

ПРОГРАММА ПРИВЕДЕНИЯ УРАВНЕНИЯ КРИВОЙ ВТОРОГО ПОРЯДКА К КАНОНИЧЕСКОМУ ВИДУ

Автор: Артемьева Анастасия Дмитриевна, студент 1 курса университета «Дубна» филиал «Протвино», кафедра физики

Научный руководитель: ст. преп. Губаева Милета Михайловна

Аннотация

Создана программа для приведения уравнения второго порядка к каноническому виду. Программа разработана на языке программирования C++ с использованием среды Microsoft Visual Studio 2013.

The program for the reduction of the equation of the second order and his reduction to a kanonicheky look is created. The program is developed in a programming language C ++ with use of the Microsoft Visual Studio 2013 environment.

Для решения многих графических задач, решаемых при создании видео с помощью компьютеров, требуется предварительное знание стандартного или канонического вида уравнения, задающего форму объекта. Представленная в данной статье программа определяет тип и вид кривой второго порядка, заданной квадратным уравнением в декартовых координатах, и определяет преобразования, приводящие это уравнение к каноническому виду.

Теоретические основы данных преобразований представлены в учебниках и классической литературе по геометрии и линейной алгебре[1][3]. Поэтому ниже приводится только описание алгоритма программы.

Кривые второго порядка – геометрическое место точек плоскости, прямоугольные координаты которых удовлетворяют уравнению общего вида: $a_{11}x^2 + a_{22}y^2 + 2a_{12}xy + 2a_{13}x + 2a_{23}y + a_{33} = 0$, в котором, по крайней мере, один из коэффициентов a_{11} , a_{12} , a_{22} отличен от нуля[4] [5].

Для любой кривой второго порядка существует прямоугольная система координат ОХУ, в которой её уравнение принимает один из следующих трёх канонических видов: для центральных линий, для нецентральных линий и для параболы.

Алгоритм программы использует инварианты преобразований. Инвариантами кривой называются такие выражения, составленные из коэффициентов ее уравнения, которые не меняются при переходе от одной прямоугольной декартовой системы координат к другой такой же системе, т. е. при поворотах осей координат и при параллельных переносах осей[2].

Для этого программа находит инварианты по формулам[1]:

– инварианты при сдвигах и поворотах

$$I_1 = a_{11} + a_{22}$$

$$I_2 = \begin{vmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{vmatrix}$$

$$I_3 = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{vmatrix}$$

– инвариант только при поворотах

$$K = \begin{vmatrix} a_{11} & a_{13} \\ a_{13} & a_{33} \end{vmatrix} + \begin{vmatrix} a_{22} & a_{23} \\ a_{23} & a_{33} \end{vmatrix}$$

Ниже представлена часть программного кода, где определяются тип и вид кривой по инвариантам:

```

//===Определяем тип кривой
1.   if (I2 > 0) printf("\tЭллиптический тип кривой\n\n");
2.   if (I2 == 0) printf("\tПараболический тип кривой\n\n");
3.   if (I2 < 0) printf("\tГиперболический тип кривой\n\n");

//===Определяем вид кривой
1.   if (I2>0 && I1*I3<0) printf("\tЭллипс\n\n");
2.   if (I2>0 && I1*I3>0) printf("\tМнимый эллипс\n\n");
3.   if (I2>0 && I3 == 0) printf("\tДве мнимые пересекающиеся
прямые(Вырожденный эллипс)\n\n");
4.   if (I2<0 && I3 != 0) printf("\tГипербола\n\n");
5.   if (I2<0 && I3 == 0) printf("\tДве пересекающиеся прямые (вырожденная
гипербола)\n\n");
6.   if (I2 == 0 && I3 != 0) printf("\tПарабола\n\n");
7.   if (I2 == 0 && I3 == 0 && K<0) printf("\tДве параллельные прямые\n\n");
8.   if (I2 == 0 && I3 == 0 && K>0)
printf("\tДве мнимых параллельных прямых\n\n");
9.   if (I2 == 0 && I3 == 0 && K == 0) printf("\tДве совпадающие прямые\n\n");

```

Всего имеется девять видов кривых.

На следующем шаге программа решает характеристическое уравнение

$$\lambda^2 - I_1\lambda + I_2 = 0,$$

которое для кривых второго порядка является квадратным, и находятся корни этого уравнения λ_1, λ_2 .

Далее программа определяет каноническое уравнение кривой второго порядка по инвариантам:

- для центральных линий ($I_2 \neq 0$, случаи 1-5)

$$\lambda_1 X^2 + \lambda_2 Y^2 + \frac{I_1}{I_2} = 0$$

- нецентральных линий ($I_2 = 0, I_3 \neq 0$, случай 6)

$$I_1 Y^2 = 2X \sqrt{-\frac{I_3}{I_2}}$$

- нецентральных линий ($I_2 = 0, I_3 = 0$ случаи 7-9)

$$I_1 X^2 + \frac{K}{I_1} = 0$$

Данная программа создана как консольное приложение в среде разработки программного обеспечения Microsoft Visual Studio 2013 на языке программирования C++.

Microsoft Visual Studio 2013 – продукт компании Microsoft, включающий интегрированную среду разработки программного обеспечения и ряд других инструментальных средств.

C++ – компилируемый, статически типизированный язык программирования общего назначения, сочетает свойства как высокоуровневых, так и низкоуровневых языков. C++ широко используется для разработки программного обеспечения, являясь одним из самых популярных языков программирования.

Интерфейс программы представлен на Рисунке 1.

Рисунок 1 – Интерфейс программы

После запуска программа:

- выводит на экран общий вид уравнения кривой 2-го порядка;
- просит ввести с клавиатуры коэффициенты этого уравнения, согласно представленному в предыдущем пункте виду;
- затем выводит на экран вид уравнения с заданными коэффициентами;
- далее, согласно описанному алгоритму, вычисляет инварианты данного уравнения;
- определяет тип кривой и выводит его на экран;
- находит, является ли уравнение кривой центральным или нецентральным;
- решает характеристическое уравнение;
- находит уравнение кривой в каноническом виде и выводит его на экран.

Разработанная программа может служить основой для написания модуля преобразований в графических программах.

Список использованной литературы.

1. В.О. Соловьёв «Курсовая работа по линейной алгебре и аналитической геометрии».
2. Д.В.Клетеник «Сборник задач по аналитической геометрии».
3. В.А. Ильин, Э.Г. Позняк «Аналитическая геометрия»
4. <http://mathhelpplanet.com>
5. <https://ru.wikipedia.org>

ПРОГРАММА ПОИСКА КОМПЛЕКТУЮЩИХ ЭВМ

Автор: Блиско Дмитрий Дмитриевич, студент 4 курса МАИ учебного центра «Интеграция», г. Серпухова, Московской области.

Научный руководитель: старший преподаватель Суханов Семен Валерьевич

Аннотация.

В данной научно-практической работе анализируется программное средство по поиску комплектующих, построение алгоритма выполнения программы, позволяющей осуществить поиск комплектующих ЭВМ в сети Internet, а также разработка самой программы.