After step 1 of the Simplex algorithm
\[ z = 3x_1 - \frac{5}{2} x_5 + 30 \]
Notice that if \( x_5 \) increases, \( z \) decreases, but if \( x_1 \) increases from 0, \( z \) increases.

§2.4 Summary of the Simplex Algorithm
See notes.

Empirical evidence suggests that the Simplex algorithm takes about \( 2m \) iterations to obtain the optimal solution.

§2.5 Adapting the Simplex algorithm to non-standard LP problems

1. Minimize the objective function \( z \):
   introduce \( z = -z \) and use
   \[ \min z = -\max (-z) = -\max z \]
   Now consider the problem with \( \max z \).

\[
\begin{align*}
\max f(x) &= -\min_{x \in \mathbb{R}} \{ -f(x) \} \\
\min f(x) &= -\max_{x \in \mathbb{R}} \{ -f(x) \}
\end{align*}
\]
2. Negative resource elements $b_j$: let $b_j = -b < 0 \quad (b > 0)$:

\[ a^T x \leq -b \quad \Rightarrow \quad -a^T x \geq b > 0 \]
\[ a^T x \geq -b \quad \Rightarrow \quad -a^T x < b > 0 \]
\[ a^T x = -b \quad \Rightarrow \quad -a^T x = b > 0. \]

Check RHS's of constraints are all $\geq 0$.

3. \[ \geq \text{ constraints: } \quad a = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} \quad x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \]

Introduce a \underline{surplus variable}: $x_{n+1}$

\[ \begin{cases} a^T x - x_{n+1} = b > 0 \\
\quad \uparrow \text{surplus} \\
\quad x_{n+1} \geq 0. \end{cases} \]

Check: $x_{n+1} = a^T x - b > 0 \quad \Rightarrow$

\[ a^T x > b, \text{ i.e. original constraint} \]

If $b = 0$, $a^T x > 0 \quad \Rightarrow \quad -a^T x < 0$

\[ \text{i.e. standard form}. \]
3. Negative decision variables $x_k \leq 0$:
   \[ \text{let } \hat{x}_k = -x_k; \text{ then } \hat{x}_k \geq 0. \]

4. Unrestricted decision variable $x_k$:
   \[ \text{let } x_k = \hat{x}_k - \bar{x}_k, \text{ where } \begin{cases} \hat{x}_k \geq 0; \\ \bar{x}_k \geq 0. \end{cases} \]

So we have 2 cases which do not reduce to standard form:
1. equality constraints $a^T x = b$
2. $\geq$ constraints $a^T x \geq b > 0$.

§5.1 Finding an initial FCP solution

Points 1 & 2 above affect the finding of an initial FCP.

Consider the example:

\[ \begin{align*}
\text{Max } z &= 3x_1 + 5x_2 \\
\text{s.t. } &x_1 \leq 4 \\
&3x_1 + 2x_2 \geq 18 \quad \text{← no longer} \\
&2x_2 \leq 12 \\
\text{with } &x_1, x_2 \geq 0.
\end{align*} \]
The feasible region is

\[ 2x_2 \leq 12 \]

\[ (0,0) \text{ is not feasible.} \]

\[ 3x_1 + 2x_2 \geq 18 \]

The feasible region is convex but the origin is not a FCP. The problem may also occur even if the origin is feasible, since it may not be a FCP.

In the example we introduce slack variables for the \(<\) constraint and a surplus variable for the \(\geq\) constraint:

\[
\begin{align*}
   x_1 + x_3 & = 4 & x_3 & \geq 0 \\
   3x_1 + 2x_2 - x_4 & = 18 & x_4 & \geq 0 \\
   2x_1 & + x_5 & \leq 12 & x_5 & \leq 0
\end{align*}
\]

If we set the non-basic variables \(x_1, x_2\) to 0, we get
\[ x_3 = 4 \]
\[-x_4 = 18 \leq \text{problem} \]
\[ x_5 = 12 \]

To get around this we introduce \underline{artificial variables} for all equality & \geq constraints: \underline{\bar{x}_6 \geq 0} \Rightarrow \]

\[ 3x_1 + 2x_2 - x_4 + \bar{x}_6 = 18 \]

To get an initial FCP we set

\[
\begin{cases}
\text{decision variables } & x_i = 0, \mu_i = 0 \\
\text{surplus variables } & x_4 = 0
\end{cases}
\]

(these are the initial non-basic variables).

We set the initial basic variables \underline{each variables} \( x_3, x_5 \)

\underline{artificial variables} \( \bar{x}_6 \geq 0 \)

equal to the RHS of their constraints.

So in the example:

\[ u_1 = 0, \mu_1 = 0, \mu_4 = 0 \]
\[ x_3 = 4, \bar{x}_6 = 18, x_5 = 12 \]

Do we have the original constraints?
In the example,
\[ x_4 - x_6 = 3x_1 + 2x_2 - 15 \geq 0 \]
We can't recover the original constraint until \( x_6 = 0 \), in which case
\[ 0 \leq x_4 = 3x_1 + 2x_2 - 15. \]
To recover the original problem we must force all artificial variables to 0.
We do this by introducing a new objective function
\[ W = - \sum_{k} x_k \leq 0 \]
for all artificial variables.
We have an initial phase of the simplex to maximize \( W \).
If the optimal of \( W \), \( W^* < 0 \), then \( x_k = 0 \) and we have recovered the original constraints — so we have a FCP of the original problem & we can use the iterations of the simplex algorithm to optimize \( Z \). If \( W^* < 0 \) there is no feasible solution to the original problem.