# Open-source software for material accountancy analysis

Proceedings of the INMM Annual Meeting
May 22-26, 2023

**Nathan Shoman***
Sandia National Laboratories†
nshoman@sandia.gov

**Pat Moosir**
Sandia National Laboratories†
mhiggin@sandia.gov

### Abstract

Material accountancy is a cornerstone of safeguards and security for nuclear fuel cycle facilities. However, the underlying methodologies used can often be difficult to find and interpret. Written in Python, the Material Performance Indicator Toolkit (MAPIT) is the first free and open-source toolkit for performing material accountancy for bulk nuclear facilities. It provides both a beginner-friendly graphical user interface and expert-focused application programming interface; both of which contains many common statistical methods used for material accountancy. This paper details the initial capabilities of MAPIT, design philosophies used to implement various statistical methodologies, efforts to generate reproducible results, and design decisions made to improve performance.

## 1 Introduction

Material accountancy is an important cornerstone of regulatory guidelines for nuclear power plants that ensures nuclear material is present and accounted for. However, the methods to achieve regulatory targets can vary from organization to organization, rely on significant levels of intuitional knowledge, and can be unapproachable to newcomers; to both industrial practitioners and academic researchers alike. We propose to solve some of these roadblocks via the Material Performance Indicator Toolkit (MAPIT). Written entirely in Python, MAPIT is the first open-source material accountancy toolkit available to everyone under the GPL-3.0 license model. The target audience for MAPIT ranges from undergraduate level engineers to seasoned experts with decades of experience. MAPIT has both a graphical user interface (GUI) and application program interface (API). Our code is freely available on GitHub[3]. We welcome feedback and comments from users and the nuclear community at large.

MAPIT, an open-source material accountancy toolkit, currently focuses on bulk material accountancy, MAPIT implements common statistical routines used by both the U.S. Nuclear Regulatory Agency (NRC) and the International Atomic Energy Agency (IAEA). Currently implemented features include:

- **Automated error propagation** of all measurement locations used to form a material balance. Efficient NumPy [1] routines are used to simulate errors for many measurement locations collected over large spans of time.
- **Implementations of common statistical tests** including material unaccounted for (MUF, sometimes called inventory difference (ID) or material balance (MB)), $\sigma_{\text{MUF}}$ (sometimes called SEID), active inventory, the standardized independent transformed MUF (SITMUF), Page's trend test, and looking to add more functionality in the future.

*Corresponding author

†Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. SAND2023-02945C

[3] https://github.com/sandialabs/MAPIT

- **Statistical test optimization**, which is currently only directly supported in the GUI, can be used to set thresholds and define performance targets.
- **Robust input/output (I/O)** capabilities in MAPIT allow users to either use their own data within MAPIT or to use an included exemplar. MAPIT supports several input formats including `.csv`, `.mat`, and `.npy` with several options to export results for further analysis.
- **Intuitive GUI** has been designed as a visual aid to provide a deeper understanding of the impact of measurement system design on material accountancy performance.
- **Rich API** is included for users familiar with Python. The same functionality of the GUI is available in well documented modules that can be integrated with other analytical pipelines, or for large-scale analysis.
- **Comprehensive documentation** for the API as well as step-by-step exercises are available alongside the MAPIT codebase on GitHub. In the future, we will publish a detailed users guide with supplementary documentation on the theory of the different methodologies used within MAPIT.

## 2 Background

There have been a variety of different efforts to control nuclear material dating back to 1954 with Eisenhower's Atoms for Peace program. Later, other agreements such as the formation of the Non-Proliferation Treaty in 1968 created an increased importance on nuclear material control. One of the more prominent early efforts in methods for material control can be found in [2]. This effort mostly centered around MUF [3] which had several shortcomings [4]. Later efforts centered around more complex statistical techniques such as material balance sequences [5, 6], SITMUF [7], GEMUF [8], and Page's trend test on SITMUF [9–12], which are concepts still widely used today. More contemporary literature has focused on efforts to leverage process data as a basis for near-real-time-accounting [13–19]. Other efforts have investigated data-driven strategies to develop alternative strategies to material accountancy all together [20–28]. Decades of research into material control has lead to important insights, but a relatively large volume of literature that can be difficult to compare and digest without extensive study. MAPIT was designed to improving access and understanding of material accountancy statistical methods by providing a transparent framework to carry out analyses.

## 3 Principles

MAPIT was designed with several different principles in mind as a means to improve understanding of statistical analyses for material accountancy:

- **Approachable:** Methods used in material accountancy are often contained in papers written by statisticians for statisticians. However, many stakeholders in the material accountancy space often do not have deep statistical backgrounds. These stakeholders still need understanding of the underlying performance metrics used in material accountancy. For example, facility designers and measurement experts would still benefit from understanding the impact of different measurement technologies on the performance of the material accountancy system. MAPIT was designed with a technically diverse set of end users in mind. Common statistical tests are implemented in a common programming language, Python, and we have plans to release a theory guide that ties together specific code blocks to theoretical concepts and historical material accountancy literature. MAPIT improves the accessibility of material control analysis while maintaining all of the statistical rigor described in the literature. This is accomplished through an intuitive GUI that is coupled with several step-by-step exercises.
- **Transparent:** Statistical approaches for material accountancy are not often expressed in computationally efficient ways, which can make it difficult to determine the best way to implement these equations efficiently. MAPIT draws upon past efforts in the modeling and simulation space to express routines for these statistical approaches. In-line code comments that tie to the original literature makes a strong connection between theory and practice.
- **Contrastive:** It is often difficult to determine which method or strategy to use when designing a material accountancy system. MAPIT was designed to aid in this task by facilitating contrastive analysis and "what-if" tools. One such example in MAPIT is a per-location breakdown of error contribution to the material balance that can be used to optimize sensor placement and compare against other best practices (shown in Fig. 2 below). I/O allow users to save and load problem configurations to more efficiently compare different material accountancy systems.
- **Efficient:** Material accountancy analysis often requires 100s to 1000s of different realizations to generate statistically robust results. This can create a particularly large compute and memory requirement. MAPIT

was designed in Python, a language commonly used in large-scale data analysis tasks. Efficient routines from NumPy [1] and SciPy [29] are used and combined with a flexible specification for high performance applications. For example, matrix operations from NumPy are used when possible which offers performance orders of magnitude over simple loops. Custom routines cache expensive variables where possible for potential reuse for subsequent calculations. When used in the GUI format, MAPIT has intuitive sliders to enable multi core processing to increase performance for systems with more resources [4].

- **Reproducible:** Simulated measurement error can make it difficult to reproduce results unless large numbers of iterations are used. Use of the MAPIT API allows users to fix the NumPy random seed for reproducible results. Modern software practices (i.e., version control, unit testing, etc.) has also been implemented to further increase confident in MAPIT calculations.

## 4 Graphical User Interface

The underlying statistics and associated setup of material accountancy calculations can be difficult to understand, particularly for stakeholders without a statistical background. A key principle behind MAPIT is approachability. Material accountancy analysis performed in MAPIT should be statistically valid, but also approachable to those unfamiliar with material accountancy. A Graphical User Interface (GUI) for MAPIT, shown in Figure 1, is used to improve approachability.

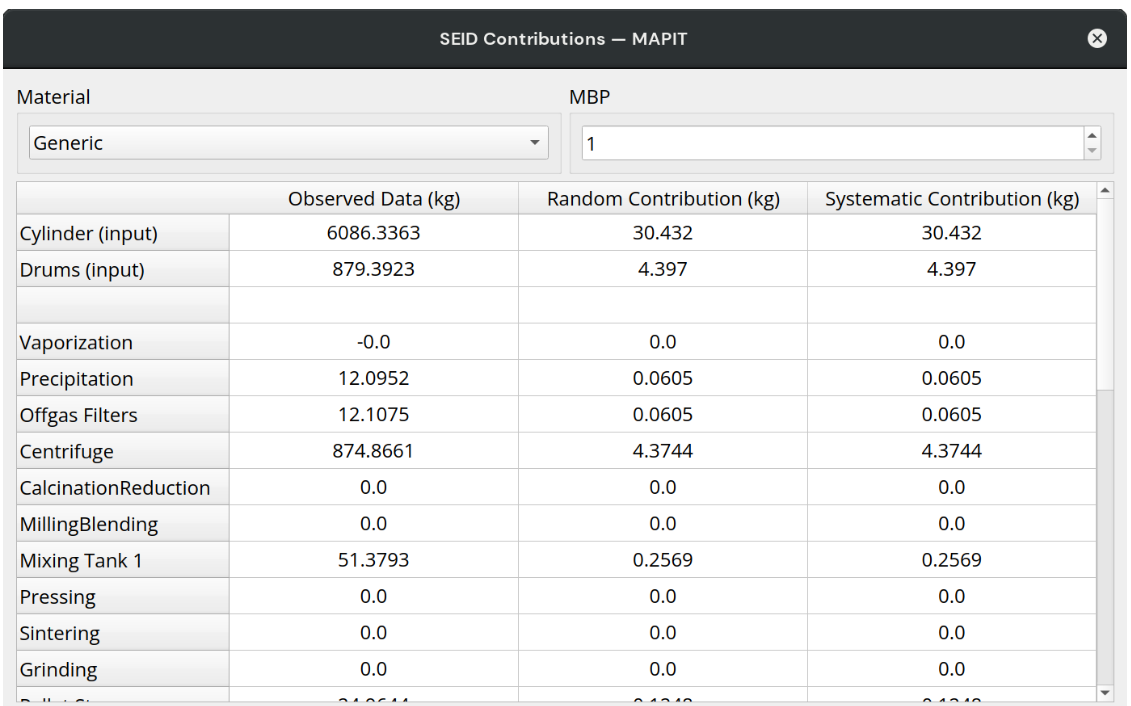

Figure 1: Overview of the MAPIT GUI

[4]Feature is planned to be integrated in to MAPIT in late 2023

The single main MAPIT window contains all the elements required to perform material accountancy analysis. An exemplar dataset generated by SNL based on STR-150 [30] is included to facilitate exploratory analysis. Alternatively, MAPIT's flexible import options allows for analysis to be performed on local datasets. The workflow in the GUI consists of a few steps:

- **Dataset selection:** In the first upper left area of the GUI, users chose to either use the exemplar dataset or to import their own data. An import wizard provided step-by-step prompts for additional data used for MAPIT.
- **Analysis:** Currently available statistical tests are shown in the next area of the GUI. Simple checkboxes enable or disable calculation of certain tests.
- **Problem setup:** This area requires a few simple parameters to setup the statistical tests described in the previous section:
  - Material balance period (MBP): The amount of time between material balance closures
  - Iterations: The number of statistical realizations; only used if simulating measurement error
  - Analysis element / isotope: The element to perform the material balance on
  - Temporal Offset: In some cases, it might be useful to skip some time before performing the first material balance, so this option is provided
  - Set Errors: If using simulated data, it is often useful to add simulated errors to each statistical realization

Following execution of MAPIT using the "Run" button, several graphical options are available. All the statistical tests that were selected during problem setup are available to plot using the controls on the right side of the GUI. Further, the GUI includes features like threshold calculations and $\sigma_{\text{MUF}}$/SEID contributions (shown in Figure 2 below).




| | Observed Data (kg) | Random Contribution (kg) | Systematic Contribution (kg) |
|---|---|---|---|
| Cylinder (input) | 6086.3363 | 30.432 | 30.432 |
| Drums (input) | 879.3923 | 4.397 | 4.397 |
| | | | |
| Vaporization | -0.0 | 0.0 | 0.0 |
| Precipitation | 12.0952 | 0.0605 | 0.0605 |
| Offgas Filters | 12.1075 | 0.0605 | 0.0605 |
| Centrifuge | 874.8661 | 4.3744 | 4.3744 |
| CalcinationReduction | 0.0 | 0.0 | 0.0 |
| MillingBlending | 0.0 | 0.0 | 0.0 |
| Mixing Tank 1 | 51.3793 | 0.2569 | 0.2569 |
| Pressing | 0.0 | 0.0 | 0.0 |
| Sintering | 0.0 | 0.0 | 0.0 |
| Grinding | 0.0 | 0.0 | 0.0 |

Figure 2: Example $\sigma$MUF/SEID contribution table

Using the tools in the MAPIT GUI, it is possible to perform "what-if" scenarios by changing measurement errors and comparing the impact on statistical test performance through the threshold functionality. Results generated using the MAPIT GUI can be exported to a CSV format for secondary analysis using the menu bar near the top of the GUI and figures shown in the plotting window can be saved in a variety of formats through the floppy icon near the bottom of the plot window. Step-by-step tutorials for using the MAPIT GUI can be found on the GitHub repository at `https://sandialabs.github.io/MAPIT/guided_exercises.html`.

## 5 Application Program Interface

While the MAPIT GUI is an important tool for visualizing key aspects of material accountancy, some workflows need integration with other tools and systems. The MAPIT API is designed to utilize the various functionality through a programmatic interface. All of the key functions mentioned in the GUI are also available through the API. The material accountancy workflow can be divided into a few key steps. MAPIT requires some assumptions about the format of the underlying data and provides some functions to help format the data. Specific details can be found in the online documentation and in our API example found here `https://github.com/sandialabs/MAPIT/blob/master/API_examples/API_basic.ipynb`.

The most common first step in a material accountancy analysis using simulated data is the application of simulated error. Using the MAPIT API, this can be achieved in a single line of code for each material balance component, shown in Code 1.

```
from MAPIT.core import Preprocessing

inputAppliedError = Preprocessing.SimErrors(
                            input = processedInput,
                            ErrorMatrix =  inputErrorMatrix,
                            iterations = IT)

inventoryAppliedError = Preprocessing.SimErrors(
                            input = processedInventory,
                            ErrorMatrix =  inventoryErrorMatrix,
                            iterations = IT)


outputAppliedError = Preprocessing.SimErrors(
                            input = processedOutput,
                            ErrorMatrix =  outputErrorMatrix,
                            iterations = IT)
```

Code 1: MAPIT API for applying simulated measurement error

After the data has been formatted and has simulated errors applied (if applicable), the next step is to apply the various statistical tests and concepts. All of the tests and quantities in MAPIT have a single function required to generate results. For example, the SITMUF function shown below in Code 2, requires the material balance components and measurement relative standard deviations used to generate the covariance matrix, and ID/MUF. These components can be provided from an external data source, or can be calculated by the MAPIT API in a previous step.

```
from MAPIT.core import StatsTests

SITMUF = StatsTests.SITMUF(inputAppliedError = inputAppliedError,
                                inventoryAppliedError = inventoryAppliedError,
                                outputAppliedError = outputAppliedError,
                                processedInputTimes = processedInputTimes,
                                processedInventoryTimes = processedInventoryTimes,
                                processedOutputTimes = processedOutputTimes,
                                ErrorMatrix = TotalErrorMatrix,
                                MBP = mbaTime,
                                MUF = MUF)
```

Code 2: MAPIT API for calculating SITMUF

The MAPIT API is also useful for large-scale analysis. Performance analysis for material accountancy systems often require large quantities of iterations to generate good statistics. The API handles this by simply adjusting the iteration parameter (where applicable). Future updates to MAPIT later in 2023 will include multi-threadding to further improve the speed of this calculation. Full documentation for the API can be found at `https://sandialabs.github.io/MAPIT/apidocs.html`.

## 6 Conclusion and Future Work

MAPIT is an open-source toolkit designed for the safeguards and security community that was designed with a range of end-user expertise in mind. With a focus on approachability, transparency, efficiency, and reproducibility, we hope that MAPIT will further strengthen the community by having a readily available material accountancy analysis platform. Our near term goals for MAPIT include performance improvements through multi-threadded calculations, documentation tying theory and literature to our code, implementing additional statistical tests, and increased functionality within the GUI.

## 7 Acknowledgements

This work was funded by Department Of Energy Office of Nuclear Energy (DOE-NE) and the Department of Energy's National Nuclear Security Administration (DOE-NNSA). Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

## References

[1] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, pp. 357–362, Sept. 2020.

[2] J. L. Jaech, "Statistical methods in nuclear material control," 1 1973.

[3] A. Goldman, R. Picard, and J. Shipley, "Statistical methods for nuclear materials safeguards: An overview," *Technometrics*, vol. 24, no. 4, pp. 267–275, 1982.

[4] T. Speed and D. Culpin, "The role of statistics in nuclear materials accounting: issues and problems," *Journal of the Royal Statistical Society: Series A (General)*, vol. 149, no. 4, pp. 281–300, 1986.

[5] R. Avenhaus and J. Jaech, "On subdividing material balances in time and/or space," *Journal of Nuclear Materials Management*, vol. 10, 1981.

[6] R. R. Picard, "Sequential analysis of material balances," *Journal of Nuclear Materials Management*, vol. 15, 1987.

[7] B. Jones, "Near real time material accountancy using SITMUF and a joint page's test: comparison with MUF and CUMUF tests," 1988.

[8] R. Seifert, *The GEMUF test: A new sequential test for detecting loss of material in a sequence of accounting periods*. IAEA, 1987.

[9] B. Jones, "Calculation of diversion detection using the sitmuf sequence and page's test: application to evaluation of facility designs," in *Proceedings of the 7th ESARDA Symposium on Safeguards and Nuclear Material Management*, 1985.

[10] B. Jones, "Calculation of diversion detection using the sitmuf sequence and page's test," in *Nuclear safeguards technology 1986*, 1987.

[11] B. Jones, "Comparison of near real time materials accountancy using sitmuf and page's test with conventional accountancy," in *Proceedings of the 9th ESARDA Symposium on Safeguards and Nuclear Material Management*, 1987.

[12] B. Jones, "Near real time materials accountancy using sitmuf and a joint pages test: improvement of the test," *ESARDA Bulletin*, vol. 16, pp. 13–19, 1989.

[13] T. Burr and M. S. Hamada, "Bayesian updating of the material balances covariance matrices with training data," *International Journal of Prognostics and Health Management*, vol. 5, 2014.

[14] T. Burr, C. Norman, C. Orton, and S. Zykov, "Non-destructive assay in the role of consistency checking," *International Workshop on Numerical Modelling of NDA Instrumentation and Methods for Nuclear Safeguards*, 2018.

[15] T. L. Burr, C. A. Coulter, J. Howell, and L. E. Wangen, "Solution monitoring: quantitative and qualitative benefits to nuclear safeguards," *Journal of Nuclear Science and Technology*, vol. 40, no. 4, pp. 256–263, 2003.

[16] T. Burr, M. S. Hamada, L. Ticknor, and J. Sprinkle, "Hybrid statistical testing for nuclear material accounting data and/or process monitoring data in nuclear safeguards," *Energies*, vol. 8, no. 1, pp. 501–528, 2015.

[17] T. Burr, M. Hamada, M. Skurikhin, and B. Weaver, "Pattern recognition options to combine process monitoring and material accounting data in nuclear safeguards," *Statistics Research Letters*, vol. 1, no. 1, pp. 6–31, 2012.

[18] T. L. BURR, C. A. COULTER, J. HOWELL, and L. E. WANGEN, "Solution monitoring: Quantitative and qualitative benefits to nuclear safeguards," *Journal of Nuclear Science and Technology*, vol. 40, no. 4, pp. 256–263, 2003.

[19] T. Burr, M. S. Hamada, L. Ticknor, and J. Sprinkle, "Hybrid statistical testing for nuclear material accounting data and/or process monitoring data in nuclear safeguards," *Energies (Basel)*, vol. 8, 1 2015.

[20] C. R. Orton, S. A. Bryan, J. M. Schwantes, T. G. Levitskaia, C. G. Fraga, and S. M. Peper, "Advanced process monitoring techniques for safeguarding reprocessing facilities," 11 2010.

[21] N. Shoman, J. Coble, and D. Meier, "Experimental performance of the multi isotop process monitor," in *Transactions of the American Nuclear Society*, vol. 113, pp. 483–485, 2015.

[22] J. B. Coble, S. E. Skutnik, S. N. Gilliam, M. Cooper, and J. Mitchell, "Integrating data sources for improved safeguards and accountancy of electrochemical fuel reprocessing systems. final report," 3 2020.

[23] H. Garcia, W.-C. Lin, and R. Carlson, "Evaluating safeguards benefits of process monitoring as compared with nuclear material accountancy," 7 2014.

[24] H. E. Garcia, "Integrated process monitoring based on systems of sensors for enhanced nuclear safeguards sensitivity and robustness," 7 2014.

[25] N. Shoman and T. Burr, "Impact of safeguards measurement errors on deep neural networks.," 8 2021.

[26] N. Shoman, B. Cipiti, T. Grimes, B. Wilson, and R. Gladen, "Insights from applied machine learning for safeguarding a purex reprocessing facility.," 8 2021.

[27] R. Gladen, T. Grimes, B. Wilson, J. Dermigny, B. B. Cipiti, and N. Shoman, "Neural assessment of non-destructive assay for material accountancy," in *Annual Meeting Proceedings of the Institute of Nuclear Material Management*, 2021.

[28] N. Shoman and P. Honnold, "Limitations for data-driven safeguards at enrichment facilities," in *Annual Meeting Proceedings of the Institute of Nuclear Material Management*, 2022.

[29] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[30] R. J. Jones, E. V. Weinstock, and W. R. Kane, "Detailed description of an ssac at the facility level for a low-enriched uranium conversion and fuel fabrication facility," tech. rep., International Atomic Energy Agency, 1984.