A New Method for "Exact Calculation" by a Digital Computer

(An Application of Modulo p Arithmetics)

H. TAKAHASI* AND Y. ISHIBASHI*

1. *Introduction*

The precision of calculations on a digital machine is usually limited by the length of the accumulator of that machine. In conventional multiplication, the less significant half of a double precision product is discarded after rounding-off. Sometimes, however, it is desired to maintain the result in its full length, that is, to hold an exact result without any rounding-off, and in this case a suitable subroutine must be used to handle the double length result.

In this paper, a new method is described which simplifies multiple-length calculations. Exact calculation is indispensable in problems related to the theory of numbers, such as, in the generation of primes, factorization, and integral solution of indefinite equations. It is not hard to find further examples of problems for which such exact calculations are effective. In the inversion of matrices, we often encounter ill-conditioned cases in which several significant digits are lost during calculation by an accidental cancellation. Floating point calculation is of little help, and may even be dangerous, because it tends to incorrectly imply high accuracy. Multiple precision computation is the only effective cure, and if the problem is extremely ill-conditioned, one must resort to treating all data as integers and perform an exact calculation without any rounding-off.

The method to be described here will offer a new technique that may be used to simplify the evaluation of complex expressions involving addition, subtraction and multiplication. The characteristic feature of this new method is to make all computations modulo an integer p. In other words, whenever a number greater than p is obtained during computation, the number is just replaced by its residue obtained by dividing it by p. Thus we have only to deal with numbers of single precision throughout, provided p lies within the capacity of the accumulator. Since the result thus obtained is the residue of the true value modulo p, it is necessary to reconstruct the true value by some means. This is done by repeating the same computation using several different moduli p_1, p_2, $\cdots$, p_l, and by solving a system of simultaneous congruence equations to determine the true value u from its residues u_1, u_2, $\cdots$, u_l. As it

* Faculty of Science, Tokyo Univ., Tokyo

is generally possible to estimate an upper limit for the true value of the expression, the number of different moduli needed to compute u unambiguously can be predetermined.

This method is, in principle, applicable only to expressions consisting of addition, subtraction and multiplication. Nevertheless, this method can be extended so as to apply to expressions involving division provided the final result can be shown to be an integer (Example; $n(n-1)(n-2)/6$). In evaluating such an expression, the "division" modulo p may be understood as a process for obtaining $x=a/b$ as the solution of an equation $bx\equiv a \pmod{p}$. It should be emphasized here that p must be a prime in order for division always to be possible.

It is interesting to note that the present method is in principle nothing but the well known checking method of "casting out nines". Let the problem to be checked be

$$1234+567=1801$$

The calculation goes as follow:

$$\begin{aligned} 1234&\equiv 1+2+3+4=10\equiv 1+0=1\\ 567&\equiv 5+6+7=18=9\equiv 0 \qquad (\mathrm{mod}\ 9)\\ 1801&\equiv 1+8+0+1=10\equiv 1 \end{aligned}$$

which just corresponds to the case $p=9$ of our method.

The method is based on the theorem:

If $A\equiv A_0,\quad B\equiv B_0 \pmod{p}$

then, $A\pm B\equiv A_0\pm B_0,\quad AB\equiv A_0B_0, \pmod{p}$

and one computes $A_0\pm B_0$ or A_0B_0 instead of computing $A\pm B$ or AB.

2. *The process of Calculation*

2.1. *Reduction Modulo p*

While the actual procedure to be followed is evident from the principle stated above, several remarks will be made in connection with practical aspects of the programs.

The process is carried out with the help of suitable subroutines which perform addition, subtraction, multiplication and, if necessary, inversion, all modulo p. Inversion modulo p means finding an x which satisfies $ax\equiv 1 \pmod{p}$. It is convenient to prepare an interpretive routine which incorporates all four of these basic arithmetic operations together with several branch instructions. These subroutines are required to operate on the operands x satisfying $0\leq x<p$, and to give a result so normalized as to lie in the same interval. Reduction of an arbitrary integer X modulo p amounts to finding a pair (q, x) which satisfies

$$X=pq+x \qquad (0\leq x<p). \tag{2.1·1}$$

This may be done quite simply if the computer has a division instruction which forms a correct remainder.

However, the following algorithm is sometimes more efficient if we choose p so that it is slightly less than the accumulator capacity. Let the length of the accumulator be s binary positions and choose p so that it may be written in the form

$$p=2^s-h, \tag{2.1·2}$$

where h is a small integer. Then, to obtain the residue of X modulo p add $X_u=\frac{X}{2^s}$ (upper half of X) multiplied by h to X_l (the lower half of X). That is,

$$X=2^sX_u+X_l \qquad 0\leq X_l<2^s \tag{2.1·3}$$

and hence

$$\begin{aligned} X'&=X_l+hX_u=X_l+(2^s-p)X_u \\ &=X+pX_u=X \pmod{p} \end{aligned} \tag{2.1·4}$$

If X' again overflows, that is, is larger than 2^s, then the process should be repeated, until the relation $0\leq X'<p$ is satisfied. This algorithm is even possible on a computer which is not equipped with a division instruction giving the correct remainder. A difficulty occurs if X' falls in the interval

$$p\leq X'<2^s. \tag{2.1·5}$$

Since in this case X' does not overflow, it does not change when the process is repeated. To avoid this difficulty, the initial value of X is made negative. Since now X_u always remains nonpositive, X never overflows in the positive direction. When X_u becomes zero, the process may be terminated.

2.2. *Calculation of a Reciprocal*

Two methods are known for the solution of the equation $ax\equiv 1 \pmod{p}$ for given a, one using Euclid's algorithm and the other using Fermat's theorem. While the quantity of arithmetic computation may be less using the former method, the program is simpler using the latter method, and for this reason the latter one was adopted.

If p is a prime, then by Fermat's theorem

$$a^{p-1}\equiv 1 \pmod{p} \tag{2.2·1}$$

so that x, the reciprocal of a, is obtained as

$$x\equiv a^{p-2} \pmod{p}. \tag{2.2·2}$$

Let the binary representation of $p-2$ be

$$p-2=\sum_{i=0}^{s-1} d_i 2^i \tag{2.2·3}$$

then x is determined by a recurrence formula:

$$\left.\begin{aligned} &p_0=1 \\ &p_i\equiv p_{i-1}^{2}\, a^{d_{s-i}} \qquad (\mathrm{mod}\ p) \\ &x\equiv p_s\,. \end{aligned}\right\} \tag{2.2·4}$$

That is, the binary digits of the integer p are tested one by one, while p_i is squared each time, and result is further multiplied by a whenever the tested digit was a "1". The process is particularly suited to binary computers. The flow chart for this process is shown in Fig. 1.

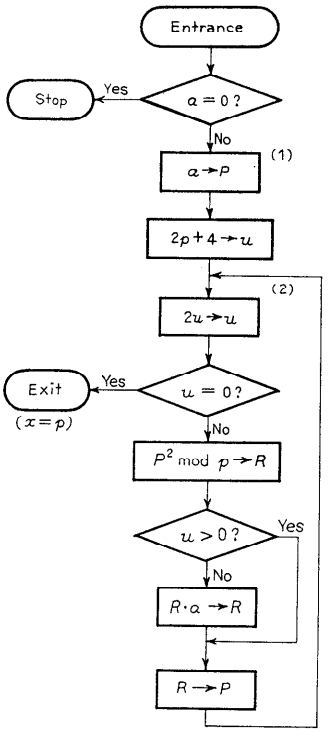

FIG. 1. Flow chart for the inversion routine modulo p.

Note (1) We assume $2^{s-2}+2<p<2^s$, and hence d_{s-1} is always 1, so that the first cycle of (2.2·4) can be done outside the loop.

(2) Successive digits of p are examined by shifting u left one position for each new digit and examining the sign digit.

(3) R denotes the lower extension of the accumulator, which in integer multiplication contains the product.

2.3. *Determination of the True Value*

Suppose y_i is obtained as a result of some calculation using p_i as modulus, so that

$$Y=y_i \pmod{p_i} \quad (i=1, 2, \cdots, n) \tag{2.3·1}$$

If the true value Y is smaller than every p, all y_i's will be equal to each other, and equal to Y. In general, however, Y will become larger than the accumulator capacity, and all y_i's will be different from each other. Then the system of simultaneous equations (2.3·1) is solved to find the definite integer Y. In order to determine Y uniquely, a condition such as

$$0 \leq Y < p_1 p_2 \cdots p_n \tag{2.3·2}$$

or

$$|Y| \frac{1}{2} p_1 p_2 \cdots p_n \tag{2.3·2'}$$

should be satisfied. The number n of primes necessary for the unambiguous determination of Y can be determined by estimating the upper limit of Y.

For the determination of Y, a sequence of Y_i's is obtained which satisfies:

$$\left.\begin{array}{ll} Y=Y_1 (=y_1) & (\text{mod } p_1) \\ Y=Y_2 & (\text{mod } p_1 p_2) \\ \cdots\cdots & \cdots\cdots \\ Y=Y_i & (\text{mod } p_1 p_2 \cdots p_i)\,. \end{array}\right\} \tag{2.3·3}$$

This can be done by finding a set of q_i's such that

$$\left.\begin{array}{l} Y_2-Y_1=q_2 p_1 \\ Y_3-Y_2=q_3 p_1 p_2 \\ \cdots\cdots \\ Y_i-Y_{i-1}=q_i p_1 p_2 \cdots p_{i-1}\,, \end{array}\right\} \tag{2.3·4}$$

and which satisfy

$$0 \leq q_i < p_{i-1}\,. \tag{2.3·5}$$

Then, since

$$Y_i-Y_{i-1} \equiv y_i-Y_{i-1} \pmod{p_i} \tag{2.3·6}$$

q_i is obtained from

$$q_i \equiv (y_i-Y_{i-1})(p_1 p_2 \cdots p_{i-1})^{-1} \pmod{p_i}\,. \tag{2.3·7}$$

Here $(p_1 p_2 \cdots p_{i-1})^{-1}$ means a reciprocal modulo p_i.

Then Y_n obtained by this process is equivalent to Y, or by rewriting

(2.3·4) we get

$$Y=(((\cdots(q_np_{n-1}+q_{n-1})p_{n-2}+q_{n-2})p_{n-3}+\cdots)p_1+q_1 \qquad (2.3\cdot8)$$

The q_i's are calculated by the equation (2.3·7) when the range of Y is given by (2.3·2), and y_i is determined by (2.3·5). It is however more convenient to choose q_i so that

$$|q_i| < p_{i-1}/2$$

when the range of Y is given by (2.3·2′). In either case, q_i for $i>i_0$ with a certain i_0 will become zero when an unnecessarily large number n is taken.

The calculation stated above requires some calculations on multiple precision numbers, namely the addition of multiple precision numbers, multiplication of a multiple precision number by a single precision number and the reduction of multiple precision numbers modulo p. The numbers $(p_1p_2\cdots p_{i-1})^{-1} \pmod{p_i}$ can be calculated in advance and be tabulated. Some p_i's, h_i's and $(p_1p_2\cdots p_{i-1})^{-1} \pmod{p_i}$ are listed in Table 1.

TABLE 1.

i	p_i	h_i	$(p_1p_2\cdots p_{i-1})^{-1} \bmod p_i$
1	343597 38337	31	1
2	38319	49	324508 63968
3	38307	61	73491 66249
4	38299	69	224977 16804
5	38289	79	211820 09803
6	38247	121	334384 41941
7	38227	141	179709 97103
8	38121	247	218289 14109
9	38059	309	272955 85313
10	38043	325	68501 32122
11	38011	357	104116 19371
12	37917	451	146250 41267
12	37869	499	5351 31435
14	37849	519	126517 94573
15	37837	531	206522 59502
16	37821	547	174572 11185
17	37813	555	140672 61446
18	37791	577	185955 21066
19	37777	591	101995 05182
20	37771	597	146932 64241

2.4. *Comparison with Conventional Multi-length Calculation*

Let us now compare the number of arithmetic operations needed in a given computation using the conventional multi-precision process. If it it assumed that an accuracy of n-tuple precision is desired, then the number of moduli is n in our method. Then n^2 multiplications are

needed in the ordinary multiplication of two numbers of n-tuple precision, while in our method, n multiplications are needed for each multiplication modulo p, and n further multiplications for reduction modulo p, so that $2n$ multiplications altogether are sufficient. When n is large, our method has a remarkable advantage. Another merit which should be emphasized is that the memory space required during calculation is the same as that required in single precision calculation, because the entire calculation can be performed using each p singly with successive p's being used one after another so that there is no need to store intermediate results for all the different moduli. Therefore our method is useful either for computers with a small memory capacity, or for problems requiring a large working memory.

3. *Exact Matrix Inversion as an Example of the Use of the Method*

The inversion of a matrix will be treated as an example, since it is one of the problems in which the effect is most conspicuous. We assumed that the elements of the given matrix are integers and that the elements of the inverted matrix are to be expressed as fractions, where the common denominator is the determinant of the given matrix and the numerators are its minor determinants. Let the set of equations associated with the given matrix be written in matrix form as follows:

$$\begin{pmatrix} a_{11}a_{12} & \cdots & a_{1n} \\ a_{21}a_{22} & \cdots & a_{2n} \\ \vdots & & \vdots \\ a_{n1}a_{n2} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} 1\,0 & \cdots & 0 \\ 0\,1 & \cdots & 0 \\ \vdots & & \\ 0\,0 & \cdots & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \tag{3·1}$$

On dividing the first row by a_{11}, one obtains

$$\begin{pmatrix} 1 & a_{12}/a_{11} & \cdots & a_{1n}/a_{11} \\ a_{21} & a'_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} 1/a_{11}\,0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \tag{3·2}$$

Similar elimination processes are repeated until the matrix on the left hand side becomes a unit matrix, with the result that the inverted matrix is obtained on the right hand side. Exact execution of this process in fractional representation usually becomes extremely complicated because of the large values of the quantities involved, and the present (mod p)-technique is very effective in reducing the computational burden. Here however, we shall employ a method whereby the use of fractions is altogether avoided. Specifically, we perform division modulo p wherever division occurs, in the sense already mentioned. Clearly, there is no need to worry about "overflow" and it is not necessary to search

for the largest element to use as a pivot. At the end of the elimination process we have a unit matrix on the left hand side.

The matrix thus obtained on the right hand side is in a certain sense the inverse of the original matrix. That is, it gives the solution of the given equation interpreted as an equation (mod p), but it is not possible to determine the true value of the inverse matrix having elements in fractional form. It should be noted, however, that all elements of the true inverted matrix may be represented as fractions whose common denominator is the determinant of the given matrix, if all the elements of the initial matrix are integers. From this, it is clear that all elements of the right hand matrix can be transformed to integers by multiplying by Δ.

The value of the determinant Δ can be obtained from the processes of elimination by forming the cumulative product of the pivotal elements used in each step of elimination; that is,

$$\Delta = a_{11}a_{22}' \cdots a_{nn}^{(n-1)}, \tag{3·3}$$

where $a_{11}, a_{22}', a_{33}'' \cdots a_{nn}^{(n-1)}$ are successive pivotal elements. This may be seen from the fact that the only operation in each step of the elimi-

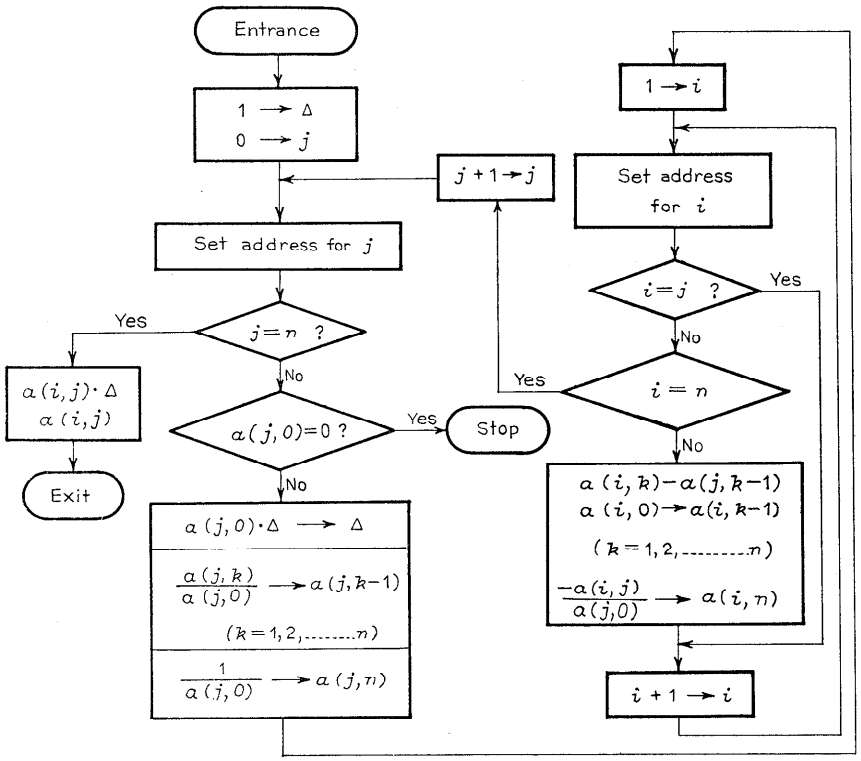

FIG. 2. Flow chart for the matrix inversion routine.

nation process that changes the value of the left hand determinant is division by the pivotal element. Thus, the determinant of the left hand side is repeatedly divided by the pivotal elements in every cycle, until it becomes equal to unity at the end of the last cycle, so that we have

$$\Delta/(a_{11}a_{22}' \cdots a_{nn}^{(n-1)})=1 . \tag{3·4}$$

A flow chart for matrix inversion using the scheme given above is shown in Fig. 2.

A difficulty will occur if any of the pivotal elements becomes zero accidentally. In a case such as this, a search is made for a non-zero element in the remaining part of the left hand matrix, and the elimination is continued; the first non-zero element to be emcountered is then used as a pivot. In this case, however, some attention should be paid to the sign of the result. If all elements of the remaining part of the left hand matrix vanish, the matrix is degenerate and no further calculation is necessary.

4. *The New Method Applied to Calculation on Polynomials*

4·1. *Principle*

It is of some interest to expand our ideas and apply them to calculations involving polynomials using the algebra of a polynomial field modulo $f(x)$, where $f(x)$ is some polynomial of x. Actually, the technique to be explained in this section consists of replacing a calculation involving polynomials $A(x), B(x), C(x), \cdots$ by a corresponding calculation involving the numerical values $A(\alpha), B(\alpha), C(\alpha), \cdots$. Since $A(\alpha)$ is equal to the residue of $A(\alpha)$ modulo $(x-\alpha)$, it is clear that we are dealing with the same technique that has been explained in the preceding paragraphs.

Suppose we are given several polynomials $A(x), B(x), C(x), \cdots$ and have to perform some sequence of algebraic operations which consists of just addition and multiplication of these data polynomials. Let the result be $P(x)$. It is evident that for any α (supposed to be real for practical reasons), $P(\alpha)$ can be obtained by making the same sequence of operations on the numerical (sample) values $A(\alpha), B(\alpha), C(\alpha), \cdots$. By repeating same sequence of operations on n different values of α, we can determine the explicit form of the polynomial uniquely using Lagrange's interpolation formula, on the assumption that $P(x)$ is a polynomial of degree not greater than $n-1$. Since an upper bound for the degree of $P(x)$ can usually be found by some consideration of the actual process for obtaining $P(x)$, it is always, in principle, possible to determine $P(x)$ unambigously by using sufficient number of sample values α for x.

The advantage of this method is evident and is quite similar to that

of the foregoing case. Specifically, it is simpler to program numerical operations than to program manipulation of algebraic expressions, and less working storage in needed. The last step in this method, which is the determination of the polynomial from the sample values, may be handled by using a standard program available for this purpose, so the time and effort required for programming individual problems will be reduced.

Calculations concerning alternating current networks are most effectively handled by this method and provide a good example for purposes of explanation. Network calculations usually involve evaluation of determinants having the elements of the form

$$L_{ik}p^2+R_{ik}p+D_{ik}$$

Although the final results required are usually numerical values, they are in general complex quantities and should be calculated for complex values of p, and hence require arithmetic operations on the complex numbers. For this reason, it is more convenient to obtain first the explicit expression for the determinants as functions of p, and then to evaluate them for complex values of p. Since the calculation of determinants involving polynomials is an extremely complex process, our method offers a powerful tool for treating such problems. The evaluation of polynomials in complex variables is a fairly standard process and may be performed by using standard subroutines. Thus, the whole problem can be dealt with simply by the use of more or less standardized routines.

Limitations of this method lie naturally in the loss of accuracy caused by the use of the interpolation formula, and its accuracy also depends essentially on the proper choice of the sample points α_i. We will not enter the details of the error analysis because this topic seems to deviate somewhat from the subject of the present paper.

If we introduce the further restriction that the coefficients of the given polynomials shall be integers, and try to make an exact calculation with these polynomials, the arithmetic modulo p again becomes applicable to the calculation of numerical values. The advantage derived from exact calculation is that we need not worry about loss of accuracy due to interpolation irrespective of our choice of the value of α. A simple choice would thus be to take $0, 1, 2, \cdots, n-1$ as α's, and to calculate $P(0)$, $P(1), \cdots, P(n-1)$ in order to determine $P(x)$.

4.2. *A Note on the Determination of the Polynomial*

It is not convenient to use the conventional Lagrange interpolation formula for the determination of the polynomial $P(x)$, from the values

$P(\alpha_1), P(\alpha_2), \cdots, P(\alpha_n)$. Instead it is much simpler to solve directly the set of equations,

$$\sum_i a_i \alpha_j^i = P(\alpha_j) \tag{4.2·1}$$

either by ordinary methods or by the method described in the foregoing sections.

Another method which would be useful in any interpolation problem consists of finding a sequence of polynomials $P_0(x), P_1(x), \cdots, P_{n-1}(x)$, according to the following rule. First let $P_0(x)$ be a polynomial of 0th degree, that is a constant, that coincides with $P(x)$ at $x=\alpha_1$. Next let $P_1(x)$ be a polynomial of degree 1 which coincides with $P(x)$ at $x=\alpha_1$ and $x=\alpha_2$, and so on. This rules can be expressed as

$$\left.\begin{aligned} &P_0=P(\alpha_1)\\ &P_1(x)=P_0+(P(\alpha_2)-P_0)\frac{x-\alpha_1}{\alpha_2-\alpha_1}\\ &\cdots\cdots\\ &P_i(x)=P_{i-1}(x)+(P(\alpha_{i+1})-P_{i-1}(\alpha_{i+1}))\frac{(x-\alpha_1)(x-\alpha_2)\cdots(x-\alpha_i)}{(\alpha_{i+1}-\alpha_1)(\alpha_{i+1}-\alpha_2)\cdots(\alpha_{i+1}-\alpha_i)}\,. \end{aligned}\right\} \tag{4.2·2}$$

The process for calculation of the polynomials $P_1, P_2, P_3, \cdots$, is simple to program. This process is, of course, still applicable if the coefficients are integers, and in this case, we can perform the exact calculation using the modulo p technique. It is then convenient to use $0, 1, \cdots, n-1$ as sample points, and (4.2·2) takes the form:

$$\left.\begin{aligned} &P_0=P(0)\\ &P_1(x)=P_0+(P(1)-P_0)\frac{x}{1}\\ &P_2(x)=P_1(x)+(P(2)-P_1(2))\frac{x(x-1)}{2\cdot 1}\\ &\cdots\cdots\\ &P_i(x)=P_{i-1}(x)+(P(i)-P_{i-1}(i))\frac{x(x-1)\cdots(x-i+1)}{i(i-1)\cdots 2\cdot 1}\\ &\quad=P_{i-1}(x)+(P(i)-P_{i-1}(i))\frac{x(x-1)\cdots(x-i-1)}{i!} \end{aligned}\right\} \tag{4.2·3}$$

The division by $i!$, which appears in the second term of the right hand side of (4.2·3) as the denominator, can be replaced by multiplication by its reciprocal modulo p. Such a substitution is permissible, just because the coefficient in the numerator is known to be a multiple of $i!$.

4.3. *"Harmonic Analysis Modulo p"*

It has already been stated that any set of distinct integers can be used as sample values in the mod p case. It is natural to expect that a suitable choice of α can simplify the subsequent calculation required for obtaining the polynomial coeffecients. An example of such a set of values is given by

$$\alpha_i \equiv \varepsilon^i \qquad (\mathrm{mod}\ p) \tag{4.3·1}$$

where ε is any one of the primitive roots of the equation

$$\varepsilon^n \equiv 1 \qquad (\mathrm{mod}\ p) \tag{4.3·2}$$

Equation (4.2·1) now has the form

$$P(\varepsilon^j) = \sum_i a_i \varepsilon^{ij} \tag{4.3·3}$$

A glance at (4.3·3) shows that $P(\varepsilon^i)$ can be regarded as the Fourier transform of the sequence a_i in 'modulo p' sense. Therefore a_i can be obtained by the inverse transformation

$$a_i \equiv n^{-1} \sum_j \varepsilon^{-ij} P(\varepsilon^j) \qquad (\mathrm{mod}\ p)\,. \tag{4.3·4}$$

This may be regarded as a kind of harmonic analysis. Here, n should of course be larger than the degree of the polynomial.

It has been assumed that the equation (4.3·2) has n distinct roots. The necessary and sufficient condition for the equation (4.3·2) to have n distinct roots, that is, the condition that x^n-1 can be completely factorized into linear factors modulo p, is that n divides $p-1$. This being the case ε is given as

$$\varepsilon \equiv \mu^{(p-1)/n} \qquad (\mathrm{mod}\ p) \tag{4.3·5}$$

where μ is any one primitive root of $\mu^{p-1} \equiv 1$ modulo p. While we have several different choices for ε, these all lead to the same result.

4.4. *Calculation on Complex Quantities*

The case $n=4$ of the harmonic analysis just explained gives a method for performing exact calculations on complex quantities without applying the rules for arithmetics with complex numbers. Let us take an integer ε such that

$$\varepsilon^2 \equiv -1 \qquad (\mathrm{mod}\ p) \tag{4.4·1}$$

The equation has a root if and only if $p \equiv 1 \pmod 4$. Now we substitute this real quantity for the imaginary factor $\sqrt{-1}$ in all given complex quantities, thus eliminating imaginary quantities. We then go through the calculation using mod p arithmetic, and let the result be denoted as

$P(\varepsilon)$. The same calculation is repeated using $-\varepsilon$ instead of ε, and the result is $P(-\varepsilon)$. Now the true result is given by $A+\sqrt{-1}B$, where

$$\left.\begin{aligned} A&=2^{-1}(P(\varepsilon)+P(-\varepsilon))\\ B&=-2^{-1}\varepsilon(P(\varepsilon)+P(-\varepsilon)) \end{aligned}\right\} \qquad (4.4\cdot 2)$$

5. *Examples*

Example 1. Calculate the determinant

$$\Delta=\begin{vmatrix} x-11 & -41 & 26 & 19 & -72 & 93 & 32 & 18 & 70\\ -89 & x-11 & -4 & 88 & -33 & -87 & -53 & -89 & -67\\ 88 & 49 & x+45 & -71 & -63 & -82 & -85 & -71 & 6\\ 55 & -93 & -55 & x+92 & 5 & 86 & 76 & -84 & 12\\ -21 & 84 & -38 & -18 & x+88 & 44 & -33 & 51 & -96\\ 34 & -20 & -95 & 4 & -51 & x+34 & -92 & 2 & -71\\ 17 & -40 & 44 & -87 & -31 & 37 & x-89 & 69 & -21\\ 47 & -25 & 85 & 25 & -25 & -51 & -30 & x+11 & 31\\ -68 & -81 & 7 & 63 & 67 & -70 & 25 & 43 & x-5 \end{vmatrix} \qquad (5\cdot 1)$$

As Δ is a polynomial of ninth degree, n should be equal to or larger than 10. Let n be 11. We have to solve

$$X^{11}\equiv 1 \pmod{p} \qquad (5\cdot 2)$$

which has a root when $p-1$ is a multiple of 11. The numbers $p=2^{35}-31$, and $2^{35}-141$ are primes that safisfy this condition. The upper limit of Δ can be covered by these two values of p. Solving the equation (5.2) we have

$$\begin{aligned} x_1&= 22840\ 82430 \pmod{(2^{35}-31)}\\ x_1&=120625\ 84333 \pmod{(2^{35}-141)} \end{aligned}$$

(Other roots are given by $x_r=x_1^r$ $(r=2,\cdots,10)$) Calculation of Δ using these values of x gives

	$p=2^{35}-31$		$p=2^{35}-141$	
$\Delta(x_1)=$	7852	04101	83663	43965
$\Delta(x_2)=$	94047	08928	120461	26152
$\Delta(x_3)=$	163778	32400	155625	87599
$\Delta(x_4)=$	232231	21473	47743	79183
$\Delta(x_5)=$	38547	22821	292059	28729
$\Delta(x_6)=$	249175	88930	141951	05338
$\Delta(x_7)=$	127625	87777	156705	32757
$\Delta(x_8)=$	286527	06171	340947	89793
$\Delta(x_9)=$	180270	24938	62970	97808

$$\Delta(x_{10})=313216\ \ 18182 \qquad 68896\ \ 74223$$
$$\Delta(x_{11})=187205\ \ 80564 \qquad 236994\ \ 63484$$

Applying a Fourier transformation to these values, the coefficients of X^m are obtained as follows;

	$p=2^{35}-31$		$p=2^{35}-141$	
x^0	233424	70269	280219	07449
x^1	296293	71739	299269	20529
x^2	121160	02332	121178	70352
x^3	3523	02825	3523	31645
x^4	223492	80914	223492	80804
x^5	340107	10490	34107	10380
x^6	343593	93753	343593	93643
x^7		1435		1435
x^8		154		154
x^9		1		1
x^{10}		0		0

From these two sets of values, the true from of Δ is obtained as

$$\Delta(x)=x^9+154x^8+1435x^7-344584x^6-349027847x^5$$
$$-12010457423x^4+9002603747119x^3+583509192441266x^2$$
$$+92427438735032x+1461674905790008195$$

Example 2. Inversion of a matrix

$$(a)=\begin{bmatrix} 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \\ 1 & 4 & 16 & 64 \\ 1 & 5 & 25 & 125 \end{bmatrix}.$$

Every step of the process of inversion mod $(2^{36}-31)$ is shown below in octal form for purposes of elucidation.

Initial equations
$$\begin{cases} 1x_1+2x_2+\ 4x_3+\ 10x_4=y_1 \\ 1x_1+3x_2+11x_3+\ 33x_4=y_2 \\ 1x_1+4x_2+20x_3+100x_4=y_2 \\ 1x_1+5x_2+31x_3+175x_4=y_4 \end{cases}$$

First step
$$\begin{cases} x_1+2x_2+\ 4x_3+\ 10x_4= & 1y_1 \\ \quad 1x_1+\ 4x_3+\ 23x_4=377777 & 77740y_1+y_2 \\ \quad 2x_2+14x_3+\ 70x_4=377777 & 777740y_1+y_3 \\ \quad 3x_2+25x_3+165x_4=377777 & 777740y_1+y_4 \end{cases}$$

Second step

$$\begin{cases} x_1+ & 5x_3+ & 23x_4=377777\ 777740y_1+ & 1y \\ x_2+377777\ 777733x_3+377777\ 777703x_4= & & 3y_1+377777\ 777737y_2 \\ & 2x_3+ & 22x_4= & 1y_1+377777\ 777737y_2+y_3 \\ & 6x_3+ & 74x_4= & 2y_1+377777\ 777736y_3+y_4 \end{cases}$$

Third step

$$\begin{cases} x_1+ & 11x_4=177777\ 777761y_1+377777\ 777740y_2+177777\ 777761y_3 \\ x_2+ & 30x_4= & 6y_1+377777\ 777731y_2+ & 3y_3 \\ x_3+377777\ 777707x_4= & 177777\ 777755y_1+ & 6y_2+177777\ 777756y_3 \\ & 6x_4=377777\ 777740y_1+ & 3y_2+377777\ 777736y_3+y_4 \end{cases}$$

Fourth step (Result)

$$\begin{cases} x_1=052525\ 252520y_1+177777\ 777761y_2+177777\ 777760y_3+325252\ 525221y_4 \\ x_2= & 21y_1+377777\ 777715y_2+ & 17y_3+377777\ 777735y_4 \\ x_3=325252\ 525211y_1+ & 23y_2+177777\ 777741y_3+252525\ 242405y_4 \\ x_4= & 2y_1+177777\ 777753y_2+ & 5y_3+177777\ 777757y_4 \end{cases}$$

$$\Delta=1\cdot1\cdot2\cdot6=12$$

$$A_{ik}\equiv(a^{-1})_{ik}\cdot\Delta$$

$$\equiv\begin{bmatrix} 170\ 377777\ 777361 & 264\ 377777\ 777661 \\ 377777\ 777603 & 344\ 377777\ 777447 & 64 \\ 30\ 377777\ 777637 & 74\ 377777\ 777717 \\ 377777\ 777737 & 6\ 377777\ 777733 & 2 \end{bmatrix}$$

Changing the last matrix into decimal form and attaching correct signs, a^{-1} is given as follows:

$$(a^{-1})=\frac{1}{12}\begin{bmatrix} 120 & -240 & 180 & -48 \\ -94 & 228 & -180 & 52 \\ 24 & -66 & 60 & -18 \\ -2 & 6 & -6 & 2 \end{bmatrix}$$