1. **向量代数**

**1，如何分辨左手坐标系和右手坐标系？**

Direct3D采用的是左手坐标系。如果我们伸出左手，并拢手指，假设它们指向的是x轴的正方向，再弯曲四指指向y轴的正方向，则最后伸直拇指的方向大约就是z轴的正方向。右手坐标系同理。

**2，点积**

**2.1 点积的代数公式和几何公式？**

点积就是向量间对应分量的乘积之和。

点积的几何关系：u·v=|u||v|cosθ。

**2.2向量v落在向量n上的正交投影的计算公式？**

如果n是单位向量：

projn(v)=(v · n) n

perpn(v)=v – projn(v)

如果n不具有单位长度：

proj(v) = (v · n) n / |n|2

**2.3格拉姆-施密特正交化方法是什么？**

在将向量vi添加到规范正交集中时，我们需要令vi减去它在现有规范正交集中其他向量{w0,w1,…,wi-1}方向上的分量，这样就可确保新加入规范正交集的向量与该集合中的其他向量互相正交。

**3，叉积**

**3.1 叉积的代数公式和几何公式？**

W=u x v = (uyvz – uzvy, uzvx – uxvz, uxvy – uyvx)

|uxv| = |u| |v| sinθ

**3.2 叉积的方向？**

如果伸出左手，使并拢的左手手指指向向量u的方向，再以0≤θ≤π的角度弯曲四指，使之指向向量v的方向，那么最后伸直的大拇指约略指向的即为w=u x v的方向，这就是所谓的左手拇指法则。

* 1. **如何通过叉积来进行正交化处理？**

1，令w0 = v0 / |v0|;

2，w2 = w0 x v1 / |w0 x v1|

3，w1 = w2 x w0

**4，XMVECTOR**

**4.1 DirectXMath的性能优势是什么？**

借助128位宽的单指令多数据寄存器，利用一条SIMD指令即可同时对4个32位浮点数或整数进行运算。

**4.2 XMVECTOR和XMFLOATn使用规则是什么？**

1. 局部变量或全局变量用XMVECTOR类型；
2. 对于类中的数据成员，使用XMFLOAT2，XMFLOAT3和XMFLOAT4类型；
3. 在运算之前，通过加载函数将XMFLOATn类型转换为XMVECTOR类型；
4. 用XMVECTOR实例进行运算；
5. 通过存储函数将XMVECTOR类型转换为XMFLOATn类型。

**4.3 传递XMVECTOR参数的规则？**

1. 前3个XMVECTOR参数应当用类型FXMVECTOR；
2. 第4个XMVECTOR参数应当用类型GXMVECTOR；
3. 第5，6个XMVECTOR参数应当用类型HXMVECTOR；
4. 其余的XMVECTOR参数应当用类型CXMVECTOR。

在编写构造函数时，前3个XMVECTOR参数用FXMVECTOR类型，其余XMVECTOR参数则用CXMVECTOR类型，另外，对于构造函数不要使用XM\_CALLCONV注解。

**4.4 XMVECTOR类型的常量实例应当用什么类型来表示？**

XMVECTORF32