计算机动画原理与技术 作业 1 报告

于泽汉 No.118039910141

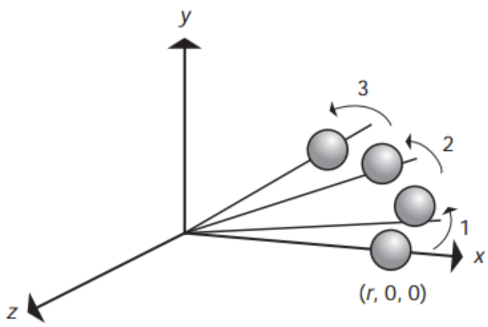
1. 旋转动画的三种方法及其舍入误差

三种方法的实现效果和对应代码见 `hw1_rotation.html`。

浏览器（推荐使用 Chrome）打开可查看效果，文本编辑器打开可查看代码。

这里为了突出效果，在产生舍入误差的地方均只保留两位小数。

方法 1



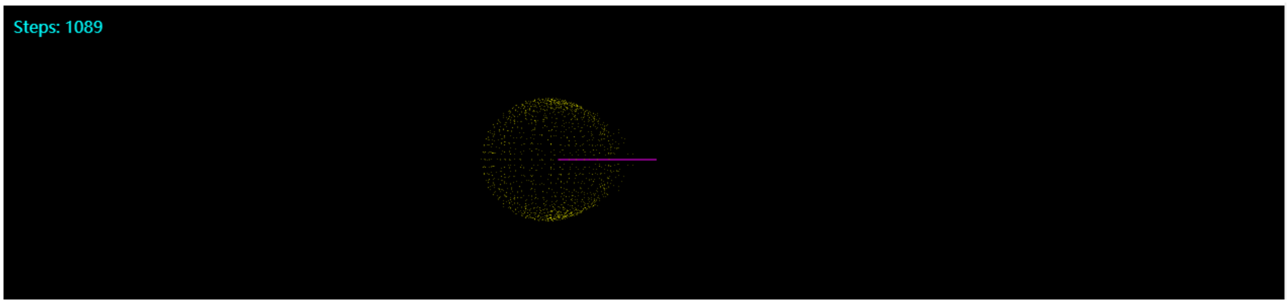

```
for each point P of the moon {
  P' = P
}
Rdz = y-axis rotation of 5 degrees
repeat until (done) {
  for each point P' of the moon {
    P' = Rdz*P'
  }
  record a frame of the animation
}
```

方法 1：每次迭代，球面上的每个点坐标，乘上一个相同的旋转固定角度的矩阵

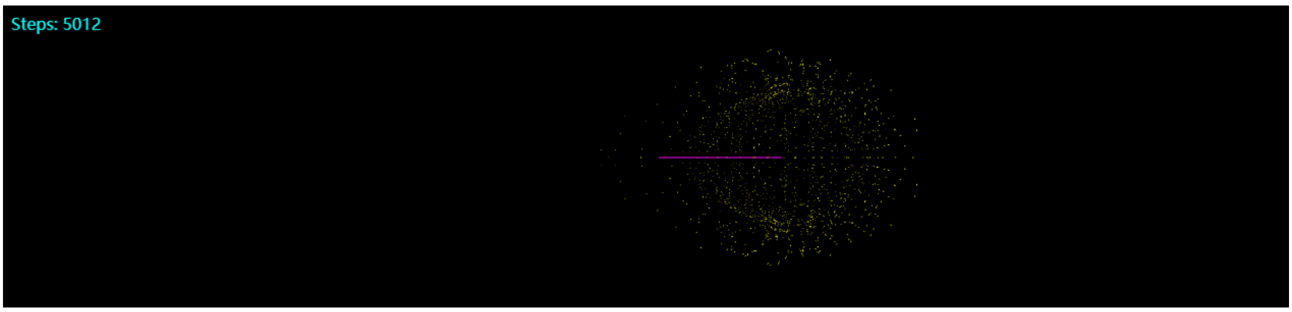
方法 1 的舍入误差会积累在球面上每个点的坐标数值中。

随着迭代次数的增多，这些点不再能形成标准的球面，而是会偏离应在的位置，并且原先在同一条经线或者纬线上的点之后也将不再共线。

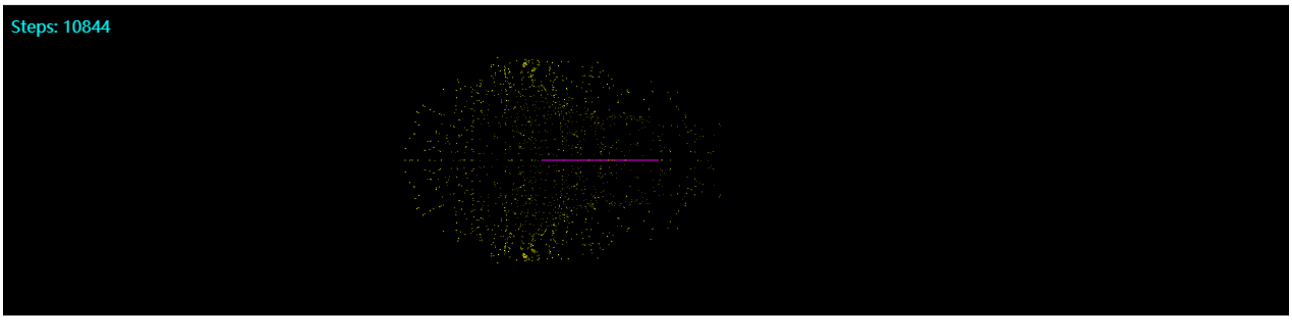
如下各图所示：



采用方法 1 迭代 1089 次



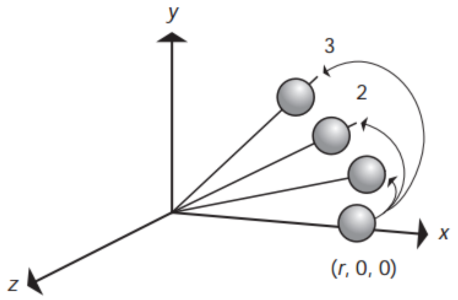
采用方法 1 迭代 5012 次



采用方法 1 迭代 10844 次

迭代次数越多，偏差的程度也就越大。

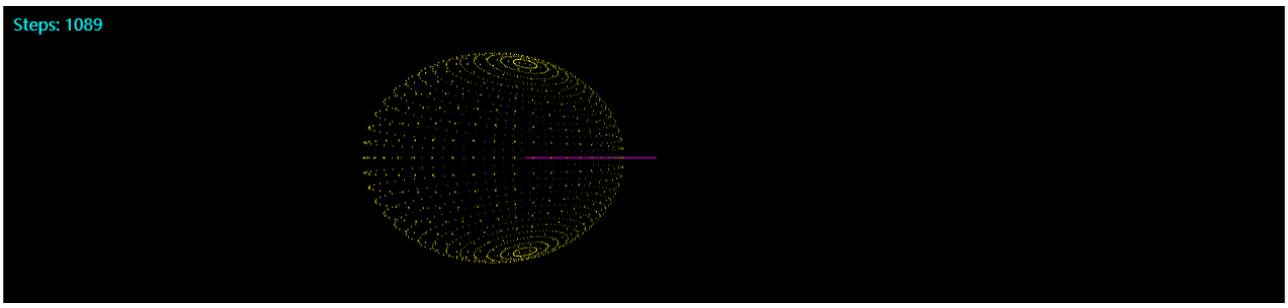
方法 2



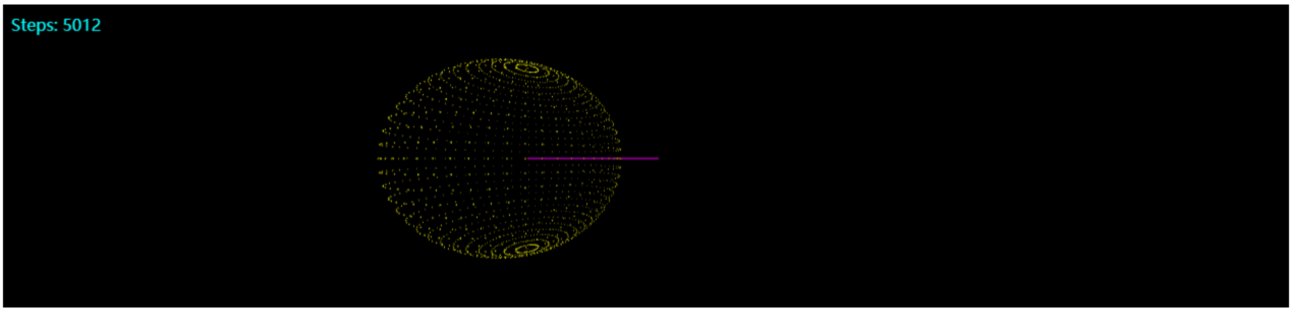

```
R = identity matrix
Rdz = z-axis rotation of 5 degrees
repeat until (done) {
  for each point P of the moon {
    P′ = R*P
  }
  record a frame of the animation
  R = R*Rdz
}
```

方法 2：每次迭代，将旋转矩阵乘上一个相同的旋转固定角度的矩阵，然后再将这个旋转矩阵作用于球面上各点

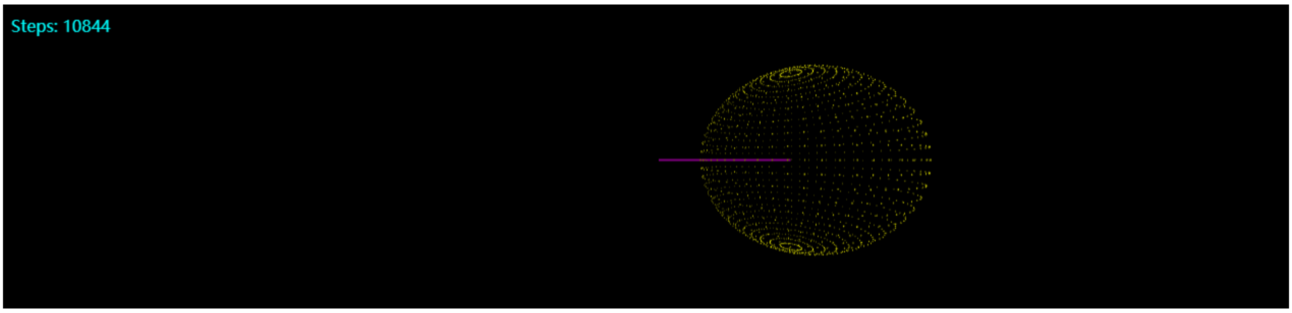
方法 2 的舍入误差会积累在变换矩阵中。
随着迭代次数的增多，变换矩阵不再是对应的旋转矩阵，因此球面上的点可能会出现一定程度的拉伸和平移。不过原先在同一条经线或纬线上的点依旧保持共线，因为虽然该矩阵不再是单纯的旋转矩阵，但是依旧作线性变换。
如下各图所示：



采用方法 2 迭代 1089 次



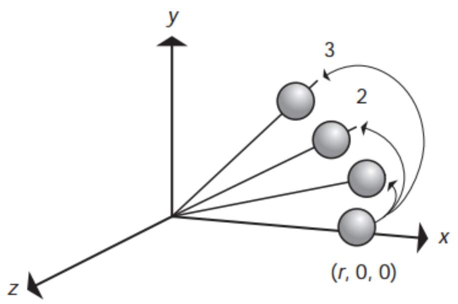
采用方法 2 迭代 5012 次



采用方法 2 迭代 10844 次

随着迭代次数的增多，偏差的程度并不像方法 1 那样明显。

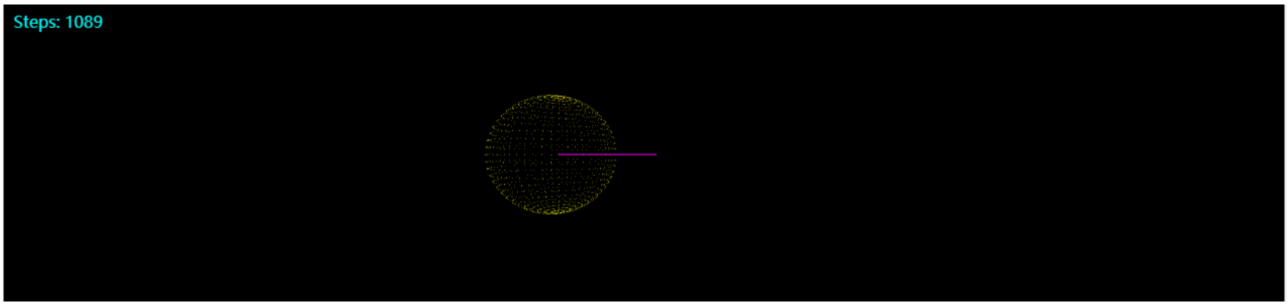
方法 3



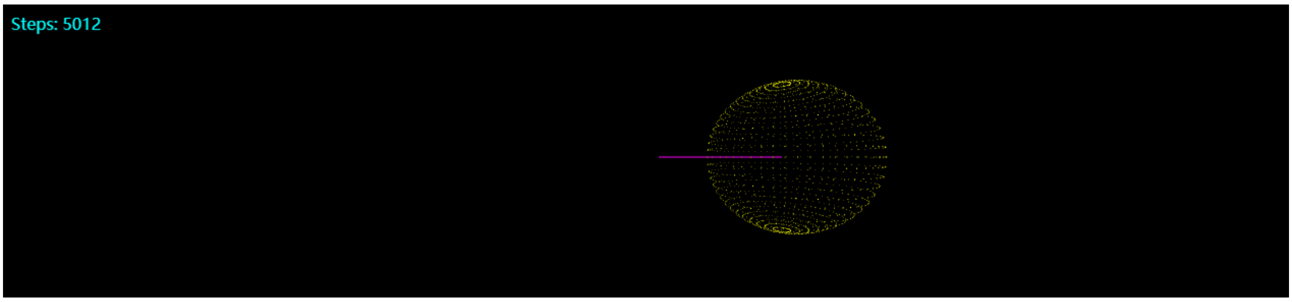

```
R = identity matrix
Rdz = z-axis rotation of 5 degrees
repeat until (done) {
  for each point P of the moon {
    P' = R*P
  }
  record a frame of the animation
  R = R*Rdz
}
```

方法 3：每次迭代，将旋转角度增加一个固定的值，再将球面上各点坐标乘上该角度对应的旋转矩阵

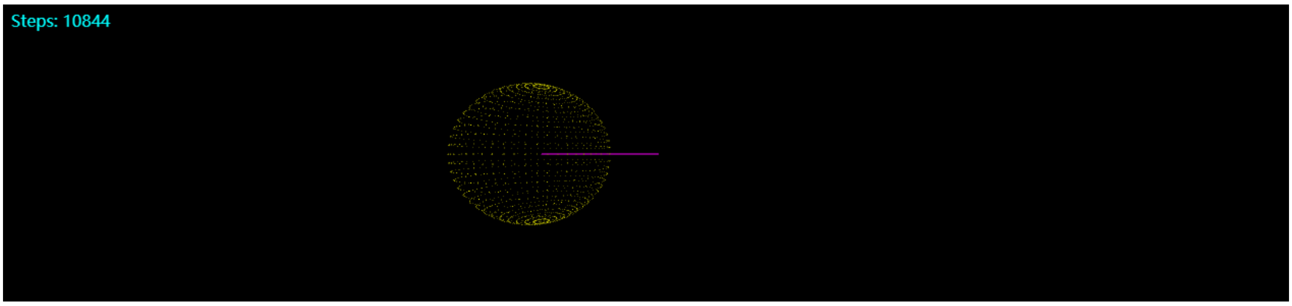
方法 3 的舍入误差会积累在旋转角度中。
随着迭代次数的增多，旋转角度出现偏差，但是依旧对应的是正常的旋转矩阵，原先在同一条经线或纬线上的点也依旧保持共线，球面上各点之间的联系没有破坏。
如下各图所示：



采用方法 3 迭代 1089 次



采用方法 3 迭代 5012 次



采用方法 3 迭代 10844 次

随着迭代次数的增多，肉眼几乎看不出偏差（因为只有角度有误差）。

2. 使用四元数实现旋转插值

旋转插值的实现效果和对应代码见 `hw1_quaternion.html`。
浏览器（推荐使用 Chrome）打开可查看效果，文本编辑器打开可查看代码。
若两个四元数分别为 q_1 和 q_2，那么对它们的插值（之后还要标准化）为：

$$slerp(q_1, q_2, t) = \frac{\sin[(1-t)\theta] \cdot q_1 + \sin[t\theta] \cdot q_2}{\sin\theta}$$

其中：θ 为 q_1 和 q_2 的夹角，可以通过公式计算得到；t 为插值系数，取值范围为 $[0, 1]$。当然，还需要考虑很多边界情况。比如，θ 较小时需要合理近似（$\sin\theta \approx \theta$）以消去接近 0 的除数；两个四元数之间角度超过 180 度时，需要对其中一个四元数取负，以使插值“不绕远路”；诸如此类，都写在代码中，这里不再赘述。