**Mini-project: Solving the N-queens problems via the Metropolis algorithm**

# 1 Finding a solution

In this first part, you should use the Metropolis algorithm in order to solve the $N$-queens problem.

Here are questions you should think about:

- what state space $S$ to choose?

- what base chain?

- what energy function $f : S \to \mathbb{R}$ should be optimized?

- how to compute/update efficiently this energy function?

- what are the resulting acceptance probabilities (and Metropolis chain)?

- what inverse temperature $\beta$ should be chosen?

- is it worth using simulated annealing here?

**Your tasks for this first part of the project:**

- Code the Metropolis algorithm, deciding on the best choices to make (which you should explain in your report).

- Show us some graphs demonstrating that your code works for $N$ around 100, 1000.

# 2 Estimating the number of solutions

For a given value of $N$, let us denote this number as $Z_\infty$, as it corresponds to the normalization constant (or partition function) for the case $\beta = \infty$ (cf. course): this number $Z_\infty$ is typically hard to compute. Currently, its precise value is known up to $N = 26$ only. We propose below some steps to evaluate this number for large values of $N$.

Let $Z_0$ be the partition function corresponding to the case $\beta = 0$ (this number is easily computable, whatever the choice you made in part 1). Theoretically, the following approach is possible in order to estimate $Z_\infty$:

- draw a configuration at random according to $\pi_0$ and check whether it is a valid configuration (i.e., with no queens attacking each other);

- repeat this a zillion times and count the number of valid configurations: this number divided by the total number of draws is an estimator of the ratio $Z_\infty/Z_0$.

Problem: this number $Z_\infty/Z_0$ is extremely small, so the variance of your estimator will be huge, unless you draw a *really huge* number of samples.

A better approach is therefore to do the following: fix a target $\beta^*$ that you think is "close enough" to $\infty$, so that $Z_{\beta^*} \simeq Z_\infty$; the first part should give you a clue about which value $\beta^*$ to choose.

Then observe that for $0 = \beta_0 < \beta_1 < \beta_2 < \ldots < \beta_T = \beta^*$, we have the following equality

$$Z_{\beta^*}/Z_0 = \prod_{t=0}^{T-1} Z_{\beta_{t+1}}/Z_{\beta_t}$$

which can be rewritten as

$$\log(Z_{\beta^*}/Z_0) = \sum_{t=0}^{T-1} \log(Z_{\beta_{t+1}}/Z_{\beta_t})$$

The advantage of this approach with respect to the first one is that if $\beta_t$ and $\beta_{t+1}$ are close enough, then the ratio $Z_{\beta_{t+1}}/Z_{\beta_t}$ is not so small and can be rewritten as follows:

$$\begin{aligned}\frac{Z_{\beta_{t+1}}}{Z_{\beta_t}} &= \frac{\sum_{x\in S} \exp(-\beta_{t+1}\, f(x))}{\sum_{y\in S} \exp(-\beta_t\, f(y))} = \frac{\sum_{x\in S} \exp(-\beta_{t+1}\, f(x) + \beta_t\, f(x) - \beta_t\, f(x))}{\sum_{y\in S} \exp(-\beta_t\, f(y))} \\ &= \sum_{x\in S} \exp(-(\beta_{t+1} - \beta_t)\, f(x))\, \pi_{\beta_t}(x) \quad \text{where} \quad \pi_{\beta_t}(x) = \frac{\exp(-\beta_t\, f(x))}{\sum_{y\in S} \exp(-\beta_t\, f(y))}\end{aligned}$$

In order to estimate this ratio, draw $M$ i.i.d. samples $X_1, \ldots, X_M$ according to $\pi_{\beta_t}$ (using the Metropolis algorithm) and compute

$$\frac{1}{M} \sum_{k=1}^{M} \exp(-(\beta_{t+1} - \beta_t)\, f(X_k))$$

Repeating then this procedure for all values of $t \in \{0, 1, \ldots, T-1\}$ (and multiplying the results) allows to estimate the ratio $Z_{\beta^*}/Z_0$, meant to be close to $Z_\infty/Z_0$; this allows finally to estimate $Z_\infty$.

**Your tasks for this second part of the project:**

- Choose values of $0 < \beta_1 < \beta_2 < \ldots < \beta_T = \beta^*$, as well as $T$, which you think will lead to a good estimator of $Z_\infty$.

- Draw a graph of $\log(Z_{\beta_t})$ as a function of $\beta_t$, using your successive estimators.

- What can you say about the variance of your estimator of $Z_\infty$?

**Deadline for your report:** Monday, December 19, 11:59 PM.

# 3 Competition

During the competition, to take place on Thursday, December 22, 12:15 PM, we will ask you to change your algorithm, so as to solve a slightly different problem. We will still specify in which format exactly we expect you to send us your solution(s).

**Your tasks for this third part of the project:** Be ready for it!