Noname manuscript No.
(will be inserted by the editor)

Projection Methods for Quantum Channel Construction

Dmitriy Drusvyatskiy · Chi-Kwong Li · Diane Christine Pelejo · Yuen-Lam Voronin · Henry Wolkowicz

Received: date / Accepted: date

Abstract We consider the problem of constructing quantum channels, if they exist, that transform a given set of quantum states $\{\rho_1, \ldots, \rho_k\}$ to another such set $\{\hat{\rho}_1, \ldots, \hat{\rho}_k\}$. In other words, we must find a *completely positive linear map*, if it exists, that maps a given set of density matrices to another given set of density matrices, possibly of different dimension. Using the theory of completely positive linear maps, one can formulate the problem as an instance of a positive semidefinite feasibility problem with highly structured constraints. The nature of the constraints makes projection based algorithms very appealing when the number of variables is huge and standard interior point-methods for semidefinite programming are not applicable. We provide empirical evidence to this effect. We moreover present heuristics for finding both high rank and low rank solutions. Our experiments are based on the *method of alternating projections* and the *Douglas-Rachford* reflection method.

D. Drusvyatskiy
Department of Mathematics, University of Washington, Seattle, WA
ddrusv@uw.edu

C-K. Li
Department of Mathematics, College of William and Mary, Williamsburg, VA
ckli@math.wm.edu

D. Pelejo
Department of Mathematics, College of William and Mary, Williamsburg, VA
dcpelejo@gmail.com

Y.-L. Voronin
Department of Computer Science, University of Colorado, Boulder, CO
E-mail: yuen-lam.voronin@colorado.edu

H. Wolkowicz
Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Canada
hwolkowicz@uwaterloo.ca

Keywords quantum channels · completely positive linear maps · alternating projection methods · Douglas-Rachford method · Choi matrix · semidefinite feasibility problem · large scale

PACS 90C22 · 65F10 · 81Q10

1 Introduction

A basic problem in quantum information science is to construct, if it exists, a *quantum channel* sending a given set of *quantum states* $\{\rho_1, \ldots, \rho_k\}$ to another set of quantum states $\{\hat{\rho}_1, \ldots, \hat{\rho}_k\}$; see e.g., [10,18,20,24,25,28] and the references therein. Quantum states ρ_j are mathematically represented as *density matrices* — positive semidefinite Hermitian matrices with trace one which we denote as A_j; while quantum channels are represented by *trace preserving completely positive linear maps* — mappings T from the space of $n \times n$ density matrices in $\mathcal{M}^n$ to $m \times m$ density matrices in $\mathcal{M}^m$ having the form

$$T(X) = \sum_{j=1}^{r} F_j X F_j^*, \tag{1}$$

for some $m \times n$ matrices $F_1, \ldots, F_r$ satisfying $\sum_{j=1}^{r} F_j^* F_j = I_n$. See [11,21,28] for more details. Here $\mathcal{M}^t$ is the vector space of $t \times t$ complex matrices.

Thus given some density matrices $A_1, \ldots, A_k \in \mathcal{M}^n$ and $B_1, \ldots, B_k \in \mathcal{M}^m$, our task is to find a trace preserving completely positive linear map T satisfying $T(A_j) = B_j$, for each $j = 1, \ldots, k$. In turn, if we let $\{E_{11}, E_{12}, \ldots, E_{nn}\}$ denote the *standard orthonormal basis of* $\mathcal{M}^n$, then a mapping T is a trace preserving completely positive linear map if, and only if, the celebrated *Choi matrix of* T, defined in block form by

$$C(T) := \begin{bmatrix} P_{11} & \ldots & P_{1n} \\ \vdots & P_{st} & \vdots \\ P_{n1} & \ldots & P_{nn} \end{bmatrix} := \begin{bmatrix} T(E_{11}) & \ldots & T(E_{1n}) \\ \vdots & T(E_{st}) & \vdots \\ T(E_{n1}) & \ldots & T(E_{nn}) \end{bmatrix} \tag{2}$$

is positive semidefinite and the trace preserving constraints, $\mathrm{trace}(P_{st}) = \delta_{st}$, hold, where δ_{st} is the *Kronecker delta*. Note that the Choi matrix $C(T)$ is a square $nm \times nm$ matrix, and hence can be very large even for moderate values of m and n. Using the Choi matrix, ones sees that our problem is equivalent to the positive semidefinite feasibility problem for $P = (P_{st})$:

$$\left\{ \begin{array}{rl} \sum_{s,t} (A_\ell)_{st} P_{st} = B_\ell, & \ell = 1, \ldots, k \\ \mathrm{trace}(P_{st}) = \delta_{st}, & 1 \le s \le t \le n \\ P \in \mathcal{H}_+^{nm} & \end{array} \right\}, \tag{3}$$

where $\mathcal{H}_+^{nm}$ denotes the set of $nm \times nm$ positive semidefinite Hermitian matrices. Moreover, the rank of the Choi matrix P has a natural interpretation: it equals the minimal number of summands needed in any representation of

the form (1) for the corresponding trace preserving completely positive map T.

The set of all positive semidefinite matrices $P \in \mathcal{H}_+^{nm}$ that satisfy the constant-trace constraint is a compact set. Therefore, problem (3) is never *weakly infeasible*, i.e., infeasible but contains an asymptotically feasible sequence $\{P^{(j)}\}_{j=1}^{\infty}$ that satisfies the constraints in the limit. e.g., [14]. This means that accurate algorithms can detect infeasibility. In the infeasible case, the *alternating projection (MAP)* and the *Douglas-Rachford (DR)* projection/reflection algorithms that we use converge to the *nearest* points between the linear manifold defined by the linear constraints and the semidefinite cone. Infeasibility is detected if these nearest points are *not equal*.

We note that one can use standard primal-dual interior point semidefinite programming packages to solve the feasibility problem. However, when the size of the problem (m, n) grows, the efficiency and especially the accuracy of the semidefinite programming approach is limited. To illustrate, even for a reasonable sized problem $m = n = 100$, the number of complex variables involved is $10^8/2$.

In this paper, we exploit the special structure of the problem and develop projection based methods to solve high dimensional problems with high accuracy. We present numerical experiments based on the *alternating projection (MAP)* and the *Douglas-Rachford (DR)* projection/reflection methods. We see that the DR method significantly outperforms MAP for this problem. Our numerical results show promise of projection based approaches for many other types of feasibility problems arising in quantum information science.

We continue in Section 2 with a description of projection methods and the details for the particular implementation for our problem. Section 3 contains the numerical experiments. This includes heuristics for maximum rank solutions in Section 3.2 and low and constrained rank solutions in Section 3.3. Our concluding remarks are in Section 4.

2 Projection methods for constructing quantum channels

2.1 General background on projection methods

We begin by describing the method of alternating projections (MAP) and the Douglas-Rachford method (DR) in full generality. To this end, consider a Euclidean space $\mathcal{E}$ with an inner product $\langle \cdot, \cdot \rangle$ and the induced norm $\|\cdot\|$. We are interested in finding a point x lying in the intersection of two nonempty closed subsets A and B of $\mathcal{E}$. For example, A may be an affine subspace of Hermitian matrices (over the reals) and B may be the convex cone of positive semidefinite Hermitian matrices (over the reals), as in our basic quantum channel problem (3). Projection based methods then presuppose that given a point $x \in \mathcal{E}$, finding a point in the nearest-point set

$$\mathrm{proj}_A(x) = \mathrm{argmin}_{a \in A}\{\|x - a\|\}$$

is easy, as is finding a point in $\mathrm{proj}_B(x)$. When A and B are closed and convex, the nearest-point sets $\mathrm{proj}_A(x)$ and $\mathrm{proj}_B(x)$ are singletons.

Given a current point $a_l \in A$, the method of alternating projections then iterates the following two steps:

$$\begin{aligned} &\text{choose} \quad b_l \in \mathrm{proj}_B(a_l); \\ &\text{choose} \quad a_{l+1} \in \mathrm{proj}_A(b_l). \end{aligned}$$

When A and B are convex and there exists a pair of nearest points of A and B, the method always generates iterates converging to such a pair. In particular, when the convex sets A and B intersect, the method converges to some point in the intersection $A \cap B$. Moreover, when the relative interiors of A and B intersect, convergence is R-linear with the rate governed by the cosines of the angles between the vectors $a_{l+1} - b_l$ and $a_l - b_l$. For details, see for example [2,3,9,17]. When A and B are not convex, analogous convergence guarantees hold, but only if the method is initialized sufficiently close to the intersection [5, 13, 22, 23].

The Douglas-Rachford algorithm takes a more asymmetric approach. Given a point $x \in \mathcal{E}$, we define the reflection operator

$$\mathrm{refl}_A(x) = \mathrm{proj}_A(x) + (\mathrm{proj}_A(x) - x).$$

The Douglas-Rachford algorithm is then a "reflect-reflect-average" method; that is, given a current iterate $x_l \in \mathcal{E}$, it generates the next iterate by the formula

$$x_{l+1} = \frac{x_l + \mathrm{refl}_A(\mathrm{refl}_B(x_l))}{2}.$$

It is known that for convex instances, the "projected iterates" converge [26]. The rate of convergence has recently been shown to be linear when A and B are affine subspaces with nonempty intersection [19, Theorem 4.6], and R-linear when A and B are general convex sets whose relative interiors intersect [29, Theorem 4.14]. The Douglas-Rachford algorithm has proven to be extremely effective empirically for many types of problems; see for example [1,4,16]. The Douglas-Rachford algorithm has been applied to nonconvex problems (even though the convergence guarantee is not known) in physics applications; see e.g. [6].

The salient point here is that for MAP and DR to be effective in practice, the nearest point mappings proj_A and proj_B must be easy to evaluate. We next observe that for the quantum channel construction problem — our basic problem — these mappings are indeed fairly easy to compute (especially the projection onto the affine subspace).

2.2 Computing projections in the quantum channel construction problem

In the current work, we always consider the space of Hermitian matrices $\mathcal{H}^{nm}$ as a Euclidean space, that is we regard $\mathcal{H}^{nm}$ as an inner product space

over the reals in the obvious way (where the inner product is defined as $\langle S, T\rangle := \mathrm{trace}(ST) \in \mathbb{R}$ for all $S, T \in \mathcal{H}^{nm}$). The trace inner product induces the Frobenius norm $\|P\| := \left(\sum_{i,j}(\mathrm{Re}\, P_{ij})^2 + (\mathrm{Im}\, P_{ij})^2\right)^{1/2}$, where $\mathrm{Re}\, P_{ij}$ and $\mathrm{Im}\, P_{ij}$ are the real and the complex parts of P_{ij}, respectively.

Recall that our basic problem is to find a Hermitian matrix $P = (P_{st})$ satisfying

$$\left\{ \begin{array}{ll} \sum_{s,t}(A_\ell)_{st}P_{st} = B_\ell, & \ell = 1, \ldots, k \\ \mathrm{trace}(P_{st}) = \delta_{st}, & 1 \le s \le t \le n \\ P \in \mathcal{H}_+^{nm} & \end{array} \right\}. \tag{4}$$

We aim to apply MAP and DR to this formulation. To this end, we first need to introduce some notation to help with the exposition. Define the linear mappings $\mathcal{L}_A : \mathcal{H}^{nm} \to \otimes_{j=1}^k \mathcal{H}^m$ and $\mathcal{L}_T : \mathcal{H}^{nm} \to \mathcal{H}^n$ by

$$\mathcal{L}_A(P) := \Big(\sum_{s,t}(A_\ell)_{st}P_{st}\Big)_\ell \quad \text{and} \quad \mathcal{L}_T(P) := \Big(\mathrm{trace}(P_{st})\Big)_{s,t},$$

and let

$$\mathcal{L}(P) := (\mathcal{L}_A(P), \mathcal{L}_T(P)) \in \big(\otimes_{j=1}^k \mathcal{H}^m\big) \times \mathcal{H}^n. \tag{5}$$

Moreover assemble the vectors

$$B = (B_1, \ldots, B_k) \quad \text{and} \quad \Delta = (\delta_{st})_{s,t}.$$

Thus we aim to find a matrix P in the intersection of $\mathcal{H}_+^{nm}$ with the affine subspace

$$\mathcal{A} := \{P \in \mathcal{H}^{nm} : \mathcal{L}(P) = (B, \Delta)\}.$$

Projecting a Hermitian matrix P onto $\mathcal{H}_+^{nm}$ is standard due to the Eckart-Young Theorem [15]. Indeed, if $P = U\,\mathrm{Diag}(\lambda_1, \ldots, \lambda_{mn})U^*$ is an eigenvalue decomposition of P, then we have

$$\mathrm{proj}_{\mathcal{H}_+^{mn}}(P) = U\,\mathrm{Diag}(\lambda_1^+, \ldots, \lambda_{mn}^+)U^*,$$

where for any real number r, we set $r^+ = \max\{0, r\}$. Thus projecting a Hermitian matrix onto $\mathcal{H}_+^{mn}$ requires a single eigenvalue decomposition — a procedure for which there are many efficient and well-tested codes (e.g., [12]).

We next describe how to perform the projection onto the affine subspace $\mathcal{A}$, that is how to solve the nearest point problem

$$\min\Big\{\frac{1}{2}\|P - \hat{P}\|^2 : \mathcal{L}(\hat{P}) = (B, \Delta)\Big\}.$$

Classically, the solution is

$$\mathrm{proj}_{\mathcal{A}}(P) = P + \mathcal{L}^\dagger R,$$

where $\mathcal{L}^\dagger$ is the Moore-Penrose generalized inverse of $\mathcal{L}$ and $R := (B, \Delta) - \mathcal{L}(P)$ is the residual. In particular, computing $\mathrm{proj}_{\mathcal{A}}(\cdot)$ requires either solving a large scale linear equation (for n^2m^2 real-valued unknowns), or finding the map $\mathcal{L}^\dagger$.

Finding the Moore-Penrose generalized inverse of a large linear mapping, like the one we have here, can often be time consuming and error prone. Luckily, the special structure of the affine constraints in our problem allow us to find $\mathcal{L}^\dagger$ both very quickly and very accurately, so that in all our experiments the time to compute the projection onto $\mathcal{A}$ is negligible compared to the computational effort needed to perform the eigenvalue decompositions. We now describe how to compute $\mathcal{L}^\dagger$ in more detail; full details can be found in Appendix A.

Henceforth, we use $\mathrm{sHvec}(A_j) \in \mathbb{R}^{n^2}$ to denote a vectorization of the matrix A_j with respect to a fixed basis of the Hermitian matrix A_j, with the following order:

$$\begin{array}{l}(i) \text{ real part of the off-diagonal of } A_j;\\ (ii) \text{ imaginary part of the off-diagonal of } A_j;\\ (iii) \text{ the diagonal of } A_j.\end{array} \tag{6}$$

We now construct the matrix $M \in \mathbb{R}^{k \times n^2}$ by declaring

$$M^T = \left[\mathrm{sHvec}(A_1)\ \mathrm{sHvec}(A_2)\ \ldots\ \mathrm{sHvec}(A_k)\right]. \tag{7}$$

We then separate M into three blocks

$$M = \left[M_{\mathrm{Re}}\ \ M_{\mathrm{Im}}\ \ M_D\right],$$

where $M_D \in \mathbb{R}^{k \times n}$ has rows formed from the diagonals of matrices A_j, and M_{Re} and M_{Im} have rows formed from the real and imaginary parts of A_j, respectively, for $j = 1, \ldots, k$. Define now the matrices

$$\begin{array}{l} M_{\mathrm{Re\,Im}\,D} := \left[M_{\mathrm{Re}}\ -M_{\mathrm{Im}}\ M_D\right], \\ N_{\mathrm{Re\,Im}\,D} := \left[\frac{1}{\sqrt{2}}\begin{bmatrix} M_{\mathrm{Re}} & M_{\mathrm{Re}} & -M_{\mathrm{Im}} & -M_{\mathrm{Im}} \\ -M_{\mathrm{Im}} & M_{\mathrm{Im}} & -M_{\mathrm{Re}} & M_{\mathrm{Re}} \end{bmatrix}\begin{bmatrix} M_D & 0 \\ 0 & M_D \end{bmatrix}\right]. \end{array} \tag{8}$$

Permuting the rows and columns of $N_{\mathrm{Re\,Im}\,D}$ in a certain way, described in Appendix A we obtain a matrix denoted by $N_{final} \in \mathbb{R}^{2k \times 2n^2}$. Then $\mathcal{L}$ can be represented in coordinates (i.e. acting on a vectorization of P) in a surprisingly simple way, namely as a matrix:

$$L := \begin{bmatrix} I_{t(m-1)} \otimes N_{final} & 0 \\ 0 & \begin{bmatrix}\left[I_{m-1} \otimes M_{\mathrm{Re\,Im}\,D} \quad 0_{k(m-1),n^2}\right] \\ \left[e_m \otimes I_{n^2}\right]^T\end{bmatrix} \end{bmatrix}, \tag{9}$$

where $\otimes$ denotes the Kronecker product, and $t(n-1)$ denotes the triangular number $t(n-1) = \frac{n(n-1)}{2}$. Let the matrix $(M_{\mathrm{Re\,Im}\,D})_{null}$ have orthonormal columns that yield a basis for $\mathrm{null}(M_{\mathrm{Re\,Im}\,D})$, i.e.,

$$\mathrm{null}(M_{\mathrm{Re\,Im}\,D}) = \mathrm{range}((M_{\mathrm{Re\,Im}\,D})_{null}).$$

The generalized inverse of the top-left block is trivial to find from N_{final}. An explicit expression for the generalized inverse of the bottom right-block can

also be found. Therefore, we get an explicit blocked structure for the Moore-Penrose generalized inverse of the complete matrix representation.

$$L^\dagger = \begin{bmatrix} I_{t(m-1)} \otimes N_{final}^\dagger & 0 \\ 0 & \begin{bmatrix} I_{m-1} \otimes M_{\mathrm{Re\,Im}\,D}^\dagger & e_{m-1} \otimes (M_{\mathrm{Re\,Im}\,D})_{null} \\ -e_{m-1}^T \otimes M_{\mathrm{Re\,Im}\,D}^\dagger & I_{n^2} - (m-1)(M_{\mathrm{Re\,Im}\,D})_{null} \end{bmatrix} \end{bmatrix}, \tag{10}$$

as claimed. Thus $L^\dagger$ is easy to construct by simply stacking various small matrices together in blocks. Moreover, this means that both evaluations Lp and $L^\dagger R$ can be *vectorized* and evaluated efficiently and accurately in MATLAB.

3 Numerical experiments

In this section, we numerically illustrate the effectiveness of the projection and reflection methods for solving quantum channel construction problems. The large/huge problems were solved on an AMD Opteron(tm) Processor 6168, 1900.089 MHz cpu running LINUX. The smaller problems were solved using an Optiplex 9020, Intel(R) Core(TM), i7-4770 CPUs, 3.40GHz,3.40 GHz, RAM 16GB running Windows 7.

For simplicity of exposition, in our numerical experiments, we set $n = m$. Moreover, we will impose the common *unital constraint* $T(I_n) = I_n$ condition along with the trace preserving constraint. The class of unital quantum channels is considered in this work because of their practical importance as they "leave the maximal mixed state invariant", see e.g., [27]. Note that the mathematics behind the solution of (3) presented in this work does not require that the TPCP map to be found has to be unital or that $m = n$ must hold. In general, one can impose the additional linear constraint $T(\frac{1}{n}I_n) = \frac{1}{m}I_m$. We note in passing that the unital constraint implies that the last constraint in each density matrix block of constraints for each i is redundant. To generate random instances for our tests we proceed as follows. We start with given integers $m = n, k$ and a value for r. We generate a Choi matrix P using r random unitary matrices $F_i, i = 1, \ldots, r$ as the *Kraus operators* and a positive probability distribution d, i.e., from (1) and (2) we set the blocks of P to be

$$P_{st} = \sum_{j=1}^{r} d_j F_j E_{st} F_j^*.$$

The specific choice of *Kraus operators* (being unitary matrices) implies that the resulting quantum channel defined by P is a convex combination of unitary conjugations [30], thus guaranteeing that the quantum channel is unital. Note that, given a density matrix X, then the trace preserving completely positive map can now be evaluated using the blocked form of P in (2) as

$$T(X) = \sum_{st} X_{st} P_{st}.$$

We then generate random density matrices $A_j, j = 1, \ldots, k$ and set B_j as the image of the corresponding trace preserving completely positive map T on A_j, for all j. This guarantees that we have a feasible instance of rank r and larger/smaller r values result in larger/smaller rank for the feasible Choi matrix P.

Also, besides constructing TPCP maps of the form $T(X) = \sum_j d_j U_j X U_j^*$, for some probability vector $(p_1, ..., p_r)$ and unitaries $U_1, \ldots, U_r$, we can also construct a general T by choosing $G_1, \ldots, G_{r-1}$ randomly, and then set $F_i = \frac{1}{M_i} G_i$, for $i = 1, \ldots, r-1$, for sufficiently large $M_i > 0$. We get $Q = I - F_1 F_1^* - \ldots - F_{r-1} F_{r-1}^* \succeq 0$. Then we let F_r be any matrix satisfying $F_r F_r^* = Q$. The map $T(X) = \sum_{j=1}^r F_j X F_j^*$ is then a TPCP map and we can take arbitrary density matrices $A_1, \ldots, A_k$, and $B_i = T(A_i)$, for $i = 1, \ldots, k$, to test our program.

3.1 Solving the basic problem with DR

We first look at our basic feasibility problem (3). We illustrate the numerical results only using the DR algorithm since we found it to be vastly superior to MAP; see Section 3.2, below. We found solutions of huge problems with surprisingly high accuracy and very few iterations. The results are presented in Table 1. As in Section 3.2 below, we use a multiple of the identity $P_0 = mnI_{mn}$ as the starting point for DR. We give the size of the problem, the number of iterations, the norm of the residual $\|Lp - R\|$ after the projection on the PSD cone at the end, the maximum value of the cosine values of angles between successive iterates indicating the linear rate of convergence, and the total computational time to perform a projection on the PSD cone. The projection on the PSD cone dominates the time of the algorithm, i.e., the total time is roughly the number of iterations times the projection time. To fathom the size of the problems considered, observe that a problem with $m = n = 170$ finds a PSD matrix of order 3×10^4 which has approximately 4.5×10^8 variables. We reiterate that the solutions are found with extremely high accuracy in very few DR iterations. Moreover, the solutions found are usually of maximum rank unless r is relatively small compared to $m = n$. This can be seen specifically in the tests for maximum rank solutions in Table 2, i.e., full rank is found for $r = 30$ decreasing till $r = 14$ for fixed $m = n = 30, k = 16$.

3.2 Heuristic for finding max-rank feasible solutions using DR and MAP

We now look at the problem of finding *high rank feasible solutions*. Recall that this corresponds to finding a trace preserving completely positive map T mapping A_i to B_i, so that T necessarily has a long operator sum representation (1). We moreover use this section to compare the DR and MAP algorithms. Our numerical tests fix $m = n, k$ and then change the value of r, i.e., the value used to generate the test problems.

$m = n, k, r$	iters	norm-residual	max-cos	PSD-proj-CPUs
90,50,90	6	5.88e-15	.7014	233.8
100,60,90	7	7.243e-15	0.8255	821.7
110,65,90	7	7.983e-15	0.8222	1484
120,70,90	8	8.168e-15	0.8256	2583
130,75,90	8	7.19e-15	0.8288	3607
140,80,90	9	8.606e-15	0.8475	5832
150,85,90	11	8.938e-15	0.8606	6188
160,90,90	11	9.295e-15	0.8718	1.079e+04
170,95,90	12	9.412e-15	0.8918	1.139e+04

Table 1 Using DR algorithm; for solving huge feasibility problems; CPU time is for one projection

The heuristic for finding a large rank solution starts by finding a (current) feasible solution $P_i, i = 1$. We use a multiple of the identity as the starting point $P_0 = mnI_{mn}$ to find the feasible point P_i using DR. We repeatedly set our current point P_c to be the barycenter of all the t feasible points currently found $P_c = \frac{1}{t}\sum_i^t P_i$. The algorithm proceeds by changing the starting point to the *other side and outside* of the PSD cone, i.e., the new starting point to find a new $P_i, i > 1$, is found by traveling in direction $d = mnI_{mn} - \text{trace}(P_c)P_c$ starting from P_c so that the new starting point $P_0 := P_c + \alpha d$ is <u>not</u> PSD. For instance, we may set $\alpha = 2^j\|d\|^2$ for sufficiently large j. We then apply the DR algorithm with the new starting point to find a new P_i until we find the current barycenter matrix $P_c \succ 0$ or no increase in the rank of P_c occurs.

Again, we see that we find very accurate solutions and solutions of maximum rank. We find that DR is much more efficient both in the number of iterations in finding a feasible solution from a given starting point and in the number of steps in our heuristic needed to find a large rank solution. In Tables 2 and 3 we present the output for several values of r when using DR and MAP, respectively. We use a randomly generated feasibility instance for each value of r but we start MATLAB with the *rng(default)* settings so the same random instances are generated. We note that the DR algorithm is successful for finding a maximum rank solution and usually after only the first step of the heuristic. The last three $r = 12, 10, 8$ values required $8, 9, 12$ steps, respectively. However, the final P solution was obtained to (a high) 9 decimal accuracy.

The MAP always requires many more iterations and at least two steps for the maximum rank solution. It then fails completely once $r \leq 12$. In fact, it reaches the maximum number of iterations while only finding a feasible solution to 3 decimals accuracy for $r = 12$ and then 2 decimals accuracy for $r = 10, 8$. We see that the cosine value has reached 1 for $r = 12, 10, 8$ and the MAP algorithm was making no progress towards convergence.

For each value of r we include:

1. the number of steps of DR that it took to find the max-rank P;

2. the minimum/maximum/mean number of iterations for the steps in finding P [1];
3. the maximum of the cosine of the angles between three successive iterates [2];
4. the value of the maximum rank found. [3]

	rank steps	min-iters	max-iters	mean-iters	max-cos	max rank
$r = 30$	1	6	6	6	7.008801e-01	900
$r = 28$	1	7	7	7	7.323953e-01	900
$r = 26$	1	7	7	7	7.550174e-01	900
$r = 24$	1	8	8	8	7.911440e-01	900
$r = 22$	1	9	9	9	8.238539e-01	900
$r = 20$	1	9	9	9	8.454781e-01	900
$r = 18$	1	11	11	11	8.730321e-01	900
$r = 16$	1	15	15	15	8.995266e-01	900
$r = 14$	1	23	23	23	9.288445e-01	900
$r = 12$	8	194	3500	1.916375e+03	9.954262e-01	900
$r = 10$	9	506	3500	2.605778e+03	9.968120e-01	900
$r = 8$	12	2298	3500	3.350833e+03	9.986002e-01	900

Table 2 Using DR algorithm; with $[m\ n\ k\ mn\ toler\ iterlimit] = [30\ 30\ 16\ 900\ 1e{-}14\ 3500]$; max/min/mean iter and number rank steps for finding max-rank of P. The 3500 here means 9 decimals accuracy attained for last step.

	rank steps	min-iters	max-iters	mean-iters	max-cos	max rank
$r = 30$	2	55	67	61	8.233188e-01	900
$r = 28$	2	65	77	71	8.513481e-01	900
$r = 26$	2	78	89	8.350000e+01	8.754098e-01	900
$r = 24$	2	100	109	1.045000e+02	9.040865e-01	900
$r = 22$	2	124	130	127	9.250665e-01	900
$r = 20$	2	156	158	157	9.432779e-01	900
$r = 18$	2	239	245	242	9.689567e-01	900
$r = 16$	2	388	407	3.975000e+02	9.847052e-01	900
$r = 14$	2	1294	1369	1.331500e+03	9.980012e-01	900
$r = 12$	2	3500	3500	3500	1.000000e+00	493
$r = 10$	2	3500	3500	3500	1.000000e+00	483
$r = 8$	2	3500	3500	3500	1.000000e+00	475

Table 3 Using MAP algorithm; with $[m\ n\ k\ mn\ toler\ iterlimit] = [30\ 30\ 16\ 900\ 1e-14\ 3500]$; max/min/mean iter and number rank steps for finding max-rank of P. The 3500 mean-iters means max iterlimit reached; low accuracy attained.

[1] Note that if the maximum value is the same as *iterlimit*, then the method failed to attain the desired accuracy *toler* for this particular value of r.

[2] This is a good indicator of the expected number of iterations.

[3] We used the *rank* function in MATLAB with the default tolerance, i.e., rank(P) is the number of singular values of P that are larger than $mn * eps(\|P\|)$, where $eps(\|P\|)$ is the positive distance from $\|P\|$ to the next larger in magnitude floating point number of the same precision. Here we note that we did not fail to find a max-rank solution with the DR algorithm.

3.3 Heuristic for finding low rank and rank constrained solutions

In quantum information science, one might want to obtain a feasible Choi matrix solution $P = (P_{ij})$ with low rank, e.g., [30, Section 4.1]. If we have a bound on the rank, then we could change the algorithm by adding a rank restriction when one projects the current iterate of $P = (P_{ij})$ onto the PSD cone. That is instead of taking the positive part of $P = (P_{ij})$, we take the *nonconvex projection*

$$P_r := \sum_{j \le r, \lambda_j > 0} \lambda_j x_j x_j^*,$$

where P has spectral decomposition $\sum_{j=1}^{mn} \lambda_j x_j x_j^*$ with $\lambda_1 \ge \cdots \ge \lambda_{mn}$.

Alternatively, we can do the following. Suppose a feasible Choi matrix $C(T) = P_c = ((P_c)_{ij})$ is found with $\operatorname{rank}(P_c) = r$. We can then attempt to find a new Choi matrix of smaller rank restricted to the face F of the PSD cone where the current P_c is in the relative interior of F, i.e., the minimal face of the PSD cone containing P_c. We do this using facial reduction, e.g., [7,8]. More specifically, suppose that $P_c = VDV^*$ is a compact spectral decomposition, where $D \in \mathcal{H}^r_{++}$ is diagonal, positive definite and has rank r. Then the minimal face F of the PSD cone containing P_c has the form $F = V\mathcal{H}^r_+V^*$. Recall $Lp = b$ denotes the matrix/vector equation corresponding to the linear constraints in our basic problem with $p = \operatorname{sHvec}(P)$, the vectorization of the Hermitian matrix P described in (6). Let $L_{i,:}$ denote the rows of the matrix representation L. We let $\operatorname{sHMat} = \operatorname{sHvec}^{-1}$ be the inverse mapping from a complex vector to a Hermitian matrix. Note that $\operatorname{sHMat} = \operatorname{sHvec}^*$, i.e., the inverse and adjoint are the same. Then each row of the equation $Lp = b$ is equivalent to

$$\langle L^*_{i,:}, \operatorname{sHvec}(P)\rangle = \langle \operatorname{sHMat}(L^*_{i,:}), V\bar{P}V^*\rangle = \langle V^* \operatorname{sHMat}(L^*_{i,:})V, \bar{P}\rangle, \quad \bar{P} \in \mathcal{H}^r_+.$$

Therefore, we can replace the linear constraints with the smaller system $\bar{L}\bar{p} = b$ with equations $\langle \bar{L}_{i,:}, \bar{p}\rangle$, where $\bar{L}_{i,:} = \operatorname{sHvec}\left(V^* \operatorname{sHMat}(L^*_{i,:})V\right)$. In addition, since the current feasible point P_c is in the relative interior of the face $V\mathcal{H}^r_+V^*$, if we start outside the PSD cone $\mathcal{H}^r_+$ for our feasibility search, then we get a singular feasible $\bar{P}$ if one exists and so have reduced the rank of the corresponding initial feasible P. We then repeat this process as long as we get a reduction in the rank.

The MAP approach we are using appears to be especially well suited for finding low rank solutions. In particular, the facial reduction works well because we are able to get extremely high accuracy feasible solutions before applying the compact spectral decomposition. If the initial P_0 that is projected onto the affine subspace is not positive semidefinite, then successive iterates on the affine subspace stay outside the semidefinite cone, i.e., we obtain a final feasible solution $\bar{P}$ that is not positive definite if one exists. Therefore, the rank of $\bar{V}\bar{V}^*$ is reduced from the rank of P. The code for this has been surprisingly successful in reducing rank. We provide some typical results for small problems in Table 4. We start with a small rank (denoted by r) feasible solution that is used to generate a feasible problem. Therefore, we know that

the minimal rank is $\leq r$. We then repeatedly solve the problem using facial reduction until a positive definite solution is found which means we cannot continue with the facial reduction. Note that we could restart the algorithm using an upper bound for the rank obtained from the last rank we obtained.

$m = n, k, r$	facial red. ranks	final norm-residual
12,10,11	100,50,44,39	1.836e-15
12,10,10	92,61,43,44	1.786e-15
20,14,20	304,105,71	9.648e-15
22,13,20	374,121,75	9.746e-15

Table 4 Using MAP algorithm with facial reduction for decreasing the rank. Requested tolerance 1e-14 satisfied at each step.

Finally, our tests indicate that the rank constrained problem, which is nonconvex, often can be solved efficiently. Moreover, this problem helps in further reducing the rank. To see this, suppose that we know a bound, *rbnd*, on the rank of a feasible P. Then, as discussed above, we change the projection onto the PSD cone by using only the largest *rbnd* eigenvalues of P. In our tests, if we use r, the value from generating our instances, then we were always successful in finding a feasible solution of rank r. Our final tests appear in Tables 5 and 6. We generate problems with initial rank r. We then start solving a constrained rank problem with starting constraint rank r_s and decrease this rank by 1 until we can no longer find a feasible solution; the final rank with a feasible solution is r_f. At each successful reduction we found a feasible solution to the requested tolerance $1e-14$.

$m = n, k, r$	starting/final constr. ranks, r_s/r_f	
12,9,15	20	7
25,16,35	45	19
30,21,38	48	27

Table 5 Using DR algorithm for rank constrained problems with ranks r_s to r_f. Requested tolerance 1e-14 satisfied at each step. See Table 6 for details on $12, 9, 15$ instance one.

Table 6 illustrates the DR algorithm for finding a low rank solution for the first instance in Table 5. We begin with starting rank 20. We see the increase in max-cos and simultaneously the number of iterations needed to find a feasible solution as the rank constraint decreases. We stop in reducing rank once we cannot find a feasible solution with the iteration limit for DR set at 3,500.

4 Conclusion

In this paper, we studied the basic problem of constructing a quantum channel that maps between given sets of quantum states. We have used the Choi

current constrained rank	max-cos	norm(residual)	iterations
20	9.5183e-01	8.6510e-15	6.4700e+02
19	9.4773e-01	9.1083e-15	6.9600e+02
18	9.5347e-01	9.8330e-15	7.4700e+02
17	9.5947e-01	9.6879e-15	8.2300e+02
16	9.6289e-01	9.9593e-15	8.9700e+02
15	9.7182e-01	9.4914e-15	9.9700e+02
14	9.7775e-01	9.3193e-15	1.1670e+03
13	9.7630e-01	9.8646e-15	1.2830e+03
12	9.8125e-01	9.6170e-15	1.4250e+03
11	9.8389e-01	9.8741e-15	1.6660e+03
10	9.8834e-01	9.8033e-15	1.9860e+03
9	9.9109e-01	9.9461e-15	2.4430e+03
8	9.9260e-01	9.1184e-15	2.9920e+03
7	9.9704e-01	4.5293e-13	3.5000e+03
6	9.9960e-01	1.5008e-05	3.5000e+03

Table 6 Using DR algorithm for rank constrained problem instance one in Table 5 with $m = n = 12, k = 9, r = 15$ and starting constrained rank 20 till final successful constrained rank 7; feasibility failed for constrained rank 6 with iteration limit 3,500.

matrix representation for completely positive maps to show that the construction is equivalent to solving a Hermitian positive semidefinite linear feasibility problem. This feasibility problem has special structure that can be exploited. We have shown the efficiency of using alternating projection and Douglas-Rachford projection/reflection algorithms for accurately solving large scale problems to high accuracy. This included finding trace preserving completely positive, TPCP, maps with high rank, as well as the nonconvex problems of finding TPCP maps with low rank.

References

1. Aragón Artacho, F., Borwein, J., Tam, M.: Recent results on Douglas-Rachford methods for combinatorial optimization problems. Journal of Optimization Theory and Applications pp. 1–30 (2013). DOI 10.1007/s10957-013-0488-0. URL http://dx.doi.org/10.1007/s10957-013-0488-0
2. Bauschke, H., Borwein, J.: On the convergence of von Neumann's alternating projection algorithm for two sets. Set-Valued Anal. **1**(2), 185–212 (1993). DOI 10.1007/BF01027691. URL http://dx.doi.org/10.1007/BF01027691
3. Bauschke, H., Borwein, J.: On projection algorithms for solving convex feasibility problems. SIAM Rev. **38**(3), 367–426 (1996). DOI 10.1137/S0036144593251710. URL http://dx.doi.org/10.1137/S0036144593251710
4. Bauschke, H., Combettes, P., Luke, D.: Phase retrieval, error reduction algorithm, and Fienup variants: a view from convex optimization. J. Opt. Soc. Amer. A **19**(7), 1334–1345 (2002). DOI 10.1364/JOSAA.19.001334. URL http://dx.doi.org.proxy.lib.uwaterloo.ca/10.1364/JOSAA.19.001334
5. Bauschke, H., Luke, D., Phan, H., Wang, X.: Restricted normal cones and the method of alternating projections: Theory. Set-Valued and Variational Anal. pp. 1–43 (2013). DOI 10.1007/s11228-013-0239-2. URL http://dx.doi.org/10.1007/s11228-013-0239-2
6. Borwein, J., Sims, B.: The Douglas-Rachford algorithm in the absence of convexity. In: Fixed-point algorithms for inverse problems in science and engineering, *Springer Optim. Appl.*, vol. 49, pp. 93–109. Springer, New York (2011). DOI 10.1007/978-1-4419-9569-8_6. URL http://dx.doi.org.proxy.lib.uwaterloo.ca/10.1007/978-1-4419-9569-8_6

7. Borwein, J., Wolkowicz, H.: Facial reduction for a cone-convex programming problem. J. Austral. Math. Soc. Ser. A **30**(3), 369–380 (1980/81)
8. Borwein, J., Wolkowicz, H.: Regularizing the abstract convex program. J. Math. Anal. Appl. **83**(2), 495–530 (1981)
9. Bregman, L.: The method of successive projection for finding a common point of convex sets. Sov. Math. Dokl **6**, 688–692 (1965)
10. Chefles, A., Jozsa, R., Winter, A.: On the existence of physical transformations between sets of quantum states. Int. J. Quant. Inf. **2**, 11–21 (2004)
11. Choi, M.: Completely positive linear maps on complex matrices. Linear Algebra and Appl. **10**, 285–290 (1975)
12. Demmel, J., Marques, O., Parlett, B., Vömel, C.: Performance and accuracy of LAPACK's symmetric tridiagonal eigensolvers. SIAM J. Sci. Comput. **30**(3), 1508–1526 (2008). DOI 10.1137/070688778. URL http://dx.doi.org.proxy.lib.uwaterloo.ca/10.1137/070688778
13. Drusvyatskiy, D., Ioffe, A., Lewis, A.: Alternating projections and coupling slope (2014)
14. Duffin, R.: Infinite programs. In: A. Tucker (ed.) Linear Equalities and Related Systems, pp. 157–170. Princeton University Press, Princeton, NJ (1956)
15. Eckart, C., Young, G.: The approximation of one matrix by another of lower rank. Psychometrika **1**, 211–218 (1936)
16. Elser, V., Rankenburg, I., Thibault, P.: Searching with iterated maps. Proceedings of the National Academy of Sciences **104**(2), 418–423 (2007). DOI 10.1073/pnas.0606359104. URL http://www.pnas.org/content/104/2/418.abstract
17. Escalante, R., Raydan, M.: Alternating projection methods, *Fundamentals of Algorithms*, vol. 8. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (2011). DOI 10.1137/1.9781611971941. URL http://dx.doi.org/10.1137/1.9781611971941
18. Fung, C.H., Li, C.K., Sze, N.S., Chau, H.: Conditions for degradability of tripartite quantum states. Tech. Rep. arXiv:1308.6359, University of Hong Kong (2012)
19. Hesse, R., Luke, D., Neumann, P.: Alternating projections and douglas-rachford for sparse affine feasibility. Signal Processing, IEEE Transactions on **62**(18), 4868–4881 (2014). DOI 10.1109/TSP.2014.2339801
20. Huang, Z., Li, C.K., Poon, E., Sze, N.S.: Physical transformations between quantum states. J. Math. Phys. **53**(10), 102,209, 12 (2012). DOI 10.1063/1.4755846. URL http://dx.doi.org/10.1063/1.4755846
21. Kraus, K.: States, effects, and operations: Fundamental notions of quantum theory. Lecture Notes in Physics, Springer-Verlag, Berlin **190** (1983)
22. Lewis, A., Luke, D., Malick, J.: Local linear convergence for alternating and averaged nonconvex projections. Found. Comput. Math. **9**(4), 485–513 (2009). DOI 10.1007/s10208-008-9036-y. URL http://dx.doi.org/10.1007/s10208-008-9036-y
23. Lewis, A., Malick, J.: Alternating projections on manifolds. Math. Oper. Res. **33**(1), 216–234 (2008). DOI 10.1287/moor.1070.0291. URL http://dx.doi.org/10.1287/moor.1070.0291
24. Li, C.K., Poon, Y.T.: Interpolation by completely positive maps. Linear Multilinear Algebra **59**(10), 1159–1170 (2011). DOI 10.1080/03081087.2011.585987. URL http://dx.doi.org/10.1080/03081087.2011.585987
25. Li, C.K., Poon, Y.T., Sze, N.S.: Higher rank numerical ranges and low rank perturbations of quantum channels. J. Math. Anal. Appl. **348**(2), 843–855 (2008). DOI 10.1016/j.jmaa.2008.08.016. URL http://dx.doi.org/10.1016/j.jmaa.2008.08.016
26. Lions, P., Mercier, B.: Splitting algorithms for the sum of two nonlinear operators. SIAM Journal on Numerical Analysis **16**(6), 964–979 (1979). DOI 10.1137/0716071. URL http://dx.doi.org/10.1137/0716071
27. Mendl, C., Wolf, M.: Unital quantum channels—convex structure and revivals of Birkhoff's theorem. Comm. Math. Phys. **289**(3), 1057–1086 (2009). DOI 10.1007/s00220-009-0824-2. URL http://dx.doi.org.proxy.lib.uwaterloo.ca/10.1007/s00220-009-0824-2
28. Nielsen, M., Chuang, I. (eds.): Quantum Computation and Quantum Information. Cambridge University Press (2000)
29. Phan, H.M.: Linear convergence of the douglas-rachford method for two closed sets (2014)

30. Watrous, J.: Distinguishing quantum operations having few kraus operators. Quant. Inf. Comp. **8**, 819–833 (2008). URL http://arxiv.org/abs/0710.0902

Acknowledgements We would like to thank the editors and referees for their careful reading and helpful comments on the paper.

A Background

A.1 Matrix representation of $\mathcal{L}$ and $\mathcal{L}^\dagger$

In this section, we show that the matrices L in (9) and $L^\dagger$ in (10) are indeed matrix representations of the linear map $\mathcal{L}$ (defined in (5)) and its Moore-Penrose generalized inverse, respectively, under a specific choice of basis of the vector space $\mathcal{H}^{nm}$.

A.1.1 Choice of orthonormal basis of $\mathcal{H}^s$

We choose the standard orthonormal basis for the vector space $\mathcal{H}^\ell$ of $\ell \times \ell$ Hermitian matrices (over the reals) as follows. Let $e_j \in \mathbb{R}^\ell$ be the j-th standard unit vector for $j = 1, \ldots, \ell$. Then $e_i e_j^T \in \mathbb{R}^{\ell \times \ell}$ is zero everywhere except the (i,j)-th entry, which is 1. For $i, j = 1, \ldots, \ell$, define the (i,j)-th basis matrix as follows:

$$E_{ij} = \begin{cases} \frac{1}{\sqrt{2}}(e_i e_j^T + e_j e_i^T) & \text{if } i < j, \\ \frac{\mathsf{i}}{\sqrt{2}}(e_j e_i^T - e_i e_j^T) & \text{if } i > j, \\ e_j e_j^T & \text{if } i = j. \end{cases} \tag{11}$$

Then $\mathcal{E}_{\text{real,offdiag}} \cup \mathcal{E}_{\text{imag,offdiag}} \cup \mathcal{E}_{\text{diag}}$ forms an orthonormal basis of $\mathcal{H}^\ell$, where
- $\mathcal{E}_{\text{real,offdiag}} := \{E_{ij} : 1 \le i < j \le \ell\}$ collects the real zero-diagonal basis matrices,
- $\mathcal{E}_{\text{imag,offdiag}} := \{E_{ij} : 1 \le j < i \le \ell\}$ collects the imaginary zero-diagonal basis matrices, and
- $\mathcal{E}_{\text{diag}} := \{E_{jj} : 1 \le j \le \ell\}$ collects the real diagonal basis matrices.

We define a total ordering $<$ on the tuples (i,j) for $i, j = 1, \ldots, \ell$, so that the matrices are ordered with $\mathcal{E}_{\text{real,offdiag}} < \mathcal{E}_{\text{imag,offdiag}} < \mathcal{E}_{\text{diag}}$ in the element-wise sense (as stated in (6)). For any $(i,j), (\tilde{i}, \tilde{j}) \in \{1, \ldots, \ell\}^2$, we say that $(i,j) \prec (\tilde{i}, \tilde{j})$ if one of the following holds.
- Case 1: $i < j$ (so that E_{ij} is a real matrix with zero diagonal).
  - $i < j$ and $\tilde{i} \ge \tilde{j}$.
  - $i < j$ and $\tilde{i} < \tilde{j}$, but $\tilde{j} > j$.
  - $i < j$ and $\tilde{i} < \tilde{j} = j$, but $\tilde{i} > i$.
- Case 2: $i > j$ (so that E_{ij} is a imaginary matrix with zero diagonal).
  In this case we must have $\tilde{i} \ge \tilde{j}$.
  - $j < i$ and $\tilde{j} = \tilde{i}$.
  - $j < i$ and $\tilde{j} < \tilde{i}$, but $\tilde{i} > i$.
  - $j < i$ and $\tilde{j} < \tilde{i} = i$, but $\tilde{j} > j$.
- Case 3: $i = j$ (so that E_{jj} is a real diagonal matrix).
  In this case we must have $\tilde{i} = \tilde{j}$.
  - $j < \tilde{j}$.

For instance, when $\ell = 3$, our orthonormal basis of choice is given in the following order:

$$E_{12} = \tfrac{1}{\sqrt{2}}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad E_{13} = \tfrac{1}{\sqrt{2}}\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad E_{23} = \tfrac{1}{\sqrt{2}}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix},$$

$$E_{21} = \tfrac{1}{\sqrt{2}}\begin{bmatrix} 0 & \mathsf{i} & 0 \\ -\mathsf{i} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad E_{31} = \tfrac{1}{\sqrt{2}}\begin{bmatrix} 0 & 0 & \mathsf{i} \\ 0 & 0 & 0 \\ -\mathsf{i} & 0 & 0 \end{bmatrix}, \quad E_{32} = \tfrac{1}{\sqrt{2}}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \mathsf{i} \\ 0 & -\mathsf{i} & 0 \end{bmatrix},$$

$$E_{11} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad E_{22} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad E_{33} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

We also work with $nm \times nm$ *block* matrices, with each block of size $m \times m$. On the space $\mathcal{H}^{nm}$, for any $1 \le i, j \le nm$, let

$$s := \left\lceil \frac{i}{m} \right\rceil, \quad t := \left\lceil \frac{j}{m} \right\rceil, \quad p := i - m(s-1), \quad \text{and} \quad q := j - m(t-1). \tag{12}$$

Note that $1 \le s, t \le n$ are the block indices and $1 \le p, q \le m$ are the intra-block indices. For instance, consider the matrix $e_i e_j^T \in \mathcal{H}^{nm}$ (having only 1 nonzero entry at position (i, j)). The nonzero entry is at the (p, q)-th entry in the (s, t)-th block (which is of size $m \times m$). The orthonormal matrices E_{ij} defined in (11) are related to the block indices (s, t) as described in Table 7.

	The nonzero $m \times m$ blocks of E_{ij}
Case 1: $s < t$	$(E_{ij})_{st} = \frac{1}{\sqrt{2}} e_p e_q^T$ $(E_{ij})_{ts} = \frac{1}{\sqrt{2}} e_q e_p^T$
Case 2: $s > t$	$(E_{ij})_{st} = -\frac{\mathrm{i}}{\sqrt{2}} e_p e_q^T$ $(E_{ij})_{ts} = \frac{\mathrm{i}}{\sqrt{2}} e_q e_p^T$
Case 3: $s = t$	$(E_{ij})_{ss} = E_{pq}$

Table 7 The nonzero blocks of E_{ij} for any fixed $i, j = 1, \ldots, nm$. (s, t, p, q) are defined in terms of (i, j) as in (12). All blocks unspecified in the table are zero blocks.

Unlike $\mathcal{H}^s$, we order the blocked orthonormal matrices E_{ij} in $\mathcal{H}^{nm}$ via the following total ordering $<$ on the set $\mathcal{I} := \{(i, j) : 1 \le i, j \le nm\}$. Let $1 \le i, j, \tilde{i}, \tilde{j} \le nm$ with $(i, j) \ne (\tilde{i}, \tilde{j})$. Then letting

$$\begin{aligned} s &:= \left\lceil \tfrac{i}{m} \right\rceil, \ t := \left\lceil \tfrac{j}{m} \right\rceil, \ p = i - m(s-1), \ q = j - m(t-1), \\ \tilde{s} &:= \left\lceil \tfrac{\tilde{i}}{m} \right\rceil, \ \tilde{t} := \left\lceil \tfrac{\tilde{j}}{m} \right\rceil, \ \tilde{p} = \tilde{i} - m(\tilde{s}-1), \ \tilde{q} = \tilde{j} - m(\tilde{t}-1), \end{aligned}$$

the relation $(i, j) < (\tilde{i}, \tilde{j})$ holds if and only if one of the following holds.

- $\{p, q\} \ne \{\tilde{p}, \tilde{q}\}$ and $(\min\{p, q\}, \ \max\{p, q\}) \prec (\min\{\tilde{p}, \tilde{q}\}, \ \max\{\tilde{p}, \tilde{q}\})$.
- $\{p, q\} = \{\tilde{p}, \tilde{q}\}$ and $(s, t) \prec (\tilde{s}, \tilde{t})$.
- $\{p, q\} = \{\tilde{p}, \tilde{q}\}$, $(s, t) = (\tilde{s}, \tilde{t})$ and $p < q$. (Then $\tilde{q} = p < q = \tilde{p}$.)

In other words, we order the 2-tuples (i, j) in $\mathcal{I}$ by grouping all those with the same intra-block index (p, q) and block indices $s < t$ together, for some $p < q$, followed by those tuples with intra-block index (q, p) and block indices $s < t$. As an example, when $m = 2$ and $n = 3$, the following list gives the first few entries of $\mathcal{I}$:

$$(1,4), \ (1,6), \ (3,6), \ (2,3), \ (2,5), \ (4,5), \ (1,2), \ (3,4), \ \ldots$$

These first 2-tuples (i, j) in $\mathcal{I}$ have the corresponding 4-tuples (s, t, p, q) defined as in (12), given as follows:

$$(1,2,1,2), \ (1,3,1,2), \ (2,3,1,2), \ (1,2,2,1), \ (1,3,2,1), \ (2,3,2,1), \ (1,1,1,2), \ (2,2,1,2), \ \ldots$$

Note that the first three 2-tuples have the same intra-block index $(p, q) = (1, 2)$. The immediately following three 2-tuples have the intra-block index $(q, p) = (2, 1)$, and so on.

A.1.2 Symmetric vectorization of Hermitian matrices

Using the ordered orthonormal basis of $\mathcal{H}^s$ described in (11) in Section A.1.1, we can define the corresponding *symmetric vectorization* of Hermitian matrices. Since any Hermitian matrix in $\mathcal{H}^s$ can be expressed as a unique linear combination of the orthonormal basis matrices E_{ij}, the map

$$\mathrm{sHvec} : \mathcal{H}^s \to \mathbb{R}^{s^2} : H \mapsto v,$$

where $v \in \mathbb{R}^{s^2}$ is the unique vector such that $H = \sum_{i,j=1}^{s} v_{ij} E_{ij}$, is well-defined. The map sHvec is a linear isometry (i.e., sHvec is a linear map and $\|\operatorname{sHvec}(H)\|^2 = \operatorname{trace}(H^2)$ for all $H \in \mathcal{H}^s$), and its adjoint is given by

$$\operatorname{sHMat} : \mathbb{R}^{s^2} \to \mathcal{H}^s : v \mapsto \sum_{i,j=1}^{s} v_{ij} E_{ij}, \tag{13}$$

which is also the inverse map of sHvec. For instance,

$$\operatorname{sHvec}\left(\begin{bmatrix} 1 & \sqrt{2}-\mathsf{i} \\ \sqrt{2}+\mathsf{i} & 3 \end{bmatrix}\right) = \begin{bmatrix} 2 & -\sqrt{2} & 1 & 3 \end{bmatrix}^T.$$

A.1.3 Ordering the rows and columns in the matrix representation of $\mathcal{L}$

In the following, we compute matrix representations L_A and L_T of the linear maps $\mathcal{L}_A : \mathcal{H}^{nm} \to \otimes_{j=1}^{m} \mathcal{H}^m$ and $\mathcal{L}_T : \mathcal{H}^{nm} \to \mathcal{H}^n$, respective. The matrix representation $\mathcal{L} = (\mathcal{L}_A(\cdot), \mathcal{L}_T(\cdot))$ is then chosen to be $L = \begin{bmatrix} L_A \\ L_T \end{bmatrix}$, with $L_A \in \mathbb{R}^{km^2 \times n^2 m^2}$ and $L_T \in \mathbb{R}^{m^2 \times n^2 m^2}$.

Any matrix representation for the linear map $\mathcal{L}_A$ (resp. $\mathcal{L}_T$) depends on the choice of the *ordered* orthonormal bases for $\mathcal{H}^{nm}$ and for $\mathcal{H}^m \times \cdots \times \mathcal{H}^m$ (resp. for $\mathcal{H}^{nm}$ and for $\mathcal{H}^m$). For $\mathcal{H}^{nm}$, we use the ordered orthonormal basis defined on Page 16 in Section A.1.1. For $\mathcal{H}^m \times \cdots \times \mathcal{H}^m$, we use the orthonormal basis

$$\begin{array}{llll} (E_{12}, 0, \ldots, 0), & (0, E_{12}, \ldots, 0), & \ldots, & (0, 0, \ldots, E_{12}), \\ (E_{21}, 0, \ldots, 0), & (0, E_{21}, \ldots, 0), & \ldots, & (0, 0, \ldots, E_{21}), \\ (E_{13}, 0, \ldots, 0), & (0, E_{13}, \ldots, 0), & \ldots, & (0, 0, \ldots, E_{13}), \\ (E_{31}, 0, \ldots, 0), & (0, E_{31}, \ldots, 0), & \ldots, & (0, 0, \ldots, E_{31}), \ \ldots, \ (0, 0, \ldots, E_{mm}). \end{array} \tag{14}$$

We first construct a matrix representation L_A of $\mathcal{L}_A$ by rows. Recall that

$$\mathcal{L}_A(P) = \left(\sum_{s,t=1}^{n} (A_1)_{st} P_{st}, \ \sum_{s,t=1}^{n} (A_2)_{st} P_{st}, \ \ldots, \ \sum_{s,t=1}^{n} (A_k)_{st} P_{st} \right)$$

for any $P \in \mathcal{H}^{nm}$, so the rows of $\mathcal{L}_A$ are determined by the linear functionals

$$\begin{cases} \mathbf{L}_{\ell,p,q,\mathrm{Re}}(P) := \sqrt{2}\mathrm{Re}\left(\sum_{s,t=1}^{n} (A_\ell)_{st} (P_{st})\right)_{pq} \\ \mathbf{L}_{\ell,p,q,\mathrm{Im}}(P) := \sqrt{2}\mathrm{Im}\left(\sum_{s,t=1}^{n} (A_\ell)_{st} (P_{st})\right)_{pq} \\ \mathbf{D}_{\ell,q}(P) := \left(\sum_{s,t=1}^{n} (A_\ell)_{st} (P_{st})\right)_{qq} \end{cases}$$

for some $\ell \in \{1, \ldots, k\}$ and $p < q \in \{1, \ldots, m\}$. Defining vectors $\alpha_{\ell,p,q,\mathrm{Re}}, \alpha_{\ell,p,q,\mathrm{Im}}, \beta_{\ell,q} \in \mathbb{R}^{(nm)^2}$ by

$$\begin{cases} \mathbf{L}_{\ell,p,q,\mathrm{Re}}(P) = (\alpha_{\ell,p,q,\mathrm{Re}})^T \operatorname{sHvec}(P), \\ \mathbf{L}_{\ell,p,q,\mathrm{Im}}(P) = (\alpha_{\ell,p,q,\mathrm{Im}})^T \operatorname{sHvec}(P), \\ \mathbf{D}_{\ell,q}(P) = (\beta_{\ell,q})^T \operatorname{sHvec}(P), \end{cases}$$

for all $\ell \in \{1, \ldots, k\}$, $p < q \in \{1, \ldots, m\}$, we get that

$$L_A = \begin{bmatrix} \alpha_{1,1,2,\mathrm{Re}} \cdots \alpha_{k,1,2,\mathrm{Re}} & \alpha_{1,1,2,\mathrm{Im}} \cdots \alpha_{k,m,m,\mathrm{Im}} & \beta_{1,1} \cdots \beta_{k,1} & \beta_{1,2} \cdots \beta_{k,m} \end{bmatrix}^T.$$

Now we proceed to find the vectors $\alpha_{\ell,p,q,\mathrm{Re}}, \alpha_{\ell,p,q,\mathrm{Im}}, \beta_{\ell,q}$.

A.1.4 Computing the rows of L_A

Fix any $\ell \in \{1,\dots,k\}$. For any $i,j \in \{1,\dots,nm\}$, let (s,t,p,q) be defined as in (12).

If $s<t$, then using Table 7 we get

$$\begin{aligned}\sum_{\tilde{s},\tilde{t}=1}^{n}(A_\ell)_{\tilde{s}\tilde{t}}(E_{ij})_{\tilde{s}\tilde{t}} &= \frac{1}{\sqrt{2}}\left((A_\ell)_{st}e_pe_q^T+(A_\ell)_{ts}e_qe_p^T\right)\\ &= \frac{1}{\sqrt{2}}\left(\mathrm{Re}\,(A_\ell)_{st}(e_pe_q^T+e_qe_p^T)+\mathsf{i}\,\mathrm{Im}\,(A_\ell)_{st}(e_pe_q^T-e_qe_p^T)\right)\end{aligned}$$

If $s>t$, then

$$\begin{aligned}\sum_{\tilde{s},\tilde{t}=1}^{n}(A_\ell)_{\tilde{s}\tilde{t}}(E_{ij})_{\tilde{s}\tilde{t}} &= \frac{\mathsf{i}}{\sqrt{2}}\left(-(A_\ell)_{st}e_pe_q^T+(A_\ell)_{ts}e_qe_p^T\right)\\ &= \frac{1}{\sqrt{2}}\left(-\mathrm{Im}\,(A_\ell)_{ts}(e_pe_q^T+e_qe_p^T)-\mathsf{i}\,\mathrm{Re}\,(A_\ell)_{ts}(e_pe_q^T-e_qe_p^T)\right)\end{aligned}$$

If $s=t$, then

$$\sum_{\tilde{s},\tilde{t}=1}^{n}(A_\ell)_{\tilde{s}\tilde{t}}(E_{ij})_{\tilde{s}\tilde{t}} = (A_\ell)_{ss}E_{pq}.$$

Fix any $\ell=1,\dots,k$ and $\hat{p}<\hat{q}$ from $\{1,\dots,m\}$. Then for all $i,j\in\{1,\dots,nm\}$, defining (s,t,p,q) as in (12), we have

$$\mathbf{L}_{\ell,\hat{p},\hat{q},\mathrm{Re}}(E_{ij}) = \begin{cases}\mathrm{Re}\,(A_\ell)_{st} & \text{if } s<t \text{ and } \{p,q\}=\{\hat{p},\hat{q}\}\\ -\mathrm{Im}\,(A_\ell)_{ts} & \text{if } s>t \text{ and } \{p,q\}=\{\hat{p},\hat{q}\}\\ (A_\ell)_{ss} & \text{if } s=t,\ p<q \text{ and } (p,q)=(\hat{p},\hat{q})\\ 0 & \text{otherwise,}\end{cases}$$

and

$$\mathbf{L}_{\ell,\hat{p},\hat{q},\mathrm{Im}}(E_{ij}) = \begin{cases}\mathrm{Im}\,(A_\ell)_{st} & \text{if } s<t \text{ and } (p,q)=(\hat{p},\hat{q})\\ -\mathrm{Im}\,(A_\ell)_{st} & \text{if } s<t \text{ and } (p,q)=(\hat{q},\hat{p})\\ -\mathrm{Re}\,(A_\ell)_{ts} & \text{if } s>t \text{ and } (p,q)=(\hat{p},\hat{q})\\ \mathrm{Re}\,(A_\ell)_{ts} & \text{if } s>t \text{ and } (p,q)=(\hat{q},\hat{p})\\ (A_\ell)_{ss} & \text{if } s=t,\ p>q \text{ and } (p,q)=(\hat{q},\hat{p})\\ 0 & \text{otherwise,}\end{cases}$$

and

$$\mathbf{D}_{\ell,\hat{q}}(E_{ij}) = \begin{cases}\sqrt{2}\mathrm{Re}\,(A_\ell)_{st} & \text{if } s<t \text{ and } p=q=\hat{q}\\ -\sqrt{2}\mathrm{Im}\,(A_\ell)_{st} & \text{if } s>t \text{ and } p=q=\hat{q}\\ (A_\ell)_{ss} & \text{if } s=t \text{ and } p=q=\hat{q}.\end{cases}$$

Therefore for any $\hat{p}<\hat{q}=1,\dots,m$ and any $i,j=1,\dots,nm$, using (s,t,p,q) defined in (12) and the definitions of $M_{\mathrm{Re}}, M_{\mathrm{Im}}, M_D$ on Page 6,

$$\begin{aligned}\left(\alpha_{\ell,\hat{p},\hat{q},\mathrm{Re}}\right)_{i,j} &= \alpha^T_{\ell,\hat{p},\hat{q},\mathrm{Re}}\ \mathrm{sHvec}(E_{ij}) = \mathbf{L}_{\ell,\hat{p},\hat{q},\mathrm{Re}}(E_{ij})\\ &= \begin{cases}\frac{1}{\sqrt{2}}(M_{\mathrm{Re}})_{\ell,st} & \text{if } s<t \text{ and } \{p,q\}=\{\hat{p},\hat{q}\}\\ -\frac{1}{\sqrt{2}}(M_{\mathrm{Im}})_{\ell,ts} & \text{if } s>t \text{ and } \{p,q\}=\{\hat{p},\hat{q}\}\\ (M_D)_{\ell,s} & \text{if } s=t,\ p<q \text{ and } (p,q)=(\hat{p},\hat{q})\\ 0 & \text{otherwise,}\end{cases}\end{aligned}$$

implying that $\alpha^T_{\ell,\hat{p},\hat{q},\mathrm{Re}}$ is one of the first k rows of the matrix $\left[I_{t(m-1)}\otimes N_{\mathrm{Re}\,\mathrm{Im}\,D}\ \ 0\right]$. (Here note that the number of pairs $(\hat{p},\hat{q})$ with $1\le\hat{p}<\hat{q}\le m$ is $t(m-1)=\frac{1}{2}m(m-1)$.

The zero block corresponds to the index pairs (i,j) with $p=q$, where (s,t,p,q) are the block indices defined in (12).) Similarly,

$$\begin{aligned}\left(\alpha_{\ell,\hat{p},\hat{q},\mathrm{Im}}\right)_{i,j} &= \alpha^T_{\ell,\hat{p},\hat{q},\mathrm{Im}}\ \mathrm{sHvec}(E_{ij}) = \mathbf{L}_{\ell,p,q,\mathrm{Im}}(E_{ij}) \\ &= \begin{cases} \frac{1}{\sqrt{2}}(M_{\mathrm{Im}})_{\ell,st} & \text{if } s<t \text{ and } (p,q)=(\hat{p},\hat{q}) \\ -\frac{1}{\sqrt{2}}(M_{\mathrm{Im}})_{\ell,st} & \text{if } s<t \text{ and } (p,q)=(\hat{q},\hat{p}) \\ -\frac{1}{\sqrt{2}}(M_{\mathrm{Re}})_{\ell,ts} & \text{if } s>t \text{ and } (p,q)=(\hat{p},\hat{q}) \\ \frac{1}{\sqrt{2}}(M_{\mathrm{Re}})_{\ell,ts} & \text{if } s>t \text{ and } (p,q)=(\hat{q},\hat{p}) \\ (M_D)_{\ell,s} & \text{if } s=t,\ p>q \text{ and } (p,q)=(\hat{q},\hat{p}) \\ 0 & \text{otherwise,} \end{cases}\end{aligned}$$

so $\alpha^T_{\ell,\hat{p},\hat{q},\mathrm{Im}}$ is one of the last k rows of the matrix $\begin{bmatrix} I_{t(m-1)} \otimes N_{\mathrm{Re\,Im}\,D} & 0 \end{bmatrix}$. Finally,

$$(\beta_{\ell,\hat{q}})_{ij} = \mathbf{D}_{\ell,\hat{q}}(E_{ij}) = \begin{cases} (M_{\mathrm{Re}})_{\ell,st} & \text{if } s<t \text{ and } p=q=\hat{q} \\ -(M_{\mathrm{Im}})_{\ell,st} & \text{if } s>t \text{ and } p=q=\hat{q} \\ (M_D)_{\ell,s} & \text{if } s=t \text{ and } p=q=\hat{q}, \end{cases}$$

so $\beta^T_{\ell,\hat{q}}$ is one of the rows of the matrix $\begin{bmatrix} 0 & I_m \otimes M_{\mathrm{Re\,Im}\,D} \end{bmatrix}$.

Hence, a matrix representation of $\mathcal{L}_A$ is given by

$$\begin{bmatrix} I_{t(m-1)} \otimes N_{\mathrm{Re\,Im}\,D} & 0 \\ 0 & I_m \otimes M_{\mathrm{Re\,Im}\,D} \end{bmatrix}.$$

A.1.5 Computing L_T

Recall the linear map $\mathcal{L}_T : \mathcal{H}^{nm} \to \mathcal{H}^n : P \mapsto [\mathrm{trace}(P_{st})]_{s,t=1,\ldots,n}$, which defines the second component of $\mathcal{L}$. We compute a matrix representation L_T of $\mathcal{L}_T$ by columns, i.e., by considering $\mathcal{L}_T(E_{ij}) \in \mathcal{H}^n$ for $i,j=1,\ldots,nm$. Defining (s,t,p,q) as in (12), we have

$$\mathcal{L}_T(E_{ij}) = \begin{cases} E_{st} & \text{if } p=q, \\ 0 & \text{otherwise.} \end{cases}$$

Hence the (i,j)-th column of L_T is given by

$$\mathrm{sHvec}(\mathcal{L}_T(E_{ij})) = \begin{cases} \mathrm{sHvec}(E_{st}) & \text{if } p=q, \\ 0 & \text{otherwise.} \end{cases}$$

This implies that $L_T = \begin{bmatrix} 0 & e_m^T \times I_{n^2} \end{bmatrix}$, where the zero block corresponds to the (i,j) pairs with $p \neq q$, and each row e_m^T in the Kronecker product corresponds to those (i,j) pairs with the same block indices (s,t) (and there are m pairs of (i,j) with the same block indices that have nonzero intra-block traces).

A.1.6 Alternative column orderings, eliminating redundant rows

Combining the results from the previous two sections, we arrive at a matrix representation of $\mathcal{L}$:

$$L = \begin{bmatrix} L_A \\ L_T \end{bmatrix} = \begin{bmatrix} I_{t(m-1)} \otimes N_{\mathrm{Re\,Im}\,D} & 0 \\ 0 & I_m \otimes M_{\mathrm{Re\,Im}\,D} \\ 0 & e_m^T \otimes I_{n^2} \end{bmatrix}. \tag{15}$$

In the final matrix representation that we use, some of the rows of the second block rows are linearly dependent of the other rows, if the linear map $\mathcal{L}_A$ contains the unital constraints.

Hence we remove those rows and replace the original matrix representation L in (15) by the following matrix:

$$\begin{aligned} L &= \begin{bmatrix} I_{t(m-1)} \otimes N_{\mathrm{Re\,Im}\,D} & 0 \\ 0 & \begin{bmatrix} I_{m-1} \otimes M_{\mathrm{Re\,Im}\,D} & 0 \end{bmatrix} \\ 0 & e_m^T \otimes I_{n^2} \end{bmatrix} \\ &= \begin{bmatrix} I_{t(m-1)} \otimes N_{\mathrm{Re\,Im}\,D} & 0 & 0 \\ 0 & I_{m-1} \otimes M_{\mathrm{Re\,Im}\,D} & 0 \\ 0 & e_{m-1}^T \otimes I_{n^2} & I_{n^2} \end{bmatrix}. \end{aligned}$$

Note that we can use alternative ordering for the off-diagonal entries inside the blocks. While this does not change the ordering of the columns in the second block column in (15) (which correspond to the diagonal entries inside the blocks), it can affect the column ordering of $N_{\mathrm{Re\,Im}\,D}$ (resulting in e.g. N_{final}).

A.1.7 Pseudoinverse of L

Using the block diagonal structure of L and the fact that

$$\begin{bmatrix} I_{m-1} \otimes M_{\mathrm{Re\,Im}\,D} & 0 \\ e_{m-1}^T \otimes I_{n^2} & I_{n^2} \end{bmatrix}^\dagger = \begin{bmatrix} I_{m-1} \otimes M_{\mathrm{Re\,Im}\,D}^\dagger & e_{m-1} \otimes (M_{\mathrm{Re\,Im}\,D})_{null} \\ -e_{m-1}^T \otimes M_{\mathrm{Re\,Im}\,D}^\dagger & I_{n^2} - (m-1)(M_{\mathrm{Re\,Im}\,D})_{null} \end{bmatrix} \tag{16}$$

(which can be easily verified to be the pseudoinverse), it is immediate that

$$\begin{aligned} L^\dagger &= \begin{bmatrix} I_{t(m-1)} \otimes N_{\mathrm{Re\,Im}\,D} & 0 \\ 0 & \begin{bmatrix} I_{m-1} \otimes M_{\mathrm{Re\,Im}\,D} & 0 \\ e_{m-1}^T \otimes I_{n^2} & I_{n^2} \end{bmatrix}^\dagger \end{bmatrix} \\ &= \begin{bmatrix} I_{t(m-1)} \otimes N_{\mathrm{Re\,Im}\,D} & 0 & 0 \\ 0 & I_{m-1} \otimes M_{\mathrm{Re\,Im}\,D}^\dagger & e_{m-1} \otimes (M_{\mathrm{Re\,Im}\,D})_{null} \\ 0 & -e_{m-1}^T \otimes M_{\mathrm{Re\,Im}\,D}^\dagger & I_n - (m-1)(M_{\mathrm{Re\,Im}\,D})_{null} \end{bmatrix}. \end{aligned}$$