TXT2NTAB
Generating NEMTAB cross-section libraries from SCALE

C. Mesado, A. Labarile, R. Miró, A. Bernal

University Research Institute for Industrial, Radiophysical and Environmental Safety (ISIRYM)

Universitat Politècnica València (UPV)

SCALE USERS GROUP WORKSHOP

September 26-28 - ORNL
INDEX

1. INTRODUCTION

2. GUIDE

3. EXAMPLE CASE

4. CONCLUSIONS

5. ONGOING WORK
1. INTRODUCTION

**TXT2NTAB** is a set of **MATLAB** scripts developed to **convert** cross sections predicted by **SCALE** (**txtfile16** format) into **NEMTAB** formatted cross section libraries (to use with **PARCS** for example).

The methodology has been **tested** with a **BWR** GE6 within the author’s **PhD** thesis and was also **verified** for a fresh fuel simulation with a **code-to-code** comparison with **CASMO**.
1. INTRODUCTION

TXT2NTAB is a set of MATLAB scripts developed to convert cross sections predicted by SCALE (txtfile16 format) into NEMTAB formatted cross section libraries (to use with PARCS for example).

Each reactor segment must be run in SCALE to provide all txtfile16 needed for a complete core representation.
1. INTRODUCTION

- TXT2NTAB was requested by the **UAM-LWR benchmark** organizers.

- They use it to generate NEMTAB to **distribute** among benchmark participants.

- Developed within the author’s **PhD thesis** (UPV).

- It handles **TRITON-NEWT** and **Polaris** modules.

- **Burn-up** dependence feature.

- It is provided with a friendly **user interface** in ASCII format.

- A friendly **manual** is also provided with examples.

- Optionally, it also generates **GEOM** auxiliary file for PARCS.
Feedback parameters...

- It is the user **responsibility** to introduce proper feedback parameters.
- These are set using **branches** in SCALE.
- There are three important things to consider.
  - Feedback **parameters** to be included: Tfuel, Dmod, Crod (this is fixed in the NEMTAB format).
  - Number of **points** for each parameter. More accurate library, but more computational time.
  - The **range** for the feedback parameters should cover the range of all possible operational conditions.
TRITON-NEWT

There are:

- *i* moderator density points
- *j* fuel temperature points
- one *boron* concentration point
- two control rod states (*in/out*)

The reactor history is constant for all branches (nominal state).

<table>
<thead>
<tr>
<th>Branch index</th>
<th>Feedback parameters</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>hdm, htm, htf, hbo, hcr=0</td>
<td>Nominal branch.</td>
</tr>
<tr>
<td>1</td>
<td>dm1, tm1, tf1, cb1, cr0</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>dm2, tm2, tf1, cb1, cr0</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>i</em></td>
<td>dmi, tmi, tf1, cb1, cr0</td>
<td><em>i</em> branches with dm, tm points.</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>ij – i+1</em></td>
<td>dm1, tm1, tfj, cb1, cr0</td>
<td>Repeat same branches <em>j</em> times with different tf points.</td>
</tr>
<tr>
<td><em>ij – i+2</em></td>
<td>dm2, tm2, tfj, cb1, cr0</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>ij</em></td>
<td>dmi, tmi, tfj, cb1, cr0</td>
<td></td>
</tr>
<tr>
<td><em>ij+1</em></td>
<td>dm1, tm1, tf1, cb1, cr1</td>
<td>Repeat <em>ij</em> branches with control rod inserted.</td>
</tr>
<tr>
<td><em>ij+2</em></td>
<td>dm2, tm2, tf1, cb1, cr1</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>ij+i</em></td>
<td>dmi, tmi, tf1, cb1, cr1</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2·<em>ij – i+1</em></td>
<td>dm1, tm1, tfj, cb1, cr1</td>
<td></td>
</tr>
<tr>
<td>2·<em>ij – i+2</em></td>
<td>dm2, tm2, tfj, cb1, cr1</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2·<em>ij</em></td>
<td>dmi, tmi, tfj, cb1, cr1</td>
<td>Total number of branches 2·ij +1</td>
</tr>
</tbody>
</table>

2. GUIDE
2. GUIDE

POLARIS

In order to maintain the analogy with txtfile16 generated by TRITON-NEWT, the first branch specified in Polaris must define the nominal state (or reactor history).

state FUEL : temp=htf
state COOL : temp=htm
state COOL : dens=hdm
state COOL : boron=hbo
state BANK : in=false

'branch - instantaneous change
read branch
'
   BRANCH:  NOM| Branch 1| Branch i| Branch ij | Branch ij+i CR| ...
   add FUEL : temp=htf  tf1  ...  tf1  ...  tfj  ...  ...  tf1  ...  ...
   COOL : temp=htm  tm1  ...  tmi  ...  tmi  ...  tmi  ...  ...
   COOL : dens=hdm  dm1  ...  dmi  ...  dmi  ...  dmi  ...  ...
   BANK : in=false  ...  ...  ...  ...  ...  ...  ...  true  ...  ...
end branch
POLARIS

- File txtfile16 generated by Polaris has some **discrepancies** compared to the txtfile16 generated by TRITON.

- User must **complete** some information for a couple of data blocks:
  - Block 3: Polaris does not write this block only for fresh fuel conditions.
  - Block 4: user must complete the feedback parameter information.

- This will be **automatized** in a future release of TXT2NTAB.

- More information can be found in the user **guide**.
Input files and data.

- fueltype.dat contains the fuel type distribution for the particular reactor (radial mapping).

- burnup.dat only provided for burn-up conditions (no fresh fuel), contains the burn-up distribution as a 3D map.

- input.dat is the input data provided by the user to define the reactor dimensions, segment information, parameters, feedback parameters...
  
  - Define as many fuel assemblies and segments as needed.
  - Option for cross section extrapolation for low moderator densities.
  - Three neutronic composition options.
  - NEMTAB generation for a specific burn-up point.
  - Optionally, generate GEOM auxiliary file for PARCS.

More information can be found in the user guide.
Input.dat example

- **Structured** with variables and cards.
- **Comments** are available with % symbol.
- In-code **checks** to reduce bugs.
- Whole description in user **guide**.
- And more ...

```plaintext
%radial reflector
ftyp 1
segm 1 1 1
fuel 0 0 0
absz -15.00 0.00 375.00 390.00

%CALCULATION PARAMETERS
pola 1  %optional, 1 if txtfile16 were obtained with Polaris module, 0 otherwise, default is 0 ("
nobu 1  %optional, 1 for fresh fuel, variable burn is then omitted, default is 1
intp linear  %optional, interpolation method accepted by MATLAB: linear or spline, default and recomm
extp 0  %optional, 1 for cross section extrapolation for low moderator densities, default is 0 ("
dlim 0.17075;  %limit for extrapolation, omitted if variable expt is 0
gexp 0  %optional, 1 to create GEOM file to be used with PARCS, default is 0
kexp 0  %optional, options are 0, 1 or 2. Option to compute number of neutronic composition, the:

%FEEDBACK PARAMETERS
%to look up in txtfile16
dm16 0.03814 0.17753 0.45632 0.73511 0.84034 0.94281 0.99829  %moderator density, increasing order
tm16 561.4 561.4 561.4 493 393 293  %moderator temperature for each moderator den.
tf16 293 879.5 1028.6 1396.5 1764.3 2132.2  %fuel temperature, increasing order
cb16 1935  %scalar only (no variation with boron concentration yet)
```
3. EXAMPLE CASE

- **Segment information** is given by IBERINCO.
- **Simplified** case to **test** the methodology.
- **Fresh** fuel conditions (computational resources).
- 6 fuel segments and 3 reflector segments.
- **Burn-up** conditions **implemented** (interpolation).
3. EXAMPLE CASE

- **Verification** with a code-to-code comparison with CASMO.

- CASMO outputs are given by IBERINCO.

- A MATLAB tool was developed to translate CASMO inputs into SCALE models.
3. EXAMPLE CASE

- **Verification** with a code-to-code comparison with **CASMO**.

- CASMO outputs are given by **IBERINCO**.

- A **MATLAB** tool was developed to translate **CASMO** inputs into **SCALE** models.

- **NEMTABs** can be obtained out of CASMO results and out of SCALE results.

- The same **PARCS model** can be run with both NEMTAB sets.

- Results comparison is used to **verify** the process.
3. EXAMPLE CASE

- Example of lattice model translated from CASMO and obtained with MATLAB tool.
- Represents only one segment of the core.
- TRITON-NEWT is used.
- 5 different UO$_2$ pin types.
- 1 Gd pin type.
- Two water rods.
3. EXAMPLE CASE

- User must decide what **feedback parameters** (branches) to include for each segment.

- Some **branches** can be **discarded** to save computational time.

- However, there is no problem if the user wants to simulate **all branches**.
Data shown hereafter are full core results obtained with **PARCSv3.2**, either with NEMTABs out of SCALE (with TXT2NTAB) or with NEMTABs out of CASMO.

<table>
<thead>
<tr>
<th></th>
<th>CASMO</th>
<th>SCALE v7-56</th>
<th>SCALE v7-252</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Default</td>
<td>MCDANCOFF</td>
</tr>
<tr>
<td>ARO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_{\text{eff}}$</td>
<td>1.03171</td>
<td>1.03601</td>
<td>1.03417</td>
</tr>
<tr>
<td>Error (pcm)</td>
<td>-</td>
<td>429.5</td>
<td>246.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>CASMO</th>
<th>SCALE v7-56</th>
<th>SCALE v7-252</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Default</td>
<td>MCDANCOFF</td>
</tr>
<tr>
<td>ARI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_{\text{eff}}$</td>
<td>0.79702</td>
<td>0.80023</td>
<td>0.79893</td>
</tr>
<tr>
<td>Error (pcm)</td>
<td>-</td>
<td>321.1</td>
<td><strong>191.5</strong></td>
</tr>
</tbody>
</table>
### 3. EXAMPLE CASE

<table>
<thead>
<tr>
<th></th>
<th>RMS$_{\text{axial}}$ (%)</th>
<th></th>
<th>RMS$_{\text{axial}}$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>V7-56</td>
<td>V7-252</td>
<td>V7-56</td>
</tr>
<tr>
<td>Default</td>
<td>0.24</td>
<td>0.21</td>
<td>11.98</td>
</tr>
<tr>
<td>MCDANCOFF</td>
<td>0.24</td>
<td>0.21</td>
<td>12.09</td>
</tr>
</tbody>
</table>

**Graphs:**

- **Left Graph:**
  - CASMO (black)
  - SCALE (red)
  - Power level vs. Axial level

- **Right Graph:**
  - CASMO (black)
  - SCALE (red)
  - Power level vs. Axial level
### 3. EXAMPLE CASE

#### ARO $\text{RMS}_{\text{radial}}$ (%)

<table>
<thead>
<tr>
<th></th>
<th>V7-56</th>
<th>V7-252</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>0.53</td>
<td>0.53</td>
</tr>
<tr>
<td>MCDANCOFF</td>
<td>0.53</td>
<td>0.53</td>
</tr>
</tbody>
</table>

#### ARI $\text{RMS}_{\text{radial}}$ (%)

<table>
<thead>
<tr>
<th></th>
<th>V7-56</th>
<th>V7-252</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>1.75</td>
<td>1.73</td>
</tr>
<tr>
<td>MCDANCOFF</td>
<td>1.74</td>
<td>1.72</td>
</tr>
</tbody>
</table>

#### Absolute radial error (%)

![Graph showing absolute radial error](image_url)

- **x-cell**
- **y-cell**

![Color bar for error levels](image_url)
With **accurate Dancoff** factors the $k_{\text{eff}}$ difference is **reduced** almost 50%.

Their influence over the axial and radial **power** profiles is **negligible**.

It was found that when accurate Dancoff factors are used, an unexpected **increase** in **memory** was required by **SCALE**.

<table>
<thead>
<tr>
<th></th>
<th>$k_{\text{eff}}$ (pcm)</th>
<th>$\text{RMS}_{\text{axial}}$ (%)</th>
<th>$\text{RMS}_{\text{radial}}$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>V7-56</td>
<td>V7-252</td>
<td>V7-56</td>
</tr>
<tr>
<td><strong>Default</strong></td>
<td>429.5</td>
<td>362.1</td>
<td>0.24</td>
</tr>
<tr>
<td><strong>MCDANCOFF</strong></td>
<td>246.0</td>
<td>244.9</td>
<td>0.24</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>$k_{\text{eff}}$ (pcm)</th>
<th>$\text{RMS}_{\text{axial}}$ (%)</th>
<th>$\text{RMS}_{\text{radial}}$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>V7-56</td>
<td>V7-252</td>
<td>V7-56</td>
</tr>
<tr>
<td><strong>Default</strong></td>
<td>321.1</td>
<td>308.9</td>
<td>11.98</td>
</tr>
<tr>
<td><strong>MCDANCOFF</strong></td>
<td>191.5</td>
<td>223.3</td>
<td>12.09</td>
</tr>
</tbody>
</table>
3. EXAMPLE CASE

- The **master library** has an effect over the $k_{\text{eff}}$ (20 – 70 pcm).

- But it is almost **irrelevant** to the power **profiles**.

- In general, the $k_{\text{eff}}$ difference is smaller when the **v7-252** is used.

<table>
<thead>
<tr>
<th>ARO</th>
<th>$k_{\text{eff}}$ (pcm)</th>
<th>RMS$_{\text{axial}}$ (%)</th>
<th>RMS$_{\text{radial}}$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>V7-56</td>
<td>V7-252</td>
<td>V7-56</td>
</tr>
<tr>
<td>Default</td>
<td>429.5</td>
<td>362.1</td>
<td>0.24</td>
</tr>
<tr>
<td>MCDANCOFF</td>
<td>246.0</td>
<td>244.9</td>
<td>0.24</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ARI</th>
<th>$k_{\text{eff}}$ (pcm)</th>
<th>RMS$_{\text{axial}}$ (%)</th>
<th>RMS$_{\text{radial}}$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>V7-56</td>
<td>V7-252</td>
<td>V7-56</td>
</tr>
<tr>
<td>Default</td>
<td>321.1</td>
<td>308.9</td>
<td>11.98</td>
</tr>
<tr>
<td>MCDANCOFF</td>
<td>191.5</td>
<td>223.3</td>
<td>12.09</td>
</tr>
</tbody>
</table>
TXT2NTAB program is developed to fill the gap between SCALE and a core simulator.

Problem-dependent cross section libraries in NEMTAB format can be obtained from TRITON or Polaris.

The choice of feedback parameters is the user responsibility.

Significant computational time could be needed with TRITON models (use of Polaris).

A friendly ASCII user interface is used to introduce the data with ease.

A user guide is provided.

4. CONCLUSIONS
5. ONGOING WORK

TXT2NTAB is actively being updated...

✓ For **U&S** analysis. Process **SAMPLER** perturbed txtfile16 to generate...
  • One set of NEMTAB/r with the **average** response.
  • One set of NEMTAB/r with their **standard deviation**.

✓ Handle **branches** of **Polaris** without the user intervention.

✓ Generate NEMTABs for different **reactor histories**.
THANK YOU!

You can forward any question or comment to ...

cmesado@iqn.upv.es

Besides you can download TXT2NTAB User Guide and the author PhD Thesis here ...

goo.gl/nbmxNp