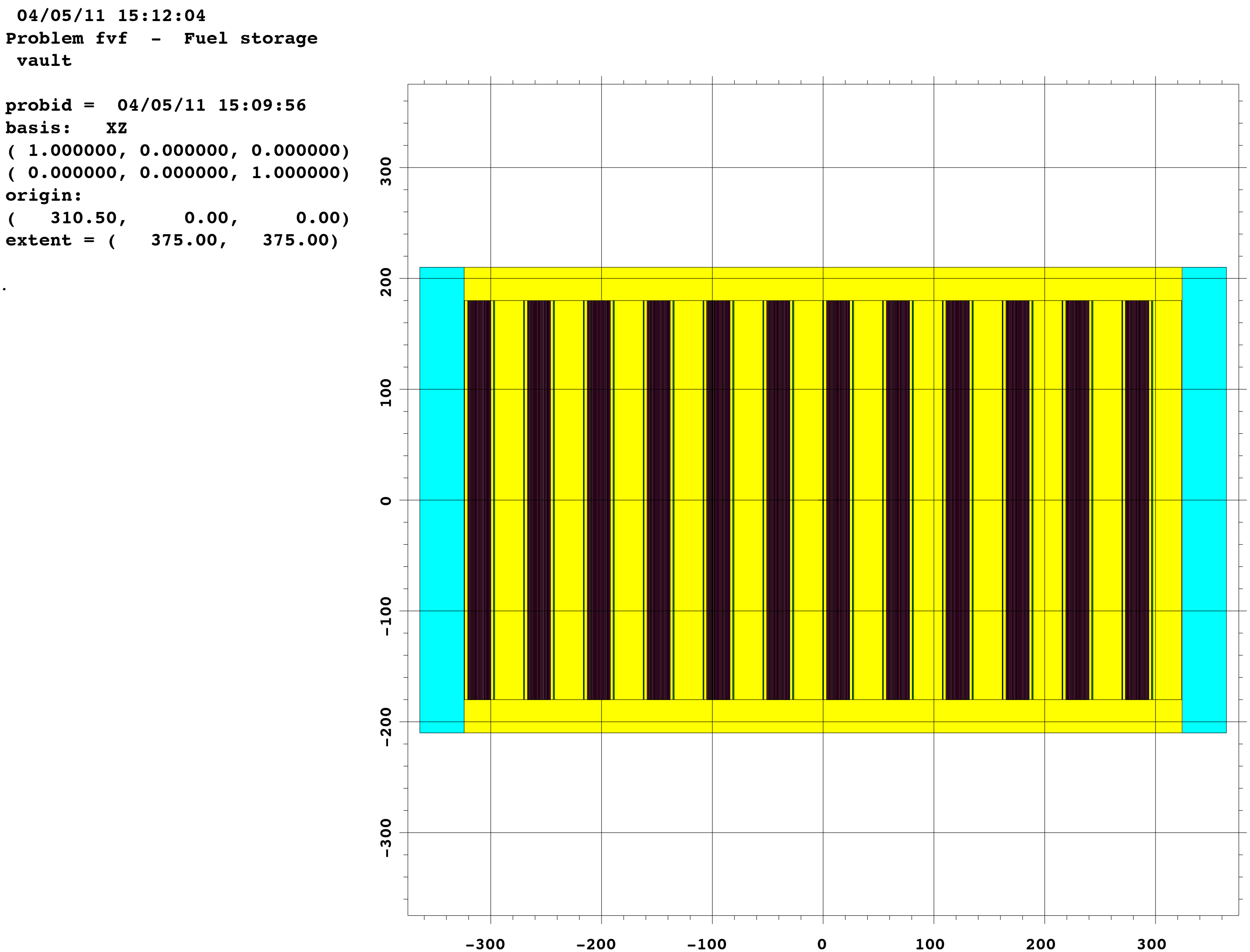

Figure 4: Fuel vault geometry xz plot

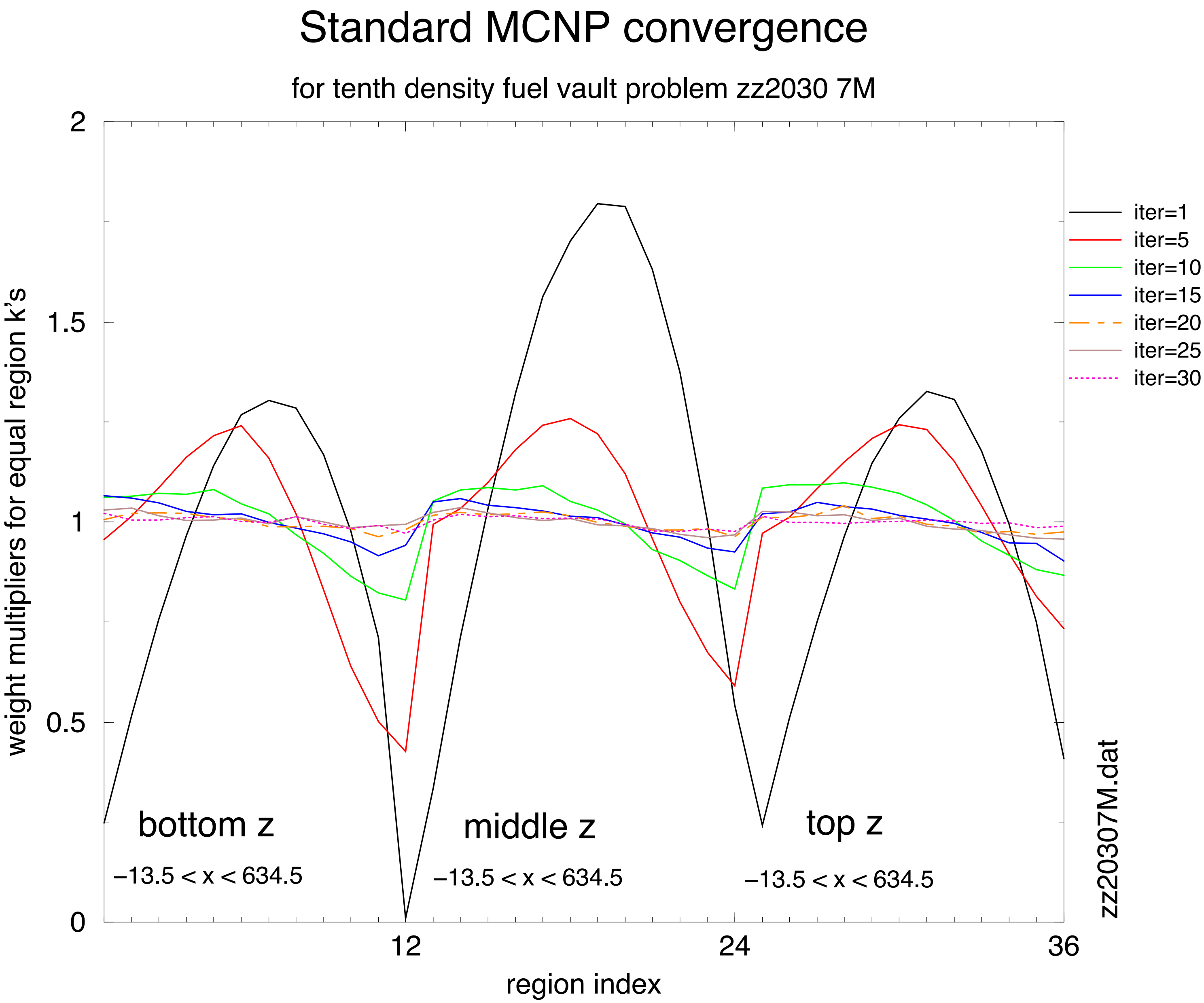

Figure 5: Convergence of Eigenvalue Estimates 1

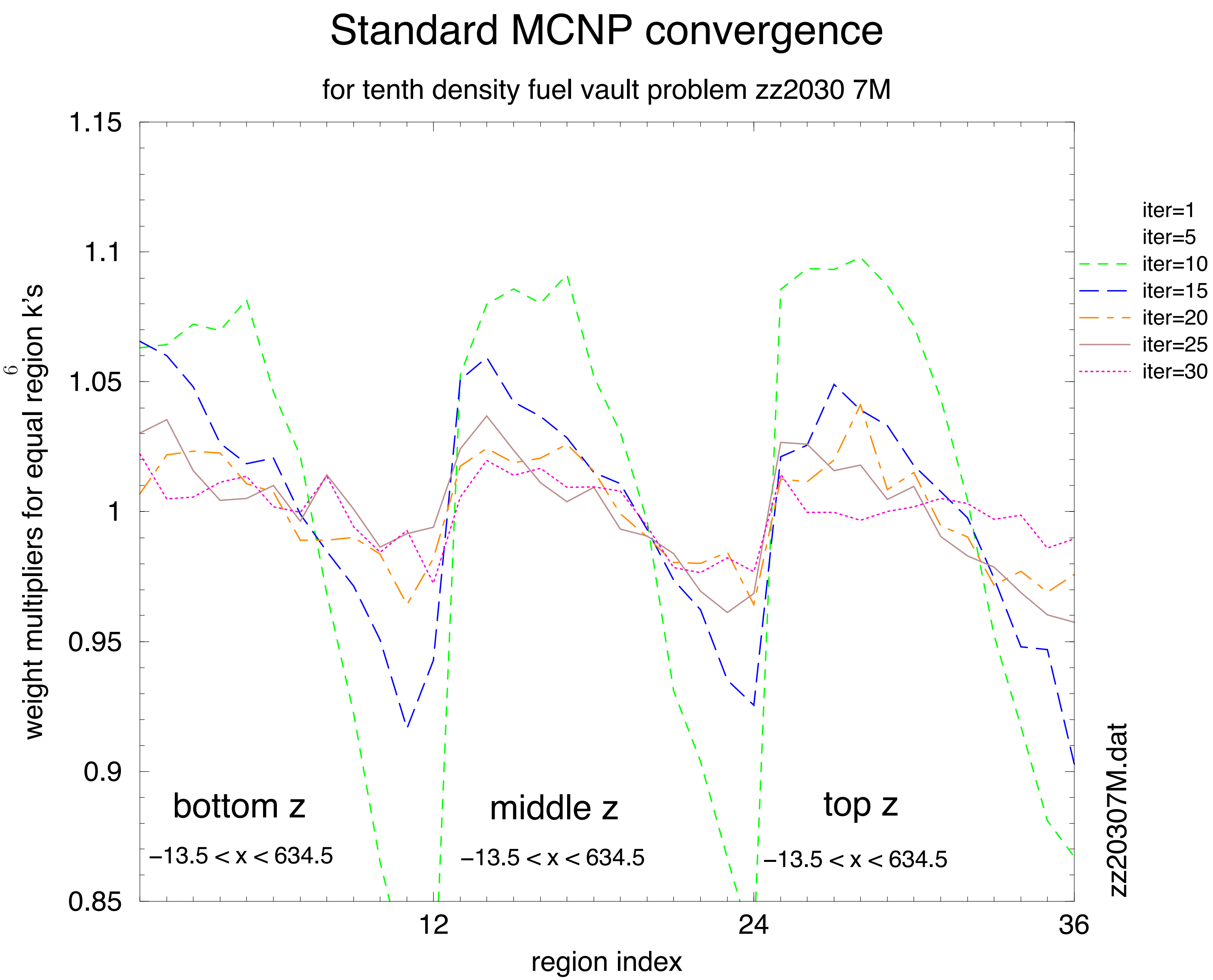

Figure 6: Convergence of Eigenvalue Estimates 2

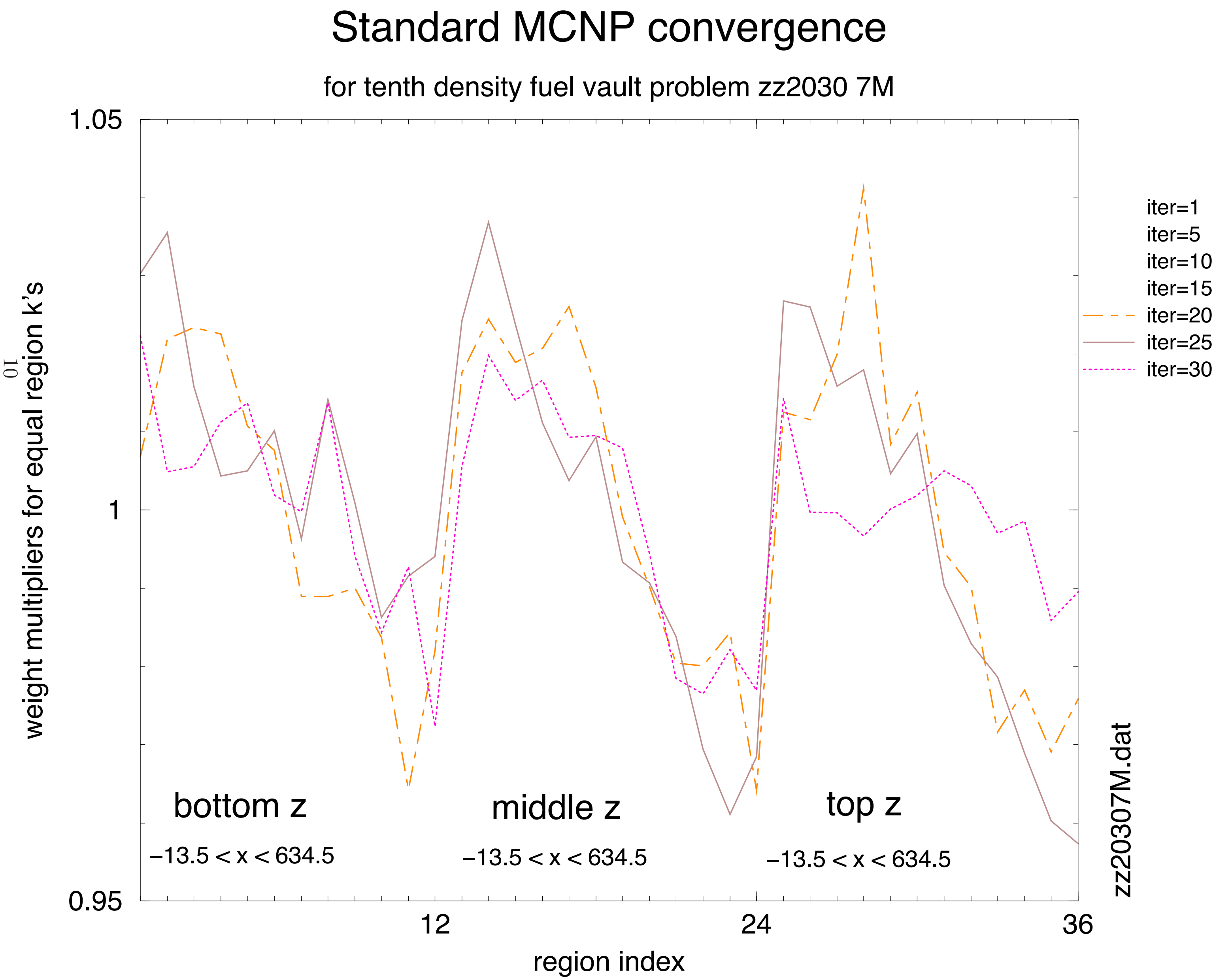

Figure 7: Convergence of Eigenvalue Estimates 3

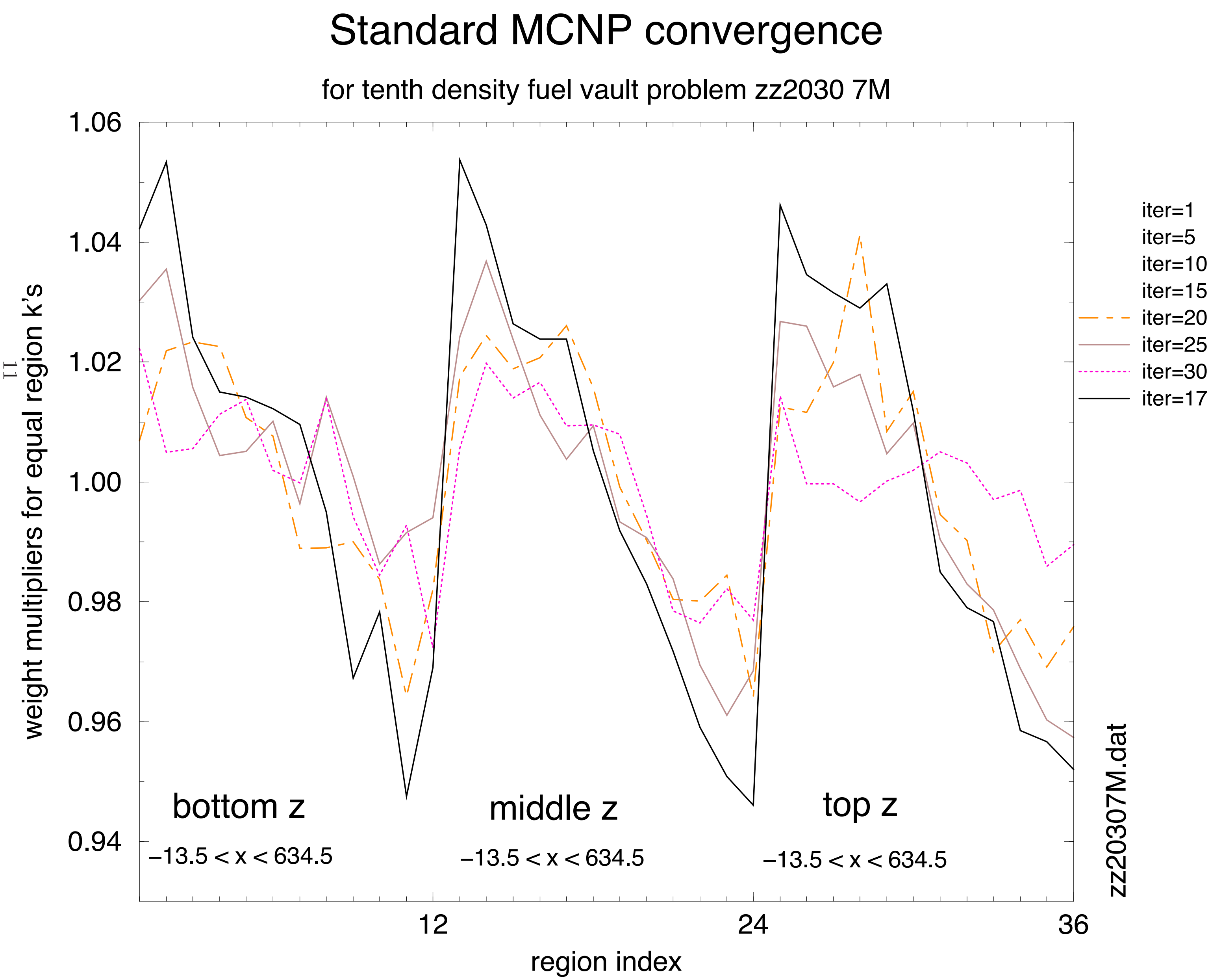

Figure 8: Convergence of Eigenvalue Estimates 4

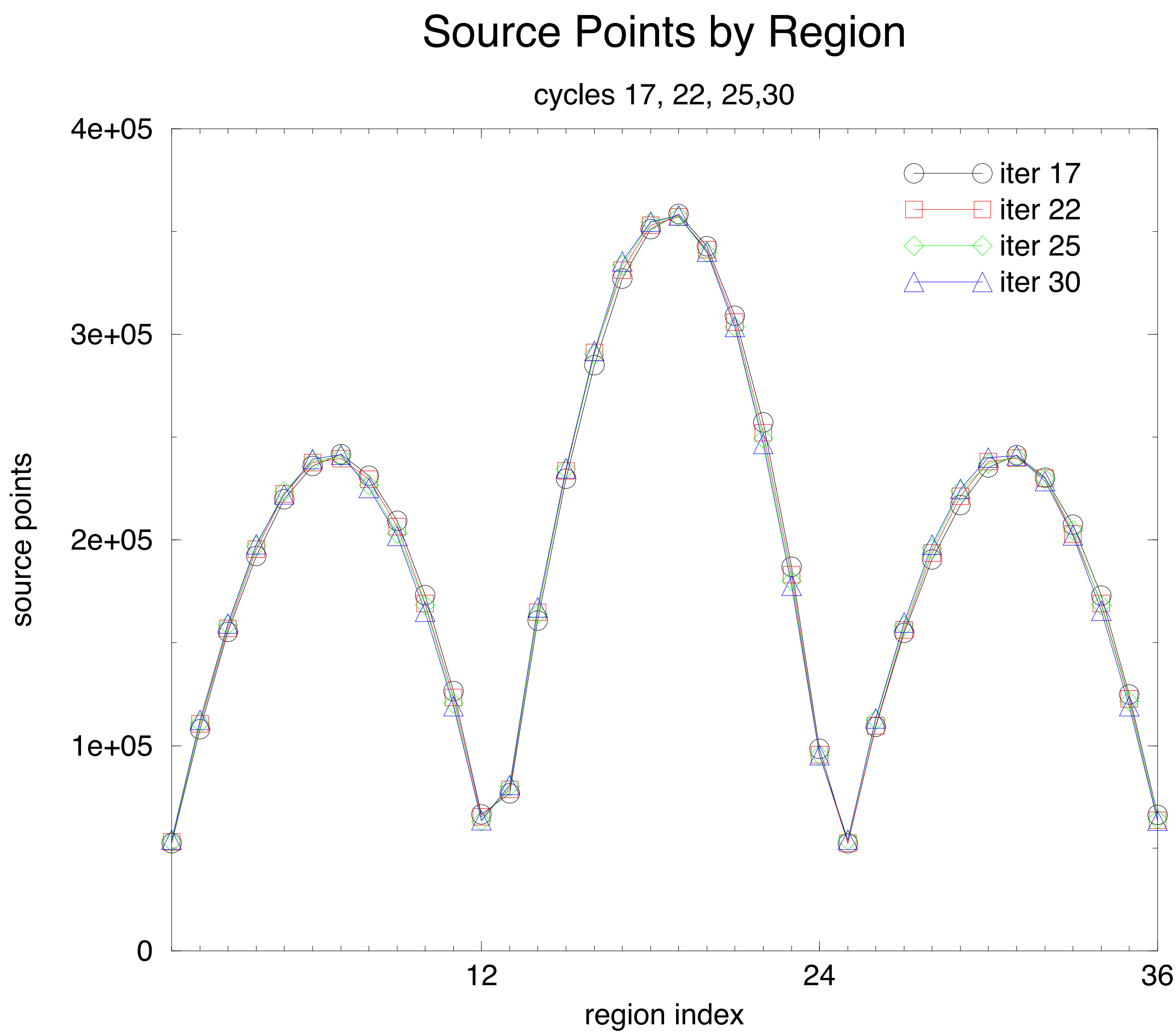

Figure 9: Source Distributions

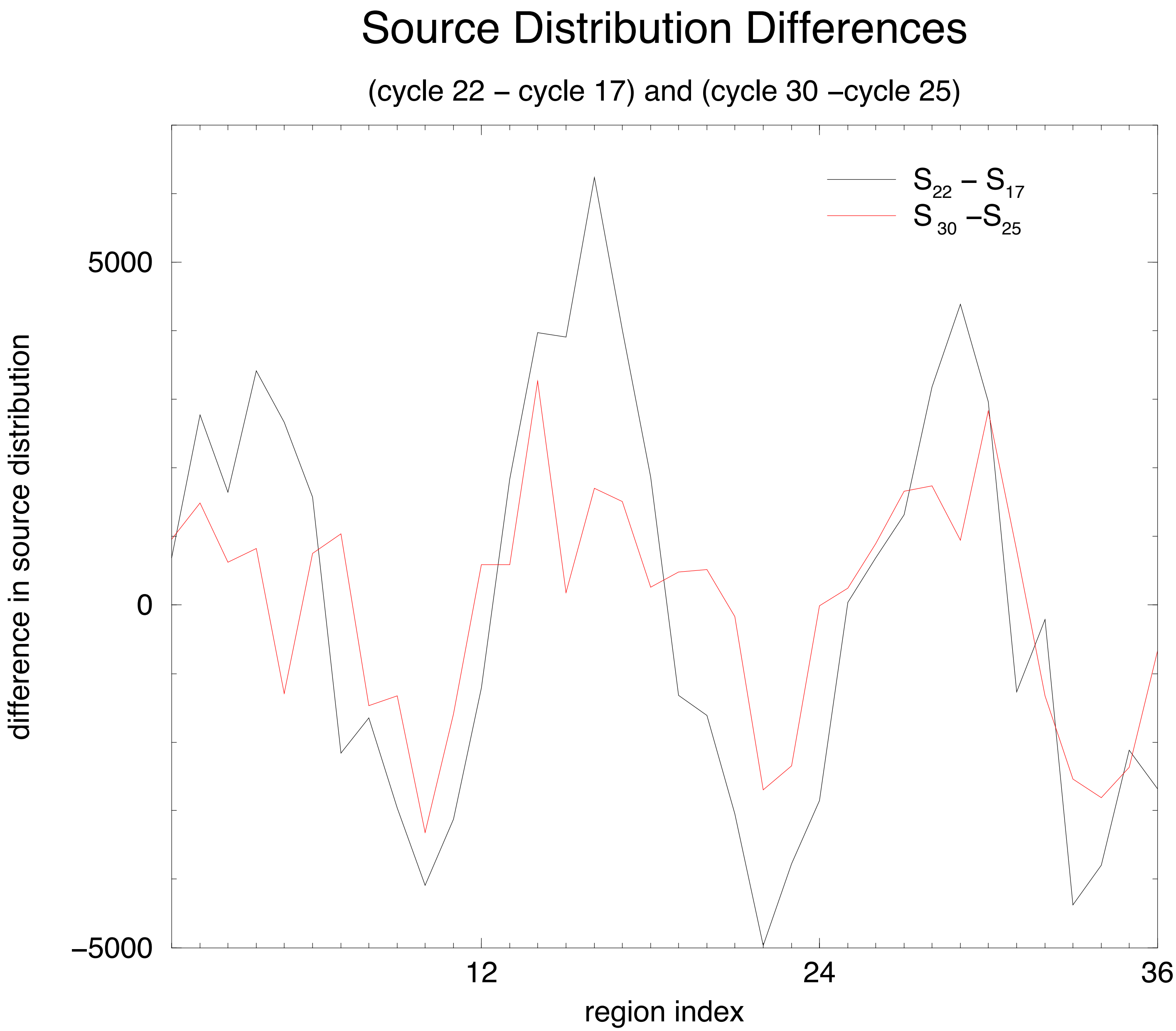

Figure 10: Difference in Source Distributions

4 Convergence Metric 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 7 million particles were used on each iteration. The initial distribution (entering iteration 1) put 3.5 million particles at the single point (625, 27.5, -179) in region 12 and the other 3.5 million particles were started uniformly over all regions.

Fig. 5 shows the change every 5 iterations 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 5 iterations on a smaller scale. (Iterations 1 and 5 are not displayed because their curves go off the graph at this scale.) Again, the weight multipliers are getting closer to 1.

5 The Multiple k Metric and MCNP's Source Entropy Check

It is worth noting that the multiple k metric is a direct measure of how far the source distribution is from convergence whereas MCNP's entropy check is a very different type of test based on the asymptotic entropy of the source distribution. Neither metric is infallible and metrics based on substantially different ideas are less likely to simultaneously indicate convergence when the source distribution has, in fact, not converged.

For example, for the test problem, MCNP printed the following at the end of the calculation, but Fig. 8 indicates that the source distribution is still converging because the weight multipliers are still trending to one.

```
keff_cycle_col=  0.338958678018840
 overall k=  0.338958678019893        2372710.74613216
   6999999.99997908

 source distribution written to file zz7M2030.s        cycle=    30
 run terminated when      30 kcode cycles were done.

 =====>       6.76 M neutrons/hr    (based on wall-clock time in mcrun)


 comment.
 comment. Average fission-source entropy for the last half of cycles:
 comment.      H=  5.03E+00  with population std.dev.=  5.29E-04
```

```
 comment.
 comment.
 comment. Cycle   17 is the first cycle having fission-source
 comment.   entropy within 1 std.dev. of the average
 comment.   entropy for the last half of cycles.
 comment.   At least this many cycles should be discarded.
 comment.
 comment. Source entropy convergence check passed.
 comment.

                              ctm =     1860.26   nrn =      111264443326
 dump   31 on file zz7M2030.r   nps =   211863087   coll =       2878427648
 tally data written to file zz7M2030.m
 mcrun  is done
```

Compare iteration 17 with iteration 30. It may be that the source distribution has not even converged after 30 iterations as the weight multipliers are not yet looking like they are unity within stochastic variation. The weight multipliers still are trending down with iteration.

Figs. 9 and 10 directly show the convergence of the source distribution (and Appendix B has the same information in array form). On the scale of Fig. 9 it is difficult to tell that the source is still converging at cycle 17. Comparing in Fig. 10 the 5 cycle source difference (cycle 22 - cycle 17) with the source difference (cycle 30 - cycle 25) one sees better that the source distribution is still converging at cycle 17. Thus, the new metric is giving somewhat different results than the entropy metric.

6 Summary and Future Work

A multiple eigenvalue based convergence metric might be a useful addition to MCNP to assess convergence. The method explicitly and directly uses the basic definition of the eigenfunction and eigenvalue. Inasmuch as MCNP's entropy measure does not make such a direct estimate of the distance from convergence, it is reasonable to expect that the new method will provide further convergence assurance in addition to MCNP's entropy method.

It is worth noting that *both* the entropy and multiple k convergence measures provide useful information and the multiple k metric is not meant to supplant the entropy metric.

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

7 Appendix A: 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
       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
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
```

8 Appendix B: Data for Cycles 17, 22, 25, and 30

```
 a[i] cycle 17, b[i] cycle 22, c[i] cycle 25, d[i] cycle 30

        a[ 1]=  52418.6341420262
        a[ 2]=  107858.751167272
        a[ 3]=  155273.865571055
        a[ 4]=  192242.597316689
        a[ 5]=  219897.253605233
        a[ 6]=  236135.976714007
        a[ 7]=  241700.657516146
        a[ 8]=  231404.050943086
        a[ 9]=  209428.903788324
        a[10]=  173247.994248485
        a[11]=  126553.802066245
        a[12]=  66576.4681801899
        a[13]=  76799.1852182547
        a[14]=  160904.447850486
        a[15]=  229872.340849693
        a[16]=  285120.744486318
        a[17]=  327370.580981526
        a[18]=  351260.865007342
        a[19]=  358660.802101521
        a[20]=  342935.311709371
        a[21]=  309211.728465707
        a[22]=  257177.519607150
        a[23]=  186944.517945407
        a[24]=  98530.6966088038
```

```
a[25]=  52536.4579953659
a[26]=  109064.947903156
a[27]=  154977.308923247
a[28]=  190615.030529054
a[29]=  216957.648315174
a[30]=  235324.190334653
a[31]=  241203.400914771
a[32]=  230316.676567789
a[33]=  207454.854991298
a[34]=  172913.494325873
a[35]=  124770.468150443
a[36]=  66337.8249518155

b[ 1]=  53110.3017566322
b[ 2]=  110633.622785026
b[ 3]=  156918.801560646
b[ 4]=  195662.529583831
b[ 5]=  222564.839312269
b[ 6]=  237711.770115115
b[ 7]=  239537.605565260
b[ 8]=  229758.130756949
b[ 9]=  206465.472702004
b[10]=  169160.402967755
b[11]=  123427.476887193
b[12]=  65366.9100303654
b[13]=  78640.9838673632
b[14]=  164876.442749935
b[15]=  233785.973782858
b[16]=  291363.319531511
b[17]=  331390.634887593
b[18]=  353129.582008798
b[19]=  357340.508808612
b[20]=  341323.179736381
b[21]=  306162.091026604
b[22]=  252211.404733718
b[23]=  183168.812815935
b[24]=  95676.7789605512
b[25]=  52577.0577594156
b[26]=  109755.220853395
b[27]=  156295.516363144
b[28]=  193796.675822450
b[29]=  221349.283108473
b[30]=  238289.034254695
b[31]=  239934.787304278
b[32]=  230104.289149141
b[33]=  203081.924486996
```

```
b[34]=  169115.382367614
b[35]=  122654.123022556
b[36]=  63659.1285983792

c[ 1]=  53203.2293913017
c[ 2]=  111092.701705472
c[ 3]=  158801.438966393
c[ 4]=  197085.609886908
c[ 5]=  223619.620094814
c[ 6]=  238564.718812300
c[ 7]=  240486.689355046
c[ 8]=  226902.694320515
c[ 9]=  203253.142248675
c[10]=  168335.174034866
c[11]=  120702.554419202
c[12]=  63179.6486598959
c[13]=  80155.8866482119
c[14]=  164055.559585073
c[15]=  234645.661629515
c[16]=  290342.745451413
c[17]=  334000.221199581
c[18]=  354634.112352684
c[19]=  357544.610735895
c[20]=  339919.369666469
c[21]=  303793.534607134
c[22]=  249411.483261461
c[23]=  180047.275924847
c[24]=  95254.2222041767
c[25]=  53849.2278280292
c[26]=  112388.704770774
c[27]=  158269.616997556
c[28]=  195941.842112121
c[29]=  223982.180457750
c[30]=  237250.687883648
c[31]=  240261.341063166
c[32]=  229996.475981038
c[33]=  204716.403823949
c[34]=  168477.393845742
c[35]=  121638.000217496
c[36]=  64196.2198432734

d[ 1]=  54162.8761630490
d[ 2]=  112584.776747226
d[ 3]=  159425.049551749
d[ 4]=  197910.197940256
d[ 5]=  222333.868845722
```

```
d[ 6]=  239318.990175641
d[ 7]=  241528.965062974
d[ 8]=  225432.431555234
d[ 9]=  201928.334099043
d[10]=  165016.455937883
d[11]=  119112.748700814
d[12]=  63765.4219536047
d[13]=  80745.5456019819
d[14]=  167323.385851933
d[15]=  234826.074246848
d[16]=  292048.531201556
d[17]=  335514.369262781
d[18]=  354892.380634094
d[19]=  358024.927579867
d[20]=  340436.086289260
d[21]=  303624.161762818
d[22]=  246715.559224156
d[23]=  177701.571856085
d[24]=  95242.8209372811
d[25]=  54098.9058361420
d[26]=  113288.450343203
d[27]=  159931.814485208
d[28]=  197682.303650653
d[29]=  224926.666158133
d[30]=  240085.634562156
d[31]=  241058.183438400
d[32]=  228672.928427570
d[33]=  202175.219579446
d[34]=  165670.152715954
d[35]=  119266.677299934
d[36]=  63527.5323004193
```

```
b[i]-a[i]   cycle 22 -cycle 17
1 691.668
2 2774.87
3 1644.94
4 3419.93
5 2667.59
6 1575.79
7 -2163.05
8 -1645.92
9 -2963.43
10 -4087.59
11 -3126.33
12 -1209.56
```

```
13 1841.8
14 3971.99
15 3913.63
16 6242.58
17 4020.05
18 1868.72
19 -1320.29
20 -1612.13
21 -3049.64
22 -4966.11
23 -3775.71
24 -2853.92
25 40.5998
26 690.273
27 1318.21
28 3181.65
29 4391.63
30 2964.84
31 -1268.61
32 -212.387
33 -4372.93
34 -3798.11
35 -2116.35
36 -2678.7

d[i]-c[i]   cycle 30 -cycle 25
1 959.647
2 1492.08
3 623.611
4 824.588
5 -1285.75
6 754.271
7 1042.28
8 -1470.26
9 -1324.81
10 -3318.72
11 -1589.81
12 585.773
13 589.659
14 3267.83
15 180.413
16 1705.79
17 1514.15
18 258.268
19 480.317
20 516.717
```

```
21 -169.373
22 -2695.92
23 -2345.7
24 -11.4013
25 249.678
26 899.746
27 1662.2
28 1740.46
29 944.486
30 2834.95
31 796.842
32 -1323.55
33 -2541.18
34 -2807.24
35 -2371.32
36 -668.688
```