向量与矩阵

行数和列数相同的矩阵，称为方阵

方阵的对角线元素是矩阵里行号和列号相同的元素

矩阵基础

1.认识矩阵

$$\begin{bmatrix} 4 & 0 & 12 \\ -5 & \sqrt{4} & 3 \\ 12 & -4/3 & -1 \\ 1/2 & 18 & 0 \end{bmatrix}$$

2.矩阵维度和记法

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}$$

M_{ij} 表示M的第i行，第j列元素。

3 方阵

行数和列数相同的矩阵，称为方阵
我们的课程中，主要讨论的范畴就是在 2*2、3*3、4*4方阵

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}$$

方阵的对角线元素就是方阵的行号和列号相同的元素；例如 3*3矩阵M的对角线元素为 m_{11}、m_{22}、m_{33}。其他元素都是非对角元素。

单位矩阵

矩阵基础

4 单位矩阵

单位矩阵，是一种特殊的对角矩阵，n维单位矩阵记做 I_n。是n * n 矩阵。对象元素为1.其他元素为0。
例如 3 * 3 单位矩阵

$$I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

单位矩阵非常特殊，因为它是矩阵乘法单位元，其基本性质是用任意1个矩阵乘以单位矩阵，都将得到原矩阵。所以在某种意义上对矩阵的作用就犹如1对于标量的作用。

向量是作为矩阵的一个特例，是多行一列或者是一列多行

矩阵转置，就是把行矩阵和列矩阵进行互换，转置的手法可以理解成沿着矩阵对角线翻折。

矩阵基础

5 向量作为矩阵使用

行向量 $\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$　　列向量 $\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$

6 矩阵转置

一个r * c 矩阵M。M的转置记做M^T，是一个 c * r 矩阵。它的列由M的行组成。可以从另方面理解。
$M_{ij}^T = M_{ji}$,即沿着矩阵的对角线翻折。

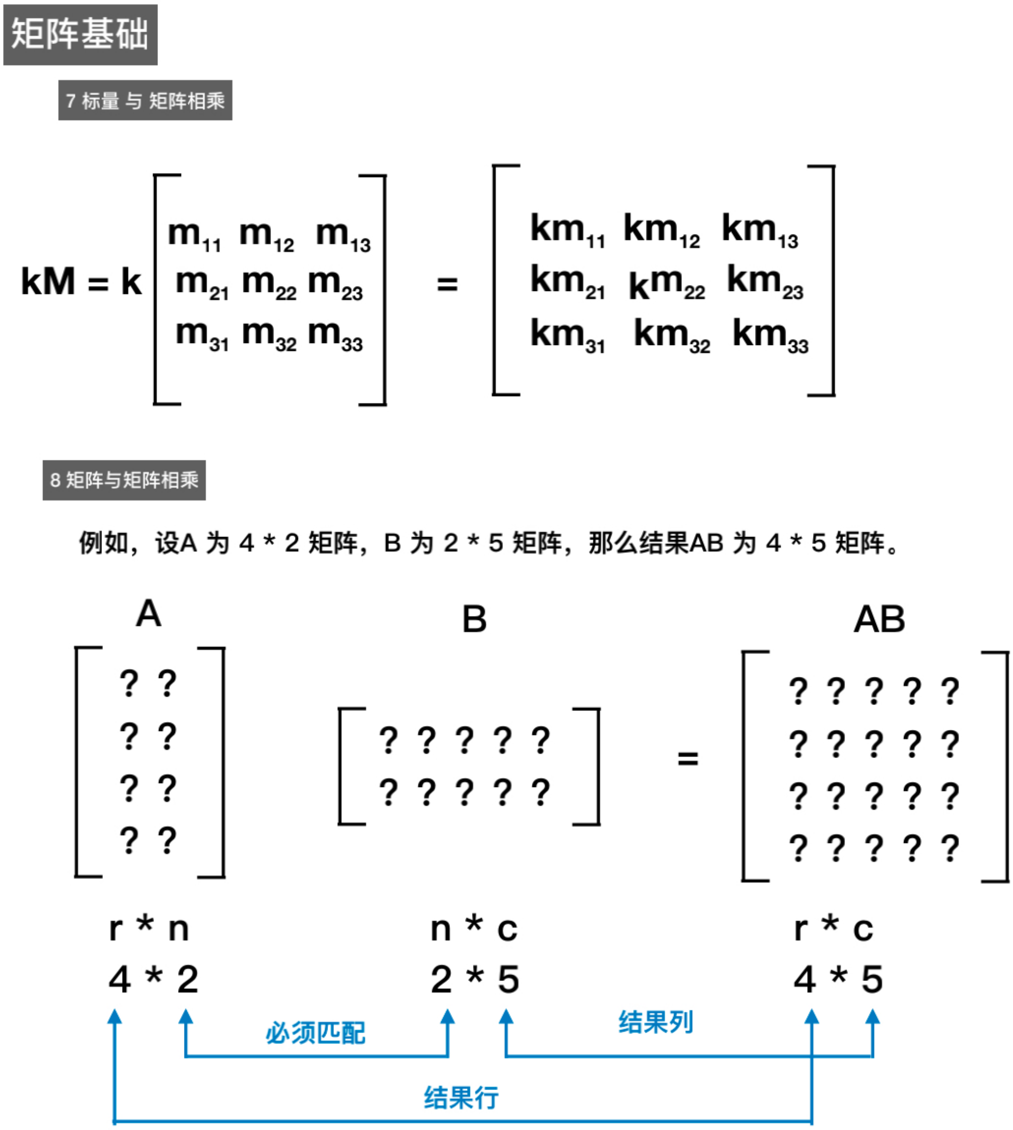
$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{bmatrix}^T = \begin{bmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{bmatrix} \qquad \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}^T = \begin{bmatrix} a & d & g \\ b & e & h \\ c & f & i \end{bmatrix}$$

对向量而言，转置将使得行向量变成列向量，是列向量变成行向量。

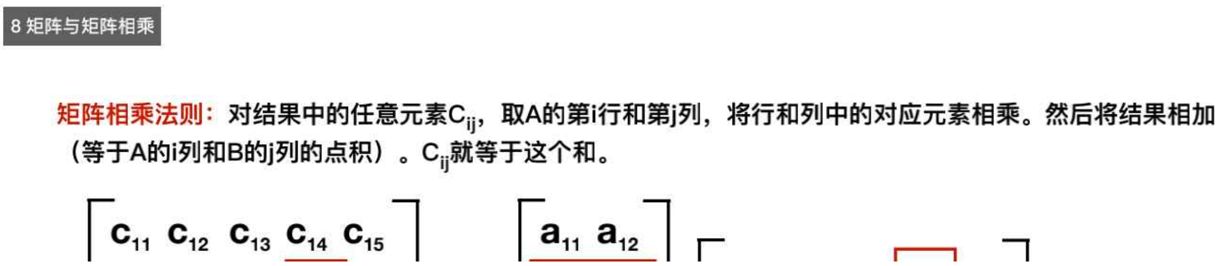
$$\begin{bmatrix} x & y & z \end{bmatrix}^T = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \qquad \begin{bmatrix} x \\ y \\ z \end{bmatrix}^T = \begin{bmatrix} x & y & z \end{bmatrix}$$

标量与矩阵相乘，就是标量与矩阵里的每一个元素相乘，这点跟变量和向量相乘一样。

矩阵与矩阵相乘，要求：矩阵1的列数=矩阵2的行数；结果：矩阵1的行数+矩阵2的列数



矩阵与矩阵相乘计算过程，矩阵相乘结果中的任意元素C(ij)，是来自矩阵1的i行与矩阵2的j列做点乘结果（类似向量间的点乘）



$$\begin{bmatrix} c_{21} & c_{22} & c_{23} & \boxed{c_{24}} & c_{25} \\ c_{31} & c_{32} & c_{33} & c_{34} & c_{35} \\ c_{41} & c_{42} & c_{43} & c_{44} & c_{45} \end{bmatrix} = \begin{bmatrix} \boxed{a_{21} \; a_{22}} \\ a_{31} \; a_{32} \\ a_{41} \; a_{42} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & b_{13} & \boxed{b_{14}} & b_{15} \\ b_{21} & b_{22} & b_{23} & \boxed{b_{24}} & b_{25} \end{bmatrix}$$

例如

$c_{24} = a_{21}b_{14} + a_{22}b_{24}$ （C的第2行第4列的元素等于A的第2行和B的第4列的点积）

当然还有另一种助记方法:

$$\begin{array}{cc} & \begin{bmatrix} b_{11} & b_{12} & b_{13} & \boxed{b_{14}} & b_{15} \\ b_{21} & b_{22} & b_{23} & \boxed{b_{24}} & b_{25} \end{bmatrix} \\ \begin{bmatrix} a_{11} & a_{12} \\ \boxed{a_{21} \; a_{22}} \\ a_{31} & a_{32} \\ a_{41} & a_{42} \end{bmatrix} & \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} & c_{15} \\ c_{21} & c_{22} & c_{23} & \boxed{c_{24}} & c_{25} \\ c_{31} & c_{32} & c_{33} & c_{34} & c_{35} \\ c_{41} & c_{42} & c_{43} & c_{44} & c_{45} \end{bmatrix} \end{array}$$

例如

$$c_{24} = a_{21}b_{14} + a_{22}b_{24}$$

矩阵相乘注意事项

1、不满足交换律，即 AB != BA

2、满足结合律，即 (AB)C = A(BC)

3、矩阵相乘的转置等于矩阵先转置再做相乘，即 (AB)T = A(T)B(T)

2 * 2 矩阵相乘完整公式

$$AB = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

$$= \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{bmatrix}$$

矩阵乘法注意事项:

1.任意矩阵M乘以方阵S,不管从哪边乘，都得到与原矩阵大小相同的矩阵。当然，前提是假定乘法有意义。如果S是单位矩阵，结果就是原矩阵M，即:MI = IM = M

矩阵，结果就是原矩阵M，即:MI = IM = M 。
2.矩阵乘法不满足交换律，即:AB != BA
3.矩阵乘法满足结合律，即:(AB)C = A(BC)。假定ABC的维数使得其乘法有意义，要注意如果(AB)C有意义，那么A(BC)就一定有意义。
4.矩阵乘法也满足与标量或向量的结合律，即:(kA)B = k(AB) = A(kB); (vA)B = v(AB);
5.矩阵积的转置相当于先转置矩阵然后以相反的顺序乘法，即：$(AB)^T = B^T A^T$

判断向量与矩阵相乘是否有意义，结果又是什么?

9 向量与矩阵的乘法

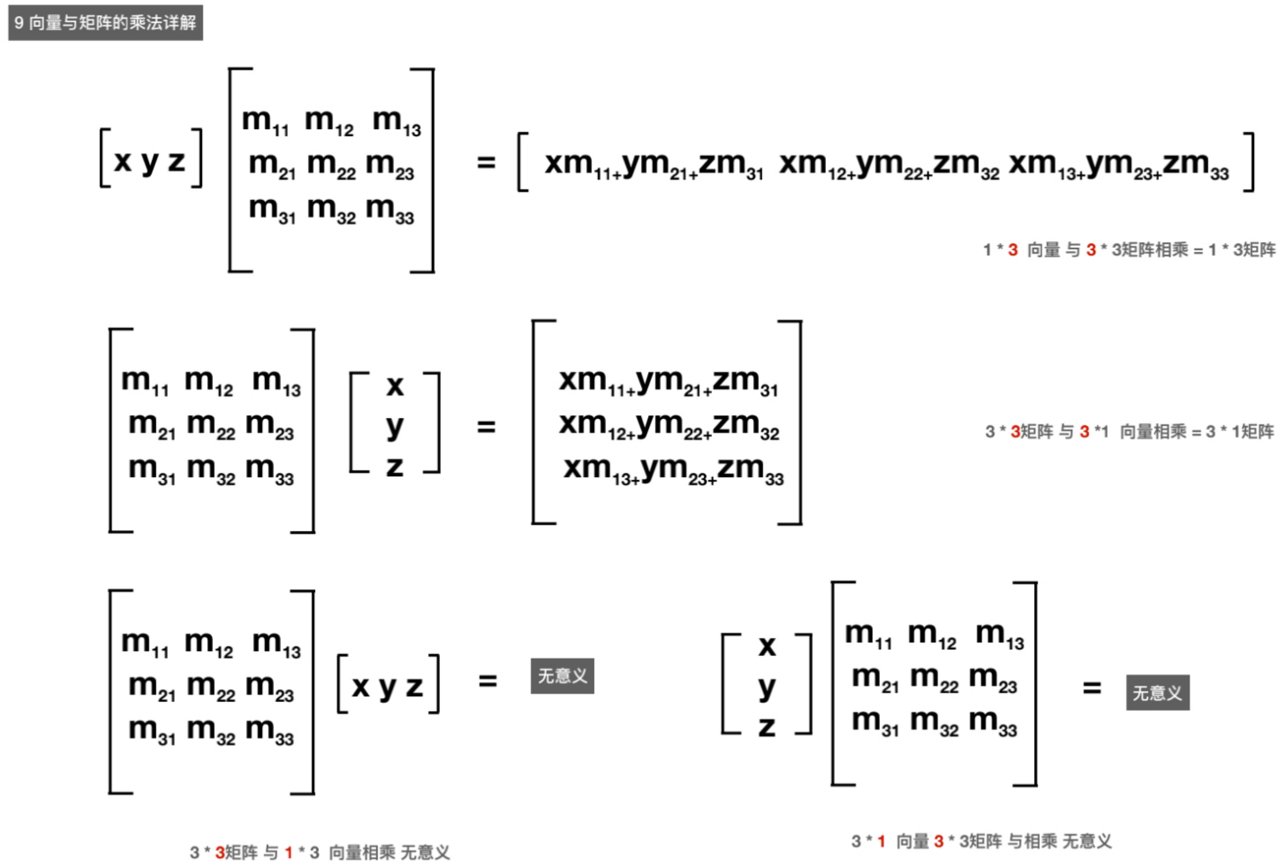
思考 向量与矩阵相乘结果是多少？是否具有意义？

$$\begin{bmatrix} x & y & z \end{bmatrix} \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} = ?$$

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \begin{bmatrix} x & y & z \end{bmatrix} = ?$$

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = ?$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} = ?$$



行向量左乘矩阵时，结果是行向量；

列向量右乘矩阵时，结果是列向量；

行向量右乘矩阵时，结果是无意义；

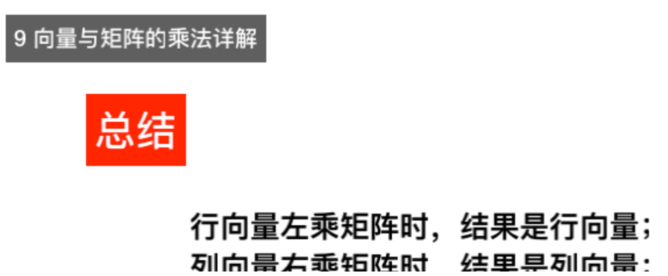
列向量左乘矩阵时，结果是无意义；

矩阵与向量相乘 注意事项：

1.结果向量中的每个元素都是原向量与矩阵中单独行或列的点积；

2.矩阵一向量乘法满足对向量加法的分配律，对于向量v,w 和 矩阵M 有，

(v + w)M = vM + wM;



列向量右乘矩阵时，结果是列向量；
行向量右乘矩阵时，结果是无意义；
列向量左乘矩阵时，结果是无意义；

矩阵与向量相乘 注意事项：
1.结果向量中的每个元素都是原向量与矩阵中单独行或列的点积；
2.矩阵一向量乘法满足对向量加法的分配律，对于向量v,w 和 矩阵M 有，
(v + w)M = vM + wM;

行向量和列向量使用场景

OpenGL 内部使用的是列向量，但是我们书写的时候是行向量

10 行向量与列向量的使用场景

为什么要使用行向量？（偏向于书写方便）
1.在文字中使用行向量的形式更加好书写；
2.用矩阵乘法实现坐标系转换时，向量左乘矩阵的形式更加方便
3.DirectX使用的是行向量

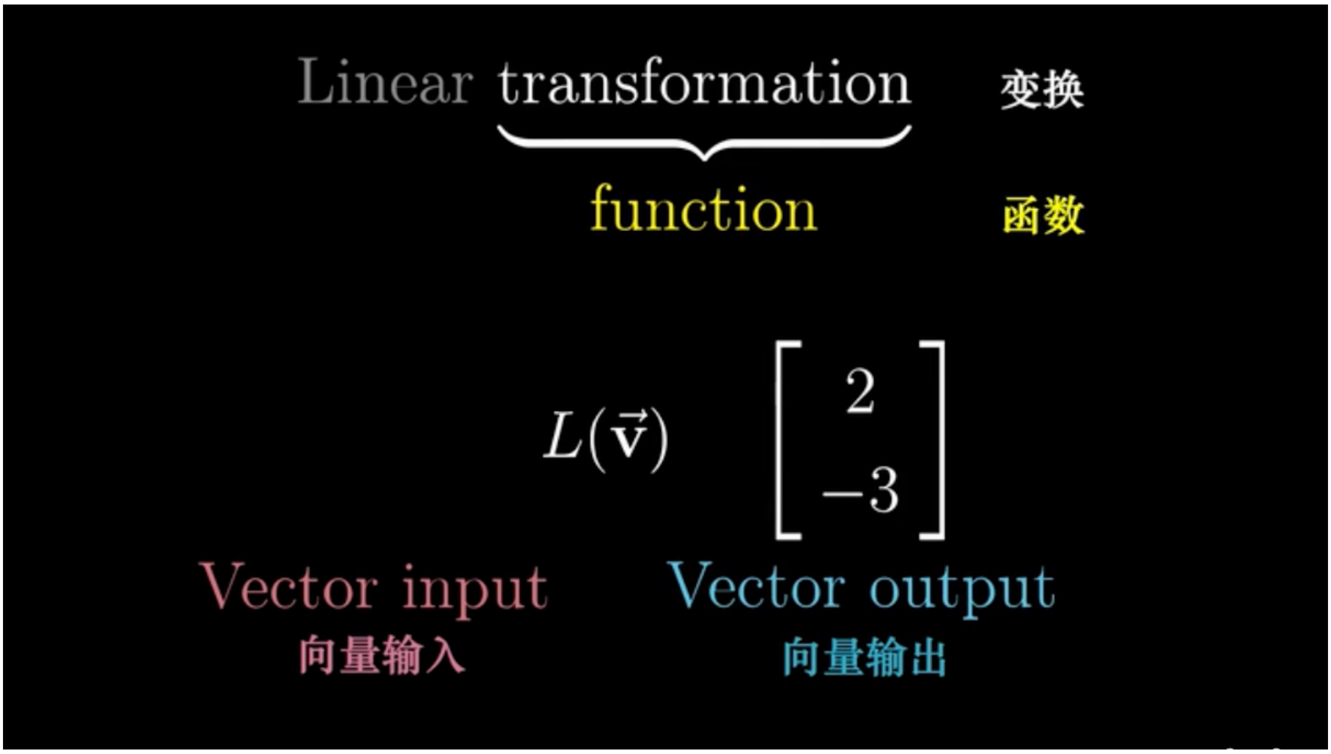
DirectX是由微软公司创建的多媒体编程接口。由C++编程语言实现。它们旨在使基于Windows 的计算机成为运行和显示具有丰富多媒体元素（例如全色图形、视频、3D 动画和丰富音频）的应用程序的理想平台。DirectX并不是一个单纯的图形API，它是由微软公司开发的用途广泛的AP

为什么要使用列向量?
1.等式中使用列向量形式更好
2.线性代数书中使用列向量
3.多本计算机图形学都是使用的列向量
4.OpenGL 使用的是列向量

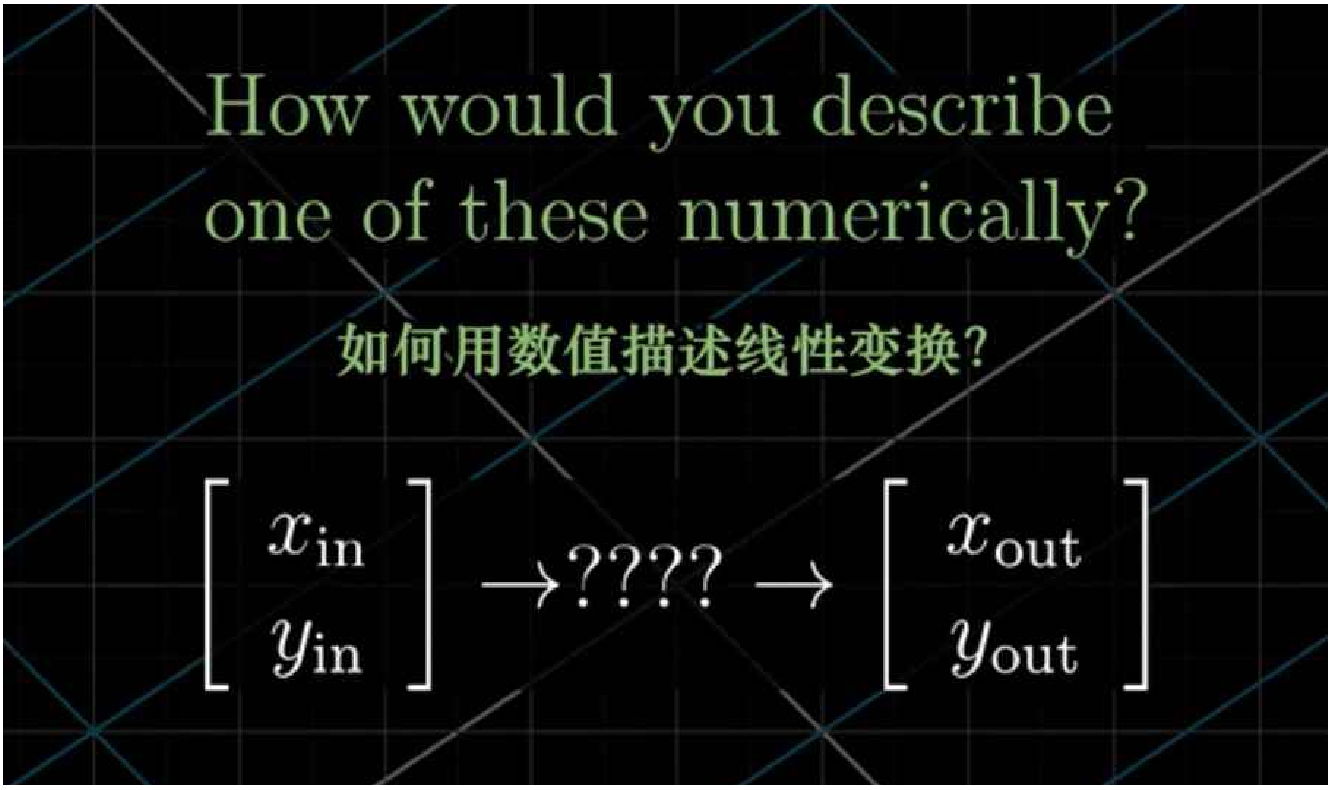
矩阵与线性变换：讲的通透

https://www.bilibili.com/video/BV1ns41167b9?from=search&seid=15604634838627420222

矩阵：矩阵可以看做是一个线性变换的函数，输入一个向量，输出一个新向量



如何用数值来描述线性变换

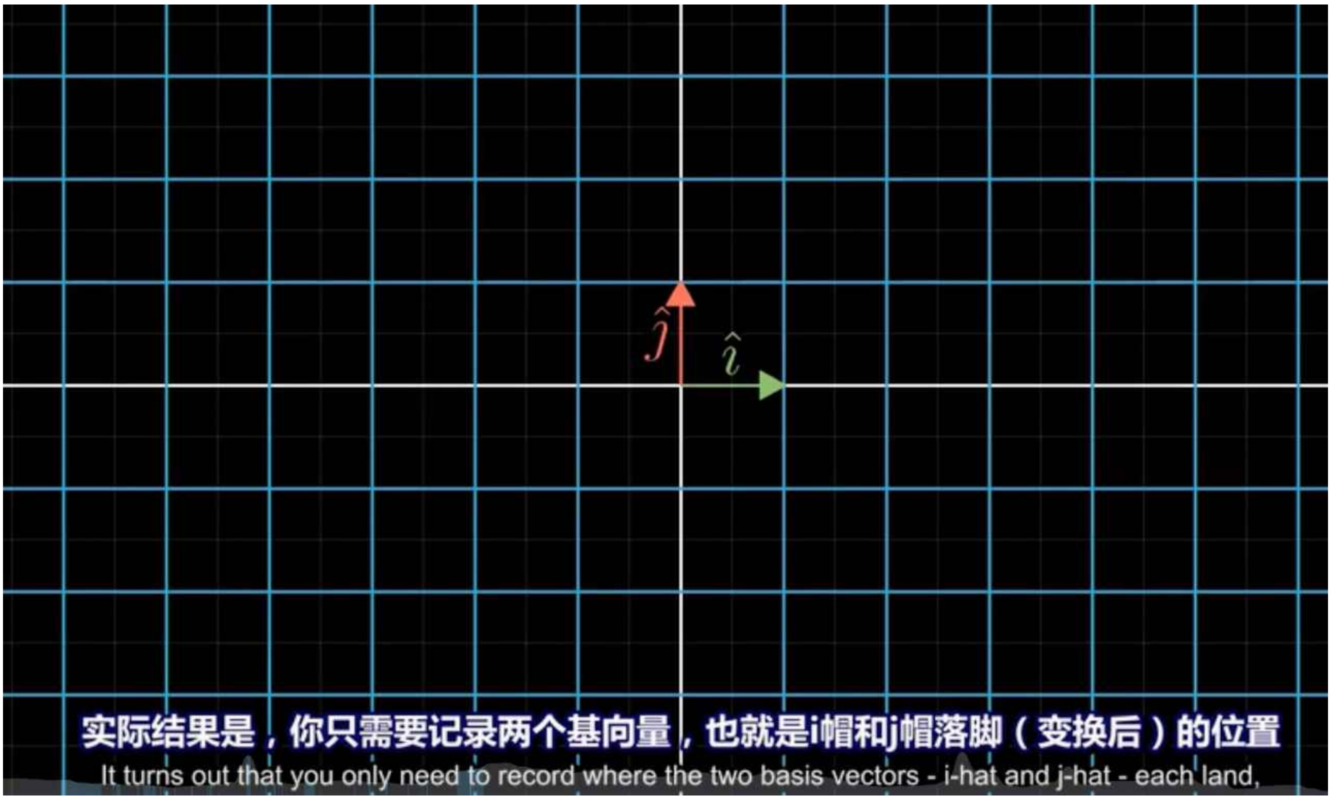


在平面空间里，需要先定义两个基向量，i和j

基向量：

1、大小固定，方向固定的向量（不等于单位向量）

2、空间变换后，基向量也会变



定义一个向量V，我们先看看向量V是怎么基于基向量而来的

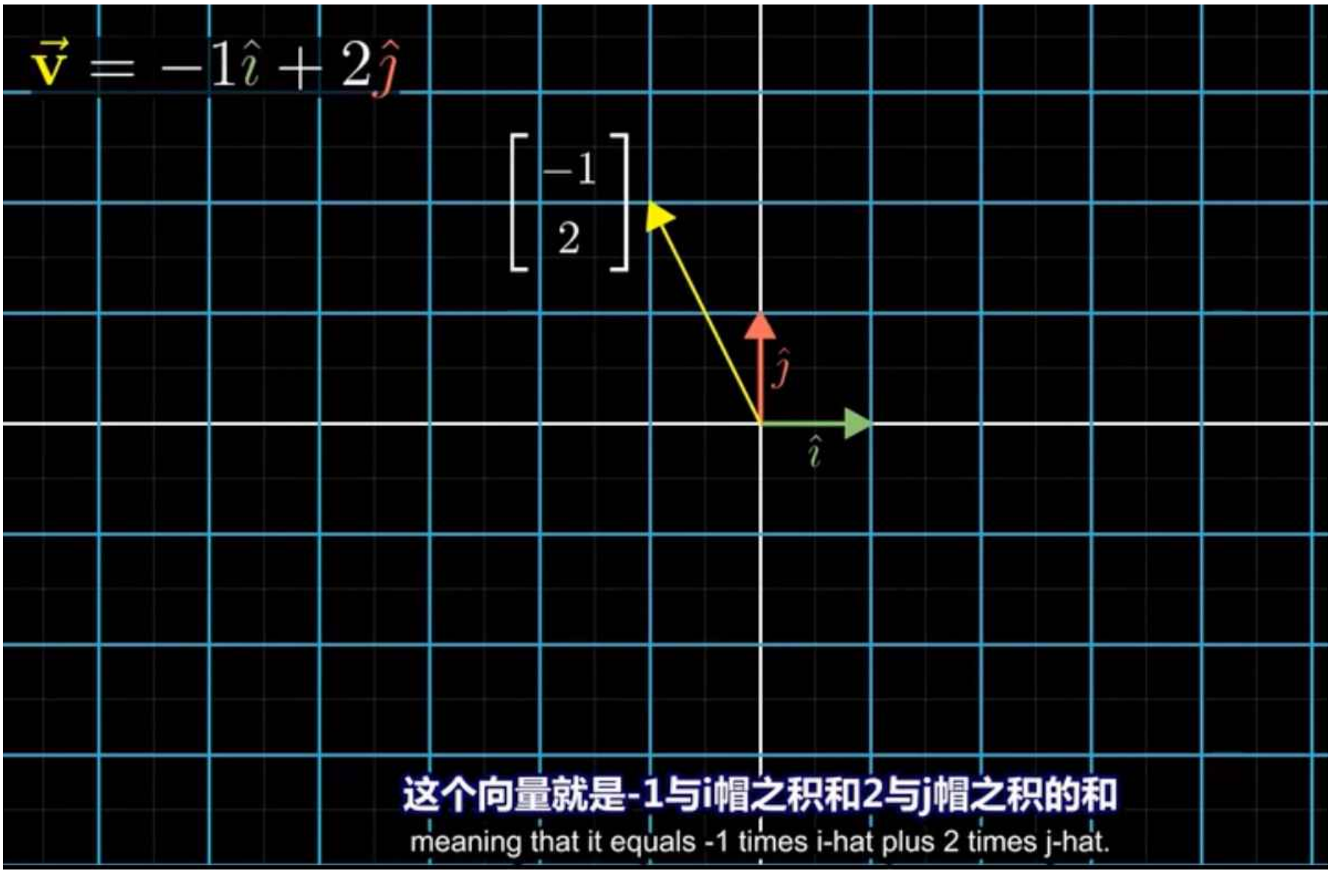
1、基向量i调整到相反方向，也就是绿色箭头要变成调整方向并且大小不变。这个过程可以由算式-1i 来表示。

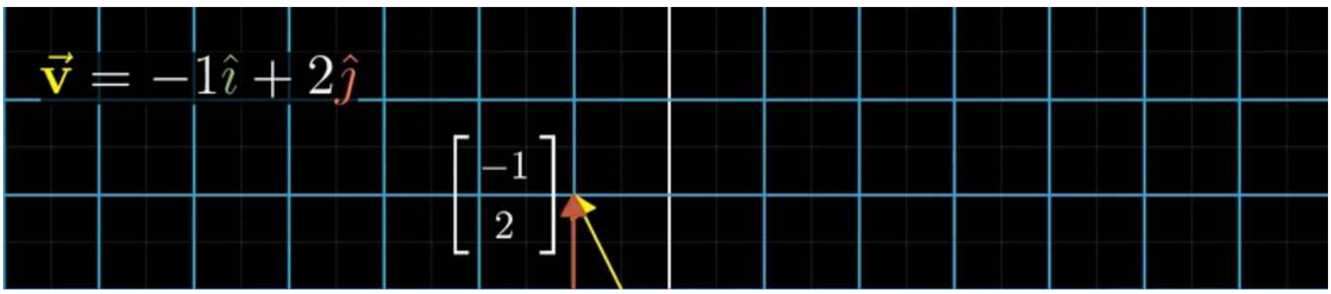
2、基向量j调整到原来的两倍，也就是红色箭头方向不变并且大小为原来两边。这个过程可以由算式2j 来表示。

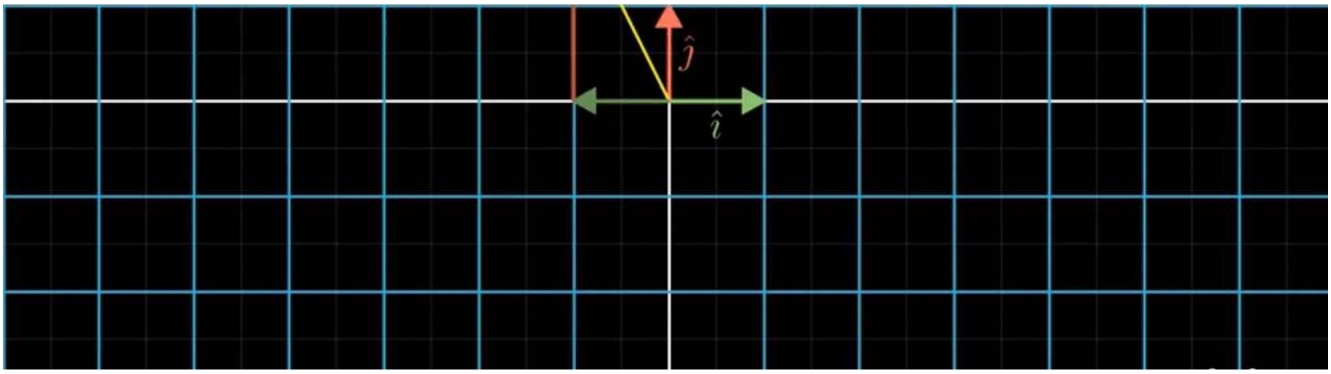
3、根据向量和的平行四边形法则，调整后的基向量j（红色箭头）要向左平移，让起始点接着调整后的基向量i（绿色箭头），

那 黄色箭头向量 = 调整后的绿色箭头向量 + 调整后的红色箭头向量，就可以推出： 向量V = -1i + 2j

所以向量V可以看做是基向量i与标量-1之积加上基向量j与标量2之积的和，算式：向量V = -1i + 2j，







假定基量i 为 [1, 0]，基量j为 [0, 1]，那么根据算式：向量V = -1i + 2j，向量V = [-1, 2]

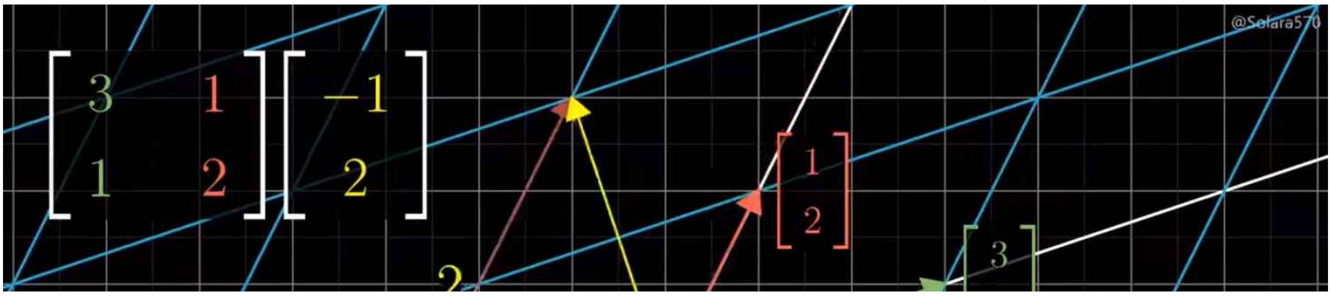
$$V = -1\begin{bmatrix}1\\0\end{bmatrix} + 2\begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}-1*1+2*0\\-1*0+2*1\end{bmatrix} = \begin{bmatrix}-1\\2\end{bmatrix}$$

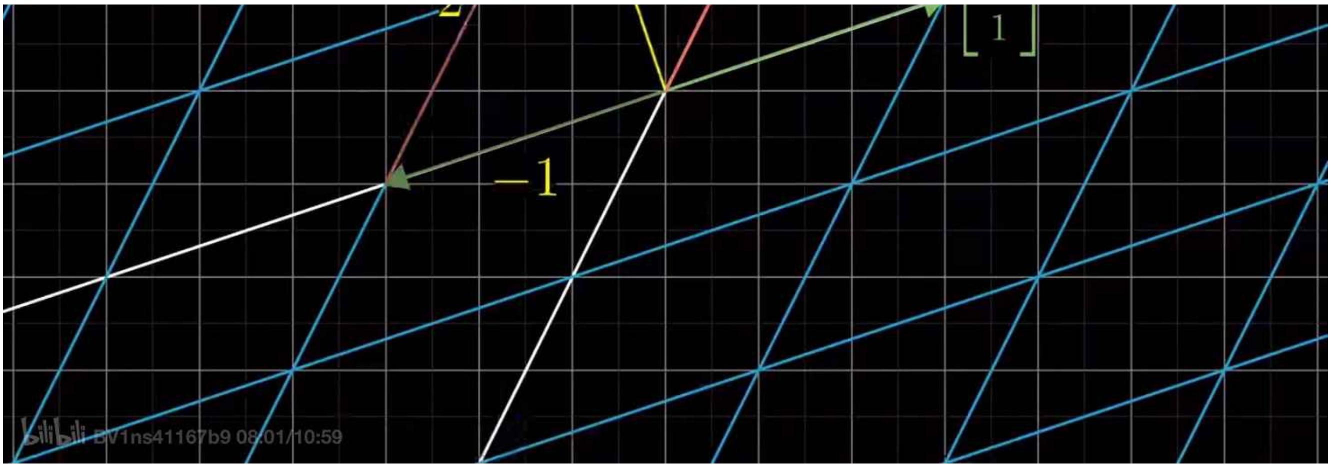
此时我们对空间做些变换，发现基向量 i 和 j，还有向量 V 都发生了变化，基向量i 为 [3, 1]，基向量j 为 [1, 2]，但是如果还是让基向量i调整到反方向，然后让基向量j调整为原来两倍，可以发现 黄色箭头向量 = 调整后的绿色箭头向量 + 调整后的红色箭头向量，这说明对空间进行线性变换时，

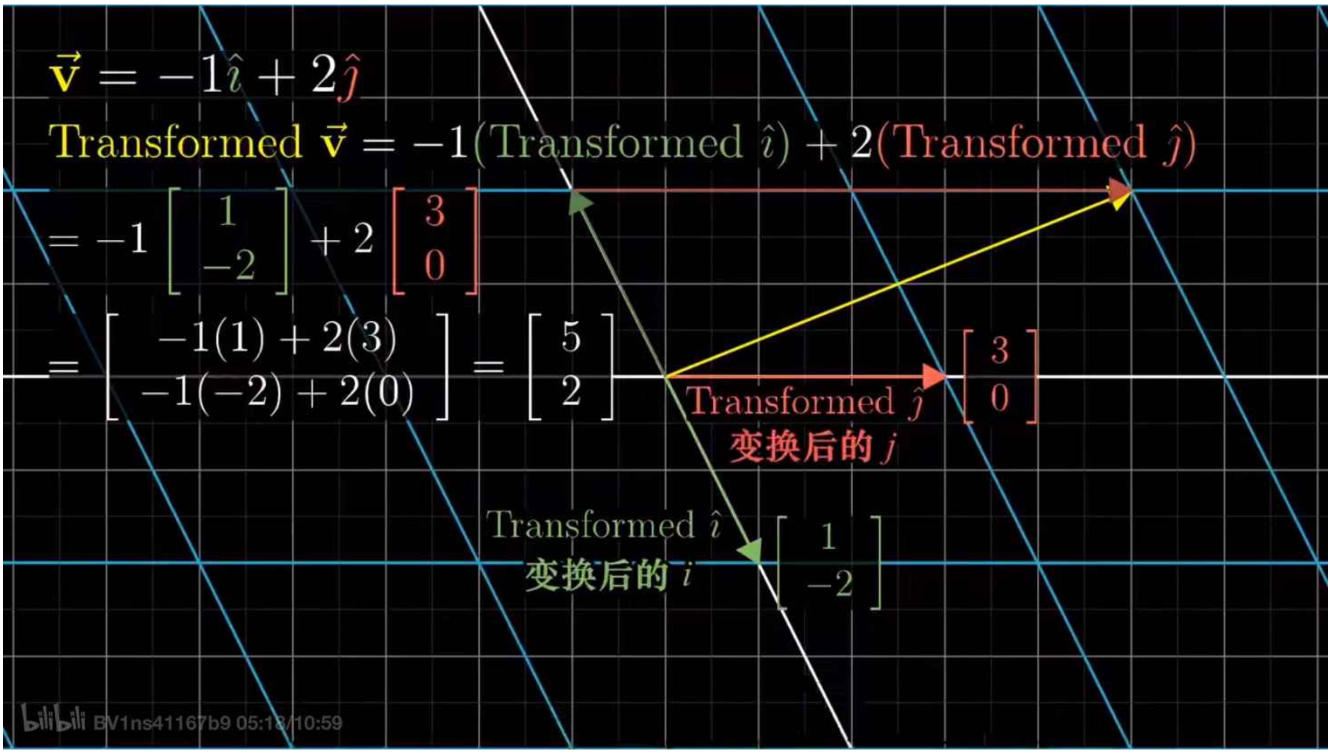
变化后的向量 = -1变换后的基向量i + 2变换后的基向量j

Transformed 向量V = 缩放因子(Transformed i) + 缩放因子(Transformed j) = 缩放新基向量再相加

意味着只要能找到变换后的基向量 i 和 j，通过缩放（拉伸和挤压）后，就一定能找到变换后的向量 V，此时向量 V = [-1, 3]







通过上面我们知道一个二维线性变换仅由四个数字完全确定，变换后的基向量 i 的坐标和变换后的基向量 j 的坐标。

通常我们把这些坐标放在一个 2x2 的格子里，称为 2x2 矩阵

我们试着使用矩阵与向量相乘的方式来看下结果，这里可以发现矩阵与向量相乘的结果与上面通过基向量的线性变换的结果是一致的，都是 向量 V = [-1, 3]

$$\begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} -1 \\ 2 \end{bmatrix} = \begin{bmatrix} -1 \\ 3 \end{bmatrix}$$

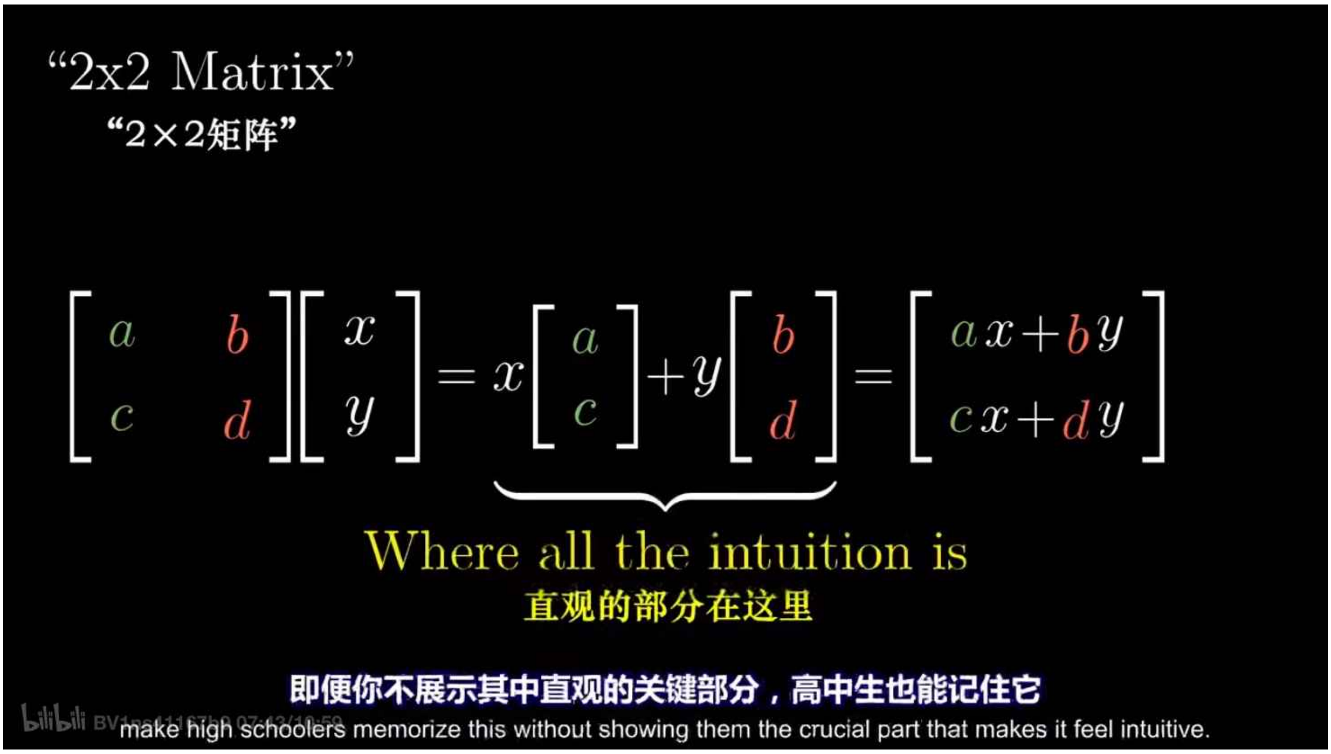
所以我们就得出了下面这个算式，这里可以得出两种理解

1、矩阵就是描述线性变换（也可以看作是线性变换的函数，后面右乘向量可以看作是入参，最后的结果出参是一个线性变换后的新向量），矩阵与向量相乘的几何意义就是把向量根据矩阵的线性变换得到一个新的向量

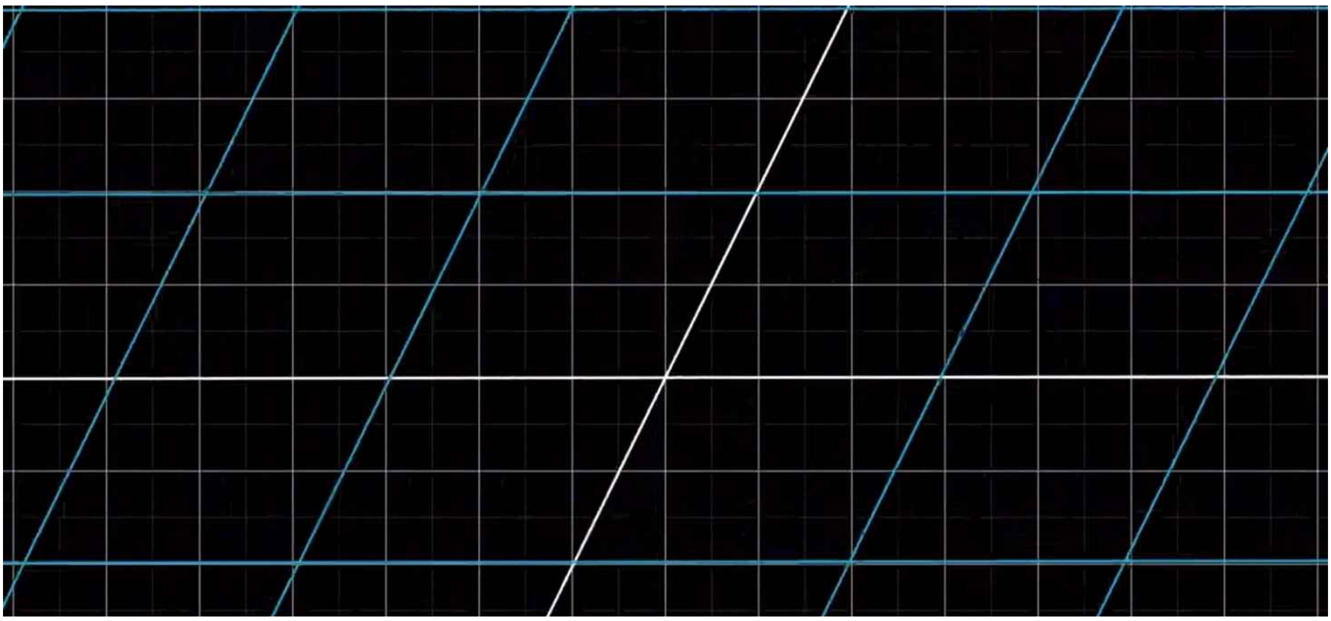
2、变换后的向量都可以通过缩放新基向量再相加得来：Transformed 向量V = 缩放因子(Transformed i) + 缩放因子(Transformed j) = 缩放新基向量再相加

$$V = \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} -1 \\ 2 \end{bmatrix} = -1 \begin{bmatrix} 3 \\ 1 \end{bmatrix} + 2 \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} -1*3+2*1 \\ -1*1+2*2 \end{bmatrix} = \begin{bmatrix} -1 \\ 3 \end{bmatrix}$$

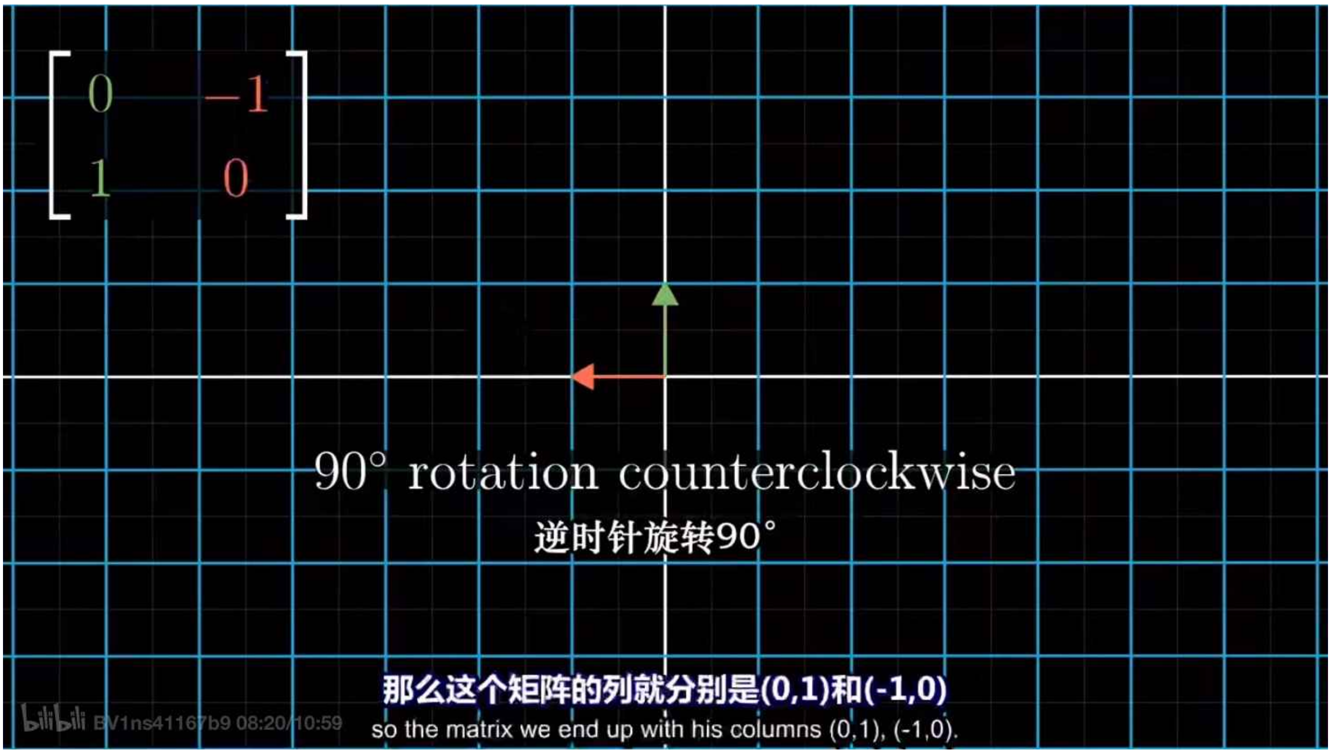
通过上面的理解，我们可以有一个通用的公式如下图所示：

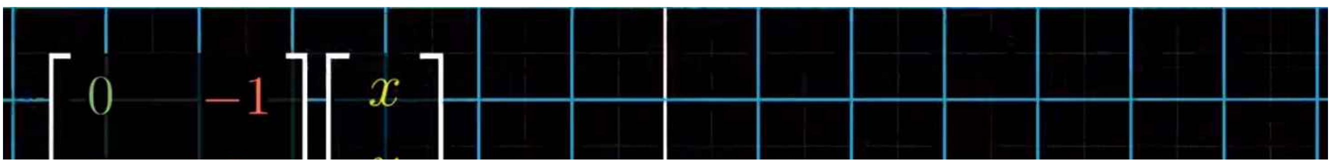


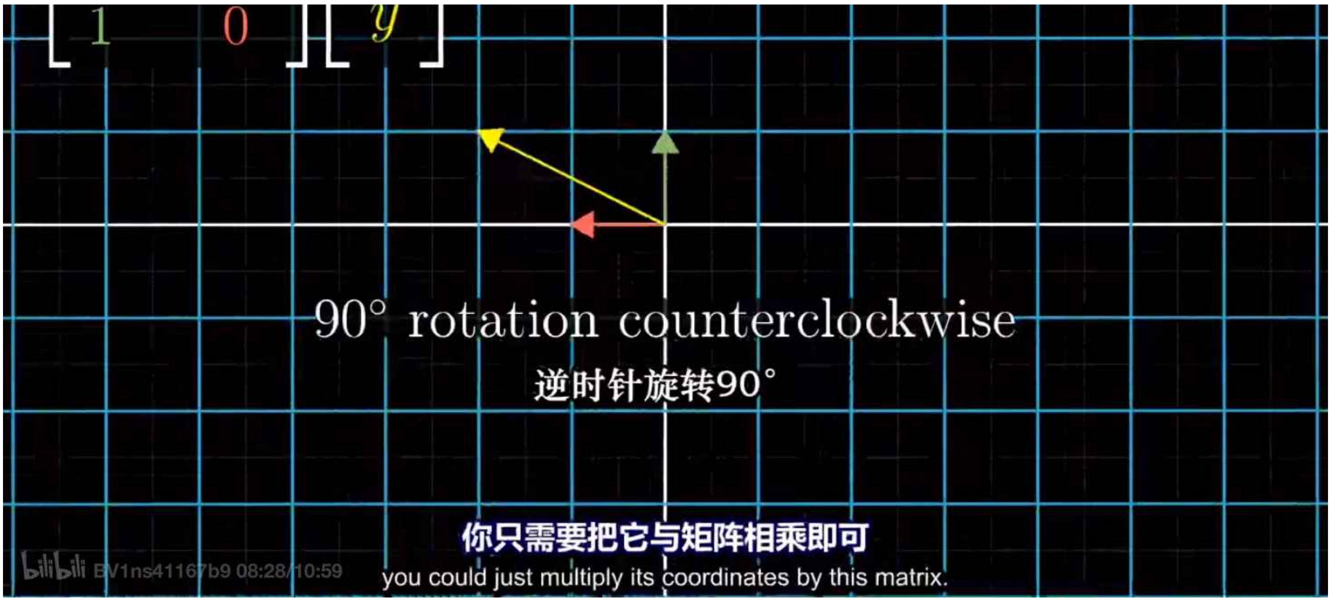
总的来说线性变换可以看作是保持网格线平行并等距分布的变换（原点不动）




总的来说，你应该把线性变换看作是“保持网格线平行并等距分布”的变换
In general, you should think of linear transformations as "keeping grid lines parallel and evenly spaced".


$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$
90° rotation counterclockwise
逆时针旋转90°
那么这个矩阵的列就分别是(0,1)和(-1,0)
bilibili BV1ns41167b9 08:20/10:59
so the matrix we end up with his columns (0,1), (-1,0).


0
−1
x



总结：

1、矩阵就是描述线性变换（也可以看作是线性变换的函数，后面右乘向量可以看作是入参，最后的结果出参是一个线性变换后的新向量），矩阵与向量相乘的几何意义就是把向量根据矩阵的线性变换得到一个新的向量

2、变换后的向量都可以通过缩放新基向量再相加得来：Transformed 向量V = 缩放因子(Transformed i) + 缩放因子(Transformed j) = 缩放新基向量再相加

3、线性变换可以看作是保持网格线平行并等距分布的变换（原点不动）

工具：

矩阵计算工具

https://matrix.reshish.com/zh/multCalculation.php

https://zhuanlan.zhihu.com/p/111337479

数学公式工具

舒适的LaTeX代码编辑体验：VS Code + LaTeX Workshop

https://zhuanlan.zhihu.com/p/106167792?utm_source=QQ_article_bottom

https://blog.csdn.net/yinqingwang/article/details/79684419

第一个demo

https://github.com/James-Yu/LaTeX-Workshop/blob/master/sample/t.tex

LaTex语法

https://www.docx2latex.com/tutorials/MaTrix-LaTeX.html

Origin Link: https://wepie.yuque.com/tcsdzz/ios_team/xbu2r9