Note about the file `lognormal_demo2.py`

The "off-the-shelf" generating vector `lattice-39101-1024-1048576.3600` (call it $\boldsymbol{z} \in \mathbb{Z}^s$) is designed to satisfy

$$\int_{[0,1]^s} f(\boldsymbol{y}) \,\mathrm{d}\boldsymbol{y} \approx \frac{1}{n} \sum_{k=0}^{n-1} f(\boldsymbol{y}^{(k,n)}),$$

where the point set is given by

$$\boldsymbol{y}^{(k,n)} = \mathrm{mod}\bigg(\frac{k\boldsymbol{z}}{n} + \boldsymbol{\Delta}, 1\bigg), \quad k = 0, \ldots, n-1,$$

for *all* $n = 2^\ell$, $\ell = 10, \ldots, 20$, with $\boldsymbol{\Delta} \sim \mathcal{U}([0,1]^s)$.

This specific generating vector is a bit special since it is *extensible* in n, i.e., the point sets satisfy

$$\boldsymbol{y}^{(2k,2^{\ell+1})} = \boldsymbol{y}^{(k,2^\ell)}, \quad k = 0, \ldots, 2^\ell - 1, \ \ell = 10, \ldots, 19,$$

meaning that we can reuse the point set of a 2^ℓ-point cubature rule for the computation of a $2^{\ell+1}$-point cubature rule in the case of this special "off-the-shelf" lattice. *Not all lattice rules are extensible in n.* However, all lattice rules *are* extensible in the dimension s: for example, this particular lattice rule can be used for any dimension $s \in \{1, \ldots, 3\,600\}$ by truncating the generating vector to the first s entries.

We can apply this lattice point set in our integration problem using the change of variables

$$\int_{\mathbb{R}^s} u(\boldsymbol{x}, \boldsymbol{y}) \prod_{j=1}^{s} \frac{\mathrm{e}^{-\frac{1}{2} y_j^2}}{\sqrt{2\pi}} \,\mathrm{d}\boldsymbol{y} = \int_{(0,1)^s} u(\boldsymbol{x}, \Phi^{-1}(\boldsymbol{w})) \,\mathrm{d}\boldsymbol{w},$$

where $\Phi^{-1}(\boldsymbol{w}) = [\Phi^{-1}(w_1), \ldots, \Phi^{-1}(w_s)]^{\mathrm{T}}$ and $\Phi(w) = \int_{-\infty}^{w} \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-\frac{1}{2} y^2} \,\mathrm{d}y$ is the cumulative distribution function (CDF) of the standard normal distribution $\mathcal{N}(0,1)$.

Note that Φ^{-1} can be accessed as `scipy.stats.norm.ppf` in Python.