The separable and non–separable formulations of convex quadratic problems in interior point methods *

Csaba Mészáros

WP 98-3

Laboratory of Operations Research and Decision Systems
Hungarian Academy of Sciences

Abstract

Quadratic programming (QP) problems are usually formulated in the general non-separable form. For convex QP problems, however, it is always possible to separate the quadratic function by reformulating the original problem. The two formulations present equvivalent optimization problems but may influence the behavior and efficiency of the optimization techniques applied. In the paper the interior point methods for large–scale convex quadratic programming are concerned. The target of our investigation is to compare how the two formulations of convex QPs influence the computational work of interior point methods and answer the question which formulation is to be used in general.

1 Introduction

Convex quadratic problems were introduced as early as the 1950s [4] and present an important class of the linearly constrained optimization. Beyond

*This work was supported in part by EPSRC grant No. GR/J52655 and Hungarian Research Fund OTKA T-016413.

the classical portfolio optimization, quadratic problems arise in wide variety of applications, including least squares problems, regression analysis, data fitting, risk management. In the paper we concern the quadratic programming problem in the form:

$$\begin{array}{rl} \min & c^T x + \frac{1}{2} x^T Q x, \\ & Ax = b, \\ & x \geq 0, \end{array} \tag{1}$$

where $A \in R^{m \times n}$, $Q \in R^{n \times n}$ is symmetric positive semidefinite and $c, x \in R^n$, $b \in R^m$. The Wolfe–dual associated with this problem can be written as

$$\begin{array}{rl} \max & b^T y - \frac{1}{2} x^T Q x, \\ & A^T y + z - Q x = c, \\ & z \geq 0, \end{array} \tag{2}$$

where $z \in R^n$ and $y \in R^m$. The constrains of the above problems define convex polyhedrons which are nonempty if a feasible primal–dual solution exists.

The quadratic problem is called *separable* if Q has no off–diagonal nonzeros. It is easy to see that any convex QP problem (1) can be converted to a separable equivalent problem of the form:

$$\begin{array}{rrcl} \min & c^T x + \frac{1}{2} p^T p, & & \\ & Ax & = & b, \\ & F^T x - p & = & 0, \\ & x & \geq & 0, \end{array} \tag{3}$$

where $p \in R^n$ and $F \in R^{n \times n}$ is such that $FF^T = Q$.

Interior point methods (IPM) are iterative algorithms for solving optimization problems. Numerical experiments show that on large–scale linearly constrained problems they can significantly outperform the methods based on the generalization of the simplex algorithm. The most impressive progress in the implementation of interior point methods were done in linear programming (LP). Recent implementation techniques in this field are well-established and highly efficient [1]. Since for separable quadratic problems the same sparsity and numeric engines can be used in IPMs as for linear programs, there are minor modifications to be performed on an IPM implementation for LP to extend its capabilities for handling separable quadratic problems. This staightforward implementation made the separable equivalent form of QPs attractive for interior point methods [3]. It is to be noted, however, that the dimension of problem (3) is usually significantly larger than that of problem (1). In the paper we will examine how the increase of the model size influences the efficiency of the computations in IPMs.

The paper is organized as follows: Section 2 presents the most involved computational task of interior point methods for quadratic programming which will be the subject of our further investigations. In section 3 we shall compare issues related to the differences in IPMs arising from the separable and non–separable forms of QPs. We will consider the infeasible primal–dual log barrier method in our computational study, presented in section 4, because this algorithm and its variants are considered the most efficient in practice among the IPMs. However, it is to be noted that the discussed computational issues are relevant to the majority of the IPMs. Section 5 contains the conclusions of the experiments and our recommendations.

2 Computation of the search directions in IPMs for QP

After Karmarkar introduced the novel interior point method for linear programming, a number of different interior point algorithms appeared in the literature (see [12]). Among those, the infeasible primal–dual log barrier methods are considered to be the most effective for solving large–scale problems [14]. Whereas the results of Monterio and Zhou [11] show the superlinear convergence behavior of these methods, computational practice indicates their usefulness in real life [10, 2].

Primal–dual interior point algorithms are iterative approaches which seek to satisfy the Karush–Kuhn–Tucker type optimality conditions for the primal–dual problem (1-2):

$$\begin{aligned} Ax &= b, && (4) \\ A^T y + z - Qx &= c, && (5) \\ Xz &= 0, && (6) \\ (x, z) &\geq 0, && (7) \end{aligned}$$

where $X = diag(x_1, ..., x_n)$. The infeasible primal–dual interior point methods generate a sequence of iterates in both the primal and dual spaces

$$(x^k, y^k, z^k) \qquad k = 0, 1, 2, ...$$

which fulfil the strict positivity condition $(x^k, z^k) > 0$, but feasibility (4,5) and optimality (6) is reached as k increases. It is to be noted that the primal-dual interior point methods generates equivalent search directions for the separable and non–separable quadratic programming problems if the embedding primal and dual constraints in the separable equivalent are satisfied.

Since the number of iterations required by interior point methods, as the practice shows, is nearly independent of problem size, the efficiency of IPMs is strongly correlated to the speed to compute one iteration. In the case of convex quadratic problems, the effort to compute one iteration is mainly related to the cost of the symmetric factorization

$$\begin{bmatrix} -Q-D & A^T \\ A & 0 \end{bmatrix} = L\hat{D}L^T, \tag{8}$$

where $D \in R^{n\times n}$ diagonal matrix with positive diagonal values, $\hat{D} \in R^{(n+m)\times(n+m)}$ diagonal and $L \in R^{(n+m)\times(n+m)}$ is lower triangular. In general the matrix on the left side of (8) is referred to as the *augmented system.*

2.1 The normal equations approach

The most straightforward method to compute the above decomposition is the normal equation approach where the negative definite $-Q-D$ block is eliminated first by performing the Cholesky decomposition

$$\hat{F}\hat{F}^T = Q + D, \tag{9}$$

where $\hat{F} \in R^{n\times n}$ is lower triangular. Then the remaining system, whose matrix is

$$A(\hat{F}\hat{F}^T)^{-1}A^T \tag{10}$$

positive definite, is factored by a second Cholesky factorization:

$$\hat{L}\hat{L}^T = A(\hat{F}\hat{F}^T)^{-1}A^T. \tag{11}$$

In this way, the decomposition (8) can be written as:

$$L\hat{D}L^T = \begin{bmatrix} \hat{F} & \\ -A\hat{F}^{-T} & \hat{L} \end{bmatrix} \begin{bmatrix} -I & \\ & I \end{bmatrix} \begin{bmatrix} \hat{F}^T & -\hat{F}^{-1}A^T \\ & \hat{L}^T \end{bmatrix}. \tag{12}$$

In practice, it is advantageous to perform a symmetric permutation on the matrices $Q + D$ and $A(\hat{F}\hat{F}^T)^{-1}A^T$ such that the fill–in during the Cholesky factorizations is kept as low as possible. An attractive property of the normal equation approach is that this permutation for both of the decompositions can be fixed prior to applying the interior point algorithm, in a separate symbolic phase where the full attention can be paid for the sparsity issues. As it was shown in [9] the two symbolic factorizations can be performed in independent steps because the structure of $A(\hat{F}\hat{F}^T)^{-1}A^T$ is unrelated to the permutation used on $Q + D$.

2.2 The augmented system approach

As an alternative to the normal equation approach, one can use the direct symmetric factorization (8) without prescribing the elimination of the $-Q-D$ block at the beginning. As it was pointed out [5] this allows more freedom in the pivot sequence, which can be advantageous in several cases. It is to be noted that the matrix to be factored is indefinite, therefore, a technique other than the Cholesky factorization has to be used. One possibility to determine a proper pivot agenda is taking numerical properties into account when performing the factorization, as it was described in [7]. Alternatively, the pivot order can be determined by symbolic factorization if the matrix is quasidefinite [13]. Vanderbei suggested extending the primal problem with dummy slack variables with zero upper and lower bounds and using a trivial transformation on the augmented system, which makes the matrix quasidefinite. Unfortunately, by this approach the modified primal optimization problem will have empty interior which can cause numerical difficulties. An other possibility is using regularization on the augmented system [8] which also provides the quasidefinite property.

In linear programming the augmented system improves on normal equations only if the constraint matrix has special structure (see [5]). The majority of such cases, for example if few dense columns appear in A, the Shur complement mechanism is also able to achieve similar performance. In the case of nonseparable quadratic programming the augmented system approach gives better results on the majority of the problems while the Shur complement method is usually inefficient [9].

3 The separable and non-separable formulations of QPs for IPM

In the case, if the quadratic programming problem is separable, the augmented system is of the same structure as that of by linear programming because the scaling matrix $Q + D$ is diagonal. The only difference is the perturbation effect of Q which has to be taken into account when performing numeric computations. Thus, numeric and symbolic procedures of an IPM implementation for linear programming can be used for the separable quadratic case.

In the other case, however, if Q is non–separable, more sophisticated computations have to be performed. A significant difference is that the original columns of A have to be transformed when computing $A\hat{F}^{-T}$, which makes the storage management also more complicated.

If the non–separable QP problem is transformed to the separable equivalent (3), then the number of rows and columns in the model are increased by n. This increase can be reduced to the rank of Q by trivial presolve steps. For the equivalent separable problem, the augmented system will have the following form:

$$\left[\begin{array}{cc|cc} -D & & A^T & F^T \\ & -I & & -I \\ \hline A & & & \\ F & -I & & \end{array}\right]$$

which can be easily reduced to the matrix

$$\left[\begin{array}{c|cc} -D & A^T & F^T \\ \hline A & & \\ F & & I \end{array}\right] \tag{13}$$

by pivoting out the diagonal block $-I$. Thus, the matrix of the normal equation system will have the following form:

$$\left[\begin{array}{cc} AD^{-1}A^T & AD^{-1}F \\ (AD^{-1}F)^T & I+FD^{-1}F^T \end{array}\right]. \tag{14}$$

Next, we will compare how many operations are necessary by the non-separable and separable cases with the considerations that rank(Q)=n, furthermore, all matrices appearing in the problem are dense. By operation we mean a pair consisting of a floating point multiplication and addition. In the case of the original non–separable form, the decomposition (12) needs

$$\frac{n^2(n-1)}{6}+\frac{mn(n+1)}{2}+\frac{nm(m+1)}{2}+\frac{m^2(m-1)}{6},$$

operations. The terms represent the number of operations needed to compute (9), $A\hat{F}^{-T}$, (10) and (11). In the case of the separable equivalent, the number of operations required is

$$\frac{nm(m+1)}{2}+\frac{mn(n+1)}{2}+\frac{n(n+1)(n+2)}{6}+\frac{(m+n)^2(m+n-1)}{6}$$

where the terms denote the operations needed to compute $AD^{-1}A^T$, $AD^{-1}F$, $FD^{-1}F^T$ and the Cholesky decomposition of (14). It is easy to see the computations on the separable equivalent require $\frac{1}{6}n^3+\frac{1}{2}n^2+\frac{1}{3}n+\frac{1}{2}m^2n+\frac{1}{2}mn^2-\frac{1}{3}mn$ more operations than by using the non-separable form.

Whereas in the above case the non–separable form is more advantageous, the separable equivalent can be preferable if Q is of low rank.

3.1 Computation of the separable equivalent

We have implemented a procedure for building the separable equivalent of a nonseparable QP problem. In our approach we use a modified Cholesky factorization to determine F. At start, Q is permuted with the minimum local fill–in ordering for keeping the number of nonzeros in F small. Then the usual column-oriented Cholesky factorization is performed. In the ith step we compute the values in the ith column of F as:

$$F_{ii} = \sqrt{Q_{ii}-\sum_{k=1}^{i-1}F_{ki}^2}$$

$$F_{ij} = \frac{Q_{ij} - \sum_{k=1}^{i-1} F_{kj}F_{ki}}{F_{ii}} \text{ for } j = i+1, ...n.$$

Since Q can be semidefinite, we compute the column in F if the following numeric criterion is fulfilled:

$$Q_{ii} - \sum_{k=1}^{i-1} F_{ki}^2 > \max\left(\frac{Q_{ii}}{10^{12}}, 10^{-15}\right). \tag{15}$$

Otherwise, we set each entry in the ith column in F to zero and exclude the column from the forthcoming elimination steps. It is to be noted that the expression on the left side of (15) cannot be negative, furthermore, if it is zero then necessarily $F_{ij} = 0$ for all $j > i$, otherwise, Q is not positive semidefinite. These criteria are also checked during our procedure to confirm the positive semidefiniteness of Q. Our process also performs the trivial presolve step which eliminates empty columns in F and the corresponding components of p.

Table 1: Problem statistics

Name	M	N	NZ(A)	QN	NZ(Q)	rank(Q)	NZ(F)	Time
cvxqp1-l	5000	10000	14998	10000	39984	9964	278219	3.19
cvxqp1-m	500	1000	1498	1000	3984	986	17563	0.28
cvxqp2-l	2500	10000	7499	10000	39984	9964	278219	3.13
cvxqp2-m	250	1000	749	1000	3984	986	17563	0.28
cvxqp3-l	7500	10000	22497	10000	39984	9964	278219	3.19
cvxqp3-m	750	1000	2247	1000	3984	986	17563	0.27
dualc1	215	9	1935	9	45	9	45	0.11
dualc2	229	7	1603	7	28	3	6	0.11
dualc5	278	8	2224	8	36	8	36	0.11
dualc8	503	8	4024	8	36	6	21	0.11
gouldqp2	349	699	1047	349	697	348	695	0.11
gouldqp3	349	699	1047	698	1395	697	1393	0.17
laser	1000	1002	3000	1002	4002	1002	4002	0.11
mosarqp1	700	2500	3422	2500	2545	2500	2545	0.16
mosarqp2	600	900	2930	900	945	900	945	0.11

ctd. →

Name	M	N	NZ(A)	QN	NZ(Q)	rank(Q)	NZ(F)	Time
q25fv47	820	1571	10400	446	59499	446	59499	1.15
qbandm	305	472	2494	25	41	25	41	0.11
qbeaconf	173	262	3375	18	27	18	27	0.06
qbore3d	233	315	1429	28	78	28	78	0.11
qbrandy	220	249	2148	16	65	16	65	0.11
qcapri	271	353	1767	56	894	56	896	0.11
qe226	223	282	2578	67	964	67	1107	0.11
qetamacr	400	688	2409	378	4447	378	4447	0.11
qfffff80	524	854	6227	278	1916	278	1916	0.17
qforplan	161	421	4563	36	582	36	582	0.11
qgfrdxpn	616	1092	2377	54	162	54	186	0.16
qgrow15	300	645	5620	38	500	38	500	0.11
qgrow22	440	946	8252	65	852	65	854	0.11
qgrow7	140	301	2612	30	357	30	357	0.11
qisrael	174	142	2269	42	698	42	698	0.11
qpilotno	975	2172	13057	94	485	94	492	0.17
qsc205	205	203	551	11	21	11	21	0.11
qscagr25	471	500	1554	28	128	28	129	0.11
qscagr7	129	140	420	8	25	8	25	0.11
qscfxm1	330	457	2589	56	733	56	733	0.11
qscfxm2	660	914	5183	74	1131	74	1131	0.11
qscfxm3	990	1371	7777	89	1221	89	1236	0.16
qscorpio	388	358	1426	22	40	22	40	0.06
qscrs8	490	1169	3182	33	121	33	121	0.11
qscsd6	147	1350	4316	96	1404	96	1412	0.11
qscsd8	397	2750	8584	140	2510	140	3926	0.16
qsctap1	300	480	1692	36	153	36	153	0.11
qsctap2	1090	1880	6714	141	777	141	777	0.16
qsctap3	1480	2480	8874	186	1047	186	1047	0.11
qseba	515	1028	4352	96	646	96	646	0.11
qshare1b	117	225	1151	18	39	18	39	0.11
qshell	536	1775	3556	405	34790	405	34830	0.66
qship04l	402	2118	6332	14	56	14	56	0.11
qship04s	402	1458	4352	14	56	14	56	0.16

ctd. →

Name	M	N	NZ(A)	QN	NZ(Q)	rank(Q)	NZ(F)	Time
qship08l	778	4283	12802	940	34965	940	34975	0.44
qship08s	778	2387	7114	538	11677	538	11687	0.22
qship12l	1151	5427	16170	2023	62228	2023	62231	0.66
qship12s	1151	2763	8178	1042	17403	1042	17406	0.27
qsierra	1227	2036	7302	122	183	122	183	0.10
qstair	356	467	3856	66	1018	66	1018	0.11
qstandat	359	1075	3031	138	804	138	804	0.11
stcqp1	2052	4097	13338	4097	26603	4097	76621	1.54
stcqp2	2052	4097	13338	4097	26603	4097	76621	1.53

During factorization F may suffer fill–in, but it is also possible that contains fewer nonzeros than Q, due to high rank deficiency in Q. Table 1 shows the dimensions of our test problems and the characteristics of F. For our experiments we have selected all non–separable quadratic problems from [6], excepting few very small ones. Figures given include the problem name, the number of rows, columns and nonzeros in the constraint matrix, the number of quadratic variables and the number of nonzeros in the lower triangular part of Q, the number of nonempty columns and the number of nonzeros in F, furthermore, the time (in seconds) spent to compute the decomposition. In our experiments we used an IBM PC with Intel Pentium MMX 200 Mhz processor. As the results show, there are problems where heavy fill–in occurs in F. It is to be noted that in the present test problem library there are no problems where the rank deficiency of Q is high, probably because those problems were formulated in separable form in advance. Let us observe that in many cases no or very little fill–in occurs when computing F.

4 Numerical results

In our numerical tests we solved QP problems in their non-separable and separable forms. The QP problems were solved with our implementation, called BPMPD to 10^{-8} relative accuracy on an IBM PC Pentium 200 Mhz machine with 64 MB of memory without presolving. Our implementation used the normal equation or the augmented system approach depending on the result of the procedure described in [5] and [9]. In our experiment we compare the number of nonzeros in L, the number of floating point operations (given in thousands) to compute one factorization, the iteration counts and the total solution times (in seconds) taken by the different formulations. Table 2 shows the results. Columns labelled "sep." refer to the separable, whereas "nsep." denotes the nonseparable case, respectively.

The computational results indicate that the nonseparable formulation is more advantageous in the case if heavy fill–in occurs in F. The other advantageous situation for the non-separable form is if the fill–in is not significant, but Q contains considerable number of off–diagonal nonzeros. In the other cases the differences are very limited, perhaps occasional..

Table 2: Comparison of non-separable and separable QPs in IPM

problem	Nonzeros in L		Operations		Iterations		Solution time	
name	sep.	nsep.	sep.	nsep.	sep.	nsep.	sep.	nsep.
cvxqp1-m	180774	66521	20635	5409	12	9	43.06	6.10
cvxqp2-m	304931	47190	55432	2973	12	10	65.53	4.06
cvxqp3-m	245313	73604	34250	6076	12	10	73.38	8.18
dualc1	2025	1971	10	10	8	9	0.28	0.28
dualc2	1645	1624	7	6	6	7	0.22	0.27
dualc5	2296	2252	10	10	8	8	0.33	0.28
dualc8	4081	4052	18	18	7	8	0.38	0.43
gouldqp2	3482	2091	6	3	8	8	0.49	0.44
gouldqp3	8003	3827	20	9	10	7	0.77	0.50
laser	19000	6000	65	12	9	9	1.37	0.83
mosarqp1	30272	20851	337	277	9	8	2.25	1.26
mosarqp2	25067	19539	356	310	9	9	1.32	0.99
q25fv47	170951	101563	15540	8558	18	18	44.10	24.06
qbandm	7282	7056	60	60	20	19	0.82	0.76
qbeaconf	6047	5984	68	68	18	17	0.66	0.60
qbore3d	4622	4425	43	42	16	16	0.50	0.54
qbrandy	5612	5345	68	66	13	15	0.50	0.55
qcapri	6916	7062	82	93	23	27	1.15	1.20
qe226	10732	6916	189	90	16	17	0.82	0.77
qetamacr	30768	18004	621	490	16	16	2.47	2.53
qfffff80	43631	26109	1189	573	22	22	5.05	3.46
qforplan	11094	9099	206	138	19	18	1.05	0.93
qgfrdxpn	4724	4027	11	7	19	16	1.04	0.99

ctd. →

Problem	Nonzeros in L		Operations		Iterations		Solution time	
name	sep.	nsep.	sep.	nsep.	sep.	nsep.	sep.	nsep.
qgrow15	13525	12336	154	129	18	19	1.54	1.42
qgrow22	20560	17604	242	169	20	22	2.48	2.25
qgrow7	6647	5752	77	60	18	20	0.77	0.77
qisrael	5308	4577	75	67	19	19	0.66	0.60
qpilotno	61547	60090	2585	2485	20	22	8.24	9.17
qsc205	1541	1424	4	4	17	18	0.33	0.38
qscagr25	4655	4039	18	13	14	14	0.60	0.61
qscagr7	1161	1113	4	4	13	13	0.27	0.27
qscfxm1	10935	6706	166	56	20	20	1.05	0.88
qscfxm2	19913	13598	312	114	24	22	2.20	1.71
qscfxm3	27433	20299	395	167	19	23	2.47	2.47
qscorpio	3250	3102	12	11	9	10	0.39	0.38
qscrs8	10088	8866	94	73	17	17	1.16	1.21
qscsd6	13820	8772	233	62	13	11	1.10	0.82
qscsd8	30345	18451	782	155	10	11	2.20	1.48
qsctap1	4690	4124	24	21	13	15	0.55	0.61
qsctap2	23979	21348	309	280	13	14	1.82	1.93
qsctap3	33302	29438	489	428	13	15	2.53	2.81
qseba	6019	5484	28	26	20	15	1.38	1.10
qshare1b	2805	2264	17	11	16	18	0.44	0.50
qshell	13056	10614	127	125	24	23	2.80	2.03
qship04l	10122	9643	43	39	13	11	1.15	1.04
qship04s	7450	7006	31	27	11	11	0.76	0.77
qship08l	144966	140742	5507	11371	14	9	19.34	14.44
qship08s	57421	32825	1384	711	13	10	5.38	2.86
qship12l	219856	196153	6093	12458	14	11	21.97	18.73
qship12s	73625	38776	1125	534	14	14	5.27	3.62
qsierra	19451	17809	161	121	15	15	2.03	1.92
qstair	21637	18810	535	438	15	16	1.65	1.65
qstandat	8105	6696	44	36	14	13	0.93	0.83
stcqp1	158055	81427	13175	6479	7	9	23.35	9.55
stcqp2	262000	164344	24912	15000	9	9	56.57	16.97

5 Conclusion

In the paper we concerned the use of the separable equivalent of non–separable convex quadratic problems in interior point methods. Whereas the use of the separable equivalent has some technical benefits during the implementation of IPMs, our computational study showed that it has no further advantage and can be significantly less efficient than the use of the non–separable form. Results show that in cases when matrix Q has large number of nonzeros and has small rank deficiency, the use of the separable equivalent is especially disadvantageous. Whereas, we presented results without presolving, we observed that quadratic presolve is more effective on the original non–separable problem.

References

[1] E.D. Andersen, J. Gondzio, Cs. Mészáros, and X. Xu. Implementation of interior point methods for large scale linear programs. In T. Terlaky, editor, *Interior point methods of mathematical programming*, pages 189–252. Kluwer Academic Publishers, 1996.

[2] T.J. Carpenter, I.J. Lustig, J.M. Mulvey, and D.F. Shanno. Higher order predictor-corrector interior point methods with application to quadratic objectives. *SIAM Journal on Optim.*, 3:696–725, 1993.

[3] T.J. Carpenter, I.J. Lustig, J.M. Mulvey, and D.F. Shanno. Separable quadratic programming via a primal-dual interior point method and its use in a sequential procedure. *ORSA Journal on Comput.*, 5:182–191, 1993.

[4] H. M. Markowitz. The optimization of a quadratic function subject to linear constraints. *Naval Res. Logist. Quart.*, 3:111–133, 1956.

[5] I. Maros and Cs. Mészáros. The role of the augmented system in interior point methods. Technical Report TR/06/95, Brunel University, Department of Mathematics and Statistics, London, 1995. to appear in European Journal of Operations Research.

[6] I. Maros and Cs. Mészáros. A repository of convex quadratic programming problems. Technical Report DOC 97/6, Imperial College, Department of Computing, London, 1997.

[7] Cs. Mészáros. The augmented variant of ipms in two–stage stochastic linear programming. *European Journal of Operational Research*, 101(2):317–327, 1997.

[8] Cs. Mészáros. On free variables in interior point methods. Technical Report DOC 97/4, Imperial College, Department of Computing, London, 1997. to appear in Optimization Methods and Software.

[9] Cs. Mészáros. On the sparsity issues of interior point methods for quadratic programming. Working paper WP 98–4, Computer and Automation Institute, Hungarian Academy of Sciences, Budapest, 1998.

[10] R. D. C. Monteiro and I. Adler. Interior path following primal-dual algorithms. Part II: Convex quadratic programming. *Math. Programming*, 44:43–66, 1989.

[11] R.D.C. Monteiro and F. Zhou. On superlinear convergence of infeasible-interior-point algorithms for linearly constrained convex programs. Technical report, School of Industrial and System Engineering, Georgia Institute of Technology, Atlanta, 1995.

[12] C Roos, T. Terlaky, and J.-Ph. Vial. *Theory and Algorithms for Linear Optimization: An Interior Point Approach.* John Wiley & Sons, 1996.

[13] R. J. Vanderbei. Symmetric quasi-definite matrices. *SIAM Journal on Optim.*, 5(1):100–113, February 1995.

[14] S. Wright. *Primal–dual interior point methods.* SIAM, T57.74.W75, University Science Center, Philadelphia, 1997.