Summarize the main points of the document about Quasi-Monte Carlo methods in Python.
• Latin Hypercube Sampling (centered, strength 1 or 2).
• Optimize a sample by minimizing $C^2$ discrepancy or performing Lloyd-Max iterations,
• Fast numerical inverse methods to sample arbitrary univariate distributions with QMC 
  (Baumgarten & Patel, 2022),
• QMC integration.

Before the release of SciPy 1.7.0, the need for these functions was partially met in the scientific 
Python ecosystem by tutorials (e.g. blog posts) and niche packages, but the functions in SciPy 
have several advantages:

• Popularity: With millions of downloads per month, SciPy is one of the most downloaded 
  scientific Python packages. New features immediately reach a wide range of users from 
  all fields.
• Performance: The low-level functions are written in compiled languages such as Cython 
  and optimized for speed and efficiency.
• Consistency: The APIs comply with the high standards of SciPy, function API reference 
  and tutorials are thorough, and the interfaces share common features complementing 
  other SciPy functions.
• Quality: As with all SciPy code, these functions were rigorously peer-reviewed for code 
  quality and are extensively unit-tested. In addition, the implementations were produced 
  in collaboration with the foremost experts in the QMC field.

Since the first release of all these new features, we have seen other libraries add support for 
and rely on SciPy’s implementations, e.g. Optuna (Ishikawa et al., 2022) and SALib (Roy 
& Iwanaga, 2022).

Acknowledgements

The authors thank professors Sergei Kucherenko (Imperial College London) and Fred Hickernell 
(Illinois Institute of Technology) for helpful discussions. The SciPy maintainer team provided 
support and help regarding the design and integration, notably Ralf Gommers (Quansight) and 
Tyler J. Reddy (Los Alamos National Laboratory).

References

Baumgarten, Christoph, & Patel, Tirth. (2022). Automatic random variate generation 
in Python. In Meghann Agarwal, Chris Calloway, Dillon Niederhut, & David Shupe 
(Eds.), Proceedings of the 21st Python in Science Conference (pp. 46–51). 
https://doi.org/10.25080/majora-212e5952-007

GitHub. https://github.com/optuna/optuna/pull/2423

for Industrial; Applied Mathematics. https://doi.org/10.1137/1.9781611970081

domains/mc/

1007/978-3-030-98319-2_4

module with quasi Monte Carlo functionality. In GitHub pull request. GitHub. https: 
//github.com/scipy/scipy/pull/10844

GitHub. https://github.com/SALib/SALib/pull/519