For a given dimension d, number of sets s and a series of measurements $M_i^k \in \mathbb{C}^d \otimes \mathbb{C}^d$ our problem is as follows:

$$\min \sum_{i,j}^{s} \sum_{k,l}^{d} -\sqrt{1 - \mathrm{tr}(M_k^i M_l^j)} \tag{1}$$

subject to:

$$M_k^i \succeq 0 \tag{2}$$

$$\sum_{k}^{d} M_k^i = \mathbb{I} \tag{3}$$

$$\mathrm{tr}(M_k^i) = 1 \tag{4}$$

$$(M_k^i)^2 = M_k^i \tag{5}$$

Need to define three functions for this method: the objective function $f(x)$, the constraint function $g(x)$ and the function $X(x)$ converting between the vector form x and the matrix form such that $X(x)$ is positive semidefinite. These functions do not need to be linear.

In our case $X(x)$ takes the form:

$$X(x) = \sum_{a}^{n} x_a A_a + B \tag{6}$$

such that it turns a vector $x \in \mathbb{R}^n$ containing the unique elements of the measurements into a matrix $X(x) \in \mathbb{R}^p \otimes \mathbb{R}^p$ which contains the real components on the block diagonals and the imaginary components on the off-diagonals, with $p = 2d^2n$. For instance, in the $d = 2$, $n = 2$ case, if each $M_i^k = R_i^k + iI_i^k$:

$$X(x) = \begin{pmatrix} R_1^1 & 0 & 0 & 0 & I_1^1 & 0 & 0 & 0 \\ 0 & R_2^1 & 0 & 0 & 0 & I_2^1 & 0 & 0 \\ 0 & 0 & R_1^2 & 0 & 0 & 0 & I_1^2 & 0 \\ 0 & 0 & 0 & R_2^2 & 0 & 0 & 0 & I_2^2 \\ I_1^1 & 0 & 0 & 0 & R_1^1 & 0 & 0 & 0 \\ 0 & I_2^1 & 0 & 0 & 0 & R_2^1 & 0 & 0 \\ 0 & 0 & I_1^2 & 0 & 0 & 0 & R_1^2 & 0 \\ 0 & 0 & 0 & I_2^2 & 0 & 0 & 0 & R_2^2 \end{pmatrix} \tag{7}$$

Many of these elements are defined in relation to others such that the vector x contains as little information as needed, whilst also forcing the submatrices of any $X(x)$ to satisfy the identity and trace constraints. This also has the benefit of meaning in our case $g(x)$ is only used for the projective constraint.

Splitting the objective function into real and imaginary components and using the identity $\mathrm{tr}(M_k^i M_l^j) = M_k^i \cdot M_l^j$:

$$f(x) = \sum_{i,j}^{s} \sum_{k,l}^{d} -\sqrt{1 - R_k^i \cdot R_l^j + I_k^i \cdot I_l^j} \tag{8}$$

Now defining the extraction matrices C_k^i, D_k^i, E_k^i and F_k^i such that $C_k^i X D_k^i = R_k^i$ and $E_k^i X F_k^i = I_k^i$ with the notation that $X = X(x)$:

$$f(x) = \sum_{i,j}^{s} \sum_{k,l}^{d} -\sqrt{1 - C_k^i X D_k^i \cdot C_l^j X D_l^j + E_k^i X F_k^i \cdot E_l^j X F_l^j} \tag{9}$$

Thus the full expression in terms of the components of x:

$$f(x) = \sum_{i,j}^{s} \sum_{k,l}^{d} -\Bigg(1 - C_k^i(\sum_a^n x_a A_a + B)D_k^i \cdot C_l^j(\sum_a^n x_a A_a + B)D_l^j \tag{10}$$

$$+E_k^i(\sum_a^n x_a A_a + B)F_k^i \cdot E_l^j(\sum_a^n x_a A_a + B)F_l^j\Bigg)^{\frac{1}{2}} \tag{11}$$

Now taking the first derivative of this, letting d_k^i be the value inside the above square root:

$$\frac{\partial f(x)}{\partial x_b} = \sum_{i,j}^{s} \sum_{k,l}^{d} -\frac{1}{2}(d_k^i)^{-\frac{1}{2}}(-C_k^i A_b D_k^i \cdot C_l^j X D_l^j - C_k^i X D_k^i \cdot C_l^j A_b D_l^j \tag{12}$$

$$+E_k^i A_b F_k^i \cdot E_l^j X F_l^j + E_k^i X F_k^i \cdot E_l^j A_b F_l^j) \tag{13}$$

Then the second derivative, remembering that d_k^i has a dependence on x:

$$\frac{\partial^2 f(x)}{\partial x_b \partial x_c} = \sum_{i,j}^{s} \sum_{k,l}^{d} \frac{1}{4} (d_k^i)^{-\frac{3}{2}} (-C_k^i A_b D_k^i \cdot C_l^j X D_l^j - C_k^i X D_k^i \cdot C_l^j A_b D_l^j \tag{14}$$

$$+ E_k^i A_b F_k^i \cdot E_l^j X F_l^j + E_k^i X F_k^i \cdot E_l^j A_b F_l^j) \tag{15}$$

$$(-C_k^i A_c D_k^i \cdot C_l^j X D_l^j - C_k^i X D_k^i \cdot C_l^j A_c D_l^j \tag{16}$$

$$+ E_k^i A_c F_k^i \cdot E_l^j X F_l^j + E_k^i X F_k^i \cdot E_l^j A_c F_l^j) \tag{17}$$

$$-\frac{1}{2} (d_k^i)^{-\frac{1}{2}} (-C_k^i A_b D_k^i \cdot C_l^j A_c D_l^j - C_k^i A_c D_k^i \cdot C_l^j A_b D_l^j \tag{18}$$

$$+ E_k^i A_b F_k^i \cdot E_l^j A_c F_l^j + E_k^i A_c F_k^i \cdot E_l^j A_b F_l^j) \tag{19}$$

Note that in the above expressions many of these terms are zero, for instance $C_k^i A_b D_k^i \cdot C_l^j A_b D_l^j = 0$ since each A matrix will only have non-zero components for a single measurement, so one of those two extractions must result in a zero matrix.

Computationally many things here are different, such that X is only ever calculated once per iteration and the extraction matrices are unneeded, instead the submatrices are extracted directly from the cached X using Eigen's "X.block()" routine, which claims $O(0)$ scaling when compiled with optimisations.

Putting these all together we can construct the Lagrangian and it's first/second derivatives, which are then used as in the paper.