TECHNICAL NOTES AND SHORT PAPERS

Some Remarks on Modular Arithmetic and Parallel Computation

By H. S. Shapiro

1. Introduction. A question that has been discussed in recent years is that of *parallel computation*. Can a given computation be broken up into independent assignments which may be performed simultaneously? Traditional methods of computation are almost entirely *serial*, with the consequence that one cannot convert extra computing capacity into significantly greater speed. Thus far the only general method which has been proposed for achieving parallelism is the use of "modular arithmetic"—that is, for some collection of relatively prime integers $m_1, \cdots m_k$ one performs the calculations (mod m_i) independently; the final result is then obtained by solving a system of simultaneous congruences. Such a procedure is possible provided that (i) the calculation consists entirely of additions and multiplications of integers* (so that the corresponding calculations (mod m_i) are justified), and (ii) each number sought in the calculation is an integer known *a priori* to lie in an interval of length $\leqq m_1 m_2 \cdots m_k$.

Modular arithmetic, when applicable, has the advantage of being free from round-off errors; moreover, addition and multiplication (mod m) are carry-free. Another feature is that in some types of calculation (for instance, tabulation of the values of a polynomial for equally spaced values of the argument) the calculation (mod m) is much simplified by the *periodic repetition* of the values being calculated. It therefore seems of interest to show how computations of practical importance may be carried out within the limitations (i) and (ii) above. In this note we discuss division, linear equations, and the first boundary value problem from the standpoint of modular arithmetic.

2. Division. Let us consider the problem of finding d binary digits of the quotient $\frac{x}{y}$ where x and y are integers, $0 < x < y$. The most natural approach is to choose an $n > 0$, and let r be the least non-negative residue of 2^n (mod y), then, writing $N = \frac{2^n - r}{y}$, N is an integer easily computed (mod m) if $(m, y) = 1$, and we have

$$\frac{xN}{2^n} \leqq \frac{x}{y},$$

these numbers differing by $\frac{xr}{2^n y} < \frac{x}{2^n}$. Hence, the integer xN, converted to binary

Received May 22, 1961. This report represents results obtained at the Institute of Mathematical Sciences, New York University, under the sponsorship of the Office of Naval Research, United States Navy.

* Division is allowable only when the modulus is relatively prime to the denominator, and the quotient is an integer.

notation, gives d digits of the quotient $\frac{x}{y}$, providing $2^{n-1-d} > x$. Here the approximation is from below; by working instead with

$$N' = \frac{2^n + s}{y}, \qquad s = y - r,$$

we get an approximation from above. The objection to this procedure is that calculation of the residues of $N(\text{mod } m_i)$ is simple only in case $(m_i, y) = 1$, and so each denominator gives rise to a certain set of "forbidden" moduli.

The following division algorithm is free from this defect. Let $b = 2^k$ be the smallest power of 2 not less than y. Then the sequence t_n defined by

$$bt_{n+1} = (b - y)t_n + x$$

converges to $\frac{x}{y}$ (for an arbitrary choice of t_0). Writing $s_n = b^n t_n$, we get

$$s_{n+1} = (b - y)s_n + b^n x. \tag{1}$$

If s_0 is chosen to be an integer, all the s_n are integers, easily computed and periodic (mod m), for all m without exception. In place of (1), we can use the convergent iteration

$$s'_{n+1} = -(y - a)s_n' + a^n x \tag{2}$$

where a is the greatest power of 2 not exceeding y. By choosing the better of (1), (2), i.e., (1) or (2) according as $\frac{b-y}{b}$ or $\frac{y-a}{a}$ is smaller (and at least one of these numbers is $<\frac{1}{3}$) we achieve good convergence. The necessary *a priori* estimate of s_n (or s_n') respectively, and the degree of approximation after n iterations, are readily obtained, and from this the magnitude of $M = \Pi m_i$ sufficient for calculation of $\frac{x}{y}$ to the required accuracy is known. Since t_n arises from s_n upon division by 2^{nk}, i.e., shifting of a binary point, the conversion of s_n from modular to binary notation gives the initial digits of $\frac{x}{y}$ directly.

A variation of this division algorithm which gives a simpler recurrence at the expense of an auxiliary calculation is this. Suppose that the (given) binary expansions of x, y are

$$x = a_0 2^p + a_1 2^{p-1} + \cdots a_p$$
$$y = b_0 2^q + b_1 2^{q-1} + \cdots b_q$$

where a_i, b_i are 0 or 1. We may suppose $a_p = b_q = 1$ since multiplication and division by powers of 2 is trivial. Then $\frac{x}{y} = 2^{p-q} f(\frac{1}{2})$, where f denotes the function

$$f(t) = \frac{a_p + a_{p-1}t + \cdots a_0 t^p}{b_q + b_{q-1}t + \cdots b_0 t^q} = 1 + c_1 t + c_2 t^2 + \cdots. \tag{3}$$

It is easy to show that, in the Taylor expansion (3), we have $| c_n | \leqq n$th Fibonacci number, so that (3) converges at least* for $| t | < \frac{\sqrt{5} - 1}{2} = .618 \cdots$. Moreover, from (3), the integers c_n are readily computed in terms of the a_i, b_i by a recurrence obtained upon cross-multiplying in (3).

But in terms of the c_n the division can be carried out, using the scheme

$$s_{n+1} = 2\, s_n + c_n\,. \tag{4}$$

If s_0 is an integer (say, $s_0 = 0$) the s_n are integers and

$$\lim_{n\to\infty} \frac{s_n}{2^n} = f\left(\frac{1}{2}\right) = 2^{q-p}\,\frac{x}{y}$$

once again, the *a priori* bounds on s_n, and the rate of convergence (which is uniformly rapid with respect to all divisions) is readily obtained.

These algorithms may be adapted, in an obvious way, to any radix.

3. Linear Equations. Given the system of k equations (in vector notation)

$$Ax = b, \qquad A = \| a_{ij} \| \tag{1}$$

where the a_{ij} and b_i are assumed to be integers, the direct adaptation for modular arithmetic is to compute $d = \det A$, and replace (1) by the system

$$Ay = db \tag{2}$$

for the *integer* variables y_i. Operating with moduli m_i such that $(d, m_i) = 1$ (assuming, of course, $d \neq 0$) the solution of (2) (mod m_i) is very simple (say, by Gaussian elimination). Crude *a priori* bounds on the y_i may be obtained (e.g., by Hadamard's determinant inequality) when they are not available from physical or other considerations. However, there is again the objection that this scheme allows "forbidden moduli" which vary with the given problem. Moreover, the solutions x_i are found as quotients, necessitating divisions which are non-trivial. These difficulties disappear when an iterative method is employed. Let us suppose, to keep the discussion simple, that by preliminary transformations (1) has been put into a form where $| a_{ii} | > \sum_{j\neq i} | a_{ij} |$. Let c_i be the integer defined by: (i) c_i has the same sign as a_{ii}; (ii) $| c_i |$ is the least power of 2 not less than $| a_{ii} |$. Let C be the diagonal matrix of the c_i, and $c = \max | c_i |$. Then the system (1) may be rewritten in the form

$$Cx = Dx + b \tag{3}$$

(writing $D = C - A$), and the iterative scheme

$$Cx_{n+1} = Dx_n + b \tag{4}$$

converges to the solution of (1), because of the supposed diagonal-dominance of A. Finally, putting $y_n = c^n x_n$ we get

$$y_{n+1} = (cC^{-1})Dy_n + (cC^{-1})bc^n. \tag{5}$$

* The fact that the series (3) converges at least in this circle, i.e., that the polynomial in the denominator cannot vanish in this circle, was discovered in another connection and pointed out to the author by D. J. Newman.

Since cC^{-1} has integer entries, all components of all y_n are integers (if y_0 is so chosen). The iterative scheme (5) is suitable for modular computation; we omit a detailed discussion of rate of convergence and *a priori* bounds.

4. First Boundary-Value Problem. When the equation $\nabla^2 u = 0$, subject to given boundary conditions, is solved numerically by the method of (square) nets, it is customary to use an iterative method of solution which is known to converge at a rate that can be estimated in terms of the geometry of the region. This problem is in principle subsumed in the above discussion, but two factors make it especially simple from the standpoint of modular computation. First, the transformation to integer variables is particularly simple and especially favorable to a solution in binary notation (owing to the special significance of the number 4 in this iteration). Second, the maximum principle gives a good *a priori* bound on the solutions. For, writing the iterative scheme symbolically as

$$u_{n+1} = Au_n + b$$

and setting $4^n u_n = vn$, we get

$$v_{n+1} = 4Av_n + 4^{n+1}b. \tag{1}$$

Suppose that the components of b are integers (i.e., that the given boundary values are integers, which is achieved by shifting a binary point). If, then, the initial values $v_0 = u_0$ are chosen to be integers lying between the smallest and greatest boundary values, all components of the v_n computed from (1) are integers, lying in the range

$$4^n B_1 \leqq v \leqq 4^n B_2$$

where B_1 and B_2 are the min (and max) of the prescribed boundary values.

5. Remarks on Other Iterative Methods. There are many other important iterative methods in numerical analysis, but not all of these seem well adapted to modular computation, because in many cases transformation to integer variables leads to integers that are too large to be computed practically, i.e., an excessively great number of moduli are required. We may illustrate this with a simple example. Suppose we try to solve the equation

$$x^2 + x - \tfrac{1}{8} = 0 \tag{1}$$

by means of the convergent iteration

$$x_{n+1} = -x_n^2 + \tfrac{1}{8}, \qquad x_0 = 0.$$

Letting $2^{2^n} x_n = y_n$ we have

$$y_{n+1} = 2^{2^{n+1}-3} - y_n^2, \qquad y_0 = 0. \tag{2}$$

The numbers y_n defined by (2) are integers, whose initial binary digits coincide with those of the positive root of (1). Moreover, calculation (mod m) of the numbers y_n from (2) is quite trivial. However, since y_n is of the order of 2^{2^n}, even 10 iterations using moduli of the order of 50 would involve us (roughly) in calculating a 1000 binary digit number, by solving a system of 200 simultaneous congruences—a lot of work to solve a quadratic equation. Newton's method would lead to the same difficulties. A feasible method for solving quadratic equations by modular computation can, however, be based upon the Taylor expansion $(1 - 4x)^{-1/2} = \sum_{n=0}^{\infty} \binom{2n}{n} x^n$.

6. Concluding Remarks. Preliminary analysis indicates that parallel computation, using modular arithmetic, is feasible for certain kinds of problems. The parallel computation envisioned here leads very swiftly to a solution encoded "in modular notation." By this is meant, a system of simultaneous congruences, whose solution (in a specified interval), written as a binary number, has as its *initial digits* the binary number which is the goal of the computation. For results of practical value it will probably be necessary, at the very least, to use moduli whose product exceeds 10^{10}. Hence the feasibility of rapid solution of large-scale systems of congruences will determine the timesaving possibilities of the method. Any *a priori* knowledge about the solution, such as might be obtainable from a preliminary rough solution, analog computation, etc., leads to a reduction in the number of necessary moduli, i.e., knowledge of r binary places reduces the product of the m_i needed by a factor 2^{-r}. Again, in such a case as the boundary value problem, where the values of the solution at neighboring net points differ by amounts which can be bounded *a priori*, this fact might lead to a considerable reduction of labor in the "conversion" phase of the problem.

7. Acknowledgment. The author wishes to thank Lt. Col. L. M. Butsch, Jr., Capt. F. M. Brown, and Capt. A. L. Calton, Jr. of the Bionics and Computer Laboratory, Wright Air Development Division, for directing his attention to the area of modular computation, and Dr. D. L. Slotnick of the Westinghouse Electric Corporation, Air Arm Division, for many stimulating discussions on the subject.

Institute of Mathematical Sciences
New York University
New York 3, N. Y.

Permutations with Restricted Position

By Frank Harary

In his book on combinatorial analysis, Riordan [4, p. 163–164] discusses permutations with restricted position and mentions an open question:
"Any restrictions of position may be represented on a square, with the elements to be permuted as column heads and the positions as row heads, by putting a cross at a row-column intersection to mark a restriction. For example, for permutations of four (distinct) elements, the arrays of restrictions for the rencontres and reduced ménage problems mentioned above are

Received July 5, 1961.