Eigen problems:
Calculating Eigen

- we knew ~~how~~ what eigenvectors are and how to calculate it.
- we can combine this idea with concept of changing basis (covered earlier)

→ what emerges is powerful tool. for performing efficient matrix operations, called diagonalization.

Sometimes we need to apply the same matrix operation MANY TIMES

Eg. Transformation matrix T, that represents the change in location of particle after a single time stamp

$$t = \begin{bmatrix} 0.9 & 0.8 \\ -1 & 0.35 \end{bmatrix} \qquad V_0 = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}$$

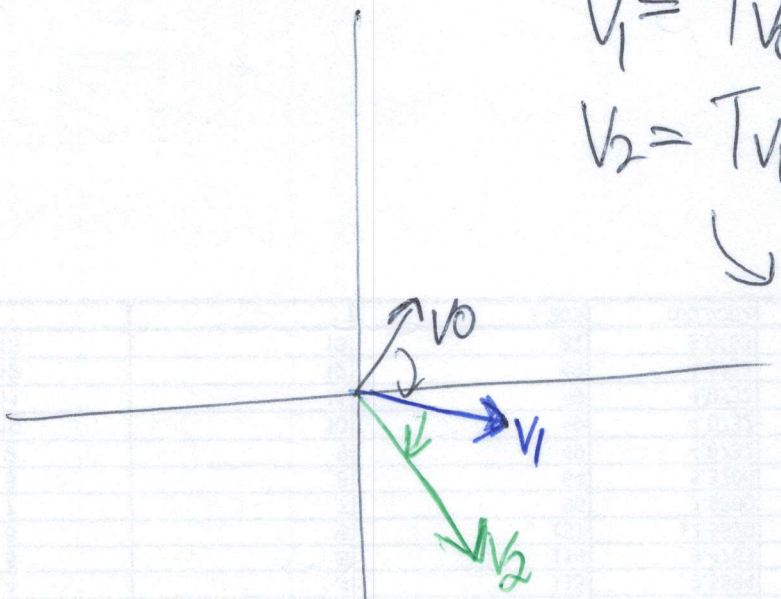
Initial position V_0, multiplied by transformation T gives us our new location

$$V_1 = TV_0$$

$$V_2 = TV_1$$

↳ same as:

$$V_2 = T(TV_0)$$



- to work out where our particle will be after 2 timesteps we can find V_2 by simply multiplying V_1 by T.
- which is the same thing as multiplying

V_0 by T 2 times

$$\therefore V_2 = TV_1$$

$$\therefore V_2 = T(TV_0)$$

$$\therefore V_2 = T^2 V_0$$

(T squared times V_0)

Imagine we want that we expect the same linear transformation operation to occur every time for n timesteps.

$$\therefore V_2 = T^2 V_0$$

$$\therefore V_n = T^n V_0$$

- We've seen the amount of work it takes to apply a single transformation 3D matrix multiplication.

So if we were to imagine that T tells us what happens in 1 second, and we want to know where our particle is in week's time, then n will be 1.2 million

i.e we will need to multiply T by itself 1 million times, which will take a while. ------

But if all the terms (except those in leading diagonal, we refer to it as a diagonal matrix.

⇒ and when raising matrices to powers, diagonal matrices make things a lot easier.

∴ all you need to do is put all terms in the diagonal to power of n., and then we'll get the answer.

$$\therefore T^n = \begin{bmatrix} a^n & 0 & 0 \\ 0 & b^n & 0 \\ 0 & 0 & c^n \end{bmatrix} \Rightarrow \text{Simple....}$$

But what if T is not a diagonal matrix.
(answer will come from Eigen analysis)

To solve this, we'll change to a basis, where our transformation T, becomes diagonal.
⇒ which is called an Eigenbasis.

We can then easily apply our power of n to the diagonalize form, and finally transforming the resultant matrix back again

Gunga T to power of n, but avoiding much of work.

Remember when changing Basis, each column of our transformed matrix supply

represents the new location of the transformed unit vectors

To build our Eigen Basis conversion matrix, we just plug in each of our eigenvectors as columns

C = Eigenvector 1, 2, 3, we using a 3Dim Example

$$\therefore C = \begin{bmatrix} x_1 & x_2 & x_3 \\ | & | & | \\ | & | & | \end{bmatrix} \quad D = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}$$

(So now multiplying by T is just a pure Scale simply meaning, it can now be represented by a diagonal matrix.

⇒ Crucially the diagonal matrix D contains the corresponding Eigenvalues of matrix T.

So close to unleashing power of EIGEN .... Final Link to see ↳ the following:

① Applying the transformation T, ↳ just the same as converting to our Eigen Basis,

② applying the diagonalizing matrix

③ and then converting Back again

$$\therefore T = C\Delta C^{-1}$$

$$T^2 = C\Delta C^{-1} C\Delta C^{-1}$$

[But multiply a matrix, then multiplying it with its inverse, is the same as doing nothing at all]

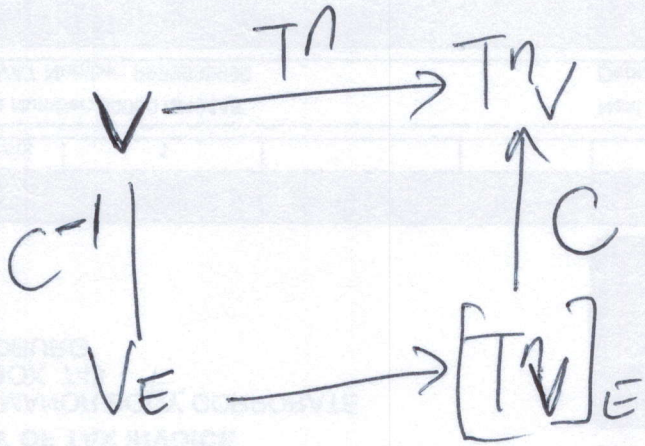
$\Rightarrow$ So we can just remove this operation.

$$= C\Delta\Delta C^{-1}$$

$$= C\Delta^2 C^{-1}$$

we can then generalize this to every power of T.

$$\therefore T^n = C\Delta^n C^{-1}$$



$\therefore$ we have method that allows us to apply a transformation matrix as many times as we like, without paying a large computational cost