Implementing Middle Product on FLINT

Daniel Scott, and Dr. Bill Hart

Introduction

Number theory is the branch of pure mathematics concerned with the properties of numbers in general, especially integers. Its origins are worldwide, and its current uses are extensive and diverse.

Fast Library for Number Theory (FLINT) is a C library for programmers wishing to perform calculations in Number Theory. Its main aims are:

- Asymptotically Fast Algorithms (algorithms that are extremely fast for very large numbers of calculations
- As fast or faster than all major competitors
- Extensively tested (making sure all algorithms give the correct answer)
- Extensively profiled (checking it is quick for all size and number of terms being calculated)

My part in this project was to research and help write algorithms using relaxed multiplication. The term relaxed multiplication was coined by Dr. Joris Van Der Hoeven in 1997 at Paris-Sud University. Since then he has published multiple papers on its implementation and use.

Before his work there were two types of algorithms for polynomial multiplication, Zealous and Lazy. Zealous uses the fastest modern algorithms, the main such being known as the Fast Fourier Transform (FFT). This transform calculates the Discrete Fourier Transform (DFT) quickly, and this is used to calculate the product in the fastest possible time. However Zealous algorithms lack the ability to use previous calculations to help future computations. Also, this algorithm type cannot have the range of its calculations extended without restarting the entire process. Lazy algorithms can be continued from their previous end point, and they also have the capacity to use earlier calculations to assist further calculations. However they tend to take longer to perform large scale calculations as they can't utilise fast algorithms such as FFT mentioned above. The purpose of relaxed multiplication is to combine the strengths of both methods.

I have been attempting to understand these papers and write up or rework relaxed algorithms in order that they may be used within FLINT.

Research

The first paper for which an algorithm was derived, was based on Dr. Joris Van Der Hoeven's 2003 paper 'Relax but don't be too lazy'. The idea I was most interested in was a relaxed multiplication algorithm whose method concerns carefully choosing the amount, position and order in which the coefficients are calculated, in order to have a fast relaxed algorithm.

Figure 1 is a graphical representation of the algorithm discussed here. The concept involves having the axes represent the coefficients of two power series *f* and *g*. The boxes represent sets of coefficients that are all calculated at the same time. The numbers within the boxes show which boxes are calculated at any one time. Within this method there is a trick based upon work of Professor Anatolii Alexeevich Karatsuba along with Dr Yu. P. Ofman (in 1962). When working with many digit numbers, multiplication is significantly more difficult and time consuming to perform than addition. Due to this fact, the Karatsuba trick performs 3 multiplications along with multiple additions instead of 4 multiplications, in order to save time. The Karatsuba trick allows us to, with specific boxes saved either temporarily or permanently in memory, to calculate all the coefficients in two specific, equally sized boxes simultaneously.

The second paper studied was Dr. Joris Van Der Hoeven's 2003 paper 'Relaxed multiplication Using the Middle Product' . When two power series are multiplied the result is truncated to order n. However most algorithms that calculate the first n coefficients of a power series product, will in fact redundantly calculate the next n coefficients. This wastes time so I did research to study algorithms that only calculate the first n coefficients that can also use the aforementioned fast algorithms.

The middle product is another Karatsuba style trick. Figure 2 illustrates the method of this algorithm, to calculate a parallelogram instead of the box shape of the first algorithm. Hence it can be used to only calculate the first n coefficients of the product. The algorithm conceived to use this middle product technique is one that breaks down the required set of products into multiple blocks. Middle products are iteratively performed on each block until every coefficient of degree less than n is calculated. A large difference between this algorithm and the first discussed is that all the coefficients of one of our power series have to be fixed, i.e. determined before the algorithm begins. Also the algorithm takes a slightly different form depending on whether n is an odd (Figure 3) or even (Figure 4) integer.

Research

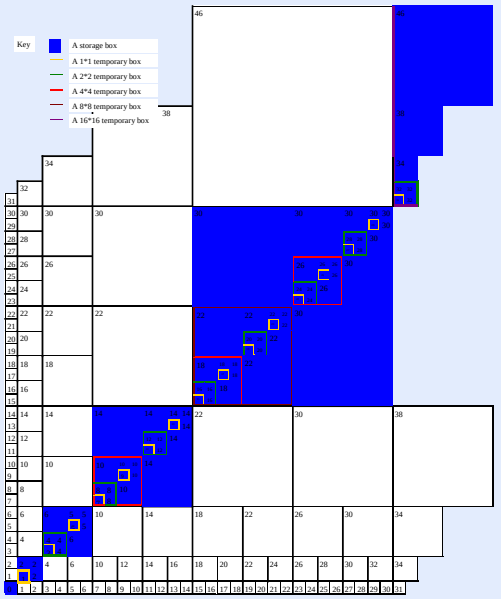

Figure 1: A diagram to show the size, position and order in which the required coefficients are calculated.

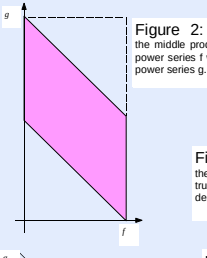

Figure 2: A representation of the middle product of an order n power series f with an order 2n-1 power series g.

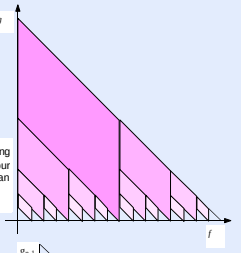

Figure 3: A diagram demonstrating the middle products that calculate our truncated product of power series (van der Hoeven, 2003).

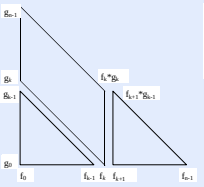

Figure 4: A diagram to show the specific terms dealt with by the algorithms three 'if' functions in the case that n is odd.

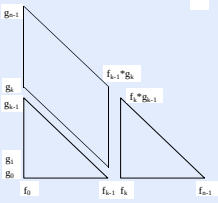

Figure 5: A diagram to show the specific terms dealt with by the algorithms three 'if' functions in the case that n is even.

Discussion

The first algorithm discussed has a very specific use in mind, along with other potential applications. There are Zealous algorithms that are in fact asympotically faster at polynomial multiplication. However within algebraic number theory (a strand of number theory that FLINT specialises in) there is the topic of p-adic arithmetic that FLINT hopes to incorporate. p-adic numbers are a different number system from conventional numbers. Terms within it are represented by power series and thus on a computer these are truncated power series. In order to use these p-adic numbers, a certain degree of accuracy is needed for each p-adic number, depending on the required function. Yet this degree of accuracy cannot be pre-determined. So Zealous algorithms based around FFT is severely disadvantaged in this use, as any lack of accuracy leads to having to rerun the entire process to a higher degree of accuracy. Yet the relaxed method discussed here can be restarted without needing to recalculate anything already processed, thus leading to large scale time savings in this important process.

The second algorithm could also be very helpful in the p-adic number field described above. Decisions on which algorithm to use being determined by what information is known and what is desired from the algorithm. Besides that it also has potential in two other areas. The first is in power series division and the square rooting of power series. Although this algorithm uses more memory than division based on tricks of Graeffe and Kung, it should be an asymptotically faster method.

The second area that this algorithm has potential in is fast exponentiation of power series. A transcendental function is one that can only be expressed fully by use of an infinite series. For ones in which a power series is known, such as the trigonometric functions or logarithms, this algorithm could be of use.

Further investigation is necessary in order to implement any of the aforementioned techniques, however, the use of the Middle Product algorithm is essential for the enhancement of the FLINT programming library.

Conclusions

- FLINT has made great progress in single limb integer arithmetic, multi-precision integer arithmetic and polynomial arithmetic over multi-precision integers.
- The relaxed multiplication algorithm I have researched, inspired by Dr. Joris Van Der Hoeven's work, has considerable potential in any power series multiplication in which it could be necessary to either expand the previous work or improve upon its accuracy.
- The Middle Product algorithm investigated displays hope for improving FLINT's ability to perform power series division or square rooting at higher speeds than previously attainable.
- The Middle Product algorithm also shows promise for asymptotically fast exponentiation of power series, a process used throughout Number Theory calculations.
- Participation in this research has given me a new found understanding and appreciation of both the field of Number Theory and modern computer practises.

Daniel Scott
D.Scott@warwick.ac.uk

Dr Bill Hart
B.Hart@warwick.ac.uk