

The NEREA Python Package for Integral Experiment Analysis: a Fission Chamber Calibration Case Study

Federico Grimaldi^{1,2,*}, Federico Di Croce^{1,2}, Antonin Krása¹, Jan Wagemans¹, and Guido Vittiglio¹
¹SCK CEN Belgian nuclear research centre, Boeretang 200, Mol, 2400, Antwerp, Belgium
²Université libre de Bruxelles, Avenue Franklin D. Roosevelt 50, Bruxelles, 1050, Belgium
(*) federico.grimaldi@sckcen.be

Abstract—NEREA is a Python package designed for integral experiment processing in nuclear reactors. In this work we present the main features of NEREA to date, including data processing for pulse height spectrum analysis, fission chamber calibration, fission rate traverse measurement processing, control rod calibration and reactivity worth calculation. NEREA offers high control of the processing flow yet keeping it smooth and user-friendly thanks to its object-oriented conception.

We showcase the capabilities of NEREA presenting the calibration of a large set of fission chambers at the BR1 MARK-III converter. NEREA allows for analysis of the impact of processing choices and for considerations over several measurements. Strengths and weaknesses of the implemented calibration methodology are also discussed, highlighting room for future work on this. This is particularly true when the measured pulse height spectrum is characterized by low peak-to-valley ratio, making the choice of a proper discrimination level challenging.

Keywords —Fission Chambers, MARK-III converter, NEREA Python API, Integral Experiments.

I. INTRODUCTION

VENUS-F is a fast spectrum zero power research reactor operated at the Belgian Nuclear Research Centre SCK CEN. Integral experiments have been performed there in support of nuclear data improvement [1] [2] [3] [4] and code development [5]. Research on fast heavy-metal-cooled reactors has recently renewed the interest in VENUS-F critical configurations, with new experimental campaigns conducted in the OFFERR European Commission framework. Moreover, since 2021, students have been more and more involved in the research activities at VENUS-F with Ph.D., Master and Bachelor theses.

With the new activities and the data produced, a renewal of the data processing codes was performed. This work presents the NEREA Python package developed for this purpose, with its assumptions and implementation choices. As an application case study, this paper also presents several results of a fission chamber calibration campaign carried out in January and February 2025 in the MARK-III facility of the BR1 reactor,

operated at SCK CEN [6].

II. STRUCTURE OF NEREA

NEREA is an object-oriented Python package available publicly on GitHub [7] which structure is schematically reported in Fig. 1. On the experiment side, a few data classes store the acquired raw data and allow for their preprocessing. Such classes can then be passed to *Experimental* classes, where the actual processing is performed.

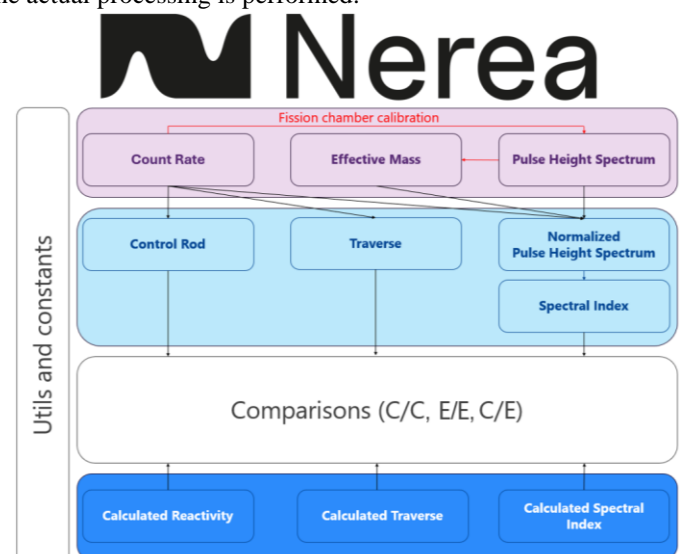


Fig. 1: Schematic of the structure of NEREA: raw-data-interfacing classes (pink), Experimental classes (light blue) and Calculated classes (dark blue).

On the calculation side, NEREA provides the user with *Calculated* classes to then enable for calculation-to-experiment comparisons. NEREA wraps *serpenTools* [8] to interface with the Serpent Monte Carlo code [9]. *Comparison* classes can couple *Experimental* and *Calculated* instances and compare them. Through the whole data handling performed by NEREA (Neutron Energy-integrated Reactor Experiment Analysis), best estimate and uncertainty values are reported one by another in *pandas.DataFrame* [10] instances.

A. Raw data interfacing classes

Each raw data interfacing class comes with one or more

classmethods that allow its creation from common raw data file formats. Attributes identifying the data source (e.g., processed detector, its deposit, ...) are implemented in NEREA and their consistency is checked through the data processing.

1) The *CountRate* class

nerea.CountRate stores time-dependent reaction rate data. *nerea.CountRate* handles time through the *datetime* module of Python.

Methods for timeseries integration and averaging are implemented and used in the processing of spectral index measurements (see section B.2). NEREA also implements methods to find the asymptotic exponential counts in a timeseries, to fit them with an exponential and finally to get the asymptotic period measured by the processed detector.

The *nerea.CountRates* class can import multiple time-dependent data from a single output file and select that with the most counts.

2) The *PulseHeightSpectrum* class

nerea.PulseHeightSpectrum instances can be instantiated reading from GENIE 2000 “TKA” files. Any *nerea.PulseHeightSpectrum* instance retains information on the counted pulses in each voltage channel and implements methods to smooth and re-bin the acquired curve.

Currently, NEREA enables processing of the measured pulse height spectra with the R-channel effective mass methodology presented in [11] [12] [13] [14]. The R-channel methodology bases the identification of the discrimination level for pulse height integration on a detector calibration measurement. Such a calibration shall be performed in a similar neutron flux spectrum as that of its application. The application and calibration spectrum are re-binned to align at the R-channel (i.e., the channel with half-maximum counts right to the maximum itself). The discrimination level for application pulse height spectrum integration is then chosen as a fraction of the R-channel (see section B.1)). Several discrimination levels are considered and the pulse height spectrum integral is calculated from each of them. If the same discrimination approach is used for both calibration and measurement acquisitions, they can be normalized to one another as described in section B.1).

Integration from custom discrimination levels and normalization by fission chamber measured mass (rather than by its effective mass coming from the detector calibration) can also be performed with NEREA. Fig. 2 shows the output of the *plot* method of this class.

The *integrate* method of the *nerea.PulseHeightSpectrum* class integrates the spectrum data from a set of discrimination levels passed by the user. This is then used in the *nerea.SpectralIndex* class for appropriate reaction rate mass normalization.

Integration can be performed taking the raw (default) or smoothed data. In such case, data renormalization can be performed within NEREA, since some smoothing methods may not be conservative of the pulse height spectrum integral.

The *calibrate* method implements the R-channel methodology to compute the fission chamber effective mass processing a calibration pulse height spectrum.

3) The *EffectiveMass* class

Spectral indices measured with fission chambers are processed by NEREA following the effective mass approach reported in [11] [12] [13] [14]. This consists in associating an equivalent

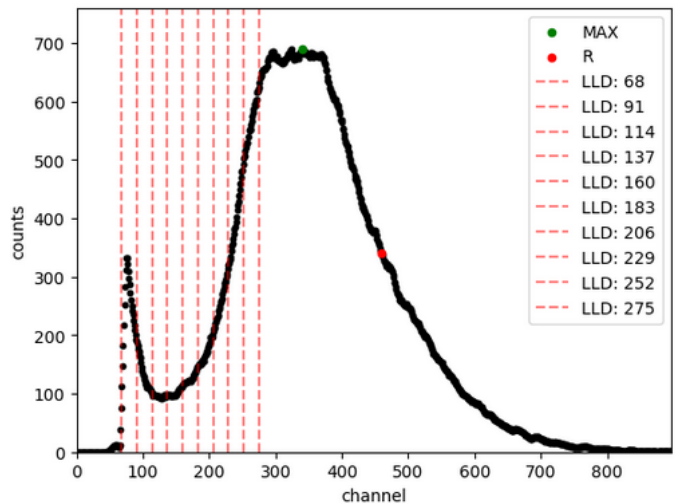


Fig. 2: Pulse height spectrum of fission chamber 19 after 10-channel moving average smoothing. In green the maximum of the spectrum found by NEREA and in red the corresponding R. Dashed lines indicate the discrimination levels (equi-spaced from 0.15 to 0.6R)

fission chamber deposit mass to several discrimination levels from which the measured pulse height spectrum is integrated. More considerations on this approach are reported in section IV.

Fission chamber mass data (either from deposit mass measurement as in [15] [16] [17] or from calibration as in section 3) are stored in the *nerea.EffectiveMass* class. The effective mass is processed and treated by NEREA assuming it has units of mass [μg].

B. Experimental classes for experiment processing

1) The *NormalizedPulseHeightSpectrum* class

The *nerea.NormalizedPulseHeightSpectrum* takes a *nerea.PulseHeightSpectrum* instance and normalizes it to a *nerea.EffectiveMass* and to a *nerea.CountRate* instance. The normalization is performed in steps:

1. Normalization per unit effective mass accounting for the discrimination levels used during calibration data processing;
2. To exclude the measurement noise, a plateau of the pulse height spectrum normalized per unit effective mass as a function of discrimination level is found (1% default tolerance, user customizable);
3. The plateau value is then normalized per unit live measurement time and per average count rate in the power monitor data stored in a *nerea.CountRate* instance.

The processed data can be plotted via the *plot* method (see Fig. 3). This displays information on the effective mass (top left), on the power monitor count rate (top right), on the processed pulse height spectrum (bottom left) and on the calculated reaction rate per unit mass and time (bottom right).

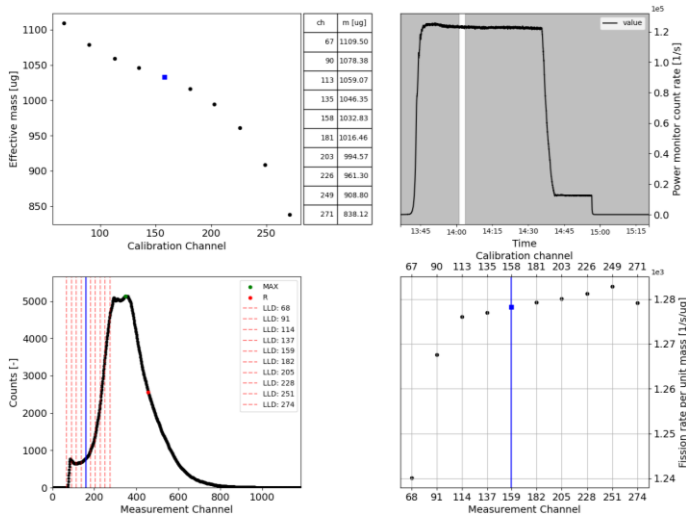


Fig. 3: Summary plot of the NormalizedPulseHeightSpectrum class. Fission chamber effective mass (top left), power monitor (top right) with measurement window in white, pulse height spectrum (bottom left) and fission rate per unit mass (bottom right). The chosen discrimination level is highlighted in blue.

2) The SpectralIndex class

A spectral index is defined as the ratio of two reaction rates. In a neutron flux ϕ , considering fission cross sections σ_f for ^{235}U and ^{238}U , for example, the $^{238}\text{U}/^{235}\text{U}$ spectral index would be defined as

$$SI\left(\frac{^{238}\text{U}}{^{235}\text{U}}\right) = \frac{\int d\tau \phi \sigma_f^{238}}{\int d\tau \phi \sigma_f^{235}},$$

where τ indicates all the dimensions of the phase space considered for integration in the fission chamber deposit. The *nerea.SpectralIndex* instances are initiated with two *nerea.NormalizedPulseHeightSpectrum*: one for the spectral index numerator and one for its denominator. A correction is subtracted from the ratio to account for impurities in the detector deposits as in [13]. Different processing choices (e.g., in terms of pulse height spectrum smoothing and of considered discrimination levels) can be passed to the numerator and denominator separately.

3) The Traverse class

NEREA is suited for processing of reactor experiments beyond spectral index measurements. The *nerea.Traverse* class takes a list of measured *nerea.CountRate*, averages their count rate over the measurement real time and normalizes them to any user-specified element in the list, corresponding to a traverse point.

4) The ControlRodCalibration class

The *nerea.ControlRodCalibration* class is initiated with a critical height and with a *nerea.CountRate* instance where calibration data are stored. Various variants of control rod calibration with positive period method (integral and differential, with and without compensation [18]) are implemented in NEREA.

From the fitted asymptotic period and the In-Hour equation, the *nerea.ControlRodCalibration* allows for calculation of the reactivity associated with each step of the control rod movement. The integral and differential control rod calibration curves are then produced and fitted. Using such fitted curves,

the reactivity worth associated with any control rod movement can be computed with the *get_reactivity_worth* method, passing the initial and final control rod position

C. Calculated and Comparison classes

NEREA interfaces with calculation data with *Calculated* classes. Respectively, the *nerea.CalculatedSpectralIndex* and *nerea.CalculatedTraverse* handle calculated values for the spectral indices and traverse and can be initiated with *classmethods* wrapping *serpentTools*.

NEREA also allows for comparison of calculated (C) to experimental (E), experimental to experimental and calculated to calculated data. This is done via *Comparison* classes, which implement calculation of C/E, E/E and C/C figures of merit. The *nerea.FrameCompare* class enables comparison of any single row *pandas.DataFrame* instances with one another, given that they come with *value* and *uncertainty* columns, which is the minimal format of any output of NEREA.

D. Considerations on the uncertainty propagation in NEREA

NEREA assumes Poisson-distributed random variables for any of the count-related quantities processed in the data-interfacing classes. The uncertainties are then propagated with the first-order momentum propagation equation, assuming no correlations. To allow for a more complete uncertainty analysis, NEREA also returns the portion of output variance associated to any input quantity involved in the processing. Such a variance portion is returned in absolute terms.

For more sophisticated analyses (e.g., due to experiment specificity or processing further than the implemented comparisons), one may require NEREA to output more than the default. This is done with the *long_output* argument of any *processing* method. With this, NEREA outputs also the best estimate of each processing input and the associated variance. One can then propagate uncertainties beyond NEREA's current implementation and explicitly account for correlations.

E. NEREA use example

```
import nerea
import pandas as pd

composition1 = pd.DataFrame({"value": 100, "uncertainty": 0}, index=["U235"])
pm_cal1 = nerea.CountRate.from_ascii("PM1.txt", detector=1, deposit_id="U235")
em1 = nerea.PulseHeightSpectrum.from_formatted_TKA("cal1.TKA").calibrate(
    nerea.constants.KNBS, composition1,
    monitor=pm_cal1, one_group_xs=nerea.constants.XS_FAST)

composition2 = pd.DataFrame({"value": 100, "uncertainty": 0}, index=["U238"])
pm_cal2 = nerea.CountRate.from_ascii("PM2.txt", detector=2, deposit_id="U235")
em2 = nerea.PulseHeightSpectrum.from_formatted_TKA("cal2.TKA").calibrate(
    nerea.constants.KNBS, composition2,
    monitor=pm_cal2, one_group_xs=nerea.constants.XS_FAST)

phs1 = nerea.PulseHeightSpectrum.from_formatted_TKA("meas1.TKA")
phs2 = nerea.PulseHeightSpectrum.from_formatted_TKA("meas2.TKA")
pm1 = nerea.CountRate.from_ascii("M1.txt", detector=1, deposit_id="U235")
pm2 = nerea.CountRate.from_ascii("M2.txt", detector=1, deposit_id="U235")
si = nerea.SpectralIndex(nerea.NormalizedPulseHeightSpectrum(phs1, em1, pm1),
    nerea.NormalizedPulseHeightSpectrum(phs2, em2, pm2))

c = nerea.CalculatedSpectralIndex.from_detectors('model_det0.m', ['d1', 'd2'])
## C/E - 1 [%] can be computed returning pandas.DataFrame
## with 'value' and 'uncertainty' columns
ce = nerea.CoverE(c, si).process(minus_one_percent=True)
```

Code 1: NEREA code to process spectral index C/E.

Code 1 reports a code snippet that can be used to process spectral index C/E results with NEREA. Fission chamber composition aside, most of it consists of data import (pulse height spectra and monitor fission chamber count rates). Formatted file import does not differ substantially from the import functions described in the previous section: the only difference is that the

object metadata are read from the file name directly in this case rather than being passed by the user.

All the processing is handled in the *calibrate* method used to get the fission chamber effective mass, and in the *process* method, that calculates the C/E in the last row. Any processing option (e.g., tolerances, fitting and smoothing parameters, ...) can be passed as a keyword argument to these functions. Iterating scripts analogous to that presented in *Code 1* have been used to process the data and produce the results presented in the next section stopping at the effective mass calculation.

III. FISSION CHAMBER CALIBRATION AT BR1 MARK-III

In January and February 2025, several fission chambers of different designs were calibrated in the MARK-III converter at BR1, where a ^{235}U prompt fission is achieved [6]. The calibration was performed following the R-channel methodology (see section II.A.2). Table I reports a list of the calibrated fission chambers with their design and the associated effective mass at 0.4 R.

TABLE I

EFFECTIVE MASS OF THE CALIBRATED FISSION CHAMBERS (PROCESSED AT 0.4 R). * INDICATES PULSE HEIGHT SPECTRA THAT CAN HARDLY BE USED FOR SPECTRAL INDEX MEASUREMENT (SEE APPENDIX).

ID	Deposit	Effective Mass [μg] (0.4 R)	Design
1	^{232}Th	886*	4 mm
2	^{234}U	250	diameter by
3	^{235}U	87	CEA
4	^{236}U	466	
5	^{238}U	500	
6	^{238}U	430	
7	nat. U	418	
8	nat. U	544	
9	^{237}Np	102	
10	^{239}Pu	22	
11	^{240}Pu	261	
12	^{240}Pu	75	
13	^{242}Pu	191	
14	^{241}Am	125	
15	^{232}Th	2058	8 mm
16	^{238}U	2626	diameter by
17	^{237}Np	364*	CEA
18	^{237}Np	321*	
19	^{235}U	1012	CFUF by
			PHOTONIS
20	^{239}Pu	20	Built-in extension

Each fission chamber was connected to an extension measured to align the fission chamber deposit midplane with the center of the MARK-III converter. The electronic chains of the measurement consisted of a high voltage power supply (+ 310 V), a charge-sensitive pre-amplifier (CANBERRA 2006), a spectroscopic amplifier (CANBERRA AFT 2025) and a multiport module. The data were acquired with the GENIE 2000 spectroscopy software. The calibration coefficient to correlate the power monitor counts with the absolute neutron

fluence in the experiment site was measured through Ni foil irradiation. The Ni foil activation was measured with a HPGe detector with an uncertainty of about 2%. As already reported in [3] [13] this is a major contributor to the effective mass uncertainty. Calibrating fission chamber in a single campaign (i.e., with a single calibration factor) makes the processed spectral index independent of the calibration factor itself, as it appears at the spectral index numerator and denominator.

A. A chronological overview

Fission chambers calibration data of campaigns at BR1 MARK-III since 2010. These campaigns featured calibration of some of the fission chambers reported in this study, as reported in Table II. No impurity correction was found in the calibration files earlier than 2025: the uncorrected effective masses are then reported for those.

As it is visible in Table II, the measured effective mass changes with time. This can be attributed to several causes, among which, fluctuations in the calibration factor, slight changes in the fission chamber design characteristics (e.g., changes in filling gas pressure), changes in the used electronics and differences in the processing. A similar time-wise comparison is also reported in [11]. Fission chamber calibration also calibrates for the data processing choices, to some extent. This means that the observed differences in effective mass could reflect similarly on the measured count rate resulting in limited changes to the final spectral index value. The recommendation we give is to reperform detector calibration before each experimental campaign to maximize consistency.

TABLE II
EFFECTIVE MASS [MG] MEASURED OVER THE YEARS AT 0.4 R WITH CALIBRATIONS AT BR1 MARK-III. *: PULSE HEIGHT SPECTRA HARDLY USED FOR SPECTRAL INDEX MEASUREMENT (SEE APPENDIX); **: REPORTED AT 0.5 R BECAUSE OF HIGH NOISE.

ID	Year					
	2010	2013	2016	2020	2022	2025
	Effective mass [μg]					
1						886*
2	206				214	250
3		84	86			87
4			442		449	466
5		488	488			500
6			419			430
7			391			418
8		528	538			544
9		101				102
10			21			22
11	232**					238**
12		72				75
13	170					191
14						125
15						2058
16						2626
17						364*
18		100*				321*
19			1217			1012
20			21			20

B. Impact of processing choices

From now on, we consider fission chamber 19 as a reference case study, which pulse height spectrum is shown in Fig. 2. The default implementation of NEREA features moving average (MA) smoothing over 10 channels. Fig. 4 shows a comparison with another suitable smoothing option: exponential weighted mean (EWM) [19] (span=10). Visually, the resulting spectra differ very little. The pulse height spectrum can also be fitted to find its maximum and R channel. NEREA implements the possibility to fit the spectrum from any channel before the maximum to the first zero on its right. The pulse height spectrum fitted with a polynomial of order 5 is shown in Fig. 4 (FIT). Before channel 270, the raw data are concatenated to the fitted ones.

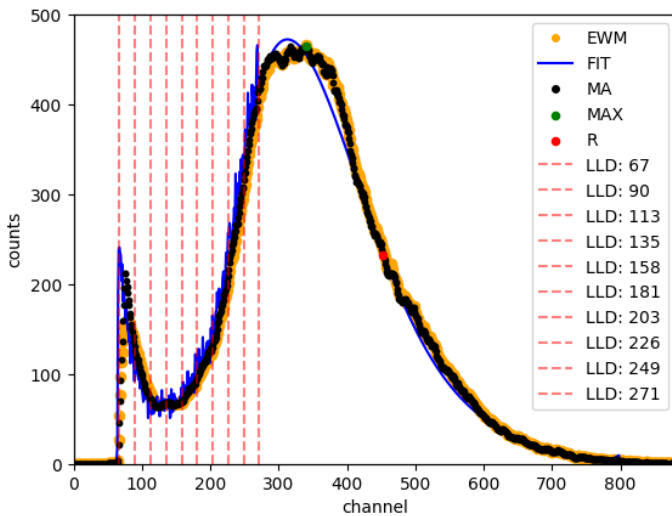


Fig. 4: Comparison of the pulse height spectrum of fission chamber 19 smoothed with moving average (black), exponential weighted mean (orange) and the fitted one (blue).

TABLE III
EFFECTIVE MASS OF FISSION CHAMBER 19. PULSE HEIGHT SPECTRUM PROCESSED WITH DIFFERENT SMOOTHING OPTIONS. RESULTS REPORTED AT 0.4 AND 0.6 R.

Smoothing method	Window, Span or Order	Effective Mass [μg] (0.4 R)	Effective Mass [μg] (0.6 R)
None	n.a.	1017.7	854.2
Moving Average	5	1013.5	825.4
	10	1012.5	821.3
	-5 to +5	1014.6	829.4
	15	1011.6	821.3
Exponential weighted mean	5	1015.4	833.1
	10	1012.5	821.3
	15	1011.6	814.0
Mean		1013.5	827.5
Absolute standard deviation		2.3	12.2

From now on, all the pulse height spectrum integrations are performed finding the R-channel of the smoothed spectrum and integrating the raw data from the so-found discrimination levels. The effective mass emerges to be quite insensitive to the smoothing options (see Table III). A larger standard deviation is found for effective mass values at 0.6 R than at 0.4 R. The

effective mass obtained fitting the pulse height spectrum is rather sensitive to the fit choice. Even if the fitted integral counts in the spectrum differ by about 0.1 % from the raw ones, the discrimination levels considered to determine the effective mass easily fall in the region of transition between raw data and fitted data (see Fig. 4). The effective mass is then quite sensitive to the choice of the channel where the fit starts. Low polynomial orders or low channels to start the fit from can distort the spectrum significantly. For this reason, the results of the pulse height spectrum fit are not reported in Table III.

Discrimination level 0.4 R (channel 181) is in the pulse height spectrum valley between noise and fission signal. Changing the discrimination level by one channel there impacts the spectrum integral less than around 0.6 R (channel 271), which is much closer to the pulse height spectrum maximum. The larger sensitivity of the integrated PHS to the discrimination level at 0.6 R than at 0.4 R explains the larger standard deviation reported in Table III.

These results depend on the measured pulse height spectrum (i.e., on the measured fission chamber). In general, the more distinctly the fission signal stands out from the measurement noise (i.e., the higher the peak-to-valley ratio) and the broader the valley region (relative to the overall spectrum), the less impact the smoothing method will have on the results.

C. Comparison of successive acquisitions

Fig. 5 shows the pulse height spectra acquired in three successive acquisitions with fission chamber 19. Nor the reactor nor the electronics were shut down between the acquisitions and no change in the fission chamber position or acquisition setting happened.

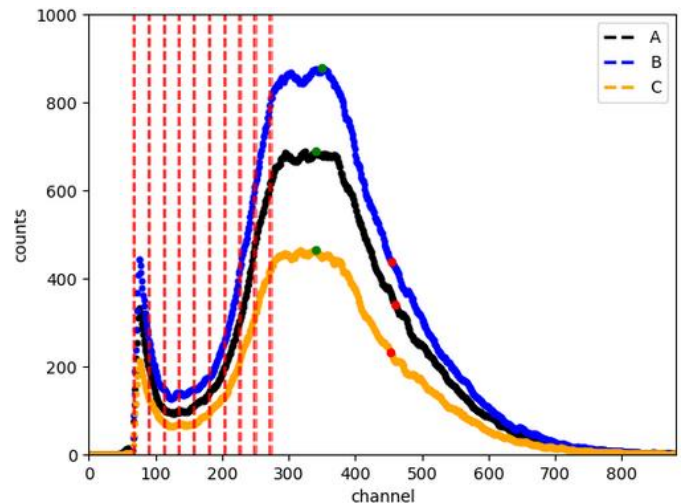


Fig. 5: Pulse height spectra of fission chamber 19 acquired in three successive acquisitions (A, B, C) after smoothing.

Table IV reports the effective mass measured in each of the three acquisitions for each of 10 discrimination levels computed as a fraction of the R channel (note that this requires rounding by NEREA). The standard deviation of the measured effective masses (A, B and C) ranges from about 0.5%, at 0.4 R, to 1% at 0.6 R. The spread of the measurements is not fully explained by the uncertainty propagated from the fission chamber counting uncertainty (0.2% - 0.3%) and the power monitor

counting uncertainty (< 0.05% - the difference in power monitor count rate between the three acquisition goes up to 0.2%).

TABLE IV
EFFECTIVE MASS [MG] OF FISSION CHAMBER 19 AFTER ACQUISITIONS A, B AND C FOR EACH OF 10 DISCRIMINATION LEVELS FRACTIONAL OF THE R-CHANNEL.

R	A	B	C
ch	Effective Mass [μg]	ch Effective Mass [μg]	ch Effective Mass [μg]
0.15	68 1095.6	68 1104.8	67 1107.8
0.20	91 1061.6	90 1069.7	90 1073.5
0.25	114 1045.9	113 1051.9	113 1056.3
0.30	137 1033.1	136 1037.9	135 1043.7
0.35	160 1019.4	158 1023.6	158 1030.1
0.40	183 1001.7	181 1006.2	181 1012.5
0.45	206 977.7	204 982.0	203 989.4
0.50	229 940.8	227 944.0	226 952.9
0.55	252 882.8	249 889.6	249 896.1
0.60	275 804.9	272 811.9	271 821.3

Table V shows how the effective mass results of acquisitions A, B and C compare. Sensitivity to the pulse height spectrum processing choice is here observed. This is particularly relevant fitting the spectrum as the number of channels fitted before the maximum and the polynomial order (first and second value in parenthesis in Table V) give rather oscillating results. On the other hand, integrating from the same channel discriminator (independently of R), the spread of the measured effective masses reduces, as shown in Table V.

TABLE V
DIFFERENCE IN EFFECTIVE MASS INTEGRATING FROM R-CHANNEL-RELATED AND ABSOLUTE DISCRIMINATION LEVELS.

Spectrum processing and discriminator criterion	Effective Mass B / Effective Mass A - 1	Effective Mass C / Effective Mass A - 1
0.6 R MA	0.87 %	2.04 %
EWM	1.36 %	2.04 %
FIT	1.34 %	2.51 %
(100, 7)		
FIT	- 0.07 %	2.05 %
(150, 7)		
FIT	0.38 %	1.60 %
(150, 9)		
Ch. Raw data	- 0.53 %	0.19 %
275		

While one could argue that the observed effective mass standard deviation between acquisitions A, B and C is of a similar order of magnitude as the counting uncertainties until channel 0.4 R, after that this is not the case.

We attribute the difference in measured effective mass between acquisitions A, B and C to the R channel calibration methodology. As it was already observed, the sensitivity of the pulse height spectrum integral to the associated discrimination level (i.e., to the integration lower bound) increases moving

towards the pulse height spectrum maximum. When the calibration methodology results in different discriminators (see Table V), the measured effective mass is affected in a way that is not reflected by any of the considered uncertainties.

IV. DISCUSSION ON THE R-CHANNEL CALIBRATION METHODOLOGY

Adoption of the R-channel to compare pulse height spectra comes with many advantages and some shortcomings. Fundamentally, the R-channel methodology consists of a pulse height spectrum re-binning (and consequent reshaping). When both the calibration and the measurement spectrum are rebinned to the same fractions of R, one can look at them in the same domain and compare when the pulse height spectrum integrals are proportional with a constant factor. This has mostly two advantages:

- the processing methodology is inherently rather independent of the acquisition chain amplification gain, as the R re-binning corrects for it;
- when the pulse height spectrum valley is well identified, the pulse height spectrum integral ratio is rather stable in that region.

These two advantages should not be underestimated: the experimental procedures and practices are simplified by the R-channel methodology, as there is no need to calibrate each fission chamber with each amplifier – pre-amplifier couple. Moreover, one could argue that a fission chamber with poor peak-to-valley ratio is not well suited for spectral index measurements and should rather be used for bare count rate comparisons as in fission traverse measurements.

On the other hand, it was observed that the R-channel methodology suffers from uncertainties associated with the determination of the R channel as this does not only require good statistics in the pulse height spectrum integral, but also in the individual channels of the maximum and of R. This can be quite hard to achieve with low-counting fission chambers and gives a visible effect also on the effective mass measured with fission chamber 19 (about 10⁵ counts after 0.6 R).

While alternatives to the R-channel calibration methodology are being considered, the recommendation we give is to perform multiple calibration and measurement acquisitions, possibly varying the electronics. Then the spread of the measured effective masses could be taken as an agnostic estimate of the calibration uncertainty, accounting not only for counting statistics uncertainty, but also for the inherent uncertainties associated with the calibration methodology.

V. CONCLUSIONS

The NEREA Python package for integral reactor experiments is presented in this paper. The living version of the code is available on GitHub [7] and the code is available on PyPi. NEREA provides a user with classes to process fission chamber acquisitions for spectral index, fission rate traverse and control rod calibration measurements.

To show the versatility of NEREA and the degree of processing control it allows, the results of a fission chamber calibration campaign performed at BR1 are presented. The results highlight

a benefit in fission chamber recalibration before each experimental campaign. The little-to-no impact of the pulse height spectrum smoothing choices implemented in NEREA on the fission chamber effective mass results was shown for a fission chamber with good signal-to-noise discrimination. An analysis of the R-channel calibration methodology implemented in NEREA is also presented. This shows that, besides its advantages, such a methodology comes with uncertainties that are not fully reflected by the observed counting statistics. While alternative options are under investigation, it is recommended to perform multiple independent calibration measurements and derive a calibration uncertainty after statistical considerations on the repeated measurement set.

ACKNOWLEDGMENT

This work was supported financially by Association Vinçotte Nuclear (AVN).

REFERENCES

- [1] A. Krása et al, "Comparative study on neutron data in integral experiments of MYRRHA mockup critical cores in the VENUS-F reactor", *EPJ Web of Conferences*, 2017.
- [2] A. Kochetkov, et al., "Integral experiments in the VENUS-F reactor", *Annals of Nuclear Energy*, 2021.
- [3] F. Grimaldi et al., "The CoRREx neutron spectrum filtering campaign at VENUS-F for calculation-to-experiment discrepancy interpretation", *Annals of Nuclear Energy*, 2025.
- [4] A. Kochetkov et al., "The Lead-Based VENUS-F Facility: Status of the FREYA Project", *EPJ Web of Conferences*, 2016.
- [5] E. Fridman et al., "Modeling of FREYA fast critical experiments with the Serpent Monte Carlo code", *Annals of Nuclear Energy*, 2017.
- [6] J. Wagemans et al., "The ²³⁵U Prompt Fission Neutron Spectrum in the BR1 Reactor at SCK•CEN", *EPJ Web of Conferences*, 2016.
- [7] F. Grimaldi, "NEREA GitHub", 2025. [Online]. Available: <https://github.com/GrimFe/NEREA>. [Accessed 2025].
- [8] A. Johnson et al., "serpentTools: A Python Package for Expediting Analysis with Serpent", *Nuclear Science and Engineering*, 2019.
- [9] J. Leppänen et al., "The Serpent Monte Carlo code: Status, development and applications in 2013", *Annals of Nuclear Energy*, 2015.
- [10] W. Mckinney, "Data Structures for Statistical Computing in Python", *Proceedings of the 9th Python in Science Conference*, 2010.
- [11] N. Blanc De Lanaute, "Développement et optimisation de méthodes de mesures neutroniques par chambre à fission auprès de réacteurs expérimentaux. Maîtrise , traitement et réduction des incertitudes", PhD thesis, 2012.
- [12] B. Geslot et al., "Method to Calibrate Fission Chambers in Campbell Mode", *International Conference on Advancements in Nuclear Instrumentation, Measurement Methods and their Applications (ANIMMA)*, 2011.
- [13] V. Lamirand et al., "Miniature fission chambers calibration in pulse mode: interlaboratory comparison at the SCK•CEN BR1 and CEA CALIBAN reactors" *International Conference on Advancements in Nuclear Instrumentation, Measurement Methods and their Applications (ANIMMA)*, 2013.
- [14] A. Kochetkov et al., "Calibration of Large Photonis Fission Chambers in Standard Neutron Fields of the BR1 Reactor", *IEEE TNS*, 2022.
- [15] V. Doolin and V. Mozhaev, "Method of fission chamber efficiency determination using uncorrelated neutron background measurement", *Nuclear Instruments and Methods*, 1972.
- [16] J. A. Grundl et al., "Measurement of absolute fission rates," *Nuclear Technology*, 1975.
- [17] M. Obu, "Preparation and Characteristics of Fission Chambers", 1981.
- [18] T. Misawa et al., *Nuclear Reactor Physics Experiments*, Kyoto University Press, 2010.
- [19] pandas, "pandas.Series.ewm," [Online]. Available: <https://pandas.pydata.org/docs/reference/api/pandas.Series.ewm.html>. [Accessed 2025].
- [20] J.-P. Hudelot, " Développement amelioration et calibration des mesures de taux de reaction neutroniques: elaboration d'une base de techniques standards", PhD thesis, 1998.

APPENDIX

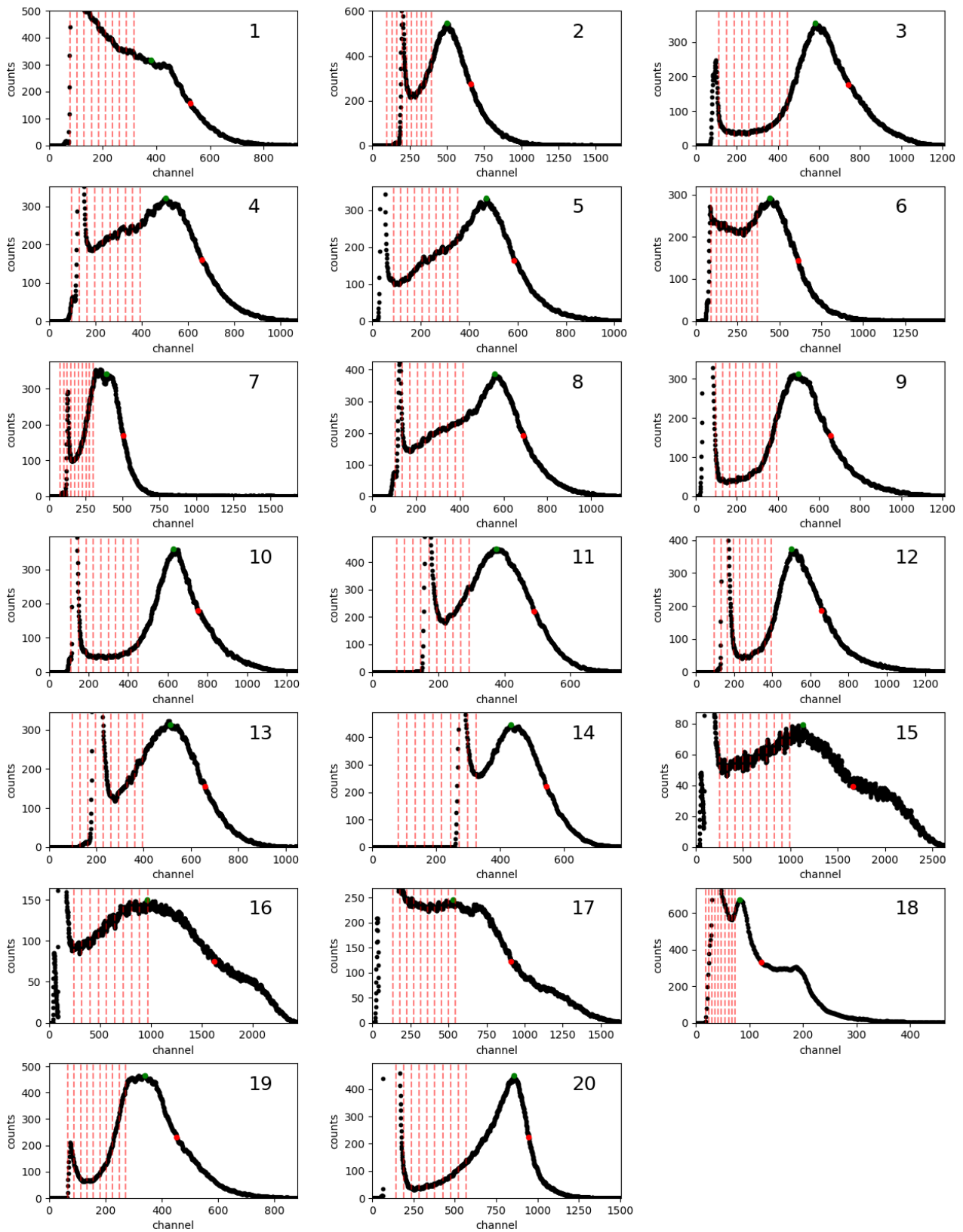


Fig. 6: Measured calibration pulse height spectra. The maximum (green) and R (red) points are highlighted. Red dashed lines indicate 10 discrimination levels as a fraction of the R-channel.