MONTE CARLO METHODS (2)

Objectives:

- Compute an integral thanks to a Monte Carlo integration;
- Sample a distribution using the inverse transform sampling method;
- Implement a simulated annealing method to minimise a cost function.

No list manipulation is allowed in this tutorial!

I. Calculation of an improper integral via Monte Carlo integration

We want to evaluate the numerical value of the integral

$$I = \int_0^1 \left(\frac{1}{x^{1/3}} + \frac{x}{10} \right) \mathrm{d}x. \tag{1}$$

Its exact value is $31/20$. In the lecture notes, we have seen that one can compute the above integral by sampling a random variable X which follows the probability distribution density $p(x)$. The objective of this exercice is to analyse how the estimate of the integral can be improved by carefully choosing $p(x)$.

Question 1: Imagine that you draw N samples from a random variable X which follows the probability distribution density $p(x)$. Show that the Monte Carlo estimate of I as a function of the N samples x_1, x_2, $\ldots$, x_N reads

$$I \approx \frac{1}{N} \sum_{i=1}^{N} \frac{1}{p(x_i)} \left(\frac{1}{x_i^{1/3}} + \frac{x_i}{10} \right). \tag{2}$$

Question 2: We first take $p(x)$ as a uniform distribution in $[0,\, 1]$ to compute the integral. Evaluate the integral for $N = 100,\, 1000, \ldots, 10^6$ samples. For each value of N, you can do the calculation 20 times and average the result. Print the estimated values of I. Then plot $|I - 31/20|$ as a function of N in a loglog plot. Comment on the accuracy of the method as a function of N.
Hint: the accuracy of Monte Carlo integration is of order $N^{-1/2}$.

Question 3: To improve accuracy, we want to perform the Monte Carlo integration using the probability distribution density

$$p(x) = \frac{2}{3x^{1/3}} \tag{3}$$

of support $I = [0,\, 1[$. Why can we expect this choice of $p(x)$ to give a better estimate of the integral?

Question 4: Define a function `sampling_p(N)` which takes as an input an integer N and returns an array of size N containing N samples $\{x_i\}$ drawn from $p(x)$ given by Eq. (3) using the inverse transform sampling method.

Question 5: Using the function defined above, evaluate the integral for $N = 100,\, 1000, \ldots, 10^6$ samples drawn from $p(x)$ given by Eq. (3). For each value of N, you can do the calculation 20 times and average the result. Print the estimated values of I. Then plot $|I - 31/20|$ as a function of N in a loglog plot. Comment on the accuracy of the method as a function of N and compare with the uniform distribution.

II. The best concert tour

A music band wants to make a concert tour in France and has selected 13 cities among the biggest ones. The group wants to start from Paris, perform only once in each city and come back to Paris eventually. Because of ecological concerns, the band wants to travel the least number of kilometers. The geographical coordinates of the 13 French cities where they want to perform are listed in the array below. We recall that on Earth, given two locations of coordinates (λ_1, ϕ_1) and (λ_2, ϕ_2) (with λ_a the longitudes measuring the W/E deviation from the Greenwich Meridian and ϕ_a the latitudes measuring the N/S deviation from the Equator), their relative distance reads

$$d_{12} = R \arccos\left[\cos\phi_1 \cos\phi_2 \cos(\lambda_1 - \lambda_2) + \sin\phi_1 \sin\phi_2\right], \tag{4}$$

with $R = 6371\,\mathrm{km}$ the average radius of the Earth.

City	Longitude (O/E) λ	Latitude (N/S) ϕ
Paris	$2^\circ\ 21'\ 07''$ (E)	$48^\circ\ 51'\ 24''$ (N)
Lyon	$4^\circ\ 49'\ 56''$ (E)	$45^\circ\ 45'\ 28''$ (N)
Toulouse	$1^\circ\ 26'\ 38''$ (E)	$43^\circ\ 36'\ 16''$ (N)
Nice	$7^\circ\ 16'\ 17''$ (E)	$43^\circ\ 41'\ 45''$ (N)
Nantes	$1^\circ\ 33'\ 10''$ (O)	$47^\circ\ 13'\ 05''$ (N)
Montpellier	$3^\circ\ 52'\ 38''$ (E)	$43^\circ\ 36'\ 43''$ (N)
Strasbourg	$7^\circ\ 45'\ 08''$ (E)	$48^\circ\ 34'\ 24''$ (N)
Bordeaux	$0^\circ\ 34'\ 46''$ (O)	$44^\circ\ 50'\ 16''$ (N)
Lille	$3^\circ\ 03'\ 48''$ (E)	$50^\circ\ 38'\ 14''$ (N)
Le Havre	$0^\circ\ 06'\ 00''$ (E)	$49^\circ\ 29'\ 24''$ (N)
Clermont-Ferrand	$3^\circ\ 04'\ 56''$ (E)	$45^\circ\ 46'\ 33''$ (N)
Limoges	$1^\circ\ 15'\ 00''$ (E)	$45^\circ\ 51'\ 00''$ (N)
Orléans	$1^\circ\ 54'\ 32''$ (E)	$47^\circ\ 54'\ 09''$ (N)

Table 1: **Geographical coordinates of the 13 French cities chosen by a band for their French concert tour.** The longitudes λ and latitudes ϕ are given in degrees (°), minutes (′) and seconds (″). We recall that $1^\circ = 60' = 3600''$.

A direct brute-force test of the $12! = 479001600$ possible paths takes about 30 minutes to converge and shows that the smallest tour is: Paris → Orléans → Limoges → Clermont-Ferrand → Lyon → Nice → Montpellier → Toulouse → Bordeaux → Nantes → Le Havre → Lille → Strasbourg → Paris. The objective of this exercise is to find circuits which minimize the total length travelled using a simulated annealing algorithm.

Question 1: Construct the $n \times n$ symmetric matrix d_{ij} `distances` of all distances between cities i and j (with n the number of cities). You can use the piece of code below giving the lists of longitudes and latitudes in degrees.

```
cities = ['Paris', 'Lyon', 'Toulouse', 'Nice', 'Nantes', 'Montpellier', 'Strasbourg',
'Bordeaux', 'Lille', 'Le Havre', 'Clermont-Ferrand', 'Limoges', 'Orleans']
longitudes = [2. + 21./60. + 7./3600., 4. + 49./60. + 56./3600., 1. + 26./60. + 38./3600.,
7. + 16./60. + 17./3600., - (1. + 33./60. + 10./3600.), 3. + 52./60. + 38./3600.,
7. + 45./60. + 8./3600., - (0. + 34./60. + 46./3600.), 3. + 3./60. + 48./3600.,
0. + 6./60., 3. + 4./60. + 56/3600., 1. + 15./60., 1. + 54./60. + 32./3600.]
latitudes = [48. + 51./60. + 24./3600., 45. + 45./60. + 28./3600., 43. + 36./60. + 16./3600.,
43. + 41./60. + 45/3600., 47. + 13./60. + 5./3600., 43. + 36./60. + 43./3600.,
48. + 34./60. + 24./3600., 44. + 50./60. + 16./3600., 50. + 38./60. + 14./3600.,
49. + 29./60. + 24./3600., 45. + 46./60. + 33./3600., 45. + 51./60.,
47. + 54./60. + 9./3600.]
```

Circuits of cities are defined by a one-dimensional array `path` of size n containing integers between 0 and n, with `path[i]` corresponding to the index in the array `cities` of the city at the i^{th} position in the path. For the optimal path, the array would be `path = [0, 12, 11, 10, 1, 3, 5, 2, 7, 4, 9, 8, 6].`

Question 2: Define a function `path_distance(city_rank, distances)` which takes as an input the one-dimensional array `path` defined above, and the matrix `distances` of pair distances between cities, and returns the total length of the circuit. Using this function, compute the length L_{min} of the smallest tour described above.

In order to quickly find good approximations of the optimal path, we propose to implement a simulated annealing scheme. The simulated annealing technique assumes that a path of length L has a probability $\mathcal{P} \propto e^{-L/T}$ with T the 'temperature' (in km^{-1}) which will be decreased during the process. In practice, we start from an arbitrary circuit connecting all cities once, starting from Paris and coming back to Paris. At each step, we propose the following procedure:

- Try to swap two cities in the path except Paris and compute the change in length ΔL of the circuit.
- Accept and reject the new trial path following the Metropolis criterion assuming that the probability of a path of total length L is proportionsl to $e^{-L/T}$ (with T the temperature).
- Update the temperature T by decreasing it by a factor $1 - \alpha$ at each step.

The simulated annealing scheme starts with an initial temperature T_{i} and ends with a final temperature T_{f}.

Question 3: Define a function `simulated_annealing(distances, alpha, Ti, Tf)` which takes as an input the two-dimensional $n \times n$ matrix `distances` containing the distances between cities, the factor α to decrease the temperature at each step, the initial temperature T_{i} and the final temperature T_{f}, implements the simulated annealing procedure, and returns a one-dimensional array `path` of size n containing integers between 0 and n, with `path[i]` corresponding to the index in the array `cities` of the city at the i^{th} position in the path.
Hint: you can take `path=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]` *as an initial path.*

Question 4: The performance of the algorithm depends on the annealing rate α. For $T_{\mathrm{i}} = 100$ and $T_{\mathrm{f}} = 10^{-5}$, run the algorithm 20 times for $\alpha = 10^{-1}$, 10^{-2}, 10^{-3}, 10^{-4} and compute the average length $\langle L \rangle(\alpha)$ of the path found by the simulated annealing procedure.

Question 5: Plot $\langle L \rangle(\alpha)$ as a function of α with a logarithmic scale for the abscissae. Is the simulated annealing algorithm able to find the minimal path?

Question 6: Plot $\langle L \rangle(\alpha) - L_{\mathrm{min}}$ as a function of α in a loglog plot. How does the difference scale with α?

III. Calculation of the volume of the unit ball via Monte Carlo integration

Compute the volume of the unit ball (the set of vectors of norm smaller or equal than 1) in D dimensions using a Monte Carlo integration. Compare with the exact results $4\pi/3$ and $\pi^5/120$ for $D = 3$ and $D = 10$ respectively. You can do the calculation 20 times and average the result for each value of D.