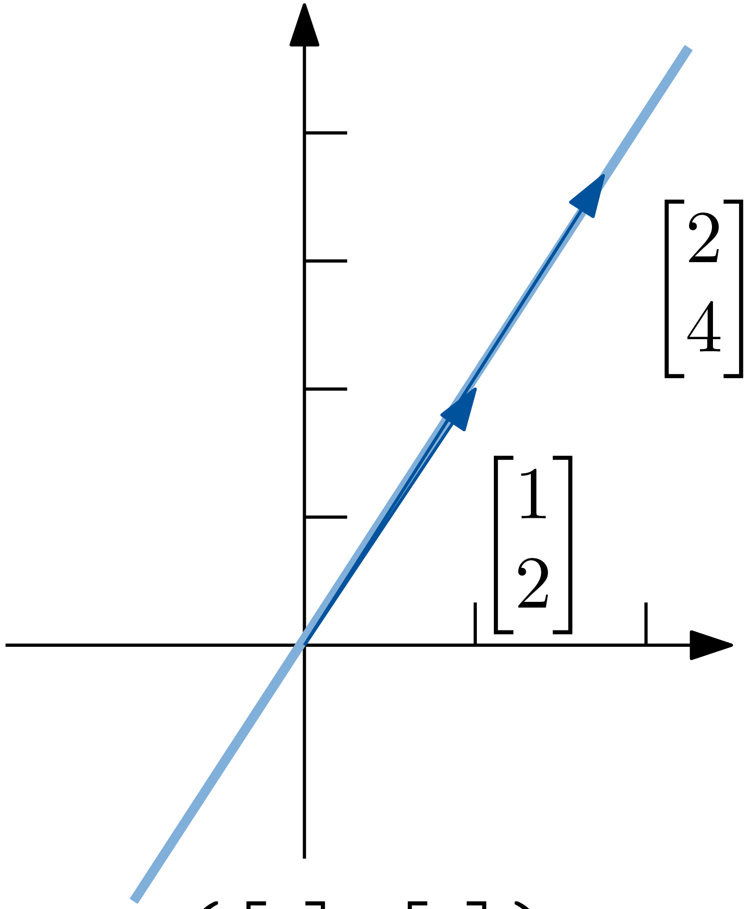
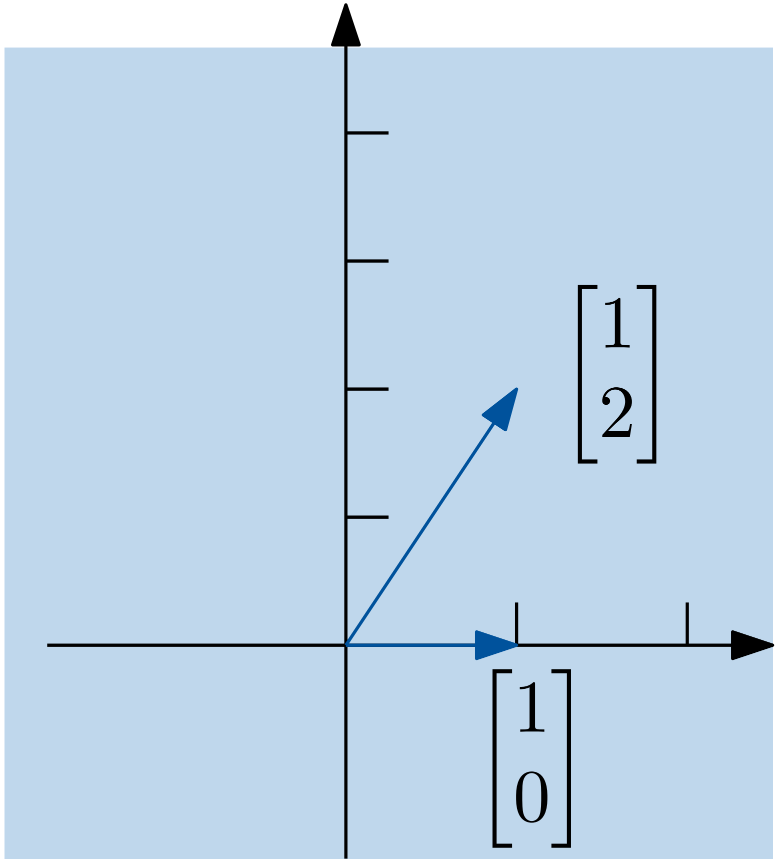
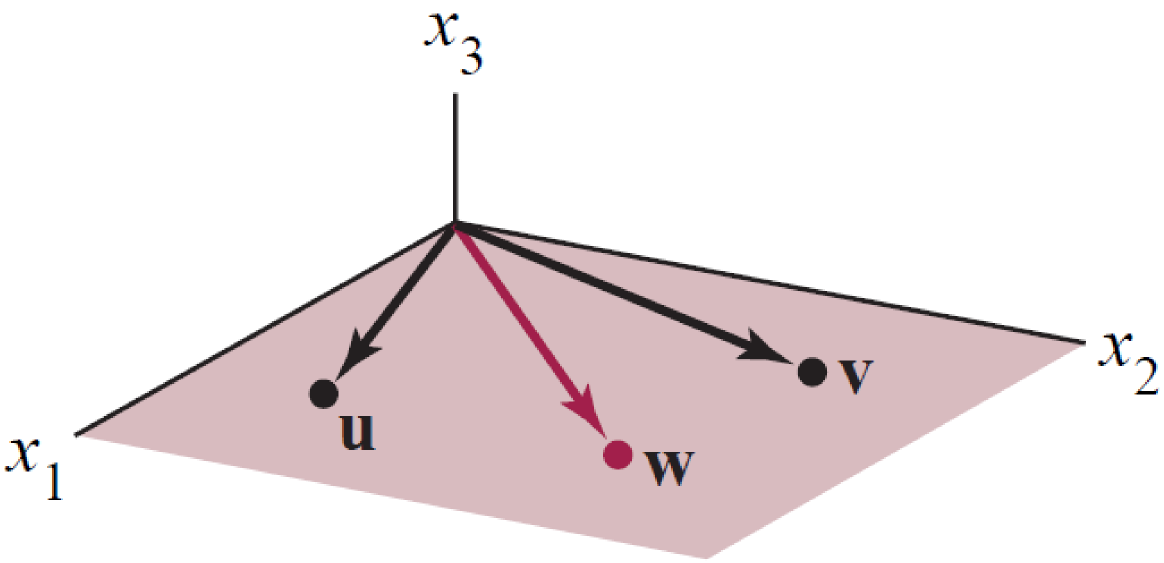

$\begin{bmatrix}2\\4\end{bmatrix}$
$\begin{bmatrix}1\\2\end{bmatrix}$

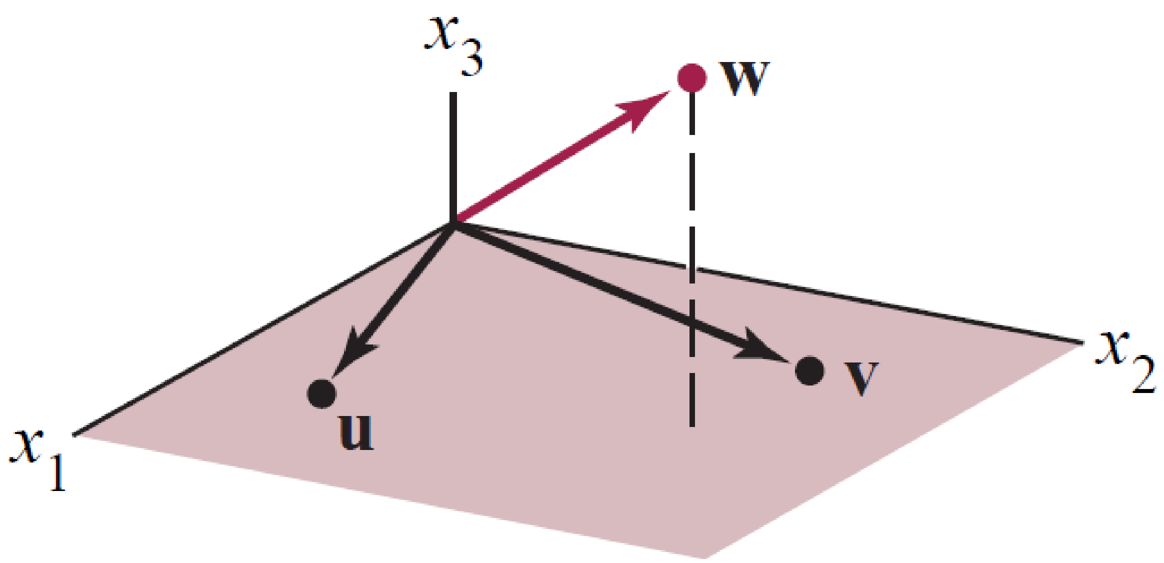
$$\text{Span}\left\{\begin{bmatrix}1\\2\end{bmatrix},\begin{bmatrix}2\\4\end{bmatrix}\right\} = \text{a line}$$


$\begin{bmatrix}1\\2\end{bmatrix}$
$\begin{bmatrix}1\\0\end{bmatrix}$

$$\text{Span}\left\{\begin{bmatrix}1\\2\end{bmatrix},\begin{bmatrix}1\\0\end{bmatrix}\right\} = \mathbb{R}^2$$



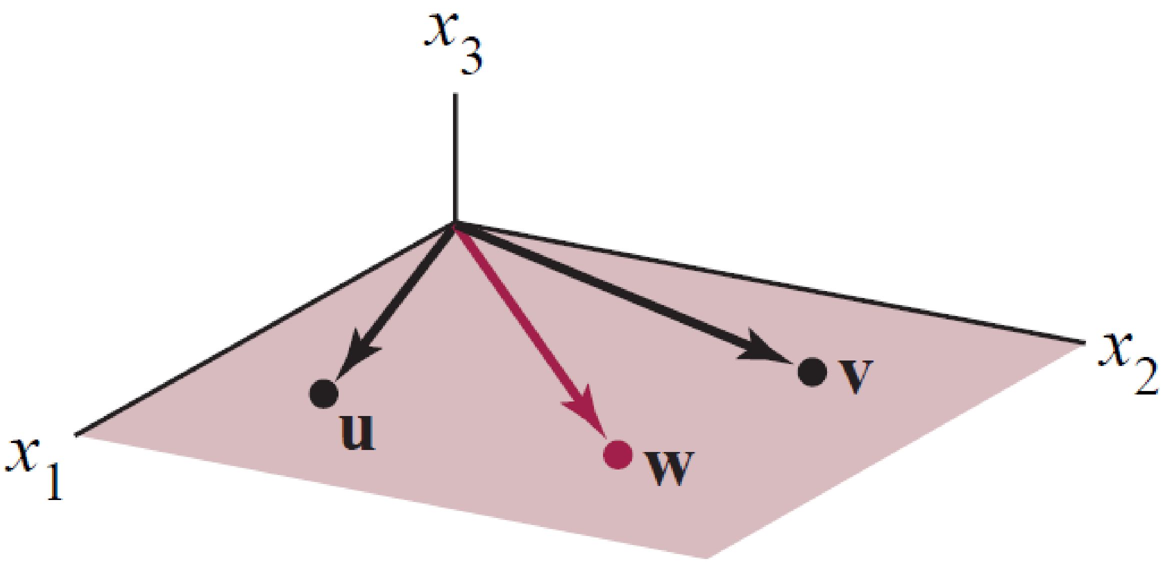
Span $\{\mathbf{u}, \mathbf{v}, \mathbf{w}\}$ = Span $\{\mathbf{u}, \mathbf{v}\}$ = a plane

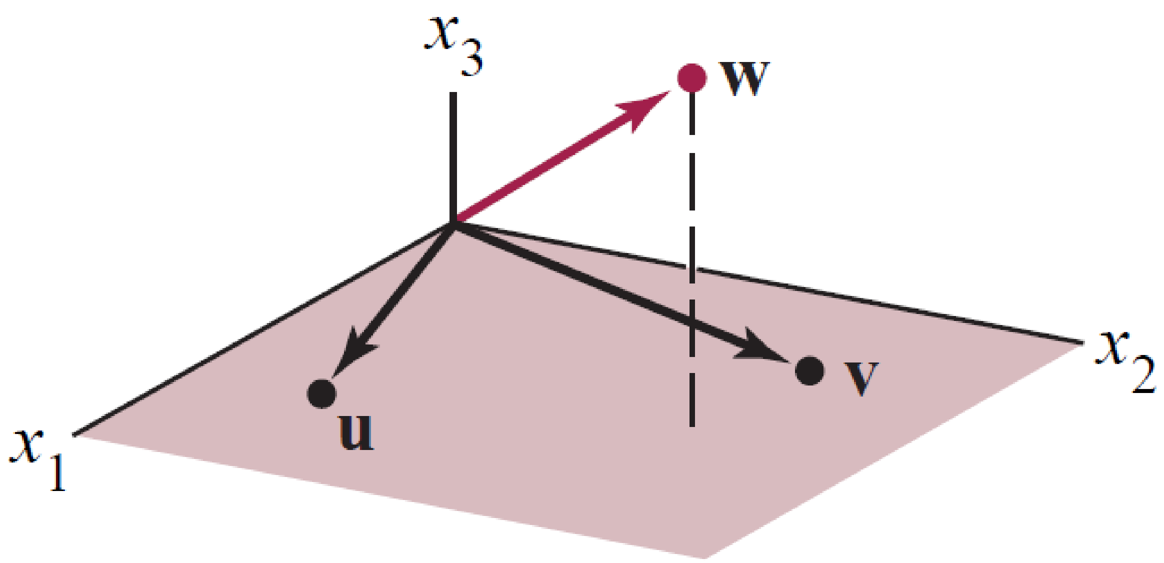


Span $\{\mathbf{u}, \mathbf{v}, \mathbf{w}\} = \mathbb{R}^3$

When do $n$ vectors span $\mathbb{R}^n$?

How to find an efficient spanning set?





$$\text{Span}\,\{\mathbf{u},\mathbf{v},\mathbf{w}\} = \text{Span}\,\{\mathbf{u},\mathbf{v}\} = \text{a plane}$$

$$\text{Span}\,\{\mathbf{u},\mathbf{v},\mathbf{w}\} = \mathbb{R}^3$$

When do $n$ vectors span $\mathbb{R}^n$? When they are a linearly independent set.

How to find an efficient spanning set? The casting out algorithm.

# §1.7: Linear Independence

**Definition**: A set of vectors $\{\mathbf{v_1}, \ldots, \mathbf{v_p}\}$ is *linearly independent* if the only solution to the vector equation

$$x_1\mathbf{v_1} + \cdots + x_p\mathbf{v_p} = \mathbf{0}$$

is the trivial solution $(x_1 = \cdots = x_p = 0)$.

# §1.7: Linear Independence

**Definition**: A set of vectors $\{\mathbf{v_1}, \dots, \mathbf{v_p}\}$ is *linearly independent* if the only solution to the vector equation

$$x_1\mathbf{v_1} + \cdots + x_p\mathbf{v_p} = \mathbf{0}$$

is the trivial solution $(x_1 = \cdots = x_p = 0)$.

The opposite of linearly independent is linearly dependent:

**Definition**: A set of vectors $\{\mathbf{v_1}, \dots, \mathbf{v_p}\}$ is *linearly dependent* if there are weights $c_1, \dots, c_p$, not all zero, such that

$$c_1\mathbf{v_1} + \cdots + c_p\mathbf{v_p} = \mathbf{0}.$$

The equation $c_1\mathbf{v_1} + \cdots + c_p\mathbf{v_p} = \mathbf{0}$ is a linear dependence relation.

$$x_1\mathbf{v_1} + \cdots + x_p\mathbf{v_p} = \mathbf{0}$$

Only solution is $x_1 = \cdots = x_p = 0$
→ linearly independent

There is a solution with some $x_i \neq 0$
→ linearly dependent

**Example**: The set $\left\{\begin{bmatrix}1\\2\end{bmatrix}, \begin{bmatrix}2\\4\end{bmatrix}\right\}$ is linearly dependent because

$$2\begin{bmatrix}1\\2\end{bmatrix} + (-1)\begin{bmatrix}2\\4\end{bmatrix} = \begin{bmatrix}0\\0\end{bmatrix}.$$

**Example**: The set $\left\{\begin{bmatrix}1\\2\end{bmatrix}, \begin{bmatrix}1\\0\end{bmatrix}\right\}$ is linearly independent because

$$x_1\begin{bmatrix}1\\2\end{bmatrix} + x_2\begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}0\\0\end{bmatrix} \implies \begin{aligned} x_1 + x_2 &= 0 \\ 2x_1 \qquad &= 0 \end{aligned} \implies x_1 = 0, x_2 = 0.$$

Some easy cases:

- Sets containing the zero vector $\{\mathbf{0}, \mathbf{v_2}, \ldots, \mathbf{v_p}\}$:

$$x_1\mathbf{0} + x_2\mathbf{v_2} + \cdots + x_p\mathbf{v_p} = \mathbf{0}$$

Some easy cases:

- Sets containing the zero vector $\{\mathbf{0}, \mathbf{v_2}, \dots, \mathbf{v_p}\}$:

$$(1)\mathbf{0} + (0)\mathbf{v_2} + \cdots + (0)\mathbf{v_p} = \mathbf{0} \qquad \text{linearly dependent}$$

Some easy cases:

- Sets containing the zero vector $\{\mathbf{0}, \mathbf{v_2}, \ldots, \mathbf{v_p}\}$:

$$(1)\mathbf{0} + (0)\mathbf{v_2} + \cdots + (0)\mathbf{v_p} = \mathbf{0}$$

linearly dependent

- Sets containing one vector $\{\mathbf{v}\}$:

$$x\mathbf{v} = \mathbf{0}$$

linearly independent if $\mathbf{v} \neq \mathbf{0}$

$\begin{bmatrix} xv_1 \\ \vdots \\ xv_n \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$. If some $v_i \neq 0$, then $x = 0$ is the only solution.

Some easy cases:

- Sets containing two vectors $\{\mathbf{u}, \mathbf{v}\}$:

$$x_1\mathbf{u} + x_2\mathbf{v} = \mathbf{0}$$

if $x_1 \neq 0$, then $\mathbf{u} = (-x_2/x_1)\mathbf{v}$. if $x_2 \neq 0$, then $\mathbf{v} = (-x_1/x_2)\mathbf{u}$.

So $\{\mathbf{u}, \mathbf{v}\}$ is linearly dependent if and only if one of the vectors is a multiple of the other.

Some easy cases:

- Sets containing two vectors $\{\mathbf{u}, \mathbf{v}\}$:

  $$x_1\mathbf{u} + x_2\mathbf{v} = \mathbf{0}$$

  if $x_1 \neq 0$, then $\mathbf{u} = (-x_2/x_1)\mathbf{v}$.　　　if $x_2 \neq 0$, then $\mathbf{v} = (-x_1/x_2)\mathbf{u}$.

  So $\{\mathbf{u}, \mathbf{v}\}$ is linearly dependent if and only if one of the vectors is a multiple of the other.

- Sets containing more vectors:

  A set of vectors is linearly dependent if and only if one of the vectors is a linear combination of the others. (Any vector with nonzero weight in the linear dependency relation will work.)

A non-trivial solution to $A\mathbf{x} = \mathbf{0}$ is a linear dependence relation between the columns of $A$.

**Theorem**: For a matrix $A$, the following are equivalent:

a. $A\mathbf{x} = \mathbf{0}$ has no non-trivial solution.
b. If $A\mathbf{x} = \mathbf{b}$ is consistent, then it has a unique solution.
c. The columns of $A$ are linearly independent.
d. rref$(A)$ has a pivot in every column (i.e. all variables are basic).

A non-trivial solution to $A\mathbf{x} = \mathbf{0}$ is a linear dependence relation between the columns of $A$.

**Theorem**: For a matrix $A$, the following are equivalent:

a. $A\mathbf{x} = \mathbf{0}$ has no non-trivial solution.
b. If $A\mathbf{x} = \mathbf{b}$ is consistent, then it has a unique solution.
c. The columns of $A$ are linearly independent.
d. rref$(A)$ has a pivot in every column (i.e. all variables are basic).

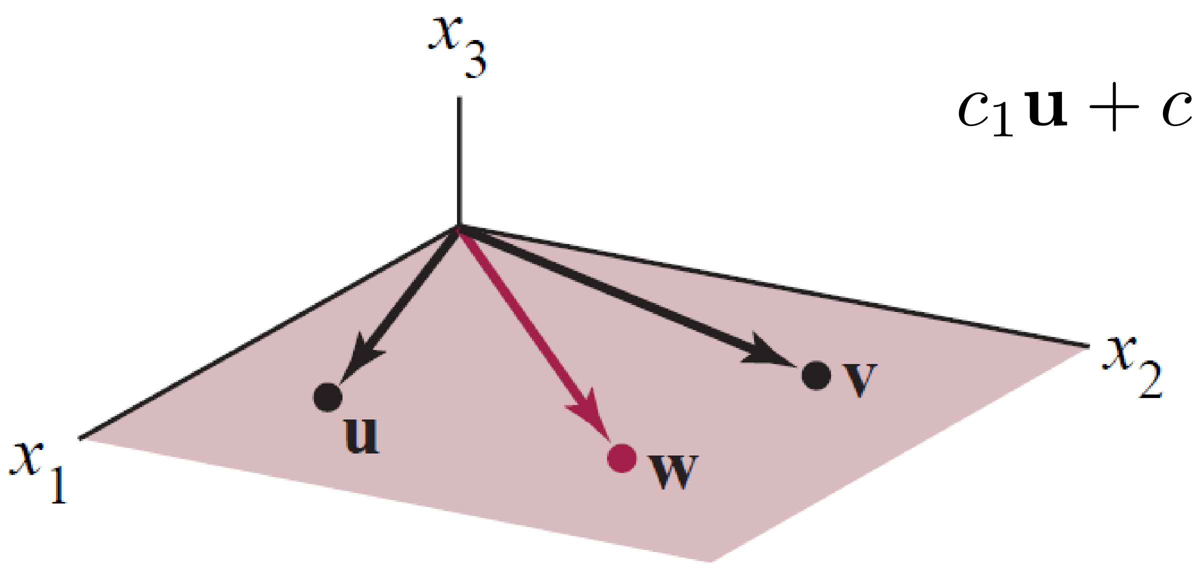
In particular: the row reduction algorithm produces at most one pivot in each row of rref$(A)$. So, if $A$ has more columns than rows (a “fat” matrix), then rref$(A)$ cannot have a pivot in every column.

So a set of more than $n$ vectors in $\mathbb{R}^n$ is always linearly dependent.

Exercise: Combine this with Theorem 4 to show that a set of $n$ linearly independent vectors span $\mathbb{R}^n$.

Problem: if $\{\mathbf{v_1}, \ldots, \mathbf{v_p}\}$ is linearly dependent, then Span $\{\mathbf{v_1}, \ldots, \mathbf{v_p}\}$ is the span of fewer vectors.

E.g. if $\mathbf{w} = a\mathbf{u} + b\mathbf{v}$, then Span $\{\mathbf{u}, \mathbf{v}, \mathbf{w}\}$ = Span $\{\mathbf{u}, \mathbf{v}\}$:

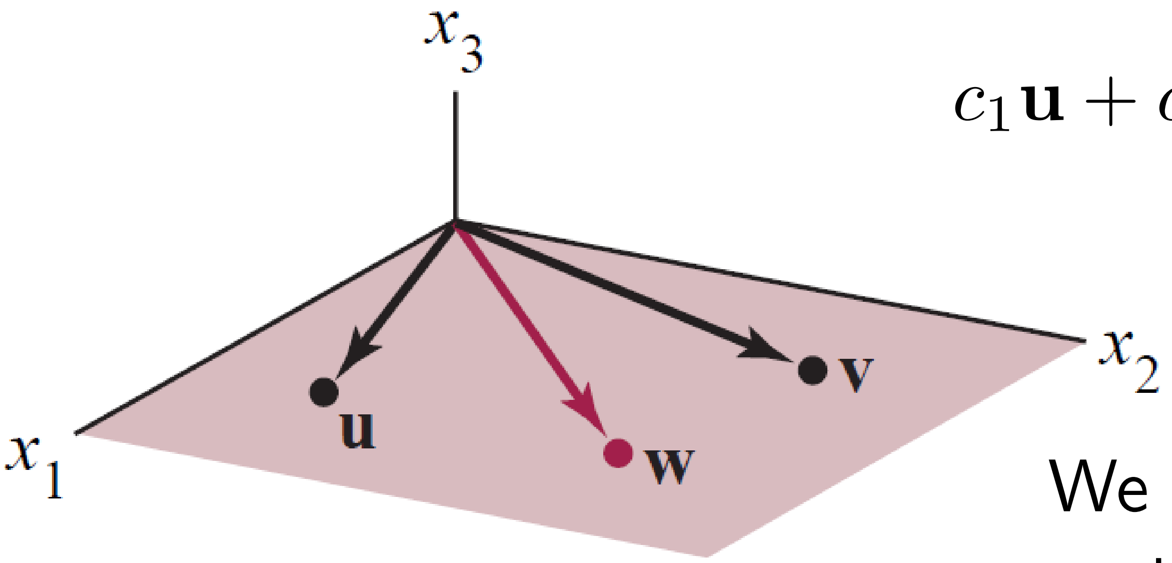


$$\begin{aligned} c_1\mathbf{u} + c_2\mathbf{v} + c_3\mathbf{w} &= c_1\mathbf{u} + c_2\mathbf{v} + c_3(a\mathbf{u} + b\mathbf{v}) \\ &= (c_1 + c_3a)\mathbf{u} + (c_2 + c_3b)\mathbf{v}. \end{aligned}$$

Problem: if $\{\mathbf{v_1}, \dots, \mathbf{v_p}\}$ is linearly dependent, then Span $\{\mathbf{v_1}, \dots, \mathbf{v_p}\}$ is the span of fewer vectors.

E.g. if $\mathbf{w} = a\mathbf{u} + b\mathbf{v}$, then Span $\{\mathbf{u}, \mathbf{v}, \mathbf{w}\}$ = Span $\{\mathbf{u}, \mathbf{v}\}$:

$$\begin{aligned} c_1\mathbf{u} + c_2\mathbf{v} + c_3\mathbf{w} &= c_1\mathbf{u} + c_2\mathbf{v} + c_3(a\mathbf{u} + b\mathbf{v}) \\ &= (c_1 + c_3a)\mathbf{u} + (c_2 + c_3b)\mathbf{v}. \end{aligned}$$



We want to remove from $\{\mathbf{v_1}, \dots, \mathbf{v_p}\}$ some vectors that are linear combinations of other $\mathbf{v_i}$s.

One answer (casting-out algorithm):

Row reduce $\begin{bmatrix} | & | & & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \dots & \mathbf{v}_p \\ | & | & & | \end{bmatrix}$ and keep the vectors in the pivot columns.

The casting-out algorithm:

**Example**: Let

$$S = \left\{ \begin{bmatrix} 0 \\ 3 \\ 1 \end{bmatrix}, \begin{bmatrix} 3 \\ -7 \\ 3 \end{bmatrix}, \begin{bmatrix} -6 \\ 8 \\ 4 \end{bmatrix}, \begin{bmatrix} 6 \\ -5 \\ -3 \end{bmatrix}, \begin{bmatrix} 4 \\ 8 \\ 2 \end{bmatrix} \right\}.$$

Find a linearly independent subset $R$ of $S$ such that Span$R =$ Span$S$.

The casting-out algorithm:

**Example**: Let

$$S = \left\{ \begin{bmatrix} 0 \\ 3 \\ 1 \end{bmatrix}, \begin{bmatrix} 3 \\ -7 \\ 3 \end{bmatrix}, \begin{bmatrix} -6 \\ 8 \\ 4 \end{bmatrix}, \begin{bmatrix} 6 \\ -5 \\ -3 \end{bmatrix}, \begin{bmatrix} 4 \\ 8 \\ 2 \end{bmatrix} \right\}.$$

Find a linearly independent subset $R$ of $S$ such that Span$R =$ Span$S$.

**Answer**:

$$\begin{bmatrix} 0 & 3 & -6 & 6 & 4 \\ 3 & -7 & 8 & -5 & 8 \\ 1 & -3 & 4 & -3 & 2 \end{bmatrix} \xrightarrow[\text{to echelon form}]{\text{row reduction}} \begin{bmatrix} 1 & -3 & 4 & 3 & 2 \\ 0 & 1 & -2 & 2 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The pivot columns are 1,2 and 5, so $R = \left\{ \begin{bmatrix} 0 \\ 3 \\ 1 \end{bmatrix}, \begin{bmatrix} 3 \\ -7 \\ 3 \end{bmatrix}, \begin{bmatrix} 4 \\ 8 \\ 2 \end{bmatrix} \right\}$ is one answer.

(The answer from the casting out algorithm is not the only answer.)

Why the casting-out algorithm works:

**Example**:

$$\begin{bmatrix} | & | & | & | & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 & \mathbf{v}_4 & \mathbf{v}_5 \\ | & | & | & | & | \end{bmatrix} = \begin{bmatrix} 0 & 3 & -6 & 6 & 4 \\ 3 & -7 & 8 & -5 & 8 \\ 1 & -3 & 4 & -3 & 2 \end{bmatrix} \xrightarrow[\text{to rref}]{\text{row reduction}} \begin{bmatrix} 1 & 0 & -2 & 3 & 0 \\ 0 & 1 & -2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Why the casting-out algorithm works:

**Example**:

$$\begin{bmatrix} | & | & | & | & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 & \mathbf{v}_4 & \mathbf{v}_5 \\ | & | & | & | & | \end{bmatrix} = \begin{bmatrix} 0 & 3 & -6 & 6 & 4 \\ 3 & -7 & 8 & -5 & 8 \\ 1 & -3 & 4 & -3 & 2 \end{bmatrix} \xrightarrow[\text{to rref}]{\text{row reduction}} \begin{bmatrix} 1 & 0 & -2 & 3 & 0 \\ 0 & 1 & -2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{rref}\left(\begin{bmatrix} | \\ \mathbf{v}_1 \\ | \end{bmatrix}\right) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$ has a pivot in every column, so $\{\mathbf{v_1}\}$ is linearly independent.

Why the casting-out algorithm works:

**Example**:

$$\begin{bmatrix} | & | & | & | & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 & \mathbf{v}_4 & \mathbf{v}_5 \\ | & | & | & | & | \end{bmatrix} = \begin{bmatrix} 0 & 3 & -6 & 6 & 4 \\ 3 & -7 & 8 & -5 & 8 \\ 1 & -3 & 4 & -3 & 2 \end{bmatrix} \xrightarrow[\text{to rref}]{\text{row reduction}} \begin{bmatrix} 1 & 0 & -2 & 3 & 0 \\ 0 & 1 & -2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$\text{rref}\left(\begin{bmatrix} | \\ \mathbf{v}_1 \\ | \end{bmatrix}\right) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ has a pivot in every column, so $\{\mathbf{v_1}\}$ is linearly independent.

$\text{rref}\left(\begin{bmatrix} | & | \\ \mathbf{v}_1 & \mathbf{v}_2 \\ | & | \end{bmatrix}\right) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$ has a pivot in every column, so $\{\mathbf{v_1}, \mathbf{v_2}\}$ is linearly independent.

## Why the casting-out algorithm works:

**Example**:

$$\begin{bmatrix} | & | & | & | & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 & \mathbf{v}_4 & \mathbf{v}_5 \\ | & | & | & | & | \end{bmatrix} = \begin{bmatrix} 0 & 3 & -6 & 6 & 4 \\ 3 & -7 & 8 & -5 & 8 \\ 1 & -3 & 4 & -3 & 2 \end{bmatrix} \xrightarrow[\text{to rref}]{\text{row reduction}} \begin{bmatrix} 1 & 0 & -2 & 3 & 0 \\ 0 & 1 & -2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{rref}\left(\begin{bmatrix} | & | & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \\ | & | & | \end{bmatrix}\right) = \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & -2 \\ 0 & 0 & 0 \end{bmatrix}$$ does not have a pivot in every column.

The solution set to $\begin{bmatrix} | & | & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \\ | & | & | \end{bmatrix} \mathbf{x} = \mathbf{0}$ is

$$\begin{aligned} x_1 \quad -2x_3 &= 0 \\ x_2 - 2x_3 &= 0 \end{aligned} \qquad \begin{aligned} x_1 &= 2x_3 \\ x_2 &= 2x_3 \\ x_3 &= \ x_3 \end{aligned}$$

Why the casting-out algorithm works:

**Example**:

$$\begin{bmatrix} | & | & | & | & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 & \mathbf{v}_4 & \mathbf{v}_5 \\ | & | & | & | & | \end{bmatrix} = \begin{bmatrix} 0 & 3 & -6 & 6 & 4 \\ 3 & -7 & 8 & -5 & 8 \\ 1 & -3 & 4 & -3 & 2 \end{bmatrix} \xrightarrow[\text{to rref}]{\text{row reduction}} \begin{bmatrix} 1 & 0 & -2 & 3 & 0 \\ 0 & 1 & -2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{rref}\left(\begin{bmatrix} | & | & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \\ | & | & | \end{bmatrix}\right) = \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & -2 \\ 0 & 0 & 0 \end{bmatrix}$$ does not have a pivot in every column.

The solution set to $\begin{bmatrix} | & | & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \\ | & | & | \end{bmatrix} \mathbf{x} = \mathbf{0}$ is $\mathbf{x} = s \begin{bmatrix} 2 \\ 2 \\ 1 \end{bmatrix}$ where $s$ can take any value.

$$\begin{aligned} x_1 \quad -2x_3 &= 0 \\ x_2 - 2x_3 &= 0 \end{aligned} \qquad \begin{aligned} x_1 &= 2x_3 \\ x_2 &= 2x_3 \\ x_3 &= \ \ x_3 \end{aligned}$$

Why the casting-out algorithm works:

**Example**:

$$\begin{bmatrix} | & | & | & | & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 & \mathbf{v}_4 & \mathbf{v}_5 \\ | & | & | & | & | \end{bmatrix} = \begin{bmatrix} 0 & 3 & -6 & 6 & 4 \\ 3 & -7 & 8 & -5 & 8 \\ 1 & -3 & 4 & -3 & 2 \end{bmatrix} \xrightarrow[\text{to rref}]{\text{row reduction}} \begin{bmatrix} 1 & 0 & -2 & 3 & 0 \\ 0 & 1 & -2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{rref}\left(\begin{bmatrix} | & | & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \\ | & | & | \end{bmatrix}\right) = \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & -2 \\ 0 & 0 & 0 \end{bmatrix}$$ does not have a pivot in every column.

The solution set to $\begin{bmatrix} | & | & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \\ | & | & | \end{bmatrix} \mathbf{x} = \mathbf{0}$ is $\mathbf{x} = s \begin{bmatrix} 2 \\ 2 \\ 1 \end{bmatrix}$ where $s$ can take any value.

Take $s = 1$: $\begin{bmatrix} | & | & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \\ | & | & | \end{bmatrix} \begin{bmatrix} 2 \\ 2 \\ 1 \end{bmatrix} = \mathbf{0}$.

So $2\mathbf{v_1} + 2\mathbf{v_2} + \mathbf{v_3} = \mathbf{0}$, so $\mathbf{v_3} = -2\mathbf{v_1} - 2\mathbf{v_2}$, a linear combination of $\mathbf{v_1}$ and $\mathbf{v_2}$. So we don't need $\mathbf{v_3}$ to get the same span.

Why the casting-out algorithm works:

**Example**:

$$\begin{bmatrix} | & | & | & | & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 & \mathbf{v}_4 & \mathbf{v}_5 \\ | & | & | & | & | \end{bmatrix} = \begin{bmatrix} 0 & 3 & -6 & 6 & 4 \\ 3 & -7 & 8 & -5 & 8 \\ 1 & -3 & 4 & -3 & 2 \end{bmatrix} \xrightarrow[\text{to rref}]{\text{row reduction}} \begin{bmatrix} 1 & 0 & -2 & 3 & 0 \\ 0 & 1 & -2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The solution set to $\begin{bmatrix} | & | & | & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 & \mathbf{v}_4 \\ | & | & | & | \end{bmatrix} \mathbf{x} = \mathbf{0}$ is $\mathbf{x} = s\begin{bmatrix} 2 \\ 2 \\ 1 \\ 0 \end{bmatrix} + t\begin{bmatrix} -3 \\ -2 \\ 0 \\ 1 \end{bmatrix}$ where $s, t$ can take any value.

Take $s = 0, t = 1$: $\begin{bmatrix} | & | & | & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 & \mathbf{v}_4 \\ | & | & | & | \end{bmatrix} \begin{bmatrix} -3 \\ -2 \\ 0 \\ 1 \end{bmatrix} = \mathbf{0}.$ So $-3\mathbf{v_1} - 2\mathbf{v_2} + 0\mathbf{v_3} + \mathbf{v_4} = \mathbf{0}$, so $\mathbf{v_4} = 3\mathbf{v_1} + 2\mathbf{v_2}$, a linear combination of the pivot columns.

Why the casting-out algorithm works:

The row reduction algorithm writes the solution set of

$$\begin{bmatrix} | & | & & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \dots & \mathbf{v}_p \\ | & | & & | \end{bmatrix} \mathbf{x} = \mathbf{0}$$

in the form $s_i\mathbf{w_i} + s_j\mathbf{w_j} + \dots$, where $x_i, x_j, \dots$ are the free variables.

For each column $\mathbf{v_i}$ corresponding to a free variable, the solution $A\mathbf{w_i} = \mathbf{0}$ allows you to write $\mathbf{v_i}$ as a linear combination of the earlier pivot columns.

So Span $\{\mathbf{v_1}, \mathbf{v_2}, \dots, \mathbf{v_p}\}$ is the same as the span of the pivot columns.

Why the casting-out algorithm works:

The row reduction algorithm writes the solution set of

$$\begin{bmatrix} | & | & & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \dots & \mathbf{v}_p \\ | & | & & | \end{bmatrix} \mathbf{x} = \mathbf{0}$$

in the form $s_i \mathbf{w_i} + s_j \mathbf{w_j} + \dots$, where $x_i, x_j, \dots$ are the free variables.

For each column $\mathbf{v_i}$ corresponding to a free variable, the solution $A\mathbf{w_i} = \mathbf{0}$ allows you to write $\mathbf{v_i}$ as a linear combination of the earlier pivot columns.

So Span $\{\mathbf{v_1}, \mathbf{v_2}, \dots, \mathbf{v_p}\}$ is the same as the span of the pivot columns.

The casting-out algorithm is a "greedy algorithm": it prefers vectors that are earlier in the set.

E.g. if you want a linearly independent subset of $\{\mathbf{u}, \mathbf{v}, \mathbf{w}\}$ with the same span, and you want $\mathbf{w}$ to be in this set, you should row-reduce $\begin{bmatrix} \mathbf{w} & \mathbf{u} & \mathbf{v} \end{bmatrix}$.