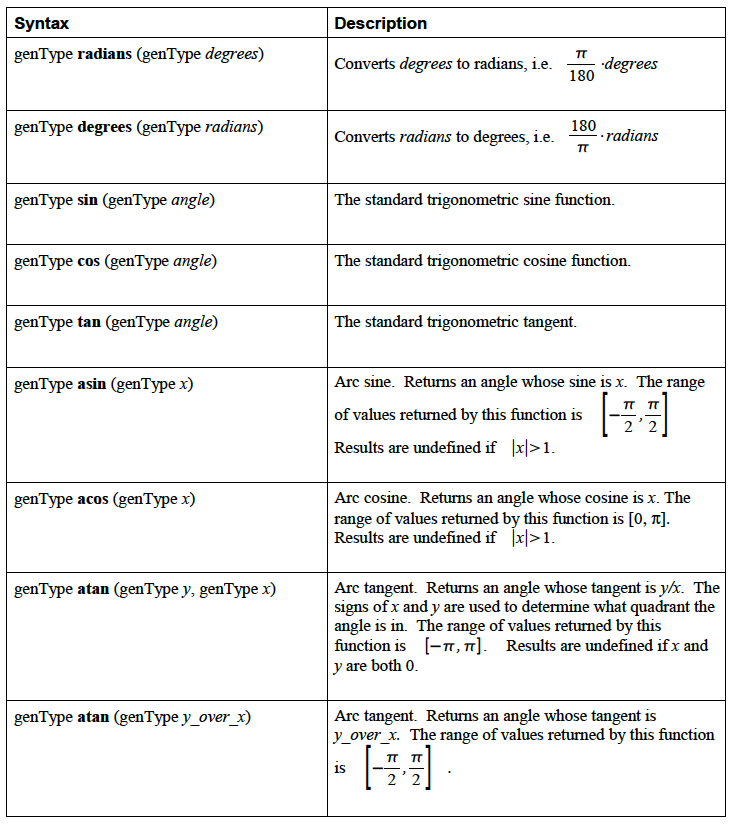
**1  角度和三角函数**

         标识为angle的函数参数假定以弧度为单位。这些函数不可能出现被0除的情况，如果被除数为0，结果是未定义的。



  radian函数是将角度转换为弧度，degrees函数是将弧度转换为角度。sin, cos, tan都是标准的三角函数。asin, acos, atan是反三角函数。genType有点像面向对象中泛型，即如果genType是float型的，那么

genType pow (genType x, genType y)

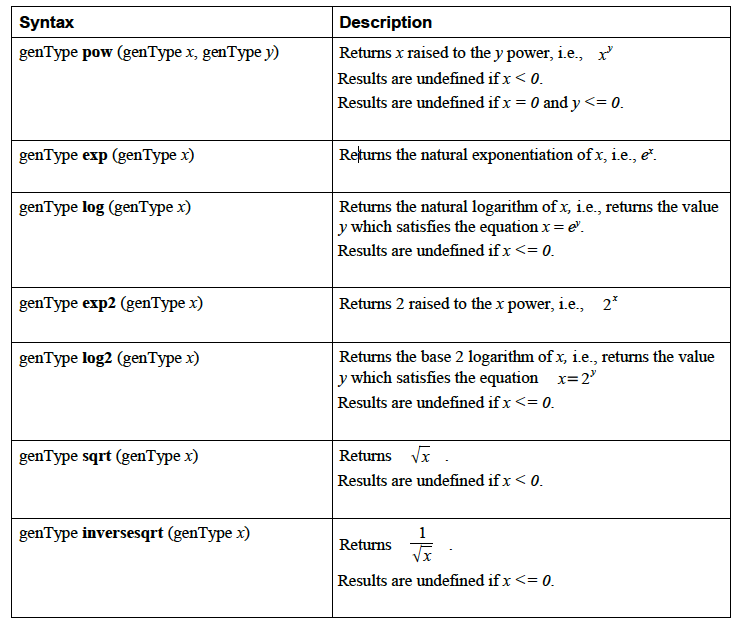
就变成了：

float pow (float x, float y)

同理，如果genType是int型的，那么就变成了：

int pow (int x, int y)；

**2  指数函数**



（1）genType pow (genType x, genType y)

          x的y次方。如果x小于0，结果是未定义的。同样，如果x=0并且y<=0,结果也是未定义的。使用时应特别注意。

（2）genType exp (genType x)

          e的x次方

（3）genType log (genType x)

          计算满足x等于e的y次方的y的值。如果x的值小于0，结果是未定义的。

（4）genType exp2 (genType x)

         计算2的x次方

（5）genType log2 (genType x)

         计算满足x等于2的y次方的y的值。如果x的值小于0，结果是未定义的。

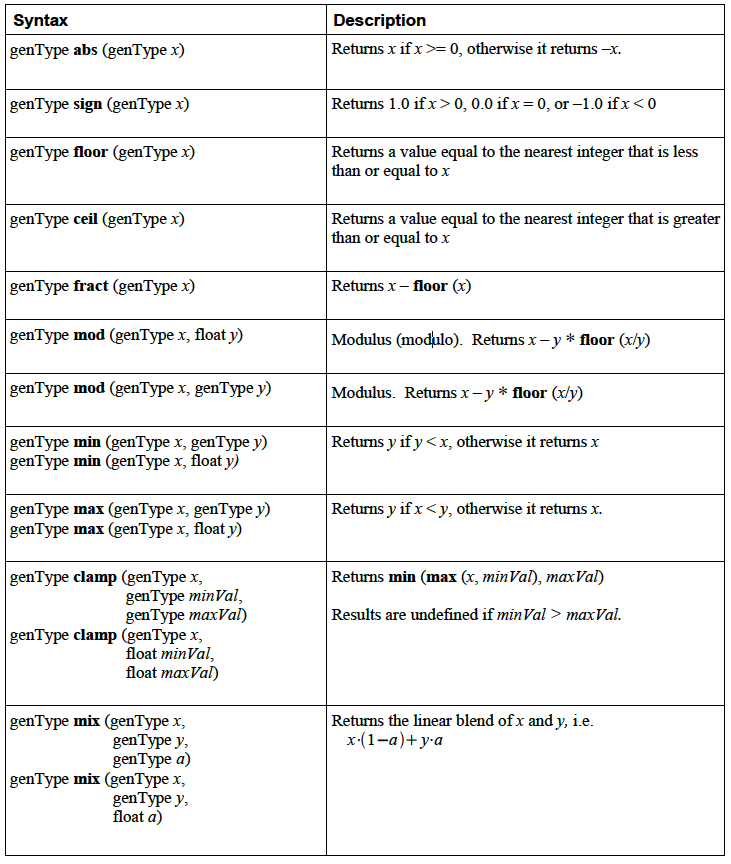
（6）genType sqrt (genType x)

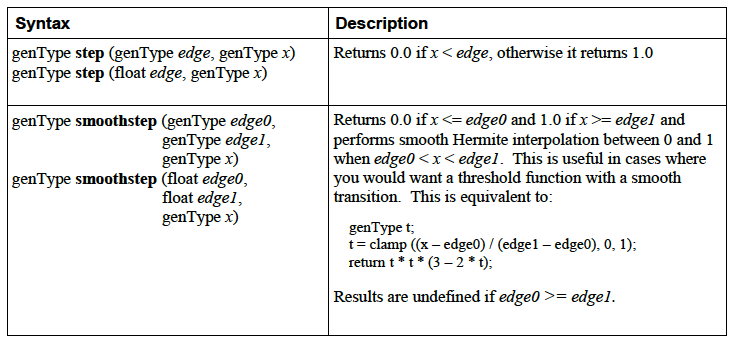
         计算x的开方。如果x小于0，结果是未定义的。

（7）genType inversesqrt (genType x)

         计算x的开方之一的值，如果x小于等于0，结果是未定义的。

**3  常用函数**





（1）genType abs (genType x)

返回x的绝对值

（2）genType sign (genType x)

如果x>0，返回1.0；如果x=0，返回0，如果x<0，返回-1.0

（3）genType floor (genType x)

返回小于等于x的最大整数值

（4）genType ceil (genType x)

返回大于等于x的最小整数值

（5）genType fract (genType x)

返回x-floor(x)，即返回x的小数部分

（6）genType mod (genType x, float y)、genType mod (genType x, genType y)

返回x – y \* floor (x/y)，即求模计算%

（7）genType min (genType x, genType y)，genType min (genType x, float y)

返回x和y的值较小的那个值。

（8）genType max (genType x, genType y)，genType max (genType x, float y)

返回x和y的值较大的那个值。

（9）genType clamp (genType x, genType minVal, genType maxVal)、genType clamp (genType x, float minVal, float maxVal)

clamp翻译为夹具，就叫夹具函数吧，这个函数是什么意思呢？看看解释的意思是：获取x和minVal之间较大的那个值，然后再拿较大的那个值和最后那个最大的值进行比较然后获取较小的那个，意思就明白了，clamp实际上是获得三个参数中大小处在中间的那个值。函数有个说明：如果minVal > minMax的话，函数返回的结果是未定的。也就是说x的值大小没有限制，但是minval的值必须比maxVal小。

（10）genType mix (genType x, genType y, genType a)、genType mix (genType x, genType y, float a)

返回线性混合的x和y，如：x⋅(1−a)+y⋅a

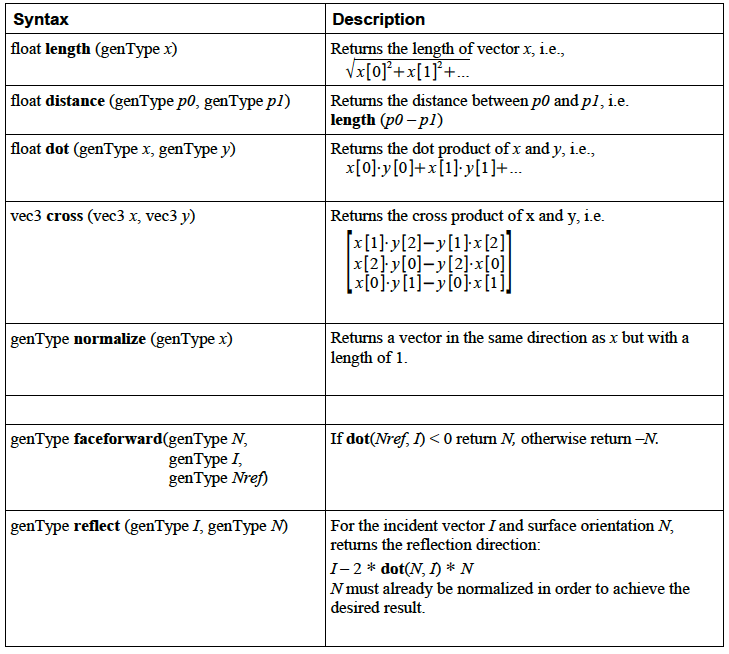
（11）genType step (genType edge, genType x)，genType step (float edge, genType x)

如果x < edge，返回0.0，否则返回1.0

（12）genType smoothstep (genType edge0,genType edge1,genType x)，genType smoothstep (float edge0,float edge1,genType x)

如果x <= edge0，返回0.0 ；如果x >= edge1 返回1.0；如果edge0 < x < edge1，则执行0~1之间的平滑埃尔米特差值。如果edge0 >= edge1，结果是未定义的。

**4  几何函数**



（1）float length (genType x)

返回向量x的长度

（2）float distance (genType p0, genType p1)

计算向量p0，p1之间的距离

（3）float dot (genType x, genType y)

向量x，y之间的点积

（4）vec3 cross (vec3 x, vec3 y)

向量x，y之间的叉积

（5）genType normalize (genType x)

标准化向量，返回一个方向和x相同但长度为1的向量

（6）genType faceforward(genType N, genType I, genType Nref)

如果Nref和I的点积小于0，返回N；否则，返回-N；

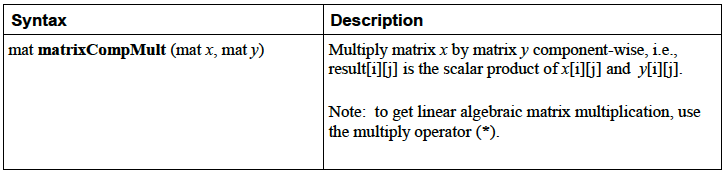
（7）genType reflect (genType I, genType N)

返回反射向量

（8）genType refract(genType I, genType N,float eta)

返回折射向量

**5  矩阵函数**

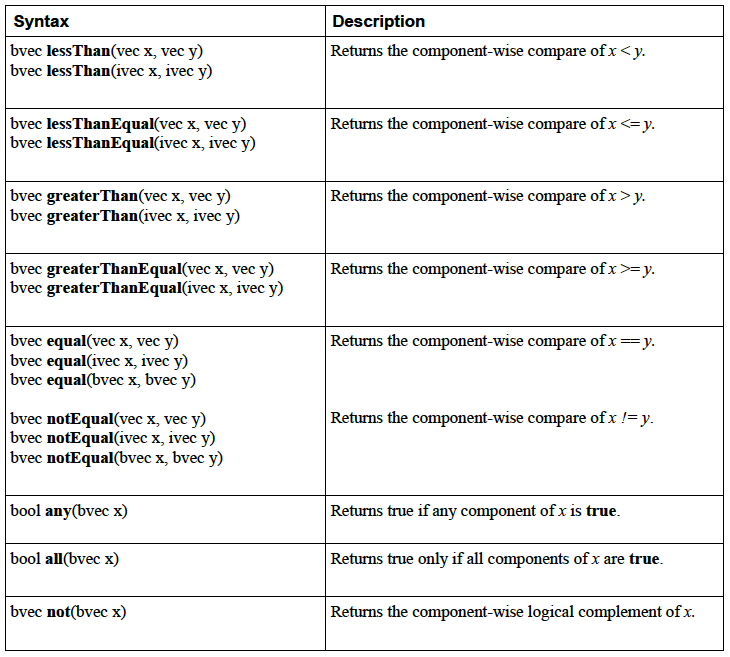


（1）mat matrixCompMult (mat x, mat y)

矩阵x乘以y，result[i][j]是 x[i][j] 和 y[i][j] 的标量积。注意，要获取线性代数矩阵的乘法，使用乘法操作符\*。

**6  向量相关函数**

       相关或相等的操作符(<, <=, >, >=, ==, !=)被定义（或保留），返回一个标量布尔值。下面, “bvec” 是表示bvec2, bvec3, or bvec4的占位符, “ivec”是ivec2, ivec3, or ivec4的占位符,  “vec” 是vec2, vec3, or vec4的占位符. 在任何情况下，输入和返回值向量的长度必须匹配。



（1）lessThan

比较x < y.

（2）lessThanEqual

比较x<=y

（3）greaterThan

比较x>y

（4）greaterThanEqual

比较x>=y

（5）equal

比较x==y

（6）notEqual

比较x!=y

（7）bool any(bvec x)

如果向量x的任何组件为true，则结果返回true。

（8）bool all(bvec x)

            如果向量x的所有组件均为true，则结果返回true。

（9）bvec not(bvec x)

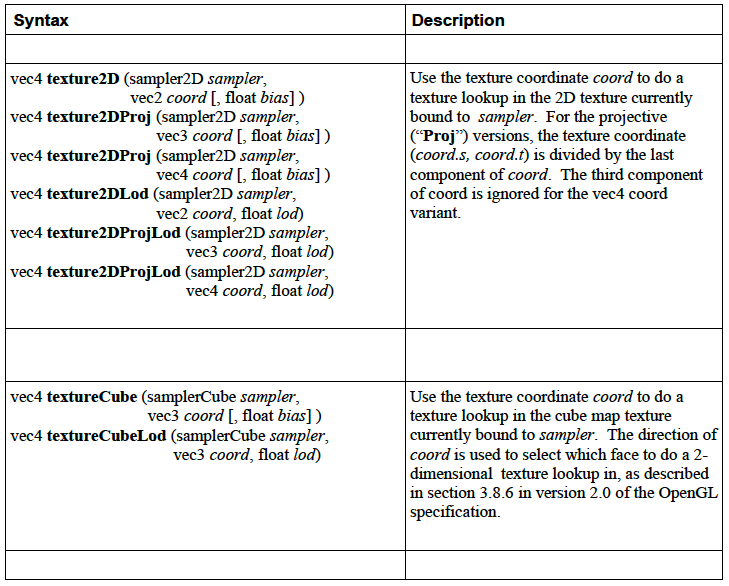
返回向量x的互补矩阵

**7  材质查找函数**

       纹理（材质）查找函数对于定点着色器和片元着色器都适用。然而，定点着色器的细节级别并不是通过固定功能计算的，所以顶点着色器和片元着色器纹理查找之间还是有一些差别的。一下函数是通过采样器访问纹理，和使用OpenGL ES API是一样的。纹理属性如尺寸，像素格式，维数，过滤方法，纹理映射匹配数，深度比较等等在OpenGL ES API中都有定义。

        在下面的函数中，bias参数对于片元着色器来说是可选的。但在定点着色器中不可使用。对于片元着色器，如果使用了bias这个参数，它被加到优先细节的计算级别中来执行纹理访问操作。如果bias没有使用，那么实现将自动选择一个默认级别。对于非纹理映射的纹理，纹理是直接被使用的。如果是纹理映射的，并且在片元着色器中执行，那么使用LOD来进行纹理查找。如果是纹理映射的，并且在顶点着色器中执行，那么使用的是基本纹理。

        以Lod结尾的内建函数只能用在顶点着色器中，在带有Lod的函数中，lod参数直接用来表示细节级别。



**8  片元处理函数**

       片元处理函数只有在片元语言中才有，但在GLSL ES中没有片元处理函数。