Lecture 1
Linear Optimization
Duality, Simplex Methods
Robert J. Vanderbei
April 14, 2012

Machine Learning Summer School
La Palma

http://www.princeton.edu/~rvdb
Linear Programming

**Standard form:**

```
maximize \ c^T x \\
subject to \ Ax \leq b \\
x \geq 0.
```

**Dictionary.** Add slack variables \( w \) and name the objective function:

\[
\begin{align*}
\zeta &= c^T x \\
w &= b - Ax \\
x, w \geq 0.
\end{align*}
\]

**Dictionary Solution.** Set \( x = 0 \) and read off values of \( w \)'s: \( w = b \).

**Feasibility.** If \( b \geq 0 \), then \( w \geq 0 \) and so dictionary solution is feasible.

**Optimality.** If \( b \geq 0 \) and \( c \leq 0 \), then the dictionary solution is optimal.
Primal Simplex Method  (used when feasible)

Dictionary:

\[
\begin{align*}
\zeta & = c^T x \\
w & = b - Ax
\end{align*}
\]

\[x, w \geq 0.\]

**Entering Variable.** Choose an index \(j\) for which \(c_j > 0\). Variable \(x_j\) is the entering variable.

**Leaving Variable.** Let \(x_j\) increase while holding all other \(x_k\)'s at zero. Figure out which slack variable hits zero first. Let's say it's \(w_i\).

**Pivot.** Rearrange equations so that entering variable \(x_j\) is on left and leaving variable \(w_i\) is on right to get a new dictionary.

\[
\begin{align*}
\zeta & = \zeta^* + \tilde{c}^T x_N \\
x_B & = \tilde{b} - \tilde{A} x_N
\end{align*}
\]

\[x, w \geq 0.\]

Here \(x_B\) denotes the vector of slack variables with \(w_i\) replaced with \(x_j\), \(x_N\) denotes the set of original variables with \(x_j\) replaced with \(w_i\), the tildes on \(A\), \(b\), and \(c\) denote the fact that the constants have changed, and \(\zeta^*\) is the value of the objective function associated with this new dictionary. Continue until dictionary solution is optimal. Click here to try.
### Unboundedness

Consider the following dictionary:

<table>
<thead>
<tr>
<th></th>
<th>Current Dictionary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>obj</strong> = 0.0</td>
<td><strong>2.0</strong> x1 + <strong>-1.0</strong> x2 + <strong>1.0</strong> x3</td>
</tr>
<tr>
<td><strong>w1</strong> = 4.0</td>
<td>- <strong>-5.0</strong> x1 - <strong>3.0</strong> x2 - <strong>-1.0</strong> x3</td>
</tr>
<tr>
<td><strong>w2</strong> = 10.0</td>
<td>- <strong>-1.0</strong> x1 - <strong>-5.0</strong> x2 - <strong>2.0</strong> x3</td>
</tr>
<tr>
<td><strong>w3</strong> = 7.0</td>
<td>- <strong>0.0</strong> x1 - <strong>-4.0</strong> x2 - <strong>3.0</strong> x3</td>
</tr>
<tr>
<td><strong>w4</strong> = 6.0</td>
<td>- <strong>-2.0</strong> x1 - <strong>-2.0</strong> x2 - <strong>4.0</strong> x3</td>
</tr>
<tr>
<td><strong>w5</strong> = 6.0</td>
<td>- <strong>-3.0</strong> x1 - <strong>0.0</strong> x2 - <strong>-3.0</strong> x3</td>
</tr>
</tbody>
</table>

- Could increase either \( x_1 \) or \( x_3 \) to increase obj.
- Consider increasing \( x_1 \).
- Which basic variable decreases to zero first?
- Answer: none of them, \( x_1 \) can grow without bound, and obj along with it.
- This is how we detect *unboundedness* with the simplex method.
Initialization

Consider the following problem:

\[
\text{maximize} \quad -3x_1 + 4x_2 \\
\text{subject to} \quad -4x_1 - 2x_2 \leq -8 \\
-2x_1 \leq -2 \\
3x_1 + 2x_2 \leq 10 \\
-x_1 + 3x_2 \leq 1 \\
-3x_2 \leq -2 \\
\]

\[x_1, x_2 \geq 0.\]

Phase-I Problem

• Modify problem by subtracting a new variable, \( x_0 \), from each constraint and
• replacing objective function with \(-x_0\)
Phase-I Problem

maximize \( -x_0 \)

subject to

\[
\begin{align*}
-x_0 - 4x_1 - 2x_2 & \leq -8 \\
-x_0 - 2x_1 & \leq -2 \\
-x_0 + 3x_1 + 2x_2 & \leq 10 \\
-x_0 - x_1 + 3x_2 & \leq 1 \\
-x_0 & \leq -2
\end{align*}
\]

\( x_0, x_1, x_2 \geq 0. \)

- Clearly feasible: pick \( x_0 \) large, \( x_1 = 0 \) and \( x_2 = 0 \).
- If optimal solution has \( \text{obj} = 0 \), then original problem is feasible.
- Final phase-I basis can be used as initial \textit{phase-II} basis (ignoring \( x_0 \) thereafter).
- If optimal solution has \( \text{obj} < 0 \), then original problem is infeasible.
Initialization—First Pivot

Applet depiction shows both the Phase-I and the Phase-II objectives:

<table>
<thead>
<tr>
<th>Current Dictionary</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td><strong>obj</strong> = 0.0</td>
</tr>
<tr>
<td>0.0</td>
</tr>
<tr>
<td><strong>w1</strong> = -8.0</td>
</tr>
<tr>
<td><strong>w2</strong> = -2.0</td>
</tr>
<tr>
<td><strong>w3</strong> = 10.0</td>
</tr>
<tr>
<td><strong>w4</strong> = 1.0</td>
</tr>
<tr>
<td><strong>w5</strong> = -2.0</td>
</tr>
</tbody>
</table>

- Dictionary is infeasible even for Phase-I.
- One pivot needed to get feasible.
- Entering variable is $x_0$.
- Leaving variable is one whose current value is most negative, i.e. $w_1$.
- After first pivot...
Going into second pivot:

<table>
<thead>
<tr>
<th>obj</th>
<th>0.0</th>
<th>+</th>
<th>0.0</th>
<th>w1</th>
<th>+</th>
<th>-3.0</th>
<th>x1</th>
<th>+</th>
<th>4.0</th>
<th>x2</th>
</tr>
</thead>
<tbody>
<tr>
<td>-8.0</td>
<td></td>
<td>+</td>
<td>-1.0</td>
<td>w1</td>
<td>+</td>
<td>4.0</td>
<td>x1</td>
<td>+</td>
<td>2.0</td>
<td>x2</td>
</tr>
<tr>
<td>x0</td>
<td>8.0</td>
<td>-</td>
<td>-1.0</td>
<td>w1</td>
<td>-</td>
<td>4.0</td>
<td>x1</td>
<td>-</td>
<td>2.0</td>
<td>x2</td>
</tr>
<tr>
<td>w2</td>
<td>6.0</td>
<td>-</td>
<td>-1.0</td>
<td>w1</td>
<td>-</td>
<td>2.0</td>
<td>x1</td>
<td>-</td>
<td>2.0</td>
<td>x2</td>
</tr>
<tr>
<td>w3</td>
<td>18.0</td>
<td>-</td>
<td>-1.0</td>
<td>w1</td>
<td>-</td>
<td>7.0</td>
<td>x1</td>
<td>-</td>
<td>4.0</td>
<td>x2</td>
</tr>
<tr>
<td>w4</td>
<td>9.0</td>
<td>-</td>
<td>-1.0</td>
<td>w1</td>
<td>-</td>
<td>3.0</td>
<td>x1</td>
<td>-</td>
<td>5.0</td>
<td>x2</td>
</tr>
<tr>
<td>w5</td>
<td>6.0</td>
<td>-</td>
<td>-1.0</td>
<td>w1</td>
<td>-</td>
<td>4.0</td>
<td>x1</td>
<td>-</td>
<td>-1.0</td>
<td>x2</td>
</tr>
</tbody>
</table>

- Feasible!
- Focus on the yellow highlights.
- Let $x_1$ enter.
- Then $w_5$ must leave.
- After second pivot...
Going into third pivot:

<table>
<thead>
<tr>
<th></th>
<th>Current Dictionary</th>
</tr>
</thead>
<tbody>
<tr>
<td>obj</td>
<td>-4.5 + -0.75 w₁ + 0.75 w₅ + 3.25 x₂</td>
</tr>
<tr>
<td>x₀</td>
<td>-2.0 + 0.0 w₁ - 1.0 w₅ - 3.0 x₂</td>
</tr>
<tr>
<td>w₂</td>
<td>2.0 - 0.0 w₁ - 1.0 w₅ - 3.0 x₂</td>
</tr>
<tr>
<td>w₃</td>
<td>3.0 - -0.5 w₁ - -0.5 w₅ - 2.5 x₂</td>
</tr>
<tr>
<td>w₄</td>
<td>7.5 - 0.75 w₁ - -1.75 w₅ - 5.75 x₂</td>
</tr>
<tr>
<td>x₁</td>
<td>4.5 - -0.25 w₁ - -0.75 w₅ - 5.75 x₂</td>
</tr>
</tbody>
</table>

- $x₂$ must enter.
- $x₀$ must leave.
- After third pivot...
End of Phase-I

Current dictionary:

| obj  | -7/3 | + | -3/4 | w1 | + | 11/6 | w5 | + | 0  | x0 |
| x2   | 2/3  | - | 0    | w1 | - | -1/3 | w5 | - | 0  | x0 |
| w2   | 4/3  | - | -1/2 | w1 | - | 1/3  | w5 | - | 0  | x0 |
| w3   | 11/3 | - | 3/4  | w1 | - | 1/6  | w5 | - | 0  | x0 |
| w4   | 2/3  | - | -1/4 | w1 | - | 7/6  | w5 | - | 0  | x0 |
| x1   | 5/3  | - | -1/4 | w1 | - | 1/6  | w5 | - | 0  | x0 |

- Optimal for Phase-I (no yellow highlights).
- \( \text{obj} = 0 \), therefore original problem is feasible.
Phase-II

Current dictionary:

<table>
<thead>
<tr>
<th></th>
<th>Current Dictionary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>obj</strong> =</td>
<td><strong>-7/3</strong> + <strong>-3/4</strong> w1 + <strong>11/6</strong> w5 + 0 x0</td>
</tr>
<tr>
<td>x2  =</td>
<td><strong>2/3</strong> - 0 w1 - <strong>-1/3</strong> w5 - 0 x0</td>
</tr>
<tr>
<td>w2  =</td>
<td><strong>4/3</strong> - <strong>-1/2</strong> w1 - <strong>1/3</strong> w5 - 0 x0</td>
</tr>
<tr>
<td>w3  =</td>
<td><strong>11/3</strong> - <strong>3/4</strong> w1 - <strong>1/6</strong> w5 - 0 x0</td>
</tr>
<tr>
<td>w4  =</td>
<td><strong>2/3</strong> - <strong>-1/4</strong> w1 - <strong>7/6</strong> w5 - 0 x0</td>
</tr>
<tr>
<td>x1  =</td>
<td><strong>5/3</strong> - <strong>-1/4</strong> w1 - <strong>1/6</strong> w5 - 0 x0</td>
</tr>
</tbody>
</table>

For Phase-II:

- Ignore column with $x_0$ in Phase-II.
- Ignore Phase-I objective row.

$w_5$ must enter. $w_4$ must leave...
### Optimal Solution

**Current Dictionary**

<table>
<thead>
<tr>
<th></th>
<th>obj</th>
<th>x0</th>
<th>x0</th>
<th>x0</th>
<th>x0</th>
<th>x0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-9/7</td>
<td>-5/14</td>
<td>11/7</td>
<td>11/7</td>
<td>11/7</td>
<td>11/7</td>
</tr>
<tr>
<td>x2</td>
<td>6/7</td>
<td>-1/14</td>
<td>2/7</td>
<td>2/7</td>
<td>2/7</td>
<td>2/7</td>
</tr>
<tr>
<td>w2</td>
<td>8/7</td>
<td>-3/7</td>
<td>-2/7</td>
<td>-2/7</td>
<td>-2/7</td>
<td>-2/7</td>
</tr>
<tr>
<td>w3</td>
<td>25/7</td>
<td>11/14</td>
<td>-1/7</td>
<td>-1/7</td>
<td>-1/7</td>
<td>-1/7</td>
</tr>
<tr>
<td>w5</td>
<td>4/7</td>
<td>-3/14</td>
<td>6/7</td>
<td>6/7</td>
<td>6/7</td>
<td>6/7</td>
</tr>
<tr>
<td>x1</td>
<td>11/7</td>
<td>-3/14</td>
<td>-1/7</td>
<td>-1/7</td>
<td>-1/7</td>
<td>-1/7</td>
</tr>
</tbody>
</table>

- **Optimal!**
- Click [here](#) to practice the simplex method on problems that may have infeasible first dictionaries.
- For instructions, click [here](#).
Degeneracy

Definitions.

A dictionary is degenerate if one or more “rhs”-value vanishes.

Example:

\[
\begin{align*}
\zeta &= 6 + w_3 + 5x_2 + 4w_1 \\
x_3 &= 1 - 2w_3 - 2x_2 + 3w_1 \\
w_2 &= 4 + w_3 + x_2 - 3w_1 \\
x_1 &= 3 - 2w_3 \\
w_4 &= 2 + w_3 - w_1 \\
w_5 &= 0 - x_2 + w_1
\end{align*}
\]

A pivot is degenerate if the objective function value does not change.

Examples (based on above dictionary):

1. If \(x_2\) enters, then \(w_5\) must leave, pivot is degenerate.
2. If \(w_1\) enters, then \(w_2\) must leave, pivot is not degenerate.
Cycling

Definition.
A cycle is a sequence of pivots that returns to the dictionary from which the cycle began.

Note: Every pivot in a cycle must be degenerate. Why?

Pivot Rules.

Definition.
Explicit statement for how one chooses entering and leaving variables (when a choice exists).

Largest-Coefficient Rule.
A common pivot rule for entering variable:

Choose the variable with the largest coefficient in the objective function.

Hope.
Some pivot rule, such as the largest coefficient rule, will be proven never to cycle.
An example that cycles using the following pivot rules:

- entering variable: largest-coefficient rule.
- leaving variable: smallest-index rule.

\[ \zeta = x_1 - 2x_2 - 2x_4 \]
\[
 w_1 = -0.5x_1 + 3.5x_2 + 2x_3 - 4x_4 \\
 w_2 = -0.5x_1 + x_2 + 0.5x_3 - 0.5x_4 \\
 w_3 = 1 - x_1.
\]

Here’s a demo of cycling (ignoring the last constraint)...
$x_1 \Leftrightarrow w_1$:

$\begin{array}{c|cccc|cccc}
& 0 & + & 1 & x_1 & + & -2 & x_2 & + & 0 & x_3 & + & -2 & x_4 \\
\text{obj} & 0 & - & 0 & + & -2 & x_2 & + & 0 & x_3 & + & -2 & x_4 \\
w_1 & 0 & - & 1/2 & x_1 & - & -7/2 & x_2 & - & -2 & x_3 & - & 4 & x_4 \\
w_2 & 0 & - & 0 & - & 0 & x_1 & - & -1 & x_2 & - & -1/2 & x_3 & - & 1/2 & x_4 \\
\end{array}$

$x_2 \Leftrightarrow w_2$:

$\begin{array}{c|cccc|cccc}
& 0 & + & -2 & w_1 & + & 5 & x_2 & + & 4 & x_3 & + & -10 & x_4 \\
\text{obj} & 0 & - & 0 & + & -2 & x_2 & + & 0 & x_3 & + & -2 & x_4 \\
x_1 & 0 & - & 2 & w_1 & - & -7 & x_2 & - & -4 & x_3 & - & 8 & x_4 \\
w_2 & 0 & - & 0 & - & 0 & w_1 & - & 5/2 & x_2 & - & 3/2 & x_3 & - & 7/2 & x_4 \\
\end{array}$

$x_3 \Leftrightarrow x_1$:

$\begin{array}{c|cccc|cccc}
& 0 & + & 0 & w_1 & + & -2 & w_2 & + & 1 & x_3 & + & -3 & x_4 \\
\text{obj} & 0 & - & 0 & + & -2 & w_2 & + & 0 & x_3 & + & -3 & x_4 \\
x_1 & 0 & - & -4/5 & w_1 & - & 14/5 & w_2 & - & 1/5 & x_3 & - & 9/5 & x_4 \\
x_2 & 0 & - & -2/5 & w_1 & - & 2/5 & w_2 & - & 3/5 & x_3 & - & 7/5 & x_4 \\
\end{array}$
$x_4 \Leftrightarrow x_2$:

```
<table>
<thead>
<tr>
<th>obj</th>
<th>+</th>
<th>1</th>
<th></th>
<th>w1</th>
<th>+</th>
<th>-4</th>
<th>w2</th>
<th>+</th>
<th>-1/2</th>
<th>x1</th>
<th>+</th>
<th>-3/2</th>
<th>x2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>-4</td>
<td></td>
<td>-7/4</td>
<td></td>
<td>9/4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>-</td>
<td>1/2</td>
<td></td>
<td>1/2</td>
<td></td>
<td>-2</td>
<td></td>
<td>-3/4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>-</td>
<td>1/2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1/2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

$w_1 \Leftrightarrow x_3$:

```
<table>
<thead>
<tr>
<th>obj</th>
<th>+</th>
<th>-2</th>
<th></th>
<th>x3</th>
<th>+</th>
<th>4</th>
<th>w2</th>
<th>+</th>
<th>3</th>
<th>x1</th>
<th>+</th>
<th>-6</th>
<th>x2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>-8</td>
<td></td>
<td>-7/2</td>
<td></td>
<td>9/2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>-</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>-</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

$w_2 \Leftrightarrow x_4$:

```
<table>
<thead>
<tr>
<th>obj</th>
<th>+</th>
<th>0</th>
<th></th>
<th>x3</th>
<th>+</th>
<th>-2</th>
<th>x4</th>
<th>+</th>
<th>1</th>
<th>x1</th>
<th>+</th>
<th>-2</th>
<th>x2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td></td>
<td>1/2</td>
<td></td>
<td>1/2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>-</td>
<td>-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1/2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>-</td>
<td>-1/2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

*Cycling is rare!* A program that generates random $2 \times 4$ fully degenerate problems was run more than *one billion* times and did not find one example!
Perturbation Method

Whenever a vanishing “rhs” appears perturb it. If there are lots of them, say $k$, perturb them all. Make the perturbations at different scales:

other data $\gg \epsilon_1 \gg \epsilon_2 \gg \cdots \gg \epsilon_k > 0$.

An Example.

Entering variable: $x_2$
Leaving variable: $w_2$
Recall current dictionary:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Current Dictionary</th>
</tr>
</thead>
<tbody>
<tr>
<td>obj</td>
<td>0.0 + 0.0 e1 + 4.0 e2 + 0.0 e3 + 14.0 x1 + -4.0 v2</td>
</tr>
<tr>
<td>v1</td>
<td>0.0 + 1.0 e1 + -1.0 e2 + 0.0 e3 + 2.0 x1 + -1.0 v2</td>
</tr>
<tr>
<td>x2</td>
<td>0.0 + 0.0 e1 + 1.0 e2 + 0.0 e3 + -3.0 x1 + 1.0 v2</td>
</tr>
<tr>
<td>v3</td>
<td>0.0 + 0.0 e1 + 1.0 e2 + 1.0 e3 + 1.0 x1 + -1.0 v2</td>
</tr>
</tbody>
</table>

Entering variable: $x_1$
Leaving variable: $w_3$
Other Pivot Rules

**Smallest Index Rule.**

Choose the variable with the smallest index (the $x$ variables are assumed to be “before” the $w$ variables).

Note: Also known as *Bland’s rule*.

**Random Selection Rule.**

Select at random from the set of possibilities.

**Greatest Increase Rule.**

Pick the entering/leaving pair so as to maximize the increase of the objective function over all other possibilities.

Note: Too much computation.
maximize \( x_1 + 2x_2 + 3x_3 \) 
subject to 
\[
\begin{align*}
    x_1 + 2x_3 & \leq 3 \\
    x_2 + 2x_3 & \leq 2 \\
    x_1, x_2, x_3 & \geq 0.
\end{align*}
\]
Efficiency

Question:

Given a problem of a certain size, how long will it take to solve it?

Two Kinds of Answers:

- **Average Case.** How long for a *typical* problem.
- **Worst Case.** How long for the *hardest* problem.

**Average Case.**

- Mathematically difficult.
- Empirical studies.

**Worst Case.**

- Mathematically tractible.
- Limited value.
Measures of Size

- Number of constraints $m$ and/or number of variables $n$.
- Number of data elements, $mn$.
- Number of nonzero data elements.
- Size, in bytes, of AMPL formulation (model+data).

Measuring Time

Two factors:
- Number of iterations.
- Time per iteration.
Klee–Minty Problem (1972)

maximize \[ \sum_{j=1}^{n} 2^{n-j} x_j \]

subject to \[ 2 \sum_{j=1}^{i-1} 2^{i-j} x_j + x_i \leq 100^{i-1} \quad i = 1, 2, \ldots, n \]
\[ x_j \geq 0 \quad j = 1, 2, \ldots, n. \]

**Example** \( n = 3 \):

maximize \[ 4 x_1 + 2 x_2 + x_3 \]

subj. to \[ x_1 \leq 1 \]
\[ 4 x_1 + x_2 \leq 100 \]
\[ 8 x_1 + 4 x_2 + x_3 \leq 10000 \]
\[ x_1, x_2, x_3 \geq 0. \]
A Distorted Cube

Constraints represent a “minor” distortion to an $n$-dimensional hypercube:

\[
0 \leq x_1 \leq 1 \\
0 \leq x_2 \leq 100 \\
\vdots \\
0 \leq x_n \leq 100^{n-1}.
\]
Analysis

Replace

$$1, 10, 100, 1000, \ldots,$$

with

$$1 = b_1 \ll b_2 \ll b_3 \ll \ldots.$$

Then, make following replacements to rhs:

\[
\begin{align*}
    b_1 &\rightarrow b_1 \\
    b_2 &\rightarrow 2b_1 + b_2 \\
    b_3 &\rightarrow 4b_1 + 2b_2 + b_3 \\
    b_4 &\rightarrow 8b_1 + 4b_2 + 2b_3 + b_4 \\
\vdots
\end{align*}
\]

Hardly a change!

Make a similar constant adjustment to objective function.

Look at the pivot tool version…
Case $n = 3$:

Now, watch the pivots...
Klee–Minty problem shows that:

Largest-coefficient rule can take $2^n - 1$ pivots to solve a problem in $n$ variables and constraints.

For $n = 70$,

$$2^n = 1.2 \times 10^{21}.$$ 

At 1000 iterations per second, this problem will take 40 billion years to solve. The age of the universe is estimated at 15 billion years.

Yet, problems with 10,000 to 100,000 variables are solved routinely every day.

Worst case analysis is just that: worst case.
Complexity

<table>
<thead>
<tr>
<th>n</th>
<th>n^2</th>
<th>n^3</th>
<th>2^n</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>27</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>64</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>25</td>
<td>125</td>
<td>32</td>
</tr>
<tr>
<td>6</td>
<td>36</td>
<td>216</td>
<td>64</td>
</tr>
<tr>
<td>7</td>
<td>49</td>
<td>343</td>
<td>128</td>
</tr>
<tr>
<td>8</td>
<td>64</td>
<td>512</td>
<td>256</td>
</tr>
<tr>
<td>9</td>
<td>81</td>
<td>729</td>
<td>512</td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td>1000</td>
<td>1024</td>
</tr>
<tr>
<td>12</td>
<td>144</td>
<td>1728</td>
<td>4096</td>
</tr>
<tr>
<td>14</td>
<td>196</td>
<td>2744</td>
<td>16384</td>
</tr>
<tr>
<td>16</td>
<td>256</td>
<td>4096</td>
<td>65536</td>
</tr>
<tr>
<td>18</td>
<td>324</td>
<td>5832</td>
<td>262144</td>
</tr>
<tr>
<td>20</td>
<td>400</td>
<td>8000</td>
<td>1048576</td>
</tr>
<tr>
<td>22</td>
<td>484</td>
<td>10648</td>
<td>4194304</td>
</tr>
<tr>
<td>24</td>
<td>576</td>
<td>13824</td>
<td>16777216</td>
</tr>
<tr>
<td>26</td>
<td>676</td>
<td>17576</td>
<td>67108864</td>
</tr>
<tr>
<td>28</td>
<td>784</td>
<td>21952</td>
<td>268435456</td>
</tr>
<tr>
<td>30</td>
<td>900</td>
<td>27000</td>
<td>1073741824</td>
</tr>
</tbody>
</table>

Sorting: fast algorithm = $n \log n$, slow algorithm = $n^2$

Matrix times vector: $n^2$
Matrix times matrix: $n^3$
Matrix inversion: $n^3$

Simplex Method:
- Worst case: $n^22^n$ operations.
- Average case: $n^3$ operations.
- Open question:
  Does there exist a variant of the simplex method whose worst case performance is polynomial?

Linear Programming:
- **Theorem:** There exists an algorithm whose worst case performance is $n^{3.5}$ operations.
Define a random problem:

```matlab
m = ceil(exp(log(400)*rand()));
n = ceil(exp(log(400)*rand()));

A = round(sigma*(randn(m,n)));  
b = round(sigma*rand(m,1));

y = round(sigma*rand(1,m));  
z = round(sigma*rand(1,n));
c = y*A - z;

A = -A;
```

Initialize a few things:

```matlab
iter = 0;
opt = 0;
```
The Main Loop:

while max(c) > eps || min(b) < -eps,
    % pick largest coefficient
    [cj, col] = max(c);
    Acol = A(:,col);

    % select leaving variable
    [t, row] = max(-Acol./b);
    if t < eps,
        opt = -1; % unbounded
        'unbounded'
        break;
    end
    Arow = A(row,:);

    a = A(row,col); % pivot element

    iter = iter+1;
end

The code for a pivot:

A = A - Acol*Arow/a;
A(row,:) = -Arow/a;
A(:,col) = Acol/a;
A(row,col) = 1/a;
brow = b(row);
b = b - Acol*brow/a;
b(row) = -brow/a;
ccol = c(col);
c = c - ccol*Arow/a;
c(col) = ccol/a;
Primal Simplex Method

number of pivots vs. $m+n$
$\text{Primal Simplex Method}$

Minimum of $m$ and $n$

Number of pivots

Iters $= 0.6893 \min(m, n)^{1.3347} \approx \frac{2}{3} \min(m, n)^{4/3}$
Average Case—AMPL Version

Declare parameters:

```ampl
param eps := 1e-9;
param sigma := 30;
param niters := 1000;
param size := 400;

param m;
param n;
param AA {1..size, 1..size};
param bb {1..size};
param cc {1..size};
param A {1..size, 1..size};
param b {1..size};
param c {1..size};
param x {1..size};
param y {1..size};
param Arow {1..size};
param Acol {1..size};
param a;
param brow;
param ccol;
param ii;
param jj;
```

Define a random problem:

```ampl
let m := ceil(exp(log(size)*Uniform01()));
let n := ceil(exp(log(size)*Uniform01()));
let {i in 1..m, j in 1..n} A[i,j] := round(sigma*Normal01());
let {i in 1..m} y[i] := round(sigma*Uniform01());
let {j in 1..n} z[j] := round(sigma*Uniform01());
let {i in 1..m} b[i] := round(sigma*Uniform01());
let {j in 1..n} c[j] := sum {i in 1..m} y[i]*A[i,j] - z[j];
let {i in 1..m, j in 1..n} A[i,j] := -A[i,j];
let {i in 1..m, j in 1..n} AA[i,j] := A[i,j];
let {i in 1..m} bb[i] := b[i];
let {j in 1..n} cc[j] := c[j];
```
The Simplex Method (Phase 2)

repeat while (max {j in 1..n} c[j]) > eps {
    let maxc := 0;
    for {j in 1..n} {
        if (c[j] > maxc) then {
            let maxc := c[j];
            let col := j;
        }
    }
    let minbovera := 1/eps;
    for {i in 1..m} {
        if (A[i,col] < -eps) then {
            if (-b[i]/A[i,col] < minbovera) then {
                let minbovera := -b[i]/A[i,col];
                let row := i;
            }
        }
    }
    if minbovera >= 1/eps then {
        let opt := -1; # unbounded
        display "unbounded";
        break;
    }
}

The code for a pivot:

let {j in 1..n} Arow[j] := A[row,j];
let {i in 1..m} Acol[i] := A[i,col];
let a := A[row,col];
let {i in 1..m, j in 1..n}
let {j in 1..n} A[row,j] := -Arow[j]/a;
let {i in 1..m} A[i,col] := Acol[i]/a;
let A[row,col] := 1/a;
let brow := b[row];
let {i in 1..m}
    b[i] := b[i] - brow*Acol[i]/a;
let b[row] := -brow/a;
let ccol := c[col];
let {j in 1..n}
    c[j] := c[j] - ccol*Arow[j]/a;
let c[col] := ccol/a;
The AMPL code can be found here:

http://orfe.princeton.edu/rvdb/307/lectures/primalsimplex.txt
Duality

Every Problem:

\[
\begin{align*}
\text{maximize} & \quad c^T x \\
\text{subject to} & \quad Ax \leq b \\
& \quad x \geq 0.
\end{align*}
\]

Has a Dual:

\[
\begin{align*}
\text{minimize} & \quad b^T y \\
\text{subject to} & \quad A^T y \geq c \\
& \quad y \geq 0.
\end{align*}
\]

Original problem is called the \textit{primal problem}.

A problem is defined by its data (notation used for the variables is arbitrary).

Dual in “Standard” Form:

\[
\begin{align*}
\text{maximize} & \quad -b^T y \\
\text{subject to} & \quad -A^T y \leq -c \\
& \quad y \geq 0.
\end{align*}
\]

Dual is \textit{negative transpose} of primal.

\textbf{Theorem 1} \quad \textit{Dual of dual is primal.}
Theorem 2  If $x$ is feasible for the primal and $y$ is feasible for the dual, then

$$c^T x \leq b^T y.$$ 

Proof.

$$c^T x \leq (A^T y)^T x \leq y^T Ax \leq (Ax)^T y \leq b^T y$$

(because $c \leq A^T y$ and $x \geq 0$)

(because $Ax \leq b$ and $y \geq 0$).
Gap or No Gap?

An important question:

Is there a gap between the largest primal value and the smallest dual value?

Answer is provided by the Strong Duality Theorem (coming later).
Simplex Method and Duality

**A Primal Problem:**

\[
\begin{align*}
\text{obj} &= 0 + (-3) x_1 + 2 x_2 + 1 x_3 \\
\text{w}_1 &= 0 - 0 x_1 - (-1) x_2 - 2 x_3 \\
\text{w}_2 &= 3 - (-3) x_1 - 4 x_2 - 1 x_3 \\
\end{align*}
\]

**Its Dual:**

\[
\begin{align*}
\text{obj} &= 0 + 0 y_1 + (-3) y_2 \\
\text{z}_1 &= 3 - 0 y_1 - 3 y_2 \\
\text{z}_2 &= -2 - 1 y_1 - (-4) y_2 \\
\text{z}_3 &= -1 - (-2) y_1 - (-1) y_2 \\
\end{align*}
\]

**Notes:**

- Dual is negative transpose of primal.
- Primal is feasible, dual is not.

Use primal to choose pivot: \(x_2\) enters, \(w_2\) leaves.
Make analogous pivot in dual: \(z_2\) leaves, \(y_2\) enters.
Second Iteration

After First Pivot:

Primal (feasible):

\[
\begin{align*}
\text{obj} &= \frac{3}{2} + \frac{-3}{2}x_1 + \frac{-1}{2}w_2 + \frac{1}{2}x_3 \\
w_1 &= \frac{3}{4} - \frac{-3}{4}x_1 - \frac{1}{4}w_2 - \frac{9}{4}x_3 \\
x_2 &= \frac{3}{4} - \frac{-3}{4}x_1 - \frac{1}{4}w_2 - \frac{1}{4}x_3
\end{align*}
\]

Dual (still not feasible):

\[
\begin{align*}
\text{obj} &= \frac{-3}{2} + \frac{-3}{4}y_1 + \frac{-3}{4}z_2 \\
z_1 &= \frac{3}{2} - \frac{3}{4}y_1 - \frac{3}{4}z_2 \\
y_2 &= \frac{1}{2} - \frac{-1}{4}y_1 - \frac{-1}{4}z_2 \\
z_3 &= \frac{-1}{2} - \frac{-9}{4}y_1 - \frac{-1}{4}z_2
\end{align*}
\]

Note: *negative transpose property intact.*

Again, use primal to pick pivot: $x_3$ enters, $w_1$ leaves.

Make analogous pivot in dual: $z_3$ leaves, $y_1$ enters.
After Second Iteration

Primal:

- Is *optimal*.

Dual:

- Negative transpose property remains intact.
- Is *optimal*.

Conclusion

Simplex method applied to primal problem (two phases, if necessary), solves both the primal and the dual.
Dual Simplex Method  (used when dual is feasible)

**Primal Dictionary:**
\[
\begin{align*}
\zeta &= 0 + c^T x \\
w &= b - Ax
\end{align*}
\]

**Dual Dictionary:**
\[
\begin{align*}
-\xi &= 0 - b^T y \\
z &= -c + A^T y
\end{align*}
\]

**Entering Variable.** Choose an index $i$ for which $b_i < 0$. Variable $y_i$ is the entering variable.

**Leaving Variable.** Let $y_i$ increase while holding all other $y_k$’s at zero. Figure out which slack variable hits zero first. Let’s say it’s $z_j$.

**Dual Pivot.** In dual problem, $y_i$ is the entering variable and $z_j$ is the leaving variable.

**Primal Pivot.** The corresponding pivot in primal problem has $w_i$ as the leaving variable and $x_j$ as the entering variable.

After the pivot, the primal and dual are still negative transposes of each other.

Continue improving the dual problem until reaching optimality.

At optimality, $\tilde{b} \geq 0$ and $\tilde{c} \leq 0$.

Hence, solution associated with primal dictionary is optimal too.
Dual Simplex Method

When: dual feasible, primal infeasible (i.e., pink on the left, not on top).

An Example. Showing both primal and dual dictionaries:

Looking at dual dictionary: $y_2$ enters, $z_2$ leaves.

On the primal dictionary: $w_2$ leaves, $x_2$ enters.

After pivot...
Dual Simplex Method: Second Pivot

Going in, we have:

Looking at dual: $y_1$ enters, $z_4$ leaves.

Looking at primal: $w_1$ leaves, $x_4$ enters.
Dual Simplex Method Pivot Rule

Refering to the primal dictionary:

- Pick leaving variable from those rows that are *infeasible*.
- Pick entering variable from a box with a negative value and which can be increased the least (on the dual side).

Next primal dictionary shown on next page...
Going in, we have:

<table>
<thead>
<tr>
<th></th>
<th>obj</th>
<th>2.7143</th>
<th>-15.7143</th>
<th>-1.4286</th>
<th>3.0</th>
<th>0.0</th>
<th>-2.2857</th>
<th>-0.1429</th>
<th>-0.5714</th>
<th>x1</th>
<th>x2</th>
<th>x3</th>
<th>v1</th>
</tr>
</thead>
<tbody>
<tr>
<td>x4</td>
<td>-5.1429</td>
<td>-0.1429</td>
<td>-0.2857</td>
<td>0.0</td>
<td>-0.4286</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x2</td>
<td>-0.1429</td>
<td>-0.5714</td>
<td>-0.1429</td>
<td>0.0</td>
<td>0.2857</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>v3</td>
<td>4.0</td>
<td>1.0</td>
<td>3.0</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Which variable must leave and which must enter?

See next page...
Answer is: $x_2$ leaves, $x_1$ enters.

Resulting dictionary is OPTIMAL:

<table>
<thead>
<tr>
<th></th>
<th>obj</th>
<th>x4</th>
<th>x1</th>
<th>w3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-17.0</td>
<td>2.75</td>
<td>0.25</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>-9.0</td>
<td>-0.25</td>
<td>-1.75</td>
<td>-7.0</td>
</tr>
<tr>
<td></td>
<td>-1.0</td>
<td>x2</td>
<td>x2</td>
<td>x2</td>
</tr>
<tr>
<td></td>
<td>-0.25</td>
<td>-0.25</td>
<td>0.25</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>0.0</td>
<td>w2</td>
<td>w2</td>
<td>w2</td>
</tr>
<tr>
<td></td>
<td>0.0</td>
<td>x3</td>
<td>x3</td>
<td>x3</td>
</tr>
<tr>
<td></td>
<td>-4.0</td>
<td>w1</td>
<td>w1</td>
<td>w1</td>
</tr>
</tbody>
</table>
**Dual-Based Phase I Method**

Example:

| obj  | 0.0 | + | -4.0 | x1 | + | 2.0 | x2 | + | 3.0 | x3 |
|------|-----|+ | -1.0 | x1 | - | -1.0 | x2 | - | -4.0 | x3 |
| w1   | 0.0 | + | 1.0  | - | 2.0 | x1 | - | -1.0 | x2 | - | 3.0 | x3 |
| w2   | 0.0 | + | 1.0  | - | 3.0 | x1 | - | -3.0 | x2 | - | -4.0 | x3 |
| w3   | -3.0| + | 1.0  | - | 1.0 | x1 | - | -1.0 | x2 | - | 1.0 | x3 |
| w4   | -1.0| + | 1.0  | - | 2.0 | x1 | - | 0.0  | x2 | - | 0.0 | x3 |

Notes:
- Two objective functions: the true objective (on top), and a fake one (below it).
- For *Phase I*, use the fake objective—it’s dual feasible.
- Two right-hand sides: the real one (on the left) and a fake (on the right).
- Ignore the fake right-hand side—we’ll use it in another algorithm later.

*Phase I—First Pivot*: $w_3$ leaves, $x_1$ enters.

After first pivot...
Dual-Based Phase I Method—Second Pivot

Current dictionary:

Dual pivot: $w_2$ leaves, $x_2$ enters.

After pivot:
Dual-Based Phase I Method—Third Pivot

Current dictionary:

Dual pivot:  
\( w_1 \) leaves,  
\( w_2 \) enters.

After pivot:  
It’s feasible!
Fourth Pivot—Phase II

Current dictionary:

It's feasible.

Ignore fake objective.

Use the real thing (top row).

Primal pivot: $x_3$ enters, $w_4$ leaves.
After pivot:

<table>
<thead>
<tr>
<th></th>
<th>obj</th>
<th>w2</th>
<th>x2</th>
<th>x1</th>
<th>x3</th>
</tr>
</thead>
<tbody>
<tr>
<td>obj</td>
<td>6.75</td>
<td>11.25</td>
<td>3.25</td>
<td>0.5</td>
<td>0.75</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-6.5</td>
<td>-1.5</td>
<td>0.0</td>
<td>-0.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6.25</td>
<td>0.25</td>
<td>-0.5</td>
<td>0.75</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Problem is *unbounded*!
Conclusion on previous slide is the essence of the strong duality theorem which we now state:

**Theorem 3**  
*If the primal problem has an optimal solution,*

\[ x^* = (x_1^*, x_2^*, \ldots, x_n^*) , \]

*then the dual also has an optimal solution,*

\[ y^* = (y_1^*, y_2^*, \ldots, y_m^*) , \]

*and*

\[ \sum_j c_j x_j^* = \sum_i b_i y_i^* . \]

---

Paraphrase:

If primal has an optimal solution, then there is no duality gap.
Duality Gap

Four possibilities:

- Primal optimal, dual optimal (no gap).
- Primal unbounded, dual infeasible (no gap).
- Primal infeasible, dual unbounded (no gap).
- Primal infeasible, dual infeasible (infinite gap).

Example of infinite gap:

\[
\begin{align*}
\text{maximize} & \quad 2x_1 - x_2 \\
\text{subject to} & \quad x_1 - x_2 \leq 1 \\
& \quad -x_1 + x_2 \leq -2 \\
& \quad x_1, x_2 \geq 0.
\end{align*}
\]
Complementary Slackness

Theorem 4  At optimality, we have

\[ x_j z_j = 0, \quad \text{for} \; j = 1, 2, \ldots, n, \]
\[ w_i y_i = 0, \quad \text{for} \; i = 1, 2, \ldots, m. \]
Proof

Recall the proof of the Weak Duality Theorem:

\[
\sum_j c_j x_j \leq \sum_j (c_j + z_j) x_j = \sum_j \left( \sum_i y_i a_{ij} \right) x_j = \sum_{ij} y_i a_{ij} x_j
\]

\[
= \sum_i \left( \sum_j a_{ij} x_j \right) y_i = \sum_i (b_i - w_i) y_i \leq \sum_i b_i y_i,
\]

The inequalities come from the fact that

\[
x_j z_j \geq 0, \quad \text{for all } j,
\]
\[
w_i y_i \geq 0, \quad \text{for all } i.
\]

By Strong Duality Theorem, the inequalities are equalities at optimality.
Parametric Self-Dual Simplex Method

An Example

\[
\begin{align*}
\text{maximize} & \quad -3x_1 + 11x_2 + 2x_3 \\
\text{subj. to} & \quad -x_1 + 3x_2 \leq 5 \\
& \quad 3x_1 + 3x_2 \leq 4 \\
& \quad 3x_2 + 2x_3 \leq 6 \\
& \quad -3x_1 - 5x_3 \leq -4 \\
& \quad x_1, x_2, x_3 \geq 0.
\end{align*}
\]

Initial Dictionary:

\[
\begin{align*}
\zeta &= -3x_1 + 11x_2 + 2x_3 \\
w_1 &= 5 + x_1 - 3x_2 \\
w_2 &= 4 - 3x_1 - 3x_2 \\
w_3 &= 6 - 3x_2 - 2x_3 \\
w_4 &= -4 + 3x_1 + 5x_3
\end{align*}
\]

Note: neither primal nor dual feasible.
Introduce a parameter \( \mu \) and perturb:

\[
\zeta = -3x_1 + 11x_2 + 2x_3 - \mu x_1 - \mu x_2 - \mu x_3
\]

\[
w_1 = 5 + \mu + x_1 - 3x_2
\]
\[
w_2 = 4 + \mu - 3x_1 - 3x_2
\]
\[
w_3 = 6 + \mu - 3x_2 - 2x_3
\]
\[
w_4 = -4 + \mu + 3x_1 + 5x_3
\]

For \( \mu \) large, dictionary is \textit{optimal}.

Question: For which \( \mu \) values is dictionary optimal?
Answer:

\[-3 - \mu \leq 0 \]
\[11 - \mu \leq 0 \quad (*)\]
\[2 - \mu \leq 0 \quad (*)\]
\[5 + \mu \geq 0 \]
\[4 + \mu \geq 0 \]
\[6 + \mu \geq 0 \]
\[\text{Note: only those marked with (*) give inequalities that omit } \mu = 0.\]

Tightest:

\[\mu \geq 11\]

Achieved by: objective row perturbation on \(x_2\).

Let \(x_2\) enter.
Who Leaves?

Do ratio test using current lowest $\mu$ value, i.e. $\mu = 11$:

$$
\begin{align*}
5 + 11 - 3x_2 & \geq 0 \\
4 + 11 - 3x_2 & \geq 0 \\
6 + 11 - 3x_2 & \geq 0 \\
-4 + 11 & \geq 0
\end{align*}
$$

Tightest:

$$
4 + 11 - 3x_2 \geq 0.
$$

Achieved by: constraint containing basic variable $w_2$.

Let $w_2$ leave.
After the pivot:

\[ \zeta = 14.67 \]
\[ -14x_1 - 3.67w_2 + 2x_3 + 0.33\mu w_2 - \mu x_3 \]

\[ w_1 = 1 + 4x_1 + w_2 \]
\[ x_2 = 1.33 + 0.33\mu - x_1 - 0.33w_2 \]
\[ w_3 = 2 + 3x_1 + w_2 - 2x_3 \]
\[ w_4 = -4 + \mu + 3x_1 + 5x_3 \]
Second Pivot

Using the *advanced* pivot tool, the current dictionary is:

![Dictionary Table]

Note: the parameter $\mu$ is not shown. *But it is there!*

Question: For which $\mu$ values is dictionary optimal? Answer:

$$
\begin{align*}
-14 & \leq 0 \\
-3.67 + 0.33\mu & \leq 0 \\
2 - \mu & \leq 0 *
\end{align*}
\quad
\begin{align*}
1 & \geq 0 \\
1.33 + 0.33\mu & \geq 0 \\
2 & \geq 0 \\
-4 + \mu & \geq 0 *
\end{align*}
$$

Tightest lower bound: $\mu \geq 4$.

Achieved by: constraint containing basic variable $w_4$. Let $w_4$ *leave.*
Who shall enter?

Recall the current dictionary:

<table>
<thead>
<tr>
<th>obj</th>
<th>14.6667</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1</td>
<td>+ 1.0</td>
</tr>
<tr>
<td>x2</td>
<td>+ 1.3333</td>
</tr>
<tr>
<td>v3</td>
<td>+ 2.0</td>
</tr>
<tr>
<td>v4</td>
<td>- 4.0</td>
</tr>
</tbody>
</table>

Do *dual-type* ratio test using current lowest \( \mu \) value, i.e. \( \mu = 4 \):

\[
14 + 0 \times 4 - 3y_4 \geq 0 \\
3.67 - 0.33 \times 4 \geq 0 \\
-2 + 1 \times 4 - 5y_4 \geq 0
\]

Tightest: \(-2 + 1 \times 4 - 5y_4 \geq 0\).

Achieved by: objective term containing nonbasic variable \( x_3 \). Let \( x_3 \) enter.
Third Pivot

The current dictionary is:

<table>
<thead>
<tr>
<th>obj</th>
<th>16.2667</th>
<th>+</th>
<th>-15.2</th>
<th>x1</th>
<th>+</th>
<th>-3.667</th>
<th>v2</th>
<th>+</th>
<th>0.4</th>
<th>v4</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1</td>
<td>1.0</td>
<td>+</td>
<td>0.0</td>
<td>-</td>
<td>-4.0</td>
<td>x1</td>
<td>-</td>
<td>-1.0</td>
<td>v2</td>
<td>-</td>
</tr>
<tr>
<td>x2</td>
<td>1.3333</td>
<td>+</td>
<td>0.3333</td>
<td>-</td>
<td>1.0</td>
<td>x1</td>
<td>-</td>
<td>0.333</td>
<td>v2</td>
<td>-</td>
</tr>
<tr>
<td>w3</td>
<td>0.4</td>
<td>+</td>
<td>0.4</td>
<td>-</td>
<td>-4.2</td>
<td>x1</td>
<td>-</td>
<td>-1.0</td>
<td>v2</td>
<td>-</td>
</tr>
<tr>
<td>x3</td>
<td>0.8</td>
<td>+</td>
<td>-0.2</td>
<td>-</td>
<td>0.6</td>
<td>x1</td>
<td>-</td>
<td>0.0</td>
<td>v2</td>
<td>-</td>
</tr>
</tbody>
</table>

| 1.0 | 0.0 | -4.0 | x1 | -1.0 | v2 | -0.2 | v4 |
| 1.0 | 0.0 | -4.0 | x1 | -1.0 | v2 | -0.2 | v4 |
| 0.4 | 0.2 | -0.8 | x1 | -0.4 | v2 | -0.8 | v4 |

Question: For which $\mu$ is dictionary optimal? Answer:

\[-15.2 + 0.6\mu \leq 0 \quad | \quad 1 + 0.33\mu \geq 0\]
\[-3.67 + 0.33\mu \leq 0 \quad | \quad 1.33 + 0.33\mu \geq 0\]
\[0.4 - 0.2\mu \leq 0 \quad * \quad 0.4 + 0.4\mu \geq 0\]

Tightest lower bound: $\mu \geq 2$.

Achieved by: objective term containing nonbasic variable $w_4$. Let $w_4$ enter.
Who shall leave? Recall the current dictionary:

```
<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>obj</td>
<td>16.2667</td>
<td>+</td>
<td>-15.2</td>
<td>x1</td>
<td>+</td>
<td>-3.6667</td>
</tr>
<tr>
<td>v1</td>
<td>1.0</td>
<td>+</td>
<td>0.0</td>
<td>-4.0</td>
<td>x1</td>
<td>-1.0</td>
</tr>
<tr>
<td>x2</td>
<td>1.3333</td>
<td>+</td>
<td>0.3333</td>
<td>-1.0</td>
<td>x1</td>
<td>-0.3333</td>
</tr>
<tr>
<td>v3</td>
<td>0.4</td>
<td>+</td>
<td>0.4</td>
<td>-4.2</td>
<td>x1</td>
<td>-1.0</td>
</tr>
<tr>
<td>x3</td>
<td>0.8</td>
<td>+</td>
<td>-0.2</td>
<td>0.6</td>
<td>x1</td>
<td>0.0</td>
</tr>
</tbody>
</table>
```

Do *primal-type* ratio test using current lowest $\mu$ value, i.e. $\mu = 2$:

1 + 0 * 2 $\geq 0$

1.33 + 0.33 * 2 $\geq 0$

0.4 + 0.4 * 2 - 0.4 $w_4$ $\geq 0$

0.8 - 0.2 * 2 + 0.2 $w_4$ $\geq 0$

Tightest: $0.4 + 0.4 * 2 - 0.4 w_4 \geq 0$.

Achieved by: constraint containing basic variable $w_3$. Let $w_3$ *leave*. 
Fourth Pivot

The current dictionary is:

![Dictionary Table]

It’s **optimal!** Also, the range of \( \mu \) values includes \( \mu = 0 \):

\[
\begin{align*}
-11 & - 1.5\mu & \leq & 0 \\
-2.67 & - 0.167\mu & \leq & 0 \\
-1 & + 0.5\mu & \leq & 0 \\
1 & & \geq & 0 \\
1.33 & + 0.33\mu & \geq & 0 \\
1 & + 1\mu & \geq & 0 \\
1 & & \geq & 0
\end{align*}
\]

That is, \( -1 \leq \mu \leq 2 \).

Range of \( \mu \) values is shown at bottom of pivot tool. Invalid ranges are highlighted in yellow.
Top Ten Reasons to Like this Method

- Freedom to pick perturbation as you like.
- Randomizing perturbation completely solves the degeneracy problem.
- Perturbations don’t have to be “small”.
- In the optimal dictionary, perturbation is completely gone—no need to remove it.
- In some real-world problems, a “natural” perturbation exists (next lecture).
- The average-case performance can be analyzed (lecture after that).

Okay, there are only 6 items in the list. SORRY.
An Example: Structural Optimization

**Forces:** $x_{ij} =$ tension in beam (aka member) $\{i, j\}$.
- $x_{ij} = x_{ji}$.
- Compression = -Tension.

**Force Balance:**

Look at joint 2:

$$x_{12} \begin{bmatrix} -1 \\ 0 \end{bmatrix} + x_{23} \begin{bmatrix} -0.6 \\ 0.8 \end{bmatrix} + x_{24} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = -\begin{bmatrix} b_1^1 \\ b_2^2 \end{bmatrix}$$

**Notations:**

$$p_i = \text{position vector for joint } i$$

$$u_{ij} = \frac{p_j - p_i}{\|p_j - p_i\|} \quad (\text{Note } u_{ji} = -u_{ij})$$

**Constraints:**

$$\sum_{j: \{i, j\} \in A} u_{ij} x_{ij} = -b_i \quad i = 1, \ldots, m.$$
Matrix Form

\[ Ax = -b \]

\[ x^T = [ x_{12} \quad x_{13} \quad x_{14} \quad x_{23} \quad x_{24} \quad x_{34} \quad x_{35} \quad x_{45} ] \]

\[
A = \begin{bmatrix}
1 & 0 & .6 \\
0 & 1 & .8 \\
-1 & 0 & 0 \\
0 & -1 & .8 \\
-.6 & .8 & 0 \\
-.8 & 0 & 0 \\
-.6 & -1 & .8 \\
-.8 & .8 & 0 \\
\end{bmatrix}, \quad b = \begin{bmatrix} b_1^1 \\ b_1^2 \\ b_2^1 \\ b_2^2 \\ b_3^1 \\ b_3^2 \\ b_4^1 \\ b_4^2 \\ b_5^1 \\ b_5^2 \end{bmatrix}.
\]

Notes:

\[ ||u_{ij}|| = ||u_{ji}|| = 1. \]

\[ u_{ij} = -u_{ji}. \]

- Each column contains a \( u_{ij} \), a \( u_{ji} \), and rest are zero.
- In one dimension, exactly a node-arc incidence matrix.
Minimum Weight Structural Design

minimize \[ \sum_{\{i,j\} \in A} l_{ij} |x_{ij}| \]
subject to \[ \sum_{j: \{i,j\} \in A} u_{ij} x_{ij} = -b_i \quad i = 1, 2, \ldots, m. \]

Not quite an LP.
Use a common trick:

\[ x_{ij} = x_{ij}^+ - x_{ij}^-, \quad x_{ij}^+, x_{ij}^- \geq 0 \]
\[ |x_{ij}| = x_{ij}^+ + x_{ij}^- \]

Reformulated as an LP:

minimize \[ \sum_{\{i,j\} \in A} (l_{ij} x_{ij}^+ + l_{ij} x_{ij}^-) \]
subject to \[ \sum_{j: \{i,j\} \in A} (u_{ij} x_{ij}^+ - u_{ij} x_{ij}^-) = -b_i \quad i = 1, 2, \ldots, m \]
\[ x_{ij}^+, x_{ij}^- \geq 0 \quad \{i, j\} \in A. \]
AMPL Model

param m default 26;   # must be even
param n default 39;

set X := {0..n};
set Y := {0..m};

set NODES := X cross Y;   # A lattice of Nodes
set ANCHORS within NODES := { x in X, y in Y :
    x == 0 && y >= floor(m/3) && y <= m-floor(m/3) };

param xload {(x,y) in NODES: (x,y) not in ANCHORS} default 0;
param yload {(x,y) in NODES: (x,y) not in ANCHORS} default 0;

param gcd {x in -n..n, y in -n..n} :=
    (if x < 0 then gcd[-x,y] else
    (if x == 0 then y else
    (if y < x then gcd[y,x] else
    (gcd[y mod x, x])
    )));

set ARCS := { (xi,yi) in NODES, (xj,yj) in NODES:
    abs( xj-xi ) <= 3    &&
    abs(yj-yi) <=3   &&
    abs(gcd[ xj-xi, yj-yi ]) == 1 &&
    ( xi > xj || (xi == xj && yi > yj) )
    };

param length {(xi,yi,xj,yj) in ARCS} := sqrt( (xj-xi)^2 + (yj-yi)^2 );
var comp {ARCS} >= 0;
var tens {ARCS} >= 0;

minimize volume:
    sum { (xi,yi,xj,yj) in ARCS }
        length[xi,yi,xj,yj] * (comp[xi,yi,xj,yj] + tens[xi,yi,xj,yj]);

subject to Xbalance {(xi,yi) in NODES: (xi,yi) not in ANCHORS}:
    sum { (xi,yi,xj,yj) in ARCS }
        ((xj-xi)/length[xi,yi,xj,yj]) * (comp[xi,yi,xj,yj]-tens[xi,yi,xj,yj])
    +
    sum { (xk,yk,xi,yi) in ARCS }
        ((xi-xk)/length[xk,yk,xi,yi]) * (tens[xk,yk,xi,yi]-comp[xk,yk,xi,yi])
    =
    xload[xi,yi];

subject to Ybalance {(xi,yi) in NODES: (xi,yi) not in ANCHORS}:
    sum { (xi,yi,xj,yj) in ARCS }
        ((yj-yi)/length[xi,yi,xj,yj]) * (comp[xi,yi,xj,yj]-tens[xi,yi,xj,yj])
    +
    sum { (xk,yk,xi,yi) in ARCS }
        ((yi-yk)/length[xk,yk,xi,yi]) * (tens[xk,yk,xi,yi]-comp[xk,yk,xi,yi])
    =
    yload[xi,yi];

let yload[n,m/2] := -1;
solve;
The Michell Bracket (1904)

Constraints: 2,138
Variables: 31,034
Time: 193 secs

Click [here](#) for parametric self-dual simplex method animation tool.
Click [here](#) for affine-scaling method animation tool.
Regression

- Means and Medians
- Least Squares Regression
- Least Absolute Deviation (LAD) Regression
- LAD via LP
- Average Complexity of Parametric Self-Dual Simplex Method
Consider 1995 Adjusted Gross Incomes on Individual Tax Returns:

<table>
<thead>
<tr>
<th>Individual</th>
<th>AGI</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b_1$</td>
<td>$25,462$</td>
</tr>
<tr>
<td>$b_2$</td>
<td>$45,110$</td>
</tr>
<tr>
<td>$b_3$</td>
<td>$15,505$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$b_{m-1}$</td>
<td>$33,265$</td>
</tr>
<tr>
<td>$b_m$</td>
<td>$75,420$</td>
</tr>
</tbody>
</table>

Real summary data is shown on the next slide...
Table 1.--1995, Individual Income Tax Returns

[All figures are estimates based on samples--money amounts are in thousands of dollars]

<table>
<thead>
<tr>
<th>Size of adjusted gross income</th>
<th>Number of returns</th>
<th>Adjusted gross income</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(1)</td>
<td>(2)</td>
</tr>
<tr>
<td>All returns</td>
<td>118,218,327</td>
<td>4,189,353,615</td>
</tr>
<tr>
<td>No adjusted gross income</td>
<td>944,141</td>
<td>55,253,648</td>
</tr>
<tr>
<td>$1 under $5,000</td>
<td>14,646,131</td>
<td>37,604,828</td>
</tr>
<tr>
<td>$5,000 under $10,000</td>
<td>13,982,404</td>
<td>104,603,365</td>
</tr>
<tr>
<td>$10,000 under $15,000</td>
<td>13,562,088</td>
<td>169,317,443</td>
</tr>
<tr>
<td>$15,000 under $20,000</td>
<td>11,385,632</td>
<td>198,418,324</td>
</tr>
<tr>
<td>$20,000 under $25,000</td>
<td>9,970,099</td>
<td>223,400,219</td>
</tr>
<tr>
<td>$25,000 under $30,000</td>
<td>7,847,862</td>
<td>215,200,244</td>
</tr>
<tr>
<td>$30,000 under $40,000</td>
<td>12,380,339</td>
<td>430,491,242</td>
</tr>
<tr>
<td>$40,000 under $50,000</td>
<td>9,098,760</td>
<td>406,638,597</td>
</tr>
<tr>
<td>$50,000 under $75,000</td>
<td>13,679,023</td>
<td>828,349,278</td>
</tr>
<tr>
<td>$75,000 under $100,000</td>
<td>5,374,489</td>
<td>458,505,650</td>
</tr>
<tr>
<td>$100,000 under $200,000</td>
<td>4,074,852</td>
<td>532,030,480</td>
</tr>
<tr>
<td>$200,000 under $500,000</td>
<td>1,007,136</td>
<td>292,117,517</td>
</tr>
<tr>
<td>$500,000 under $1,000,000</td>
<td>178,374</td>
<td>120,347,093</td>
</tr>
<tr>
<td>$1,000,000 or more</td>
<td>86,998</td>
<td>227,582,987</td>
</tr>
<tr>
<td>Taxable returns</td>
<td>89,252,989</td>
<td>4,007,580,441</td>
</tr>
<tr>
<td>Nontaxable returns</td>
<td>28,965,338</td>
<td>181,773,174</td>
</tr>
</tbody>
</table>
Means and Medians

Median:

\[ \hat{x} = b^{1+m/2} \approx 22,500. \]

Mean:

\[ \bar{x} = \frac{1}{m} \sum_{i=1}^{m} b_i = \frac{4,189,353,615,000}{118,218,327} = 35,437. \]
Mean’s Connection with Optimization

\[ \bar{x} = \arg \min_x \sum_{i=1}^{m} (x - b_i)^2. \]

Proof:

\[
\begin{align*}
  f(x) &= \sum_{i=1}^{m} (x - b_i)^2 \\
  f'(x) &= \sum_{i=1}^{m} 2(x - b_i) \\
  f'(&\bar{x}) = 0 \quad \implies \quad \bar{x} = \frac{1}{m} \sum_{i=1}^{m} b_i \\
  \lim_{x \to \pm\infty} f(x) &= +\infty \quad \implies \quad \bar{x} \text{ is a minimum}
\end{align*}
\]
Median’s Connection with Optimization

\[ \hat{x} = \arg\min_x \sum_{i=1}^{m} |x - b_i|. \]

Proof:

\[ f(x) = \sum_{i=1}^{m} |x - b_i| \]

\[ f'(x) = \sum_{i=1}^{m} \text{sgn}(x - b_i) \quad \text{where } \text{sgn}(x) = \begin{cases} 
1 & x > 0 \\
0 & x = 0 \\
-1 & x < 0 
\end{cases} \]

\[ = (\# \text{ of } b_i's \text{ smaller than } x) - (\# \text{ of } b_i's \text{ larger than } x). \]

If \( m \) is odd:
Parametric Self-Dual Simplex Method: Data

<table>
<thead>
<tr>
<th>Name</th>
<th>m</th>
<th>n</th>
<th>iters</th>
</tr>
</thead>
<tbody>
<tr>
<td>25fv47</td>
<td>777</td>
<td>1545</td>
<td>5089</td>
</tr>
<tr>
<td>80bau3b</td>
<td>2021</td>
<td>9195</td>
<td>10514</td>
</tr>
<tr>
<td>adlittle</td>
<td>53</td>
<td>96</td>
<td>141</td>
</tr>
<tr>
<td>afiro</td>
<td>25</td>
<td>32</td>
<td>16</td>
</tr>
<tr>
<td>agg2</td>
<td>481</td>
<td>301</td>
<td>204</td>
</tr>
<tr>
<td>agg3</td>
<td>481</td>
<td>301</td>
<td>193</td>
</tr>
<tr>
<td>bandm</td>
<td>224</td>
<td>379</td>
<td>1139</td>
</tr>
<tr>
<td>beaconfd</td>
<td>111</td>
<td>172</td>
<td>113</td>
</tr>
<tr>
<td>blend</td>
<td>72</td>
<td>83</td>
<td>117</td>
</tr>
<tr>
<td>bnl1</td>
<td>564</td>
<td>1113</td>
<td>2580</td>
</tr>
<tr>
<td>bnl2</td>
<td>1874</td>
<td>3134</td>
<td>6381</td>
</tr>
<tr>
<td>boeing1</td>
<td>298</td>
<td>373</td>
<td>619</td>
</tr>
<tr>
<td>boeing2</td>
<td>125</td>
<td>143</td>
<td>168</td>
</tr>
<tr>
<td>bore3d</td>
<td>138</td>
<td>188</td>
<td>227</td>
</tr>
<tr>
<td>brandy</td>
<td>123</td>
<td>205</td>
<td>585</td>
</tr>
<tr>
<td>czprob</td>
<td>689</td>
<td>2770</td>
<td>2635</td>
</tr>
<tr>
<td>d6cube</td>
<td>403</td>
<td>6183</td>
<td>5883</td>
</tr>
<tr>
<td>degen2</td>
<td>444</td>
<td>534</td>
<td>1421</td>
</tr>
<tr>
<td>degen3</td>
<td>1503</td>
<td>1818</td>
<td>6398</td>
</tr>
<tr>
<td>e226</td>
<td>162</td>
<td>260</td>
<td>598</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>m</th>
<th>n</th>
<th>iters</th>
</tr>
</thead>
<tbody>
<tr>
<td>nesm</td>
<td>646</td>
<td>2740</td>
<td>5829</td>
</tr>
<tr>
<td>recipe</td>
<td>74</td>
<td>136</td>
<td>80</td>
</tr>
<tr>
<td>sc105</td>
<td>104</td>
<td>103</td>
<td>92</td>
</tr>
<tr>
<td>sc205</td>
<td>203</td>
<td>202</td>
<td>191</td>
</tr>
<tr>
<td>sc50a</td>
<td>49</td>
<td>48</td>
<td>46</td>
</tr>
<tr>
<td>sc50b</td>
<td>48</td>
<td>48</td>
<td>53</td>
</tr>
<tr>
<td>scagr25</td>
<td>347</td>
<td>499</td>
<td>1336</td>
</tr>
<tr>
<td>scagr7</td>
<td>95</td>
<td>139</td>
<td>339</td>
</tr>
<tr>
<td>scfxm1</td>
<td>282</td>
<td>439</td>
<td>531</td>
</tr>
<tr>
<td>scfxm2</td>
<td>564</td>
<td>878</td>
<td>1197</td>
</tr>
<tr>
<td>scfxm3</td>
<td>846</td>
<td>1317</td>
<td>1886</td>
</tr>
<tr>
<td>scorpion</td>
<td>292</td>
<td>331</td>
<td>411</td>
</tr>
<tr>
<td>scrs8</td>
<td>447</td>
<td>1131</td>
<td>783</td>
</tr>
<tr>
<td>scsd1</td>
<td>77</td>
<td>760</td>
<td>172</td>
</tr>
<tr>
<td>scsd6</td>
<td>147</td>
<td>1350</td>
<td>494</td>
</tr>
<tr>
<td>scsd8</td>
<td>397</td>
<td>2750</td>
<td>1548</td>
</tr>
<tr>
<td>sctap1</td>
<td>284</td>
<td>480</td>
<td>643</td>
</tr>
<tr>
<td>sctap2</td>
<td>1033</td>
<td>1880</td>
<td>1037</td>
</tr>
<tr>
<td>sctap3</td>
<td>1408</td>
<td>2480</td>
<td>1339</td>
</tr>
<tr>
<td>seba</td>
<td>449</td>
<td>896</td>
<td>766</td>
</tr>
<tr>
<td>Name</td>
<td>$m$</td>
<td>$n$</td>
<td>iters</td>
</tr>
<tr>
<td>----------</td>
<td>------</td>
<td>------</td>
<td>-------</td>
</tr>
<tr>
<td>etamacro</td>
<td>334</td>
<td>542</td>
<td>1580</td>
</tr>
<tr>
<td>fff800</td>
<td>476</td>
<td>817</td>
<td>1029</td>
</tr>
<tr>
<td>finnis</td>
<td>398</td>
<td>541</td>
<td>680</td>
</tr>
<tr>
<td>fit1d</td>
<td>24</td>
<td>1026</td>
<td>925</td>
</tr>
<tr>
<td>fit1p</td>
<td>627</td>
<td>1677</td>
<td>15284</td>
</tr>
<tr>
<td>forplan</td>
<td>133</td>
<td>415</td>
<td>576</td>
</tr>
<tr>
<td>ganges</td>
<td>1121</td>
<td>1493</td>
<td>2716</td>
</tr>
<tr>
<td>greenbea</td>
<td>1948</td>
<td>4131</td>
<td>21476</td>
</tr>
<tr>
<td>grow15</td>
<td>300</td>
<td>645</td>
<td>681</td>
</tr>
<tr>
<td>grow22</td>
<td>440</td>
<td>946</td>
<td>999</td>
</tr>
<tr>
<td>grow7</td>
<td>140</td>
<td>301</td>
<td>322</td>
</tr>
<tr>
<td>israel</td>
<td>163</td>
<td>142</td>
<td>209</td>
</tr>
<tr>
<td>kb2</td>
<td>43</td>
<td>41</td>
<td>63</td>
</tr>
<tr>
<td>lotfi</td>
<td>134</td>
<td>300</td>
<td>242</td>
</tr>
<tr>
<td>maros</td>
<td>680</td>
<td>1062</td>
<td>2998</td>
</tr>
</tbody>
</table>
A Regression Model for Algorithm Efficiency

**Observed Data:**

\[
\begin{align*}
    t & = \text{# of iterations} \\
    m & = \text{# of constraints} \\
    n & = \text{# of variables}
\end{align*}
\]

**Model:**

\[
t \approx 2^\alpha (m + n)^\beta
\]

**Linearization:** Take logs:

\[
\log t = \alpha \log 2 + \beta \log(m + n) + \epsilon
\]

↑ error
Solve several instances (say $k$ of them):

\[
\begin{bmatrix}
\log t_1 \\
\log t_2 \\
\vdots \\
\log t_k
\end{bmatrix}
= 
\begin{bmatrix}
\log 2 \log (m_1 + n_1) \\
\log 2 \log (m_2 + n_2) \\
\vdots \\
\log 2 \log (m_k + n_k)
\end{bmatrix}
\begin{bmatrix}
\alpha \\
\beta
\end{bmatrix}
+ 
\begin{bmatrix}
\epsilon_1 \\
\epsilon_2 \\
\vdots \\
\epsilon_k
\end{bmatrix}
\]

In matrix notation:

\[
b = Ax + \epsilon
\]

**Goal:** Find $x$ that “minimizes” $\epsilon$. 

Regression Model Continued
Least Squares Regression

**Euclidean Distance:** \( \| x \|_2 = \left( \sum_i x_i^2 \right)^{1/2} \)

**Least Squares Regression:** \( \bar{x} = \arg\min_x \| b - Ax \|_2^2 \)

**Calculus:**

\[
f(x) = \| b - Ax \|_2^2 = \sum_i \left( b_i - \sum_j a_{ij} x_j \right)^2
\]

\[
\frac{\partial f}{\partial x_k}(\bar{x}) = \sum_i 2 \left( b_i - \sum_j a_{ij} \bar{x}_j \right) (-a_{ik}) = 0, \quad k = 1, 2, \ldots, n
\]

Rearranging,

\[
\sum_i a_{ik} b_i = \sum_i \sum_j a_{ik} a_{ij} \bar{x}_j, \quad k = 1, 2, \ldots, n
\]

In matrix notation,

\[
A^T b = A^T A \bar{x}
\]

Assuming \( A^T A \) is invertible,

\[
\bar{x} = \left( A^T A \right)^{-1} A^T b
\]
Least Absolute Deviation Regression

**Manhattan Distance:** \[ \|x\|_1 = \sum_i |x_i| \]

**Least Absolute Deviation Regression:** \( \hat{x} = \arg\min_x \|b - Ax\|_1 \)

**Calculus:**

\[
f(x) = \|b - Ax\|_1 = \sum_i \left| b_i - \sum_j a_{ij}x_j \right|
\]

\[
\frac{\partial f}{\partial x_k}(\hat{x}) = \sum_i \frac{b_i - \sum_j a_{ij}\hat{x}_j}{\left| b_i - \sum_j a_{ij}\hat{x}_j \right|} (-a_{ik}) = 0, \quad k = 1, 2, \ldots, n
\]

Rearranging,

\[
\sum_i \frac{a_{ik}b_i}{\epsilon_i(\hat{x})} = \sum_i \sum_j \frac{a_{ik}a_{ij}\hat{x}_j}{\epsilon(\hat{x})}, \quad k = 1, 2, \ldots, n
\]

In matrix notation,

\[
A^T E(\hat{x}) b = A^T E(\hat{x}) A \hat{x}, \quad \text{where } E(\hat{x}) = \text{Diag}(\epsilon(\hat{x}))^{-1}
\]

Assuming \( A^T E(\hat{x}) A \) is invertible,

\[
\hat{x} = \left( A^T E(\hat{x}) A \right)^{-1} A^T E(\hat{x}) b
\]
Least Absolute Deviation Regression—Continued

An implicit equation.

Can be solved using *successive approximations*:

\[
\begin{align*}
x^0 &= 0 \\
x^1 &= \left( A^T E(x^0) A \right)^{-1} A^T E(x^0) b \\
x^2 &= \left( A^T E(x^1) A \right)^{-1} A^T E(x^1) b \\
&\vdots \\
x^{k+1} &= \left( A^T E(x^k) A \right)^{-1} A^T E(x^k) b \\
&\vdots \\
\hat{x} &= \lim_{k \to \infty} x^k
\end{align*}
\]
Least Absolute Deviation Regression via Linear Programming

\[ \min \sum_{i} \left| b_{i} - \sum_{j} a_{ij}x_{j} \right| \]

Equivalent Linear Program:

\[ \min \sum_{i} t_{i} \]

\[ -t_{i} \leq b_{i} - \sum_{j} a_{ij}x_{j} \leq t_{i} \quad i = 1, 2, \ldots, m \]
AMPL Model

param m;
param n;

set I := {1..m};
set J := {1..n};

param A {I,J};
param b {I};

var x{J};
var t{I};

minimize sum_dev:
    sum {i in I} t[i];

subject to lower_bound {i in I}:
    -t[i] <= b[i] - sum {j in J} A[i,j]*x[j];

subject to upper_bound {i in I}:
    b[i] - sum {j in J} A[i,j]*x[j] <= t[i];
Parametric Self-Dual Simplex Method

Thought experiment:

- $\mu$ starts at $\infty$.
- In reducing $\mu$, there are $n + m$ barriers.
- At each iteration, one barrier is passed—the others move about randomly.
- To get $\mu$ to zero, we must on average pass half the barriers.
- Therefore, on average the algorithm should take $(m + n)/2$ iterations.

Using 69 real-world problems from the Netlib suite...

Least Squares Regression:

\[
\begin{bmatrix}
\bar{\alpha} \\
\bar{\beta}
\end{bmatrix} = \begin{bmatrix}
-1.03561 \\
1.05152
\end{bmatrix} \implies T \approx 0.488(m + n)^{1.052}
\]

Least Absolute Deviation Regression:

\[
\begin{bmatrix}
\hat{\alpha} \\
\hat{\beta}
\end{bmatrix} = \begin{bmatrix}
-0.9508 \\
1.0491
\end{bmatrix} \implies T \approx 0.517(m + n)^{1.049}
\]
A log–log plot of $T$ vs. $m + n$ and the $L^1$ and $L^2$ regression lines.
Parametric Self−Dual Simplex Method

\[ \text{iters} = 0.4165(m + n)^{0.9759} \]
Parametric Self–Dual Simplex Method

\[ \text{iters} = 1.4880 \min(m, n)^{1.3434} \]
Lecture 3
Interior Point Methods
and Nonlinear Optimization

Robert J. Vanderbei
April 16, 2012

Machine Learning Summer School
La Palma

http://www.princeton.edu/~rvdb
Example: Basis Pursuit Denoising
$L^1$-Penalized Regression

A trade-off between two objectives:

1. Least squares regression: $\min \frac{1}{2} \|Ax - b\|_2^2$.

2. Sparsity of the solution as encouraged by minimizing $\sum_j |x_j|$.

Trade-off:

$$\min \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|x\|_1.$$

Ideal value for $\lambda$ is unknown.

May wish to try many different values hoping to find a good one.

**Suggestion:**

- Change least-squares regression to least-absolute-value regression,
- formulate the problem as a parametric linear programming problem, and
- solve it for all values of $\lambda$ using the parametric simplex method.

This is an important problem in machine learning.
Interior-Point Methods
What Makes LP Hard?

**Primal**

maximize \( c^T x \)
subject to \( Ax + w = b \)
\( x, w \geq 0 \)

**Dual**

minimize \( b^T y \)
subject to \( A^T y - z = c \)
\( y, z \geq 0 \)

**Complementarity Conditions**

\( x_j z_j = 0 \quad j = 1, 2, \ldots, n \)
\( w_i y_i = 0 \quad i = 1, 2, \ldots, m \)
Can’t write $xz = 0$. The product $xz$ is undefined.

Instead, introduce a new notation:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \implies X = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}$$

Then the complementarity conditions can be written as:

$$XZe = 0$$
$$WYe = 0$$
Optimality Conditions

\[
\begin{align*}
Ax + w &= b \\
A^T y - z &= c \\
ZX e &= 0 \\
WY e &= 0 \\
w, x, y, z &\geq 0
\end{align*}
\]

Ignore (temporarily) the nonnegativities.

2\(n\) + 2\(m\) equations in 2\(n\) + 2\(m\) unknowns.

Solve’em.

Hold on. Not all equations are linear.

*It is the nonlinearity of the complementarity conditions that makes LP fundamentally harder than solving systems of equations.*
The Interior-Point Paradigm

Since we’re ignoring nonnegativities, it’s best to replace complementarity with $\mu$-complementarity:

\[
\begin{align*}
Ax + w &= b \\
A^T y - z &= c \\
ZXe &= \mu e \\
WYe &= \mu e
\end{align*}
\]

Start with an arbitrary (positive) initial guess: $x, y, w, z$.

Introduce *step directions*: $\Delta x, \Delta y, \Delta w, \Delta z$.

Write the above equations for $x + \Delta x, y + \Delta y, w + \Delta w,$ and $z + \Delta z$:

\[
\begin{align*}
A(x + \Delta x) + (w + \Delta w) &= b \\
A^T(y + \Delta y) - (z + \Delta z) &= c \\
(Z + \Delta Z)(X + \Delta X)e &= \mu e \\
(W + \Delta W)(Y + \Delta Y)e &= \mu e
\end{align*}
\]
Rearrange with “delta” variables on left and drop nonlinear terms on left:

\[
\begin{align*}
A \Delta x + \Delta w &= b - Ax - w \\
A^T \Delta y - \Delta z &= c - A^T y + z \\
Z \Delta x + X \Delta z &= \mu e - ZX e \\
W \Delta y + Y \Delta w &= \mu e - WY e
\end{align*}
\]

This is a \emph{linear} system of \(2m + 2n\) equations in \(2m + 2n\) unknowns.

Solve’em.

Dampen the step lengths, if necessary, to maintain positivity.

Step to a new point:

\[
\begin{align*}
x &\leftarrow x + \theta \Delta x \\
y &\leftarrow y + \theta \Delta y \\
w &\leftarrow w + \theta \Delta w \\
z &\leftarrow z + \theta \Delta z
\end{align*}
\]

(\(\theta\) is the scalar damping factor).
Recall equations

\[
\begin{align*}
A \Delta x + \Delta w &= b - Ax - w \\
A^T \Delta y - \Delta z &= c - A^T y + z \\
Z \Delta x + X \Delta z &= \mu e - ZXe \\
W \Delta y + Y \Delta w &= \mu e - WYe
\end{align*}
\]

Solve for \( \Delta z \)

\[
\Delta z = X^{-1} (\mu e - ZXe - Z \Delta x)
\]

and for \( \Delta w \)

\[
\Delta w = Y^{-1} (\mu e - WYe - W \Delta y).
\]

Eliminate \( \Delta z \) and \( \Delta w \) from first two equations:

\[
\begin{align*}
A \Delta x - Y^{-1} W \Delta y &= b - Ax - \mu Y^{-1} e \\
A^T \Delta y + X^{-1} Z \Delta x &= c - A^T y + \mu X^{-1} e
\end{align*}
\]
Pick a smaller value of $\mu$ for the next iteration.

Repeat from beginning until current solution satisfies, within a tolerance, optimality conditions:

**Primal feasibility** $b - Ax - w = 0$.

**Dual feasibility** $c - A^T y + z = 0$.

**Duality gap** $b^T y - c^T x = 0$.

**Theorem.**
- Primal infeasibility gets smaller by a factor of $1 - \theta$ at every iteration.
- Dual infeasibility gets smaller by a factor of $1 - \theta$ at every iteration.
- If primal and dual are feasible, then duality gap decreases by a factor of $1 - \theta$ at every iteration (if $\mu = 0$, slightly slower convergence if $\mu > 0$).
Hard/impossible to “do” an interior-point method by hand.

Yet, easy to program on a computer (solving large systems of equations is routine).

LOQO implements an interior-point method.

Setting option loqo_options ’verbose=2’ in AMPL produces the following “typical” output:
<table>
<thead>
<tr>
<th>Iter</th>
<th>Primal</th>
<th>Dual</th>
<th>Sig</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Obj Value</td>
<td>Infeas</td>
<td>Obj Value</td>
</tr>
<tr>
<td>1</td>
<td>-7.8000000e+03</td>
<td>1.55e+03</td>
<td>5.5076028e-01</td>
</tr>
<tr>
<td>2</td>
<td>2.6725737e+05</td>
<td>7.84e+01</td>
<td>1.0917132e+00</td>
</tr>
<tr>
<td>3</td>
<td>1.1880365e+05</td>
<td>3.92e+00</td>
<td>4.5697310e-01</td>
</tr>
<tr>
<td></td>
<td>4.5697310e-01</td>
<td>2.02e-13</td>
<td>4.5697310e-01</td>
</tr>
<tr>
<td>4</td>
<td>6.7391043e+03</td>
<td>2.22e-01</td>
<td>7.2846138e-01</td>
</tr>
<tr>
<td>5</td>
<td>9.5202841e+02</td>
<td>3.12e-02</td>
<td>5.4810461e+00</td>
</tr>
<tr>
<td>6</td>
<td>2.1095320e+02</td>
<td>6.03e-03</td>
<td>2.7582307e+01</td>
</tr>
<tr>
<td>7</td>
<td>8.5669013e+01</td>
<td>1.36e-03</td>
<td>4.2343105e+01</td>
</tr>
<tr>
<td>8</td>
<td>5.8494756e+01</td>
<td>3.42e-04</td>
<td>4.6750024e+00</td>
</tr>
<tr>
<td>9</td>
<td>5.1228667e+01</td>
<td>8.85e-05</td>
<td>4.7875326e+00</td>
</tr>
<tr>
<td>10</td>
<td>4.9466277e+01</td>
<td>2.55e-05</td>
<td>4.8617380e+00</td>
</tr>
<tr>
<td>11</td>
<td>4.8792989e+01</td>
<td>1.45e-06</td>
<td>4.8736603e+01</td>
</tr>
<tr>
<td>12</td>
<td>4.8752154e+01</td>
<td>7.26e-08</td>
<td>4.8749328e+01</td>
</tr>
<tr>
<td>13</td>
<td>4.8750108e+01</td>
<td>3.63e-09</td>
<td>4.8749966e+01</td>
</tr>
<tr>
<td>14</td>
<td>4.8750005e+01</td>
<td>1.81e-10</td>
<td>4.8749998e+01</td>
</tr>
<tr>
<td>15</td>
<td>4.8750000e+01</td>
<td>9.07e-12</td>
<td>4.8750000e+01</td>
</tr>
</tbody>
</table>

---

**OPTIMAL SOLUTION FOUND**
Start with an optimization problem—in this case LP:

\[ \begin{align*}
\text{maximize} & \quad c^T x \\
\text{subject to} & \quad Ax \leq b \\
& \quad x \geq 0
\end{align*} \]

Use slack variables to make all inequality constraints into nonnegativities:

\[ \begin{align*}
\text{maximize} & \quad c^T x \\
\text{subject to} & \quad Ax + w = b \\
& \quad x, w \geq 0
\end{align*} \]

Replace nonnegativity constraints with logarithmic barrier terms in the objective:

\[ \begin{align*}
\text{maximize} & \quad c^T x + \mu \sum_j \log x_j + \mu \sum_i \log w_i \\
\text{subject to} & \quad Ax + w = b
\end{align*} \]
Incorporate the equality constraints into the objective using *Lagrange multipliers*:

\[
L(x, w, y) = c^T x + \mu \sum_j \log x_j + \mu \sum_i \log w_i + y^T (b - Ax - w)
\]

Set derivatives to zero:

\[
c + \mu X^{-1} e - A^T y = 0 \quad \text{(deriv wrt } x)\\
\mu W^{-1} e - y = 0 \quad \text{(deriv wrt } w)\\
b - Ax - w = 0 \quad \text{(deriv wrt } y)\\
\]

Introduce *dual complementary variables*:

\[
z = \mu X^{-1} e
\]

Rewrite system:

\[
c + z - A^T y = 0\\
X Z e = \mu e\\
W Y e = \mu e\\
b - Ax - w = 0
\]
Logarithmic Barrier Functions

Plots of $\mu \log x$ for various values of $\mu$: 

![Graph showing plots of $\mu \log x$ for different values of $\mu$.]
Lagrange Multipliers

maximize \( f(x) \)
subject to \( g(x) = 0 \)

maximize \( f(x) \)
subject to \( g_1(x) = 0 \)
\( g_2(x) = 0 \)
Nonlinear Optimization
Outline

• Algorithm
  – Basic Paradigm
  – Step-Length Control
  – Diagonal Perturbation
The Interior-Point Algorithm
Introduce Slack Variables

- Start with an optimization problem—for now, the simplest NLP:
  \[
  \begin{align*}
  \text{minimize} & \quad f(x) \\
  \text{subject to} & \quad h_i(x) \geq 0, \quad i = 1, \ldots, m
  \end{align*}
  \]

- Introduce slack variables to make all inequality constraints into nonnegativities:
  \[
  \begin{align*}
  \text{minimize} & \quad f(x) \\
  \text{subject to} & \quad h(x) - w = 0, \quad w \geq 0
  \end{align*}
  \]
Associated Log-Barrier Problem

- Replace nonnegativity constraints with logarithmic barrier terms in the objective:

\[
\text{minimize} \quad f(x) - \mu \sum_{i=1}^{m} \log(w_i)
\]

subject to \( h(x) - w = 0 \)
First-Order Optimality Conditions

• Incorporate the equality constraints into the objective using Lagrange multipliers:

\[ L(x, w, y) = f(x) - \mu \sum_{i=1}^{m} \log(w_i) - y^T(h(x) - w) \]

• Set all derivatives to zero:

\[ \nabla f(x) - \nabla h(x)^T y = 0 \]
\[ -\mu W^{-1} e + y = 0 \]
\[ h(x) - w = 0 \]
Symmetrize Complementarity Conditions

• Rewrite system:

\[ \nabla f(x) - \nabla h(x)^T y = 0 \]
\[ WY e = \mu e \]
\[ h(x) - w = 0 \]
Apply Newton’s Method

- Apply Newton’s method to compute *search directions*, $\Delta x$, $\Delta w$, $\Delta y$:

$$
\begin{bmatrix}
H(x, y) & 0 & -A(x)^T \\
0 & Y & W \\
A(x) & -I & 0 \\
\end{bmatrix}
\begin{bmatrix}
\Delta x \\
\Delta w \\
\Delta y \\
\end{bmatrix}
= 
\begin{bmatrix}
-\nabla f(x) + A(x)^T y \\
\mu e - WY e \\
-h(x) + w \\
\end{bmatrix}.
$$

Here,

$$
H(x, y) = \nabla^2 f(x) - \sum_{i=1}^{m} y_i \nabla^2 h_i(x)
$$

and

$$
A(x) = \nabla h(x)
$$

- Note: $H(x, y)$ is positive semidefinite if $f$ is convex, each $h_i$ is concave, and each $y_i \geq 0$. 

Reduced KKT System

- Use second equation to solve for $\Delta w$. Result is the reduced KKT system:

$$
\begin{bmatrix}
-H(x, y) & A^T(x) \\
A(x) & WY^{-1}
\end{bmatrix}
\begin{bmatrix}
\Delta x \\
\Delta y
\end{bmatrix} =
\begin{bmatrix}
\nabla f(x) - A^T(x)y \\
-h(x) + \mu Y^{-1}e
\end{bmatrix}
$$

- Iterate:

$$
x^{(k+1)} = x^{(k)} + \alpha^{(k)} \Delta x^{(k)} \\
w^{(k+1)} = w^{(k)} + \alpha^{(k)} \Delta w^{(k)} \\
y^{(k+1)} = y^{(k)} + \alpha^{(k)} \Delta y^{(k)}
$$
Convex vs. Nonconvex Optimization Probs

Nonlinear Programming (NLP)

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad h_i(x) = 0, \quad i \in \mathcal{E}, \\
& \quad h_i(x) \geq 0, \quad i \in \mathcal{I}.
\end{align*}
\]

NLP is \textit{convex} if

\begin{itemize}
  \item $h_i$’s in equality constraints are affine;
  \item $h_i$’s in inequality constraints are concave;
  \item $f$ is convex;
\end{itemize}

NLP is \textit{smooth} if

\begin{itemize}
  \item All are twice continuously differentiable.
\end{itemize}
For convex nonquadratic optimization, it does not suffice to choose the steplength $\alpha$ simply to maintain positivity of nonnegative variables.

- Consider, e.g., minimizing
  \[ f(x) = (1 + x^2)^{1/2}. \]
- The iterates can be computed explicitly:
  \[ x^{(k+1)} = -(x^{(k)})^3. \]
- Converges if and only if $|x| \leq 1$.
- Reason: away from 0, function is too linear.
Step-Length Control

A filter-type method is used to guide the choice of steplength $\alpha$. Define the dual normal matrix:

$$N(x, y, w) = H(x, y) + A^T(x)W^{-1}YA(x).$$

**Theorem** Suppose that $N(x, y, w)$ is positive definite.

1. If current solution is primal infeasible, then $(\Delta x, \Delta w)$ is a descent direction for the infeasibility $\|h(x) - w\|$.

2. If current solution is primal feasible, then $(\Delta x, \Delta w)$ is a descent direction for the barrier function.

Shorten $\alpha$ until $(\Delta x, \Delta w)$ is a descent direction for either the infeasibility or the barrier function.
Nonconvex Optimization: Diagonal Perturbation

- If $H(x, y)$ is not positive semidefinite then $N(x, y, w)$ might fail to be positive definite.
- In such a case, we lose the descent properties given in previous theorem.
- To regain those properties, we perturb the Hessian: $\tilde{H}(x, y) = H(x, y) + \lambda I$.
- And compute search directions using $\tilde{H}$ instead of $H$.

Notation: let $\tilde{N}$ denote the dual normal matrix associated with $\tilde{H}$.

**Theorem** If $\tilde{N}$ is positive definite, then $(\Delta x, \Delta w, \Delta y)$ is a descent direction for
1. the primal infeasibility, $\|h(x) - w\|$;
2. the noncomplementarity, $w^T y$. 
Notes:

- *Not necessarily* a descent direction for *dual infeasibility*.

- A *line search* is performed to find a value of $\lambda$ within a factor of 2 of the smallest permissible value.
Nonconvex Optimization: Jamming

Theorem If the problem is convex and and the current solution is not optimal and ..., then for any slack variable, say $w_i$, we have $w_i = 0$ implies $\Delta w_i \geq 0$.

- To paraphrase: for convex problems, as slack variables get small they tend to get large again. This is an antijamming theorem.
- A recent example of Wächter and Biegler shows that for nonconvex problems, jamming really can occur.
- Recent modification:
  - if a slack variable gets small and
  - its component of the step direction contributes to making a very short step,
  - then increase this slack variable to the average size of the variables the “mainstream” slack variables.
- This modification corrects all examples of jamming that we know about.
Modifications for General Problem Formulations

• Bounds, ranges, and free variables are all treated implicitly as described in *Linear Programming: Foundations and Extensions (LP:F&E)*.

• Net result is following reduced KKT system:

\[
\begin{bmatrix}
-(H(x, y) + D) & A^T(x) \\
A(x) & E
\end{bmatrix}
\begin{bmatrix}
\Delta x \\
\Delta y
\end{bmatrix}
= \begin{bmatrix}
\Phi_1 \\
\Phi_2
\end{bmatrix}
\]

• Here, *D* and *E* are *positive definite* diagonal matrices.

• Note that *D* helps reduce frequency of diagonal perturbation.

• Choice of barrier parameter *µ* and initial solution, if none is provided, is described in the paper.

• Stopping rules, matrix reordering heuristics, etc. are as described in *LP:F&E*. 
The language is called **AMPL**, which stands for *A Mathematical Programming Language*.

The “official” document describing the language is a book called “AMPL” by Fourer, Gay, and Kernighan. Amazon.com sells it for $78.01.

There are also online tutorials:

- [https://webspace.utexas.edu/sdb382/www/teaching/ce4920/ampl_tutorial.pdf](https://webspace.utexas.edu/sdb382/www/teaching/ce4920/ampl_tutorial.pdf)
- [http://www2.isye.gatech.edu/~jswann/teaching/AMPLTutorial.pdf](http://www2.isye.gatech.edu/~jswann/teaching/AMPLTutorial.pdf)
- Google: “AMPL tutorial” for several more.
NEOS Info

NEOS is the Network Enabled Optimization Server supported by our federal government and located at Argonne National Lab.

To submit an AMPL model to NEOS...

- visit http://www.neos-server.org/neos/,
- click on the icon,
- scroll down to the Nonlinearly Constrained Optimization list,
- click on LOQO [AMPL input],
- scroll down to Model File:,
- click on Choose File,
- select a file from your computer that contains an AMPL model,
- scroll down to e-mail address:,
- type in your email address, and
- click Submit to NEOS.

Piece of cake!
The Homogeneous Self-Dual Method
The Homogeneous Self-Dual Problem

**Primal-Dual Pair**

\[
\begin{align*}
\text{maximize} & \quad c^T x \\
\text{subject to} & \quad Ax \leq b \\
& \quad x \geq 0
\end{align*}
\]

\[
\begin{align*}
\text{minimize} & \quad b^T y \\
\text{subject to} & \quad A^T y \geq c \\
& \quad y \geq 0
\end{align*}
\]

**Homogeneous Self-Dual Problem**

\[
\begin{align*}
\text{maximize} & \quad 0 \\
\text{subject to} & \quad -A^T y + c\phi \leq 0 \\
& \quad Ax - b\phi \leq 0 \\
& \quad -c^T x + b^T y \leq 0 \\
& \quad x, \quad y, \quad \phi \geq 0
\end{align*}
\]
In Matrix Notation

maximize $0$
subject to
$$
\begin{bmatrix}
0 & -A^T & c \\
A & 0 & -b \\
-c^T & b^T & 0
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
\phi
\end{bmatrix} \leq
\begin{bmatrix}
0 \\
0 \\
0
\end{bmatrix}
$$

$x, y, \phi \geq 0$.

HSD is self-dual (constraint matrix is skew symmetric).

HSD is feasible ($x = 0, y = 0, \phi = 0$).

HSD is homogeneous—i.e., multiplying a feasible solution by a positive constant yields a new feasible solution.

Any feasible solution is optimal.

If $\phi$ is a null variable, then either primal or dual is infeasible (see text).
**Theorem.** Let \((x, y, \phi)\) be a solution to HSD. If \(\phi > 0\), then

- \(x^* = x/\phi\) is optimal for primal, and
- \(y^* = y/\phi\) is optimal for dual.

**Proof.**

- \(x^*\) is primal feasible—obvious.
- \(y^*\) is dual feasible—obvious.

Weak duality theorem implies that \(c^T x^* \leq b^T y^*\).

3rd HSD constraint implies reverse inequality.

Primal feasibility, plus dual feasibility, plus no gap implies optimality.
In New Notation:

\[
\begin{bmatrix}
0 & -A^T & c \\
A & 0 & -b \\
-c^T & b^T & 0
\end{bmatrix} \rightarrow A
\begin{bmatrix}
x \\ y \\ \phi
\end{bmatrix} \rightarrow x \quad n + m + 1 \rightarrow n
\]

maximize \quad 0
subject to \quad Ax + z = 0
\quad x, z \geq 0
More Notation

Infeasibility: \( \rho(x, z) = Ax + z \)
Complementarity: \( \mu(x, z) = \frac{1}{n} x^T z \)

Nonlinear System

\[
A(x + \Delta x) + (z + \Delta z) = \delta(Ax + z) \\
(X + \Delta X)(Z + \Delta Z)e = \delta \mu(x, z)e
\]

Linearized System

\[
A \Delta x + \Delta z = -(1 - \delta) \rho(x, z) \\
Z \Delta x + X \Delta z = \delta \mu(x, z)e - XZe
\]
Algorithm

Solve linearized system for \((\Delta x, \Delta z)\).

Pick step length \(\theta\).

Step to a new point:
\[
\bar{x} = x + \theta \Delta x, \quad \bar{z} = z + \theta \Delta z.
\]

Even More Notation

\[
\bar{\rho} = \rho(\bar{x}, \bar{z}), \quad \bar{\mu} = \mu(\bar{x}, \bar{z})
\]
Theorem 2

1. \( \Delta z^T \Delta x = 0 \).
2. \( \bar{\rho} = (1 - \theta + \theta \delta) \rho \).
3. \( \bar{\mu} = (1 - \theta + \theta \delta) \mu \).
4. \( \bar{X} \bar{Z} e - \bar{\mu} e = (1 - \theta)(X Z e - \mu e) + \theta^2 \Delta X \Delta Z e \).

Proof.

1. Tedious but not hard (see text).
2.

\[
\bar{\rho} = A(x + \theta \Delta x) + (z + \theta \Delta z) \\
= Ax + z + \theta (A \Delta x + \Delta z) \\
= \rho - \theta (1 - \delta) \rho \\
= (1 - \theta + \theta \delta) \rho.
\]
3. 

\[ \bar{x}^T \bar{z} = (x + \theta \Delta x)^T (z + \theta \Delta z) \]
\[ = x^T z + \theta (z^T \Delta x + x^T \Delta z) + \theta^2 \Delta x^T \Delta z \]
\[ = x^T z + \theta e^T (\delta \mu e - XZ e) \]
\[ = (1 - \theta + \theta \delta)x^T z. \]

Now, just divide by \( n \).

4. 

\[ \bar{X} \bar{Z} e - \bar{\mu} e = (X + \theta \Delta X)(Z + \theta \Delta Z)e - (1 - \theta + \theta \delta)\mu e \]
\[ = XZe - \mu e + \theta(X \Delta z + Z \Delta x + (1 - \delta)\mu e) + \theta^2 \Delta X \Delta Z e \]
\[ = (1 - \theta)(XZe - \mu e) + \theta^2 \Delta X \Delta Z e. \]
Neighborhoods of \( \{(x, z) > 0 : x_1z_1 = x_2z_2 = \cdots = x_nz_n\} \)

\[ \mathcal{N}(\beta) = \{(x, z) > 0 : \|XZe - \mu(x, z)e\| \leq \beta\mu(x, z)\} \]

Note: \( \beta < \beta' \) implies \( \mathcal{N}(\beta) \subset \mathcal{N}(\beta') \).

**Predictor-Corrector Algorithm**

**Odd Iterations–Predictor Step**

Assume \((x, z) \in \mathcal{N}(1/4)\).

Compute \((\Delta x, \Delta z)\) using \(\delta = 0\).

Compute \(\theta\) so that \((\bar{x}, \bar{z}) \in \mathcal{N}(1/2)\).

**Even Iterations–Corrector Step**

Assume \((x, z) \in \mathcal{N}(1/2)\).

Compute \((\Delta x, \Delta z)\) using \(\delta = 1\).

Put \(\theta = 1\).
Predictor-Corrector Algorithm

In Complementarity Space

Let

\[ u_j = x_j z_j \quad j = 1, 2, \ldots, n. \]
Well-Definedness of Algorithm

Must check that preconditions for each iteration are met.

*Technical Lemma.*

1. If $\delta = 0$, then $\|\Delta X \Delta Ze\| \leq \frac{n}{2} \mu$.

2. If $\delta = 1$ and $(x, z) \in \mathcal{N}(\beta)$, then $\|\Delta X \Delta Ze\| \leq \frac{\beta^2}{1-\beta} \mu/2$.

*Proof.* Tedious *and* tricky. See text.
**Theorem.**

1. After a predictor step, \((\bar{x}, \bar{z}) \in \mathcal{N}(1/2)\) and \(\bar{\mu} = (1 - \theta)\mu\).

2. After a corrector step, \((\bar{x}, \bar{z}) \in \mathcal{N}(1/4)\) and \(\bar{\mu} = \mu\).

**Proof.**

1. \((\bar{x}, \bar{z}) \in \mathcal{N}(1/2)\) by definition of \(\theta\).

   \[
   \bar{\mu} = (1 - \theta)\mu \text{ since } \delta = 0.
   \]

2. \(\theta = 1\) and \(\beta = 1/2\). Therefore,

   \[
   \|\bar{X}\bar{Z}e - \bar{\mu}e\| = \|\Delta X \Delta Z e\| \leq \mu/4.
   \]

   Need to show also that \((\bar{x}, \bar{z}) > 0\). Intuitively clear (see earlier picture) but proof is tedious. See text.
Complexity Analysis

Progress toward optimality is controlled by the stepsize $\theta$.

**Theorem.** In predictor steps, $\theta \geq \frac{1}{2\sqrt{n}}$.

**Proof.**

Consider taking a step with step length $t \leq 1/2\sqrt{n}$:

$$x(t) = x + t\Delta x, \quad z(t) = z + t\Delta z.$$ 

From earlier theorems and lemmas,

$$\|X(t)Z(t)e - \mu(t)e\| \leq (1 - t)\|XZe - \mu e\| + t^2\|\Delta X\Delta Ze\|$$

$$\leq (1 - t)\frac{\mu}{4} + t^2\frac{n\mu}{2}$$

$$\leq (1 - t)\frac{\mu}{4} + \frac{\mu}{8}$$

$$\leq (1 - t)\frac{\mu}{4} + (1 - t)\frac{\mu}{4}$$

$$= \frac{\mu(t)}{2}.$$ 

Therefore $(x(t), z(t)) \in \mathcal{N}(1/2)$ which implies that $\theta \geq 1/2\sqrt{n}$. 

Since
\[ \mu^{(2k)} = (1 - \theta^{(2k-1)})(1 - \theta^{(2k-3)}) \cdots (1 - \theta^{(1)})\mu^{(0)} \]
and \(\mu^{(0)} = 1\), we see from the previous theorem that
\[ \mu^{(2k)} \leq \left(1 - \frac{1}{2\sqrt{n}}\right)^k. \]

Hence, to get a small number, say \(2^{-L}\), as an upper bound for \(\mu^{(2k)}\) it suffices to pick \(k\) so that:
\[ \left(1 - \frac{1}{2\sqrt{n}}\right)^k \leq 2^{-L}. \]

This inequality is implied by the following simpler one:
\[ k \geq 2 \log(2) L \sqrt{n}. \]

Since the number of iterations is \(2k\), we see that \(4 \log(2) L \sqrt{n}\) iterations will suffice to make the final value of \(\mu\) be less than \(2^{-L}\).

Of course,
\[ \rho^{(k)} = \mu^{(k)} \rho^{(0)} \]
so the same bounds guarantee that the final infeasibility is small too.
Just a final remark: If primal and dual problems are feasible, then algorithm will produce a solution to HSD with $\phi > 0$ from which a solution to original problem can be extracted. See text for details.