Linear algebra

Mathematical Foundations

- Geometry (2D, 3D)
- Trigonometry
- Vector and affine spaces
  - Points, vectors, and coordinates
- Dot and cross products
- Linear transforms and matrices

3D Geometry

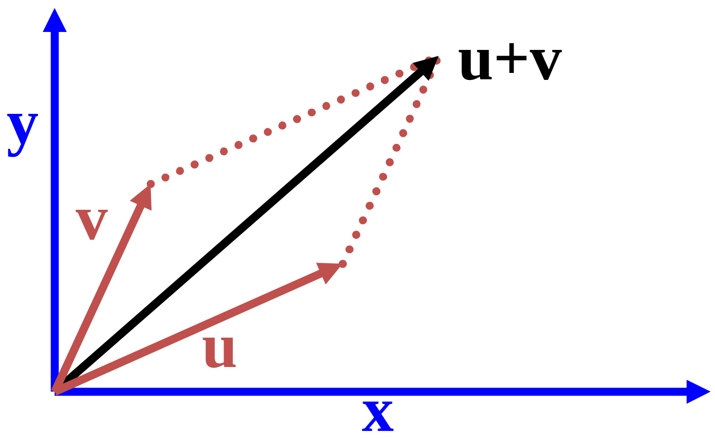
- To model, animate, and render 3D scenes, we must specify:
  - Location
  - Displacement from arbitrary locations
  - Orientation
- We'll look at two types of spaces:
  - *Vector spaces*
  - *Affine spaces*
- We will often be sloppy about the distinction

Vector Spaces

- Formed by a collection of “vectors”
- Two types of elements:
  - Scalars (real numbers): $\alpha, \beta, \gamma, \delta, \ldots$
  - Vectors (n-tuples): $\mathbf{u}, \mathbf{v}, \mathbf{w}, \ldots$

Vector Spaces: A Familiar Example

- Vectors in a 2D plane is:
  - Vectors are "arrows" rooted at the origin
  - Scalar multiplication "streches" the arrow, changing its length (   ) but not its direction
  - Addition uses the "trapezoid rule":



Vector Spaces: Another Familiar Example

- Ordered tripples:
  - $(x1, y1, z1) + (x2, y2, z2) = (x1 + x2, y1 + y2, z1 + z2)$
    and
  - $\alpha \cdot (x, y, z) = (\alpha x, \alpha y, \alpha z)$

Vector Spaces

- Supports two operations
  (which do not exit a given vector space):
  - *Associative* and *commutative*
    addition operation $\mathbf{u} + \mathbf{v}$, with:
    - Identity $\mathbf{o}$ $\quad \mathbf{v} + \mathbf{o} = \mathbf{v}$ $\quad$ ("zero" vector)
    - Inverse **-** $\quad \mathbf{v} + (-\mathbf{v}) = \mathbf{o}$
  - Scalar multiplication:
    - Distributive rule: $\quad \alpha(\mathbf{u} + \mathbf{v}) = \alpha\mathbf{u} + \alpha\mathbf{v}$

      $(\alpha + \beta)\mathbf{u} = \alpha\mathbf{u} + \beta\mathbf{u}$

Vector Spaces

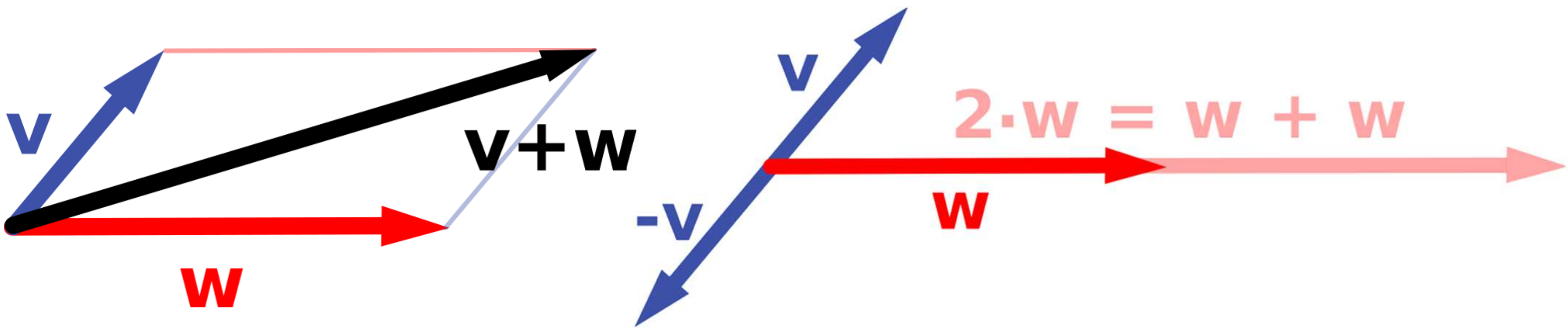
- A *linear combination* of vectors results in a new vector:

$$\mathbf{v} = \alpha_1\mathbf{v}_1 + \alpha_2\mathbf{v}_2 + \ldots + \alpha_n\mathbf{v}_n$$

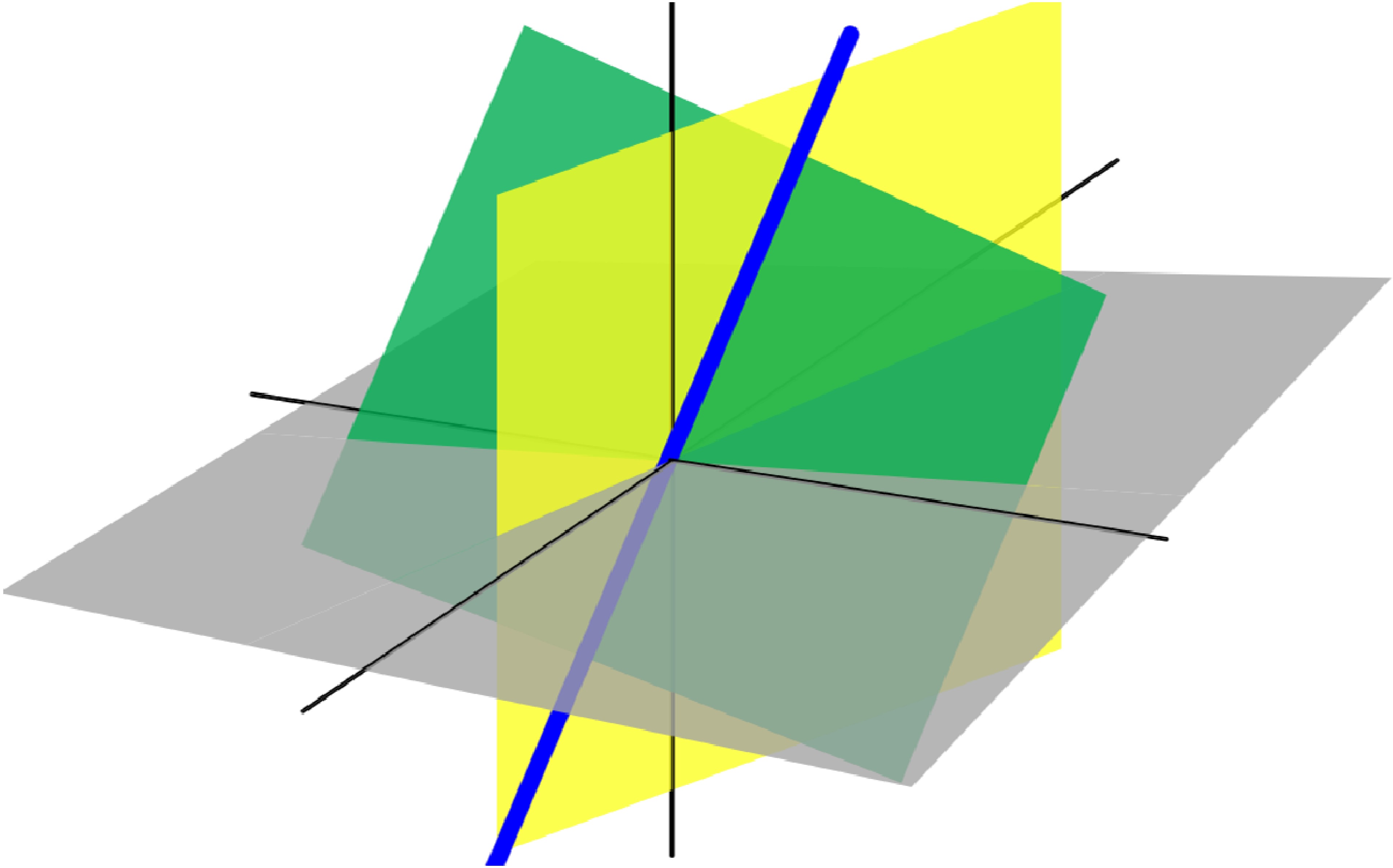
Example

Linear Hull = Subspace

- Given a set of vectors the **linear hull** is the set of all the vectors that can be created out of linear combinations of those vectors.
- This is called a subspace



Subspaces



Basis

- If the only set of scalars such that

$$\alpha_1 \mathbf{v}_1 + \alpha_2 \mathbf{v}_2 + \ldots + \alpha_n \mathbf{v}_n = \mathbf{0}$$

is $$\alpha_1 = \alpha_2 = \ldots = \alpha_3 = \mathbf{0}$$

then we say the vectors are *linearly independent*

- The *dimension* of a space is the greatest number of linearly independent vectors possible in a vector set
- For a vector space of dimension *n*, any set of *n* linearly independent vectors form a

Example

Coordinates

- Given a basis for a vector space:
  - Each vector in the space is a *unique* linear combination of the basis vectors
  - The *coordinates* of a vector are the scalars from this linear combination
  - Best-known example: Cartesian coordinates
  - Note that a given vector **v** will have different coordinates for different bases

Example

Dot Product

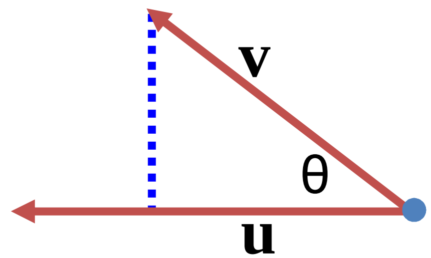
- The *dot product* or, more generally, *inner product* of two vectors is a scalar:

$$\mathbf{v}_1 \bullet \mathbf{v}_2 = < \mathbf{v}_1 , \mathbf{v}_2 > = \mathbf{x}_1\mathbf{x}_2 + \mathbf{y}_1\mathbf{y}_2 + \mathbf{z}_1\mathbf{z}_2 \quad \text{(in 3D)}$$

- Useful for many purposes
  - Computing the length of a vector: $|\mathbf{v}| = \text{sqrt}(\mathbf{v} \bullet \mathbf{v})$
  - *Normalizing* a vector, making it unit-length
  - Computing the angle between two vectors:

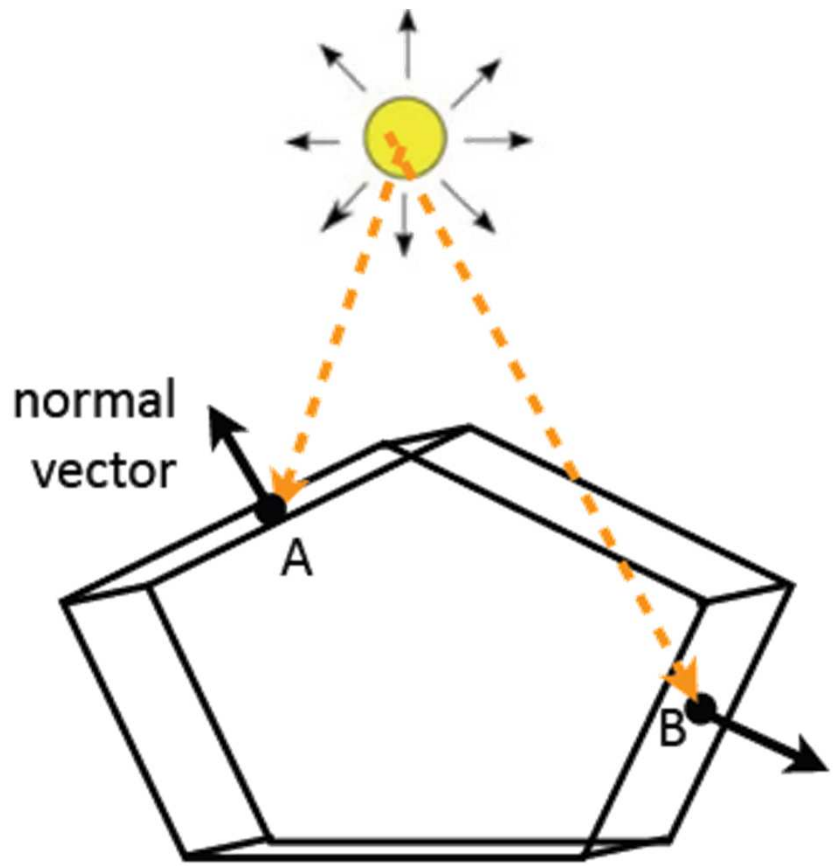
    $\mathbf{u} \bullet \mathbf{v} = |\mathbf{u}|\ |\mathbf{v}| \cos(\theta)$
  - Checking two vectors for *orthogonality*
  - *Projecting* one vector onto another

    $|\mathbf{v}| \cos(\theta) = \mathbf{u} \bullet \mathbf{v}\ |\mathbf{u}|^{-1}$



Dot Product App – Back-face culling

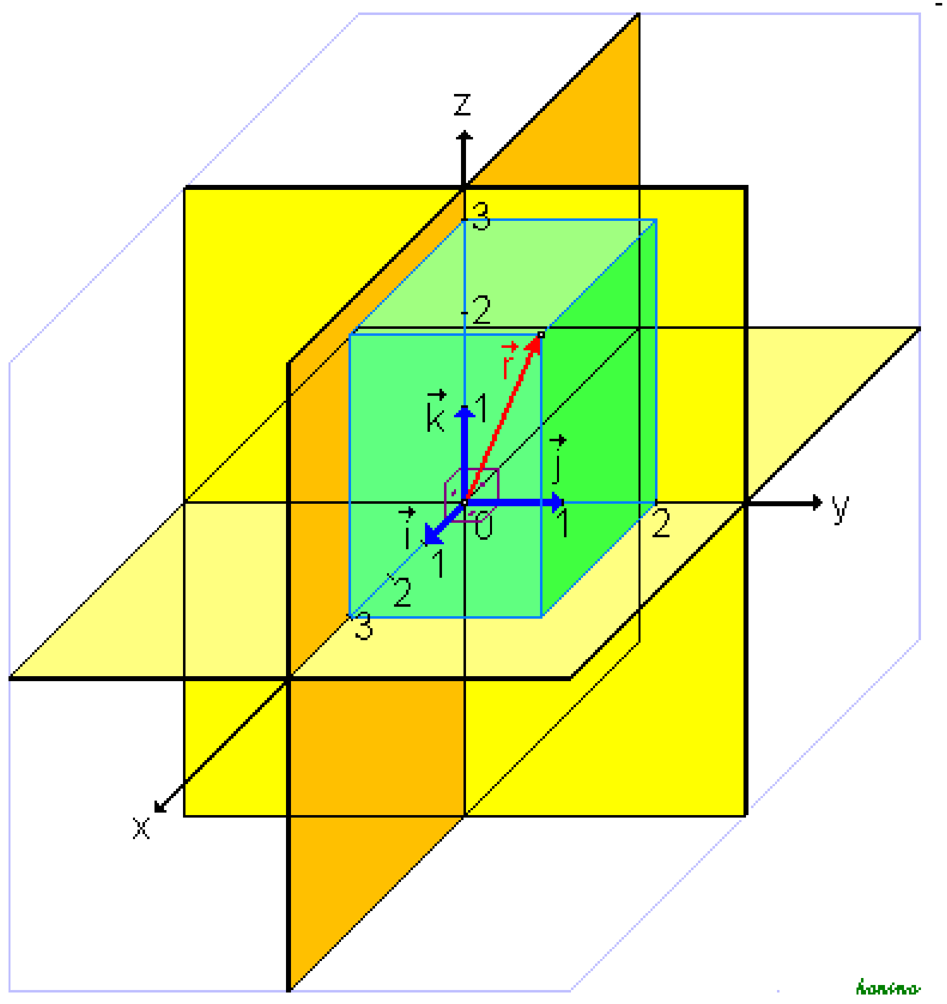
- Determining “front” vs. “back” facing triangles



Orthonormal Basis

Example

- A basis is called orthonormal iff
- Each basis vector $\mathbf{b}_i$ has the length one
  $||\mathbf{b}_i|| = <\mathbf{b}_i, \mathbf{b}_i> = 1$
- Basis vectors are pair wise orthogonal
  $<\mathbf{b}_i, \mathbf{b}_k> = 0$ for all $i \neq k$



Orthonormal Basis

- Coordinates for this kind of basis are simple to calculate $\mathbf{B} = (\mathbf{b}_1 \ldots \mathbf{b}_n)$

  $(\mathbf{v}_1)_B = <\mathbf{v}, \mathbf{b}_1>$

  ..

  $(\mathbf{v}_n)_B = <\mathbf{v}, \mathbf{b}_n>$

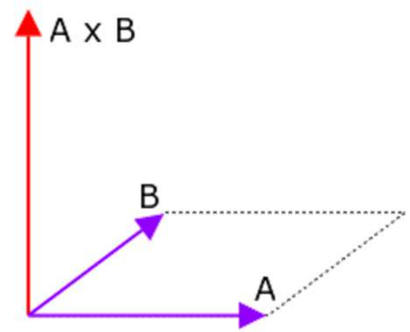
- Another advantage we will see later (matrix)

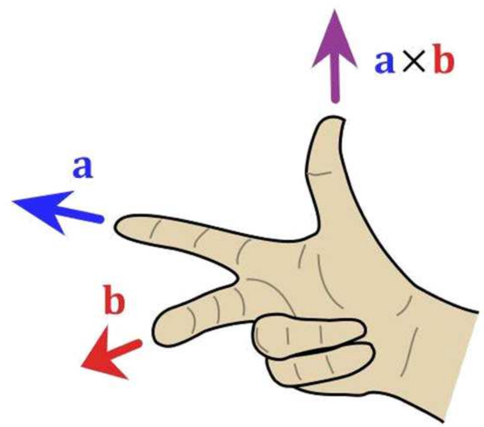
Example

Cross Product

- The *cross product* or *vector product* of two vectors is a vector:

$$\mathbf{v}_1 \times \mathbf{v}_2 = \begin{bmatrix} y_1 z_2 - y_2 z_1 \\ -(x_1 z_2 - x_2 z_1) \\ x_1 y_2 - x_2 y_1 \end{bmatrix}$$





- Is orthogonal to both
- Direction dictated by right-hand rule
- Handy for finding surface orientation
  - Lighting
  - Visibility

Matrices

- By convention, matrix element $\mathbf{M}_{rc}$ is located at row *r* and column *c*:

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} & \cdots & \mathbf{M}_{1n} \\ \mathbf{M}_{21} & \mathbf{M}_{22} & \cdots & \mathbf{M}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{M}_{m1} & \mathbf{M}_{m2} & \cdots & \mathbf{M}_{mn} \end{bmatrix}$$

- By (OpenGL) convention, vectors are columns:

$$\mathbf{v} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \end{bmatrix}$$

Linear Transformations

- A *linear transformation*:
  - Maps one vector to another
  - Preserves linear combinations
- Thus behavior of linear transformation is completely determined by what it does to a basis
- Linear combination of basis vectors

Example

Matrices and Linear Transformations

- Matrix-vector multiplication applies a linear transformation to a vector:

$$\mathbf{M} \bullet \mathbf{v} = \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} & \mathbf{M}_{13} \\ \mathbf{M}_{21} & \mathbf{M}_{22} & \mathbf{M}_{23} \\ \mathbf{M}_{31} & \mathbf{M}_{32} & \mathbf{M}_{33} \end{bmatrix} \begin{bmatrix} \mathbf{v}_x \\ \mathbf{v}_y \\ \mathbf{v}_z \end{bmatrix}$$

- Recall how to do matrix multiplication
- Matrix multiplication as a linear combination of basis vectors

Matrices and Bases

- Matrix multiplication as a linear combination of basis vectors
- $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3]$

$$\begin{bmatrix} & & \\ \mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3 \\ & & \end{bmatrix} \begin{bmatrix} \mathbf{v}_x \\ \mathbf{v}_y \\ \mathbf{v}_z \end{bmatrix} = \mathbf{B} \begin{bmatrix} \mathbf{v}_x \\ \mathbf{v}_y \\ \mathbf{v}_z \end{bmatrix}$$

Vectors and Matrices

- Vector algebra operations can be expressed in this matrix form
  - Dot product:

$$\mathbf{a}\bullet\mathbf{b}=\begin{bmatrix}a_x & a_y & a_z\end{bmatrix}\begin{bmatrix}b_x\\ b_y\\ b_z\end{bmatrix}=\alpha$$

  - Cross product:
    - Note: use right-hand rule!

$$\mathbf{a}\times\mathbf{b}=\begin{bmatrix}0 & -a_z & a_y\\ a_z & 0 & -a_x\\ -a_y & a_x & 0\end{bmatrix}\begin{bmatrix}b_x\\ b_y\\ b_z\end{bmatrix}=\begin{bmatrix}c_x\\ c_y\\ c_z\end{bmatrix}=\mathbf{c}$$

$$\mathbf{a}\bullet\mathbf{c}=\mathbf{0}$$

$$\mathbf{b}\bullet\mathbf{c}=\mathbf{0}$$

Matrix Transformations

- A *sequence* or *composition* of linear transformations corresponds to the product of the corresponding matrices
  - Note: the matrices to the *right* affect vector first
  - Note: order of matrices matters!
- The *identity matrix* **I** has no effect in multiplications
- Some (not all) matrices have an inverse

  $\mathbf{M}^{-1}\,\mathbf{M} = \mathbf{I}$ $\qquad\qquad$ $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$
- Inverse of orthonormalmatrix

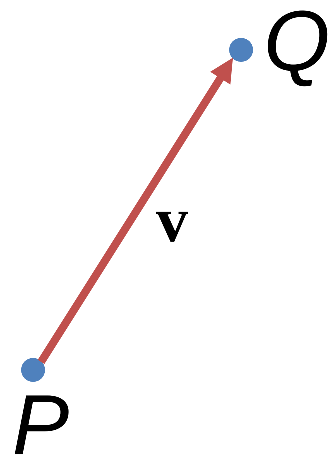
  $\mathbf{M}^{T}\,\mathbf{M} = \mathbf{I}$

Vectors And Points

- We commonly use vectors to represent:
  - Direction (i.e., orientation)
  - Points in space (i.e., location)
  - Displacements from point to point
- But we want points and directions to behave differently
  - Ex: To *translate* something means to move it without changing its orientation
  - Translation of a point = different point
  - Translation of a direction = same direction

Affine Spaces

- To be more rigorous, we need an explicit notion of *position*
- *Affine spaces* add a third element to vector spaces: *points* (P, Q, R, …)
- Points support these operations
  - Point-point subtraction: $Q - P = \mathbf{v}$
    - Result is a vector pointing from P to Q
  - Vector-point addition: $P + \mathbf{v} = Q$
    - Result is a new point
    - $P + \mathbf{0} = P$
  - Note that the addition of two points is not defined



Affine Spaces

- Points, like vectors, can be expressed in coordinates
  - The definition uses an affine combination
  - Net effect is same: expressing a point in terms of a basis
- Thus the common practice of representing points as vectors with coordinates
- Be careful to avoid nonsensical operations
  - Point + point
  - Scalar * point

Example (center point)

Affine Lines: An Aside

- Parametric representation of a *line* with a direction vector **d** and a point P_1 on the line:

  $$P(\alpha) = P_{origin} + \alpha\mathbf{d}$$

- Restricting $0 \leq \alpha$ produces a *ray*
- Setting **d** to $P - Q$ and restricting $0 \geq \alpha \geq 1$ produces a *line segment* between P and Q