

我是8位的

I am 8 bits, what about you?

对于向量几何可以留到高数探究, 可以从
方程组入手

随笔 - 205, 文章 - 0, 评论 - 103, 阅读 - 101万

导航

博客园
首页
新随笔
联系
订阅 
管理

2022年3月						
日	一	二	三	四	五	六
27	28	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

公告

你的支持是我的动力
欢迎关注微信公众号“我是8位的”



昵称：我是8位的
园龄：4年7个月
粉丝：288
关注：5
+加关注

盖楼抽奖

#她的梦想在发光#

HWD科技女性故事有奖征集

分享最打动的科技女性故事

活动时间：2022年3月8日-3月18日

马上参与



搜索

 找找看
 谷歌搜索

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

积分与排名

积分 - 457097
排名 - 1198

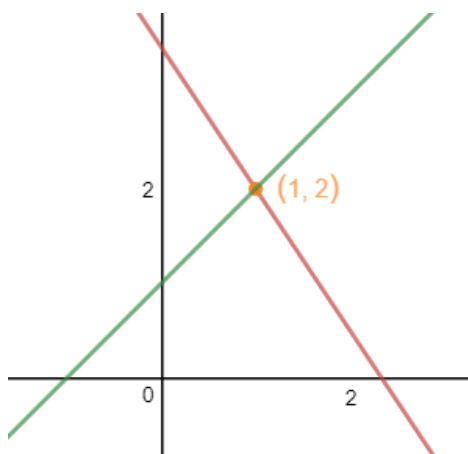
线性代数笔记5——平面方程与矩阵

线性方程的几何意义

二元线性方程

$$\begin{cases} 3x + 2y = 7 \\ -6x + 6y = 6 \end{cases}$$

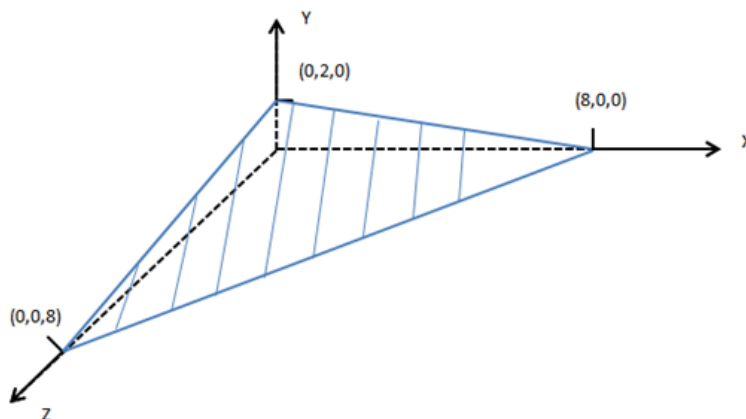
该方程是一个二元线性方程组，包含两个方程，每个方程是一条直线，两条直线的交点就是该方程有唯一解，这就是二元线性方程的几何意义。



平面方程

空间内不在同一直线上的三点构成一个平面，平面方程可表示为 $ax + by + cz = d$ 。平面方程也称为三元线性方程。

方程 $x + 4y + z = 8$ ，在xyz三个坐标轴上的截距分别是(8,0,0),(0,2,0),(0,0,8)，下图是该函数在坐标轴上的示意图：



需要注意的是，平面是无限延伸的。

根据法向量求平面方程

随笔分类 (211)

★★资源下载★★(1)
Java并发编程(1)
程序员的数学(24)
单变量微积分(31)
多变量微积分(24)
概率(24)
机器学习(27)
软件设计(1)
数据分析(6)
数据结构与算法(27)
随笔(5)
线性代数(34)
项目管理(2)
转载(4)

随笔档案 (205)

2021年2月(1)
2020年3月(2)
2020年2月(6)
2020年1月(4)
2019年12月(7)
2019年11月(15)
2019年9月(3)
2019年8月(6)
2019年7月(1)
2019年6月(8)
2019年5月(3)
2019年4月(5)
2019年3月(7)
2019年2月(3)
2019年1月(7)
更多

阅读排行榜

1. 使用Apriori进行关联分析 (一) (29768)
2. 线性代数笔记12——列空间和零空间 (28772)
3. FP-growth算法发现频繁项集 (一)——构建FP树(24430)
4. 寻找“最好” (2) ——欧拉-拉格朗日方程(23099)
5. 多变量微积分笔记3——二元函数的极值(22772)

评论排行榜

1. 隐马尔可夫模型 (一) (8)
2. 线性代数笔记12——列空间和零空间 (7)
3. 线性代数笔记3——向量2 (点积) (7)
4. FP-growth算法发现频繁项集 (一)——构建FP树(5)
5. 寻找“最好” (2) ——欧拉-拉格朗日方程(4)

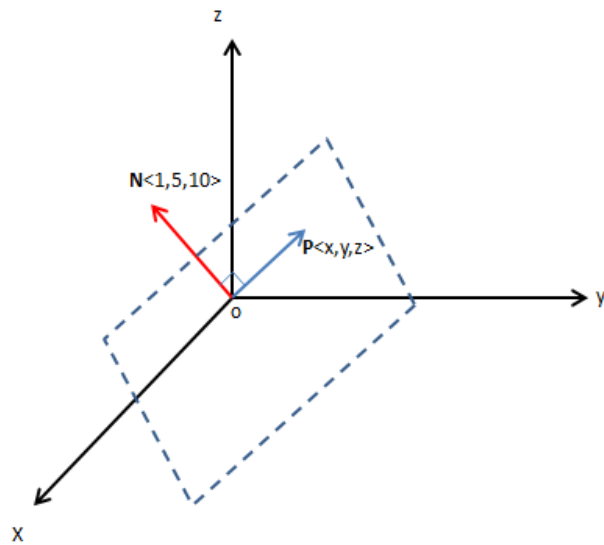
推荐排行榜

1. 寻找“最好” (2) ——欧拉-拉格朗日方程(7)
2. FP-growth算法发现频繁项集 (一)——构建FP树(7)
3. 线性代数笔记3——向量2 (点积) (6)
4. FP-growth算法发现频繁项集 (二)——发现频繁项集(5)
5. 隐马尔可夫模型 (一) (5)

最新评论

1. Re:线性代数笔记3——向量2 (点积)
如果点积小于0, 即夹角小于90°, 这个写错了吧。应该是夹角大于90°
--猫猫猫猫猫大人

现在需要找到一个过原点的平面, 它有一个过原点的法向量是 $\langle 1, 5, 10 \rangle$ 。



如上图所示, $P = \langle x, y, z \rangle$ 是所求平面上的向量, 法向量 $N \perp OP$, 因此:

$$N \cdot P = 0$$

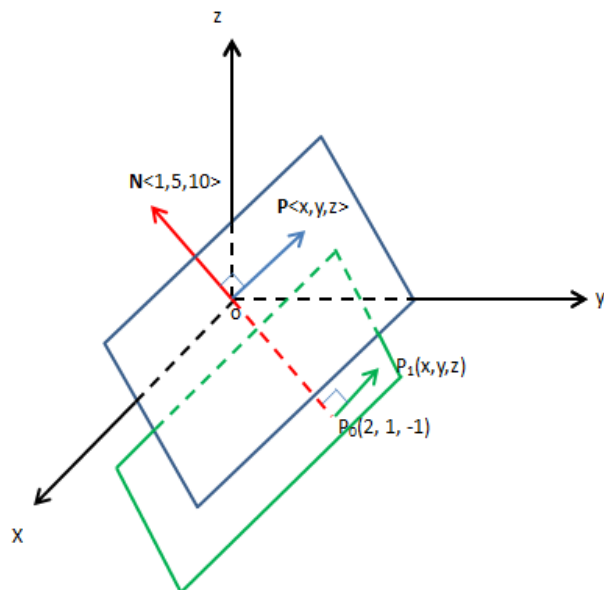
$$\langle 1, 5, 10 \rangle \cdot \langle x, y, z \rangle = 0$$

$$x + 5y + 10z = 0$$

这就是平面方程。

再看一个稍微不同点的问题, 一个平面的法向量是 $N = \langle 1, 5, 10 \rangle$, 该平面经过 $P_0(2, 1, -1)$, 求该平面方程。

由于拥有同一个法向量, 所以这是与上一个平面平行的平面:



平面上的任意点 P_1 是 (x, y, z) , 向量 $P_0P_1 \perp N$:

2. Re:线性代数笔记10——矩阵的LU分解写的很好，不过LU分解的前提是错的，LU分解只需要第三个条件，如果允许行置换就是下面写到的PLU，可以分解所有矩阵

--wiki3D

3. Re:单变量微积分笔记20——三角替换1 (sin和cos) 很nice

--尹保棕

4. Re:线性代数笔记24——微分方程和exp(At)

有些图片挂了呢

--ccchendada

5. Re:寻找“最好” (2) ——欧拉-拉格朗日方程

提个issue，最速降线中

$v = \{2gh\}^{1/2}$ 与配图不一致，建议以起点为原点，向右伸出x轴，向下伸出y轴建立坐标系

--trustInU

$$N \cdot P_0 P_1 = 0$$

$$\langle 1, 5, 10 \rangle \cdot \langle x - 2, y - 1, z + 1 \rangle = 0$$

$$x + 5y + 10z = -3$$

上面两个方程唯一的不同点就是 $ax + by + cz = d$ 中的 d ，其它参数对应了穿过原点的法向量，实际上， d 两个平行平面的距离。根据这个特点，可以很快求得第二个平面方程：

$$x + 5y + 10z = d$$

$$\text{代入 } P_0: 2 + 5 \times 1 + 10 \times (-1) = d \Rightarrow d = -3$$

示例

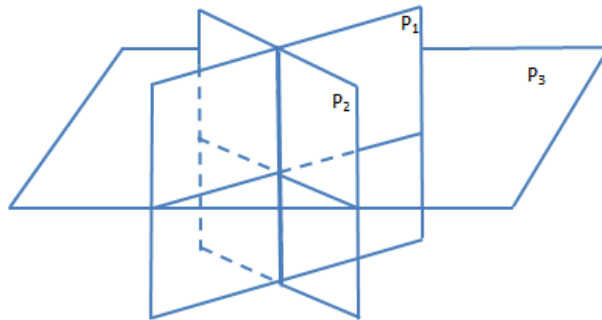
向量 $\mathbf{V} = \langle 1, 2, -1 \rangle$ 与平面 $x + y + 3z = 5$ 的关系？

平面的法向量 $\mathbf{N} = \langle 1, 1, 3 \rangle$ ，容易看出， $\mathbf{V} \cdot \mathbf{N} = 1 \times 1 + 2 \times 1 + (-1) \times 3 = 0$ ， $\mathbf{V} \perp \mathbf{N}$ ，向量 $\mathbf{V}$ 与平面平行。需要注意的是，向量不是点（实际上向量有无数点）， $\langle 1, 2, -1 \rangle$ 不同于 $(1, 2, -1)$ ，在没有特殊说明的情况下，可以认为向量从原点出发。如果向量 $\mathbf{V}$ 从原点出发， $\mathbf{V}$ 经过点 $(1, 2, -1)$ ，但该点并不在平面上。

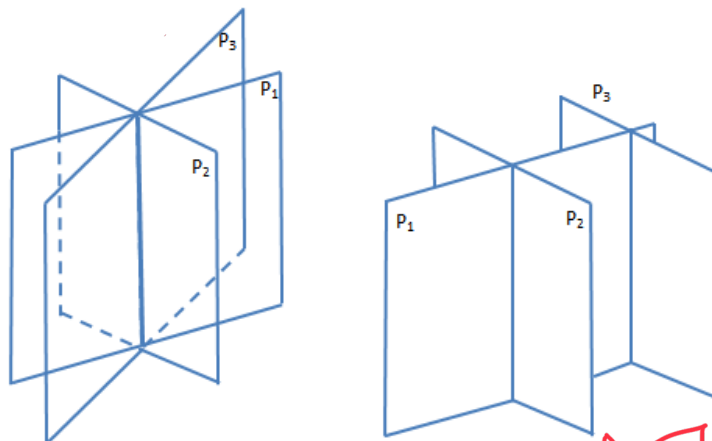
平面方程组的解

$$\begin{cases} x + z = 1 \\ x + y = 2 \\ x + 2y + 3z = 3 \end{cases}$$

三元线性方程组 $x + 2y + 3z = 3$ ，设三个平面分别是 P_1, P_2, P_3 ，该方程组有唯一解，即这三个平面相交于一点，三个方程两两相交于一条直线：



平面方程组也可能出现无解的情况，一种典型的情况是三个平面平行。如果 $P_1 \cap P_2 \neq \varnothing$ ，即二者相交于一条直线，根据 P_3 的位置，平面方程组可能有唯一解，无解，或有无数解。下面是无数解和无解的情况：



无数解和无解

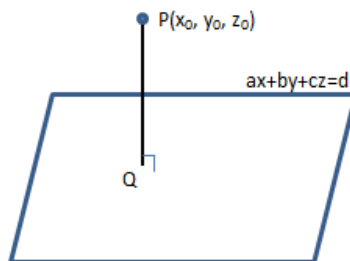
总结一下，如果 P_1 与 P_2 相交，它们的交线：

1. 与 P_3 相交于一点，则方程组有唯一解；
2. 在 P_3 上，则方程组有无数解；
3. 与 P_3 平行，且不在 P_3 上，方程组无解。

当然，如果 P_3 与 P_1 或 P_2 中一个相同，则结集是一个平面。

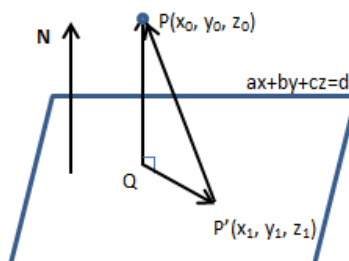
点到平面的距离

平面方程是 $ax + by + cz = d$ ，平面外一点 $P = (x_0, y_0, z_0)$ ，求该点到平面的距离。



PQ垂直于平面，现在要求PQ的长度，但是并不知道Q点的具体数值。

设 $P' = (x_1, y_1, z_1)$ 是平面上的一点，现在将问题转换为向量：



向量 $\overrightarrow{QP}$ 是 $\overrightarrow{P'P}$ 在法向量 $\mathbf{N}$ 方向上的分量，也就是 $\overrightarrow{P'P}$ 与 $\mathbf{N}$ 相同方向的单位向量的点积。（可参考《[线性代数笔记3——向量2（点积）](#)》）设距离为 D ，则：

$$\pm D = \mathbf{P}' \mathbf{P} \cdot \mathbf{u}$$

$$= \mathbf{P}' \mathbf{P} \cdot \frac{\mathbf{N}}{|\mathbf{N}|}$$

$$= \langle x_1 - x_0, y_1 - y_0, z_1 - z_0 \rangle \cdot \frac{\langle a, b, c \rangle}{\sqrt{a^2 + b^2 + c^2}}$$

$$= \frac{a(x_1 - x_0) + b(y_1 - y_0) + c(z_1 - z_0)}{\sqrt{a^2 + b^2 + c^2}}$$

$$= \frac{(ax_1 + by_1 + cz_1) - (ax_0 + by_0 + cz_0)}{\sqrt{a^2 + b^2 + c^2}}$$

$$= \frac{d - (ax_0 + by_0 + cz_0)}{\sqrt{a^2 + b^2 + c^2}}$$

$$D = \frac{|ax_0 + by_0 + cz_0 - d|}{\sqrt{a^2 + b^2 + c^2}}$$

求 P 在法向量 N 方向上的分量

求解线性方程

当然可以使用初中的代数知识求解线性方程组，这里主要讨论如何用矩阵求解。

消元法

$$\text{求解} \begin{cases} -x + 2y - z = 9 \\ 3x - 7y - 2z = -20 \\ 2x + 2y + z = 2 \end{cases}$$

首先将方程组以矩阵的方式表示：

$$\left[\begin{array}{ccc|c} -x & 2y & -z & 9 \\ 3x & -7y & -2z & -20 \\ 2x & 2y & z & 2 \end{array} \right]$$

该矩阵称为增广矩阵。由于是线性方程组，可以省略未知数：

$$\left[\begin{array}{ccc|c} -1 & 2 & -1 & 9 \\ 3 & -7 & -2 & -20 \\ 2 & 2 & 1 & 2 \end{array} \right]$$

现在可以对其进行消元，首先消去 x ，方法与普通代数法类似：

$$\left[\begin{array}{ccc|c} -1 & 2 & -1 & 9 \\ 3 + (-1) \times 3 & -7 + 2 \times 3 & -2 + (-1) \times 3 & -20 + 9 \times 3 \\ 2 + (-1) \times 2 & 2 + 2 \times 2 & 1 + (-1) \times 2 & 2 + 9 \times 2 \end{array} \right] = \left[\begin{array}{ccc|c} -1 & 2 & -1 & 9 \\ 0 & -1 & -5 & 7 \\ 0 & 6 & -1 & 20 \end{array} \right]$$

用同样的方法对 y 消元：

$$\left[\begin{array}{ccc|c} -1 & 2 & -1 & 9 \\ 0 & -1 & -5 & 7 \\ 0 & 6 + (-1) \times 6 & -1 + (-5) \times 6 & 20 + 7 \times 6 \end{array} \right] = \left[\begin{array}{ccc|c} -1 & 2 & -1 & 9 \\ 0 & -1 & -5 & 7 \\ 0 & 0 & -31 & 62 \end{array} \right]$$

矩阵第三行对应 $-31z = 62$, $z = -2$

$$\text{最终可解得} \begin{cases} x = 1 \\ y = 3 \\ z = -2 \end{cases}$$

$$\begin{aligned} -x + 2y - z &= 9 \\ -y - 5z &= 7 \\ -31z &= 62 \end{aligned}$$

可以看出，消元法本质上与初中的代数法没有区别，只是换了一种较为简单的表现形式，对于多元线性方程组，其消元过程十分繁琐。

矩阵法

$$\text{求解} \begin{cases} 3x + 2y = 7 \\ -6x + 6y = 6 \end{cases}$$

这里需要使用列向量的概念，列向量是一个 $n \times 1$ 的矩阵，即矩阵由一个含有 n 个元素的列所组成：列向量的转置是一个行向量，反之亦然。

将上面的方程组用矩阵和向量表示：

$$\begin{bmatrix} 3 & 2 \\ -6 & 6 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 7 \\ 6 \end{bmatrix}$$

$$\text{令 } A = \begin{bmatrix} 3 & 2 \\ -6 & 6 \end{bmatrix}, x = \begin{bmatrix} x \\ y \end{bmatrix}, b = \begin{bmatrix} 7 \\ 6 \end{bmatrix}, x \text{ 和 } b \text{ 是列向量,}$$

$$\underbrace{\begin{bmatrix} 3 & 2 \\ -6 & 6 \end{bmatrix}}_A \underbrace{\begin{bmatrix} x \\ y \end{bmatrix}}_x = \underbrace{\begin{bmatrix} 7 \\ 6 \end{bmatrix}}_b$$

$$Ax = b$$

$$A^{-1} = \frac{1}{\det(A)} \begin{bmatrix} 6 & -2 \\ 6 & 3 \end{bmatrix}$$

$$A^{-1}Ax = A^{-1}b$$

$$Ex = A^{-1}b$$

$$x = A^{-1}b$$

$$\begin{bmatrix} \frac{6}{30} & \frac{-2}{30} \\ \frac{6}{30} & \frac{3}{30} \end{bmatrix}$$

$$\begin{bmatrix} \frac{1}{5} & -\frac{1}{15} \\ \frac{1}{5} & \frac{1}{10} \end{bmatrix}$$

实际上可看作 $x = b/A$ ，有点意思了，可以通过一个除法运算直接求得方程的解。

$$\begin{bmatrix} x \\ y \end{bmatrix} = x = A^{-1}b = \begin{bmatrix} 1/5 & -1/15 \\ 1/5 & 1/10 \end{bmatrix} \begin{bmatrix} 7 \\ 6 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$\text{解得} \begin{cases} x = 1 \\ y = 2 \end{cases}$$

对于多元线性方程组，使用矩阵法求解比消元法简单的多。

$$\text{我们用python求解消元法中的方程组} \begin{cases} -x + 2y - z = 9 \\ 3x - 7y - 2z = -20 \\ 2x + 2y + z = 2 \end{cases}$$

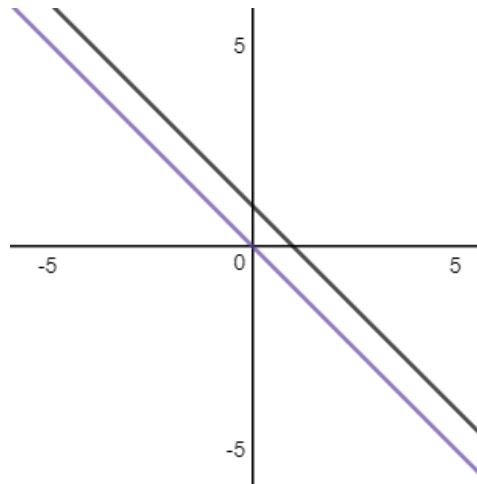
```
1 import numpy as np
2
3 a = np.matrix([[ -1, 2, -1], [ 3, -7, -2], [ 2, 2, 1]])
4 c = np.matrix([[ 9, -20, 2]]).T
5 result = a**(-1) * c
6 print(result)
```

```
[[ -1.]
 [  3.]
 [ -2.]]
```

无解的方程组

线性方程组在用矩阵向量法转换后，如果矩阵 A 是奇异矩阵， A^{-1} 没有定义，该方程组无解。对于二元线性方程组来说，其几何意义是两条平行的直线。

如 $\begin{cases} x+y=1 \\ x+y=0 \end{cases}$, 该方程组无解, $A = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ 是奇异矩阵。下图是该方程组在坐标轴上的图像:



多个线性方程组

现在来看下面的矩阵:

$$\begin{bmatrix} 1 & 3 \\ 2 & 7 \end{bmatrix} \begin{bmatrix} a & c \\ b & d \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

如果把这个矩阵根据乘法展开, 将得到两个矩阵:

$$\begin{bmatrix} 1 & 3 \\ 2 & 7 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 1 & 3 \\ 2 & 7 \end{bmatrix} \begin{bmatrix} c \\ d \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

这就可以看作是两个线性方程组: $AX_1 = b_1$, $AX_2 = b_2$

消元法求逆矩阵

上面的示例中, $\begin{bmatrix} a & c \\ b & d \end{bmatrix}$ 是 $\begin{bmatrix} 1 & 3 \\ 2 & 7 \end{bmatrix}$ 的逆, 现在尝试用消元法求解。增广矩阵形式:

$$\underbrace{\begin{bmatrix} 1 & 3 \\ 2 & 7 \end{bmatrix}}_A \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_I = AI$$

我们的目标是将其变为 IA^{-1} :

$$\underbrace{\begin{bmatrix} 1 & 3 \\ 2 & 7 \end{bmatrix}}_A \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_I \rightarrow \begin{bmatrix} 1 & 3 & 1 & 0 \\ 0 & 1 & -2 & 1 \end{bmatrix} \rightarrow \underbrace{\begin{bmatrix} 1 & 0 & 7 & -3 \\ 0 & 1 & -2 & 1 \end{bmatrix}}_{IA^{-1}}$$

示例

示例1

求下面的平面方程:

- 已知平面的法向量 $\mathbf{N} = \langle 1, 2, 3 \rangle$, 平面过点 $(1, 0, -1)$
- 平面过原点且平行于两个向量 $\mathbf{A} = \langle 1, 0, -1 \rangle$ 和 $\mathbf{B} = \langle -1, 2, 0 \rangle$
- 平面过点 $P_1(1, 2, 0)$, $P_2(3, 1, 1)$, $P_3(2, 0, 0)$
- 平面与a中的平面平行, 且经过点 $(1, 2, 3)$

a.

平面方程 $ax + by + cz = d$, $\mathbf{N} = \langle 1, 2, 3 \rangle = \langle a, b, c \rangle$, 所以平面方程是 $x + 2y + 3z = d$

将点 $(1, 0, -1)$ 代入平面方程, $1 + 0 - 3 = -2 = d$

平面方程是 $x + 2y + 3z = -2$

b.

平面方程 $ax + by + cz = d$

$\therefore \mathbf{A}, \mathbf{B}$ 过原点, 且与平面平行, 并且平面过原点

$\therefore \mathbf{A}, \mathbf{B}$ 在平面上, $d = 0$

已知平面上两个个向量从同一点出发的向量, 计算平面的法向量:

$$\begin{aligned} \mathbf{N} = \mathbf{A} \times \mathbf{B} &= \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ 1 & 0 & -1 \\ -1 & 2 & 0 \end{vmatrix} \\ &= \langle \mathbf{i} \begin{vmatrix} 0 & -1 \\ 2 & 0 \end{vmatrix}, -\mathbf{j} \begin{vmatrix} 1 & -1 \\ -1 & 0 \end{vmatrix}, \mathbf{k} \begin{vmatrix} 1 & 0 \\ -1 & 2 \end{vmatrix} \rangle \\ &= \langle 2, 1, 2 \rangle \end{aligned}$$

平面方程是 $2x + y + 2z = 0$

根据叉积算法向量可参考《[线性代数笔记4——向量3 \(叉积\)](#)》

c.

$$\mathbf{P}_3 \mathbf{P}_1 = \mathbf{P}_1 - \mathbf{P}_3 = \langle 1 - 2, 2 - 0, 0 - 0 \rangle = \langle -1, 2, 0 \rangle$$

$$\mathbf{P}_3 \mathbf{P}_2 = \mathbf{P}_2 - \mathbf{P}_3 = \langle 3 - 2, 1 - 0, 1 - 0 \rangle = \langle 1, 1, 1 \rangle$$

$$\mathbf{N} = \mathbf{P}_3 \mathbf{P}_1 \times \mathbf{P}_3 \mathbf{P}_2$$

$$\begin{aligned} &= \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ -1 & 2 & 0 \\ 1 & 1 & 1 \end{vmatrix} \\ &= \langle \mathbf{i} \begin{vmatrix} 2 & 0 \\ 1 & 1 \end{vmatrix}, -\mathbf{j} \begin{vmatrix} -1 & 0 \\ 1 & 1 \end{vmatrix}, \mathbf{k} \begin{vmatrix} -1 & 2 \\ 1 & 1 \end{vmatrix} \rangle \\ &= \langle 2, 1, -3 \rangle \end{aligned}$$

平面方程 $2x + y - 3z = d$, 取任意点代入, $d = 4$ 。平面方程是 $2x + y - 3z = 4$

d.

a的平面是 $x + 2y + 3z = -2$, 由于该平面平行于a, 所以该平面是 $x + 2y + 3z = d$ 。

将点 $(1, 2, 3)$ 代入, $1 + 2 \times 2 + 3 \times 3 = 14 = d$

平面方程是 $x + 2y + 3z = 14$

示例2

求原点到平面 $2x + y - 2z = 4$ 的距离。

$$D = \frac{|ax_0 + by_0 + cz_0 - d|}{\sqrt{a^2 + b^2 + c^2}} = \frac{|2 \times 0 + 1 \times 0 - 2 \times 0 - 4|}{\sqrt{2^2 + 1^2 + (-2)^2}} = \frac{4}{3}$$

总结

- 1. 二元线性方程组的几何意义是平面上的两条直线，其解是二者的交点
- 2. 三元线性方程组的几何意义是三维空间上的三个平面，可能存在唯一解、无数解或无解
- 3. 平面方程用 $ax + by + cz = d$ ，点到平面的距离

$$D = \frac{|ax_0 + by_0 + cz_0 - d|}{\sqrt{a^2 + b^2 + c^2}}$$

- 4. 如果线性方程组对应的矩阵是奇异矩阵，则该方程组无解

作者：[我是8位的](#)

出处：<http://www.cnblogs.com/bigmonkey>

本文以学习、研究和分享为主，如需转载，请联系本人，标明作者和出处，非商业用途！

随笔

分类: [线性代数](#)

标签: [矩阵](#), [线性方程](#), [平面方程](#)

[好文要顶](#)[关注我](#)[收藏该文](#)



[我是8位的](#)
关注 - 5
粉丝 - 288
[+加关注](#)

1

推荐

0

反对

« 上一篇: [线性代数笔记4——向量3 \(叉积\)](#)
» 下一篇: [线性代数笔记6——直线和曲线的参数方程](#)

posted on 2018-01-08 01:25 [我是8位的](#) 阅读(12156) 评论(0) 编辑 收藏 举报

[刷新评论](#) [刷新页面](#) [返回顶部](#)

 登录后才能查看或发表评论，立即 [登录](#) 或者 [逛逛](#) [博客园首页](#)

【推荐】华为 HWD 2022 故事征集，分享最打动你的科技女性故事
【推荐】华为开发者专区，与开发者一起构建万物互联的智能世界

广告 X

yo zodcs.com

永中DCS_文档在线预览解决方案

兼容不同版本的Office、PDF等，特定格式的合作，支持Windows、信创环境下一键部署

打开 >

编辑推荐:

- 革命性创新，动画杀手铜 @scroll-timeline
- 戏说领域驱动设计（十二）—— 服务
- ASP.NET Core 6框架揭秘实例演示[16]：内存缓存与分布式缓存的使用
- .Net Core 中无处不在的 Async/Await 是如何提升性能的？
- 分布式系统改造方案 —— 老旧系统改造篇

#她的梦想在发光#

HWD科技女性故事有奖征集

活动时间: 2022年3月8日~3月18日



最新新闻:

- 乔布斯的创业搭档：他缺乏工程师才能，不得不锻炼营销能力来弥补
 - 美国大厂码农薪资曝光：年薪18万美元，够养家，不够买海景房
 - 两张照片就能转视频！Google提出FLIM帧插值模型
 - Android 再推“杀手级”功能，可回收 60% 存储空间
 - 溺在理财暴雷潮的投资人：本金63万，月兑25元不够卖菜
- » 更多新闻...

Powered by:
博客园

Copyright © 2022 我是8位的
Powered by .NET 6 on Kubernetes