Quadratic Programming: Theory and Algorithms

Zola Donovan\textsuperscript{1} and Murong Xu\textsuperscript{1}

\textsuperscript{1}Department of Mathematics
West Virginia University
Morgantown, WV USA

March 3, 2015
1 Motivating Examples
1 Motivating Examples
2 The Quadratic Programming Problem
Outline

1 Motivating Examples
2 The Quadratic Programming Problem
3 Optimality Conditions
Outline

1. Motivating Examples
2. The Quadratic Programming Problem
3. Optimality Conditions
4. Interior-Point Methods
1 Motivating Examples
2 The Quadratic Programming Problem
3 Optimality Conditions
4 Interior-Point Methods
5 Examples and QP Software
6 References
The Casino Game

Example (1)

Suppose you are given the choice of playing one of two games at a casino.

Game X has a 5% chance of winning $1000, and a 95% chance of winning nothing.

Game Y has a 5% chance of winning $5000.

If you lose however, you have to pay the casino $200.

You are allowed to play this game one time.

Which game would you choose to play?
Example (1)

Suppose you are given the choice of playing one of two games at a casino. Game X has a 5% chance of winning $1000, and a 95% chance of winning nothing. Game Y has a 5% chance of winning $5000. If you lose however, you have to pay the casino $200. You are allowed to play this game one time. Which game would you choose to play?
The Casino Game

Example (1)

- Suppose you are given the choice of playing one of two games at a casino.
The Casino Game

Example (1)
- Suppose you are given the choice of playing one of two games at a casino.
- Game X has a 5% chance of winning $1000, and a 95% chance of winning nothing.
Example (1)

- Suppose you are given the choice of playing one of two games at a casino.
- Game X has a 5% chance of winning $1000, and a 95% chance of winning nothing.
- Game Y has a 5% chance of winning $5000.
Example (1)

- Suppose you are given the choice of playing one of two games at a casino.
- Game X has a 5% chance of winning $1000, and a 95% chance of winning nothing.
- Game Y has a 5% chance of winning $5000. If you lose however, you have to pay the casino $200.
Example (1)

- Suppose you are given the choice of playing one of two games at a casino.
- Game X has a 5% chance of winning $1000, and a 95% chance of winning nothing.
- Game Y has a 5% chance of winning $5000. If you lose however, you have to pay the casino $200.
- You are allowed to play this game one time.
The Casino Game

Example (1)

- Suppose you are given the choice of playing one of two games at a casino.
- Game X has a 5% chance of winning $1000, and a 95% chance of winning nothing.
- Game Y has a 5% chance of winning $5000. If you lose however, you have to pay the casino $200.
- You are allowed to play this game one time.
- Which game would you choose to play?
Portfolio Optimization

We wish to invest $1000.00 in stocks A, B, and C for a one month period. We buy a stock at some dollar amount per share in the beginning of the month, and sell it at some dollar amount per share at the end of the month. The rate of return of each stock is a random variable with some expected value. Our goal is to invest in such a way that the expected end-of-month return is at least $50.00 or 5%.
We wish to invest $1000.00 in stocks A, B, and C for a one month period. We buy a stock at some dollar amount per share in the beginning of the month, and sell it at some dollar amount per share at the end of the month. The rate of return of each stock is a random variable with some expected value. Our goal is to invest in such a way that the expected end-of-month return is at least $50.00 or 5%. 

Example (2)
Portfolio Optimization

Example (2)

- We wish to invest $1000.00 in stocks A, B, and C for a one month period.
Example (2)

- We wish to invest $1000.00 in stocks A, B, and C for a one month period.
- We buy a stock at some dollar amount per share in the beginning of the month,
Example (2)

- We wish to invest $1000.00 in stocks A, B, and C for a one month period.
- We buy a stock at some dollar amount per share in the beginning of the month, and sell it at some dollar amount per share at the end of the month.
Example (2)

- We wish to invest $1000.00 in stocks A, B, and C for a one month period.
- We buy a stock at some dollar amount per share in the beginning of the month, and sell it at some dollar amount per share at the end of the month.
- The rate of return of each stock is a random variable with some expected value.
Portfolio Optimization

Example (2)

- We wish to invest $1000.00 in stocks A, B, and C for a one month period.
- We buy a stock at some dollar amount per share in the beginning of the month, and sell it at some dollar amount per share at the end of the month.
- The rate of return of each stock is a random variable with some expected value.
- Our goal is to invest in such a way that the expected end-of-month return is at least $50.00 or 5%.
Optimization Approach

An optimization approach to the decision problems:
1. Build a mathematical model of the decision problem.
2. Analyze available quantitative data to use in the mathematical model.
3. Use a numerical method to solve the mathematical model.
4. Infer the actual decision from the solution to the mathematical model.
An optimization approach to the decision problems:
An optimization approach to the decision problems:

- Build a mathematical model of the decision problem.
An optimization approach to the decision problems:

- Build a mathematical model of the decision problem.
- Analyze available quantitative data to use in the mathematical model.
Optimization Approach

An optimization approach to the decision problems:

- Build a mathematical model of the decision problem.
- Analyze available quantitative data to use in the mathematical model.
- Use a numerical method to solve the mathematical model.
An optimization approach to the decision problems:

- Build a mathematical model of the decision problem.
- Analyze available quantitative data to use in the mathematical model.
- Use a numerical method to solve the mathematical model.
- Infer the actual decision from the solution to the mathematical model.
The Quadratic Programming Problem

Quadratic programming (QP) refers to the problem of optimizing a quadratic function, subject to linear equality and inequality constraints. The general quadratic program can be stated as follows:

\[ \begin{align*}
\text{Optimize} & \quad \frac{1}{2} x^T Q x + c^T x \\
\text{subject to} & \quad Ax \leq b, \quad A_e = b, \quad \text{or} \quad Ax \geq b
\end{align*} \]

QPs are special classes of nonlinear optimization problems, and contain linear programming problems as special cases.
The quadratic programming (QP) problem

- Quadratic programming (QP) refers to the problem of optimizing a quadratic function,
The quadratic programming (QP) problem

- Quadratic programming (QP) refers to the problem of optimizing a quadratic function, subject to linear equality and inequality constraints.
The Quadratic Programming Problem

The quadratic programming (QP) problem

- Quadratic programming (QP) refers to the problem of optimizing a quadratic function, subject to linear equality and inequality constraints.
- The general quadratic program can be stated as follows:
The quadratic programming (QP) problem

- Quadratic programming (QP) refers to the problem of optimizing a quadratic function, subject to linear equality and inequality constraints.

- The general quadratic program can be stated as follows:

\[
\text{Optimize } \frac{1}{2} \cdot x^T \cdot Q \cdot x + c^T \cdot x
\]
Quadratic programming (QP) refers to the problem of optimizing a quadratic function, subject to linear equality and inequality constraints.

The general quadratic program can be stated as follows:

$$\text{Optimize } \frac{1}{2} \cdot \mathbf{x}^T \cdot \mathbf{Q} \cdot \mathbf{x} + \mathbf{c}^T \cdot \mathbf{x}$$

$$\mathbf{A} \cdot \mathbf{x} \{\leq, =, \text{or } \geq\} \mathbf{b}$$
The Quadratic Programming Problem

The quadratic programming (QP) problem refers to the problem of optimizing a quadratic function, subject to linear equality and inequality constraints.

The general quadratic program can be stated as follows:

Optimize $\frac{1}{2} \cdot x^T \cdot Q \cdot x + c^T \cdot x$

$A \cdot x \{\leq, =, \text{or} \geq\} b$

$x \geq 0$
The quadratic programming (QP) problem

- Quadratic programming (QP) refers to the problem of optimizing a quadratic function, subject to linear equality and inequality constraints.

- The general quadratic program can be stated as follows:

\[
\text{Optimize } \frac{1}{2} \cdot x^T \cdot Q \cdot x + c^T \cdot x \\
A \cdot x \{\leq, =, \text{or } \geq\} b \\
x \geq 0
\]

- QPs are special classes of nonlinear optimization problems,
The Quadratic Programming Problem

The quadratic programming (QP) problem

- Quadratic programming (QP) refers to the problem of optimizing a quadratic function, subject to linear equality and inequality constraints.
- The general quadratic program can be stated as follows:

\[
\text{Optimize } \frac{1}{2} \cdot x^T \cdot Q \cdot x + c^T \cdot x \\
A \cdot x \ \{\leq, =, \text{or} \geq\} \ b \\
\quad \ x \geq 0
\]

- QPs are special classes of nonlinear optimization problems, and contain linear programming problems as special cases.
The Quadratic Programming Problem

In standard form, QPs may be represented as follows:

$$\min_x f(x) = \frac{1}{2} x^T Q x + c^T x \quad \text{subject to} \quad A x = b, \quad x \geq 0$$
The quadratic programming (QP) problem

- In standard form,
The quadratic programming (QP) problem

- In standard form, QPs may be represented as follows:
The quadratic programming (QP) problem

- In standard form, QPs may be represented as follows:

\[ \min_{x} \ f(x) := \frac{1}{2} \cdot x^T \cdot Q \cdot x + c^T \cdot x \]
The quadratic programming (QP) problem

In standard form, QPs may be represented as follows:

$$
\min_x f(x) := \frac{1}{2} \cdot x^T \cdot Q \cdot x + c^T \cdot x \\
A \cdot x = b
$$
The quadratic programming (QP) problem

In standard form, QPs may be represented as follows:

\[
\min_x \quad f(x) := \frac{1}{2} \cdot x^T \cdot Q \cdot x + c^T \cdot x \\
A \cdot x = b \\
x \geq 0
\]
The Quadratic Programming Problem

Positive semidefinite matrix $Q$

Recall that, when $Q$ is a positive semidefinite matrix, i.e., when $y^T \cdot Q \cdot y \geq 0$ for all $y$, the objective function of the problem is a convex function of $x$. In this case, a local minimizer of the objective function is also a global minimizer.

The figure below shows the graph and contours of a quadratic function with a positive semidefinite $Q$.

Figure: Graph and contours of a convex function
Motivating Examples

The Quadratic Programming Problem

Optimality Conditions

Interior-Point Methods

Examples and QP Software

References

The Quadratic Programming Problem

**Positive semidefinite matrix** $\mathbf{Q}$

- Recall that, when $\mathbf{Q}$ is a positive semidefinite matrix,
Positive semidefinite matrix $Q$

- Recall that, when $Q$ is a positive semidefinite matrix, i.e., when $\mathbf{y}^T \cdot Q \cdot \mathbf{y} \geq 0$ for all $\mathbf{y}$,
The Quadratic Programming Problem

Positive semidefinite matrix $Q$

Recall that, when $Q$ is a positive semidefinite matrix, i.e., when $y^T \cdot Q \cdot y \geq 0$ for all $y$, the objective function of the problem is a convex function of $x$. 

...
Recall that, when $Q$ is a positive semidefinite matrix, i.e., when $y^T \cdot Q \cdot y \geq 0$ for all $y$, the objective function of the problem is a convex function of $x$.

In this case, a local minimizer of the objective function is also a global minimizer.
Recall that, when $Q$ is a positive semidefinite matrix, i.e., when $y^T \cdot Q \cdot y \geq 0$ for all $y$, the objective function of the problem is a convex function of $x$.

In this case, a local minimizer of the objective function is also a global minimizer.

The figure below shows the graph and contours of a quadratic function with a positive semidefinite $Q$. 
The Quadratic Programming Problem

Positive semidefinite matrix $Q$

- Recall that, when $Q$ is a positive semidefinite matrix, i.e., when $y^T \cdot Q \cdot y \geq 0$ for all $y$, the objective function of the problem is a convex function of $x$.
- In this case, a local minimizer of the objective function is also a global minimizer.
- The figure below shows the graph and contours of a quadratic function with a positive semidefinite $Q$.

![Graph and contours of a convex function](image)

Figure: Graph and contours of a convex function
The Quadratic Programming Problem

When $Q$ is not a positive semidefinite matrix (either indefinite or negative semidefinite), the objective function is nonconvex, and may have local minimizers that are not global minimizers.

The figure below shows the graph and contours of a quadratic function with an indefinite $Q$.

Figure: Graph and contours of a nonconvex function
Indefinite or negative semidefinite matrix $Q$

- When $Q$ is not a positive semidefinite matrix (either indefinite or negative semidefinite),
Motivating Examples
The Quadratic Programming Problem
Optimality Conditions
Interior-Point Methods
Examples and QP Software
References

The Quadratic Programming Problem

Indefinite or negative semidefinite matrix $Q$

- When $Q$ is not a positive semidefinite matrix (either indefinite or negative semidefinite), the objective function is nonconvex,
Indefinite or negative semidefinite matrix $Q$

- When $Q$ is not a positive semidefinite matrix (either indefinite or negative semidefinite), the objective function is nonconvex, and may have local minimizers that are not global minimizers.
The Quadratic Programming Problem

Indefinite or negative semidefinite matrix $Q$

- When $Q$ is not a positive semidefinite matrix (either indefinite or negative semidefinite), the objective function is nonconvex, and may have local minimizers that are not global minimizers.

- The figure below shows the graph and contours of a quadratic function with an indefinite $Q$. 

Figure: Graph and contours of a nonconvex function

Z. Donovan and M. Xu
Optimization Methods in Finance
Indefinite or negative semidefinite matrix $Q$

- When $Q$ is not a positive semidefinite matrix (either indefinite or negative semidefinite), the objective function is nonconvex, and may have local minimizers that are not global minimizers.

- The figure below shows the graph and contours of a quadratic function with an indefinite $Q$.

**Figure:** Graph and contours of a nonconvex function
The Quadratic Programming Problem

As in linear programming, we can develop a dual of quadratic programming problems. The dual of \( f(x) \) is given below:

\[
\begin{align*}
\text{max} & \quad x^T y - \frac{1}{2} x^T Q x + s A^T y - Q x + s = c \\
x & \geq 0, \quad s \geq 0
\end{align*}
\]

Note that, unlike the case of linear programming, the variables of the primal quadratic programming problem also appear in the dual QP.
The dual of the QP problem

As in linear programming, we can develop a dual of quadratic programming problems.
As in linear programming, we can develop a dual of quadratic programming problems. The dual of $f(x)$ is given below:
As in linear programming, we can develop a dual of quadratic programming problems. The dual of \( f(x) \) is given below:

\[
\begin{align*}
\max_{x, y, s} & \quad b^T y - \frac{1}{2} x^T Q x \\
\text{s.t.} & \quad A^T y - Q x + s = c \\
& \quad x \geq 0, \quad s \geq 0
\end{align*}
\]
The Quadratic Programming Problem

As in linear programming, we can develop a dual of quadratic programming problems.

The dual of $f(x)$ is given below:

$$
\max_{x,y,s} \quad b^T \cdot y - \frac{1}{2} \cdot x^T \cdot Q \cdot x
$$

$$
A^T \cdot y - Q \cdot x + s = c
$$

where $x \geq 0, s \geq 0$. Note that, unlike the case of linear programming, the variables of the primal quadratic programming problem also appear in the dual QP.
As in linear programming, we can develop a dual of quadratic programming problems. 

The dual of \( f(x) \) is given below:

\[
\begin{align*}
\max_{x,y,s} \quad & b^T \cdot y - \frac{1}{2} \cdot x^T \cdot Q \cdot x \\
A^T \cdot y - Q \cdot x + s &= c \\
x \geq 0, \quad s \geq 0
\end{align*}
\]
As in linear programming, we can develop a dual of quadratic programming problems.

The dual of $f(x)$ is given below:

$$\max_{x, y, s} \ b^T \cdot y - \frac{1}{2} \cdot x^T \cdot Q \cdot x$$

$$A^T \cdot y - Q \cdot x + s = c$$

$$x \geq 0, s \geq 0$$

Note that, unlike the case of linear programming,
As in linear programming, we can develop a dual of quadratic programming problems. The dual of $f(x)$ is given below:

\[
\max_{x, y, s} \quad b^T \cdot y - \frac{1}{2} \cdot x^T \cdot Q \cdot x \\
A^T \cdot y - Q \cdot x + s = c \\
x \geq 0, \quad s \geq 0
\]

Note that, unlike the case of linear programming, the variables of the primal quadratic programming problem also appear in the dual QP.
Karush-Kuhn-Tucker (KKT) Optimality Conditions

Suppose that $x$ is a local optimal solution of the QP such that it satisfies $A \cdot x = b$, $x \geq 0$, and assume that $Q$ is a positive semidefinite matrix. Then, there exist vectors $y$ and $s$ such that the following conditions hold:

$$A^T \cdot y - Q \cdot x + s = c$$

$$s \geq 0$$

$$x_i \cdot s_i = 0, \forall i$$

Furthermore, $x$ is a global optimal solution.
Motivating Examples
The Quadratic Programming Problem
Optimality Conditions
Interior-Point Methods
Examples and QP Software
References

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Karush-Kuhn-Tucker Theorem (as applied to the QP problem)

- Suppose that \( x \) is a local optimal solution of the QP such that it satisfies
  \[ A \cdot x = b, \ x \geq 0, \]
Karush-Kuhn-Tucker (KKT) Optimality Conditions

Karush-Kuhn-Tucker Theorem (as applied to the QP problem)

Suppose that \( x \) is a local optimal solution of the QP such that it satisfies
\[
A \cdot x = b, \quad x \geq 0,
\]
and assume that \( Q \) is a positive semidefinite matrix.
Karush-Kuhn-Tucker (KKT) Optimality Conditions

Karush-Kuhn-Tucker Theorem (as applied to the QP problem)

- Suppose that \( \mathbf{x} \) is a local optimal solution of the QP such that it satisfies \( \mathbf{A} \cdot \mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0} \), and assume that \( \mathbf{Q} \) is a positive semidefinite matrix.

- Then, there exist vectors \( \mathbf{y} \) and \( \mathbf{s} \) such that the following conditions hold:
Karush-Kuhn-Tucker (KKT) Optimality Conditions

Karush-Kuhn-Tucker Theorem (as applied to the QP problem)

1. Suppose that $x$ is a local optimal solution of the QP such that it satisfies $A \cdot x = b$, $x \geq 0$, and assume that $Q$ is a positive semidefinite matrix.

2. Then, there exist vectors $y$ and $s$ such that the following conditions hold:

$$A^T \cdot y - Q \cdot x + s = c$$
Karush-Kuhn-Tucker (KKT) Optimality Conditions

Karush-Kuhn-Tucker Theorem (as applied to the QP problem)

- Suppose that \( \mathbf{x} \) is a local optimal solution of the QP such that it satisfies \( \mathbf{A} \cdot \mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0 \), and assume that \( \mathbf{Q} \) is a positive semidefinite matrix.

- Then, there exist vectors \( \mathbf{y} \) and \( \mathbf{s} \) such that the following conditions hold:

\[
\mathbf{A}^T \cdot \mathbf{y} - \mathbf{Q} \cdot \mathbf{x} + \mathbf{s} = \mathbf{c}
\]

\[
\mathbf{s} \geq 0
\]
Karush-Kuhn-Tucker (KKT) Optimality Conditions

Karush-Kuhn-Tucker Theorem (as applied to the QP problem)

- Suppose that $x$ is a local optimal solution of the QP such that it satisfies $A \cdot x = b$, $x \geq 0$, and assume that $Q$ is a positive semidefinite matrix.
- Then, there exist vectors $y$ and $s$ such that the following conditions hold:
  
  $$
  A^T \cdot y - Q \cdot x + s = c \\
  s \geq 0 \\
  x_i \cdot s_i = 0, \forall i
  $$

Z. Donovan and M. Xu

Optimization Methods in Finance
Karush-Kuhn-Tucker (KKT) Optimality Conditions

**Karush-Kuhn-Tucker Theorem (as applied to the QP problem)**

- Suppose that \( \mathbf{x} \) is a local optimal solution of the QP such that it satisfies \( A \cdot \mathbf{x} = b, \mathbf{x} \geq 0 \), and assume that \( Q \) is a positive semidefinite matrix.

- Then, there exist vectors \( \mathbf{y} \) and \( \mathbf{s} \) such that the following conditions hold:

  \[
  A^T \cdot \mathbf{y} - Q \cdot \mathbf{x} + \mathbf{s} = c \\
  \mathbf{s} \geq 0 \\
  x_i \cdot s_i = 0, \forall i
  \]

- Furthermore, \( \mathbf{x} \) is a global optimal solution.
Karush-Kuhn-Tucker (KKT) Optimality Conditions

More about the KKT theorem

The positive semidefiniteness condition related to the Hessian of the Lagrangian function in the KKT theorem is automatically satisfied for convex quadratic programming problems, and therefore is not included in the theorem above.

If vectors \( x, y, \) and \( s \) satisfy conditions of the KKT theorem as well as the primal feasibility conditions

\[
A \cdot x = b \quad x \geq 0
\]

then \( x \) is a global optimal solution.

In other words, all 5 conditions are both necessary and sufficient for \( x, y, \) and \( s \) to describe a global optimal solution of the QP problem.
Karush-Kuhn-Tucker (KKT) Optimality Conditions

More about the KKT theorem
Karush-Kuhn-Tucker (KKT) Optimality Conditions

More about the KKT theorem

- The positive semidefiniteness condition related to the Hessian of the Lagrangian function in the KKT theorem
Karush-Kuhn-Tucker (KKT) Optimality Conditions

More about the KKT theorem

- The positive semidefiniteness condition related to the Hessian of the Lagrangian function in the KKT theorem is automatically satisfied for convex quadratic programming problems,
Karush-Kuhn-Tucker (KKT) Optimality Conditions

More about the KKT theorem

- The positive semidefiniteness condition related to the Hessian of the Lagrangian function in the KKT theorem is automatically satisfied for convex quadratic programming problems, and therefore is not included in the theorem above.
The positive semidefiniteness condition related to the Hessian of the Lagrangian function in the KKT theorem is automatically satisfied for convex quadratic programming problems, and therefore is not included in the theorem above.

If vectors $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{s}$ satisfy conditions of the KKT theorem as well.
More about the KKT theorem

- The positive semidefiniteness condition related to the Hessian of the Lagrangian function in the KKT theorem is automatically satisfied for convex quadratic programming problems, and therefore is not included in the theorem above.
- If vectors $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{s}$ satisfy conditions of the KKT theorem as well as the primal feasibility conditions.
More about the KKT theorem

- The positive semidefiniteness condition related to the Hessian of the Lagrangian function in the KKT theorem is automatically satisfied for convex quadratic programming problems, and therefore is not included in the theorem above.

- If vectors \( x, y, \) and \( s \) satisfy conditions of the KKT theorem as well as the primal feasibility conditions

\[
A \cdot x = b
\]
Karush-Kuhn-Tucker (KKT) Optimality Conditions

More about the KKT theorem

- The positive semidefiniteness condition related to the Hessian of the Lagrangian function in the KKT theorem is automatically satisfied for convex quadratic programming problems, and therefore is not included in the theorem above.

- If vectors \( x, y, \) and \( s \) satisfy conditions of the KKT theorem as well as the primal feasibility conditions

\[
A \cdot x = b \\
x \geq 0
\]
Karush-Kuhn-Tucker (KKT) Optimality Conditions

More about the KKT theorem

- The positive semidefiniteness condition related to the Hessian of the Lagrangian function in the KKT theorem is automatically satisfied for convex quadratic programming problems, and therefore is not included in the theorem above.

- If vectors $x$, $y$, and $s$ satisfy conditions of the KKT theorem as well as the primal feasibility conditions

$$A \cdot x = b$$
$$x \geq 0$$

then $x$ is a global optimal solution.
Karush-Kuhn-Tucker (KKT) Optimality Conditions

More about the KKT theorem

- The positive semidefiniteness condition related to the Hessian of the Lagrangian function in the KKT theorem is automatically satisfied for convex quadratic programming problems, and therefore is not included in the theorem above.

- If vectors $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{s}$ satisfy conditions of the KKT theorem as well as the primal feasibility conditions

$$A \cdot \mathbf{x} = \mathbf{b}$$

$$\mathbf{x} \geq 0$$

then $\mathbf{x}$ is a global optimal solution.

- In other words,
Karush-Kuhn-Tucker (KKT) Optimality Conditions

More about the KKT theorem

- The positive semidefiniteness condition related to the Hessian of the Lagrangian function in the KKT theorem is automatically satisfied for convex quadratic programming problems, and therefore is not included in the theorem above.

- If vectors \( \mathbf{x}, \mathbf{y}, \) and \( \mathbf{s} \) satisfy conditions of the KKT theorem as well as the primal feasibility conditions

\[
A \cdot \mathbf{x} = \mathbf{b} \\
\mathbf{x} \geq 0
\]

then \( \mathbf{x} \) is a global optimal solution.

- In other words, all 5 conditions are both necessary and sufficient for \( \mathbf{x}, \mathbf{y}, \) and \( \mathbf{s} \) to describe a global optimal solution of the QP problem.
Karush-Kuhn-Tucker (KKT) Optimality Conditions

In a manner similar to linear programming, the optimality conditions can be seen as a collection of conditions for:

1. **Primal feasibility:**
   \[ A \cdot x = b, \quad x \geq 0 \]

2. **Dual feasibility:**
   \[ A^T \cdot y - Q \cdot x + s = c, \quad s \geq 0 \]

3. **Complementary slackness:** for each \( i = 1, \ldots, n \) we have
   \[ x_i \cdot s_i = 0 \]
In a manner similar to linear programming, the optimality conditions can be seen as a collection of conditions for:

1. **Primal feasibility:**
   \[ A \cdot x = b, \quad x \geq 0 \]

2. **Dual feasibility:**
   \[ A^T \cdot y - Q \cdot x + s = c, \quad s \geq 0 \]

3. **Complementary slackness:** For each \( i = 1, \ldots, n \), we have
   \[ x_i \cdot s_i = 0 \]

Karush-Kuhn-Tucker (KKT) Optimality Conditions

**KKT conditions**
Karush-Kuhn-Tucker (KKT) Optimality Conditions

KKT conditions

- In a manner similar to linear programming,
Karush-Kuhn-Tucker (KKT) Optimality Conditions

KKT conditions

In a manner similar to linear programming, the optimality conditions can be seen as a collection of conditions for:
In a manner similar to linear programming, the optimality conditions can be seen as a collection of conditions for:

- **primal feasibility**: \( A \cdot x = b, \ x \geq 0 \);
In a manner similar to linear programming, the optimality conditions can be seen as a collection of conditions for:

1. **Primal feasibility:** \( A \cdot x = b, \ x \geq 0 \);
2. **Dual feasibility:** \( A^T \cdot y - Q \cdot x + s = c, \ s \geq 0 \);
Karush-Kuhn-Tucker (KKT) Optimality Conditions

In a manner similar to linear programming, the optimality conditions can be seen as a collection of conditions for:

1. **Primal feasibility**: \( A \cdot x = b, \ x \geq 0 \);

2. **Dual feasibility**: \( A^T \cdot y - Q \cdot x + s = c, \ s \geq 0 \);

3. **Complementary slackness**: for each \( i = 1, \ldots, n \) we have \( x_i \cdot s_i = 0 \).
Exercise 3

Consider the following quadratic program

\[
\begin{align*}
&\text{min} & & 2x_1^2 + x_2^2 + 4x_3^2 + x_1 - 2x_2 + 3x_3 = 6 \\
& & & 2x_1 - 2x_2 + 3x_3 = 12 \\
& & & x_1, x_2, x_3 \geq 0
\end{align*}
\]
Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3
Exercise 3

Consider the following quadratic program:

\[
\begin{align*}
\min & \quad 2x_1^2 + x_2^2 + 4x_3^2 + x_1 + 2x_2 - x_3 = 6 \\
& -2x_1 - 2x_2 + 3x_3 = 12 \\
& x_1, x_2, x_3 \geq 0
\end{align*}
\]
Exercise 3

Consider the following quadratic program

\[ \min \quad 2 \cdot x_1^2 + x_2^2 + 4 \cdot x_3^2 \]
Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

Consider the following quadratic program

\[
\begin{align*}
\text{min} & \quad 2 \cdot x_1^2 + x_2^2 + 4 \cdot x_3^2 \\
\text{s.t.} & \quad x_1 + 2 \cdot x_2 - x_3 = 6
\end{align*}
\]
Exercise 3

Consider the following quadratic program

\[
\begin{align*}
\text{min} & \quad 2 \cdot x_1^2 + x_2^2 + 4 \cdot x_3^2 \\
& \quad x_1 + 2 \cdot x_2 - x_3 = 6 \\
& \quad 2 \cdot x_1 - 2 \cdot x_2 + 3 \cdot x_3 = 12
\end{align*}
\]
Consider the following quadratic program

\[
\begin{align*}
\text{min} & \quad 2 \cdot x_1^2 + x_2^2 + 4 \cdot x_3^2 \\
& x_1 + 2 \cdot x_2 - x_3 = 6 \\
& 2 \cdot x_1 - 2 \cdot x_2 + 3 \cdot x_3 = 12 \\
& x_1, x_2, x_3 \geq 0
\end{align*}
\]
Is the quadratic function convex? Yes.

\[ Q = \begin{bmatrix}
4 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 8
\end{bmatrix} \]

which is a positive semidefinite matrix.

Set up the KKT conditions for the optimal solution in matrix form, and show how you would solve for \( x \) and \( y \).
Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

Is the quadratic function convex? Yes.

\[ Q = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 8 \end{bmatrix} \]

which is a positive semidefinite matrix.

Set up the KKT conditions for the optimal solution in matrix form, and show how you would solve for \( x \) and \( y \).
Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

Is the quadratic function convex?
Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

- Is the quadratic function convex? Yes.

\[ Q = \begin{bmatrix}
4 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 8
\end{bmatrix} \]

which is a positive semidefinite matrix.

Set up the KKT conditions for the optimal solution in matrix form, and show how you would solve for \( x \) and \( y \).
Exercise 3

- Is the quadratic function convex? Yes.

\[ Q = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 8 \end{bmatrix} \]
Is the quadratic function convex? Yes.

\[ Q = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 8 \end{bmatrix} \] which is a positive semidefinite matrix.
Is the quadratic function convex? Yes.

\[ Q = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 8 \end{bmatrix} \] which is a positive semidefinite matrix.

Set up the KKT conditions for the optimal solution in matrix form,
Is the quadratic function convex? Yes.

\[
Q = \begin{bmatrix}
4 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 8
\end{bmatrix}
\]

which is a positive semidefinite matrix.

Set up the KKT conditions for the optimal solution in matrix form, and show how you would solve for \( x \) and \( y \).
Karush-Kuhn-Tucker (KKT) Optimality Conditions

Our KKT conditions for the optimal solution in matrix form is as follows:

\[
\begin{bmatrix}
Q A^T A & 0 \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
x \\
y
\end{bmatrix}
+ 
\begin{bmatrix}
s \\
0
\end{bmatrix}
= 
\begin{bmatrix}
c \\
b
\end{bmatrix}
\]

Where

\[
Q = \begin{bmatrix}
4 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 8
\end{bmatrix},
A = \begin{bmatrix}
1 & 2 \\
-1 & 2 \\
-2 & 3
\end{bmatrix},
c = \begin{bmatrix}
0 \\
0 \\
0
\end{bmatrix},
b = \begin{bmatrix}
6 \\
12
\end{bmatrix}
\]

Z. Donovan and M. Xu
Optimization Methods in Finance
Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

Our KKT conditions for the optimal solution in matrix form is as follows:

\[
\begin{bmatrix}
-QA^T & 0 \\
A & \mathbf{0}
\end{bmatrix}
\begin{bmatrix}
\mathbf{x} \\
\mathbf{y}
\end{bmatrix}
+ \begin{bmatrix}
\mathbf{s} \\
\mathbf{0}
\end{bmatrix} = \begin{bmatrix}
c \\
b
\end{bmatrix}
\]

Where

\[
Q = \begin{bmatrix}
4 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 8
\end{bmatrix},
A = \begin{bmatrix}
1 & 2 \\
-1 & -2 \\
2 & 3
\end{bmatrix},
c = \begin{bmatrix}
0 \\
0 \\
0
\end{bmatrix},
b = \begin{bmatrix}
6 \\
12
\end{bmatrix}
\]
Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

Our KKT conditions for the optimal solution in matrix form is as follows:

\[
\begin{bmatrix}
-\mathbf{Q} \mathbf{A}^T \\
\mathbf{0}
\end{bmatrix}
\begin{bmatrix}
\mathbf{x} \\
\mathbf{y}
\end{bmatrix}
+
\begin{bmatrix}
\mathbf{s} \\
\mathbf{0}
\end{bmatrix}
=
\begin{bmatrix}
\mathbf{c} \\
\mathbf{b}
\end{bmatrix}
\]

Where

\[
\mathbf{Q} = \begin{bmatrix}
4 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 8
\end{bmatrix},
\]

\[
\mathbf{A} = \begin{bmatrix}
1 & 2 & -1 \\
2 & -2 & 3
\end{bmatrix},
\]

\[
\mathbf{c} = \begin{bmatrix}
0 \\
0 \\
0
\end{bmatrix},
\]

\[
\mathbf{b} = \begin{bmatrix}
6 \\
12
\end{bmatrix}
\]
Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

Our KKT conditions for the optimal solution in matrix form is as follows:

\[
\begin{bmatrix}
-Q & A^T \\
A & 0
\end{bmatrix}
\]
Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

Our KKT conditions for the optimal solution in matrix form is as follows:

\[
\begin{bmatrix}
-Q & A^T \\
A & 0
\end{bmatrix}
\begin{bmatrix}
x \\
y
\end{bmatrix}
\]
Exercise 3

Our KKT conditions for the optimal solution in matrix form is as follows:

\[
\begin{bmatrix}
-Q & A^T \\
A & 0
\end{bmatrix}
\begin{bmatrix}
x \\
y
\end{bmatrix}
+
\begin{bmatrix}
s \\
0
\end{bmatrix}
\]
Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

- Our KKT conditions for the optimal solution in matrix form is as follows:

$$
\begin{bmatrix}
-\mathbf{Q} & \mathbf{A}^T \\
\mathbf{A} & 0
\end{bmatrix}
\begin{bmatrix}
x \\
y
\end{bmatrix}
+ 
\begin{bmatrix}
\mathbf{s} \\
0
\end{bmatrix}
= 
\begin{bmatrix}
\mathbf{c} \\
\mathbf{b}
\end{bmatrix}
$$

Where

$$
\mathbf{Q} = \begin{bmatrix}
4 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 8
\end{bmatrix},
\mathbf{A} = \begin{bmatrix}
1 & 2 & -1 \\
2 & -2 & 3
\end{bmatrix},
\mathbf{c} = \begin{bmatrix}
0 \\
0 \\
0
\end{bmatrix},
\mathbf{b} = \begin{bmatrix}
6 \\
12
\end{bmatrix}
$$
Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

Our KKT conditions for the optimal solution in matrix form is as follows:

\[
\begin{bmatrix}
-Q & A^T \\
A & 0
\end{bmatrix}
\begin{bmatrix}
x \\
y
\end{bmatrix}
+
\begin{bmatrix}
s \\
0
\end{bmatrix}
=
\begin{bmatrix}
c \\
b
\end{bmatrix}
\]
Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

Our KKT conditions for the optimal solution in matrix form is as follows:

\[
\begin{bmatrix}
-Q & A^T \\
A & 0
\end{bmatrix}
\begin{bmatrix}
x \\
y
\end{bmatrix}
+
\begin{bmatrix}
s \\
0
\end{bmatrix}
=
\begin{bmatrix}
c \\
b
\end{bmatrix}
\]

Where
Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

Our KKT conditions for the optimal solution in matrix form is as follows:

\[
\begin{bmatrix}
-Q & A^T \\
A & 0
\end{bmatrix}
\begin{bmatrix}
x \\
y
\end{bmatrix}
+
\begin{bmatrix}
s \\
0
\end{bmatrix}
=
\begin{bmatrix}
c \\
b
\end{bmatrix}
\]

Where

\[
Q = \begin{bmatrix}
4 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 8
\end{bmatrix},
\]

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

Our KKT conditions for the optimal solution in matrix form is as follows:

\[
\begin{bmatrix}
-Q & A^T \\
A & 0
\end{bmatrix}
\begin{bmatrix}
x \\
y
\end{bmatrix}
+
\begin{bmatrix}
s \\
0
\end{bmatrix}
=
\begin{bmatrix}
c \\
b
\end{bmatrix}
\]

Where

\[
Q = \begin{bmatrix}
4 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 8
\end{bmatrix},
\]

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

Our KKT conditions for the optimal solution in matrix form is as follows:

\[
\begin{bmatrix}
-Q & A^T \\
A & 0
\end{bmatrix}
\begin{bmatrix}
x \\
y
\end{bmatrix}
+
\begin{bmatrix}
s \\
0
\end{bmatrix}
=
\begin{bmatrix}
c \\
b
\end{bmatrix}
\]

Where

\[
Q = \begin{bmatrix}
4 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 8
\end{bmatrix},
\]
Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

Our KKT conditions for the optimal solution in matrix form is as follows:

\[
\begin{bmatrix}
-Q & A^T \\
A & 0
\end{bmatrix}
\begin{bmatrix}
x \\
y
\end{bmatrix}
+
\begin{bmatrix}
s \\
o
\end{bmatrix}
=
\begin{bmatrix}
c \\
b
\end{bmatrix}
\]

Where

\[
Q = \begin{bmatrix}
4 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 8
\end{bmatrix},
A = \begin{bmatrix}
1 & 2 & -1 \\
2 & -2 & 3
\end{bmatrix},
\]
Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

Our KKT conditions for the optimal solution in matrix form is as follows:

\[
\begin{bmatrix}
-Q & A^T \\
A & 0
\end{bmatrix}
\begin{bmatrix}
x \\
y
\end{bmatrix}
+\begin{bmatrix}
s \\
0
\end{bmatrix}
=\begin{bmatrix}
c \\
b
\end{bmatrix}
\]

Where

\[
Q = \begin{bmatrix}
4 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 8
\end{bmatrix}, \quad A = \begin{bmatrix}
1 & 2 & -1 \\
2 & -2 & 3
\end{bmatrix}, \quad c = \begin{bmatrix}
0 \\
0
\end{bmatrix}
\]
Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

Our KKT conditions for the optimal solution in matrix form is as follows:

\[
\begin{bmatrix}
-Q & A^T \\
A & 0
\end{bmatrix}
\begin{bmatrix}
x \\
y
\end{bmatrix}
+ 
\begin{bmatrix}
s \\
0
\end{bmatrix}
=
\begin{bmatrix}
c \\
b
\end{bmatrix}
\]

Where

\[
Q = \begin{bmatrix}
4 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 8
\end{bmatrix},
A = \begin{bmatrix}
1 & 2 & -1 \\
2 & -2 & 3
\end{bmatrix},
c = \begin{bmatrix}
0 \\
0
\end{bmatrix},
b = \begin{bmatrix}
6 \\
12
\end{bmatrix}
\]
Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

We get the following linear system:

\[
\begin{bmatrix}
-4 & 0 & 0 & 1 & 2 \\
0 & -2 & 0 & 2 & -2 \\
0 & 0 & -8 & 1 & 3 \\
1 & 2 & -1 & 0 & 0 \\
2 & -2 & 3 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\begin{bmatrix}
x_1 \\ x_2 \\ x_3 \\ y_1 \\ y_2
\end{bmatrix}
\end{bmatrix}
\begin{bmatrix}
0 \\ 0 \\ 0 \\ 0 \\ 0
\end{bmatrix}
= 
\begin{bmatrix}
0 \\ 0 \\ 0 \\ 6 \\ 12
\end{bmatrix}
\]

After solving the system, we find that

\(
x = (5.045, 1.194, 1.433)
\)

is an optimal solution with

\(
y = (7.522, 6.328)
\)

and

\(
s = (0, 0, 0)
\)
The Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

We get the following linear system:

\[
\begin{bmatrix}
-4 & 0 & 0 & 1 & 2 \\
0 & -2 & 0 & 2 \\
-2 & -8 & -1 & 3 \\
1 & 2 & -1 & 0 & 0 \\
2 & -2 & 3 & 0 & 0 \\
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
y_1 \\
y_2 \\
\end{bmatrix}
+
\begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
0 \\
\end{bmatrix}
=
\begin{bmatrix}
0 \\
0 \\
0 \\
6 \\
12 \\
\end{bmatrix}
\]

After solving the system, we find that

\( x = (5.045, 1.194, 1.433) \)

is an optimal solution with

\( y = (7.522, 6.328) \)

and

\( s = (0, 0, 0) \).
Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

- We get the following linear system:
Exercise 3

We get the following linear system:

\[
\begin{bmatrix}
-4 & 0 & 0 & 1 & 2 \\
0 & -2 & 0 & 2 & -2 \\
0 & 0 & -8 & -1 & 3 \\
1 & 2 & -1 & 0 & 0 \\
2 & -2 & 3 & 0 & 0
\end{bmatrix}
\]
Exercise 3

We get the following linear system:

\[
\begin{bmatrix}
-4 & 0 & 0 & 1 & 2 \\
0 & -2 & 0 & 2 & -2 \\
0 & 0 & -8 & -1 & 3 \\
1 & 2 & -1 & 0 & 0 \\
2 & -2 & 3 & 0 & 0 \\
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
y_1 \\
y_2 \\
\end{bmatrix}
= \begin{bmatrix}
0 \\
0 \\
6 \\
12 \\
0
\end{bmatrix}
\]
Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

We get the following linear system:

\[
\begin{bmatrix}
-4 & 0 & 0 & 1 & 2 \\
0 & -2 & 0 & 2 & -2 \\
0 & 0 & -8 & -1 & 3 \\
1 & 2 & -1 & 0 & 0 \\
2 & -2 & 3 & 0 & 0 \\
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
y_1 \\
y_2 \\
\end{bmatrix}
+ \begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
0 \\
\end{bmatrix}
= \begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
0 \\
\end{bmatrix}
\]
Exercise 3

We get the following linear system:

\[
\begin{bmatrix}
-4 & 0 & 0 & 1 & 2 \\
0 & -2 & 0 & 2 & -2 \\
0 & 0 & -8 & -1 & 3 \\
1 & 2 & -1 & 0 & 0 \\
2 & -2 & 3 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
y_1 \\
y_2
\end{bmatrix}
+ \begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
0
\end{bmatrix}
= \begin{bmatrix}
0 \\
0 \\
0 \\
6 \\
12
\end{bmatrix}
\]
Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

We get the following linear system:

\[
\begin{bmatrix}
-4 & 0 & 0 & 1 & 2 \\
0 & -2 & 0 & 2 & -2 \\
0 & 0 & -8 & -1 & 3 \\
1 & 2 & -1 & 0 & 0 \\
2 & -2 & 3 & 0 & 0 \\
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
y_1 \\
y_2 \\
\end{bmatrix}
+ 
\begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
0 \\
\end{bmatrix}
= 
\begin{bmatrix}
0 \\
0 \\
0 \\
6 \\
12 \\
\end{bmatrix}
\]

After solving the system,
Exercise 3

We get the following linear system:

\[
\begin{bmatrix}
-4 & 0 & 0 & 1 & 2 \\
0 & -2 & 0 & 2 & -2 \\
0 & 0 & -8 & -1 & 3 \\
1 & 2 & -1 & 0 & 0 \\
2 & -2 & 3 & 0 & 0 \\
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
y_1 \\
y_2 \\
\end{bmatrix}
+
\begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
0 \\
\end{bmatrix}
=
\begin{bmatrix}
0 \\
0 \\
0 \\
6 \\
12 \\
\end{bmatrix}
\]

After solving the system, we find that \( \mathbf{x} = (5.045, 1.194, 1.433) \) is an optimal solution.
Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

- We get the following linear system:

\[
\begin{bmatrix}
-4 & 0 & 0 & 1 & 2 \\
0 & -2 & 0 & 2 & -2 \\
0 & 0 & -8 & -1 & 3 \\
1 & 2 & -1 & 0 & 0 \\
2 & -2 & 3 & 0 & 0 \\
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
y_1 \\
y_2 \\
\end{bmatrix}
+
\begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
0 \\
\end{bmatrix}
=
\begin{bmatrix}
0 \\
0 \\
0 \\
6 \\
12 \\
\end{bmatrix}
\]

- After solving the system, we find that \( \mathbf{x} = (5.045, 1.194, 1.433) \) is an optimal solution with \( \mathbf{y} = (7.522, 6.328) \) and \( \mathbf{s} = (0, 0, 0) \).
Interior-Point Methods

Interior-Point Method (IPM) finds primal-dual solutions \((x, y, s)\) by applying variants of Newton's method to the optimality conditions and modifying the search directions and step lengths so that \(x \geq 0\) and \(s \geq 0\) are satisfied strictly at every iteration.

Rewrite the Optimality Conditions

\[
F(x, y, s) = \begin{bmatrix}
A^T y - Q x + s - c \\
A x - b
\end{bmatrix} = \begin{bmatrix}
0 \\
0
\end{bmatrix},
\]

\(X\) and \(S\) are diagonal matrices such that \(X_{ii} = x_i\) and \(X_{ij} = 0, i \neq j\), and similarly for \(S\).
Interior-Point Methods

Interior-Point Method

The Quadratic Programming Problem
Optimality Conditions
Interior-Point Methods
Examples and QP Software
References

**Interior-Point Method**

Interior-Point Method (IPM) finds primal-dual solutions \((x, y, s)\) by applying variants of Newton's method to the optimality conditions and modifying the search directions and step lengths so that \(x \geq 0\) and \(s \geq 0\) are satisfied strictly at every iteration.

Rewrite the Optimality Conditions

\[
F(x, y, s) = \begin{bmatrix}
A^T y - Qx + s - c \\
Ax - b
\end{bmatrix} = \begin{bmatrix}
0 \\
0
\end{bmatrix},
\]

\((x, s) \geq 0\)

where \(X\) and \(S\) are diagonal matrices such that \(X_{ii} = x_i\) and \(X_{ij} = 0, i \neq j\), and similarly for \(S\).
Interior-Point Method

*Interior-Point Method* (IPM) finds primal-dual solutions \((x, y, s)\) by applying variants of Newton’s method to the optimality conditions.
Interior-Point Method

*Interior-Point Method*(IPM) finds primal-dual solutions \((x, y, s)\) by applying variants of Newton’s method to the optimality conditions and modifying the search directions and step lengths so that \(x \geq 0\) and \(s \geq 0\) are satisfied strictly at every iteration.
Interior-Point Method

*Interior-Point Method* (IPM) finds primal-dual solutions \((x, y, s)\) by applying variants of Newton’s method to the optimality conditions and modifying the search directions and step lengths so that \(x \geq 0\) and \(s \geq 0\) are satisfied strictly at every iteration.

Rewrite the Optimality Conditions
Interior-Point Methods

Interior-Point Method

*Interior-Point Method* (IPM) finds primal-dual solutions \((x, y, s)\) by applying variants of Newton’s method to the optimality conditions and modifying the search directions and step lengths so that \(x \geq 0\) and \(s \geq 0\) are satisfied strictly at every iteration.

Rewrite the Optimality Conditions

\[
F(x, y, s) = \begin{bmatrix} \quad A^T y - Q x + s - c \quad \\ - A x - b \\ X S e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad (x, s) \geq 0
\]
Interior-Point Method

*Interior-Point Method* (IPM) finds primal-dual solutions \((x, y, s)\) by applying variants of Newton’s method to the optimality conditions and modifying the search directions and step lengths so that \(x \geq 0\) and \(s \geq 0\) are satisfied strictly at every iteration.

Rewrite the Optimality Conditions

\[
F(x, y, s) = \begin{bmatrix}
A^T y - Q x + s - c \\
A x - b \\
X S e
\end{bmatrix} = \begin{bmatrix}
0 \\
0 \\
0
\end{bmatrix}, \quad (x, s) \geq 0
\]

\(X\) and \(S\) are diagonal matrices such that \(X_{ii} = x_i\) and \(X_{ij} = 0, i \neq j\), and similarly for \(S\).
Remark

$x_i \cdot s_i$ is a nonlinear constraint, so we cannot solve this system using linear system solution methods such as Gaussian elimination. Since the system is square we can apply Newton's method. The existence of nonnegative constraints creates a difficulty, otherwise we can use Newton's method directly. Since we are generating iterates for both the primal and dual problems, this version of IPMs are often called primal-dual interior-point methods.

Strategy of Applying A Modified Newton's Method

1. Identify an initial solution $(x_0, y_0, s_0)$, which satisfies the first two constraints (linear) and $(x_0, s_0) > 0$, but not the third one.

2. Generate new points $(x_k, y_k, s_k)$ that also satisfy these same conditions and get progressively closer to satisfying the third constraint.
Motivating Examples
The Quadratic Programming Problem
Optimality Conditions
Interior-Point Methods
Examples and QP Software
References

Strategy

Remark

1. Identify an initial solution $(x_0, y_0, s_0)$, which satisfies the first two constraints (linear) and $(x_0, s_0) > 0$, but not the third one.
2. Generate new points $(x_k, y_k, s_k)$ that also satisfy these same conditions and get progressively closer to satisfying the third constraint.

Z. Donovan and M. Xu
Optimization Methods in Finance
Strategy

Remark

- $x_i \cdot s_i$ is a nonlinear constraint, so we cannot solve this system using linear system solution methods such as Gaussian elimination.
**Strategy**

**Remark**

- $x_i \cdot s_i$ is a nonlinear constraint, so we cannot solve this system using linear system solution methods such as Gaussian elimination.
- Since the system is square we can apply Newton's method.
Strategy

Remark

- $x_i \cdot s_i$ is a nonlinear constraint, so we cannot solve this system using linear system solution methods such as Gaussian elimination.
- Since the system is square we can apply Newton's method.
- The existence of nonnegative constraints creates a difficulty, otherwise we can use Newton's method directly.
Motivating Examples
The Quadratic Programming Problem
Optimality Conditions
Interior-Point Methods
Examples and QP Software
References

Strategy

Remark

- $x_i \cdot s_i$ is a nonlinear constraint, so we cannot solve this system using linear system solution methods such as Gaussian elimination.
- Since the system is square we can apply Newton’s method.
- The existence of nonnegative constraints creates a difficulty, otherwise we can use Newton’s method directly.
- Since we are generating iterates for both the primal and dual problems, this version of IPMs are often called *primal-dual interior-point methods*.  

Z. Donovan and M. Xu
Optimization Methods in Finance
**Strategy**

**Remark**

- $x_i \cdot s_i$ is a nonlinear constraint, so we cannot solve this system using linear system solution methods such as Gaussian elimination.

- Since the system is square we can apply Newton’s method.

- The existence of nonnegative constraints creates a difficulty, otherwise we can use Newton’s method directly.

- Since we are generating iterates for both the primal and dual problems, this version of IPMs are often called *primal-dual interior-point methods*.

**Strategy of Applying A Modified Newton’s Method**
Stratety

Remark

- $x_i \cdot s_i$ is a nonlinear constraint, so we cannot solve this system using linear system solution methods such as Gaussian elimination.
- Since the system is square we can apply Newton's method.
- The existence of nonnegative constraints creates a difficulty, otherwise we can use Newton’s method directly.
- Since we are generating iterates for both the primal and dual problems, this version of IPMs are often called *primal-dual interior-point methods*.

Strategy of Applying A Modified Newton’s Method

1. Identify an initial solution $(x^0, y^0, s^0)$, which satisfies the first two constraints (linear) and $(x^0, s^0) > 0$, but not the third one.
**Strategy**

**Remark**

- $x_i \cdot s_i$ is a nonlinear constraint, so we cannot solve this system using linear system solution methods such as Gaussian elimination.
- Since the system is square we can apply Newton’s method.
- The existence of nonnegative constraints creates a difficulty, otherwise we can use Newton’s method directly.
- Since we are generating iterates for both the primal and dual problems, this version of IPMs are often called *primal-dual interior-point methods*.

**Strategy of Applying A Modified Newton’s Method**

1. Identify an initial solution $(x^0, y^0, s^0)$, which satisfies the first two constraints (linear) and $(x^0, s^0) > 0$, but not the third one.
2. Generate new points $(x^k, y^k, s^k)$ that also satisfy these same conditions and get progressively closer to satisfying the third constraint.
Algorithms for IPMs with pure Newton direction
Algorithms for IPMs with pure Newton direction

Definition

Feasible set: $F := \{(x, y, s) : A \cdot x = b, A^T \cdot y - Q \cdot x + s = c, x \geq 0, s \geq 0\}$

$(x, y, s) \in F$ is a feasible point.

Strictly feasible set: $F_0 := \{(x, y, s) : A \cdot x = b, A^T \cdot y - Q \cdot x + s = c, x > 0, s > 0\}$

$(x, y, s) \in F_0$ is a strictly feasible solution, which lies in the interior of the region defined by those constraints rather than being on the boundary. So $F_0$ is the relative interior of the set $F$. 

Z. Donovan and M. Xu
Optimization Methods in Finance
Algorithms for IPMs with pure Newton direction

Definition

- Feasible set: $\mathcal{F} := \{(x, y, s) : A \cdot x = b, A^T \cdot y - Q \cdot x + s = c, x \geq 0, s \geq 0\}$
Definition

Feasible set: \( \mathcal{F} := \{(x, y, s) : A \cdot x = b, A^T \cdot y - Q \cdot x + s = c, x \geq 0, s \geq 0\} \)

\((x, y, s) \in \mathcal{F}\) is a feasible point.
Algorithms for IPMs with pure Newton direction

Definition

- Feasible set: $\mathcal{F} := \{(x, y, s) : A \cdot x = b, A^T \cdot y - Q \cdot x + s = c, x \geq 0, s \geq 0\}$

  $(x, y, s) \in \mathcal{F}$ is a feasible point.

- Strictly feasible set:

  $\mathcal{F}^0 := \{(x, y, s) : A \cdot x = b, A^T \cdot y - Q \cdot x + s = c, x > 0, s > 0\}$
Algorithms for IPMs with pure Newton direction

Definition

- Feasible set: $\mathcal{F} := \{(x, y, s) : A \cdot x = b, A^T \cdot y - Q \cdot x + s = c, x \geq 0, s \geq 0\}$
  
  $(x, y, s) \in \mathcal{F}$ is a feasible point.

- Strictly feasible set:
  $\mathcal{F}^0 := \{(x, y, s) : A \cdot x = b, A^T \cdot y - Q \cdot x + s = c, x > 0, s > 0\}$

  $(x, y, s) \in \mathcal{F}^0$ is a strictly feasible solution, which lies in the interior of the region defined by those constraints rather than being on the boundary.
Algorithms for IPMs with pure Newton direction

**Definition**

- Feasible set: \( \mathcal{F} := \{(x, y, s) : A \cdot x = b, A^T \cdot y - Q \cdot x + s = c, x \geq 0, s \geq 0\} \)
  
  \((x, y, s) \in \mathcal{F}\) is a feasible point.

- Strictly feasible set:
  
  \( \mathcal{F}^0 := \{(x, y, s) : A \cdot x = b, A^T \cdot y - Q \cdot x + s = c, x > 0, s > 0\} \)

  \((x, y, s) \in \mathcal{F}^0\) is a strictly feasible solution, which lies in the *interior* of the region defined by those constraints rather than being on the boundary. So \( \mathcal{F}^0 \) is the relative interior of the set \( \mathcal{F} \).
Exercise 4

Consider the quadratic programming problem given below:

\[
\begin{align*}
\text{min} & \quad x_1 \cdot x_2 + x_2^1 + 3 \cdot x_2^2 + 2 \cdot x_2^3 + 2 \cdot x_1 + x_2 + 3 \cdot x_3 \\
\text{subject to} & \quad x_1 + x_2 + x_3 = 1 \\
& \quad x_1 - x_2 = 0 \\
& \quad x_1, x_2, x_3 \geq 0
\end{align*}
\]

The current primal-dual estimate of the solution is \(x_k = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})^T\), \(y_k = (1, \frac{1}{2})^T\), and \(s_k = (\frac{3}{2}, \frac{11}{6}, \frac{10}{3})^T\).

Is \((x, y, s)\) \(\in F\)? How about \(F_0\)?

Z. Donovan and M. Xu

Optimization Methods in Finance
Exercise 4

Consider the quadratic programming problem given below:

\[
\begin{aligned}
\min & \quad x_1 \cdot x_2 + x_2^1 + 3 \cdot x_2^2 + 2 \cdot x_2^3 + 2 \cdot x_1 + x_2 + 3 \cdot x_3 \\
\text{s.t.} & \quad x_1 + x_2 + x_3 = 1 \\
& \quad x_1 - x_2 = 0 \\
& \quad x_1, x_2, x_3 \geq 0
\end{aligned}
\]

The current primal-dual estimate of the solution is \(x_k = (1, 1, 1)^T\), \(y_k = (1, 1, 2, 2)^T\), and \(s_k = (3, 2, 10, 3)^T\).

Is \((x, y, s)\) ∈ \(F\)?

How about \(F_0\)?

Z. Donovan and M. Xu

Optimization Methods in Finance
Exercise 4

Consider the quadratic programming problem given below:

\[
\begin{align*}
\min & \quad \frac{1}{2} x_1 \cdot x_2 + x_2^1 + 3 x_2^2 + 2 x_2^3 + 2 x_1 + x_2 + 3 x_3 \\
\text{s.t.} & \quad x_1 + x_2 + x_3 = 1 \\
& \quad x_1 - x_2 = 0 \\
& \quad x_1, x_2, x_3 \geq 0
\end{align*}
\]
Exercise 4

Consider the quadratic programming problem given below:

$$\min \quad x_1 \cdot x_2 + x_1^2 + \frac{3}{2} \cdot x_2^2 + 2 \cdot x_3^2$$
$$+ 2 \cdot x_1 + x_2 + 3 \cdot x_3$$
Exercise 4

Consider the quadratic programming problem given below:

\[
\begin{align*}
\min & \quad x_1 \cdot x_2 + x_1^2 + \frac{3}{2} \cdot x_2^2 + 2 \cdot x_3^2 \\
& + 2 \cdot x_1 + x_2 + 3 \cdot x_3 \\
& x_1 + x_2 + x_3 = 1
\end{align*}
\]
Exercise 4

Consider the quadratic programming problem given below:

\[
\begin{align*}
\min & \quad x_1 \cdot x_2 + x_1^2 + \frac{3}{2} \cdot x_2^2 + 2 \cdot x_3^2 \\
& \quad + 2 \cdot x_1 + x_2 + 3 \cdot x_3 \\
& \quad x_1 + x_2 + x_3 = 1 \\
& \quad x_1 - x_2 = 0
\end{align*}
\]
Exercise 4

Consider the quadratic programming problem given below:

\[
\begin{align*}
\text{min} & \quad x_1 \cdot x_2 + x_1^2 + \frac{3}{2} \cdot x_2^2 + 2 \cdot x_3^2 \\
& \quad + 2 \cdot x_1 + x_2 + 3 \cdot x_3 \\
& \quad x_1 + x_2 + x_3 = 1 \\
& \quad x_1 - x_2 = 0 \\
& \quad x_1, x_2, x_3 \geq 0
\end{align*}
\]
Exercise 4

Consider the quadratic programming problem given below:

\[
\begin{align*}
\min & \quad x_1 \cdot x_2 + x_1^2 + \frac{3}{2} \cdot x_2^2 + 2 \cdot x_3^2 \\
& \quad + 2 \cdot x_1 + x_2 + 3 \cdot x_3 \\
& \quad x_1 + x_2 + x_3 = 1 \\
& \quad x_1 - x_2 = 0 \\
& \quad x_1, x_2, x_3 \geq 0
\end{align*}
\]

The current primal-dual estimate of the solution \( x^k = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})^T \), \( y^k = (1, \frac{1}{2})^T \), and \( s^k = (\frac{3}{2}, \frac{11}{6}, \frac{10}{3})^T \).
Exercise 4

Consider the quadratic programming problem given below:

\[
\begin{align*}
\min & \quad x_1 \cdot x_2 + x_1^2 + \frac{3}{2} \cdot x_2^2 + 2 \cdot x_3^2 \\
& \quad + 2 \cdot x_1 + x_2 + 3 \cdot x_3 \\
& \quad x_1 + x_2 + x_3 = 1 \\
& \quad x_1 - x_2 = 0 \\
& \quad x_1, x_2, x_3 \geq 0
\end{align*}
\]

The current primal-dual estimate of the solution \(x^k = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)^T\), \(y^k = \left(1, \frac{1}{2}\right)^T\), and \(s^k = \left(\frac{3}{2}, \frac{11}{6}, \frac{10}{3}\right)^T\). Is \((x, y, s) \in \mathcal{F}\)?
Exercise 4

Consider the quadratic programming problem given below:

\[
\begin{align*}
\min & \quad x_1 \cdot x_2 + x_1^2 + \frac{3}{2} \cdot x_2^2 + 2 \cdot x_3^2 \\
& \quad + 2 \cdot x_1 + x_2 + 3 \cdot x_3 \\
& \quad x_1 + x_2 + x_3 = 1 \\
& \quad x_1 - x_2 = 0 \\
& \quad x_1, x_2, x_3 \geq 0
\end{align*}
\]

The current primal-dual estimate of the solution \( \mathbf{x}^k = \left( \frac{1}{3}, \frac{1}{3}, \frac{1}{3} \right)^T \), \( \mathbf{y}^k = \left( 1, \frac{1}{2} \right)^T \), and \( \mathbf{s}^k = \left( \frac{3}{2}, \frac{11}{6}, \frac{10}{3} \right)^T \). Is \( (x, y, s) \in \mathcal{F} \)? How about \( \mathcal{F}^0 \)?
Algorithms for IPMs with pure Newton direction
Two Basic Ingredients of IPMs
Algorithms for IPMs with pure Newton direction

Two Basic Ingredients of IPMs

1. A measure that can be used to evaluate and compare the quality of alternative solutions and search directions.
Two Basic Ingredients of IPMs

1. A measure that can be used to evaluate and compare the quality of alternative solutions and search directions.

2. A method to generate a better solution, with respect to the measure just mentioned, from a non-optimal solution.
Algorithms for IPMs with pure Newton direction
Algorithms for IPMs with pure Newton direction

Pure Newton Step

Assume that we have a current estimate \((x_k, y_k, s_k)\) of the optimal solution to the problem. The Newton step from this point is determined by solving the following system of linear equations:

\[
J(x_k, y_k, s_k) \begin{bmatrix} \Delta x_k \\ \Delta y_k \\ \Delta s_k \end{bmatrix} = -F(x_k, y_k, s_k),
\]

where \(J(x_k, y_k, s_k)\) is the Jacobian of the function \(F\) and \([\Delta x_k, \Delta y_k, \Delta s_k]^T\) is the search direction.
Pure Newton Step

Assume that we have a current estimate \((x^k, y^k, s^k)\) of the optimal solution to the problem.
Algorithms for IPMs with pure Newton direction

Pure Newton Step

Assume that we have a current estimate \((x^k, y^k, s^k)\) of the optimal solution to the problem.

The Newton step from this point is determined by solving the following system of linear equations:
Algorithms for IPMs with pure Newton direction

Pure Newton Step

Assume that we have a current estimate \((x^k, y^k, s^k)\) of the optimal solution to the problem.

The Newton step from this point is determined by solving the following system of linear equations:

\[
J(x^k, y^k, s^k) \begin{bmatrix} \Delta x^k \\ \Delta y^k \\ \Delta s^k \end{bmatrix} = -F(x^k, y^k, s^k),
\]
Motivating Examples
The Quadratic Programming Problem
Optimality Conditions
Interior-Point Methods
Examples and QP Software
References

Algorithms for IPMs with pure Newton direction

Pure Newton Step

Assume that we have a current estimate \((x^k, y^k, s^k)\) of the optimal solution to the problem.

The Newton step from this point is determined by solving the following system of linear equations:

\[
J(x^k, y^k, s^k) \begin{bmatrix}
\Delta x^k \\
\Delta y^k \\
\Delta s^k
\end{bmatrix} = -F(x^k, y^k, s^k),
\]

where \(J(x^k, y^k, s^k)\) is the Jacobian of the function \(F\) and \([\Delta x^k, \Delta y^k, \Delta s^k]^T\) is the search direction.
Algorithms for IPMs with pure Newton direction

First, we observe that

\[ J(x_k, y_k, s_k) = \begin{bmatrix} -Q & A^T & I \\ A & 0 & 0 \\ \kappa^T & s_k & X_k \end{bmatrix} \]

where, \( X_k \) and \( S_k \) are diagonal matrices with the components of the vectors \( x_k \) and \( s_k \) along their diagonals.

Furthermore, if \((x_k, y_k, s_k) \in F_0\), then

\[ F(x_k, y_k, s_k) = \begin{bmatrix} 0 & 0 & \kappa^T \end{bmatrix} \begin{bmatrix} S_k \\ X_k^T \end{bmatrix} e \]
Pure Newton Step

First, we observe that

\[ J(x_k, y_k, s_k) = \begin{bmatrix} -Q & A^T & I \\ 0 & A_0 & 0 \\ 0 & 0 & S_k \end{bmatrix} X_k \]

where, \( X_k \) and \( S_k \) are diagonal matrices with the components of the vectors \( x_k \) and \( s_k \) along their diagonals.

Furthermore, if \((x_k, y_k, s_k) \in F_0\), then

\[ F(x_k, y_k, s_k) = \begin{bmatrix} 0 & 0 & X_k S_k e \end{bmatrix} \]
First, we observe that

\[
J(x_k, y_k, s_k) = \begin{bmatrix} -Q & A^T & I \\ A & 0 & 0 \\ X & S_k & 0 \end{bmatrix}
\]

where, \(X_k\) and \(S_k\) are diagonal matrices with the components of the vectors \(x_k\) and \(s_k\) along their diagonals.

Furthermore, if \((x_k, y_k, s_k) \in F_0\), then

\[
F(x_k, y_k, s_k) = \begin{bmatrix} 0 & 0 & X_k S_k e \end{bmatrix}
\]
Algorithms for IPMs with pure Newton direction

Pure Newton Step

First, we observe that

\[ J(x^k, y^k, s^k) = \begin{bmatrix} -Q & A^T & I \\ A & 0 & 0 \\ S^k & 0 & X^k \end{bmatrix} \]
Algorithms for IPMs with pure Newton direction

Pure Newton Step

First, we observe that

$$J(x^k, y^k, s^k) = \begin{bmatrix} -Q & A^T & I \\ A & 0 & 0 \\ S^k & 0 & X^k \end{bmatrix}$$

where, $x^k$ and $s^k$ are diagonal matrices with the components of the vectors $x^k$ and $s^k$ along their diagonals.
Pure Newton Step

First, we observe that

\[
J(x^k, y^k, s^k) = \begin{bmatrix}
-Q & A^T & I \\
A & 0 & 0 \\
S^k & 0 & X^k \\
\end{bmatrix}
\]

where, \( X^k \) and \( S^k \) are diagonal matrices with the components of the vectors \( x^k \) and \( s^k \) along their diagonals.

Furthermore, if \((x^k, y^k, s^k) \in \mathcal{F}^0\), then
First, we observe that

$$
J(x^k, y^k, s^k) = \begin{bmatrix}
-Q & A^T & I \\
A & 0 & 0 \\
S^k & 0 & X^k
\end{bmatrix}
$$

where, $X^k$ and $S^k$ are diagonal matrices with the components of the vectors $x^k$ and $s^k$ along their diagonals. Furthermore, if $(x^k, y^k, s^k) \in F^0$, then

$$
F(x^k, y^k, s^k) = \begin{bmatrix}
0 \\
0 \\
X^kS^ke
\end{bmatrix}
$$
Algorithms for IPMs with pure Newton direction
Motivating Examples
The Quadratic Programming Problem
Optimality Conditions
Interior-Point Methods
Examples and QP Software
References

Algorithms for IPMs with pure Newton direction

Pure Newton Step

The Newton equation reduces to

\[
\begin{bmatrix}
-\mathbf{Q} \mathbf{A}^T \\
\mathbf{I}
\end{bmatrix}
\begin{bmatrix}
\Delta \mathbf{x}_k \\
\Delta \mathbf{y}_k \\
\Delta \mathbf{s}_k
\end{bmatrix}
= 
\begin{bmatrix}
\mathbf{0} \\
\mathbf{-X}_k \mathbf{S}_k \mathbf{e}
\end{bmatrix}.
\]

So by solving this equation system, we can find the search step for the (k+1)th iteration.

Exercise 5
Consider the quadratic programming problem given in Exercise 4 and the current primal-dual estimate of the solution \( \mathbf{x}_k = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})^T \), \( \mathbf{y}_k = (1, \frac{1}{2})^T \), and \( \mathbf{s}_k = (\frac{3}{2}, \frac{11}{6}, \frac{10}{3})^T \). Form and solve the Newton equation for this problem at \( (\mathbf{x}_k, \mathbf{y}_k, \mathbf{s}_k) \).
Algorithms for IPMs with pure Newton direction

Pure Newton Step

The Newton equation reduces to

\[
\begin{bmatrix}
-\mathbf{Q} \mathbf{A}^T \\
\mathbf{A}
\end{bmatrix}
\begin{bmatrix}
\Delta \mathbf{x}^k \\
\Delta \mathbf{y}^k \\
\Delta \mathbf{s}^k
\end{bmatrix}
= 
\begin{bmatrix}
0 \\
-\mathbf{X}^k \mathbf{S}^k \mathbf{e}
\end{bmatrix}.
\]

So by solving this equation system, we can find the search step for \((k+1)\)th iteration.

Exercise 5

Consider the quadratic programming problem given in Exercise 4 and the current primal-dual estimate of the solution

\[
\mathbf{x}^k = \begin{bmatrix} 1 \\ 3 \\ 1 \end{bmatrix},
\mathbf{y}^k = \begin{bmatrix} 1 \\ 1 \end{bmatrix},
\mathbf{s}^k = \begin{bmatrix} 2 \\ 11 \\ 6 \\ 10 \end{bmatrix}.
\]

Form and solve the Newton equation for this problem at \((\mathbf{x}^k, \mathbf{y}^k, \mathbf{s}^k)\).
Algorithms for IPMs with pure Newton direction

**Pure Newton Step**

The Newton equation reduces to

\[
\begin{bmatrix}
-\mathbf{Q} & \mathbf{A}^T & \mathbf{I} \\
\mathbf{A} & 0 & 0 \\
\mathbf{S}^k & 0 & \mathbf{X}^k
\end{bmatrix}
\begin{bmatrix}
\Delta \mathbf{x}^k \\
\Delta \mathbf{y}^k \\
\Delta \mathbf{s}^k
\end{bmatrix}
= \begin{bmatrix}
0 \\
0 \\
-\mathbf{X}^k \mathbf{S}^k \mathbf{e}
\end{bmatrix}.
\]
Algorithms for IPMs with pure Newton direction

Pure Newton Step

The Newton equation reduces to

\[
\begin{bmatrix}
-Q & A^T & I \\
A & 0 & 0 \\
S^k & 0 & X^k
\end{bmatrix}
\begin{bmatrix}
\Delta x^k \\
\Delta y^k \\
\Delta s^k
\end{bmatrix} =
\begin{bmatrix}
0 \\
0 \\
-X^k S^k e
\end{bmatrix}.
\]

So by solving this equation system, we can find the search step for \((k + 1)\)th iteration.
Algorithms for IPMs with pure Newton direction

Pure Newton Step

The Newton equation reduces to

\[
\begin{bmatrix}
-Q & A^T & I \\
A & 0 & 0 \\
S^k & 0 & X^k
\end{bmatrix}
\begin{bmatrix}
\Delta x^k \\
\Delta y^k \\
\Delta s^k
\end{bmatrix}
= \begin{bmatrix}
0 \\
0 \\
-X^kS^ke
\end{bmatrix}.
\]

So by solving this equation system, we can find the search step for \((k + 1)\)th iteration.

Exercise 5

Consider the quadratic programming problem given in Exercise 4 and the current primal-dual estimate of the solution \(x^k = (1, 1, 1)^T\), \(y^k = (1, 1, 2)^T\), and \(s^k = (3, 11, 6, 10, 3)^T\).

Form and solve the Newton equation for this problem at \((x^k, y^k, s^k)\).
Algorithms for IPMs with pure Newton direction

### Pure Newton Step

The Newton equation reduces to

\[
\begin{bmatrix}
-Q & A^T & I \\
A & 0 & 0 \\
S^k & 0 & X^k \\
\end{bmatrix}
\begin{bmatrix}
\Delta x^k \\
\Delta y^k \\
\Delta s^k \\
\end{bmatrix}
= \begin{bmatrix}
0 \\
0 \\
-X^k S^k e \\
\end{bmatrix}.
\]

So by solving this equation system, we can find the search step for \((k + 1)\)th iteration.

### Exercise 5

Consider the quadratic programming problem given in Exercise 4
Algorithms for IPMs with pure Newton direction

Pure Newton Step

The Newton equation reduces to

\[
\begin{bmatrix}
-Q & A^T & I \\
A & 0 & 0 \\
S^k & 0 & X^k \\
\end{bmatrix}
\begin{bmatrix}
\Delta x^k \\
\Delta y^k \\
\Delta s^k \\
\end{bmatrix}
=
\begin{bmatrix}
0 \\
0 \\
-X^k S^k e \\
\end{bmatrix}.
\]

So by solving this equation system, we can find the search step for \((k + 1)\)th iteration.

Exercise 5

Consider the quadratic programming problem given in Exercise 4 and the current primal-dual estimate of the solution \(x^k = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)^T\),
Algorithms for IPMs with pure Newton direction

Pure Newton Step

The Newton equation reduces to

\[
\begin{bmatrix}
-Q & A^T & I \\
A & 0 & 0 \\
S^k & 0 & \mathbf{X}^k
\end{bmatrix}
\begin{bmatrix}
\Delta \mathbf{x}^k \\
\Delta \mathbf{y}^k \\
\Delta s^k
\end{bmatrix}
= \begin{bmatrix}
0 \\
0 \\
-\mathbf{X}^k \mathbf{S}^k \mathbf{e}
\end{bmatrix}.
\]

So by solving this equation system, we can find the search step for \((k + 1)\)th iteration.

Exercise 5

Consider the quadratic programming problem given in Exercise 4 and the current primal-dual estimate of the solution \(\mathbf{x}^k = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)^T\), \(\mathbf{y}^k = (1, \frac{1}{2})^T\).
Algorithms for IPMs with pure Newton direction

Pure Newton Step

The Newton equation reduces to

\[
\begin{bmatrix}
-Q & A^T & I \\
A & 0 & 0 \\
S^k & 0 & X^k
\end{bmatrix}
\begin{bmatrix}
\Delta x^k \\
\Delta y^k \\
\Delta s^k
\end{bmatrix}
= \begin{bmatrix}
0 \\
0 \\
-X^k S^k e
\end{bmatrix}.
\]

So by solving this equation system, we can find the search step for \((k + 1)\)th iteration.

Exercise 5

Consider the quadratic programming problem given in Exercise 4 and the current primal-dual estimate of the solution \(\mathbf{x}^k = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)^T\), \(\mathbf{y}^k = (1, \frac{1}{2})^T\), and \(\mathbf{s}^k = (\frac{3}{2}, \frac{11}{6}, \frac{10}{3})^T\).
Algorithms for IPMs with pure Newton direction

### Pure Newton Step

The Newton equation reduces to

\[
\begin{bmatrix}
-Q & A^T & I \\
A & 0 & 0 \\
S^k & 0 & X^k \\
\end{bmatrix}
\begin{bmatrix}
\Delta x^k \\
\Delta y^k \\
\Delta s^k \\
\end{bmatrix}
= \begin{bmatrix}
0 \\
0 \\
-\Delta x^k S^k e \\
\end{bmatrix}.
\]

So by solving this equation system, we can find the search step for \((k + 1)\)th iteration.

### Exercise 5

Consider the quadratic programming problem given in Exercise 4 and the current primal-dual estimate of the solution \(x^k = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})^T\), \(y^k = (1, \frac{1}{2})^T\), and \(s^k = (\frac{3}{2}, \frac{11}{6}, \frac{10}{3})^T\). Form and solve the Newton equation for this problem at \((x^k, y^k, s^k)\).
Algorithms for IPMs with pure Newton direction

In our case, this action may not be permissible, since the Newton step may take us to a new point that does not necessarily satisfy the nonnegativity constraints $x \geq 0$ and $s \geq 0$. To avoid such violations, we seek a step-size parameter $\alpha_k \in (0, 1]$ such that $x_k + \alpha_k \Delta x_k > 0$ and $s_k + \alpha_k \Delta s_k > 0$.

Once we determine the step-size parameter, we choose the next iterate as $(x_{k+1}, y_{k+1}, s_{k+1}) = (x_k, y_k, s_k) + \alpha_k (\Delta x_k, \Delta y_k, \Delta s_k)$.
Step-size Parameter

In our case, this action may not be permissible, since the Newton step may take us to a new point that does not necessarily satisfy the nonnegativity constraints $x \geq 0$ and $s \geq 0$. To avoid such violations, we shall seek a step-size parameter $\alpha_k \in (0, 1]$ such that $x_k + \alpha_k \cdot \Delta x_k > 0$ and $s_k + \alpha_k \cdot \Delta s_k > 0$.

Once we determine the step-size parameter, we choose the next iterate as $(x_k, y_k, s_k) = (x_k, y_k, s_k) + \alpha_k \cdot (\Delta x_k, \Delta y_k, \Delta s_k)$. 

Z. Donovan and M. Xu
Optimization Methods in Finance
In our case, this action may not be permissible,
In our case, this action may not be permissible, since the Newton step may take us to a new point that does not necessarily satisfy the nonnegativity constraints.
In our case, this action may not be permissible, since the Newton step may take us to a new point that does not necessarily satisfy the nonnegativity constraints \( x \geq 0 \) and \( s \geq 0 \).
In our case, this action may not be permissible, since the Newton step may take us to a new point that does not necessarily satisfy the nonnegativity constraints $x \geq 0$ and $s \geq 0$.

To avoid such violations, we will seek a step-size parameter $\alpha^k \in (0, 1]$ such that $x^k + \alpha^k \cdot \Delta x^k > 0$ and $s^k + \alpha^k \cdot \Delta s^k > 0$. 
Step-size Parameter

- In our case, this action may not be permissible, since the Newton step may take us to a new point that does not necessarily satisfy the nonnegativity constraints \( x \geq 0 \) and \( s \geq 0 \).
- To avoid such violations, we sill seek a step-size parameter \( \alpha^k \in (0, 1] \) such that \( x^k + \alpha^k \cdot \Delta x^k > 0 \) and \( s^k + \alpha^k \cdot \Delta s^k > 0 \).
- Once we determine the step-size parameter, we choose the next iterate as

\[
(x^{k+1}, y^{k+1}, s^{k+1}) = (x^k, y^k, s^k) + \alpha^k \cdot (\Delta x^k, \Delta y^k, \Delta s^k).
\]
Motivating Examples
The Quadratic Programming Problem
Optimality Conditions
Interior-Point Methods
Examples and QP Software
References

Algorithms for IPMs with pure Newton direction

Weakness of IPMs with pure Newton direction
We often can take only a small step along the direction \( \alpha_k \ll 1 \) before violating the condition \( x_k + \alpha_k \cdot \Delta x_k > 0 \) and \( s_k + \alpha_k \cdot \Delta s_k > 0 \); hence, the pure Newton direction often does not allow us to make much more progress towards a solution.

Modify the basic Newton procedure in two important ways
1. They bias the search direction toward the interior of the nonnegative orthant \((x, s) \geq 0\) so that we can move further along the direction before one of the components of \((x, s)\) becomes negative.
2. They keep the component of \((x, s)\) from moving "too close" to the boundary of the nonnegative orthant. Search directions computed from points that are close to the boundary tend to be distorted, and little progress can be made along them.

Z. Donovan and M. Xu
Optimization Methods in Finance
Weakness of IPMs with pure Newton direction

We often can take only a small step along the direction ($\alpha_k \ll 1$) before violating the condition $x_{k+\alpha_k \cdot \Delta x_k} > 0$ and $s_{k+\alpha_k \cdot \Delta s_k} > 0$; hence, the pure Newton direction often does not allow us to make much more progress towards a solution.

Modify the basic Newton procedure in two important ways:
1. They bias the search direction toward the interior of the nonnegative orthant $(x, s) \geq 0$ so that we can move further along the direction before one of the components of $(x, s)$ becomes negative.
2. They keep the component of $(x, s)$ from moving “too close” to the boundary of the nonnegative orthant. Search directions computed from points that are close to the boundary tend to be distorted, and little progress can be made along them.
Weakness of IPMs with pure Newton direction

- We often can take only a small step along the direction \( \alpha^k \ll 1 \) before violating the condition \( x^k + \alpha^k \cdot \Delta x^k > 0 \) and \( s^k + \alpha^k \cdot \Delta s^k > 0 \);
We often can take only a small step along the direction ($\alpha^k \ll 1$) before violating the condition $x^k + \alpha^k \cdot \Delta x^k > 0$ and $s^k + \alpha^k \cdot \Delta s^k > 0$; hence, the pure Newton direction often does not allow us to make much more progress towards a solution.
Algorithms for IPMs with pure Newton direction

Weakness of IPMs with pure Newton direction

- We often can take only a small step along the direction ($\alpha_k \ll 1$) before violating the condition $\mathbf{x}^k + \alpha_k \cdot \Delta \mathbf{x}^k > 0$ and $\mathbf{s}^k + \alpha_k \cdot \Delta \mathbf{s}^k > 0$; hence, the pure Newton direction often does not allow us to make much more progress towards a solution.

Modify the basic Newton procedure in two important ways

- They bias the search direction toward the interior of the nonnegative orthant $(\mathbf{x}, \mathbf{s}) \geq 0$ so that we can move further along the direction before one of the components of $(\mathbf{x}, \mathbf{s})$ becomes negative.
- They keep the component of $(\mathbf{x}, \mathbf{s})$ from moving “too close” to the boundary of the nonnegative orthant. Search directions computed from points that are close to the boundary tend to be distorted, and little progress can be made along them.
Algorithms for IPMs with pure Newton direction

Weakness of IPMs with pure Newton direction

- We often can take only a small step along the direction ($\alpha^k \ll 1$) before violating the condition $x^k + \alpha^k \cdot \Delta x^k > 0$ and $s^k + \alpha^k \cdot \Delta s^k > 0$; hence, the pure Newton direction often does not allow us to make much more progress towards a solution.

Modify the basic Newton procedure in two important ways

1. They bias the search direction toward the interior of the nonnegative orthant $(x, s) \geq 0$
We often can take only a small step along the direction ($\alpha^k \ll 1$) before violating the condition $x^k + \alpha^k \cdot \Delta x^k > 0$ and $s^k + \alpha^k \cdot \Delta s^k > 0$; hence, the pure Newton direction often does not allow us to make much more progress towards a solution.

Modify the basic Newton procedure in two important ways

1. They bias the search direction toward the interior of the nonnegative orthant $(x, s) \succeq 0$ so that we can move further along the direction before one of the components of $(x, s)$ becomes negative.
Weakness of IPMs with pure Newton direction

- We often can take only a small step along the direction ($\alpha^k \ll 1$) before violating the condition $x^k + \alpha^k \cdot \Delta x^k > 0$ and $s^k + \alpha^k \cdot \Delta s^k > 0$; hence, the pure Newton direction often does not allow us to make much more progress towards a solution.

Modify the basic Newton procedure in two important ways

1. They bias the search direction toward the interior of the nonnegative orthant $(x, s) \geq 0$ so that we can move further along the direction before one of the components of $(x, s)$ becomes negative.

2. They keep the component of $(x, s)$ from moving “too close” to the boundary of the nonnegative of orthant.
We often can take only a small step along the direction \( \alpha_k \ll 1 \) before violating the condition \( x^k + \alpha_k \cdot \Delta x^k > 0 \) and \( s^k + \alpha_k \cdot \Delta s^k > 0 \); hence, the pure Newton direction often does not allow us to make much more progress towards a solution.

1. They bias the search direction toward the interior of the nonnegative orthant \((x, s) \geq 0\) so that we can move further along the direction before one of the components of \((x, s)\) becomes negative.
2. They keep the component of \((x, s)\) from moving “too close” to the boundary of the nonnegative of orthant. Search directions computed from points that are close to the boundary tend to be distorted,
Weakness of IPMs with pure Newton direction

- We often can take only a small step along the direction \( \alpha^k \ll 1 \) before violating the condition \( x^k + \alpha^k \cdot \Delta x^k > 0 \) and \( s^k + \alpha^k \cdot \Delta s^k > 0 \); hence, the pure Newton direction often does not allow us to make much more progress towards a solution.

Modify the basic Newton procedure in two important ways

1. They bias the search direction toward the interior of the nonnegative orthant \((x, s) \geq 0\) so that we can move further along the direction before one of the components of \((x, s)\) becomes negative.

2. They keep the component of \((x, s)\) from moving “too close” to the boundary of the nonnegative of orthant. Search directions computed from points that are close to the boundary tend to be distorted, and little progress can be made along them.
The Central Path

The central path $C$ is an arc of strictly feasible points (any point in $C$ is in $F_0$) that plays a vital role in the theory of primal-dual algorithm. It is parameterized by a scalar $\tau > 0$, and the points $(x_\tau, y_\tau, s_\tau)$ on the central path are obtained as solutions of the following system:

$$
F(x_\tau, y_\tau, s_\tau) = \begin{bmatrix}
0 \\
0 \\
\tau \cdot e
\end{bmatrix},
(x_\tau, s_\tau) > 0.
$$

Then, the central path $C$ is defined as

$$
C = \{ (x_\tau, y_\tau, s_\tau) : \tau > 0 \}.
$$

The third constraint can be rewritten as

$$(x_\tau)_i \cdot (s_\tau)_i = \tau, \quad \forall \ i.$$
The Central Path

Algorithm for IPMs with centered Newton direction

The Central Path

Z. Donovan and M. Xu
Algorithms for IPMs with centered Newton direction

The Central Path

The central path $C$ is an arc of strictly feasible points (any point in $C$ is in $\mathcal{F}^0$) that plays a vital role in the theory of primal-dual algorithm.
Algorithms for IPMs with centered Newton direction

The Central Path

The central path $C$ is an arc of strictly feasible points (any point in $C$ is in $\mathcal{F}^0$) that plays a vital role in the theory of primal-dual algorithm. It is parameterized by a scalar $\tau > 0$, ...
The Central Path

The central path $C$ is an arc of strictly feasible points (any point in $C$ is in $F^0$) that plays a vital role in the theory of primal-dual algorithm. It is parameterized by a scalar $\tau > 0$, and the points $(x_\tau, y_\tau, s_\tau)$ on the central path are obtained as solutions of the following system:
Algorithms for IPMs with centered Newton direction

The Central Path

The central path $C$ is an arc of strictly feasible points (any point in $C$ is in $\mathcal{F}^0$) that plays a vital role in the theory of primal-dual algorithm. It is parameterized by a scalar $\tau > 0$, and the points $(x_\tau, y_\tau, s_\tau)$ on the central path are obtained as solutions of the following system:

$$F(x_\tau, y_\tau, s_\tau) = \begin{bmatrix} 0 \\ 0 \\ \tau \cdot e \end{bmatrix}, (x_\tau, s_\tau) > 0.$$
The Central Path

The central path $\mathcal{C}$ is an arc of strictly feasible points (any point in $\mathcal{C}$ is in $\mathcal{F}^0$) that plays a vital role in the theory of primal-dual algorithm. It is parameterized by a scalar $\tau > 0$, and the points $(x_\tau, y_\tau, s_\tau)$ on the central path are obtained as solutions of the following system:

$$
F(x_\tau, y_\tau, s_\tau) = \begin{bmatrix} 0 \\ 0 \\ \tau \cdot e \end{bmatrix}, (x_\tau, s_\tau) > 0.
$$

Then, the central path $\mathcal{C}$ is defined as
The central path $C$ is an arc of strictly feasible points (any point in $C$ is in $\mathcal{F}^0$) that plays a vital role in the theory of primal-dual algorithm. It is parameterized by a scalar $\tau > 0$, and the points $(x_\tau, y_\tau, s_\tau)$ on the central path are obtained as solutions of the following system:

$$F(x_\tau, y_\tau, s_\tau) = \begin{bmatrix} 0 \\ 0 \\ \tau \cdot e \end{bmatrix}, (x_\tau, s_\tau) > 0.$$ 

Then, the central path $C$ is defined as

$$C = \{(x_\tau, y_\tau, s_\tau) : \tau > 0\}.$$
Algorithms for IPMs with centered Newton direction

The Central Path

The central path $C$ is an arc of strictly feasible points (any point in $C$ is in $\mathcal{F}^0$) that plays a vital role in the theory of primal-dual algorithm. It is parameterized by a scalar $\tau > 0$, and the points $(x_\tau, y_\tau, s_\tau)$ on the central path are obtained as solutions of the following system:

$$
F(x_\tau, y_\tau, s_\tau) = \begin{bmatrix}
0 \\
0 \\
\tau \cdot e
\end{bmatrix}, \quad (x_\tau, s_\tau) > 0.
$$

Then, the central path $C$ is defined as

$$
C = \{(x_\tau, y_\tau, s_\tau) : \tau > 0\}.
$$

The third constraint can be rewritten as
The Central Path

The central path $C$ is an arc of strictly feasible points (any point in $C$ is in $\mathcal{F}^0$) that plays a vital role in the theory of primal-dual algorithm. It is parameterized by a scalar $\tau > 0$, and the points $(x_\tau, y_\tau, s_\tau)$ on the central path are obtained as solutions of the following system:

$$
F(x_\tau, y_\tau, s_\tau) = 
\begin{bmatrix}
0 \\
0 \\
\tau \cdot e
\end{bmatrix},
(x_\tau, s_\tau) > 0.
$$

Then, the central path $C$ is defined as

$$
C = \{(x_\tau, y_\tau, s_\tau) : \tau > 0\}.
$$

The third constraint can be rewritten as

$$
(x_\tau)_i \cdot (s_\tau)_i = \tau, \forall i.
$$
The Central Path

Remark

Instead of the complementary condition, we require the products
\((x\tau)_i \cdot (s\tau)_i\)
have the same value for all \(i\).

The system has a unique solution for every \(\tau > 0\), provided that \(F_0\) is nonempty.

As \(\tau \to 0\), the conditions defining the points on the central path approximate the
set of optimality conditions more and more closely.

If \(F_0\) is nonempty, \((x\tau, y\tau, s\tau)\) will converge to an optimal solution of the problem.
Remark

Instead of the complementary condition, we require the products $\left(x_\tau \cdot s_\tau \right)_i$ have the same value for all $i$. The system has a unique solution for every $\tau > 0$, provided that $\mathcal{F}_0$ is nonempty. As $\tau \to 0$, the conditions defining the points on the central path approximate the set of optimality conditions more and more closely. If $\mathcal{F}_0$ is nonempty, $(x_\tau, y_\tau, s_\tau)$ will converge to an optimal solution of the problem.
Remark

Instead of the complementary condition, we require the products \((x_\tau)_i \cdot (s_\tau)_i\) have the same value for all \(i\).
Remark

- Instead of the complementary condition, we require the products $(x_\tau)_i \cdot (s_\tau)_i$ have the same value for all $i$.
- The system has a unique solution for every $\tau > 0$, provided that $\mathcal{F}^0$ is nonempty.
Remark

- Instead of the complementary condition, we require the products \((x_\tau)_i \cdot (s_\tau)_i\) have the same value for all \(i\).

- The system has a unique solution for every \(\tau > 0\), provided that \(\mathcal{F}^0\) is nonempty.

- As \(\tau \to 0\), the conditions defining the points on the central path approximate the set of optimality conditions more and more closely.
Remark

- Instead of the complementary condition, we require the products $(x_\tau)_i \cdot (s_\tau)_i$ have the same value for all $i$.

- The system has a unique solution for every $\tau > 0$, provided that $\mathcal{F}^0$ is nonempty.

- As $\tau \to 0$, the conditions defining the points on the central path approximate the set of optimality conditions more and more closely.

If $\mathcal{F}^0$ is nonempty, $(x_\tau, y_\tau, s_\tau)$ will converge to an optimal solution of the problem.
The Central Path
The Central Path
The Central Path
The Central Path
The Central Path
The Central Path
The Central Path
The Central Path

Recall the quadratic programming problem given in Exercise 4 and the current primal-dual estimate of the solution $x_k = (1, 3, 1)^T$, $y_k = (1, 1, 2)^T$, and $s_k = (3, 2, 11, 6, 10, 3)^T$. Verify that $(x_k, y_k, s_k)$ is not on the central path.
Recall the quadratic programming problem given in Exercise 4 and the current primal-dual estimate of the solution $x_k = (1, 1, 1)^T$, $y_k = (1, 1, 2)^T$, and $s_k = (3, 2, 11, 6, 10, 3)^T$. Verify that $(x_k, y_k, s_k)$ is not on the central path.
The Central Path

Exercise 6

Recall the quadratic programming problem given in Exercise 4
Recall the quadratic programming problem given in Exercise 4 and the current primal-dual estimate of the solution.
Exercise 6

Recall the quadratic programming problem given in Exercise 4 and the current primal-dual estimate of the solution $\mathbf{x}^k = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)^T$, $\mathbf{y}^k = \left(\frac{1}{3}, \frac{1}{2}\right)^T$, and $\mathbf{s}^k = \left(\frac{3}{2}, \frac{1}{6}, \frac{1}{3}\right)^T$. Verify that $(\mathbf{x}^k, \mathbf{y}^k, \mathbf{s}^k)$ is not on the central path.
Exercise 6

Recall the quadratic programming problem given in Exercise 4 and the current primal-dual estimate of the solution $x^k = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)^T$, $y^k = (1, \frac{1}{2})^T$, $s^k = \left(\frac{2}{3}, \frac{1}{2}, \frac{1}{3}\right)^T$.
Exercise 6

Recall the quadratic programming problem given in Exercise 4 and the current primal-dual estimate of the solution $x^k = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)^T$, $y^k = \left(1, \frac{1}{2}\right)^T$, and $s^k = \left(\frac{3}{2}, \frac{11}{6}, \frac{10}{3}\right)^T$. 

Verify that $(x^k, y^k, s^k)$ is not on the central path.
Recall the quadratic programming problem given in Exercise 4 and the current primal-dual estimate of the solution $\mathbf{x}^k = \begin{pmatrix} \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{pmatrix}^T$, $\mathbf{y}^k = \begin{pmatrix} 1 \\ \frac{1}{2} \end{pmatrix}^T$, and $\mathbf{s}^k = \begin{pmatrix} \frac{3}{2} \\ \frac{11}{6} \\ \frac{10}{3} \end{pmatrix}^T$. Verify that $(\mathbf{x}^k, \mathbf{y}^k, \mathbf{s}^k)$ is not on the central path.
IPMs with Centered Newton directions

To get over the weakness with pure Newton directions, most interior-point methods take a step toward points on the central path \( C \) corresponding to predetermined value of \( \tau \). Since such directions are aiming for central points, they are called centered directions.
To get over the weakness with pure Newton directions, most interior-point methods take a step toward points on the central path $C$ corresponding to predetermined value of $\tau$. Since such directions are aiming for central points, they are called centered directions.
To get over the weakness with pure Newton directions, most interior-point methods take a step toward points on the central path $C$ corresponding to predetermined value of $\tau$. 
To get over the weakness with pure Newton directions, most interior-point methods take a step toward points on the central path $C$ corresponding to predetermined value of $\tau$. Since such directions are aiming for central points, they are called centered directions.
IPMs with Centered Newton directions

Description

A centered direction is obtained by applying Newton update to the following system:

\[
\hat{F}(x, y, s) = \begin{bmatrix}
A^T \cdot y - Q \cdot x + s - c \\
A \cdot x - b \\
X \cdot S \cdot e - \tau \cdot e
\end{bmatrix} = \begin{bmatrix} 0 \\
0 \\
0 \end{bmatrix}.
\]

Since the Jacobian of \(\hat{F}\) is identical to the Jacobian of \(F\), proceeding as the previous Newton equation, we obtain the following (modified) Newton equation for the centered direction:

\[
\begin{bmatrix}
-Q A^T \\
A \\
0 \\
0 \\
S_k \\
X_k \\
\end{bmatrix} \begin{bmatrix}
\Delta x_k \\
\Delta y_k \\
\Delta s_k \\
\end{bmatrix} = \begin{bmatrix} 0 \\
\tau \cdot e - X_k \cdot S_k \cdot e \\
\end{bmatrix}.
\]
Motivating Examples
The Quadratic Programming Problem
Optimality Conditions
Interior-Point Methods
Examples and QP Software
References

IPMs with Centered Newton directions

Description
IPMs with Centered Newton directions

Description

A centered direction is obtained by applying Newton update to the following system:
A centered direction is obtained by applying Newton update to the following system:

\[
\hat{F}(x, y, s) = \begin{bmatrix}
A^T \cdot y - Q \cdot x + s - c \\
A \cdot x - b \\
X \cdot S \cdot e - \tau \cdot e
\end{bmatrix} = \begin{bmatrix}
0 \\
0 \\
0
\end{bmatrix}.
\]
A centered direction is obtained by applying Newton update to the following system:

\[
\begin{bmatrix}
A^T \cdot y - Q \cdot x + s - c \\
A \cdot x - b \\
X \cdot S \cdot e - \tau \cdot e
\end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.
\]

Since the Jacobian of \( \hat{F} \) is identical to the Jacobian of \( F \),
Description

A centered direction is obtained by applying Newton update to the following system:

\[
\hat{F}(x, y, s) = \begin{bmatrix}
    A^T \cdot y - Q \cdot x + s - c \\
    A \cdot x - b \\
    X \cdot S \cdot e - \tau \cdot e
\end{bmatrix} = \begin{bmatrix}
    0 \\
    0 \\
0
\end{bmatrix}.
\]

Since the Jacobian of \( \hat{F} \) is identical to the Jacobian of \( F \), proceeding as the previous Newton equation,
IPMs with Centered Newton directions

**Description**

A centered direction is obtained by applying Newton update to the following system:

\[
\hat{F}(x, y, s) = \begin{bmatrix}
    A^T \cdot y - Q \cdot x + s - c \\
    A \cdot x - b \\
    X \cdot S \cdot e - \tau \cdot e
\end{bmatrix} = \begin{bmatrix}
    0 \\
    0 \\
    0
\end{bmatrix}.
\]

Since the Jacobian of \( \hat{F} \) is identical to the Jacobian of \( F \), proceeding as the previous Newton equation, we obtain the following (modified) Newton equation for the centered direction:
IPMs with Centered Newton directions

Description

A centered direction is obtained by applying Newton update to the following system:

\[
\hat{F}(x, y, s) = \begin{bmatrix}
A^T \cdot y - Q \cdot x + s - c \\
A \cdot x - b \\
X \cdot S \cdot e - \tau \cdot e
\end{bmatrix} = \begin{bmatrix} 0 \\
0 \\
0 \end{bmatrix}.
\]

Since the Jacobian of \( \hat{F} \) is identical to the Jacobian of \( F \), proceeding as the previous Newton equation, we obtain the following (modified) Newton equation for the centered direction:

\[
\begin{bmatrix}
-Q & A^T & I \\
A & 0 & 0 \\
S^k & 0 & X^k
\end{bmatrix} \begin{bmatrix}
\Delta x_c^k \\
\Delta y_c^k \\
\Delta s_c^k
\end{bmatrix} = \begin{bmatrix} 0 \\
0 \\
\tau \cdot e - X^k \cdot S^k \cdot e \end{bmatrix}.
\]
IPMs with Centered Newton directions

We introduce a centering parameter $\sigma \in [0, 1]$ and a duality gap (or average complementarity) $\mu$ defined by:

$$
\mu(x, s) := \sum_{i=1}^{n} x_i \cdot s_i 
$$

Note that, when $(x, y, s)$ satisfy the conditions $A \cdot x = b$, $x \geq 0$ and $A^T \cdot y - Q \cdot x + s = c$, $s \geq 0$, then $(x, y, s)$ are optimal if and only if $\mu(x, s) = 0$.

If $\mu$ is large, then we are far away from the solution. Therefore, $\mu$ serves as a measure of optimality for feasible points – the smaller the duality gap, the closer the point of optimality.

For a central point $(x^\tau, y^\tau, s^\tau)$ we have

$$
\mu(x^\tau, s^\tau) = \sum_{i=1}^{n} (x^\tau)_i \cdot (s^\tau)_i
$$

Z. Donovan and M. Xu  
Optimization Methods in Finance
We introduce a centering parameter $\sigma \in [0, 1]$ and a duality gap (or average complementarity) $\mu$ defined by:

$$
\mu(x, s) := \sum_{i=1}^{n} x_i \cdot s_i \quad \text{in} \quad n = x^T \cdot s.
$$

Note that, when $(x, y, s)$ satisfy the conditions $A \cdot x = b$, $x \geq 0$ and $A^T \cdot y - Q \cdot x + s = c$, $s \geq 0$, then $(x, y, s)$ are optimal if and only if $\mu(x, s) = 0$.

If $\mu$ is large, then we are far away from the solution. Therefore, $\mu$ serves as a measure of optimality for feasible points – the smaller the duality gap, the closer the point of optimality.

For a central point $(x_\tau, y_\tau, s_\tau)$ we have

$$
\mu(x_\tau, s_\tau) = \sum_{i=1}^{n} (x_\tau)_i \cdot (s_\tau)_i \quad \text{in} \quad \sum_{i=1}^{n} \tau_n = \tau.
$$
We introduce a *centering parameter* \( \sigma \in [0, 1] \) and a duality gap (or average complementarity) \( \mu \) defined by:
IPMs with Centered Newton directions

Description

We introduce a *centering parameter* $\sigma \in [0, 1]$ and a duality gap (or average complementarity) $\mu$ defined by:

$$
\mu = \mu(x, s) := \sum_{i=1}^{n} \frac{x_i \cdot s_i}{n} = \frac{x^T \cdot s}{n}.
$$
We introduce a *centering parameter* $\sigma \in [0, 1]$ and a duality gap (or average complementarity) $\mu$ defined by:

$$\mu = \mu(x, s) := \frac{\sum_{i=1}^{n} x_i \cdot s_i}{n} = \frac{x^T \cdot s}{n}.$$

Note that, when $(x, y, s)$ satisfy the conditions $A \cdot x = b, x \geq 0$ and $A^T \cdot y - Q \cdot x + s = c, s \geq 0$, then $(x, y, s)$ are optimal if and only if $\mu(x, s) = 0$. 
IPMs with Centered Newton directions

Description

We introduce a centering parameter $\sigma \in [0, 1]$ and a duality gap (or average complementarity) $\mu$ defined by:

$$\mu = \mu(x, s) := \frac{\sum_{i=1}^{n} x_i \cdot s_i}{n} = \frac{x^T \cdot s}{n}.$$

Note that, when $(x, y, s)$ satisfy the conditions $A \cdot x = b, x \geq 0$ and $A^T \cdot y - Q \cdot x + s = c, s \geq 0$, then $(x, y, s)$ are optimal if and only if $\mu(x, s) = 0$. If $\mu$ is large, then we are far away from the solution.
IPMs with Centered Newton directions

Description

We introduce a centering parameter $\sigma \in [0, 1]$ and a duality gap (or average complementarity) $\mu$ defined by:

$$
\mu = \mu(x, s) := \frac{\sum_{i=1}^{n} x_i \cdot s_i}{n} = \frac{x^T \cdot s}{n}.
$$

Note that, when $(x, y, s)$ satisfy the conditions $A \cdot x = b, x \geq 0$ and $A^T \cdot y - Q \cdot x + s = c, s \geq 0$, then $(x, y, s)$ are optimal if and only if $\mu(x, s) = 0$. If $\mu$ is large, then we are far away from the solution. Therefore, $\mu$ serves as a measure of optimality for feasible points.
Description

We introduce a centering parameter $\sigma \in [0, 1]$ and a duality gap (or average complementarity) $\mu$ defined by:

$$\mu = \mu(x, s) := \frac{\sum_{i=1}^{n} x_i \cdot s_i}{n} = \frac{x^T \cdot s}{n}.$$

Note that, when $(x, y, s)$ satisfy the conditions $A \cdot x = b, x \geq 0$ and $A^T \cdot y - Q \cdot x + s = c, s \geq 0$, then $(x, y, s)$ are optimal if and only if $\mu(x, s) = 0$. If $\mu$ is large, then we are far away from the solution. Therefore, $\mu$ serves as a measure of optimality for feasible points – the smaller the duality gap, the closer the point of optimality.
IPMs with Centered Newton directions

Description

We introduce a centering parameter $\sigma \in [0, 1]$ and a duality gap (or average complementarity) $\mu$ defined by:

$$
\mu = \mu(x, s) := \frac{\sum_{i=1}^{n} x_i \cdot s_i}{n} = \frac{x^T \cdot s}{n}.
$$

Note that, when $(x, y, s)$ satisfy the conditions $A \cdot x = b, x \geq 0$ and $A^T \cdot y - Q \cdot x + s = c, s \geq 0$, then $(x, y, s)$ are optimal if and only if $\mu(x, s) = 0$. If $\mu$ is large, then we are far away from the solution. Therefore, $\mu$ serves as a measure of optimality for feasible points – the smaller the duality gap, the closer the point of optimality.

For a central point $(x_\tau, y_\tau, s_\tau)$ we have
IPMs with Centered Newton directions

**Description**

We introduce a *centering parameter* $\sigma \in [0, 1]$ and a duality gap (or average complementarity) $\mu$ defined by:

$$
\mu = \mu(x, s) := \frac{\sum_{i=1}^{n} x_i \cdot s_i}{n} = \frac{x^T \cdot s}{n}.
$$

Note that, when $(x, y, s)$ satisfy the conditions $A \cdot x = b, x \geq 0$ and $A^T \cdot y - Q \cdot x + s = c, s \geq 0$, then $(x, y, s)$ are optimal if and only if $\mu(x, s) = 0$. If $\mu$ is large, then we are far away from the solution. Therefore, $\mu$ serves as a measure of optimality for feasible points – the smaller the duality gap, the closer the point of optimality.

For a central point $(x_\tau, y_\tau, s_\tau)$ we have

$$
\mu(x_\tau, s_\tau) = \frac{\sum_{i=1}^{n} (x_\tau)_i \cdot (s_\tau)_i}{n} = \frac{\sum_{i=1}^{n} \tau}{n} = \tau.
$$
Using a simple change in notation, the centered direction can now be described as the solution of the following system:

\[
\begin{bmatrix}
-\mathbf{Q} & \mathbf{A}^T \\ \mathbf{A} & 0 & 0 & \mathbf{S} \\
\end{bmatrix}
\begin{bmatrix}
\Delta \mathbf{x}_k \\ \Delta \mathbf{c}_k \\ \Delta \mathbf{y}_k \\ \Delta \mathbf{s}_k 
\end{bmatrix} =
\begin{bmatrix}
0 \\ 0 \\ \sigma_k \cdot \mu_k \cdot \mathbf{e} - \mathbf{X}_k \cdot \mathbf{S}_k \cdot \mathbf{e}
\end{bmatrix},
\]

where \( \mu_k := \mu_k(\mathbf{x}_k, \mathbf{s}_k) = (\mathbf{x}_k)^T \cdot \mathbf{s}_k \) and \( \sigma_k \in [0, 1] \) is a user-defined quantity describing the ratio of the duality gap at the target central point and the current point. When \( \sigma_k = 1 \), we have a pure centering direction.
Using a simple change in notation, the centered direction can now be described as the solution of the following system:

\[
\begin{bmatrix}
-\mathbf{Q} & \mathbf{A}^T \\
\mathbf{A} & \mathbf{0} \\
\end{bmatrix}
\begin{bmatrix}
\Delta \mathbf{x}_k \\
\Delta \mathbf{y}_k \\
\Delta \mathbf{s}_k \\
\end{bmatrix}
= \begin{bmatrix}
\mathbf{0} \\
\mathbf{0} \\
\sigma_k \cdot \mathbf{\mu}_k \cdot \mathbf{e} - \mathbf{X}_k^T \cdot \mathbf{S}_k \cdot \mathbf{e} \\
\end{bmatrix},
\]

where \( \mathbf{\mu}_k := \mathbf{\mu}(\mathbf{x}_k, \mathbf{s}_k) = (\mathbf{x}_k^T \cdot \mathbf{s}_k) \) and \( \sigma_k \in [0, 1] \) is a user-defined quantity describing the ratio of the duality gap at the target central point and the current point. When \( \sigma_k = 1 \), we have a pure centering direction.
Using a simple change in notation, the centered direction can now be described as the solution of the following system:

\[
\begin{bmatrix}
-\mathbf{Q} & \mathbf{A}^T & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{S} \\
\mathbf{A} & \mathbf{0} & \mathbf{0} & \mathbf{X} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \sigma_k \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{e} \\
\end{bmatrix}
\begin{bmatrix}
\Delta \mathbf{x} \\
\Delta \mathbf{c} \\
\Delta \mathbf{y} \\
\Delta \mathbf{s} \\
\end{bmatrix}
= \begin{bmatrix}
\mathbf{0} \\
\mathbf{0} \\
\sigma_k \cdot \mathbf{mu} \\
\mathbf{e} - \mathbf{X} \cdot \mathbf{S} \cdot \mathbf{e} \\
\end{bmatrix},
\]

where \( \mathbf{\mu}_k := \mathbf{\mu}(\mathbf{x}_k, \mathbf{s}_k) = (\mathbf{x}_k)^T \cdot \mathbf{s}_k \) and \( \sigma_k \in [0, 1] \) is a user-defined quantity describing the ratio of the duality gap at the target central point and the current point. When \( \sigma_k = 1 \), we have a pure centering direction.
IPMs with Centered Newton directions

Description

Using a simple change in notation, the centered direction can now be described as the solution of the following system:

\[
\begin{bmatrix}
-Q & A^T & I \\
A & 0 & 0 \\
S^k & 0 & X^k \\
\end{bmatrix}
\begin{bmatrix}
\Delta x^k_c \\
\Delta y^k_c \\
\Delta s^k_c \\
\end{bmatrix}
= \begin{bmatrix}
0 \\
0 \\
\sigma^k \cdot \mu^k \cdot e - X^k \cdot S^k \cdot e \\
\end{bmatrix},
\]

where \( \mu^k := \mu(x^k, s^k) = (x^k)^T \cdot s^k \) and \( \sigma^k \in [0, 1] \) is a user-defined quantity describing the ratio of the duality gap at the target central point and the current point. When \( \sigma^k = 1 \), we have a pure centering direction.
IPMs with Centered Newton directions

Description

Using a simple change in notation, the centered direction can now be describe as the solution of the following system:

\[
\begin{bmatrix}
-Q & A^T & I \\
A & 0 & 0 \\
S^k & 0 & X^k
\end{bmatrix}
\begin{bmatrix}
\Delta x^k_c \\
\Delta y^k_c \\
\Delta s^k_c
\end{bmatrix}
= \begin{bmatrix}
0 \\
0 \\
\sigma^k \cdot m\mu^k \cdot e - X^k \cdot S^k \cdot e
\end{bmatrix},
\]

where \( \mu^k := \mu(x^k, s^k) = \frac{(x^k)^T \cdot s^k}{n} \) and \( \sigma^k \in [0, 1] \) is a user defined quantity describing the ratio of the duality gap at the target central point and the current point.
IPMs with Centered Newton directions

Description

Using a simple change in notation, the centered direction can now be described as the solution of the following system:

\[
\begin{bmatrix}
-Q & A^T & I \\
A & 0 & 0 \\
S^k & 0 & X^k \\
\end{bmatrix}
\begin{bmatrix}
\Delta x_c^k \\
\Delta y_c^k \\
\Delta s_c^k \\
\end{bmatrix} =
\begin{bmatrix}
0 \\
0 \\
\sigma^k \cdot m^k \cdot e - X^k \cdot S^k \cdot e \\
\end{bmatrix},
\]

where \(\mu^k := \mu(x^k, s^k) = \frac{(x^k)^T \cdot s^k}{n}\) and \(\sigma^k \in [0, 1]\) is a user-defined quantity describing the ratio of the duality gap at the target central point and the current point. When \(\sigma^k = 1\), we have a pure centering direction.
IPMs with Centered Newton directions

Remark

We have three ways to choose to target a central point:

At a lower level than our current point \((\tau < \mu(x, s))\);

At the same level as our current point \((\tau = \mu(x, s))\);

At a higher level than our current point \((\tau > \mu(x, s))\).

In most circumstances, the third option is not a good choice as it targets a central point that is "farther" than the current iterate to the optimal solution. Therefore, we will always choose \(\tau \leq \mu(x, s)\) in defining centered directions.
IPMs with Centered Newton directions

Remark

We have three ways to choose to target a central point:

1. At a lower level than our current point ($\tau < \mu(x, s)$);
2. At the same level as our current point ($\tau = \mu(x, s)$);
3. At a higher level than our current point ($\tau > \mu(x, s)$).

In most circumstances, the third option is not a good choice as it targets a central point that is "farther" than the current iterate to the optimal solution. Therefore, we will always choose $\tau \leq \mu(x, s)$ in defining centered directions.
Remark

We have three ways to choose to target a central point:
Remark

We have three ways to choose to target a central point:

- At a lower level than our current point \( \tau < \mu(x, s) \);
Remark

We have three ways to choose to target a central point:

- At a lower level than our current point ($\tau < \mu(x, s)$);
- At the same level as our current point ($\tau = \mu(x, s)$);
Remark

We have three ways to choose to target a central point:

- At a lower level than our current point ($\tau < \mu(x, s)$);
- At the same level as our current point ($\tau = \mu(x, s)$);
- At a higher level than our current point ($\tau > \mu(x, s)$).
Remark

We have three ways to choose to target a central point:

- At a lower level than our current point ($\tau < \mu(x, s)$);
- At the same level as our current point ($\tau = \mu(x, s)$);
- At a higher level than our current point ($\tau > \mu(x, s)$).

In most circumstances, the third option is not a good choice as it targets a central point that is "farther" than the current iterate to the optimal solution.
Remark

We have three ways to choose to target a central point:

- At a lower level than our current point ($\tau < \mu(x, s)$);
- At the same level as our current point ($\tau = \mu(x, s)$);
- At a higher level than our current point ($\tau > \mu(x, s)$).

In most circumstances, the third option is not a good choice as it targets a central point that is "farther" than the current iterate to the optimal solution.

Therefore, we will always choose $\tau \leq \mu(x, s)$ in defining centered directions.
Generic Interior Point Algorithm

With these basic concepts in hand, we can define a general primal-dual interior point algorithm. Choose $(x_0, y_0, s_0) \in F_0$. For $k = 0, 1, 2, ...$ repeat the following steps.
With these basic concepts in hand, we can define a general primal-dual interior point algorithm.
With these basic concepts in hand, we can define a general primal-dual interior point algorithm.

Choose \((x^0, y^0, s^0) \in \mathcal{F}^0\). For \(k = 0, 1, 2, \ldots\) repeat the following steps.
Generic Interior Point Algorithm

Choose $\sigma_k \in [0, 1]$, let $\mu_k = (x_k)^T \cdot s_k$. Solve

$$
\begin{bmatrix}
- Q A^T & I \\
A & 0 & 0 \\
S_k & 0 & X_k \\
\end{bmatrix}
\begin{bmatrix}
\Delta x_k \\
\Delta c_k \\
\Delta s_k \\
\end{bmatrix} =
\begin{bmatrix}
0 \\
0 \\
\sigma_k \cdot \mu_k \cdot e - X_k \cdot S_k \cdot e \\
\end{bmatrix}.
$$

Choose $\alpha_k$ such that $x_k + \alpha_k \cdot \Delta x_k > 0$, $s_k + \alpha_k \cdot \Delta s_k > 0$. Set $(x_{k+1}, y_{k+1}, s_{k+1}) = (x_k, y_k, s_k) + \alpha_k \cdot (\Delta x_k, \Delta y_k, \Delta s_k)$, and $k := k + 1$. 

Z. Donovan and M. Xu
Optimization Methods in Finance
Generic Interior Point Algorithm

General Interior Point Algorithm II

<table>
<thead>
<tr>
<th>( \sigma_k \in [0, 1] )</th>
<th>( \mu_k = (x_k^T s_k^n) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Choose ( \alpha_k ) such that ( x_k + \alpha_k \cdot \Delta x_k &gt; 0 ), ( s_k + \alpha_k \cdot \Delta s_k &gt; 0 )</td>
<td></td>
</tr>
<tr>
<td>Set ((x_{k+1}, y_{k+1}, s_{k+1}) = (x_k, y_k, s_k) + \alpha_k \cdot (\Delta x_k, \Delta y_k, \Delta s_k))</td>
<td></td>
</tr>
</tbody>
</table>
Choose $\sigma^k \in [0, 1]$, let $\mu^k = \frac{(x^k)^T \cdot s^k}{n}$. Solve
Choose $\sigma^k \in [0, 1]$, let $\mu^k = \frac{(x^k)^T s^k}{n}$. Solve

$$
\begin{bmatrix}
-Q & A^T & I \\
A & 0 & 0 \\
S^k & 0 & X^k \\
\end{bmatrix}
\begin{bmatrix}
\Delta x^k_c \\
\Delta y^k_c \\
\Delta s^k_c \\
\end{bmatrix}
= 
\begin{bmatrix}
0 \\
0 \\
\sigma^k \cdot \mu^k \cdot e - X^k \cdot S^k \cdot e \\
\end{bmatrix}.
$$
Generic Interior Point Algorithm

**General Interior Point Algorithm II**

- Choose \( \sigma^k \in [0, 1] \), let \( \mu^k = \frac{(x^k)^T \cdot s^k}{n} \). Solve

\[
\begin{bmatrix}
-Q & A^T & I \\
A & 0 & 0 \\
S^k & 0 & X^k \\
\end{bmatrix}
\begin{bmatrix}
\Delta x^k_c \\
\Delta y^k_c \\
\Delta s^k_c \\
\end{bmatrix} =
\begin{bmatrix}
0 \\
0 \\
\sigma^k \cdot \mu^k \cdot e - X^k \cdot S^k \cdot e \\
\end{bmatrix}.
\]

- Choose \( \alpha^k \) such that

\[x^{k+1} = x^k + \alpha^k \cdot \Delta x^k, \quad s^{k+1} = s^k + \alpha^k \cdot \Delta s^k, \quad k := k + 1.\]
Generic Interior Point Algorithm

General Interior Point Algorithm II

- Choose $\sigma^k \in [0, 1]$, let $\mu^k = \frac{(x^k)^T \cdot s^k}{n}$. Solve

$$
\begin{bmatrix}
-Q & A^T & I \\
A & 0 & 0 \\
S^k & 0 & X^k \\
\end{bmatrix}
\begin{bmatrix}
\Delta x^k_c \\
\Delta y^k_c \\
\Delta s^k_c \\
\end{bmatrix} =
\begin{bmatrix}
0 \\
0 \\
\sigma^k \cdot \mu^k \cdot e - X^k \cdot S^k \cdot e \\
\end{bmatrix}.
$$

- Choose $\alpha^k$ such that

$$x^k + \alpha^k \cdot \Delta x^k > 0, \quad s^k + \alpha^k \cdot \Delta s^k > 0.$$
Choose $\sigma^k \in [0, 1]$, let $\mu^k = \frac{(x^k)^T s^k}{n}$. Solve

$$
\begin{bmatrix}
  -Q & A^T & I \\
  A & 0 & 0 \\
  S^k & 0 & X^k 
\end{bmatrix}
\begin{bmatrix}
  \Delta x_c^k \\
  \Delta y_c^k \\
  \Delta s_c^k 
\end{bmatrix}
= 
\begin{bmatrix}
  0 \\
  0 \\
  \sigma^k \cdot \mu^k \cdot e - X^k \cdot S^k \cdot e 
\end{bmatrix}.
$$

Choose $\alpha^k$ such that

$$x^k + \alpha^k \cdot \Delta x^k > 0, \quad s^k + \alpha^k \cdot \Delta s^k > 0.$$
Choose $\sigma^k \in [0, 1]$, let $\mu^k = \frac{(x^k)^T s^k}{n}$. Solve

$$
\begin{bmatrix}
-Q & A^T & I \\
A & 0 & 0 \\
S^k & 0 & X^k
\end{bmatrix}
\begin{bmatrix}
\Delta x^k_c \\
\Delta y^k_c \\
\Delta s^k_c
\end{bmatrix} =
\begin{bmatrix}
0 \\
0 \\
\sigma^k \cdot \mu^k \cdot e - X^k \cdot S^k \cdot e
\end{bmatrix}.
$$

Choose $\alpha^k$ such that

$$
x^k + \alpha^k \cdot \Delta x^k > 0, s^k + \alpha^k \cdot \Delta s^k > 0.
$$

Set

$$
(x^{k+1}, y^{k+1}, s^{k+1}) = (x^k, y^k, s^k) + \alpha^k \cdot (\Delta x^k, \Delta y^k, \Delta s^k),
$$

Z. Donovan and M. Xu

Optimization Methods in Finance
Generic Interior Point Algorithm

General Interior Point Algorithm II

Choose $\sigma^k \in [0, 1]$, let $\mu^k = \frac{(x^k)^T \cdot s^k}{n}$. Solve

$$\begin{bmatrix}
-Q & A^T & I \\
A & 0 & 0 \\
S^k & 0 & X^k
\end{bmatrix}
\begin{bmatrix}
\Delta x_c^k \\
\Delta y_c^k \\
\Delta s_c^k
\end{bmatrix} =
\begin{bmatrix}
0 \\
0 \\
\sigma^k \cdot \mu^k \cdot e - X^k \cdot S^k \cdot e
\end{bmatrix}.$$  

Choose $\alpha^k$ such that

$$x^k + \alpha^k \cdot \Delta x^k > 0, s^k + \alpha^k \cdot \Delta s^k > 0.$$  

Set

$$(x^{k+1}, y^{k+1}, s^{k+1}) = (x^k, y^k, s^k) + \alpha^k \cdot (\Delta x^k, \Delta y^k, \Delta s^k),$$  

and $k := k + 1$.  

Z. Donovan and M. Xu  
Optimization Methods in Finance
Starting From an Infeasible Point
Starting From an Infeasible Point

Motivation

The generic interior point method starts with a strictly feasible iterate.
Starting From an Infeasible Point

Motivation

The generic interior point method starts with a strictly feasible iterate.

It is not practical since finding such a starting point is not always a trivial task.
Starting From an Infeasible Point

Motivation

The generic interior point method starts with a strictly feasible iterate.

It is not practical since finding such a starting point is not always a trivial task.

Fortunately, however, we can accommodate infeasible starting points with a small modification of the linear system we solve in each iteration.
Starting From an Infeasible Point

We only require that the initial point \((x_0, y_0, s_0)\) satisfy the nonnegativity restrictions strictly: \(x_0 > 0\) and \(s_0 > 0\).

Then the Newton equation from an infeasible point \((x_k, y_k, s_k)\) is reduced to

\[
\begin{bmatrix}
- Q A^T I \\
A_0 0 \\
S_k 0
\end{bmatrix}
\begin{bmatrix}
\Delta x_k \\
\Delta y_k \\
\Delta s_k
\end{bmatrix}
=
\begin{bmatrix}
c + Q \cdot x_k - A^T \cdot y_k - s_k \tau \cdot e - X_k \cdot S_k \cdot e
\end{bmatrix}
\]

We no longer have zeros in the first and second blocks of the right-hand-side vector since we are not assuming that the iterates satisfy \(A \cdot x_k = b\) and \(A^T \cdot y_k - Q \cdot x_k + s_k = c\).

Replacing the linear system in this case, the algorithms can work simultaneously.
Starting From an Infeasible Point

Modification

- We only require that the initial point \((x^0, y^0, s^0)\) satisfy the nonnegativity restrictions strictly: \(x^0 > 0\) and \(s^0 > 0\).
Starting From an Infeasible Point

Modification

- We only require that the initial point \((x^0, y^0, s^0)\) satisfy the nonnegativity restrictions strictly: \(x^0 > 0\) and \(s^0 > 0\).
- Then the Newton equation from an infeasible point \((x^k, y^k, s^k)\) is reduced to
Starting From an Infeasible Point

Modification

- We only require that the initial point \((x^0, y^0, s^0)\) satisfy the nonnegativity restrictions strictly: \(x^0 > 0\) and \(s^0 > 0\).
- Then the Newton equation from an infeasible point \((x^k, y^k, s^k)\) is reduced to

\[
\begin{bmatrix}
-Q & A^T & I \\
A & 0 & 0 \\
S^k & 0 & X^k
\end{bmatrix}
\begin{bmatrix}
\Delta x^k \\
\Delta y^k \\
\Delta s^k
\end{bmatrix}
= \begin{bmatrix}
c + Q \cdot x^k - A^T \cdot y^k - s^k \\
b - A \cdot x^k \\
\tau \cdot e - X^k \cdot S^k \cdot e
\end{bmatrix}
\]
Starting From an Infeasible Point

Modification

- We only require that the initial point \((x^0, y^0, s^0)\) satisfy the nonnegativity restrictions strictly: \(x^0 > 0\) and \(s^0 > 0\).

- Then the Newton equation from an infeasible point \((x^k, y^k, s^k)\) is reduced to

\[
\begin{bmatrix}
-Q & A^T & I \\
A & 0 & 0 \\
S^k & 0 & X^k
\end{bmatrix}
\begin{bmatrix}
\Delta x^k \\
\Delta y^k \\
\Delta s^k
\end{bmatrix}
= \begin{bmatrix}
c + Q \cdot x^k - A^T \cdot y^k - s^k \\
b - A \cdot x^k \\
\tau \cdot e - X^k \cdot S^k \cdot e
\end{bmatrix}
\]

We no longer have zeros in the first and second blocks of the right-hand-side vector since we are not assuming that the iterates satisfy...
Starting From an Infeasible Point

Modification

- We only require that the initial point \((x^0, y^0, s^0)\) satisfy the nonnegativity restrictions strictly: \(x^0 > 0\) and \(s^0 > 0\).
- Then the Newton equation from an infeasible point \((x^k, y^k, s^k)\) is reduced to

\[
\begin{bmatrix}
-Q & A^T & I \\
A & 0 & 0 \\
S^k & 0 & X^k
\end{bmatrix}
\begin{bmatrix}
\Delta x^k \\
\Delta y^k \\
\Delta s^k
\end{bmatrix}
= \begin{bmatrix}
c + Q \cdot x^k - A^T \cdot y^k - s^k \\
b - A \cdot x^k \\
\tau \cdot e - X^k \cdot S^k \cdot e
\end{bmatrix}
\]

We no longer have zeros in the first and second blocks of the right-hand-side vector since we are not assuming that the iterates satisfy \(A \cdot x^k = b\).
Starting From an Infeasible Point

**Modification**

- We only require that the initial point \((x^0, y^0, s^0)\) satisfy the nonnegativity restrictions strictly: \(x^0 > 0\) and \(s^0 > 0\).

- Then the Newton equation from an infeasible point \((x^k, y^k, s^k)\) is reduced to

\[
\begin{bmatrix}
-Q & A^T & I \\
A & 0 & 0 \\
S^k & 0 & X^k
\end{bmatrix}
\begin{bmatrix}
\Delta x^k \\
\Delta y^k \\
\Delta s^k
\end{bmatrix}
= \begin{bmatrix}
c + Q \cdot x^k - A^T \cdot y^k - s^k \\
b - A \cdot x^k \\
\tau \cdot e - X^k \cdot S^k \cdot e
\end{bmatrix}
\]

We no longer have zeros in the first and second blocks of the right-hand-side vector since we are not assuming that the iterates satisfy \(A \cdot x^k = b\) and \(A^T \cdot y^k - Q \cdot x^k + s^k = c\).
Starting From an Infeasible Point

**Modification**

- We only require that the initial point \((x^0, y^0, s^0)\) satisfy the nonnegativity restrictions strictly: \(x^0 > 0\) and \(s^0 > 0\).

- Then the Newton equation from an infeasible point \((x^k, y^k, s^k)\) is reduced to

\[
\begin{bmatrix}
-Q & A^T & I \\
A & 0 & 0 \\
S^k & 0 & X^k
\end{bmatrix}
\begin{bmatrix}
\Delta x^k \\
\Delta y^k \\
\Delta s^k
\end{bmatrix}
= 
\begin{bmatrix}
c + Q \cdot x^k - A^T \cdot y^k - s^k \\
b - A \cdot x^k \\
\tau \cdot e - X^k \cdot S^k \cdot e
\end{bmatrix}
\]

We no longer have zeros in the first and second blocks of the right-hand-side vector since we are not assuming that the iterates satisfy \(A \cdot x^k = b\) and \(A^T \cdot y^k - Q \cdot x^k + s^k = c\).

- Replacing the linear system in this case, the algorithms can work simultaneously.
Markowitz, in his Nobel prize winning work, showed that a rational investor's notion of minimizing risk can be closely approximated by minimizing the variance of the return of the investment portfolio.
Solving Motivating Example 2

Optimization model for Example 2: Quantifying the notion of “risk”

- Markowitz, in his Nobel prize winning work,
Solving Motivating Example 2

Optimization model for Example 2: Quantifying the notion of “risk”

- Markowitz, in his Nobel prize winning work, showed that a rational investor’s notion of minimizing risk can be closely approximated
Markowitz, in his Nobel prize winning work, showed that a rational investor’s notion of minimizing risk can be closely approximated by minimizing the variance of the return of the investment portfolio.
Recall that:

We wish to invest $1000.00 in stocks A, B, and C for a one month period.

We buy a stock at some dollar amount per share in the beginning of the month, and sell it at some dollar amount per share at the end of the month.

The rate of return of each stock is a random variable with some expected value.

Our goal is to invest in such a way that the expected end-of-month return is at least $50.00 or 5%.
Solving Motivating Example 2

Recall that:

We wish to invest $1000.00 in stocks A, B, and C for a one month period. We buy a stock at some dollar amount per share in the beginning of the month, and sell it at some dollar amount per share at the end of the month. The rate of return of each stock is a random variable with some expected value. Our goal is to invest in such a way that the expected end-of-month return is at least $50.00 or 5%.
We wish to invest $1000.00 in stocks A, B, and C for a one month period. We buy a stock at some dollar amount per share in the beginning of the month, and sell it at some dollar amount per share at the end of the month. The rate of return of each stock is a random variable with some expected value. Our goal is to invest in such a way that the expected end-of-month return is at least $50.00 or 5%.
Recall that:

- We wish to invest $1000.00 in stocks A, B, and C for a one month period.
Example

- Recall that:
- We wish to invest $1000.00 in stocks A, B, and C for a one month period.
- We buy a stock at some dollar amount per share in the beginning of the month,
Solving Motivating Example 2

Example

- Recall that:
  - We wish to invest $1000.00 in stocks A, B, and C for a one month period.
  - We buy a stock at some dollar amount per share in the beginning of the month, and sell it at some dollar amount per share at the end of the month.
Example

- Recall that:
  - We wish to invest $1000.00 in stocks A, B, and C for a one month period.
  - We buy a stock at some dollar amount per share in the beginning of the month, and sell it at some dollar amount per share at the end of the month.
  - The rate of return of each stock is a random variable with some expected value.
Example

- Recall that:
  - We wish to invest $1000.00 in stocks A, B, and C for a one month period.
  - We buy a stock at some dollar amount per share in the beginning of the month, and sell it at some dollar amount per share at the end of the month.
  - The rate of return of each stock is a random variable with some expected value.
  - Our goal is to invest in such a way that the expected end-of-month return is at least $50.00 or 5%.
Solving Motivating Example 2

Optimization model for Example 2: The decision variables $x_i$, $i = 1, 2, 3$, denote the dollars invested in stock $i$. Since we have a total of $1000.00 to invest, then the $x_i$'s should satisfy:

\[ \sum_{i=1}^{3} x_i \leq 1000 \]
\[ x_i \geq 0, \quad i = 1, 2, 3 \]
Solving Motivating Example 2

Optimization model for Example 2: The decision variables

Our decision variables are $x_i$, $i = 1, 2, 3$, denoting the dollars invested in stock $i$. Since we have a total of $1000.00 to invest, then the $x_i$'s should satisfy:

$$3 \sum_{i=1}^{3} x_i \leq 1000$$

$$x_i \geq 0, \quad i = 1, 2, 3$$
Solving Motivating Example 2

Optimization model for Example 2: The decision variables

- Our decision variables are $x_i$, $i = 1, 2, 3$, denoting the dollars invested in stock $i$. 
Solving Motivating Example 2

Optimization model for Example 2: The decision variables

- Our decision variables are $x_i, i = 1, 2, 3$, denoting the dollars invested in stock $i$.
- Since we have a total of $1000.00 to invest, then the $x_i$'s should satisfy:
Optimization model for Example 2: The decision variables

- Our decision variables are \( x_i, i = 1, 2, 3 \), denoting the dollars invested in stock \( i \).
- Since we have a total of $1000.00 to invest, then the \( x_i \)'s should satisfy:

\[
\sum_{i=1}^{3} x_i \leq 1000
\]
Solving Motivating Example 2

Optimization model for Example 2: The decision variables

- Our decision variables are $x_i, i = 1, 2, 3$, denoting the dollars invested in stock $i$.
- Since we have a total of $\$1000.00$ to invest, then the $x_i$’s should satisfy:

\[
\sum_{i=1}^{3} x_i \leq 1000
\]

\[
x_i \geq 0, \ i = 1, 2, 3
\]
Solving Motivating Example 2

Optimization model for Example 2: Covariance matrix

Let $Q$ denote the covariance matrix of rates of stock returns.

The classical mean-variance model consists of minimizing portfolio risk, as measured by $\sum x^T Q x$, and subject to a set of constraints.

Since we want to have an expected return of at least $50.00, then

$$\sum \bar{r}_i \cdot x_i \geq 50$$

Where $\bar{r}_i$ is the expected value of the random variable corresponding to the monthly return per dollar for stock $i$. 

Z. Donovan and M. Xu

Optimization Methods in Finance
Let \( Q \) denote the covariance matrix of rates of stock returns.

The classical mean-variance model consists of minimizing portfolio risk, as measured by 
\[
\frac{1}{2} x^T Q x,
\]
and subject to a set of constraints. Since we want to have an expected return of at least $50.00, then 
\[
\sum_{i=1}^{n} \bar{r}_i x_i \geq 50,
\]
where \( \bar{r}_i \) is the expected value of the random variable corresponding to the monthly return per dollar for stock \( i \).
Solving Motivating Example 2

Optimization model for Example 2: Covariance matrix

- Let **Q** denote the covariance matrix of rates of stock returns.
- The classical mean-variance model consists of minimizing portfolio risk, as measured by
Solving Motivating Example 2

Optimization model for Example 2: Covariance matrix

- Let $Q$ denote the covariance matrix of rates of stock returns.
- The classical mean-variance model consists of minimizing portfolio risk, as measured by

$$\frac{1}{2} \cdot x^T \cdot Q \cdot x,$$

Where $\bar{r}_i$ is the expected value of the random variable corresponding to the monthly return per dollar for stock $i$. 
Solving Motivating Example 2

Optimization model for Example 2: Covariance matrix

- Let $Q$ denote the covariance matrix of rates of stock returns.
- The classical mean-variance model consists of minimizing portfolio risk, as measured by
  \[
  \frac{1}{2} \cdot x^T \cdot Q \cdot x,
  \]
  and subject to a set of constraints.
Optimization model for Example 2: Covariance matrix

- Let $Q$ denote the covariance matrix of rates of stock returns.
- The classical mean-variance model consists of minimizing portfolio risk, as measured by
  $$\frac{1}{2} \cdot x^T \cdot Q \cdot x,$$
  and subject to a set of constraints.
- Since we want to have an expected return of at least $50.00, then
Let \( Q \) denote the covariance matrix of rates of stock returns.

The classical mean-variance model consists of minimizing portfolio risk, as measured by

\[
\frac{1}{2} \cdot x^T \cdot Q \cdot x,
\]

and subject to a set of constraints.

Since we want to have an expected return of at least $50.00, then

\[
\sum_{i=1}^{3} \bar{r}_i \cdot x_i \geq 50
\]
Let $\mathbf{Q}$ denote the covariance matrix of rates of stock returns.

The classical mean-variance model consists of minimizing portfolio risk, as measured by

$$
\frac{1}{2} \cdot \mathbf{x}^T \cdot \mathbf{Q} \cdot \mathbf{x},
$$

and subject to a set of constraints.

Since we want to have an expected return of at least $50.00, then

$$
\sum_{i=1}^{3} \bar{r}_i \cdot x_i \geq 50
$$

Where $\bar{r}_i$ is the expected value of the random variable corresponding to the monthly return per dollar for stock $i$. 
Solving Motivating Example 2

Optimization model for Example 2

Using matrices and vectors, our optimization model can be compactly stated as follows:

$$\begin{align*}
\min & \quad x^T \cdot Q \cdot x \\
\text{subject to} & \quad e^T \cdot x \leq 1000 \\
& \quad \bar{r}^T \cdot x \geq 50 \\
& \quad x \geq 0
\end{align*}$$

Where $x$ is the decision vector of size $n$ ($n$ is the number of stocks, $n = 3$ in our example), $e$ is an $n$-vector of ones, $\bar{r}$ is the $n$-vector of expected returns of the stocks, and $Q$ is the $n \times n$ covariance matrix.
Solving Motivating Example 2

Optimization model for Example 2

Using matrices and vectors, our optimization model can be compactly stated as follows:

\[
\begin{align*}
\min \quad & x^T \cdot Q \cdot x \\
\text{subject to} \quad & e^T \cdot x \leq 1000 \\
& \bar{r}^T \cdot x \geq 50 \\
& x \geq 0
\end{align*}
\]

Where \( x \) is the decision vector of size \( n \) (\( n \) is the number of stocks, \( n = 3 \) in our example), \( e \) is an \( n \)-vector of ones, \( \bar{r} \) is the \( n \)-vector of expected returns of the stocks, and \( Q \) is the \( n \times n \) covariance matrix.
Solving Motivating Example 2

Optimization model for Example 2

Using matrices and vectors, our optimization model can be compactly stated as follows:

\[
\begin{align*}
\min & \quad x^T \cdot Q \cdot x \\
e^T \cdot x & \leq 1000 \\
\bar{r}^T \cdot x & \geq 50 \\
x & \geq 0
\end{align*}
\]

Where \(x\) is the decision vector of size \(n\) (\(n\) is the number of stocks, \(n = 3\) in our example), \(e\) is an \(n\)-vector of ones, \(\bar{r}\) is the \(n\)-vector of expected returns of the stocks, and \(Q\) is the \(n \times n\) covariance matrix.
Solving Motivating Example 2

**Optimization model for Example 2**

- Using matrices and vectors, our optimization model can be compactly stated as follows:

$$\min x^T \cdot Q \cdot x$$
Solving Motivating Example 2

Optimization model for Example 2

Using matrices and vectors, our optimization model can be compactly stated as follows:

\[
\begin{align*}
\min & \quad x^T \cdot Q \cdot x \\
\text{subject to} & \quad e^T \cdot x \leq 1000 \\
& \quad x \geq 0
\end{align*}
\]
Solving Motivating Example 2

Using matrices and vectors, our optimization model can be compactly stated as follows:

\[
\begin{align*}
\min & \quad x^T \cdot Q \cdot x \\
\text{subject to} & \quad e^T \cdot x \leq 1000 \\
& \quad \bar{r}^T \cdot x \geq 50
\end{align*}
\]
Using matrices and vectors, our optimization model can be compactly stated as follows:

\[
\min \quad x^T \cdot Q \cdot x \\
\quad e^T \cdot x \leq 1000 \\
\quad \bar{r}^T \cdot x \geq 50 \\
\quad x \geq 0
\]
Solving Motivating Example 2

Optimization model for Example 2

Using matrices and vectors, our optimization model can be compactly stated as follows:

$$\min \quad x^T \cdot Q \cdot x$$

$$e^T \cdot x \quad \leq \quad 1000$$

$$\bar{r}^T \cdot x \quad \geq \quad 50$$

$$x \quad \geq \quad 0$$

Where $x$ is the decision vector of size $n$ ($n$ is the number of stocks, $n = 3$ in our example),
Using matrices and vectors, our optimization model can be compactly stated as follows:

\[
\begin{align*}
\min & \quad x^T \cdot Q \cdot x \\
\quad & \quad e^T \cdot x \leq 1000 \\
\quad & \quad \bar{r}^T \cdot x \geq 50 \\
\quad & \quad x \geq 0
\end{align*}
\]

Where \( x \) is the decision vector of size \( n \) (\( n \) is the number of stocks, \( n = 3 \) in our example), \( e \) is an \( n \)-vector of ones,
Solving Motivating Example 2

Using matrices and vectors, our optimization model can be compactly stated as follows:

\[
\begin{align*}
\text{min} & \quad x^T \cdot Q \cdot x \\
\text{subject to} & \quad e^T \cdot x \leq 1000 \\
& \quad \bar{r}^T \cdot x \geq 50 \\
& \quad x \geq 0
\end{align*}
\]

Where \( x \) is the decision vector of size \( n \) (\( n \) is the number of stocks, \( n = 3 \) in our example), \( e \) is an \( n \)-vector of ones, \( \bar{r} \) is the \( n \)-vector of expected returns of the stocks, and
Solving Motivating Example 2

**Optimization model for Example 2**

Using matrices and vectors, our optimization model can be compactly stated as follows:

\[
\begin{align*}
\text{min } & \quad x^T \cdot Q \cdot x \\
\text{such that } & \quad e^T \cdot x \leq 1000 \\
& \quad \bar{r}^T \cdot x \geq 50 \\
& \quad x \geq 0
\end{align*}
\]

Where \(x\) is the decision vector of size \(n\) (\(n\) is the number of stocks, \(n = 3\) in our example), \(e\) is an \(n\)-vector of ones, \(\bar{r}\) is the \(n\)-vector of expected returns of the stocks, and \(Q\) is the \(n \times n\) covariance matrix.
Using MATLAB and Optimization Toolbox Function \texttt{quadprog}

Solving our motivating example using MATLAB

For our example, suppose that

\[
\begin{bmatrix}
0 & 0.171 & 0.0033 & 0.0012 \\
0 & 0.033 & 0.0059 & 0.0045 \\
0 & 0.012 & 0.0045 & 0.0630
\end{bmatrix},
\]

and

\[
\begin{bmatrix}
0.026 \\
0.008 \\
0.074
\end{bmatrix}
\]

So,

\[
A = \begin{bmatrix}
1 & 1 & 1 \\
-0.026 & -0.008 & -0.074
\end{bmatrix}, \quad b = \begin{bmatrix}
1000 \\
-50
\end{bmatrix}
\]

Z. Donovan and M. Xu

Optimization Methods in Finance
Using MATLAB and Optimization Toolbox Function \texttt{quadprog}

Solving our motivating example using MATLAB

For our example, suppose that $Q = \begin{bmatrix} 0 & 0.0171 & 0 \\ 0.0033 & 0 & 0.0012 \\ 0.0033 & 0.0059 & 0.0045 \\ 0 & 0.0012 & 0.0045 & 0.0063 \end{bmatrix}$, and $\bar{r} = \begin{bmatrix} 0.026 \\ 0.008 \\ 0.074 \end{bmatrix}$.

So, $A = \begin{bmatrix} 1 & 1 & 1 \\ -0.026 & -0.008 & -0.074 \end{bmatrix}$, and $b = \begin{bmatrix} 1000 \\ -50 \end{bmatrix}$.
Solving our motivating example using MATLAB

- For our example, suppose that

\[ Q = \begin{bmatrix} 0 & 0.0171 & 0.0033 & 0.0012 \\ 0.0171 & 0.0033 & 0.0059 & 0.0045 \\ 0.0033 & 0.0059 & 0.0630 & 0.0045 \\ 0.0012 & 0.0045 & 0.0630 & 0.0045 \end{bmatrix} \]

\[ \bar{r} = \begin{bmatrix} 0.026 \\ 0.008 \\ 0.074 \end{bmatrix} \]

So,

\[ A = \begin{bmatrix} 1 & 1 & 1 \\ -0.026 & -0.008 & -0.074 \end{bmatrix} \]

\[ b = \begin{bmatrix} 1000 \\ -50 \end{bmatrix} \]
For our example, suppose that 

\[
Q = \begin{bmatrix}
0.0171 & 0.0033 & 0.0012 \\
0.0033 & 0.0059 & 0.0045 \\
0.0012 & 0.0045 & 0.0630 \\
\end{bmatrix}
\]
Using MATLAB and Optimization Toolbox Function \texttt{quadprog}

Solving our motivating example using MATLAB

- For our example, suppose that

\[
Q = \begin{bmatrix}
0.0171 & 0.0033 & 0.0012 \\
0.0033 & 0.0059 & 0.0045 \\
0.0012 & 0.0045 & 0.0630
\end{bmatrix}, \text{ and}
\]

Z. Donovan and M. Xu  Optimization Methods in Finance
Using MATLAB and Optimization Toolbox Function \texttt{quadprog}

Solving our motivating example using MATLAB

- For our example, suppose that

\[
Q = \begin{bmatrix}
0.0171 & 0.0033 & 0.0012 \\
0.0033 & 0.0059 & 0.0045 \\
0.0012 & 0.0045 & 0.0630 \\
\end{bmatrix}, \quad \text{and} \quad \bar{r} = \begin{bmatrix}
0.026 \\
0.008 \\
0.074 \\
\end{bmatrix}
\]
Using MATLAB and Optimization Toolbox Function \texttt{quadprog}

Solving our motivating example using MATLAB

- For our example, suppose that

\[
Q = \begin{bmatrix}
0.0171 & 0.0033 & 0.0012 \\
0.0033 & 0.0059 & 0.0045 \\
0.0012 & 0.0045 & 0.0630
\end{bmatrix}, \quad \text{and} \quad \bar{r} = \begin{bmatrix}
0.026 \\
0.008 \\
0.074
\end{bmatrix}
\]

- So,

\[
A = \begin{bmatrix}
1 & 1 & 1 \\
-0.026 & -0.008 & -0.074
\end{bmatrix}
\]
Solving our motivating example using MATLAB

- For our example, suppose that

\[
Q = \begin{bmatrix}
0.0171 & 0.0033 & 0.0012 \\
0.0033 & 0.0059 & 0.0045 \\
0.0012 & 0.0045 & 0.0630
\end{bmatrix}, \quad \text{and} \quad \bar{r} = \begin{bmatrix}
0.026 \\
0.008 \\
0.074
\end{bmatrix}
\]

- So,

\[
A = \begin{bmatrix}
1 & 1 & 1 \\
-0.026 & -0.008 & -0.074
\end{bmatrix}, \quad \text{and}
\]
Solving our motivating example using MATLAB

For our example, suppose that

\[
Q = \begin{bmatrix}
0.0171 & 0.0033 & 0.0012 \\
0.0033 & 0.0059 & 0.0045 \\
0.0012 & 0.0045 & 0.0630
\end{bmatrix}, \quad \text{and} \quad \bar{r} = \begin{bmatrix}
0.026 \\
0.008 \\
0.074
\end{bmatrix}
\]

So,

\[
A = \begin{bmatrix}
1 & 1 & 1 \\
-0.026 & -0.008 & -0.074
\end{bmatrix}, \quad \text{and} \quad b = \begin{bmatrix}
1000 \\
-50
\end{bmatrix}
\]
Using MATLAB and Optimization Toolbox Function **quadprog**
Solving our motivating example using MATLAB

We find that our optimal solution is $x^* = \begin{bmatrix} 500 \\ 0 \\ 500 \end{bmatrix}$. 

Using MATLAB and Optimization Toolbox Function `quadprog`
Using MATLAB and Optimization Toolbox Function \texttt{quadprog}

Solving our motivating example using MATLAB

- We find that our optimal solution is

\[
\begin{pmatrix}
500 \\
0 \\
500
\end{pmatrix}
\]
Solving our motivating example using MATLAB

- We find that our optimal solution is \( \mathbf{x}^* = \begin{bmatrix} 500 \\ 0 \\ 500 \end{bmatrix} \).
Using MATLAB and Optimization Toolbox Function \texttt{quadprog}

\[ \begin{align*}
\min & \quad x_1^2 + 3x_1 + 4x_2 \\
\text{s.t.} & \quad -x_1 + 3x_2 \geq 15 \\
& \quad 2x_1 + 5x_2 \leq 100 \\
& \quad 3x_1 + 4x_2 \leq 80 \\
& \quad x_1, x_2 \geq 0 \\
\end{align*} \]

We find that our optimal solution is \( x^* = (0, 5) \).
Using MATLAB and Optimization Toolbox Function \texttt{quadprog}

Solving another example using MATLAB

\[
\min_{x} \quad \frac{1}{2} \cdot x_1^2 + 3 \cdot x_1 + 4 \cdot x_2
\]
Using MATLAB and Optimization Toolbox Function \texttt{quadprog}

Solving another example using MATLAB

\[
\min \quad \frac{1}{2} \cdot x_1^2 + 3 \cdot x_1 + 4 \cdot x_2 \\
\quad x_1 + 3 \cdot x_2 \geq 15
\]
Motivating Examples
The Quadratic Programming Problem
Optimality Conditions
Interior-Point Methods
Examples and QP Software
References

Using MATLAB and Optimization Toolbox Function \texttt{quadprog}

Solving another example using MATLAB

\[
\min_x \frac{1}{2} x_1^2 + 3 \cdot x_1 + 4 \cdot x_2 \\
\text{s.t. } x_1 + 3 \cdot x_2 \geq 15 \\
2 \cdot x_1 + 5 \cdot x_2 \leq 100
\]
Using MATLAB and Optimization Toolbox Function \texttt{quadprog}

Solving another example using MATLAB

\[
\begin{align*}
\min_{x} & \quad \frac{1}{2} x_1^2 + 3 \cdot x_1 + 4 \cdot x_2 \\
\quad & x_1 + 3 \cdot x_2 \geq 15 \\
\quad & 2 \cdot x_1 + 5 \cdot x_2 \leq 100 \\
\quad & 3 \cdot x_1 + 4 \cdot x_2 \leq 80 
\end{align*}
\]
Using MATLAB and Optimization Toolbox Function \texttt{quadprog}

Solving another example using MATLAB

\[
\min_{x} \quad \frac{1}{2} \cdot x_1^2 + 3 \cdot x_1 + 4 \cdot x_2 \\
x_1 + 3 \cdot x_2 \quad \geq \quad 15 \\
2 \cdot x_1 + 5 \cdot x_2 \quad \leq \quad 100 \\
3 \cdot x_1 + 4 \cdot x_2 \quad \leq \quad 80 \\
x_1, x_2 \quad \geq \quad 0
\]
**Motivating Examples**

**The Quadratic Programming Problem**

**Optimality Conditions**

**Interior-Point Methods**

**Examples and QP Software**

**References**

Using MATLAB and Optimization Toolbox Function `quadprog`

Solving another example using MATLAB

\[
\begin{align*}
\min_x & \quad \frac{1}{2} x_1^2 + 3 \cdot x_1 + 4 \cdot x_2 \\
& \quad x_1 + 3 \cdot x_2 \geq 15 \\
& \quad 2 \cdot x_1 + 5 \cdot x_2 \leq 100 \\
& \quad 3 \cdot x_1 + 4 \cdot x_2 \leq 80 \\
& \quad x_1, x_2 \geq 0 \\
\end{align*}
\]

\[
Q = \begin{bmatrix}
1 & 0 \\
0 & 0 \\
\end{bmatrix},
\]

Z. Donovan and M. Xu

Optimization Methods in Finance
Solving another example using MATLAB

\[
\min_x \quad \frac{1}{2} x_1^2 + 3 \cdot x_1 + 4 \cdot x_2 \\
\quad x_1 + 3 \cdot x_2 \geq 15 \\
\quad 2 \cdot x_1 + 5 \cdot x_2 \leq 100 \\
\quad 3 \cdot x_1 + 4 \cdot x_2 \leq 80 \\
\quad x_1, x_2 \geq 0
\]

\[
Q = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad c = \begin{bmatrix} 3 \\ 4 \end{bmatrix}
\]
Using MATLAB and Optimization Toolbox Function \texttt{quadprog}

Solving another example using MATLAB

\[
\begin{align*}
\min_{x} & \quad \frac{1}{2} \cdot x_1^2 + 3 \cdot x_1 + 4 \cdot x_2 \\
\text{s.t.} & \quad x_1 + 3 \cdot x_2 \geq 15 \\
& \quad 2 \cdot x_1 + 5 \cdot x_2 \leq 100 \\
& \quad 3 \cdot x_1 + 4 \cdot x_2 \leq 80 \\
& \quad x_1, x_2 \geq 0
\end{align*}
\]

\[
Q = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad c = \begin{bmatrix} 3 \\ 4 \end{bmatrix}, \quad A = \begin{bmatrix} -1 & -3 \\ 2 & 5 \\ 3 & 4 \end{bmatrix},
\]
Solving another example using MATLAB

\[
\begin{align*}
\min_{x} & \quad \frac{1}{2} x_1^2 + 3 \cdot x_1 + 4 \cdot x_2 \\
& \quad x_1 + 3 \cdot x_2 \geq 15 \\
& \quad 2 \cdot x_1 + 5 \cdot x_2 \leq 100 \\
& \quad 3 \cdot x_1 + 4 \cdot x_2 \leq 80 \\
& \quad x_1, x_2 \geq 0
\end{align*}
\]

\[Q = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad c = \begin{bmatrix} 3 \\ 4 \end{bmatrix}, \quad A = \begin{bmatrix} -1 & -3 \\ 2 & 5 \\ 3 & 4 \end{bmatrix}, \quad b = \begin{bmatrix} -15 \\ 100 \\ 80 \end{bmatrix}\]
Using MATLAB and Optimization Toolbox Function \texttt{quadprog}

Solving another example using MATLAB

\[
\begin{align*}
\min_{x} & \quad \frac{1}{2} \cdot x_1^2 + 3 \cdot x_1 + 4 \cdot x_2 \\
& \quad x_1 + 3 \cdot x_2 \geq 15 \\
& \quad 2 \cdot x_1 + 5 \cdot x_2 \leq 100 \\
& \quad 3 \cdot x_1 + 4 \cdot x_2 \leq 80 \\
& \quad x_1, x_2 \geq 0
\end{align*}
\]

\[
Q = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad c = \begin{bmatrix} 3 \\ 4 \end{bmatrix}, \quad A = \begin{bmatrix} -1 & -3 \\ 2 & 5 \\ 3 & 4 \end{bmatrix}, \quad b = \begin{bmatrix} -15 \\ 100 \\ 80 \end{bmatrix}
\]

We find that our optimal solution is \(x^* = (0, 5)\).
Using MATLAB and Optimization Toolbox Function \texttt{quadprog}

Solving another example using MATLAB

\[
\begin{align*}
\min_{\mathbf{x}} & \quad \frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{c}^T \mathbf{x} \\
\text{s.t.} & \quad \mathbf{A} \mathbf{x} \leq \mathbf{b} \\
& \quad \mathbf{x}^T \mathbf{x} \geq 0
\end{align*}
\]

\[
\begin{align*}
\mathbf{Q} &= \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, & \mathbf{c} &= \begin{bmatrix} 3 \\ 4 \end{bmatrix}, & \mathbf{A} &= \begin{bmatrix} -1 & -3 \\ 2 & 5 \\ 3 & 4 \end{bmatrix}, & \mathbf{b} &= \begin{bmatrix} -15 \\ 100 \\ 80 \end{bmatrix}
\end{align*}
\]

We find that our optimal solution is \( \mathbf{x}^* = (0, 5) \).
Motivating Examples
The Quadratic Programming Problem
Optimality Conditions
Interior-Point Methods
Examples and QP Software
References

References

5. Website: https://inst.eecs.berkeley.edu/~ee127a/book/login/exa_quad_fcn_cvx.html

References
References

References


References


References


References


5. Website: https://inst.eecs.berkeley.edu/~ee127a/book/login/exa_quad_fcn_cvx.html