Academic Chatter
@AcademicChatter
Follow

Coding tips

Google frequently
Ask for help
You are not your code
Frustration isn't failure
Try a variety of tutorials/books/resources
Comment generously
Debugging is learning
Feeling clueless is ok
Coding is thinking, give it time
Celebrate your own wins

Have fun!

1:11 PM - 13 Nov 2019

238 Retweets 879 Likes

22 238 879

Academic Chatter
@AcademicChatter

Join the #AcademicChatter community & connect with grad students, ECRs, & senior academics. Tag us with # or @AcademicChatter. Book reviews: @ACReviewofBooks

instagram.com/academicchatte…

Joined July 2017

CH40208: TOPICS IN COMPUTATIONAL CHEMISTRY

WORKING WITH VECTORS AND MATRICES (IN PYTHON)

FIRST: HOUSEKEEPING

- Data Analysis Exercise worked example

FIRST: HOUSEKEEPING

- Data Analysis Exercise worked example
- Drew's "office hours" 2:00–5:00*.

* today 2:00–3:45

THIS WEEK: VECTORS AND MATRICES (INTRODUCTION TO LINEAR ALGEBRA)

- Assuming no previous knowledge!
- First, the maths:
  - What are vectors and matrices, and how do they work mathematically?
- Then, the Python:
  - Using `numpy` for vector and matrix algebra
  - Examples & exercises:
    - vectors: atomic positions and interatomic distances
    - matrices: molecular rotations using rotation matrices

THIS WEEK: VECTORS AND MATRICES (INTRODUCTION TO LINEAR ALGEBRA)

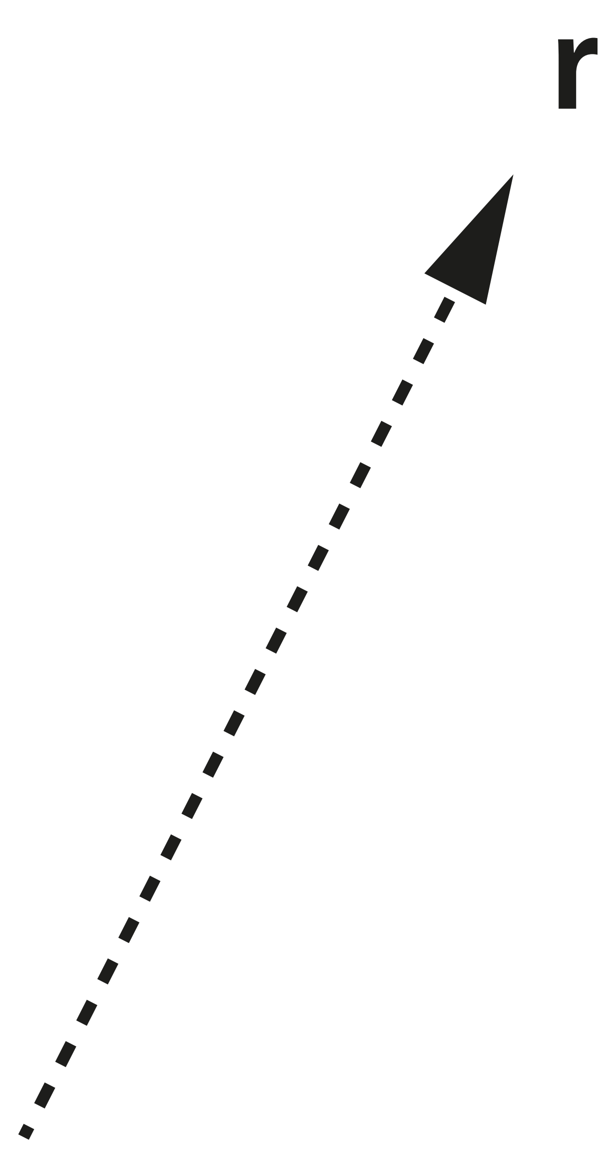
- This is not a maths course.

THIS WEEK: VECTORS AND MATRICES (INTRODUCTION TO LINEAR ALGEBRA)

- This is not a maths course.
- But …
  - 3blue1brown YouTube videos!
  - Lorena Barba "Land on Vector Spaces"

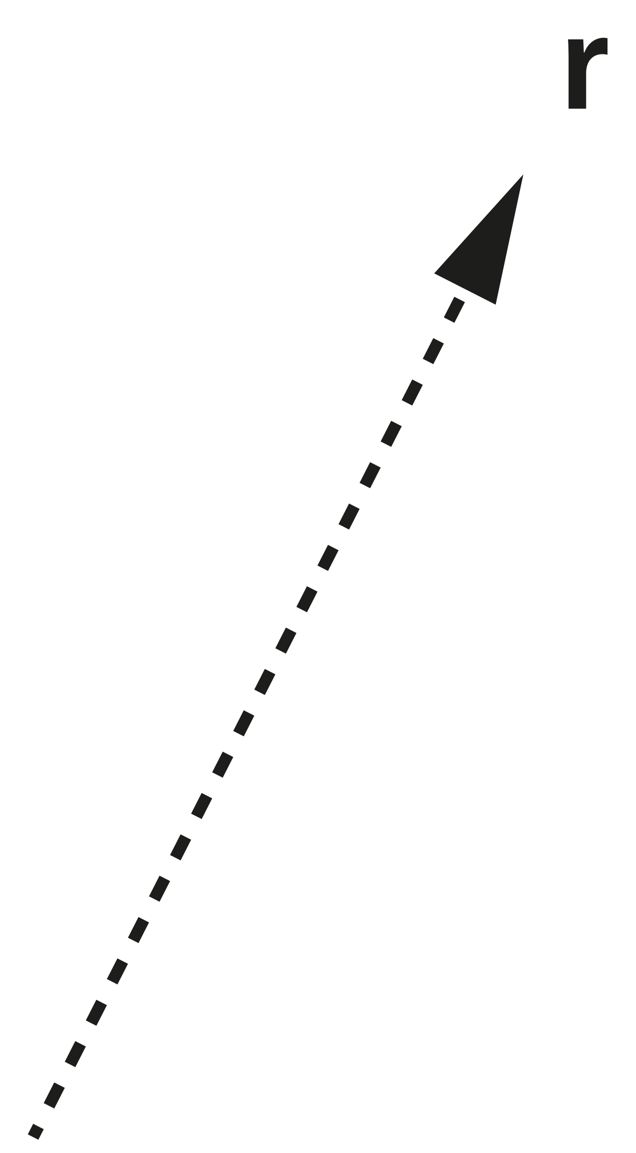
VECTORS

- Have **magnitude** and **direction**
(differs from **scalars**, which only have **magnitude**).



VECTORS

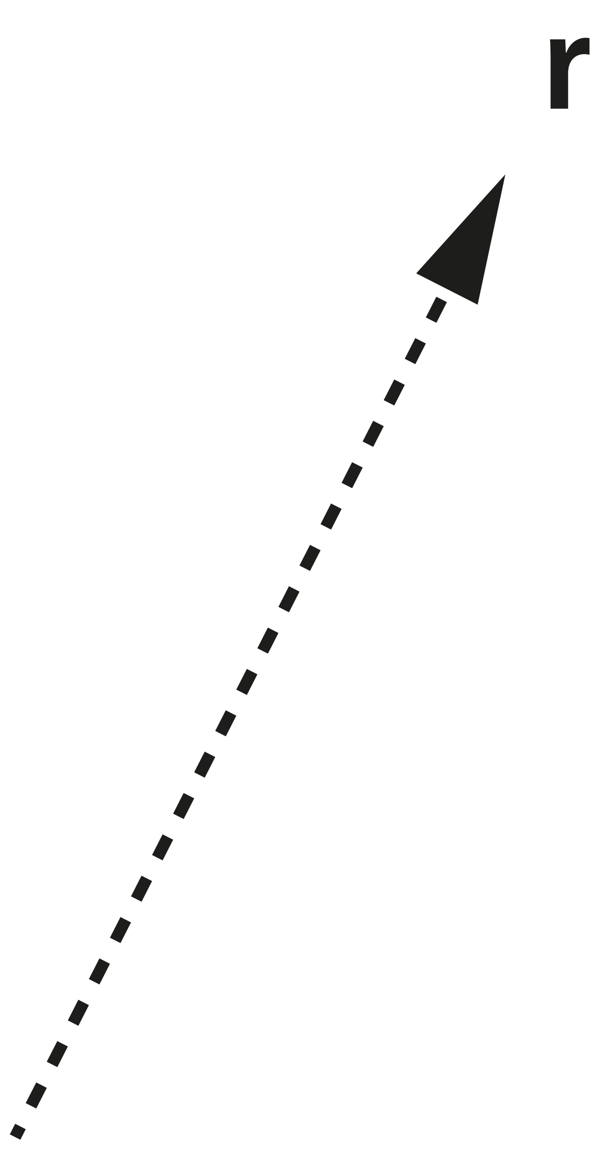
- Have **magnitude** and **direction**
  (differs from **scalars**, which only have **magnitude**).
- Examples:
  - atomic positions, velocities, accelerations, forces.



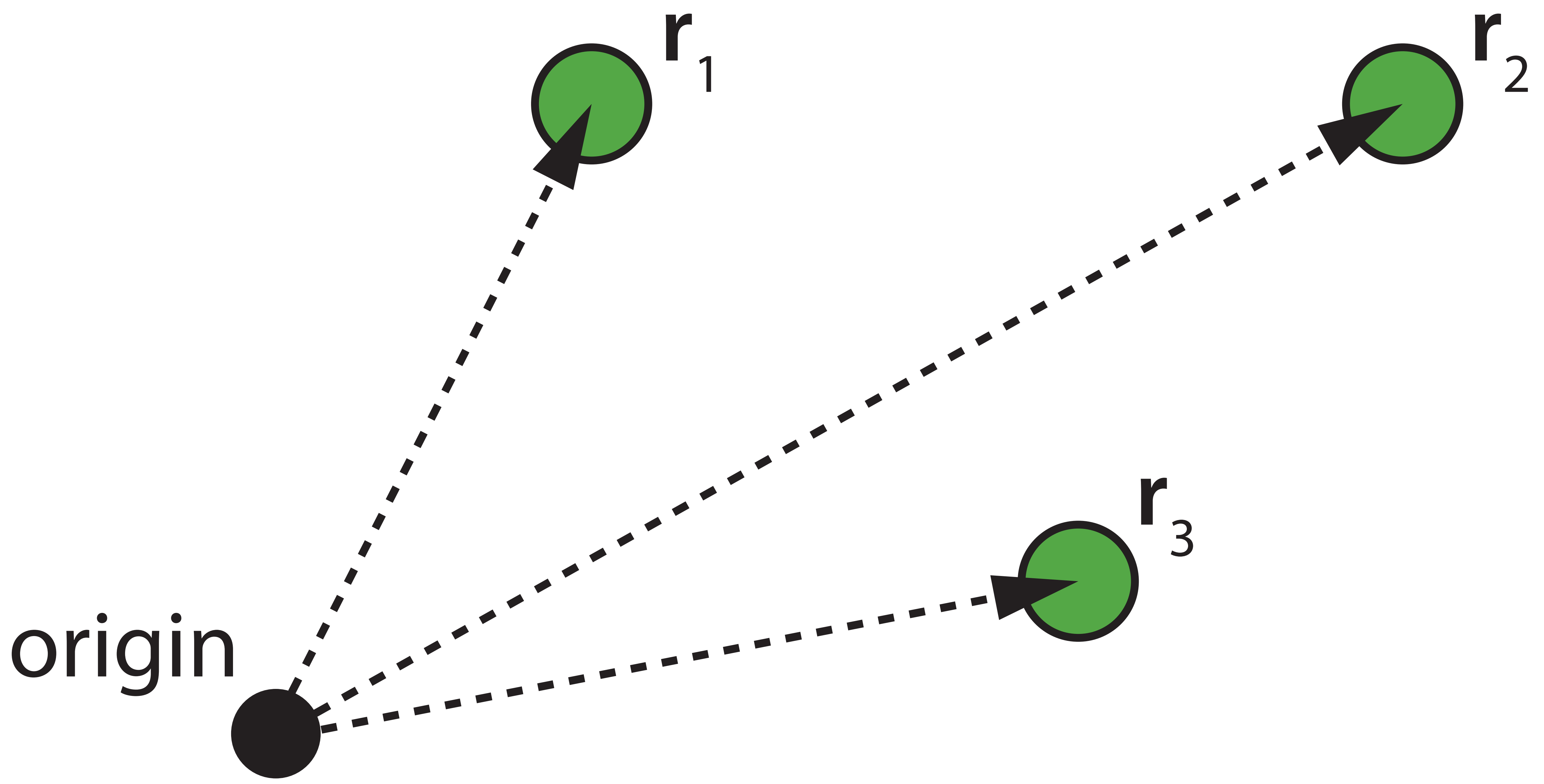
VECTORS

- Have **magnitude** and **direction**
(differs from **scalars**, which only have **magnitude**).
- Examples:
    - atomic positions, velocities, accelerations, forces.
    - angular momentum
    (molecular rotations & spin → magnetism)
    - wavefunctions

$$\Psi = \sum_i c_i \phi_i$$

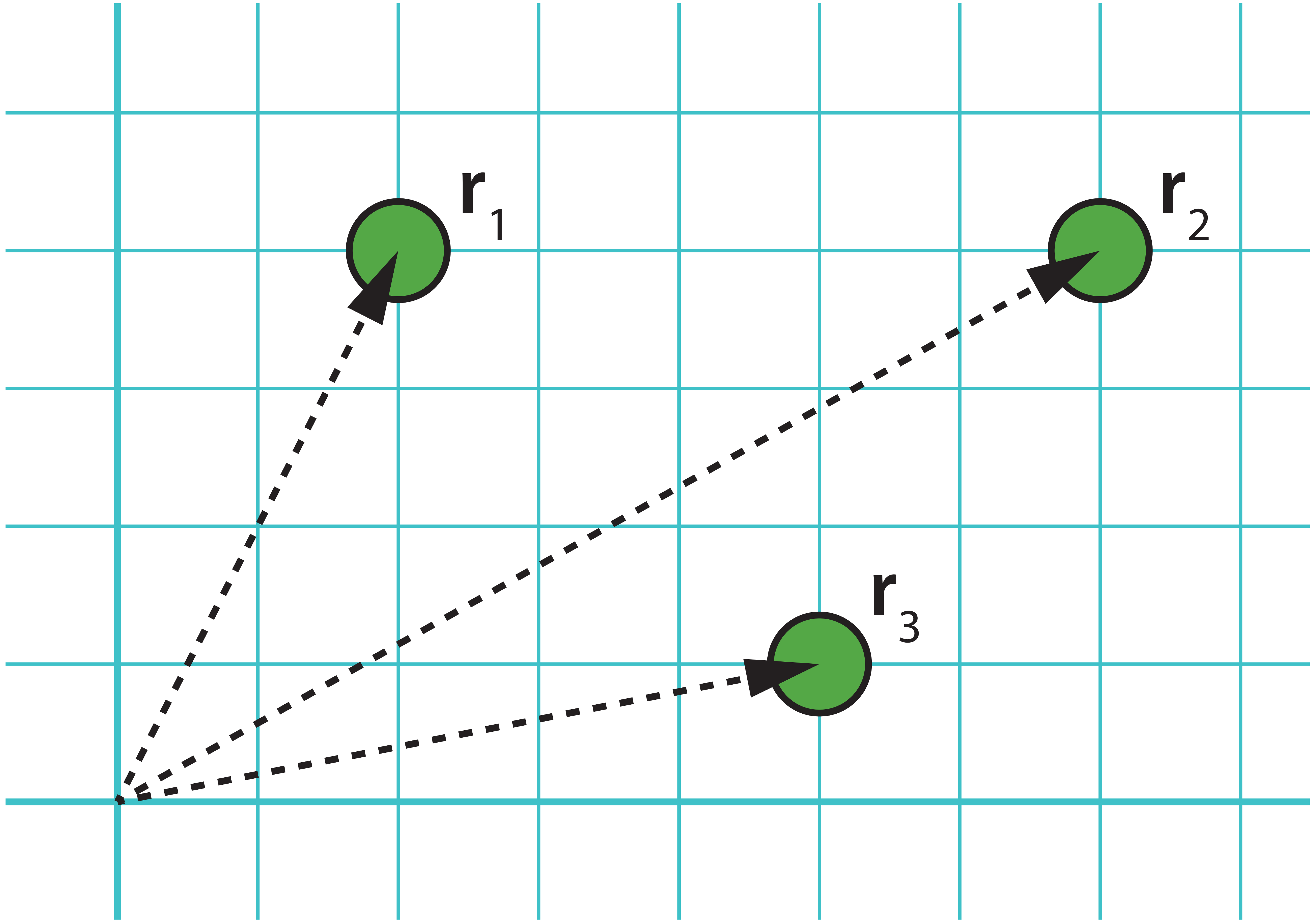


EXAMPLE: ATOM POSITIONS



- positions are defined relative to some reference point (the origin)

EXAMPLE: ATOM POSITIONS



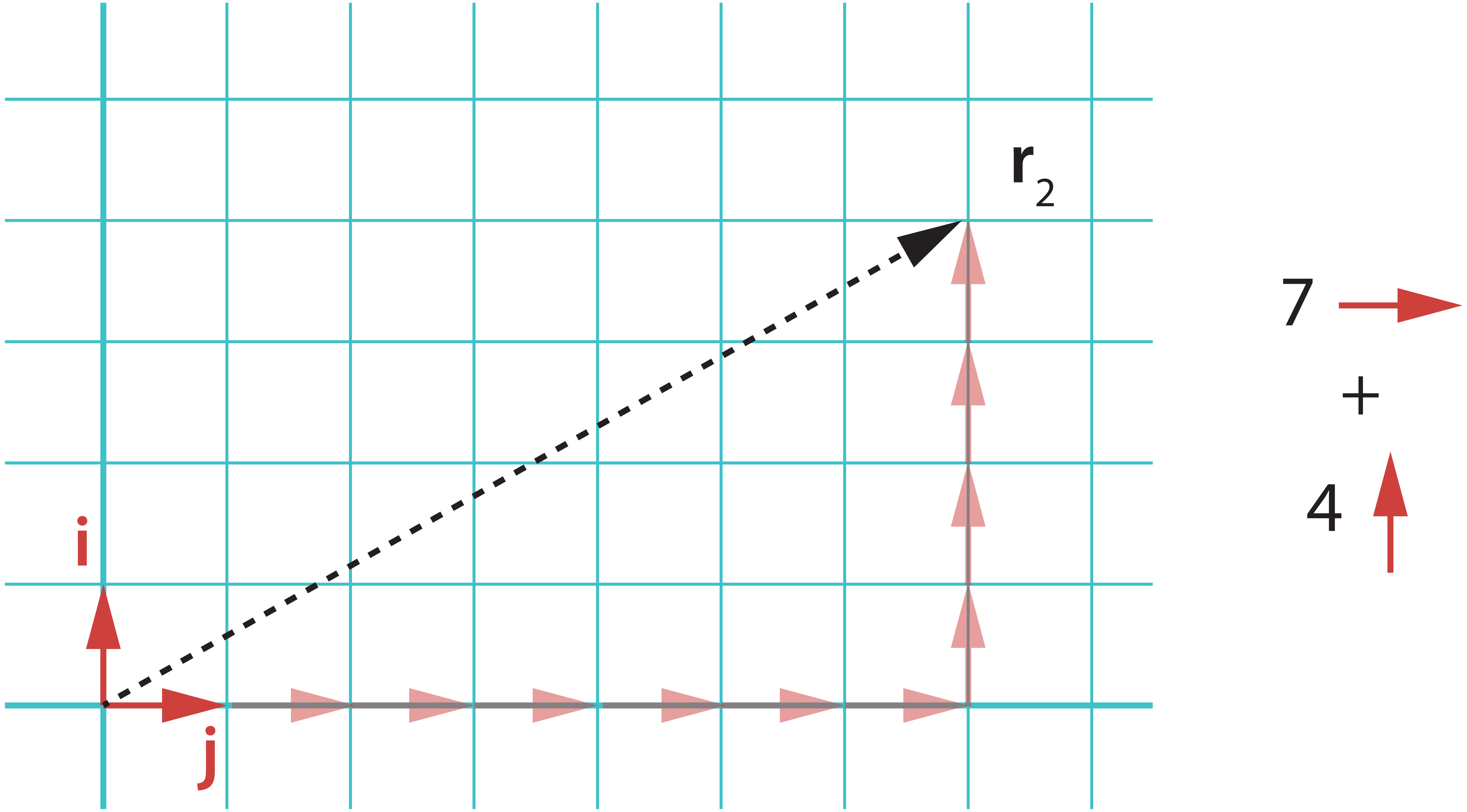
$\mathbf{r}_1 = (2,4)$

$\mathbf{r}_2 = (7,4)$

$\mathbf{r}_3 = (5,1)$

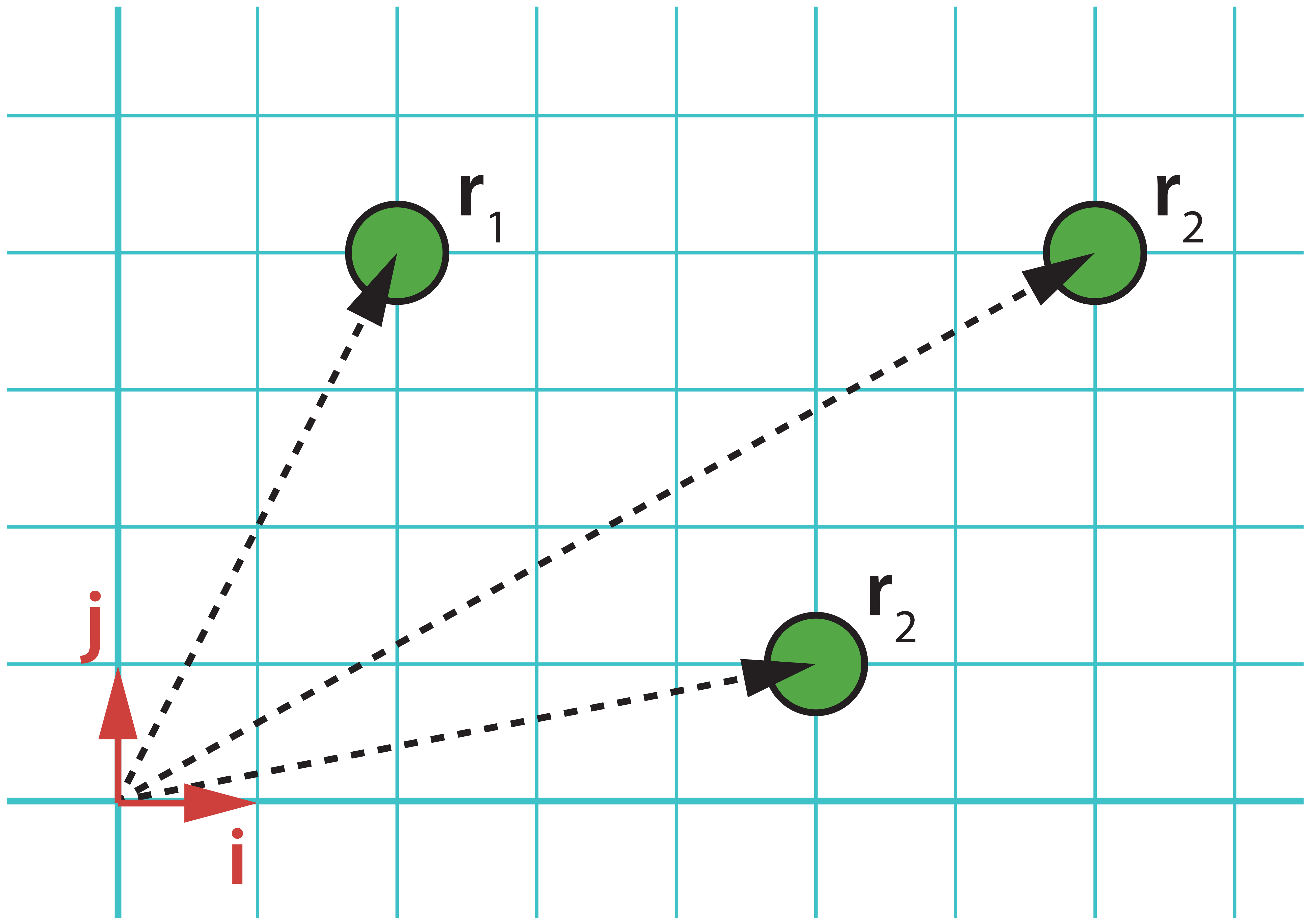
- Cartesian coordinates (e.g. week 2 — interatomic distances)

EXAMPLE: ATOM POSITIONS



- **Basis vectors** are implied by the coordinate system
- All position vectors are linear combinations of the basis vectors: $\mathbf{r}_\alpha = x_\alpha \mathbf{i} + y_\alpha \mathbf{j}$

EXAMPLE: ATOM POSITIONS



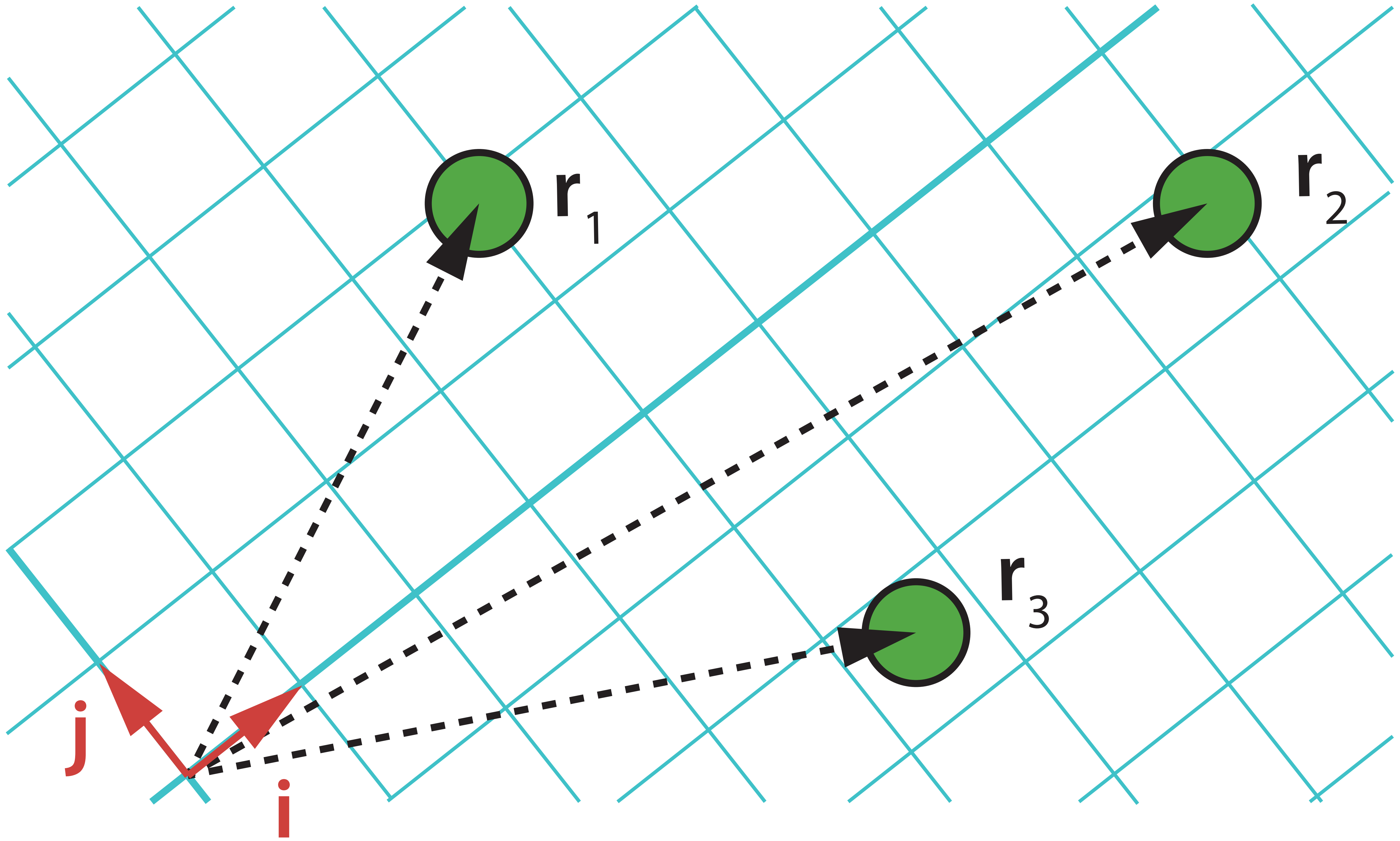
$$\mathbf{r}_1 = 2\mathbf{i} + 4\mathbf{j}$$

$$\mathbf{r}_2 = 7\mathbf{i} + 4\mathbf{j}$$

$$\mathbf{r}_3 = 5\mathbf{i} + 1\mathbf{j}$$

- (x,y) coordinates give coefficients in linear combination of basis vectors: $\mathbf{r}_\alpha = x_\alpha \mathbf{i} + y_\alpha \mathbf{j}$

WHAT IF WE CHOOSE A DIFFERENT BASIS?



$\mathbf{r}_1 = 4\mathbf{i} + 2\mathbf{j}$

$\mathbf{r}_2 = 7.9\mathbf{i} - 1.3\mathbf{j}$

$\mathbf{r}_3 = 4.6\mathbf{i} + 2.3\mathbf{j}$

- We can represent the same positions using different coefficients (because the basis is different) → $\mathbf{r}'_\alpha = x_\alpha \mathbf{i}' + y_\alpha \mathbf{j}'$

VECTOR NOTATION

- Bold upright symbols: $\mathbf{r}$, $\mathbf{i}$, $\mathbf{j}$
- Arrows: $\vec{r}$, $\vec{i}$, $\vec{j}$

VECTOR NOTATION

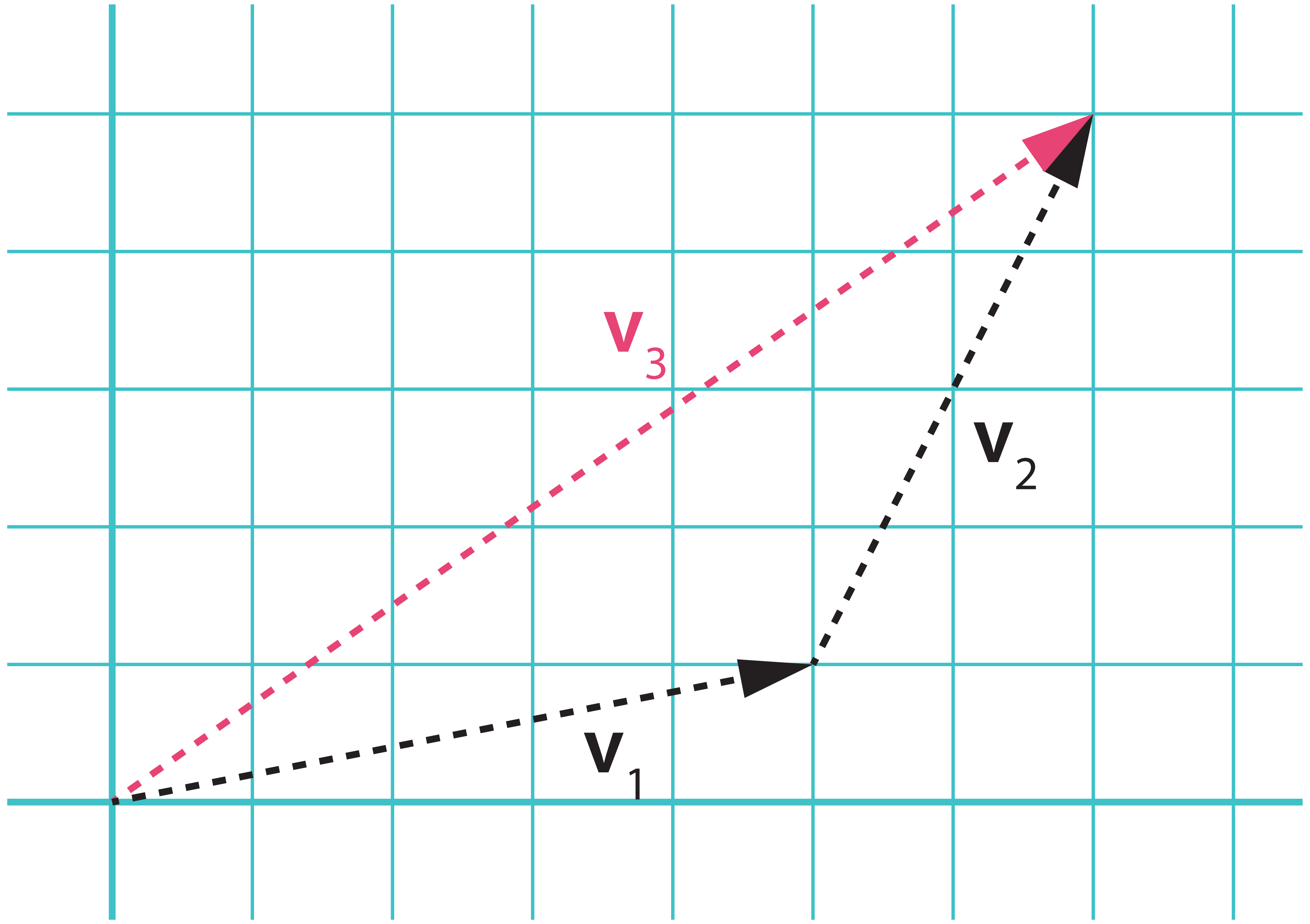
- Bold upright symbols: $\mathbf{r}, \mathbf{i}, \mathbf{j}$
- Arrows: $\vec{r}, \vec{i}, \vec{j}$
- List of coefficients: $(3{,}4)$
- Column vector: $\begin{bmatrix} 3 \\ 4 \end{bmatrix}$

VECTORS USING NUMPY

DEMO

VECTOR ADDITION, SUBTRACTION, SCALING, AND "MULTIPLICATION"

- Vector addition



$\mathbf{v}_1 = (5,1)$

$\mathbf{v}_2 = (2,4)$

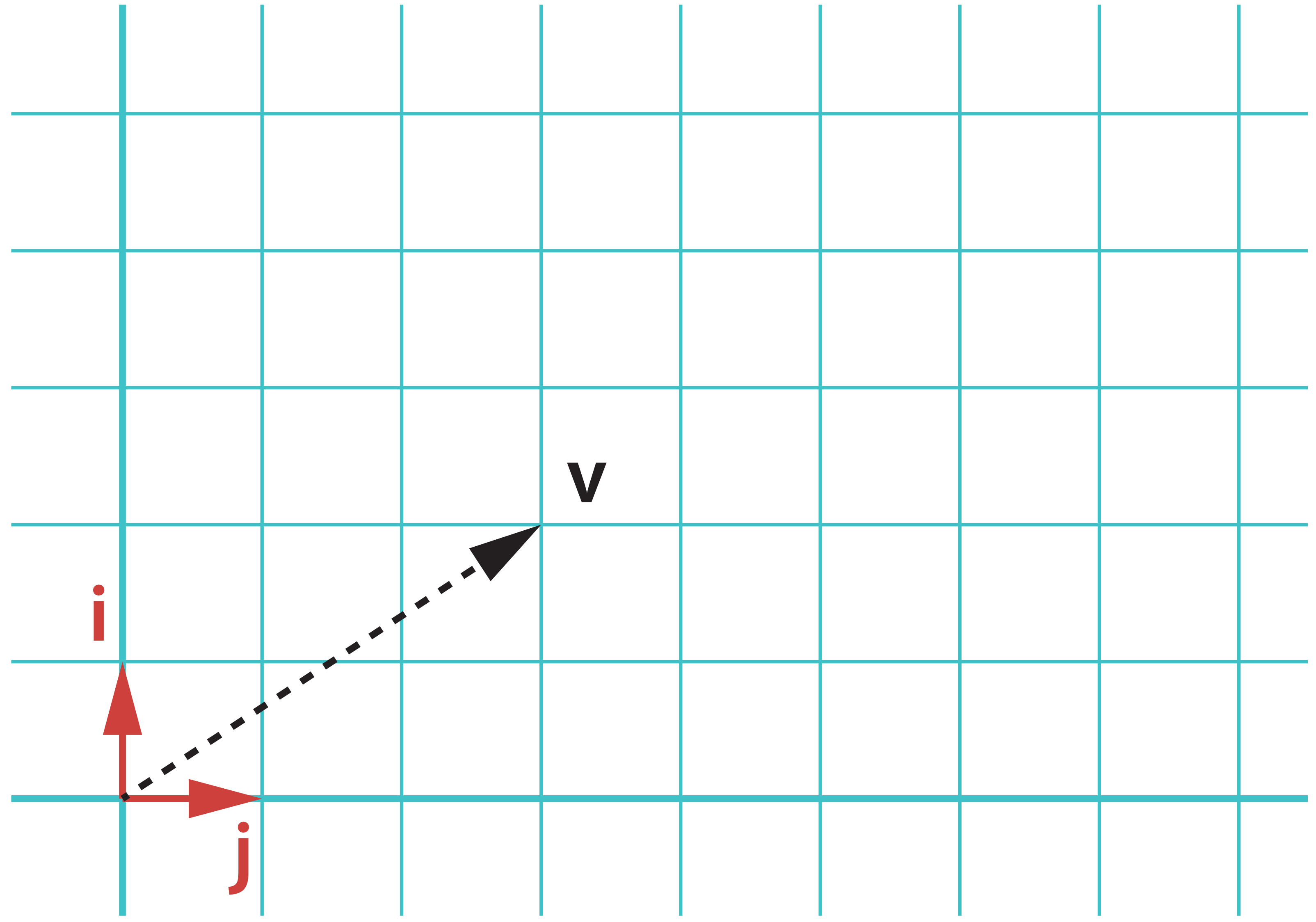
$\mathbf{v}_3 = (5+2, 2+4)$

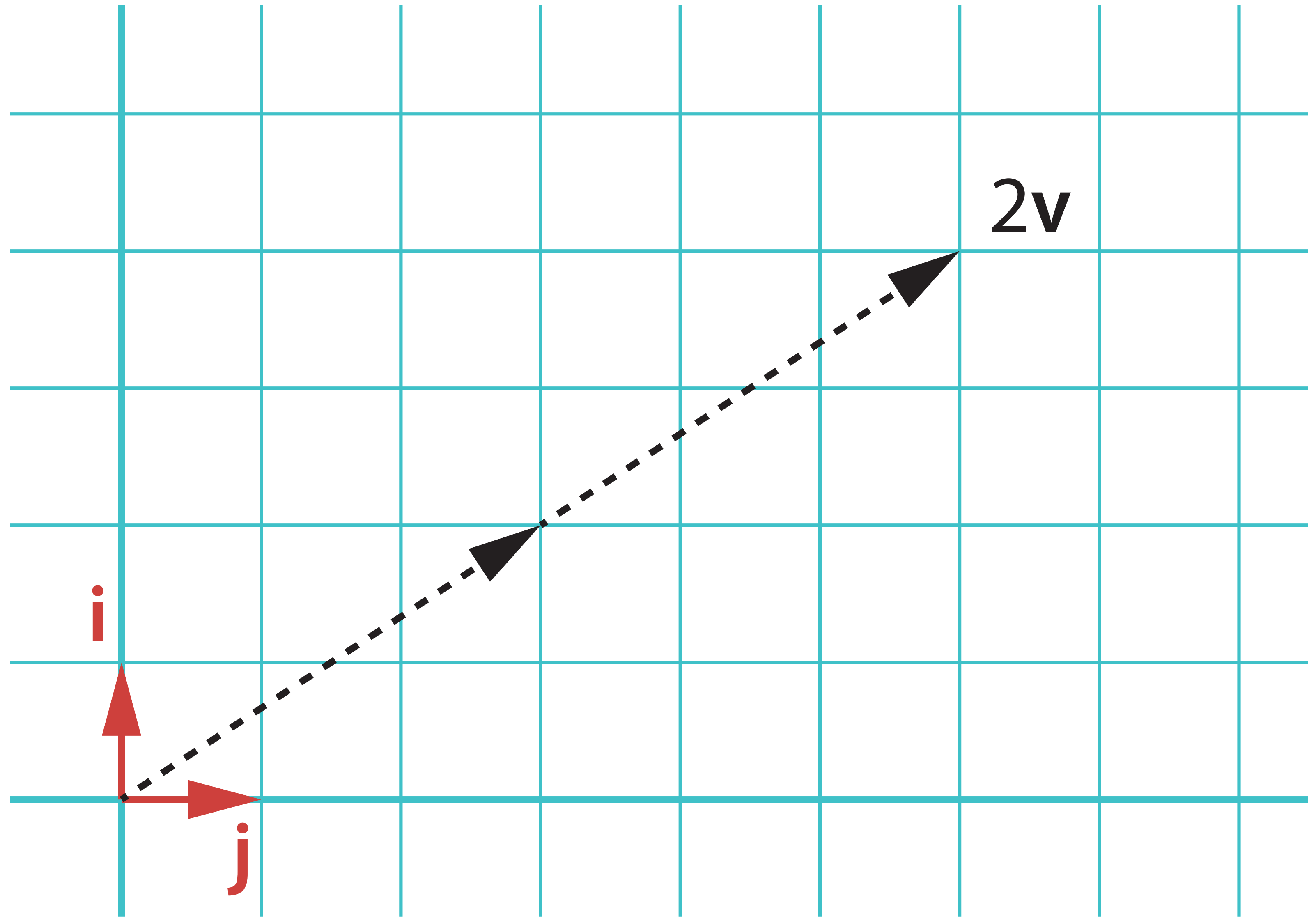
$= (7,6)$

VECTOR ADDITION, SUBTRACTION, SCALING, AND “MULTIPLICATION”

- Vector scaling

$$\mathbf{v} = \begin{bmatrix} 2 \\ 3 \end{bmatrix} \qquad \mathbf{v} \times 2 = \begin{bmatrix} 2 \times 2 \\ 3 \times 2 \end{bmatrix} = \begin{bmatrix} 4 \\ 6 \end{bmatrix}$$





VECTOR ADDITION, SUBTRACTION, SCALING, AND “MULTIPLICATION”

- dot product

$$\mathbf{a} \cdot \mathbf{b} = \sum_i a_i b_i = a_1 b_1 + a_2 b_2 + \ldots + a_n b_n$$

VECTOR ADDITION, SUBTRACTION, SCALING, AND "MULTIPLICATION"

- dot product

$$\mathbf{a} \cdot \mathbf{b} = \sum_i a_i b_i = a_1 b_1 + a_2 b_2 + \ldots + a_n b_n$$

$$\mathbf{a} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

VECTOR ADDITION, SUBTRACTION, SCALING, AND "MULTIPLICATION"

- dot product

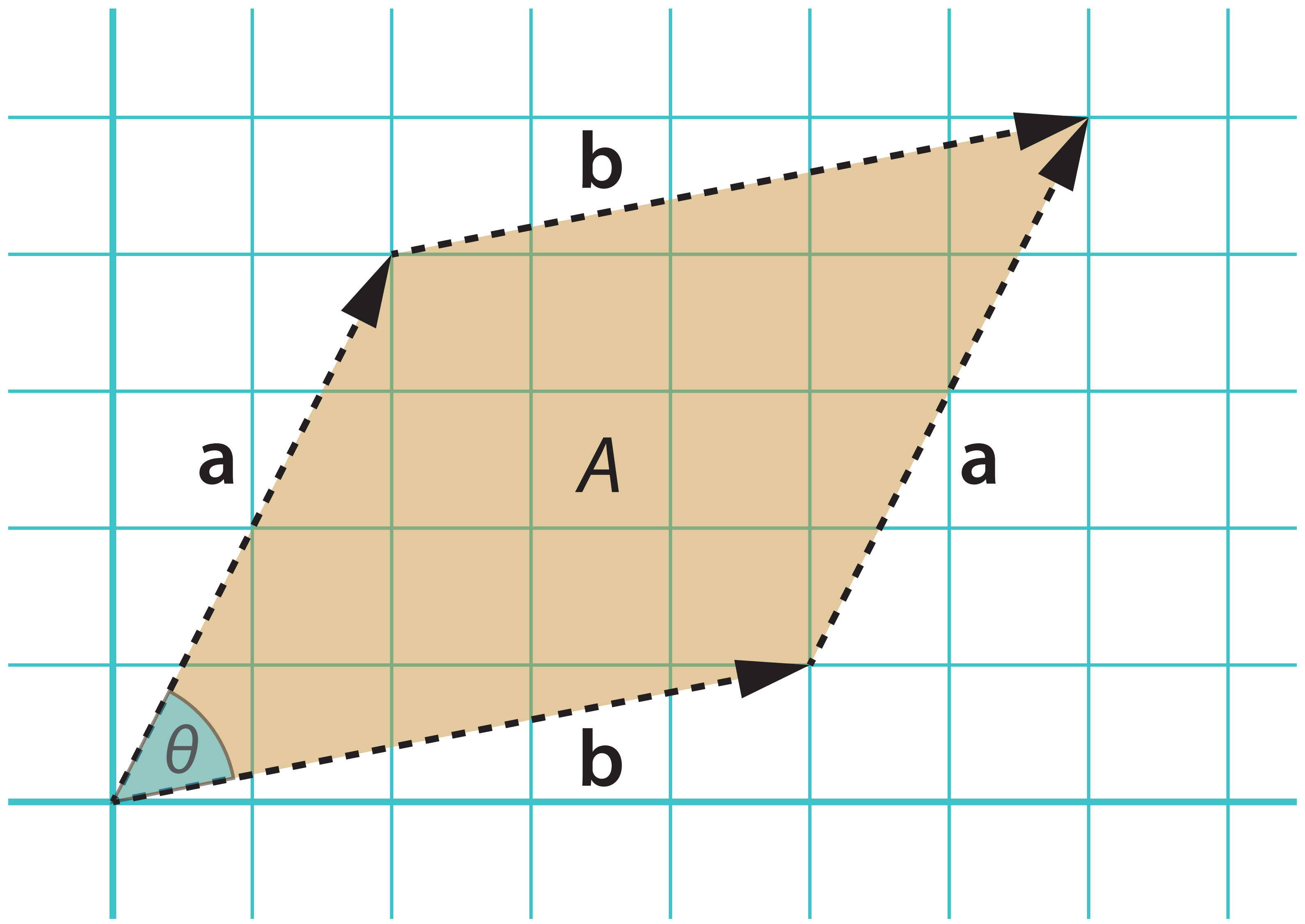
$$\mathbf{a} \cdot \mathbf{b} = \sum_i a_i b_i = a_1 b_1 + a_2 b_2 + \ldots + a_n b_n$$

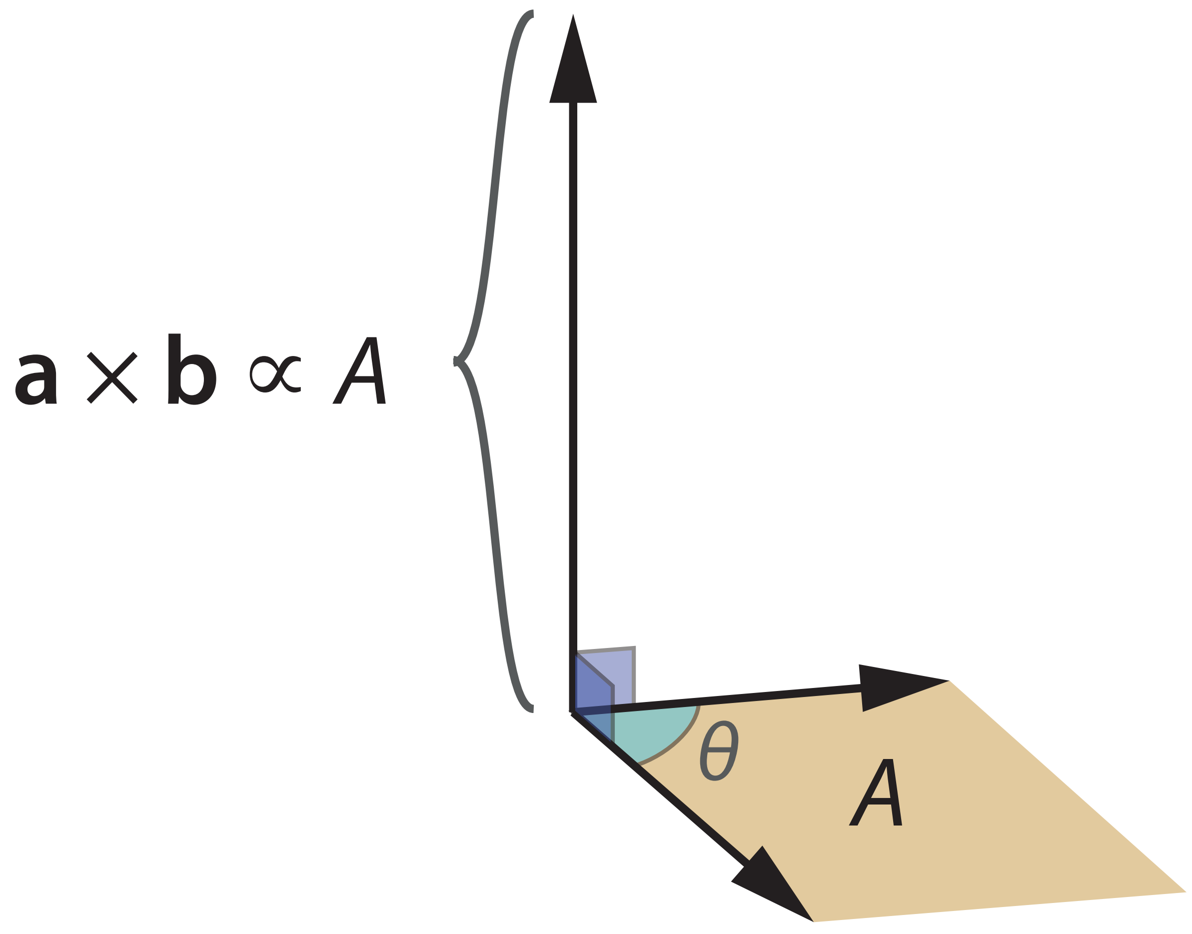
$$\mathbf{a} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

$$\mathbf{a} \cdot \mathbf{b} = \begin{bmatrix} 2 \\ 3 \end{bmatrix} \cdot \begin{bmatrix} 3 \\ 1 \end{bmatrix} = (2 \times 3) + (3 \times 1) = 9$$

VECTOR ADDITION, SUBTRACTION, SCALING, AND "MULTIPLICATION"

- cross product (more complicated — here for completion → 3D geometry)





$$\mathbf{a} \times \mathbf{b} = \| \mathbf{a} \| \| \mathbf{b} \| \sin\theta \, \mathbf{n}$$

VECTOR ALGEBRA USING NUMPY

DEMO

EXERCISE 1

7 Problems

7.1 Interatomic distances

Write code that can take the x, y, and z coordinates of three atoms and calculate the distances r_{ij} between each pair. For each pair of atoms, print the interatomic distance.

The equation for the distance r_{ij} between two atoms i and j is,

$$r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}. \tag{2}$$

Remember: Plan the structure of your program before you start to write any code.

Download the `molecule1.txt` and `molecule2.txt` files from Moodle[4], and copy these into your working directory (e.g. `H:/CH40208/week2`. Each file contains three columns, labelled x, y, and z, which you can read the atomic coordinates from. To calculate the distances between each pair of atoms you will need to use a pair of *nested* loops. What do these distances tell you about the shapes of these molecules?

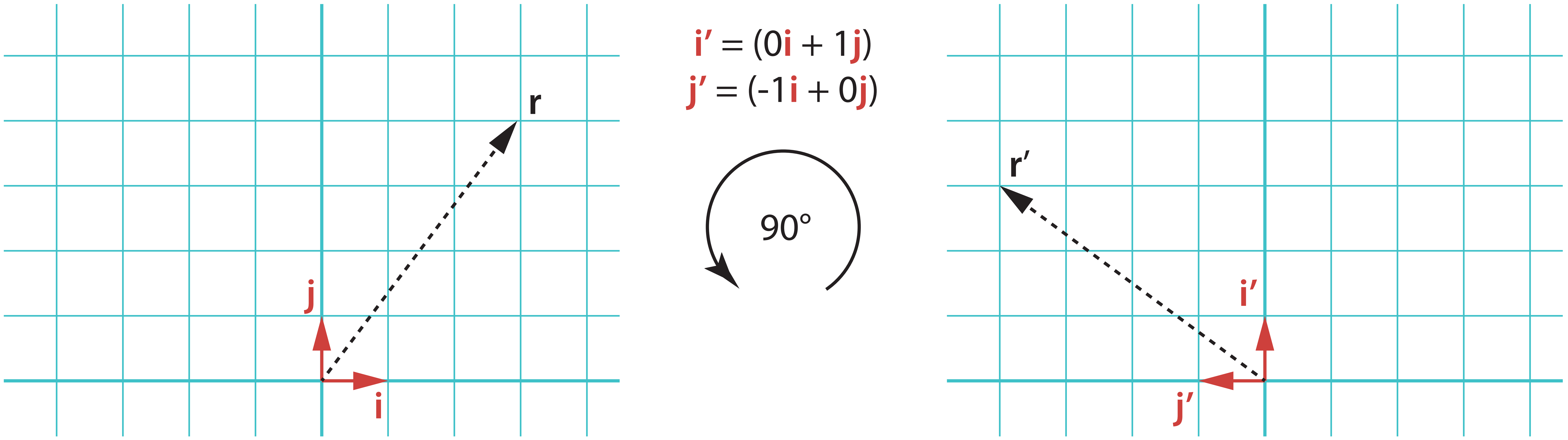
- copy and edit (or start from scratch) & rewrite using `numpy` vectors and `np.dot()`

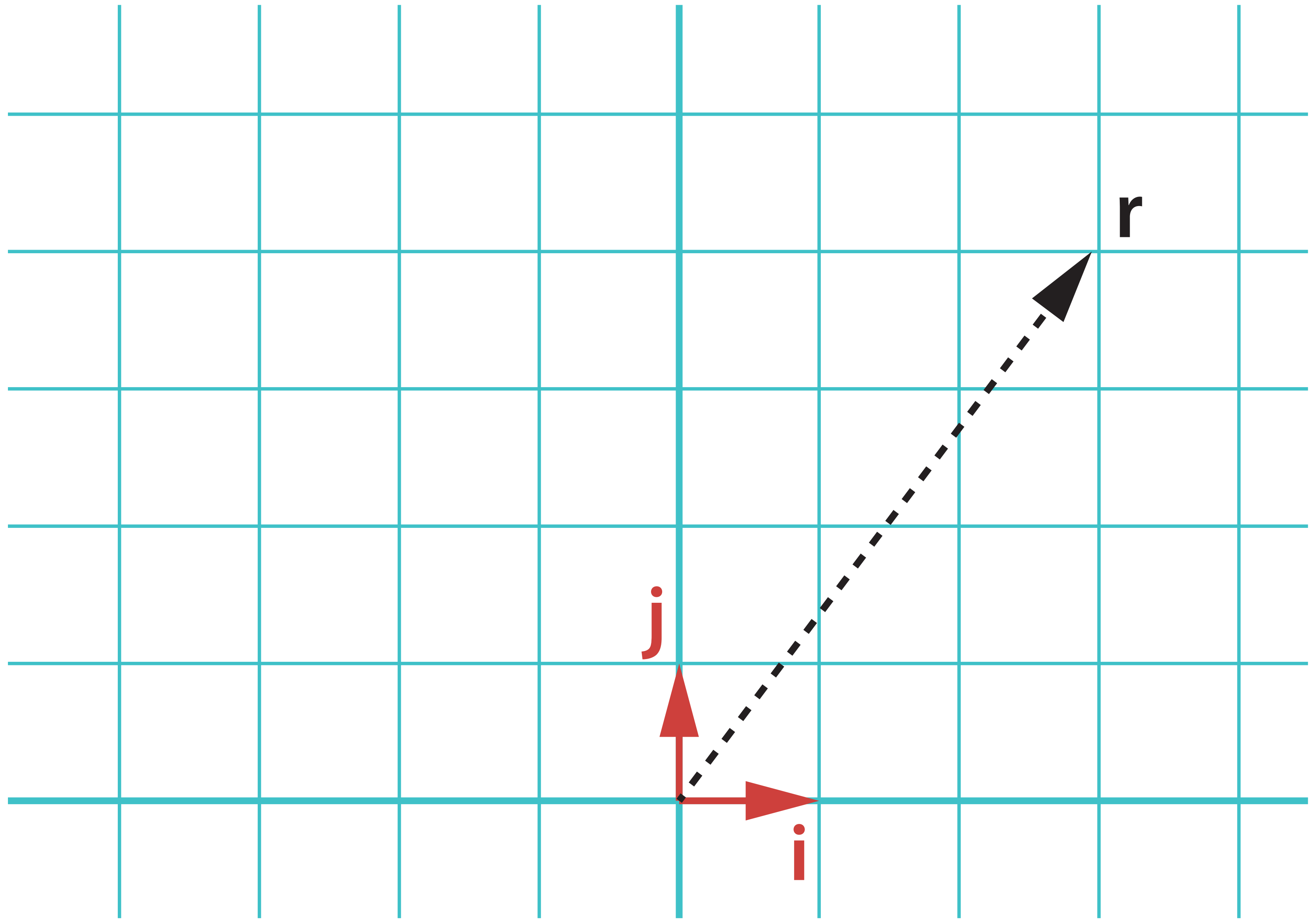
MATRICES AS LINEAR TRANSFORMATIONS

- What is a matrix?

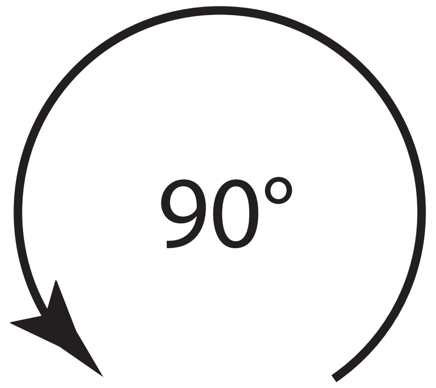
MATRICES AS LINEAR TRANSFORMATIONS

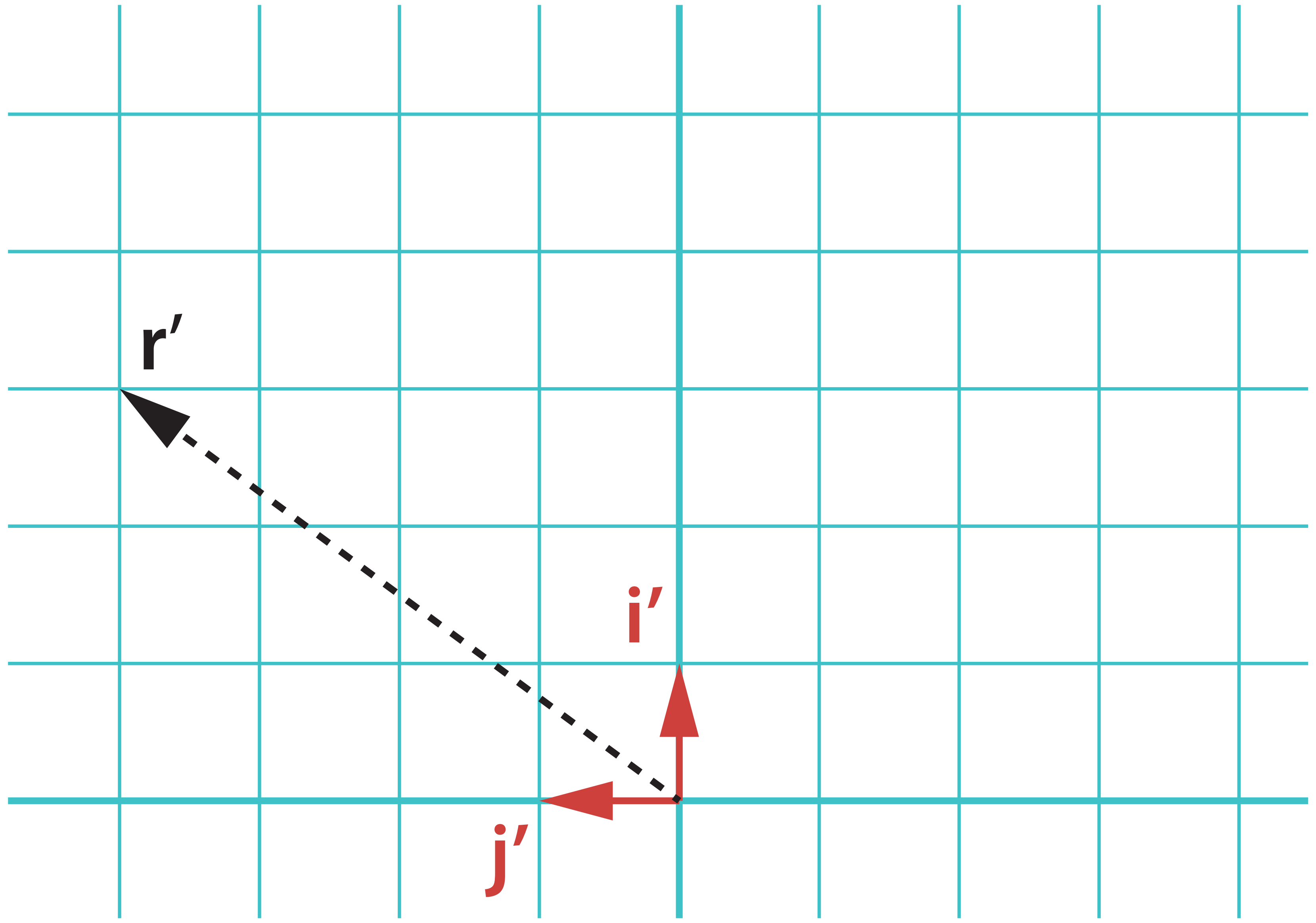
- Change of basis $\mathbf{i} \rightarrow \mathbf{i}',\ \mathbf{j} \rightarrow \mathbf{j}'$
- Example: rotation by 90°.





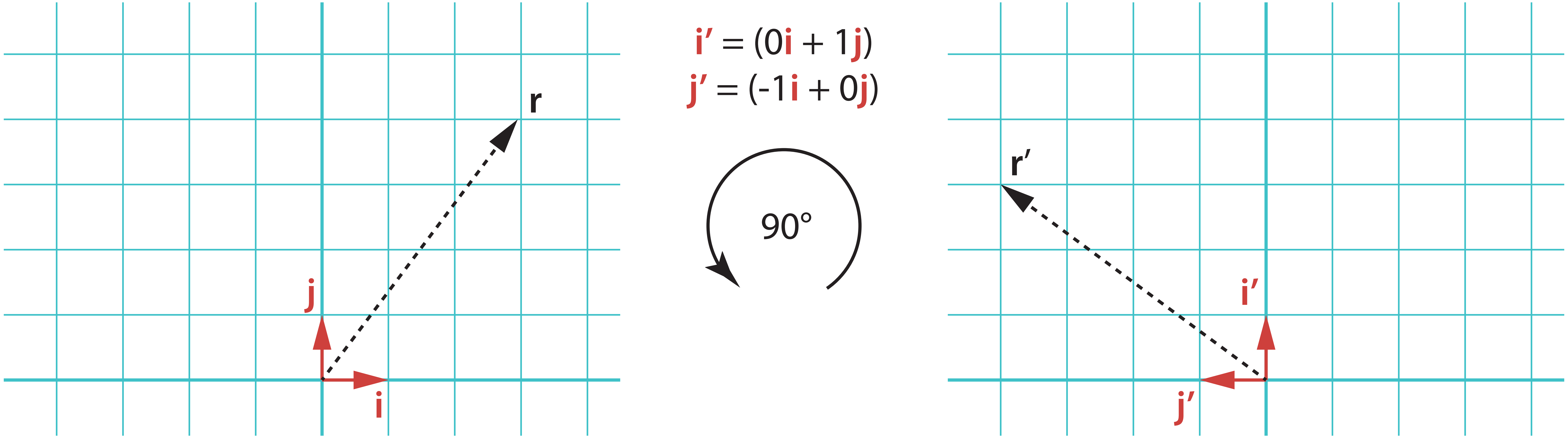
$\mathbf{i}' = (0\mathbf{i} + 1\mathbf{j})$
$\mathbf{j}' = (-1\mathbf{i} + 0\mathbf{j})$





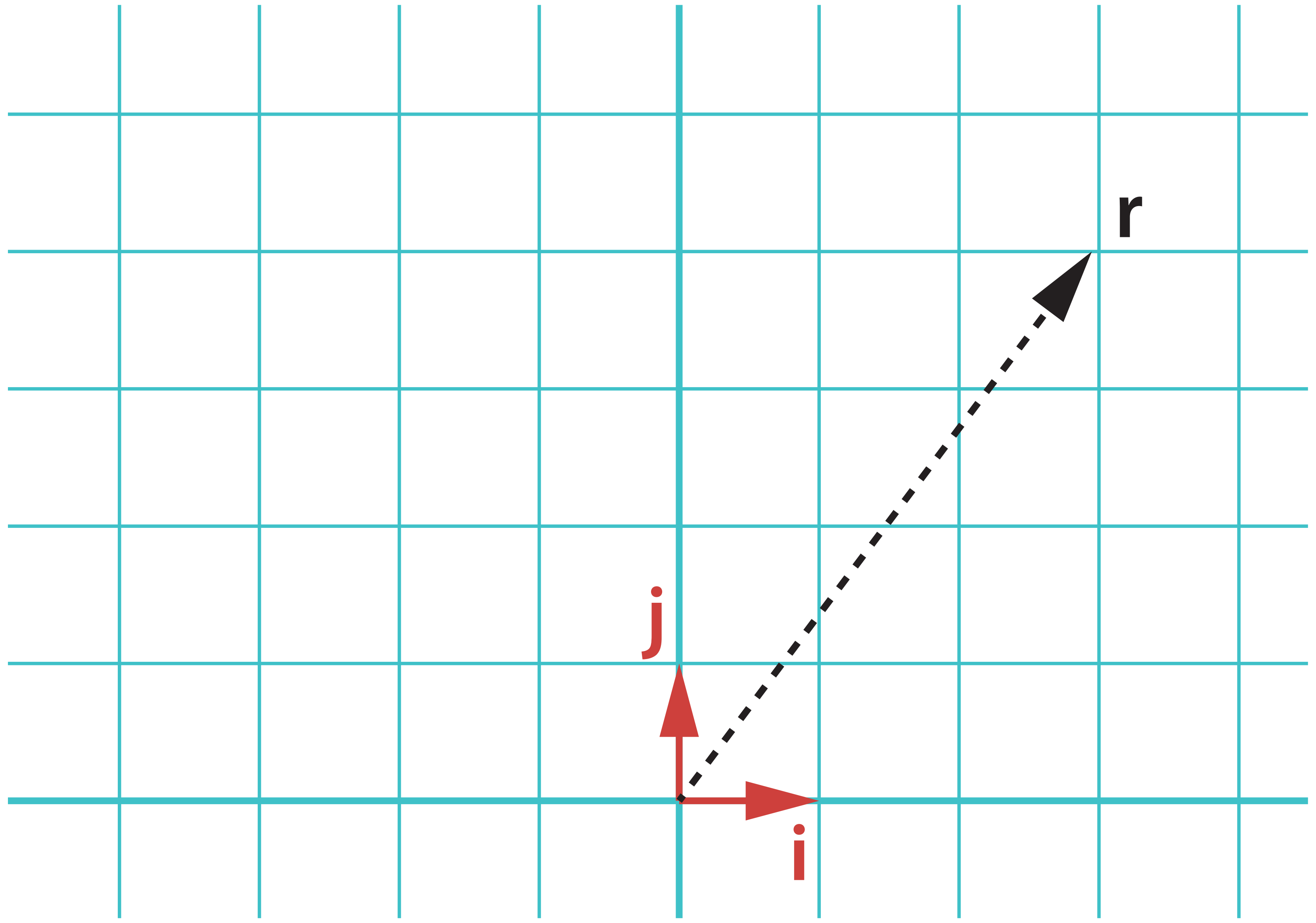
$$\mathbf{r} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

$$\mathbf{r}' = \begin{bmatrix} 3 \\ 4 \end{bmatrix}' = \begin{bmatrix} ? \\ ? \end{bmatrix}$$



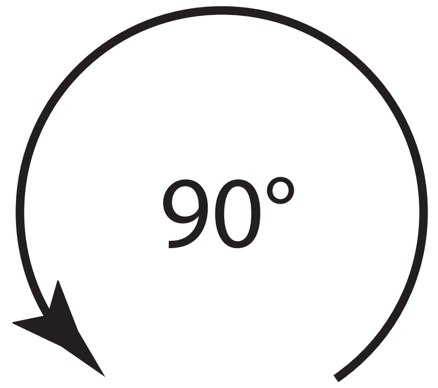
$$\mathbf{r} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

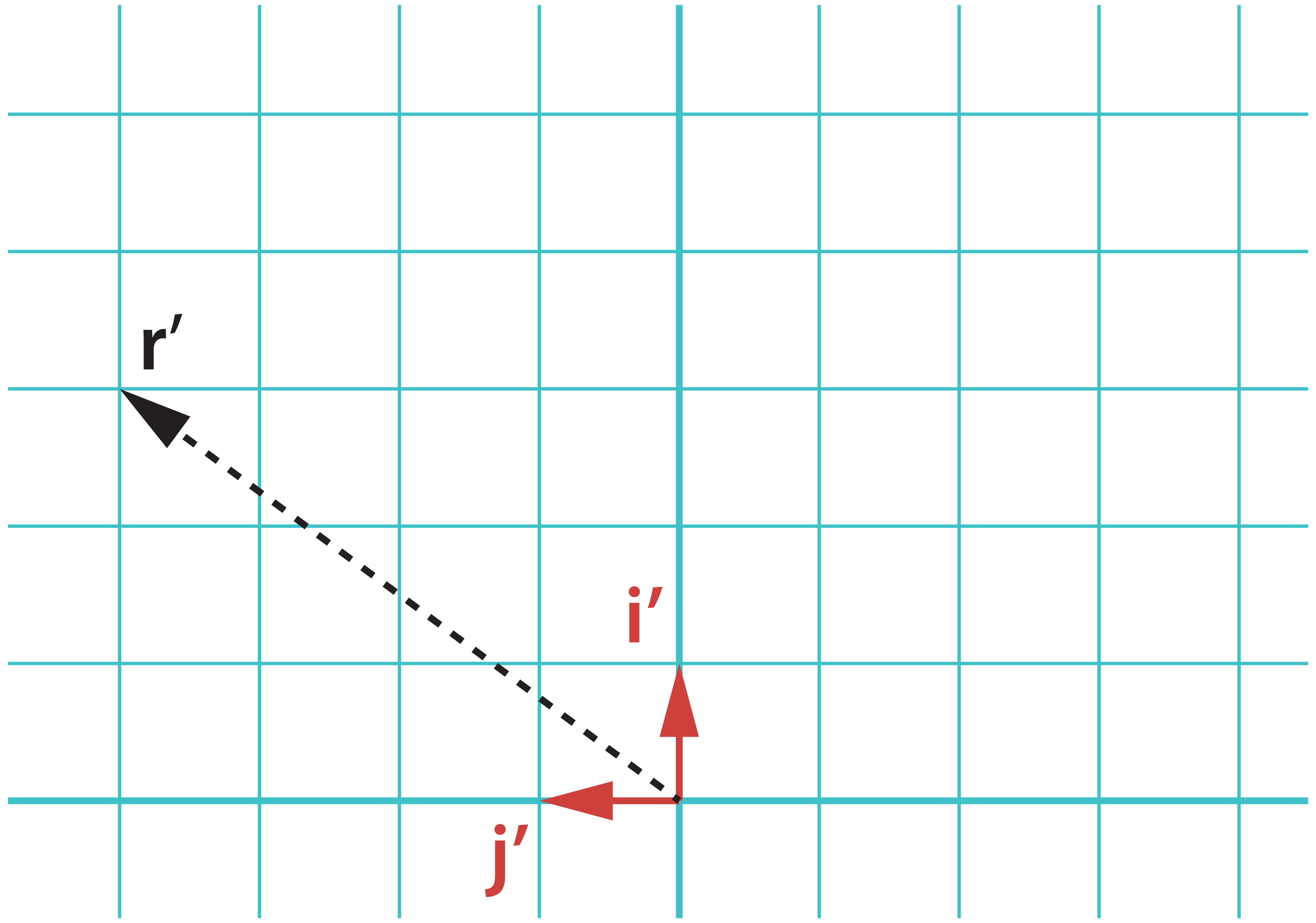
$$\mathbf{r}' = \begin{bmatrix} 3 \\ 4 \end{bmatrix}' = (3\mathbf{i}' + 4\mathbf{j}')$$



$\mathbf{i}' = (0\mathbf{i} + 1\mathbf{j})$

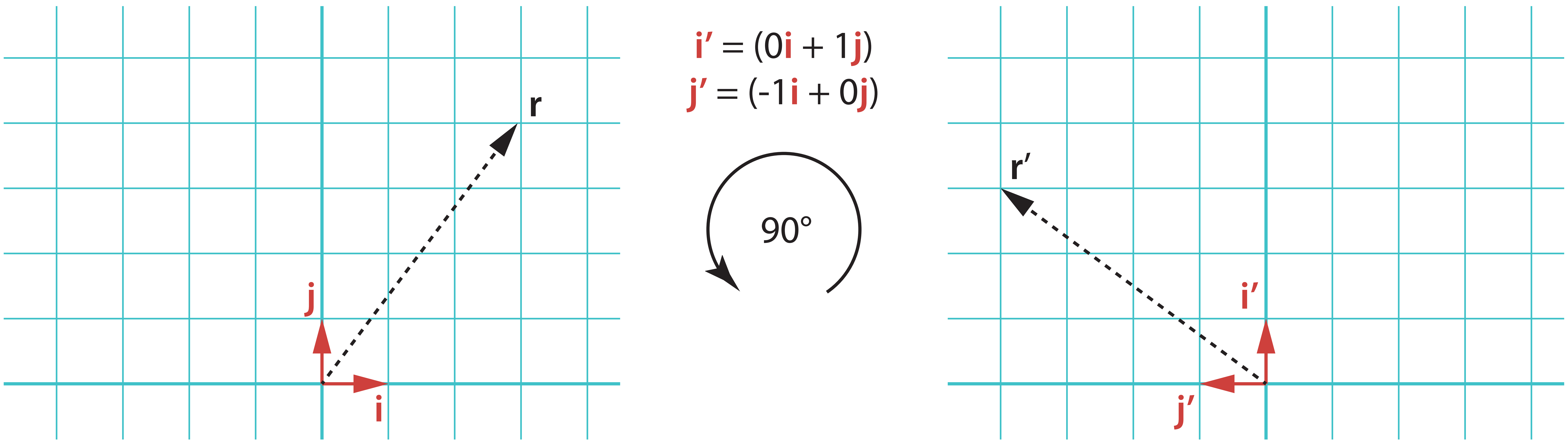
$\mathbf{j}' = (-1\mathbf{i} + 0\mathbf{j})$





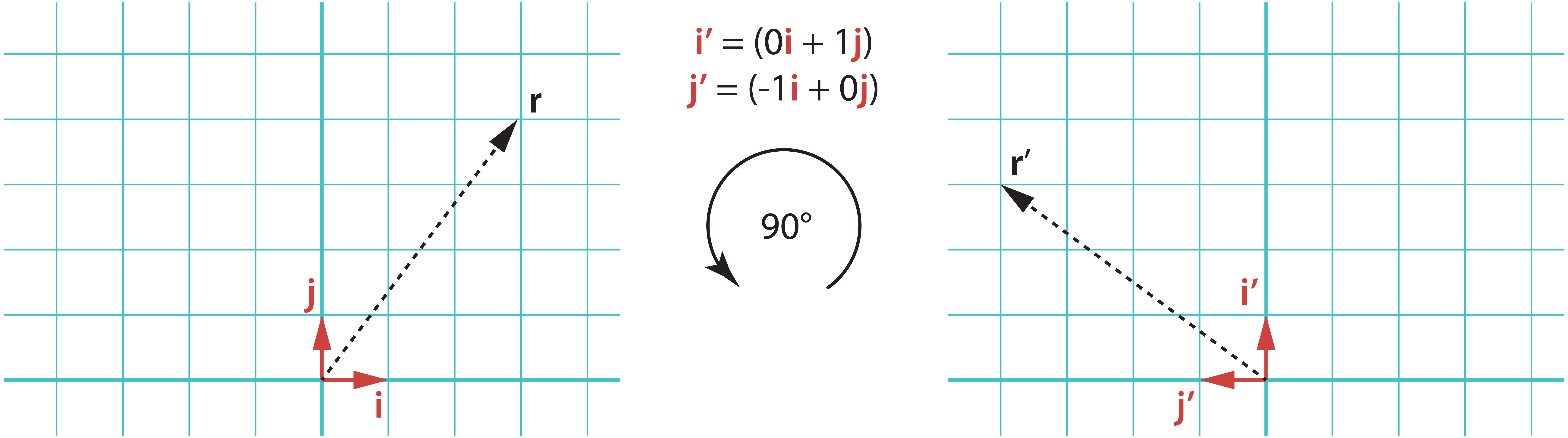
$$\mathbf{i} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad \mathbf{i}' = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\mathbf{j} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad \mathbf{j}' = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$$



$$\mathbf{r}' = \begin{bmatrix} 3 \\ 4 \end{bmatrix}' = (3\mathbf{i}' + 4\mathbf{j}')$$

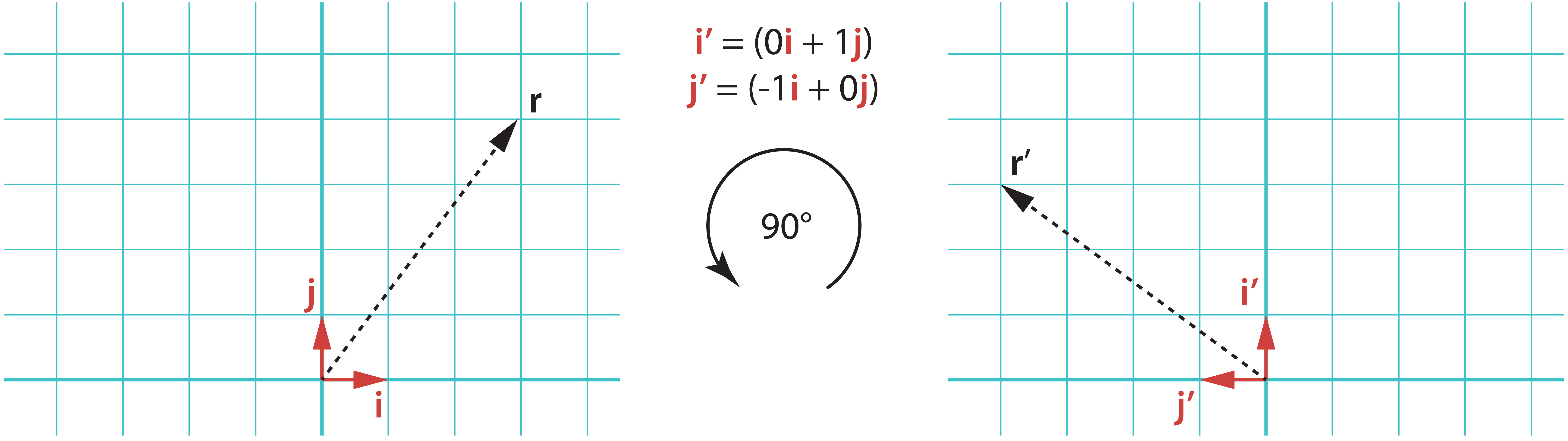
$$\mathbf{r}' = 3\begin{bmatrix} 0 \\ 1 \end{bmatrix} + 4\begin{bmatrix} -1 \\ 0 \end{bmatrix} = \begin{bmatrix} -4 \\ 3 \end{bmatrix}$$



$$\mathbf{r}' = \begin{bmatrix} 3 \\ 4 \end{bmatrix}' = (3\mathbf{i}' + 4\mathbf{j}')$$

$$\mathbf{r}' = 3\begin{bmatrix} 0 \\ 1 \end{bmatrix} + 4\begin{bmatrix} -1 \\ 0 \end{bmatrix} = \begin{bmatrix} -4 \\ 3 \end{bmatrix}$$

- can express as **matrix** × **vector** → $\mathbf{r}' = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 3 \\ 4 \end{bmatrix}$



$$\mathbf{r}' = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

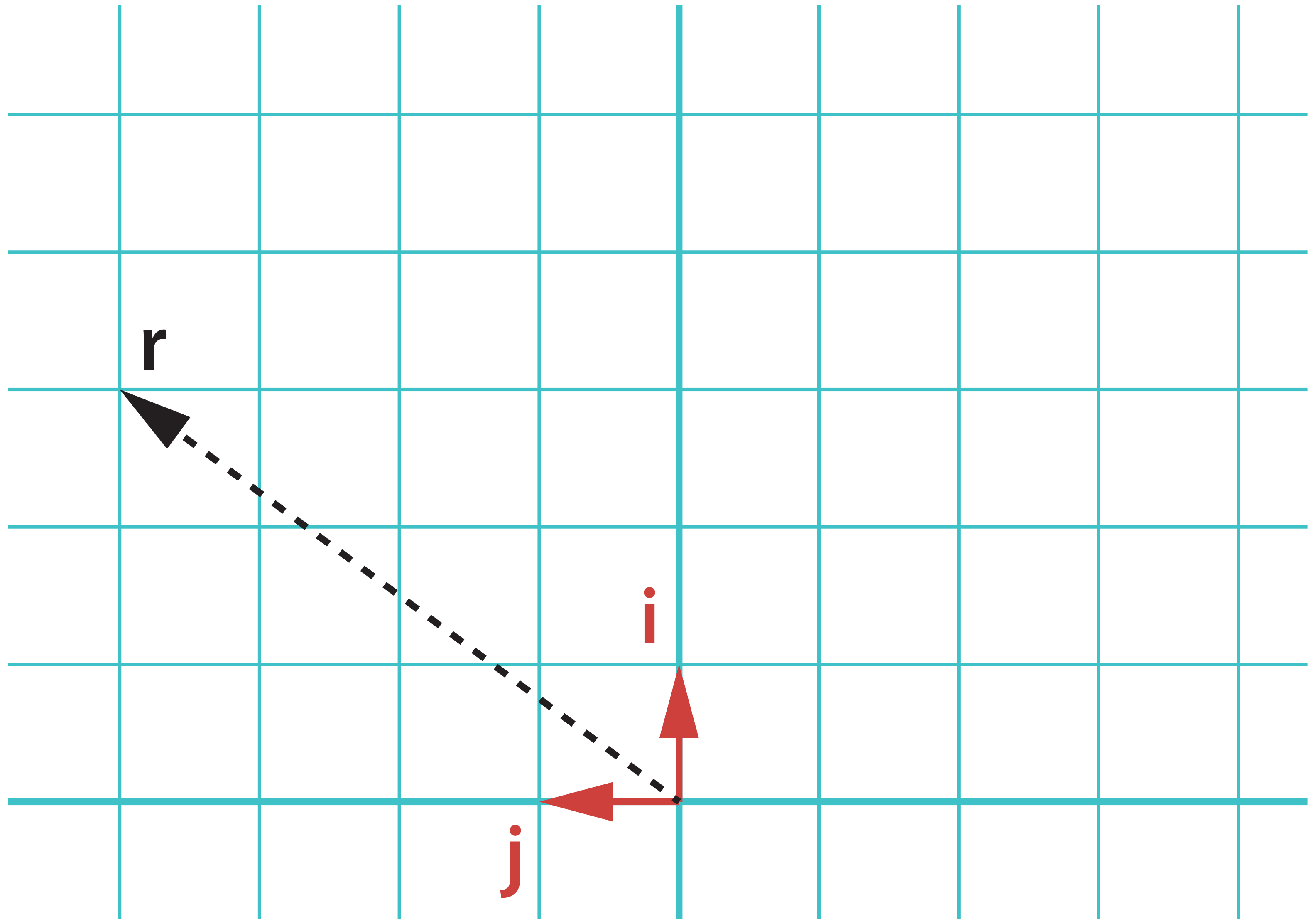
- a matrix describes a basis set transformation.
- **i** and **j** are mapped to the columns of the matrix (in 2D cases)

MATRIX ALGEBRA USING NUMPY

DEMO

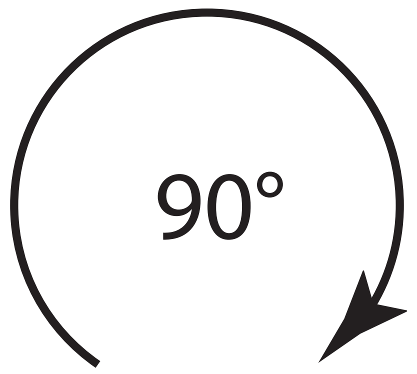
INVERTING MATRICES

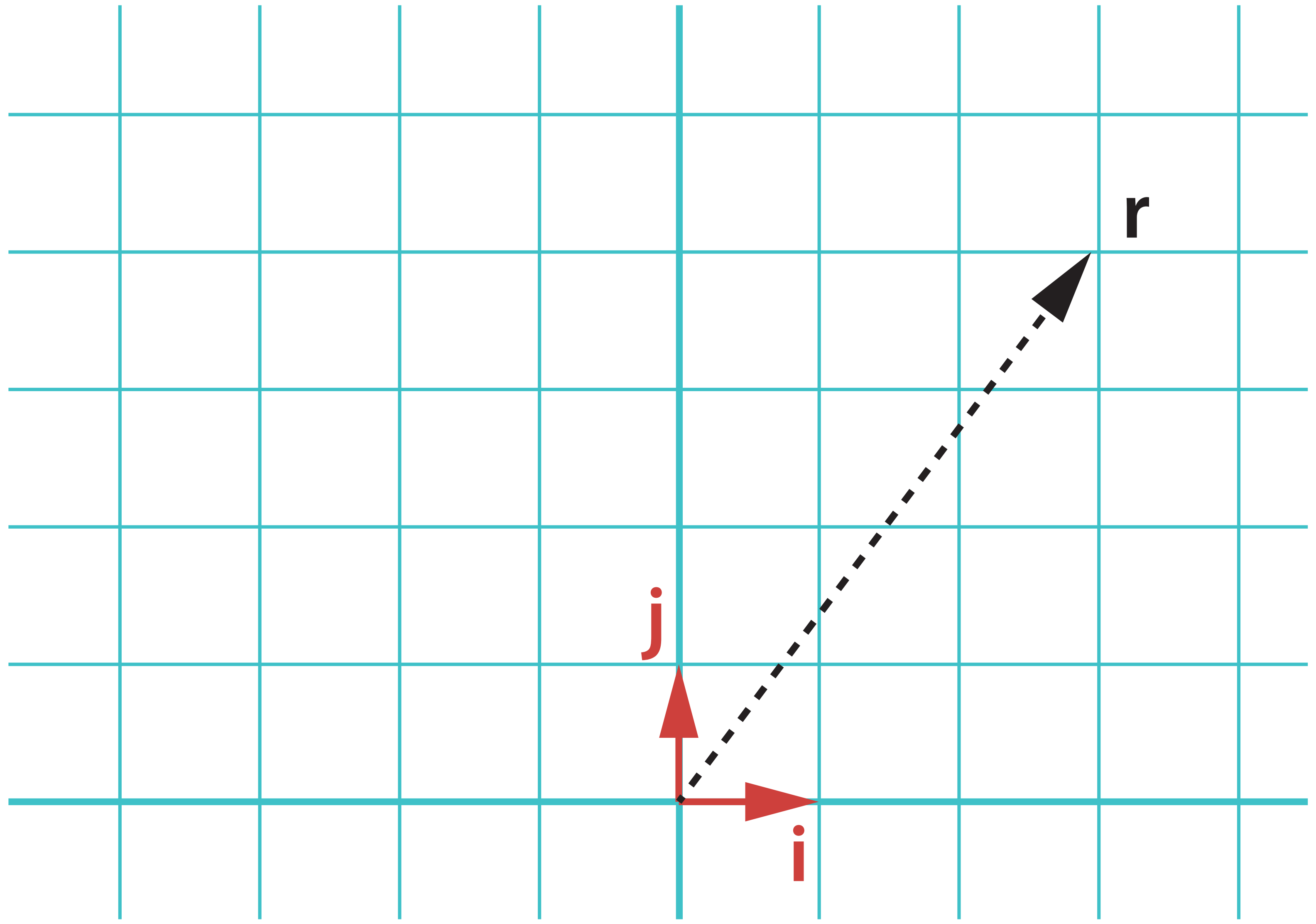
- Defining the inverse operation (rotation by 90° clockwise)



$\mathbf{i}' = (0\mathbf{i} - 1\mathbf{j})$

$\mathbf{j}' = (1\mathbf{i} + 0\mathbf{j})$





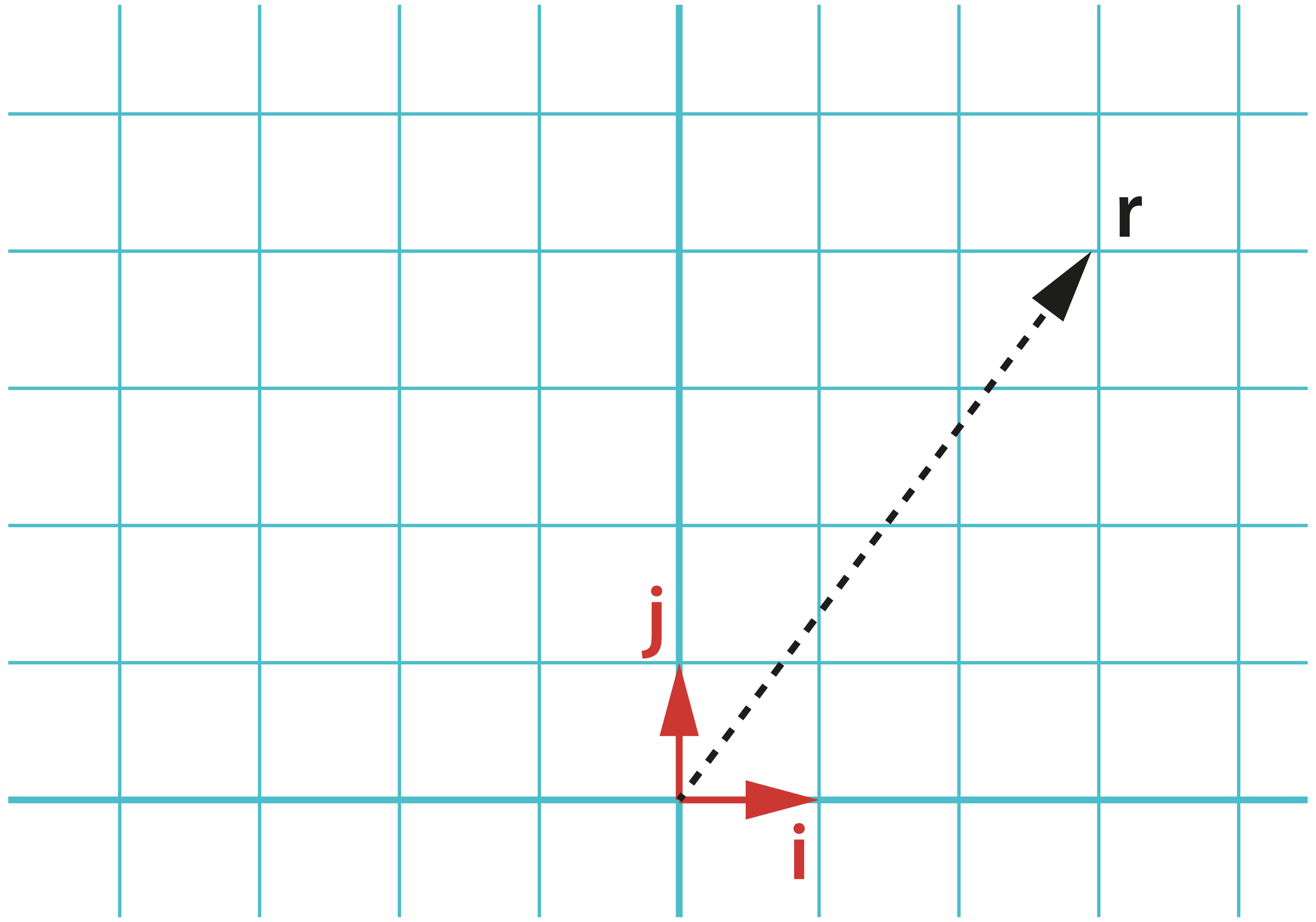
$\mathbf{r} = 3\mathbf{i} + 4\mathbf{j}$

$\mathbf{r}' = 3\mathbf{i}' + 4\mathbf{j}'$

MATRIX INVERSION USING NUMPY

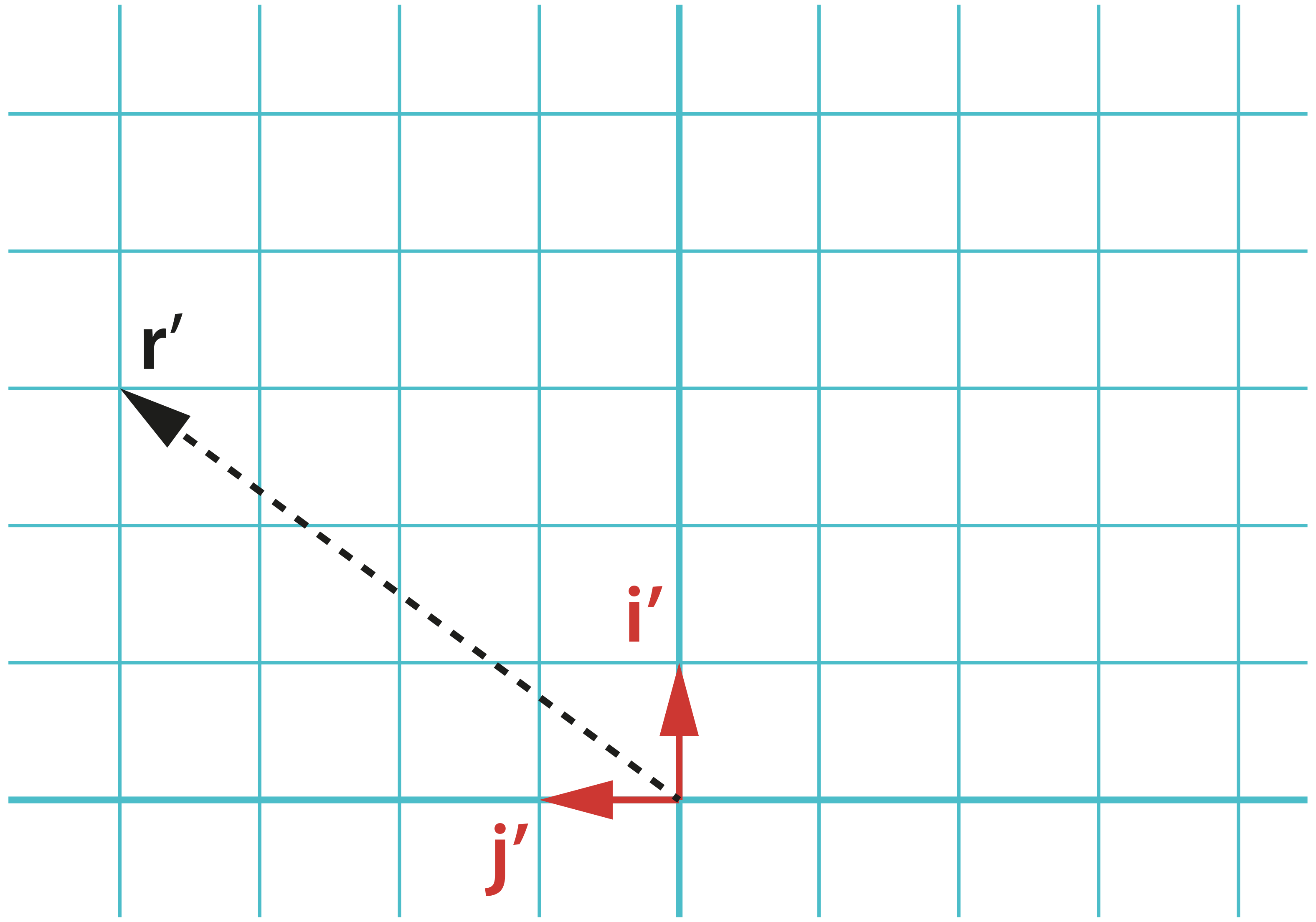
DEMO

MATRIX × MATRIX



$\mathbf{i}' = (0\mathbf{i} + 1\mathbf{j})$
$\mathbf{j}' = (-1\mathbf{i} + 0\mathbf{j})$

90°



$\mathbf{i}'' = (?\mathbf{i} + ?\mathbf{j})$
$\mathbf{j}'' = (?\mathbf{i} + ?\mathbf{j})$

90°

$$\mathbf{i} \rightarrow \mathbf{i}' \rightarrow \mathbf{i}''$$

$$\mathbf{j} \rightarrow \mathbf{j}' \rightarrow \mathbf{j}''$$

MATRIX × MATRIX

$$\mathbf{i}' = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad \mathbf{i}'' = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad \mathbf{i}'' = 0 \begin{bmatrix} 0 \\ 1 \end{bmatrix} + 1 \begin{bmatrix} -1 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$$

$$\mathbf{j}' = \begin{bmatrix} -1 \\ 0 \end{bmatrix} \qquad \mathbf{j}'' = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} -1 \\ 0 \end{bmatrix} \qquad \mathbf{j}'' = -1 \begin{bmatrix} 0 \\ 1 \end{bmatrix} + 0 \begin{bmatrix} -1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$

$$\mathbf{MM} = \begin{bmatrix} \mathbf{i}'' & \mathbf{j}'' \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$$

MATRIX × MATRIX

DEMO

EXERCISE 2

3 Problem

This week we will look at the rotation of a molecule on a surface (such that the z coordinate does not change), for example a water molecule on the surface of some crystal. The ability to rotate a molecule in space is extremely important in computational chemistry, for example in drug discovery a ligand molecule may be rotated with a binding pocket of a protein molecule in order to evaulate the lowest energy interaction and offering insight into ligand design as a result. To help with visualisation of the molecule on a surface, we have provided a **module** on Moodle (`visualisation.py`) that uses the `matplotlib` library to enable visual inspection of the molecule on a surface.

Finally, by replacing elements of Equation 3 with those from Equation 1 we may obtain straight-forward transformations that lead from (x, y) to (x', y'),

$$
\begin{aligned}
x' &= x \cos\beta - y \sin\beta, \\
y' &= y \cos\beta + x \sin\beta.
\end{aligned}
\tag{4}
$$

Using the transformations outlined in Equation 4, you now must create a module named `transform` that includes a function called `rotation` that will take three variables `x`, `y`, and `angle`. This function will perform a rotation of `angle` on `x` and `y` to produce `x_new` and `y_new`, which are returned from the function. Use the `visualisation.show` function to observe the rotation of the water molecule.

- copy and edit (or start from scratch) & rewrite using **numpy** vectors and rotation matrices (hint: rewrite Eqn 4 as a matrix equation)

EXERCISE 2

$$x' = x\cos\beta - y\sin\beta$$
$$y' = x\sin\beta + y\cos\beta$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

- copy and edit (or start from scratch) & rewrite
  using `numpy` vectors and rotation matrices (hint: rewrite Eqn 4 as a matrix equation)