CONTINUED FRACTION ARITHMETIC IMPLEMENTATION WITH GOSPER ALGORITHM

Mert Esen

Kevser Yolcu

Continued Fractions

Continued fractions(CF) offer a means of concrete representation for arbitrary real numbers. The continued fraction expansion of a real number is an alternative to the representation of such a number as a (possibly infinite) decimal. [1]

Consider, for example, the rational number $\frac{415}{93}$, which is around 4.4624. As a first approximation, start with 4, which is the integer part; $\frac{415}{93}$ = 4 + $\frac{43}{93}$. Note that the fractional part is the reciprocal of $\frac{93}{43}$ which is about 2.1628. Use the integer part, 2, as an approximation for the reciprocal to obtain a second approximation of 4 + $\frac{1}{2}$ = 4.5;

$\frac{93}{43}$ = 2 + $\frac{7}{3}$. The remaining fractional part, $\frac{7}{43}$, is the reciprocal of $\frac{43}{7}$, and $\frac{43}{7}$ is around 6.1429. Use 6 as an approximation for this to obtain 2 + $\frac{1}{6}$ as an approximation for $\frac{93}{43}$ and 4 + $\frac{1}{2+\frac{1}{6}}$, about 4.4615, as the third approximation; $\frac{43}{7}$ = 6 + $\frac{1}{7}$. Finally, the fractional part, $\frac{1}{7}$ is the reciprocal of 7, so its approximation in this scheme, 7, is exact ($\frac{7}{1}$ = 7 + $\frac{0}{1}$) and produces the exact expression 4 + $\frac{1}{2+\frac{1}{6+\frac{1}{7}}}$ for $\frac{415}{93}$. In another way [4, 2, 6, 7]. [2]

Gosper Algorithm and Continued Fraction Arithmetic [4]

It is possible to perform arithmetic directly on continued fractions. It is possible to do this with infinite continued fractions because, unlike traditional decimal or binary arithmetic, continued fraction arithmetic yields the most significant number first. [3]

Let say X, Y, and Z are continued fractions;

X = $[x_0, x_1, x_2, \dots, x_n]$,

Y = $[y_0, y_1, y_2, \dots, y_n]$,

Z = $[z_0, z_1, z_2, \dots, z_n]$.

We want to apply four basic operations

(+, -, /, *) between X and Y to find result Z.

In Gosper Algorithm (GA) a, b, c, d, e, f, g, h are coefficients that change according to operation. We represent a, b, c, d, e, f, g, h as a matrix,

$$\begin{pmatrix} a & b & c & d \\ e & f & g & h \end{pmatrix}$$

They will change according to operation needs to apply.

Addition: $\begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$

Substraction: $\begin{pmatrix} 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$

Division: $\begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}$

Multiplication : $\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$

GA uses these coefficients to generate next steps when calculating arithmetic between CF. $x_0, x_1, x_2, \dots, x_n$ and $y_0, y_1, y_2, \dots, y_n$ will be used as inputs, and output terms will be result of Z as $[z_0, z_1, z_2, \dots, z_n]$ according to rules below:

1) If $\frac{a}{e} = \frac{b}{f} = \frac{c}{g} = \frac{d}{h}$ is true, keep $z_k = \frac{a}{e}$ which is k represents … starts with 0, and apply output formula to generate next matrix(next a, b, c, d, e, f, g, h).

$$\begin{pmatrix} e & f & g & h \\ a - ez_k & b - fz_k & c - gz_k & d - hz_k \end{pmatrix}$$

2) If $\frac{a}{e}, \frac{b}{f}, \frac{c}{g}, \frac{d}{h}$ are not number like ∞ or $\frac{0}{0}$, we have to stop calculating (there should be no result for all of them). In this step our result will be output terms we find before as Z = $[z_0, z_1, z_2, \dots, z_n]$.

3) If $|\frac{b}{f} - \frac{a}{e}| > |\frac{c}{g} - \frac{a}{e}|$ is true apply input formula below for x_k.

$$\begin{pmatrix} b & a + bx_k & d & c + dx_k \\ f & e + fx_k & h & g + hx_k \end{pmatrix}$$

4) Otherwise apply input formula below for y_k.

$$\begin{pmatrix} c & d & a + cy_k & b + dy_k \\ g & h & e + gy_k & f + hy_k \end{pmatrix}$$

Check these rules in every step orderly.

Gosper Algorithm Implementation

In our program we basically use rules we describe previous part. The program takes 2 fraction number and calculates continued fraction representation of it. Applies arithmetic operation between these two numbers.

The project can reachable from Github

https://github.com/kevseryolcu/ContinuedFractionArithmetic

To run the program run CFArithmeticsApp.java. You should give two fraction number as input and operation as string.

```
Input first fraction:
13/11
Input second fraction:
1/2
Input arithmetic operation(add, sub, mul, di
add
```

It will generate and display all GA paths.

```
Results:
Y Input: 0
1 0 0 1
0 0 1 0
----------------
Y Input: 2
0 1 1 2
1 0 2 0
----------------
X Input: 1
1 1 2 3
0 1 0 2
----------------
X Input: 5
1 6 3 17
1 5 2 10
----------------
Output: 1
1 5 2 10
0 1 1 7
----------------
Y Input: null
2 10 2 10
1 7 1 7
----------------
X Input: 2
10 22 10 22
7 15 7 15
----------------
Output: 1
7 15 7 15
3 7 3 7
----------------
Output: 2
3 7 3 7
1 1 1 1
----------------
X Input: null
7 7 7 7
1 1 1 1
----------------
Output: 7
1 1 1 1
0 0 0 0
----------------
continued fraction: [1; 1, 2, 7]
fraction: 37/22
```

References

1- William F. Hammond, Continued Fractions and the Euclidean Algorithm, 1997
2- Continued fraction. In Wikipedia. from https://en.wikipedia.org/wiki/Continued_fraction
3- Luther Tychonievich, Continued Fractions, 2011
4- Arithmetic with Continued Fractions, Mark Jason Dominus, 2005