# Automated reasoning 2016

## Lecture 2: Theory of linear rational arithmetic (LRA)

Instructor: Ashutosh Gupta

TIFR, India

Compile date: 2016-02-26

# Where are we and where are we going?

We have

- 

We will

- 

Topic 2.1

# Theory of rational linear arithmetic

# Rational linear arithmetic

**Rational linear arithmetic:** quantifier-free first order formulas with structure $\Sigma = (\{+/2, 0, 1, \dots\}, \{< /2\})$ with a set of axioms

Note: We have seen the axioms in the previous course.

# Rational linear arithmetic

**Rational linear arithmetic:** quantifier-free first order formulas with structure $\Sigma = (\{+/2, 0, 1, \dots\}, \{< /2\})$ with a set of axioms

Note: We have seen the axioms in the previous course.

We will discuss a number of methods to find satisfiability of conjunction of linear inequalities.

- Fourier-Motzkin
- Simplex

We will not cover following methods

- Loos-Weispfenning quantifier elimination
- Omega test method
- Elliposid method
- Kermakar's method

# Rational linear arithmetic

**Rational linear arithmetic:** quantifier-free first order formulas with structure $\Sigma = (\{+/2, 0, 1, \dots\}, \{< /2\})$ with a set of axioms

Note: We have seen the axioms in the previous course.

We will discuss a number of methods to find satisfiability of conjunction of linear inequalities.

- Fourier-Motzkin
- Simplex

We will not cover following methods

- Loos-Weispfenning quantifier elimination
- Omega test method
- Elliposid method
- Kermakar's method

We present the above methods using non-strict linear inequalities. However, they are extendable to strict inequalities, equalities, dis-equalities.

# Fourier-Motzkin

The algorithm proceeds by eliminating variables one by one. After eliminating all the variables, if the input reduces to $\top$ then only the input is satisfiable.

# Fourier-Motzkin

The algorithm proceeds by eliminating variables one by one. After eliminating all the variables, if the input reduces to $\top$ then only the input is satisfiable.

For each variable $x$, any conjunction of linear inequalities can be transformed into the following form.

$$\bigwedge_{j=1}^{m} s_j \leq x \wedge \bigwedge_{i=1}^{l} x \leq t_i \wedge \bigwedge_{k=1}^{n} u_k \leq 0$$

# Fourier-Motzkin

The algorithm proceeds by eliminating variables one by one. After eliminating all the variables, if the input reduces to $\top$ then only the input is satisfiable.

For each variable $x$, any conjunction of linear inequalities can be transformed into the following form.

$$\bigwedge_{j=1}^{m} s_j \leq x \wedge \bigwedge_{i=1}^{l} x \leq t_i \wedge \bigwedge_{k=1}^{n} u_k \leq 0$$

$$\Updownarrow$$

$$\bigwedge_{j=1}^{m} \bigwedge_{i=1}^{l} s_j \leq t_i \wedge \bigwedge_{k=1}^{n} u_k \leq 0$$

# Fourier-Motzkin

The algorithm proceeds by eliminating variables one by one. After eliminating all the variables, if the input reduces to $\top$ then only the input is satisfiable.

For each variable $x$, any conjunction of linear inequalities can be transformed into the following form.

$$\bigwedge_{j=1}^{m} s_j \leq x \wedge \bigwedge_{i=1}^{l} x \leq t_i \wedge \bigwedge_{k=1}^{n} u_k \leq 0$$

$$\Updownarrow$$

$$\bigwedge_{j=1}^{m} \bigwedge_{i=1}^{l} s_j \leq t_i \wedge \bigwedge_{k=1}^{n} u_k \leq 0$$

equisatisfiable but without $x$

# Fourier-Motzkin

The algorithm proceeds by eliminating variables one by one. After eliminating all the variables, if the input reduces to $\top$ then only the input is satisfiable.

For each variable $x$, any conjunction of linear inequalities can be transformed into the following form.

$$\bigwedge_{j=1}^{m} s_j \leq x \wedge \bigwedge_{i=1}^{l} x \leq t_i \wedge \bigwedge_{k=1}^{n} u_k \leq 0$$

$$\Updownarrow$$

$$\bigwedge_{j=1}^{m} \bigwedge_{i=1}^{l} s_j \leq t_i \wedge \bigwedge_{k=1}^{n} u_k \leq 0$$

equisatisfiable but without $x$

## Exercise 2.1

*a. Add support for equality, dis-equality, and strict inequalities*

*b. What is the complexity?*

# Example: Fourier-Motzkin

## Example 2.1

*Consider:* $-x_1 + x_2 + 2x_3 \leq 0 \wedge x_1 - x_2 \leq 0 \wedge x_1 - x_3 \leq 0 \wedge 1 - x_3 \leq -1$

# Example: Fourier-Motzkin

## Example 2.1

*Consider:* $-x_1 + x_2 + 2x_3 \leq 0 \wedge x_1 - x_2 \leq 0 \wedge x_1 - x_3 \leq 0 \wedge 1 - x_3 \leq -1$

*Suppose we eliminate $x_1$ first. We transform the constraints into our format.*

# Example: Fourier-Motzkin

## Example 2.1

*Consider:* $-x_1 + x_2 + 2x_3 \leq 0 \wedge x_1 - x_2 \leq 0 \wedge x_1 - x_3 \leq 0 \wedge 1 - x_3 \leq -1$

*Suppose we eliminate $x_1$ first. We transform the constraints into our format.*

$\underbrace{x_2 + 2x_3 \leq x_1}_{x_1 \text{ lower bounded}}$

# Example: Fourier-Motzkin

## Example 2.1

*Consider:* $-x_1 + x_2 + 2x_3 \leq 0 \wedge x_1 - x_2 \leq 0 \wedge x_1 - x_3 \leq 0 \wedge 1 - x_3 \leq -1$

*Suppose we eliminate $x_1$ first. We transform the constraints into our format.*

$$\underbrace{x_2 + 2x_3 \leq x_1}_{x_1 \text{ lower bounded}} \wedge \underbrace{(x_1 \leq x_2 \wedge x_1 \leq x_3)}_{x_1 \text{ upper bounded}}$$

# Example: Fourier-Motzkin

## Example 2.1

*Consider:* $-x_1 + x_2 + 2x_3 \leq 0 \wedge x_1 - x_2 \leq 0 \wedge x_1 - x_3 \leq 0 \wedge 1 - x_3 \leq -1$

*Suppose we eliminate $x_1$ first. We transform the constraints into our format.*

$$\underbrace{x_2 + 2x_3 \leq x_1}_{x_1 \textit{ lower bounded}} \wedge \underbrace{(x_1 \leq x_2 \wedge x_1 \leq x_3)}_{x_1 \textit{ upper bounded}} \wedge \underbrace{-x_3 + 1 \leq 0}_{x_1 \textit{ does not occur}}$$

# Example: Fourier-Motzkin

## Example 2.1

*Consider:* $-x_1 + x_2 + 2x_3 \leq 0 \wedge x_1 - x_2 \leq 0 \wedge x_1 - x_3 \leq 0 \wedge 1 - x_3 \leq -1$

*Suppose we eliminate $x_1$ first. We transform the constraints into our format.*

$$\underbrace{x_2 + 2x_3 \leq x_1}_{x_1 \textit{ lower bounded}} \wedge \underbrace{(x_1 \leq x_2 \wedge x_1 \leq x_3)}_{x_1 \textit{ upper bounded}} \wedge \underbrace{-x_3 + 1 \leq 0}_{x_1 \textit{ does not occur}}$$

*Eliminated constraints:* $x_2 + 2x_3 \leq x_2 \wedge x_2 + 2x_3 \leq x_3 \wedge -x_3 + 1 \leq 0$

# Example: Fourier-Motzkin

## Example 2.1

*Consider:* $-x_1 + x_2 + 2x_3 \leq 0 \land x_1 - x_2 \leq 0 \land x_1 - x_3 \leq 0 \land 1 - x_3 \leq -1$

*Suppose we eliminate $x_1$ first. We transform the constraints into our format.*

$$\underbrace{x_2 + 2x_3 \leq x_1}_{x_1 \text{ lower bounded}} \land \underbrace{(x_1 \leq x_2 \land x_1 \leq x_3)}_{x_1 \text{ upper bounded}} \land \underbrace{-x_3 + 1 \leq 0}_{x_1 \text{ does not occur}}$$

*Eliminated constraints:* $x_2 + 2x_3 \leq x_2 \land x_2 + 2x_3 \leq x_3 \land -x_3 + 1 \leq 0$

*After simplification:* $x_3 \leq 0 \land x_2 + x_3 \leq 0 \land -x_3 + 1 \leq 0$

# Example: Fourier-Motzkin

## Example 2.1

*Consider:* $-x_1 + x_2 + 2x_3 \leq 0 \wedge x_1 - x_2 \leq 0 \wedge x_1 - x_3 \leq 0 \wedge 1 - x_3 \leq -1$

*Suppose we eliminate $x_1$ first. We transform the constraints into our format.*

$$\underbrace{x_2 + 2x_3 \leq x_1}_{x_1 \text{ lower bounded}} \wedge \underbrace{(x_1 \leq x_2 \wedge x_1 \leq x_3)}_{x_1 \text{ upper bounded}} \wedge \underbrace{-x_3 + 1 \leq 0}_{x_1 \text{ does not occur}}$$

*Eliminated constraints:* $x_2 + 2x_3 \leq x_2 \wedge x_2 + 2x_3 \leq x_3 \wedge -x_3 + 1 \leq 0$

*After simplification:* $x_3 \leq 0 \wedge x_2 + x_3 \leq 0 \wedge -x_3 + 1 \leq 0$

*Since $x_2$ is bounded only from one side, we can trivially eliminate $x_2$:*
$x_3 \leq 0 \wedge -x_3 + 1 \leq 0$

# Example: Fourier-Motzkin

## Example 2.1

*Consider:* $-x_1 + x_2 + 2x_3 \leq 0 \wedge x_1 - x_2 \leq 0 \wedge x_1 - x_3 \leq 0 \wedge 1 - x_3 \leq -1$

*Suppose we eliminate $x_1$ first. We transform the constraints into our format.*

$$\underbrace{x_2 + 2x_3 \leq x_1}_{x_1 \text{ lower bounded}} \wedge \underbrace{(x_1 \leq x_2 \wedge x_1 \leq x_3)}_{x_1 \text{ upper bounded}} \wedge \underbrace{-x_3 + 1 \leq 0}_{x_1 \text{ does not occur}}$$

*Eliminated constraints:* $x_2 + 2x_3 \leq x_2 \wedge x_2 + 2x_3 \leq x_3 \wedge -x_3 + 1 \leq 0$

*After simplification:* $x_3 \leq 0 \wedge x_2 + x_3 \leq 0 \wedge -x_3 + 1 \leq 0$

*Since $x_2$ is bounded only from one side, we can trivially eliminate $x_2$:*
$x_3 \leq 0 \wedge -x_3 + 1 \leq 0$

*Eliminating $x_3$:* $1 \leq 0$

# Example: Fourier-Motzkin

## Example 2.1

*Consider:* $-x_1 + x_2 + 2x_3 \leq 0 \wedge x_1 - x_2 \leq 0 \wedge x_1 - x_3 \leq 0 \wedge 1 - x_3 \leq -1$

*Suppose we eliminate* $x_1$ *first. We transform the constraints into our format.*

$\underbrace{x_2 + 2x_3 \leq x_1}_{x_1 \textit{ lower bounded}} \wedge \underbrace{(x_1 \leq x_2 \wedge x_1 \leq x_3)}_{x_1 \textit{ upper bounded}} \wedge \underbrace{-x_3 + 1 \leq 0}_{x_1 \textit{ does not occur}}$

*Eliminated constraints:* $x_2 + 2x_3 \leq x_2 \wedge x_2 + 2x_3 \leq x_3 \wedge -x_3 + 1 \leq 0$

*After simplification:* $x_3 \leq 0 \wedge x_2 + x_3 \leq 0 \wedge -x_3 + 1 \leq 0$

*Since* $x_2$ *is bounded only from one side, we can trivially eliminate* $x_2$*:*
$x_3 \leq 0 \wedge -x_3 + 1 \leq 0$

*Eliminating* $x_3$*:* $1 \leq 0 \leftarrow$ *false formula therefore unsat*

# Example: Fourier-Motzkin

## Example 2.1

*Consider:* $-x_1 + x_2 + 2x_3 \leq 0 \land x_1 - x_2 \leq 0 \land x_1 - x_3 \leq 0 \land 1 - x_3 \leq -1$

*Suppose we eliminate $x_1$ first. We transform the constraints into our format.*

$$\underbrace{x_2 + 2x_3 \leq x_1}_{x_1 \text{ lower bounded}} \land \underbrace{(x_1 \leq x_2 \land x_1 \leq x_3)}_{x_1 \text{ upper bounded}} \land \underbrace{-x_3 + 1 \leq 0}_{x_1 \text{ does not occur}}$$

*Eliminated constraints:* $x_2 + 2x_3 \leq x_2 \land x_2 + 2x_3 \leq x_3 \land -x_3 + 1 \leq 0$

*After simplification:* $x_3 \leq 0 \land x_2 + x_3 \leq 0 \land -x_3 + 1 \leq 0$

*Since $x_2$ is bounded only from one side, we can trivially eliminate $x_2$:*
$x_3 \leq 0 \land -x_3 + 1 \leq 0$

*Eliminating $x_3$:* $1 \leq 0 \leftarrow$ *false formula therefore unsat*

Both complexity and practical performance of the algorithm are bad.

Topic 2.2

## Simplex

# Simplex

Simplex was originally for linear optimization

A variation of simplex is used to check satisfiablity, which is called incremental simplex

- incremental simplex takes atoms one by one and finds satisfying assignment of so for taken atoms before processing the next atom.

# Incremental simplex as theory solver

push(): Add new atom to the simplex state.

# Incremental simplex as theory solver

push(): Add new atom to the simplex state.

pop(): inexpensive operation (we will see the details)

# Incremental simplex as theory solver

push(): Add new atom to the simplex state.

pop(): inexpensive operation (we will see the details)

unsatCore(): again inexpensive operation

# Simplex - notation

Consider the following constraints in matrix form

$$Ax \leq b,$$

where $A$ is a $m \times n$ matrix.

# Simplex - notation

Consider the following constraints in matrix form

$$Ax \leq b,$$

where $A$ is a $m \times n$ matrix.

By introducing fresh variables, we transform the above into

$$\begin{bmatrix} -\mathtt{I} & A \end{bmatrix} \begin{bmatrix} s \\ x \end{bmatrix} = 0 \text{ and } s \leq b.$$

# Simplex - notation

Consider the following constraints in matrix form

$$Ax \leq b,$$

where $A$ is a $m \times n$ matrix.
By introducing fresh variables, we transform the above into

$$\begin{bmatrix} -\mathtt{I} & A \end{bmatrix} \begin{bmatrix} s \\ x \end{bmatrix} = 0 \text{ and } s \leq b.$$

$s$ are called slack variables. Since there is no reason to distinguish $x$ and $s$ in simplex, $A$ will refer to $\begin{bmatrix} -\mathtt{I} & A \end{bmatrix}$ and $x$ will refer to $\begin{bmatrix} s \\ x \end{bmatrix}$.

# Simplex - notation

Consider the following constraints in matrix form

$$Ax \leq b,$$

where $A$ is a $m \times n$ matrix.
By introducing fresh variables, we transform the above into

$$\begin{bmatrix} -\mathtt{I} & A \end{bmatrix} \begin{bmatrix} s \\ x \end{bmatrix} = 0 \text{ and } s \leq b.$$

$s$ are called slack variables. Since there is no reason to distinguish $x$ and $s$ in simplex, $A$ will refer to $\left[\begin{smallmatrix} -\mathtt{I} & A \end{smallmatrix}\right]$ and $x$ will refer to $\left[\begin{smallmatrix} s \\ x \end{smallmatrix}\right]$.

The above constraints will be denoted by

$$Ax = 0 \text{ and } \bigwedge_{i=1}^{m+n} l_i \leq x_i \leq u_i.$$

$l_i$ and $u_i$ are $+\infty$ and $-\infty$ if there is no lower and upper bound, respectively.

# Simplex - notation (contd.)

$$Ax = 0 \text{ and } \bigwedge_{i=1}^{m+n} l_i \leq x_i \leq u_i$$

- $A$ is $m \times (m+n)$ matrix.

# Simplex - notation (contd.)

$$Ax = 0 \text{ and } \bigwedge_{i=1}^{m+n} l_i \leq x_i \leq u_i$$

- $A$ is $m \times (m+n)$ matrix.
- Since $Ax = 0$ defines an $n$-dim subspace in $(m+n)$-dim space, if we choose values of $n$ variables then we fix values of the other $m$ variables.

# Simplex - notation (contd.)

$$Ax = 0 \text{ and } \bigwedge_{i=1}^{m+n} l_i \leq x_i \leq u_i$$

- $A$ is $m \times (m+n)$ matrix.
- Since $Ax = 0$ defines an $n$-dim subspace in $(m+n)$-dim space, if we choose values of $n$ variables then we fix values of the other $m$ variables.
- We will refer to $i$th column of $A$ as the column corresponding to $x_i$.

# Simplex - notation (contd.)

$$Ax = 0 \text{ and } \bigwedge_{i=1}^{m+n} l_i \leq x_i \leq u_i$$

- $A$ is $m \times (m+n)$ matrix.
- Since $Ax = 0$ defines an $n$-dim subspace in $(m+n)$-dim space, if we choose values of $n$ variables then we fix values of the other $m$ variables.
- We will refer to $i$th column of $A$ as the column corresponding to $x_i$.

## Example 2.2

*Consider:* $-x + y \leq -2 \wedge x \leq 3$

# Simplex - notation (contd.)

$$Ax = 0 \text{ and } \bigwedge_{i=1}^{m+n} l_i \leq x_i \leq u_i$$

- $A$ is $m \times (m+n)$ matrix.
- Since $Ax = 0$ defines an $n$-dim subspace in $(m+n)$-dim space, if we choose values of $n$ variables then we fix values of the other $m$ variables.
- We will refer to $i$th column of $A$ as the column corresponding to $x_i$.

## Example 2.2

*Consider:* $-x + y \leq -2 \wedge x \leq 3$

We have slack variables $s_1$ and $s_2$.

# Simplex - notation (contd.)

$$Ax = 0 \text{ and } \bigwedge_{i=1}^{m+n} l_i \leq x_i \leq u_i$$

- $A$ is $m \times (m+n)$ matrix.
- Since $Ax = 0$ defines an $n$-dim subspace in $(m+n)$-dim space, if we choose values of $n$ variables then we fix values of the other $m$ variables.
- We will refer to $i$th column of $A$ as the column corresponding to $x_i$.

## Example 2.2

*Consider:* $-x + y \leq -2 \wedge x \leq 3$

We have slack variables $s_1$ and $s_2$. In matrix form,

$$\left[\begin{array}{cc:cc} -1 & 0 & -1 & 1 \\ 0 & -1 & 1 & 0 \end{array}\right] \left[\begin{array}{c} s_1 \\ s_2 \\ \hdashline x \\ y \end{array}\right] = 0 \qquad \begin{array}{c} s_1 \leq -2 \\ s_2 \leq 3 \end{array}$$

# Simplex - basic and nonbasic variables

## Definition 2.1

*Simplex assumes all the columns of* $-\mathtt{I}$ *occur in* $A$*.*

# Simplex - basic and nonbasic variables

## Definition 2.1

*Simplex assumes all the columns of $-\mathtt{I}$ occur in $A$. The variables corresponding to the columns are called basic variables. Others are called nonbasic variables.*

# Simplex - basic and nonbasic variables

## Definition 2.1

*Simplex assumes all the columns of $-\mathtt{I}$ occur in $A$.The variables corresponding to the columns are called basic variables. Others are called nonbasic variables.*

Note: the assumption is true about initial $A$

# Simplex - basic and nonbasic variables

## Definition 2.1

*Simplex assumes all the columns of $-\mathtt{I}$ occur in $A$.The variables corresponding to the columns are called basic variables. Others are called nonbasic variables.*

Note: the assumption is true about initial $A$

## Definition 2.2

*Let $B$ be the set of indexes for basic variables and $NB \triangleq 1..n - B$.*
*For $j \in B$, let $k_j$ be a row s.t. $A_{k_j j} = -1$.*

# Simplex - basic and nonbasic variables

## Definition 2.1

*Simplex assumes all the columns of $-\mathtt{I}$ occur in $A$. The variables corresponding to the columns are called basic variables. Others are called nonbasic variables.*

Note: the assumption is true about initial $A$

## Definition 2.2

*Let $B$ be the set of indexes for basic variables and $NB \triangleq 1..n - B$. For $j \in B$, let $k_j$ be a row s.t. $A_{k_j j} = -1$.*

## Example 2.3

$$\begin{bmatrix} -1 & 0 & -1 & 1 \\ 0 & -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ x \\ y \end{bmatrix} = 0 \qquad \begin{matrix} s_1 \leq -2 \\ s_2 \leq 3 \end{matrix}$$

*Currently, $s_1$ and $s_2$ are basic and $x$ and $y$ are nonbasic.*

# Simplex - basic and nonbasic variables

## Definition 2.1

*Simplex assumes all the columns of* $-\mathtt{I}$ *occur in* $A$*.The variables corresponding to the columns are called basic variables. Others are called nonbasic variables.*

Note: the assumption is true about initial $A$

## Definition 2.2

*Let* $B$ *be the set of indexes for basic variables and* $NB \triangleq 1..n - B$*.*
*For* $j \in B$*, let* $k_j$ *be a row s.t.* $A_{k_j j} = -1$*.*

## Example 2.3

$$\begin{bmatrix} -1 & 0 & -1 & 1 \\ 0 & -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ x \\ y \end{bmatrix} = 0 \qquad \begin{array}{c} s_1 \leq -2 \\ s_2 \leq 3 \end{array}$$

*Currently,* $s_1$ *and* $s_2$ *are basic and* $x$ *and* $y$ *are nonbasic.*
$B = \{1, 2\}$, $NB = \{3, 4\}$, $k_1 = 1$, *and* $k_2 = 2$.

# Simplex - current assignment

## Definition 2.3

*Simplex maintains* current assignment $v : x \to \mathbb{Q}$ *s.t.*

- $Av = 0$,
- *nonbasic variables satisfy their bounds, and,*
- *consequently values for basic variables in* $v$ *are fixed and* $v$ *may* violate *a bound of* at most *one basic variable.*

# Simplex - current assignment

## Definition 2.3

*Simplex maintains* current assignment $v : x \to \mathbb{Q}$ *s.t.*

- $Av = 0$,
- *nonbasic variables satisfy their bounds, and,*
- *consequently values for basic variables in $v$ are fixed and $v$ may violate a bound of at most one basic variable.*

Explained later why "at most" one

# Simplex - current assignment

## Definition 2.3

*Simplex maintains current assignment* $v : x \to \mathbb{Q}$ *s.t.*

- $Av = 0$,
- *nonbasic variables satisfy their bounds, and,*
- *consequently values for basic variables in* $v$ *are fixed and* $v$ *may violate a bound of at most one basic variable.*

Explained later why "at most" one

## Example 2.4

$$\left[\begin{array}{cc:cc} -1 & 0 & -1 & 1 \\ 0 & -1 & 1 & 0 \end{array}\right] \left[\begin{array}{c} s_1 \\ s_2 \\ \hdashline x \\ y \end{array}\right] = 0$$

$$s_1 \leq -2$$
$$s_2 \leq 3$$

# Simplex - current assignment

## Definition 2.3

*Simplex maintains* current assignment $v : x \rightarrow \mathbb{Q}$ *s.t.*

- $Av = 0$,
- *nonbasic variables satisfy their bounds, and,*
- *consequently values for basic variables in* $v$ *are fixed and* $v$ *may* violate *a bound of* at most *one basic variable.*

Explained later why "at most" one

## Example 2.4

$$\left[\begin{array}{cc|cc} -1 & 0 & -1 & 1 \\ 0 & -1 & 1 & 0 \end{array}\right] \left[\begin{array}{c} s_1 \\ s_2 \\ \hline x \\ y \end{array}\right] = 0 \qquad \begin{array}{c} s_1 \leq -2 \\ s_2 \leq 3 \end{array}$$

Initially, $v = \{\underbrace{x \mapsto 0, y \mapsto 0}_{\text{Choose values for nonbasic variables, others follow!}}, s_1 \mapsto 0, s_2 \mapsto 0\}$

# Simplex - current assignment

## Definition 2.3

*Simplex maintains current assignment* $v : x \to \mathbb{Q}$ *s.t.*

- $Av = 0$,
- *nonbasic variables satisfy their bounds, and,*
- *consequently values for basic variables in* $v$ *are fixed and* $v$ *may violate a bound of at most one basic variable.*

Explained later why "at most" one

## Example 2.4

$$\left[\begin{array}{cc:cc} -1 & 0 & -1 & 1 \\ 0 & -1 & 1 & 0 \end{array}\right]\left[\begin{array}{c} s_1 \\ s_2 \\ \hdashline x \\ y \end{array}\right] = 0$$

Currently violated

$s_1 \leq -2$

$s_2 \leq 3$

Initially, $v = \{\underbrace{x \mapsto 0, y \mapsto 0}_{\text{Choose values for nonbasic variables, others follow!}}, s_1 \mapsto 0, s_2 \mapsto 0\}$

# Simplex - state

For variable $i \in NB$, $v(x_i)$ is equal to one of the existing bounds of $x_i$ and if no bounds exists for $x_i$ then $v(x_i) = 0$.

# Simplex - state

For variable $i \in NB$, $v(x_i)$ is equal to one of the existing bounds of $x_i$ and if no bounds exists for $x_i$ then $v(x_i) = 0$.

### Definition 2.4

*A bound on $x_i$ is called active if $v(x_i)$ is equal to the bound.*
*We will mark the active bounds by $^*$.*

### Definition 2.5

*The NB set and bound activity defines the current state of simplex.*

# Simplex - state

For variable $i \in NB$, $v(x_i)$ is equal to one of the existing bounds of $x_i$ and if no bounds exists for $x_i$ then $v(x_i) = 0$.

## Definition 2.4

*A bound on $x_i$ is called active if $v(x_i)$ is equal to the bound.*
*We will mark the active bounds by $^*$.*

## Definition 2.5

*The NB set and bound activity defines the current state of simplex.*

## Example 2.5

$$\begin{bmatrix} -1 & 0 & -1 & 1 \\ 0 & -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ x \\ y \end{bmatrix} = 0 \qquad \begin{array}{c} s_1 \leq -2 \\ s_2 \leq 3 \end{array}$$

*Since all nonbasic variables have no bounds, no bound is marked active.*

# Simplex - pivot operation

If $v$ violates a bound constraint of some basic variable, then simplex correct it by applying pivot operation

## Definition 2.6

*Let us suppose $x_j$ is basic, column $j$ has $-1$ at row $k$, and $x_i$ is nonbasic.*

$$
\begin{array}{cc}
 & \begin{array}{ccccccc} & & j & & i & & \\ & & \downarrow & & \downarrow & & \end{array} \\
k \longrightarrow & \begin{bmatrix} & \vdots & 0 & \vdots & a & \vdots & \\ & \dots & -1 & \dots & b & \dots & \\ & \vdots & 0 & \vdots & c & \vdots & \end{bmatrix}
\end{array}
$$

# Simplex - pivot operation

If $v$ violates a bound constraint of some basic variable, then simplex correct it by applying pivot operation

## Definition 2.6

*Let us suppose $x_j$ is basic, column $j$ has $-1$ at row $k$, and $x_i$ is nonbasic. A pivot operation between $i$ and $j$ exchanges the roll between $x_i$ and $x_j$.*

$$\begin{array}{cc} & \begin{array}{ccccc} & j & & i & \\ & \downarrow & & \downarrow & \end{array} \\ k \longrightarrow & \begin{bmatrix} \vdots & 0 & \vdots & a & \vdots \\ \dots & -1 & \dots & b & \dots \\ \vdots & 0 & \vdots & c & \vdots \end{bmatrix} \end{array}$$

# Simplex - pivot operation

If $v$ violates a bound constraint of some basic variable, then simplex correct it by applying pivot operation

### Definition 2.6

*Let us suppose $x_j$ is basic, column $j$ has $-1$ at row $k$, and $x_i$ is nonbasic. A pivot operation between $i$ and $j$ exchanges the roll between $x_i$ and $x_j$. The pivot operation applies row operations until column $i$ has $-1$ at row $k$ and all other entries in the column are zero.*

$$
\begin{array}{cc}
 & \begin{array}{ccccccc} & j & & i & \\ & \downarrow & & \downarrow & \end{array} \\
k \longrightarrow & \begin{bmatrix} \vdots & 0 & \vdots & a & \vdots \\ \dots & -1 & \dots & b & \dots \\ \vdots & 0 & \vdots & c & \vdots \end{bmatrix}
\end{array}
$$

# Simplex - pivot operation

If $v$ violates a bound constraint of some basic variable, then simplex correct it by applying pivot operation

## Definition 2.6

*Let us suppose $x_j$ is basic, column $j$ has $-1$ at row $k$, and $x_i$ is nonbasic. A pivot operation between $i$ and $j$ exchanges the roll between $x_i$ and $x_j$. The pivot operation applies row operations until column $i$ has $-1$ at row $k$ and all other entries in the column are zero.*

After pivot operation between $i$ and $j$

$$
\begin{array}{c c}
 & \begin{array}{ccccc} & j & & i & \\ & \downarrow & & \downarrow & \end{array} \\
k \longrightarrow & \begin{bmatrix} \vdots & 0 & \vdots & a & \vdots \\ \dots & -1 & \dots & b & \dots \\ \vdots & 0 & \vdots & c & \vdots \end{bmatrix}
\end{array}
=>
\begin{bmatrix} \vdots & \frac{a}{b} & \vdots & 0 & \vdots \\ \dots & \frac{1}{b} & \dots & -1 & \dots \\ \vdots & \frac{c}{b} & \vdots & 0 & \vdots \end{bmatrix}
$$

# Simplex - choosing non-basic column for pivot

Wlog, let $1 \in B$ , $k_1 = 1$, and $v(x_1)$ violates $u_1$. We need to decrease $v(x_1)$.
We call $v(x_1) - u_1$ violation difference.

# Simplex - choosing non-basic column for pivot

Wlog, let $1 \in B$ , $k_1 = 1$, and $v(x_1)$ violates $u_1$. We need to decrease $v(x_1)$.
We call $v(x_1) - u_1$ violation difference.

Since $x_1 = \sum_{i \in NB} a_{1i}x_i$, we need to change $v(x_i)$ of some $x_i$ s.t. $a_{1i}x_i$ decreases

# Simplex - choosing non-basic column for pivot

Wlog, let $1 \in B$ , $k_1 = 1$, and $v(x_1)$ violates $u_1$. We need to decrease $v(x_1)$.
We call $v(x_1) - u_1$ violation difference.
Since $x_1 = \sum_{i \in NB} a_{1i}x_i$, we need to change $v(x_i)$ of some $x_i$ s.t. $a_{1i}x_i$ decreases

### Definition 2.7

*A column $i \in NB$ is suitable if*

- *$x_i$ is unbounded,*
- *$x_i = u_i$ and $a_{1i} > 0$, or*
- *$x_i = l_i$ and $a_{1i} < 0$.*

*$i$ is selected suitable column if $i$ is the smallest suitable column.*

# Simplex - choosing non-basic column for pivot

Wlog, let $1 \in B$ , $k_1 = 1$, and $v(x_1)$ violates $u_1$. We need to decrease $v(x_1)$.
We call $v(x_1) - u_1$ violation difference.

Since $x_1 = \sum_{i \in NB} a_{1i}x_i$, we need to change $v(x_i)$ of some $x_i$ s.t. $a_{1i}x_i$ decreases

## Definition 2.7

*A column $i \in NB$ is suitable if*

- *$x_i$ is unbounded,*
- *$x_i = u_i$ and $a_{1i} > 0$, or*
- *$x_i = l_i$ and $a_{1i} < 0$.*

*$i$ is selected suitable column if $i$ is the smallest suitable column.*

## Example 2.6

$$\begin{bmatrix} -1 & 0 & -1 & 1 \\ 0 & -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ x \\ y \end{bmatrix} = 0 \qquad \begin{matrix} s_1 \leq -2 \\ s_2 \leq 3 \end{matrix}$$

*Column 3 and 4 are suitable.*

# Simplex - choosing non-basic column for pivot

Wlog, let $1 \in B$ , $k_1 = 1$, and $v(x_1)$ violates $u_1$. We need to decrease $v(x_1)$.
We call $v(x_1) - u_1$ violation difference.

Since $x_1 = \sum_{i \in NB} a_{1i}x_i$, we need to change $v(x_i)$ of some $x_i$ s.t. $a_{1i}x_i$ decreases

### Definition 2.7

*A column $i \in NB$ is suitable if*

- *$x_i$ is unbounded,*
- *$x_i = u_i$ and $a_{1i} > 0$, or*
- *$x_i = l_i$ and $a_{1i} < 0$.*

*$i$ is selected suitable column if $i$ is the smallest suitable column.*

### Example 2.6

$$\begin{bmatrix} -1 & 0 & -1 & 1 \\ 0 & -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ x \\ y \end{bmatrix} = 0 \qquad \begin{array}{c} s_1 \leq -2 \\ s_2 \leq 3 \end{array}$$

### Exercise 2.2

*Write other cases that are ignored due to "wlog"*

*Column 3 and 4 are suitable.*

# Simplex - choosing basic column for pivot

$v$ satisfies all bounds except $u_1$. Change in $v(x_i)$ may lead to violations, because $x_i$ appears in the definitions of basic variables.

# Simplex - choosing basic column for pivot

$v$ satisfies all bounds except $u_1$. Change in $v(x_i)$ may lead to violations, because $x_i$ appears in the definitions of basic variables.

Wlog, let $x_i$ is unbounded and $a_{1i} < 0$. Therefore, we need to increase $v(x_i)$.

# Simplex - choosing basic column for pivot

$v$ satisfies all bounds except $u_1$. Change in $v(x_i)$ may lead to violations, because $x_i$ appears in the definitions of basic variables.

Wlog, let $x_i$ is unbounded and $a_{1i} < 0$. Therefore, we need to increase $v(x_i)$.

## Definition 2.8

*We need to find the maximum allowed change.*

$$ch := min\bigcup_{j\in B}\{\frac{v(x_j)-u_j}{a_{k_ji}}|a_{k_ji}<0\}\cup\{\frac{v(x_j)-l_j}{a_{k_ji}}|a_{k_ji}>0\}$$

*Let $j$ be the smallest index for which the above min is attained.*

# Simplex - choosing basic column for pivot

$v$ satisfies all bounds except $u_1$. Change in $v(x_i)$ may lead to violations, because $x_i$ appears in the definitions of basic variables.

Wlog, let $x_i$ is unbounded and $a_{1i} < 0$. Therefore, we need to increase $v(x_i)$.

### Definition 2.8

*We need to find the maximum allowed change.*

$$ch := min \bigcup_{j \in B} \{ \frac{v(x_j) - u_j}{a_{k_j i}} | a_{k_j i} < 0 \} \cup \{ \frac{v(x_j) - l_j}{a_{k_j i}} | a_{k_j i} > 0 \}$$

*Let $j$ be the smallest index for which the above min is attained.*

### Example 2.7

*We change $x$(selected suitable column) to reduce violation difference.*

# Simplex - choosing basic column for pivot

$v$ satisfies all bounds except $u_1$. Change in $v(x_i)$ may lead to violations, because $x_i$ appears in the definitions of basic variables.

Wlog, let $x_i$ is unbounded and $a_{1i} < 0$. Therefore, we need to increase $v(x_i)$.

What are the other cases?

## Definition 2.8

*We need to find the maximum allowed change.*

$$ch := min \bigcup_{j \in B} \{ \frac{v(x_j) - u_j}{a_{k_j i}} | a_{k_j i} < 0 \} \cup \{ \frac{v(x_j) - l_j}{a_{k_j i}} | a_{k_j i} > 0 \}$$

*Let $j$ be the smallest index for which the above min is attained.*

## Example 2.7

*We change $x$(selected suitable column) to reduce violation difference.*

*Since $v(y) = 0$ and we are varying $x$, $s_1 = -x$ and $s_2 = x$.*
*We have $s_1 \leq -2$, and $s_2 \leq 0$.*

# Simplex - choosing basic column for pivot

$v$ satisfies all bounds except $u_1$. Change in $v(x_i)$ may lead to violations, because $x_i$ appears in the definitions of basic variables.

Wlog, let $x_i$ is unbounded and $a_{1i} < 0$. Therefore, we need to increase $v(x_i)$.

## Definition 2.8

*We need to find the maximum allowed change.*

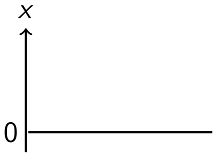
$$ch := min \bigcup_{j \in B} \{ \frac{v(x_j) - u_j}{a_{k_j i}} | a_{k_j i} < 0 \} \cup \{ \frac{v(x_j) - l_j}{a_{k_j i}} | a_{k_j i} > 0 \}$$

*Let $j$ be the smallest index for which the above min is attained.*

## Example 2.7

*We change $x$(selected suitable column) to reduce violation difference.*

*Since $v(y) = 0$ and we are varying $x$, $s_1 = -x$ and $s_2 = x$. We have $s_1 \leq -2$, and $s_2 \leq 0$.*



# Simplex - choosing basic column for pivot

$v$ satisfies all bounds except $u_1$. Change in $v(x_i)$ may lead to violations, because $x_i$ appears in the definitions of basic variables.

Wlog, let $x_i$ is unbounded and $a_{1i} < 0$. Therefore, we need to increase $v(x_i)$.

## Definition 2.8

*We need to find the maximum allowed change.*

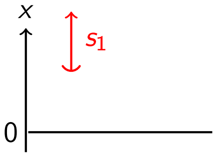
$$ch := min\bigcup_{j\in B}\{\frac{v(x_j)-u_j}{a_{k_ji}}|a_{k_ji}<0\}\cup\{\frac{v(x_j)-l_j}{a_{k_ji}}|a_{k_ji}>0\}$$

*Let $j$ be the smallest index for which the above min is attained.*

## Example 2.7

*We change $x$(selected suitable column) to reduce violation difference.*

*Since $v(y) = 0$ and we are varying $x$, $s_1 = -x$ and $s_2 = x$.*
*We have $s_1 \leq -2$, and $s_2 \leq 0$.*



# Simplex - choosing basic column for pivot

$v$ satisfies all bounds except $u_1$. Change in $v(x_i)$ may lead to violations, because $x_i$ appears in the definitions of basic variables.

Wlog, let $x_i$ is unbounded and $a_{1i} < 0$. Therefore, we need to increase $v(x_i)$.

What are the other cases?

### Definition 2.8

*We need to find the maximum allowed change.*

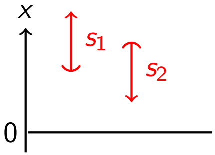
$$ch := min \bigcup_{j \in B} \{ \frac{v(x_j) - u_j}{a_{k_j i}} | a_{k_j i} < 0\} \cup \{ \frac{v(x_j) - l_j}{a_{k_j i}} | a_{k_j i} > 0\}$$

*Let $j$ be the smallest index for which the above min is attained.*

### Example 2.7

*We change $x$(selected suitable column) to reduce violation difference.*

*Since $v(y) = 0$ and we are varying $x$, $s_1 = -x$ and $s_2 = x$.*
*We have $s_1 \leq -2$, and $s_2 \leq 0$.*
*Therefore, $s_1$ allows $2 \leq x$*



# Simplex - choosing basic column for pivot

$v$ satisfies all bounds except $u_1$. Change in $v(x_i)$ may lead to violations, because $x_i$ appears in the definitions of basic variables.

Wlog, let $x_i$ is unbounded and $a_{1i} < 0$. Therefore, we need to increase $v(x_i)$.

What are the other cases?

## Definition 2.8

*We need to find the maximum allowed change.*

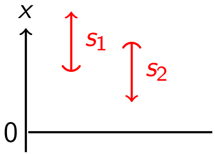
$$ch := min \bigcup_{j \in B} \{ \frac{v(x_j) - u_j}{a_{k_j i}} | a_{k_j i} < 0 \} \cup \{ \frac{v(x_j) - l_j}{a_{k_j i}} | a_{k_j i} > 0 \}$$

*Let $j$ be the smallest index for which the above min is attained.*

## Example 2.7

*We change $x$(selected suitable column) to reduce violation difference.*

*Since $v(y) = 0$ and we are varying $x$, $s_1 = -x$ and $s_2 = x$.*
*We have $s_1 \leq -2$, and $s_2 \leq 0$.*
*Therefore, $s_1$ allows $2 \leq x$ and $s_2$ allows $x \leq 3$.*



# Simplex - choosing basic column for pivot

$v$ satisfies all bounds except $u_1$. Change in $v(x_i)$ may lead to violations, because $x_i$ appears in the definitions of basic variables.

Wlog, let $x_i$ is unbounded and $a_{1i} < 0$. Therefore, we need to increase $v(x_i)$.

What are the other cases?

## Definition 2.8

*We need to find the maximum allowed change.*

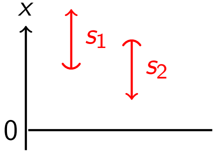
$$ch := min \bigcup_{j \in B} \{ \frac{v(x_j) - u_j}{a_{k_j i}} | a_{k_j i} < 0\} \cup \{ \frac{v(x_j) - l_j}{a_{k_j i}} | a_{k_j i} > 0\}$$

*Let $j$ be the smallest index for which the above min is attained.*

## Example 2.7

*We change $x$(selected suitable column) to reduce violation difference.*

*Since $v(y) = 0$ and we are varying $x$, $s_1 = -x$ and $s_2 = x$.*
*We have $s_1 \leq -2$, and $s_2 \leq 0$.*
*Therefore, $s_1$ allows $2 \leq x$ and $s_2$ allows $x \leq 3$.*
*Clearly, $ch = 3$ and $j = 2$.*



# Simplex - pivoting operation to reduce violation difference

We carry $ch$ and $j$ from the last slide. Wlog, $ch = \frac{v(x_j) - u_j}{a_{k_j i}}$.

# Simplex - pivoting operation to reduce violation difference

We carry $ch$ and $j$ from the last slide. Wlog, $ch = \frac{v(x_j) - u_j}{a_{k_j i}}$.

Now there are three possibilities

- If $ch = u_i = +\infty$, pivot between $i$ and 1 and activate $u_1$
- If $ch > (u_i - l_i)$, we assign $v(x_i) = l_i$ update values of basic variables and no pivoting
- Otherwise, we choose $j$ for the basic variable for pivoting and apply pivoting between $i$ and $j$. We activate $u_j$ bound on variable $x_j$.

# Simplex - pivoting operation to reduce violation difference

We carry $ch$ and $j$ from the last slide. Wlog, $ch = \frac{v(x_j) - u_j}{a_{k_j i}}$.

Now there are three possibilities

- If $ch = u_i = +\infty$, pivot between $i$ and 1 and activate $u_1$
- If $ch > (u_i - l_i)$, we assign $v(x_i) = l_i$ update values of basic variables and no pivoting
- Otherwise, we choose $j$ for the basic variable for pivoting and apply pivoting between $i$ and $j$. We activate $u_j$ bound on variable $x_j$.

If violation still persists then look for further pivot operations.

### Theorem 2.1

*Pivoting operation never increases violation difference*

# Simplex - pivoting operation to reduce violation difference

We carry $ch$ and $j$ from the last slide. Wlog, $ch = \frac{v(x_j)-u_j}{a_{k_j i}}$.

Now there are three possibilities

- If $ch = u_i = +\infty$, pivot between $i$ and 1 and activate $u_1$
- If $ch > (u_i - l_i)$, we assign $v(x_i) = l_i$ update values of basic variables and no pivoting
- Otherwise, we choose $j$ for the basic variable for pivoting and apply pivoting between $i$ and $j$. We activate $u_j$ bound on variable $x_j$.

If violation still persists then look for further pivot operations.

### Theorem 2.1

*Pivoting operation never increases violation difference*

### Example 2.8

*After pivoting between* 3 *and* 2*.*

$$\begin{bmatrix} -1 & -1 & 0 & 1 \\ 0 & 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ x \\ y \end{bmatrix} = 0 \qquad \begin{matrix} s_1 \leq -2 \\ s_2 \leq 3^* \end{matrix}$$

# Simplex - pivoting operation to reduce violation difference

We carry $ch$ and $j$ from the last slide. Wlog, $ch = \frac{v(x_j)-u_j}{a_{k_j i}}$.

Now there are three possibilities

- If $ch = u_i = +\infty$, pivot between $i$ and 1 and activate $u_1$
- If $ch > (u_i - l_i)$, we assign $v(x_i) = l_i$ update values of basic variables and no pivoting
- Otherwise, we choose $j$ for the basic variable for pivoting and apply pivoting between $i$ and $j$. We activate $u_j$ bound on variable $x_j$.

If violation still persists then look for further pivot operations.

### Theorem 2.1

*Pivoting operation never increases violation difference*

### Example 2.8

*After pivoting between* 3 *and* 2*.*

$$\begin{bmatrix} -1 & -1 & 0 & 1 \\ 0 & 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ x \\ y \end{bmatrix} = 0 \qquad \begin{array}{c} s_1 \leq -2 \\ s_2 \leq 3^* \end{array}$$

*Now $v$ is satisfying.*

# Incremental simplex and single violation assumption

Before adding next atom, incremental simplex finds a solution of atoms added so far.

# Incremental simplex and single violation assumption

Before adding next atom, incremental simplex finds a solution of atoms added so far.

New atom $ax \leq b$ is added in the following steps.

- A fresh slack variable $s$ is introduced
- $s = ax$ is added as a row in $A$ and $s \leq b$ is added in the bounds
- The new row may have non-zeros in basic columns. They are removed by row operations on the new row.
- $s$ is added to $B$.

# Incremental simplex and single violation assumption

Before adding next atom, incremental simplex finds a solution of atoms added so far.

New atom $ax \leq b$ is added in the following steps.

- A fresh slack variable $s$ is introduced
- $s = ax$ is added as a row in $A$ and $s \leq b$ is added in the bounds
- The new row may have non-zeros in basic columns. They are removed by row operations on the new row.
- $s$ is added to $B$.

Therefore, current assignment can only violate the bound of $s$.

# Example: inserting a new atom

## Example 2.9

*Let us add atom* $-2x - y \leq -8$ *in our running example*

# Example: inserting a new atom

## Example 2.9

*Let us add atom $-2x - y \leq -8$ in our running example*
*We add a new slack variable $s_3$ and a corresponding row.*

$$\begin{bmatrix} -1 & 0 & 0 & -2 & -1 \\ 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} s_3 \\ s_1 \\ s_2 \\ x \\ y \end{bmatrix} = 0 \qquad \begin{array}{c} s_3 \leq -8 \\ s_1 \leq -2 \\ s_2 \leq 3^* \end{array}$$

# Example: inserting a new atom

## Example 2.9

*Let us add atom $-2x - y \leq -8$ in our running example*
*We add a new slack variable $s_3$ and a corresponding row.*

$$\begin{bmatrix} -1 & 0 & 0 & -2 & -1 \\ 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} s_3 \\ s_1 \\ s_2 \\ x \\ y \end{bmatrix} = 0 \qquad \begin{matrix} s_3 \leq -8 \\ s_1 \leq -2 \\ s_2 \leq 3^* \end{matrix}$$

*After remove basic variables ($\{s_1, x\}$) from the top row*

$$\begin{bmatrix} -1 & 0 & -2 & 0 & -1 \\ 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} s_3 \\ s_1 \\ s_2 \\ x \\ y \end{bmatrix} = 0 \qquad \begin{matrix} s_3 \leq -8 \\ s_1 \leq -2 \\ s_2 \leq 3^* \end{matrix}$$

# Example: inserting a new atom

## Example 2.9

*Let us add atom $-2x - y \leq -8$ in our running example*
*We add a new slack variable $s_3$ and a corresponding row.*

$$\begin{bmatrix} -1 & 0 & 0 & -2 & -1 \\ 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} s_3 \\ s_1 \\ s_2 \\ x \\ y \end{bmatrix} = 0 \qquad \begin{matrix} s_3 \leq -8 \\ s_1 \leq -2 \\ s_2 \leq 3^* \end{matrix}$$

*After remove basic variables ($\{s_1, x\}$) from the top row*

$$\begin{bmatrix} -1 & 0 & -2 & 0 & -1 \\ 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} s_3 \\ s_1 \\ s_2 \\ x \\ y \end{bmatrix} = 0 \qquad \begin{matrix} s_3 \leq -8 \\ s_1 \leq -2 \\ s_2 \leq 3^* \end{matrix}$$

## Exercise 2.3

*Now $s_3$ is violated. Pivot if possible.*

# Simplex - iterations

Simplex is a sequence of pivot operations

- If simplex fails to find a suitable column for some violation then input is unsat.
- If a state is reached without violation then $v$ is a satisfying assignment.

## Example 2.10

*$s_3$ is still in violation.*

$$\begin{bmatrix} -1 & -1 & -3 & 0 & 0 \\ 0 & 1 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} s_3 \\ s_1 \\ s_2 \\ x \\ y \end{bmatrix} = 0 \qquad \begin{array}{c} s_3 \leq -8 \\ s_1 \leq -2^* \\ s_2 \leq 3^* \end{array}$$

*Now, we can not find a suitable column.*

*Therefore, the constraints are unsat.*

# Example 2.11

*Run simplex on* $x_1 \leq 5 \wedge 4x_1 + x_2 \leq 25 \wedge -2x_1 - x_2 \leq -25$

## Example 2.11

*Run simplex on* $x_1 \leq 5 \wedge 4x_1 + x_2 \leq 25 \wedge -2x_1 - x_2 \leq -25$

After push of the first atom

$$\begin{bmatrix} -1 & 1 \end{bmatrix} \begin{bmatrix} s_1 \\ x_1 \end{bmatrix} = 0 \qquad s_1 \leq 5 \qquad v = \{x_1 \mapsto 0, s_1 \mapsto 0\}$$

## Example 2.11

*Run simplex on* $x_1 \leq 5 \wedge 4x_1 + x_2 \leq 25 \wedge -2x_1 - x_2 \leq -25$

After push of the first atom

$$\begin{bmatrix} -1 & 1 \end{bmatrix} \begin{bmatrix} s_1 \\ x_1 \end{bmatrix} = 0 \qquad s_1 \leq 5 \qquad v = \{x_1 \mapsto 0, s_1 \mapsto 0\}$$

After push of the second atom

$$\begin{bmatrix} -1 & 0 & 4 & 1 \\ 0 & -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} s_2 \\ s_1 \\ x_1 \\ x_2 \end{bmatrix} = 0 \qquad \begin{matrix} s_1 \leq 5 \\ s_2 \leq 25 \end{matrix} \qquad v = \{\_ \mapsto 0\}$$

## Example 2.11

*Run simplex on* $x_1 \leq 5 \wedge 4x_1 + x_2 \leq 25 \wedge -2x_1 - x_2 \leq -25$

After push of the first atom

$$\begin{bmatrix} -1 & 1 \end{bmatrix}\begin{bmatrix} s_1 \\ x_1 \end{bmatrix} = 0 \qquad s_1 \leq 5 \qquad v = \{x_1 \mapsto 0, s_1 \mapsto 0\}$$

After push of the second atom

$$\begin{bmatrix} -1 & 0 & 4 & 1 \\ 0 & -1 & 1 & 0 \end{bmatrix}\begin{bmatrix} s_2 \\ s_1 \\ x_1 \\ x_2 \end{bmatrix} = 0 \qquad \begin{array}{c} s_1 \leq 5 \\ s_2 \leq 25 \end{array} \qquad v = \{\_ \mapsto 0\}$$

After push of the last atom

$$\begin{bmatrix} -1 & 0 & 0 & -2 & -1 \\ 0 & -1 & 0 & 4 & 1 \\ 0 & 0 & -1 & 1 & 0 \end{bmatrix}\begin{bmatrix} s_3 \\ s_2 \\ s_1 \\ x_1 \\ x_2 \end{bmatrix} = 0 \qquad \begin{array}{c} s_1 \leq 5 \\ s_2 \leq 25 \\ s_3 \leq -25 \end{array} \qquad v = \{\_ \mapsto 0\}$$

## Example 2.11

*Run simplex on* $x_1 \leq 5 \wedge 4x_1 + x_2 \leq 25 \wedge -2x_1 - x_2 \leq -25$

After push of the first atom

$$\begin{bmatrix} -1 & 1 \end{bmatrix} \begin{bmatrix} s_1 \\ x_1 \end{bmatrix} = 0 \qquad s_1 \leq 5 \qquad v = \{x_1 \mapsto 0, s_1 \mapsto 0\}$$

After push of the second atom

$$\begin{bmatrix} -1 & 0 & 4 & 1 \\ 0 & -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} s_2 \\ s_1 \\ x_1 \\ x_2 \end{bmatrix} = 0 \qquad \begin{array}{c} s_1 \leq 5 \\ s_2 \leq 25 \end{array} \qquad v = \{\_ \mapsto 0\}$$

After push of the last atom

$$\begin{bmatrix} -1 & 0 & 0 & -2 & -1 \\ 0 & -1 & 0 & 4 & 1 \\ 0 & 0 & -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} s_3 \\ s_2 \\ s_1 \\ x_1 \\ x_2 \end{bmatrix} = 0 \qquad \begin{array}{c} s_1 \leq 5 \\ s_2 \leq 25 \\ s_3 \leq -25 \end{array} \qquad v = \{\_ \mapsto 0\}$$

## Exercise 2.4

*Finish the run*

# Popping atoms

If we want to remove some atom from simplex state, we

# Popping atoms

If we want to remove some atom from simplex state, we

- make the corresponding slack variable basic variable and
- remove the corresponding row and bound constraints on the slack variable

# Unsat core extraction

If input is unsat, then there must be a basic variable whose value is violated.

- we collect the slack variables that appear in the row that defines the basic variable
- the atoms corresponding to the slack variables are part of unsat core

# Simplex- geometric intuition

Now we will connect the algorithm with a geometric intuition.

# Simplex- geometric intuition

Now we will connect the algorithm with a geometric intuition.

For ease of exposition, we will assume that all $l_i$s and $u_i$s are finite.
This restriction can be easily dropped.

# Simplex- geometric intuition

Now we will connect the algorithm with a geometric intuition.

For ease of exposition, we will assume that all $l_i$s and $u_i$s are finite.
This restriction can be easily dropped.

We will add supper script $p$ to various objects to denote their value at $p$th iteration.

# Simplex- geometric intuition

Now we will connect the algorithm with a geometric intuition.

For ease of exposition, we will assume that all $l_i$s and $u_i$s are finite.
This restriction can be easily dropped.

We will add supper script $p$ to various objects to denote their value at $p$th iteration.

For example, $A^p$ is the value of $A$ at $p$th iteration.

# Simplex - geometric intuition: meaning of suitable column

Let us introduce the following object in each iterations

- Let $\mu^p$ be a row vector of length $2(m+n)$ s.t.

$$\begin{bmatrix} 1 & \underbrace{0}_{m-1} & \mu^p \end{bmatrix} \begin{bmatrix} A^p \\ \mathrm{I} \\ -\mathrm{I} \end{bmatrix} = \begin{bmatrix} -1 & \underbrace{0}_{m+n-1} \end{bmatrix}$$

$$\mu^p_k = \begin{cases} -A^p_{1k} & k \in NB^p \text{ and } u_k \text{ is active at } p\text{th iteration} \\ A^p_{1(k-(m+m))} & (k-(m+n)) \in NB^p \text{ and } l_{k-(m+n)} \text{ is active at } p\text{th} \\ 0 & \text{otherwise} \end{cases}$$

# Simplex - geometric intuition: meaning of suitable column

Let us introduce the following object in each iterations

- Let $\mu^p$ be a row vector of length $2(m+n)$ s.t.

$$\begin{bmatrix} 1 & \underbrace{0}_{m-1} & \mu^p \end{bmatrix} \begin{bmatrix} A^p \\ \mathtt{I} \\ -\mathtt{I} \end{bmatrix} = \begin{bmatrix} -1 & \underbrace{0}_{m+n-1} \end{bmatrix}$$

$$\mu^p_k = \begin{cases} -A^p_{1k} & k \in NB^p \text{ and } u_k \text{ is active at } p\text{th iteration} \\ A^p_{1(k-(m+m))} & (k-(m+n)) \in NB^p \text{ and } l_{k-(m+n)} \text{ is active at } p\text{th} \\ 0 & \text{otherwise} \end{cases}$$

## Theorem 2.2

*Let $i'$ be the smallest index for which $\mu^p$ has a negative number and $i$ be the selected suitable column for the next pivoting. Then,*

$$i = \begin{cases} i' & i' \leq m+n \\ i'-(m+n) & \textit{otherwise.} \end{cases}$$

# Simplex - geometric intuition: meaning of suitable column

Let us introduce the following object in each iterations

- Let $\mu^p$ be a row vector of length $2(m+n)$ s.t.

$$\begin{bmatrix} 1 & \underbrace{0}_{m-1} & \mu^p \end{bmatrix} \begin{bmatrix} A^p \\ \mathrm{I} \\ -\mathrm{I} \end{bmatrix} = \begin{bmatrix} -1 & \underbrace{0}_{m+n-1} \end{bmatrix}$$

$$\mu^p_k = \begin{cases} -A^p_{1k} & k \in NB^p \text{ and } u_k \text{ is active at } p\text{th iteration} \\ A^p_{1(k-(m+m))} & (k-(m+n)) \in NB^p \text{ and } l_{k-(m+n)} \text{ is active at } pth \\ 0 & \text{otherwise} \end{cases}$$

## Theorem 2.2

*Let $i'$ be the smallest index for which $\mu^p$ has a negative number and $i$ be the selected suitable column for the next pivoting. Then,*

$$i = \begin{cases} i' & i' \leq m+n \\ i'-(m+n) & \textit{otherwise.} \end{cases}$$

## Exercise 2.5

*Prove the above.*

# Simplex - geometric intuition: update direction

Selection of suitable column induces the idea of update direction

- Let $y^p$ be a vector of length $m + n$. $y^p$ indicates the direction of change due to pivot operation after $p$th iteration.

# Simplex - geometric intuition: update direction

Selection of suitable column induces the idea of update direction

- Let $y^p$ be a vector of length $m+n$. $y^p$ indicates the direction of change due to pivot operation after $p$th iteration.
  Let $i \in NB^p$ be the selected suitable column.
  - $l_i$ is active

$$y_j^p = \begin{cases} 1 & j = i, \\ A^p_{k_j i} & j \in B^p \\ 0 & \text{otherwise} \end{cases}$$

# Simplex - geometric intuition: update direction

Selection of suitable column induces the idea of update direction

- Let $y^p$ be a vector of length $m+n$. $y^p$ indicates the direction of change due to pivot operation after $p$th iteration.
  Let $i \in NB^p$ be the selected suitable column.
  - $l_i$ is active

$$y^p_j = \begin{cases} 1 & j = i, \\ A^p_{k_j i} & j \in B^p \\ 0 & \text{otherwise} \end{cases}$$

  - $u_i$ is active

$$y^p_j = \begin{cases} -1 & j = i, \\ -A^p_{k_j i} & j \in B^p \\ 0 & \text{otherwise} \end{cases}$$

# Simplex - geometric intuition: update direction

Selection of suitable column induces the idea of update direction

- Let $y^p$ be a vector of length $m+n$. $y^p$ indicates the direction of change due to pivot operation after $p$th iteration.
  Let $i \in NB^p$ be the selected suitable column.
  - $l_i$ is active
  $$y^p_j = \begin{cases} 1 & j = i, \\ A^p_{k_j i} & j \in B^p \\ 0 & \text{otherwise} \end{cases}$$
  - $u_i$ is active
  $$y^p_j = \begin{cases} -1 & j = i, \\ -A^p_{k_j i} & j \in B^p \\ 0 & \text{otherwise} \end{cases}$$

Exercise 2.6

*Show* $[-1\ 0]y^p > 0$

# Simplex - geometric intuition: limit on update

- The change in direction $y$ only violate bounds on basic variables

$$ch := min \bigcup_{j \in 1..m} \{ \frac{u_j - v(x_j)}{y_j^p} | y_j^p > 0 \} \cup \{ \frac{l_j - v(x_j)}{-y_j^p} | y_j^p > 0 \}$$

Let $j$ be the smallest index for which the above min is attained, which is used for pivoting.

## Exercise 2.7

*Check the basis column $j$ selected above is same as the pivot basis column selected earlier*

# Simplex - termination

## Lemma 2.1

*Simplex terminates.*

## Proof.

In every step the violation difference $(v(x_1) - u_1)$ reduces or stays same.

# Simplex - termination

## Lemma 2.1

*Simplex terminates.*

## Proof.

In every step the violation difference $(v(x_1) - u_1)$ reduces or stays same.

Since there are finitely many states, simplex terminates if $v(x_1) - u_1$ cannot stay same forever.

# Simplex - termination

## Lemma 2.1

*Simplex terminates.*

## Proof.

In every step the violation difference $(v(x_1) - u_1)$ reduces or stays same.

Since there are finitely many states, simplex terminates if $v(x_1) - u_1$ cannot stay same forever.

For that we prove that same state can not repeat in a simplex run.

# Simplex - termination

## Lemma 2.1

*Simplex terminates.*

## Proof.

In every step the violation difference $(v(x_1) - u_1)$ reduces or stays same.

Since there are finitely many states, simplex terminates if $v(x_1) - u_1$ cannot stay same forever.

For that we prove that same state can not repeat in a simplex run.

Wlog, let us suppose the states of $s$th and $t$th iterations of simplex is same and there is no change in $v(x_1) - u_1$ from $p$ to $q$.

# Simplex - termination

## Lemma 2.1

*Simplex terminates.*

## Proof.

In every step the violation difference $(v(x_1) - u_1)$ reduces or stays same.

Since there are finitely many states, simplex terminates if $v(x_1) - u_1$ cannot stay same forever.

For that we prove that same state can not repeat in a simplex run.

Wlog, let us suppose the states of $s$th and $t$th iterations of simplex is same and there is no change in $v(x_1) - u_1$ from $p$ to $q$.

Let $r$ be the largest index column which left and reentered $NB$ at iteration $p$ and $q$ respectively, where $s \leq p < q \leq t$.

# Simplex - termination(contd.)

Now Consider,

$$\begin{bmatrix} 1 & \underbrace{0}_{m-1} & \mu^p \end{bmatrix} \begin{bmatrix} A^p \\ \mathtt{I} \\ -\mathtt{I} \end{bmatrix} y^q = [-1 \;\; 0] y^q > 0$$

Now we will show that the above term cannot be $> 0$.

# Simplex - termination(contd.)

Now Consider,

$$\begin{bmatrix} 1 & \underbrace{0}_{m-1} & \mu^p \end{bmatrix} \begin{bmatrix} A^p \\ \mathtt{I} \\ -\mathtt{I} \end{bmatrix} y^q = [-1 \;\; 0] y^q > 0$$

Now we will show that the above term cannot be $> 0$.

Let us apply a different calculation on the above term.

$$\begin{bmatrix} 1 & \underbrace{0}_{m-1} & \mu^p \end{bmatrix} \begin{bmatrix} A^p \\ \mathtt{I} \\ -\mathtt{I} \end{bmatrix} y^q$$

# Simplex - termination(contd.)

Now Consider,

$$\begin{bmatrix} 1 & \underbrace{0}_{m-1} & \mu^p \end{bmatrix} \begin{bmatrix} A^p \\ \mathrm{I} \\ -\mathrm{I} \end{bmatrix} y^q = [-1 \;\; 0] y^q > 0$$

Now we will show that the above term cannot be $> 0$.

Let us apply a different calculation on the above term.

$$\begin{bmatrix} 1 & \underbrace{0}_{m-1} & \mu^p \end{bmatrix} \begin{bmatrix} A^p \\ \mathrm{I} \\ -\mathrm{I} \end{bmatrix} y^q = \begin{bmatrix} 1 & \underbrace{0}_{m-1} & \mu^p \end{bmatrix} \begin{bmatrix} A^p y^q \\ \mathrm{I} y^q \\ -\mathrm{I} y^q \end{bmatrix}$$

# Simplex - termination(contd.)

Now Consider,

$$\begin{bmatrix} 1 & \underbrace{0}_{m-1} & \mu^p \end{bmatrix} \begin{bmatrix} A^p \\ \mathtt{I} \\ -\mathtt{I} \end{bmatrix} y^q = [-1 \ \ 0] y^q > 0$$

Now we will show that the above term cannot be $> 0$.

Let us apply a different calculation on the above term.

$$\begin{bmatrix} 1 & \underbrace{0}_{m-1} & \mu^p \end{bmatrix} \begin{bmatrix} A^p \\ \mathtt{I} \\ -\mathtt{I} \end{bmatrix} y^q = \begin{bmatrix} 1 & \underbrace{0}_{m-1} & \mu^p \end{bmatrix} \begin{bmatrix} A^p y^q \\ \mathtt{I} y^q \\ -\mathtt{I} y^q \end{bmatrix}$$

$$= \begin{bmatrix} 1 & \underbrace{0}_{m-1} & \mu^p \end{bmatrix} \begin{bmatrix} 0 \\ \mathtt{I} y^q \\ -\mathtt{I} y^q \end{bmatrix} = \mu^p \begin{bmatrix} y^q \\ -y^q \end{bmatrix}$$

# Termination

Let $\hat{y}^p \triangleq \begin{bmatrix} y^q \\ -y^q \end{bmatrix}$

Now we show every $\mu_j \hat{y}_j^p$ is non-positive.

- $j \in B^p$ or $j - n \in B^p$ or $j$th bound is inactive, $\mu_j^p = 0$
- $j \in NB^p$ or $j - n \in NB^p$, and $j$th bound is active
  - $j > r$, $y_i^q = 0$
  - $j = r$, $\underbrace{u_r^p < 0}$ and $y_r^q > 0(why?)$

    because $r$ is selected to leave $NB^p$

  - $j > r$, $\overbrace{u_j^p \geq 0}$ and $y_j^q \leq 0$ (why?)

# Simplex - complexity

Simplex is average time linear and worst case exponential.

# Simplex - complexity

Simplex is average time linear and worst case exponential.

In practice, none of the above complexities are observed

# Simplex - complexity

Simplex is average time linear and worst case exponential.

In practice, none of the above complexities are observed

Ellipsoid method is a polynomial time algorithm for linear constraints. In practice, simplex performs better in many classes of problems.

# End of Lecture 2