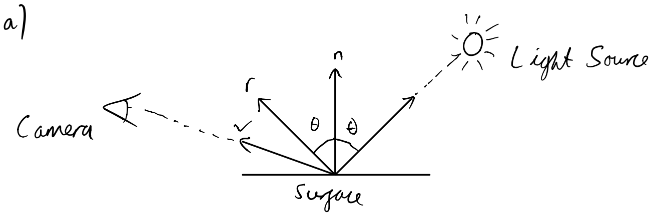
1) a)



b) Ambient term: k_A (constant everywhere, independent of light, camera or vertex position)

Diffuse term $\frac{k_d (n.C) \Phi_s}{d^2}$: Depends on light position relative to the surface normal, the smaller the angle between these vectors, the larger this term. Inversely proportional to the distance from the light source as well.

Specular term $\frac{k_s (v.r)^q \Phi_s}{d^2}$: Depends on how close the camera is to the reflection of the light. q gives the shininess of the object. Also inversely proportional to the distance from the object.

c) Blinn Phong uses halfway vectors instead to calculate the angle between the vectors.

This is advantageous because it is computationally faster to implement compared to regular Phong shading. It doesn't look quite as nice as Phong shading and the shininess coefficient must also be adjusted to achieve results which are similar.

d) Find centre of quadrilateral

$$\frac{p_1 + p_2 + p_3 + p_4}{4} = \begin{pmatrix} \frac{1.5 + 1.5 - 1.5 - 1.5}{4} \\ \frac{1.5 - 1.5 - 1.5 + 1.5}{4} \\ \frac{3 + 3 + 3 + 3}{4} \end{pmatrix}$$

$$= \begin{pmatrix} 0 \\ 0 \\ 3 \end{pmatrix}$$

Light source at $\begin{pmatrix} 0 \\ 0 \\ 4 \end{pmatrix}$

$L = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$ already normalised

$V = -\begin{pmatrix} 0 \\ 0 \\ 3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}$ when normalised

Find normal to the plane

$\overrightarrow{P_1P_2} = \begin{pmatrix} 0 \\ -3 \\ 0 \end{pmatrix}$ $\qquad$ $\overrightarrow{P_1P_3} = \begin{pmatrix} -3 \\ -3 \\ 0 \end{pmatrix}$

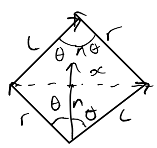
$n = \overrightarrow{P_1P_2} \times \overrightarrow{P_1P_3} = \begin{pmatrix} 0 \\ 0 \\ -9 \end{pmatrix}$

$$\begin{matrix} 0 & -3 \\ -3 & -3 \\ 0 & 0 \\ 0 & -3 \\ -3 & -3 \end{matrix}$$

$n = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}$ when normalised

Check this is in the direction of V

$n.V = 1$ +ve, so the normal is in the correct direction



$r = 2x - L$ where x is the scaler projection of L onto n

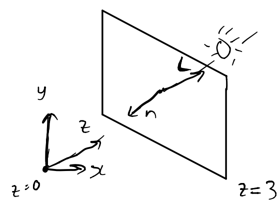
$x = n(L.n)$

$r = 2n(L.n) - L$

$r = -2\begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$

$r = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$

$l = k_a + (k_d(n.L) + k_s(V.$



Light source is the other side of the quadrilateral, only the ambient term remains

$k_a = 1$

$L = 1$

e) The vertex shader is provided with information about a vertex and can pass on this information as well as selecting a colour for the vertex.

The geometry shader can accept a primitive and modify it in some way to output a new primitive.

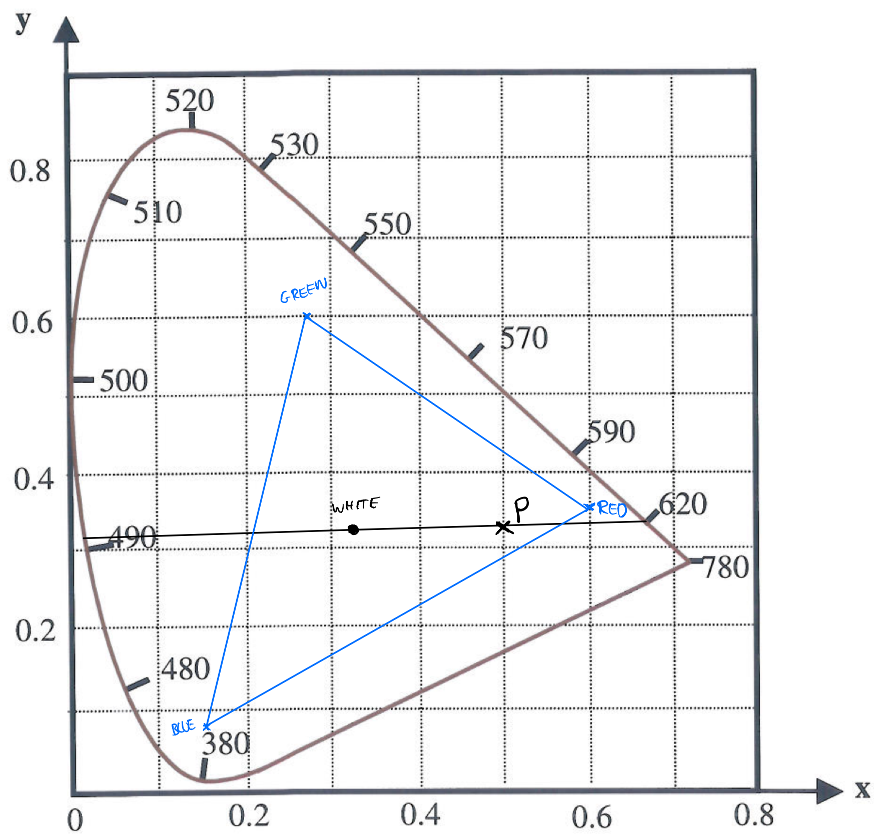
The fragment shader acts on fragments which are effectively parts of the scene which have been assigned to a pixel, it then selects the colour for these fragments.

f) i) Vertex Shader since Gouraud Shading shades vertices and then interpolates the colour.

ii) Fragment Shader since Phong Shading interpolates surface normals and then calculates the colour for each pixel.

2) a) $x = \frac{r}{r+g+b}$ $y = \frac{g}{r+g+b}$ $z = \frac{b}{r+g+b}$

$x = 0.5$ $y = 0.33$ $z = 0.33$



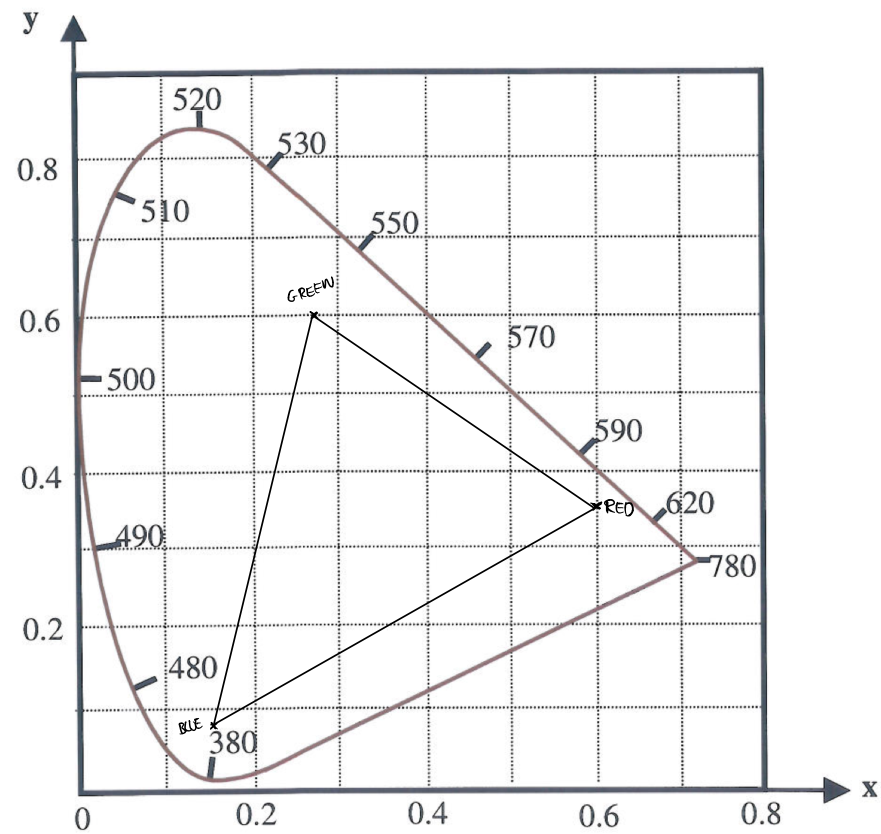
White occurs at (0.33, 0.33, 0.33)

Wavelength is roughly 620nm

c) Color occurs roughly halfway along the line so the saturation is probably about 0.5

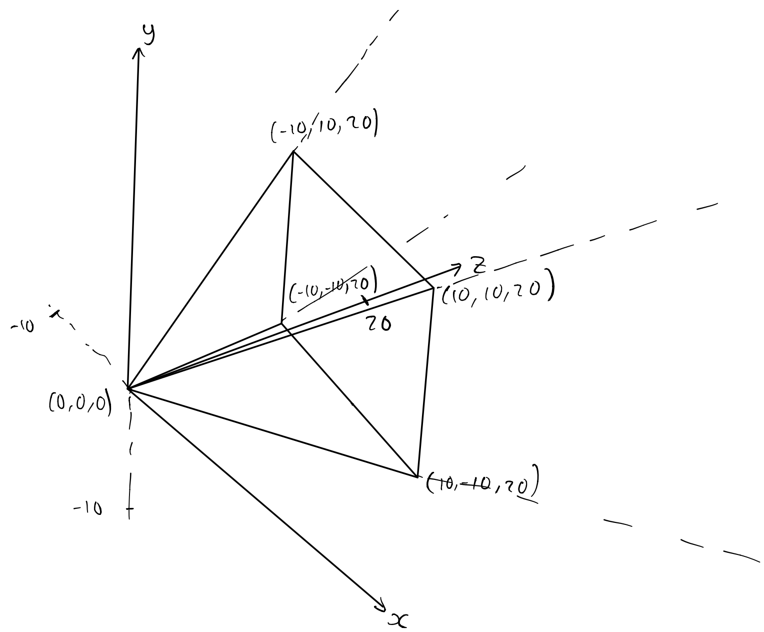
d) Complement wavelength is about 491nm

e)



f) Diagram must be convex since it represents mixing of light, you must be able to draw a line between any two points and everywhere within that line is the corresponding mix of light and must therefore be contained within the diagram.

3)



Choose a point

$p = \begin{pmatrix}0\\0\\50\end{pmatrix}$ is always in the viewing plane

Front Plane

$$\begin{array}{ccc} a & b & c \\ \begin{pmatrix}10\\-10\\20\end{pmatrix} & \begin{pmatrix}10\\10\\20\end{pmatrix} & \begin{pmatrix}-10\\10\\20\end{pmatrix}\end{array} \qquad \begin{array}{cc} \vec{ab} & \vec{ac} \\ \begin{pmatrix}0\\20\\0\end{pmatrix} & \begin{pmatrix}-20\\20\\0\end{pmatrix}\end{array}$$

$$n = \vec{ab} \times \vec{ac}$$

$$= \begin{pmatrix}0\\0\\40\end{pmatrix} \qquad = \begin{pmatrix}0\\0\\1\end{pmatrix} \text{ normalized}$$

$$\begin{array}{cc} 0 & -20 \\ 20 & 20 \\ 0 & 0 \\ 0 & -20 \\ 20 & 20 \end{array}$$

Is $(p-a)$ in the same direction as n?

$(p-a) \cdot n = 120$ +ve, so this is an inner surface normal

Back plane is parallel to front plane, with normal in the opposite direction

$$n = \begin{pmatrix}0\\0\\-1\end{pmatrix}$$

Top Plane

$$\begin{array}{ccc} a & b & c \\ \begin{pmatrix}0\\0\\0\end{pmatrix} & \begin{pmatrix}10\\10\\20\end{pmatrix} & \begin{pmatrix}-10\\10\\20\end{pmatrix}\end{array} \qquad \begin{array}{cc} \vec{ab} & \vec{ac} \\ \begin{pmatrix}10\\10\\20\end{pmatrix} & \begin{pmatrix}-10\\10\\20\end{pmatrix}\end{array}$$

$$\begin{array}{cc} 10 & -10 \\ 10 & 10 \\ 20 & 20 \\ 10 & -10 \\ 10 & 10 \end{array}$$

$(n-a)\cdot p = +ve$, correct direction

$$n = \begin{pmatrix} 0 \\ -400 \\ 200 \end{pmatrix} \qquad n = \begin{pmatrix} 0 \\ -2 \\ 1 \end{pmatrix} \frac{\sqrt{5}}{5}$$

By symmetry, the bottom plane has normal $\begin{pmatrix} 0 \\ 2 \\ 1 \end{pmatrix} \frac{\sqrt{5}}{5}$

Right Plane

$$a = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad b = \begin{pmatrix} 10 \\ 10 \\ 20 \end{pmatrix} \quad c = \begin{pmatrix} 10 \\ -10 \\ 20 \end{pmatrix} \qquad \vec{ab} = \begin{pmatrix} 10 \\ 10 \\ 20 \end{pmatrix} \quad \vec{ac} = \begin{pmatrix} 10 \\ -10 \\ 20 \end{pmatrix}$$

$$\begin{matrix} 10 & 10 \\ 10 & -10 \\ 20 & 20 \\ 10 & 10 \\ 10 & -10 \end{matrix}$$

$$n = \begin{pmatrix} 400 \\ 0 \\ -200 \end{pmatrix}$$

$(p-a)\cdot n = -ve$,
so reverse the normal

$$n = \begin{pmatrix} -2 \\ 0 \\ 1 \end{pmatrix} \frac{\sqrt{5}}{5}$$

By symmetry, left plane has normal $\begin{pmatrix} 2 \\ 0 \\ 1 \end{pmatrix} \frac{\sqrt{5}}{5}$

Plane	Normal	Homogeneous Normal
Front	$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 1 \\ -10 \end{pmatrix}$
Back	$\begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ -1 \\ 200 \end{pmatrix}$
Top	$\begin{pmatrix} 0 \\ -2 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 0 \\ -2 \\ 1 \\ 0 \end{pmatrix}$
Bottom	$\begin{pmatrix} 0 \\ 2 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 2 \\ 1 \\ 0 \end{pmatrix}$
Left	$\begin{pmatrix} 2 \\ 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 2 \\ 0 \\ 1 \\ 0 \end{pmatrix}$
Right	$\begin{pmatrix} -2 \\ 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} -2 \\ 0 \\ 1 \\ 0 \end{pmatrix}$

b) Point $\begin{pmatrix}100\\100\\210\\1\end{pmatrix}$ · Front = +ve
· Back = −ve, so it is not inside the viewing volume

$\begin{pmatrix}95\\85\\180\\1\end{pmatrix}$ · Front = +ve
· Back = +ve
· Top = +ve
· Bottom = $170+180$ = +ve
· Left = +ve
· Right = $-190+180$ = −ve, so not inside

$\begin{pmatrix}70\\70\\170\\1\end{pmatrix}$ · Front = +ve
· Back = +ve
· Top = $-140+170$ = +ve
· Right = $-140+170$ = +ve
· Left = +ve
· Bottom = +ve

So it is inside the plane

$\begin{pmatrix}70\\90\\160\end{pmatrix}$ · Front = +ve
· Back = +ve
· Top = $-180+160$ = −ve So not inside

Only the point $\begin{pmatrix}70\\70\\170\end{pmatrix}$ is inside the viewing volume

d) Projecting, since z, divide x and y

$$\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&\frac{1}{\delta}&0\end{pmatrix}\begin{pmatrix}x\\y\\z\\1\end{pmatrix} = \begin{pmatrix}x\\y\\z\\z/\delta\end{pmatrix}$$

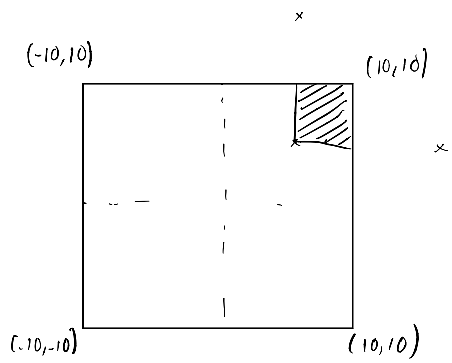
$$= \begin{pmatrix}\frac{x\delta}{z}\\\frac{y\delta}{z}\\\delta\end{pmatrix}$$

Multiply by focal length (20) and divide by z coordinate

$$\begin{pmatrix}10\\10\\40\end{pmatrix} \Rightarrow \begin{pmatrix}5\\5\\20\end{pmatrix}$$

$$\begin{pmatrix}10\\30\\40\end{pmatrix} \Rightarrow \begin{pmatrix}5\\15\\20\end{pmatrix}$$

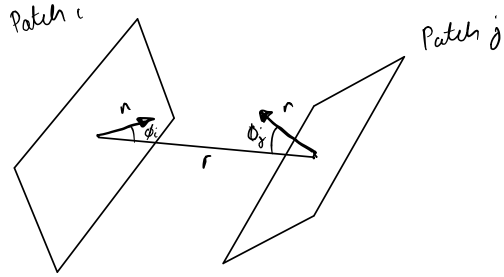
$$\begin{pmatrix}30\\10\\40\end{pmatrix} \Rightarrow \begin{pmatrix}15\\5\\20\end{pmatrix}$$



e) Clipping is much faster to calculate in 2D, since it is just a range check. However, this may mean more computation is used to project objects which would never be seen in the viewing plane.

In general, if most objects are visible, clipping in 2D is more efficient, if most objects are not visible, clipping in 3D will be more efficient.

4) a)



The form factors capture a lot of information about the two patches. The angles between the normals of the patches represent how much the patches face each other. A_j is the area of patch j, because if it is a larger patch, more light can be emitted from it. r is the distance between the two patches, and because light spreads out as a sphere, πr^2 represents the surface area of this sphere at patch i

b) In the full matrix solution, the system of equations is solved in its entirety to calculate the illumination of each patch.

In progressive refinement, numerical methods are used to iteratively get closer to the solution of the system of equations.

Progressive refinement may be used to a certain level if changes need to be made to the scene regularly, since it is computationally much faster. The full matrix solution may be used when a final, perfect looking result is required.

c) This equation does not take into account the position of the camera, so it would be impossible to calculate specular reflections.

d) Patch i centroid: $\begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}$

Patch j centroid $\begin{pmatrix} 5 \\ 1 \\ 4 \end{pmatrix}$

Patch i normal: $n_i = \begin{pmatrix} 3 \\ 0 \\ 0 \end{pmatrix} \times \begin{pmatrix} 0 \\ 3 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 9 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$

3 0
0 3
0 0
3 0
0 3

Patch j normal:

$$n_j = \begin{pmatrix}0\\0\\3\end{pmatrix} \times \begin{pmatrix}0\\3\\3\end{pmatrix} = \begin{pmatrix}-1\\0\\0\end{pmatrix}$$

$$\begin{array}{cc} 0 & 0 \\ 0 & 3 \\ 3 & 3 \\ 0 & 0 \\ 0 & 3 \end{array}$$

$$r = |c_j - c_i| = \left\| \begin{pmatrix}3\\0\\3\end{pmatrix} \right\| = 3\sqrt{2}$$

$$r^2 = 18$$

$$A_i = \left\| \begin{pmatrix}3\\0\\0\end{pmatrix} \times \begin{pmatrix}0\\3\\0\end{pmatrix} \right\| \frac{1}{2} = 4.5$$

$$A_j = \left\| \begin{pmatrix}0\\0\\3\end{pmatrix} \times \begin{pmatrix}0\\3\\3\end{pmatrix} \right\| \frac{1}{2} = 4.5$$

$$a \cdot b = |a||b| \cos\phi$$

$$\cos\phi_i = \frac{(c_j - c_i) \cdot n_i}{r}$$

$$= \frac{3}{3\sqrt{2}} = \frac{\sqrt{2}}{2}$$

$$\cos\phi_j = \frac{(c_i - c_j) \cdot n_j}{r} = \frac{3}{3\sqrt{2}} = \frac{\sqrt{2}}{2}$$

$$F_{ij} = \frac{0.5\,|A_j|}{\pi r^2} = 0.0398$$

$$F_{ji} = \frac{0.5\,|A_i|}{\pi r^2} = 0.0398$$

e) The hemicube method involves placing a hemicube over the patch i for which form factors F_{ij} are to be calculated. The hemicube is divided into pixels and then a delta form factor calculated for the pixels on the sides and on the top of the cube. A patch j can then be projected onto the hemicube and the form factor F_{ij} is the sum of the delta form factors of the pixels occupied by the projection.