Egyptian Fraction Decomposition: Rational Functions as Sums of Polynomial Reciprocals

Tim Banks, Dr. Neil Epstein, Kalkedan Malefia, and Kaitlyn Sullivan

Mason Experimental Geometry Lab

December 6, 2024

Introduction

The Ancient Egyptians expressed proper fractions as sums of unit fractions with a numerator, 1, and a positive integer as its denominator. Our project explores the application of this unique design on "proper" polynomial fraction, $\frac{f}{g}$,where $\deg(f) \leq \deg(g)$. However, "improper" polynomial fraction where the $\deg(f) > \deg(g)$ can not be written Egyptianally. Although Egyptian fractions have been around for millennia, we were inspired by algorithms by mathematicians such as Pierce, Engel, Fibonacci, and Golomb to express rational numbers as Egyptian fractions.

- The **Pierce** and **Engel** decompositions express fractions as a sum (or alternating sum) of reciprocals where the denominators are multiples of the original denominator.
- **Fibonacci's Greedy Algorithm** iteratively subtracts the largest unit fraction from the given fraction until the remainder itself becomes a unit fraction.
- **Golomb's Algorithm** repeats the process of taking the multiplicative inverse of the denominator modulo the numerator, continuing until only a unit fraction remains.

These algorithms form the foundation for extending Egyptian fraction decomposition to rational functions. By imitating these three methods–Pierce-Engel, Fibonacci, and Golomb– on polynomial fractions, we were able to analyze the limitations and patterns of each algorithm.

Early History

Egyptian fraction notation was developed in the Middle Kingdom of Egypt around 2000 BCE, appearing in many early texts but wasn't improved until the time of the Rhind and Moscow Mathematical Papyri, around 1800 BCE. A hieroglyph character that looks like a mouth or oval was placed above a number to indicate the reciprocal of that number. They also used special symbols to represent common fractions such as $\frac{2}{3}$.

Our Process

We began by analyzing the Pierce-Engel algorithm for rational functions as documented by Dr. Epstein. This inspired us to implement these algorithms using computer programs like Mathematica to visualize how rational functions could be expressed as sums of polynomial reciprocals. We developed Mathematica notebooks to evaluate the Pierce-Engel method, Fibonacci-style algorithms, and Golomb-style algorithms. These tools expedited our experimentation, enabling us to focus on establishing bounds on the number of terms and degrees in the decompositions while comparing the effectiveness and elegance of the decompositions.

Theorem (Pierce-Engel Style (2))

Let $f, g \in k[x]$ where k is a field. Assume that $\deg(f) \leq \deg(g)$ then there is a uniquely determined list of nonzero polynomials $h_0, h_1, \ldots, h_n$ such that $n \leq \deg(f)$, $0 < \deg(h_i) < \deg(h_{i+1})$ whenever $0 \leq i < n$, and

$$\frac{f}{g} = \frac{1}{h_0} - \frac{1}{h_0 h_1} + \ldots + \frac{(-1)^n}{h_0 h_1 \ldots h_n}$$

Remarks

This algorithm utilizes the Division Algorithm for Polynomials. Given $g = f h_0 + r_0$, we can derive the following, $\frac{1}{h_0} = \frac{f}{g} + \frac{r_0}{g h_0} \Leftrightarrow \frac{f}{g} = \frac{1}{h_0} - \frac{1}{h_0}(\frac{r_0}{g})$ and $\frac{1}{h_1} = \frac{r_0}{g} + \frac{r_1}{g h_1} \Leftrightarrow \frac{r_0}{g} = \frac{1}{h_1} - \frac{1}{h_1}(\frac{r_1}{g})$. As this algorithm continues, the results take the form $\frac{(-1)^i}{h_0 h_1 \ldots h_{i-1}}(\frac{r_{i-1}}{g})$.
Where the degree of each remainder is $\deg(f) > \deg(r_0) > \deg(r_1) > \ldots > \deg(r_n)$. Additionally, Pierce-Engel decomposition can be written as an ascending continued fraction.

Example

In the following output code of polynomial decomposition for $\frac{f}{g}$, the $h_0, h_1, \ldots h_n$ take the form written above and $n \leq \deg(f)$.

```
f = x^2 + 2 x + 3;
g = x^3 + 3 x^2 + x + 1;
```

$$= \frac{1}{1+x} + \frac{16}{(1+x)(-1+10x+4x^2)} - \frac{18}{(1+x)(-1+10x+4x^2)(1+x+3x^2+x^3)}$$

```
f = 2 x + 3;
g = x^3 + 3 x^2 + x + 1;
```

$$= \frac{8}{-5+6x+4x^2} - \frac{23}{(-5+6x+4x^2)(1+x+3x^2+x^3)}$$

```
f = 11 x + 3;
g = 13 x^2 + 15 x + 8;
```

$$= \frac{121}{126+143x} - \frac{590}{(126+143x)(8+15x+13x^2)}$$

Theorem (Fibonacci Style)

Let $f, g \in k[x]$, where k is a field, of degrees d, e respectively, where $d \leq e$. Then there is a list of polynomials $h_0, h_1, \ldots, h_n$ such that $n \leq d$, $\deg(h_0) = e - d$, $\deg(h_i) \geq i + 2(e-d) + \sum_{j=1}^{i-1} \deg(h_j)$ whenever $1 \leq i \leq n$, and

$$\frac{f}{g} = \frac{1}{h_0} - \frac{1}{h_1} + \ldots + \frac{(-1)^n}{h_n}.$$

Remarks

This algorithm utilizes the Division Algorithm for Polynomials.
Given $g = f h_0 + r_0$ then $\frac{f}{g} = \frac{1}{h_0} - \frac{r_0}{g h_0} \Leftrightarrow \frac{r_0}{h_0 g} = \frac{1}{h_0} - \frac{r_1}{g h_0 h_1}$ As this algorithm continues, the results take the form $\frac{r_i}{g h_0 \ldots h_i} = \frac{1}{h_{i+1}} - \frac{r_{i+1}}{g h_0 \ldots h_{i+1}}$ where, $\deg(f) > \deg(r_0) > \deg(r_1) > \cdots > \deg(r_n)$. Additionally, the resulting decomposition becomes increasingly complex and burdensome. It grows both in size and intricacy, making it less elegant and more challenging to work with compared to other methods.

Example

In the following output code of polynomial decomposition for $\frac{f}{g}$, notice the degrees of polynomials $h_0, h_1, \ldots, h_n$ grow more rapidly. Furthermore, the algorithm produces large numbers resulting in a complex decomposition that lacks efficiency.

```
f = x^2 + 2 x + 3;
g = x^3 + 3 x^2 + x + 1;
```

$$= \frac{1}{1+x} + \frac{1}{\frac{7}{32} + \frac{9x}{16} + \frac{7x^2}{8} + \frac{x^3}{4}} - \frac{18}{7+32x+92x^2+164x^3+207x^4+162x^5+60x^6+8x^7}$$

```
f = 11 x + 3;
g = 13 x^2 + 15 x + 8;
```

$$= \frac{1}{\frac{126}{121} + \frac{13x}{11}} - \frac{590}{1008+3034x+3783x^2+1859x^3}$$

```
f = x^2 + 1;
g = x^3 + 2 x^3 + x^2 + 1;
```

$$= \frac{1}{1+3x} + \frac{3}{3+x+6x^2+9x^3} - \frac{3}{3+10x+12x^2+46x^3+66x^4+54x^5+108x^6+81x^7}$$

Theorem (Golomb Style Algorithm for Polynomials)

Let $f, g \in k[x]$, where k is a field, of degrees d, e respectively, where $d \leq e$. Suppose $\gcd(f, g) = 1$, then there exist polynomials $g_1, \cdots, g_n$ where $n \leq d$, $\deg(g) > \deg(g_1) > \cdots > \deg(g_n)$, and a nonzero $c \in k$, such that

$$\frac{f}{g} = \frac{1}{g g_1} + \frac{1}{g_1 g_2} + \cdots + \frac{1}{g_{n-1} g_n} + \frac{1}{c g_n}$$

In particular, there are at most $d + 1$ terms, and all denominators have degree $\leq 2e - 1$

Remarks

This involves solving equations with $\deg(f_{i+1}) < \deg(f_i)$ and $\deg(g_{i+1}) < \deg(g_i)$ of the form

$$g_{i+1} f_i = f_{i+1} g_i + 1 \Leftrightarrow \frac{f_i}{g_i} = \frac{f_{i+1}}{g_{i+1}} + \frac{1}{g_i g_{i+1}}$$

This is possible because of the Sylvester Resultant.
In addition to the bound on number of terms and the bound on maximum denominator polynomial degree, the Golomb algorithm also produces "friendlier", smaller coefficient values in comparison to the other algorithms.

Example

In the following code outputs of polynomial decompositions for $\frac{f}{g}$, notice the maximum degree of the denominators is 2e - 1. Notice the difference in terms when the degree of f decreases.

```
f =x^2+2x+3;
g = x^3+3x^2+x+1;
```

$$= -\frac{9}{4\left(-\frac{9}{4}-\frac{9x}{4}\right)} + \frac{1}{\left(-\frac{9}{4}-\frac{9x}{4}\right)\left(\frac{5}{18}-\frac{5x}{18}-\frac{x^2}{9}\right)} + \frac{1}{\left(\frac{5}{18}-\frac{5x}{18}-\frac{x^2}{9}\right)(1+x+3x^2+x^3)}$$

```
f =2x+3;
g = x^3+3x^2+x+1;
```

$$= -\frac{8}{23\left(\frac{5}{23}-\frac{6x}{23}-\frac{4x^2}{23}\right)} + \frac{1}{\left(\frac{5}{23}-\frac{6x}{23}-\frac{4x^2}{23}\right)(1+x+3x^2+x^3)}$$

Now notice the comparative simplicity of this decomposition result to the results of other algorithms.

```
f = 11 x+3;
g = 13 x^2+15 x+8;
```

$$= -\frac{121}{590\left(-\frac{63}{295}-\frac{143x}{590}\right)} + \frac{1}{\left(-\frac{63}{295}-\frac{143x}{590}\right)(8+15x+13x^2)}$$

Conclusion and Future Work

In creating computer scripts for the Pierce-Engel, Fibonacci, and Golomb style algorithms, future computations and the analysis process has become much easier. This fulfills the goal for this semester that we decided on in early fall: to make experimenting and analyzing the behavior of these algorithms faster. A complex hand computation has been reduced to seconds using the code we have developed this semester. This makes the work of potential future semesters much more streamlined, allowing all time to be dedicated to analysis rather than spending time on long hand computations.
For the future, the main focus would be determining minimum and maximum degree bounds for the denominators as well as bounds on the minimum and maximum number of terms for each algorithm. So far, we have found in the Pierce-Engel and Fibonacci algorithm that as the number of terms increases, so does the maximum degree of the denominators. For Fibonacci in particular, the degree increases greatly as the terms get large, creating very large degrees in the polynomial reciprocals. In comparison, the Golomb algorithms has a bound on the maximum degree. The maximum degree of a denominator polynomial is $\leq 2\deg(g) - 1$ and decreases each time, creating a very efficient algorithm.
It can be concluded that the most efficient algorithm is the Golomb style algorithm due to it's bound on degrees and the fact that the degrees decrease with each iteration. In comparison, the Pierce-Engel and Fibonacci terms increase with each iteration, with the Fibonacci increasing greatly as the number of terms increases. For all algorithms, the number of summands is at most equal to $\deg(f) + 1$.
In future semesters, exploration of more algorithms would further expand knowledge on efficient methods of rational function decomposition, allowing for more observation on numbers of terms, degree bounds, and other behaviors of algorithms. Work will be continued with Golomb's algorithm as we write a paper on our findings regarding the degree bound and the code that we have completed this semester.

Acknowledgements

Special thanks to Dr. Neil Epstein for being the faculty member overseeing the research project, as well as our graduate mentor Tim Banks.

References

1. Crawford, T. (n.d.). Egyptian fractions. Tom Rocks Math. https://tomrocksmaths.com/wp-content/uploads/2023/08/m-s-egyptian-fractions.pdf
2. Epstein, N. (2024). Rational Functions as Sums of Reciprocals of Polynomials. The American Mathematical Monthly, 131(9), 794–801. https://doi.org/10.1080/00029890.2024.2379749
3. Solomon W. Golomb. (1962) An Algebraic Algorithm for the Representation Problems of the Ahmes Papyrus, The American Mathematical Monthly, 69(8), pp. 785-786.