FINDING THE GREATEST COMMON DIVISOR OF POLYNUMBERS USING MODERN PROGRAMMING LANGUAGES

Jumanazarova Shodiya Otojon qizi
Urgench State Pedagogical Institute
Teacher of the Department of Physics and Mathematics
E-mail: shodiyajumanazarova219@gmail.com

ARTICLE INFO

ARTICLE HISTORY:
Received:10.04.2025
Revised: 11.04.2025
Accepted:13.04.2025

ABSTRACT:

This article presents the Euclidean algorithm for finding the greatest common divisor of polynomials using the modern programming language Python. It also discusses the mathematical foundations of this algorithm and its implementation in Python.

KEYWORDS:
polynomial, Euclidean algorithm, Python programming language, algebra, computation.

Introduction. Nowadays, in order to increase the effectiveness of education, various modern interactive methods, information and communication tools, and handouts are used in organizing the lesson process. As a result of the teacher's rational use of various educational technologies in the lesson process, students' curiosity increases and cognitive abilities develop. If information technologies are used in each lesson, during the learning process the student acquires information not only on the subject, but also on the principles of operation of computer tools and new programs, and enriches computer literacy.

Today's task of education is to teach students to operate independently in the conditions of an increasingly growing information and educational environment, to use the flow of information rationally. For this, it is necessary to create the opportunity for them to work independently on a continuous basis.

Literature review. Literature devoted to algebra and number theory

Mathematics has always attracted the attention of humanity with its theoretical and practical aspects. Algebra and number theory is one of its most important and relevant areas. In this area, the works of D.R. Saparboyeva "Algebra and Number Theory" and A.G. Kurosh "Higher Algebra Course" are of great importance. Below we will consider the scientific and practical significance of these books.

This book, written by D.R. Saparboyeva, is an important source for studying the basic concepts of algebra and number theory. The book covers algebraic equations, matrices, determinants, the most basic concepts in number theory and their practical application methods.

Main features:

1. Convenient for students and teachers: This work is written in an understandable style, which creates convenience for students. Each topic is explained simply and consistently, which greatly helps students in mastering knowledge.

2. Leading source in Uzbek: While most of the scientific literature on algebra is available in foreign languages, this book is written in Uzbek with in-depth and clear explanations. This solves the language barrier and makes the subject even easier to master.

3. Enriched with practical problems: The book provides many examples to reinforce theoretical knowledge. Through these problems, students can learn to apply theory in practice.

This work by D.R. Saparboyeva serves as a solid source of knowledge for students, teachers and teachers interested in mathematics.

The textbook "Higher Algebra Course" written by A.G. Kurosh is dedicated to the in-depth study of fundamental knowledge in higher algebra. This work stands out from other literature with its theoretical perfection and scientific style. The book describes in detail algebraic structures, group theory, rings, fields and other algebraic concepts.

Main features:

1. Deep disclosure of the theory: The book provides a deep scientific analysis of algebraic concepts and theories. It perfectly describes the theoretical foundations of groups, rings and fields.

2. Academic approach: This work is written in a scientific style and consistency. This allows it to be used as a basic manual in scientific research and higher education.

3. For university students and researchers: The book is intended primarily for students and researchers who want to gain knowledge in higher mathematics. It plays an important role in preparing students for scientific research.

4. High scientific level of the author: A.G. Kurosh is one of the leading experts in algebra, and his works are of classical importance in science. This book is an indispensable resource for those who want to conduct thorough research in algebra.

Euclidean algorithm for finding the GCD of polynomials. Algebra and computational methods, which are one of the main branches of mathematics, are used to solve many problems. Polynomials are algebraic expressions, and finding their greatest common divisor (GCD) is important for many problems. Euclidean algorithm is one of the effective solutions in this process.

Euclidean algorithm . Let's see how Euclidean algorithm, known for integers, and its results are applied to polynomials. We assume that the degree of the polynomial is not less than the degree of the polynomial and divide by ni. We denote the resulting quotient and remainder by and , respectively. It is known that the degree of is less than the degree of . Now, dividing by ni, we denote the quotient and remainder by and . Again, taking into account that the degree of is less than that of , we divide by ; we denote the resulting divisor and remainder by and , etc. In general, we divide each remainder by the next remainder. As a result, we obtain remainders of decreasing degrees: . The number of these remainders is necessarily finite, since their degrees form a sequence of decreasing (but non-negative) integers, and such a sequence cannot be infinite. Therefore, the above division process is finite, and we arrive at a remainder that is exactly divisible by the previous remainder.

Any polynomial of degree 1 in any number field P is an irreducible polynomial in that field. Indeed, a polynomial of degree less than 1 can only be of degree 0. However, it is impossible to write a polynomial of degree 1 as the product of two polynomials of degree 0. First, a polynomial f (x) of higher degree that is irreducible in the field P may be irreducible in a larger field that includes P. For example, a polynomial that is irreducible in the field of rational numbers is irreducible in the field of real numbers. If we factorize a polynomial, we conclude that if it is real, it belongs to the real field, and if it is complex, it belongs to the complex field.

For example: .

Basics of the Euclidean algorithm

1. The Euclidean algorithm is based on iteratively calculating the LCM of two polynomials, and is implemented through the following steps:

2. Determine the two given polynomials.

3. If the second polynomial is zero, the first polynomial is taken as the result.

4. Otherwise, divide the first polynomial by the second polynomial and determine the remainder.

5. Repeat the process, taking the second polynomial as the first polynomial and the remainder as the second polynomial.

6. When the remainder is zero, the last divisor is the LCM.

Python Code Examples

The following Python code examples demonstrate the process of finding the LCF of polynomials in various situations.

Example 1: ECUB for simple polynomials

```
import sympy as sp

def gcd_polynomials(p1, p2):
while p2 != 0:
p1, p2 = p2, sp.rem(p1, p2)
return p1

x = sp.Symbol('x')
p1 = x**3 + 2*x**2 + x + 1
p2 = x**2 + x

ecub = gcd_polynomials(p1, p2)
print(f"ECUB of polynomials: {ecub}")
```

Example 2: ECUB for high degree polynomials

```
import sympy as sp

def gcd_polynomials(p1, p2):
 while p2 != 0:
 p1, p2 = p2, sp.rem(p1, p2)
 return p1

x = sp.Symbol('x')
p3 = x**5 + 3*x**4 + 2*x**3 + x + 7
p4 = x**4 + x**3 + x**2 + x
```

```
ekub_1 = gcd_polynomials(p3, p4)
print(f"EKUB of large degree polynomials: {ekub_1}")
```

Example 3: EKUB for polynomials with different coefficients

```
import sympy as sp

def gcd_polynomials(p1, p2):
while p2 != 0:
p1, p2 = p2, sp.rem(p1, p2)
return p1

x = sp.Symbol('x')
p5 = 5*x**3 + 10*x**2 + 15*x + 20
p6 = 10*x**2 + 5*x + 5

ekub_3 = gcd_polynomials(p5, p6)
print(f"Elektropolynomials with different coefficients "ECUB of polynomials: {ekub_3}")
```

Advantages of Euclidean Algorithm

Accuracy – This algorithm provides accurate results.

Speed – It can be repeated in a short time.

Practical application – Suitable for applications in algebra and calculus.

In mathematics, finding the greatest common divisor (GCD) of polynomials is often useful. In this article, we will develop a GCD calculation software using Flask as a web application.

Result. The main part of the code in the Python programming language looks like this:

Below is the Python code written using Flask. This code calculates the GCD of two polynomials and presents the result to the user via a web interface:

```
from flask import Flask, request, jsonify, render_template
from sympy import symbols, gcd, expand

app = Flask(__name__)
```

```
def find_gcd_of_polynomials(poly1, poly2, variable):
"""
Finds the greatest common divisor (GCD) of two given polynomials.
:param poly1: First polynomial
:param poly2: Second polynomial
:param variable: Variable of the polynomial
:return: GCD
"""
return gcd(poly1, poly2)
@app.route('/')
def index():
return render_template('index.html')

@app.route('/compute_gcd', methods=['POST'])
def compute_gcd():
data = request.form
t = symbols('x')
poly1 = expand(data.get('poly1'))
 poly2 = expand(data.get('poly2'))
 gcd_poly = find_gcd_of_polynomials(poly1, poly2, t)
 return render_template('index.html', gcd_result=str(gcd_poly), poly1=data.get('poly1'), poly2=data.get('poly2'))

if __name__ == '__main__':
 app.run(debug=True).
```

The following areas of future research on the Euclidean algorithm can be explored:

- Optimization through parallel programming
- Advanced approaches to polynomials
- Automated methods for determining the greatest common divisor of polynomials using AI

Conclusion. The Euclidean algorithm is an effective and convenient method for determining the greatest common divisor of polynomials. Implementing this algorithm using the Python programming language allows you to automate mathematical calculations and perform them more efficiently. This article covers the basic principles of the Euclidean algorithm and its application in the Python program. The principle of operation of the algorithm in various cases is explained through examples. In the future, attention can be paid to the issues of further development and optimization of this algorithm.

References:

1. Khojiyev J.Kh. Fayneleyib A.S. "Algebra and number theory". Tashkent, "Uzbekistan", 2001.
2. A.G. Kurosh. "Course of higher algebra". Tashkent, Teacher, 1976.
3. Saotov Yu.U. "Higher Mathematics". Part 3. Tashkent, 1993.
4. Jumaniyozov Q. Methods of solving examples and problems in mathematics. Ed. TDPU. T-2014.
5. Shodmonova Sh., Mirsagatova N., Ibragimova G., Mirsolieva M. "Educational Technologies" Methodological Guide, T. - 2011. 35.
6. R. Iskandarov, R. Nazarov. "Algebra and Number Theory", Part 1-2. Tashkent, 1963.
7. R. Iskandarov. "Higher Algebra", Part 1. Tashkent, 1963.
8. Saotov Yu.U. "Higher Mathematics". Part 2. Tashkent, 1991.

ent, 1991.