CAQM: Convexity Analysis of Quadratic Maps

Anatoly Dymarsky, Elena Gryazina, Boris Polyak, Sergei Volodin

Name	**Input**	**Call**	**Description**	**Return value**	**Exception**
Random map	Dimensions n, m	`get_random_f(n, m, is_complex)`	Generates random map f	`[A, b]`	None
Value at x	The point $x \in X$	`quadratic_map(A, b, x)`	Calculates $f(x)$	$y = f(x)$	None
Product $c \cdot A$	Normal vector c	`get_Ac(A, c)`	Calculates $c \cdot A$	$A_c = c \cdot A$	None
Get H_c	$c, y \in \mathbb{R}^m$	`get_H_c(A, b, c, y)`	$H_c = \begin{pmatrix} A_c & b_c \\ b_c' & -(c, y) \end{pmatrix}$	H_c	None
Minimize $z(c)$	c, c_+, step β	`minimize_z_c(A, b, c, c_plus, beta_initial, max_step)`	Calculates $\inf\limits_{c \in C_-} z(c)$	`[z, c_array, z_array]`	If failed
$\mathbb{R}^n$ **projection**		`project(A, b, c, x_0, delta_c, normal, search_area_size)`	Projects $c + \Delta c$ to C_-	`[c_new, lambda]`	If failed
$\mathbb{C}^n$ **projection**		`project_descent(A, b, c, normal_1, normal_2)`	Projects c to C_-	`[c_new, distance]`	If failed
Gradient $\frac{\partial z}{\partial c}$	Normal c	`get_dz_dc(A, b, c)`	Calculates Q, $\nabla z(c)$, normal vectors n_1, n_2	`[Q, Q_inv, k, v, lambda_min, z, dz_dc, normal_re, normal_im, drho_dc]`	None
Change of basis	c_+	`change_basis(A, b, c_plus)`	$\begin{cases} x = S(x' + x_0) \\ y = y' + y_0 \end{cases}$ s.t. $\begin{cases} c_+ \cdot A_0 = I \\ c_+ \cdot b_0 = 0 \end{cases}$	`[A_new, b_new, x0, y0]`	None