

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	<code>get_random_f(n, m, is_complex)</code>	Generates random map f	$[A, b]$	None
Value at x	The point $x \in X$	<code>quadratic_map(A, b, x)</code>	Calculates $f(x)$	$y = f(x)$	None
Product $c \cdot A$	Normal vector c	<code>get_Ac(A, c)</code>	Calculates $c \cdot A$	$A_c = c \cdot A$	None
Get H_c	$c, y \in \mathbb{R}^m$	<code>get_H.c(A, b, c, y)</code>	$H_c = \begin{pmatrix} A_c & b_c \\ b'_c & -(c, y) \end{pmatrix}$	H_c	None
Minimize $z(c)$	$c, c_+, \text{step } \beta$	<code>minimize_z.c(A, b, c, c_plus, beta_initial, max_step)</code>	Calculates $\inf_{c \in C_-} z(c)$	$[z, c_array, z_array]$	If failed
\mathbb{R}^n projection		<code>project(A, b, c, x_0, delta_c, normal, search_area_size)</code>	Projects $c + \Delta c$ to C_-	$[c_new, \text{lambda}]$	If failed
\mathbb{C}^n projection		<code>project_descent(A, b, c, normal_1, normal_2)</code>	Projects c to C_-	$[c_new, \text{distance}]$	If failed
Gradient $\frac{\partial z}{\partial c}$	Normal c	<code>get_dz.dc(A, b, c)</code>	Calculates $Q, \nabla z(c)$, normal vectors n_1, n_2	$[Q, Q_inv, k, v, \text{lambda_min}, z, dz_dc, \text{normal_re}, \text{normal_im}, \text{drho_dc}]$	None
Change of basis	c_+	<code>change_basis(A, b, c_plus)</code>	$\begin{cases} x = S(x' + x_0) \\ y = y' + y_0 \end{cases} \quad \text{s.t.} \quad \begin{cases} c_+ \cdot A_0 = I \\ c_+ \cdot b_0 = 0 \end{cases}$	$[A_new, b_new, x_0, y_0]$	None
Boundary point ∂F_c	Normal c	<code>boundary_point(A, b, c)</code>	Calculates pre-image $x: f(x) \in \partial F_c$	$[x]$	If $c \cdot A < 0$