

# LRCM MIN<sup>1</sup>

Raoul Pietersz<sup>2</sup>, Igor Grubišić<sup>3</sup>

**Abstract.** This note is the documentation of a Matlab template designed to minimize an arbitrary function over the set of correlation matrices with a prescribed rank. The template and this documentation is a modification of the SG MIN template (Edelman & Lippert 2000).

**Acknowledgement.** The template and this documentation is a copy-paste of the SG MIN template (Edelman & Lippert 2000), adapted to minimize over the space of low-rank correlation matrices. The LRCM MIN template owes its insightful template setup and computational speed to the developers of the SG MIN template, A. Edelman and R. Lippert.

**Key words:** Matlab, correlation, rank, geometric programming, Thomson problem

**JEL Classification:** G13

## 1 Introduction

We discuss a template that minimizes an arbitrary objective function over the set of correlation matrices with rank less than or equal to a given rank. Application areas include finance (finding the nearest low-rank correlation matrix) and electro-mechanics (the Thomson problem).

Suppose we wish to minimize  $F(Y)$  over  $Y$  an  $n \times d$  matrix such that  $\text{diag}(YY^T) = I$ , where  $Y^T$  denotes the transpose. Here  $F(\cdot)$  should be invariant under the right action of the orthonormal group:  $F(YQ) = F(Y)$  for any  $d \times d$  orthonormal matrix  $Q$ . The LRCM MIN template is designed to carry out such minimization. The associated theory stems partly from Grubišić & Pietersz (2004), in which geometric programming is applied to

---

<sup>1</sup>Date: 3 November 2004.

<sup>2</sup>Erasmus Research Institute of Management, Erasmus University Rotterdam, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands (email: pietersz@few.eur.nl, tel: +31 10 4088932) and Product Development Group (HQ7011), ABN AMRO Bank, P.O. Box 283, 1000 EA Amsterdam, The Netherlands

<sup>3</sup>Mathematical Institute, University of Utrecht, P.O. Box 80010, 3508 TA Utrecht, The Netherlands (email: grubisic@math.uu.nl, tel: +31 30 2531529)

optimization over low-rank correlation matrices. For more details the reader is referred to Grubišić & Pietersz (2004). For an accessible and excellent introduction to geometric programming, via concrete examples for the Stiefel and Grassmann manifolds, see Edelman, Arias & Smith (1998).

## 2 Matlab Templates

The templates are ready for immediate use or as a departure point for generalization, e.g. problems including low-rank correlation matrices plus other variables. In the simplest mode, the user need only supply the function  $F(\cdot)$  to be minimized (and its first and second derivatives, optionally) in `F.m` (in `dF.m` and `ddF.m`) and an initial guess  $Y_0$ . The initial guess  $Y_0$  needs to lie on the ‘Cholesky’ manifold, i.e., the upper part should be lower triangular,  $Y_0(i, j) = 0$ ,  $i = 1, \dots, d-1$ ,  $j = i+1, \dots, d$ , and  $\text{diag}(Y_0 Y_0^T) = I$ . Then all that is required is a single call to `lrcm_min` (denoting Low-Rank Correlation Matrix Minimization).

```
[fn, Yn]=lrcm_min(Y0)
```

For example, if the function `F.m` has the form

```
function f=F(Y)
a=sqrt(3/8); % approx 0.6124
C=[1 a a; a 1 5/6; a 5/6 1];
f=0.5*norm(C-Y*Y', 'fro')^2;
```

we can call e.g. `lrcm_min([1 0; .6 .8; .8 .6])`.

Edelman & Lippert (2000) recommend to provide first derivative information:

```
function df=dF(Y)
a=sqrt(3/8); % approx 0.6124
C=[1 a a; a 1 5/6; a 5/6 1];
df=2*(Y*Y'-C)*Y;
```

The code can do finite differences, but it is more slow and problematic. Second derivative information can also be provided by the user:

```
function ddf=ddF(Y,H)
a=sqrt(3/8); % approx 0.6124
C=[1 a a; a 1 5/6; a 5/6 1];
ddf=2.0*((Y*Y'-C)*H+(Y*H'+H*Y')*Y);
```

argument	description
mode	{‘newton’, ‘dog’, ‘prcg’, ‘frcg’, ‘steep’}
verbose	{‘verbose’, ‘quiet’}
speedup	{‘speedon’, ‘speedoff’}
maxiter	maximum number of iterations
ftol	first convergence tolerance
gradtol	second convergence tolerance

Table 1: A short list of the optional arguments for `lrcm_min`.

According to Edelman and Lippert (2000) this is not nearly as crucial for speed as providing first derivative information, but it may improve accuracy. An example test call where  $F(\cdot)$  is known to have optimal value<sup>4</sup>  $1/36 \approx 0.0278$  is

```

iter    grad      F(Y)      step type
0   1.624770e-001  5.140162e-002  none
1   1.711515e-002  2.795564e-002  Newton step
2   6.277607e-005  2.777778e-002  Newton step
3   4.441285e-009  2.777778e-002  Newton step
4   2.983180e-009  2.777778e-002  Newton step

```

```
ans =
```

```
0.0278
```

The full calling sequence to `lrcm_min` is

```
[fn, Yn]=lrcm_min(Y0,mode,verbose,maxiter,ftol,gradtol)
```

where `Y0` is required and the optional arguments are (see Table 1):

**mode** selects the optimization method that will be used. ‘newton’ selects Newton’s method with a conjugate gradient based inversion of the Hessian. ‘dog’ selects a dog-leg step algorithm which interpolates a steepest descent and a Newton’s step. ‘frcg’ selects Fletcher-Reeves conjugate gradient. ‘prcg’ selects Polak-Ribière conjugate gradient which has the advantage that it does not require a very accurate Hessian (and thus, the safest of the methods if one uses a finite difference approximation

---

<sup>4</sup>For a proof, see the appendix.

to implement `ddF.m`). Lastly, ‘steep’ lets the implementation perform a steepest descent method. The option ‘newton’ is the default.

**verbose** determines whether the function will display reports on each iteration while the function executes. When this argument is ‘verbose’ (the default) data will be displayed. When this argument is ‘quiet’ then no convergence data is displayed.

**speedup** If ‘speedon’ is selected, then code does not strictly require the matrices to be of Cholesky form, which empirically has shown to be more efficient. If ‘speedoff’ is selected (the default), then all intermediate matrices are in Cholesky form.

**maxiter, ftol and gradtol** We declare convergence if both of two conditions are true: `grad/gradinit < gradtol` (default `1e-7`) and `(f-fold)/f < ftol` (default `1e-10`) or when the number of iterations exceeds `maxiter` (default 20). Here `gradinit` is the initial magnitude of the gradient and `fold` is the value  $F(Y)$  at the last iteration.

The string arguments can be supplied in any order; the numeric arguments have to be supplied in the order `maxiter`, `ftol` and `gradtol`.

### 3 Sample Problems and Their Differentials

This section provides two example problems that may be solved by LRCM MIN. The effort involved in applying the template is in calculating the differential. The differential  $dF(Y)$  is given by

$$(dF(Y))_{ij} = \frac{\partial F}{\partial Y_{ij}}.$$

An implicit definition of the differential is given by

$$\langle dF(Y), H \rangle = \frac{d}{dt} F(Y(t))|_{t=0}$$

where  $Y(t)$  is any curve for which  $\dot{Y}(0) = H$ . Edelman & Lippert (2000, section 8.3.3) provide useful hints for calculating explicit expressions for the differential via the implicit definition of the differential – the reader is referred there if such calculation of the differential be required. As a check it is recommended to use the finite difference `dF.m` code supplied in the subdirectory `finitediff` to check derivations before proceeding.

The software needs a function called `ddF.m` which returns  $\frac{d}{dt}dF(Y(t))|_{t=0}$  for  $\dot{Y}(0) = H$ . The sort of second derivative information required by the software is easier to derive than the first. If one has an analytic expression for  $dF(Y)$ , then one need only differentiate.

It is strongly suggested that one use an analytic expression for computing  $dF(Y)$ , as the finite difference code for it requires a large number of function evaluations ( $2nd$ ).

### 3.1 Finding the Nearest Low-Rank Correlation Matrix

The problem of finding the nearest low-rank correlation matrix is the minimization of  $\|C - YY^T\|$  over the stratified space  $\text{diag}(YY^T) = I$ . This minimization determines the nearest matrix  $\hat{C}$  to  $C$  such that  $\hat{C}$  is a correlation matrix of rank  $d$ . The differential of  $F(Y) = \frac{1}{2}\|YY^T - C\|_F^2 = \frac{1}{2}\langle\psi, \psi\rangle$ , with  $\psi := YY^T - C$ , is  $dF(Y) = 2\psi Y$ . This can be seen as follows:

$$\begin{aligned}\frac{d}{dt}F(Y(t)) &= \langle\dot{\psi}, \psi\rangle \\ &= \langle HY^T + YH^T, \psi\rangle \\ &= \langle HY^T, \psi\rangle + \langle YH^T, \psi\rangle \\ &= \langle H, 2\psi Y\rangle = \langle H, dF(Y)\rangle, \quad \forall H.\end{aligned}$$

The second derivative of  $F(Y)$  is given by the equation

$$ddF(Y) = 2(\psi H + (YH^T + HY^T)Y).$$

An initial feasible point is obtained by a well-known technique believed to have been introduced by Flury (1988). We first perform an eigenvalue decomposition  $C = QDQ^T$  with  $Q$  an orthonormal matrix and with  $D$  a diagonal matrix containing the eigenvalues,  $D_{11} \geq \dots \geq D_{nn}$ . Then we define  $Y$  by assigning to each row

$$Y_i = \frac{\{Q_d \sqrt{\max(D_d, 0)}\}(i, :)}{\|\{Q_d \sqrt{\max(D_d, 0)}\}(i, :)\|}$$

where  $Q_d$  consists of the first  $d$  columns of  $Q$  and where  $D_d$  is the principal sub-matrix of  $D$  of degree  $d$ . The scaling is to ensure that each row of  $Y$  is of unit length. If row  $i$  is a priori of zero length, then we choose  $Y_i$  to be an arbitrary vector in  $\mathbb{R}^d$ . The Flury procedure has been implemented in `guess.m`. Its calling sequence is

`Y0=guess(C,d)`

with `C` the original matrix and with `d` the desired rank. `guess.m` also brings the matrix `Y0` in the desired Choleksy form.

### 3.2 The Thomson Problem

The Thomson problem is concerned with minimizing the potential energy of  $n$  charged particles on the 2-sphere  $S^2 \subset \mathbb{R}^3$  ( $d = 3$ ). The associated potential is given by

$$F(Y) = \sum_{i=1}^n \sum_{j=i+1}^n \frac{1}{\|Y_i - Y_j\|_2}.$$

Note that the potential energy is invariant under the action of the orthogonal group; the minimization problem is thus over the set of rank-3 correlation matrices. This problem has a long history, see for example the website [www.ogre.nu/sphere.htm](http://www.ogre.nu/sphere.htm).

The potential energy function has been implemented in `F.m` in the subdirectory `thomson problem`. Its differential in explicit form is unknown to us, therefore the finite difference functions apply in this case. An initial feasible point is chosen at random by the function `thomson_guess.m`. Its calling sequence is

```
Y0=thomson_guess(n,d)
```

with `n` the number of charged particles and with `d` the embedding dimension.

## 4 Numerical Examples

We have provided implementations for the sample problems described.

We present the examples and their verbose outputs. We have included these outputs so the reader can check that his/her copy of the package is executing properly. The user should be running Matlab (version 6 or compatible) in the following directory:

```
>> ls
clamp.m      Fline.m      invdgrad_MINRES.m  lrcm_prcg.m
connection.m grad.m        ip.m               lrcm_steep.m
ddF.m        gradline.m   lrcm_dog.m         move.m
dF.m         guess.m      lrcm_frcg.m        svdplus.m
dFline.m     Hessian.m    lrcm_min.m         tangent.m
F.m          invdgrad_CG.m lrcm_newton.m
```

By default the functions `F.m` and its derivative functions corresponding to finding the nearest low-rank correlation matrix are present in the base directory. Probably also present are the subdirectories

```
distance minimization/  
doc/  
finitediff/  
matlab workspaces/  
problem from doc/  
sample problem/  
thomson problem/ weighted distance minimization/
```

Both `>> ls 'distance minimization'` and `>> ls 'weighted distance minimization'` should give:

```
ddF.m dF.m F.m guess.m svdplus.m
```

```
>> ls doc  
conv_chart_corr.eps      lrcm min.tex  
conv_chart_thomson.eps   lrcm min.pdf  lrcm_min.bib
```

```
>> ls finitediff  
ddF.m dF.m
```

```
>> ls 'matlab workspaces'  
C.mat example from doc.mat
```

The workspace 'C' contains a matrix used in a convergence run test in Grubišić & Pietersz (2004). The workspace 'example from doc' contains the matrix described in the appendix.

Both `>> ls 'problem from doc'` and `>> ls 'sample problem'` should give:

```
ddF.m dF.m F.m
```

```
>> ls 'thomson problem'  
ddF.m dF.m F.m thomson_guess.m
```

Before running 'distance minimization' a global `FParameters` should be created that has a field `C` containing the target correlation matrix  $C$ . The 'weighted distance minimization' requires additionally a field `W` that contains the weights matrix. The 'sample problem' requires an  $n \times d$  matrix `Ystar` as additional field to `FParameters`. For the 'Thomson problem' or for the 'problem from doc' a field is not necessary. (For the Thomson problem, the energy potential function has no parameters.)

## 4.1 A Sample Problem of Finding the Nearest Low-Rank Correlation Matrix

In this example we try to find the rank-4 correlation matrix nearest to a randomly generated  $12 \times 12$  correlation matrix  $C$ .

```
>> rand('state',0); randn('state',0);
>> n=12; d=4;
>> C=gallery('randcorr',n);
>> global FParameters; FParameters.C=C;
>> [fn,Yn]=lrcm_min(guess(C,d),100);
  iter   grad      F(Y)      step type
  0   3.666272E+00   9.614876E+00   none
      invdgrad: Hessian not positive definite, CG terminating early
  1   1.782298E+00   8.426158E+00   Newton step
      invdgrad: Hessian not positive definite, CG terminating early
  2   9.028751E-01   8.200722E+00   Newton step
      invdgrad: Hessian not positive definite, CG terminating early
(...iterations later...)
 10   1.476235E-02   7.947614E+00   Newton step
      invdgrad: max iterations reached inverting the hessian by CG
 11   3.019603E-03   7.947583E+00   Newton step
 12   1.352081E-05   7.947583E+00   Newton step
 13   2.709074E-07   7.947583E+00   Newton step
```

Figure 1 shows the convergence curve for this run.

## 4.2 A Sample Thomson Problem

In this example we try to find a minimum energy configuration of 12 charged particles on the 2-sphere ( $n=12$ ,  $d=3$ ).

```
>> copyfile('thomson problem/*.m', '*.m');
>> rand('state',0); randn('state',0);
>> n=12; d=3;
>> [fn,Yn]=lrcm_min(clamp(thomson_guess(n,d)));
  iter   grad      F(Y)      step type
  0   1.365679E+01   5.990261E+01   none
      invdgrad: Hessian not positive definite, CG terminating early
  1   7.667772E+00   5.139003E+01   steepest step
      invdgrad: Hessian not positive definite, CG terminating early
```



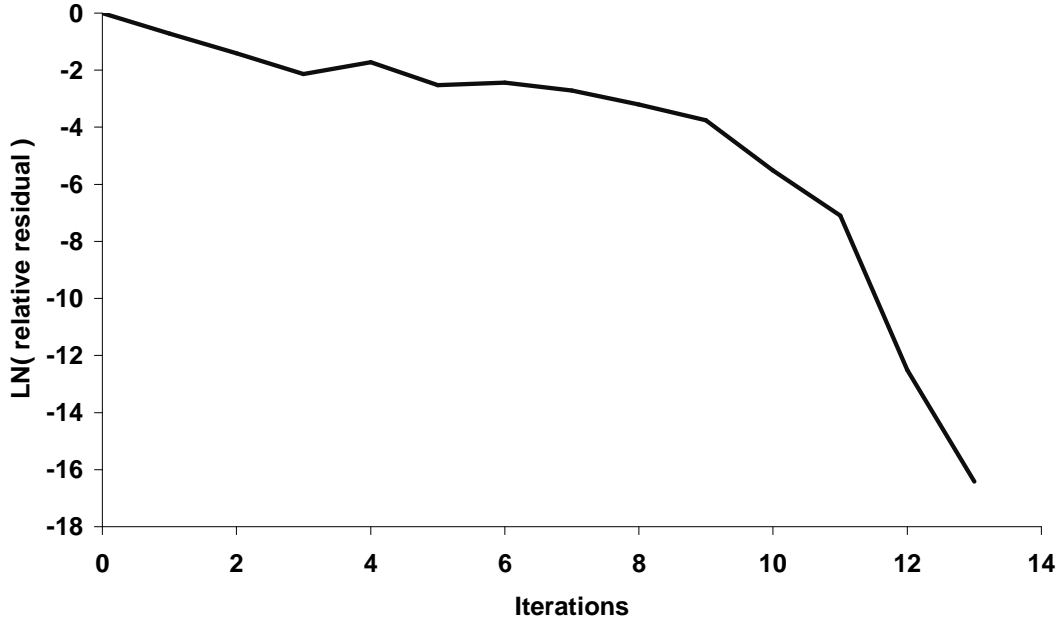


Figure 1: Finding the nearest low-rank correlation matrix: Sample problem convergence curve. The natural logarithm of the relative residual  $\ln(\|\nabla F(Y_i)\|/\|\nabla F(Y_0)\|)$  versus the number of iterations.

```

2  1.969602E+00    4.977655E+01    Newton step
   invdgrad: Hessian not positive definite, CG terminating early
3  1.090684E+00    4.943760E+01    Newton step
   invdgrad: Hessian not positive definite, CG terminating early
4  7.275627E-01    4.931665E+01    Newton step
   invdgrad: Hessian not positive definite, CG terminating early
5  2.312192E-01    4.927771E+01    steepest step
   invdgrad: Hessian not positive definite, CG terminating early
6  3.787738E-01    4.923130E+01    Newton step
   invdgrad: max iterations reached inverting the hessian by CG
7  3.162751E-01    4.918279E+01    Newton step
   invdgrad: max iterations reached inverting the hessian by CG
8  6.030232E-02    4.916636E+01    Newton step
   invdgrad: max iterations reached inverting the hessian by CG
9  6.587839E-03    4.916526E+01    Newton step
10 1.583339E-05    4.916525E+01    Newton step
11 1.083864E-06    4.916525E+01    Newton step

```

Figure 2 shows the convergence curve for this run.

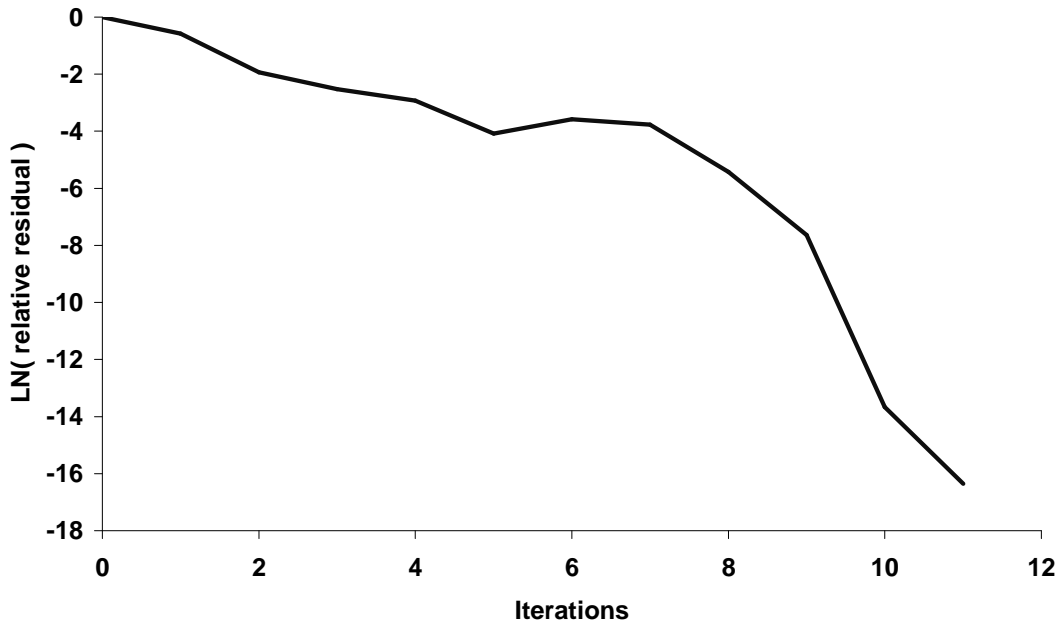


Figure 2: Distributing charged particles on a sphere: Sample problem convergence curve. The natural logarithm of the relative residual  $\ln(\|\nabla F(Y_i)\|/\|\nabla F(Y_0)\|)$  versus the number of iterations.

## 5 Concluding Remarks

Edelman & Lippert (2000, section 8.5) describe the structure of the SG MIN template and how one could modify it. This description applies equally well to LRCM MIN since it is nothing more but a modification of SG MIN. The reader is referred to Edelman & Lippert (2000, section 8.6) for a treatment of the differential geometry and geometric programming applied in the template.

The LRCM MIN template is probably the most efficient implementation known at the time of writing this paper for finding low-rank correlation matrix approximations to given correlation matrices. This has been shown numerically at least for a particular setup in Grubišić & Pietersz (2004, section 6). This makes the LRCM MIN template interesting for financial practitioners since financial correlation matrices (historic or implied) tend to be of full rank whereas due to computational time constraints, it is more desirable to price with a low-factor model – and thus a low-rank correlation matrix will be input to such models. We hope that this documentation and the template will provide these practitioners easy access to the geometric programming solution of finding such low-rank correlation matrices.

Another application in the finance area could possibly be the following: The template has the potential for calibration of the BGM model of Brace, Gątarek & Musiela (1997). The BGM model is an interest rate derivatives valuation model. LRCM MIN could in principle be used to calibrate BGM jointly to caps, swaptions and forward (swap) rate correlations. Namely the parameters of the model can be viewed as certain volatility parameters plus a given-rank correlation matrix. This gives rise to a product manifold, for which the part of the low-rank correlation matrices can be extended from LRCM MIN.

## References

- Brace, A., Gątarek, D. & Musiela, M. (1997), ‘The market model of interest rate dynamics’, *Mathematical Finance* **7**(2), 127–155.
- Edelman, A. & Lippert, R. (2000), Nonlinear eigenvalue problems with orthogonality constraints (section 8.3), *in* Z. Bai, J. Demmel, J. Dongarra, A. Ruhe & H. van der Vorst, eds, ‘Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide’, SIAM, Philadelphia.
- Edelman, A., Arias, T. A. & Smith, S. T. (1998), ‘The geometry of algorithms with orthogonality constraints’, *SIAM Journal of Matrix Analysis and its Applications* **20**(2), 303–353.
- Flury, B. (1988), *Common Principal Components and Related Multivariate Models*, J. Wiley & Sons, New York.
- Grubišić, I. & Pietersz, R. (2004), Efficient rank reduction of correlation matrices, Working paper, Utrecht University, Utrecht.
- Wu, L. (2003), ‘Fast at-the-money calibration of the libor market model using lagrange multipliers’, *Journal of Computational Finance* **6**(2), 39–77.
- Zhang, Z. & Wu, L. (2003), ‘Optimal low-rank approximation to a correlation matrix’, *Linear Algebra and its Applications* **364**, 161–187.

## A The numerical example of section 2

In this section we show that with respect to the rank-3 correlation matrix

$$(1) \quad C := \begin{pmatrix} 1 & \alpha & \alpha \\ \alpha & 1 & 5/6 \\ \alpha & 5/6 & 1 \end{pmatrix}, \quad \alpha := \sqrt{3/8} \approx 0.6124,$$

the globally nearest correlation matrix of rank 2 is given by

$$(2) \quad C_0 := \begin{pmatrix} 1 & \alpha & \alpha \\ \alpha & 1 & 1 \\ \alpha & 1 & 1 \end{pmatrix}.$$

This fact is obtained by the Lagrange multiplier results of Zhang & Wu (2003) and Wu (2003). Denote by  $\mathcal{R}_{n,d}$  and  $\mathcal{C}_{n,d}$  the  $n \times n$  matrices (correlation matrices respectively) of rank  $d$ . Define the Lagrangian for  $X \in \mathcal{R}_{n,d}$  and for Lagrange multipliers  $\lambda \in \mathbb{R}^n$

$$\mathcal{L}(X, \lambda) := -\frac{1}{2}\|C - X\|_F^2 - \lambda^T \text{diag}(C - X) = -\frac{1}{2}\|C + \text{diag}(\lambda) - X\|_F^2 + \frac{1}{2}\|\lambda\|_2^2.$$

Define

$$(3) \quad V(\lambda) := \min\{\mathcal{L}(X, \lambda) ; X \in \mathcal{R}_{n,d}\}.$$

This minimization problem can be solved via an eigenvalue decomposition

$$C + \text{diag}(\lambda) = QDQ^T,$$

with  $Q$  an orthonormal matrix and  $D$  the diagonal matrix containing the eigenvalues,  $|D_{11}| \geq \dots \geq |D_{nn}|$ . It is straightforward to show that the solution to problem (3) is then given by  $X^* = Q_d D_d Q_d^T$  with  $Q_d$  consisting of the first  $d$  columns of  $Q$  and with  $D_d$  the principal sub-matrix of  $D$  of degree  $d$ . Suppose now that there exists a Lagrange multiplier  $\lambda_0$  with  $C_0$  the associated solution to problem (3) such that the diagonal of  $C_0$  consists of ones. Then  $C_0$  is the rank- $d$  correlation matrix globally nearest to  $C$ , since for all  $X \in \mathcal{C}_{n,d}$  we have

$$\frac{1}{2}\|C - X\|_F^2 \stackrel{(a)}{=} -\mathcal{L}(X, \lambda_0) \stackrel{(b)}{\geq} -V(\lambda_0) \stackrel{(c)}{=} \frac{1}{2}\|C - C_0\|_F^2.$$

Here (in)equality

(a) is obtained from the property that  $X \in \mathcal{C}_{n,d}$ ,

(b) is by definition of  $V$ , and

(c) is by assumption that  $C_0$  has unit diagonal.

Now for the matrix  $C$  of equation (1) consider the Lagrange multiplier  $\lambda_0 := (0, 1/6, 1/6)^T$ . The matrix  $C + \text{diag}(\lambda)$  has eigenvalues  $\{5/2, 1/2, 1/3\}$ . A normalized eigenvector corresponding to eigenvalue  $1/3$  is given by  $e_0 := (0, \sqrt{1/2}, -\sqrt{1/2})^T$ . Thus the associated solution to problem (3) is given by  $C + \text{diag}(\lambda) - 1/3e_0e_0^T$ , which turns out to be equal to the matrix  $C_0$  of equation (2). Since this matrix has unit diagonal, we conclude that it is the rank-2 correlation matrix globally nearest to  $C$ .