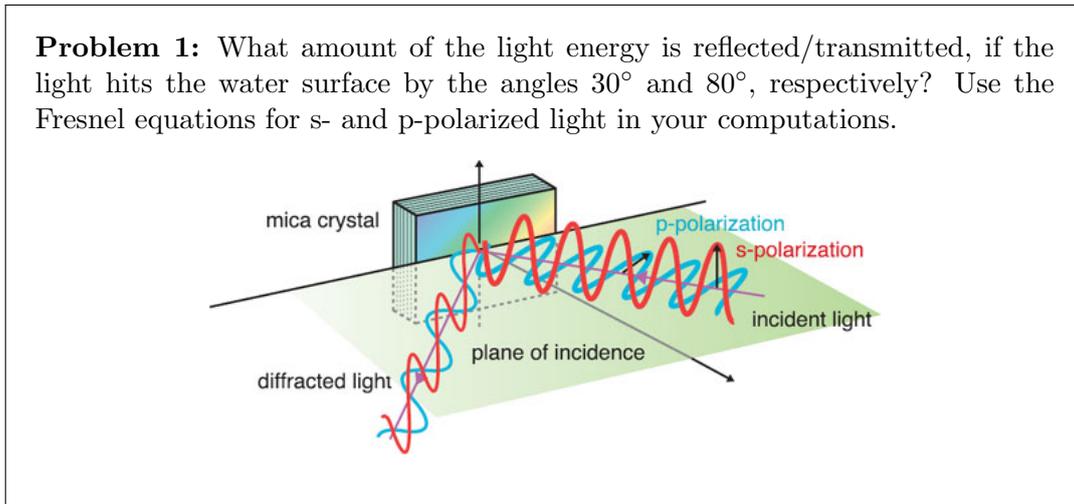


Problems to solve during the (out)break

PART 1 – STEP-BY-STEP SOLUTIONS

Problem 1: What amount of the light energy is reflected/transmitted, if the light hits the water surface by the angles 30° and 80° , respectively? Use the Fresnel equations for s- and p-polarized light in your computations.



The amount of reflected (or transmitted) light energy depends on the viewing angle with respect to the surface. In real life, we encounter this phenomenon mainly if we are looking at the water surface.

If we look at the surface from above, we see the fish swimming below. The reason is, that the larger amount of light is transmitted than reflected. Similarly, if we look at the surface from a side, we may see the reflection of the ambient nature, because the larger amount of incoming light is reflected, rather than transmitted below the surface.

This phenomenon is described by Fresnel equations, which says that the amount of the reflectance R and transmittance T depends on the polarization of the incoming ray. As seen in the figure above, the ray is decomposed into the polarized rays:

- **s-polarization** – the vector perpendicular to the plane of incidence (from German *senkrecht*),
- **p-polarization** – the vector lying the plane of incidence (parallel).

Now consider the angle of incidence (i. e. the angle of reflection) $\theta_i = 30^\circ$. From Snell's law we immediately get that the angle of transmittance $\theta_t \approx 22.083^\circ$, since refractive indices for air and water are $\eta_1 = 1.00026$ and $\eta_2 = 1.33$, respectively.

Now the amount of the (overall) reflectance R is computed as the average of the s-reflectance R_s and p-reflectance R_p . These are given by

$$\begin{aligned} R_s &= \left| \frac{\eta_1 \cos \theta_i - \eta_2 \cos \theta_t}{\eta_1 \cos \theta_i + \eta_2 \cos \theta_t} \right|^2 \approx \left| \frac{0.866 - 1.232}{0.866 + 1.232} \right|^2 \approx 3.043\%, \\ R_p &= \left| \frac{\eta_1 \cos \theta_t - \eta_2 \cos \theta_i}{\eta_1 \cos \theta_t + \eta_2 \cos \theta_i} \right|^2 \approx \left| \frac{0.927 - 1.152}{0.927 + 1.152} \right|^2 \approx 1.171\%, \end{aligned} \quad (1)$$

thus the overall reflectance is $R = \frac{R_s + R_p}{2} \approx 2.107\%$.

The s-transmittance and p-transmittance are computed as $T_s = 1 - R_s$ and $T_p = 1 - R_p$, respectively, since the law of the conservation of energy holds. Thus the overall transmittance is computed again as an average

$$T = \frac{T_s + T_p}{2} = \frac{(1 - R_s) + (1 - R_p)}{2} = 1 - \frac{R_s + R_p}{2} = 1 - R \approx 97.983\%. \quad (2)$$

The obtained numbers correspond with the interpretation, that in this case we see almost clearly the life below the water surface, since only about 2% of the light energy is reflected.

The case of $\theta_i = 80^\circ$ is computed analogously and yields $R \approx 34.65\%$.

Problem 2: Consider the scene with the golden ball and the point lamp. The viewer stands in the point $V = (-1, 0, 5)^\top$, and the position of the lamp is $L = (10, 0, 20)^\top$. The color of the light is $(1, 1, 1)$, and the ambient coefficient is 5%. Gold is characterized by the following values:

$$k_a = (0.247, 0.199, 0.075),$$

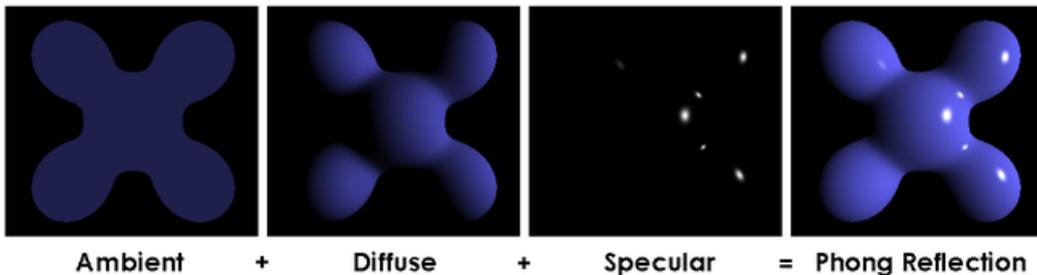
$$k_d = (0.752, 0.606, 0.226),$$

$$k_s = (0.628, 0.556, 0.366),$$

$$n_s = 51.2.$$

What color is seen in the point $P = (0, 0, 1)^\top$ with the normal vector $\mathbf{n} = (0, 0, 1)^\top$, if we use the Phong illumination model?

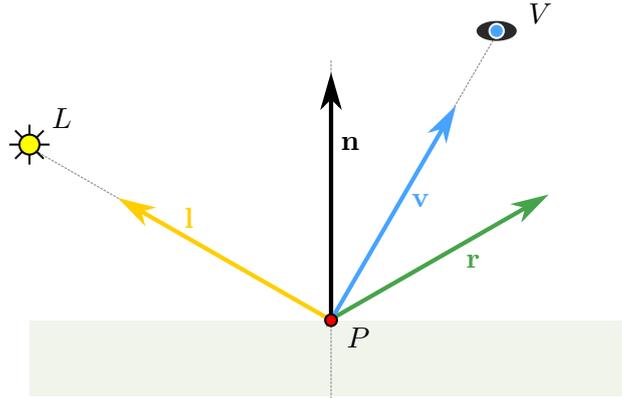
The Phong illumination model mixes the final light intensity from three separate parts – **ambient**, **diffuse** and **specular**. Although this model is not physically based, its main advantage is the presence of specular reflections on a surface (in contrary to the Lambertian reflectance).



A material is characterized by quadruplet (k_a, k_d, k_s, n_s) , where k_a, k_d, k_s are ambient, diffuse and specular colors (given by RGB values scaled to unit interval), respectively and shininess (scalar). Similarly the ambient, diffuse and specular color may be assigned to the light source, these colors are denoted by I_a, I_d and I_s , respectively. Then the final intensity at the certain point is computed as

$$I = I_{amb} + I_{diff} + I_{spec}. \quad (3)$$

The ambient part $I_{amb} = k_a I_a$ is just the addition to the final intensity (it is independent on the position of the light and viewer).



The diffuse part $I_{diff} = k_d I_d (\mathbf{l} \cdot \mathbf{n})$ depends on the position of the light (i. e. the vector of the incoming ray, see the figure below).

The specular part $I_{spec} = k_s I_s (\mathbf{r} \cdot \mathbf{v})^{n_s}$ is dependent on the position of the viewer.

To sum up, the intensity I is computed as

$$I = k_a I_a + k_d I_d (\mathbf{l} \cdot \mathbf{n}) + k_s I_s (\mathbf{r} \cdot \mathbf{v})^{n_s}. \quad (4)$$

In our problem, the parameters of the material – gold – are given.

In practice the diffuse and specular color are often identified, i. e. $I_d = I_s = (1, 1, 1)$. The ambient color of the light is obtained by scaling the color by the ambient coefficient, i. e. $I_a = 0.05(1, 1, 1) = (0.05, 0.05, 0.05)$.

Now, we proceed to the computation of the vectors of the incoming ray \mathbf{l} , the viewing vector \mathbf{v} and the reflected vector \mathbf{r} . Keep in mind, that all vectors need to be unit:

$$\begin{aligned} \mathbf{l} &= \frac{L - P}{\|L - P\|} \approx (0.466, 0, 0.885), \\ \mathbf{v} &= \frac{V - P}{\|V - P\|} \approx (-0.243, 0, 0.97), \\ \mathbf{r} &= 2(\mathbf{l} \cdot \mathbf{n})\mathbf{n} - \mathbf{l} \approx (-0.466, 0, 0.885), \end{aligned} \quad (5)$$

where \mathbf{r} is obtained using the formula proved in the problem 2 of the previous part.

Now plugging the computed values into the equation (4) we obtain

$$\begin{aligned} I &= k_a I_a + k_d I_d (\mathbf{l} \cdot \mathbf{n}) + k_s I_s (\mathbf{r} \cdot \mathbf{v})^{n_s} \\ &= (0.247, 0.199, 0.075)(0.05, 0.05, 0.05) + (0.752, 0.606, 0.226)(1, 1, 1) 0.885 \\ &\quad + (0.628, 0.556, 0.366)(1, 1, 1) 0.97^{51.2} \\ &= (0.012, 0.001, 0) + (0.666, 0.536, 0.2) + (0.132, 0.117, 0.077) \\ &= (0.81, 0.654, 0.277) \approx RGB(207, 167, 71). \end{aligned} \quad (6)$$

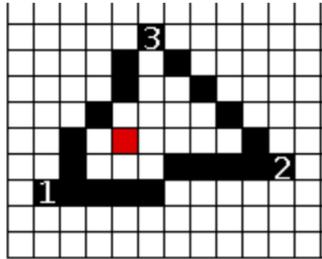
Problem 3: Consider a pyramid $V_0V_1V_2V_3$, where $V_0 = (0, 0, 0)$, $V_1 = (0, 1, 0)$, $V_2 = (1, 0, 0)$, $V_3 = (0, 0, 1)$.

1. Determine the normal vectors in the vertices V_0, V_1, V_2, V_3 .
2. Using the Phong illumination model, the colors of the vertices were calculated as follows:

$$c(V_1) = (0.6, 0.1, 0), \quad c(V_2) = (0.6, 0.2, 0), \quad c(V_3) = (0.7, 0.15, 0).$$

Calculate the color of the pixel (4, 4) on the display of the triangle $V'_1V'_2V'_3$ (see the figure), if it is obtained by the projection of the pyramid onto the camera and the Gouraud shading is used.

3. How would have been the color computed, if the Phong shading was used? Compute the normal vector in the pixel (4, 4), which is used in the color calculation using the Phong shading.



1. Our aim is to obtain vertex normals which reflect the shape of our object – a pyramid. Since we are working with a polyhedron, the vertex normals may be obtained from the face normals.

Since, three of the faces lie in the coordinate planes, the face normals are obvious:

$$\begin{aligned} \Delta V_1V_2V_3 &\rightsquigarrow \mathbf{n}_0 = (1, 1, 1), & \Delta V_0V_2V_3 &\rightsquigarrow \mathbf{n}_1 = (0, -1, 0), \\ \Delta V_0V_1V_3 &\rightsquigarrow \mathbf{n}_2 = (-1, 0, 0), & \Delta V_0V_1V_2 &\rightsquigarrow \mathbf{n}_3 = (0, 0, -1). \end{aligned} \quad (7)$$

Now we may proceed to the computation of the vertex normals by simple averaging of the normals of the faces adjacent to the vertex:

$$\begin{aligned} \mathbf{n}_{V_0} &= \frac{\mathbf{n}_1 + \mathbf{n}_2 + \mathbf{n}_3}{3} = (-1/3, -1/3, -1/3), \\ \mathbf{n}_{V_1} &= \frac{\mathbf{n}_0 + \mathbf{n}_2 + \mathbf{n}_3}{3} = (0, 1/3, 0), \\ \mathbf{n}_{V_2} &= \frac{\mathbf{n}_0 + \mathbf{n}_1 + \mathbf{n}_3}{3} = (1/3, 0, 0), \\ \mathbf{n}_{V_3} &= \frac{\mathbf{n}_0 + \mathbf{n}_1 + \mathbf{n}_2}{3} = (0, 0, 1/3). \end{aligned} \quad (8)$$

Note that the direction of the vertex normals are heavily dependent on the lengths of the face normals. The direction of the vector is unique, since we always consider outwards pointing normals.

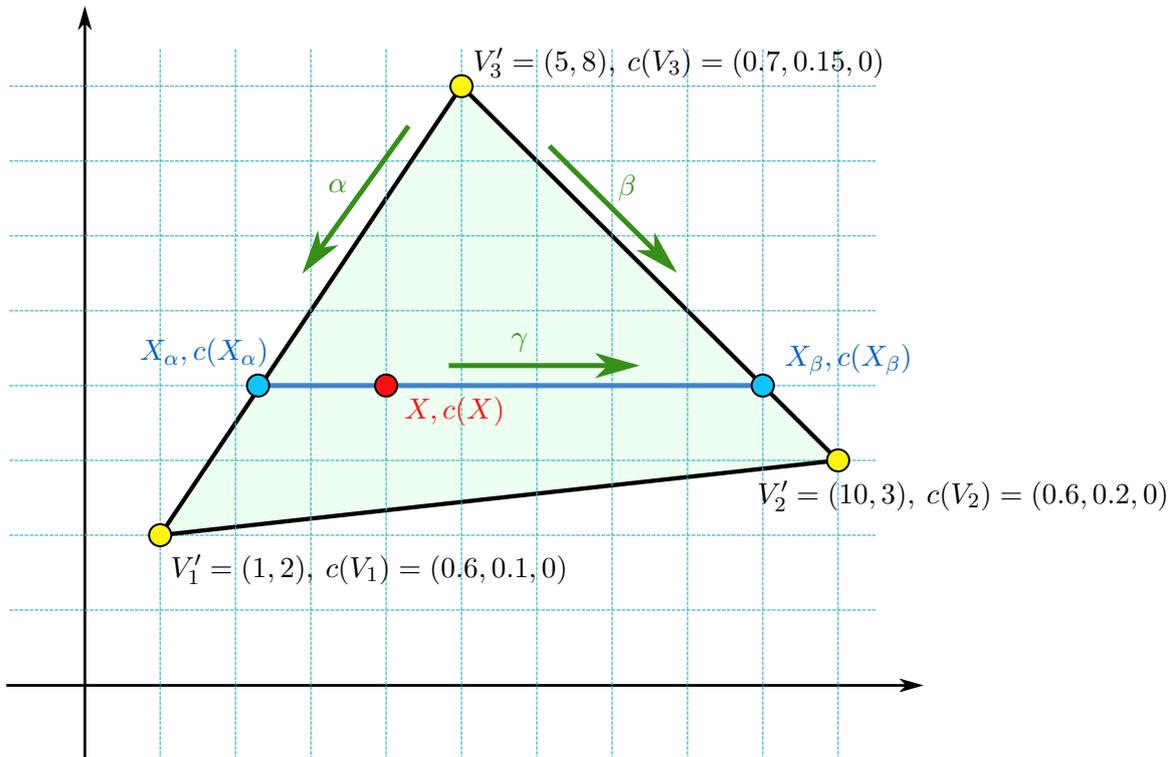
In the equation (7) we see, that the vector \mathbf{n}_0 is the only non-unit vector. How would the result change if we normalized \mathbf{n}_0 ?

There is no unique way of computation of the vertex normals from the face normals. The more advanced ways of computing the vertex normals are:

- **mean weighted equally** $\mathbf{n} = \sum_{i=1}^n \frac{\mathbf{n}_i}{\|\mathbf{n}_i\|}$ – average computed from normalized face normals,
- **mean weighted by lengths** $\mathbf{n} = \frac{1}{n} \sum_{i=1}^n \mathbf{n}_i$ – average reflecting the lengths of the face normals (used in our solution),
- **mean weighted by angles** $\mathbf{n} = \sum_{i=1}^n \sin \alpha_i \frac{\mathbf{n}_i}{\|\mathbf{n}_i\|}$, where α_i is the angle of the polygon near the vertex.

Note that the computed vertex normals are not automatically unit. We always need to normalize them if we are going to use them in the Phong shading.

2. The situation is depicted in the following figure: To obtain the color of the point X



we will use the bilinear interpolation.

In the first step we construct a horizontal line passing through the point X , which intersects the triangle in the points X_α and X_β . Now, the point X may be expressed as

$$X = (1 - \gamma)X_\alpha + \gamma X_\beta, \quad (9)$$

which is actually the first linear interpolation. In the same fashion we express the points X_α and X_β (the second linear interpolation):

$$\begin{aligned} X_\alpha &= (1 - \alpha)V'_3 + \alpha V'_1, \\ X_\beta &= (1 - \beta)V'_3 + \beta V'_2. \end{aligned} \quad (10)$$

By plugging the equations (10) into the equation (9) we get

$$X = (1 - \gamma)((1 - \alpha)V'_3 + \alpha V'_1) + \gamma((1 - \beta)V'_3 + \beta V'_2). \quad (11)$$

Since we are using the Gouraud shading, i. e. we need to apply the bilinear interpolation to the colors, so we use the same parameters α, β and γ :

$$c(X) = (1 - \gamma)((1 - \alpha)c(V'_3) + \alpha c(V'_1)) + \gamma((1 - \beta)c(V'_3) + \beta c(V'_2)). \quad (12)$$

These parameters are unknown and our task is to determine them.

We start with the determination of the parameters α and β . Since the points X_α and X_β lie on the line $y = 4$, we need to compute their x -coordinates and as a by-product we get also the parameters α and β . This is done by plugging the y -coordinates into the equation (10):

$$\begin{aligned} 4 &= (1 - \alpha)8 + \alpha 2 &\rightsquigarrow &\alpha = 2/3 &\rightsquigarrow &X_\alpha = (7/3, 4), \\ 4 &= (1 - \beta)8 + \beta 3 &\rightsquigarrow &\beta = 4/5 &\rightsquigarrow &X_\beta = (9, 4). \end{aligned} \quad (13)$$

Similarly, we obtain the parameter γ by plugging the x -coordinates of the respective points into the equation (9):

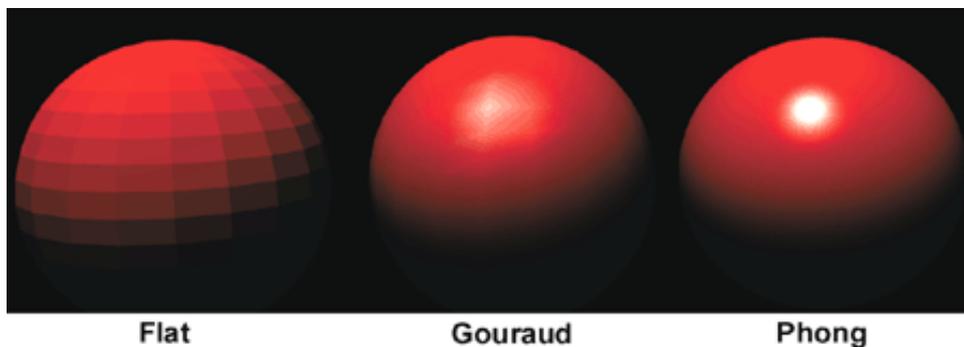
$$4 = (1 - \gamma)7/3 + \gamma 9 \quad \rightsquigarrow \quad \gamma = 1/4. \quad (14)$$

Finally, we insert the values α, β and γ into the equation (12) to obtain the color of the point X and we get $c(X) = (0.63, 0.11, 0) = RGB(161, 28, 0)$.

3. In the Gouraud shading we firstly compute the color of the vertices of the polygon (e. g. using by Phong or Lambertian illumination model) and then we apply the bilinear interpolation to the colors.

However, in the case of Phong shading (not to be confused with Phong illumination), we firstly interpolate the vertex normal of the polygon and then compute the color in the given point and its computed normal.

The main difference between Gouraud and Phong shading is the rendering of the specular reflections (as seen in the figure below), while Phong shading produces more realistic ones.



The bilinear interpolation of normals in our case is

$$\mathbf{n}_X = (1 - \gamma)((1 - \alpha)\mathbf{n}'_3 + \alpha\mathbf{n}'_1) + \gamma((1 - \beta)\mathbf{n}'_3 + \beta\mathbf{n}'_2), \quad (15)$$

where the vectors $\mathbf{n}'_1, \mathbf{n}'_2$ and \mathbf{n}'_3 are obtained by normalization (since we need unit vectors) of the vectors $\mathbf{n}_{V_1}, \mathbf{n}_{V_2}$ and \mathbf{n}_{V_3} , respectively.

By plugging the parameters α, β and γ from the previous part into the equation (15) we get, that $\mathbf{n}_X = (1/5, 1/2, 3/10)$. To compute the color of the point X we use the normalized vector $\mathbf{n}_X / \|\mathbf{n}_X\|$.