COMPRESSION OF PINWISE NUCLIDE CONCENTRATIONS IN CMS5

Joshua Hykes
Studsvik Scandpower, Inc.
101 N. 3rd Street, Suite 202, Wilmington NC USA

joshua.hykes@studsvik.com

ABSTRACT

Pinwise nuclide number density (ND) data from the lattice physics code CASMOS5 is compressed using a preconditioned truncated singular value decomposition (TSVD) to reduce storage requirements. Previously only assembly-average or single-pin NDs were optionally saved in the CMS5 few-group cross section library. However, backend analysis has prompted the desire to have pin-by-pin NDs available in the library for use by the SNF code. Adding this data set significantly increases the size of the library, particularly for lattices modeled in full assembly geometry (that is, not in half or octant symmetry). To reduce the required storage, the SVD is used to approximate the entire data with a reduced basis. In the four test cases, compression ratios of 2.7 to 8.5 were achieved for the PWR cases, with maximum errors less than 0.1%. However, the rodded BWR segment proved more difficult, with an average compression ratio of 1.6. One advantage of this technique is that the compression ratio is higher for full-symmetry cases, where the need for the compression is also highest.

KEYWORDS: isotopes, compression, SVD

1. INTRODUCTION

The level of nuclide detail in nodal core simulators continues to increase as users perform increasingly sophisticated analysis. In the previous generation code, Studsvik’s SIMULATE-3 tracked a handful of nuclides in each node. Studsvik’s most up-to-date simulator SIMULATE5 [1] tracks 50+ nuclides per node and a handful of actinides on a finer mesh. However, users continue to push for more detailed information, in this case pin-by-pin isotopic compositions from SIMULATE5. The primary uses for such data are backend analysis and as an input to a Monte Carlo simulation of one or more burnup fuel assemblies for reference calculations. (The inclusion of this data set does not imply that the nodal code would itself track or deplete the pinwise nuclides, but rather that the pinwise data would be available for post analysis. Although there may be ways to use the TSVD for neutronic calculations, that is not explored here.) Currently, SIMULATE5 and the spent nuclear fuel backend companion code SNF are limited to extracting pin isotopic number densities for only one pin per assembly. If multiple pins are required, then multiple SIMULATE5 libraries must be created. The downside to storing pin-by-pin nuclide data in the library is the additional storage space required. In preliminary tests using uncompressed data, most libraries doubled or tripled in size. Compressing the pinwise nuclide number densities would reduce the storage requirements and make it more practical to store all the pinwise isotopic data in the library file. To estimate the
potential increase in the library, we consider the following estimates of the various parameters involved:

- Number of fuel pins per lattice. Typically only an eighth or half of the pins need to be stored because of symmetry. A 17x17 octant has about 40 fuel pins.
- Number of nuclides tracked. CASMO5 [2] tracks several hundred nuclides. The precise number varies for several reasons, but 350 is a reasonable estimate.
- Number of depletion statepoints per case matrix, estimated as 300.
- Segments per library, estimated as 30.
- Bytes per single-precision float, 4.

Multiplying these factors gives 0.5 GB. Some libraries could be significantly larger, for example, if the lattice physics calculation was performed in full symmetry (in which case this estimate would increase to 4 GB), or if the number of segments in the core was larger. Compressing the nuclide NDs would make adding this data set to the library more palatable. Intuitively, one would expect that some compression is possible, since many fuel pins begin with the same composition, and then deplete in a similar environment.

Recently, Tomatis studied compression algorithms for pin-by-pin power form factors [3]. Performing the compression on each 2D map was not very successful, but when compressing the 2D maps (over all statepoints) simultaneously, a compression ratio greater than 10 was achieved. Tomatis compared the Hotelling Transform and the Discrete Cosine Transform, finding that the HT produced smaller errors for the same compression ratio. The present work is similar in purpose, but the data to compress is different (nuclide number densities versus power form factors). Although their goal was not data compression, Abdo et al. used the SVD to develop a surrogate model for estimating the isotopics in a single fuel region throughout a depletion calculation [4].

2. METHODS

Here we use the truncated SVD to capture the most important subspace of the data, a similar method as the Hotelling Transform used in Ref. [3]. (The SVD and related techniques have been developed and studied in many fields, so the idea goes by a multitude of names, one of the more popular being principal component analysis (PCA).) The SVD for matrix $A$ is $A = U \Sigma V^*$ [5]. Assuming that there are $n$ nuclides and $p$ pins, the $A$ matrix has size $n \times p$. There is a separate $A$ matrix for each statepoint. (These separate $A$ matrices could alternately be combined; see below.) We compute the SVD [6] with the library LAPACK [7], whether through a Python interface for initial testing, or linking directly with the library from CMSLINK5, the utility code responsible for processing various CASMO5 outputs into the library read by SIMULATE5. The truncated SVD approximates $A$ by keeping only the $r$ largest singular values and corresponding singular vectors. The truncated SVD is $A_r = U_r \Sigma_r V_r^*$, where $U_r$ is $n \times r$, $\Sigma_r$ is $r \times r$, and $V_r^*$ is $r \times p$.

For numerical accuracy in the calculation of the SVD, it is helpful if the elements in the $A$ matrix are close to 1.0. This will not be the case for the nuclide NDs, since they vary across several orders of magnitude. However, this can be accomplished by normalizing the rows of $A$:

$$\hat{A}_{ij} = \frac{A_{ij}}{\mu_i}, \quad \mu_i = \frac{1}{p} \sum_j A_{ij} . \tag{1}$$
This normalizes the pin map for each nuclide, which is also how CASMO prints the maps. The normalization increases the storage requirements slightly because \( \bar{\mu} \) must be stored.

One could likewise normalize each row of \( A \) with the mean and standard deviation:

\[
\hat{A}_{ij} = \frac{A_{ij} - \mu_i}{\sigma_i}, \quad \sigma_i^2 = \frac{1}{p} \sum_j (A_{ij} - \mu_i)^2.
\]

This scheme gave a similar compression performance as observed for the first normalization, so the first normalization is used.

If one stores the truncated SVD instead of \( A \), then the compressed storage size is:

\[
N_{\text{compressed}} = n + r + r \cdot n + r \cdot p = n + r(1+n+p)
\]

and the compression ratio \( \text{CR} \) is

\[
\text{CR} = \frac{n \cdot p}{n + r(1+n+p)}.
\]

A higher \( \text{CR} \) is favorable as it indicates better compression.

As suggested in Ref [3], one could form a combined \( A \) for NDs at all statepoints, and then perform the SVD. In a few tests of this procedure versus doing the compression individually at each statepoint, the individual statepoint method achieved a better or comparable compression ratio. It is also simpler for the database program to process one statepoint at a time. One difficulty of using the combined \( A \) is that some isotopes will change over many orders of magnitude. This makes the normalization of \( A \) less useful because the elements of \( A \) will still span several orders of magnitude (unless an additional transformation is introduced, such as a logarithm, which is not investigated here).

The compression error is computed as

\[
E_{r,ij} = |\hat{A}_{r,ij} - A_{ij}|
\]

Assuming that \( A \) has already been normalized, this error is effectively a relative error. In what follows, we use the median of \( E \) as a representative, average value of the error, and the maximum as the worst case. The smallest rank \( r \) of the truncated SVD is chosen such that \( \|E_r\|_{\infty} \leq \epsilon \), where \( \epsilon \) is a user-defined error tolerance.

### 3. RESULTS

Four example cases are considered here to test a variety of factors which might affect the effectiveness of the compression method. Three PWR and one BWR lattices are included. The BWR bundle is expected to be the most challenging because of the cruciform control blade, which induces a flux and burnup tilt across the bundle, thus leading to a larger variety of isotopic compositions. The example cases include gadolinia and IFBA burnable absorbers. The gadolinia especially creates pins with different powers and burnups. Although CASMO5 tracks multiple regions within each Gd-bearing fuel pin, the compression is done with pin-averaged compositions.

The test cases span a variety of symmetries (octant, half, and full). This affects the number of fuel pins and has a strong effect on the \( \text{CR} \). The four test cases are:
Figure 1: Compression ratios for various TSVD ranks $r$.

- $17 \times 17$ PWR with IFBA in octant symmetry with 39 fuel pins (denoted as P-17-IFBA-1/8),
- $17 \times 17$ PWR with Gd in octant symmetry with 39 fuel pins (P-17-Gd-1/8),
- $14 \times 14$ PWR with Gd in full symmetry with 179 fuel pins (P-14-Gd-1),
- $10 \times 10$ BWR with Gd in half symmetry with 49 fuel pins (B-10-Gd-1/2).

For each case, the depletions from the standard ‘S5C’ CASMO5 case matrix are executed, which include the base depletion, moderator and fuel temperature histories, boron history, and rodded depletion. Approximately 320 isotopes are tracked for the IFBA case, and 360 for the Gd cases.

Figure 1 provides the compression ratios achieved while varying the TSVD rank $r$. The few points on the lower left are from the initial statepoints with fresh fuel, where there are only a few isotopes and thus not much to compress. The points along the main, longer curves are from burned fuel. The lattice in full geometry (P-14-Gd-1) has the largest CRs, while the other cases are more closely grouped.

Figure 2 shows the compression error for each statepoint, with the maximum and median error over all fuel pins. Each statepoint is plotted as a light dot, so darker areas indicate that many statepoints overlap. The maximum error decreases relatively quickly for the PWR cases, while the decrease is slower for the BWR case. This is expected given the flux and power gradient across the BWR bundle due to the inserted control blade. See the further discussion in the next section.

For the PWR cases, an error less than 1% is achieved with $r = 10$ to 15. The maximum error in the BWR case does not reach this level until $r = 25$. If the maximum error was not as important as the median, then $r = 15$ might be satisfactory for the BWR case as well.
Figure 2: Compression error as the compression level is varied. The 1\% and 0.1\% levels are indicated by gray lines.
Table 1: Compression performance for the case matrix calculation for four lattices using a maximum error tolerance of $10^{-3}$.

<table>
<thead>
<tr>
<th>Sym.</th>
<th>TSVD rank</th>
<th>Compr. Ratio</th>
<th>Median</th>
<th>Abs. error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TSVD rank</td>
<td>Compr. Ratio</td>
<td>Median</td>
<td>Abs. error</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17x17 PWR with IFBA</td>
<td>1/8</td>
<td>2</td>
<td>11.5</td>
<td>18</td>
</tr>
<tr>
<td>17x17 PWR with Gd</td>
<td>1/8</td>
<td>2</td>
<td>12.0</td>
<td>15</td>
</tr>
<tr>
<td>14x14 PWR with Gd</td>
<td>1</td>
<td>2</td>
<td>15.1</td>
<td>50</td>
</tr>
<tr>
<td>11x11 BWR with Gd</td>
<td>1/2</td>
<td>3</td>
<td>27.0</td>
<td>38</td>
</tr>
</tbody>
</table>

While the previous results were for varying compression levels, Table 1 shows the CRs when a maximum error tolerance of 0.1% is specified. A sufficiently large $r$ is chosen independently for each statepoint to ensure that the maximum error meets the criterion. The octant PWR cases achieve CRs around 3 on average, while the full PWR case has an average CR of 8.5. The BWR is the least effective with an average CR of 1.6. Judging from the slope of the max curve in Figure 2, choosing a tolerance of 0.5% would have allowed significant reduction in $r$ and increase of CR.

SVD runtime costs  One downside to the compression described here is the computational cost of the SVD. In the context of CMS5, the SVDs would be performed in the library utility program CMSLINK5. Since this program is typically run once, the runtime is of secondary importance. Nonetheless, for a set of 24 cases, CMSLINK5 runtime increased by an average of 70% when enabling the pin-by-pin ND storage with no compression. When enabling the SVD compression, the runtime increased by a further 70%. This was using the reference BLAS that comes with LAPACK. Presumably some gains would be possible with an optimized BLAS. When timing the individual SVD factorizations, the time ranged from 3 to 26 ms for 39 to 179 fuel pins and 359 nuclides.

3.1. Discussion of SVD basis vectors

The previous section presented the CRs and corresponding errors using the SVD compression method. These are the primary parameters that would interest users of the algorithm. Nonetheless, it is interesting to examine the SV decomposition and particularly the SVD basis vectors. The three factors in the decomposition are presented, along with observations about their primary features. Except for the singular value comparison in Figure 3, the following basis vectors come solely from a BWR case with an inserted control blade, selecting the burnup point at 7 MWd/kgU when the burnable absorber Gd is still burning out.

First, the singular values (the diagonal of $\Sigma$) are plotted in Figure 3, for a $17 \times 17$ PWR and a $10 \times 10$ rodded BWR. For the PWR assembly, the singular values rapidly decay by a factor of 1000, implying that a small TSVD rank would be appropriate (maybe $r = 4$ to 8). However, the SVs for the rodded BWR bundle decrease gradually, meaning that a higher rank is necessary to achieve acceptable accuracy. This behavior is consistent with expectations and the above results.
Figure 3: Singular values for the PWR and rodded BWR cases at 7 MWd/kgU.

Figure 4: The first two vectors from U for the isotopic basis for the rodded BWR case at 7 MWd/kgU.

Figure 4 shows the first two isotopic basis vectors (the first two columns of $U$). The first vector is bimodal, with the values close to zero corresponding to the burnable absorber (BA) gadolinium isotopes (which CASMO5 tracks separately from the gadolinium fission products). Thus, the first vector has split the isotopes into two groups, BA Gd isotopes and the rest. The second basis vector has more features (with the BA Gd isotopes still clearly visible). One feature is the many peaks with magnitude of about 0.08. These peaks correspond to trace nuclides which appear equally in all pins due to numerical precision issues.

Figure 5 shows the first three pin-by-pin basis vectors (from $V$). For reference, the control blade and gadolinium pins are indicated in the figures. The $V$ basis vectors have done a good job of grouping the BA pins. In addition, the pins with similar power levels (for example, the pins next to the control blade) are grouped.
4. CONCLUSIONS

The truncated SVD has been examined here for compression of pin-by-pin nuclide number densities. This pin-by-pin data will be saved into the few-group cross section library read by SIMULATE5, whereas previously only the isotopics for the segment average or a single pin were saved. The new pin-by-pin data will be used for backend analysis (using the SNF code) and for generating equivalent inputs for Monte Carlo reference calculations. Without compression, the pin-by-pin NDs can increase cross section library file by tens of percent (for octant PWR) to more than doubling the size (for full assemblies). With the truncated SVD compression, we have achieved compression ratios of 1.5 to 8, while ensuring that the maximum compression error is less than 0.1%. If this error tolerance is relaxed, then better compression can be attained. The full PWR lattices were the most amenable to compression. This is fortuitous, as this is where the compression is most needed. On the other hand, BWR lattices with control blades inserted proved a particular challenge for this method, caused by the flux gradient across the bundle.

REFERENCES