Optimum Time for Multiplication on a Digital Computer

By H. H. Johnson

Methods of multiplication used on serial binary computers are discussed with reference to the optimum time required on a machine with a cyclic main store. From this point of view the methods are shown to be of four main types and the time to be allowed for multiplication in optimum programming is calculated in a number of cases. The possible time-saving of reversing the multiplier and multiplicand is also considered.

1. Introduction

Methods of multiplication on serial binary machines vary from the standard pencil-and-paper method to multiplication by a fast multiplier (Bowden, 1953). The time taken for a multiplication by both these methods is independent of the arrangement of digits in the multiplier. For some methods, however, the time is not invariant, so that for an optimum-coded machine using a delay-type store it is essential to know the expected time for a multiplication, so that maximum time-saving may be obtained, by inserting the next order in the optimum position.

In a machine of this type each order includes an indication of the operation to be performed and the address of the next order; other addresses may be specified also. Wilkinson (1955) gives a detailed account of the optimum coding of a $3 + 1$ address machine with a three level store. For the majority of operations, e.g. addition, subtraction, doubling, halving, some logical orders and magnitude tests, the time taken is known and fixed for the same operation. Thus the next order may be placed, subject to availability, so that it may be called in immediately the previous operation is completed. Further, if the next order is placed in an earlier position, it will not be available until after a complete period of the delay store, and if in a later position, a certain amount of time will be wasted. For operations such as multiplication and division, with variable time for completion, the optimum position for the next order is such that over all possible times the expected time is minimized.

It may be necessary in the following methods to allow time for control instructions, but these are functions of the overall construction of the computer and, as such, do not affect the basic time for a multiplication.

2. Constant Time Methods

2.1 *Pencil-and-Paper Method*

In this method the partial products of the multiplicand and each digit of the multiplier are formed in turn and totalled to give the final product. Very few additional circuits are required, and for any N-digit multiplier the time taken is constant and is that required for N additions. This method was used in the optimum-programmed transistor digital computer built at Manchester University and described by Kilburn, Grimsdale and Webb (1956). The correct sign of the product was produced by a device which caused the last partial product to be subtracted when the multiplier was negative.

2.2 *Use of a Fast Multiplier*

For any N-digit multiplier, N "and" circuits, N-1 adders and N-1 delay circuits are required to form the partial products of each digit of the multiplier with each digit of the multiplicand, which are then totalled in parallel to form the final product. The time taken for a multiplication is constant and is only two word times. This method has been modified in Mercury (Lonsdale and Warburton, 1956) and other computers, with the result that a reduction in the number of circuits by $1/k$ approximately increases the constant time for a multiplication to $2k$ word times.

3. Variable Time Methods

3.1 *The Short-cut Method*

This method, which was used on the A.P.E.(X) computer and the HEC4E computer (Bird 1956), is similar to the short-cut method used on hand calculators. Using this type of multiplication no adjustment for the sign of the product is required (Booth and Booth, 1953). The digits of the multiplier are examined for changes from one to zero, and vice-versa, commencing at the least-significant end. The multiplicand is subtracted from the shifted partial product at the end of a sequence of zeros, and added at the end of a sequence of ones. If the least-significant digit of the multiplier is a one, it is necessary to consider this as the end of a sequence of zeros. This difficulty may be overcome by having a preceding digit preset to zero.

The total time for a multiplication is $(r + s)$ time units, where r is the number of changes of digit and s is the number of sequences of either digit of length greater than one. This assumes that one time unit must be used to count a sequence and to shift the partial product the required number of places, and one time unit must be used to perform the required addition or subtraction at the end of a sequence.

If $P(n, m)$ and $Q(n, m)$ are the probabilities of m changes and sequences in an n-digit binary number with the two least-significant digits alike and unlike respectively, then

$$P(n+1, m+1) = \tfrac{1}{2}P(n, m+1) + \tfrac{1}{2}Q(n, m) \quad (3.1.1)$$

and $$Q(n+1, m+1) = \tfrac{1}{2}P(n, m) + \tfrac{1}{2}Q(n, m) \quad (3.1.2)$$

or $$(2EF - F)P(n, m) = Q(n, m) \quad (3.1.3)$$

and $$(2EF - 1)Q(n, m) = P(n, m) \quad (3.1.4)$$

where

$$EP(n, m) = P(n+1, m) \text{ and } FP(n, m) = P(n, m+1), \text{etc.}$$

Further, if $R(n, m)$ is the probability of m changes and sequences in $(n + 1)$ digits, when the least significant digit is always zero, then

$$R(n, m) = \tfrac{1}{2}P(n, m) + \tfrac{1}{2}P(n, m-1) + \tfrac{1}{2}Q(n, m-1) \quad (3.1.5)$$

or $$2FR(n, m) = (F + 1)P(n, m) + Q(n, m). \quad (3.1.6)$$

Eliminating $Q(n, m)$ between 3.1.3, 3.1.4 and 3.1.6 gives

$$2F(2EF - 1)R(n, m) = [(2EF - 1)(F + 1) + 1]P(n, m) \quad (3.1.7)$$

and $$2FR(n, m) = [F(2E - 1) + F + 1]P(n, m) \quad (3.1.8)$$

Whence

$$(4E^2F^2 - 2EF^2 - 2EF + F - 1)R(n, m) = 0 \quad (3.1.9)$$

and writing $R(n,m) = 2^{-n}X(n, m)$ gives

$$(E^2F^2 - EF^2 - EF + F - 1)X(n, m) = 0 \quad (3.1.10)$$

with boundary conditions

$$X(n, 0) = 0 \quad (3.1.a)$$
$$X(n, 1) = 1 \quad (3.1.b)$$
$$X(n, m) = 0 \quad (m > n). \quad (3.1.c)$$

Values for $X(32, m)$ are given in Table 1.

Table 1

Probability of a Multiplication Time of m Time Units (times 2^{32})

m	$X(32, m) = Z(32, m)$	m	$X(32, m) = Z(32, m)$
0	0	17	46266655
1	1	18	80159046
2	2	19	130589887
3	33	20	199587733
4	91	21	285303525
5	579	22	379927900
6	1788	23	468904185
7	7183	24	532801020
8	21658	25	552539405
9	67731	26	516942938
10	186494	27	429444381
11	495363	28	309691889
12	1218999	29	187361603
13	2842325	30	89896332
14	6228860	31	30737759
15	12890265	32	5702887
16	25148779		

Therefore $2^{-N}X(N, m)$ is the probability of a multiplication by an N-digit multiplier taking m time units. If k time units are allowed for a multiplication in optimum programming, then the expected time $T\ (M,N.k)$ is given by

$$T(M, N, k) = k \sum_{m=1}^{k} 2^{-N}X(N, m) + (M + k) \sum_{m=k+1}^{N} 2^{-N}X(N, m) \quad (N < M + k + 1)$$

$$= (M + k) - M \sum_{m=1}^{k} 2^{-N}X(N, m) \quad (3.1.11)$$

where M time units is the period of the main store. If the multiplication order is in address $p(i)$, the address of the next instruction, as specified in the multiplication order, will be $p(i + k)$. Thus the next order will be available in k time units, if $m \leqslant k$, or in $k + M$ time units, if $k < m \leqslant k + M$. $T(M, N, k)$ has a minimum value $T(M, N, K)$ if, for k $= K$,

$$\Delta T(M, N, k) > 0 \text{ and } \nabla T(M, N, k) < 0 \quad (3.1.d)$$

both differences being with respect to k.

Now

$$\Delta T(M, N, k) = (M + k + 1) - (M + k) - M \sum_{m=1}^{k+1} 2^{-N}X(N, m) + M \sum_{m=1}^{k} 2^{-N}X(N, m)$$
$$= 1 - M2^{-N}X(N, k + 1).$$

The conditions $(3.1.d)$ therefore become

$$X(N, k + 1) < 2^N/M \text{ and } X(N, k) > 2^N/M. \quad (3.1.e)$$

Thus the optimum time, K time units, to allow for a multiplication is such that

$$X(N, K) > 2^N/M > X(N, K + 1) \ (N < M + K + 1).$$

3.11 *Cyclic Store with a Small Period*

If M is small, then $T(M, N, k)$

$$= k \sum_{m=1}^{k} 2^{-N}X(N, m) + (k + M) \sum_{m=k+1}^{M+k} 2^{-N}X(N, m) + (k + 2M) \sum_{m=M+k+1}^{N} 2^{-N}X(N, m)$$
$$(M + k + 1 \leqslant N < 2M + k + 1)$$

$$= (k + 2M) - M \sum_{m=k+1}^{M+k} 2^{-N}X(N, m) - 2M \sum_{m=1}^{k} 2^{-N}X(N, m).$$

Hence

$$\Delta T(M, N, k) = 1 - 2^{-N}M[X(N, M + k + 1) + X(N, k + 1)]$$

and the optimum time K is given by

$$X(N, M + K) + X(N, K) > 2^N/M > X(N, M + K + 1) + X(N, K + 1)$$

$$(M + K + 1 \leqslant N < 2M + K + 1).$$

In general the optimum time K is given by

$$\sum_{i=0}^{\alpha} X(N, K + iM) > 2^N/M > \sum_{i=0}^{\alpha} X(N, K + 1 + iM)$$

$$(\alpha M + K + 1 \leqslant N < (\alpha + 1)M + K + 1)$$

3.2 *A Modification*

If it is possible to avoid using a complete time unit to count a sequence of similar digits and to shift the partial product the required number of places, then the time for a multiplication is $r + 1$ time units, where r is the number of changes of digit in the multiplier. The additional time unit is required to shift and count the multiplier once.

If $S(n, m)$ is the probability of m changes of digit in an $(n + 1)$-digit number, then

$$S(n + 1, m) = \tfrac{1}{2}S(n, m) + \tfrac{1}{2}S(n, m - 1) \qquad (3.2.1)$$

or

$$(2EF - F - 1)S(n, m) = 0. \qquad (3.2.2)$$

Writing $S(n, m) = 2^{-n}W(n, m)$ gives

$$(EF - F - 1)W(n, m) = 0 \qquad (3.2.3)$$

with boundary conditions

$$W(n, 0) = 2 \qquad (3.2.a)$$

$$W(n, n) = 2 \qquad (3.2.b)$$

$$W(n, m) = 0 \quad (m > n). \qquad (3.2.c)$$

Equation 3.2.3 may be written $(\Delta F - 1)W(n, m) = 0$.

Therefore $(F - \Delta^{-1})W(n, m) = A$ (a constant) which gives

$$W(n, m) = \Delta^{-m}A = \Delta^{-m+1}nA = \Delta^{-m+2}n(n + 1)A/2!$$

$$= An(n - 1) \ldots (n - m + 1)/m! = An!/[(n - m)!m!]$$

whence $\quad W(n, m) = 2n!/[(n - m)!\, m!]$.

Thus the probability of m changes of digit in an $(N + 1)$-digit number, whose last digit is zero, is $2^{-N}Y(N, m) = 2^{-N}N!/[(N - m)!m!]$.

Values of $Y(32, m)$ are given in Table 2.

By the method of 3.11, the optimum time $K + 1$ to allow for a multiplication is given by

$$\sum_{i=0}^{\alpha} Y(N, K + iM) > 2^N/M > \sum_{i=0}^{\alpha} Y(N, K + 1 + iM)$$

$$(\alpha M + K + 1 \leqslant N < (\alpha + 1)M + K + 1).$$

Table 2

m	$Y(32, m)$	m
0	1	32
1	32	31
2	496	30
3	4960	29
4	35960	28
5	201376	27
6	906192	26
7	3365856	25
8	10518300	24
9	28048800	23
10	64512240	22
11	129024480	21
12	225792840	20
13	347373600	19
14	471435600	18
15	565722720	17
16	601080390	16

3.3 *Modification of the Pencil-and-Paper Method*

It is interesting to compare the method of the previous section with a modified version of the standard pencil-and-paper method. If the multiplier may be scanned for the next most significant unit digit, the partial product shifted and the multiplicand added to the partial product all in one time unit, then the total time for a multiplication is $(m + 1)$ time units, where m is the number of unit digits in the multiplier and one time unit is allowed for the total scanning. Now the probability of m unit digits in an N-digit binary number is $\binom{N}{m}2^{-N}$, so that the optimum time to allow for a multiplication is identical with the result of 3.2.

3.4 *The Halving and Doubling Method*

In the binary scale this method becomes identical with the method of 2.1 (Booth and Booth, 1953). When the binary multiplier is halved, the remainder is one, if and only if the least-significant digit before halving was one; in this case the multiplicand is added to the partial product. If the remainder is zero, then the least-significant digit must have been a zero and the multiplicand is not added to the partial product. In either case the multiplicand is doubled and the process repeated. If a sequence of zeros may be detected, the time for a multiplication, by either method, may generally be reduced, depending on the structure of the multiplier, since when the remainder is zero no addition is required. If r is the number of ones and s the number of isolated zeros and sequences of zeros in the multiplier, then the time for a multiplication is $(r + s)$ time units.

If $L(n, m)$ and $M(n, m)$ are the probabilities of m ones, isolated zeros and sequences of zeros in n-digit numbers ending in zero and one, respectively, then

$$L(n, m) = \tfrac{1}{2}L(n - 1, m) + \tfrac{1}{2}M(n - 1, m - 1) \quad (3.4.1)$$

and

$$M(n, m) = \tfrac{1}{2} L(n-1, m-1) + \tfrac{1}{2} M(n-1, m-1) \quad (3.4.2)$$

or

$$(2EF - F)\, L(n, m) = M(n, m) \quad (3.4.3)$$

and

$$(2EF - 1) M(n, m) = L(n, m). \quad (3.4.4)$$

Further, if $N(n, m)$ is the probability of m ones, isolated zeros and sequences of zeros in any n-digit number,

$$N(n, m) = \tfrac{1}{2} L(n, m) + \tfrac{1}{2} M(n, m). \quad (3.4.5)$$

Eliminating $L(n, m)$ and $M(n, m)$ between 3.4.3, 3.4.4 and 3.4.5 gives

$$(4E^2F^2 - 2EF^2 - 2EF + F - 1)\; N(n, m) = 0. \quad (3.4.6)$$

Substituting $N(n, m) = 2^{-n} Z(n, m)$ gives

$$(E^2F^2 - EF^2 - EF + F - 1) Z(n, m) \quad (3.4.7)$$

with boundary conditions

$$Z(n, 0) = 0 \quad (3.4.a)$$

$$Z(n, 1) = 1 \quad (3.4.b)$$

$$Z(n, m) = 0 \quad (m > n). \quad (3.4.c)$$

By comparison with 3.1.10 and associated boundary conditions 3.1.*a*, 3.1.*b* and 3.1.*c*, we see that $Z(n, m) \equiv X(n, m)$; thus the optimum time for a multiplication, K, is the same as that of the short-cut method given in 3.1 and 3.11.

3.5 *The Modified Short-Cut Method*

This process is described by Lehman (1958), who gives an earlier verbal reference. Basically a pair of consecutive digits, $d(i)$ and $d(i+1)$, where $d(i+1)$ is more significant than $d(i)$, of the multiplier are examined. If these digits are equal, no alteration is made to the partial product, and the pair of digits $d(i+1)$ and $d(i+2)$ are examined. If the digits examined are unequal, the multiplicand is subtracted from or added to the partial product if $d(i)$ is greater or less than $d(i+1)$, respectively, and the digits $d(i+2)$ and $d(i+3)$ are examined. Lehman shows that no adjustment for the sign of the product is required. Tocher (1958) proves that this modified ternary representation of the binary number

$$\sum_{i=0}^{\infty} d(i) 2^i \text{ in the form } \sum_{i=0}^{\infty} (-)^{s(i)} e(i) 2^i,$$

where $d(i), e(i), s(i) = 0, 1.$ $i = 0, 1, 2, \ldots$, gives the minimum digit representation, i.e. when

$$e(i+1) = \{d(i+1) - d(i)\}^2 \{1 - e(i)\}$$

and $s(i) = d(i+1)$. Two cases will be considered.

3.51 *First Modification* (*Third Method*)

The total time for a multiplication is $(r + s)$ time units where r is the number of pairs of unlike digits causing an addition or subtraction, and s is the number of sequences of like digits of length greater or equal to one, not included in the pairs of digits.

Table 3

m	$U(32, m)$
0	0
1	1
2	3
3	68
4	300
5	2496
6	11900
7	63424
8	274464
9	1131616
10	4140752
11	13827648
12	41363712
13	110319616
14	257856768
15	516163584
16	851094528
17	1080331776
18	921956608
19	417364992
20	75254784
21	3784704
22	23552

Then, if $A(n, m)$ is the probability of m pairs and sequences in $(n + 1)$ digits, where the least-significant digit is always zero,

$$(4E^3F^2 - 2E^2F^2 - 2EF + F - 1) A(n, m) = 0 \quad (3.51.1)$$

Writing $A(n, m) = 2^{-n} U(n, m)$ gives

$$U(n+3, m+2) - U(n+2, m+2) - 2U(n+1, m+1) + 2U(n, m+1) - 2U(n, m) = 0 \quad (3.51.2)$$

with boundary conditions

$$U(n, 0) = 0 \quad (3.51.a)$$

$$U(n, 1) = 1 \quad (3.51.b)$$

$$U(n, m) = 0 \; \{m > [(2n+3)/3]\} \quad (3.51.c)$$

Values for $U(32, m)$ are given in Table 3.

Hence, by the method of 3.11, the optimum time, K, to allow for a multiplication is given by

$$\sum_{i=0}^{\infty} U(N, K + iM) > 2^N/M > \sum_{i=0}^{\infty} U(N, K + 1 + iM)$$

$$(\alpha M + K + 1 \leqslant N < (\alpha + 1) M + K + 1).$$

3.52 *Second Modification.* (*Fourth Method*)

If, as in 3.2, a complete time unit is not necessary to count a sequence of similar digits and to shift the partial product, then the time for a multiplication is $r + 1$ time units, where r is the number of pairs of unlike digits causing an addition or subtraction.

If $I(n, m)$ and $J(n, m)$ are the probabilities of m unlike

Table 4

m	$V(32, m)$
0	1
1	63
2	1800
3	30856
4	353808
5	2864160
6	16839680
7	72864000
8	232581888
9	543921664
10	916844544
11	1083543552
12	859955200
13	428654592
14	120324096
15	15597568
16	589824

pairs in an $(n+1)$-digit binary number with, respectively, a sequence or a pair at the most significant end, then

$$I(n+1, m+1) = \tfrac{1}{2}I(n, m+1) + J(n, m+1) \quad (3.52.1)$$

and $$J(n+1, m+1) = \tfrac{1}{2}I(n, m) \quad (3.52.2)$$

or $$(2E-1)I(n, m) = 2J(n, m) \quad (3.52.3)$$

and $$2EFJ(n, m) = I(n, m). \quad (3.52.4)$$

Further, if $B(n, m)$ is the probability of m pairs in $(n+1)$ digits when the least-significant digit is always zero, then

$$B(n, m) = I(n, m) + J(n, m). \quad (3.52.5)$$

Therefore $$(2E^2F - EF - 1)B(n, m) = 0. \quad (3.52.6)$$

Writing $B(n, m) = 2^{-n}V(n, m)$ gives

$$V(n+2, m+1) - V(n+1, m+1) - 2V(n, m) = 0 \quad (3.52.7)$$

with boundary conditions

$$V(n, 0) = 1 \quad (3.52.a)$$

$$V(2n+1, n) = 2^{n+1} \quad (3.52.b)$$

$$V(n, m) = 0\{m > [(n+2)/2]\}. \quad (3.52.c)$$

Values for $V(32, m)$ are given in Table 4.

Hence, by the method of 3.11, the optimum time, $K+1$, to allow for a multiplication is given by

$$\sum_{i=0}^{\alpha} V(N, K+iM) > 2^N/M > \sum_{i=0}^{\alpha} V(N, K+1+iM)$$
$$(\alpha M + K + 1 \leqslant N < (\alpha+1)M + K + 1).$$

4. Selection of Multiplier

It has been shown that in some methods of multiplication the time taken depends solely on the arrangement of digits in the multiplier. It would, however, be feasible to use the multiplicand as the multiplier, if the multiplication would thereby be completed more quickly. This would involve a small number of additional circuits, and a certain number of time units would have to be allowed for comparing the two numbers.

If this choice is possible and P time units are needed to select the best number as multiplier; then if $2^{-n}C(n, m)$ is the probability of a multiplication by an n-digit number taking m time units, the probability of a multiplication of two n-digit numbers taking m time units is

$$2^{-2n}D(n, m) = 2^{-2n}\left[C(n, m)\left(\sum_{i=m}^{n} C(n, i)\right) + C(n, m)\left(\sum_{i=m+1}^{n} C(n, i)\right)\right].$$

Hence, following 3.11, the optimum time to allow for a multiplication of two N-digit binary numbers is $K+P$ or $K+P+1$, according to whether additional time is required to shift the partial product, where K is such that

$$\sum_{i=0}^{\alpha} D(N, K+iM) > 2^{2N}/M > \sum_{i=0}^{\alpha} D(N, K+1+iM)$$
$$(\alpha M + K + 1 \leqslant N < (\alpha+1)M + K + 1)$$

Table 5

Optimum time for the Multiplication of two 32-digit Binary Numbers

	MULTIPLIER UNSELECTED			MULTIPLIER SELECTED	
	FIRST METHOD (3.1 3.4)	SECOND METHOD (3.2 3.3)		FIRST METHOD (3.1 3.4)	SECOND METHOD (3.2 3.3)
M	K	K	M	K	K
8	3	$1+1$	8	$1+P$	$1+P+1$
16	12	$3+1$	16	$10+P$	$2+P+1$
32	29	$20+1$	32	$28+P$	$19+P+1$
64	30	$21+1$	64	$29+P$	$20+P+1$
128	30	$22+1$	128	$29+P$	$21+P+1$
	THIRD METHOD (3.51)	FOURTH METHOD (3.52)		THIRD METHOD (3.51)	FOURTH METHOD (3.52)
M	K	K	M	K	K
8	2	$4+1$	8	$1+P$	$4+P+1$
16	3	$13+1$	16	$2+P$	$13+P+1$
32	19	$13+1$	32	$18+P$	$13+P+1$
64	20	$14+1$	64	$18+P$	$13+P+1$
128	20	$14+1$	128	$19+P$	$13+P+1$

5. Discussion

Variable time methods of multiplication have been seen to be of four types. By allowing the optimum time for a multiplication, instead of N time units, where N is the number of digits in each number, the percentage time savings on the average, for $N = 32$, are $9 \cdot 4$, $34 \cdot 4$, $40 \cdot 6$* and $56 \cdot 2$ by the first, second, third and fourth types, respectively. The last figure represents a very remarkable saving in machine time, the middle two are well worth consideration, whilst even the first saving would be useful, particularly in calculations involving a predominance of multiplication time.

* Tocher (*loc. cit.*) shows for large N a direct saving of digits of 33% of digits and a corresponding direct decrease in multiplication time; the further decrease is due to the optimal strategy employed in positioning the next instruction.

The prior selection of the "best" multiplier is shown to be of no benefit, except possibly in three of the cases considered, and only then if the choice can be made in one time unit.

6. Acknowledgements

The author wishes to thank Dr. A. D. Booth of Birkbeck College for his advice and encouragement, and Dr. J. G. Freeman of Bradford Institute of Technology for his continued support.

References

BOWDEN, B. V. (Ed.) (1953). *Faster Than Thought*. London: Pitman & Sons Ltd., p. 56.

LONSDALE, K., and WARBURTON, E. T. (1956). "Mercury: A High Speed Digital Computer," *Proc. I.E.E.*, Vol. 103, Part B, p. 2093 M.

BIRD, R. (1956). "The HEC4E Computer," *Proc. I.E.E.*, Vol. 103, Part B, p. 2052 M.

BOOTH, A. D., and BOOTH, KATHLEEN H. V. (1953). *Automatic Digital Calculators*. London: Butterworths, pp. 44–6 and 56–8.

KILBURN, T., GRIMSDALE, R. L., and WEBB, D. C. (1956). "A Transistor Digital Computer with a Magnetic-Drum Store," *Proc. I.E.E.*, Vol. 103, Part B, p. 2043 M.

LEHMAN, M. (1958). "Short-Cut Multiplication and Division in Automatic Binary Digital Computers," *Proc. I.E.E.*, Vol. 105, Part B, p. 2693 M.

TOCHER, K. D. (1958). "Techniques of Multiplication and Division for Automatic Binary Computers," *Quar. Jour. Mech. and Applied Maths.*, Vol. XI, p. 3.

WILKINSON, J. H. (1955). "An Assessment of the System of Optimum Coding used on the Pilot Automatic Computing Engine at the National Physical Laboratory," *Phil. Trans. Roy. Soc.*, A.946, Vol. 148, pp. 253–81.

Forthcoming Publication of the Proceedings of the 1960 PICC Symposium, Rome

A *Symposium on the numerical treatment of ordinary differential equations, integral and integro-differential equations* took place during the week of 20–24 September 1960 at the Mathematical Institute of the University of Rome. This Symposium was organized by the Provisional International Computation Centre (PICC).

The Symposium opened with a report delivered by Professor Walther of Darmstadt (Germany) on the methods presently employed in the treatment of integral and integro-differential equations. The different methods are classified in categories according to the nature of the problem, the type of solution desired, and the numerical and/or electronic techniques available. Dr. Genuys (Paris) then presented a second report, also very complete, on the methods of treating ordinary differential equations. Like Professor Walther, Dr. Genuys examined each method in relation to the practical possibilities of processing by modern electronic computers.

After this introduction, more than 50 specialists, divided into three study groups (Section I: Ordinary differential equations; Section II: Integral and integro-differential equations; Section III: Applications), spoke on the particular problems with which they dealt and how the practical and theoretical difficulties which they encountered had been overcome.

The Symposium was attended by about 200 mathematicians from the following countries: Austria, Belgium, Czechoslovakia, Finland, France, Germany, Greece, Hungary, Ireland, Israel, Italy, Japan, the Netherlands, Poland, Rumania, Sweden, Switzerland, United Kingdom, United States of America, Yugoslavia.

Of a more general and philosophic nature was the lecture delivered by Professor Lanczos (Dublin) on the possibilities offered by modern electronic computers, closely allied with a penetrating criticism of approximation processes and convergence caprices.

The final session was devoted to an outstanding speech by Professor R. Courant of New York, who explained his personal conclusion on the requirements of scientific research in this highly technical century, and the problems of training, at the highest level, young specialists in the field of automatic computation.

The Symposium, on the whole, presented a fairly complete picture of the actual state of this important section of mathematical sciences. Its success was largely due to the careful preparatory work furnished by the Italian representative to the PICC, Professor Aldo Ghizzetti, Rome.

The *Proceedings* (about 700 pages) will be published by Birkhäuser Verlag (Basel/Stuttgart) early in 1961.