Для приведения квадратичной формы к каноническому виду требуется найти ортогональное преобразование координат, которое приведет матрицу квадратичной формы к диагональному виду.

Шаги для нахождения ортогонального преобразования координат:

1. Найдите собственные значения и собственные векторы матрицы квадратичной формы.

2. Постройте ортонормированный базис из собственных векторов.

3. Составьте матрицу преобразования, где столбцы - это ортонормированный базис.

Давайте выполним эти шаги для данной квадратичной формы:

1. Найдем собственные значения и собственные векторы. Решим характеристическое уравнение:

|A - λI| = 0

где A - матрица квадратичной формы, λ - собственное значение, I - единичная матрица.

Матрица A:

|2-λ 4 4 |

| 4 2-λ -4 |

| 4 -4 2-λ|

Выполним вычисления:

(2-λ)((2-λ)(2-λ)-(-4)(-4)) - (4)(4(2-λ)-4(4)) + (4)(4(-4)-(-4)(4)) = 0

(2-λ)(λ^2 - 4λ + 4 + 16) - 16(2-λ) + 64(2-λ) = 0

(2-λ)(λ^2 - 4λ + 20) = 0

Получаем кубическое характеристическое уравнение:

λ^3 - 4λ^2 + 20λ - 40 = 0

Решим его численно и найдем три собственных значения λ1, λ2 и λ3.

2. Найдем собственные векторы, соответствующие каждому собственному значению. Для каждого значения подставим его обратно в уравнение (A - λI)v = 0 и решим систему уравнений, где v - собственный вектор.

Найденные собственные значения и соответствующие им собственные векторы:

Собственное значение λ1 = 12:

Соответствующий собственный вектор:

v1 = (1, -1, -1)

Собственное значение λ2 = -8:

Соответствующий собственный вектор:

v2 = (1, 1, -1)

Собственное значение λ3 = 2:

Соответствующий собственный вектор:

v3 = (1, -1, 1)

3. Построим ортонормированный базис из собственных векторов. Для этого нормализуем каждый собственный вектор, разделив его на его длину:

v1\_normalized = v1 / ||v1||

v2\_normalized = v2 / ||v2||

v3\_normalized = v3 / ||v3||

где ||v|| - длина вектора v.

Теперь имеем ортонормированный базис из трех векторов:

v1\_normalized = (1/√3, -1/√3, -1/√3)

v2\_normalized = (1/√3, 1/√3, -1/√3)

v3\_normalized = (1/√3, -1/√3, 1/√3)

4. Составим матрицу преобразования, где столбцы - это ортонормированный базис:

P = (v1\_normalized | v2\_normalized | v3\_normalized)

где "|" обозначает столбцовую конкатенацию.

Подставим значения:

P = (1/√3 1/√3 1/√3

-1/√3 1/√3 -1/√3

-1/√3 -1/√3 1/√3)

Матрица P является ортогональной матрицей, так как ее столбцы являются ортонормированными векторами. Это и будет ортогональным преобразованием координат, которое приводит квадратичную форму к каноническому виду.

Для проверки преобразуем исходную квадратичную форму с помощью ортогонального преобразования:

K' = (𝑥1' 𝑥2' 𝑥3') P^T A P (𝑥1' 𝑥2' 𝑥3')^T

где P^T - транспонированная матрица P.

Подставим значения:

K' = (𝑥1' 𝑥2' 𝑥3') (1/√3 -1/√3 -1/√3

1/√3 1/√3 -1/√3

1/√3 -1/√3 1/√3)

(2 4 4

4 2 -4

4 -4 2)

(1/√3 1/√3 1/√3)

= (𝑥1' 𝑥2' 𝑥3') (12 0 0

0 -8 0

0 0 2)

(1/√3 1/√3 1/√3)^T

= (1/√3 𝑥1' + 1/√3 𝑥2' + 1/√3 𝑥3') 12

+ (1/√3 𝑥1' + 1/√3 𝑥2' - 1/√3 𝑥3') (-8)

+ (1/√3 𝑥1' - 1/√3 𝑥2' + 1/√3 𝑥3') 2

Таким образом, ортогональное преобразование приводит исходную квадратичную форму к следующему каноническому виду:

K' = 12(1/√3 𝑥1' + 1/√3 𝑥2' + 1/√3 𝑥3')^2

- 8(1/√3 𝑥1' + 1/√3 𝑥2' - 1/√3 𝑥3')^2

+ 2(1/√3 𝑥1' - 1/√3 𝑥2' + 1/√3 𝑥3')^2

Предоставленная квадратичная форма K = (𝑥1 𝑥2 𝑥3) A (𝑥1 𝑥2 𝑥3), где матрица A =

| 2 4 4 |

| 4 2 -4 |

| 4 -4 2 |, задает поверхность второго порядка.

Для определения названия этой поверхности, мы можем проанализировать характеристики квадратичной формы.

1. Собственные значения:

Найдем собственные значения матрицы A, решив характеристическое уравнение det(A - λI) = 0.

Det(A - λI) = | 2-λ 4 4 |

| 4 2-λ -4 |

| 4 -4 2-λ |

Выполним вычисления:

(2-λ)((2-λ)(2-λ)-(-4)(-4)) - (4)(4(2-λ)-4(4)) + (4)(4(-4)-(-4)(4)) = 0

(2-λ)(λ^2 - 4λ + 4 + 16) - 16(2-λ) + 64(2-λ) = 0

(2-λ)(λ^2 - 4λ + 20) = 0

Решив квадратное уравнение, получаем три собственных значения:

λ1 = 12, λ2 = -8, λ3 = 2.

2. Собственные векторы:

Для каждого собственного значения найдем соответствующий ему собственный вектор, решив систему уравнений (A - λI)v = 0.

Для собственного значения λ1 = 12:

(A - λ1I)v1 = (A - 12I)v1 = 0

| -10 4 4 | | x1 | | 0 |

| 4 -10 -4 | | x2 | = | 0 |

| 4 -4 -10 | | x3 | | 0 |

Решив данную систему уравнений, получаем собственный вектор v1 = (1, -1, -1).

Для собственного значения λ2 = -8:

(A - λ2I)v2 = (A + 8I)v2 = 0

| 10 4 4 | | x1 | | 0 |

| 4 10 -4 | | x2 | = | 0 |

| 4 -4 10 | | x3 | | 0 |

Решив данную систему уравнений, получаем собственный вектор v2 = (1, 1, -1).

Для собственного значения λ3 = 2:

(A - λ3I)v3 = (A - 2I)v3 = 0

| 0 4 4 | | x1 | | 0 |

| 4 0 -4 | | x2 | = | 0 |

| 4 -4 0 | | x3 | | 0 |

Решив данную систему уравнений, получаем собственный вектор v3 = (1, -1, 1).

Теперь, имея собственные значения и собственные векторы, мы можем определить название поверхности второго порядка:

- Для собственные значения λ1 и λ2, соответствующие собственным векторам v1 и v2, у нас есть два отрицательных собственных значения, что указывает на гиперболический тип поверхности.

- Для собственное значение λ3, соответствующее собственному вектору v3, у нас есть одно положительное собственное значение, что указывает на эллиптический тип поверхности.

Следовательно, поверхность второго порядка, заданная данной квадратичной формой, является комбинацией гиперболической и эллиптической поверхностей.

Для изображения поверхности второго порядка в пространстве (𝑥1, 𝑥2, 𝑥3), нам потребуется использовать трехмерный график.

Код Python:

import numpy as np

import matplotlib.pyplot as plt

from mpl\_toolkits.mplot3d import Axes3D

# Каноническая форма квадратичной формы

A = np.array([[12, 0, 0],

[0, -8, 0],

[0, 0, 2]])

# Генерация координатной сетки

x1 = np.linspace(-10, 10, 100)

x2 = np.linspace(-10, 10, 100)

x1, x2 = np.meshgrid(x1, x2)

# Вычисление значения функции k(x1, x2, x3)

x3 = np.zeros\_like(x1)

k = A[0, 0] \* x1\*\*2 + A[1, 1] \* x2\*\*2 + A[2, 2] \* x3\*\*2

# Создание 3D-построения

fig = plt.figure()

ax = fig.add\_subplot(111, projection='3d')

# Построение поверхности

ax.plot\_surface(x1, x2, k, cmap='viridis')

# Настройка меток осей

ax.set\_xlabel('x1')

ax.set\_ylabel('x2')

ax.set\_zlabel('k(x1, x2, x3)')

# Показать график

plt.show()

Полученный график в Jupyter Notebook:

