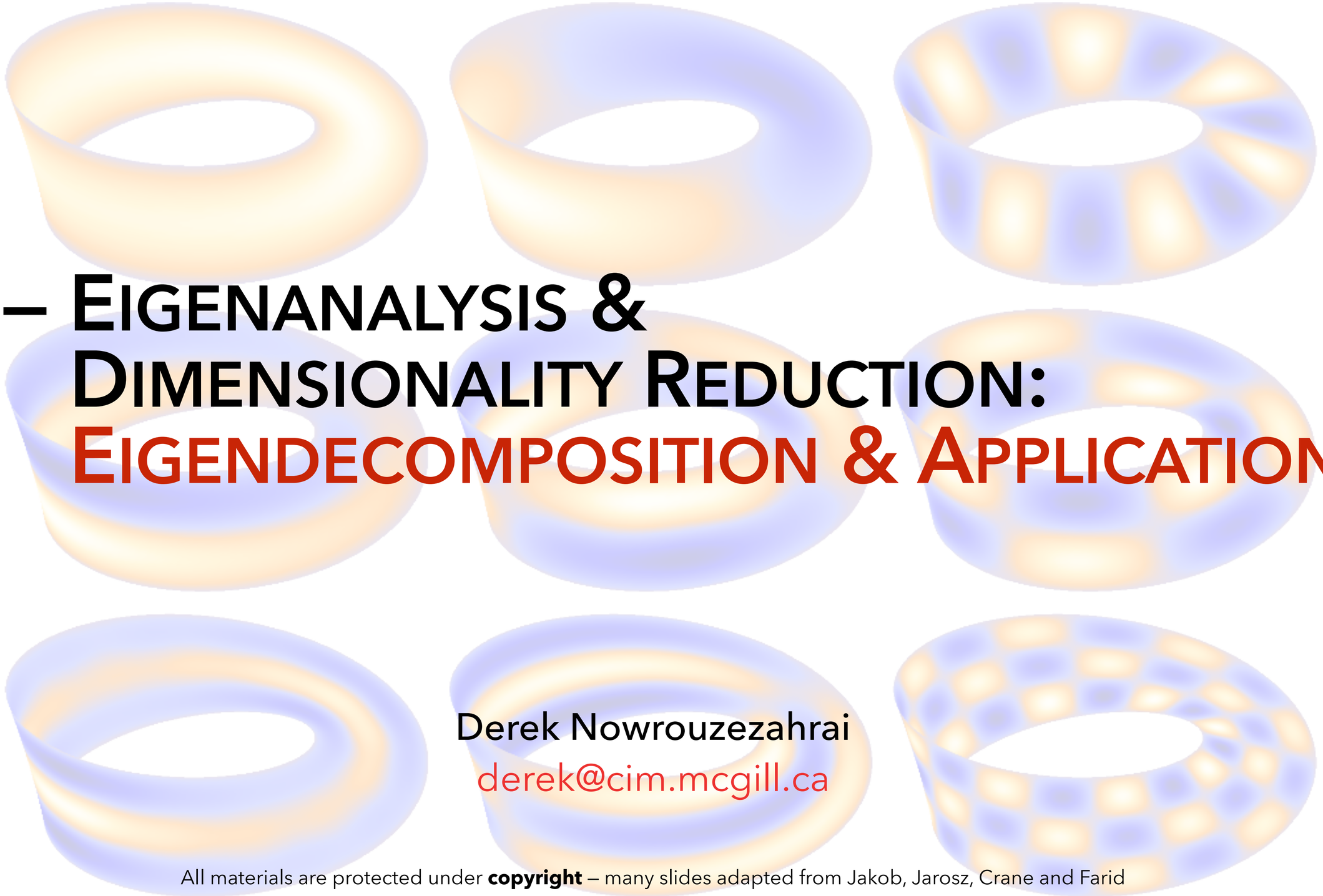


4A – Eigenanalysis & Dimensionality Reduction: Eigendecomposition & Applications

Derek Nowrouzezahrai
derek@cim.mcgill.ca

General Eigendecomposition

Recall – Eigenvectors & eigenvalues

$$\mathbf{A}\mathbf{v}_i = \lambda_i \mathbf{v}_i$$

Recall – Eigenvectors & eigenvalues

eigenvalue eigenvector

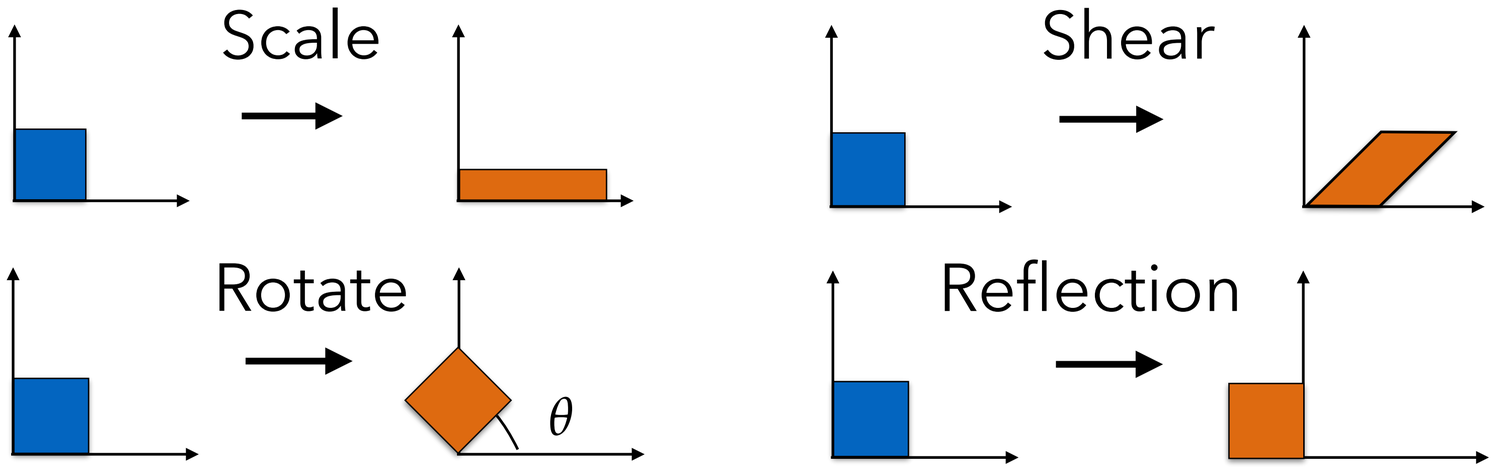
$$\mathbf{A}\mathbf{v}_i = \lambda_i \mathbf{v}_i$$

where

$$\mathbf{A} \in \mathbb{R}^{n \times n}, \ \lambda \in \mathbb{R}$$
$$\mathbf{v} \in \mathbb{R}^n, \mathbf{v} \neq 0$$

Eigenvectors – geometric interpretation

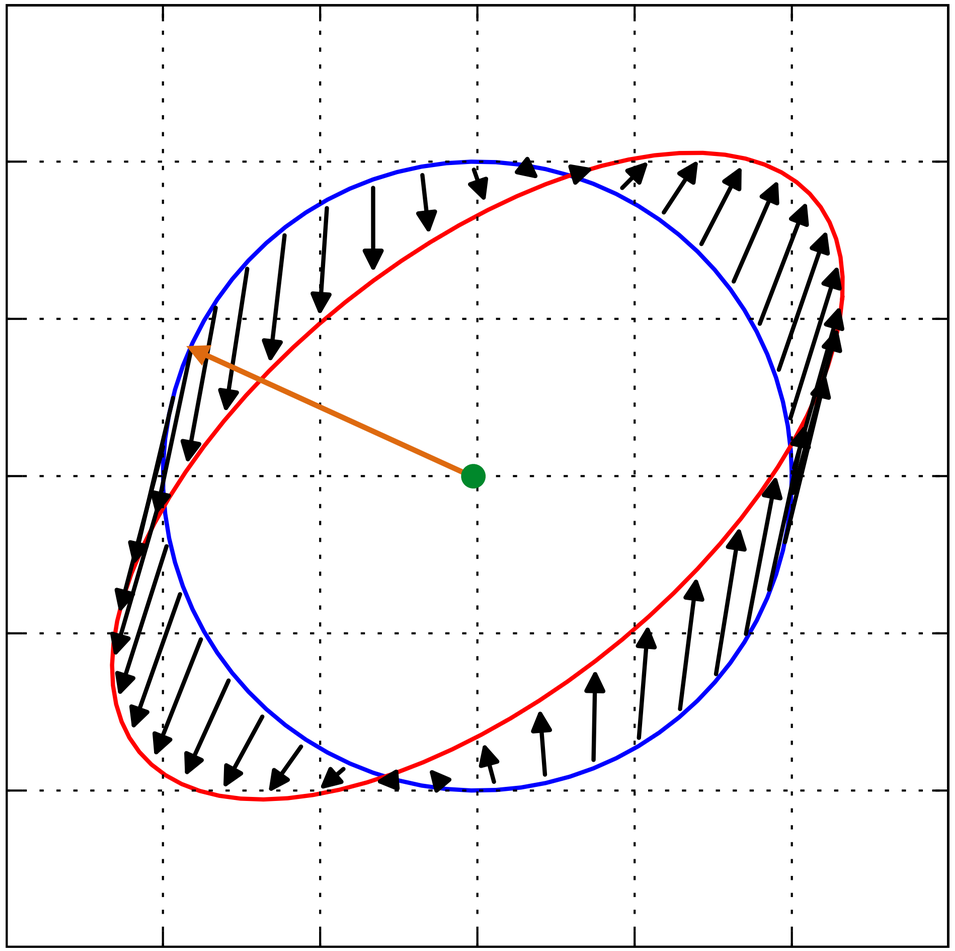
Old News: linear maps can be interpreted as geometric transformations



In general, **any combination** of these transformation, and in **any dimension**

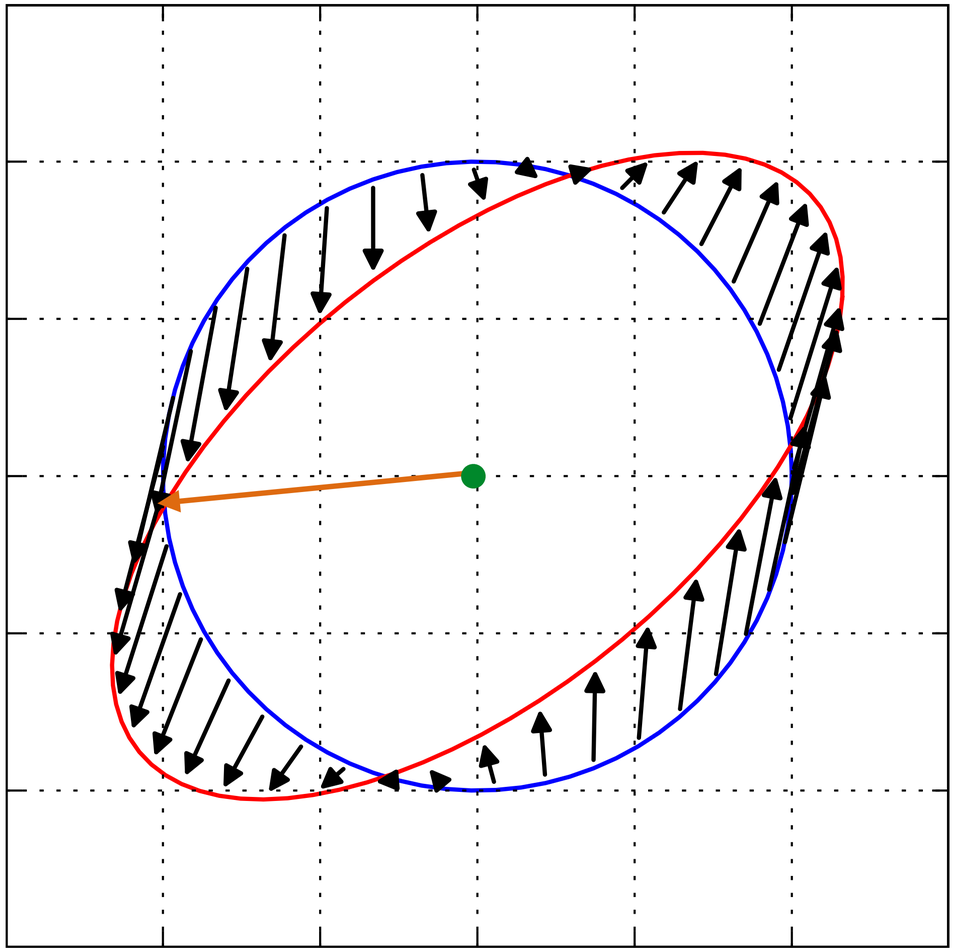
Eigenvectors – geometric interpretation

$$\mathbf{A}\mathbf{v}_i \neq \lambda_i \mathbf{v}_i$$



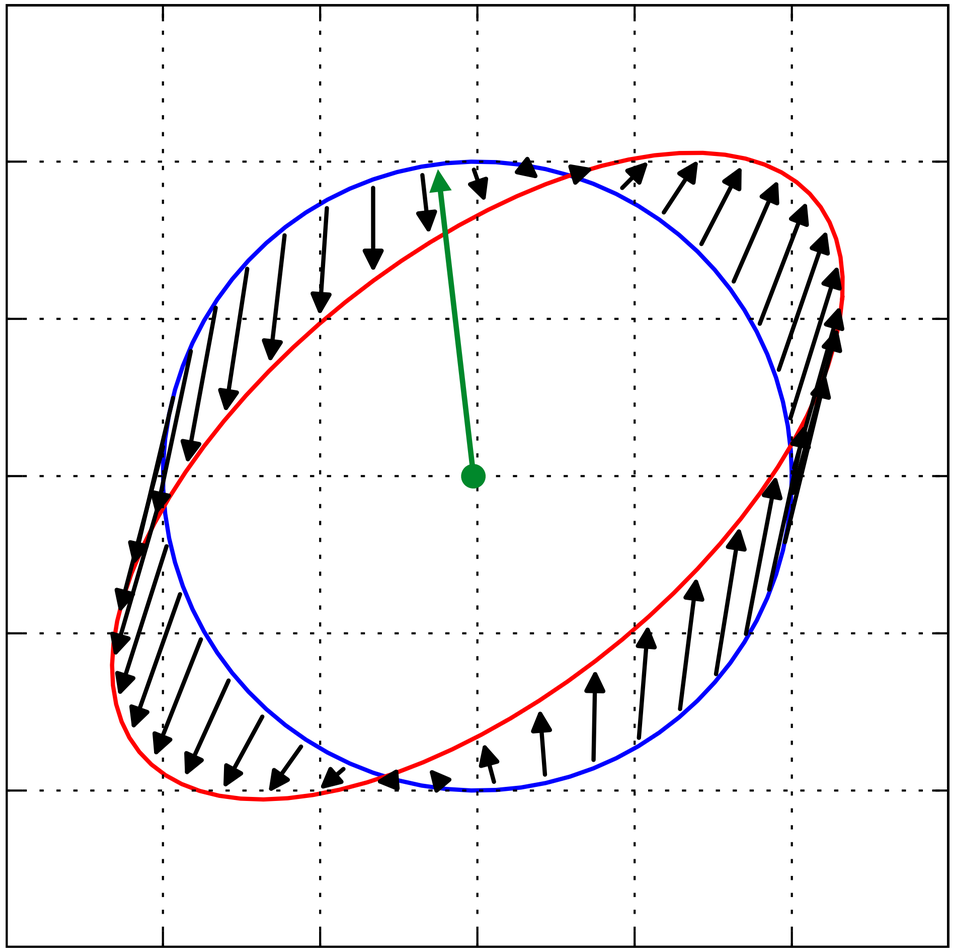
Eigenvectors – geometric interpretation

$$\mathbf{A}\mathbf{v}_i \neq \lambda_i \mathbf{v}_i$$



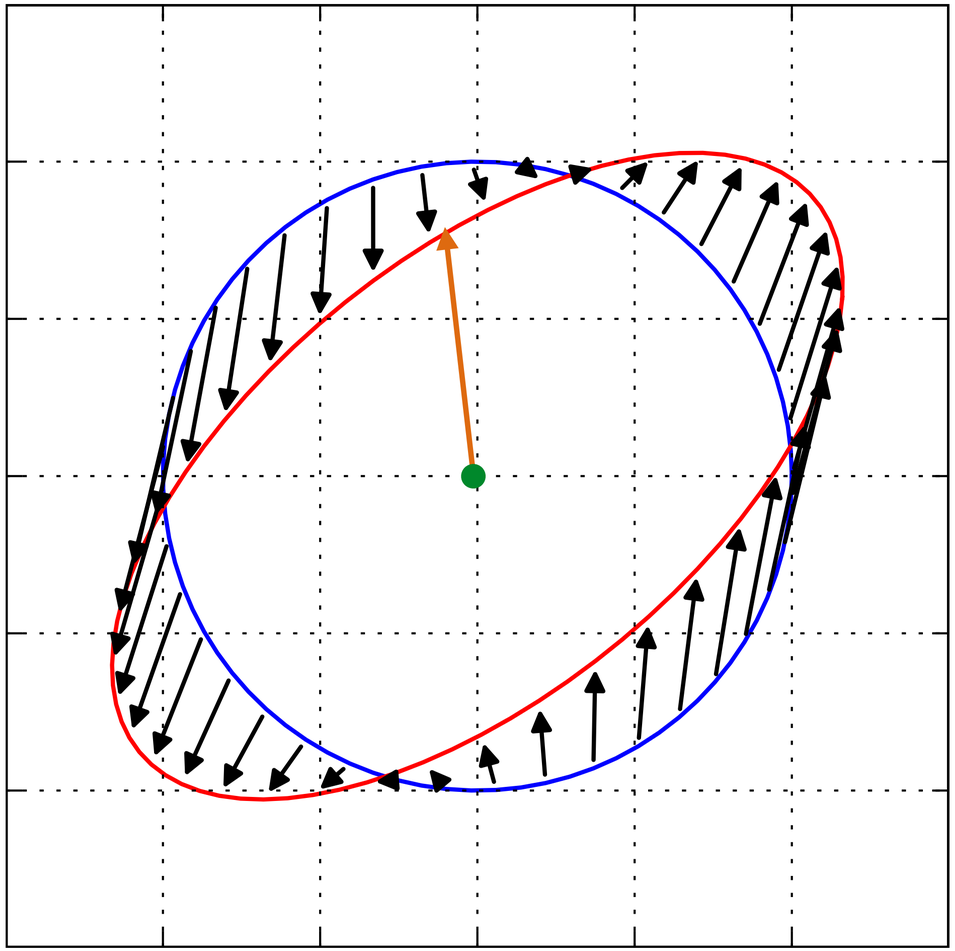
Eigenvectors – geometric interpretation

$$\mathbf{A}\mathbf{v}_1 = \lambda_1 \mathbf{v}_1$$



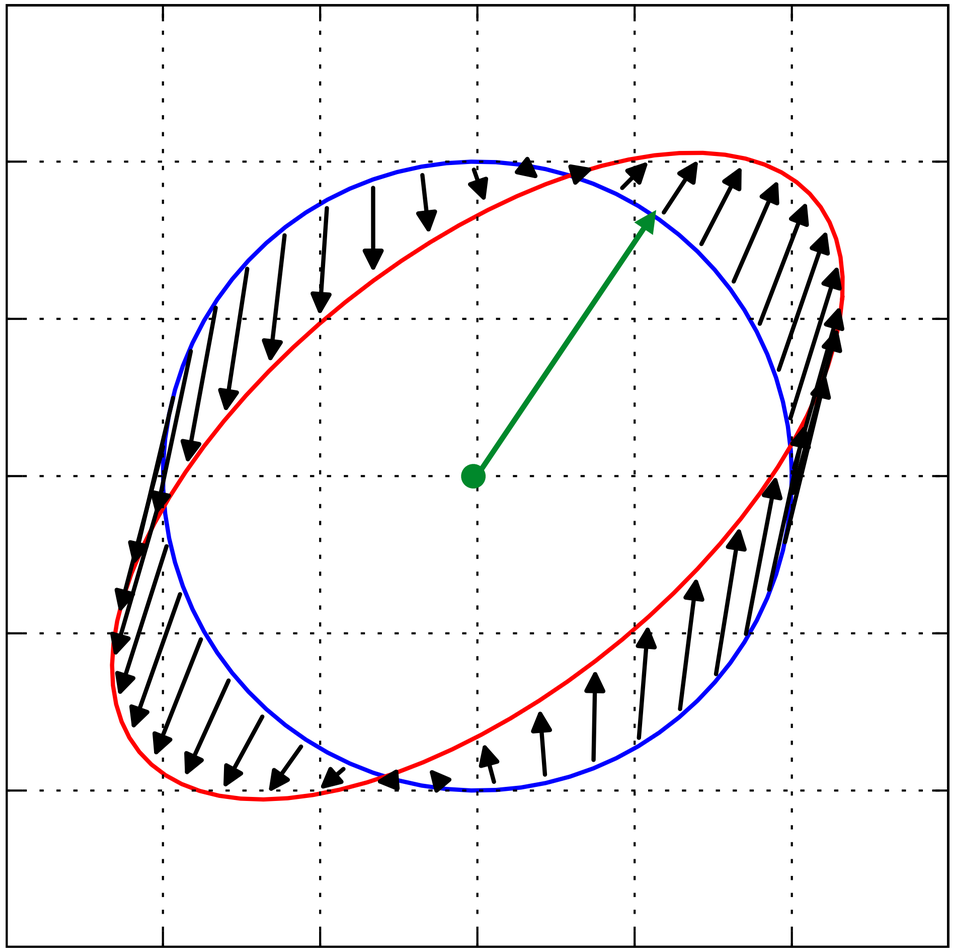
Eigenvectors – geometric interpretation

$$\mathbf{A}\mathbf{v}_1 = \lambda_1 \mathbf{v}_1$$



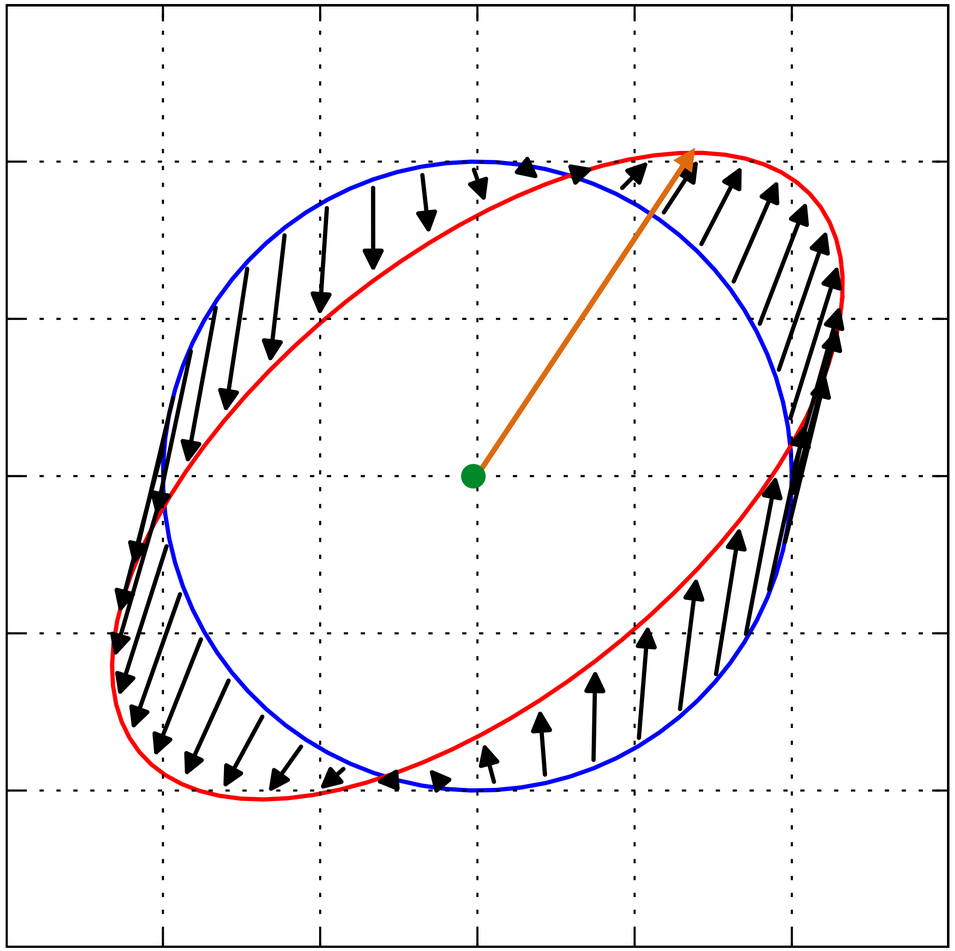
Eigenvectors – geometric interpretation

$$\mathbf{A}\mathbf{v}_2 = \lambda_2 \mathbf{v}_2$$



Eigenvectors – geometric interpretation

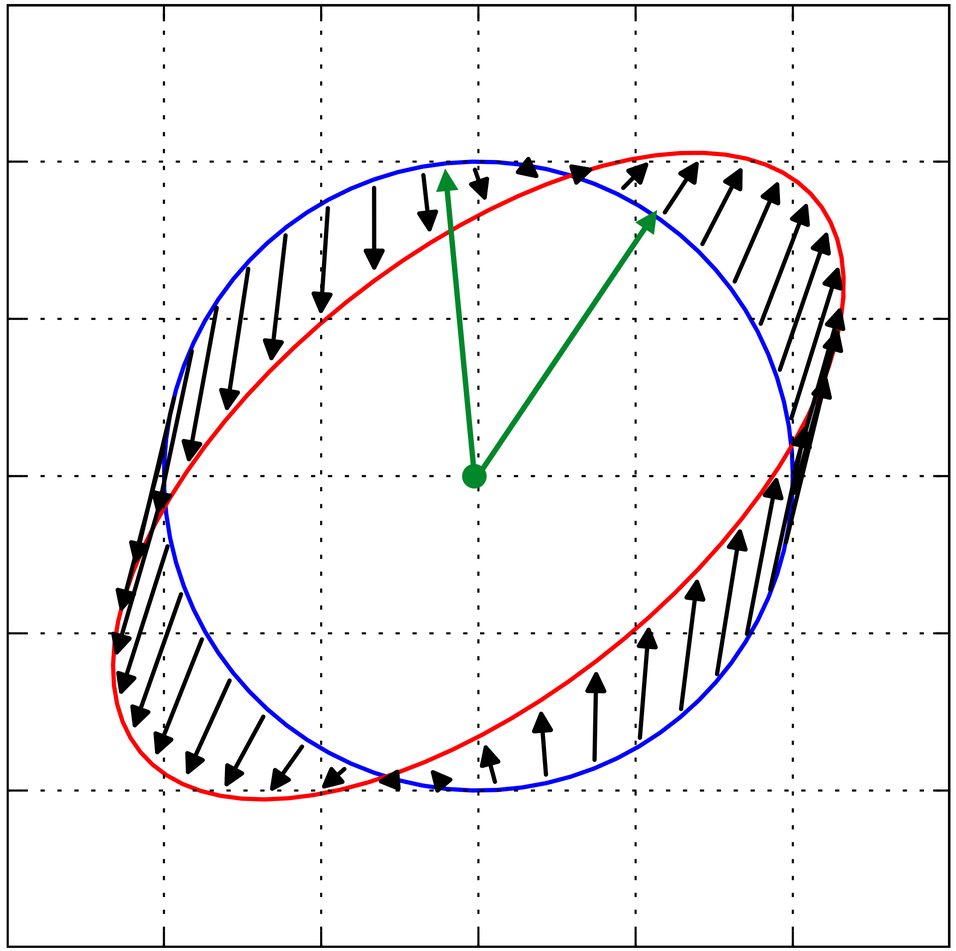
$$\mathbf{A}\mathbf{v}_2 = \lambda_2 \mathbf{v}_2$$



Eigenvectors – geometric interpretation

$$\mathbf{A}\mathbf{v}_i = \lambda_i \mathbf{v}_i$$

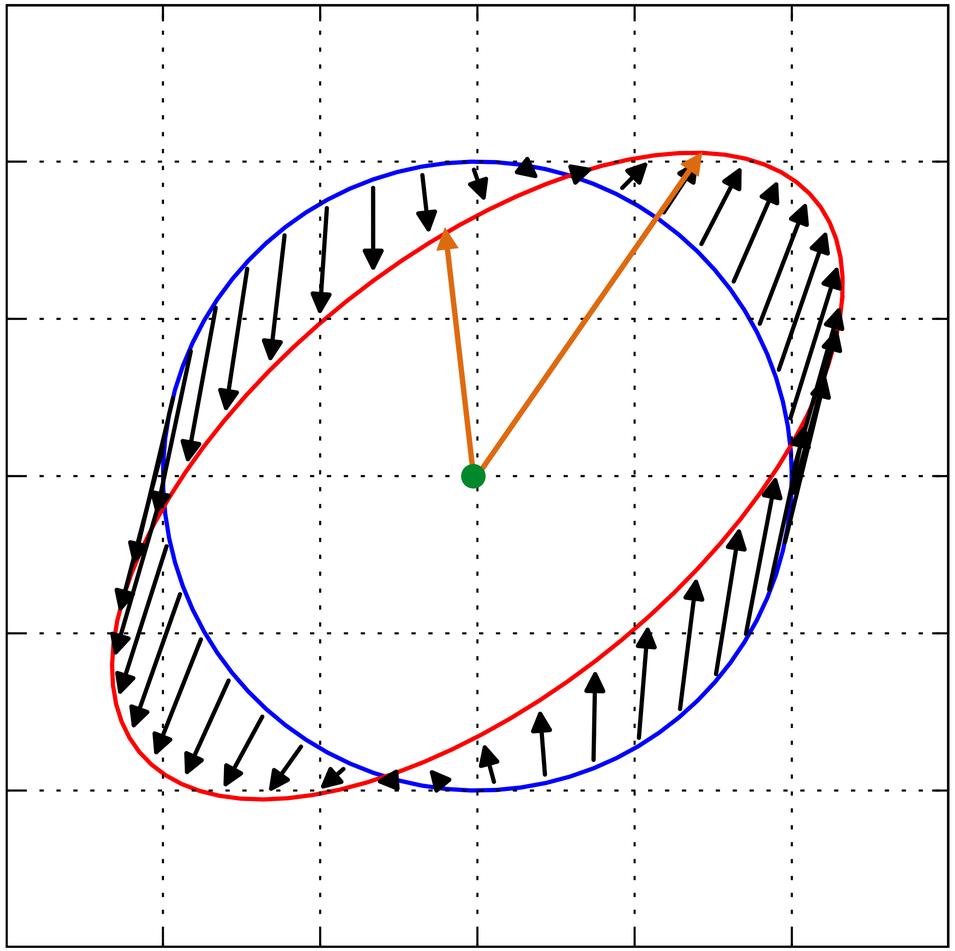
```
# some random matrix
A = array([[ 1.16043581,  0.0566787 ],
           [ 0.66722123,  0.85777919]])
λ, V  = scipy.linalg.eig(A)
```



Eigenvectors – geometric interpretation

$$\mathbf{A}\mathbf{v}_i = \lambda_i \mathbf{v}_i$$

```
# some random matrix
A = array([[ 1.16043581,  0.0566787 ],
           [ 0.66722123,  0.85777919]])
λ, V  = scipy.linalg.eig(A)
```



Eigenvectors of non-singular matrices

For non-singular matrices $\mathbf{A}$:

- there are exactly n distinct eigenvalue/eigenvector pairs
- the eigenvectors $\mathbf{v}_i$ form a basis for $\mathbb{R}^n$
  - the basis is *not necessarily* orthogonal

Therefore, a matrix with the $\mathbf{v}_i$ as its columns

$$\mathbf{V} = \begin{pmatrix} | & & | \\ \mathbf{v}_1 & \cdots & \mathbf{v}_n \\ | & & | \end{pmatrix}$$

has rank n (i.e., full rank) and so is non-singular

Eigenvectors of non-singular matrices

$$\mathbf{V} = \begin{pmatrix} | & & | \\ \mathbf{v}_1 & \cdots & \mathbf{v}_n \\ | & & | \end{pmatrix}$$

The matrix $\mathbf{V}$ has interesting geometric properties:

- noting that $\mathbf{V}^{-1}\mathbf{v}_i$ is the ith column of $\mathbf{V}^{-1}\mathbf{V} = \mathbf{I}$, we have $\mathbf{V}^{-1}\mathbf{v}_i = \mathbf{e}_i$

From this, we can show that $\mathbf{V}^{-1}\mathbf{A}\mathbf{V} = \mathrm{diag}(\lambda_1,\ldots,\lambda_n) \equiv \mathbf{\Lambda}$

- can use the $\mathbf{v}_i$ to form two change of bases that *diagonalize* $\mathbf{A}$
- the direction of $\mathbf{v}_i$ (and not its magnitude) is most important
  - it is not uncommon to normalize the $\mathbf{v}_i$ by convention

Eigendecomposition

From this diagonalization of $\mathbf{A}$ — namely $\mathbf{V}^{-1}\mathbf{A}\mathbf{V} = \mathbf{\Lambda}$ — we can derive the **eigendecomposition** of $\mathbf{A}$ by rearranging for $\mathbf{A}$ as:

$$\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$$

$$\mathbf{V} = \begin{pmatrix} | & & | \\ \mathbf{v}_1 & \cdots & \mathbf{v}_n \\ | & & | \end{pmatrix} \qquad \mathbf{\Lambda} = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}$$

matrix with eigenvectors

diagonal matrix with eigenvalues

Geometric interpretation

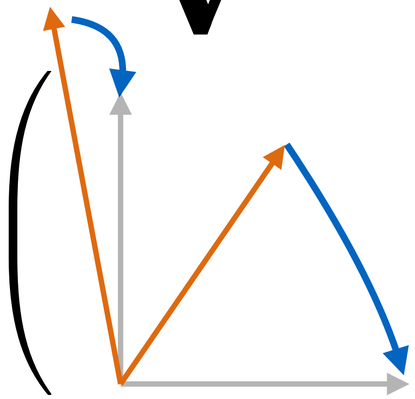
The **eigendecomposition** expresses a general linear transformation $\mathbf{A}$ as the product of three transforms:

$$\mathbf{A} =$$

Geometric interpretation

The **eigendecomposition** expresses a general linear transformation $\mathbf{A}$ as the product of three transforms:

$$\mathbf{A} = \mathbf{V}^{-1} \cdot \left(\quad \right)$$

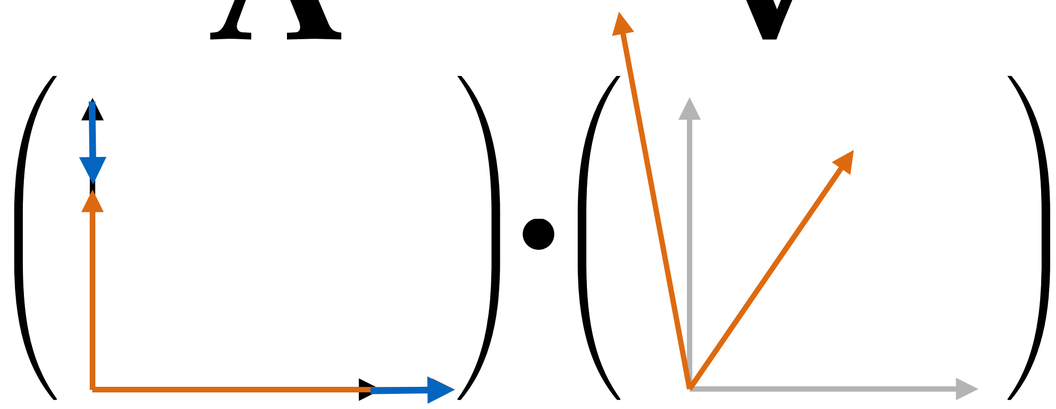


change of basis
(not necessarily
orthogonal!)

Geometric interpretation

The **eigendecomposition** expresses a general linear transformation $\mathbf{A}$ as the product of three transforms:

$$\mathbf{A} = \qquad \overset{\mathbf{\Lambda}}{\cdot \left(\quad \right)} \overset{\mathbf{V}^{-1}}{\cdot \left(\quad \right)}$$



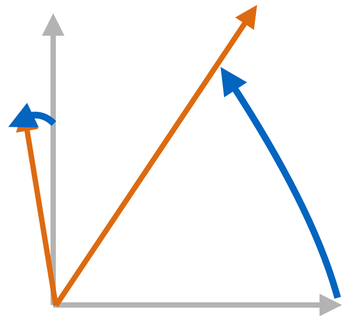
non-uniform scale

change of basis (not necessarily orthogonal!)

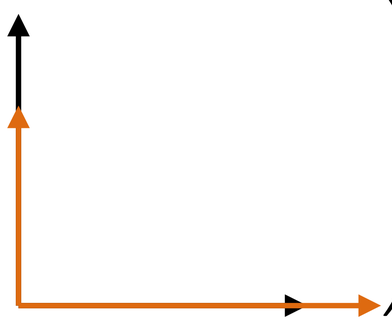
Geometric interpretation

The **eigendecomposition** expresses a general linear transformation $\mathbf{A}$ as the product of three transforms:

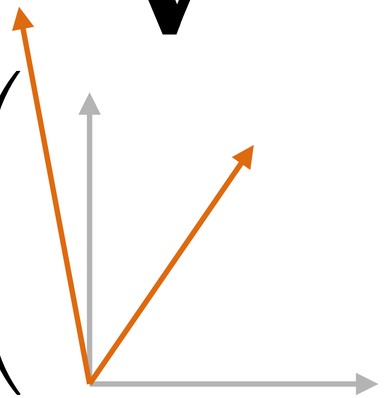
$$\mathbf{A} = \overset{\mathbf{V}}{\left(\right.}$$



$$\left.\right) \cdot \overset{\mathbf{\Lambda}}{\left(\right.}$$



$$\left.\right) \cdot \overset{\mathbf{V}^{-1}}{\left(\right.}$$



$$\left.\right)$$

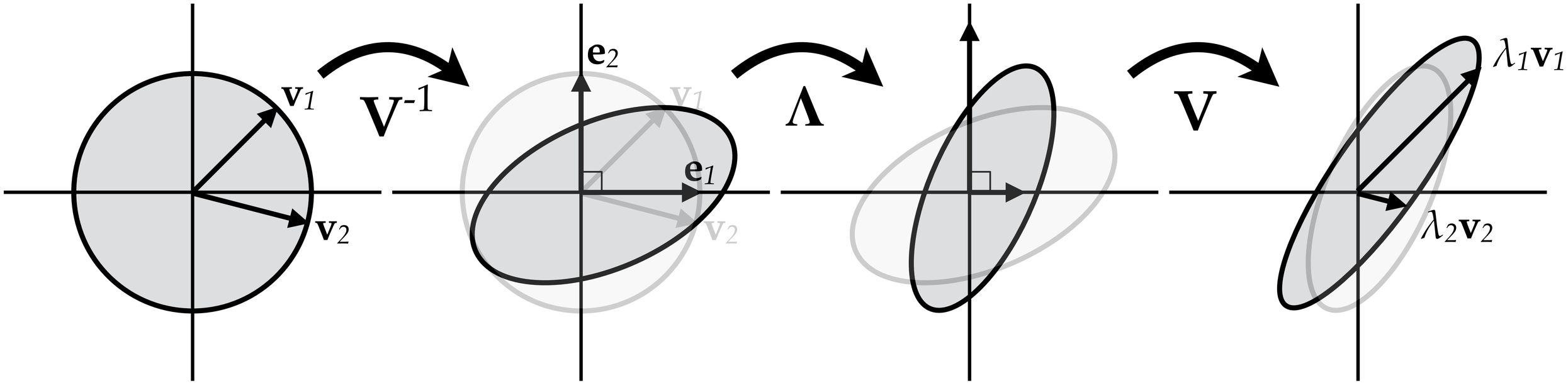
(inverse) change of basis

non-uniform scale

change of basis (not necessarily orthogonal!)

Geometric interpretation

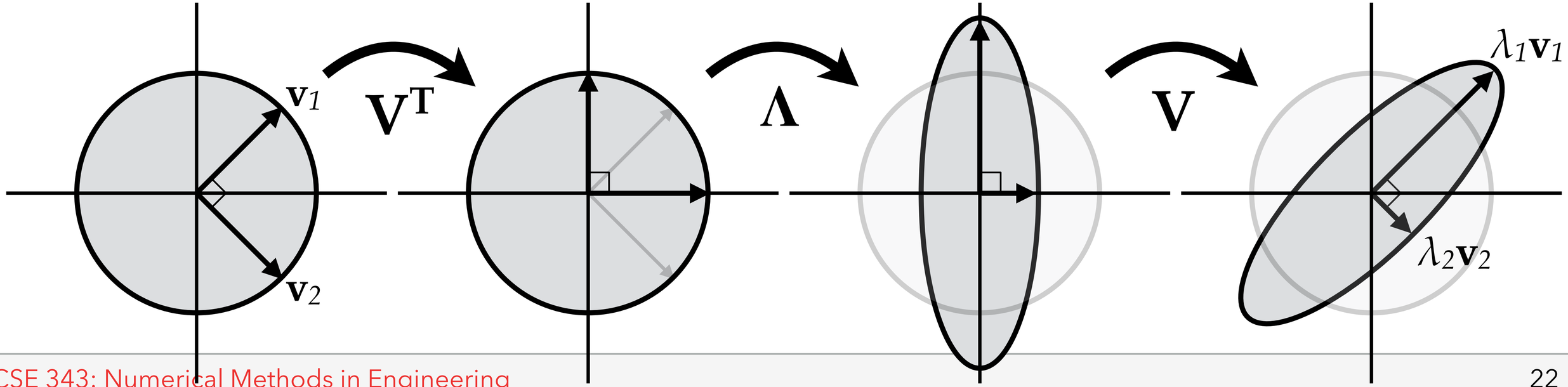
$$\mathbf{A} = \underbrace{\left(\quad\right)}_{\mathbf{V}} \cdot \underbrace{\left(\quad\right)}_{\mathbf{\Lambda}} \cdot \underbrace{\left(\quad\right)}_{\mathbf{V}^{-1}}$$



Eigendecomposition of a Symmetric $\mathbf{A}$

For symmetric $\mathbf{A}$, we can find (unit) eigenvectors – with unique eigenvalues – that are mutually orthonormal

- here, the change of basis $\mathbf{V}$ only has a **rotation component**
- so $\mathbf{V}^{-1} = \mathbf{V}^{\mathbf{T}}$, the diagonalization is $\mathbf{V}^{\mathbf{T}}\mathbf{A}\mathbf{V} = \mathbf{\Lambda}$, and the eigendecomposition is $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{\mathbf{T}}$



Summary and Further Interpretation

The eigendecomposition uses $\mathbf{V}$ (and its inverse) to isolate the non-uniform scaling effects of the transform $\mathbf{A}$

- this allows us to reason about the *repetitive application of* $\mathbf{A}$
  - $\mathbf{A} \cdot \mathbf{A} \cdot \mathbf{A} \cdot \cdots \mathbf{A}$ (repeated k times) performs a transformation with the same eigenvectors, but with scaled eigenvalues

$$\text{If } \mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1} \text{ then } \mathbf{A}^k = \mathbf{V}\mathbf{\Lambda}^k\mathbf{V}^{-1}$$

  - $\mathbf{V}$ transforms the space spanned by the eigenvectors into a coordinate system that aligns the directions that shrink/grow (as a result of the application of $\mathbf{A}$) about $\mathbf{e}_i$

Summary and Further Interpretation

The eigendecomposition of non-singular $\mathbf{A}$ has only non-zero eigenvalues

Given the eigendecomposition of a non-singular $\mathbf{A}$, we can compute the (eigendecomposition of the) inverse of $\mathbf{A}$ as

$$\text{If } \mathbf{A} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{-1} \text{ then } \mathbf{A}^{-1} = \mathbf{V}\boldsymbol{\Lambda}^{-1}\mathbf{V}^{-1}$$

We haven't yet discussed **how to determine** (i.e., **compute**) the eigendecomposition…

Computing an Eigendecomposition

Computing an Eigendecomposition

Algebraically, we can express the eigenvalues as the solutions to the following ***characteristic equation***: $\det(\mathbf{A} - \lambda \mathbf{I}) = 0$

- we can arrive at this degree-n **characteristic polynomial** $p(\lambda)$ by isolating for eigenvalues in $\mathbf{A}\mathbf{v}_i = \lambda_i \mathbf{v}_i$

The roots of a polynomial of degree > 4 cannot be expressed in closed form [Abel 1824]

- what are the pragmatic implications of this mathematical limitation?

Practical Eigendecomposition

We take advantage of the <u>matrix power property</u> of the eigendecomposition to build an *iterative*, numerical algorithm for computing eigenvalue/eigenvector pairs

$$\mathbf{A}^k = \mathbf{V}\mathbf{\Lambda}^k\mathbf{V}^{-1} \qquad \mathbf{\Lambda}^k = \begin{pmatrix} \lambda_1^k & & \\ & \ddots & \\ & & \lambda_n^k \end{pmatrix}$$

- as $k \rightarrow \infty$, the value of the largest eigenvalue λ_1^k dominates over that of the remaining eigenvalues

Power method

We can show that by repetitively applying $\mathbf{A}$ to any vector $\mathbf{x}$, the resulting product will converge to the eigenvector associated with the largest eigenvalue:

$$\mathbf{A}^k\mathbf{x} = \underbrace{(\mathbf{A}\cdot\ldots\cdot\mathbf{A})}_{k\gg 1 \text{ times}}\mathbf{x} \approx \mathbf{V}\widehat{\mathbf{\Lambda}}\mathbf{V}^{-1}\mathbf{x} \qquad \widehat{\mathbf{\Lambda}} = \begin{pmatrix} \lambda_1^k & & & \\ & 0 & & \\ & & \ddots & \\ & & & 0 \end{pmatrix}$$

A naive implementation of this *iterative* algorithm is:

```
x = np.random.random(A.shape[0])
for i in range(num_iterations):
    x = A @ x
```

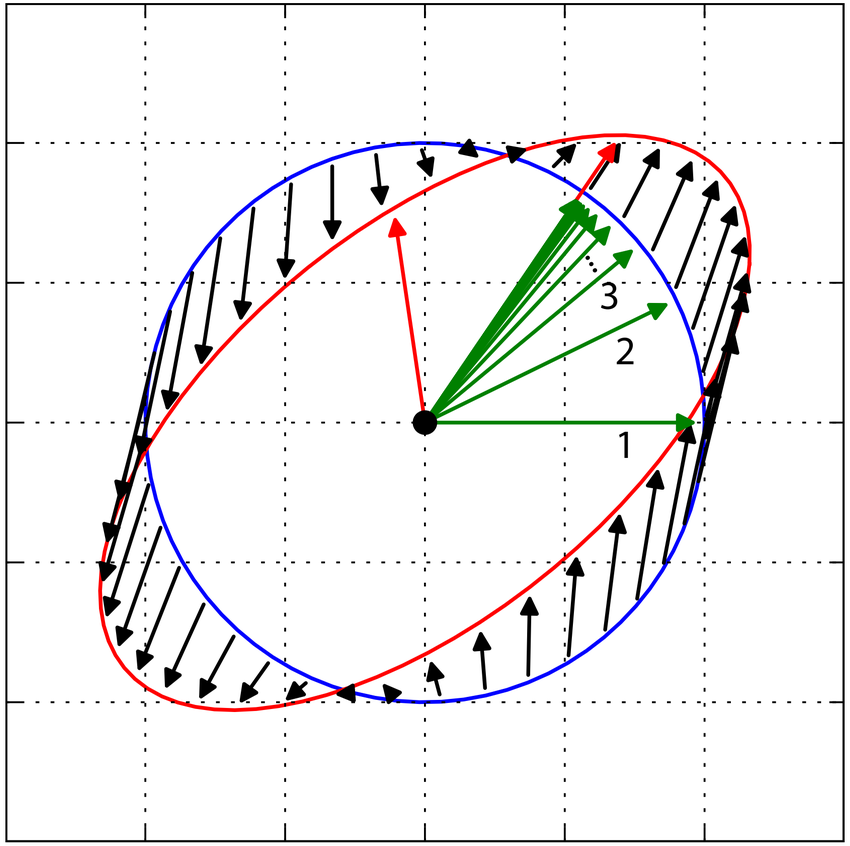
Power method

$$\mathbf{A}^k\mathbf{x} = \underbrace{(\mathbf{A}\cdot\ldots\cdot\mathbf{A})}_{k\gg 1 \text{ times}}\mathbf{x} \approx \mathbf{V}\widehat{\mathbf{\Lambda}}\mathbf{V}^{-1}\mathbf{x} \qquad \widehat{\mathbf{\Lambda}} = \mathrm{diag}(\lambda_1^k, 0, \ldots, 0)$$

A more robust algorithm computes a sequence of normalized vectors $\frac{\mathbf{A}\mathbf{x}}{\|\mathbf{A}\mathbf{x}\|}, \frac{\mathbf{A}^2\mathbf{x}}{\|\mathbf{A}^2\mathbf{x}\|}, \frac{\mathbf{A}^3\mathbf{x}}{\|\mathbf{A}^3\mathbf{x}\|}, \ldots$ which converges to $\mathbf{v}_1$

```
x = np.random.random(A.shape[0])
for i in range(num_iterations):
    x = A @ x
    x /= np.linalg.norm(x)
```

Power method – iteration



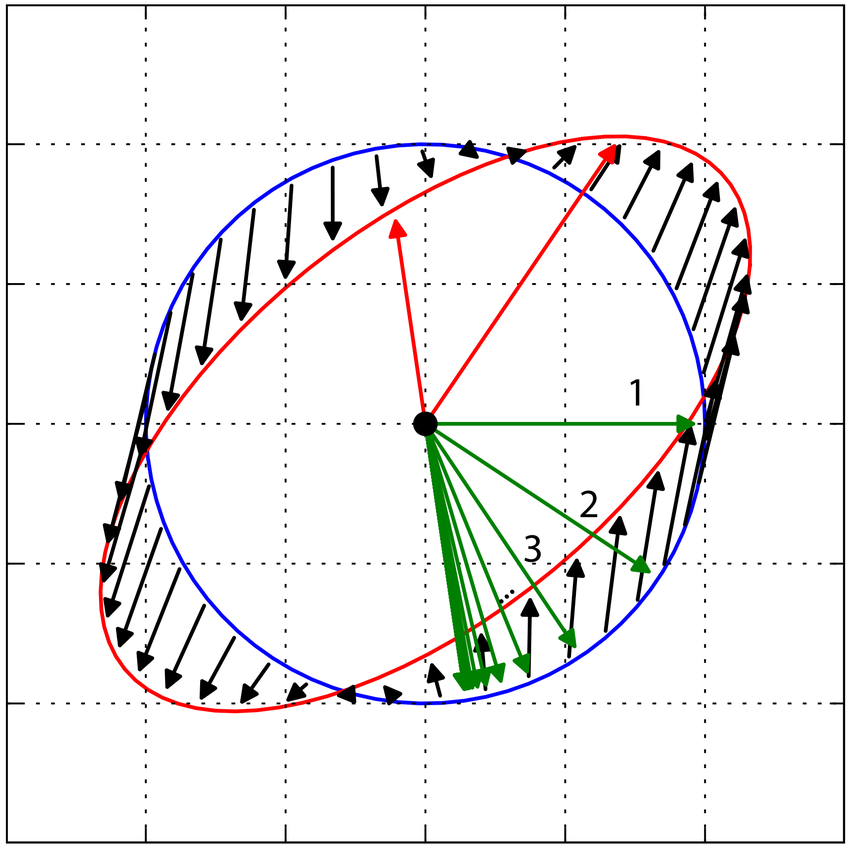
Power method

We can similarly obtain an estimate of the eigenvector associated to the **smallest** eigenvalue by computing the sequence $\frac{\mathbf{A}^{-1}\mathbf{x}}{|\mathbf{A}^{-1}\mathbf{x}\|}, \frac{\mathbf{A}^{-2}\mathbf{x}}{\|\mathbf{A}^{-2}\mathbf{x}\|}, \frac{\mathbf{A}^{-3}\mathbf{x}}{\|\mathbf{A}^{-3}\mathbf{x}\|}$, which converges to $\mathbf{v}_n$ since

$$\mathbf{A}^{-1} = \mathbf{V}\mathbf{\Lambda}^{-1}\mathbf{V}^{-1}$$

and as before, as $k \to \infty$, now the value of the **inverse of the smallest eigenvalue** $1/\lambda_n^k$ dominates over that of the remaining eigenvalues (of the inverse matrix)

Power method – inverse iteration



Eigendecomposition pragmatics…

The power method will only converge if your initial guess is not (numerically) orthogonal to the eigenvector

- in practice, you may run the iteration many times to sanity check
- it only converges if a single, dominant eigenvalue exists
- if eigenvalues are not well spaced, iterations may be unstable

Power method (and the inverse method) only approximate the eigenvectors associated to the largest and smallest eigenvalues

- getting other eigenvectors requires more advanced techniques

More gotchas….

Eigendecomposition pragmatics…

The details of more advanced eigendecomposition algorithms are beyond the scope of this course, but a few things to keep in mind if you pursue this avenue in the future:

- we've only discussed eigenvalue-eigenvector **pairs** $(\mathbf{v}_i, \lambda_i)$ but things are more complicated: the pair is really a **triple** $(\mathbf{v}_i, \mathbf{w}_i, \lambda_i)$
  - where not only do we have the usual $\mathbf{A}\mathbf{v}_i = \lambda_i \mathbf{v}_i$ – where the $\mathbf{v}_i$ are referred tp more generally as *right eigenvectors* – but we also have $\mathbf{w}_i^T \mathbf{A} = \lambda_i \mathbf{w}_i^T$ – where the *left eigenvectors* $\mathbf{w}_i$ must also satisfy an additional constraint w.r.t. the eigenvalue λ_i

Eigenanalysis Summary

Generally speaking, you can rely on existing algorithms for eigendecomposition

- unless you go into research in this area, or
- if you only require the eigenvector associated to largest/smallest unique eigenvalue, if it exists

Just as important to understanding what an eigendecomposition represents is getting a sense of the many places it can be applied

- we will discuss several applications of eigendecomposition in, e.g., overdetermined model fitting, dimensionality reduction, and (very briefly) solving systems of differential equations