Initial Setup

We now begin to setup the initial framework of our search for the lowest energy configuration of Au^{10}. We restrict the system to some suitable subset of $\omega \subset \mathbb{R}^2$, but since each atom has 2 degrees of freedom, we still end up with a very large configuration space,

$$\omega \times \ldots \times \omega := \Omega \subset \mathbb{R}^{20}$$

The sheer scale of the problem gives rise to issues, that were not present in the $1d$-version of the problem that we initially considered. In order to overcome this problem, we rely on a couple of tricks.

1.0.1 *Equivalent and feasible configurations*

As we have already discussed DFT energy calculations are expensive. In order to limit the amount of calculations we have to perform, an SQL database is created for storing previously explored configurations. We also implement a series of cheap equivalence tests. These should catch configurations, that on the surface level seem different, but are in fact just different representations of the same configuration. Formally, a configuration is a vector,

$$z = \begin{pmatrix} (x_1, y_1) \\ \vdots \\ (x_n, y_n) \end{pmatrix} \in \Omega.$$

We say that two configurations z_1, z_2 are equivalent, if either of the following relations hold,

$$z_1 \sim z_2 \iff \begin{cases} \exists \alpha \in \mathbb{R} \mid z_1 = z_2 + \alpha \\ rotational matrix \\ permtation \end{cases}.$$

These equivalences corresponds to translational- and rotational transformations, aswell as permutation of the atoms in the configuration. The assumption is, that configurations are energy-invariant under these transformations. Besides these checks, we can also implement some *feasibility* checks. Meaning, is the configuration physically allowed. The atoms in the system are not, physically, allowed to overlap. This criteria allows us to constrict Ω even further.

1.0.2 *Latin Hypercube Sampling*

1.0.3 *Branch and bound*

The size of the configuration space Ω, means that optimization of our otherwise inexpensive aquistion becomes infeasible. This is where the simple closed form of **EI** becomes relevant. Notice, that it is monotonic,

$$calculations.$$

We can exploit this property in a *branch and bound algorithm* $(BnB\ )$,