City Research Online

City St George's, University of London

Citation: Asimit, V., Wang, R., Zhou, F. & Zhu, R. (2025). Efficient Positive Semidefinite Matrix Approximation by Iterative Optimisations and Gradient Descent Method. Risks, 13(2), 28. doi: 10.3390/risks13020028

This is the published version of the paper.

This version of the publication may differ from the final published version. To cite this item please consult the publisher's version.

Permanent repository link: https://openaccess.city.ac.uk/id/eprint/34629/

Link to published version: https://doi.org/10.3390/risks13020028

Copyright and Reuse: Copyright and Moral Rights remain with the author(s) and/or copyright holders. Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge, unless otherwise indicated, provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way. For full details of reuse please refer to City Research Online policy.

City Research Online: http://openaccess.city.ac.uk/ publications@citystgeorges.ac.uk

risks

MDPI

Article

Efficient Positive Semidefinite Matrix Approximation by Iterative Optimisations and Gradient Descent Method

Vali Asimit [1], Runshi Wang [1,*], Feng Zhou [2] and Rui Zhu [1]

[1] Bayes Business School, City St George's, University of London, 106 Bunhill Row, London EC1Y 8TZ, UK; asimit@city.ac.uk (V.A.); rui.zhu@city.ac.uk (R.Z.)
[2] MRC Biostatistics Unit, University of Cambridge, Cambridge CB2 0SR, UK; feng.zhou@city.ac.uk or feng.zhou@mrc-bsu.cam.ac.uk
* Correspondence: runshi.wang@bayes.city.ac.uk

Abstract: We devise two algorithms for approximating solutions of PSDisation, a problem in actuarial science and finance, to find the nearest valid correlation matrix that is positive semidefinite (PSD). The first method converts the PSDisation problem with a positive semidefinite constraint and other linear constraints into iterative Linear Programmings (LPs) or Quadratic Programmings (QPs). The LPs or QPs in our formulation give an upper bound of the optimal solution of the original problem, which can be improved during each iteration. The biggest advantage of this iterative method is its great flexibility when working with different choices of norms or with user-defined constraints. Second, a gradient descent method is designed specifically for PSDisation under the Frobenius norm to measure how close the two metrices are. Experiments on randomly generated data show that this method enjoys better resilience to noise while maintaining good accuracy. For example, in our experiments with noised data, the iterative quadratic programming algorithm performs best in more than 41% to 67% of the samples when the standard deviation of noise is 0.02, and the gradient descent method performs best in more than 70% of the samples when the standard deviation of noise is 0.2. Examples of applications in finance, as well as in the machine learning field, are given. Computational results are presented followed by discussion on future improvements.

Keywords: nearest correlation matrix; positive semidefinite; semidefinite programming

check for updates

Academic Editor: Mogens Steffensen

Received: 29 November 2024
Revised: 26 January 2025
Accepted: 5 February 2025
Published: 7 February 2025

Citation: Asimit, Vali, Runshi Wang, Feng Zhou, and Rui Zhu. 2025. Efficient Positive Semidefinite Matrix Approximation by Iterative Optimisations and Gradient Descent Method. *Risks* 13: 28. https://doi.org/10.3390/risks13020028

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

A correlation matrix is a square matrix summarising correlation coefficients between each pair of variables. This is used by many financial or insurance companies to determine how a group of risks are dependent from each other (Pearson 1895). It is useful across various fields to help us understand the relationship between variables. For example, in finance, a correlation matrix is used in optimisation to build a portfolio to minimise the volatility. It also enables efficient computation of cardinality constrained efficient frontiers to solve mean-variance portfolio selection problems (Steuer et al. 2024). In actuarial science, insurance companies use correlation matrices to measure insurers' exposure to risks and calculate the Solvency Capital Requirement (SCR) under the Solvency II Standard Formula (Milhaud et al. 2018). Another application is to work with copulas for aggregation of risks in a more complex capital model (Milhaud et al. 2018). In genomics science, correlation matrices help identify relationship between genes and construct relevance networks and association networks (Opgen-Rhein and Strimmer 2007). They also play a vital role in genes classifications. In machine learning and data science, correlation matrices are widely

used to identify key patterns of data through dimensionality reduction techniques such as Principal Component Analysis (PCA) (Pearson 1901).

In all these applications, the correlation matrices are required to be symmetric and Positive Semidefinite (PSD), with diagonal entries all equal to one. In practice, however, the correlation matrix estimated from the empirical data is rarely PSD due to a variety of reasons, including data incompleteness, noise, rounding, manual adjustment, or inconsistent computing approaches, etc. (Cutajar et al. 2017). For example, when correlations between variables are estimated in groups and then joined together, the resulting correlation matrix can be non-PSD. There are many other factors that contribute to this problem. The data collected in many scenarios can be incomplete and may need further repair. The correlations may have been estimated using inconsistent approaches or have been manually adjusted according to requirements. All of these reasons lead to an invalid estimation of the correlation matrix (Cutajar et al. 2017). Therefore, a preprocessing method is needed to obtain a valid PSD correlation matrix based on the empirical matrix in order to perform any further analysis. The problem of finding such a correlation matrix nearest to an empirical matrix is called PSDisation (Milhaud et al. 2018).

Currently most state-of-the-art PSDisation methods focus on finding the nearest correlation matrix with respect to the F-norm, of which some apply to H-norm as well. Over those algorithms, the Newton Method (NM) (Qi and Sun 2006) and the augmented Lagrangian Method (ALD) (Qi and Sun 2011) are the most efficient in producing optimal solutions. However, depending on the industrial situations, insurance companies may wish to measure the similarity using an alternative norm other than the F-norm, which makes some algorithms more time-consuming or invalid. Another PSDisation method that is widely used in industry is the shrinking method (Higham et al. 2016). It is a simple and efficient method that can improve the result of the alternating projections algorithm (APM) (Cutajar et al. 2017). However, a target matrix that lies inside the cone of PSD matrices is needed for the shrinking method; therefore, its performance depends heavily on the choice of the target matrix.

Ledoit and Wolf (2004) introduced a shrinkage method in order to improve the stability of the estimated covariance matrix that is not positive definite or ill-conditioned. This often happens when the matrix size is large and the number of observations is smaller than, or comparable, to the number of variables. It achieves its goal by reducing the variance of the estimator, and thus reducing the Mean Squared Error (MSE) via a computationally inexpensive approach. This involves a shrinkage of the sample covariance matrix (unbiased) towards a target matrix without variance, usually an identity matrix. The optimal shrinkage intensity parameter can be computed analytically to minimise the MSE. The similar idea can be used to obtain a PSD correlation matrix starting from an invalid one. Although the shrinkage method is not defined as a PSDisation method, it has been widely used in many applications in different fields. We will include this method in our experiments on real datasets to compare its performance with other PSDisation methods.

Therefore, most algorithms in the current literature primarily operate under the F-norm or the H-norm for the general PSDisation problems. However, they tend to lack flexibility when addressing more specific needs in specialised cases. To tackle this limitation, we propose two new methods to approximate the solution of the PSDisation problem, which offer greater flexibility and provide robust solutions to practical real-world challenges. The first algorithm works by solving a series of optimisations, such as Linear Programming (LP) or Quadratic Programming (QP). Instead of solving the original problem, which requires the correlation matrix $\mathbf{X}$ to be PSD, our formulation requires $\mathbf{X} = \mathbf{U}^T\mathbf{Q}\mathbf{U}$, where $\mathbf{Q}$ belongs to the convex cone of diagonally dominant symmetric matrices with non-negative diagonal entries and thus all constraints are linear. This is a rich subset of the PSD matrices set,

but the optimisations can be solved more efficiently due to the linearity of the constraints. The main advantage of this method is its flexibility as our formulation is adapted directly from the original problem and is thus effective on any choice of norms, including the F-norm, the Chebyshev norm, and the H-norm, or potentially a combination of any of these norms above. The advantages of using these norms will be illustrated in the results of our experiments. The second algorithm is to find the nearest correlation matrix to the initial matrix using an iterative gradient projection method under the F-norm. This can be achieved simply by repeatedly taking a step along the gradient of the objective function and then projecting the matrix back to the PSD cone. This algorithm is easy to implement and enjoys good efficiency while maintaining good accuracy.

We will compare our two proposed algorithms with the state-of-the-art NM, ALD, and the shrinkage method. Extensive experiments demonstrate that our first iterative algorithm could achieve results comparable to state-of-the-art methods, while showing more flexibility in handling complex constraints or different choices of norms. Our second gradient descent method provides slightly less accurate results but runs faster than the first algorithm and is more resilient to noise than some state-of-the-art methods and can therefore be more reliable in practical situations.

This paper is organised as follows. In Section 2, we introduce the notations and formally define the PSDisation problem. Section 3 summarises some of the most popular PSDisation algorithms that exist in the current literature, together with the shrinkage method by Ledoit and Wolf (2004). In Section 4, we introduce our two approaches to the PSDisation problem. Experiment results with simulated data, as well as real data, for both algorithms are reported in Section 5 and Section 6, respectively. Finally, in Section 7, we conclude this paper and propose some future work.

2. Problem Formulation

We now introduce the notations we use throughout this paper and the background knowledge of the PSDisation problem. We denote the set of $n \times n$ real matrices by $\mathbb{R}^{n\times n}$. A real-valued symmetric matrix $\mathbf{Z} \in \mathbb{R}^{n\times n}$ is Positive Definite (PD) if $\boldsymbol{y}^T\mathbf{Z}\boldsymbol{y}$ is positive for any nonzero real vector $\boldsymbol{y} \in \mathbb{R}^n$. It is Positive Semidefinite (PSD) if $\boldsymbol{y}^T\mathbf{Z}\boldsymbol{y}$ is non-negative for any nonzero real vector $\boldsymbol{y}$. Equivalently, a real symmetric matrix is PSD if all its eigenvalues are non-negative. We write $\mathbf{Z} \succeq 0$ if $\mathbf{Z}$ is PSD ($\mathbf{Z} \succ 0$ for PD). We use $\text{diag}(\mathbf{Z})$ to denote the vector of all diagonal entries of $\mathbf{Z}$. Let $\mathbf{I}$ be the identity matrix and $\mathbf{1}$ be the all-ones vector. The operator $||\cdot||$ over the set of real matrices represents a generic matrix norm, for which different choices are signified by a subscript and four choices are explored in this paper; namely, for a $\mathbf{M} \in \mathbb{R}^{n\times n}$, these norms are as follows:

1. *Frobenious norm (or F-norm)*: $||\mathbf{M}||_\mathbf{F} = \sqrt{\sum_{i=1}^n \sum_{j=1}^n m_{ij}^2} = \sqrt{\text{Tr}(\mathbf{M^T M})}$;
2. *W-norm*: $||\mathbf{M}||_\mathbf{W} = ||\mathbf{W}^{\frac{1}{2}}\mathbf{M}\mathbf{W}^{\frac{1}{2}}||_\mathbf{F}$, where $\mathbf{W} \in \mathbb{R}^{n\times n}$ is a square matrix with positive entries;
3. *H-norm*: $||\mathbf{M}||_\mathbf{H} = \sqrt{\sum_{i=1}^n \sum_{j=1}^n h_{ij} m_{ij}^2}$;
4. *Chebyshev norm (or max norm)*: $||\mathbf{M}||_\mathbf{C} = \max_{i,j}(|m_{ij}|)$,

where m_{ij} is the (i, j) entry of $\mathbf{M}$.

The PSDisation problem we consider in this paper is defined as follows. Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be a symmetric matrix that is not PSD with diagonal entries all equal to 1. It represents an empirical correlation matrix of n random variables $Y_1, \ldots, Y_n$, with standard deviations $\sigma_{Y_1}, \ldots, \sigma_{Y_n}$ whose (i, j) entry is given by

$$a_{ij} = \text{corr}(Y_i, Y_j) = \frac{\text{cov}(Y_i, Y_j)}{\sigma_{Y_i}\sigma_{Y_j}}, \text{ if } \sigma_{Y_i}\sigma_{Y_j} > 0.$$

We aim to find the nearest PSD matrix $\mathbf{X}$ of the same form. The problem can be written as follows.

$$\min_{\mathbf{X}} \ ||\mathbf{A}-\mathbf{X}||^2 \quad s.t.\ \mathbf{X} \succeq 0, \quad \text{diag}(\mathbf{X}) = \mathbf{1}. \tag{1}$$

As we wish to find the nearest correlation matrix $\mathbf{X}$ to the empirical matrix $\mathbf{A}$, the choice of the matrix norm $||\cdot||$ determines how "nearest" is defined.

The F-norm (Van Loan and Golub 1983) is a trivial choice for defining how close two matrices are and is widely used in most of the PSDisation literature. However, recent studies have explored the use of different norms in order to meet more practical demands. The W-norm is a weighted norm commonly used in numerical mathematics (Higham 2002) that allows us to force some elements of $\mathbf{X}$ to be closer to the corresponding entries in $\mathbf{A}$. Setting $\mathbf{W}$ to the identity matrix will retrieve the F-norm. However, when applied to actuarial science, an insurance company may be more interested in the H-norm (Milhaud et al. 2018). The H-norm is similar to the F-norm but allows one to assign weights to $\mathbf{M}$ on an element-by-element basis. Therefore, in the situation where one has prior knowledge on the correlation matrix, the H-norm may be preferred. For example, when one is more confident in some entries of the estimated correlation matrix $\mathbf{A}$ and wants to fix them, one can allocate more weights on those entries so that a small change results in a huge penalty when measuring the distance between matrices. The H-norm is equivalent to the F-norm when $\mathbf{H}$ is an all-ones matrix. Finally, the Chebyshev norm specifies an element-wise ceiling for the largest difference between the entries of $\mathbf{X}$ and $\mathbf{A}$, which can be particularly useful and more robust to outliers when m_{ij} is large.

It is worth mentioning that the choice of norms should be determined according to specific applications and there is no guarantee which one would work best. Therefore, we attempt to develop a more flexible algorithm that can be generalised to work with different norm choices.

3. Related Work

This section summarises some most popular existing solutions on PSDisation, as well as the shrinkage method to fix the ill-conditioned covariance matrix.

One of the early algorithms to tackle the PSDisation problem is the alternating projections method (APM) (Higham 2002). It finds the nearest PSD correlation matrix by iterative projections onto two convex sets, the PSD matrices cone $\mathcal{S}$, and the set of matrices $\mathcal{U}$ with diagonal values all equal to one. The alternating projections method stated that by repeatedly projecting $\mathbf{A}$ onto $\mathcal{S}$ and then onto $\mathcal{U}$, we could finally obtain the nearest matrix $\mathbf{X}$ on the intersection of $\mathcal{S}$ and $\mathcal{U}$ (Boyle and Dykstra 1986). Furthermore, to achieve the convergence to the optimal solution, a correction needs to be made according to Dykstra (1983). In general, the alternating projections algorithm converges linearly at best (Deutsch and Hundal 1997) and can be applied under the F-norm as well as the W-norm. It shows good potential on the H-norm if an efficient algorithm of projection onto $\mathcal{S}$ can be found.

Higham (2002) suggested another way to tackle this problem, that is, to formulate the optimisation as a Semidefinite Programming (SDP), which can then be solved using any powerful SDP solver. For the F-norm, the object $||\mathbf{A}-\mathbf{X}||^2$ is a second-order function of $\mathbf{X}$, so a transformation into the standard SDP primal form is needed before any general SDP solver can be applied. According to Higham (2002), the problem can be written as a standard SDP with $(n^2+n+2)^2$ variables and $\frac{1}{2}n^4+\frac{1}{2}3n^2+n+1$ constraints. Although the SDP problem can be further reformulated and simplified, to our knowledge, there is still currently no SDP solver that is efficient enough to solve such an SDP easily.

Rebonato and Jäckel (2011) introduced a method to produce a feasible correlation matrix near $\mathbf{A}$ based on decomposition of a valid correlation matrix by representing it

using angular parameters on a multidimensional unit hypersphere. The constraints on $\mathbf{X}$ are therefore automatically satisfied. Then, by solving an optimisation problem over the parameters, we expect to recover a feasible $\mathbf{X}$ near $\mathbf{A}$ under a norm of our choice. One advantage of the Hypersphere Decomposition Method (HDM) is that this algorithm allows us to perform PSDisation with the H-norm, so we are able to weight each entry with more flexibility. Furthermore, the computation is efficient and the converge speed is fast on small matrices. However, according to Milhaud et al. (2018), the hypersphere decomposition method may suffer from local optimum, making the solution less accurate. Several side effects are discussed in their paper. For instance, when changing the order of the input variables, the output of this method differs, which should not happen to any proper PSDisation algorithms theoretically.

The Spectral Decomposition Method (SDM) (Rebonato and Jäckel 2011) is a slightly rough approach but can give an acceptable solution even faster. The idea is to set all negative eigenvalues to zero after performing a spectral decomposition of the initial matrix. A rescaling is then carried out to ensure the resulting correlation matrix has diagonal elements all equal to one. The procedure is extremely fast as it contains no iterations. The solution found by the spectral decomposition method can be an approximation of the true optimal of the PSDisation problem, or can be used as a starting point for other PSDisation methods (Rebonato and Jäckel 2011). However, the result cannot be considered as accurate and it lacks the flexibility to be adapt with different choice of norms.

Qi and Sun (2006) introduced a Newton-type algorithm to find the nearest correlation matrix under the F-norm. It is designed to work with the F-norm as well as the W-norm using the semismooth Newton method combined with the conjugate gradient solver. It has a quadratic convergence rate, which is superior to the APM, and is effective on data with thousands of dimensions. Further improvements in this method were suggested by Borsdorf and Higham (2010). Qi and Sun (2011) extended their work to the H-norm case based on an augmented Lagrangian Dual Approach (ALD) by avoiding computing the projection onto the PSD cone under the H-norm. An upper bound or lower bound for each entry of the correlation matrix can also be set.

Ledoit and Wolf (2004) proposed a shrinkage method for estimating the covariance matrix in order to fix the ill-conditioned sample covariance matrix calculated from observations in high-dimensional settings, which is likely to occur when the number of samples available is smaller than the number of variables. It achieves higher accuracy and better stability by shrinking the sample covariance matrix towards a simple structured target matrix. This significantly reduces the variance of the estimator without introducing too much bias. The degree of shrinkage is called shrinkage intensity and can be optimised to minimise the MSE in a closed form. The shrinkage method has been particularly useful in finance applications such as portfolio optimisation or risk management. It is also commonly adopted in statistics or and machine learning fields such as genomics and bioinformatic analysis. Higham et al. (2016) developed a shrinking method using a similar idea for restoring positive semidefiniteness. It was designed to handle the correlation matrix, where one wants to keep some diagonal blocks fixed. Instead of minimising MSE using the observation data, it uses a bisection method to find the optimal solution to the PSDisation problem that lies on the path between the initial non-PSD matrix $\mathbf{A}$ and a target PSD matrix. The bisection method enjoys great simplicity and requires few iterations before achieving a good tolerance. However, the outcome of this method depends heavily on the choice of the target matrix. Indeed, the target matrix can be chosen as any PSD matrix that is near to $\mathbf{A}$. Therefore, the shrinking method is not considered suitable for finding the global optimal of the PSDisation problem. Cutajar et al. (2017) explored the shrinking method as a way to improve the results of the alternating projections method. Furthermore, any block-structured

constraints that are linear can be applied in this method. We will include the more general and widely used shrinkage method by Ledoit and Wolf (2004) in our experiments to see how our PSDisation algorithms and other state-of-the-art methods perform on real data applications in finance and machine learning comparing with this technique.

Overall, most algorithms may become computationally expensive when additional constraints are introduced to the problem. To the best of our knowledge, there is currently no algorithm that is able to solve the general PSDisation problem under the Chebyshev norm efficiently.

4. Proposed PSDisation Methods

In this section, we propose two methods to approximate the nearest correlation matrix. Currently most PSDisation methods are designed specifically for the F-norm. Some of them can be adapted to work with the H-norm with the cost of a more complex formulation or a longer computational time. We want to produce an algorithm that is more flexible with the objective and the constraints and more efficient than traditional SDP solvers. Therefore we propose our first proposed method, which involves iteratively solving a series of linearly constrained quadratic optimisations or linear optimisations. We obtain an approximation of the optimal value of the PSDisation problem by directly minimising the objective function at each iteration, where we have the flexibility to easily choose different norms to work with. The other approach we propose is based on iteratively taking a step along the gradient to reduce the objective function and then projecting the resulting matrix onto the PSD matrices set with diagonal one according to the spectral decomposition method (Rebonato and Jäckel 2011), which is designed for solving the PSDisation problem with respect to the F-norm. Experiments show that our gradient descent approach enjoys good efficiency and robustness. MATLAB implements for both algorithms are made available online.

4.1. Iterative Quadratic/Linear Programming

Consider the PSDisation problem in (1), which requires $\mathbf{X}$ to be PSD in its constraints. The PSD constraint can be reduced; therefore, the optimisation problem is transferred into quadratic or linear optimisation problems with linear constraints.

A square matrix $\mathbf{X} \in \mathbb{R}^{n\times n}$ is called diagonally dominant if

$$|x_{ii}| \geq \sum_{j\neq i} |x_{ij}| \;\; \forall\, 1 \leq i \leq n,$$

where x_{ij} is the (i,j) entry of $\mathbf{X}$. It is well known that a symmetric diagonally dominant matrix with non-negative diagonal entries is PSD, which makes up a rich subset of PSD matrices. Therefore, we could approximate the optimisation problem (1) stated above by solving convex optimisation problems of the following form in an iterative manner,

$$\begin{aligned} \min_{\mathbf{X},\mathbf{Q}} \; & ||\mathbf{A}-\mathbf{X}||^2 \\ s.t. \; & \text{diag}(\mathbf{X}) = \mathbf{1}, \\ & \mathbf{X} = \mathbf{U}_h^T \mathbf{Q} \mathbf{U}_h, \\ & \mathbf{Q} = \mathbf{Q}^T, \\ & \mathbf{Q} \text{ is diagonally dominant with non-negative diagonal.} \end{aligned} \tag{2}$$

Note that $\mathbf{U}_1 = \mathbf{I}$ and $\mathbf{U}_h = \text{Chol}(\mathbf{X}^*_{h-1})$ for $h \geq 2$, where $\mathbf{X}^*_{h-1}$ is the optimal solution from the previous optimisation. $\mathbf{U} = \text{Chol}(\mathbf{X})$ represents the Cholesky decomposition of $\mathbf{X}$ such that $\mathbf{U}$ is an upper triangular matrix satisfying $\mathbf{X} = \mathbf{U}^T\mathbf{U}$.

Lemma 1. *Problem* (2) *gives an upper bound of Problem* (1).

Proof. The matrix $\mathbf{Q}$ is guaranteed to be PSD in (2) since it is symmetric and diagonally dominant with non-negative diagonal entries; therefore, $\mathbf{X} = \mathbf{U}_h^T \mathbf{Q} \mathbf{U}_h$ is also PSD. Problem (2) minimises the objective function over a subset of PSD matrices; therefore, Lemma 1 holds. □

Lemma 2. *Problem* (2) *is feasible for iteration* h ($h \geq 2$) *if it is feasible for iteration* $h-1$.

Proof. This proof is trivial as $\mathbf{X}_{h-1}^* = \mathbf{U}_h^T \mathbf{I} \mathbf{U}_h$ and the identity matrix $\mathbf{I}$ is symmetric and diagonally dominant. Therefore, the solution of problem in iteration $h-1$ is feasible for iteration h. □

Let us denote the objective function by $f(\cdot)$:

$$f(\mathbf{X}) = ||\mathbf{A} - \mathbf{X}||^2.$$

Lemmas 1 and 2 indicate that an optimal solution always exists for (2) as long as the problem in the first problem is feasible and the optimal solution in each iteration is at least as good as that from the previous iteration, i.e.,

$$f(\mathbf{X}_h^*) \leq f(\mathbf{X}_{h-1}^*) \quad \forall h \geq 2.$$

Given that the problems are feasible and the optimal solution is PSD (which is true when numerical computation is concerned), it can be further shown that the optimal value of Problem (2) decreases strictly after each iteration unless it reaches the optimal value of Problem (1) (Ahmadi and Hall 2017). In this case, the optimal objective value in each iteration in Problem (2) will finally converge as it is monotonically decreasing and bounded below by the true optimal value of Problem (1). In our numerical experiments, the optimal after each iteration always finally converges to the true optimal and strong empirical evidence shows a fast convergence. Our proposed method is shown in Figure 1 and Algorithm 1 below.

Let us take the F-norm as an example. Note that $||\mathbf{A} - \mathbf{X}||_{\mathbf{F}}^2 = \boldsymbol{x}^T \boldsymbol{x} - 2\boldsymbol{a}^T \boldsymbol{x} + \boldsymbol{a}^T \boldsymbol{a}$ where $\boldsymbol{x} = \text{vec}(\mathbf{X})$ and $\boldsymbol{a} = \text{vec}(\mathbf{A})$ and vec is the vectorisation function that transforms a matrix into a column vector. We further rewrite the objective function and the constraint that $\mathbf{Q}$ is diagonally dominant with non-negative diagonal by adding slack variables $\mathbf{R}$ into the optimisation,

$$\begin{aligned}
\min_{\mathbf{X},\mathbf{Q},\mathbf{R}} \ & \frac{1}{2}\boldsymbol{x}^T \boldsymbol{x} - \boldsymbol{a}^T \boldsymbol{x} \\
s.t. \ & \text{diag}(\mathbf{X}) = \mathbf{1}, \\
& q_{ii} \geq \sum_{j \neq i} r_{ij}, && 1 \leq i \leq n \\
& -r_{ij} \leq q_{ij} \leq r_{ij}, && 1 \leq i \neq j \leq n \\
& \mathbf{X} = \mathbf{U}_h^T \mathbf{Q} \mathbf{U}_h, \\
& \mathbf{Q} = \mathbf{Q}^T,
\end{aligned}$$

where q_{ij} and r_{ij} are the (i,j) entries of $\mathbf{Q}$ and $\mathbf{R}$, respectively. Now, all constraints become linear and we obtain a series of iterative QPs. Each QP has $3n^2$ variables and $4n^2 - n$ constraints, which can be solved efficiently.

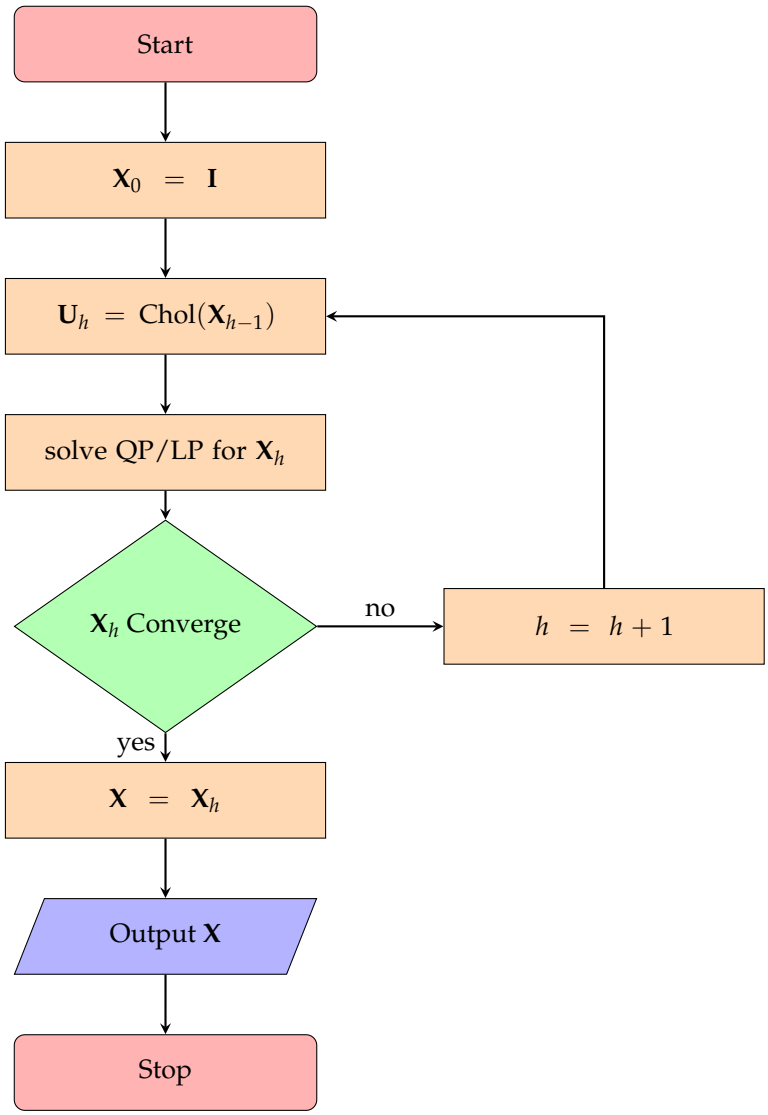

Figure 1. Flowchart of Algorithm 1: An iterative quadratic/linear programming algorithm to solve PSDisation under some matrix norm.

Algorithm 1: An iterative quadratic/linear programming algorithm to solve PSDisation under some matrix norm.

$\mathbf{X}_0 = \mathbf{I}$;
while *not converged* **do**
  $\mathbf{U}_h = \text{Chol}(\mathbf{X}_{h-1})$;
  find $\mathbf{X}_h$ by solving

$$\min_{\mathbf{X}_h, \mathbf{Q}_h} ||\mathbf{A} - \mathbf{X}_h||^2$$
$$s.t.\ \text{diag}(\mathbf{X}_h) = \mathbf{1},$$
$$\mathbf{X}_h = \mathbf{U}_h^T \mathbf{Q}_h \mathbf{U}_h,$$
$$\mathbf{Q}_h = \mathbf{Q}_h^T,$$
$$\mathbf{Q}_h \text{ is diagonally dominant with non-negative diagonal;}$$

  $h = h + 1$;
end

Similarly, if the H-norm is applied in the objective function, the iterative QPs can be written as

$$\begin{aligned}
\min_{\mathbf{X},\mathbf{Q},\mathbf{R}} \ & \frac{1}{2}\boldsymbol{x}^T \operatorname{Diag}(\boldsymbol{h})\boldsymbol{x} - \boldsymbol{a}^T \operatorname{Diag}(\boldsymbol{h})\boldsymbol{x} \\
s.t. \ & \operatorname{diag}(\mathbf{X}) = \mathbf{1}, \\
& q_{ii} \geq \sum_{j \neq i} r_{ij}, && 1 \leq i \leq n \\
& -r_{ij} \leq q_{ij} \leq r_{ij}, && 1 \leq i \neq j \leq n \\
& \mathbf{X} = \mathbf{U}_h^T \mathbf{Q} \mathbf{U}_h, \\
& \mathbf{Q} = \mathbf{Q}^T.
\end{aligned}$$

where $\boldsymbol{h} = \operatorname{vec}(\mathbf{H})$ and $\operatorname{Diag}(\boldsymbol{h})$ is an $n^2 \times n^2$ diagonal matrix with the elements of $\boldsymbol{h}$ on the main diagonal. This general formulation makes our method more flexible when an insurance company wants to assign weight to each entry of the correlation matrix, exploiting prior knowledge of the data.

Next, we consider the situation where we optimise with respect to the Chebyshev norm. According to Cutajar et al. (2017), PSDisation over the F-norm is likely to result in a correlation matrix $\mathbf{X}$ in which some entries differ significantly from the initial matrix $\mathbf{A}$, while other entries have relatively smaller deviations, giving the minimum F-norm optimal. However, this is not preferred by some insurance companies as they aim to minimise the maximum discrepancy between corresponding entries from the valid correlation matrix $\mathbf{X}$ and the initial matrix $\mathbf{A}$. In this case, minimising over the Chebyshev norm becomes an alternative choice. With our iterative approximating method, this can be obtained by solving a series of LPs of the following form:

$$\begin{aligned}
\min_{\mathbf{X},\mathbf{Q},\mathbf{R},t} \ & t \\
s.t. \ & -t \leq a_{ij} - x_{ij} \leq t, && 1 \leq i \neq j \leq n \\
& \operatorname{diag}(\mathbf{X}) = \mathbf{1}, \\
& q_{ii} \geq \sum_{j \neq i} r_{ij}, && 1 \leq i \leq n \\
& -r_{ij} \leq q_{ij} \leq r_{ij}, && 1 \leq i \neq j \leq n \\
& \mathbf{X} = \mathbf{U}_h^T \mathbf{Q} \mathbf{U}_h, \\
& \mathbf{Q} = \mathbf{Q}^T.
\end{aligned}$$

Note that in this formulation, minimising $||\mathbf{A} - \mathbf{X}||_{\mathbf{C}}^2$ can be equivalently accomplished by minimising $||\mathbf{A} - \mathbf{X}||_{\mathbf{C}}$; thus, only LPs instead of QPs need to be solved.

Experiments show that our iterative algorithm proposed for approximating optimal solutions for PSDisation problems offers great flexibility under different norms while maintaining a good accuracy. Details are offered in Section 5.

4.2. Gradient Descent

Next we propose our second algorithm, where we will focus on working under the F-norm only as there is currently no known method that could project $\mathbf{X}$ onto the PSD set under the H-norm. This method is therefore less flexible in the choice of norms. However, it works more efficiently and provides more steady results on real data with noise.

Consider the objective in (1) and note that the gradient of its objective function under the F-norm is $\frac{\partial}{\partial \mathbf{X}}||\mathbf{A} - \mathbf{X}||_{\mathbf{F}} = \frac{\mathbf{X}-\mathbf{A}}{||\mathbf{A}-\mathbf{X}||_{\mathbf{F}}}$. To project a matrix $\mathbf{X}$ onto the PSD cone $\mathcal{S}_+^n$, first perform the eigendecomposition such that $\mathbf{X} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1}$, where $\mathbf{Q} \in \mathbb{R}^{n \times n}$ is the square

matrix whose columns are eigenvectors of $\mathbf{X}$, and $\mathbf{\Lambda} = \mathrm{Diag}(\boldsymbol{\lambda})$ is the diagonal matrix whose diagonal elements are the corresponding eigenvalues λ_i of $\mathbf{X}$. Define $\mathbf{\Lambda}_+ = \mathrm{Diag}(\boldsymbol{\lambda}_+)$, where $\lambda_{+i} = \max(\lambda_i, 0)$. Next, in order to ensure that the resulting matrix has diagonal $\mathbf{1}$, we calculate the scaling matrix $\mathbf{T} = \mathrm{Diag}(\boldsymbol{t})$, where the weighting parameter t_i is given by

$$t_i = \left(\sum_{m=1}^{n} q_{im}^2 \lambda_{+m} \right)^{-1/2}.$$

Then, $\mathbf{X}_+ = \mathbf{T}\mathbf{Q}\mathbf{\Lambda}_+\mathbf{Q}^{-1}\mathbf{T}$ is the projection of $\mathbf{X}$ onto the cone of PSD matrices.

Our proposed method is shown in Figure 2 and Algorithm 2 below. $\mathbf{X}$ is updated iteratively by taking a gradient step followed by a projection onto the PSD cone until converged. According to our experiments, the gradient step γ can be initialised as a large value to speed up the convergence, and it will quickly decrease during iterations.

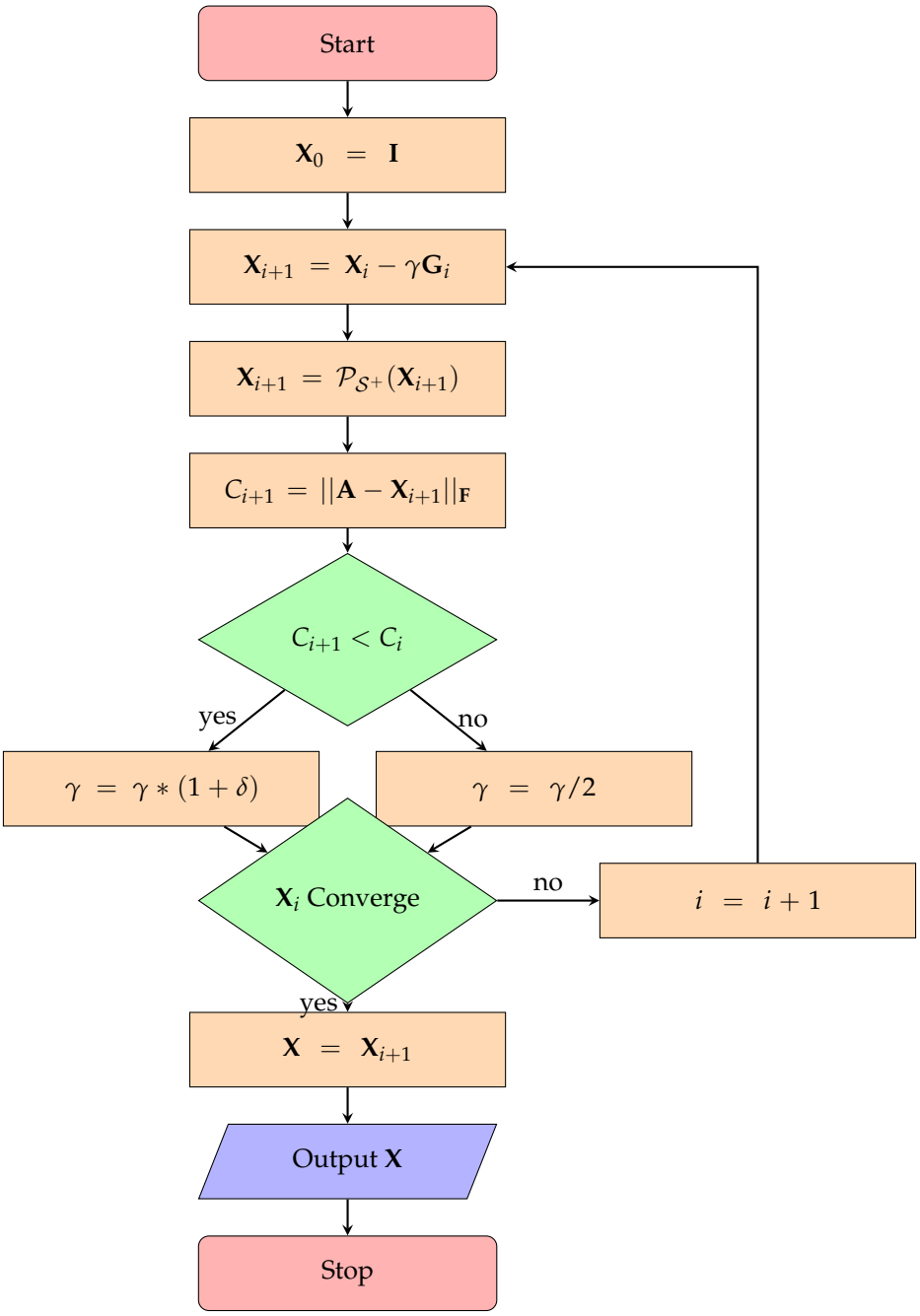

Figure 2. Flowchart of Algorithm 2: The gradient descent algorithm for PSDisation under the F-norm.

Algorithm 2: The gradient descent algorithm for PSDisation under the F-norm

$\mathbf{X}_0 = \mathbf{I}$;
while *not converged* **do**
  $\mathbf{X}_{i+1} = \mathbf{X}_i - \gamma \mathbf{G}_i$;
  $\mathbf{X}_{i+1} = \mathcal{P}_{\mathcal{S}^+}(\mathbf{X}_{i+1})$;
  $C_{i+1} = ||\mathbf{A} - \mathbf{X}_{i+1}||_{\mathbf{F}}$;
  if $C_{i+1} < C_i$ **then**
    $\gamma = \gamma * (1 + \delta)$;
  else
    $\gamma = \gamma / 2$;
  end
  $i = i + 1$;
end

5. Experimental Results on Simulated Data

To illustrate the robustness and flexibility of our algorithms in different settings, we designed a series of tests based on both simulated and real datasets. In this section, we explain and show the experimental results on simulated data and compare our methods with the state-of-the-art methods. In the next section, we give examples of some practical applications. All experiments were run using MATLAB R2023a Prerelease (9.14.0.2137306) with Financial Toolbox v6.5, Optimization Toolbox v9.5, and Statistics and Machine Learning Toolbox v12.5, and Gurobi Optimizer 10.0.1.

5.1. Experiments on Iterative Quadratic/Linear Programming Algorithm with Simulated Data

To test our iterative quadratic/linear programming algorithm, we randomly generate the initial matrix $\mathbf{A}$ of different dimensions (i.e., with dimensions of 5, 10, 25, 50, and 75) that is not PSD. For $\mathbf{A}$ generated from each dimension settings, we solve for the nearest valid correlation matrix $\mathbf{X}$ using our method under the F-norm and the Chebyshev norm, which lead to iterative quadratic programming (IQP) and iterative linear programming (ILP), respectively. The stopping criteria is set to achieve a solution with error less than 0.1% in each iteration. We calculate the Frobenius distance and the Chebyshev distance between $\mathbf{X}$ and $\mathbf{A}$ from the IQP and ILP and compare with those given by the APM, the Newton method and the SDM, which are all designed for minimising the F-norm. The APM and the Newton method work very fast and can both achieve solutions with accuracy tolerance far lower than 0.0001; therefore, the experiment settings have no big impact on testing results. We just use the default settings by the authors here (Higham 2002; Qi and Sun 2006). The results are shown in Tables 1 and 2 for the F-norm and the Chebyshev norm, respectively. The best performances in each group are shown in bold.

Table 1. The F-norm $||\mathbf{A} - \mathbf{X}||_{\mathbf{F}}$ obtained from different algorithms on different dimensions of the initial matrix $\mathbf{A}$. (Best performances are marked in bold).

Dimension of A	**5**	**10**	**25**	**50**	**75**
Newton	**0.3420**	**0.8341**	**5.0057**	**13.0331**	**20.6889**
APM	**0.3420**	**0.8341**	**5.0057**	**13.0331**	**20.6889**
SDM	0.3537	0.8602	5.1716	13.5887	21.5025
IQP	**0.3420**	0.8352	5.0245	13.2406	21.2224
ILP	0.3819	1.0815	7.6554	15.9193	24.9860

Table 2. The Chebyshev norm $||\mathbf{A}-\mathbf{X}||_{\mathbf{C}}$ obtained from different algorithms on different dimensions of the initial matrix $\mathbf{A}$. (Best performances are marked in bold).

Dimension of A	**5**	**10**	**25**	**50**	**75**
Newton	0.1313	0.2128	0.6473	0.7605	0.8313
APM	0.1313	0.2128	0.6473	0.7605	0.8313
SDM	0.1416	0.2265	0.6225	0.7278	0.7512
IQP	0.1304	0.2131	0.6245	0.7314	0.8397
ILP	**0.0854**	**0.1375**	**0.4293**	**0.4310**	**0.4805**

To give a clearer view on how these algorithms compare with each other, we use the Frobenius distance and the Chebyshev distance obtained via the Newton method as a benchmark and calculate the relative change of the two distances obtained from other algorithms in percentage. For example, the relative change of the Frobenius distance obtained from the IQP algorithm compared to the Newton method is given by $\frac{||\mathbf{A}-\mathbf{X}_{\text{IQP}}||_{\mathbf{F}}-||\mathbf{A}-\mathbf{X}_{\text{Newton}}||_{\mathbf{F}}}{||\mathbf{A}-\mathbf{X}_{\text{Newton}}||_{\mathbf{F}}} \times 100\%$. In Tables 3 and 4, a negative percentage means that the optimal solution obtained from this method is better than that from the Newton method in the corresponding norm, while a positive percentage represents a worse result.

Table 3. Relative change in the F-norm $||\mathbf{A}-\mathbf{X}||_{\mathbf{F}}$ compared with the Newton method as a percentage. (Best performances are marked in bold).

Dimension of A	**5**	**10**	**25**	**50**	**75**
Newton	**0**	**0**	**0**	**0**	**0**
APM	**0**	**0**	**0**	**0**	**0**
SDM	3.43	3.13	3.31	4.26	3.93
IQP	0.02	0.14	0.3763	1.59	2.58
ILP	11.67	29.66	52.93	22.15	20.77

Table 4. Relative change in the Chebyshev norm $||\mathbf{A}-\mathbf{X}||_{\mathbf{C}}$ compared with the Newton method as a percentage. (Best performances are marked in bold).

Dimension of A	**5**	**10**	**25**	**50**	**75**
Newton	0	0	0	0	0
APM	0	0	0	0	0
SDM	7.79	6.45	−3.84	−4.30	−9.64
IQP	−0.70	0.16	−3.52	−3.82	1.01
ILP	**−34.98**	**−35.37**	**−33.68**	**−43.33**	**−42.21**

Furthermore, we report the computing time of each run in Table 5. The lowest running times are shown in bold.

It can be easily concluded from Table 3 (and also see Table 5 for efficiency) that both the APM and the Newton method give solutions with the minimum Frobenius distance, which they aim to minimise, while the Newton method works slightly faster in high dimensions. (Further experiments show that the ALD approach also gives the same optimal solutions, which we omit in the above tables.) The SDM, on the other hand, is not as accurate as the Newton method and the APM but is the fastest algorithm overall as it does not require iterations. The IQP method gives solutions slightly worse than the Newton method in terms of the F-norm, but the difference is not significant, especially in low dimensions. These

results from the IQP can be improved by setting a lower tolerance and hence increasing the number of iterations but will cost more time to converge. The ILP method is designed to minimise with respect to the Chebyshev norm and therefore does not give solid results in the F-norm as other algorithms mentioned above. However, the nearest correlation matrix generated by the ILP method has a much lower Chebyshev distance than by the Newton method which intends to minimise the F-norm (see Table 4). This result indicates that minimising over the Chebyshev norm can indeed be a good alternative for PSDisation since considering the F-norm will potentially result in huge deviations in some entries of $\mathbf{X}$, which are not favoured by some insurance companies (Cutajar et al. 2017). It is worth mentioning that both the IQP and ILP methods take a long time to run when the dimension of $\mathbf{A}$ is high, this could potentially be improved by optimising the QP and LP solvers in the future work. For example, possible improvements could be to carefully track the optimisations and keep an active set to significantly reduce the number of constraints or to manually set a starting point for each optimisation based on the solution from the previous iteration.

Table 5. Running time of different algorithms on different dimensions of the initial matrix $\mathbf{A}$ (in seconds). (Best performances are marked in bold).

Dimension of A	**5**	**10**	**25**	**50**	**75**
Newton	0.0024	0.0038	0.0058	0.0062	0.0350
APM	0.0005	0.0010	0.0068	0.0281	0.0615
SDM	**0.0001**	**0.0001**	**0.0003**	**0.0008**	**0.0025**
IQP	0.0315	0.1260	3.84	546	8180
ILP	0.0206	0.0937	2.35	719	5661

To illustrate the flexibility of our IQP algorithm and its potential to achieve good accuracy, our next step is to test when weights are assigned to each coefficient, i.e., to work under the H-norm. Similarly as before, we generate the initial matrix $\mathbf{A}$ of different dimensions. This time, we apply the ALD method and our IQP algorithm to find the nearest correlation matrix to $\mathbf{A}$ under the F-norm and the H-norm, respectively, where each entry of the weight matrix $\mathbf{H}$ is randomly generated from a uniform distribution. We also test these methods on PSDisation problems with constraints; that is, we let $x_{ij} = 0$ if $|a_{ij}| < 0.1$, and we also request $x_{ij} > 0$ if $a_{ij} > 0.5$ and $x_{ij} < 0$ if $a_{ij} < -0.5$. Table 6 shows the optimal values under different settings.

It can be concluded that both ALD and IQP work effectively on constrained or unconstrained PSDisation problems based on the F-norm or the H-norm. Our IQP algorithm runs slowly when dimension is high and produces slightly worse results than ALD, but the optimal values are still competitive. For example, when working under the F-norm with dimension 10, IQP gives optimal solutions of 2.5060 and 2.5411 in the unconstrained and constrained PSDisation tasks, respectively, which are close to those given by ALD, 2.5032 and 2.5401. On the other hand, IQP gives slightly better results than ALD in constrained problems with the H-norm when the dimension is low. For example, in the constrained task under the H-norm with dimension 10, the optimal solution given by IQP is 1.6501, which is better than 1.6631 by ALD. In addition, our iterative algorithm shows more flexibility when problem settings are combined with the H-norm (IQP) and the Chebyshev norm (ILQ), where the ALD method becomes infeasible.

Table 6. Optimal solution by ALD and IQP for constrained and unconstrained PSDisation problems under the F-norm and the H-norm on different dimensions of the initial matrix **A**. (Best performances are marked in bold).

Dimension of A				**10**	**20**	**40**	**70**
# of equality constraints (=0)				4	19	74	263
# of inequality constraints (<0/>0)				25	92	367	1208
Optimal solution	Unconstrained	F-norm	ALD	**2.5032**	**6.2625**	**14.8958**	**30.2514**
			IQP	2.5060	6.2789	15.1565	30.7230
		H-norm	ALD	**1.6149**	**3.9136**	**10.0493**	**20.3044**
			IQP	1.6188	3.9360	10.2596	20.8868
	Constrained	F-norm	ALD	**2.5401**	**6.4399**	**15.2334**	**30.8318**
			IQP	2.5411	6.4562	15.3771	31.2218
		H-norm	ALD	1.6631	4.0789	10.4666	**20.9562**
			IQP	**1.6501**	**4.0707**	**10.4355**	21.1677

5.2. Experiments on Gradient Descent Method with Simulated Data

In order to test the performance of the gradient descent method on PSDisation problems, we compare ALD, IQP, GD, and SDM on unconstrained problems of different dimensions under the F-norm. SDM is a trivial method that projects the matrix onto the PSD cone directly and can therefore be considered as a baseline for this experiment. Table 7 compares the convergence times and optimal solutions of all four algorithms.

Table 7. Convergence time and optimal solution by ALD, IQP, GD, and SDM for unconstrained PSDisation problems under the F-norm on different dimensions of the initial matrix **A**. (Best performances are marked in bold).

Dimension of A		**10**	**20**	**40**	**70**	**100**	**200**
Convergence time (unconstrained, F-norm)	ALD	0.0029 s	0.0027 s	0.0120 s	0.0076 s	0.0163 s	0.0364 s
	IQP	0.17 s	1.52 s	2 min 19 s	2 h 29 min	-	-
	GD	0.03 s	0.06 s	0.16 s	0.07 s	0.08 s	0.26 s
	SDM	**0.0003 s**	**0.0004 s**	**0.0007 s**	**0.0011 s**	**0.0017 s**	**0.0047 s**
Optimal solution (unconstrained, F-norm)	ALD	**2.5032**	**6.2625**	**14.8958**	**30.2514**	**44.9199**	**96.3681**
	IQP	2.5060	6.2789	15.1565	30.7230	-	-
	GD	2.5316	6.2754	15.0992	30.8536	46.0570	98.7712
	SDM	2.5667	6.5599	15.5756	31.3282	46.5758	99.3325

As shown in Table 7, GD provides comparable results to ALD. Though slightly slower than ALD, GD works much more efficient than general SDP solvers. Furthermore, we want to illustrate the advantages of the Gradient Descent (GD) method over the ALD algorithm. Consider the situation where in practice, the correlation matrices obtained by some insurance companies are usually inaccurate. We expect our PSDisation algorithm to work better when error exists in the input matrix according to the testing results.

In the following experiment, we randomly generate a correlation matrix **A** that is not PSD. Noise from normal distribution is then added to each non-diagonal entry of **A** and $\mathbf{A}'$ represents the estimated correlation matrix. We use GD, ALD, and IQP to calculate the nearest PSD correlation matrix **X** to $\mathbf{A}'$ under the F-norm and compare $||\mathbf{A} - \mathbf{X}||_{\mathbf{F}}$, the distance between **X**, and the actual initial matrix **A**.

Table 8 shows the average results of the three algorithms tested on 100 different initial matrices of dimension 40 with noise of standard deviation 0.2. It can be concluded that

while GD provides slightly inaccurate solutions to the PSDisation optimisation problem, it is less influenced by noise and thus leads to better results than ALD overall. In all 100 testing examples, GD gives matrix $\mathbf{X}$, which is closer to the initial matrix $\mathbf{A}$. In practice, GD can be more resilient to noise, with a larger standard deviation than ALD. Details of the experimental results are given in Figure 3.

Table 8. Average Frobenius distance from $\mathbf{X}$ to $\mathbf{A}$ and $\mathbf{A}'$ by different algorithms over 100 tests in experiment settings. (Best performances are marked in bold).

	$\|\|\mathbf{A}' - \mathbf{X}\|\|_{\mathbf{F}}$	$\|\|\mathbf{A} - \mathbf{X}\|\|_{\mathbf{F}}$
ALD	**4.43**	5.70
IQP	4.45	5.72
GD	4.60	**5.49**

Figure 3 plots the percentages of the best performance each PSDisation algorithm achieves with different dimensions of the correlation matrix and different standard deviations of noise. The best performance is decided by giving the shortest distance between $\mathbf{X}$ and $\mathbf{A}$, where $\mathbf{X}$ is calculated by applying different PSDisation algorithms on the noised matrix $\mathbf{A}'$. The tests are repeated 100 times for matrices of different dimensions and noise of different levels.

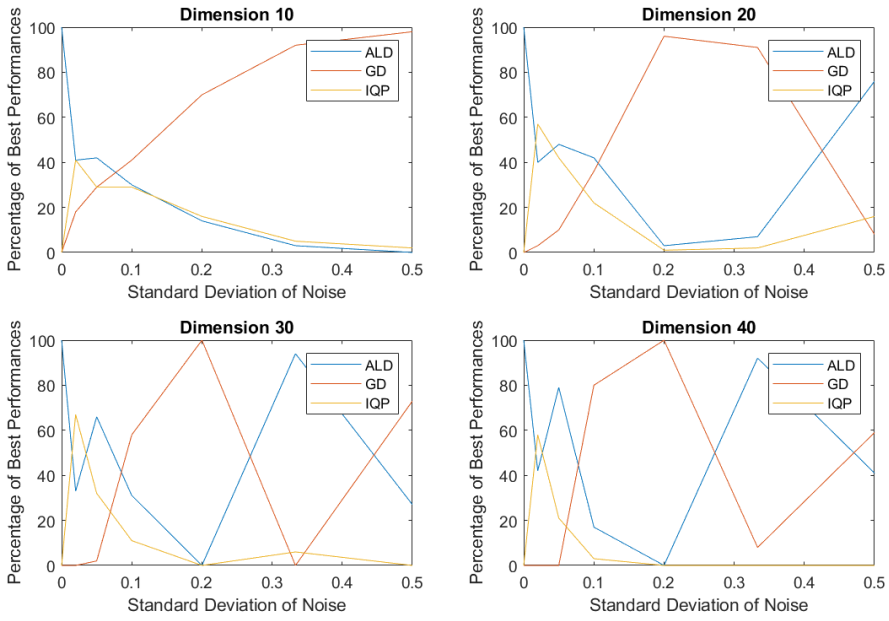

Figure 3. Percentage of best performances by ALD, GD, and IQP versus the standard deviation of the noise in PSDisation tests with different dimensions of the initial matrix $\mathbf{A}$.

It can be concluded that despite the fact that ALD performs best on accurate initial matrices (when noise does not exist), IQP and GD can be good alternatives when the input comes with noise, especially for matrices with large dimension. IQP works well when the noise is small; e.g., in 67% of our tests, IQP gives the smallest $||\mathbf{A} - \mathbf{X}||_{\mathbf{F}}$ for matrices with dimension 30 and noise with standard deviation 0.02. When the noise becomes slightly larger (for a standard deviation between 0.1 and 0.2), GD performs significantly better.

5.3. Experiments with Simulated Data Rounded to Multiples of 25%

We present results of comparisons between ALD, IQP, GD, and SDM on PSDisation problems under the F-norm in practical settings where entries of the correlation matrix

are rounded. Without knowing the actual correlations, the way the European Commission creates the correlation matrix for the solvency capital requirement insurance models is by picking the best possible choices for the correlation matrix entries from the set $\{-75\%, -50\%, -25\%, 0\%, 25\%, 50\%, 75\%\}$. The correlation matrix $\mathbf{A}$ estimated this way is usually not PSD, but certain entries of $\mathbf{A}$ can be expected to be positive or negative as actuaries expect some risks to be positively or negatively correlated based on domain knowledge.

Inspired by the above practical scenario, our experiment is designed as follows. First, a PSD matrix $\mathbf{A}_t$ is generated, which we assume is the true correlation matrix. Then 250 observations are generated from the t-distribution with three degrees of freedom and are used to calculate the sample correlation matrix $\mathbf{A}_s$. We obtain our estimated correlation matrix $\mathbf{A}'_s$ by rounding each entry of the sample correlation matrix $\mathbf{A}_s$ to 25%. We also round the true correlation matrix $\mathbf{A}_t$ to 25% to obtain $\mathbf{A}'_t$. Now, $\mathbf{A}'_s$ and $\mathbf{A}'_t$ are not PSD. We perform an unconstrained PSDisation on $\mathbf{A}'_s$ under the F-norm, and the solution is denoted as $\mathbf{X}$. We also test constrained PSDisation by ALD and IQP. The constraints are set so that the entries of $\mathbf{X}$ have the same signs as $\mathbf{A}_s$, which we assume are available to actuaries as domain knowledge.

The tests are conducted repeatedly on matrices of dimensions 10, 20, 30, and 40. The unconstrained PSDisations are conducted using ALD, IQP, GD, and SDM. In our experiment settings, only around 85% to 89% of the non-diagonal entries of $\mathbf{X}$ have the same sign as $\mathbf{A}_s$. In the constrined settings, both ALD(con) and QP(con) can 100% satisfy the same sign constraints, either strictly or to some small tolerance. The results are shown in Tables 9 and 10.

It can be easily concluded from Table 9 that ALD is the best in terms of minimising the objective, $||\mathbf{A}'_s - \mathbf{X}||_\mathbf{F}$, while IQP could produce very similar results. On the other hand, GD gives slightly worse solutions than ALD and IQP, but still better than the baseline method SDM. Comparing $||\mathbf{A}_s - \mathbf{X}||_\mathbf{F}$, unsurprisingly, we see that constrained PSDisation shows its advantages if prior information is available. While ALD allows restrictions on upper or lower bounds of entries of $\mathbf{X}$, IQP could potentially allow for any linear constraints and thus can be more flexible. Nonlinear constraints can also be set if efficiency is not the main focus. For example, we could restrict the entries of $\mathbf{X}$ to be multiples of 25%. This leads to iterative mixed-integer programming, which is beyond the scope of this paper.

Table 9. Average Frobenius distance from $\mathbf{X}$ to $\mathbf{A}'_s$ and $\mathbf{A}'{}_s$ by different algorithms over 500 tests in experiment settings. (Best performances are marked in bold).

	n	**10**	**20**	**30**	**40**
$\|\|\mathbf{A}'_s - \mathbf{X}\|\|_\mathbf{F}$	**ALD**	**0.1387**	**0.3940**	**0.6991**	**1.0320**
	IQP	0.1388	0.3945	0.7003	1.0347
	GD	0.1453	0.4198	0.7512	1.1252
	SDM	0.1462	0.4247	0.7633	1.1479
	ALD(con)	0.1421	0.4081	0.7262	1.0735
	IQP(con)	0.1421	0.4086	0.7276	1.0760
$\|\|\mathbf{A}_s - \mathbf{X}\|\|_\mathbf{F}$	**ALD**	0.6429	1.2817	1.8862	2.4612
	IQP	0.6429	1.2820	1.8870	2.4633
	GD	0.6433	1.2837	1.8910	2.4796
	SDM	0.6440	1.2887	1.9019	2.5017
	ALD(con)	0.6351	**1.2499**	**1.8217**	**2.3611**
	IQP(con)	**0.6350**	1.2501	1.8223	2.3624

Table 10. Average Frobenius distance from $\mathbf{X}$ to $\mathbf{A}'_t$ and $\mathbf{A'}_t$ by different algorithms over 500 tests in experiment settings. (Best performances are marked in bold).

	n	**10**	**20**	**30**	**40**
$\|\|\mathbf{A}'_t - \mathbf{X}\|\|_\mathbf{F}$	**ALD**	1.6960	3.5396	5.2255	7.0881
	IQP	1.6960	3.5398	5.2261	7.0896
	GD	1.6851	3.4839	5.1133	6.8849
	SDM	**1.6814**	**3.4640**	**5.0635**	**6.7970**
	ALD(con)	1.6923	3.5240	5.1953	7.0412
	IQP(con)	1.6923	3.5240	5.1959	7.0425
$\|\|\mathbf{A}_t - \mathbf{X}\|\|_\mathbf{F}$	**ALD**	1.5294	3.2316	4.7367	6.4601
	IQP	1.5294	3.2318	4.7373	6.4617
	GD	1.5169	3.1710	4.6136	6.2364
	SDM	**1.5128**	**3.1499**	**4.5599**	**6.1427**
	ALD(con)	1.5252	3.2135	4.7015	6.4052
	IQP(con)	1.5252	3.2138	4.7022	6.4067

On the other hand, comparing $||\mathbf{A}'_t - \mathbf{X}||_\mathbf{F}$ and $||\mathbf{A}_t - \mathbf{X}||_\mathbf{F}$ in Table 10, SDM and GD constantly give PSD matrix $\mathbf{X}$, which is closer to the true correlation matrix $\mathbf{A}_t$. Therefore, we draw the conclusion that although relatively inaccurate in minimising the objective function, $||\mathbf{A}'_s - \mathbf{X}||_\mathbf{F}$, in practice, SDM and GD are more resilient to noise generated in real data, which coincides with our observations in the previous section.

6. Experiments Results on Real Datasets

The potential of our two proposed PSDisation methods has been illustrated with synthetic/simulated data. In this section, we test them with real-life data with applications in (financial) portfolio construction and dimension (data) reduction. We compare our IQP/ILP and GD with other state-of-the-art PSDisation algorithms. The shrinkage method is also included as it is widely used in finance, econometrics, and statistics to tackle the ill-conditioned or non-PSD matrix problem.

6.1. Experiments with Assets Daily Return Data

We now present an example of PSDisation applications with real-life financial data that consist of 50 public companies from the NASDAQ stock market that started to be listed in this marker at different points in time, i.e., assets have different observation periods that depend on the first trading day of each asset. That is, we use the daily asset returns since 2010 to calculate the pairwise correlation. This is useful when building a portfolio and calculating the portfolio's (aggregate) risk position measured through risk measures such as Value at Risk (VaR) and Conditional Value at Risk (CVaR). However, the resulting correlation matrix $\mathbf{A}$ is non-PSD and has negative eigenvalues since (i) missing values exist for each stock at certain periods of time, and (ii) each asset pair has different overlapping observation period. A solution to this issue is to use PSDisation algorithms as portfolio construction means, computing weights for each asset, and such computation requires PSD empirical correlation matrix. Alternatively, we can use some imputation techniques to replace the missing data with substituted values, but this does not guarantee a PSD empirical covariance matrix due to (ii). Imputation techniques are statistically based that require no domain knowledge, or one may rely on imputation methodologies that are specific to financial applications; e.g., one may replace the missing returns with risk-free

returns, which is the rate of the 10-year rated government Treasury note. Such imputations are the last resort solution with an unknowable impact on the application in hand.

We perform PSDisation on the calculated correlation matrix **A** to obtain **X**. The testing algorithms include IQP, ILP, GD, ALD, and SDM. The shrinkage method (Ledoit and Wolf 2004) is not designed to work with missing data. Therefore, we impute the data before applying the shrinkage method. We use two methods to fill the missing entries, namely, replacing them with mean observed values of each variable and filling using the multivariate imputation by chained equations method (MICE). Replacing with mean value is a simple approach to handling missing data but may not be appropriate for estimating correlations when there are too many missing entries. MICE uses a sequence of conditional models for imputation such that each variable is modelled conditionally on other variables. The shrinkage method is then used to estimate the correlation matrix **X**, and we call them Shrinkage-Mean and Shrinkage-MICE for the two different imputation strategies. We calculate the Frobenius distance and the Chebyshev distance between **A** and **X**, and the results are shown in Table 11.

Table 11. Frobenius distance and Chebyshev distance between **A** and **X** by different algorithms on NASDAQ stocks return correlation matrix. (Best performances are marked in bold).

	$\|\|\mathbf{A}-\mathbf{X}\|\|_{\mathbf{F}}$	$\|\|\mathbf{A}-\mathbf{X}\|\|_{\mathbf{C}}$
ILP	1.0836	**0.0236**
IQP	0.3765	0.0872
ALD	**0.3764**	0.0886
GD	0.3958	0.0836
SDM	0.4085	0.0931
Shrinkage-Mean	2.5625	0.4025
Shrinkage-MICE	2.3884	0.4367

The same experiment was also carried out with mixed choices of top 25 SP500 companies and 25 NASDAQ stocks, where the correlation matrix calculated is invalid. The results are shown in Table 12.

Table 12. Frobenius distance and Chebyshev distance between **A** and **X** by different algorithms on a mixture of NASDAQ and SP500 stocks return correlation matrix. (Best performances are marked in bold).

	$\|\|\mathbf{A}-\mathbf{X}\|\|_{\mathbf{F}}$	$\|\|\mathbf{A}-\mathbf{X}\|\|_{\mathbf{C}}$
ILP	0.5605	**0.0121**
IQP	0.2269	0.0542
ALD	**0.2268**	0.0547
GD	0.2488	0.0602
SDM	0.2534	0.0620
Shrinkage-Mean	2.4080	0.3935
Shrinkage-MICE	1.8739	0.3142

It can be concluded from Tables 11 and 12 that the optimal minimum Frobenius distance solution by our IQP method is very close to the state-of-the-art ALD method, while maintaining an even lower Chebyshev distance. This ensures that the maximum deviation of correlation for each pair of stocks by IQP is smaller than that by ALD. We also noticed that in this real financial data experiment, the speed of convergence of IQP is faster, and the optimal solution is closer to ALD compared with those in previous simulated data,

providing evidence that IQP is potentially suitable for dealing with real dataset. In addition, GD performs slightly better than the baseline method SDM in this stocks data experiment in terms of both the F-norm and the Chebyshev norm. In Table 11, GD achieves an even better Chebyshev norm with NASDAQ stocks than ALD, IQP, and SDM, although this is not guaranteed in all experimental settings. ILP gives a much lower Chebyshev norm than other methods as expected, since it is designed to do so. The shrinkage method is not designed to minimise any of the norms and thus does give comparable results under these criteria. However, it aims to give a more stable and well-conditioned estimate of the correlation matrix. We now evaluate how the above methods perform in the application of portfolio optimisation.

We divide the above assets return data into two parts. Data since January 2010 until June 2021 serve as historical observations from which to estimate the correlation matrix using either PSDisation algorithms or the shrinkage method after imputation. We then build a minimum variance portfolio with the correlation matrix from each method and test how it performs in the following 20 months. The annualised average return, annualised standard deviation, and the Sharpe ratio for the NASDAQ stock portfolio and the mixed portfolio are shown in Tables 13 and 14. An equal weight portfolio is provided as a benchmark.

Table 13. Annualised average return, annualised standard deviation, and Sharpe ratio of NASDAQ stocks minimum variance portfolios based on correlation matrices estimated using different methods. (Best performances are marked in bold).

Method	Average Return (%)	Standard Deviation (%)	Sharpe Ratio
ALD	−15.6085	19.7047	−0.7921
GD	−14.7107	19.3380	−0.7607
ILP	−13.6843	20.6439	−0.6629
IQP	−15.6040	19.7084	−0.7917
SDM	−14.6754	19.3006	−0.7604
Shrinkage-Mean	**−11.5543**	**18.0076**	−0.6416
Shrinkage-MICE	−12.9444	18.7709	−0.6896
Equal weight	−18.9210	31.2105	−0.6062

Table 14. Annualised average return, annualised standard deviation, and Sharpe ratio of a mixture of NASDAQ and SP500 stocks minimum variance portfolios based on correlation matrices estimated using different methods. (Best performances are marked in bold).

Method	Average Return (%)	Standard Deviation (%)	Sharpe Ratio
ALD	6.5146	15.1593	0.4297
GD	6.5242	15.1410	0.4309
ILP	**7.0063**	15.2558	**0.4593**
IQP	6.5072	15.1598	0.4292
SDM	6.5160	**15.1395**	0.4304
Shrinkage-Mean	3.4394	15.1660	0.2268
Shrinkage-MICE	6.1097	15.3055	0.3992
Equal weight	−2.6196	24.7886	−0.1057

Judging from annualised standard deviation, we see that all minimum variance portfolios have much lower volatility compared to the equal weight portfolio. In the NASDAQ portfolio (Table 13), the portfolios built with the shrinkage method achieve both higher annualised return and lower standard deviation. Portfolios with GD and SDM also have

comparable results. However, since the annualised return is negative, it is not easy to determine which is the best. In the mixed portfolio (Table 14), SD achieves the lowest standard deviation, while the portfolio with ILP gives the best return and Sharpe ratio. We would like to point out that the shrinkage method only works without missing values in the data. Therefore, different data imputation methods can influence its performance. Table 14 gives a good example of this where Shrinkage-Mean and Shrinkage-MICE portfolios show a significant difference in their returns. Another note to make is that in this experiment, all portfolios are built with correlation matrices estimated using historical data and are then fixed for 20 months. The results might be different if the correlation matrices are estimated based on moving windows and the portfolios are updated over time.

6.2. Experiments with Machine Learning Data for Applications in PCA

Principal Component Analysis (PCA) (Pearson 1901) is a popular statistical tool to reduce the dimension of the dataset while maintaining as much information as possible from the data, which helps visualising the data or performing further actions in different fields of studies. The covariance matrix of the features is calculated before it turns into an eigenvalue problem (Jolliffe 2002). A standardisation of the data is usually needed when the features are in different scales; or alternatively, the correlation matrix can be used instead of the covariance matrix. If missing values are present in the dataset, a standard solution is to delete all instances that contain missing values and then calculate the correlation matrix. However, ignoring such instances means that less information can be used from the whole dataset, which may lead to a decrease in the quality of the PCA. We seek to use as much information as possible from the dataset by calculating the correlation between attributes in a pairwise manner. This may cause the resulting correlation matrix to be non-PSD; therefore, a PSDisation process is needed.

We use the *wine* data available from the UCI Machine Learning Repository (Dua and Graff 2017) as an example to show the advantages of using PSDisation on pairwisely calculated correlation matrix. The data consist of 13 attributes of chemical analysis of wines from different cultivars in Italy. The target is categorical with three classes. The features are in different scales so it makes sense to use the correlation matrix in PCA. We first drop 20% of the values from the dataset randomly so that it has missing values. We calculate the pairwised correlation matrix which is non-PSD and thus invalid. Then, PSDisation is performed on the invalid correlation matrix, after which eigenvectors are calculated to transfer the standardised data with missing values. The variance of each principal component of the transformed data is reported in Table 15. We use IQP algorithm for PSDisation in this example. We also perform PCA according to the correlation matrix calculated by omitting all instances that have missing values. Figure 4 shows the cumulative sum of the variance for each principal component of the transformed data by different methods, where *all PCA* stands for PCA performed with all instances with missing values omitted, and *pairwise PCA* stands for PCA using the pairwise correlation matrix with PSDisation. Since PCA aims to capture as large a variance in the data via the top several principal components as possible, a curve that increases faster within the first several principal components indicates a better PCA result.

Table 15. Variance of each principal component in the transformed wine data. (Higher variance in each row is marked in bold).

Principal Component	All PCA	Pairwise PCA
1	3.1258	**3.2692**
2	1.5018	**1.6485**

Table 15. *Cont.*

Principal Component	All PCA	Pairwise PCA
3	0.7359	**1.0723**
4	0.6901	**0.7908**
5	**0.7454**	0.7164
6	0.5511	**0.5894**
7	**0.7294**	0.5543
8	**0.4103**	0.3897
9	**0.4756**	0.3526
10	**0.3532**	0.2875
11	**0.3712**	0.2895
12	**0.3508**	0.2604
13	**0.3436**	0.1637

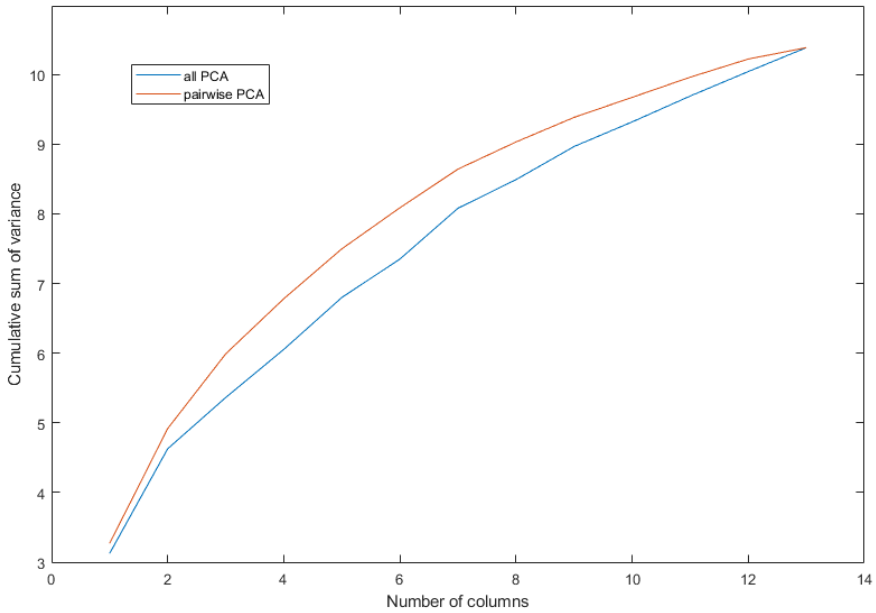

Figure 4. Cumulative sum of the variance of each principal component in the transformed wine data.

Table 15 indicates that with the use of pairwised correlation, the transformed data have higher variance in the first several principle components than all PCA. Therefore, more important information can be captured by the first few principal components via pairwise PCA.

Furthermore, Table 16 summarises the weight of each feature that the first and second principle components contain via PCA on full data and pairwise PCA and all PCA on data with missing values. The weights of the first three features with the highest weights are marked in bold. Since PCA is calculated from the full dataset, we want the results of pairwise PCA or all PCA to stay as close to the results of PCA as possible so that it captures more information; i.e., we want the first and second principal components in pairwise PCA or all PCA to both assign similar weights to each feature as those assigned by PCA. According to Table 16, the weights assigned by pairwise PCA and PCA are very close to each other. In contrast, while all PCA selects the same feature with highest weights as original PCA, the weights of the other features it gives are generally much further away than pairwise PCA. We may conclude that pairwise PCA preserves as much information as possible from data with missing values and thus produces more similar results to PCA

from the full dataset than all PCA. This indicates that pairwise PCA works better than all PCA on datasets with missing entries, and PSDisation helps to achieve this.

Table 16. Weights of each feature assigned by the first and second principal components in the wine data via PCA, pairwise PCA, and all PCA. Top: weights of features in the first principal component; bottom: weights of features in the second principal component. (Top three weights in each column are marked in bold).

Features	PCA	Pairwise PCA	All PCA
	Top		
Flavanoids	**0.4229**	**0.4212**	**0.3793**
Total phenols	**0.3947**	**0.3753**	**0.3494**
OD280/OD315 of diluted wines	**0.3762**	**0.3633**	0.3334
Proanthocyanins	0.3134	0.3046	0.1521
Nonflavanoid phenols	0.2985	0.2872	0.2453
Hue	0.2967	0.3090	0.2823
Proline	0.2868	0.3022	**0.3558**
Malic acid	0.2452	0.2237	0.2489
Alcalinity of ash	0.2393	0.2685	0.3481
Alcohol	0.1443	0.1886	0.3408
Magnesium	0.1420	0.1640	0.1399
Color intensity	0.0886	0.0550	0.0809
Ash	0.0021	0.0010	0.0893
	Bottom		
Color intensity	**0.5300**	**0.5522**	**0.5696**
Alcohol	**0.4837**	**0.4429**	0.2506
Proline	**0.3649**	**0.3603**	0.2577
Ash	0.3161	0.3533	**0.3538**
Magnesium	0.2996	0.2679	**0.3946**
Hue	0.2792	0.2824	0.3146
Malic acid	0.2249	0.2240	0.0492
OD280/OD315 of diluted wines	0.1645	0.1697	0.2766
Total phenols	0.0650	0.0788	0.0717
Proanthocyanins	0.0393	0.0280	0.2065
Nonflavanoid phenols	0.0288	0.0769	0.0902
Alcalinity of ash	0.0106	0.0287	0.1363
Flavanoids	0.0034	0.0049	0.1147

To further show the impact of missing values on the data and different methods of imputation data before PCA, we perform the following classification experiment. We split the dataset into 70% training set and 30% testing set. For the training set, we randomly drop 20% of the data so that the correlation matrix estimated in a pairwise manner is not PSD. We use different PSDisation methods to obtain the PSD correlation matrix and then perform PCA. In comparison, we use all instances without missing values to perform all PCA. We also include PCA using the correlation matrix estimated by the shrinkage method with the two data imputation strategies as explained previously in our test. As a benchmark, PCA is performed directly on data where the missing entries are replaced with mean values. For

each of these PCA methods, the first three principle components are chosen to build a k Nearest Neighbours (kNN) model. We then use the kNN model to predict the labels with PCA-transformed data in the testing set and calculate the prediction accuracy. The test is repeated 50 times, and average testing accuracies are shown in Table 17. In each model, k is tuned using leave-one-out cross-validation.

Table 17. Average testing accuracy of kNN models built on data after PCA using different correlation matrix estimating methods. (Best performance is marked in bold).

	Testing Accuracy
all PCA	0.9143
pairwise PCA with ALD	0.9581
pairwise PCA with IQP	0.9581
pairwise PCA with ILP	**0.9589**
pairwise PCA with GD	0.9581
pairwise PCA with SDM	0.9581
PCA with Shrinkage-Mean	0.9555
PCA with Shrinkage-MICE	0.9585
PCA on data filled with mean values	0.9558

According to Table 17, all PCA performs the worst in the classification task, indicating that dropping too much data in the training set may lead to undesirable behaviours of the model. Pairwise PCA, on the other hand, gives better predictions, of which ILP achieves the best prediction accuracy. This shows that in some scenarios, optimising under the Chebyshev norm instead of the F-norm can practically be a better choice. The shrinkage method performs similarly to pairwise PCA, and it also depends on data imputation methods. However, there is not enough evidence to judge between pairwise PCA and PCA using the shrinkage method.

7. Conclusions

In this paper, we proposed two new approaches, the iterative quadratic/linear programming method and the gradient descent method, to tackle the PSDisation problem for actuarial analysis in order to find the nearest correlation matrix.

IQP/ILP is a flexible algorithm that approximates the nearest correlation matrix by solving a series of optimisations with linear constraints. Despite being more time-consuming than traditional methods, the IQP method can achieve optimal solutions that are comparable to the APM, Newton, or ALD method when working with the F-norm. There is currently no efficient PSDisation algorithm in the literature that minimises the Chebyshev norm as ILP does. In our experiments, the ILP and IQP methods produce good results under the Chebyshev norm and the H-norm, respectively. In addition, it is crucial to understand that our formulation of the problem is very flexible and thus can handle norms of any choice. It is also possible to combine different norms in our formulation (e.g., to minimise $\alpha||\mathbf{A}-\mathbf{X}||_{\mathbf{F}}^2+(1-\alpha)||\mathbf{A}-\mathbf{X}||_{\mathbf{C}}^2$, where α is a weighting parameter). Future works can be undertaken to increase the speed of solving the quadratic or linear optimisation problem in each iteration to increase the efficiency in order to cope with larger problems.

The GD method repeats the following: taking a step in the opposite direction of the gradient, projecting the matrix onto the PSD cone via spectral decomposition, and scaling the matrix such that the diagonal is 1. This method is effective in unconstrained PSDisation problems with respect to the F-norm. Experiments show the efficiency and robustness of the algorithm; it is more resilient to noise and shows good potential in practical scenarios.

Future attempts to modify the GD algorithm can be considered in order that it can be applied under different choices of norms or to solve problems with more constraints. For example, one possible extension of this could be to explore a subgradient approach or to use smooth approximation to work with norms that are not differentiable.

Author Contributions: Conceptualisation, V.A.; methodology, V.A.; software, R.W. and F.Z.; validation, V.A. and R.Z.; formal analysis, V.A.; investigation, R.W.; resources, V.A. and F.Z.; data curation, R.W.; writing—original draft preparation, R.W.; writing—review and editing, V.A., R.W., and R.Z.; visualisation, R.W.; supervision, V.A. and R.Z.; project administration, V.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

Ahmadi, Amir Ali, and Georgina Hall. 2017. Sum of squares basis pursuit with linear and second order cone programming. *Algebraic and Geometric Methods in Discrete Mathematics* 685: 27–53.

Borsdorf, Rüdiger, and Nicholas J. Higham. 2010. A preconditioned newton algorithm for the nearest correlation matrix. *IMA Journal of Numerical Analysis* 30: 94–107. [CrossRef]

Boyle, James P., and Richard L. Dykstra. 1986. A method for finding projections onto the intersection of convex sets in hilbert spaces. In *Advances in Order Restricted Statistical Inference*. New York: Springer, pp. 28–47.

Cutajar, Stefan, Helena Smigoc, and Adrian O'Hagan. 2017. Actuarial risk matrices: the nearest positive semidefinite matrix problem. *North American Actuarial Journal* 21: 552–64. [CrossRef]

Deutsch, Frank, and Hein Hundal. 1997. The rate of convergence for the method of alternating projections, ii. *Journal of Mathematical Analysis and Applications* 205: 381–405. [CrossRef]

Dua, Dheeru, and Casey Graff. 2017. UCI Machine Learning Repository. Available online: http://archive.ics.uci.edu/ml (accessed on 1 November 2024).

Dykstra, Richard L. 1983. An algorithm for restricted least squares regression. *Journal of the American Statistical Association* 78: 837–42. [CrossRef]

Higham, Nicholas J. 2002. Computing the nearest correlation matrix—A problem from finance. *IMA Journal of Numerical Analysis* 22: 329–43. [CrossRef]

Higham, Nicholas J., Natasa Strabic, and Vedran Sego. 2016. Restoring definiteness via shrinking, with an application to correlation matrices with a fixed block. *SIAM Review* 58: 245–63. [CrossRef]

Jolliffe, Ian T. 2002. *Principal Component Analysis for Special Types of Data*. New York: Springer.

Ledoit, Olivier, and Michael Wolf. 2004. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis* 88: 365–411. [CrossRef]

Milhaud, Xavier, Victorien Poncelet, and Clement Saillard. 2018. Operational choices for risk aggregation in insurance: Psdization and scr sensitivity. *Risks* 6: 36. [CrossRef]

Opgen-Rhein, Rainer, and Korbinian Strimmer. 2007. Accurate ranking of differentially expressed genes by a distribution-free shrinkage approach. *Statistical Applications in Genetics and Molecular Biology* 6: 9. [CrossRef] [PubMed]

Pearson, Karl. 1895. Notes on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London* 58: 240–42.

Pearson, Karl. 1901. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2: 559–72. [CrossRef]

Qi, Houduo, and Defeng Sun. 2006. A quadratically convergent newton method for computing the nearest correlation matrix. *SIAM Journal on Matrix Analysis and Applications* 28: 360–85. [CrossRef]

Qi, Houduo, and Defeng Sun. 2011. An augmented lagrangian dual approach for the h-weighted nearest correlation matrix problem. *IMA Journal of Numerical Analysis* 31: 491–511. [CrossRef]

Rebonato, Riccardo, and Peter Jäckel. 2011. The Most General Methodology to Create a Valid Correlation Matrix for Risk Management and Option Pricing Purposes. Available online: https://ssrn.com/abstract=1969689 (accessed on 29 November 2024).

Steuer, Ralph E., Yue Qi, and Maximilian Wimmer. 2024. Computing cardinality constrained portfolio selection efficient frontiers via closest correlation matrices. *European Journal of Operational Research* 313: 628–36. [CrossRef]
Van Loan, Charles F., and Gene H. Golub. 1983. *Matrix Computations*. Baltimore: Johns Hopkins University Press.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.