NBS TECHNICAL NOTE 572

UNITED STATES
DEPARTMENT OF
COMMERCE
PUBLICATION

On Computer Performance Measurement Programming

Measuring Indexing Adroitness By Isolating Complex Primes

U.S.
DEPARTMENT
OF
COMMERCE
National
Bureau
of
Standards

NATIONAL BUREAU OF STANDARDS

The National Bureau of Standards[1] was established by an act of Congress March 3, 1901. The Bureau's overall goal is to strengthen and advance the Nation's science and technology and facilitate their effective application for public benefit. To this end, the Bureau conducts research and provides: (1) a basis for the Nation's physical measurement system, (2) scientific and technological services for industry and government, (3) a technical basis for equity in trade, and (4) technical services to promote public safety. The Bureau consists of the Institute for Basic Standards, the Institute for Materials Research, the Institute for Applied Technology, the Center for Computer Sciences and Technology, and the Office for Information Programs.

THE INSTITUTE FOR BASIC STANDARDS provides the central basis within the United States of a complete and consistent system of physical measurement; coordinates that system with measurement systems of other nations; and furnishes essential services leading to accurate and uniform physical measurements throughout the Nation's scientific community, industry, and commerce. The Institute consists of a Center for Radiation Research, an Office of Measurement Services and the following divisions:

> Applied Mathematics—Electricity—Heat—Mechanics—Optical Physics—Linac Radiation[2]—Nuclear Radiation[2]—Applied Radiation[2]—Quantum Electronics[3]—Electromagnetics[3]—Time and Frequency[3]—Laboratory Astrophysics[3]—Cryogenics[3].

THE INSTITUTE FOR MATERIALS RESEARCH conducts materials research leading to improved methods of measurement, standards, and data on the properties of well-characterized materials needed by industry, commerce, educational institutions, and Government; provides advisory and research services to other Government agencies; and develops, produces, and distributes standard reference materials. The Institute consists of the Office of Standard Reference Materials and the following divisions:

> Analytical Chemistry—Polymers—Metallurgy—Inorganic Materials—Reactor Radiation—Physical Chemistry.

THE INSTITUTE FOR APPLIED TECHNOLOGY provides technical services to promote the use of available technology and to facilitate technological innovation in industry and Government; cooperates with public and private organizations leading to the development of technological standards (including mandatory safety standards), codes and methods of test; and provides technical advice and services to Government agencies upon request. The Institute also monitors NBS engineering standards activities and provides liaison between NBS and national and international engineering standards bodies. The Institute consists of the following technical divisions and offices:

> Engineering Standards Services—Weights and Measures—Flammable Fabrics—Invention and Innovation—Vehicle Systems Research—Product Evaluation Technology—Building Research—Electronic Technology—Technical Analysis—Measurement Engineering.

THE CENTER FOR COMPUTER SCIENCES AND TECHNOLOGY conducts research and provides technical services designed to aid Government agencies in improving cost effectiveness in the conduct of their programs through the selection, acquisition, and effective utilization of automatic data processing equipment; and serves as the principal focus within the executive branch for the development of Federal standards for automatic data processing equipment, techniques, and computer languages. The Center consists of the following offices and divisions:

> Information Processing Standards—Computer Information—Computer Services —Systems Development—Information Processing Technology.

THE OFFICE FOR INFORMATION PROGRAMS promotes optimum dissemination and accessibility of scientific information generated within NBS and other agencies of the Federal Government; promotes the development of the National Standard Reference Data System and a system of information analysis centers dealing with the broader aspects of the National Measurement System; provides appropriate services to ensure that the NBS staff has optimum accessibility to the scientific information of the world, and directs the public information activities of the Bureau. The Office consists of the following organizational units:

> Office of Standard Reference Data—Office of Technical Information and Publications—Library—Office of Public Information—Office of International Relations.

---

[1] Headquarters and Laboratories at Gaithersburg, Maryland, unless otherwise noted; mailing address Washington, D.C. 20234.
[2] Part of the Center for Radiation Research.
[3] Located at Boulder, Colorado 80302.

UNITED STATES DEPARTMENT OF COMMERCE
Maurice H. Stans, Secretary
NATIONAL BUREAU OF STANDARDS • Lewis M. Branscomb, Director

NBS TECHNICAL NOTE 572

ISSUED APRIL 1971
Nat. Bur. Stand. (U.S.), Tech. Note 572, 25 pages (Apr. 1971)
CODEN: NBTNA

On Computer Performance Measurement Programming Measuring Indexing Adroitness By Isolating Complex Primes

George W. Reitwiesner

Office of Information Processing Standards
Center for Computer Sciences and Technology
National Bureau of Standards
Washington, D.C. 20234

NBS Technical Notes are designed to supplement the Bureau's regular publications program. They provide a means for making available scientific data that are of transient or limited interest. Technical Notes may be listed or referred to in the open literature.

For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C., 20402.
(Order by SD Catalog No. C 13.46:572). Price 35 cents.

On Computer Performance Measurement Programming

Measuring Indexing Adroitness by Isolating Complex Primes

George W. Reitwiesner

This writing, describing a computer performance test program, is concerned not primarily with specific measurements, but rather with a procedure for making measurement regarding specific properties of computer operation.

The program is written in a particular problem-oriented programming language; therefore assessment perforce spans the effects of the computer hardware, of the programming language, and of the intervening compiler processes.

The objective of the test is to assess adroitness in certain indexing operations. Assessment is accomplished by measuring execution time of a recursive programming loop.

The test problem was chosen as a convenient artifice to use certain specific indexing-type operations in the programming employed for solution.

The test program performs a simple computation for which the solution is completely definitive, yet for which both the solution and the time for achieving it are variable under parameters whose values are introduced as program input data.

Key words: Assessment; complex; composite; computer; criteria; evaluation; Gaussian primes; indexing; measurement; performance; prime; program; test.

Introduction

This paper describes a digital computer performance test program which was developed to meet a particular set of test criteria.

A programmed digital computer performance test should assess some single basic characteristic of digital computer performance. It should do so objectively in the sense of being insulated against possible effects of especial hardware and logical design features. It should be elevated above the mundane by a touch of the esoteric, yet its employment of the characteristic under test should occur in as natural an environment as possible. It should provide a unique result under conditions which permit the recording of some measurement (such as program execution time) to assess performance in obtaining that result. That result should not be reasonably determinable by any other means, should be variable under the control of parameters whose values are entered as input data at the outset of the performance of the test, and should be precisely reproducible on all processors which operate under the same programming language.

The test program described in this writing has been designed to permit objective assessment of the single characteristic of adroitness in indexing operations, as revealed in the execution of a particularly constructed innermost recursive programming loop. It is written in the FORTRAN problem-oriented language. Two principal indices are used in this innermost loop; one index is administered under a DO statement, and the other is advanced by an integer arithmetic statement; both indices are used in integer arithmetic comparison (IF) statements, and both are used to select a particular element in a two-dimensional array.

When it runs to normal completion, the test program performs the moderately esoteric exercise of isolating all the complex primes over a selected region of the complex plane. In effecting this isolation, the innermost loop employs indexing operations of the type described above in the rather natural programming situation of traversing along the outer edge of an origin-centered circular arc: to locate points at which that arc coincides with intersections of integer grid lines.

Assessment is made by measuring program execution time; and the data which are printed upon completion of the test run include a plot of the unique pattern of complex primes over the selected region, together with certain peculiar counts, such as the multiplicity of recognition of compositeness among non-primes and the frequency of a specially selected index change during execution.

Parametric control is available for the assignment of the exact dimensions of the selected region and, optionally, to effect premature program termination upon completion of a preassigned portion of the total work required for completing the isolation. Apart from program execution time, the results are precisely reproducible on any processor operating under the programming language used.

General Discussion

This writing describes a test program for a computer. The first half of the text describes the test in general terms; the second half presents an analysis of pertinent details.

The objective of the test is to assess adroitness in indexing operations. Indexing in computer programming is a complicated subject; the test described here covers only one aspect of that subject. It assesses the adroitness with which two indices perform conventional operations under non-conventional control and non-conventional operations under conventional control: both indices are used to access elements in a two-dimensional array, and both are used in arithmetic control statements; one index is administered under a loop-control statement, and the other is advanced by an arithmetic assignment statement.

The program is written in FORTRAN. It is sufficiently simple and brief that its translation to another programming language may be performed easily.

The assessment, perforce, spans the effects of the computer hardware and logical design, of the computer-instruction-level language which is generated by the compiler, and of the technique by which the compiler effects the indexing operations required.

Assessment is made by measuring the time of execution of a multi-echelon recursive programming loop. Two time-measuring techniques are provided: interrogating a program-accessible internal clock, if available, and manual timing between pauses. Either, or both, may be selected.

The test program has not been designed to represent an efficient computational procedure. We are not concerned here with efficiency in that sense. We are concerned with indexing adroitness.

Adroitness in basic aspects of indexing operations is demanded by the particular manner in which the innermost recursive loop is programmed. The pertinent features of the structure of this loop are shown in the sample program of Table I.

```
            J=JN
            DO 6 I=IO,IN
            IF(I-J) 2,2,7
          1 J=J-1
          2 IF(M(I,J)- Etc.) 7,5,3
   Loop   3 IF(I-J) 4,7,7
          4 IF(M(I,J-1)-Etc.) 6,1,1
          5 Other Operations
          6 CONTINUE
          7 CONTINUE
```

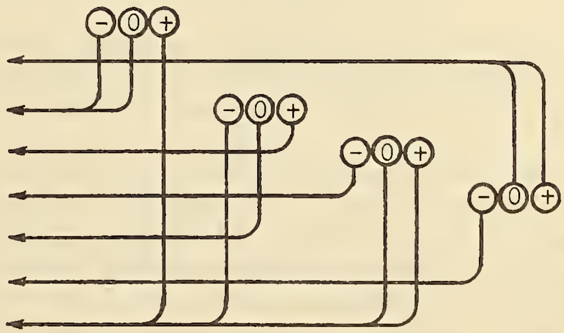


Table I. Sample Program

This sample program, being illustrative, is not complete; and it is detailwise slightly different from that employed in the test program; however, insofar as concerns the pertinent features to which we address our argument, it is logically essentially equivalent to the innermost loop of that program.

A feature of this loop which is of especial interest in regard to demanding indexing adroitness is that each of its first six statements, beginning with the DO 6 I=IO,IN statement and ending with the statement numbered 4, is a basic block terminating statement as defined in paragraph 10.2.7 of the descriptions of the FORTRAN language both in American National Standards Institute document X3.9-1966 and in the Communications of the Association for Computing Machinery, vol. 7, No. 10, October 1964, pages 623-624.

Our interest lies in the net performance resulting from the techniques by which indices span their ranges of application and the manners in which they are employed in the recursive looping which they govern.

In our sample program the indices span their ranges under two techniques: one index is governed by the conventional DO 6 I=IO,IN statement, and the other is (negatively) advanced by the J=J-1 conventional integer arithmetic statement. In the body of the recursion, each index is employed in each of two ways: to access a two-dimensional array (M(I,J)), and in effecting integer magnitude comparisons (IF(I-J) Etc.). Other operations occur in the innermost loop, but (for representative values of the governing parameters) sufficiently infrequently to be disregarded.

The test problem employs only integer numerical data.

The test problem performs a simple computation for which the solution is completely definitive, yet for which both the solution and the time required for achieving it are variable under control of three integer parameters. The roles of these three parameters are described in the following several paragraphs.

The test program employs complex integers, also known as Gaussean integers: complex numbers with integer coefficients. It searches a prescribed region of the complex plane to locate all composite complex integers: those which are the products of other complex integers of non-unit modulus.

Two of the parameters limit the range of interest over the complex plane to an origin-centered cross, with four axes of symmetry which separate the region into eight congruent sectors, as shown in Figure 1.

The larger of these parameters is the upper limit which the magnitude of either coefficient of a complex number is permitted to assume; thus it bounds the coefficient of larger magnitude. The other of these parameters is the upper limit which the magnitude of the other coefficient is permitted to assume; thus it bounds the coefficient of smaller magnitude. (When these parameters are equal, the cross in Figure 1 degenerates to a square.)

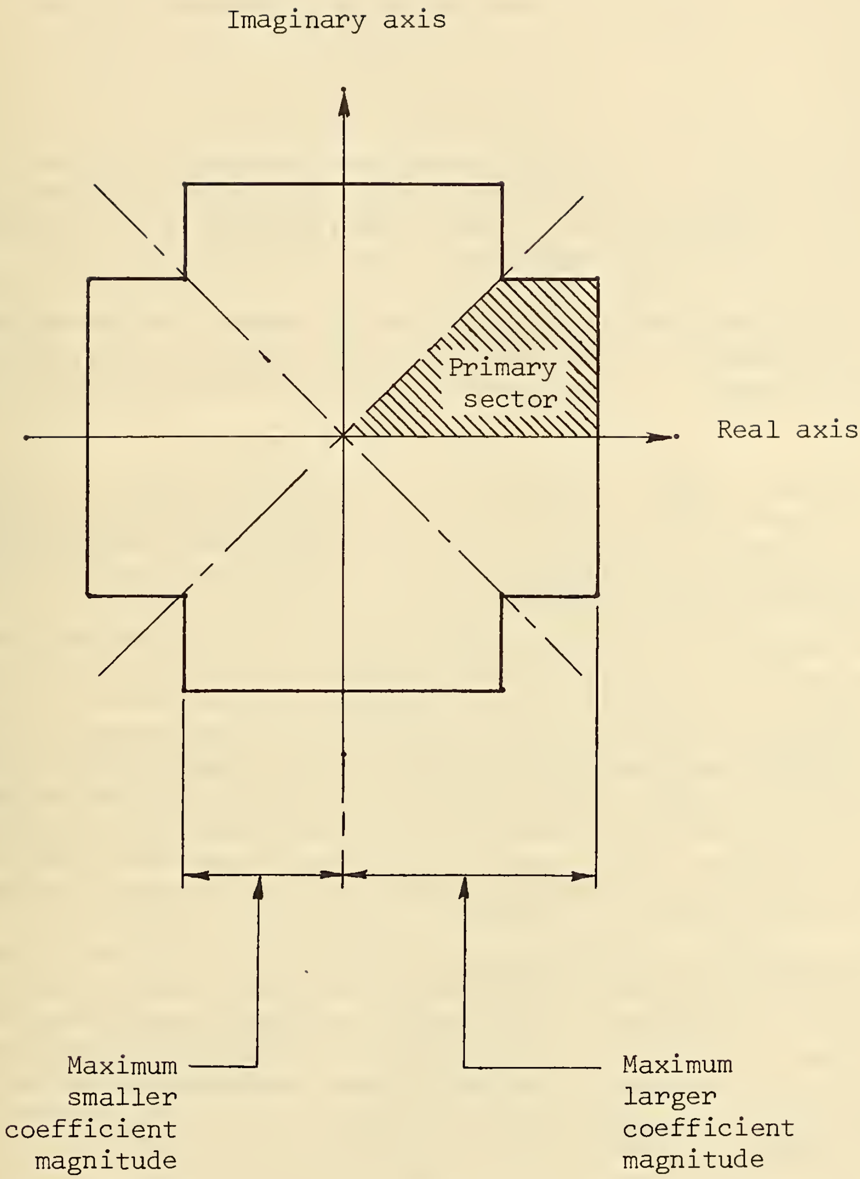

Imaginary axis
Primary sector
Real axis
Maximum smaller coefficient magnitude
Maximum larger coefficient magnitude

Figure 1.

The test program can and does employ the eight-fold symmetry remarked above. It develops data which, through that symmetry, apply throughout the region of interest. It does so, using only complex integers which are contained in (inside or on a boundary of) a particular one of the eight sectors: that one for which both coefficients are non-negative and for which the imaginary coefficient does not exceed the real coefficient. We call this the primary sector; it is shaded in Figure 1.

We concentrate attention on the primary sector.

Complex integers on the real-equals-imaginary diagonal boundary of this sector we call diagonal complex integers. They have the form: x+xi. Except for x=1, they all are composite (factoring into x and 1+i); and, trivially, 1+i is not composite.

The test program requires the performance of comparisons among moduli of complex integers. These comparisons are made, instead, upon their corresponding squares. This insures that only integer numerical data are employed.

For each complex integer contained in the primary sector, the program records two items of data: the square of the modulus, and a counter-marker which counts the number of times, if any, that that complex integer is recognized as composite.

There is an exception: only one data item is recorded for each diagonal complex integer: the square of the modulus.

Taking advantage of the eight-fold symmetry, the program stores the two (triangularly-arrayed) sets of data in a single square array: the squares of the moduli are stored on and below the principal diagonal of that array; and the counter-markers are stored above that diagonal. The program maintains a separate single count of the total number of times compositeness is recognized among all the diagonal complex integers.

The third parameter controls program termination: it is the upper limit on the number of times the program will recognize compositeness among diagonal complex integers. The program terminates prematurely when that number reaches the value of this parameter; however, the program runs to a normal termination when the value of this parameter is zero or exceeds the maximum that that number can reach under the conditions (sector dimensions) imposed by the other parameters.

When it runs to normal termination, the test program searches out and labels (with a count) every composite complex integer of the primary sector, leaving unlabeled (uncounted) only the complex primes (also called Gaussean primes), contained in the region of interest. This distinction (composite vs. prime) is not necessarily valid when the program terminates prematurely under the control of the third parameter.

The precise search procedure is detailed below. In general terms it is a counterpart (albeit a redundant and cumbersome one) to the familiar sieve procedure for isolating real primes in a sequence of real numbers by eliminating all multiples of smaller integers (primes) other than 1.

We are not principally interested in, per se, identifying complex primes.

We employ our sieve search procedure for isolating complex primes merely as an (academically appealing) artifice to afford many-fold reiteration of a particular inner programming loop which has properties of interest to us, and to yield a totally definitive result. For any particular selection of values of the three governing parameters, this definitive result embodies: the (conditionally valid) distinction of composite vs. prime; the compositeness recognition counts remarked above; and one further count which is incorporated for incidental information: the program counts the number of times the J=J-1 index change represented by statement 1 of the sample program of Table 1 is performed throughout the total course of execution.

Upon termination, the program prints two sets of data.

One set contains a miscellany:
(1) the three governing parameters,
(2) the compositeness recognition count for diagonal complex integers,
(3) the sum of all compositeness recognition counts for non-diagonal complex integers,
(4) the sum of the above two counts (3) and (2),
(5) the largest compositeness recognition count among the non-diagonal complex integers and the coefficients of the complex integer associated thereto (or of a particular one of them if duplicate maximum counts exist),
(6) the number of executions performed of the J=J-1 index change of statement 1, and
(7) the execution time of the solution, in seconds. (The execution time is set to zero when internal timing is not employed.)

The other set is a two-dimensional plot of the complex integers which were not recognized as composite, adjusted to include 1+i.

The test program is displayed in appendix A.

It employs exactly one input data card, typified in one of the early comment lines. The first five columns of this card contain certain data which are used for the plot. The next 25 columns contain five five-digit parameters: the first three of these parameters have been accounted above; the last two are auxiliary parameters which select timing and printing options as described in further comment lines. (When internal timing is employed, a suitable factor must be defined (where indicated by the appropriate comment line in the programming) to convert the measured internal timing to seconds.)

A sample output of the test program is displayed in Appendix B.

Analysis of Details

The remainder of this writing presents a more precise analysis of the problem solved by the test program, and of the particular programming procedure employed. It does this in three steps, covering: general matters, complex integer multiplication, and programming details.

For analytical precision, we enumerate the eight equal sectors of the complex plane as I, II, ... VIII, proceeding counterclockwise, beginning with the primary sector which is shaded in Figure 1.

We denote the generic complex integer f, of integer coefficients g and h, as f=gr+hi, where r and i are unit real and immaginary vectors.

We denote the square of the modulus as modsq or m(); thus the modsq of f=gr+hi is $m(f) = g^2 + h^2$.

We have defined (in the text above) the three terms: complex integer, composite complex integer, and complex prime.

We recognize four complex integers of unit modulus: +r, -r, +i, -i; we call them unit complex integers or unit complex primes.

For non-negative g and h, there are in general exactly eight composite complex integers whose coefficients (magnitudes) are g and h (in either order): we refer to them as derivants of each other and denote them f(k,j), as follows:

f(0,0) = gr+hi	f(1,0) = gr-hi	(1)
f(0,1) = -hr+gi	f(1,1) = -hr-gi	
f(0,2) = -gr-hi	f(1,2) = -gr+hi	
f(0,3) = hr-gi	f(1,3) = hr+gi.	

They derive from each other by the rules:

$$f(k,j+1) = (-1)^k(i)(f(k,j)) \qquad (2)$$

f(k+1,j) = congugate of f(k,j),

where k+1 is taken modulo 2, and j+1 is taken modulo 4; i.e.

f(2,j) = f(0,j)	(independent of j)	(3)
f(k,4) = f(k,0)	(independent of k).	

Through obvious symmetries, one derivant of any complex integer is contained in each of the eight sectors of the complex plane, and all eight have equal modsq: $m(f(k,j)) = g^2 + h^2$, independent of k and j.

(We do not belabor the degenerate cases in which the eight derivants of a complex integer are not all distinct, viz: when they coincide in pairs because their coefficients vanish or have equal magnitude.)

The particular derivant in the primary sector we call the primary derivant; it is f(0,0) = gr+hi, for non-negative g and h, with h less than (or =) g.

Figure 2 displays the eight derivants of four complex integers, a,b,c,d, where (to accommodate analysis of complex integer multiplication) the several derivants of c and d represent the products of the derivants of a and b, whence m(c)=m(d) applies, and both c and d lie on the arc (shown) of radius equal to the square root of m(c) and m(d).

In Figure 2 we locate the derivants of a,b,c,d by the symbol o.

One or more other complex integers may have the same modsq as c and d (and thus also lie on the arc in Figure 2) but not be products of derivants of a and b; for later reference, we denote any such complex integer as e and employ the symbol x to locate its eight derivants in Figure 2.

We denote the coefficients of a,b,c,d, and e according to:

a(0,0) = pr+qi (4)
b(0,0) = sr+ti
c(0,0) = vr+wi
d(0,0) = yr+zi
e(0,0) = mr+ni (if e exists).

Turning now to multiplication, we observe that for arbitrary k',j',k",j", there exist 8^2 = 64 ways of multiplying some derivant of a(k',j') by some derivant of b(k",j"). (Trivially, we do not distinguish the order of multiplication: (a)(b) is the same as (b)(a).) By virtue of the factor i in (2), there exists a four-fold duplication among the 64 products. Upon elimination of duplications, there remain 16 distinct complex integers in two sets (c and d) of eight derivants (except when coincidence occurs through degeneration).

We aim to accommodate complex integer multiplication in general, while restricting our attention to the primary sector. To do so, we seek a rule for expressing the primary derivant of each of these two sets of products in terms of the primary derivants of the factors.

One member of one set is the product of the primary derivants of the factors; it is, perforce, contained in one of the two sectors I,II; transformation from sector II to sector I under (2) is straightforward. One member of the other set is the product of the primary derivant of one of the factors and the conjugate of the primary derivant of the other factor; it is, perforce, contained in one of the sectors I,VIII; transformation from sector VIII to sector I under (2) is straightforward.

Thus, employing (4), we have the two multiplications

(a(0,0))(b(0,0)) = (ps-qt)r + (qs+pt)i = either c(0,0) or c(1,3) (5)
(a(0,0))(b(1,0)) = (ps+qt)r + (qs-pt)i = either d(0,0) or d(1,0),

and the rule we seek is

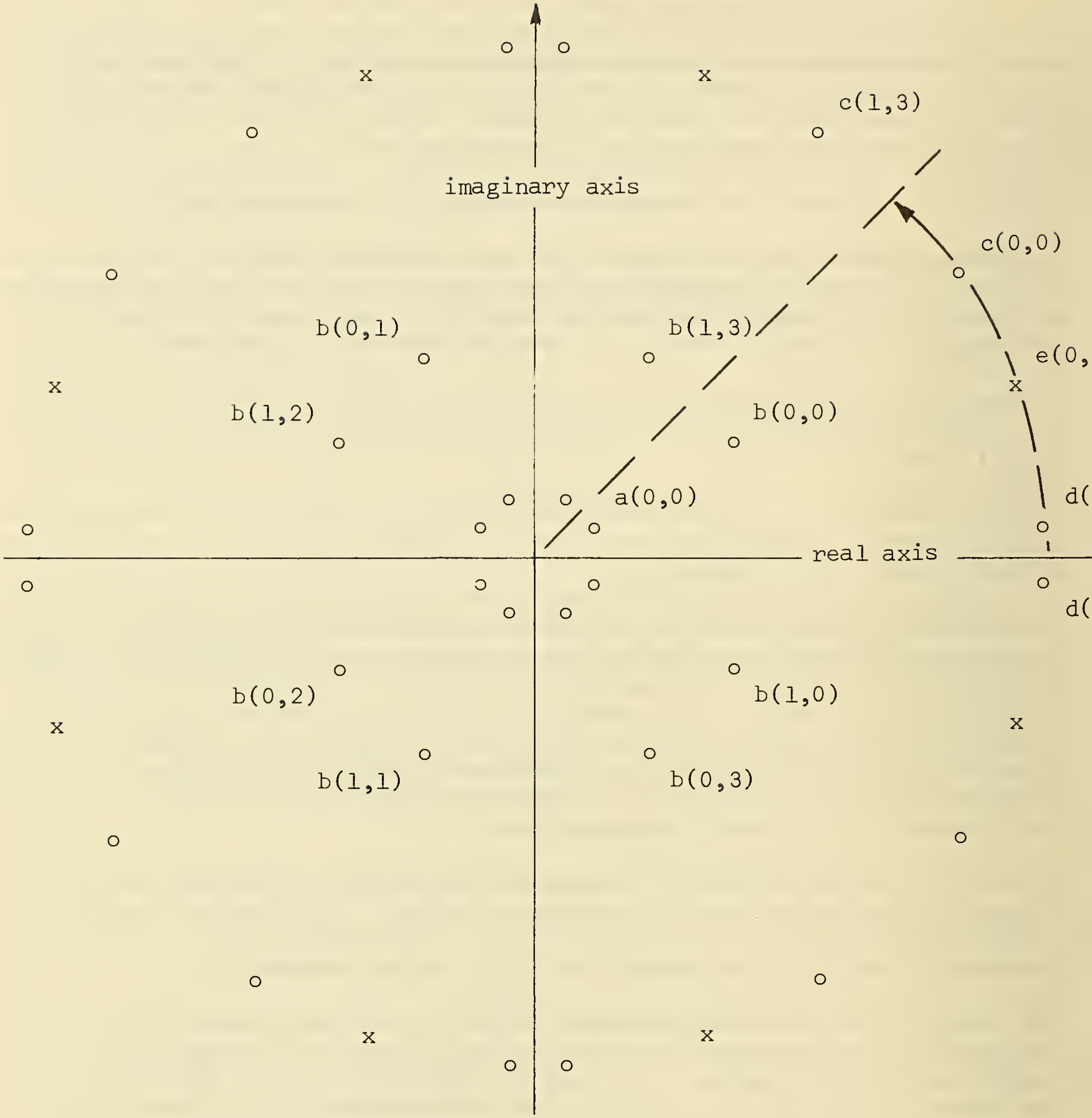

imaginary axis
real axis
c(1,3)
c(0,0)
e(0,
d(
d(
b(0,1)
b(1,3)
b(1,2)
b(0,0)
a(0,0)
b(0,2)
b(1,0)
b(1,1)
b(0,3)

Figure 2.

```
c(0,0) = (max((ps-qt),(pt+qs)))r + (min((ps-qt),(pt+rs)))i        (6)
d(0,0) =                 (ps+qt)r + (magnitude of (pt-qs))i.
```

In specific illustration, we consider the assignments

```
p= 2     q= 1     whence a(0,0) =  2r +  i     and m(a) =   5   (7)
s= 7     t= 4     whence b(0,0) =  7r + 4i     and m(b) =  65
v=15     w=10     whence c(0,0) = 15r +10i     and m(c) = 325
y=18     z= 1     whence d(0,0) = 18r +  i     and m(d) = 325.
```

Then, clearly, for

```
a(0,0) = 2r +  i                                                  (7')
b(0,0) = 7r + 4i,
```

the precise multiplications (5) yield

```
(a(0,0))(b(0,0)) = c(1,3) = 10r + 15i                              (8)
(a(0,0))(b(1,0)) = d(1,0) = 18r -   i,
```

and rule (6) directly yields

```
c(0,0) = 15r + 10i                                                (7")
d(0,0) = 18r +   i.
```

And we observe that in this specific illustrative case, there indeed does exist an e:

```
m=17     n= 6     whence e(0,0) = 17r + 6i     and m(e) = 325.  (7e)
```

We are now equipped to consider complex integer multiplication in general, while restricting out attention to the primary sector of the complex plane.

We turn to the detailed structure of the sieve-search procedure of the test program.

For integer parameters (JN) and (IN) which are greater than 1, with (JN) at least as large as (IN), we define the region of interest (primary sector) as that containing the complex integers f(0,0) = gr + hi for values of g up to (JN) and values of h up to (IN).

We denote the maximum permissible modsq over the region of interest as M; clearly: $M = (JN)^2 + (IN)^2$.

We reject, as irrelevant, all complex integers whose modsq exceeds M.

In Figure 3 we expand Figure 2 to show additional data: (IN), (JN), and M (or, rather, the square root of M). (For the specific assignments of (7) (and (7e)), there apply here: (IN)=12, (JN)=16, and M=400.)

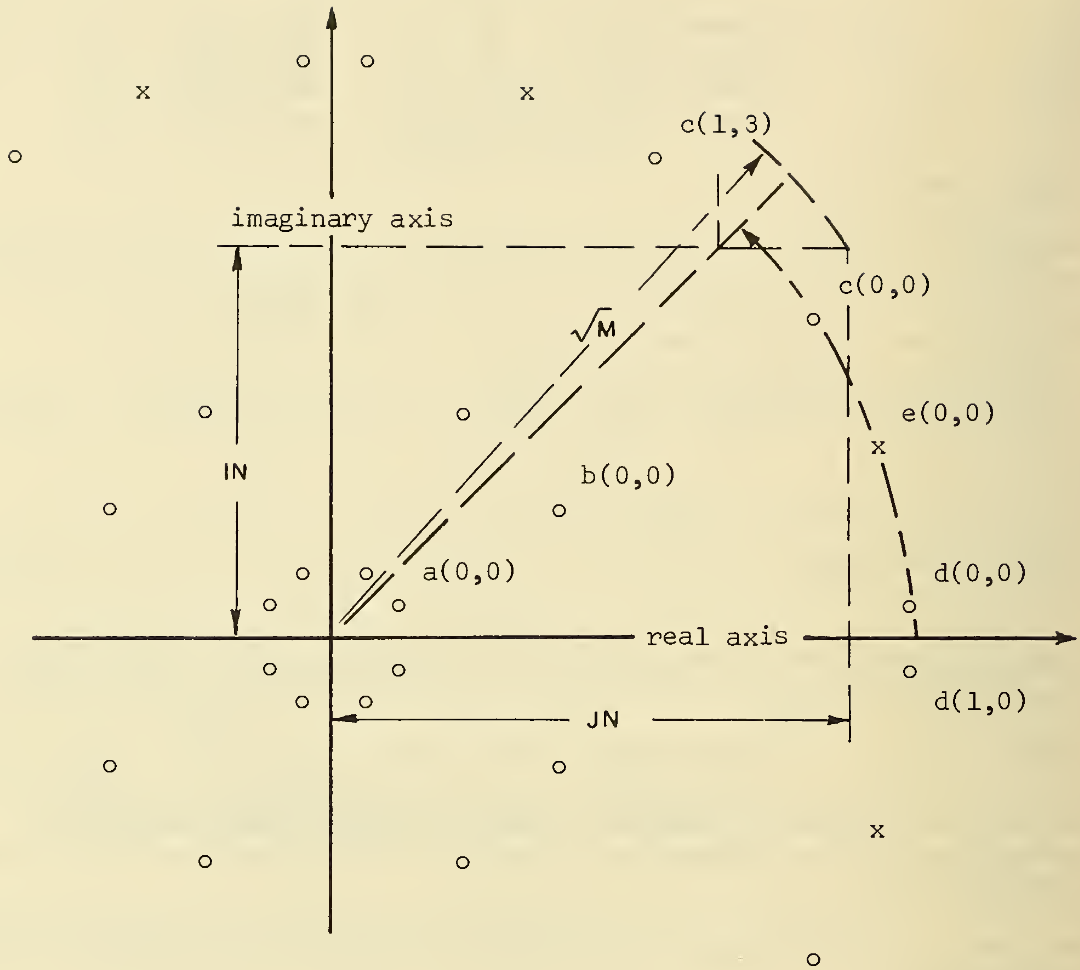


Figure 3.

To discuss our sieve-search procedure, we consider the primary derivants of the two products of the two pairs (k=1,2) of complex integers:

$$a(0,0) = pr + qi \qquad (9)$$
$$b(k,0) = sr + \emptyset ti$$

where

$$\emptyset = (-1)^k. \qquad (10)$$

We have the obvious constraints:

$$(m(pr+qi))^2 = (p^2+q^2)^2 \quad \text{less than or equal to } M \qquad (11a)$$

$$(m(pr+0i))^2 = (p^2)^2 \quad \text{less than or equal to } M \qquad (11p)$$

$$(m(pr+qi))(m(sr+\emptyset ti)) = (p^2+q^2)(s^2+t^2) \quad \text{less than or equal to } M \qquad (11b)$$

$$(m(pr+qi))(m(sr+0i)) = (p^2+q^2)(s^2) \quad \text{less than or equal to } M, \qquad (11s)$$

where (11p) and (11s) are included in (11a) and (11b), respectively.

We force these products to include all composite complex integers in the region of interest. We do so by: (1) requiring that a(0,0) assume, in turn, the succession of values:

1. $a(0,0) = pr+qi = r + i$ (12a)
2. $a(0,0) = pr+qi = 2r$
3. $a(0,0) = pr+qi = 2r + i$
4. $a(0,0) = pr+qi = 2r + 2i$
5. $a(0,0) = pr+qi = 3r$
6. $a(0,0) = pr+qi = 3r + i$
7. $a(0,0) = pr+qi = 3r + 2i$

etc. etc. etc.,

subject to (11a) until (11p) is violated; and (2) for each a(0,0), requiring that b(k,0) assume, in turn, the succession of values:

1. $b(k,0) = sr+\emptyset ti = pr + \emptyset qi$ (12b)
2. $b(k,0) = sr+\emptyset ti = pr + \emptyset(q+1)i$
3. $b(k,0) = sr+\emptyset ti = pr + \emptyset(q+2)i$

etc. etc. etc.

n-q+1. $b(k,0) = sr+\emptyset ti = pr + \emptyset ni$ (n = min (p,(IN)))
n-q+2. $b(k,0) = sr+\emptyset ti = (p+1)r$
n-q+3. $b(k,0) = sr+\emptyset ti = (p+1)r + \emptyset i$
n-q+4. $b(k,0) = sr+\emptyset ti = (p+1)r + 2\emptyset i$

etc. etc. etc.,

subject to (11b) until (11s) is violated.

(A straightforward procedure for merely isolating complex primes would be that of: excluding from (12a) and (12b) all previously recognized composite complex integers; and forming, for each residual pairs (a(0,0) and b(k,0)), the associated primary derivants of the products under (6), ignoring any products which lie outside the region of interest, and marking as composite those which do not. We choose a more complicated (and redundant) procedure, for two reasons: for the especial purpose of employing an innermost loop which has the particular index-employing features we wish to engage, and, incidentally, to elevate representative solution time to reasonably recognizable magnitude.)

The test program employs its innermost programming loop (typified in Table I) immediately subordinate to an obvious four-echelon loop execution of (12a) and (12b), precisely and fully as detailed above.

For each of the pairs a(0,0) and b(k,0) which are developed by (12a) and (12b), the innermost loop forms the product (m(a))(m(b)) and searches the region of interest to discover complex integers whose modsq are equal to that product. Except in occasional degenerate cases, there exist, perforce, at least two such complex integers: the c(0,0) and d(0,0) of (6); but one or both of these may lie outside the region of interest. And there may exist more, as illustrated in (7e).

This search is made generally along the outside edge of the arc shown in Figure 3 --- more precisely: it is made, in the direction shown by the arrow-head, along the outer edge of such portion of that arc as is contained in the region of interest. Indeed, the innermost loop acquires the particular features to which we address our argument by the combined circumstances of: (1) traversing along the outer edge of the arc and (2) accessing the stored table of modsq.

For each complex integer for which this equality is discovered, the inner loop: (1) ascertains whether or not it is indeed one of the primary derivants c(0,0) or d(0,0) of (6); (2) ignores it if it is not; and (3) raises the pertinent count if it is.

```
                                               COMPLEX SIEVE 6 OCTOBER 66   PAGE
CARD SIGNAL TO BEGIN NEW PAGE IN LISTING   COMPLEX SIEVE 6 OCTOBER 66.
C     SIEVE PROGRAM FOR SEPARATING COMPLEX COMPOSITE INTEGERS AND PRIMES
C THE * MARKS THE SEVERAL POINTS AT WHICH THIS PROGRAM MUST BE UPDATED *
C* 0/   39   40  999    1    0
C     ABOVE IS A SAMPLE OF THE INPUT DATA CARD
C     THE FIRST COLUMN IS IRRELEVANT ON THE DATA CARD
C     COLUMNS 2-4 CONTAIN OPTIONAL CHARACTERS FOR TABLE PRINTING
C     THE FIFTH COLUMN IS IRRELEVANT ON THE DATA CARD
C     COLUMNS 6-30 CONTAIN FIVE FIVE-DIGIT INTEGER PARAMETERS
C     THE ROLES OF THE FIRST THREE PARAMETERS ARE ACCOUNTED BELOW
C     THE FIFTH PARAMETER GOVERNS THE METHOD USED FOR TIMING
C     IF ITIME=+1 THE PROGRAM WILL PAUSE TWICE TO PERMIT MANUAL TIMING
C     IF ITIME= 0 THE PROGRAM WILL EMPLOY AUTOMATIC CLOCK INTERROGATION
C     AND TWO APPROPRIATE INTERROGATION STATEMENTS MUST BE EMPLOYED AND
C     ELAPSED TIME MUST BE EXPRESSED IN SECONDS THROUGH SUBSTITUTING AN
C     APPROPRIATE ASSIGNMENT TO THE QUANTITY FACTOR WHERE SHOWN BELOW
C     IF ITIME=-1 BOTH MANUAL AND AUTOMATIC TIMING APPLY
C     THE FOURTH PARAMETER GOVERNS THE PRINTOUT
C     IF ILABEL = 1, THE CHARACTER IN COLUMN 2 MARKS PRIMES
C     IF ILABEL = 1, THE BLANK ( ) IN COLUMN 3 MARKS COMPOSITE NUMBERS
C     IF ILABEL = 1, THE CHARACTER IN COLUMN 4 MARKS THE UNUSED CORNER
C     IF ILABEL = 0, THE DIGIT 0 MARKS COMPOSITE NUMBERS
C     IF ILABEL = 0, THE DIGIT 1 MARKS PRIMES
C
C     EMPLOY TABLE M(-,-) OF SQUARES OF MODULI AND INITIAL PRIME MARKERS
C
C     E                1  1  1  1  1  -  -   ...,JN=7   MARKERS ARE      -
C     X                1  1  1  1  1  -  -          6   ABOVE THE      -
C     A  F  JNO=6      1  1  1  1 32 41 52   ...IN=5   DIAGONAL     *
C     M  O   AND       1  1  1 18 25 36 45          4             *
C     P  R  INO=4      1  1  8 13 20 29 40          3           *    SQUARES
C     L                1  2  5 10 17 26 37          2         *    ARE BELOW
C     E                0  1  4  9 16 25 36          1       *    THE DIAGONAL
C
C     THE PROGRAMMING SEARCHES TABLE M(-,-) FOR COMPOSITE NUMBERS UNDER
C     A PARTICULAR SCHEME AND COUNTS THE NUMBER OF RECOGNITIONS OF EACH
C     UNDER THE SEARCH SCHEME EVERY POSSIBLE PRODUCT IS FORMED OF
C     COMPLEX NUMBERS WITHIN THE RANGE OF INTEREST IN THE COMPLEX PLANE
C
      DIMENSION MARKS(5)
C     LJP MUST BE AT LEAST 5 LESS THAN THE NUMBER OF PRINTING POSITIONS
C     LJP MUST BE AT LEAST 100 AND MUST NOT EXCEED 195
C     LJN MUST NOT EXCEED LJP
C*    DIMENSION LABLE(LJP)
      DIMENSION LABLE(115)
C*    DIMENSION M(LJN,LJN) (UPPER CORNER (ABOVE IN,IN) WILL NOT BE USED)
      DIMENSION M(        )
C*    LJP=
      LJP=115
C*    LJN=
      LJN=
C*    THE ABOVE ELEVEN LINES MUST BE COMPATABLE
C     CHECK FOR LJN .LE. LJP AND 100 .LE. LJP .LE. 195
      IF(LJN-LJP)                                          40,   40,  999
   40 IF(100-LJP)                                          41,   41,  999
   41 IF(LJP-195)                                          42,   42,  999
   42 CONTINUE
```

```
CARD SIGNAL TO BEGIN NEW PAGE IN LISTING
C     IDENTIFY I/O UNITS ...., IDI FOR INPUT AND IDO FOR OUTPUT
C*    IDI=
      IDI=
C*    IDO=
      IDO=
C     FORMATS
  100 FORMAT(1H )
  101 FORMAT(1H1)
  102 FORMAT(9I6,I8,F10.3)
  103 FORMAT( 5A1,5I5)
  106 FORMAT(54H1  MIN   MAX  DIAG  DIAG  COMP TOTAL INDEX INDEX TALLY
     1       18H     KEY   TIME IN)
  107 FORMAT(54H COEFF COEFF LIMIT COUNT COUNT COUNT  IMAX  JMAX   MAX
     1       18H   COUNT   SECONDS)
C*110 FORMAT(1H I3,1H (LJP)I1)
  110 FORMAT(1H I3,1H  115 I1)
C*111 FORMAT(1H I3,1H (LJP)A1)
  111 FORMAT(1H I3,1H  115 A1)
C*130 FORMAT(  5X,(LJP    )I1)
  130 FORMAT(  5X,      115 I1)
C*131 FORMAT( 15X,(LJP- 10)I1)
  131 FORMAT( 15X,      105 I1)
C*132 FORMAT(105X,(LJP-100)I1)
  132 FORMAT(105X,       15 I1)
C*    THE ABOVE TEN LINES MUST BE COMPATABLE
C     READ AND WRITE AND CHECK PARAMETERS (2.LE.INO.LE.JNO.LT.LNJ)
C     INO,JNO ARE LIMITS ON VALUES OF COEFFICICENTS OF COMPLEX NUMBERS
C     IN=INO+1 AND JN=JNO+1 ARE CORRESPONDING TABLE INDEX LIMITS
C     KN IS MAXIMUM ADMISSIBLE NUMBER OF PRODUCTS ON PRINCIPAL DIAGONAL
C     THE PROGRAMMING ABORTS UPON THE (KN)TH RECOGNITION OF A COMPLEX
C     NUMBER ON THE DIAGONAL RENDERING PRIME IDENTIFICATION INCONCLUSIVE
C     EXCEPT KN=0 PERMITS UNLIMITED PRODUCTS ON THE PRINCIPAL DIAGONAL
      READ  (IDI,103) (MARKS(I),I=1,5),     INO,JNO,KN,ILABLE,ITIME
      WRITE (IDO,101)
      WRITE (IDO,102)                       INO,JNO,KN,ILABLE,ITIME
      IF(2-INO)                                         162,  162,  999
  162 IF(INO-JNO)                                       163,  163,  999
  163 IF(JNO-LJN)                                       164,  999,  999
  164 IN=INO+1
      JN=JNO+1
C     COMPUTE SQUARES OF MODULI AND SET PRIME MARKERS AS IN TABLE ABOVE
      DO 167 J=1,LJN
  167 M(1,J)=(J-1)**2
      DO 169 I=2,LJN
      DO 169 J=I,LJN
      M(J,I-1)=1
  169 M(I,J)=M(1,I)+M(1,J)
C     INITIALIZE COUNTERS FOR DIAGONAL PRODUCTS, COMPOSITE NUMBERS, KEY
      KO=0
      KC=0
      KEY=0
```

```
CARD SIGNAL TO BEGIN NEW PAGE IN LISTING
C      PRESET TIMING READINGS TO ZEROS
       TIMEON=0.
       TIMEUP=0.
C      INTERROGATE CLOCK
       IF(ITIME)                                       190,  191,  190
   190 PAUSE
       IF(ITIME)                                       191,  191,  199
   191 CONTINUE
C      INSERT CLOCK INTERROGATION HERE TO DETERMINE TIMEON
C*     CALL CLOCK FUNCTION FOR TIMEON
       CALL
       TIMEON=
   199 CONTINUE
C      ESTABLISH COMPOSITE NUMBERS AS PRODUCTS OF LOWER AND UPPER FACTORS
C      RANGE LOWER FACTOR, EXCLUDING UNIT PRIME AND EXCESSIVE MODULUS
       IE=2
C      IE=1 WILL BE SET AS FINAL ACTION OF EACH ITERATION OF LOOP JL=2,JN
       DO 298 JL=2,JN
       IF(M(1,JL)**2-M(IN,JN))                         207,  207,  299
   207 DO 297 IL=IE,IN
       IF(IL-JL)                                       212,  212,  298
C      RANGE UPPER FACTOR, EXCLUDING EXCESSIVE PRODUCT MODULUS
   212 DO 296 JU=JL,JN
       IF(M(IL,JL)*M(1,JU)-M(IN,JN))                   217,  217,  297
   217 DO 295 IU=1,IN
       IF(IU-JU)                                       222,  222,  296
C      EXCLUDE IU LESS THAN IL WHEN JL EQUALS JU
   222 IF(JL-JU)                                       227,  223,  999
   223 IF(IL-IU)                                       227,  227,  295
C      COMPUTE RADIUS MODULUS (SQUARED)
   227 MODRAD=M(IL,JL)*M(IU,JU)
C      EXCLUDE EXCESSIVE PRODUCT MODULUS
       IF(MODRAD-M(IN,JN))                             232,  232,  296
C      LOCATE INITIAL INDICIES FOR SEARCH FOR PRODUCT MODULUS
   232 IO=1
       J=JN
       IF(MODRAD-M(1,JN))                              242,  252,  247
C      INITIATE SEARCH FROM REAL AXIS AT (1,J)
   242 DO 244 J=2,JN
       IF(MODRAD-M(1,J))                               252,  252,  244
   244 CONTINUE
C      INITIATE SEARCH FROM REAL=(JU-1) AT (IO,JN)
   247 DO 249 IO=1,IN
       IF(MODRAD-M(IO,J))                              252,  252,  249
   249 CONTINUE
C      COMPUTE (INDICES OF) REAL COORDINATES OF PRODUCTS
   252 JX=(JL-1)*(JU-1)-(IL-1)*(IU-1)+1
       JR=(JL-1)*(IU-1)+(JU-1)*(IL-1)+1
       IF(JX-JR)                                       255,  256,  256
   255 JX=JR
   256 JR=(JL-1)*(JU-1)+(IL-1)*(IU-1)+1
```

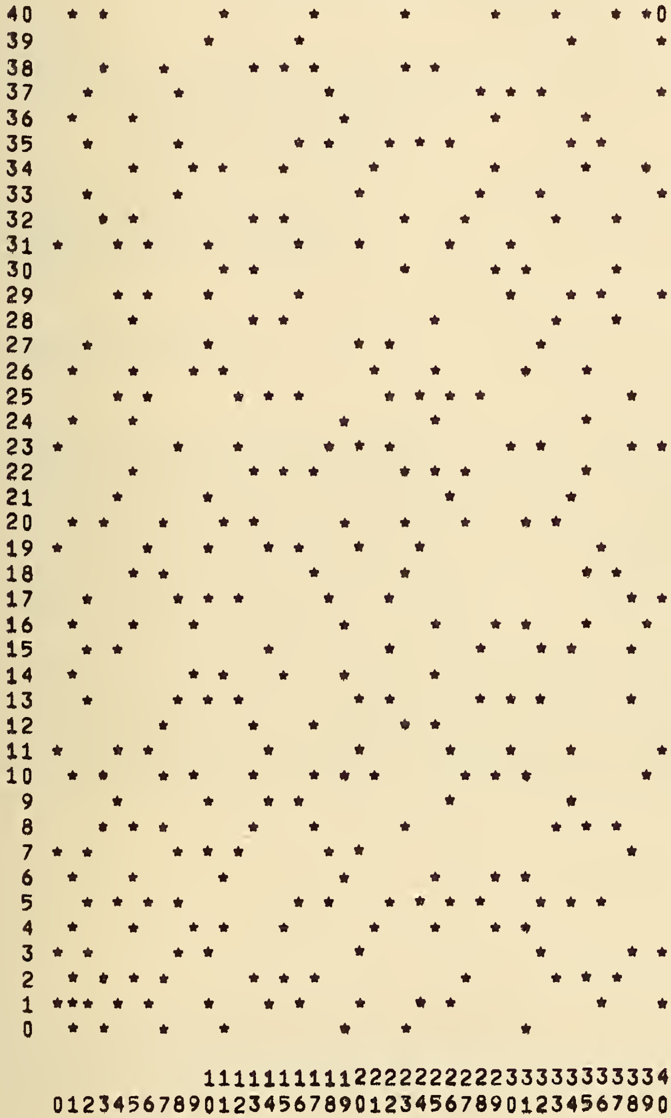
```
CARD SIGNAL TO BEGIN NEW PAGE IN LISTING
C      LOCATE PRODUCT MODULI (EXCEPTING THOSE OUTSIDE REGION OF CONCERN)
       DO 279 I=IO,IN
C      BRACKET MODULUS WITH INDEX J
       IF(I-J)                                           261,  261,  295
  260  J=J-1
       KEY=KEY+1
  261  IF(MODRAD-M(I,J))                                 262,  266,  999
  262  IF(I-J)                                           263,  295,  295
  263  IF(MODRAD-M(I,J-1))                               260,  260,  279
C      CHECK PRODUCT COORDINATE AGREEMENT FOR FIRST RECOGNITION
  266  IF(J-JR)                                          273,  269,  279
C      SEPARATE DIAGONAL CASE FOR FIRST RECOGNITION
  269  IF(I-J)                                           270,  287,  295
C      TALLY NON-DIAGONAL CASE
  270  M(J,I)=M(J,I)+1
C      EXCLUDE DUPLICATE CASES FROM SECOND RECOGNITION CHECK
       IF(JX-JR)                                         279,  295,  295
C      CHECK PRODUCT COORDINATE AGREEMENT FOR SECOND RECOGNITION
  273  IF(J-JX)                                          295,  280,  279
C      CONTINUE SEARCH
  279  CONTINUE
C      SECOND NON-DIAGONAL PRODUCT MODULUS NOT LOCATED WITHIN RANGE OF IN
       GO TO 295
C      SEPARATE DIAGONAL CASE FOR SECOND RECOGNITION
  280  IF(I-J)                                           283,  287,  295
C      TALLY NON-DIAGONAL CASE
  283  M(J,I)=M(J,I)+1
C      TERMINATE SEARCH ON SECOND (NON-DIAGONAL) TALLY
       GO TO 295
C      TALLY AND COUNT DIAGONAL CASE
  287  IF(KN)                                            999,  295,  288
  288  KO=KO+1
       IF(KO-KN)                                         295,  299,  999
C      LOOP ENDINGS
  295  CONTINUE
  296  CONTINUE
  297  CONTINUE
  298  IE=1
C      END OF REITERATION
  299  CONTINUE
```

```
CARD SIGNAL TO BEGIN NEW PAGE IN LISTING
C      INTERROGATE CLOCK
       IF(ITIME)                                              300,  300,  301
  300 CONTINUE
C      INSERT CLOCK INTERROGATION HERE TO DETERMINE TIMEUP
C*     CALL CLOCK FUNCTION FOR TIMEUP
       CALL
       TIMEUP=
       IF(ITIME)                                              301,  302,  302
  301 PAUSE
  302 CONTINUE
C      COMPUTE ELAPSED TIME IN SECONDS BY APPROPRIATELY ASSIGNED FACTOR
       FACTOR=1.
C*     FACTOR=
       FACTOR=
       TIME=(TIMEUP-TIMEON)*FACTOR
C      FINAL SETTING OF DIAGONAL AND FILL-OUT OF POSITIVE QUARTER-PLANE
C      AND SEARCH FOR MAXIMUM NON-DIAGONAL COMPOSITE NUMBER TALLY
       MAXC=0
       DO 306 I=1,IN
C      SET COMPOSITE DIAGONAL MARKS (ADJUST M(2,2) LATER)
       M(I,I)=0
       DO 306 J=I,JN
       IF(MAXC-M(J,I))                                        303,  304,  304
C      RECORD LARGEST COUNT DATA
  303 MAXC=M(J,I)-1
       MAXJ=J-1
       MAXI=I-1
  304 IF(M(J,I)-1)                                            306,  306,  305
  305 KC=KC+M(J,I)-1
       M(J,I)=0
  306 M(I,J)=M(J,I)
C      FINAL ADJUSTMENT FOR LONE DIAGONAL PRIME M(2,2)=1
       M(2,2)=1
C      COMPUTE TOTAL TALLY
       KT=KC+KO
```

```
CARD SIGNAL TO BEGIN NEW PAGE IN LISTING
C       OUTPUT
C       WRITE KEY DATA
        WRITE (IDO,106)
        WRITE (IDO,107)
        WRITE (IDO,102) INO,JNO,KN,KO,KC,KT,MAXI,MAXJ,MAXC,KEY,TIME
        WRITE (IDO,100)
C       WRITE POSITIVE QUARTER PLANE
        DO 320 I=1,JN
        NN=JN
        IF(I-(JN-IN))                                     313,  313,  314
   313  NN=IN
   314  K=JN-(I-1)
        DO 315 L=1,JN
   315  LABLE(L)=MARKS(4)
        DO 316 L=1,NN
        KK=3-M(K,L)
   316  LABLE(L)=MARKS(KK)
        L=K-1
        IF(ILABLE)                                        320,  319,  317
C       CHECK DISTINCTION BETWEEN CHARACTERS IN DATA CARD COLUMNS 2 AND 3
   317  IF(MARKS(2)-MARKS(3))                             318,  319,  318
   318  WRITE (IDO,111) L,(LABLE(KK),KK=1,JN)
        GO TO 320
   319  WRITE (IDO,110) L,(M(K,KK),KK=1,NN)
   320  CONTINUE
C       WRITE INDICES
        DO 322 I=1,10
   322  LABLE(I)=I-1
C       HUNDREDS LINE
        IF(JN-100)                                        324,  324,  327
   324  WRITE (IDO,100)
        GO TO 332
   327  DO 328 I=101,JN
   328  LABLE(I)=LABLE(2)
        WRITE (IDO,132) (LABLE(I),I=101,JN)
C       TENS LINE
   332  IF(JN-10)                                         334,  334,  337
   334  WRITE (IDO,100)
        GO TO 347
   337  DO 340 I=11,JN
        J=(I-1)/10+1
        IF(J-10)                                          340,  340,  339
   339  J=J-10
   340  LABLE(I)=LABLE(J)
        WRITE (IDO,131) (LABLE(I),I= 11,JN)
C       UNITS LINE
        DO 344 I=11,JN
   344  LABLE(I)=LABLE(I-10)
   347  WRITE (IDO,130) (LABLE(I),I=  1,JN)
        WRITE (IDO,101)
C       TERMINAL
   999  STOP
         END
```

Appendix B Sample Output

MIN COEFF	MAX COEFF	DIAG LIMIT	DIAG COUNT	COMP COUNT	TOTAL COUNT	INDEX IMAX	INDEX JMAX	TALLY MAX	KEY COUNT	TIME IN SECONDS
39	40	999	138	1839	1977	20	40	14	8636	0



FORM NBS-114A (1-71)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET	1. PUBLICATION OR REPORT NO. NBS TN 572	2. Gov't Accession No.	3. Recipient's Accession No.

4. TITLE AND SUBTITLE

On Computer Performance Measurement Programming-Measuring Indexing Adroitness by Isolating Complex Primes

5. Publication Date

April 1971

6. Performing Organization Code

7. AUTHOR(S)

George W. Reitwiesner

8. Performing Organization

9. PERFORMING ORGANIZATION NAME AND ADDRESS

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
WASHINGTON, D.C. 20234

10. Project/Task/Work Unit No.

6102171

11. Contract/Grant No.

None

12. Sponsoring Organization Name and Address

National Bureau of Standards
Center for Computer Sciences & Technology

13. Type of Report & Period Covered

Interim 1970-1971

14. Sponsoring Agency Code

15. SUPPLEMENTARY NOTES

16. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.)

This writing describes a test program for a computer. It is concerned not with specific measurements taken, but rather with a procedure for making measurements regarding a specific property of computer operation.

The program is written in a particular problem-oriented programming language; therefore assessment spans the effects of the computer hardware, of the programming language, and of the compiling process.

The objective of the test is to assess adroitness in certain indexing operations. Assessment is accomplished by measuring execution time of a recursive programming loop.

The test problem was chosen as a convenient artifice to use specific indexing-type operations in the programming employed for solution.

The test problem performs a simple computation for which the solution is completely definitive, yet for which both the solution and the time for achieving it are variable under parameters whose values are introduced as program input data.

17. KEY WORDS (Alphabetical order, separated by semicolons)

Assessment; complex; composite; computer; criteria; evaluation; Gaussian primes; indexing; measurement; performance; prime; program; test.

18. AVAILABILITY STATEMENT	19. SECURITY CLASS (THIS REPORT)	21. NO. OF PAGES
[X] UNLIMITED.	UNCLASSIFIED	23
[ ] FOR OFFICIAL DISTRIBUTION. DO NOT RELEASE TO NTIS.	20. SECURITY CLASS (THIS PAGE) UNCLASSIFIED	22. Price $0.35

USCOMM-DC 66244-P71

Latest developments in the subject area of this publication, as well as in other areas where the National Bureau of Standards is active, are reported in the NBS Technical News Bulletin. See following page.

HOW TO KEEP ABREAST OF NBS ACTIVITIES

Your purchase of this publication indicates an interest in the research, development, technology, or service activities of the National Bureau of Standards.

The best source of current awareness in your specific area, as well as in other NBS programs of possible interest, is the TECHNICAL NEWS BULLETIN, a monthly magazine designed for engineers, chemists, physicists, research and product development managers, librarians, and company executives.

If you do not now receive the TECHNICAL NEWS BULLETIN and would like to subscribe, and/or to review some recent issues, please fill out and return the form below.

(cut here)

Mail to: Office of Technical Information and Publications
National Bureau of Standards
Washington, D. C. 20234

Name ______________________________

Affiliation ______________________________

Address ______________________________

City ____________ State ______ Zip ______

☐ Please send complimentary past issues of the Technical News Bulletin.

☐ Please enter my 1–yr subscription. Enclosed is my check or money order for $3.00 (additional $1.00 for foreign mailing).
Check is made payable to: SUPERINTENDENT OF DOCUMENTS.

TN 572

☆ U.S. GOVERNMENT PRINTING OFFICE : 1971—431-997/281

NBS TECHNICAL PUBLICATIONS

PERIODICALS

JOURNAL OF RESEARCH reports National Bureau of Standards research and development in physics, mathematics, chemistry, and engineering. Comprehensive scientific papers give complete details of the work, including laboratory data, experimental procedures, and theoretical and mathematical analyses. Illustrated with photographs, drawings, and charts.

Published in three sections, available separately:

• Physics and Chemistry

Papers of interest primarily to scientists working in these fields. This section covers a broad range of physical and chemical research, with major emphasis on standards of physical measurement, fundamental constants, and properties of matter. Issued six times a year. Annual subscription: Domestic, $9.50; foreign, $11.75*.

• Mathematical Sciences

Studies and compilations designed mainly for the mathematician and theoretical physicist. Topics in mathematical statistics, theory of experiment design, numerical analysis, theoretical physics and chemistry, logical design and programming of computers and computer systems. Short numerical tables. Issued quarterly. Annual subscription: Domestic, $5.00; foreign, $6.25*.

• Engineering and Instrumentation

Reporting results of interest chiefly to the engineer and the applied scientist. This section includes many of the new developments in instrumentation resulting from the Bureau's work in physical measurement, data processing, and development of test methods. It will also cover some of the work in acoustics, applied mechanics, building research, and cryogenic engineering. Issued quarterly. Annual subscription: Domestic, $5.00; foreign, $6.25*.

TECHNICAL NEWS BULLETIN

The best single source of information concerning the Bureau's research, developmental, cooperative and publication activities, this monthly publication is designed for the industry-oriented individual whose daily work involves intimate contact with science and technology—*for engineers, chemists, physicists, research managers, product-development managers, and company executives.* Annual subscription: Domestic, $3.00; foreign, $4.00*.

*** Difference in price is due to extra cost of foreign mailing.**

NONPERIODICALS

Applied Mathematics Series. Mathematical tables, manuals, and studies.

Building Science Series. Research results, test methods, and performance criteria of building materials, components, systems, and structures.

Handbooks. Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

Special Publications. Proceedings of NBS conferences, bibliographies, annual reports, wall charts, pamphlets, etc.

Monographs. Major contributions to the technical literature on various subjects related to the Bureau's scientific and technical activities.

National Standard Reference Data Series. NSRDS provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated.

Product Standards. Provide requirements for sizes, types, quality and methods for testing various industrial products. These standards are developed cooperatively with interested Government and industry groups and provide the basis for common understanding of product characteristics for both buyers and sellers. Their use is voluntary.

Technical Notes. This series consists of communications and reports (covering both other agency and NBS-sponsored work) of limited or transitory interest.

Federal Information Processing Standards Publications. This series is the official publication within the Federal Government for information on standards adopted and promulgated under the Public Law 89–306, and Bureau of the Budget Circular A–86 entitled, Standardization of Data Elements and Codes in Data Systems.

Consumer Information Series. Practical information, based on NBS research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide useful background knowledge for shopping in today's technological marketplace.

NBS Special Publication 305, Supplement 1, Publications of the NBS, 1968-1969. When ordering, include Catalog No. C13.10:305. Price $4.50; foreign, $5.75.

Order NBS publications from: Superintendent of Documents
Government Printing Office
Washington, D.C. 20402

U.S. DEPARTMENT OF COMMERCE
WASHINGTON, D.C. 20230

OFFICIAL BUSINESS

PENALTY FOR PRIVATE USE, $300

POSTAGE AND FEES PAID
U.S. DEPARTMENT OF COMMERCE