Assignment 1

Design and Analysis of Algorithm

IDAA432C

Automate the division of any generic polynomial by linear polynomial

---

Submitted By:
Akshay Gupta **(IIT2017505)**
Naman Deept **(IIT2017507)**
Snigdha Dobhal **(IIT2017506)**

March 27, 2019

Automate the division of polynomial by a linear polynomial

Akshay Gupta IIT2017505, Naman Deept IIT2017507 Snigdha Dobhal IIT2017506

March 27, 2019

1 Abstract

This paper introduces different approach to automate the polynomial division for the general case then goes on to find out the special case of dividing a generic polynomial by the linear polynomial and also emphasizes upon the optimal approach to the solution .

Keywords: *Long Division*, *Synthetic Division* ,*Ruffini's Rule*

2 Introduction

Long division: Long division approach is the basic approach to divide the two polynomials to get the desired resulting polynomial. This rule can even automate the task of any generic polynomial division with the **figure** (or the numerator) having higher degree than that of the **correspondence** (the denominator),but the only disadvantage here is that in this case , we have both greater time consumption as well as higher memory usage. Polynomial long division is an algorithm that implements the Euclidean division of polynomials, which starting from two polynomials A (the dividend) and B (the divisor) produces, if B is not zero, a quotient Q and a remainder R such that

$$A = B \times Q + R$$

and the degree of R is lower than the degree of B (can also be 0). These conditions uniquely define Q and R, which means that Q and R do not depend on the method used to compute them. The result R=0 occurs if and only if the polynomial A has B as a factor. Thus long division is a means for testing whether one polynomial has another as a factor, and, if it does, for factoring it out. For example, if a root r of A is known, it can be factored out by dividing A by $x - \beta$.

Synthetic division: is a method of performing Euclidean division of polynomials, with less writing and fewer calculations than occur with polynomial long division. We are now concentrated towards obtaining the optimum in space as well as time. This polynomial is indeed useful in conserving our space complexity by reducing the number of variables and even to some extent the time complexity by reducing the number of variables in our manipulations. But the only disadvantage is that we need to consider only the case of division with linear polynomials ,it is not valid for the generic case. i.e we can consider the correspondence for automation with the only polynomial of the form x - , where α can be any real number.

The advantages of synthetic division are that it allows one to calculate without writing variables, it uses few calculations, and it takes significantly less space on paper than long division. Also, the subtractions in long division are converted to additions by switching the signs at the very beginning, preventing sign errors,thereby reducing space complexity.

Synthetic division for linear denominators is also called as **Ruffini's rule**.

3 Proposed Method

Input: Given any polynomial :

$$p(x) = a_0 + a_1.x + a_2.x^2 + a_3.x^3 + ... + a_n.x^n$$

The objective is to automate the division of such polynomial with a linear polynomial $h(x) = x - \beta$. Here in the said approaches, it has been considered , the polynomial as an array of some real numbers with entries in descending order of the powers of the polynomial. that is , for the given polynomial $p(x)$ our visualization of the polynomial will be in the form of an array which will be :

$$A[p(x)] =< a_n, a_{n-1}, a_{n-2}, ..., a_1, a_0 >$$

Hence , accordingly finding the resulting array from this array and claiming it to be the another polynomial array in the same descending order.

Perspective of an array: Likewise it is the perspective function , introd cunighere the concept of perspective of an array ,It will be containing 1 lesser element than the actual array, and will be defined as:

$$P(A) = < \frac{a_{n-1}}{a_n}, \frac{a_{n-2}}{a_n}, \frac{a_{n-3}}{a_n}, ..., \frac{a_0}{a_n} >$$

Perspective of the array is also referred to as the normalization of the array.The advantage of having this perspective array is that it'll reduce the proneness while automation since the linear polynomial has coefficient of superior term as 1.

3.1 Approach for this problem

In th sicase , classical approach for the long division problem will be used ,A sample example is demonstrated below: Division of polynomial $p(x) = 14.x^3 + 0.x^2 + 24.x + 19$ with linear polynomial $h(x) = x + 2$. The hand automation will be as follows:

$$
\begin{array}{r|rrrr}
 & 14x^2 & -28x & + & 80 \\
\hline
x+2 & 14x^3 + & 0x^2 & +24x & + \quad 19 \\
 & 14x^3 + & 28x^2 & & \\
\cline{2-3}
 & & -28x^2 & +24x & \\
 & & -28x^2 & -56x & \\
\cline{3-4}
 & & & 80x & + \quad 19 \\
 & & & 80x & + \quad 160 \\
\cline{4-5}
 & & & & -141
\end{array}
$$

Here are the steps in dividing polynomials using the long division method:

1. Arrange the indices of the polynomial in descending order. Replace the missing term(s) with 0.
2. Divide the first term of the dividend (the polynomial to be divided) by the first term of the divisor. This gives the first term of the quotient.
3. Multiply the divisor by the first term of the quotient.
4. Subtract the product from the dividend then bring down the next term. The difference and the next term will be the new dividend. Note: Remember the rule in subtraction "change the sign of the subtrahend then proceed to addition".
5. Repeat step 2 4 to find the second term of the quotient. Continue the process until a remainder is obtained. This can be zero or is of lower index than the divisor.
6. If the divisor is a factor of the dividend, you will obtain a remainder equal to zero. If the divisor is not a factor of the dividend, you will obtain a remainder whose index is lower than the index of the divisor.

3.1.1 Polynomial Long Division:

Input:
Numerator (N) is an array of coefficient with i^{th} index denoting the degree
Denominator (D) is an array of coefficient of size 2 denoting the monic polynomial. polynomial
Algorithm

```
procedure DEGREE(p)
    if all the elements are less than 0 then
        return −∞
    end if
    return the index of last non-zero element of P
end procedure
```

```
procedure POLYNOMIALLONGDIVISION(N,D)
    // N,D,q,r are vectors
    if degree(D)<0 then
        return error
    end if
    q ← 0
    while degree(N)≥degree(D) do
        d ← D shifted right by (degree(N)-degree(D))
        q(degree(N) − degree(D)) ← N(degree(N))/D(degree(D))
        d ← d × q(degree(N) − degree(D))
        N ← N − d
    end while
    r ← N
    return (q,r)
end procedure
```

3.1.2 Synthetic Division:

In algebra, synthetic division is a method of performing Euclidean division of polynomials, with less writing and fewer calculations than occur with polynomial long division. It is mostly taught for division by binomials of the form

$x - a$, where a is any number (real number).

Synthetic division procedure for division of a given polynomial $p(x) = x^3 - 12.x^2 - 42$ with the linear polynomial $h(x) = x - 3$ is illustrated below:

1. Write the coefficients in array form like:

$$\begin{array}{|rrrr} 1 & -12 & 0 & -42 \\ & & & \\ \hline \end{array}$$

2. Negate the coefficients of the divisor.

$$-1x \quad +3$$

3. Write in every coefficient of the divisor but the first one on the left.

$$\begin{array}{c|rrrr} & 1 & -12 & 0 & -42 \\ 3 & & & & \\ \hline \end{array}$$

4. "Drop" the first coefficient after the bar to the last row.

$$\begin{array}{c|rrrr} & 1 & -12 & 0 & -42 \\ 3 & & & & \\ \hline & 1 & & & \end{array}$$

5. Multiply the dropped number by the number before the bar, and place it in the next column.

$$\begin{array}{c|rrrr} & 1 & -12 & 0 & -42 \\ 3 & & 3 & & \\ \hline & 1 & & & \end{array}$$

6. Perform an addition in the next column.

$$\begin{array}{c|rrrr} & 1 & -12 & 0 & -42 \\ 3 & & 3 & -27 & -81 \\ \hline & 1 & -9 & -27 & -123 \end{array}$$

7. Repeat the previous two steps and the following is obtained:

$$\frac{x^3 - 12x^2 - 42}{x - 3} = x^2 - 9x - 27 - \frac{123}{x - 3}$$

From our portion of synthetic division we can conclude from the above that any specific i^{th} index of the resulting array (polynomial result equivalent) obtained is given as considering our polynomial as $p(x)$ and linear monic polynomial as $h(x) = x - \beta$:
$r_i = \sum_{j=0}^{j=i} \beta^j . a_{n-i+j}$
where r_i denotes the i^th index of the resultant polynomial array.

3.1.3 Algorithm:

Input: Here Num is the given polynomial of any degree n
and the Den is any monic polynomial .

procedure SYNTHETICDIVISION(Num,Den)
$out \leftarrow copy(Num)$
$norm \leftarrow Den[0]$
for i in $length(Num) - length(Den) + 1$ **do**
$out[i] \leftarrow out[i]/norm$
$coeff \leftarrow out[i]$
for j in $length(Den)$ **do**
$out[i+j] \leftarrow out[i+j] - Den[j] \times coeff$
end for
end for
return out
end procedure

4 Experimental Results

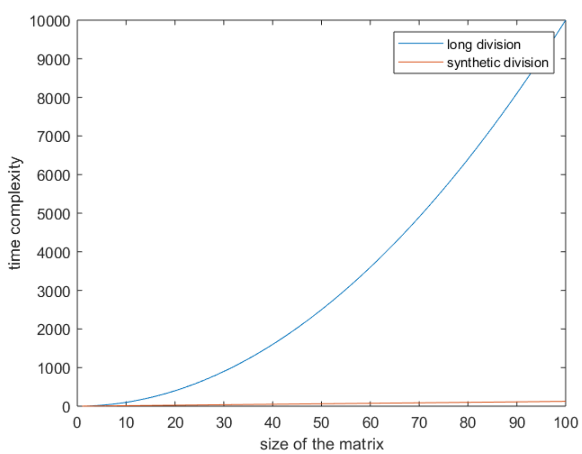

Figure 1: Graph showing the comparisons of the long division algorithm versus synthetic division algorithm

4.1 Complexity Analysis and Explanation

4.1.1 Polynomial Long Division

Consider the division to take place using long division of 2 arbitrary numbers ,in long division method we have a quadratic nature of growth with the raise in digits of the maximum of the numbers (after taking the modulus ,so that negative part will be excluded) .Hence the time taken in long division approach will be given as:

$$t_\Theta \propto (log_{10}max(|x|,|y|))^2$$

In the given problem statement , the denominator (D) is always a monic polynomial and hence of degree 1 while the numerator (N) could be of any degree n.
To calculate the coefficients of $\frac{N}{D}$, the algorithm traverses for $deg(N) - deg(D)$ times and then calculate the desired coefficients by traversing once again for degree(N) times . As in this case $deg(D)$ is 1 so time complexity will be proportional to that of $(deg(N) - 1)) \times deg(N)$.
In both the best case and the worst case , the specified algorithm traverses through this many times of iterations at least and hence this algorithm has a time complexity that of $\Theta(deg(N)^2)$.

4.1.2 Synthetic division

In our synthetic division approach we need to consider our resulting array whose any arbitrary i^th element is given by: $res_i = \sum_{j=0}^{j=i} \beta^j.a_{n-i+j}$.Where res_i denotes the i^th element of the resulting array. We are utilized only with a single loop where inside every loop we need to update the value of our sum with sum as $sum \times \beta$ where β is the value of the constant term of the monomial (linear polynomial) ,and after this we again need to add the i^th index of the corresponding dividend array to this sum, and this sum value is now assigned to the i^th index of the resulting array ,Since every iteration inside the loop is useful ,consider the first updating takes c_1 units of time ,second updating takes c_2 units of time, and assigning takes c_3 units of time . Since this runs on every iteration and the total number of iteration is the degree of the polynomial or the length of our numerator polynomial array (say n),we can say that our time t_θ is given by :

$$t_\Theta = (c_1 + c_2 + c_3).n$$
$$\text{Or, } t_\Theta \propto \text{n.}$$

Hence the complexity in this case will be $\Theta(n)$,here n is the degree of the polynomial or the length of the desired polynomial array of numerator.

5 Conclusion

Here indeed the better approach preferred the synthetic division approach in both the space as well as time constraint ,since it eliminates the usage of extra variables and also reduces our complexity from $\Theta(n^2)$ to $\Theta(n)$, where n is the size of our polynomial array, and similarly we can extend this concept to division of natural numbers as well wherein the complexity will be again reduced from $(log_{10}n)^2$ to $(log_{10}n)$ where n is the maximum of the numbers of $|x|$ and $|y|$.

6 References

1.Introduction to Algorithms(Second Edition)
Author:Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, Clifford Stein
2.Wikipedia: https://en.wikipedia.org/wiki/Synthetic division.
3.Wikipedia: https://en.wikipedia.org/wiki/Long division