

- Implement Pseudocode for `omc_integral(n)`:

- Generate n iid samples

$$\{iid Y_i \sim U(0, 1) : i = 1, 2, \dots, n\};$$

- Compute n X samples by

$$\{X_i = h(Y_i) : i = 1, 2, \dots, n\};$$

- Take average of X_i 's

- Demonstrate convergence rate numerically by doing the following:

- Fix a batch number $m = 100$;

- For i in range(5, 10):

- * run m times of `omc_integral(n = 2i)`, store it into $\{\alpha_{ij} : j = 1, \dots, m\}$.

- * compute standard deviation (`numpy.std`) of $\{\alpha_{ij} : j = 1, \dots, m\}$, save it to σ_i .

- plot and find slope (`scipy.stats.linregress`) for the data

$$\{(i, -\log_2 \sigma_i) : i = 5, \dots, 10\}.$$

- Find its convergence rate by the following procedure:

Compute RMSE (root MSE) for $\hat{\alpha}_n$ in terms of $Cn^{-\alpha}$. We say α as the convergence rate.