80[F].—A. GLODEN, *Nouvelle extension des solutions de la congruence* $x^4 + 1 \equiv 0$ *(mod p) pour p entre* $6 \cdot 10^5$ *et* 10^7. Mss. in possession of the author, 11 rue Jean Jaurès, Luxembourg, and in the Library of Brown University.

This manuscript table gives solutions x of the congruence mentioned in the title for approximately 1300 primes p beyond the limit 600000 set by previous tables [1] and under ten millions. With a few exceptions, only one solution x is given for each p (instead of the usual pair) and in each $x < 40000$. The table is a byproduct of the results of factoring numbers of the form $x^4 + 1$.

D. H. L.

[1] For references to previous tables of this sort see *MTAC* v. 1, p. 6, v. 2, p. 71, 210, v. 3, p. 96.

81[F].—A. GLODEN, *Table de factorisations des nombres* $N^8 + 1$ *pour* $N \leq 400$. Mss. in possession of the author, 11 rue Jean Jaurès, Luxembourg, and in the Library of Brown University.

The table of CUNNINGHAM [1] giving the factors of $N^8 + 1$ for $N \leq 200$ is extended in this manuscript not only by doubling the upper limit of N but also by raising limit 100000 of the smallest prime factor omitted to 600000. Of the 400 entries of the table 140 are complete factorizations, 213 are composite but incompletely factored, while only 47 are of entirely unknown character, beyond the fact that their factors lie above 600000. The smallest number of this latter kind is $\frac{1}{2}(43^8 + 1) = 5844100138801$. The author hopes to raise his 600000 to 800000 by extending his already extensive tables of solutions of the quartic congruence [2] $x^4 + 1 \equiv 0 \pmod{p}$.

A comparison of this table with that of CUNNINGHAM reveals the following errata in the latter.

p. 140	$y = 86$	insert the small factor 61057
	$y = 148$	delete the factor 97
p. 141	$y = 125$	for semicolon read full stop.

D. H. L.

[1] A. J. C. CUNNINGHAM, *Binomial Factorisations*, v. 1, London 1923, p. 140–141.
[2] See *MTAC*, v. 2, p. 71–72, 210–211, 252, 300.

AUTOMATIC COMPUTING MACHINERY

Edited by the Staff of the Machine Development Laboratory of the National Bureau of Standards. Correspondence regarding the Section should be directed to Dr. E. W. CANNON, 418 South Building, National Bureau of Standards, Washington 25, D. C.

TECHNICAL DEVELOPMENTS

Our contribution under this heading, appearing earlier in this issue, is "The solution of simultaneous linear equations with the aid of the 602 calculating punch," by FRANK M. VERZUH.

DISCUSSIONS

A New General Method for Finding Roots of Polynomial Equations

The problem of finding all of the roots of polynomial equations of fairly high degree arises so frequently that a routine for accomplishing this automatically on a high-speed digital computer would be of considerable practical value. However, for every one of the standard methods for finding the roots of a polynomial equation, there are some exceptional cases in which the particular method applied fails to work.

Horner's method and Newton's method require a good initial approximation, to prevent

the resulting sequence of approximations from diverging (as in the example $f(x) = x^3 - 5x$, where $x_0 = 1$) or involving division by zero. The method of false position will fail completely for complex roots and for real roots of even multiplicity. Both Lagrange's method and the method of Sturm functions fail for complex roots. The Graeffe method will furnish the absolute value of each root and the total number of roots having each absolute value, but it will not give the roots themselves if there are several complex roots of equal modulus. Since in the case of real polynomials the complex roots always occur in pairs of equal modulus, further refinements of the method are necessary.[1] The refined methods either fail or become extremely complicated in the case of several pairs of roots of equal modulus. In using the Graeffe method efficiently, it is also necessary to exercise considerable judgment to know which one of the possible modifications to use or to know when to stop the squaring process. Bernoulli's method does not converge if there is more than one root of largest absolute value. A modification of Bernoulli's method [2] will give the absolute value if there are n roots having this absolute value, but a different method is required for each different n. Also, considerable judgment must be exercised to know which method to use or when to change methods.

In the case of a mathematician working with a desk computer, exceptional cases will not cause much trouble, since they occur rarely, and, by the use of intelligent judgment, they can be detected and an appropriate change of method made. But it would be an imposing task (even without considering the limitations necessitated by the proposed memory sizes of all the digital machines now under construction) to set up routines for detecting and handling every one of the possible combinations of exceptional cases that can arise. It would be of advantage to have a single method which will always give all of the roots, regardless of their positions or multiplicities.

The method outlined below is a first attempt at such a universal method. It is admittedly not a speedy or efficient one, since, in fact, it would require a prohibitive amount of time to carry out the method except on high-speed digital calculating machinery, and even on these machines the process would be time consuming. But it has the advantage that regardless of the nature of the original polynomial, this method will always converge to a root, to as much accuracy as the machine uses. Furthermore, the convergence always takes place within a fixed number of steps (independent of the degree of the polynomial equation).

After finding the "first" root, the equation can be reduced in degree by removal of this root and the other roots found by repeating the same routine. Thus, the entire set of roots and their multiplicities can be found by a machine without the necessity of any human intervention during the problem.

The proposed method is as follows: Given any point p_n which is the nth approximation to a root, the following operations produce the $(n+1)^{\text{th}}$ approximation. Expand the polynomial around the point p_n by synthetic division. Then, by performing the Graeffe process a fixed number of times, find (to some known degree of accuracy in terms of relative error) R, the absolute value of the root of the transformed equation which is smallest in modulus. Since R is a function of p_n, it will be denoted by $R(p_n)$, and geometrically it is the distance from the point p_n to the nearest root of the original polynomial.

The number actually obtained by applying the Graeffe root-squaring process a fixed number of times will be denoted by $R^*(p_n)$, where $(1 + d_n)R^*(p_n) = R(p_n)$, and by using enough significant figures in the synthetic division and the Graeffe process it is possible to insure that the error term d_n satisfies $|d_n| << 1$.

Since there will be at least one root of the polynomial equation near the circumference C_n of the circle of radius $R^*(p_n)$ about the center p_n, if a suitable set, S_n, of points (such as the vertices of a regular inscribed heptagon or octagon [3]) on C_n are chosen, S_n will contain at least one point s such that $R^*(s) \leq \frac{1}{2}R^*(p_n)$. Choose any one, s (e.g., the first one which is tried) of these points, and call it p_{n+1}.

Starting with the initial approximation $p_0 = 0$ and finding each successive p_n by iterative applications of the above procedure, one can easily verify by induction from $R^*(p_{n+1}) \leq \frac{1}{2}R^*(p_n)$ that $R^*(p_n) \leq 2^{-n}R^*(p_0)$.

Then, if p denotes a root of the original equation which is nearest to p_n, it follows that $|p| \geq R(p_0) = (1 + d_0)R^*(p_0)$.

The relative error of p_n, considered as an approximation to p, is

$$\frac{|p_n - p|}{|p|} = \frac{R(p_n)}{|p|} \leq \frac{(1 + d_n)R^*(p_n)}{(1 + d_0)R^*(p_0)} \leq \frac{(1 + d_n)2^{-n}R^*(p_0)}{(1 + d_0)R^*(p_0)} = \frac{1 + d_n}{1 + d_0} 2^{-n} \approx 2^{-n}.$$

It is clear that on a binary computer the nth approximation p_n is good to about n significant figures, if the computer carries several extra significant figures in the calculation to assure that $|d_n|$ is always sufficiently small.

This method is universal in that it does not depend on the location or multiplicities of the roots or on the degree of the original equation.

Since the Graeffe process easily gives the number of roots having the absolute value R, the multiplicity of the root to which p_n converges would be apparent without further calculation.

Complex numbers can arise in the course of the calculation, even when the original coefficients are real numbers. For this reason, and also because very large powers of 2 or 10 can arise as exponents, the use of corresponding complex, floating-point addition and multiplication computer subroutines will be necessary. Such subroutines make it possible for the original equation to have complex coefficients covering a very wide range of magnitudes without involving extra programing.

The time required to find one root by this method (as calculated from the number of multiplications and additions) will be proportional to the square of the degree of the original polynomial. It follows that the time required to find all roots will be proportional to the cube of the degree of the polynomial.

Several modifications of this method could be made which would speed up convergence, but probably not sufficiently to make the method actually feasible for extensive machine use. The constant $\frac{1}{2}$, the number of points in S_n, and the degree of accuracy to which R should be calculated, could be adjusted to minimize the time. Also, it would save time to test for convergence during the computation rather than to iterate a fixed number of times.

The author wishes to express his appreciation to Dr. L. B. TUCKERMAN, JR., Dr. E. W. CANNON, & Mr. GEORGE GOURRICH, all of the NBS, for their suggestions and assistance in developing this method.

EDWARD F. MOORE

Brown University

[1] E. BODEWIG, "On Graeffe's method for solving algebraic equations," *Quart. Appl. Math.*, v. 4, 1946, p. 177–190.

[2] BERNARD DIMSDALE, "On Bernoulli's method for solving algebraic equations," *Quart. Appl. Math.*, v. 6, 1948, p. 77–81.

[3] The choice of an octagon will allow enough overlap to permit this method to converge even if $|d_n|$ is fairly large.

BIBLIOGRAPHY Z–VIII

1. ANDREW D. BOOTH & KATHLEEN H. V. BRITTEN, *Coding for ARC*, [Automatic Relay Computer]. Second edition, July 1948, 39 leaves. British Rubber Producers' Research Association, Welwyn Garden City, Herts., England. 20.3 × 25.4 cm. Not available for distribution.
2. ANDREW D. BOOTH & KATHLEEN H. V. BRITTEN, *General Considerations in the Design of an All-Purpose Electronic Digital Computer.* Second edition, Aug. 1947, 24 leaves, mimeographed. British Rubber Producers' Research Association, Welwyn Garden City. 21.6 × 27.9 cm. Not available for distribution. Copies of nos. 1 and 2 may be consulted in the Library of The Institute for Advanced Study.
3. ARNOLD A. COHEN & WILLIAM R. KEYE, "Selective alteration of digital data in a magnetic drum computer memory," transcript of a paper presented at the Institute of Radio Engineers, National Convention, May 1,