NUMERICAL LINEAR ALGEBRA AND OPTIMIZATION

Edited by
Ya-xiang Yuan

Science Press
Beijing/New York

Numerical Linear Algebra and Optimization

Proceedings of
the 2003' International Conference
on Numerical Optimization
and Numerical Linear Algebra

Edited by

Ya-xiang Yuan

State Key Laboratory of Scientific and Engineering Computing,
Institute of Computational Mathematics
and Scientific/Engineering Computing,
Academy of Mathematics and System Sciences,
Chinese Academy of Sciences

SCIENCE PRESS
Beijing/New York

Responsible Editors: Lin Peng   Chen Yuzhuo

Published by Science Press
16 Donghuangchenggen North Street
Beijing 100717, China
Printed in Beijing

ISBN 7-03-011216-0

Preface

The "2003' International Conference on Numerical Optimization and Numerical Linear Algebra" was held at Guilin from Oct 7-10, 2003. The conference was organized by Laboratory of Scientific and Engineering Computing, Institute of Computational Mathematics and Scientific/Engineering Computing, Academy of Mathematics and Systems Science of the Chinese Academy of Sciences, and Guangxi University.

This conference was the 4th of the biennial conferences, after the first one in Qingdao(1997), the second one in Nanjing(1999) and the third in Dunhuang(2003). About 80 participants attended this conference. It is my great pleasure and honour that Professor M.J.D. Powell (who supervised my Ph.D. about twenty years ago), FRS, of University of Cambridge, came and gave the conference's first invited lecture. I am also very glad that we had invited speakers such as Binsheng He, Masao Fukushima, Xingsi Li, Qin Ni, Linqun Qi, M. Raydan, E.W. Sachs, Wenyu Sun, T. Terlaky, Chuanlong Wang, Zengxi Wei, Naihua Xiu, Shufang Xu, Jason Zhang, Liansheng Zhang, Shao-liang Zhang, and Zhenyue Zhang, who reported the recent advances in numerical optimization and numerical linear algebra at the conference. Apart from the academic talks at the conference, there were also informal discussions on the developments of optimization and numerical linear algebra, particularly about promotions of international exchanges and collaborations in these two areas. The conference organized a one-day sightseeing tour along the beautiful Li River, which was enjoyed by all the participants.

I would like to thank the National Natural Science Foundation of China, State Key Laboratory of Scientific and Engineering Computing, Institute of Computational Mathematics and Scientific/Engineering Computing of Chinese Academy of Sciences, and Guangxi University for the financial supports which made the conference possible. I want to thank my colleague Professor Dai Yuhong, my post-doctors Dr Wang Zhouhong and Dr Shi Zhenjun, and my students Li Gaidi, Wang Liping, Wen Zaiwen, Yan Tao, Xia Yong, Zheng ZhenZhen and Professor Dai's student Qiqing Hu, for their helps in the organization of the conference. I would also like to thank Professor Zengxi Wei of Guangxi University for his assistance on the local arrangements. Finally, I am very grateful to Wen Zaiwen for his helps in the editing of the proceedings.

Ya-xiang Yuan

Contents

Preface

Part I Invited Lectures

1
The Use of Low-Rank Updates in Interior-Point Methods 3
E.D. Andersen C. Roos T. Terlaky T. Trafalis J.P. Warners

2
Study on Approximate Proximal Point Algorithms for Monotone Variational Inequalities 15
Bingsheng He Maijian Qian Yamei Wang

3
Some Optimality Conditions for Minimax Problems and Nonlinear Programs 42
Elijah Polak Liqun Qi

4
On Updating the Inverse of a KKT Matrix 56
M.J.D. Powell

5
Low-Cost Methods for Nonlinear Large-Scale Matrix Equations 79
Marcos Raydan

6
Extended Conjugate Residual Methods For Solving Nonsymmetric Linear Systems 88
Tomohiro Sogabe Shaoliang Zhang

7
Necessary Conditions and Sufficient Conditions for Non-Decomposable Two-Stage Min-Max Optimizations 100
Chuanlong Wang Zongben Xu

8
Some Recent Progress In Unconstrained Nonlinear Optimization 110
Zengxin Wei Guohang Yu

9
Superlinear Convergence $\mathcal{UV}$-Algorithms for D.C. Minimization 142
Zunquan Xia Li-Ping Pang Wei Wang

10
Principal Manifolds and Nonlinear Dimensionality Reduction via Tangent Space Alignment 156
Zhenyue Zhang Hongyuan Zha

Part II Contributed Papers

11
Preconditioned Multiple Search Direction Conjugate Gradient Method 187
Tongxiang Gu Xingping Liu Xuebin Chi

12
Existence and Uniqueness of the AHO Search Direction in Primal-Dual Interior-Point Algorithms for the Quadratic Semidefinite Programming 195
Jingjing Huang Fengmin Xu Yongxiang Ling

13
A Multisplitting and Schwarz Iteration Scheme For Solving Implicit Complementarity Problems 203
Chenliang Li Jingping Zeng Donghui Li

14
New Insights into Penalty Functions 210
Xingsi Li Shaohua Pan

15
Global Convergence Properties of the Conjugate Descent Method with Armijo-type Line Searches 224
Shujun Lian Changyu Wang

16
A New Class of Memory Gradient Methods with Inexact Line Searches 232
Zhenjun Shi

17
Convergence Properties of Nonmonotone Spectral Projected Gradient Methods 248
C.Y. Wang Q. Liu X.M. Yang

18
A Subspace Trust Region Method for Large Scale Unconstrained Optimization 265
Zhouhong Wang Zaiwen Wen Yaxiang Yuan

19
A Trust Region Method for Large Scale Inverse Problems in Atmospheric Image Restoration 275
Zaiwen Wen Yanfei Wang

20
Number Theoretic Global Optimization Searching Algorithm Based On Evolution 288
Shounan Zhong Weihua Wang Xuexiao You

Part I Invited Lectures

1 The Use of Low-Rank Updates in Interior-Point Methods

E.D. Andersen

MOSEK ApS, C/O Symbion Science Park
Fruebjergvej 3, Boks 16, DK-2100 Copenhagen O, Denmark

C. Roos

Faculty of Information Technology and Systems
Delft University of Technology,
P.O. Box 5031, 2600 GA Delft, The Netherlands

T. Terlaky

Department Computing and Software
McMaster University,
1280 Main Street West, Hamilton, ON, Canada, L8S 4K1

T. Trafalis [*]

Department Industrial Engineering
Oklahoma State University, Oklahoma, USA

J.P. Warners

Faculty of Information Technology and Systems
Delft University of Technology,
P.O. Box 5031, 2600 GA Delft, The Netherlands

Abstract: The most computational intensive part of interior point methods for solving various kinds of optimization problems is the solution of large linear systems. This note discusses several applications of linear systems with low-rank updates which arise in interior point methods and can efficiently be solved. It is also briefly discussed how this method is related to the widely used Sherman-Morrison formula. The relation with the normal equation and augmented system approach, used in interior point methods for linear optimization to calculate the search directions, is discussed. Further, applications in potential reduction methods, neural network training and conic quadratic optimization are discussed as well.

[*] The major part of the research was done while the fourth author visited the the Department SSOR of the TU Delft. He gratefully acknowledges the financial support of the department.

Keywords: Linear programming, interior point methods, linear equation systems, positive definite matrix, rank-one update, Sherman-Morrison formula.

1 Introduction

When applying an interior point method to solve optimization problems, large linear equation systems like

$$Qx = q,$$

where Q is an $n \times n$ matrix and $q \in \in \mathbb{R}^n$ turn up regularly. The coefficient matrix Q in such linear systems is frequently sparse (less than 1% of the coefficients is nonzero). By exploiting the sparsity structure of Q one can solve such linear systems with hundreds of thousands or even millions of variables. As the density of the coefficient matrix increases, i.e. the percentage of nonzero coefficients increases, the efficiency of the solution procedures deteriorates. One possible way of getting a dense coefficient matrix is that the original matrix Q is modified by a rank one (or more generally by a low-rank update) [9; 8; 19]. Even a rank-one update by the all-one vector e results in the completely dense matrix $Q + ee^T$; without utilizing this special structure, large linear systems with such coefficient matrices would not be solvable. Dense matrices which are a sum of a sparse matrix plus some low-rank terms arise frequently in practice. In fact as has been observed by Fiacco and McCormick [4, p. 185], then the Hessian of many functions can easily be written as a sum of a diagonal matrix and some low-rank terms and they call such functions factorable. This implies if an optimization problem containing only factorable functions is solved using a Newton based method, then the coefficient matrix will contain low-rank terms. Moreover, the coefficient matrix can frequently be written as a sum of a sparse matrix plus a dense low-rank update. This for example happens when solving large linear optimization problems having dense columns using interior point methods [2; 3; 6; 13; 16]. Contrary to our approach, dense linear systems obtained by low rank updates are solved frequently by using preconditioned conjugate gradient methods [8].

This paper is organized as follows. In Section 2 the approach to solve linear systems with low-rank updates is reviewed; see also [7]. Section 2.3 discusses how the method relates to a clever use of the Sherman-Morrison [5] formula. In Section 3.1, the use of low rank updates in interior point algorithms to solve linear optimization problems is discussed. In particular, efficient handling of dense columns in the coefficient matrix of the linear optimization problem is considered. Finally, in Section 3 further applications are discussed.

2 Solving Linear Systems with Low-rank Updates

In this section we first review a method to solve linear systems with low rank updates. It can also be found in [7]. Subsequently, for completeness we verify its correctness. Finally, the relation with the well-known Sherman Morrison formula is made explicit.

2.1 The solution procedure

Large, typically sparse linear equation systems have to be solved in many applications of mathematics. Frequently the coefficient matrix of a linear system is a low-rank update of a non-singular (in many cases positive definite) matrix. Here we discuss the solution of such linear systems. To be concrete, let us consider the following system of linear equations with the unknown vector $x \in \in \mathbb{R}^n$,

$$(Q + RS^{\mathrm{T}})x = q, \tag{2.1}$$

where Q is an $n \times n$ matrix, R and S are $n \times k$ matrices and $q \in \in \mathbb{R}^n$ are given. We make the following assumptions.

Assumptions:

1. The matrix Q is nonsingular.
2. The rank of both R and S are equal to $k \leq n$.
3. The matrix $Q + RS^{\mathrm{T}}$ is nonsingular.
4. The matrix Q is sparse, while RS^{T} is dense.

We will see later that Assumptions 1–4 guarantee that the system (2.1) is solvable efficiently.

As it is intuitively clear, solving a linear system

$$Qx = q, \tag{2.2}$$

where the sparse matrix Q is the coefficient matrix is computationally cheap, while directly solving equation system (2.1) with the dense coefficient matrix is computationally expensive. Our primary goal is to utilize the sparsity of the matrix Q. As the first step we reformulate (2.1) as follows

$$Qx = q - RS^{\mathrm{T}}x = q - Ry.$$

For a moment neglecting the fact that the vector $y = S^{\mathrm{T}}x$ depends on the unknown vector x we can decompose the solution process into the following steps.

Algorithm 2.1.

Step 1. *Determine x_0, the solution of the sparse system $Qx_0 = q$.*

Step 2. *Determine the solution of the sparse system $QU = R$, where U is an $n \times k$ matrix matrix of unknowns.*

Step 3. *Now we consider how to find an appropriate vector y. By the definition of y we have*

$$y = S^{\mathrm{T}}x = S^{\mathrm{T}}(x_0 - Uy),$$

or equivalently, y is the solution of the system

$$(I + S^{\mathrm{T}}U)y = S^{\mathrm{T}}x_0. \tag{2.3}$$

Step 4. *Let U be the $n \times k$ matrix with column vectors u_j. Then we have the solution*

$$x = x_0 - Uy.$$

The above procedure involves the solution of $k+1$ sparse linear systems (in Steps 1 and 2), all with the same sparse coefficient matrix Q, hence these can be solved easily.

In the next section we will verify that the k-dimensional linear equation system (2.3) has a unique solution and that in the way we just indicated, the linear equation system (2.1) can indeed be solved efficiently.

Before doing that, let us make a simple estimation of the computational complexity under Assumption 4. By a direct approach the linear system (2.1) with a dense coefficient matrix can be solved in $\mathcal{O}(n^3)$ arithmetic operations. Assume that the matrix Q is sparse and so the equation system $Qx_0 = q$ can be solved in $\rho < \mathcal{O}(n^3)$ arithmetic operations. We have to solve $k+1$ such systems and a small dense system with k unknowns. Thus the total complexity becomes $(k+1)\rho + \mathcal{O}(k^3)$. Note that, in many applications matrix Q is either (multi)diagonal, or block-diagonal with small diagonal blocks. As a consequence one has $\rho = \mathcal{O}(n)$. Indeed, in many of these cases it holds that $k \leq \mathcal{O}(\sqrt{n})$, so the total computational complexity becomes $\mathcal{O}(n\sqrt{n})$, which is a factor $n\sqrt{n}$ better than the direct approach.

2.2 Verification

As it is proposed above the linear system (2.1) is solved by solving $k+1$ linear equation systems with the same coefficient matrix Q and a small $k \times k$ linear system as follows.

Let $x_0 \in \in \mathbb{R}^n$ be such that $Qx_0 = q$ and $u_j \in \in \mathbb{R}^n$ such that $Qu_j = r_j$ for $j = 1, \cdots, k$, where r_j denotes the jth column of matrix R and finally, let $y \in \in \mathbb{R}^k$ be such that $(I + S^{\mathrm{T}}U)y = S^{\mathrm{T}}x_0$, where I denotes the k-dimensional identity matrix and $U = [u_1, \cdots, u_k]$. We prove the following theorem.

Theorem 2.2. *If Assumptions 1–3 hold then the unique solution of the linear system (2.1) is given by*

$$x = x_0 - Uy.$$

Proof. First note that by Assumption 3 the solution x exists and it is unique. By Assumption 1, x_0 and the r_j are unique as well. Further, by Assumption 2 the vectors r_j, $j = 1, \cdots, k$ are linearly independent and then, by Assumption 1, the solution vectors u_j, $j = 1, \cdots, k$ are linearly independent as well, i.e. $\mathrm{rank}(U) = k$. Now by proving that the coefficient matrix $I + S^{\mathrm{T}}U$ is nonsingular we verify that y is unique as well. Assume to the contrary that there is a nonzero vector $w \in \in \mathbb{R}^k$ such that $(I + S^{\mathrm{T}}U)w = 0$, or equivalently $S^{\mathrm{T}}Uw = -w$. Then, by multiplying the nonsingular matrix $Q + RS^{\mathrm{T}}$ by the nonzero vector Uw and using $Qu_j = r_j$ we have

$$(Q + RS^{\mathrm{T}})Uw = (QU)w + R(S^{\mathrm{T}}Uw) = Rw - Rw$$

yielding a contradiction.

Finally by simple calculations the reader may verify that $(Q + RS^{\mathrm{T}})(x_0 - Uy) = q$ which completes the proof. □

2.3 The Sherman-Morrison Formula

There is an intimate relation between the solution procedure we reviewed and the well known Sherman-Morrison formula [5], given below. For a survey of the Sherman-Morrison formula we refer the reader to Hager [7].

Sherman-Morrison formula:

$$(Q + RS^{\mathrm{T}})^{-1} = Q^{-1} - Q^{-1}R(I + S^{\mathrm{T}}Q^{-1}R)^{-1}S^{\mathrm{T}}Q^{-1} \tag{2.4}$$

We can make two important observations.

- Having proved the correctness of our procedure the Sherman-Morrison formula (2.4) can be derived from our procedure. This can be done as follows. By definition – we have $x_0 = Q^{-1}q$, $U = Q^{-1}R$ and $y = (I + S^{\mathrm{T}}U)^{-1}S^{\mathrm{T}}x_0$ – the solution x is given by

  $$\begin{aligned} x = x_0 - Uy &= Q^{-1}q - Q^{-1}R(I + S^{\mathrm{T}}U)^{-1}S^{\mathrm{T}}x_0 \\ &= (Q^{-1} - Q^{-1}R(I + S^{\mathrm{T}}Q^{-1}R)^{-1}S^{\mathrm{T}}Q^{-1})q. \end{aligned}$$

  By Assumption 3 we can write $x = (Q + RS^{\mathrm{T}})^{-1}q$. Comparing this with the above expression and observing that Theorem 2.2 holds for all right-hand-side vector $q \in \in \mathbb{R}^n$ the Sherman-Morrison formula (2.4) follows.

- On the other hand, one can derive our procedure by carefully analyzing the Sherman-Morrison formula (2.4). Using again that the solution of (2.1) can be written as $x = (Q + RS^{\mathrm{T}})^{-1}q$ we have

  $$x = (Q^{-1} - Q^{-1}R(I + S^{\mathrm{T}}Q^{-1}R)^{-1}S^{\mathrm{T}}Q^{-1})q.$$

  Here all the "inverse matrix – vector" products have to be replaced by the solution of a linear equation system. Thus we have the expression $Q^{-1}q$ which is equivalent to x_0; the expression $Q^{-1}R$ which is equivalent to U; and finally, having these done, the expression

  $$(I + S^{\mathrm{T}}Q^{-1}R)^{-1}S^{\mathrm{T}}Q^{-1}q = (I + S^{\mathrm{T}}U)^{-1}S^{\mathrm{T}}x_0$$

  which is equivalent to z.

3 Applications in Interior Point Methods

3.1 Normal Equations Versus Augmented System in Linear Optimization

When solving linear programming problems with interior point methods one has to solve at each iteration [2; 3; 6], either the so-called *normal equation*

$$(AD^{-2}A^{\mathrm{T}})x = q \tag{3.1}$$

or the so-called *augmented system*

$$\begin{pmatrix} 0 & A \\ A^{\mathrm{T}} & D^2 \end{pmatrix} \begin{pmatrix} x \\ x^0 \end{pmatrix} = \begin{pmatrix} -q \\ 0 \end{pmatrix}, \tag{3.2}$$

where matrix A is the $m \times n$ constraints coefficient matrix of the linear programming problem with $\text{rank}(A) = m$, the matrix D is an appropriate n dimensional positive definite diagonal matrix and $q \in \in \mathbb{R}^n$. The exact values and definitions of D and q are specific for the different algorithmic variants. The reader easily verifies that the solutions x of both systems (3.1) and (3.2) are identical.

Early implementations of interior point methods were based on the normal equation approach [12]. People recognized that dense columns of the matrix A result in unacceptable fill-in in the coefficient matrix $AD^{-2}A^{\mathrm{T}}$ of the normal equation. Let us have a closer look at this problem. For simplicity let us assume that the problem is scaled in such a way that $D = I$, and that all the dense columns are the last columns, thus matrix A can be partitioned as $A = [A_s, A_d]$ where A_s contains the sparse and A_d contains the dense columns. Now (3.1) gives

$$(A_sA_s^{\mathrm{T}} + A_dA_d^{\mathrm{T}})x = q. \tag{3.3}$$

In the case $A_sA_s^{\mathrm{T}}$ is nonsingular, this system is in the form (2.1) with $Q = A_sA_s^{\mathrm{T}}$, hence can be solved efficiently. Thus our procedure described above can be used efficiently to handle dense columns in linear programming when solved by interior point methods. When $A_sA_s^{\mathrm{T}}$ is singular a further trick will be needed.

Having a closer look at the augmented system (3.2) an interesting observation can be made. By substituting $A = [A_s, A_d]$ we have

$$\begin{pmatrix} 0 & A_s & A_d \\ A_s^{\mathrm{T}} & I & 0 \\ A_d^{\mathrm{T}} & 0 & I \end{pmatrix} \begin{pmatrix} x \\ x^1 \\ x^2 \end{pmatrix} = \begin{pmatrix} -q \\ 0 \\ 0 \end{pmatrix}. \tag{3.4}$$

By performing the pivots at the central unit matrix block we have

$$\begin{pmatrix} -A_sA_s^{\mathrm{T}} & 0 & A_d \\ A_s^{\mathrm{T}} & I & 0 \\ A_d^{\mathrm{T}} & 0 & I \end{pmatrix} \begin{pmatrix} x \\ x^1 \\ x^2 \end{pmatrix} = \begin{pmatrix} -q \\ 0 \\ 0 \end{pmatrix}, \tag{3.5}$$

i.e. one has to solve the system

$$\begin{pmatrix} -A_sA_s^{\mathrm{T}} & A_d \\ A_d^{\mathrm{T}} & I \end{pmatrix} \begin{pmatrix} x \\ x^2 \end{pmatrix} = \begin{pmatrix} -q \\ 0 \end{pmatrix}. \tag{3.6}$$

The sparse part is nonsingular. If in (3.6) we would pivot on the unit matrix block we would get the dense normal equation (3.3). Instead, assuming (again, as above) that the matrix $Q = A_sA_s^{\mathrm{T}}$ is nonsingular, we pivot on the left upper block which results in

$$\begin{pmatrix} I & -Q^{-1}A_d \\ 0 & I + A_d^{\mathrm{T}}Q^{-1}A_d \end{pmatrix} \begin{pmatrix} x \\ x^2 \end{pmatrix} = \begin{pmatrix} Q^{-1}q \\ -A_d^{\mathrm{T}}Q^{-1}q \end{pmatrix}. \tag{3.7}$$

Now the reader readily recognizes that the right hand side of (3.7) can be calculated by solving $Qx = q$ and one obtains $x_0 = Q^{-1}q$. Similarly, the matrix $U = Q^{-1}A_d$ can easily be calculated as well, and then clearly the equation $(I + A_d^{\mathrm{T}}Q^{-1}A_d)x^2 = -A_d^{\mathrm{T}}x_0$ is equivalent to solving $(I + A_d^{\mathrm{T}}U)x^2 =$

$-A_d^T x_0$. Hence our procedure can be interpreted as a specific implementation of the augmented system approach.[†]

The sparse part is singular. In the case $A_s A_s^T$ is singular, we cannot apply the above procedure. To handle this case efficiently a clever trick of K.D. Andersen [3] is needed. The solution process up to formula (3.6) can be performed independently of the (non)singularity of $A_s A_s^T$, but then, due to the singularity of $A_s A_s^T$ we cannot pivot on this block. To ensure the nonsingularity of this sparse block a nonsingular, possibly diagonal subsidiary matrix H is used. The reader easily verifies that the system

$$\begin{pmatrix} -A_s A_s^T - HH^T & A_d & H \\ A_d^T & I & 0 \\ H^T & 0 & I \end{pmatrix} \begin{pmatrix} x \\ x^2 \\ x^0 \end{pmatrix} = \begin{pmatrix} -q \\ 0 \\ 0 \end{pmatrix}. \qquad (3.8)$$

is equivalent to the linear system (3.6), but here the sparse left-upper block is nonsingular, thus our procedure is applicable again.[‡]

3.2 Karmarkar's Potential Function in Combinatorial Optimization

Let us consider the binary feasibility problem (BFP).

$$\text{(BFP)} \qquad \text{find } x \in \{-1, 1\}^n \text{ such that } Ax \leq b.$$

Here $A \in \in \mathbb{R}^{t \times n}$, $b \in \in \mathbb{R}^t$. Problems in this format are also considered in [8; 9]. Karmarkar [8; 9] presented a potential reduction algorithm to solve problems of the form (BFP). After relaxing the integrality constraints to linear constraints $-e \leq x \leq e$ (here e denotes the all-one vector of length n), he proposed to optimize a nonconvex potential function in order to get integer solutions. Let us assume that an interior solution x (i.e. $Ax < b$, $-e < x < e$) is given. Then Karmarkar's potential function is given by

$$\phi(x) = \log \sqrt{n - x^T x} - \frac{1}{m} \sum_{j=1}^{m} s_j. \qquad (3.9)$$

where the variables s_j are the slack variables associated with all the inequality constraints involved in the relaxed problem (note that $m := t + 2n$ and the slack variables are $s_j = b_j - a_j^T x$, $1 \leq j \leq t$; $s_j = 1 - x_{j-t}$, $t + 1 \leq j \leq t + n$; and $s_j = 1 + x_{j-(t+n)}$, $t + n + 1 \leq j \leq t + 2n$). To minimize Karmarkar's potential function (3.9) a trust region algorithm is implemented. In this algorithm

[†] Observe that in actual implementations of the augmented system approach the last mentioned pivots on the block $-A_s A_s^T$ are not performed explicitly; instead, a Bunch-Parlett factorization is made. The augmented system approach with Bunch-Parlett factorization automatically does the job of efficiently handling dense columns in interior point methods.

[‡] Note that if the linear optimization problem is preprocessed so that the problem is transformed to the canonical (inequalities and nonnegative variables) form, then this trick is not needed. In this case the left upper sparse block of the linear system has the form $-A_s A_s^T + \overline{D}$, where $\overline{D}$ is a positive diagonal matrix. Because this matrix is nonsingular, our procedure is applicable without further adjustment.

linear systems involving the Hessian of (3.9) as the coefficient matrix, are solved repeatedly. This Hessian matrix is given by

$$H_\phi(\mu) = -\frac{1}{f_0}I - \frac{2}{f_0^2}xx^{\mathrm{T}} + (\frac{1}{m} + \mu)\overline{A}^{\mathrm{T}}S^{-2}\overline{A},$$

where $f_0 = n - x^{\mathrm{T}}x$, $\overline{A} = (A^{\mathrm{T}}, I, -I)^{\mathrm{T}}$ and S is the diagonal matrix of the slack variables (for more details see [8; 9; 19]). Due to the term xx^{T} the matrix H_ϕ is completely dense. Assuming that matrix A is sparse, the matrix $H_\phi(\mu)$ is a rank-one update of the sparse matrix $-\frac{1}{f_0}I + (\frac{1}{m} + \mu)\overline{A}^{\mathrm{T}}S^{-2}\overline{A}$, hence the theory discussed in this paper is applicable to minimize Karmarkar's potential function efficiently. However, an additional requirement has to be satisfied; namely the parameter μ must be such that $H_\phi(\mu)$ is positive semidefinite, while at the same time the norm of the solution $x = (H_\phi(\mu))^{-1}h$ (here h denotes the right hand side of the linear system) must be sufficiently large. Note that this norm decreases when μ increases. Thus, it is not possible just to choose a large value for μ (to make sure that $H_\phi(\mu)$ is positive semidefinite). Now we show that we can use the rank-one update scheme and still check whether $H_\phi(\mu)$ is positive definite.

We have the following theorem.

Theorem 3.1. *Assume that $R \in \in \mathbb{R}^{n\times n}$ is symmetric. Then $R - zz^{\mathrm{T}}$ is positive definite if and only if R is positive definite and $z^{\mathrm{T}}R^{-1}z < 1$.*

Proof. Using the well known Schur complement reformulation we know that $R - zz^{\mathrm{T}}$ is positive definite if and only if the matrix

$$\begin{bmatrix} R & z \\ z^{\mathrm{T}} & 1 \end{bmatrix}$$

is positive definite. Thus, R must be positive definite and it easily follows that $z^{\mathrm{T}}Rz < 1$, or equivalently, $z^{\mathrm{T}}R^{-1}z < 1$ if and only if $R - zz^{\mathrm{T}}$ is positive definite. □

Hence we obtain the following scheme to solve the system

$$(R - zz^{\mathrm{T}})x = q, \tag{3.10}$$

with the additional condition that $R - zz^{\mathrm{T}}$ be positive semidefinite.

(1) Determine the Cholesky factor U of R. If during this process R is discovered to be non positive definite, stop.

(2) With the help of U, solve $Rx_0 = z$. If $z^{\mathrm{T}}R^{-1}z = z^{\mathrm{T}}x_0 \geq 1$, $R - zz^{\mathrm{T}}$ is not positive semidefinite, so stop.

(3) Solve $Rx_1 = q$, again using U.

(4) Let $\alpha := \frac{x_1^{\mathrm{T}}z}{1-x_0^{\mathrm{T}}z}$. Then $x = x_1 + \alpha x_0$ solves system (1.2).

This technique is used in the computational results reported in [19].

3.3 An application in unsupervised neural network training

In unsupervised neural network training, the objective is to categorize or discover features or regularities in a given set of training data. An approach proposed by Linsker [10] is to maximize the output variance. This gives rise to a nonconvex quadratic optimization problem of the following form.

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}(C + ee^{\mathrm{T}})x + c^{\mathrm{T}}x \\ \text{subject to} & 0 \leq x \leq e. \end{array}$$

Here the matrix C is related to the *auto correlation* matrix which is (in this case) negative definite. In [17] a potential reduction approach to solve the above problem is developed. This involves solving linear systems with the matrix $C + ee^{\mathrm{T}}$ as a coefficient matrix. Obviously, if C is sparse, the technique described in this paper can be used to speedup the computations.

3.4 Markowitz portfolio model

The well-known Markowitz portfolio problem can be modelled by the following convex quadratic optimization problem

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|Fx\|^2 + c^{\mathrm{T}}x \\ \text{subject to} & Ax = b, \\ & x \geq 0. \end{array} \tag{3.11}$$

where A is either just the all-one vector ($A = e^{\mathrm{T}}$) or a matrix with significantly less rows than columns. F is a $k \times n$ dense matrix, where in practice k quite frequently is much smaller than n. Now if (3.11) is solved using one of the popular interior point methods [15], then in each iteration a linear equation system having the coefficient matrix

$$A(D + F^{\mathrm{T}}F)^{-1}A^{\mathrm{T}}$$

should be solved. In this case D is a positive definite diagonal matrix. One possible way of solving (3.4) is of course to form the inverse of $(D + F^{\mathrm{T}}F)$ explicitly. However, the technique presented in Section 2 is directly applicable which in this case leads to computing the inverse (or a Cholesky factorization) of a $k \times k$ matrix instead of a $n \times n$ matrix. Given k is much smaller than n a large computational saving is realized. It should be noted that Fiacco and McCormick [4, p. 186] and later Vanderbei and Carpenter [18, p. 22] have noted that this particular structure in the portfolio optimization problem can be exploited to speed up the computations.

3.5 Conic quadratic optimization

Recently, so-called conic quadratic optimization or second order cone optimization has received much attention due to the wide applicability of the model, see [11]. A conic quadratic optimization problem can be stated as follows