EFFICIENT NTT ALGORITHMS FOR PRIME NUMBERS

Ryszard Stasiński

Instytut Elektroniki i Telekomunikacji
Politechnika Poznańska
Piotrowo 3A, PL-60-965 Poznań, POLAND

RESUME

Dans le conférence on présente la construction des algoritmes pour calculer NTTs d'après l'idée de Rader. Pour N=p, où p est un nombre prime, non-Fermat, les algoritmes besoins seulement $O(p(d_1+d_2+...d_t))$ opérations, au contraire de $O(pd_1d_2...d_t)$ pour le moyen direct, d_i sont les mutuellement primes diviseurs de p-1. Ça signifie, que construits d'après cette idée algoritmes pour transformation de Mersenne ne sont pas moin effectifs que ceux pour autres NTTs. En général, d'après l'idée on peut construiredes améliores NTT modules, utilisés dans les "FFT" algoritmes pour NTT. Le conférence contient aussi quelques remarques concernant adoption pour NTT des algoritmes dérives originalement pour DFT.

SUMMARY

In the paper the construction of Rader's number theoretic transform algorithms is described. It is shown that for N=p being non-Fermat prime numbers the algorithms require significantly less operations than the known ones. Namely, the number of operations is reduced from $O(pd_1d_2...d_t)$ to $O(p(d_1+d_2+...+d_t))$, where d_i are mutually prime divisors of p-1. If applied to Mersenne transforms the approach results in algorithms which computational complexities are not higher than those for other NTTs, e.g. pseudo-Fermat ones. In general, the method can be used for improving small-N NTT modules in FFT-like algorithms. The paper contains also some general remarks on the transformation of DFT algorithms into those for number theoretic transforms.

1. INTRODUCTION

Number theoretic transforms (NTT) are used for efficient computation of convolutions using a special hardware. The most important NTTs are Mersenne and Fermat transforms [1]. It is well known that the arithmetic for Mersenne transform is especially simple. Unfortunately, for the simplest realizations its dimensions are equal to p, or 2p, p being prime numbers. For other NTTs the problem of efficient computation of transforms for sizes being prime numbers is analogous to efficient computation of small-N modules for the discrete Fourier transform (DFT).

The idea of NTT can be treated as an offspring of the idea of the DFT concept, so it can be expected that some DFT algorithms can be adapted to compute NTTs, too. Namely, for N being prime numbers Rader's algorithms are of interest [2], [3]. The same is true for polynomial transforms (PTs), and, indeed, it was shown that the use of Rader's PT algorithms for N being prime numbers results in dramatic reduction of the number of operations [4]. Similarly as for PTs, the computational complexity of a multiplication in an NTT strongly depends on the form of a multiplier (a shift vs a full ring multiplication), so, the adaption of DFT algorithms to the NTT case is linked with some limitations. Note that similarities between NTTs and PTs are not accidental, as it was shown that some NTTs can be treated as PTs for digits [1].

In the paper the construction of Rader's NTT algorithms is described. The approach consists in transforming the problem of computing an $N=p^r$-point DFT into that of computing $(p-1)p^s$-point convolutions, p is an odd prime, $s=r-1,r-2,\ldots,0$; [3]. It appears that if the convolutions are mapped into multidimensional ones on the basis of the rule from [5], the form of their coefficients remains the same as for NTTs, while the computation of the p-1-point convolution brakes down into t stages, where t is a number of mutually prime divisors of p-1. Then, the number of operations decreases from $O(pd_1d_2\ldots d_t)$ to $O(p(d_1+d_2+\ldots+d_t))$, $p-1=d_1d_2\ldots d_t$. The method gives very good results for N=p not being Fermat numbers.

2. ALGORITHMS

The most general definition of the DFT is the following one‡:

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{kn}, \quad k=0,1,\ldots,N-1. \tag{1}$$

where X(k), x(n), W_N are elements of a commutable ring, and W_N is a (primitive) root of unity of order N. In the case of NTTs this is a ring of integers modulo M≥N, or, sometimes, its extended version [6].

The most important feature of NTTs is that for some M W_N are simply powers of 2:

$$W_N = 2^{kn}\mathrm{mod}\ M \tag{2}$$

Moreover, in the case of M being a Mersenne (not necessarily prime) number the computations are made simply in one's-complement arithmetic. Notice, however that as:

$M = 2^p - 1$, p is prime

$2^p \bmod M = 1$, and $2^r \bmod M \neq 1$ if $0<r<p$(3)

hence N in (1) N=p. Till now this fact was taken as an important limitation, as efficient NTT algorithms for N being prime numbers were not known. The use of more complicated arithmetics, and/or other M solve the problem only partially. Namely, in the case of NTTs the choice of N values is strongly restricted, and independent of the computational complexity criterion. [1], [6].

The Rader's DFT algorithm [2] exist for N being powers of prime numbers [3], [7]. It consists in an observation that:

$$W_N^{kn} \equiv W_N^{(kn)\mathrm{mod}\ N} \tag{4}$$

which means that calculations of the product kn can be made in a ring modulo N. If N is (a power of) a prime number, the ring becomes an (extended) Galois field. We are interested in cases when $N=p^r$, and p is an odd prime number. The Rader's algorithms for $N=2^r$ exist [7], in the case of NTTs they are, however, neither effective, nor simple. For such N the elements of the field not being divisors of zero form a multiplicative group $\langle a_i \rangle$ which is cyclic, i.e.:

$$a_{i+m} = a_i a_m,\ i+m \text{ is taken modulo K}, \tag{5}$$

where K is the rank of the group. The divisors of zero are simply multiples of p, hence:

$$K = p^r - p^{r-1} = (p-1)p^{r-1} \tag{6}$$

So, the formula on the DFT (1) can be rewritten as follows:

$$X(a_k) = \hat{X}(a_k) + \sum_{n=0}^{K-1} x(a_n)W_N^{a_{n+k}} \tag{7a}$$

$$X(pk') = \sum_{n=0}^{N-1} x(n)W_N^{k'np}, \quad k'=0,1,\ldots,N/p-1; \tag{7b}$$

where:

$$\hat{X}(a_k) = \sum_{n'=0}^{N/p-1} x(pn')W_N^{kn'p} \tag{7c}$$

The summation in (7a) is equivalent to the K-point circular correlation, which can be computed using circular convolution algorithms. X(pk') and $\hat{X}(k)$ can be computed using N/p-point DFT algorithms [3].

Notice that P^{r-1} and p-1 numbers in (6) are mutually prime. It means that the convolution can be mapped into multidimensional $d_1 \times d_2 \times \ldots \times d_t \times p^{r-1}$-point one, where d_i are mutually prime [5], and:

$$p-1 = \prod_{i=1}^{t} d_i \tag{8}$$

The p^{r-1}-point convolution is de facto a polynomial product modulo cyclotomic polynomial for Z^N-1 [3], however, this fact need not be used here.

3. ARITHMETICAL COMPLEXITY OF ALGORITHM

Consider N=p. In this case (7):

$$X(pk') = X(0) = \sum_{n=0}^{N-1} x(n) \tag{9a}$$

$$\hat{X}(k') = x(0) \tag{9b}$$

If we compute circular convolutions directly, the coefficients of convolutions are [1], [5]:

$$W_N^{a_{m_i}D_i}, \quad m_i=0,1,\ldots,d_i-1;\ D_i=(p-1)/d_i \tag{10}$$

see (8), with $W_N=2$ (2). Any further transformation of algorithms causes that the coefficients become more complicated. This means that the computation of Rader's NTT algorithm requires 2(p-1) additions (9) plus operations due to direct computation of t-dimensional $d_1 \times d_2 \times \ldots \times d_t$-point circular convolution. An d-point convolution can be

‡ Some authors consider (1) as the definition of NTT. The DFT is then the NTT for complex numbers, see e.g. [6].

computed using d^2 shifts and $d(d-1)$ additions, so, the overall algorithm requires:

$$S(N=p) = (p-1) \sum_{i=1}^{t} d_i \text{ shifts, and} \qquad (11a)$$

$$A(N=p) = (p-1)[2+ \sum_{i=1}^{t} (d_i -1)] \text{ additions.} \qquad (11b)$$

In the case of direct method:

$$\sum_{\substack{n=0 \\ n\neq i}}^{N-1} W_N^{\,n} = -W_N^{\,i},\ i=0,1,\ldots,N-1; \qquad (12)$$

which means that (1):

$$X(N-1) = x(0) - \sum_{k=0}^{N-2} \sum_{n=1}^{N-1} x(n) W_N^{\,kn} \qquad (13)$$

the fact was used in [1] for improving PT algorithms. Taking into account (13) the direct method results in:

$$S(p) = (p-1)(p-2) \text{ shifts, and} \qquad (14a)$$

$$A(p) = (p-1)p \text{ additions.} \qquad (14b)$$

Comparing (11) and (14) we can see that the Rader's NTT algorithm require asymptotically $O(p\Sigma_i d_i)$ operations, in contrast to $O(p\Pi_i d_i)$ for the direct method. Table I shows that indeed, except for N being Fermat prime numbers improvements due to Rader's NTT algorithm are dramatic. Reductions of numbers of operations are especially big when p-1 has many small divisors, e.g. for p=13, 31, 61, 71, 127, and 211. For Fermat prime numbers Rader's NTT algorithm is identical to the "ordinary" direct method, hence, the results are somewhat worse than those implied by (14).

The Rader's NTT algorithms for powers of a prime are not interesting here. Namely, they contain p-1 (in fact p-2 [4]) p^{r-1}-point, and p^{r-1} (p-1)-point circular convolutions to be computed, $N=p^r$. For the FFT-like algorithms the operations consist of rp^{r-1} (p-1)-point algorithms. Of course, p has no divisors, so, even for r=2 FFT-like algorithms are better than Rader's ones, see also [4].

4. SUMMARY AND CONCLUSION

In the paper the construction of number theoretic transform algorithms using the idea of Rader is described. It is observed that if N=p is an odd non-Fermat prime number the approach results in a class of algorithms having computational complexity of rank $O(p\Sigma_i d_i)$, where d_i are mutually prime divisors of p-1, in contrast to $O(p\Pi_i d_i)$ for direct method. As it is shown in Table I, the new algorithms are really very efficient, especially for big numbers of divisors of p-1.

The introduction of Rader's NTT algorithms causes that the computational complexity of long Mersenne transforms reduces to the level characteristic of other NTTs. Consider, for example, the 68-point pseudo-Fermat transform [1]. For the structure of the prime factor algorithm (2x2)x17 it requires 4x240 shifts due to 17-point algorithm, Table I, plus (2-1)17 ones for rotation factors, which gives 977 shifts, and 2(2x34)+4x272=1224 additions, Table I. As can be seen, the algorithm is worse than the Mersenne ones for N=61, 67, 71, and only slightly better than that for N=73. Additionally, the pseudo-Fermat transform requires a special stage of final reductions, and has approximately two times smaller number of effective bits of results [1]. Of course, the idea may be used for improving non-Mersenne NTTs, too.

T A B L E I

Numbers of operations for NTT algorithms for N=p, p is a prime number.

p	p-1	Σd_i	Rader (11) shifts/adds	Direct (14) shifts/adds
2	1	1	1/2	0/2
3	2	2	4/6	2/6
5	4	4	16/20	12/20
7	2x3	5	30/30	30/42
11	2x5	7	70/70	90/110
13	4x3	7	84/84	132/156
17	16	16	256/272	240/272
19	2x9	11	198/198	306/342
23	2x11	13	286/286	462/506
29	4x7	11	308/308	756/812
31	2x3x5	10	300/270	870/930
37	4x9	13	468/468	1260/1332
41	8x5	13	520/520	1560/1640
43	2x3x7	12	504/462	1722/1806
47	2x23	25	1150/1150	2070/2162
53	4x13	17	884/884	2652/2756
59	2x29	31	1798/1798	3306/3422
61	4x3x5	12	720/660	3540/3660
67	2x3x11	16	1056/990	4290/4422
71	2x5x7	14	980/910	4830/4970
73	8x9	17	1224/1224	5112/5256
79	2x3x13	18	1404/1326	6006/6162
89	8x11	19	1672/1672	7656/7832
103	2x3x17	22	2244/2142	10302/10506
127	2x9x7	18	2268/2142	15750/16002
211	2x3x5x7	17	3570/3150	43890/44310

[1] H.J. Nussbaumer, "Fast Fourier transform and convolution algorithms", Springer-Verlag, 1981.

[2] C.M. Rader, "Discrete Fourier transform when the number of data samples is prime", Proc. IEEE, vol. 56, 1968, pp. 1107-1108.

[3] R. Stasiński, "Easy generation of small-N discrete Fourier transform algorithms", IEE Proc., Pt. G, 1986, pp. 133-139.

[4] R. Stasiński, "Improved algorithms for computing polynomial transforms", Proc. DSP-87, Florence, Sept. 1987, pp. 93-96.

[5] R.C. Agarwal, J.W. Cooley, "New algorithms for digital convolution", IEEE Trans., vol. ASSP-25, 1977, pp. 392-409.

[6] J.-B. Martens, M.C. Vanwormhoudt, "Convolutions of long integer sequences by means of number theoretic transforms over residue class polynomial rings", IEEE Trans., vol. ASSP-31, 1983, pp. 1125-1134.

[7] R. Stasiński, Rader-Winograd's DFT algorithms for $N=2^r$", Proc. EUSIPCO-86, The Hague, Sept. 1986, pp. 81-84.