Problem 1

1.1

Name: Junior Garcia
netID: jfg388

M_1 = translation by (1, 2, 3)

$$= \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

M_2 = rotation by 90° (counterclockwise) along the x-axis

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos 90 & \sin 90 & 0 \\ 0 & -\sin 90 & \cos 90 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M_3 = M_2 M_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 3 \\ 0 & -1 & 0 & -2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

1.2

Bezier curve of degree 2 with control points:

$P_0 = (0,0)$

$P_1 = (1,0)$

$P_2 = (1,1)$

$$p(t; p_0, p_1, p_2) = (1-t)\,p(t; p_0, p_1) + t\,p(t; p_1, p_2)$$

$$= (1-t)^2 p_0 + 2(1-t)t\,p_1 + t^2 p_2 \quad \text{where } 0 \leq t \leq 1$$

Simplified:

$$(1 - 2t + t^2)p_0 + (2t - 2t^2)p_1 + t^2 p_2$$

$$x(t) = 2t - 2t^2 + t^2$$

$$= t^2 + 2t$$

$$y(t) = t^2$$

Any point in the bezier curve can be defined as

$[t^2 + 2t, t^2]$ for any t in the range $[0, 1]$

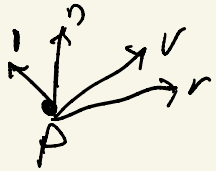
($t^2 + 2t$: x; t^2: y)

1.3

Hidden surface removal in webgl.

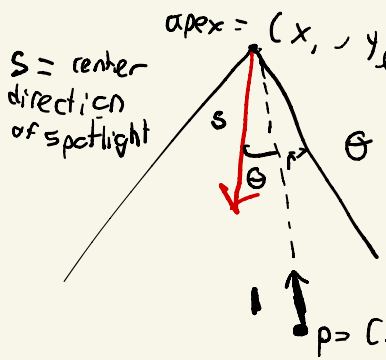
This algorithm could fail whenever there are two mesh triangles that have the same euclidean distance from the camera to the centroid of the mesh but one is actually in front of the other in terms of the z-direction. This will cause both triangles to be drawn even though one is infront of the other which will be costly and unnecesary.

1.4



$$I = k_d I_d \max(l \cdot n, 0) + k_a I_a + k_s I_s (\max(v \cdot r, 0))^{\alpha}$$

We need to change the light source vector so that it contributes light in a cone-like fashion



$\theta = l \cdot S$ = angle between l and S

$$r = \sqrt{(x_l - x)^2 + (y_l - y)^2 + (z_l - z)^2}$$ → distance from point to light source

$$\vec{l(x)} = \cos\theta \cdot \frac{1}{r} \cdot \frac{\text{apex} - \text{point}}{\| \text{apex} - \text{point} \|}$$

$$\vec{l(x)} = l \cdot S \cdot \frac{1}{r} \cdot \frac{\text{apex} - \text{point}}{\| \text{apex} - \text{point} \|}$$

$$\vec{l(x)} = \frac{l \cdot S}{r} \cdot \frac{\text{apex} - \text{point}}{\| \text{apex} - \text{point} \|}$$

Now we can substitute this l in the original Phong equation

1.9

If we increase the frustum's volume, that is, we increase the distance between the near plane and the far plane, we might lose accuracy in rendering the objects within the frustum properly. I can imagine that depth buffer test will loose accuracy for objects close to the far plane and that lightning will be compromised as the frustum distents the meshes in the scene to accomodate for large objects.

1.6

Plane equation $= 3x + 4y - z = 2$

any point $p = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$

Plane A can be written as a matrix as $\begin{bmatrix} 3 \\ 4 \\ -1 \\ 2 \end{bmatrix}$

unit normal of palne $= \frac{\hat{n}}{\|\hat{n}\|} = \frac{3, 4, 1}{\sqrt{3^2+4^2+1^2}} = < \frac{3}{\sqrt{26}}, \frac{4}{\sqrt{26}}, \frac{1}{\sqrt{26}} >$

Divide everything by $\sqrt{26}$ in the original equation to get the same plane

$$\frac{3}{\sqrt{26}}x + \frac{4}{\sqrt{26}}y - \frac{1}{\sqrt{26}}z - \frac{2}{\sqrt{26}} = 0$$

Plane A is new matrix $= \begin{bmatrix} 3/\sqrt{26} \\ 4/\sqrt{26} \\ -1/\sqrt{26} \\ -2/\sqrt{26} \end{bmatrix}$

The gquared distance of a point q to the plane is: $(A^T q)^2$

Simplified, this is $= q^T (AA^T) q$

$$AA^T = \underset{4\times 1}{\begin{bmatrix} \frac{3}{\sqrt{26}} \\ \frac{4}{\sqrt{26}} \\ \frac{-1}{\sqrt{28}} \\ \frac{-2}{\sqrt{26}} \end{bmatrix}} \times \underset{1\times 4}{\begin{bmatrix} \frac{3}{\sqrt{26}} & \frac{4}{\sqrt{26}} & \frac{-1}{\sqrt{28}} & \frac{-2}{\sqrt{26}} \end{bmatrix}}$$

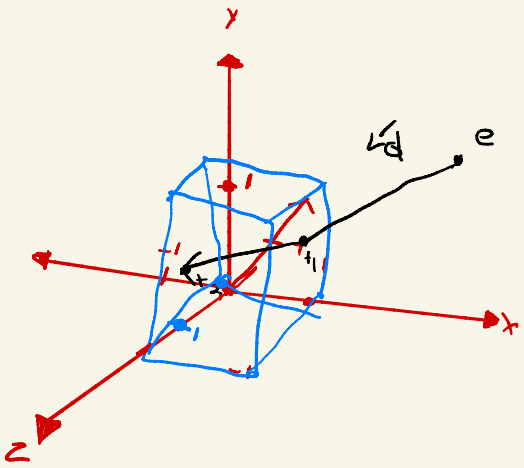
$$= \begin{bmatrix} \frac{9}{26} & \frac{6}{13} & \frac{-3}{26} & \frac{-3}{13} \\ \frac{6}{13} & \frac{8}{13} & \frac{-2}{13} & \frac{-4}{13} \\ \frac{-3}{26} & \frac{-2}{13} & \frac{1}{26} & \frac{1}{13} \\ \frac{-3}{13} & \frac{-4}{13} & \frac{1}{13} & \frac{2}{13} \end{bmatrix} = Q$$

1.7

If we have a mesh composed of a lot mesh triangles, we can speed up ray tracing by reducing the number of triangles we check for intersection. To do this, we can subdivide the mesh using bounding volume hierarchies. We first check whether the ray intersect the bounding box. If it does intersect the box, we then check whether it insects with the objects inside. That way, we reduce the number of checks we do with the ray.

Problem 2

P2.1



You want an interval $[t_1, t_2)$ that makes points emanating from ray $R(\vec{e}, \vec{d})$

$$e = \begin{bmatrix} e_x \\ e_y \\ e_z \end{bmatrix} \quad d = \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix}$$

This cuboid is composes of 6 faces. We can simplify this problem by checking when the ray intersects 1 of the planes. We also need to make sure that the ray intersects on the constrained plane, that is, it falls within the perimeter defined by the face of the cuboid. After ensuring this setup is correct, we can then generalize this procedure for all faces of the cube.

Plane $x = 1$

$e + t\hat{d} = (1, y, z)$

$$\begin{bmatrix} e_x \\ e_y \\ e_z \end{bmatrix} + t\begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} = \begin{bmatrix} 1 \\ y \\ z \end{bmatrix}$$

$e_x + td_x = 1$

$e_y + td_y = y \in [-1, 1]$

$e_z + td_z = z \in [-1, 1]$

* for t to be valid, it needs to satisfy all the conditions above. The same aplies for all constrained planes.

Plane $x = -1$

$e_x + td_x = -1.$

$e_y + td_y = y \in [-1, 1]$

$e_z + td_z = z \in [-1, 1]$

Plane $y = -1$

$e_x + td_x = x \in [-1, 1]$

$e_y + td_y = -1$

$e_z + td_z = z \in [-1, 1]$

Plane $y = 1$

$e_x + td_x = x \in [-1, 1]$

$e_y + td_y = 1$

$e_z + td_z = z \in [-1, 1]$

Plane $z = -1$

$e_x + td_x = x \in [-1, 1]$

$e_y + td_y = y \in [-1, 1]$

$e_z + td_z = -1$

Plane $z = 1$

$e_x + td_x = x \in [-1, 1]$

$e_y + td_y = y \in [-1, 1]$

$e_z + td_z = 1$

Using the method above, we can check if there exists a t such that the ray intersects one of the constrained planes. We check for all of the faces, and if we find a face that intersects, we know that one of the remaining faces will also intersect as the ray moves in a linear fashion. In order to determine t_1 and t_2, we calculate the euclidean distance between e and the two cuboid intersection points. The smaller distance defines t_1 and the longer t_2. Once we know our distances, we can calculate t_1 and t_2 as follows

$t_1 = d^{-1}(I_1 - e)$ where I_1 is the point defined by the closest intersection plane from point e

$t_2 = d^{-1}(I_2 - e)$ where I_2 is the point defined by the farthest intersection plane from point e.

If the ray does not intersect the cuboid, we can return an empty set.

Problem 2.2

Using the constrained plane aproach described in 2.1, we can further generalize the approach for any cuboid. That is, we check whether the ray intersects any of the faces of the cube. If it intersects one face, we know it must intersect at least any other face, and t_1 and t_2 are defined by calculating the ecclidean distance between e and the cuboid's intersection points. For each plane in the cuboid, which are the planes defined by $x_1, x_2, y_1, y_2, z_1, z_2$, we do the following:

x_1 case: Ray intersects face x_1 if there is a t such that

$e_x + td_x = x_1$ AND

$e_y + td_y = y \in [y_1, y_2]$ AND

$e_z + td_z = z \in [z_1, z_2]$

We repeat this procedure for all faces of the cuboid just like we did in 2.1. If the cuboid intersects two faces of the cuboid, t_1 and t_2 can be defined just as it was in 2.1:

$t_1 = d^{-1}(I_1 - e)$ where I_1 is the point defined by the closest intersection plane from point e

$t_2 = d^{-1}(I_2 - e)$ where I_2 is the point defined by the farthest intersection plane from point e.

If the ray does not intersect the cuboid, we can return an empty set

Problem 2.3

We can use an approach similar to the one defined for **the bounding hierarchies** solution. Using the procedure defined in both 2.1 and 2.2, we first check if there exist a t such that

Plane x_1:

$e_x + t d_x = x_1$ AND

$e_y + t d_y = y \in [C_{1y}, C_{1y2}]$ AND

$e_z + t d_z = z \in [C_{1z}, C_{1z}]$

After we check for all faces and find the two intersection points just like we did in 2.1 and 2.2. we then need to exclude all the points from the line segment or the ray in C_1 that are not in C_2. We do this by checking if the ray also intersects any of the faces in C_2. after we find those points in C_2, we have accumulated a set of possible intersection points that can be use to define t_1 and

t_2. T_1 will be the be defined by the intersection point in C_1 closet to e. T_2 will be defined by the 2nd closest from the possible set of points defined previously. After choosing this two points, we need to make sure that:

e/

$$t_1 = d^{-1}\left(\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} - e\right) \quad \text{such that} \quad x \in [C_1x_1, C_1x_2] \qquad x \notin [C_2x_1, C_2x_2]$$

$$t_2 = d^{-1}\left(\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} - e\right) \qquad y \in [C_1y_1, C_1y_2 \qquad y \notin [C_2y_1, C_2y_2]$$

$$z \in [C_1z_1, C_1z_2] \qquad z \notin [C_2z_1, C_2z_2]$$

If this is not satisfied, then there is no satisfactory answer and we can return an empty set

Problem 3

Rotation with respect to a point

translation by dx

$$\begin{bmatrix} \cos\theta & -\sin\theta & -a\cos\theta + b\sin\theta + a \\ \sin\theta & \cos\theta & -a\sin\theta - b\cos\theta + b \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \cos\theta & -\sin\theta & (\cos\theta \cdot dx) - a\cos\theta + b\sin\theta + a \\ \sin\theta & \cos\theta & (\sin\theta \cdot dx) - a\sin\theta - b\cos\theta + b \\ 0 & 0 & 1 \end{bmatrix}$$

$a = 0.5(1-t) + 3.5(t)$ → changing axis of rotation with respect to t

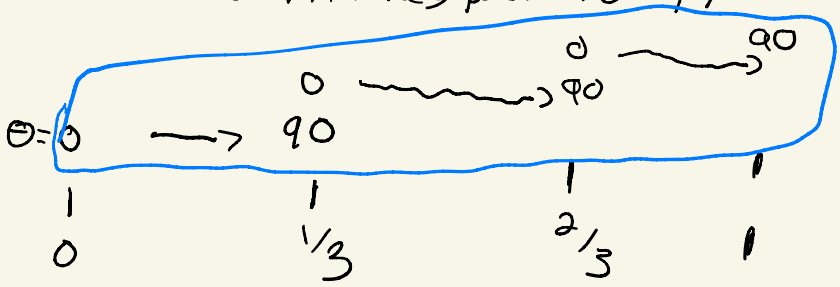
$b = 0.5$

$\theta = -2\pi t$

also changing angle with respect to t

With this approach, I am parametrizing the x coordinate and the angle of rotation as the cube moves in the x direction with respect to time t. the y

coordinate of the angle of rotation stays constant as it remains the same throughout t. Even though this result looks somewhat satisfactory, it is not the result required by the question. However, if we could somehow parametrize the angle of rotation such that it changes like below with respect to t;



and we parametrize x as:

$$x = 1(1-t) + 3(t)$$

then we could possibly achieve the desired goal.