Quadratic programming problems - a review on algorithms and applications (Active-set and interior point methods)

Dr. Abebe Geletu

Ilmenau University of Technology
Department of Simulation and Optimal Processes (SOP)
Topics

- Introduction
- Quadratic programming problems with equality constraints
- Quadratic programming problems with inequality constraints
- Primal Interior Point methods for Quadratic programming problems
- Primal-Dual-Interior Point methods Quadratic programming problems
- Linear model-predictive control (LMPC) and current issues
- References and Resources
Introduction - Quadratic optimization (is not programming)

- A general quadratic optimization (programming) problem

\[
(QP) \quad \min_x \left\{ \frac{1}{2} x^\top Q x + q^\top x \right\}
\]

s.t.

\[
Ax = a; \\
Bx \leq b; \\
x \geq 0,
\]

where \( Q \in \mathbb{R}^{n \times n} \) symmetric (not necessarily positive definite), \( A \in \mathbb{R}^{m_1 \times n}, B \in \mathbb{R}^{m_2 \times n} \) and \( m_1 \leq n \).

- Sometimes

  instead of \( Ax = a \) we write \( a_i^\top x = a_i, i = 1, \ldots, m_1 \)

  instead of \( Bx = b \) we write \( b_j^\top x \leq b_j, j = 1, \ldots, m_2 \);

where \( a_i^\top \) is the \( i-th \) row of \( A \) and \( b_j^\top \) is the \( j-th \) row of \( B \).
Introduction...

Some applications of QP’s:
- Least Square approximations and estimation
- Portfolio optimization
- Signal and image processing, computer vision, etc.
- Optimal control, linear model predictive control, etc
- PDE-constrained optimization problems in CFD, CT, topology/shape optimization, etc
- Sequential quadratic programming (SQP) methods for NLP
- etc.

What has been achieved to date for the solution of nonlinear optimization problems has been really attained through methods of quadratic optimization and techniques of numerical linear algebra. As a result nowadays it is not surprising to see a profound interest in QP’s and their real-time computing. QP’s are now the driving force behind modern control technology.
QP introduction...SQP in brief

Given a constrained optimization problem

\[(NLP) \quad \min_x f(x) \]
\[s.t. \]
\[h_i(x) = 0, i = 1, 2, \ldots, p; \]
\[g_j(x) \leq 0, j = 1, 2, \ldots, m; \]

with Lagrange function: \( \mathcal{L}(x, \lambda, \mu) = f(x) + \lambda^\top h(x) + \mu^\top g(x). \)

A rough scheme for the SQP algorithm:

Step 0: start from \( x^0 \)

Step \( k \): \( x^{k+1} = x^k + \alpha_k d_k \), where

- the search-direction \( d^k \) is computed using a quadratic programming problem:

\[(QP)_k \quad \min_d \left\{ \frac{1}{2} d^\top H_k d + \nabla f(x_k)^\top d \right\} \]
\[s.t. \quad \nabla h_i(x_k)^\top d + h_i(x_k) = 0, i = 1, 2, \ldots, p; \]
\[\nabla g_j(x)^\top d + g_j(x_k) \leq 0, j \in \mathcal{A}(x^k) \subset \{1, 2, \ldots, m\}; \]

where \( H_k \) is the Hessian of the Lagrangian \( \nabla_x^2 \mathcal{L}(x^k, \lambda^k, \mu^k) \) or a Quasi-Newton approximation of it.

- the step-length \( \alpha_k \) is determined by a 1D minimization of a merit function \( \mathcal{M} \left( x^k + \alpha d^k \right) \) - a function that guarantees a sufficient decrease in the objective \( f(x) \) and satisfaction of constraints along \( d^k \) with an appropriate step length \( \alpha_k \).
QP introduction - trajectory tracking for autonomous vehicles

Figure: Trajectory tracking in a safe corridor

\[ \min_u \frac{1}{2} \int_{t_0}^{t_f} \left\{ [x(t) - s(t)]^\top M [x(t) - u(t)] + u(t)^\top Ru(t) \right\} \]

s.t.
\[
\dot{x}(t) = Ax(t) + Bu(t), \quad x(t_0) = x_0, \quad x(t_f) = x_f; \]
\[
x_{\min}(t) \leq x(t) \leq x_{\max}(t); \]
\[
u_{\min} \leq u(t) \leq u_{\max}; \]
\[
t_0 \leq t \leq t_f. \]
Introduction to QP - Special classes

- Tracking problems for fast systems are better treated using model-predictive control (MPC).

- Practical engineering applications frequently lead to large-scale QP’s.
- Seldom are these problems solved analytically.
- Numerical methods strongly depended on
  - the properties of the matrix $Q$ - $Q$ positive definite or not
  - if there are only equality constraints $Ax = b$
  - if the are only bound constraints $x_{min} \leq x \leq x_{max}$
  - if matrices exhibit sparsity properties and/or block structures
  - etc.
Some classification of QP’s ....

**Unconstrained QP**

\[
(QP) \quad \min_x \left\{ \frac{1}{2} x^\top Q x + q^\top x \right\}.
\]

**Box-constrained QP**

\[
(QP) \quad \min_x \left\{ \frac{1}{2} x^\top Q x + q^\top x \right\}
\quad a \leq x \leq b.
\]

**Equality constrained QP**

\[
(QP)_E \quad \min_x \left\{ \frac{1}{2} x^\top Q x + q^\top x \right\}
\quad A x = a.
\]

**Inequality constrained QP**

\[
(QP)_I \quad \min_x \left\{ \frac{1}{2} x^\top Q x + q^\top x \right\}
\quad A x = a
\quad B x \geq b.
\]
Introduction to QP - Special classes

(I) Unconstrained QP

\[ \min_x \left\{ \frac{1}{2} x^\top Q x + q^\top x \right\} \]

- **Known method:** Conjugate gradient methods (CG).
  - Basically CG is used when the matrix \( Q \) is symmetric and positive definite.

- **Variants of CG:**
  - Hestens-Steifel 1952;
  - Fletcher-Reeves 1964;
  - Polak-Ribiere 1969

As in all iterative methods, CG methods may require preconditioning techniques to guarantee convergence to the correct solution ⇒ leading to preconditioned CG (PCGG). This is necessary for large-scale QP’s.

Read:
- CG without an agonizing pain by J. R. Shewchuk.
- Matrix Computations by Golub & Van Loan.
Introduction to QP - Special classes

(II) Box-constrained QP’s

\[
\min_x \left\{ \frac{1}{2} x^\top Q x + q^\top x \right\}
\]
\[
s.t.
\]
\[
x_{min} \leq x \leq x_{max}.
\]

▶ This form of QP commonly arises, eg., :

- in image recovery and deblurring, for instance, with lower and upper bounds on the gray-levels of the image;
- in linear quadratic control or linear model predictive control, with bound constraints on the control, etc. In fact, if

\[
\min_x \left\{ \frac{1}{2} x^\top Q x + u^\top R u \right\}
\]
\[
s.t.
\]
\[
Ax + Bu = b
\]
\[
u_{min} \leq u \leq u_{max}.
\]

Theoretically, we can solve for \(x\) in terms of \(u\) (quasi-sequential) so that \(x = A^{-1} (b - Bu)\) so that we have

\[
\min_x \left\{ \frac{1}{2} \left[ A^{-1} (b - Bu) \right]^\top Q \left[ A^{-1} (b - Bu) \right] + u^\top R u \right\}
\]
\[
s.t. \quad u_{min} \leq u \leq u_{max}.
\]
Box constrained QP’s

- **Known methods**: Iterative projection algorithms.
- **Variants**:
  - The Brazilai-Borwein method (Brazilai & Borwein 1988)
  - The Nesetrov gradient method (Nesterov 1983)
  - Trust region methods (Celis, Dennis and Tapia 1985)

**Important extensions and application-specific modifications of the above: gradient projection methods.**
- Trust Region Methods by A. R. Conn, N. I. M. Gould, and Ph. L. Toint

**Some recent engineering applications:**

Quadratic programming problems - a review on algorithms and applications (Active-set and interior point methods)
Equality constrained QP’s

(III) Equality constrained QP’s

\[
\text{(QP)} \quad \min_x \left\{ \frac{1}{2} x^\top Q x + q^\top x \right\}
\]
\[
s.t. \quad Ax = a.
\]

(1)

There are two cases:
(a) \( Q \) is symmetric and positive semi-definite \( \Rightarrow \) QP is convex.
(b) \( Q \) is symmetric but not positive semi-definite \( \Rightarrow \) QP is non-convex.

**Note:** The issue that wether \( A \) is of full-rank or rank-deficient (i.e. \( \text{rank}(A) = m_1 \) or \( \text{rank}(A) < m_1 \)) is not serious.
Equality constrained QP’s

- Lagrange function: \( \mathcal{L}(x, \lambda) = \left\{ \frac{1}{2} x^\top Q x + q^\top x \right\} + \lambda^\top (A x - b) \)
- KKT conditions (KKT-equation):
  \[
  \begin{align*}
  Qx + A^\top \lambda &= -q, \\
  Ax &= b.
  \end{align*}
  \Rightarrow \begin{bmatrix} Q & A^\top \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} -q \\ b \end{bmatrix} =: K
  \]

⇒ the optimization problem is reduced to the solution of a (possibly large-scale) system of linear equations.
- \( K \) is a symmetric matrix; but it may or may not be positive definite.

Quadratic programming problems - a review on algorithms and applications (Active-set and interior point methods)

TU Ilmenau
Equality constrained QP’s

Some suggestions:

(A) A numerical analyst approach:

- If $K$ is positive definite, then use a pre-conditioned conjugate gradient (PCG) method.
- If $K$ is not positive definite, then transform $Ky = p$ to $(K^T K)y = K^T p$ and then apply PCG. Notes that the matrix $(K^T K)$ is positive definite.

- Almost all state-of-the-art linear algebra packages include a PCG solver.

Find more, eg., from: Matrix computations, 2nd ed., by Golub and Van Loan.

(B) An optimization specialist approach:

- Use (gradient) projection along with the conjugate gradient method.

Recommended reading:


Quadratic programming problems - a review on algorithms and applications (Active-set and interior point methods)

TU Ilmenau
Equality constrained QP’s

**NB:**
- If $Q$ is positive definite, the solution of the KKT-equation is a solution for the QP (convex optimization problem);
- otherwise, the solution of the PCG solver needs to be verified.

**Special case:** when $Q$ is positive definite and $A$ has full rank ($\text{rank}(A) = m_1$), the matrix $\begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix}$ is invertible.

- **Small-scale quadratic optimization problems** can be solved directly to obtain the solution

$$x^* = -Q^{-1}q + Q^{-1}A^T \left(AQ^{-1}A^T\right)^{-1} \left(b + AQ^{-1}q\right)$$
$$\lambda^* = - \left(AQ^{-1}A^T\right)^{-1} \left(b + AQ^{-1}q\right).$$

- However, in general, avoid inversion of a matrix in computations.

- **Small- to medium-scale** KKT-equations can be efficiently solved by using a direct-linear algebra algorithm, eg., using Choleski or QR factorizations.

**Further recommended reading:**
- Practical optimization by Gill, Murray and Wright.

### Quadratic programming problems - a review on algorithms and applications (Active-set and interior point methods)
QP with inequality constraints

\[
\begin{align*}
(QP)_I & \quad \min_x \left\{ \frac{1}{2} x^\top Q x + q^\top x \right\} \\
\text{s.t.} & \quad Ax = a. \\
& \quad Bx \leq b.
\end{align*}
\]

There are two classes of algorithms:
- the active-set method (ASM)
- the interior point method (IPM).
Active Set Method

Strategy:
- Start from an arbitrary point $x^0$
- Find the next iterate by setting $x^{k+1} = x^k + \alpha_k d^k$, where $\alpha_k$ is a step-length and $d^k$ is search direction.

Question
- How to determine the search direction $d^k$?
- How to determine the step-length $\alpha_k$?

(A) Determination of the search direction:
- At the current iterate $x^k$ determine the index set of active the inequality constraints

$$A^k = \{j \mid b_j^\top x^k - b_j = 0, j = 1, \ldots, m_2\}.$$
Quadratic optimization ... Active set Method

- Solve the direction finding problem

\[
\min_d \left\{ \frac{1}{2} (x^k + d)^\top Q(x^k + d) + q^\top (x^k + d) \right\}
\]

\[
\text{s.t.}
\]

\[
a_i^\top (x^k + d) = a_i, \ i = 1, \ldots, m_1;
\]

\[
b_j^\top (x^k + d) = b_j, \ j \in \mathcal{A}^k.
\]

Expand

- \( \frac{1}{2} (x^k + d)^\top Q(x^k + d) + q^\top (x^k + d) = \frac{1}{2} d^\top Qd + \frac{1}{2} d^\top Qx^k + \frac{1}{2} (x^k)^\top Qd + \frac{1}{2} (x^k)^\top Qx^k + q^\top d + q^\top x^k \)

- \( a_i^\top (x^k + d) = a_i \Rightarrow a_i^\top d = a_i - a_i^\top x^k = 0. \) Similarly, \( b_j^\top d = b_j - b_j^\top x^k = 0. \)

- Simplify these expressions and drop constants to obtain:

\[
\min_d \left\{ \frac{1}{2} d^\top Qd + [Qx^k + q]^\top d \right\}
\]

\[
\text{s.t.}
\]

\[
Ad = 0,
\]

\[
\tilde{B}d = 0;
\]

where

\[
\tilde{B} = \begin{bmatrix} \vdots \\ b_j^\top \\ \vdots \end{bmatrix}, \ j \in \mathcal{A}^k. \ Set \ g^k = Qx^k + q.
\]
Quadratic optimization ... Active set Method

- To obtain the search direction $d^k$, solve the equality constrained QP:

$$\min_d \left\{ \frac{1}{2} d^\top Q d + [g^k]^\top d \right\}$$

s.t.

$$Ad = 0,$$
$$\tilde{B}d = 0.$$  

The KKT optimality conditions lead to the system:

$$Qd + g^k + A^\top \lambda + \tilde{B}^\top \tilde{\mu} = 0,$$
$$Ad = 0, \quad \Rightarrow \begin{bmatrix} Q & A^\top & \tilde{B}^\top \\ A & \bigcirc & \bigcirc \\ \tilde{B} & \bigcirc & \bigcirc \end{bmatrix} \begin{bmatrix} d \\ \lambda \\ \tilde{\mu} \end{bmatrix} = \begin{bmatrix} -g^k \\ 0 \\ 0 \end{bmatrix} \quad (\ast)$$

where $\lambda$ and $\tilde{\mu}$ are Lagrange multipliers corresponding to equality and active inequality constraints, respectively.
Quadratic optimization … Active set Method

- Apply algorithms for equality constrained QP’s to obtain $d^k$.

- If $d^k$ is a solution of QP, then there are $\lambda_k^k$ and $\tilde{\mu}_k$ such that

  (KKT conditions for the equality constrained QP)

  \[
  Qd^k + g^k + A^T \lambda^k + \tilde{B}^T \tilde{\mu}^k = 0,
  \]

  \[
  Ad^k = 0,
  \]

  \[
  \tilde{B}d^k = 0.
  \]

- There are two cases: either $d^k = 0$ or $d^k \neq 0$.

**Case 1**: $d^k = 0$. Then, the system above reduces to

  \[
  g^k + A^T \lambda^k + \tilde{B}^T \tilde{\mu}^k = 0, \quad (**)\]

**Case 1-a**: If $\tilde{\mu}^k \geq 0$, (i.e. $\tilde{\mu}_j^k \geq 0$, $j \in A^k$), then corresponding to the non-active constraints we can set $\mu_j^k = 0$ for $j \in \{1, 2, \ldots, m_2\} \setminus A^k$. Hence, we have for the original problem

  \[
  g^k + A^T \lambda^k + B\mu^k = 0,
  \]

  \[
  Ax^k - a = 0,
  \]

  \[
  Bx^k - b \leq 0,
  \]

  \[
  (Bx^k - b_k)^T \mu^k = 0
  \]

  $\mu_k \geq 0$.

Quadratic programming problems - a review on algorithms and applications (Active-set and interior point methods)
Quadratic optimization … Active set Method

• If $d^k = 0$ and $\tilde{\mu}_k \geq 0$, $x^{k+1} = x^k$ is a KKT point. Stop!

• **Case 1-b:** If some components of $\tilde{\mu}^k$ are negative, then $x^k$ is not an optimal solution. Let $\mu_{j_0} = \min \{\tilde{\mu}_j \mid \tilde{\mu}_j < 0, j \in A^k\}$. Remove the index $j_0$ from $A^k$ and solve the quadratic programming problem

$$
\min_d \left\{ \frac{1}{2} d^\top Q d + [g^k]^\top d \right\}
$$

s.t.

$$
a_i^\top d = 0, \\
b_j^\top d = 0, j \in A^k \setminus \{j_0\}.
$$

Then the direction obtained is descent direction $d^k$ for $(QP)_I$. 

Case 2: $d^k \neq 0$.

- Determine a step-length $\alpha_k$ that guarantees $x^k + \alpha_k d^k$ is feasible to (QP)$_I$.

We need to choose $\alpha_k$ so that $Ax^{k+1} = A(x^k + \alpha_k d^k) = a$ and $Bx^{k+1} = B(x^k + \alpha_k d^k) \leq b$.

- For the equality constraints $Ax = b$, since $x^k$ is feasible to (QP)$_I$ and $Ad_k = 0$ (in (QP)$_E$), we have

\[
Ax^{k+1} = A(x^k + \alpha_k d^k) = Ax^k + \alpha_k Ad_k = a + 0 = a
\]

So the equality constraints are satisfied for any $\alpha_k$.

- For the active inequality constraints $Bx \leq b$, similarly we have $b_j^T x^{k+1} = b_j^T (x^k + \alpha_k d^k) = b_j \leq b_j, j \in \mathcal{A}_k$ for any $\alpha_k$.

- For the inactive inequality constraints at $x^k$, i.e. $j \notin \mathcal{A}_k$, we have $b^T x^k < b_j$. Thus, we need to determine $\alpha_k$ so that $b_j^T (x^k + \alpha_k d^k) \leq b_j$ holds true for $j \notin \mathcal{A}_k$. Hence,

\[
b_j^T x^k + \alpha_k b_j^T d^k) \leq b_j \Rightarrow \alpha_k b_j^T d^k \leq b_j - b_j^T x^k.
\]

(i) If $b_j^T d^k \leq 0$, then $\alpha_k b_j^T d^k \leq b_j - b_j^T x^k$ is satisfied for any $\alpha_k > 0$, since $0 \leq b_j - b_j^T x^k$ (recall that $x^k$ is feasible for (QP)$_I$).

(ii) If $b_j^T d^k > 0$, then choose $\alpha_k$

\[
\alpha_k = \frac{b_j - b_j^T d^k}{b_j^T d^k}.
\]
Quadratic optimization ... Active set Method

• A common $\alpha_k$ that guarantees the satisfaction of all constraints is

$$\alpha_k = \min \left\{ 1, \frac{b_j - b_j^\top d^k}{b_j^\top d^k} \mid j \notin A^k \text{ and } b_j^\top d^k > 0 \right\}.$$ 

• Updating the active index set:

Observe that if $\alpha_k < 1$, then $\alpha_k = \frac{b_{j_0} - b_{j_0}^\top d^k}{b_{j_0}^\top d^k}$ for some $j_0 \notin A^k$. This implies that $b_{j_0}^\top (x^k + \alpha_k d^k) = b_{j_0}$. That is, the inequality constraint corresponding to the index $j_0$ becomes active.

• If $\alpha_k < 1$, then update the active index set as

$$A^{k+1} = A^k \cup \{j_0\}.$$
Algorithm 1: An active-set algorithm for QP

1: Give a start vector $x^0$;
2: Identify the active index set $A^0$;
3: Set $k = 0$;
4: while (no convergence) do
5:  Compute $g^k = Qx^k + q$;
6:  Obtain $d^k, \lambda^k$ and $\tilde{\mu}^k$ by solving the KKT-equations for
7:    \[
    \min_d \left\{ \frac{1}{2} d^T Q d + [g^k]^T d \right\}
    \]
8:    s.t.
9:    \[
    a_i^T d = 0,
    \]
10:   \[
    b_j^T d = 0, j \in A^k.
    \]
11: if $(d^k = 0)$ then
12:    if $\tilde{\mu}^k \geq 0$ then
13:        STOP! $x^k$ is a KKT point.
14:    else
15:        $\tilde{\mu} j_0 = \min \{ \tilde{\mu}_j | \tilde{\mu}_j < 0, j \in A^k \}$
16:    end if
17: end if
18: end while((continue..))
16. if \( d^k \neq 0 \) 17. Compute the step-length

\[
\alpha_k = \min \left\{ 1, \frac{b_j - b_j^\top d^k}{b_j^\top d^k} \mid j \notin A^k \text{ and } b_j^\top d^k > 0 \right\}.
\]

18. Update \( x^{k+1} = x^k + \alpha_k d^k \).

19. Update active index-set: if \( \alpha_k = 1 \), then \( A^{k+1} = A^k \), else \( A^{k+1} = A^k \cup \{j_0\} \), where \( \alpha_k = \frac{b_{j_0} - b_{j_0}^\top d^k}{b_{j_0}^\top d^k} \) for \( b_{j_0}^\top d^k > 0 \).

20. Update \( k \leftarrow k + 1 \).
21. end if
22. end while

**Important issues:**

- How to determine a starting iterate \( x^0 \) for the active-set algorithm.
- ASM requires an efficient strategy for the determination of active sets \( A^k \) at each \( x^k \).
Quadratic optimization ... Active set Method

Advantages of ASM:
• Since only active constraints are consider at each iteration \( x^k \), (QP)_E usually has only a few constraints and can be solved fast. ⇒ Large-scale (QP)_I’s are easy to solve.
• In many cases the active set varies slightly from step-to-step, making active set method efficient. ⇒ Data obtained from the current QP_E can be used to solve the next QP_E known as warm starting.
• All iterates \( x^k \) are feasible to (QP)_I. This is an important property, eg, in SQP.

Disadvantages of ASM:
• Since the active-set \( A^k \) may vary from step to step, the structure and properties, eg. sparsity, of constraint matrices may change.
• ASM may become slower near to the optimal solution;.
• For some problems ASM can be computationally expensive.
Consider

\[
(QP) \quad \min_x \left\{ \frac{1}{2} x^\top Q x + q^\top x \right\}
\]

\[s.t.\]

\[Ax = a.\]
\[x \geq 0,\]

where \( A \in \mathbb{R}^{m \times n} \).

Any other constraints can be transformed to the above form.

- If there is \( Bx \leq b, \) then add slack variables \( s \) to obtain \( Bx + s = b, \ s \geq 0 \) so that the constraints become

\[
\begin{bmatrix}
A & 0 \\
B & I_n
\end{bmatrix}
\begin{bmatrix}
x \\
s
\end{bmatrix}
=
\begin{bmatrix}
a \\
b
\end{bmatrix}
\]
\[\begin{bmatrix}
x \\
s
\end{bmatrix}
\geq
0.
\]

- If there is a box constraint \( x_{min} \leq x \leq x_{max}, \) then split it in two constraints

\[
x_{max} - x \geq 0
\]
\[
x - x_{min} \geq 0.
\]
Quadratic optimization... PD Interior point methods

⇒ Use the logarithmic barrier function for the non-negative variables \( x \).
⇒ The barrier problem associated to \((QP)\) will be

\[
(BQP)_\mu \quad \min_x \begin{cases} 
\frac{1}{2} x^\top Q x + q^\top x - \mu \sum_{j=1}^{n} \log x_i \\
\end{cases}
\]

s.t.

\[Ax = a.\]

• The Lagrange function of \((BQP)_\mu\)

\[
\mathcal{L}_\mu(x, \lambda) = \frac{1}{2} x^\top Q x + q^\top x - \lambda^\top (Ax - a) - \mu \sum_{j=1}^{n} \log x_i,
\]

where \( x > 0 \).
Quadratic optimization... PD Interior point methods

- The KKT condition for \((BQP)_\mu\) will be

\[
\frac{\partial L_\mu}{\partial x} = 0 \Rightarrow Qx + q - A^\top \lambda - \mu X^{-1}e = 0, \ x > 0.
\]

\[
\frac{\partial L_\mu}{\partial \lambda} = 0 \Rightarrow Ax = b,
\]

where \(e = (1, \ldots, 1)\).

\[
\Rightarrow Qx - A^\top \lambda - \mu X^{-1}e = -q
\]

\[
Ax = b.
\]

- Define \(s = \mu X^{-1}e\). Then \(s \in \mathbb{R}^n, \ s > 0\) and \(x_is_i = \mu, \ i = 1, \ldots, n\).

- The KKT condition for \((BQP)_\mu\) takes the form

\[
\Rightarrow Qx - A^\top \lambda - s = -q
\]

\[
Ax = b
\]

\[
x_is_i = \mu, \ i = 1, \ldots n.
\]
For each fixed parameter $\mu$, the KKT condition for $(\text{BQP})_\mu$ is a system of non-linear equations (nonlinearity due to the product $x_i s_i = \mu$).

- There are $2n + m_1$ variables $(x, \lambda, s)$. Set

$$F_\mu(x, \lambda, s) = \begin{bmatrix} Qx - A^\top \lambda - s - q \\ Ax - b \\ XS - \mu e \end{bmatrix},$$

where $X = \text{diag}(x) \in \mathbb{R}^{n \times n}$ and $S = \text{diag}(s) \in \mathbb{R}^{n \times n}$.

- For each fixed $\mu > 0$, the equation

$$F_\mu(x, \lambda, s) = 0.$$

should be solved.
Quadratic optimization... PD Interior point methods

- The system can be solved using a Newton-like algorithm.

**A general Newton algorithm:**

**Step 0:** Give an initial iterate \((x^0, \lambda^0, s^0)\) where \((x^0, s^0) > 0\).

**Step k:** Given \((x^k, \lambda^k, s^k)\):

- solve the system of equations

\[
J_{F\mu} \left( x^k, \lambda^k, s^k \right) d = -F_\mu \left( x^k, \lambda^k, s^k \right)
\]

- Determine a step-length \(\alpha_k\).

- Set

\[
\begin{align*}
x^{k+1} &= x^k + \alpha_k \Delta x_k \\
\lambda^{k+1} &= \lambda^k + \alpha_k \Delta \lambda_k \\
s^{k+1} &= s^k + \alpha_k \Delta s_k.
\end{align*}
\]
Quadratic optimization... PD Interior point methods

- At each iteration the system of linear equations

\[ J_{F_\mu} \left( x^k, \lambda^k, s^k \right) d = -F \left( x^k, \lambda^k, s^k \right) \]

should be solved.
- The Jacobian matrix \( J_{F_\mu} \left( x^k, \lambda^k, s^k \right) \) has the special form

\[
J_{F_\mu} \left( x^k, \lambda^k, s^k \right) = \begin{bmatrix}
Q & -A^T & -I \\
A & 0 & 0 \\
S^k & 0 & X^k \\
\end{bmatrix},
\]

where \( X^k = \text{diag}(x^k) \) and \( S^k = \text{diag}(s^k) \).

In interior-point method, most computation effort is spent for solving the system of linear equations to determine \( d^k \). This should be accomplished by an efficient linear algebra solver.
There are various primal-dual interior-point algorithms with modifications on the above general algorithm. * Path-following Algorithm * Affine-scaling Algorithm, * Mehrotra predicator-corrector Algorithm etc.

Algorithm 2: Path-following Algorithm for QP

1: Choose $\sigma_{min}$;
2: Choose initial iterates $(x^0, \lambda^0, s^0)$ with $(x^0, s^0) > 0$.
3: Set $k = 0$;
4: while $((x^k)^\top s^k / n) > \varepsilon$ do
5:   Choose $\sigma_k \in [\sigma_{min}, \sigma_{max}]$;
6:   Solve the system
   
   $$J_{F, \mu}(x^k, \lambda^k, s^k) d = \begin{bmatrix}
   Qx - A^\top \lambda - s - q \\
   Ax - b \\
   XS - \sigma_k \mu_k e
   \end{bmatrix},$$
   
   to determine $d^k = (\Delta x_k, \Delta \lambda_k, \Delta s_k)$.
7:   Choose $\alpha_{max}$ the largest value such that
   
   $$(x^k, s^k) + \alpha (\Delta x_k, \Delta s_k) > 0;$$
8:   Set $\alpha_k = \min \{1, \eta_k \alpha_{max}\}$ for some $\eta_k \in (0, 1)$ and $\mu_k = \frac{(x^k)^\top s^k}{n}$;
9:   Set $(x^{k+1}, \lambda^{k+1}, s^{k+1}) = (x^k, \lambda^k, s^k) + \alpha_k (\Delta x_k, \Delta \lambda_k, \Delta s_k)$
10: Update $k \leftarrow k + 1$.
11: end while
Quadratic optimization... PD Interior point methods

- Commonly used termination criteria is

$$\frac{(x^k)^T s^k}{n} \leq \varepsilon.$$

for some termination tolerance $\varepsilon > 0$, eg. $\varepsilon = 0.001$, etc.

**Some strategies for choice of centering parameter:**

(a) $\sigma_k = 0$, $k = 1, 2, \ldots$, - affine-scaling approach;

(b) $\sigma_k = 1$, $k = 1, 2, \ldots$,

(c) $\sigma_k \in [\sigma_{min}, \sigma_{max}] = 1$, $k = 1, 2, \ldots$ Commonly, $\sigma_{min} = 0.01$ and $\sigma_{max} = 0.75$ (path following method)

(d) $\sigma_k = 1 - \frac{1}{\sqrt{n}}$, $k = 1, 2, \ldots$, (with $\alpha_k = 1$ - short-step path-following method)
Quadratic optimization... PD Interior point methods

Advantages of PDIPM:
• Can be used to solve medium- to large-scale QPs ⇒ so important in optimal control.
• Fast convergence properties.
• Structure of the Jacobian matrix does not change a lot.
• The Jacobian matrix $J_{F_{\mu_k}}$ takes a special structure, which can be exploited by a linear algebra solver.

Disadvantages of PDIPM:
• Usually requires a strictly feasible initial iterate $(x^0, s^0)$ which is not trivial to obtain.
• All constraints are used in the computation of the search direction $d^k$, even if some are not active.
⇒ The computation of $d^k$ may be only attained through iterative linear algebra solvers.
Some current Issues

- **Optimization methods for real-time control.**
  Applications for fast real-time systems control:
  - Control of autonomous (ground, aerial or underwater) vehicles and Robots computation of QP’s in microseconds.
  ⇒ Require hardware (embedded systems) implementation of optimization algorithms.

- **Parallel implementation on graphic processors.**
  - Computer graphic cards provided an opportunity for parallel linear algebra computation.
  ⇒ Large QP’s can solved on shared memory PC with graphic processors.
  Example:
    - Graphic cards with processors: Latest NVIDIA Graphic cards, AMD Athlon graphic cards.
    - Programming Languages: CUDA, OpenCL (Athlon), etc.

- **Parallel implementation on shared and distributed memory (multiprocess) computers.**
  For instance in the
    - numerical solution of optimization problems with partial differential constrains;
      Applications: CFD optimization, Computer Tomography, Thermodynamics, etc
    - large-scale stochastic simulation and optimization.
Resources

Some use full links:

- A quadratic programming page, N. Gould and P. Toint
  URL: http://www.numerical.rl.ac.uk/qp/qp.html
- Computational Optimization Laboratory, Y. Ye, Stanford University,
  URL: http://www.stanford.edu/ yyye/Col.html
- Convex Optimization, Stephen P. Boyd, Stanford University,
  URL: http://www.stanford.edu/ boyd/software.html (Book, software, etc)
- The HSL Mathematical Software Library (HSL=Harwell Subroutine Library),
  Numerical Analysis Group at the STFC Rutherford Appleton Laboratory
  URL: http://www.hsl.rl.ac.uk/

Software:

- MINQ: General Definite and Bound Constrained Indefinite Quadratic Programming, Arnold Neumaier, University of Vienna
  URL: http://www.mat.univie.ac.at/ neum/software/minq/
- qpOASES - Online Active Set Strategy, H.J. Ferreau, University of Lauven (Belgium),
  URL: http://www.kuleuven.be/optec/software/qpOASES/
- QPC - Quadratic Programming in C Adrian Wills, University of Newcastle,
  URL: http://sigpromu.org/quadprog/
- OOQP is an object-oriented C++ package, M. Gertz and S. Wright
  URL: http://pages.cs.wisc.edu/ swright/ooqp/
- Interior point Optimizer (IpOpt), A. Wächter and C. Laird URL: https://projects.coin-or.org/Ipopt/
- NLPy is a Python package for numerical optimization, D. Orban, Ecole Polytechnique de Montreal,
  URL: http://nlpy.sourceforge.net/
- Python based optimization solvers
  URL: http://wiki.python.org/moin/NumericAndScientific

Quadratic programming problems - a review on algorithms and applications (Active-set and interior point methods)

TU Ilmenau
Some references on active set methods:

Some references on interior point methods:
Some references on real-time control:
- A. Wills and A. Mills and B. Ninness: FPGA implementation of an interior-point solution for linear model predictive control. The 18th IFAC World Congress, Milano (Italy), August 28 - September 2, 2011.
References

Some classical references on conjugate gradient methods:
- J.R. Shewchuk: An Introduction to the Conjugate Gradient Method Without the Agonizing Pain.
  (URL: http://www.cs.cmu.edu/~quake-papers/painless-conjugate-gradient.pdf)

Some references on box-constrained QP:

Quadratic programming problems - a review on algorithms and applications (Active-set and interior point methods)