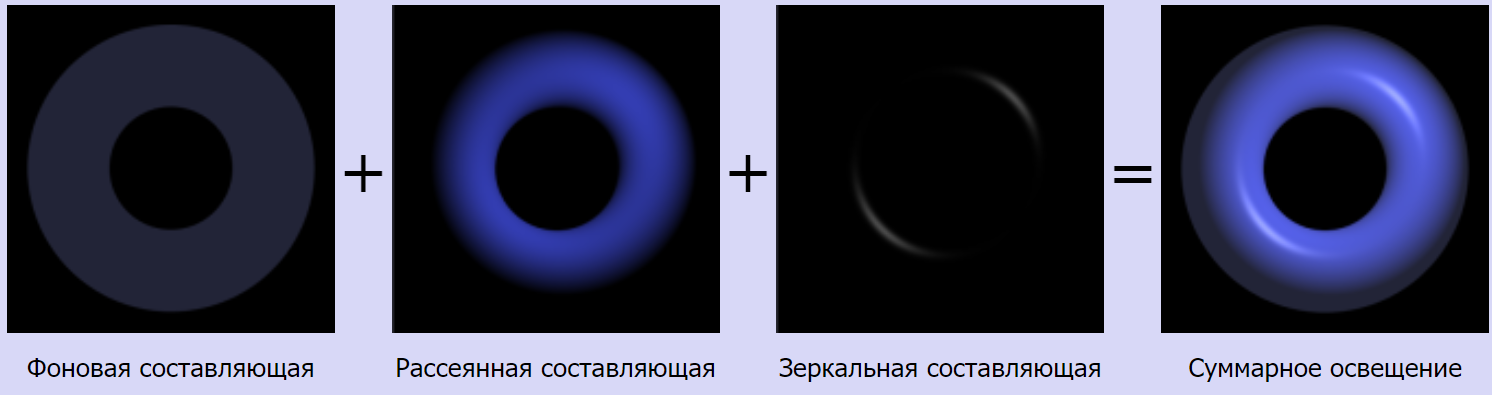
**Модель отражения Фонга.**

**Модель отражения Блинна-Фонга**

**Модель Фонга**

Модель отражения Фонга является довольно популярной и широкоиспользуемой в компьютерной графике, несмотря на то, что существуют более точные физические модели. Она была предложена в статье Ву Тонг Фонга (Bui Tuong Phong) в 1975 году.

Пусть заданы точечный источник света, расположенный в некоторой точке, поверхность, которая будет освещаться и наблюдатель. Будем считать, что наблюдатель точечный. Каждая точка поверхности имеет свои координаты и в ней определена нормаль к поверхности. Её освещенность складывается из трех компонент: фоновое освещение (ambient), рассеянный свет (diffuse) и бликовая составляющая (specular). Свойства источника определяют мощность излучения для каждой из этих компонент, а свойства материала поверхности определяют её способность воспринимать каждый вид освещения.



Для удобства все векторы, описанные ниже, берутся единичными. В этом случае косинус угла между ними совпадает со скалярным произведением.

1. **Фоновое освещение** — это постоянная в каждой точке величина надбавки к освещению. Вычисляется фоновая составляющая освещения как:



где:

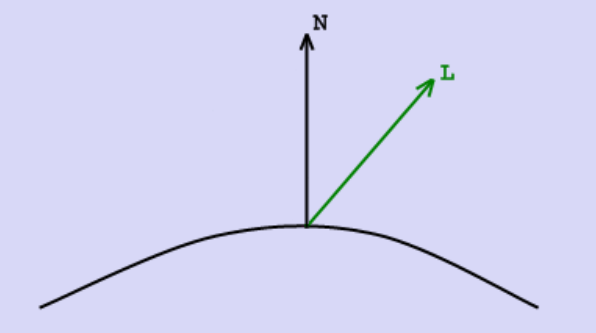
 - фоновая составляющая освещенности в точке;

 - свойство материала воспринимать фоновое освещение;

 - мощность фонового освещения.

Из формулы выше видно, что фоновая составляющая освещенности не зависит от пространственных координат освещаемой точки и источника. Поэтому при моделировании освещения, в большинстве случае, не имеет смысла брать более одного фонового источника света. Часто просто задается некое глобальное фоновое освещение всей сцены.

1. **Рассеянный свет** при попадании на поверхность рассеивается равномерно во все стороны. При расчете такого освещения учитывается только ориентация поверхности (нормаль) и направление на источник света. Рассеянная составляющая рассчитывается по закону косинусов:





где:

 - рассеянная составляющая освещенности в точке;

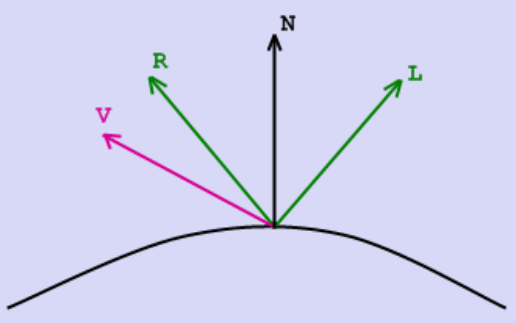
 - свойство материала воспринимать рассеянное освещение;

 - мощность рассеянного освещения;

 - направление **из** точки **на** источник;

 - вектор нормали в точке.

1. **Зеркальный свет** при попадании на поверхность подчиняется следующему закону: “Падающий и отраженный лучи лежат в одной плоскости с нормалью к отражающей поверхности в точке падения, и эта нормаль делит угол между лучами на две равные части”. То есть отраженная составляющая освещенности в точке зависит от того, насколько близки направления на наблюдателя и отраженного луча. Это можно выразить следующей формулой:





где:

 - зеркальная составляющая освещенности в точке;

 - коэффициент зеркального отражения;

 - мощность зеркального освещения;

 - направление отраженного луча;

 - направление на наблюдателя;

 - коэффициент блеска, свойство материала.

Именно зеркальное отражение представляет наибольший интерес, но в то же время его расчет требует больших вычислительных затрат. При фиксированном положении поверхности относительно источников света фоновая и рассеянные составляющие освещения могут быть просчитаны единожды для всей сцены, т.к. их значение не зависит от направления взгляда. С зеркальной составляющей этот фокус не сработает и придется пересчитывать её каждый раз, когда взгляд меняет свое направление.

Во всех вычислениях выше, для рассеянной и зеркальной компонен, если скалярное произведение в правой части меньше нуля, то соответствующая компонента освещенности полагается равной нулю.

**Упрощенный расчет зеркальной компоненты освещенности. Модель Блинна-Фонга**

Для расчета отраженной компоненты требуется выполнить довольно громоздкие вычисления. Существует модель Блинна-Фонга, представляющая собой модель Фонга с упрощенным расчетом зеркального отражения. Вычислим в каждой точке вектор полупути  (halfway vector):



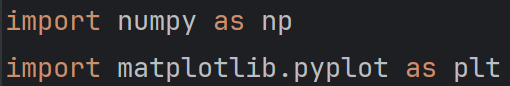
который показывает ориентацию площадки, на которой будет максимальное отражение. Тогда величину  можно заменить величиной .

При этом  и, в общем случае, соотношение между ними зависит от пространственной связи векторов ,  и . Вектор  называется вектором полупути, т.к. если все три вектора ,  и  лежат в одной плоскости, то угол между  и  составляет половину угла между  и .

Модель отражения Блинна-Фонга никогда в точности не совпадает с моделью Фонга, однако можно подобрать соответствующие значения *α* и *β*, для которых распределения зеркальной составляющей по поверхности для обеих моделей будут очень близкими. Вместе с тем, в ряде случаев модель Блинна-Фонга требует значительно меньше вычислений, например в случае направленного бесконечно-удаленного источника.

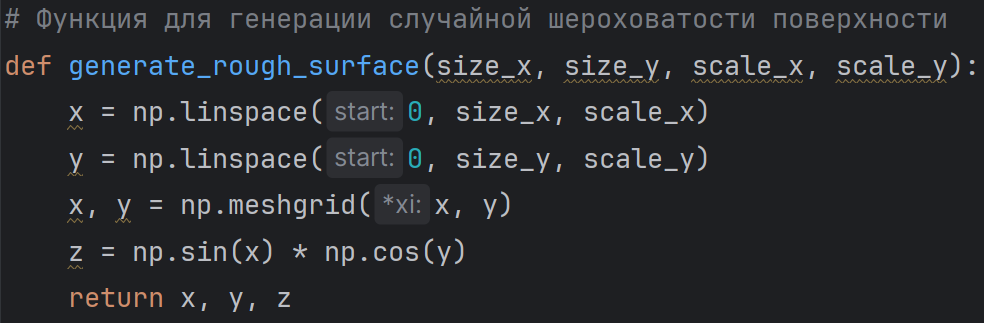
**Моделирование отражения света от шероховатой поверхности с использованием модели Блинна-Фонга.**

**Импорт библиотек**



Здесь импортируются необходимые библиотеки: *numpy* для численных вычислений и *matplotlib* для визуализации.

**Генерация шероховатой поверхности**



Эта функция создает 2D сетку координат *x* и *y* и вычисляет значения *z* как функцию от *x* и *y*.

Аргументы функции:

- *size\_x*, *size\_y*: диапазон значений по осям *x* и *y*. Например, если *size\_x* и *size\_y* равно 10, то значения по осям *x* и *y* будут варьироваться от 0 до 10 соответственно;

- *scale\_x*, *scale\_y*: количество точек в каждом измерении (по осям *x* и *y* соответственно), то есть размер сетки. Например, если *scale\_x* и *scale\_y* равно 100 и 100, то будет создана сетка 100 на 100 точек.

Внутренние шаги функции:

1. Создание линейных массивов *x* и *y*:

- *np.linspace(start, stop, num)* создает одномерный массив из *num* равномерно распределенных точек между *start* и *stop*;

- Здесь создаются два одномерных массива *x* и *y*, каждый из которых содержит *scale\_x* и *size\_y* точек в диапазоне от 0 до *size\_x* и *size\_y* соответственно;

2. Создание 2D сетки координат x и y:

- *np.meshgrid(x, y)* создает двумерные сетки координат из одномерных массивов *x* и *y*. В результате получаются два двумерных массива *x* и *y*, которые можно использовать для построения 3D поверхности;

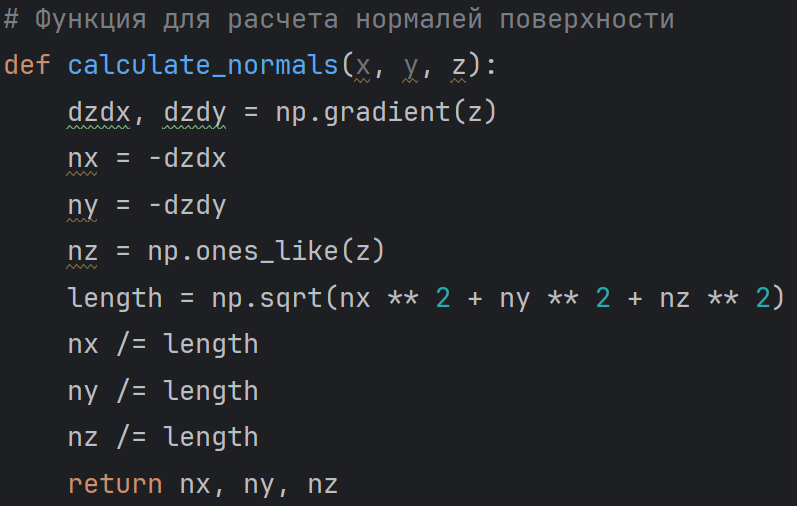
- В массиве *x* каждая строка содержит одинаковые значения из исходного одномерного массива *x*;

- В массиве *y* каждый столбец содержит одинаковые значения из исходного одномерного массива *y*.

3. Вычисление значений *z*:

- Здесь создается двумерный массив *z*, который определяет высоту поверхности в каждой точке сетки.

**Расчет нормалей поверхности**



Эта функция вычисляет нормали к поверхности:

Аргументы функции:

- *x*: координаты сетки по оси *x*;

- *y*: координаты сетки по оси *y*;

- *z*: высоты поверхности (координаты по оси *z*).

Внутренние шаги функции:

1. Вычисление градиентов *dzdx* и *dzdy*:

- *np.gradient(z)* вычисляет численные градиенты массива *z* по его осям;

- *dzdx* представляет собой частную производную *z* по направлению *x*;

- *dzdy* представляет собой частную производную *z* по направлению *y*.

2. Вычисление компонент нормали *nx*, *ny* и *nz*:

- *nx* и *ny* являются отрицательными градиентами *dzdx* и *dzdy* соответственно. Это обеспечивает правильное направление нормали;

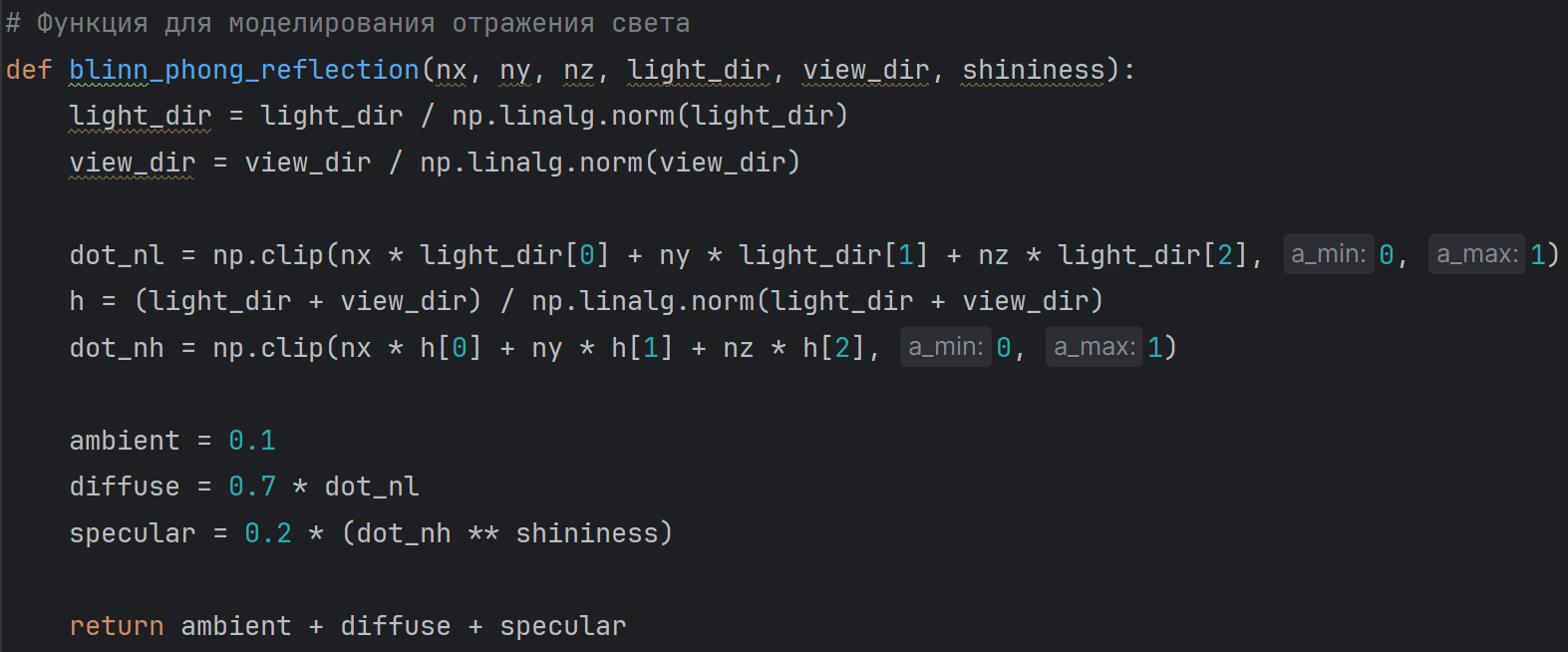
- *nz* устанавливается в 1, так как нормали направлены вверх и имеют компоненту *z*.

3. Нормализация нормалей:

- Вычисляется длина вектора нормали для каждой точки: *length = np.sqrt(nx\*\*2 + ny\*\*2 + nz\*\*2)*;

- Каждая компонента нормали делится на длину вектора, чтобы нормализовать вектор (привести его к единичной длине).

**Моделирование отражения света (модель Блинна-Фонга)**



Эта функция вычисляет отражение света от поверхности с использованием модели освещения Блинна-Фонга, которая включает амбиентное, диффузное и зеркальное (спекулярное) освещение:

Аргументы функции:

- *nx*: компонент *x* нормалей к поверхности;

- *ny*: компонент *y* нормалей к поверхности;

- *nz*: компонент *z* нормалей к поверхности;

- *light\_dir*: направление света (вектор);

- *view\_dir*: направление обзора (вектор);

- *shininess*: коэффициент блеска, определяющий, насколько зеркальным является отражение.

Вектор ;

Вектор *;*

Вектор *;*

Коэффициент .

Внутренние шаги функции:

1. Нормализация направлений света и обзора:

Нормализуем векторы *light\_dir* и *view\_dir*, деля каждый из них на его длину (*np.linalg.norm*), чтобы получить единичные векторы.

1. Расчет косинуса угла между нормалью и направлением света (*dot\_nl*):

- Вычисляем скалярное произведение нормали и направления света  (*nx \* light\_dir[0] + ny \* light\_dir[1] + nz \* light\_dir[2]*).

- Функция *np.clip* ограничивает значения в диапазоне от 0 до 1, чтобы исключить отрицательные значения (когда угол между нормалью и направлением света превышает 90 градусов) и значения больше 1.

1. Расчет вектора полупути:

- Вектор полупути *h* вычисляется как нормализованный вектор, являющийся суммой нормализованных векторов *light\_dir* и *view\_dir* . Это промежуточный вектор, используемый в модели освещения Блинна-Фонга.

1. Расчет косинуса угла между нормалью и вектором полупути (*dot\_nh*):

- Вычисляем скалярное произведение нормали и вектора полупути  (*nx \* h[0] + ny \* h[1] + nz \* h[2]*).

5. Расчет интенсивности отражения:

- Вычисляем интенсивность отражения путем скалярного произведения нормалей поверхности (*nx*, *ny*, *nz*) и направления света (*light\_dir*).

- *light\_dir* и *view\_dir* нормализуются;

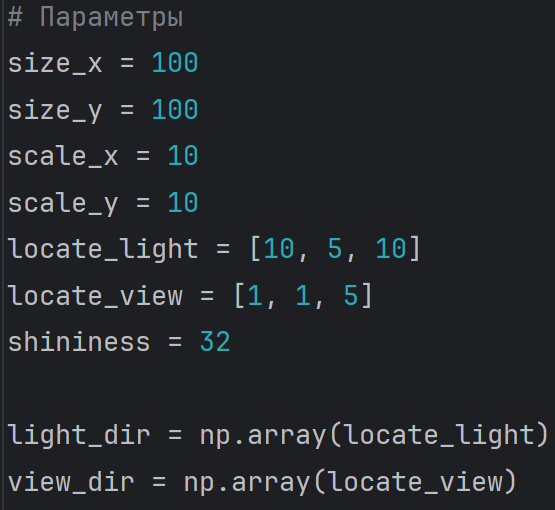
- *dot\_nl* - косинус угла между нормалью и направлением света, обрезается до диапазона [0, 1];

- *h* - вектор полупути (halfway), который является нормализованной суммой направлений света и обзора;

- *dot\_nh* - косинус угла между нормалью и вектором полупути, также обрезается до диапазона [0, 1];

- *ambient*, *diffuse* и *specular* компоненты освещения рассчитываются и суммируются.

**Параметры**



Здесь задаются параметры:

- *size\_x* - диапазон значений по оси *x*;

- *size\_y* - диапазон значений по оси *y*;

- *scale\_x* - количество точек по оси *x*;

- *scale\_y* - количество точек по оси *x*;

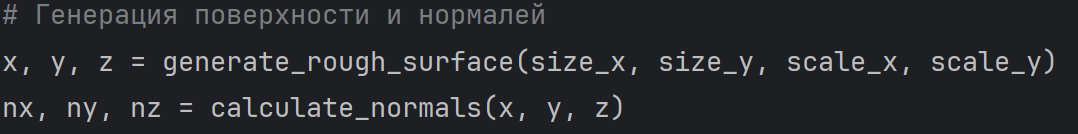
- *locate\_light* - направление света;

- *locate\_view* - направление обзора;

- *shininess* - коэффициент блеска для модели Блинна-Фонга.

**Генерация и отображение поверхности и отражения.**

**Генерация поверхности и нормалей**

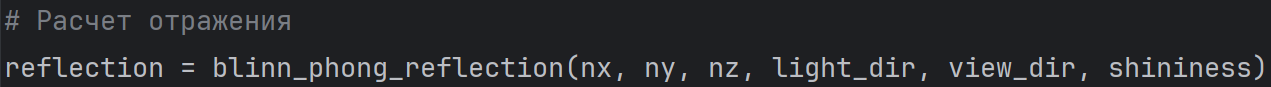


Здесь используется ранее определенные функции *generate\_rough\_surface* и *calculate\_normals* для создания сетки поверхности и вычисления нормалей к этой поверхности:

- *x*, *y*, *z* - координаты поверхности;

- *nx*, *ny*, *nz* - компоненты нормалей к поверхности.

**Расчет отражения**



Эта строка кода использует функцию *blinn\_phong\_reflection* для вычисления интенсивности света, отраженного от поверхности, с учетом нормалей, направления света и направления обзора:

- *reflection* - массив интенсивностей света, отраженного от поверхности, рассчитанный с использованием модели Блинна-Фонга.

**Визуализация**



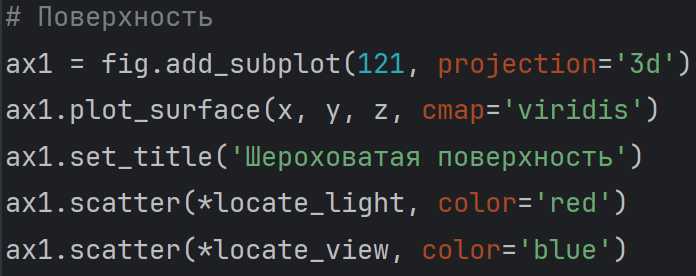
Эта часть кода отвечает за визуализацию поверхности и отражения света с использованием библиотеки *matplotlib*:

1. Создание фигуры:



- Создается фигура размером 12x6 дюймов для размещения двух графиков.

2. Визуализация поверхности:

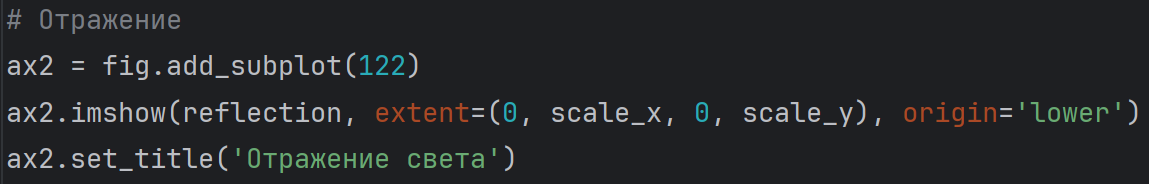
 - Добавляется 3D-подграфик (ax1) для визуализации поверхности;

- *plot\_surface(x, y, z, cmap='viridis')* рисует поверхность с использованием цветовой карты *viridis*;

- Устанавливается заголовок для подграфика;

- Устанавливаются две точки: источника свет (красного цвета), наблюдателя (синего цвета).

3. Визуализация отражения:

 - Добавляется подграфик (ax2) для визуализации отражения света;

- *imshow(reflection, extent=(0, scale\_x, 0, scale\_y), origin='lower')* рисует изображение отражения света;

- Устанавливается заголовок для подграфика.

4. Отображение фигуры:



- Отображает фигуру с двумя подграфиками.