**Developing Shading Functions for Surfaces using Mathematica**

by

Randy Westhoff
Bemidji State University

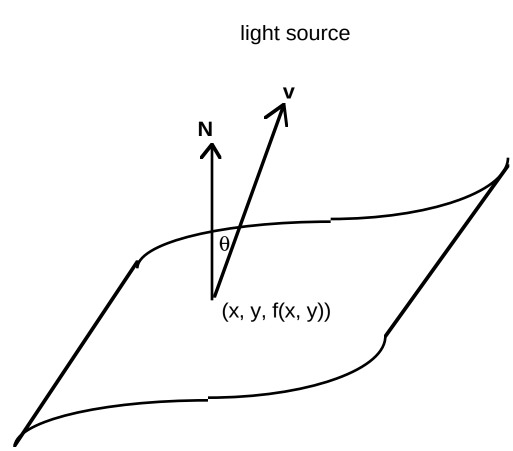
In an effort to bring more realistic applications of calculus into my multivariable calculus class I developed a project which asks groups of students to develop shading functions that simulate lighting from various point and infinite light sources for surfaces in space. I posed the problem to my students as if they were working for a computer graphics company which was designing a program that would allow a user to build 3–D models and then light the models in a number of ways. Their job was to come up with a flexible lighting plan that would allow the user to

a. adjust ambient light
b. adjust light sources position (either a point or infinite) and brightness
c. adjust the shininess of the surface

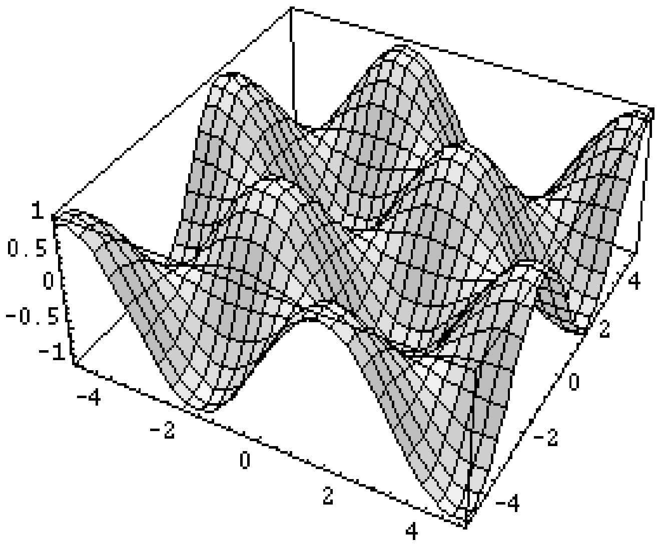
I emphasized that they were to write up their report professionally as if they were turning it in to a supervisor who was not a mathematician but had some background in mathematics. They were also asked to include Mathematica graphics that demonstrated their models capabilities.

The mathematics involved in this project is interesting but not beyond the level of a typical calculus student. Assume we are given a surface defined by $z = f(x,y)$. We wish to develop a "shading function" $s(x,y)$ that assigns a gray level between 0 and 1 (1 being white and 0 being black) to each point $(x, y, f(x,y))$ on the surface. The illumination from either a point or infinite light source will vary across the surface depending on whether or not there is an unobstructed path from the light source to the point on the surface (If not, then the point is in a shadow.) and the angle $\theta$ the normal vector $\mathbf{N} = \nabla(z - f(x,y))$ to the surface at the point makes with the incoming light ray. If the angle is close to $0^o$ the illumination will be greatest and if the angle is close to $90^o$ the illumination will be negligible. Assume first that the light source is infinite coming from the direction - $\mathbf{v}$. Then

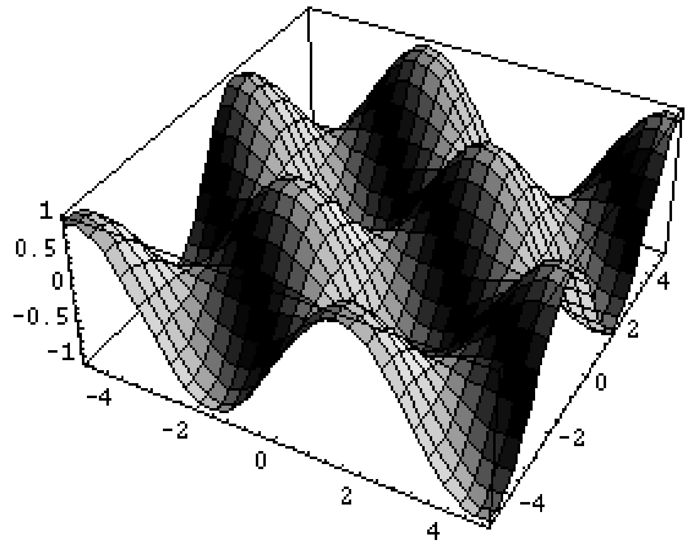
$$\cos\theta = \frac{\mathbf{N}\cdot\mathbf{v}}{|\mathbf{N}|\,|\mathbf{v}|}.$$



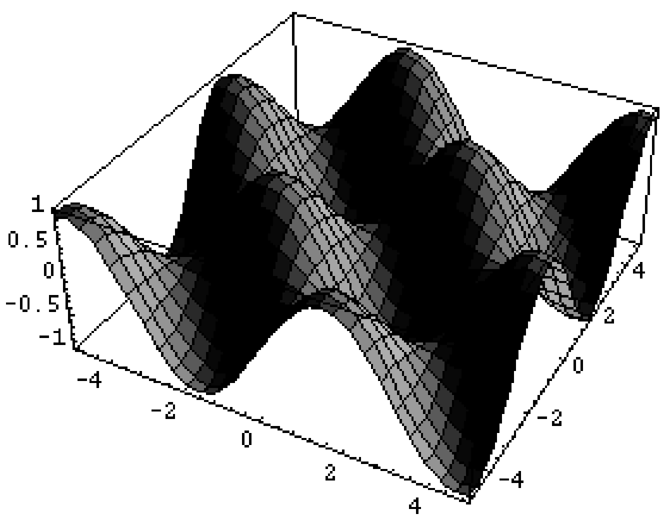
Since $\cos\theta = 1$ when $\theta = 0^o$ and $\cos\theta = 0$ when $\theta = 90^o$ this expression can serve as a shading function for parts of the surface lit by the light source. By taking $s(x,y) = \max\{0, \cos\theta\}$ we obtain a shading function which also assigns 0 (black) to areas that do not face the light source ($90^o < \theta < 180^o$). For example, if $f(x,y) = \sin(x)\cos(x)$ then the shaded graphs for several infinite light sources are



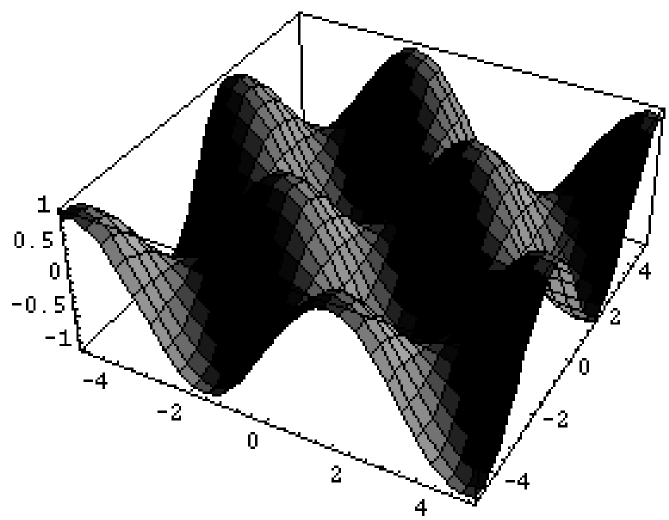

**v** = **k**




**v** = **i** + **j** + **k**




**v** = 3**i** + 3**j** +**k**

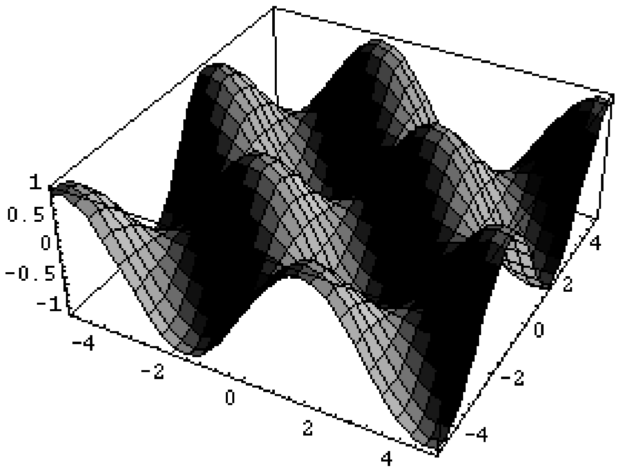



**v** = 5**i** + 5**j** + **k**

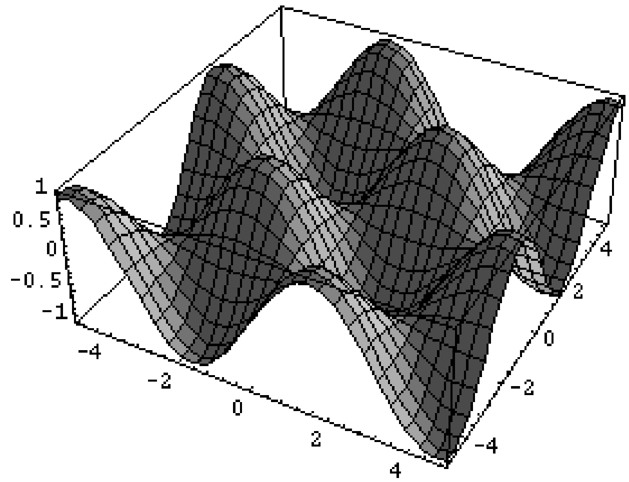
Once the basic model is developed it is rather easy to modify it to include variable ambient lighting and brightness. These are accounted for in the model

$$s(x,y) = \max\{a, b\cos(\theta)\} \quad \text{where}$$

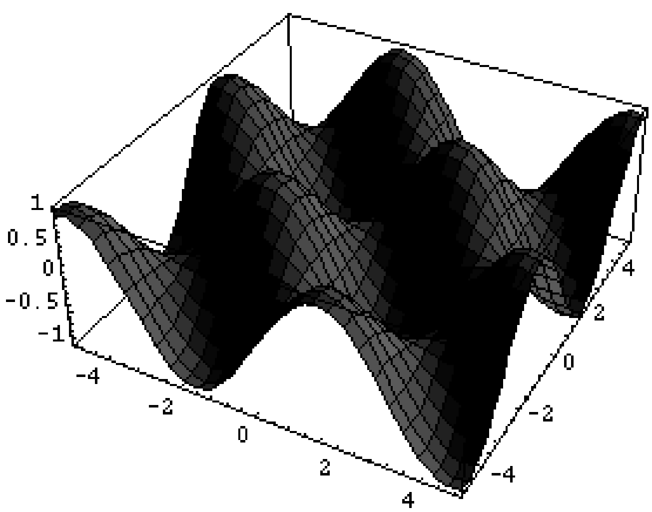
$a$ = ambient light setting, $0 \le a \le 1$
$b$ = brightness setting, $0 \le b \le 1$



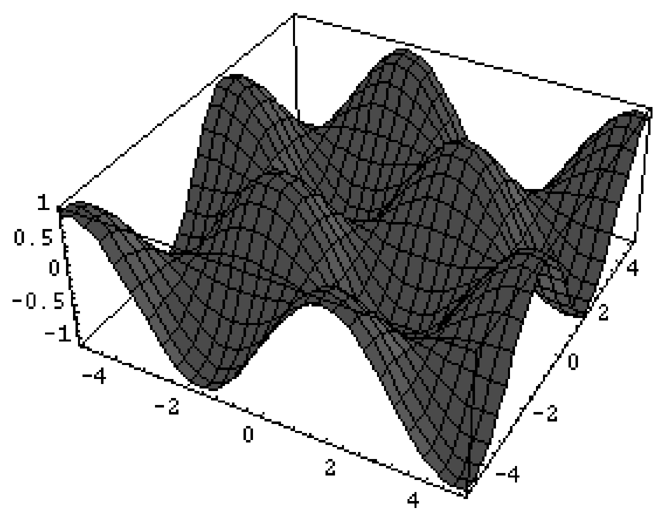

a = 0, b = 1, n = 1, **v** = 2**i** + 2**j** + **k**




a = .3, b = 1, n = 1, **v** = 2**i** + 2**j** + **k**




a = 0, b = .5, n = 1, **v** = 2**i** + 2**j** + **k**



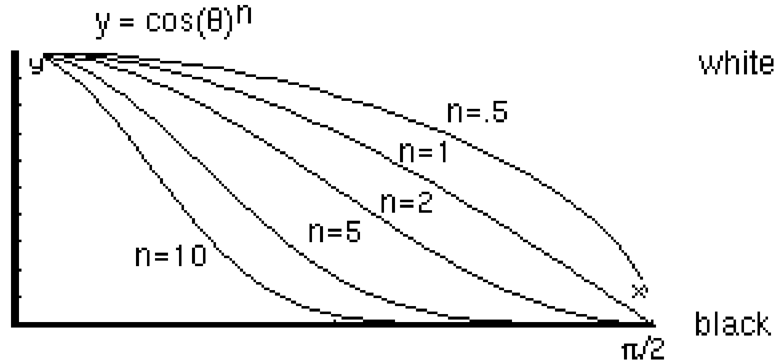

a = .3, b = .5, n = 1, **v** = 2**i** + 2**j** + **k**

The shinier the surface the faster the illumination drops off as the angle the light source makes with the normal increases from $0^o$ to $90^o$. One model that creates this effect is

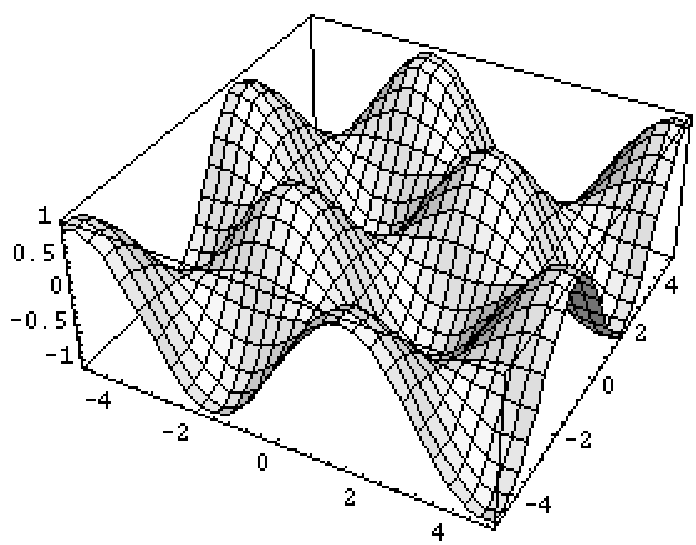
$$s(x,y) = \max\{a, b\cos(\theta)^n\} \quad \text{where}$$

$$n = \text{shininess setting}, \; n \geq 0.$$

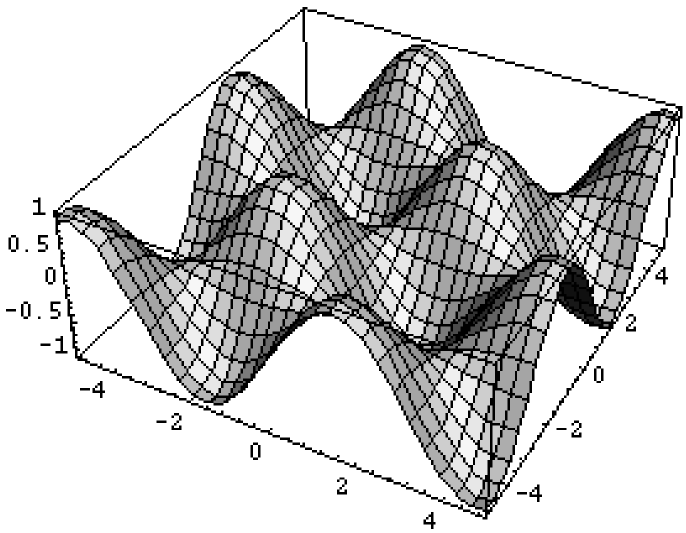
By raising $\cos(\theta)$ to various powers the illumination drops off faster or slower depending on whether $n > 1$ or $0 < n < 1$.



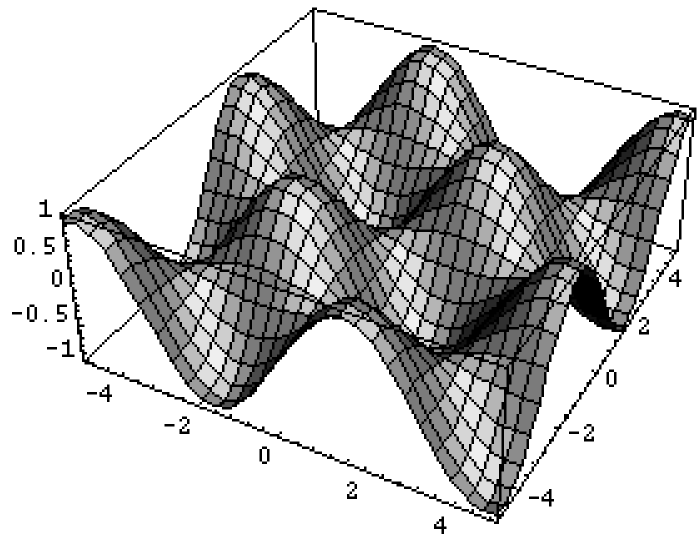

Below are some examples of this effect.




$a = 0$, $b = 1$, $n = 1$, $\mathbf{v} = \mathbf{i} - \mathbf{j} + 3\mathbf{k}$

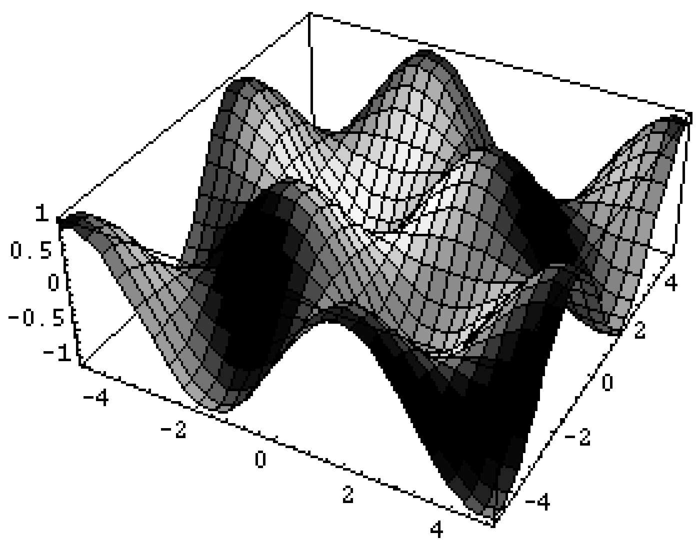



a = 0, b = 1, n = 3, **v** = **i** - **j** + 3**k**

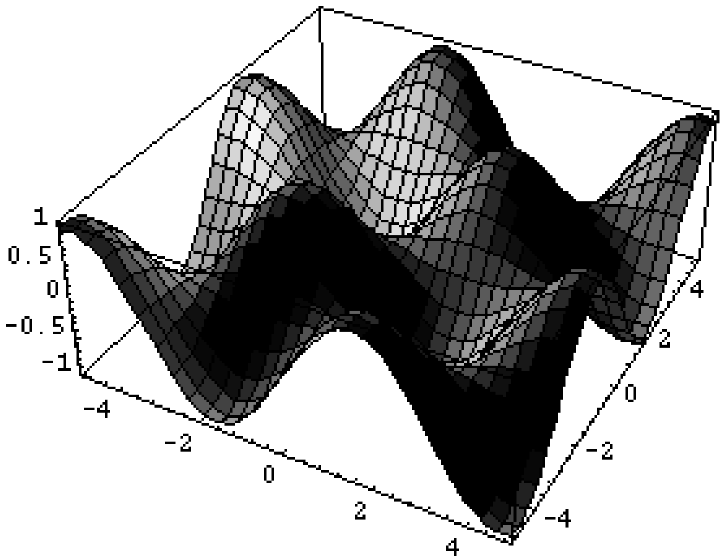



a = 0, b = 1, n = 5, **v** = **i** - **j** + 3**k**

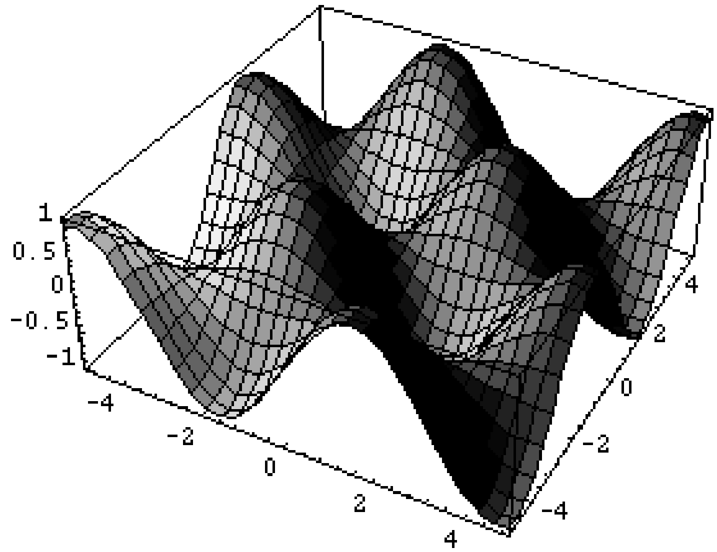
To create similar models using point rather than infinite light sources one only needs to replace **v** by the vector (u - x)**i** + (v - y)**j** + (w - f(x,y))**k** where (u, v, w) is the point light source.




a = 0, b = 1, n = 1, (u, v, w) = (0, 0, 2)




a = 0, b = 1, n = 1, (u, v, w) = (-4, 0, 2)




a = 0, b = 1, n = 1, (u, v, w) = (-4, -4, 2)

The difficulty I encountered in this project was determining how much to tell my students. They need some information to get started and I chose to outline the basic infinite light source model for them. I also gave hints as their models progressed as to how to incorporate variable ambient light, brightness and shininess into their models. The shininess adjustment gave them the most difficulty. I required them to provide rationale for why each constant in the final model had the desired effect. They also demonstrated the effect of varying each constant using Mathematica in a fashion similar to what I did earlier. Overall I felt the project was a great success. It provided them an interesting nontrivial application of the calculus they were learning as well as the opportunity to work together as a team.