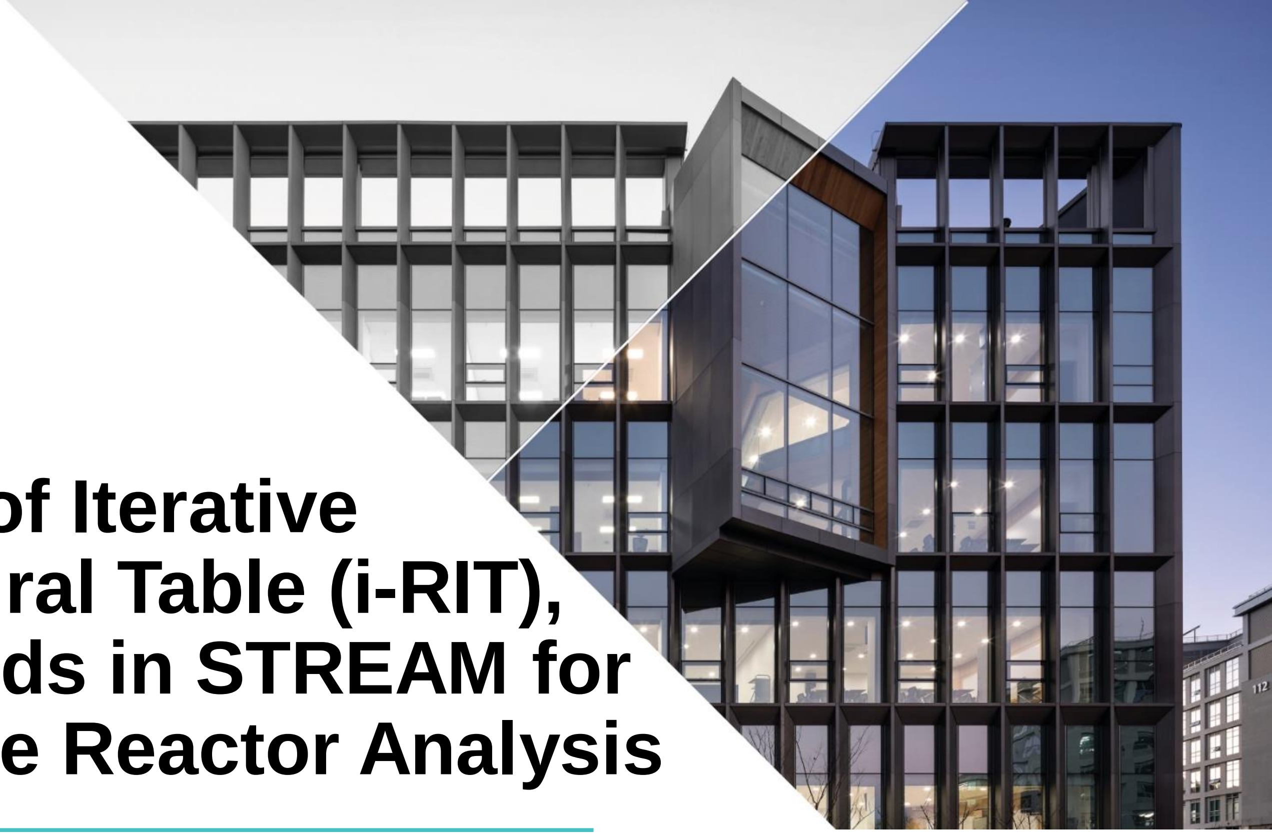


Implementation of Iterative Resonance Integral Table (i-RIT), Subgroup Methods in STREAM for High Temperature Reactor Analysis

2018. 5. 18 (Friday)

Chidong Kong

Chang Keun Jo

Deokjung Lee

CONTACT

Ulsan National Institute of Science and Technology
Address 50 UNIST-gil, Ulju-gun, Ulsan, 44919, Korea
Tel. +82 52 217 0114 **Web.** www.unist.ac.kr

CORE Computational Reactor physics & Experiment lab
Tel. +82 52 217 2940 Web. reactorcore.unist.ac.kr

Table of Contents

Introduction

- **As one of the generation four nuclear reaction designs, a very high temperature gas-cooled reactor (VHTR) has been in the spotlight with its high safety feature.**

- **For VHTR compact problem, iterative Resonance Integral Table (i-RIT) method has been implemented in STREAM.**

- **For VHTR compact problem, subgroup method has been implemented in STREAM.**

- **The two methods have been tested in STREAM with a newly generated STREAM 190G and 220G libraries.**

Methodology

▪ i-RIT(b) method

- **S. G. Hong, K.-S. Kim, "Iterative resonance self-shielding methods using resonance integral table in heterogeneous transport lattice calculation," *Ann. Nucl. Energy*, 38, pp. 32-43, 2011.**

$$\Omega \cdot \nabla \psi + \left(\Sigma_a + \lambda \Sigma_p^{mat}\right)\psi = \frac{1}{4\pi}\lambda \Sigma_p^{mat}$$

1. Initial guess

2. Solve MOC

$$\sigma_b^{MOC} = \frac{1}{N_r} \cdot \frac{\Sigma_a \phi}{(1-\phi)}$$

3. Get RI from library

$$RI(\sigma_b^{MOC})$$

4. Obtain effective σ_a^r

$$\sigma_a^r = \frac{RI(\sigma_b^{MOC})/\Delta u}{1-\left(RI(\sigma_b^{MOC})/\Delta u\right)/\sigma_b^{MOC}}$$

5. Update Σ_a

$$\Sigma_a = \sum_{r \in R} N_r \sigma_a^r$$

6. Iteration until Σ_a converges

Methodology

▪ i-RIT(a) method

- **S. G. Hong, K.-S. Kim, “Iterative resonance self-shielding methods using resonance integral table in heterogeneous transport lattice calculation,” *Ann. Nucl. Energy*, 38, pp. 32-43, 2011.**
- **Method of equivalence XS tabulation**

$$\Sigma_{b,c,g}^{MOC}(\sigma_{a,c,r,g,m}) = \sum_i \lambda \Sigma_p^i + \Sigma_{eq,c,g}(\sigma_{a,c,r,g,m}) \xrightarrow{\text{tabulation}} \Sigma_{eq,c,g}(\sigma_{a,c,r,g,m}) = \Sigma_{b,c,g}^{MOC}(\sigma_{a,c,r,g,m}) - \sum_i \lambda \Sigma_p^i$$

$$\sigma_{b,k,g}^{(l)} = \frac{\sum_i \lambda \Sigma_p^i}{N_k}$$

1. Initial guess

2. Update $\sigma_{a,k,g}^{(l)}$

$$\sigma_{a,k,g}^{(l)} = \frac{R_a(\sigma_{b,k,g}^{(l)})}{1 - \frac{R_a(\sigma_{b,k,g}^{(l)})}{\sigma_{b,k,g}^{(l)}}}$$

3. Update $\sigma_{k,g}^{*,(l)}$

$$\sigma_{k,g}^{*,(l)} = \frac{R_{a,\infty,c,r,g}}{R_{a,\infty,c,k,g}} \sigma_{a,k,g}^{(l)}$$

4. Update $\Sigma_{eq,k,g}^{(l+1)}$

$$\Sigma_{eq,k,g}^{(l+1)} = G_{eq,c,g}(\sigma_{k,g}^{*,(l)})$$

5. Update $\sigma_{b,k,g}^{(l+1)}$

6. Iteration until $\sigma_{b,k,g}$ converges

Methodology

- **Subgroup method**

$$\phi_k = \frac{\sum_i \lambda_i N_i \sigma_p^i + \Sigma_e^k}{N_r \sigma_{ak} + \sum_i \lambda_i N_i \sigma_p^i + \Sigma_e^k} = \frac{\sigma_b}{\sigma_{ak} + \sigma_b}, \quad \text{where} \;\; \sigma_b = \frac{1}{N_r}\left(\sum_i \lambda_i N_i \sigma_p^i + \Sigma_e^k\right)$$

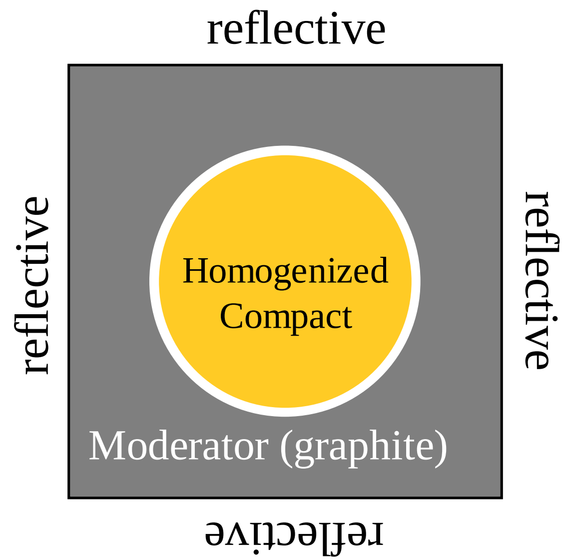
$$\sigma_{bk} = \frac{\sigma_{ak}\phi_k}{(1-\phi_k)}$$

$$\sigma_{a,eff} = \frac{\sum_{k=1}^{n} \omega_k \sigma_{ak} \dfrac{\sigma_{bk}}{\sigma_{ak} + \sigma_{bk}}}{\sum_{k=1}^{n} \omega_k \dfrac{\sigma_{bk}}{\sigma_{ak} + \sigma_{bk}}}$$

- **Subgroup Parameter Generation in STREAM**
  - **Generated in homogeneous 0-D problem**
  - **Using NJOY code**

Problem Description

- **VHTR homogenized compact problem**
  - **Fuel: UO_2+Graphite**
  - **Moderator: Graphite**
  - **Packing fraction: 1 ~ 60% (Total 40 cases)**



Region	Material	Geometry	Radius or Pitch [cm]
Fuel	UO_2 + graphite	Cylinder	0.6225 (Radius)
Gap	Helium	Cylinder	0.6350 (Radius)
Moderator	Graphite	Rectangle	1.749165 (Pitch)

Verification of STREAM Solver Module

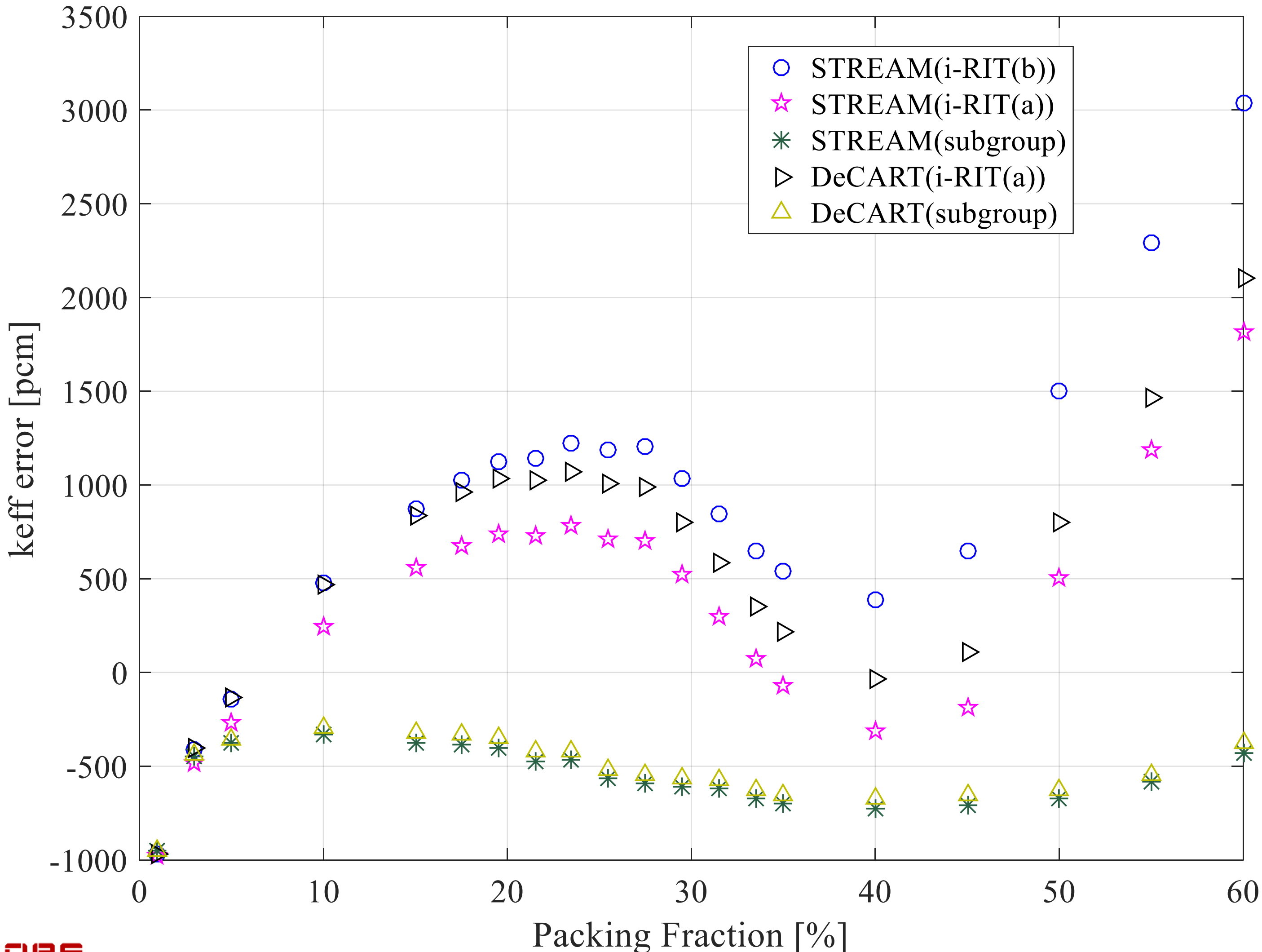
- **Using the same DeCART 190G library**

Packing Fraction [%]	*MCS reference	STREAM (i-RIT(b))	STREAM (i-RIT(a))	STREAM (Subgroup)	DeCART (i-RIT(a))	DeCART (Subgroup)
1	1.50988	1.50020	1.50008	1.50035	1.50023	1.50039
3	1.63471	1.63060	1.62991	1.63022	1.63067	1.63033
5	1.60550	1.60410	1.60285	1.60171	1.60417	1.60189
10	1.50458	1.50932	1.50701	1.50128	1.50924	1.50161
15	1.42293	1.43167	1.42853	1.41920	1.43127	1.41969
17.5	1.38857	1.39886	1.39535	1.38476	1.39819	1.38527
19.5	1.36376	1.37497	1.37116	1.35974	1.37406	1.36026
21.5	1.34140	1.35278	1.34866	1.33666	1.35162	1.33720
23.5	1.32004	1.33231	1.32789	1.31541	1.33076	1.31585
25.5	1.30125	1.31313	1.30840	1.29559	1.31130	1.29605
27.5	1.28312	1.29513	1.29017	1.27717	1.29304	1.27763
29.5	1.26611	1.27645	1.27129	1.26001	1.27414	1.26048
31.5	1.25023	1.25871	1.25325	1.24401	1.25609	1.24449
33.5	1.23578	1.24228	1.23648	1.22907	1.23933	1.22955
35	1.22554	1.23096	1.22486	1.21852	1.22770	1.21900
40	1.19420	1.19805	1.19103	1.18696	1.19386	1.18745
45	1.16738	1.17387	1.16551	1.16031	1.16843	1.16081
50	1.14460	1.15958	1.14963	1.13785	1.15258	1.13835
55	1.12494	1.14789	1.13678	1.11911	1.13961	1.11952
60	1.10764	1.13805	1.12577	1.10333	1.12868	1.10385

*MCS solutions have 15~25 pcm of standard deviations

Verification of STREAM Solver Module

- **Using the same DeCART 190G library**
  - **Well-matched between STREAM and DeCART solutions**

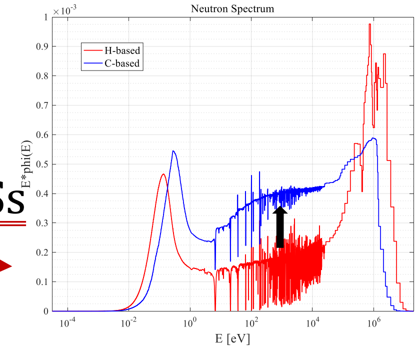


UNIST CORE

C-Based 190G STREAM Library Production

- **NJOY 99.364**

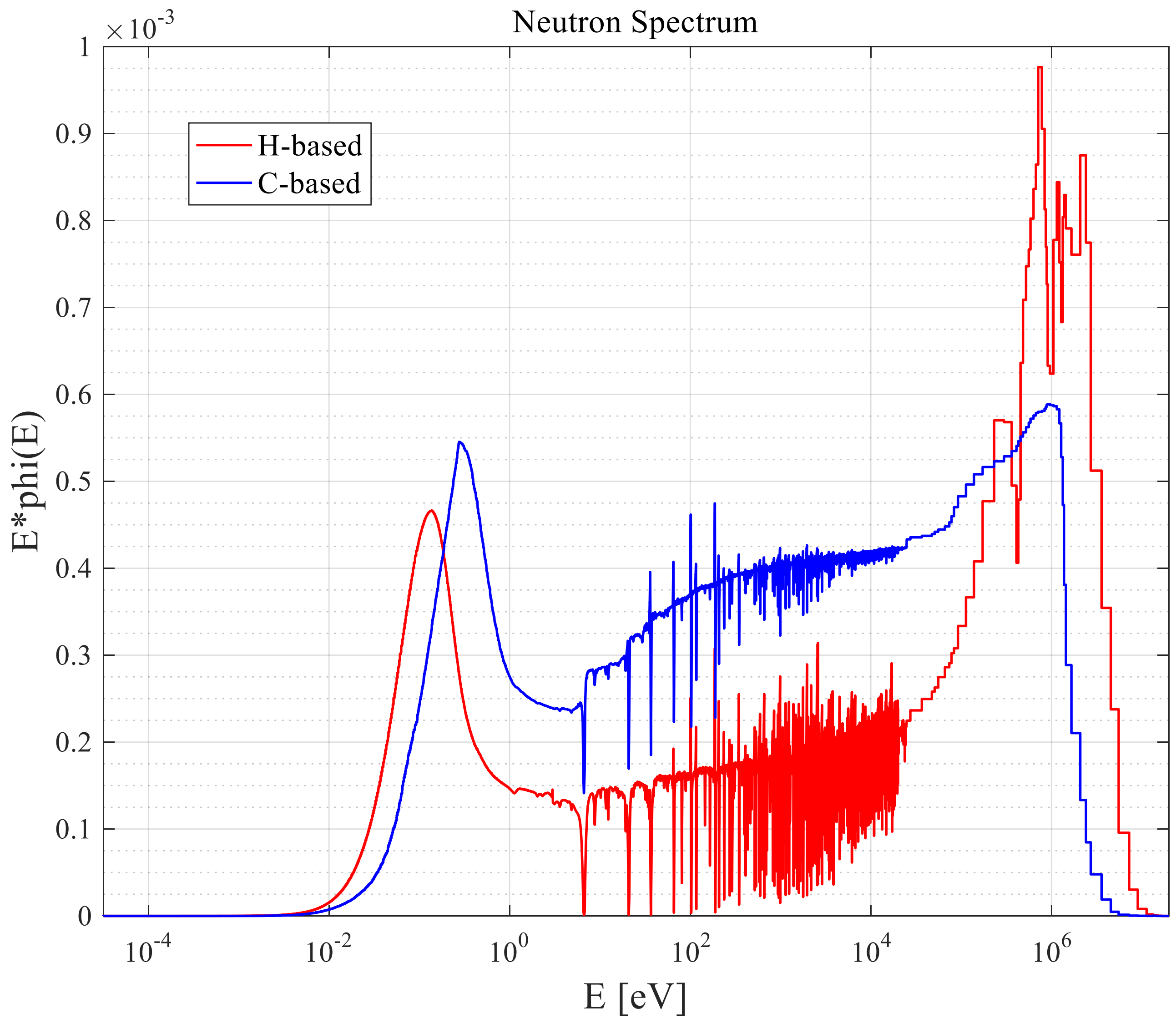
RECONR	Getting ENDF resonance parameters, and processing XSs in resolved resonance region for some nuclides
BROADR	Processing Doppler broadening
UNRESR	Processing XSs in unresolved resonance region
HEATR	Processing heating XSs and radiation damage XSs
THERMR	Processing S(α,β) XSs
GROUPR	Collasing multi-group XSs from point-wise XSs



- **Post-processing of NJOY output**
  - **Merging outputs of each nuclide and generating a multi-group data file**
- **RUP, IR parameter**
  - **190G RUP and IR parameters**

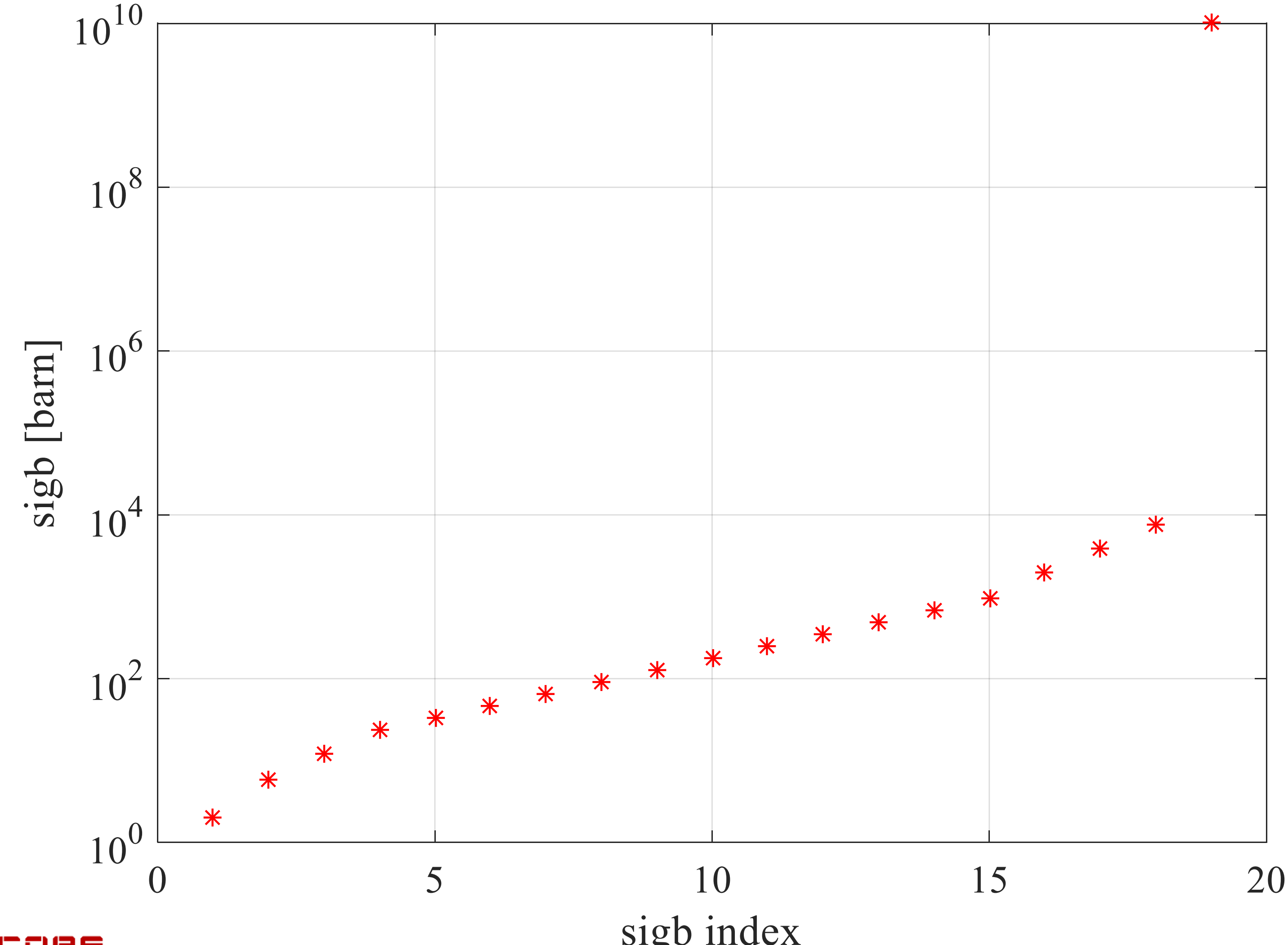
C-Based 190G Spectrum Correction

- **Spectrum Difference between H-based and C-based Problems**



Subgroup Parameter Generation

- **Subgroup Parameter Generation**
  - **19 background XSs for ^{235}U and ^{238}U in STREAM**

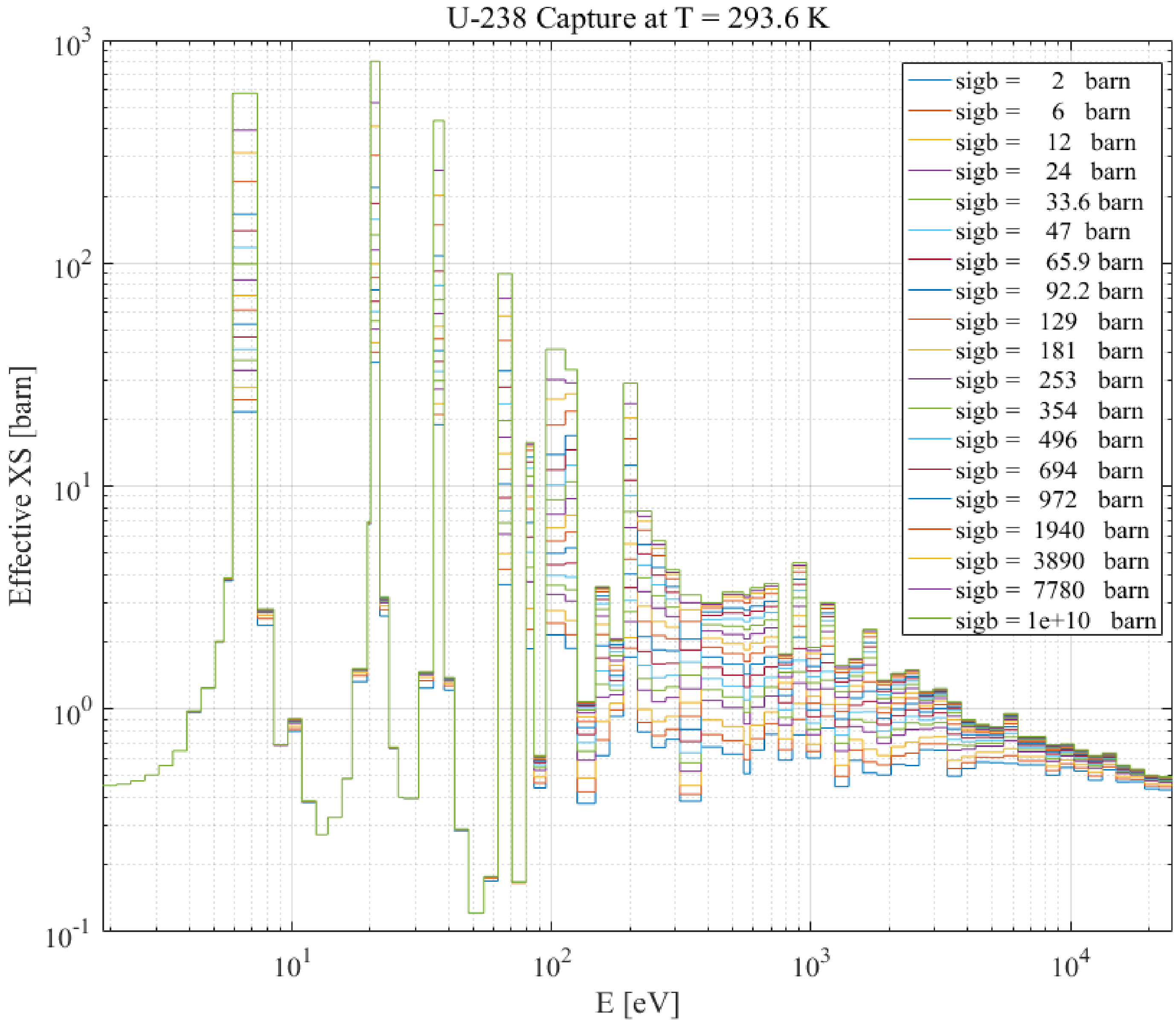


Subgroup Parameter Generation

- **NJOY-generating Effective XSs**
  - **Representing resonance integrals (RIs)**
  - **Total 56 sets of effective cross sections for each resonance energy group**
    - 2 nuclides : ^{235}U, ^{238}U
    - 2 reactions : Capture, Fission
    - 7 temperatures : 293.6 K, 600 K, 900 K, 1200 K, 1500 K, 1800 K, 2100 K
    - 2 multi-group libraries : 190G, 220G

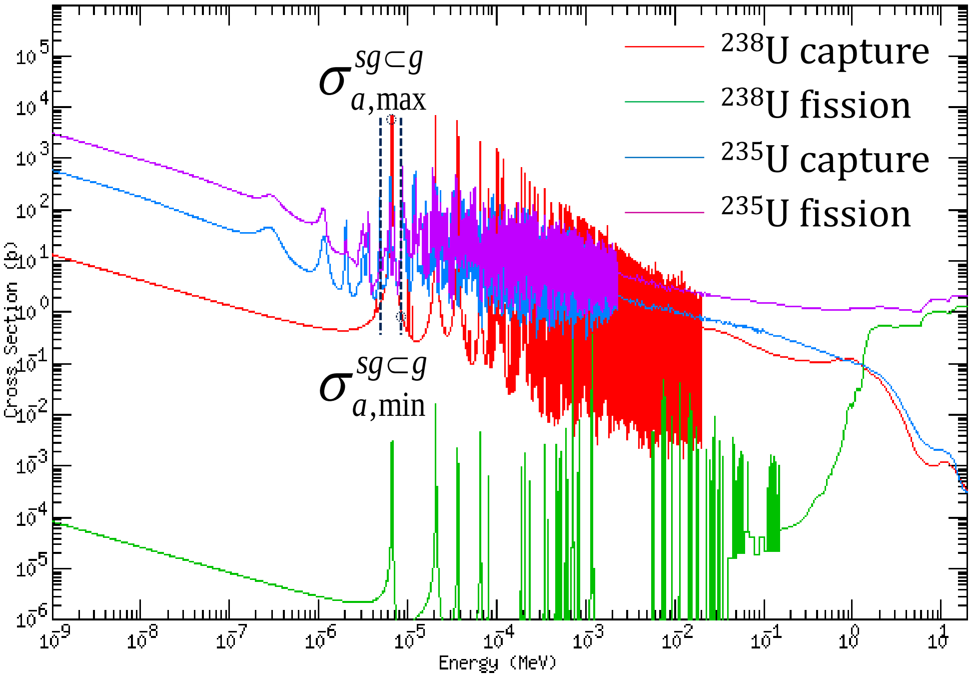
Subgroup Parameter Generation

- **NJOY-generating Effective XSs**



Subgroup Parameter Generation

- **Determination of Subgroup Level**
  1) **Get continuous energy XS for the resonance energy groups**
  2) **Get the minimum and the maximum XSs in the given energy group**
  3) **Get a difference between the minimum and the maximum**
  4) **Divide the difference into *N*+1 in linear scale**
  5) **Generation *N* subgroup levels with the same logarithm interval**



$$\sigma_{an}^{sg\subset g} = \sigma_{a,\min}^{sg\subset g} \cdot e^{\frac{n}{N+1}\times\left(\log\left(\sigma_{a,\max}^{sg\subset g}\right)-\log\left(\sigma_{a,\min}^{sg\subset g}\right)\right)}$$

in logarithm scale,

where $1 \le n \le N$ and $N = 7$

UNIST CORE

Subgroup Parameter Generation

- **Subgroup Weight Generation**

$$\sigma_{ak,eff} \cong \frac{\sum_{n=1}^{7}\left(\omega_n \cdot \sigma_{an} \cdot \frac{\sigma_{bk}}{\sigma_{an}+\sigma_{bk}}\right)}{1-\sum_{n=1}^{7}\left(\omega_n \cdot \frac{\sigma_{an}}{\sigma_{an}+\sigma_{bk}}\right)} \quad where \;\; 1 \le k \le 19$$

$$\left[\sigma_{ak,eff} - \sum_{n=1}^{7}\left(\omega_n \cdot \sigma_{an} \cdot \frac{\sigma_{ak,eff}+\sigma_{bk}}{\sigma_{an}+\sigma_{bk}}\right)\right]^2 \cong 0 \;\; and \;\; \sum_{n=1}^{7}\omega_n \cong 1$$

$$\mathrm{AW} = \Sigma \;\Leftrightarrow\; \begin{bmatrix} \sigma_{a1} \cdot \frac{\sigma_{a1,eff}+\sigma_{b1}}{\sigma_{a1}+\sigma_{b1}} & \cdots & \sigma_{a7} \cdot \frac{\sigma_{a1,eff}+\sigma_{b1}}{\sigma_{a7}+\sigma_{b1}} \\ \vdots & \ddots & \vdots \\ \sigma_{a1} \cdot \frac{\sigma_{a19,eff}+\sigma_{b19}}{\sigma_{a1}+\sigma_{b19}} & \cdots & \sigma_{a7} \cdot \frac{\sigma_{a19,eff}+\sigma_{b19}}{\sigma_{a7}+\sigma_{b19}} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \vdots \\ \omega_7 \end{bmatrix} \cong \begin{bmatrix} \sigma_{a1,eff} \\ \sigma_{a2,eff} \\ \vdots \\ \sigma_{a18,eff} \\ \sigma_{a19,eff} \end{bmatrix}$$

$$19 \times 7 \qquad\qquad 7 \times 1 \qquad 19 \times 1$$

Subgroup Parameter Generation

▪ Subgroup Weight Generation

$$A^{T} AW = A^{T}\Sigma$$

⇕

$$\begin{bmatrix} \sum_{k=1}^{19}\left(\sigma_{a1}\cdot\frac{\sigma_{ak,eff}+\sigma_{bk}}{\sigma_{a1}+\sigma_{bk}}\right)^{2} & \cdots & \sum_{k=1}^{19}\left(\sigma_{a1}\cdot\sigma_{a7}\cdot\frac{\left(\sigma_{ak,eff}+\sigma_{bk}\right)^{2}}{\left(\sigma_{a1}+\sigma_{bk}\right)\left(\sigma_{a7}+\sigma_{bk}\right)}\right) \\ \vdots & \ddots & \vdots \\ \sum_{k=1}^{19}\left(\sigma_{a7}\cdot\sigma_{a1}\cdot\frac{\left(\sigma_{ak,eff}+\sigma_{bk}\right)^{2}}{\left(\sigma_{a7}+\sigma_{bk}\right)\left(\sigma_{a1}+\sigma_{bk}\right)}\right) & \cdots & \sum_{k=1}^{19}\left(\sigma_{a7}\cdot\frac{\sigma_{ak,eff}+\sigma_{bk}}{\sigma_{a7}+\sigma_{bk}}\right)^{2} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \vdots \\ \omega_7 \end{bmatrix} \cong \begin{bmatrix} \sigma_{a1}\sum_{k=1}^{19}\left(\sigma_{ak,eff}\cdot\frac{\sigma_{ak,eff}+\sigma_{bk}}{\sigma_{a1}+\sigma_{bk}}\right) \\ \vdots \\ \sigma_{a7}\sum_{k=1}^{19}\left(\sigma_{ak,eff}\cdot\frac{\sigma_{ak,eff}+\sigma_{bk}}{\sigma_{a7}+\sigma_{bk}}\right) \end{bmatrix}$$

$$7\times 7 \qquad\qquad 7\times 1 \qquad\qquad 7\times 1$$

Subgroup Parameter Generation

▪ Subgroup Weight Generation

- **MATLAB algorithm**

$\omega_n^{(0)}$ initialization

Start loop (i)

$$A^T A W^{(i)} = A^T \lambda^{(i)} \Sigma \quad \text{where} \quad \lambda^{(i)} = \sum_{n=1}^{N} \omega_n^{(i)}$$

Least square fitting using *fminsearch* function

$$W^{(i+1)} = W^{(i)} / sum(W^{(i)})$$

$$\lambda^{(i+1)} = \sum_{n=1}^{N} \omega_n^{(i+1)}$$

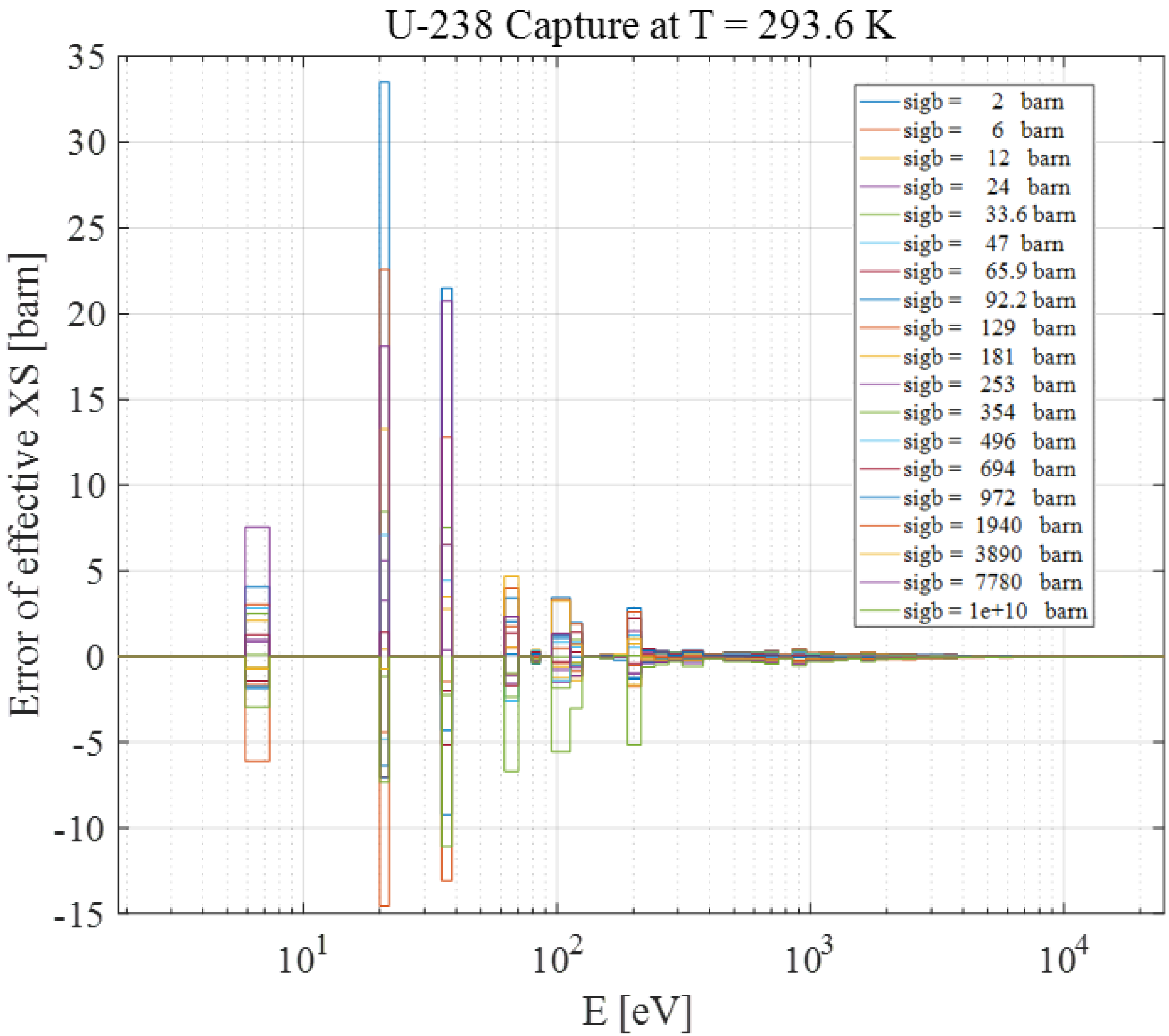
$$\textbf{if} \quad \frac{\sum_{n=1}^{N} \left(\omega_n^{(i+1)} - \omega_n^{(i)} \right)^2}{N} < eps \quad \Rightarrow \quad \textbf{break}$$

End loop (i)

UNIST CORE

Sum(wgt * sign * phi) / sum(wgt * phi) – siga(effective)

- **Error Check : 190G Subgroup Weight**



UNIST CORE

Numerical Results

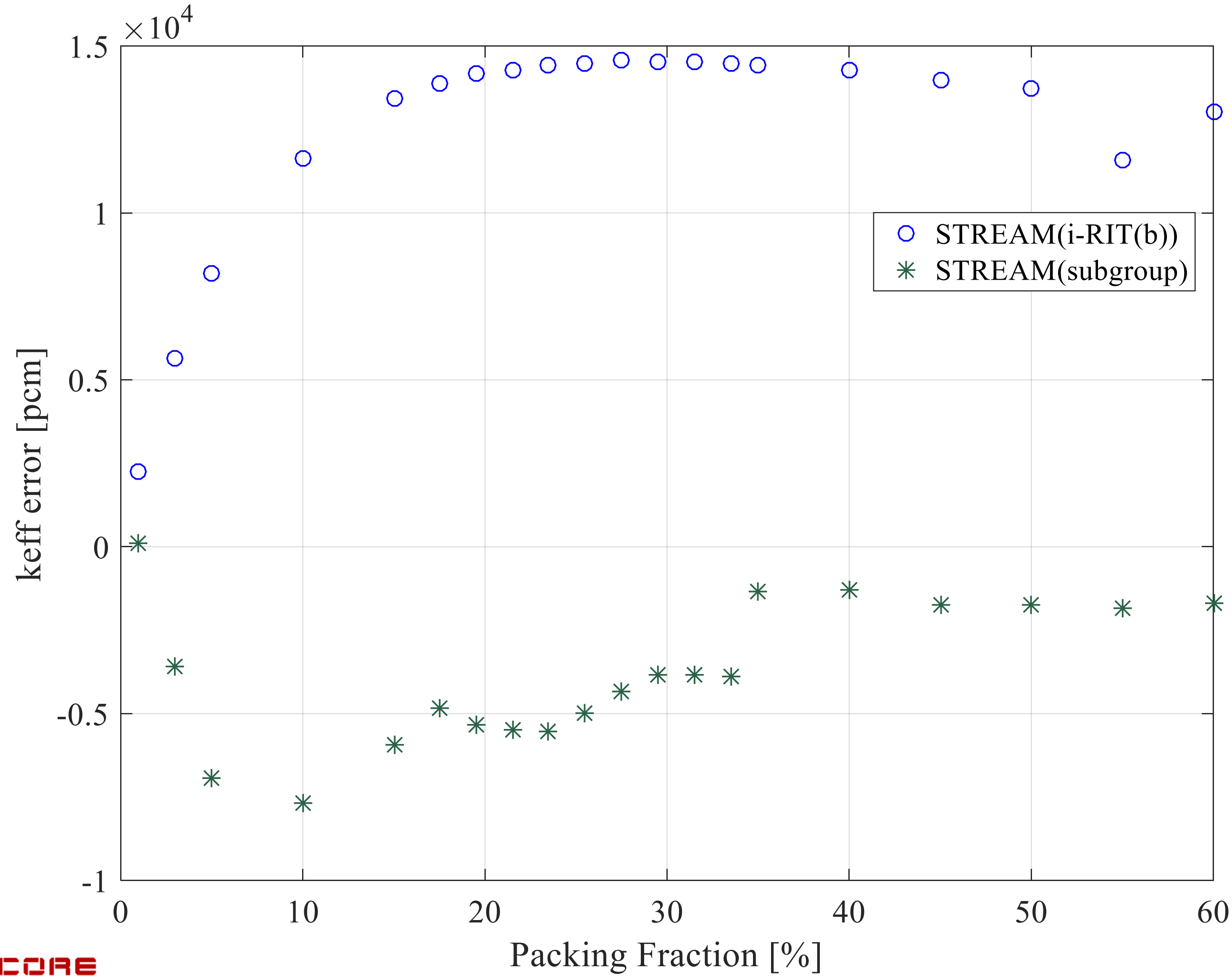
- **Using a new STREAM 190G library**

Packing Fraction [%]	*MCS reference	STREAM (i-RIT(b))	STREAM (Subgroup)
1	1.50988	1.53254	1.51079
3	1.63471	1.69131	1.59898
5	1.60550	1.68741	1.53638
10	1.50458	1.62109	1.42773
15	1.42293	1.55719	1.36357
17.5	1.38857	1.52727	1.34041
19.5	1.36376	1.50563	1.31065
21.5	1.34140	1.48435	1.28642
23.5	1.32004	1.46440	1.26461
25.5	1.30125	1.44595	1.25134
27.5	1.28312	1.42884	1.23995
29.5	1.26611	1.41158	1.22780
31.5	1.25023	1.39551	1.21179
33.5	1.23578	1.38050	1.19678
35	1.22554	1.36989	1.21199
40	1.19420	1.33693	1.18124
45	1.16738	1.30731	1.14978
50	1.14460	1.28189	1.12694
55	1.12494	1.24078	1.10648
60	1.10764	1.23784	1.09081

*MCS solutions have 15~25 pcm of standard deviations

Numerical Results

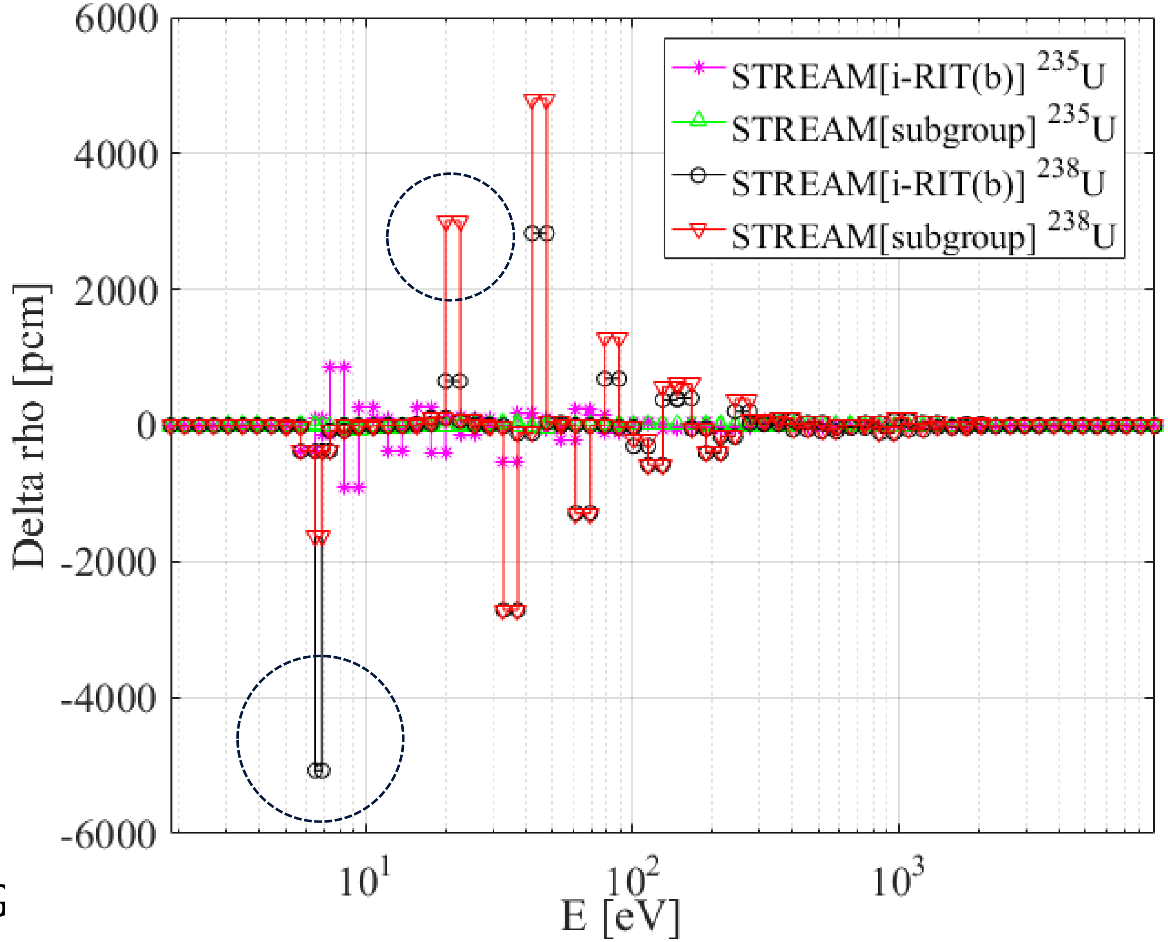
- **Using a new STREAM 190G library**



UNIST CORE

Numerical Results

- **Using a new STREAM 220G library**



190G → 220G

6.476 ~ 6.868 eV : **10** divisions (0.04 eV interval)

19.947 ~ 22.603 eV : **22** divisions (0.12 eV interval)

UNIST CORE

Numerical Results

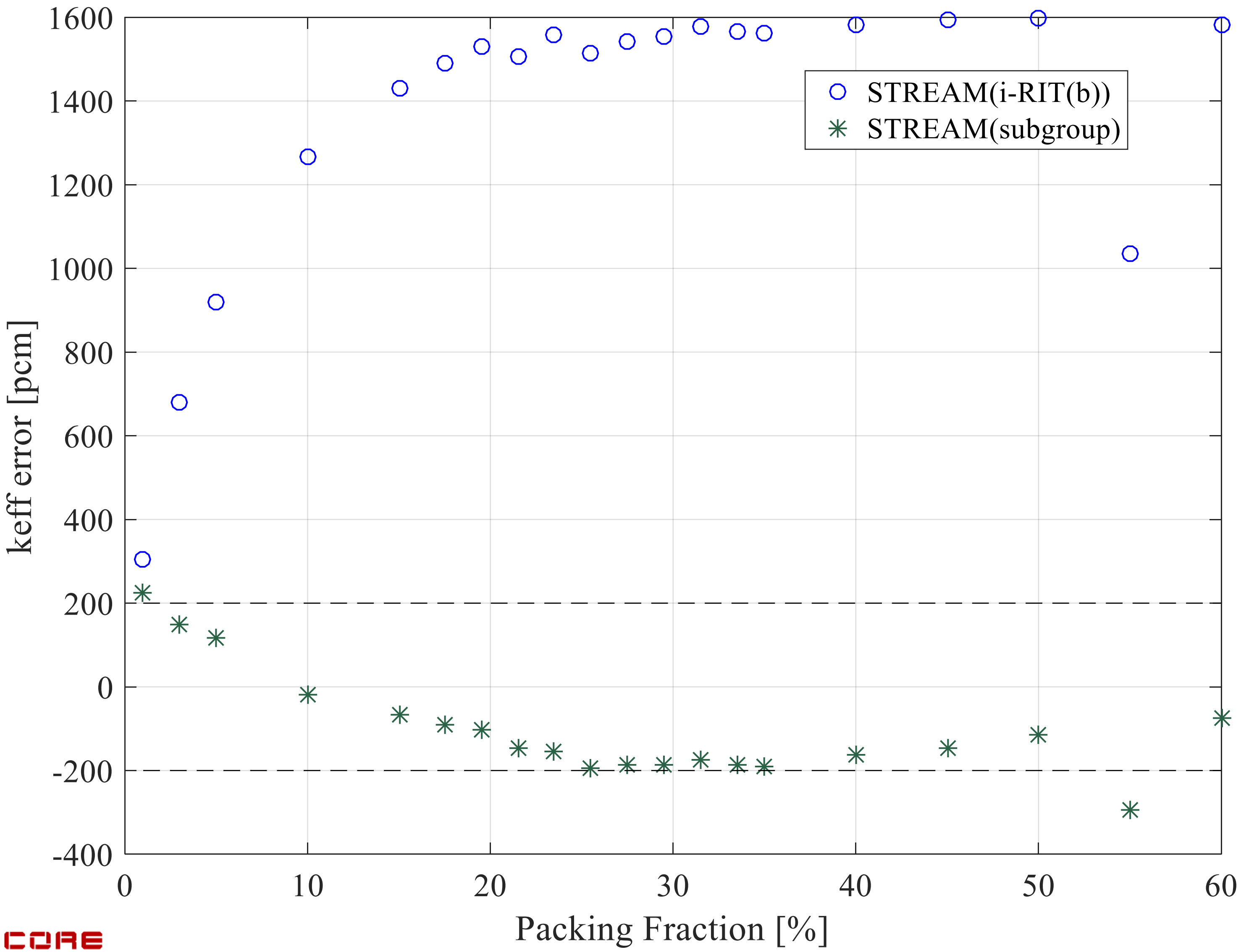
- **Using a new STREAM 220G library**

Packing Fraction [%]	*MCS reference	STREAM (i-RIT(b))	STREAM (Subgroup)
1	1.50988	1.51294	1.51214
3	1.63471	1.64152	1.63618
5	1.60550	1.61468	1.60666
10	1.50458	1.51724	1.50441
15	1.42293	1.43725	1.42225
17.5	1.38857	1.40347	1.38765
19.5	1.36376	1.37907	1.36272
21.5	1.34140	1.35647	1.33995
23.5	1.32004	1.33564	1.31848
25.5	1.30125	1.31640	1.29931
27.5	1.28312	1.29853	1.28124
29.5	1.26611	1.28165	1.26425
31.5	1.25023	1.26602	1.24849
33.5	1.23578	1.25144	1.23391
35	1.22554	1.24116	1.22364
40	1.19420	1.21004	1.19258
45	1.16738	1.18333	1.16590
50	1.14460	1.16059	1.14344
55	1.12494	1.13530	1.12199
60	1.10764	1.12348	1.10691

*MCS solutions have 15~25 pcm of standard deviations

Numerical Results

- **Using a new STREAM 220G library**



Conclusions

- **i-RIT and subgroup methods were implemented in STREAM and they were verified with the DeCART library.**
- **For a newly generated 190G STREAM library, both methods showed large keff errors compared to MCS reference keffs.**
- **For a newly generated 220G STREAM library slicing into resonance peaks, both methos showed improved accuracy.**
- **Especially, the subgroup method showed very high accuracy with keff errors below 300 pcm.**

Future Plan

- **Subgroup methods will be tested in STREAM with STREAM's 72G PWR library.**

- **Two-term expansion (flux-sigb relation) of the subgroup method will be implemented in STREAM to get higher accuracy.**

$$\phi_k = \frac{\sum_i \lambda_i N_i \sigma_p^i + \Sigma_e^k}{N_r \sigma_{ak} + \sum_i \lambda_i N_i \sigma_p^i + \Sigma_e^k} = \frac{\sigma_b}{\sigma_{ak} + \sigma_b}, \quad \text{where} \quad \sigma_b = \frac{1}{N_r}\left(\sum_i \lambda_i N_i \sigma_p^i + \Sigma_e^k \right)$$

One-term based

$$\sigma_{bk} = \frac{\sigma_{ak}\phi_k}{(1-\phi_k)}$$

$$\sigma_{a,eff} = \frac{\sum_{k=1}^{n} \omega_k \sigma_{ak} \frac{\sigma_{bk}}{\sigma_{ak}+\sigma_{bk}}}{\sum_{k=1}^{n} \omega_k \frac{\sigma_{bk}}{\sigma_{ak}+\sigma_{bk}}}$$

UNIST CORE