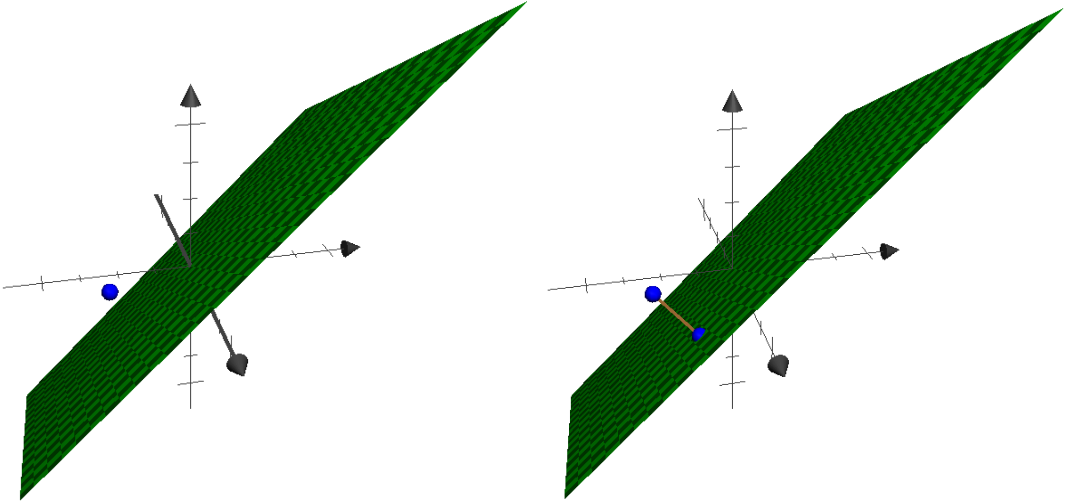
# Finding the closest point in a plane

**Goal:** Given a point $\mathbf{b}$ and a plane, find the point in the plane closest to $\mathbf{b}$.



# Finding the closest point in a plane

**Goal:** Given a point $\mathbf{b}$ and a plane, find the point in the plane closest to $\mathbf{b}$.

By translation, we can assume the plane includes the origin.

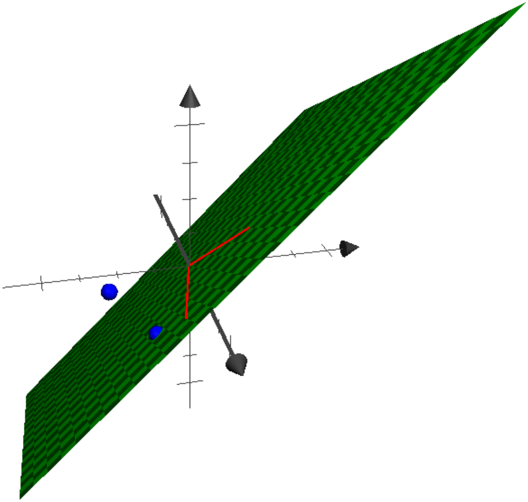
The plane is a vector space $\mathcal{V}$. Let $\{\mathbf{v}_1, \mathbf{v}_2\}$ be a basis for $\mathcal{V}$.

**Goal:** Given a point $\mathbf{b}$, find the point in Span $\{\mathbf{v}_1, \mathbf{v}_2\}$ closest to $\mathbf{b}$.

**Example:**

$$\mathbf{v}_1 = [8, -2, 2] \text{ and } \mathbf{v}_2 = [4, 2, 4]$$

$\mathbf{b} = [5, -5, 2]$
point in plane closest to $\mathbf{b}$: $[6, -3, 0]$.



# Closest-point problem in in higher dimensions

**Goal:** An algorithm that, given a vector $\mathbf{b}$ and vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$, finds the vector in Span $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ that is closest to $\mathbf{b}$.

**Special case:** We can use the algorithm to determine whether $\mathbf{b}$ lies in Span $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$:
If the vector in Span $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ closest to $\mathbf{b}$ is $\mathbf{b}$ itself then clearly $\mathbf{b}$ is in the span; if not, then $\mathbf{b}$ is not in the span.

Let $A = \left[\begin{array}{c|c|c} \mathbf{v}_1 & \cdots & \mathbf{v}_n \end{array}\right]$.

Using the linear-combinations interpretation of matrix-vector multiplication, a vector in Span $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ can be written $A\mathbf{x}$.

Thus testing if $\mathbf{b}$ is in Span $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ is equivalent to testing if the equation $A\mathbf{x} = \mathbf{b}$ has a solution.

**More generally:**
Even if $A\mathbf{x} = \mathbf{b}$ has no solution, we can use the algorithm to find the point in $\{A\mathbf{x} \; : \; \mathbf{x} \in \mathbb{R}^n\}$ closest to $\mathbf{b}$.

**Moreover:** We hope to extend the algorithm to also find the best solution $\mathbf{x}$.

## Closest point and coefficients

Not enough to find the point $p$ in Span $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ closest to $\mathbf{b}$....
We need an algorithm to find the representation of $p$ in terms of $\mathbf{v}_1, \ldots, \mathbf{v}_n$.

**Goal:** find the coefficients $x_1, \ldots, x_n$ so that $x_1\,\mathbf{v}_1 + \cdots + x_n\,\mathbf{v}_n$ is the vector in Span $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ closest to $\mathbf{b}$.

**Equivalent:** Find the vector $\mathbf{x}$ that minimizes $\left\| \begin{bmatrix} \mathbf{b} \end{bmatrix} - \left[\begin{array}{c|c|c} \mathbf{v}_1 & \cdots & \mathbf{v}_n \end{array}\right] \begin{bmatrix} \mathbf{x} \end{bmatrix} \right\|$

**Equivalent:** Find the vector $\mathbf{x}$ that minimizes $\left\| \begin{bmatrix} \mathbf{b} \end{bmatrix} - \left[\begin{array}{c|c|c} \mathbf{v}_1 & \cdots & \mathbf{v}_n \end{array}\right] \begin{bmatrix} \mathbf{x} \end{bmatrix} \right\|^2$

**Equivalent:** Find the vector $\mathbf{x}$ that minimizes $\left\| \begin{bmatrix} \mathbf{b} \end{bmatrix} - \left[\begin{array}{c} \mathbf{a}_1 \\ \hline \vdots \\ \hline \mathbf{a}_m \end{array}\right] \begin{bmatrix} \mathbf{x} \end{bmatrix} \right\|^2$

**Equivalent:** Find the vector $\mathbf{x}$ that minimizes $(\mathbf{b}[1] - \mathbf{a}_1 \cdot \mathbf{x})^2 + \cdots + (\mathbf{b}[m] - \mathbf{a}_m \cdot \mathbf{x})^2$

This last problem was addressed using gradient descent in Machine Learning lab.

# Closest point and least squares

Find the vector $\mathbf{x}$ that minimizes $\left\| \left[ \begin{array}{c} \\ \mathbf{b} \\ \\ \end{array} \right] - \left[ \begin{array}{c|c|c} & & \\ \mathbf{v}_1 & \cdots & \mathbf{v}_n \\ & & \end{array} \right] \left[ \begin{array}{c} \\ \mathbf{x} \\ \\ \end{array} \right] \right\|^2$

**Equivalent:** Find the vector $\mathbf{x}$ that minimizes $(\mathbf{b}[1] - \mathbf{a}_1 \cdot \mathbf{x})^2 + \cdots + (\mathbf{b}[m] - \mathbf{a}_m \cdot \mathbf{x})^2$

This problem is called *least squares* ("méthode des moindres carrés", due to Adrien-Marie Legendre but often attributed to Gauss)

**Equivalent:** Given a matrix equation $A\mathbf{x} = \mathbf{b}$ that might have no solution, find the best solution available in the sense that the norm of the error $\mathbf{b} - A\mathbf{x}$ is as small as possible.

- There is an algorithm based on Gaussian elimination.
- We will develop an algorithm based on orthogonality (used in `solver`)

  Much faster and more reliable than gradient descent.