**Constraint Type & Search Space:**

In Scipy, both the SLSQP optimizer and the Trust-Region Constrained Algorithm (trust-constr) optimizer can be used to solve constrained minimization problems. Specifically, the SLSQP optimizer uses bounds to define the search space, while the trust-constr optimizer uses constraints to define the search space.

In the SLSQP optimizer, we need to pass bounds as a parameter to limit the search space. Bounds can be a sequence that contains the lower and upper bounds of each variable. During the search process, the optimizer only considers variable values within this range.

On the other hand, in the trust-constr optimizer, we need to pass constraints as a parameter to define the search space. These constraints can be equality or inequality constraints, linear or nonlinear. During the search process, the optimizer considers variable values that satisfy all constraints.

COBYLA optimizer in Scipy is more similar to SLSQP optimizer in terms of using bounds to define the search space. Specifically, COBYLA optimizer uses the "rhobeg" parameter to define the initial step size for each variable, which is similar to defining the bounds in SLSQP optimizer. In addition, COBYLA optimizer can handle inequality constraints, similar to SLSQP optimizer.

However, COBYLA optimizer does not require the explicit expression of constraints in the form of functions, unlike trust-constr optimizer. Therefore, COBYLA optimizer is more suitable for solving problems with simple constraints that can be easily expressed as bounds, but less suitable for complex constraints that require explicit functions.

**Norm:**

The norm2\_sq function calculates the squared Euclidean distance between the linear function A.dot(x\_choose) and a target vector b. It can be written as:

|| A.x\_choose - b ||^2 = (A.x\_choose - b)^T (A.x\_choose - b)

where || . || denotes the L2 norm, ^T denotes transpose, and A is a matrix determined by the input vector x\_choose.

This function is used as the objective function in an optimization problem that seeks to find the x\_choose that minimizes the distance between A.dot(x\_choose) and b. The optimization algorithm used in this case is specified by the method parameter in the optimize.minimize function. The optimization is subject to constraints specified by the constraints parameter, which can be either inequality constraints on x\_choose or nonlinear constraints on the output of A.dot(x\_choose).

In the code provided, the optimization is performed with three different optimization methods: SLSQP, trust-constr, and COBYLA. The optimization seeks to find the x\_choose that minimizes the distance between A.dot(x\_choose) and b, subject to the constraints that the output values lie within a certain range [LB, UB]. These constraints are implemented as inequality constraints on either x\_choose or the output of the linear function using nonlinear constraints.

The make\_matrix function constructs the matrix A, which defines the linear function A.dot(x\_choose). It constructs A using the input vector x\_choose and the degree specified by the parameter k. The resulting optimization problem is then solved using the specified optimization method and constraints, and the resulting coefficients c of the linear function are used to plot the estimated function.

In summary, the norm2\_sq function calculates the distance between a linear function A.dot(x\_choose) and a target vector b, which is used as the objective function in the optimization problem. The optimization seeks to find the x\_choose that minimizes the distance subject to constraints, using various optimization methods and constraints. The make\_matrix function constructs the matrix A, which defines the linear function A.dot(x\_choose).