Separation of Zeros of the Riemann Zeta-Function*

By R. Sherman Lehman

1. Introduction. The Riemann zeta-function $\zeta(s)$ is the analytic function of $s = \sigma + it$ defined by the formula

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}$$

for $\sigma > 1$. It was conjectured by Riemann that all of the zeros of $\zeta(s)$, other than the zeros at the negative even integers, lie on the line $\sigma = \frac{1}{2}$. Extensive verifications of the Riemann hypothesis using high-speed computers have been made by Lehmer [1], [2], who established that the hypothesis holds for the first 25000 zeros in the upper half plane, and by Meller [3], who showed that it holds for the first 35337 zeros. In this paper we describe computations made with an IBM 7090 at the Computer Center of the University of California at Berkeley which establish that there are exactly 250000 zeros of $\zeta(s)$ for which $0 < t < 170571.35$, all of which lie on the line $\sigma = \frac{1}{2}$ and are simple. The major part of the calculation was done using a program which separates zeros of $\zeta(\frac{1}{2} + it)$ automatically in most cases. This program used the Riemann-Siegel asymptotic formula with only the first term of the asymptotic expansion retained so that the error bound derived by Titchmarsh [4] could be used. An ALGOL version of the program is given in Section 7.

2. Error Analysis. In formulas for which numerical bounds are of importance we shall use ϑ to denote a number satisfying $|\vartheta| \leqq 1$. The number denoted will, in general, be different for different occurrences.

If

$$\kappa(\tau) = (1/2\pi) \operatorname{Im} \{\log \Gamma(\tfrac{1}{4} + \pi i \tau)\} - \tfrac{1}{2}\tau \log \pi, \tag{2.1}$$

then the function

$$Z(\tau) = \exp(2\pi i \kappa(\tau))\zeta(\tfrac{1}{2} + 2\pi i \tau) \tag{2.2}$$

is real for real τ (see [4, p. 235]). It can be shown [4, p. 247] that for $\tau \geqq 8$

$$\kappa(\tau) = \tfrac{1}{2}(\tau \log \tau - \tau - \tfrac{1}{8}) + \vartheta \cdot 0.0006\, \tau^{-1}. \tag{2.3}$$

We remark that our notation follows that of Turing [5]. It differs from that of Haselgrove [6] who uses t rather than τ as the argument. Consequently, Haselgrove's $Z(t)$ is denoted by $Z(t/2\pi)$ in our notation.

Theorem 2 of [4], when some minor numerical errors are corrected, states that for $\tau \geqq 8$

$$Z(\tau) = 2 \sum_{n=1}^{m} \frac{\cos 2\pi(\tau \log n - \kappa(\tau))}{n^{1/2}} + (-1)^{m-1}\tau^{-1/4}h(\xi) \exp\{-i/96\pi\tau + \omega_6 + i\omega_7\} + R \tag{2.4}$$

Received March 11, 1965.

* This research was supported in part by the Air Force Office of Scientific Research under Contract No. AF-AFOSR 553-64.

where

$$m = [\tau^{1/2}], \qquad \xi = \tau^{1/2} - m, \qquad |\omega_6| = \frac{7}{160\pi^2\tau^2}, \qquad |\omega_7| < \frac{1}{30\pi^2\tau^2},$$

$$h(\xi) = \frac{\cos 2\pi(\xi^2 - \xi - 1/16)}{\cos 2\pi\xi}$$

and the following inequality holds:

$$|R| < \left(\frac{0.4652}{1 - 0.813\tau^{-1/3}} + \frac{0.4168}{1 - 0.489\tau^{-1/2}}\right)\tau^{-3/4} + \frac{0.969}{\tau^{5/12}}\,10^{-0.4\tau^{1/3}}$$
$$+ \frac{0.456}{\tau^{5/12}}\,10^{-0.26\tau^{1/3}} + \frac{0.309}{\tau^{3/4}}\,10^{-0.45\tau} + \frac{0.655}{\tau^{3/4}}\,10^{-0.77\tau} + 0.065\;10^{-3\tau}.$$

For $\tau \geqq 1500$ we obtain by a straightforward numerical calculation using $|h(\xi)| < 1$ (which results from (2.6) and (2.7) below)

$$Z(\tau) = 2\sum_{n=1}^{m} \frac{\cos 2\pi(\tau \log n - \kappa(\tau))}{n^{1/2}}$$
$$+ (-1)^{m-1}\tau^{-1/4}h(\xi) + \vartheta\cdot 0.929\tau^{-3/4}.$$

Turing [5, p. 103] has shown that this formula holds with a slightly increased error if $m = [\tau^{1/2}] - 1$ and $\xi = \tau^{1/2} - m$ has a value slightly greater than 1. If $1 \leqq \xi < 2$ then it suffices to increase the error term by

$$4\pi(m+1)^{-1/2}\left\{\frac{(\xi-1)^3}{3(m+1)} + \frac{0.0006}{(m+1)^2}\right\} + 2\left\{\frac{1}{(m+1)^{1/2}} - \frac{1}{(m+\xi)^{1/2}}\right\}.$$

Consequently

$$Z(\tau) = 2\sum_{n=1}^{m} \frac{\cos 2\pi(\tau \log n - \kappa(\tau))}{n^{1/2}} \tag{2.5}$$
$$+ (-1)^{m-1}\tau^{-1/4}h(\xi) + \vartheta\cdot 0.93\tau^{-3/4}$$

if $\tau \geqq 1500$ and m is an integer such that

$$0 \leqq \xi = \tau^{1/2} - m < 1.0005.$$

For numerical purposes it is convenient to replace $h(\xi)$ by a polynomial approximation. Turing [5] has given a quadratic approximation with a rigorously determined error term, but we shall need a more accurate approximation. Let

$$\phi(z) = h\left(\frac{1-z}{2}\right) = \frac{\cos \pi(z^2/2 + 3/8)}{\cos \pi z}. \tag{2.6}$$

Since $\phi(z)$ is an even entire function, we have

$$\phi(z) = \sum_{n=0}^{\infty} c_n z^n$$

with $c_{2k+1} = 0$ $(k = 0, 1, 2, \cdots)$. We obtain a polynomial approximation to $\phi(z)$ by truncating the power series and estimating the remainder.

We have

$$c_n = \frac{1}{2\pi i}\int_C \frac{\phi(z)}{z^{n+1}}\,dz$$

where C is a positively oriented contour about the origin, which we shall take to be the rectangle with corners at $\pm 2 \pm 2i$. If $z = x + iy$ with x and y real, then

$$| \sinh \pi y | \leqq | \cos \pi z | \leqq \cosh \pi y$$

and

$$\operatorname{Im} (z^2/2 + 3/8) = xy.$$

Also, when x is an integer $| \cos \pi(x + iy) | = \cosh \pi y$. Hence on the vertical line segments of the contour

$$| \phi(z) | \leqq \frac{\cosh \pi xy}{\cosh \pi y} = \frac{\cosh 2\pi y}{\cosh \pi y} \leqq \frac{\cosh 4\pi}{\cosh 2\pi} < \frac{\cosh 4\pi}{\sinh 2\pi}$$

while on the horizontal line segments

$$| \phi(z) | \leqq \frac{\cosh \pi xy}{| \sinh \pi y |} = \frac{\cosh 2\pi x}{\sinh 2\pi} \leqq \frac{\cosh 4\pi}{\sinh 2\pi}.$$

Thus

$$| c_n | \leqq \frac{16 \cosh 4\pi}{2\pi \cdot 2^{n+1} \sinh 2\pi} < \frac{700}{2^n}.$$

The remainder when the series is truncated after the term $c_n z^n$ is thus less than

$$\frac{700}{2^n} \cdot \frac{| z |/2}{1 - | z |/2}.$$

If $\xi = (1 - z)/2$, then $| z | < 1.001$ when $0 < \xi < 1.0005$. It then follows that the error made by omitting terms after $c_{24} z^{24}$ is less than $5 \cdot 10^{-5}$. We can estimate the error committed by omitting additional terms of the series and by rounding the coefficients to 5 decimal places by using the values of the coefficients c_n given by Lehmer [2] and Miller in [6]. We find

$$\begin{aligned} \phi(z) &= 0.38268 + 0.43724 z^2 + 0.13238 z^4 \\ &\quad - 0.01361 z^6 - 0.01357 z^8 - 0.00162 z^{10} + \vartheta \cdot 4.9 \cdot 10^{-4} \end{aligned} \tag{2.7}$$

provided $| z | < 1.001$.

Now let us estimate the error made when computing a value of $Z(\tau)$ by means of a digital computer using (2.3), (2.5) and (2.7). First, we shall carry out our analysis without considering in detail the particular properties of the machine used, and then we shall specialize our discussion to two methods used to compute $Z(\tau)$ with an IBM 7090.

We assume that τ is given exactly as a digital number. Let ϵ_1 be a bound for the absolute error in computing $\log n$, ϵ_2 a bound for the absolute error in computing $\log \tau$, ϵ_3 a bound for the absolute error in computing $\cos 2\pi x$ when x is given exactly as a digital number between 0 and 1, and ϵ_4 a bound for the absolute error in computing $n^{-1/2}$. The bounds ϵ_1, ϵ_2, ϵ_3, ϵ_4 and also ϵ_5, ϵ_6, and ϵ_7 which occur in the following argument are all assumed to be constants independent of n and x. In order to simplify the error analysis we shall assume that the sum in (2.5) is accumulated using fixed-point arithmetic. We also assume that $\tau^{1/2}$ is computed sufficiently accurately so that an integer m can be determined for which $0 \leqq \tau^{1/2} - m < 1.0005$

so that (2.5) and (2.7) are valid. (Actually, it should be possible to assure that $0 \leqq \tau^{1/2} - m < 1$ but in our computation a programming error prevented this.)

First, from (2.3) we compute $\kappa(\tau)$ and then obtain $\tau \log n - \kappa(\tau)$ with an error less than

$$\epsilon_1\tau + \tfrac{1}{2}\epsilon_2\tau + 0.0006\tau^{-1} + \epsilon_5$$

where ϵ_5 is a bound for the additional error due to rounding in performing the arithmetical operations. We then obtain $\cos 2\pi(\tau \log n - \kappa(\tau))$ with an error less than

$$2\pi\epsilon_1\tau + \pi\epsilon_2\tau + 0.0012\pi\tau^{-1} + 2\pi\epsilon_5 + \epsilon_3 .$$

Each term in the sum in (2.5) is then obtained with an error less than

$$(2\pi\epsilon_1\tau + \pi\epsilon_2\tau + 0.0012\pi\tau^{-1} + 2\pi\epsilon_5 + \epsilon_3)(n^{-1/2} + \epsilon_4) + \epsilon_4 + \epsilon_6$$

where ϵ_6 is a bound for the rounding error in the multiplication. Since the sum is accumulated using fixed-point arithmetic, a bound for the error in the sum can be obtained by summing these bounds. We have

$$2 \sum_{n=1}^{m} (n^{-1/2} + \epsilon_4) < 2\epsilon_4 m + 4m^{1/2} < 4.02\tau^{1/4},$$

provided $\epsilon_4 < 0.01m^{-1/2}$ as we shall assume. Finally, let $(4.9 \cdot 10^{-4} + \epsilon_7)\tau^{-1/4}$ be a bound for the absolute error in computing $\tau^{-1/4}h(\xi)$ using a digital approximation for the polynomial in (2.7). Combining the error estimates, we see that $Z(\tau)$ is computed with an error less than

$$\begin{aligned}(2.8)\quad & (26\epsilon_1 + 13\epsilon_2)\tau^{5/4} + (2\epsilon_4 + 2\epsilon_6)\tau^{1/2}(4.1\epsilon_3 + 26\epsilon_5)\tau^{1/4} \\ & \qquad + (4.9 \cdot 10^{-4} + \epsilon_7)\tau^{-1/4} + 0.95\tau^{-3/4}.\end{aligned}$$

In our computation with the IBM 7090 we used two different methods for computing $Z(\tau)$. The first was designed for maximum speed and in most cases was sufficiently accurate to determine the sign of $Z(\tau)$. Whenever that was not the case, a slower but more accurate method was used. In the following discussion we shall assume that $1500 < \tau < 100000$.

We first sketch how error bounds for the slower method were obtained. Values of $\log \tau$ were obtained by means of a standard double-precision subroutine. Because of complications in the error analysis of the double-precision subroutines, the bound $\epsilon_2 = 5 \cdot 10^{-14}$ which we obtained is probably quite conservative. The same subroutine was also used to evaluate $\log n$, but since in this case only 316 values are required, a comparison with a 16 decimal place table [7] was possible. In this way it was shown that $\epsilon_1 = 3 \cdot 10^{-15}$ is permissible.

The cosines were computed by means of a single-precision fixed-point subroutine programmed by the author for which the error was proved to be less than $\epsilon_3 = 2 \cdot 10^{-10}$. The values of $n^{-1/2}$ were obtained by means of a double-precision square root subroutine and then rounded to 27 binary places. Thus $\epsilon_4 = 3.8 \cdot 10^{-9}$ is permissible. For bounds on the rounding errors we found that we could take $\epsilon_5 = 2^{-35}$, $\epsilon_6 = 2^{-28}$ and $\epsilon_7 = 10^{-5}$. Substituting these values into (2.8), we obtain the following error bound for the more accurate method:

$$(2.9)\quad 7.3 \cdot 10^{-13}\tau^{5/4} + 1.52 \cdot 10^{-8}\tau^{1/2} + 1.6 \cdot 10^{-9}\tau^{1/4} + 0.0005\tau^{-1/4} + 0.95\tau^{-3/4}.$$

In the faster but less accurate method a table-look-up scheme was used to compute cosines with the table occupying half of the memory of the computer. The table of $\cos 2\pi x$ for $0 \leqq x \leqq 1$ contained 2^{14} entries which were generated by the previously mentioned subroutine. Since the value at the nearest tabulated point was used, the error in computing a cosine could not exceed $2\pi \cdot 2^{-15} + 2 \cdot 10^{-10}$. Hence we let $\epsilon_3 = 1.92 \cdot 10^{-4}$. The values of $\log n$ were obtained from those used in the more accurate method by rounding to 32 binary places. Thus we let $\epsilon_1 = 1.2 \cdot 10^{-10}$. The parts of the computation concerned with $\log \tau$, $n^{-1/2}$ and $\phi(z)$ were the same as for the more accurate method. Hence we can take ϵ_2, ϵ_4, ϵ_5, ϵ_6 and ϵ_7 the same as before. Substituting into (2.8), we obtain for the faster method the error bound

$$(2.10) \quad 3.2 \cdot 10^{-9} \tau^{5/4} + 1.52 \cdot 10^{-8} \tau^{1/2} + 8 \cdot 10^{-4} \tau^{1/4} + 0.0005 \tau^{-1/4} + 0.95 \tau^{-3/4}.$$

3. The Verification Program. The method that we use for verifying the Riemann hypothesis for $0 < t \leqq T$ is the following: By counting sign changes of $Z(\tau)$ for $0 < \tau < T/2\pi$ we obtain a lower bound for the number of zeros of $\zeta(s)$ on the line $\sigma = \frac{1}{2}$ for which $0 < t < T$. We then show that this lower bound is equal to $N(T)$, the total number of zeros of $\zeta(s)$ for which $0 < \sigma < 1$, $0 < t \leqq T$. In this section we shall explain the design of the program for separating zeros. The method for determining $N(T)$ will be discussed in Section 5.

It is known [8, p. 179] that

$$(3.1) \qquad N(T) = 2\kappa(T/2\pi) + 1 + S(T),$$

where, if T is not the ordinate of a zero of $\zeta(s)$,

$$(3.2) \qquad S(T) = (1/\pi) \arg \zeta(\tfrac{1}{2} + iT)$$

with the value of the argument obtained by continuous variation along the line from $\infty + iT$ to $\frac{1}{2} + iT$ starting with the value 0.

If n is a nonnegative integer, then let τ_n be the positive real number for which $2\kappa(\tau_n) = n$. These points τ_n are called the Gram points. The interval (τ_n, τ_{n+1}), which we also denote by I_n, is called the nth Gram interval. The statement, that there is exactly one zero of $\zeta(\frac{1}{2} + 2\pi i\tau)$ in each Gram interval so that the nth positive zero of $\zeta(\frac{1}{2} + 2\pi i\tau)$ is in I_{n-2}, is known as Gram's law. This "law" holds when $n < 127$ but fails for $n = 127$. It continues to hold in most cases as far as calculations have been performed. Consequently, in designing a program to separate zeros of $Z(\tau)$ it is advantageous to evaluate $Z(\tau)$ at points g_n which are approximations to the Gram points τ_n.

It is known (see [8, p. 189] or Section 5 below) that

$$\int_0^T S(t)\, dt = O(\log T).$$

Using this fact, one can prove without difficulty that the number of successive integers n for which $0 < 2\pi\tau_n < T$ and $N(2\pi\tau_n) \neq n + 1$ is $O(\log^2 T)$. Thus we expect a one-to-one correspondence between zeros and Gram intervals with the zero corresponding to an interval located either in the interval or in a nearby Gram interval. Consequently the verification program was designed so that when a failure of Gram's law occurred, an attempt was made to find enough zeros nearby to maintain the one-to-one correspondence.

We shall say that a failure of Gram's law of type $\nu_0\nu_1 \cdots \nu_l$ occurs at n if the following three conditions are satisfied:

(i) There are exactly ν_k zeros in the Gram interval I_{n+k} for $k = 0, 1, \cdots, l$;

(ii) $\nu_0 + \nu_1 + \cdots + \nu_l = l + 1$;

(iii) $\nu_0 + \nu_1 + \cdots + \nu_k \neq k + 1$ for $0 \leqq k < l$.

For example, the first failure of Gram's law occurs at 125 and is of type 0 2 because there are no zeros in I_{125} and two zeros in I_{126}.

Before writing a program for verifying the Riemann hypothesis, we had constructed a table of the first 10000 zeros of $\zeta(s)$ giving the ordinates γ_n with about 7 decimal place accuracy and the corresponding values of $2\kappa(\gamma_n/2\pi)$ to 4 decimals. This table confirmed the result of Lehmer [1] that for the first 10000 Gram intervals all failures of Gram's law are of the following five types:

$$0\,2, \quad 2\,0, \quad 0\,3\,0, \quad 2\,1\,0, \quad 0\,1\,2.$$

We note, however, that we found one more failure of each of the last two types than Lehmer did. A failure of type 2 1 0 occurs at 9807, and one of type 0 1 2 occurs at 9971. Also a failure of type 2 1 0 occurs at 6412 rather than at 6411 as reported in [1].

We designed our verification program to handle automatically only failures of Gram's law which are of the above types and are such that the zeros are far enough apart to be separated by a calculation based on (2.5). For all other cases the machine was instructed to print a report of the failure and then continue the computation.

In Section 7 we give the complete text of our verification program in ALGOL. The program is arranged so that the major part of the computation is performed by the three procedures *sign Z*, *Gram*, and *two zeros found*. We shall first describe these procedures briefly and then sketch how they are used in the verification program.

The procedure *sign Z* is used to determine the sign of $Z(\tau)$. It returns the value $+1$ if $Z(\tau)$ is shown to be positive, -1 if it is shown to be negative, and 0 if the procedure is unable to determine the sign. $Z(\tau)$ is first computed using the faster, less accurate method of Section 2, and then the computation is repeated with the more accurate method if the first one is unable to determine the sign with certainty. In order to leave some margin for safety the procedure was constructed to return the value 0 if the value computed was not greater in absolute value than twice the error bound obtained in Section 2. The mathematics concerned with error analysis for computations seems to be especially liable to error. If a program is run with an error bound which is later found to be incorrect, then the entire computation will be wasted. (Such a misfortune was reported in [5].) We chose the above device to give some protection against this possibility. It should perhaps be noted here that there is a second order error in the computation due to errors in computing the error bound. These errors are all quite small and in fact it is easy to show that the error bounds (2.9) and (2.10) are large enough to cover these additional errors.

The procedure *Gram* is used to obtain points g_n which are approximations to the Gram points τ_n. If we consider the transformation $\phi(\tau) = (n + \frac{1}{8})/(\log \tau - 1)$, then it is easily shown that for $n \geqq 7$ the iterates defined by $\tau_n^{(k+1)} = \phi(\tau_n^{(k)})$

($k = 0, 1, 2, \cdots$) converge to the fixed point τ_n^* of the transformation provided $\tau_n^{(0)}$ is sufficiently near τ_n^*. Also, by (2.3) we have $\tau_n = \tau_n^* + O(1/n)$. For $n \geqq 30$ it is also easy to verify that $0 < \phi'(\tau) < K$ for $\frac{1}{2}\tau_n^* \leqq \tau < \infty$ where K is a constant less than 1. Consequently, if $\tau_n^{(0)}$ is chosen in this interval, then the iterates will converge to τ_n^* and the absolute value of the difference between successive iterates will decrease. In the procedure *Gram* this iteration scheme is employed with g_n used as an initial approximation to g_{n+1}. The iteration is discontinued when because of round-off error the distance between successive iterates ceases to decrease. An exact error analysis of this procedure was not undertaken because all that is necessary for our verification is that it furnish a monotone increasing sequence of points g_n.

Given an interval (a, c) with $Z(a)$ and $Z(c)$ having the same sign, the procedure *two zeros found* is used to try to show that $Z(\tau)$ has at least two zeros in the interval (a, c). This is done by a bisection process which splits the interval into a maximum of 2^8 subintervals. At each stage a search is made for a point where $Z(\tau)$ has the opposite sign. If such a point is found, the bisection is discontinued and the value **true** is returned. If the bisection process fails to locate a point where $Z(\tau)$ has the opposite sign, then the value **false** is returned.

Now we shall sketch how these procedures are used. The strategy used by the verification program is to set up a one-to-one correspondence between approximate Gram intervals and located zeros. At each approximate Gram point g_n, the procedure *sign Z* is used to determine the sign of $Z(g_n)$. If the value 0 is returned, which means the sign is in doubt, then a new approximate Gram point is chosen nearby where the sign can be determined with certainty.

Because of the heuristic assumption that all failures of Gram's law are of the five types found in the first 10000 Gram intervals, the program proceeds with the assumption that each failure begins with a Gram interval where the sign of $Z(\tau)$ is the same at both end points. Consequently, as long as the sign of $Z(\tau)$ alternates at successive approximate Gram points, the one-to-one correspondence between Gram intervals and real zeros of $Z(\tau)$ is maintained; and the program proceeds ignoring the possible existence of additional unlocated zeros.

If at the points g_n, g_{n+1}, g_{n+2} the sign distribution is $+ + +$ or $- - -$, then the procedure *two zeros found* is used to try to show that there are two zeros in the interval (g_n, g_{n+2}) and thereby maintain the one-to-one correspondence. If at the points g_n, g_{n+1}, g_{n+2}, g_{n+3} the sign distribution is $+ + - -$ or $- - + +$, then the sign of $Z(\tau)$ is determined at $q = (g_{n+1} + g_{n+2})/2$ (or at a nearby point in exceptional cases). Then *two zeros found* is used to try to show that there are two zeros in (g_n, q) if $Z(\tau)$ has the same sign at g_n and q and two zeros in (q, g_{n+3}) in the other case.

In all cases where *two zeros found* fails to prove the existence of two zeros in a specified interval and also in all cases where the sign distribution does not follow one of the patterns described, a special report is made which indicates the type of failure. In each such case where the verification program has failed to maintain the one-to-one correspondence between zeros and Gram intervals, enough information is printed to determine how many zeros are missing and then the verification is resumed. The treatment of these exceptional cases will be discussed in Section 4.

In addition to reporting on exceptional cases the program keeps a count of failures of Gram's law. Instead of counting the failures directly, a count was kept of the number of displacements of zeros across Gram points required to obtain the actual distribution from one in which there is exactly one zero in each Gram interval. Thus the count of displacements was increased by one for each failure of type 0 2 or 2 0 and by two for each failure of type 0 3 0, 2 1 0 or 0 1 2. It should be noted, however, that this displacement count is not exact because in our computation instead of using the Gram points τ_n we use approximate Gram points g_n. For another machine, or even for the same machine with a different logarithm subroutine, the count of displacements could be expected to differ slightly.

In writing the verification program the innermost loop of the program, which occurs in the evaluation of the sum

$$\sum_{n=1}^{m} \frac{\cos 2\pi(\tau \log n - \kappa(\tau))}{n^{1/2}},$$

was first written directly in the machine code of the IBM 7090 with attention given to minimizing execution time. Next, the entire program was written in ALGOL and tested with the BC-ALGOL interpretive system, an implementation of ALGOL (developed at the University of California, Berkeley) which, although it is too slow for production running of large computations, is useful for testing programs. The procedure for determining the sign of $Z(\tau)$ was then translated into machine code using a symbolic assembly language. The remainder of the program, for which running speed was of less importance, was translated into FORTRAN IV and then compiled to obtain the final version of the program.

4. Results of Computations. Except for some runs which were made to test it the verification program was not used to investigate the first 10000 Gram in-intervals because they had already been studied in more detail earlier. The program was run for blocks of 1000 Gram intervals for $10000 \leqq n < 50000$ and for blocks of 5000 Gram intervals for $50000 \leqq n < 250000$. The results of these runs are summarized in Table 1 for sets of 10000 successive Gram intervals. The number of failures of Gram's law, as indicated by the column labeled "Displacements", shows only a slow increase within the range of the computation. The column labeled "Sign doubtful" records the number of times the procedure *sign Z* was unable to determine the sign of $Z(\tau)$ at an approximate Gram point. When this happened, a new approximate Gram point was chosen nearby. The comparatively large number of such cases at the top of the table occurs because the truncation bound $0.93\tau^{-3/4}$ of (2.5) is relatively large there.

The program encountered a total of 104 cases which it was not able to handle automatically. In each of these cases one or more reports were printed indicating the type of failure and the number of missing zeros. These cases were then examined with the aid of further programs. The last six columns of Table 1 give information about these cases.

The verification program will look beyond the last Gram interval of a block to find a missing zero, but it never looks before the first interval of a block. Thus, if $N(2\pi g_n) \neq n + 1$ at the beginning of a block, a report of a special case may occur even though the failure is of a type which ordinarily is handled automatically. The column labeled "Out of phase" records the number of these cases. Each of these

TABLE 1

Index of initial Gram interval	Dis-placements	Sign doubtful	Failure types					
			Out of phase	0 1 3 0	0 3 1 0	0 1 1 2	2 1 1 0	Miscellaneous
10000	1013	32		2				1
20000	1102	23	2	1				2
30000	1115	16	1		1			
40000	1161	13		1	1			
50000	1120	12		1				
60000	1142	11			1	1		1
70000	1198	12		2	2			1
80000	1204	10		1		2	1	2
90000	1178	15			1		1	1
100000	1211	8	1	1	1			1
110000	1192	8						
120000	1208	8	2			3	2	
130000	1237	11	1	2				
140000	1200	7	2	1	1	2	1	
150000	1262	4	1		1	1	2	
160000	1195	7	1	1	1	2	1	
170000	1265	5			1		1	1
180000	1223	11		1			1	
190000	1285	7	1	4	2			
200000	1264	1		1			3	2
210000	1277	7	1	2	2		2	
220000	1226	5		1	2	1		
230000	1260	6	1	1	2		3	2
240000	1259	2			1	2		1
Totals....	28797	241	14	23	20	14	18	15

14 cases was handled by running the verification program for a small block of 20 intervals overlapping the beginning of the larger block.

Of the 90 remaining cases 75 correspond to failures of types 0 1 3 0, 0 3 1 0, 0 1 1 2 and 2 1 1 0. We shall give detailed information on the 15 cases classified as "Miscellaneous" below. In all but 9 of these 90 cases the zeros could be separated by using values of $Z(t/2\pi)$ tabulated with steps of 0.1 in t. (In particular, all cases for which $t > 80000$ were handled in this way.) The other cases required a finer mesh.

In five cases failures occur in which the program fails to find three zeros in (g_n, g_{n+3}) even though the failures are of the types 0 3 0, 0 1 2 or 2 1 0. This happens because the program was not designed to handle automatically a case in which all three zeros are on one side of the point $(g_{n+1} + g_{n+2})/2$. These failures occur at 171382, 206715, 209783, 233173, and 234500. In one other case a failure of type 2 1 0 occurred which was not handled automatically. This failure actually occurs at 25094, but a report was made for 25093 because of the presence of a zero near enough to the right end point of I_{25093} so that the sign was indeterminate at g_{25094}. A new approximate Gram point was then chosen to the left of the zero in I_{25093}.

A failure of type 0 3 1 1 0 occurs at 243021, and at 68084 a failure occurs which is either of type 0 1 1 2 or 0 1 1 3 0.

In five cases there were pairs of zeros which could be separated by the verification program if the error bound (2.9) was used but not if twice the error bound was used. These close pairs are located in the Gram interval I_n for $n = 24198, 73996, 82551, 87759, 106071$.

Finally, there were two cases where the formula (2.5) with the error bound (2.9) did not have enough accuracy to separate a close pair of zeros. In order to handle these cases we constructed a subroutine to compute $Z(\tau)$ with an absolute error of less than 10^{-6} for $10 < \tau < 40000$ which used the Euler-Maclaurin sum formula (see [2, pp. 102–103]) together with the formulas

$$| Z(\tau) |^2 = \{\operatorname{Re} \zeta(\tfrac{1}{2} + 2\pi i\tau)\}^2 + \{\operatorname{Im} \zeta(\tfrac{1}{2} + 2\pi i\tau)\}^2,$$

$$\operatorname{sgn} Z(\tau) = (-1)^{n-1} \operatorname{sgn} \{\operatorname{Im} \zeta(\tfrac{1}{2} + 2\pi i\tau)\} \quad \text{for} \quad \tau \text{ in } I_n .$$

In each case it was sufficient to calculate $Z(\tau)$ at one point to establish that a sign change occurs. The first of these close pairs, which is located in I_{18858}, has been reported by Lehmer [2] and Meller [3]. The second pair is probably the closest pair among the first 250000 zeros. The zeros are located at about

$$\tfrac{1}{2} + 71732.9012i \quad \text{and} \quad \tfrac{1}{2} + 71732.9159i.$$

The maximum value of $| \zeta(s) |$ on the line $\sigma = \frac{1}{2}$ between these two zeros is about 0.0005. The first zero is located in I_{95246} while the second is in I_{95247}, and these are the only zeros in these Gram intervals. Thus, unlike the close pairs which have been noticed previously, there is no failure of Gram's law in connection with this pair.

The Euler-Maclaurin formula was also used to separate the close pairs of zeros in I_{82551}, I_{87759} and I_{106071}, giving an additional check of the computation in these cases.

The total machine time used in production runs was about 125 minutes. Of this, 100 minutes were used in running the main verification program while about 24 minutes were used in handling the 90 special cases discussed above. Less than one minute was used in carrying out the computation described in Section 5 to complete the verification. The main verification program was still quite efficient at the upper end of the range of the computation. For the 25000 Gram intervals beginning with the 225000th the running time required was 12.42 minutes, so that about 33 zeros were separated per second. On the other hand an evaluation of $Z(t/2\pi)$ by means of the Euler-Maclaurin formula for t of the order of magnitude 70000 required about one minute of machine time.

5. Completion of the Verification. In order to gather additional information, a modified version of the verification program was run for $249900 \leqq n \leqq 250100$ with information printed out each time $Z(\tau)$ was evaluated. This information included the value of $\kappa(\tau)$ at each of the points. Taking into account the computational error, we established in this way that for the approximate Gram points g_n

$$(5.1) \qquad | 2\kappa(g_n) - n | < 0.01 \qquad (249900 \leqq n \leqq 250100).$$

We shall now describe how the information from this run was used to complete the verification that if

$$T_0 = 2\pi g_{250000} = 170571.358 \cdots ,$$

there are exactly 250000 zeros of $\zeta(s)$ for which $0 < t < T_0$, all of which lie on the line $\sigma = \frac{1}{2}$ and are simple. That there are at least 250000 zeros of $\zeta(s)$ for which $\sigma = \frac{1}{2}, 0 < t < T_0$ follows from the results of Section 4 together with a small amount of information about the zeros found in the last few approximate Gram intervals before g_{250000}. To complete the verification of the Riemann hypothesis for $0 < t \leqq T_0$ we shall prove that $N(T_0)$, the total number of zeros in the rectangle $0 < \sigma < 1$, $0 < t \leqq T_0$, is equal to 250000. The method that we use is due to Turing [5]. Theorem 4 in [5] would be adequate for our purposes, but, unfortunately, although the theorem is correct, the proof given in [5] contains several mistakes. Consequently, we shall use the following weaker theorem which will be proved in Section 6.

THEOREM 1. *If*

$$S_1(T) = \int_0^T S(t)\, dt \tag{5.2}$$

and $T_2 > T_1 > 100$, *then*

$$| S_1(T_2) - S_1(T_1) | < 3.1 \log T_2 + 4.8.$$

We obtained the following distribution of zeros in the approximate Gram intervals starting with the interval with left end point g_{250000}:

$$\underline{2}, 1, 1, 1, 1, 1, 1, 1, 1, 1; \quad 1, 1, 1, 1, 1, 0, \underline{2}, 1, 1, 1; \quad 1, 1, 1, 2, \bar{0}, 1, 1, 1, 1, 1;$$

$$1, 1, 1, 1, 1, 1, 1, 1, 2, \bar{0}; \quad 1, 1, 1, 1, 1, 1, 0, \underline{1}, \underline{2}, 1; \quad 1, 1, 1, 1, 1, 2, \bar{1}, \bar{0}, 0, \underline{2};$$

$$1, 1, 0, \underline{2}, 0, \underline{2}, 1, 1, 1, 2; \quad \bar{0}, 1, 1, 1, 1, 1, 1, 1, 1, 1; \quad 1, 1, 1, 1, 1, 1, 2, \bar{0}, 1, 1;$$

$$0, \underline{2}, 1, 1, 1, 2, \bar{0}, 1, 1, 1.$$

Thus, in these 100 approximate Gram intervals we find 101 zeros. (The extra zero in the first interval is compensated for by the absence of a zero in the previous interval.)

Let

$$N(2\pi g_{250000}) = 250000 + q$$

where q is a nonnegative integer. The integer q is even because complex zeros or unseparated real zeros of $Z(\tau)$ must occur in pairs. We prove that $q = 0$ by showing that $q < 2$. We apply the equation

$$S(2\pi\tau) = N(2\pi\tau) - 1 - 2\kappa(\tau)$$

at $\tau = g_n$ for $250000 \leqq n \leqq 250099$. At 8 points (corresponding to the underlined numbers in the above listing), the first of which is g_{250000}, we have

$$S(2\pi g_n) > q - 1 - 0.01; \tag{5.3}$$

at 7 points (corresponding to the overlined numbers)

$$S(2\pi g_n) > q + 1 - 0.01; \tag{5.4}$$

and at the remaining 85 points

$$S(2\pi g_n) > q - 0.01. \tag{5.5}$$

Also by (5.1) $S(2\pi\tau)$ cannot decrease by more than 1.02 in an approximate Gram interval. Consequently, if $T_1 = 2\pi g_{250000}$ and $T_2 = 2\pi g_{250100}$, then

$$\int_{T_1}^{T_2} S(t)\,dt = 2\pi \int_{T_1/2\pi}^{T_2/2\pi} S(2\pi\tau)\,d\tau > 2\pi(q - 1.03)(g_{250100} - g_{250000}) + 2\pi(L_7 - L_8)$$

where L_7 is the total length of the 7 intervals (g_n, g_{n+1}) for which we obtained (5.4) and L_8 is the total length of the 8 intervals for which we obtained (5.3). The approximate Gram intervals all have lengths between 0.096 and 0.099 and have total length greater than 9.79. Hence, if $q \geqq 2$,

$$\int_{T_1}^{T_2} S(t)\,dt > 2\pi\{(0.97)(9.79) + (0.672 - 0.792)\} > 58,$$

while on the other hand by Theorem 1

$$\int_{T_1}^{T_2} S(t)\,dt < 43,$$

a contradiction. Therefore, $q = 0$ and the first 250000 zeros of $\zeta(s)$ for which $0 < t < 170571.35$ are on the line $\sigma = \frac{1}{2}$.

6. Proof of Theorem 1. For the proof of Theorem 1 we need several lemmas.

LEMMA 1. *If* $T_2 > T_1 > 0$, *then*

$$\pi\{S_1(T_2) - S_1(T_1)\} = \int_{1/2+iT_2}^{\infty+iT_2} \log|\zeta(s)|\,ds - \int_{1/2+iT_1}^{\infty+iT_1} \log|\zeta(s)|\,ds.$$

Proof. To define $\log \zeta(s)$ uniquely we consider the branch of the logarithm which is real for $s > 1$ and is obtained by analytic continuation in the complex plane cut from $-\infty$ to 1 along the real axis and cut along the line from $-\infty + i\gamma$ to $\rho = \beta + i\gamma$ for each complex zero ρ of $\zeta(s)$. In the cut plane we consider the half-strip S for which $\sigma > \frac{1}{2}$, $T_1 < t < T_2$. Then for each zero ρ for which $\sigma \geqq \frac{1}{2}$, $T_1 \leqq t \leqq T_2$ we delete from S any points which lie in a closed disk of radius ϵ about ρ, where ϵ is a small positive number. It is easily seen that if ϵ is sufficiently small, then the disks are disjoint and the remaining domain D_ϵ is simply connected.

We apply Cauchy's theorem integrating $\log \zeta(s)$ over the boundary of D_ϵ and then let $\epsilon \to 0$. For each small ϵ the integral vanishes because $\log \zeta(s)$ has no singularities in D_ϵ and is exponentially small as $\sigma \to \infty$. As $\epsilon \to 0$ the contributions to the integral of portions of the boundary which are circular arcs all approach 0 because $\log \zeta(s)$ has only logarithmic singularities at the zeros ρ. In integrating over portions of the boundary along a cut the direction of integration is opposite for the two banks of the cut. Since on opposite banks the values of $\log \zeta(s)$ have the same real part, the total contribution to the real part of the integral from the horizontal segments other than those along $t = T_1$ and $t = T_2$ is zero. Applying (3.2) we obtain the lemma.

LEMMA 2. *If* $\sigma > 1$ *then* $|\log|\zeta(s)|| < \log \sigma/(\sigma - 1)$.

Proof. We have

$$|\zeta(s)| = \left|\sum_{n=1}^{\infty}\frac{1}{n^s}\right| \leqq \sum_{n=1}^{\infty}\frac{1}{n^\sigma} < 1 + \int_1^\infty \frac{dx}{x^\sigma} = \frac{\sigma}{\sigma - 1}$$

and

$$\left|\frac{1}{\zeta(s)}\right| = \left|\sum_{n=1}^{\infty}\frac{\mu(n)}{n^s}\right| \leqq \sum_{n=1}^{\infty}\frac{|\mu(n)|}{n^\sigma} < \sum_{n=1}^{\infty}\frac{1}{n^\sigma} < \frac{\sigma}{\sigma - 1}.$$

LEMMA 3. *If* $\sigma \geqq -\frac{3}{2}$ *and* $t \geqq 2\pi$ *then* $|\zeta(s)| < \frac{1}{4}t^2$.

Proof. Applying the formula [8, p. 14]

$$\zeta(s) = \frac{1}{s - 1} + \frac{1}{2} + s\int_1^\infty \frac{[x] - x + \frac{1}{2}}{x^{s+1}}\,dx$$

which is valid for $\sigma > 0$, we find that for $\sigma \geqq \frac{1}{2}$, $t > 0$

$$|\zeta(s)| \leqq \frac{1}{t} + \frac{1}{2} + \frac{|s|}{2}\int_1^\infty \frac{dx}{x^{\sigma+1}} = \frac{1}{t} + \frac{1}{2} + \frac{1}{2}\left(1 + \frac{t^2}{\sigma^2}\right)^{1/2} < \frac{1}{t} + \frac{1}{2} + t\left(1 + \frac{1}{8t^2}\right)$$

and hence for $\sigma \geqq \frac{1}{2}$, $t \geqq 2\pi$

$$|\zeta(s)| < t + 1 < \tfrac{1}{4}t^2, \tag{6.1}$$

which establishes the lemma for $\sigma \geqq \frac{1}{2}$.

We estimate $\zeta(s)$ for $-\frac{3}{2} \leqq \sigma < \frac{1}{2}$ by applying the functional equation

$$\zeta(1 - s) = \frac{2}{(2\pi)^s}\Gamma(s)\cos\frac{1}{2}\pi s\,\zeta(s).$$

For $\sigma > 0$, $t > 0$

$$\log|\Gamma(\sigma + it)| \leqq -\tfrac{1}{2}\pi t + (\sigma - \tfrac{1}{2})\log t + \tfrac{1}{2}\log(2\pi) + \omega_3$$

where [4, p. 237]

$$|\omega_3| \leqq \frac{(\sigma - \frac{1}{2})\sigma^2}{2t^2} + \frac{1}{12|s|} + \frac{1}{720\,\sigma t^2}.$$

Hence, for $\frac{1}{2} \leqq \sigma \leqq \frac{5}{2}$, $t \geqq 2\pi$

$$|\omega_3| \leqq \frac{25}{4t^2} + \frac{1}{12t} + \frac{1}{360t^2} < 0.172 < \log 1.19,$$

and therefore

$$\left|\frac{\Gamma(s)}{(2\pi)^s}\right| \leqq 1.19e^{-\pi t/2}\left(\frac{t}{2\pi}\right)^{\sigma-1/2}.$$

Also, for $t \geqq 2\pi$

$$|\cos\tfrac{1}{2}\pi s| \leqq \tfrac{1}{2}(e^{\pi t/2} + e^{-\pi t/2}) \leqq \tfrac{1}{2}(1.001)e^{\pi t/2}.$$

Thus for $\frac{1}{2} \leqq \sigma \leqq \frac{5}{2}$, $t \geqq 2\pi$

$$|\zeta(1 - \sigma + it)| = |\zeta(1 - s)| \leqq 1.2(t/2\pi)^{\sigma-1/2}|\zeta(s)|. \tag{6.2}$$

Hence for $\frac{1}{2} \leqq \sigma \leqq \frac{3}{2}$, $t \geqq 2\pi$ we obtain by (6.1)

$$|\zeta(1 - \sigma + it)| \leqq 1.2\left(\frac{t}{2\pi}\right)(t + 1) < \frac{1}{4}t^2;$$

and for $\frac{3}{2} \leqq \sigma \leqq \frac{5}{2}$, $t \geqq 2\pi$ we obtain by Lemma 2

$$|\zeta(1 - \sigma + it)| \leqq 1.2\left(\frac{t}{2\pi}\right)^2 \frac{\sigma}{\sigma - 1} \leqq 3.6\left(\frac{t}{2\pi}\right)^2 < \frac{1}{4}t^2,$$

completing the proof of the lemma.

LEMMA 4. *Let $f(s)$ be an analytic function regular for $|s - s_0| \leqq R'$ such that $f(s_0) \neq 0$ and such that $|f(s)/f(s_0)| \leqq M$ on $|s - s_0| = R'$. Let $s_1, s_2, \cdots, s_n$ be the zeros of $f(s)$ in the circle $|s - s_0| \leqq R$ where $0 < R < R'$ with multiple zeros repeated. Then if $|s - s_0| \leqq r < R$,*

$$\left| \log|f(s)| - \left\{ \log|f(s_0)| + \sum_{k=1}^{n} \log\left|\frac{s - s_k}{s_0 - s_k}\right| \right\} \right| \leqq \frac{2r}{R - r}\left\{\log M + n \log \frac{R}{R' - R}\right\}.$$

Proof. Consider the function

$$g(s) = \frac{f(s)}{\prod_{k=1}^{n}(s - s_k)}$$

which is regular for $|s - s_0| \leqq R'$ and does not vanish for $|s - s_0| \leqq R$. On $|s - s_0| = R'$ we have

$$\left|\frac{g(s)}{g(s_0)}\right| = \left|\frac{f(s)}{f(s_0)} \prod_{k=1}^{n}\left(\frac{s_0 - s_k}{s - s_k}\right)\right| \leqq \left|\frac{f(s)}{f(s_0)}\right|\left(\frac{R}{R' - R}\right)^n$$

and hence

$$\left|\frac{g(s)}{g(s_0)}\right| \leqq M\left(\frac{R}{R' - R}\right)^n.$$

By the maximum principle this inequality must hold for $|s - s_0| \leqq R'$. The function $h(s) = \log\{g(s)/g(s_0)\}$ with $h(s_0) = 0$ is thus regular for $|s - s_0| \leqq R$ and

$$\operatorname{Re} h(s) \leqq \log M + n \log \frac{R}{R' - R}.$$

Applying the Borel-Carathéodory theorem [9, p. 174] with circles of radius r and R, we obtain

$$|h(s)| \leqq \frac{2r}{R - r}\left\{\log M + n \log \frac{R}{R' - R}\right\}.$$

The lemma follows by using the definitions of $g(s)$ and $h(s)$ and the fact that $|\operatorname{Re} h(s)| \leqq |h(s)|$.

To prove Theorem 1 we show that the inequality

$$\left|\int_{1/2+it}^{\infty+it} \log|\zeta(s)|\, ds\right| < 4.8 \log t + 7.5 \tag{6.3}$$

holds for $t > 100$. Then by applying the inequality for $t = T_1$ and $t = T_2$ we obtain by Lemma 1

$$\pi \mid S_1(T_2) - S_1(T_1) \mid < 9.6 \log T_2 + 15$$

from which the theorem follows immediately.

We break the integral in (6.3) into two parts, one going from $\frac{1}{2} + it$ to $s_0 = 1.1 + it$ and the other from s_0 to $\infty + it$. The latter is easily estimated since for $\sigma > 1$

$$\mid \log \mid \zeta(s) \mid \mid \leqq \mid \log \zeta(s) \mid = \left| \sum_p \sum_{n=1}^{\infty} \frac{1}{np^{ns}} \right| \leqq \sum_p \sum_{n=1}^{\infty} \frac{1}{np^{n\sigma}},$$

and hence by Lemma 2

$$\begin{aligned} \left| \int_{s_0}^{\infty+it} \log \mid \zeta(s) \mid ds \right| &\leqq \int_{1.1}^{\infty} \left\{ \sum_p \sum_{n=1}^{\infty} \frac{1}{np^{n\sigma}} \right\} d\sigma \\ &= \sum_p \sum_{n=1}^{\infty} \frac{1}{n^2 p^{1.1n} \log p} \leqq \frac{1}{\log 2} \sum_p \sum_{n=1}^{\infty} \frac{1}{np^{1.1n}} \qquad (6.4) \\ &\leqq \frac{\log \zeta(1.1)}{\log 2} \leqq \frac{\log 11}{\log 2} < 3.5. \end{aligned}$$

To estimate the integral from $\frac{1}{2} + it$ to $1.1 + it$ we apply Lemma 4 with $f(s) = \zeta(s)$, $s_0 = 1.1 + it$, $r = 0.6$, $R = 0.9$ and $R' = 2.6$. If s_k is a zero of $\zeta(s)$ in the circle $\mid s - s_0 \mid \leqq 0.9$ with Re $s_k = \sigma_k$, then because $\mid s_0 - s_k \mid < 1$ we have

$$\begin{aligned} \int_{1/2+it}^{s_0} \log \left| \frac{s - s_k}{s_0 - s_k} \right| ds &\geqq \int_{1/2+it}^{s_0} \log \mid s - s_k \mid ds \geqq \int_{0.5}^{1.1} \log \mid \sigma - \sigma_k \mid d\sigma \qquad (6.5) \\ &= \int_{0.5-\sigma_k}^{1.1-\sigma_k} \log \mid u \mid du \geqq \int_{-0.3}^{0.3} \log \mid u \mid du > -1.33 \end{aligned}$$

since $\log \mid u \mid$ is an even function which is monotone increasing for $u > 0$.

It is easily seen that when s is on the line between $\frac{1}{2} + it$ and $s_0 = 1.1 + it$, and s' is in the strip $0 < \text{Re } s' < 1$, the quantity $\mid s - s' \mid / \mid s_0 - s' \mid$ takes its maximum value for $s = \frac{1}{2} + it$, $s' = 1 + it$. Consequently

$$\int_{1/2+it}^{s_0} \log \left| \frac{s - s_k}{s_0 - s_k} \right| ds < \int_{1/2+it}^{s_0} \log 5 \, ds = 0.6 \log 5 < 1.33. \qquad (6.6)$$

On the circle $\mid s - s_0 \mid = 2.6$ we have $\sigma. \geqq -\frac{3}{2}$. Hence by Lemma 3 we can let $M = \frac{1}{4}(t + 2.6)^2 / \mid \zeta(s_0) \mid$. Then for $t > 100$

$$\begin{aligned} \log M &\leqq \log \tfrac{1}{4} + 2 \log t + 2 \log 1.026 - \log \mid \zeta(s_0) \mid \\ &\leqq 2 \log t - \log \mid \zeta(s_0) \mid - \tfrac{4}{3}. \end{aligned}$$

Using Lemmas 2 and 4 and the inequalities (6.5) and (6.6), we obtain

$$\begin{aligned} &\left| \int_{1/2+it}^{s_0} \log \mid \zeta(s) \mid ds \right| \\ &\qquad \leqq 0.6 \left| \log \mid \zeta(s_0) \mid \right| + 1.33n + \frac{1.2r}{R - r} \left\{ \log M + n \log \frac{R}{R' - R} \right\} \\ &\qquad \leqq 3 \log 11 - 3.2 + 4.8 \log t + (1.33 - 2.4 \log 17/9)n \\ &\qquad \leqq 4.8 \log t + 4 \end{aligned}$$

because $1.33 - 2.4 \log (17/9) < -0.19 < 0$. Combining this inequality with (6.4), we obtain (6.3).

7. The ALGOL Program. In this section we give the complete text of the main program used in the verification. The program uses three nonlocal procedures besides the standard ALGOL functions. The procedures *print2* and *print4* are used for output of 2 or 4 integers, respectively; and the procedure *input* is used for input of an integer. If this program is used with another machine or with another system of performing real arithmetic, the expression E for the error bound in the procedure *sign Z* should be changed.

We remark that in the procedure *sign Z* it would have been better to replace the statement

```
    if m × m > tau then m := m−1
```

by

```
    if m × m > tau then m := m−1
    else if (m+1) × (m+1) < tau then m := m+1
```

since this would have guaranteed that $0 < \tau^{1/2} - m < 1$ and permitted simplification of the considerations in Section 2.

```
begin
integer j, k, n, s, sa, sb, sc, sd, first, last, displacements, doubtful;
real a, b, c, d, q, u, v;
array cs[0:16383], log1, rsqrt[1:400], log2[1:400];
comment The array log2 should contain elements with double-precision accuracy;
integer procedure sign Z(tau); value tau; real tau;
comment This procedure assigns to the function designator the value +1 if
Z(tau) is positive, −1 if Z(tau) is negative, and 0 if the procedure cannot determine
the sign;
begin integer j, m, n;
  real E, k, s, sg, sum, t1, t2, x, Z;
  Boolean first;
  real procedure fractional part(x); value x; real x; fractional part := x − entier(x);
  s := sqrt(tau);
  m := entier(s); if m × m > tau then m := m−1;
  comment The arithmetic in the following statement should be performed in
  double-precision;
  k := fractional part(0.5 × tau × (ln(tau) − 1) − 0.0625);
  first := true; sum := 0;
  comment The next loop is the innermost loop of the program;
  for n := 1 step 1 until m do
  begin t1 := abs(fractional part(tau × log1[n]) − k);
        j := entier(2 ↑ 14 × t1);
        sum := sum + cs[j] × rsqrt[n]
  end;
  sg := if m = (m ÷ 2) × 2 then −1 else 1;
  x := (1 − 2 × (s−m)) ↑ 2;
  t2 := ((((−0.00162 × x − 0.01357) × x − 0.01361) × x
        + 0.13238) × x + 0.43724) × x + 0.38268;
```

```
check: Z := 2 × sum + sg × t2/sqrt(s);
  E := if first then
           3.2_{10}−9 × tau ↑ (5/4) + 1.6_{10}−8 × tau ↑ (1/2)
           + 8_{10}−4 × tau ↑ (1/4) + 5_{10}−4 × tau ↑ (−1/4) + 0.95 × tau ↑
             (−3/4)
       else
           7.3_{10}−13 × tau ↑ (5/4) + 1.6_{10}−8 × tau ↑ (1/2)
           + 1.6_{10}−9 × tau ↑ (1/4) + 5_{10}−4 × tau ↑ (−1/4) + 0.95 × tau ↑
             (−3/4);
  if abs(Z) > 2 × E then begin sign Z := sign(Z); go to exit end;
  if ¬ first then begin sign Z := 0; go to exit end;
  first := false; sum := 0;
  comment If abs(Z) is less than twice the error bound, the sum is recomputed
  using double-precision arithmetic;
  for n := 1 step 1 until m do
  begin t1 := abs(fractional part(tau × log2[n]) − k);
        sum := sum + cos(6.283185307179586 × t1) × rsqrt[n]
  end;
  go to check;
exit:
end;

procedure Gram(n, a, b); value n; integer n; real a, b;
comment Using a as an initial approximation the n+1th Gram point is calcu
lated and assigned to b;
begin real t1, t2, t3, difference;
  t1 := a; difference := _{10}10;
iterate: t2 := (n+1.125)/(ln(t1) − 1); t3 := abs(t2−t1);
  if t3 < difference then
  begin t1 := t2; difference := t3; go to iterate end;
  b := t2
end;

Boolean procedure two zeros found(a, c, s);
value a, c, s; real a, c; integer s;
comment This procedure searches for two zeros in the interval (a, c). The param-
eter s gives the sign of Z at a and c. The function designator is assigned the value
true if the zeros are found and the value false if they are not;
begin real h, t; integer i, j;
  h := (c−a)/4;
  for i := 1 step 1 until 7 do
  begin for j := 1 step 1 until 2 ↑ i do
        begin t := a + (2 × j − 1) × h;
        if sign Z(t) = −s then
        begin two zeros found := true; go to complete end
        end;
        h := h/2
  end;
  two zeros found := false;
complete:
end;

comment The main program begins at this point;
initialize:
  for j := 0 step 1 until 16383 do
  cs[j] := cos(6.2831853072 × (j/16384 + 1/32768));
  for j := 1 step 1 until 400 do
  begin log1[j] := ln(j); rsqrt[j] := 1/sqrt(j);
    comment The following statement should be performed using double-precision arithmetic;
    log2[j] := ln(j)
  end;
start: displacements := doubtful := 0; input(first); input(last);
  n := first; if n = 0 then go to exit;
  Gram(n−1, n, a); sa := sign Z(a); if sa = 0 then go to reset;
continue: if n > last then
  begin print4(first, last, displacements, doubtful); go to start end;
  Gram(n, a, b); k := 0;
L1: sb := sign Z(b);
  if sb = −sa then
normal: begin n := n+1; a := b; sa := sb; go to continue end;
  if sb = 0 then
  begin doubtful := doubtful + 1; b := b − (b−a)/128;
    if k > 32 then go to error exit; k := k+1; go to L1
  end;
  comment If the signs are the same at a and b, then a failure of Gram's law has
  occurred;
  displacements := displacements + 1;
  Gram(n+1, b, c);
L2: sc := sign Z(c);
  if sc = 0 then
  begin doubtful := doubtful + 1; c := c − (c−b)/128;
    if k > 32 then go to error exit; k := k+1; go to L2
  end;
  if sc = sa then
  begin comment In this case an attempt is made to find two zeros in the interval
    (a, c). Most failures of Gram's law fall under this case;
    if two zeros found(a, c, sa) then
    begin n := n+2; a := c; go to continue end;
    print2(n, 1); n := n+2; a := c; go to reset
  end;
  displacements := displacements + 1;
  Gram(n+2, c, d); sd := sign Z(d);
  if sd ≠ −sa then
```

```
    begin print2(n, 2); n := n+3; a := d; go to reset end;
    comment In the following case an attempt is made to find three zeros in the
    interval (a, d);
    q := (b+c)/2;
L3: s := sign Z(q);
    if s = 0 then
    begin q := q − (c−b)/128;
      if k > 32 then go to error exit; k := k+1; go to L3
    end;
    if s = sa then begin u := a; v := q end
    else begin u := q; v := d end;
    n := n+3; a := d; sa := sd;
    if two zeros found(u, v, s) then go to continue
    else begin print2(n−3, 3); go to reset end;
reset: Gram(n, a, b); sb := sign Z(b);
    if sb = −sa then go to normal;
    print2(n, 4); n := n+1; a := b; sa := sb; go to reset;
    comment. The next statement is executed if the sign of Z is left undetermined
    at too many points. This probably indicates insufficient accuracy in the pro-
    cedure sign Z for the range considered;
error exit: print2(n, 5);
exit:
end
```

8. Acknowledgment. The author wishes to express his indebtedness to the referee for many helpful remarks, including a simplification of the proof of Lemma 2 and detection of a programming error which required modification of the error analysis.

University of California
Berkeley, California

1. D. H. Lehmer, "On the roots of the Riemann zeta-function," *Acta Math.*, v. 95, 1956, pp. 291–298. MR **19,** 121.

2. D. H. Lehmer, "Extended computation of the Riemann zeta-function," *Mathematika*, v. 3, 1956, pp. 102–108. MR **19,** 121.

3. N. A. Meller, "Computations connected with the check of Riemann's hypothesis," *Dokl. Akad. Nauk SSSR*, v. 123, 1958, p. 246–248. (Russian) MR **20** #6396.

4. E. C. Titchmarsh, "The zeros of the Riemann zeta-function," *Proc. Roy. Soc. London*, v. 151, 1935, p. 234–255; v. 157, 1936, p. 261–263.

5. A. M. Turing, "Some calculations of the Riemann zeta-function," *Proc. London Math. Soc.*, 3, v. 3, 1953, pp. 99–117. MR **14,** 1126.

6. C. B. Haselgrove & J. C. P. Miller, *Tables of the Riemann Zeta-Function*, Royal Society Mathematical Tables, Vol. 6, Cambridge University Press, New York, 1960. MR **22** #8679.

7. National Bureau of Standards, *Table of Natural Logarithms*, Vol. I, New York, 1941. MR **2,** 366.

8. E. C. Titchmarsh, *The Theory of the Riemann Zeta-Function*, Oxford Univ. Press, New York, 1951. MR **13,** 741.

9. E. C. Titchmarsh, *The Theory of Functions*, Oxford Univ. Press, New York, 1939.