AFDELING NUMERIEKE WISKUNDE
(DEPARTMENT OF NUMERICAL MATHEMATICS)

NW 113/81 OKTOBER

J. VAN DE LUNE, H.J.J. TE RIELE & D.T. WINTER

RIGOROUS HIGH SPEED SEPARATION OF ZEROS OF RIEMANN'S ZETA FUNCTION

amsterdam 1981

stichting
mathematisch
centrum

MC

AFDELING NUMERIEKE WISKUNDE
(DEPARTMENT OF NUMERICAL MATHEMATICS)

NW 113/81 OKTOBER

J. VAN DE LUNE, H.J.J. TE RIELE & D.T. WINTER

RIGOROUS HIGH SPEED SEPARATION OF ZEROS OF RIEMANN'S ZETA FUNCTION

kruislaan 413 1098 SJ amsterdam

Printed at the Mathematical Centre, 413 Kruislaan, Amsterdam.

1980 Mathematics subject classification: 10H05, 10-04, 65GXX

Rigorous high speed separation of zeros of Riemann's zeta function

by

J. van de Lune, H.J.J. te Riele & D.T. Winter

THIS WORK IS DEDICATED TO PROFESSOR A. VAN WIJNGAARDEN, ON THE OCCASION OF HIS 65TH BIRTHDAY, NOVEMBER 2, 1981, AND ON THE OCCASION OF HIS RETIREMENT AS DIRECTOR FROM THE MATHEMATICAL CENTRE.

ABSTRACT

This report presents the following result, obtained by extensive computations: the first 200,000,001 zeros of the Riemann zeta function $\zeta(s)$ in the critical strip are simple and lie on the line $\mathrm{Re}(s) = \frac{1}{2}$. This extends Brent's previous bound 156,800,001.

The FORTRAN/COMPASS program, by which the new bound was obtained, is included.

KEY WORDS & PHRASES: *Riemann hypothesis, Riemann zeta function, Riemann-Siegel formula, Rosser's rule*

CONTENTS PAGE

1. INTRODUCTION

Riemann's zeta function is the meromorphic function $\zeta: \mathbb{C} \setminus \{1\} \to \mathbb{C}$, which, for $\mathrm{Re}(s) > 1$, may be represented explicitly by

$$\zeta(s) = \sum_{n=1}^{\infty} n^{-s}, \quad (s = \sigma+it).$$

It is well known (see TITCHMARSH [21, Chapters II and X]) that

$$\xi(s) := \tfrac{1}{2}s(s-1)\pi^{-\frac{s}{2}}\,\Gamma(\tfrac{s}{2})\zeta(s)$$

is an entire function of order 1, satisfying the functional equation

$$\xi(s) = \xi(1-s),$$

so that

$$\Xi(s) := \xi(\tfrac{1}{2}+is),$$

being an even entire function of order 1, has an infinity of zeros. The Riemann Hypothesis is the statement that all zeros of $\Xi(s)$ are real, or, equivalently, that all non-real zeros of $\zeta(s)$ lie on the "critical" line $\sigma = \frac{1}{2}$. Since $\zeta(\bar{s}) = \overline{\zeta(s)}$ we may restrict ourselves to the half plane $t > 0$. To this day, Riemann's Hypothesis has neither been proved nor disproved.

Numerical investigations related to this unsolved problem were initiated by Riemann himself and later on continued more systematically by the writers listed below (including their progress).

Investigator	Year	The first N complex zeros of $\zeta(s)$ are simple and lie on $\sigma = \frac{1}{2}$
GRAM [10]	1903	N = 15
BACKLUND [1]	1914	N = 79
HUTCHINSON [12]	1925	N = 138

Those listed above utilized the Euler-Maclaurin summation formula and performed their computations by hand or desk calculator whereas those listed below applied the Riemann-Siegel formula in conjunction with electronic computing devices.

TITCHMARSH [20]	1935/36	N = 1,041
LEHMER [14,15]	1956	N = 25,000
MELLER [16]	1958	N = 35,337
LEHMAN [13]	1966	N = 250,000
ROSSER, YOHE & SCHOENFELD [19]	1968	N = 3,500,000
BRENT [3]	1979	N = 81,000,001

An excellent explanatory account of most of these computations may be found in EDWARDS [8]. From our recent correspondence with BRENT [5] we learned that in the meanwhile he has extended his result to N = 156,800,001. For the present, we have extended this result to N = 200,000,001. We also have, independently, extended Brent's former bound N = 81,000,001 to N = 120,000,000, but after hearing that he had already reached N = 156,800,001 (and had stopped there) we continued our computations from there on. A joint paper with Brent on these results will be published elsewhere.

For an easy understanding of this report we recommend the reader to study Brent's paper [3], also because our notation is virtually the same as his.

The numerical verification of the Riemann Hypothesis consists of the separation of the zeros of a real-valued function Z(t), given in section 3.0. Our program (see section 5) is based, essentially, on the method of ROSSER et al. [19], although we have developed a different strategy of searching for sign changes of Z(t) in Gram blocks of length $L \geq 2$ (see section 2). This enabled us to bring down the average number of Z-evaluations, needed to separate a zero from its predecessor, from about 1.41 (see BRENT [3]), to about 1.21. From the statistics in section 4 it follows that

we could not have reduced this figure below 1.135. It may be noted here that about 98% of the running time was spent on evaluating Z(t). Our program was executed on a CDC system consisting of two CYBER 750 computers and ran about ten times as fast as the Univac 1100/42 program of BRENT [3].

2. THE STRATEGY FOR FINDING THE REQUIRED NUMBER OF SIGN CHANGES OF Z(t) IN A GRAM BLOCK OF LENGTH $L \geq 2$.

For convenience we recall some definitions. A Gram block of length $L(\geq 1)$ is an interval $B_j = [g_j, g_{j+L})$ such that g_j and g_{j+L} are good Gram points and $g_{j+1}, \ldots, g_{j+L-1}$ are bad Gram points. For $L = 1$ we have a Gram interval. The j-th ($j = -1,0,1,2,3,\ldots$) Gram point g_j is defined as the unique number in $[7,\infty)$ satisfying $\theta(g_j) = j\pi$ (see equation (3.2)). A Gram point g_j is called good if $(-1)^j Z(g_j) > 0$ and bad otherwise. A Gram block B_j of length L is said to satisfy "Rosser's rule" if Z(t) has at least L zeros in B_j.

As in ROSSER et al. [19] and BRENT [3] our strategy of searching for the required number of zeros of Z(t) in a given Gram block is essentially based on this rule.

In order to reduce the number of Z-evaluations as much as possible, we first observe that after having determined a Gram block B_j of length $L \geq 2$, we already have implicitly detected L-2 sign changes of Z(t). Hence, the problem reduces to finding the "missing two". Next we observe that these missing two (if existing) must both lie in one and the same Gram interval of the block B_j. Some preliminary experiments with our program revealed that in the majority of cases the missing two are situated in one of the outer Gram intervals of B_j. Therefore, we first search in (g_j, g_{j+1}) or (g_{j+L-1}, g_{j+L}) according to which of $\text{abs}(Z(g_j) + Z(g_{j+1}))$ and $\text{abs}(Z(g_{j+L-1}) + Z(g_{j+L}))$ is the smallest. In the selected interval an efficient parabolic search routine SRCH2A is invoked (see lines 4720 - 5240 of the program). If this routine terminates without having found the missing two zeros, the other outer Gram interval of the block is treated in the same manner. In case the missing two are still not found, we call search routine SRCH2B resp. SRCH3, if $L = 2$ resp. $L > 2$.

SRCH2B (lines 5840 - 6230) searches for the missing two by putting a

grid of increasing refinement on the Gram block $B_j = (g_j, g_{j+2})$ until one (and hence two) sign change is found, or the maximal search depth is reached. In the latter case the missing two were not found or, in case of non-existence, we have found an exception to Rosser's rule (which occurs very rarely, cf. table 4.2).

SRCH3 (lines 6460-6670) searches for the missing two in the inner L-2 Gram intervals of B_j by means of an adaption of a search routine described by LEHMAN [13]. SRCH3 is always applied to the composing Gram intervals and never to the block as a whole. In addition, this search is performed in a zig-zag manner, moving from the periphery of the block towards its centre. Such a zig-zag cycle is repeated a number of times with a grid of increasing refinement. For more details, we refer to the source text.

If at some instant one of our search routines has detected the missing two, a new Gram block is set up and we continue as described above. In the opposite case the program prints a message and a "plot" of Z(t) corresponding to the whole Gram block under investigation and proceeds by pretending(!) that the missing two were found indeed. These plots of Z(t) were inspected afterwards (if necessary) "by hand". So far, the missing two were always easily found either in the Gram block under consideration or in an adjacent Gram block (compare BRENT [3, section 4]).

After having covered the range $(g_{156,000,000}, g_{200,000,000})$ we ran the computation a little further, and found 4 Gram blocks in $(g_{200,000,000}, g_{200,000,004})$, all of them satisfying Rosser's rule. By applying Theorem 3.2 of BRENT [3] we completed the proof of our claim that the first N = 200,000,001 zeros of $\zeta(s)$ are simple and lie on $\sigma = \frac{1}{2}$.

We intend to extend our computations in the near future.

3. COMPUTATION OF Z(t) AND ERROR ANALYSIS

3.0. Introduction.

The unambiguous determination of the sign of Z(t) requires a rigorous bound for the error, committed when one actually computes Z(t) on a computer.

In our program we actually used two methods (A and B) for evaluating Z(t).

Method A is a very fast and efficient method which usually gives the correct sign of $Z(t)$.

Method B is a comparatively slow, but very accurate method which is invoked when $|Z(t)|$ is too small for method A.

We used the well-known Riemann-Siegel formula (with two correction terms in either case):

$$(3.1)\qquad Z(t) = 2\sum_{k=1}^{m} k^{-\frac{1}{2}} \cos[t.\ln(k) - \theta(t)] + (-1)^{m-1}\tau^{-\frac{1}{4}} \sum_{j=0}^{1} \Phi_j(z)\tau^{-j/2} + \\ + R_1(t),$$

where $m = \lfloor \tau^{\frac{1}{2}} \rfloor$, $\tau = t/(2\pi)$, $z = 1-2(\tau^{\frac{1}{2}}-m)$,

$$(3.2)\qquad \theta(t) = \arg[\pi^{-\frac{1}{2}it}\Gamma(\tfrac{1}{4}+\tfrac{1}{2}it)] \ , \quad \theta(t) \text{ continuous and } \theta(0) = 0,$$

$$(3.3)\qquad \Phi_0(z) = \cos[\pi(4z^2+3)/8]/\cos(\pi z) =: \sum_{k=0}^{\infty} c_{2k}^{(0)} z^{2k},$$

$$(3.4)\qquad \Phi_1(z) = \Phi^{(3)}(z)/(12\pi^2) := \sum_{k=0}^{\infty} c_{2k+1}^{(1)} z^{2k+1}.$$

The last term $R_1(t)$ will be dropped in our actual computation. GABCKE [9] and BRENT & SCHOENFELD [4] have given bounds on $R_n(t)$ (here, n+1 denotes the number of terms in the second sum in (3.1)). We used the bound (GABCKE [9])

$$(3.5)\qquad |R_1(t)| < 0.053t^{-5/4} < 0.0054\tau^{-5/4}, \text{ for } t \geq 200.$$

The floating point machine approximations of Z by means of methods A and B will be denoted by $\tilde{Z}_A$ and $\tilde{Z}_B$, respectively. Throughout this section, the result of the floating point machine approximation of some quantity q will be denoted by $\tilde{q}$.

We present an error analysis which accounts for *all* possible errors in $\tilde{Z}$, for any t (resp. τ) in the range,

$$(3.6)\qquad 3.5 \times 10^7 < t < 3.72 \times 10^8 \text{ (resp. } 5.5 \times 10^6 < \tau < 5.92 \times 10^7).$$

This covers the range of zero #81,000,001 till zero #1,000,000,000 of $\zeta(s)$ in the critical strip, which we had originally planned to investigate ($\gamma_{81,000,001} \approx 35,018,261.166$, $\gamma_{1,000,000,000} \approx 371,870,203.837$).

The computations were carried out on a CDC CYBER 750 computer having a 60-bit word, and single-precision (SP) and double-precision (DP) floating-point arithmetic using 48- and 96- bit binary fractions, respectively. In the sequel we will frequently work with the unit roundoffs $\varepsilon_s = 2^{-47}$ and $\varepsilon_d := 2^{-95}$.

3.1. Computation of Z(t).

At the start of the program four tables are precomputed:

(i) $\ln(k)$ for $1 \le k \le m_0$ in DP, where m_0 is large enough to cover the range of the current job;

(ii) $k^{-\frac{1}{2}}$ for $1 \le k \le m_0$ in DP, truncated to SP;

(iii) $\cos(2\pi k.2^{-13})$ for $0 \le k \le 2^{13} + 1$ in DP, truncated to SP;

(iv) $\cos(2\pi(k+1)2^{-13})$ $\cos(2\pi k2^{-13})$ for $0 \le k \le 2^{13}$ in DP, truncated to SP.

Methods A and B run essentially as follows.

Method A. Given a τ as a DP floating point number, $t = 2\pi\tau$ and $\theta(t)$ are computed in DP; $f^{(1)} := \mathrm{frac}\{\theta(t)(2\pi)^{-1}\}$ is computed in DP, and truncated to SP. Next, the main loop (corresponding to the first sum in (3.1)) is executed. This loop has been written in COMPASS (machine language of the CYBER) and optimized using the specific properties of the CYBER'S central processing units. One cycle of the loop executes in about 2.1 μ sec. $f^{(2)} := \mathrm{frac}\{\tau \ln(k)\}$ (where $\ln(k)$ is looked up in the precomputed table) is computed as follows: the DP product of τ and $\ln(k)$ is decreased with the integer part of the SP product of τ and $\ln(k)$ and the result is truncated to SP. This programming "trick" (see lines 8320-8480 in section 5) saves a considerable amount of time in the main loop. $x = \mathrm{abs}(f^{(1)}-f^{(2)})$ is computed in SP, and $\cos(2\pi x)$ is approximated by linear interpolation in the precomputed cosine-table, using the precomputed cosine-difference table. The result is multiplied by the precomputed $k^{-\frac{1}{2}}$ and the product is accumulated in an SP sum. End of the main loop. Next, the two terms in the asymptotic

expansion of the Riemann-Siegel formula (3.1) are approximated using the truncated Taylor series expansions

$$\Phi_0(z) \cong \sum_{k=0}^{N_0} c_{2k}^{(0)} z^{2k} \quad \text{and} \quad \Phi_1(z) \cong \sum_{k=0}^{N_1} c_{2k+1}^{(1)} z^{2k+1}. \tag{3.7}$$

The total correction is computed and added to 2 times the sum obtained in the main loop. The computations after the main loop are carried out in SP.

Method B. The same as method A, with *all* computations in DP. The value of $\cos(2\pi x)$ is computed using the available standard DP library function DCOS.

3.2. Error analysis.

In our error analysis we assume that τ is exactly representable as a floating point number. The positive integer m $(= \lfloor \tau^{\frac{1}{2}} \rfloor)$ is *exactly* computed from τ by testing the inequalities $m^2 \le \tau < (m+1)^2$ and by correcting the machine-computed value, if necessary. Now let

$$s(t) := 2 \sum_{k=1}^{m} k^{-\frac{1}{2}} \cos[t \cdot \ln(k) - \theta(t)] \qquad (t = 2\pi\tau). \tag{3.8}$$

By $\tilde{s}(\tilde{t})$ we denote the computed value of $s(t)$, where errors may be made in the computation of t, $\ln(k)$, $\theta(t)$, $t \cdot \ln(k) - \theta(t)$, $\cos(.)$, $k^{-\frac{1}{2}}$ and the final inner product. The following lemma accounts for *all* these errors.

LEMMA 3.1. *Suppose that* $|t-\tilde{t}| \le \delta_0 t$, $|\ln(k) - L(k)| \le \delta_1 \ln(k)$ *for* $k = 1,2,\ldots,m$, *and* $|\theta(u)-\tilde{\theta}(u)| \le \delta_2 \theta(u)$; *let* $f_k := \mathrm{frac}\{\tau L(k) - \tilde{\theta}(\tilde{t})(2\pi)^{-1}\}$ *and suppose that* $|f_k - \tilde{f}_k| \le \delta_3$ *for* $k = 1,2,\ldots,m$. *Moreover, suppose that* $|\cos(x) - \tilde{c}(x)| \le \delta_4$ *for* $0 \le x \le 2\pi + h$, *where* h *is fixed* (the reason for the occurrence of this (small) number h in this lemma will be clarified in section (3.3)), $|k^{-\frac{1}{2}} - \tilde{k}^{-\frac{1}{2}}| \le \delta_5 k^{-\frac{1}{2}}$ *for* $k = 1,2,\ldots,m$, *and that the inner product of the two vectors with components* $(1 \le k \le m)$ $\tilde{k}^{-\frac{1}{2}}$ *and* $\tilde{c}(2\pi\tilde{f}_k)$, *respectively, is computed in floating point arithmetic, with a relative error in the basic arithmetic operations* $(+,-,*$ *and* $/)$ *bounded by* ε. *Then we have*

$$\begin{aligned} |s(t)-\tilde{s}(\tilde{t})| \le\ & 4\pi\tau^{5/4}\ln(\tau)[2\delta_0+\delta_1(1+\delta_0)+\delta_2] + \\ & + 4\tau^{\frac{1}{4}}[2\pi\delta_3+\delta_4+(1+\delta_4)\{\delta_5+(1+\delta_5)((1+\varepsilon)^m-1)\}]. \end{aligned} \tag{3.9}$$

This lemma is similar to lemma 5.3 of BRENT [5], the difference being that we explicitly account for *all* possible errors in the computation of s(t). The proof is routine and uses the technique of backward error analysis (cf. WILKINSON [22]) for the inner product computation (cf. PARLETT [17, pp. 30-32]) and for the other basic arithmetic operations.

Let

$$\chi(\tau) := (-1)^{m-1}\tau^{-\frac{1}{4}}[\Phi_0(z) + \tau^{-\frac{1}{2}}\Phi_1(z)]. \tag{3.10}$$

By $\tilde{\chi}(\tau)$ we denote the computed value of $\chi(\tau)$ where errors may be made in the computation of $\tau^{-\frac{1}{2}}, \tau^{-\frac{1}{4}}, z, \Phi_0(z)$, $\Phi_1(z)$, and in the other arithmetic operations. The following lemma accounts for *all* these errors.

LEMMA 3.2. *Let* ε *be as in lemma* 3.1 *and let the relative error in the square root computation be bounded by* $a\varepsilon$. *Moreover, suppose that* $|z-\tilde{z}| \le \delta_6$ *and that* $\Phi_0(z)$ *and* $\Phi_1(z)$ *are approximated by* $\tilde{\Phi}_0(z) := \sum_{k=0}^{N_0} \tilde{c}_{2k}^{(0)} z^{2k}$ *and* $\tilde{\Phi}_1(z) := \sum_{k=0}^{N_1} \tilde{c}_{2k+1}^{(1)} z^{2k+1}$, *respectively, where* $|c_{2k}^{(0)} - \tilde{c}_{2k}^{(0)}| \le \delta_7$ *and* $|c_{2k+1}^{(1)} - \tilde{c}_{2k+1}^{(1)}| \le \delta_7$. *Then*

$$\begin{aligned}|\chi(\tau)-\tilde{\chi}(\tau)| \le\ & \tau^{-\frac{1}{4}}[2\delta_6+2\delta_7(N_0+1)+\frac{1}{(N_0+1)!}(\tfrac{\pi}{2})^{N_0+1}+(5N_0+2a+4)\varepsilon] + \\ & + \tau^{-3/4}[\tfrac{1}{4}\delta_6+2\delta_7(N_1+1)+\frac{1}{6}\frac{N_1+5/2}{(N_1+1)!}(\tfrac{\pi}{2})^{N_1+1}+(5N_1+3a+7)\varepsilon].\end{aligned} \tag{3.11}$$

In the proof of this lemma, which we shall omit, use is made of the inequalities $|\Phi_0(z)| \le 1$, $|\Phi_1(z)| \le 1$, $|\Phi_0'(z)| \le 1$ and $|\Phi_1'(z)| \le \frac{1}{4}$ for $|z| \le 1$ (see [9, Theorem 1, p.60]) and of the bounds given in [9, Theorem 2, p.62] on the error induced by truncating the infinite series in (3.3) and (3.4).

3.3. Derivation of $\delta_0, \ldots, \delta_7$ for methods A and B.

δ_0. Since τ is (assumed to be) exact and we work with a DP value of $t(= 2\pi\tau)$, for both methods, one may easily show that $\delta_0 = 1.01 \times 10^{-28} (> 2^{-93})$ is a safe choice.

δ_1. The values of ln(k), $1 \le k \le 7700$ (where k covers the range corresponding to (3.6)) were computed in DP and then compared with the corresponding numbers computed by means of Brent's multiple-precision package [2] with an accuracy of 40 digits. Taking into account the maximal observed relative error, we can safely take $\delta_1 = 5.1 \times 10^{-29} (> 2^{-94})$ for both methods.

δ_2. The function $\theta(t)$ was computed in DP from the formula

$$\theta(t) = \frac{t}{2}(\ln(\tau)-1) - \frac{\pi}{8} + \frac{1}{48t} + \frac{7}{5760t^3} + \frac{31}{80640t^5} + R\theta(t) \qquad (3.12)$$

where $|R\theta(t)| < \frac{1}{3322t^7}$ (see GABCKE [9, p.4]). According to the manual [6] the maximum relative error made by the DP standard FORTRAN library function DLOG (for the computation of ln(.)) is bounded by $1.8 \times 10^{-27} (< 128\varepsilon_d)$.

In table 3.1 (next page) we present, in detailed steps, a backward error analysis of the algorithm for the computation of $\theta(t)$, following the FORTRAN-source text given in lines 4620-4700 of our program.

Here, we use the following notation: $x(1+\varepsilon^{(j)})$ denotes an approximation of x, with $|\varepsilon^{(j)}| \le j\varepsilon_d$. If $\square$ stands for any of the basic DP arithmetic operations +,-,* and /, and if we want to compute $\alpha \square \beta$, then we may assume in our analysis that the resulting machine number is $(\alpha \square \beta)(1+\varepsilon^{(1)})$. From the last line in the table, it follows that it is sufficient to take $\delta_2 = 3.6 \times 10^{-27}$ $(>139\varepsilon_d)$.

δ_3. Method A. Before the main loop is started, the value of $f^{(1)} = \mathrm{frac}\{\tilde{\theta}(\tilde{t})(2\pi)^{-1}\}$ is precomputed in DP, and truncated to SP. The absolute error introduced should be $\le \varepsilon_s$; however, in our program we first computed $\tilde{\theta}(\tilde{t})$ mod 2π, and next $f^{(1)}$ by multiplying by $(2\pi)^{-1}$. Therefore, a safe bound on this error is $5\varepsilon_s$. In the main loop the value of $f^{(2)} = \mathrm{frac}\{\tau.L(k)\}$ is computed with an absolute error bounded by $2\varepsilon_s$. Using these two bounds it follows that we can safely choose $\delta_3 = 5 \times 10^{-14} (>7\varepsilon_s)$.

Table 3.1

Backward error analysis of the computation of $\theta(t)$ (τ exact); all computations are in DP.

	FORTRAN-text	value-obtained in the machine	comment
1.	DTWOPIN	$(2\pi)^{-1}(1+\varepsilon^{(2)})$	precomputed in main program
2.	DTAU	τ	τ is assumed to be exactly representable in the machine
3.	DT=DTAU*DTWOPI	$(t(1+\varepsilon^{(4)})$	2 and 3 actually do not occur in the source-text, but the analysis is expressed in this way because τ is supposed to be exact, instead of t
4.	DTINV=1.DO/DT	$(t^{-1}(1+\varepsilon^{(6)})$	
5.	DLOG(DTAU)	$\ln(\tau)(1+\varepsilon^{(128)})$	
6.	-1.DO	$(\ln(\tau)-1)(1+\varepsilon^{(130)})$	subtract 1 from $\ln(\tau)$
7.	*.5DO	$\frac{1}{2}(\ln(\tau)-1)(1+\ ^{(130)})$	exact multiplication by $\frac{1}{2}$
8.	*DT	$\frac{1}{2}t(\ln(\tau)-1)(1+\varepsilon^{(135)})$	multiplication by t
9.	-DPISL8	$(\frac{1}{2}t(\ln(\tau)-1)-\frac{\pi}{8}(1+\varepsilon^{(137)})$	subtract $\frac{\pi}{8}$ (precomputed)
10.	DCNST1	$\frac{1}{48}(1+\varepsilon^{(1)})$	precomputed
11.	DCNST2	$\frac{7}{5760}(1+\varepsilon^{(1)})$	
12.	DCNST3	$\frac{31}{80640}(1+\varepsilon^{(1)})$	
13.	DCNST3*DTINV	$\frac{31}{80640t}(1+\varepsilon^{(9)})$	
14.	*DTINV	$\frac{31}{80640t^2}(1+\varepsilon^{(17)})$	
15.	+DCNST2	$(\frac{31}{80640t^2}+\frac{7}{5760})(1+\varepsilon^{(19)})$	
16.	*DTINV	$(\frac{31}{80640t^3}+\frac{7}{5760t})(1+\varepsilon^{(27)})$	
17.	*DTINV	$(\frac{31}{80640t^4}+\frac{7}{5760t^2})(1+\varepsilon^{(35)})$	
18.	+DCNST1	$(\frac{31}{80640t^4}+\frac{7}{5760t^2}+\frac{1}{48})(1+\varepsilon^{(37)})$	
19.	*DTINV	$(\frac{31}{80640t^5}+\frac{7}{5760t^3}+\frac{1}{48t})(1+\varepsilon^{(45)})$	
20.	+ "9"	$\theta(t)(1+\varepsilon^{(139)})$	

Due to the programming "trick" mentioned in 3.1 we must take into account the possibility that the *computed* value of $f^{(2)}$ may be (slightly) larger than 1 by an amount which is bounded by $2.5\varepsilon_s \tau L(k)$. In the t-range (3.6) this excess is bounded by 10^{-5}. Instead of correcting $f^{(2)}$ by subtracting 1, which is needed only very rarely, we use *one* extra element in the cosine interpolation table beyond $\cos(2\pi)$, viz. $\cos(2\pi+h)$, where $h = 2\pi.2^{-13} \approx$ $\approx 7.7 \times 10^{-4} (> 10^{-5})$.

δ_3. Method B. For method B the absolute error in the DP computed value of $\tau L(k) - \tilde{\theta}(\tilde{t})$ can be shown to be bounded by $139\varepsilon_d|\tilde{\theta}(\tilde{t}) - \tau L(k)| \leq 139\varepsilon_d|\tilde{\theta}(\tilde{t})|$. Since $\theta(t) < \pi\tau\ln(\tau)$ and $\tau < 5.93 \times 10^7$ it follows that we may take $\delta_3 = 1.2 \times 10^{-17}$, for method B.

δ_4. In method A the cosine-values are approximated by linear interpolation in the table of $\cos(2\pi\frac{k}{8192})$, $0 \leq k \leq 8192$, using a second table of cosine-differences. The interpolation error is bounded by $\frac{1}{8}(2\pi \times 2^{-13})^2$ (with exact arithmetic). The error induced by the inexact arithmetic in the SP evaluation of the linear interpolation formula is bounded in absolute value by 8×2^{-47}. So $\delta_4 = 7.36 \times 10^{-8}$ is a safe choice for method A.

In method B the cosine-values are approximated by the DP standard FORTRAN library function DCOS. According to the manual [6], a safe choice is $\delta_4 = 1.5 \times 10^{-27}$ for method B.

δ_5. In method A the values of $k^{-\frac{1}{2}}$ are computed in DP and truncated to SP, so that $\delta_5 = 7.2 \times 10^{-15} (> \varepsilon_s)$. In method B the DP values of $k^{-\frac{1}{2}}$ are used. A comparison with the corresponding 40D values obtained with Brent's multiple-precision package [2] shows that we may safely take $\delta_5 = 1.01 \times 10^{-28}$ $(> 2^{-93})$.

δ_6. In both methods, the number $z = 1-2(\tau^{\frac{1}{2}}-\lfloor\tau^{\frac{1}{2}}\rfloor)$ is computed in DP. In method A it is truncated to SP, so that we may take $\delta_6 = 7.2 \times 10^{-15}$ $(> \varepsilon_s)$. In method B the absolute error in z can be safely bounded by $10\varepsilon_d\tau^{\frac{1}{2}} < 2.0 \times 10^{-24}$ so that we may take $\delta_6 = 2.0 \times 10^{-24}$ for method B.

δ_7. The coefficients $c_{2k}^{(0)}$ and $c_{2k+1}^{(1)}$ in the Taylor series expansions of $\Phi_0(z)$ and $\Phi_1(z)$, respectively, were taken from [7] (and compared with [11] and [9]) in such a way that for method A a safe choice is $\delta_7 = 5 \times 10^{-14}$ and for method B: $\delta_7 = 5 \times 10^{-28}$.

In Table 3.2 we have collected the values of $\delta_0,\ldots,\delta_7$ for both methods.

Table 3.2.

Values of $\delta_0,\ldots,\delta_7$ for methods A and B

method	δ_0	δ_1	δ_2	δ_3	δ_4	δ_5	δ_6	δ_7
A	1.01×10^{-28}	5.1×10^{-29}	3.6×10^{-27}	5×10^{-14}	7.36×10^{-8}	7.2×10^{-15}	7.2×10^{15}	5×10^{-14}
B	1.01×10^{-28}	5.1×10^{-29}	3.6×10^{-27}	1.2×10^{-17}	1.5×10^{-27}	1.01×10^{-28}	2.0×10^{-24}	5×10^{-28}

3.4. The error bounds on $\tilde{Z}$ for methods A and B.

To complete the error analysis we apply lemmas 3.1 and 3.2 with $\delta_0,\ldots,\delta_7$ as given in Table 3.2, $\varepsilon = \varepsilon_s = 2^{-47}$, $a = 10$, $N_0 = 16$ and $N_1 = 17$ for method A, and $\varepsilon = \varepsilon_d = 2^{-95}$, $a = 10$, $N_0 = N_1 = 29$ for method B; including the inherent error (3.5) this yields

$$(3.13)\qquad |Z(t)-\tilde{Z}_A(\tilde{t})| \le 3\times10^{-7}\tau^{1/4} + 8.6\times10^{-12}\tau^{-1/4} + 5.4\times10^{-3}\tau^{-5/4} + 5\times10^{-26}\tau^{5/4}\ln(\tau)$$

and

$$(3.14)\qquad |Z(t)-\tilde{Z}_B(\tilde{t})| \le 5.4\times10^{-3}\tau^{-5/4} + 3.1\times10^{-16}\tau^{1/4} + 4.1\times10^{-24}\tau^{-1/4} + 5\times10^{-26}\tau^{5/4}\ln(\tau).$$

Instead of these variable bounds, we actually used in our program the extremely conservative *fixed* bounds $\varepsilon_1 = 10^{-4}$ or 2×10^{-4} in (3.13) and $\varepsilon_2 = 2.5\times10^{-6}$ in (3.14). In the τ-range (3.6) we could safely have taken $\varepsilon_1 = 2.7\times10^{-5}$ and $\varepsilon_2 = 2.0\times10^{-11}$. In case $|\tilde{Z}_A(\tilde{t})|$ was less than ε_1, a few rather small shifts with $\tilde{t}$ were tried. If still no "clear" value was found with method A, method B was invoked. Until now not a single $\tilde{t}$ was "offered" for which method B could not determine the sign of $Z(t)$ rigorously.

To give an impression of the actual accuracy of methods A and B we present in Table 3.3 a difference table of $\tilde{Z}_B$ (76,969,020.001+k×0.001) for

$k = 0,1,\ldots,24$, with 15 differences. Moreover, since $|\tilde{Z}_B(t)-\tilde{Z}_A(t)|$ is an accurate approximation of the error in $\tilde{Z}_A(t)$, we have computed this difference for 10,000 random values of t. The maximal difference was less than 2×10^{-6} (compare our bound $\varepsilon_1 = 10^{-4}$).

Table 3.3

t	$\tilde{Z}_B(t)$	DIFFERENCES
		first ↓ second ↓ third ↓ fourth
76969020.001	-.2895571176471424382 1	
		-.6175E-02
76969020.002	-.2957321354954573483	.1183E-03
		-.6057E-02 .3185E-06
76969020.003	-.3017889009C261723635	.1186E-03 -.9351E-08
		-.5938E-02 .3092E-06 -.1720E-10
76969020.004	-.3077270953199985145O	.1189E-03 -.9368E-08 .6569E-12
		-.5819E-02 .2998E-06 -.1654E-10 .1031E-14
76969020.005	-.3135464095501810851 1	.1192E-03 -.9385E-08 .6579E-12 -.4388E-16
		-.5700E-02 .2904E-06 -.1588E-10 .9875E-15 .8428E-18
76969020.006	-.3192465437641553647 5	.1195E-03 -.9401E-08 .6589E-12 -.4304E-16 -.5507E-17
		-.5581E-02 .2810E-06 -.1523E-10 .9445E-15 -.4664E-17 .1666E-16
76969020.007	-.3248272075178627358 1	.1198E-03 -.9416E-08 .6599E-12 -.4771E-16 .1116E-16 -.4087E-16
		-.5461E-02 .2716E-06 -.1457E-10 .8968E-15 .6491E-17 -.2421E-16 .9040E-16
76969020.008	-.3302881197680796886 8	.1200E-03 -.9431E-08 .6608E-12 -.4121E-16 -.1306E-16 .4952E-16 -.1882E-15
		-.5341E-02 .2622E-06 -.1390E-10 .8556E-15 -.6565E-17 .2531E-16 -.9778E-16 .3767E-15
76969020.009	-.3356290088876430534 6	.1203E-03 -.9445E-08 .6616E-12 -.4778E-16 .1226E-16 -.4826E-16 .1885E-15
		-.5221E-02 .2528E-06 -.1324E-10 .8078E-15 .5690E-17 -.2295E-16 .9069E-16 -.3564E-15
76969020.010	-.3408496126800153767 2	.1205E-03 -.9458E-08 .6624E-12 -.4209E-16 -.1069E-16 .4243E-16 -.1679E-15
		-.5100E-02 .2433E-06 -.1258E-10 .7657E-15 -.5003E-17 .1948E-16 -.7725E-16 .4184E-15
76969020.011	-.3459496783931895466 1	.1208E-03 -.9470E-08 .6632E-12 -.4709E-16 .8785E-17 -.3483E-16 .2504E-15
		-.4979E-02 .2338E-06 -.1192E-10 .7186E-15 .3782E-17 -.1535E-16 .1732E-15 -.1919E-14
76969020.012	-.3509289627329318106 5	.1210E-03 -.9482E-08 .6639E-12 -.4331E-16 -.6563E-17 .1383E-15 -.1668E-14
		-.4858E-02 .2243E-06 -.1125E-10 .6753E-15 -.2781E-17 .1230E-15 -.1495E-14 .1188E-13
76969020.013	-.3557872318753623784 7	.1212E-03 -.9494E-08 .6646E-12 -.4609E-16 .1164E-15 -.1357E-14 .1021E-13
		-.4737E-02 .2148E-06 -.1059E-10 .6292E-15 .1127E-15 -.1234E-14 .8716E-14 -.5077E-13
76969020.014	-.3605242614788728437 7	.1215E-03 -.9504E-08 .6652E-12 .6756E-16 -.1117E-14 .7360E-14 -.4056E-13
		-.4616E-02 .2053E-06 -.9924E-11 .6968E-15 -.1004E-14 .6126E-14 -.3184E-13 .1519E-12
76969020.015	-.3651398366953797070 7	.1217E-03 -.9514E-08 .6659E-12 -.9360E-15 .5009E-14 -.2448E-13 .1114E-12
		-.4494E-02 .1958E-06 -.9258E-11 -.2392E-15 .4005E-14 -.1836E-13 .7954E-13 -.3341E-12
76969020.016	-.3696337521809133236 1	.1219E-03 -.9523E-08 .6657E-12 .3069E-14 -.1335E-13 .5506E-13 -.2227E-12
		-.4372E-02 .1863E-06 -.8592E-11 .2830E-14 -.9342E-14 .3670E-13 -.1431E-12 .5567E-12
76969020.017	-.3740058121055416476 0	.1220E-03 -.9532E-08 .6685E-12 -.6273E-14 .2335E-13 -.8808E-13 .3340E-12
		-.4250E-02 .1768E-06 -.7924E-11 -.3442E-14 .1401E-13 -.5138E-13 .1908E-12 -.7157E-12
76969020.018	-.3782558301626280758 0	.1222E-03 -.9540E-08 .6650E-12 .7740E-14 -.2803E-13 .1028E-12 -.3817E-12
		-.4128E-02 .1672E-06 -.7259E-11 .4297E-14 -.1401E-13 .5138E-13 -.1908E-12
76969020.019	-.3823836295774236296 8	.1224E-03 -.9547E-08 .6693E-12 -.6274E-14 .2336E-13 -.8809E-13
		-.4005E-02 .1577E-06 -.6589E-11 -.1977E-14 .9344E-14 -.3670E-13
76969020.020	-.3863890431149906461 1	.1225E-03 -.9554E-08 .6674E-12 .3070E-14 -.1335E-13
		-.3883E-02 .1481E-06 -.5922E-11 .1093E-14 -.4004E-14
76969020.021	-.3902719130874614188 5	.1227E-03 -.9560E-08 .6685E-12 -.9341E-15
		-.3760E-02 .1386E-06 -.5253E-11 .1589E-15
76969020.022	-.3940320913606274938 1	.1228E-03 -.9565E-08 .6686E-12
		-.3637E-02 .1290E-06 -.4585E-11
76969020.023	-.3976694393598[159461	.1230E-03 -.9569E-08
		-.3514E-02 .1194E-06
76969020.024	-.4011838280753710856 2	.1231E-03
		-.3391E-02
76969020.025	-.4045751380667828135 5	

4. SOME STATISTICS

In Table 4.1 we list the number of Gram blocks of type (j,k), $1 \le j \le 8$, $1 \le k \le j$, in the interval $[g_{156,800,000}, g_{200,000,000}]$, as they were actually counted by our program. On the line with j=2 we also list the number of exceptions to Rosser's rule, all of which turned out to be blocks of length 2 without any zeros, and which could, of course, neither be classified as type (2,1) nor as (2,2). The entries in parentheses give the approximate percentages with respect to the total number of blocks of length j, given in the final column.

Our main purpose of presenting this table is to render support to our strategy of dealing with Gram blocks of length $L \ge 2$. The table shows that our strategy is successful for $2 \le L \le 5$. However, for $L \ge 6$ the missing two zeros show an increasing tendency to lie either in (g_{j+1}, g_{j+2}) or in (g_{j+L-2}, g_{j+1-1}). Only one of the 93 blocks of length L = 7 has its missing two zeros in one of the outer Gram intervals! Since our program may shift the argument of $\tilde{Z}$ slightly, the figures in this table may not be the exact numbers of Gram blocks of various types, but only (very good) approximations (the total number of shifts made by the program was always less than 400 (with $\varepsilon_1 = 0.0001$) in a run of 2,500,000 zeros).

Table 4.1.

Number of Gram blocks of type (j,k), $1 \le j \le 8$, $1 \le k \le j$, in the interval $[g_{156,800,000}, g_{200,000,000}]$

j ↓ / k →	1	2	3	4	5	6	7	8	Total
1	30,162,315								30,162,315
2	2,279,944 (50)	2,281,054 (50)	(number of exceptions to Rosser's rule: 43)						4,561,041
3	479,720 (47)	53,497 (5)	480,613 (47)						1,013,830
4	87,367 (46)	8,592 (4)	8,499 (4)	87,150 (45)					191,608
5	7,581 (38)	1,811 (9)	948 (5)	1,882 (9)	7,678 (39)				19,900
6	156 (12)	337 (27)	119 (10)	126 (10)	366 (29)	147 (12)			1,251
7	0	29	17	3	17	26	1		93
8	0	0	1*)	0	0	1*)	1*)	0	3

*) viz. B_n, for n = 165,152,159, 175,330,804 and 181,390,731.

In table 4.2 we present the 43 exceptions to Rosser's rule, found by our program, i.e., the indices n for which $B_n = [g_n, g_{n+2})$ is a Gram block of length 2 without any zeros.

Table 4.2.

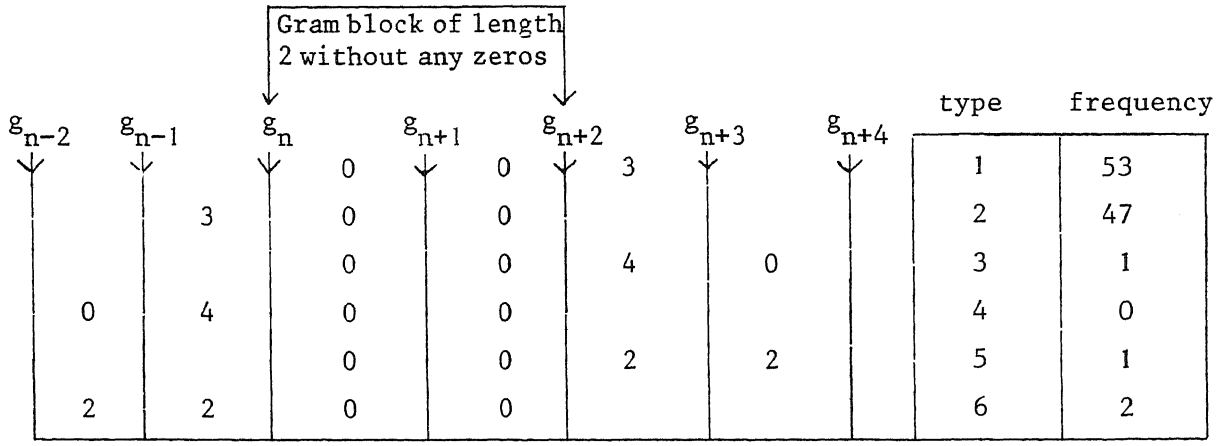
43 Exceptions to Rosser's rule, i.e. indices n for which the Gram block $B_n = [g_n, g_{n+2})$ of length 2 contains no zeros.

n	type	n	type	n	type
157,260,687	2	172,000,993	2	191,116,405	2
157,269,224	1	173,289,941	1	191,165,600	2
157,755,123	1	173,737,614	2	191,297,535	5
158,298,485	2	174,102,513	1	192,485,616	1
160,369,051	2	174,284,990	1	193,264,636	6
162,962,787	1	174,500,513	1	194,696,968	1
163,724,709	1	175,710,609	1	195,876,805	1
164,198,114	2	176,870,844	2	195,916,549	2
164,689,301	1	177,332,733	2	196,395,161	2
164,880,229	2	177,902,862	2	196,676,303	1
166,201,932	1	179,979,095	1	197,889,883	2
168,573,836	1	181,233,727	2	198,014,122	1
169,750,763	1	181,625,435	1	199,235,289	1
170,375,507	1	182,105,257	6		
170,704,880	2	182,223,560	2		

In addition to the types 1,2 and 3 introduced by BRENT [3], we have introduced the types 4,5 and 6, the definitions of which should be clear from Table 4.3. In this table we also list the frequencies of their occurrence in $(g_{-1}, g_{200,000,000})$. Here we used the counts of BRENT [5] in $(g_{-1}, g_{156,800,000})$.

Table 4.3.

Various types of exceptions to Rosser's rule and their frequencies in $(g_{-1}, g_{200,000,000})$.



Finally, from tables 4.1 and 4.2 we have counted the following numbers of Gram intervals in $(g_{156,800,000}, g_{200,000,000})$ containing exactly m($0 \leq m \leq 3$) zeros.

m	0	1	2	3
#	5,864,038	31,548,236	5,711,414	76,312
%	13.6	73.0	13.2	0.2

5. THE PROGRAM

In this section we present the source-text of our program. It should be noted that the present version is a slightly polished version of the original program (without any essential changes).

```
      PROGRAM RHCHECK (OUTPUT,TAPE1=OUTPUT,STATIN,TAPE2=STATIN,
     $                 STATOU,TAPE3=STATOU)
      IMPLICIT DOUBLE (D)
C*********************************************************************
C                    *************************
C                              *****
C.....SET THE DIMENSIONS OF DLN(.) AND SQRTINV(.) PROPERLY !
C.....THESE TWO ARRAYS MUST BE SUFFICIENTLY LONG FOR THE EVALUATION
C.....OF Z(T) AND DZ(DT) IN THE RANGE TO BE INVESTIGATED.
C.....THEIR DIMENSIONS MUST BE AT LEAST INT(SQRT(TMAX/TWOPI)).
      COMMON/MCHZDZ1/DLN(3700)
      COMMON/MCHZDZ2/SQRTINV(3700),PREPCOS(8200),PREPDIF(8200)
C.....THROUGHOUT THIS PROGRAM THE ARRAYS PREPCOS(.) AND PREPDIF(.)
C.....MUST HAVE DIMENSIONS AT LEAST 8193.
C                              *****
C                    *************************
C*********************************************************************
      COMMON/S2PARA/AT(513),AZ(513),NEXT(512),ACCEPT(512)
      COMMON/MCH1/INTRVAL(10,10)
      COMMON/MCH2/NBLOCL(10)
      COMMON/BLOC1/DPI,DPIINV,DTWOPI,DTWOPIN,DPISL8
      COMMON/BLOC2/PI,TWOPI
      COMMON/BLOC3/GRID,GRIDIN
      COMMON/BLOC4/DCNST1,DCNST2,DCNST3
      COMMON/BLOC5/EPS,EPSDBLE,NZEVALU
      COMMON/BLOC6/ZTIME
      COMMON/BLOC7/NSHIFTS
      COMMON/BLOC8/DC0(30),DC1(30),DC2(30),DC3(30)
      DIMENSION TDIM(11),ZDIM(11)
      LOGICAL ACCEPT,STAT
      TOTTIME=SECOND(CP)
C*********************************************************************
C                    *************************
C                              *****
      EPS    = .000 200 000
      EPSDBLE= .000 002 500
      LASTN  =  197 500 000
      LPRINT =            6
      MDIMENS=        3 700
      NPREP  =        8 200
      NRANGE =    2 500 000
      STAT   = .TRUE.
      ZMAX   =           80.
C                              *****
C                    *************************
C*********************************************************************
      PI=DPI=4.D0*DATAN(1.D0) $ DPIINV=1.D0/DPI $ DPISL8=DPI*.125D0
      TWOPI=DTWOPI=2.D0*DPI $ DTWOPIN=.5D0*DPIINV
      WRITE(1,41)LASTN,NRANGE
 41   FORMAT(*1THIS RUN STARTS WITH LASTN=*,I14,/,/,
     $* AND NRANGE                =*,I14,/,/,/)
      NMAX=LASTN+NRANGE $ G=GRAM(NMAX,TWOPI*NMAX/ALOG(FLOAT(NMAX)))
      LASTM=1.+SQRT(G/TWOPI)
      IF(LASTM.LE.MDIMENS)GOTO 43
      WRITE(1,42)MDIMENS,LASTM
 42   FORMAT(* AT THE START MDIMENS=*,I5,* IS TOO SMALL *,/,
     $* REPLACE MDIMENS BY *,I5,* AND INCREASE THE CORRESPONDING *,/,
     $* DIMENSIONS OF DLN(.) AND SQRTINV(.) *)
      STOP
C.....THE ARRAYS DC(30) CAN EASILY BE EXTENDED TO DC(>30).
C.....SEE: HIGH PRECISION COEFFICIENTS RELATED TO THE ZETA FUNCTION
C.....BY F.D.CRARY AND J.BARKLEY ROSSER (UNIV.WISCONSIN,REPORT # 1344).
C.....ALSO SEE: W.GABCKE,DISSERTATION,GOTTINGEN,1979.
 43   DC0(01)=+.3826834323 6508977172 8459984030D+00
```

```
650=      DC0(02)=+.4372404680 7752044936 0296467371D+00
660=      DC0(03)=+.1323765754 8034352332 4035267392D+00
670=      DC0(04)=-.1360502604 7674188654 9831887091D-01
680=      DC0(05)=-.1356762197 0103580887 9156705835D-01
690=      DC0(06)=-.1623725323 1444652828 5462529413D-02
700=      DC0(07)=+.2970535373 3379690783 1272833995D-03
710=      DC0(08)=+.7943300879 5214695880 1639026488D-04
720=      DC0(09)=+.4655612461 4504505037 0634021603D-06
730=      DC0(10)=-.1432725163 0955105754 0824631206D-05
740=      DC0(11)=-.1035484711 2312946075 0074156774D-06
750=      DC0(12)=+.1235792708 3861738056 1257626231D-07
760=      DC0(13)=+.1788108385 7954904985 6667814071D-08
770=      DC0(14)=-.3391414389 9270359069 4062189788D-10
780=      DC0(15)=-.1632663390 2565905101 3740529710D-10
790=      DC0(16)=-.3785109318 5412203828 5464720019D-12
800=      DC0(17)=+.9327423259 2017248456 6232063987D-13
810=      DC0(18)=+.5221843015 9781368553 1389314785D-14
820=      DC0(19)=-.3350673072 7442637895 1509035795D-15
830=      DC0(20)=-.3412426522 8117264940 8098710456D-16
840=      DC0(21)=+.5751203341 4323991603 3950179516D-18
850=      DC0(22)=+.1489530136 3211505454 7562777573D-18
860=      DC0(23)=+.1256537271 7021416853 3042817661D-20
870=      DC0(24)=-.4721295250 1434256689 5398813667D-21
880=      DC0(25)=-.1326906936 3039619992 7354130926D-22
890=      DC0(26)=+.1105343999 5121418344 5378225423D-23
900=      DC0(27)=+.5499646377 5274655111 4010449998D-25
910=      DC0(28)=-.1823137650 2318026280 6410898095D-26
920=      DC0(29)=-.1568940373 7720880146 8682982319D-27
930=      DC0(30)=+.1583963508 8238011610 6597605378D-29
940=C     ..................................................
950=      DC1(01)=+.2682510262 8375347029 9914039557D-01
960=      DC1(02)=-.1378477342 6351853049 8704525899D-01
970=      DC1(03)=-.3849125048 2235082228 7364153632D-01
980=      DC1(04)=-.9871066299 0620764720 1214704619D-02
990=      DC1(05)=+.3310759760 8584043329 0907695130D-02
1000=     DC1(06)=+.1464780857 7954150824 9779656198D-02
1010=     DC1(07)=+.1320794062 4876963675 1614474944D-04
1020=     DC1(08)=-.5922748701 8471413232 2349952819D-04
1030=     DC1(09)=-.5980242585 3734485877 1083507452D-05
1040=     DC1(10)=+.9641322456 1698263526 7298532985D-06
1050=     DC1(11)=+.1833473372 2714411760 0167936578D-06
1060=     DC1(12)=-.4467087562 7178335995 6079422715D-08
1070=     DC1(13)=-.2709635082 1772743216 9262839871D-08
1080=     DC1(14)=-.7785288654 3158510462 9482308521D-10
1090=     DC1(15)=+.2343762601 0893688532 4845504871D-10
1100=     DC1(16)=+.1583017278 9987521642 1622264263D-11
1110=     DC1(17)=-.1211994157 3723791246 6463447380D-12
1120=     DC1(18)=-.1458378116 1108307017 5828548170D-13
1130=     DC1(19)=+.2878630525 8131917504 5582128002D-15
1140=     DC1(20)=+.8662862902 1237241225 2825288793D-16
1150=     DC1(21)=+.8430722727 1370412715 6002253146D-18
1160=     DC1(22)=-.3630807223 0973462001 7324618110D-18
1170=     DC1(23)=-.1162669821 2838296719 4138886292D-19
1180=     DC1(24)=+.1097548671 1527531815 9018328340D-20
1190=     DC1(25)=+.6157399020 4684271038 8147079097D-22
1200=     DC1(26)=-.2290928006 7678471513 9638263100D-23
1210=     DC1(27)=-.2203281174 8848795343 7959827044D-24
1220=     DC1(28)=+.2476025180 0402785082 8527421518D-26
1230=     DC1(29)=+.5954277215 5836578022 7268286395D-27
1240=     DC1(30)=+.3261202074 6795952615 3375631907D-29
1250=C    ..................................................
1260=     DC2(01)=+.5188542830 2931684937 8458151923D-02
1270=     DC2(02)=+.3094658388 0634746033 4567436096D-03
1280=     DC2(03)=-.1133594107 8229373382 1824352559D-01
```

```
1290=       DC2(04)=+.2233045741 9581447720 5712552758D-02
1300=       DC2(05)=+.5196637408 8623302051 1692695307D-02
1310=       DC2(06)=+.3439914407 6208336694 6559135799D-03
1320=       DC2(07)=-.5910648427 4705828217 3225230308D-03
1330=       DC2(08)=-.1022997254 7935857454 4278675227D-03
1340=       DC2(09)=+.2088839221 6992755408 0732961742D-04
1350=       DC2(10)=+.5927665493 0965359578 9199648498D-05
1360=       DC2(11)=-.1642383836 2436275977 6903028478D-06
1370=       DC2(12)=-.1516119970 0940682861 7346053972D-06
1380=       DC2(13)=-.5907803698 2066679629 2279025398D-08
1390=       DC2(14)=+.2091151485 9478188977 7455551897D-08
1400=       DC2(15)=+.1781564958 3292351053 7997018788D-09
1410=       DC2(16)=-.1616407245 5353830752 8557694445D-10
1420=       DC2(17)=-.2380696249 6667615707 2107403801D-11
1430=       DC2(18)=+.5398265295 5425949181 8200414834D-13
1440=       DC2(19)=+.1975014219 6969515273 3087335885D-13
1450=       DC2(20)=+.2333286873 2882634831 0481530059D-15
1460=       DC2(21)=-.1118751761 0048080208 2004838090D-15
1470=       DC2(22)=-.4164009488 8837671885 0112283643D-17
1480=       DC2(23)=+.4446081109 2918830289 0304350093D-18
1490=       DC2(24)=+.2854611478 3637144545 7338742698D-19
1500=       DC2(25)=-.1191323143 0037894304 9718475053D-20
1510=       DC2(26)=-.1298163436 0736498946 7099023133D-21
1520=       DC2(27)=+.1612376317 8033262338 7796586632D-23
1530=       DC2(28)=+.4382497519 8873440596 5525842464D-24
1540=       DC2(29)=+.2718638957 6555759138 8203562714D-26
1550=       DC2(30)=-.1145889650 6774580369 7439455793D-26
1560=C      ..........................................
1570=       DC3(01)=+.1339716090 7194569042 6983572995D-02
1580=       DC3(02)=-.3744215136 3793937046 6416186446D-02
1590=       DC3(03)=+.1330317891 9321468120 3185472240D-02
1600=       DC3(04)=+.2265466076 5471787114 7603199052D-02
1610=       DC3(05)=-.9548499998 5067304151 1225515765D-03
1620=       DC3(06)=-.6010038458 9636039120 7580587580D-03
1630=       DC3(07)=+.1012885828 6776621953 3443494181D-03
1640=       DC3(08)=+.6865733449 2998256424 5742836487D-04
1650=       DC3(09)=-.5985366791 5385981593 0593385329D-06
1660=       DC3(10)=-.3331659851 2399471290 4355366984D-05
1670=       DC3(11)=-.2191928910 2435081057 1848421923D-06
1680=       DC3(12)=+.7890884245 6814944105 5524826157D-07
1690=       DC3(13)=+.9414685081 2952621516 5246515671D-08
1700=       DC3(14)=-.9570116210 8834803018 8072284774D-09
1710=       DC3(15)=-.1876313745 3470662796 8129705778D-09
1720=       DC3(16)=+.4437837679 3233993274 6470898497D-11
1730=       DC3(17)=+.2242673850 5617353248 4110685731D-11
1740=       DC3(18)=+.3627686865 7352436894 0825563792D-13
1750=       DC3(19)=-.1763980955 0821581607 8311214981D-13
1760=       DC3(20)=-.7960765246 7867777572 9034517928D-15
1770=       DC3(21)=+.9419651490 5896907639 1489502569D-16
1780=       DC3(22)=+.7133103854 5696578245 5666792464D-17
1790=       DC3(23)=-.3289910584 5546243211 7966525849D-18
1800=       DC3(24)=-.4180730374 8984592913 6292487056D-19
1810=       DC3(25)=+.5550542071 6463337897 8211640266D-21
1820=       DC3(26)=+.1787044190 6260123858 7176363531D-21
1830=       DC3(27)=+.1331280396 4656094286 2973430146D-23
1840=       DC3(28)=-.5818610611 0909875161 7921659609D-24
1850=       DC3(29)=-.1401903608 8526555374 3649670980D-25
1860=       DC3(30)=+.1464132021 1626254148 9977525019D-26
1870=C      ..........................................
1880=       DO 20 I=1,10
1890=       NBLOCL(I)=0
1900=       DO 10 J=1,10
1910=       INTRVAL(I,J)=0
1920=  10   CONTINUE
```

```
1930= 20   CONTINUE
1940=      DO 30 I=1,MDIMENS
1950=      DI=I $ DLN(I)=DLOG(DI) $ SQRTINV(I)=DSQRT(1.D0/DI)
1960= 30   CONTINUE
1970=C......"8192" IS AN ABSOLUTE CONSTANT FOR THIS PROGRAM !
1980=      GRID=DGRID=DTWOPI/8192.D0 $ GRIDIN=8192.D0/DTWOPI
1990=      PREPCOS(1)=1. $ DPC1=1.D0
2000=      DO 40 I=2,NPREP
2010=      DARG=(I-1)*DGRID $ DPC2=DCOS(DARG)
2020=      PREPCOS(I)=DPC2 $ PREPDIF(I-1)=DPC2-DPC1 $ DPC1=DPC2
2030= 40   CONTINUE
2040=      NZEVALU=NEW00=NSHIFTS=0 $ ZTIME=0. $ NFIRST=LASTN
2050=C.....NZEVALU= # OF Z-EVALUATIONS.
2060=C.....NZEVALU IS SET TO 0 AND HENCE "ECORATE" IS MEASURED LOCALLY.
2070=C.....THE FOLLOWING THREE CONSTANTS ARE USED IN DTHETA(DT).
2080=      DCNST1=1.D0/48.D0 $ DCNST2=7.D0/5760.D0 $ DCNST3=31.D0/80640.D0
2090=C
2100=C.....                    END OF PREPARATIONS
2110=C*****************************************************************************
2120=C*****************************************************************************
2130=C                         ESSENTIAL START OF THE PROGRAM.
2140=C*****************************************************************************
2150=C*****************************************************************************
2160=C
2170=C.....LASTN IS THE INDEX OF THE LAST KNOWN ZERO.
2180=C.....IT IS ADVISED TO TAKE LASTN FROM THE OUTPUT OF THE PREVIOUS RUN.
2190= 48   N=LASTN-1
2200=      G0=GRAM(N,TWOPI*N/ALOG(FLOAT(N)))
2210=      TIME=SECOND(CP) $ Z0=Z(G0)
2220=      ZTIME=ZTIME+(SECOND(CP)-TIME) $ NZEVALU=NZEVALU+1
2230=      IF(ABS(Z0).GT.EPS)GOTO 55
2240=      WRITE(1,50)LASTN,G0,Z0
2250= 50   FORMAT(/,/,* BAD START WITH LASTN=*,I14,/,
2260=     $* Z0(*,F20.6,*)=*,F20.6,/,* THIS STARTING VALUE IS TOO SMALL.*,/,
2270=     $* LASTN IS DECREASED BY 1 DUE TO THIS "UNCLEAR" VALUE OF Z(T).*)
2280=      GOTO 59
2290= 55   IF(Z0*((-1)**N).LT.0.)GOTO 56
2300=      IF(ABS(Z0).LT.ZMAX)GOTO 60
2310=      ZMAX=ZMAX+(ABS(Z0)-ZMAX)*.5
2320=      WRITE(1,67)N,G0,Z0
2330=      GOTO 60
2340= 56   WRITE(1,57)N,G0
2350= 57   FORMAT(/,/,*  LASTN IS DECREASED BY 1 DUE TO THE BAD INITIAL
2360=     $ GRAM POINT G(*,I14,*)=*,F20.6,/,/)
2370= 59   LASTN=LASTN-1 $ GOTO 48
2380= 60   IF(NEW00.LT.NRANGE)GOTO 65
2390=C.....PREPARATION FOR SOME FINAL OUTPUT
2400=      GN=GRAM(N,G0) $ M=SQRT(GN/TWOPI) $ TOTTIME=SECOND(CP)-TOTTIME
2410=C.....N+1 IS THE "LASTN" FOR THE NEXT RUN.
2420=      WRITE(1,61)NFIRST,N+1,GN,NZEVALU,FLOAT(NZEVALU)/NEW00,
2430=     $NSHIFTS,ZTIME/NZEVALU,ZTIME,TOTTIME,TOTTIME/NZEVALU,M,
2440=     $EPS,EPSDBLE
2450= 61   FORMAT(1H1,/,
2460=     $* NFIRST ( WAS INPUT FOR THIS RUN )          =*,I14,/,/,
2470=     $* LASTN  ( INPUT FOR NEXT RUN )              =*,I14,/,/,
2480=     $* THE CORRESPONDING GRAM POINT               =*,F36.6,/,/,
2490=     $* NZEVALU                                    =*,I14,/,/,
2500=     $* ECORATE                                    =*,F34.4,/,/,
2510=     $* NSHIFTS                                    =*,I14,/,/,
2520=     $* AVERAGE TIME FOR ONE Z - EVALUATION        =*,F34.4,/,/,
2530=     $* TOTAL TIME USED FOR ALL Z - EVALUATIONS    =*,F34.4,/,/,
2540=     $* TOTAL TIME USED IN THIS RUN                =*,F34.4,/,/,
2550=     $* AVERAGE TOTAL TIME FOR ONE Z - EVALUATION  =*,F34.4,/,/,
2560=     $* LAST M (= SUMMATION RANGE IN Z(T))         =*,I14,/,/,
```

```
2570=      $* EPS                                                    =*,F40.10,/,/,
2580=      $* EPSDBLE                                                =*,F40.10,/,/,/,/)
2590=       INTRVAL(1,1)=NBLOCL(1)
2600=       DO 64 I=1,10
2610=       WRITE(1,62)I,NBLOCL(I),(INTRVAL(I,J),J=1,I)
2620= 62    FORMAT(* I=*,I2,* # BLOCKL=*,I8,* $ *,2I8,I7,I6,I5,I4,4(I3),/)
2630= 64    CONTINUE
2640=       IF(STAT)CALL STATIST(N+1)
2650=C.....END OF JOB.
2660=       STOP
2670=C.....WE ARE GOING TO SET UP A GRAM BLOCK OF LENGTH LBLOC.
2680= 65    G1=GRAM(N+1,G0) $ TIME=SECOND(CP) $ Z1=Z(G1)
2690=       ZTIME=ZTIME+(SECOND(CP)-TIME) $ NZEVALU=NZEVALU+1
2700=       IF(ABS(Z1).GT.EPS)GOTO 66
2710=       CALL COMPZ(Z1,G0,G1) $ GOTO 69
2720= 66    IF(ABS(Z1).LT.ZMAX)GOTO 69
2730=       ZMAX=ZMAX+.5*(ABS(Z1)-ZMAX)
2740=       WRITE(1,67)N+1,G1,Z1
2750= 67    FORMAT(/,* LARGE VALUE FOR Z ........ *,
2760=      $* G(*,I14,*)=*,F18.6,*  CORRESPONDING Z=*,F18.6,/)
2770= 69    IF(Z0*Z1.GT.0.)GOTO 70
2780=C.....WE ENCOUNTER A GRAM INTERVAL ( LBLOC = 1).
2790=       NEW00=NEW00+1 $ N=N+1 $ G0=G1 $ Z0=Z1
2800=       NBLOCL(1)=NBLOCL(1)+1 $ GOTO 60
2810=C.....EXCEPTION TO GRAM'S LAW. LBLOC WILL BE > 1.
2820= 70    LBLOC=1 $ TDIM(1)=G0 $ ZDIM(1)=Z0
2830= 80    LBLOC=LBLOC+1 $ TDIM(LBLOC)=G1 $ ZDIM(LBLOC)=Z1
2840=       G2=GRAM(N+LBLOC,G1) $ TIME=SECOND(CP) $ Z2=Z(G2)
2850=       ZTIME=ZTIME+(SECOND(CP)-TIME) $ NZEVALU=NZEVALU+1
2860=       IF(ABS(Z2).GT.EPS)GOTO 85
2870=       CALL COMPZ(Z2,G1,G2) $ GOTO 89
2880= 85    IF(ABS(Z2).LE.ZMAX)GOTO 89
2890=       ZMAX=ZMAX+.5*(ABS(Z2)-ZMAX)
2900=       WRITE(1,67)N+LBLOC,G2,Z2
2910= 89    IF(Z1*Z2.GT.0.)GOTO 90
2920=       G1=G2 $ Z1=Z2 $ GOTO 80
2930= 90    TDIM(LBLOC+1)=G2 $ ZDIM(LBLOC+1)=Z2
2940=       NBLOCL(LBLOC)=NBLOCL(LBLOC)+1
2950=C.....THE GRAM BLOCK IS READY AND PRESERVED IN TDIM(.) AND ZDIM(.).
2960=C.....NBLOCL(I) CONTAINS THE NUMBER OF GRAM BLOCKS OF LENGTH I.
2970=C.....NOTE THAT IN THIS GRAM BLOCK WE HAVE ALREADY DETECTED LBLOC-2
2980=C.....ZEROS. THE MISSING TWO ARE FIRSTLY SOUGHT IN THE TWO OUTER
2990=C.....INTERVALS OF THE BLOCK.
3000=       I1=1 $ I2=LBLOC
3010=       IF(ABS(ZDIM(I1)+ZDIM(I1+1)).LT.ABS(ZDIM(I2)+ZDIM(I2+1)))GOTO 95
3020=       I1=LBLOC $ I2=1
3030= 95    CALL SRCH2A(TDIM(I1),TDIM(I1+1),ZDIM(I1),ZDIM(I1+1),NFOUND,I1)
3040=       IF(NFOUND.LT.2)GOTO 120
3050=       INTRVAL(LBLOC,I1)=INTRVAL(LBLOC,I1)+1
3060= 100   NEW00=NEW00+LBLOC
3070=       IF(LBLOC.GE.LPRINT)GOTO 215
3080= 110   N=N+LBLOC $ G0=TDIM(LBLOC+1) $ Z0=ZDIM(LBLOC+1) $ GOTO 60
3090= 120   CALL SRCH2A(TDIM(I2),TDIM(I2+1),ZDIM(I2),ZDIM(I2+1),NFOUND,I2)
3100=       IF(NFOUND.LT.2)GOTO 125
3110=       INTRVAL(LBLOC,I2)=INTRVAL(LBLOC,I2)+1 $ GOTO 100
3120= 125   IF(LBLOC.GT.2)GOTO 160
3130=       CALL SRCH2B(TDIM(1),ZDIM(1),TDIM(3),NFOUND,INTVAL,7,0)
3140=       IF(NFOUND.EQ.0)GOTO 140
3150=       INTRVAL(2,INTVAL)=INTRVAL(2,INTVAL)+1 $ NEW00=NEW00+2 $ GOTO 110
3160= 140   A=TDIM(1) $ B=TDIM(3)
3170=       WRITE(1,145)A,B,N,N+LBLOC
3180= 145   FORMAT(/,40(* !*),/,* EVEN SRCH2B DID NOT YIELD ANY ZEROS BETWEEN
3190=      $*,/,* T1=*,F20.6,*  AND  T2=*,F20.6,/,
3200=      $* WE LIST SOME VALUES OF Z(T) IN THIS RANGE *,/,
```

```
3210=     $* WE ADD TWO ZEROS TO NEW00 AND CONTINUE *,/,
3220=     $* THIS HAPPENED BETWEEN N=*,I14,* AND N+LBLOC=*,I14,/)
3230=      H=(B-A)/11.
3240=      DO 155 I=1,12
3250=      TIME=SECOND(CP) $ ZA=Z(A)
3260=      ZTIME=ZTIME+(SECOND(CP)-TIME) $ NZEVALU=NZEVALU+1
3270=      WRITE(1,150)A,ZA
3280= 150  FORMAT(* T=*,F20.6,*     Z(T)=*,F28.14)
3290=      A=A+H
3300= 155  CONTINUE
3310=      GOTO 100
3320= 160  LASTK=LBLOC-2 $ LASTKP1=LASTK+1 $ IDEPTH=2
3330=      NUM1=INTRVAL(LBLOC,2) $ NUM2=INTRVAL(LBLOC,LBLOC-1)
3340=      IF(NUM1.LT.NUM2)GOTO 202
3350=      DO 180 II=2,7
3360=      ISIGN=-1 $ IP=LASTKP1 $ IDEPTH=2*IDEPTH
3370=      DO 170 K=1,LASTK
3380=C.....THE NEXT LINE CONTROLS THE ZIG-ZAG SEARCH OF SRCH3.
3390=      IP=IP+ISIGN*(LASTKP1-K) $ ISIGN=-ISIGN
3400=      CALL SRCH3(TDIM(IP+1),ZDIM(IP+1),TDIM(IP+2),NFOUND,IDEPTH)
3410=      IF(NFOUND.EQ.3)GOTO 200
3420= 170  CONTINUE
3430= 180  CONTINUE
3440=C.....SRCH2A AND SRCH3 HAVE NOT FOUND THE MISSING TWO.
3450=C.....WE SEARCH AGAIN IN THE MOST SUSPICIOUS OUTER GRAM INTERVAL OF
3460=C.....THE GRAM BLOCK BY MEANS OF SRCH2C.
3470=C.....IN STEAD OF SRCH2C ONE MAY ALSO EXPERIMENT WITH SRCH2B HERE.
3480= 185  IF(ABS(ZDIM(1)+ZDIM(2)).GT.ABS(ZDIM(LBLOC)+ZDIM(LBLOC+1)))GOTO 187
3490=      A =TDIM(1)      $ B =TDIM(2)        $ INDEX=1
3500=      ZA=ZDIM(1)      $ ZB=ZDIM(2)        $ GOTO 1940
3510= 187  A =TDIM(LBLOC) $ B =TDIM(LBLOC+1) $ INDEX=LBLOC
3520=      ZA=ZDIM(LBLOC) $ ZB=ZDIM(LBLOC+1)
3530=C.....FROM THE ABOVE 5 LINES IT IS CLEAR ON WHICH INTERVAL WE BET.
3540= 1940 IF(ABS(ZA).LT.ABS(ZB))GOTO 1945
3550=      AUX=A $ A=B $ B=AUX
3560= 1945 WRITE(1,1946)A,B
3570= 1946 FORMAT(/,* WE CALL SRCH2C BETWEEN *,/,
3580=     $* T1=*,F16.6,*  AND  T2=*,F16.6,/)
3590=      CALL SRCH2C(A,B,ZA,NFOUND,6)
3600=C.....THE LAST PARAMETER IN SRCH2C REGULATES THE SEARCH DEPTH.
3610=      IF(NFOUND.EQ.2)GOTO 1997
3620=C.....IT APPEARS TO BE USELESS TO SEARCH IN THE OTHER OUTER INTERVAL.
3630=      A=TDIM(1) $ B=TDIM(LBLOC+1)
3640=      WRITE(1,195)A,B,LBLOC,LBLOC-2,LBLOC,N,N+LBLOC
3650= 195  FORMAT(/,40(* !*),/,* ? VIOLATION OF ROSSER'S RULE ? BETWEEN *,/,
3660=     $* T1=*,F20.6,* AND T2=*,F20.6,/,
3670=     $* # OF SIGN CHANGES SHOULD BE LBLOC=*,I3,* WE FOUND ONLY L=*,I3,/,
3680=     $* WE ADD LBLOC=*,I3,* ZEROS TO NEW00 *,/,
3690=     $* THE POSSIBLE ERROR SHOULD LATER BE DETECTED "BY HAND" *,/,
3700=     $* WE LIST SOME VALUES OF Z(T) AND CONTINUE *,/,
3710=     $* THIS HAPPENED BETWEEN N=*,I14,* AND N+LBLOC=*,I14,/,/)
3720=      NPRINT=7*LBLOC+1 $ H=(B-A)/(NPRINT-1)
3730=      DO 1995 KK=1,NPRINT
3740=      TIME=SECOND(CP) $ ZA=Z(A)
3750=      ZTIME=ZTIME+(SECOND(CP)-TIME) $ NZEVALU=NZEVALU+1
3760=      WRITE(1,1990)A,ZA
3770= 1990 FORMAT(* T=*,F20.6,*       Z(T)=*,F28.14)
3780=      A=A+H
3790= 1995 CONTINUE
3800=C.....WE PRETEND (!) THAT THE MISSING TWO WERE FOUND !
3810=C.....CLASSIFICATION OF THE BLOCK IS IMPOSSIBLE AT THIS INSTANT.
3820=      GOTO 100
3830=C.....WE CAN NOW CLASSIFY THE GRAM BLOCK.
3840= 1997 INTRVAL(LBLOC,INDEX)=INTRVAL(LBLOC,INDEX)+1 $ GOTO 100
```

```
3850= 200  INTRVAL(LBLOC,IP+1)=INTRVAL(LBLOC,IP+1)+1 $ GOTO 100
3860= 202  DO 208 II=2,7
3870=      ISIGN=+1 $ IP=0 $ IDEPTH=2*IDEPTH
3880=      DO 206 K=1,LASTK
3890=C.....THE NEXT LINE CONTROLS THE ZIG-ZAG SEARCH OF SRCH3.
3900=      IP=IP+ISIGN*(LASTKP1-K) $ ISIGN=-ISIGN
3910=      CALL SRCH3(TDIM(IP+1),ZDIM(IP+1),TDIM(IP+2),NFOUND,IDEPTH)
3920=      IF(NFOUND.EQ.3)GOTO 200
3930= 206  CONTINUE
3940= 208  CONTINUE
3950=      GOTO 185
3960=C.....IF LBLOC >= NPRINT WE PRINT SOME INTERMEDIATE STATISTICS.
3970=C.....ECORATE:= SHORT FOR "ECONOMY RATE".
3980= 215  ECORATE =FLOAT(NZEVALU)/NEW00
3990=      WRITE(1,230)LBLOC,GO,G2,ECORATE,NSHIFTS,(NBLOCL(I),I=1,10)
4000= 230  FORMAT(/,* GRAM-BLOCK OF LENGTH *,I3,*  BETWEEN *,/,
4010=     $* T1=*,F21.6,*  AND         T2=*,F21.6,/,
4020=     $* ECORATE=.....*,F8.4,/,
4030=     $* NSHIFTS=*,I14,/,
4040=     $* BLOCKLENGTHS *,I9,I8,8(I6),/,/,/)
4050=      INTRVAL(1,1)=NBLOCL(1)
4060=      DO 232 I=1,10
4070=      WRITE(1,231)I,(INTRVAL(I,J),J=1,I)
4080= 231  FORMAT(* I=*,I3,10I7)
4090= 232  CONTINUE
4100=      GOTO 110
4110=      END
4120=
4130=      REAL FUNCTION GRAM(N,START)
4140=C.....GRAM(N,START) RETURNS A FAIRLY ACCURATE APPROXIMATION OF
4150=C.....THE N-TH GRAM POINT.
4160=      COMMON/BLOC2/PI,TWOPI
4170=      COMMON/BLOC3/GRID,GRIDIN
4180=      GI=START/TWOPI
4190= 10   GI1=(GI+N+.125)/ALOG(GI)
4200=      IF(ABS((GI-GI1)/GI1).LT.1.E-13)GOTO 20
4210=      GI=GI1 $ GOTO 10
4220= 20   GRAM=GI1*TWOPI
4230=      RETURN
4240=      END
4250=
4260=      SUBROUTINE COMPZ(ZT1,TREFER,T1)
4270=      IMPLICIT DOUBLE (D)
4280=      COMMON/BLOC5/EPS,EPSDBLE,NZEVALU
4290=      COMMON/BLOC6/ZTIME
4300=      COMMON/BLOC7/NSHIFTS
4310=      TORIG=T1 $ ZTORIG=ZT1
4320=      CALL PRSHIF(T1,ZT1)
4330=      DO 10 NZ=1,4
4340=      T1=TREFER+(T1-TREFER)*.95 $ NSHIFTS=NSHIFTS+1
4350=      TIME=SECOND(CP) $ ZT1=Z(T1)
4360=      ZTIME=ZTIME+(SECOND(CP)-TIME) $ NZEVALU=NZEVALU+1
4370=      IF(ABS(ZT1).GT.EPS)RETURN
4380=      CALL PRSHIF(T1,ZT1)
4390= 10   CONTINUE
4400=      T1=TORIG
4410=      WRITE(1,20)
4420= 20   FORMAT(* ..... WE ENTER DZ(DT) ..... *)
4430=      DO 30 NZ=1,10
4440=      DT1=T1=TREFER+(T1-TREFER)*.99 $ NSHIFTS=NSHIFTS+1
4450=      TIME=SECOND(CP) $ ZT1=DZT1=DZ(DT1)
4460=      ZTIME=ZTIME+(SECOND(CP)-TIME) $ NZEVALU=NZEVALU+1
4470=      IF(DABS(DZT1).GT.EPSDBLE)RETURN
4480=      CALL PRSHIF(T1,ZT1)
```

```
4490= 30    CONTINUE
4500=       WRITE(1,40)
4510= 40    FORMAT(/,* SERIOUS DIFFICULTIES IN FINDING A GOOD DZ. WE STOP. *)
4520=       STOP
4530=       END
4540=
4550=       SUBROUTINE PRSHIF(T,ZT)
4560=C.....ANNOUNCEMENT OF SHIFTS.
4570=       WRITE(1,10)T,ZT
4580= 10    FORMAT(* .... WE MAKE A SHIFT AT T=*,F30.14,* ..... ZT=*,F10.8)
4590=       RETURN
4600=       END
4610=
4620=       DOUBLE FUNCTION DTHETA(DT)
4630=       IMPLICIT DOUBLE (D)
4640=       COMMON/BLOC1/DPI,DPIINV,DTWOPI,DTWOPIN,DPISL8
4650=       COMMON/BLOC4/DCNST1,DCNST2,DCNST3
4660=       DTINV=1.D0/DT $ DTAU=DT*DTWOPIN
4670=       DTHETA=DT*.5D0*(DLOG(DTAU)-1.D0)-DPISL8+
4680=      $((DCNST3*DTINV*DTINV+DCNST2)*DTINV*DTINV+DCNST1)*DTINV
4690=       RETURN
4700=       END
4710=
4720=       SUBROUTINE SRCH2A(A,B,ZA,ZB,NFOUND,I)
4730=C.....SEARCH FOR TWO ZEROS BETWEEN A AND B.
4740=C.....SRCH2A IS CALLED ONLY IF ZA*ZB > 0.
4750=       COMMON/S2PARA/AT(513),AZ(513),NEXT(512),ACCEPT(512)
4760=       COMMON/BLOC5/EPS,EPSDBLE,NZEVALU
4770=       COMMON/BLOC6/ZTIME
4780=       LOGICAL ACCEPT,CONVEX
4790=       NFOUND=0
4800=       IF(I.EQ.1)GOTO 1
4810=       AT(1)=A $ AT(513)=B
4820=       AZ(1)=ZA $ AZ(513)=ZB $ GOTO 2
4830= 1     AT(1)=B $ AT(513)=A
4840=       AZ(1)=ZB $ AZ(513)=ZA
4850= 2     T=(A+B)*.5 $ TIME=SECOND(CP) $ ZT=Z(T)
4860=       ZTIME=ZTIME+(SECOND(CP)-TIME) $ NZEVALU=NZEVALU+1
4870=       IF(ABS(ZT).GT.EPS)GOTO 5
4880=       CALL COMPZ(ZT,A,T)
4890= 5     IF(ZA*ZT.GT.0.)GOTO 10
4900=       NFOUND=2 $ RETURN
4910= 10    AT(257)=T $ AZ(257)=ZT
4920=       ACCEPT(  1)=.FALSE. $ NEXT(  1)=256
4930=       ACCEPT(257)=.FALSE. $ NEXT(257)=256
4940=       CONVEX=.FALSE.
4950=       IF(ABS(ZT).LT.AMIN1(ABS(ZA),ABS(ZB)))CONVEX=.TRUE.
4960=       INDEX=1
4970= 20    CALL PARA(INDEX,INDOUT,NFOUND,CONVEX)
4980=C.....NOTE THAT PARA IS CALLED ONLY IN SRCH2A.
4990=       IF(NFOUND.GT.0)RETURN
5000=       IF(INDOUT.LT.513)GOTO 40
5010=C.....THE WHOLE INTERVAL [ A,B ] HAS BEEN SCANNED NOW BY PARA.
5020=C.....WE CHECK WHETHER PARA HAS DISCOVERED CONVEXITY OF ABS(Z(T)).
5030= 30    IF(.NOT.CONVEX)RETURN
5040=C.....PARA HAS DETECTED CONVEXITY, SO THAT WE SHOULD TRY TO FIND
5050=C.....THE MISSING TWO ON [ A,B ].
5060=C.....THE VALUE OF "INTVAL" IN THE NEXT LINE IS IRRELEVANT.
5070=       CALL SRCH2B(A,ZA,B,NFOUND,INTVAL,8,0)
5080=       WRITE(1,35)A,B
5090= 35    FORMAT(/,* IN SRCH2A, PARA DETECTED CONVEXITY BETWEEN *,/,
5100=      $* A=*,F16.6,*  AND  B=*,F16.6,*  HENCE WE CALLED SRCH2B *,/)
5110=       INDEX=1
5120= 36    WRITE(1,38)INDEX,AT(INDEX),AZ(INDEX)
```

```
5130= 38    FORMAT(* INDEX=*,I4,*, AT=*,F20.6,* AZ=*,F16.10)
5140=       IF(INDEX.GE.513)RETURN
5150=       INDEX=INDEX+NEXT(INDEX)
5160=       GOTO 36
5170=C.....TRANSFER THE SEARCH TO THE NEXT INTERVAL.
5180= 40    INDEX=INDOUT
5190= 50    IF(.NOT.ACCEPT(INDEX))GOTO 20
5200=C.....SINCE ACCEPT(INDEX)=.TRUE., WE SKIP THE NEXT INTERVAL
5210=C.....[ AT(INDEX),AT(INDEX+NEXT(INDEX)) ].
5220=       INDEX=INDEX+NEXT(INDEX)
5230=       IF(INDEX.LT.513)50,30
5240=       END
5250=
5260=       SUBROUTINE PARA(INDEX,INDOUT,NFOUND,CONVEX)
5270=C.....PARA ( SHORT FOR PARABOLA ) IS CALLED ONLY IN SRCH2A SO THAT IT
5280=C.....IS APPLIED ONLY TO INTERVALS [ A,B ] WITH ZA*ZB > 0.
5290=C.....DEFINE: N=NEXT(INDEX), T1=AT(INDEX), T3=AT(INDEX+N), N2=N/2.
5300=C.....PARA TRIES TO FIND TWO ZEROS IN THE INTERVAL [ T1,T3 ] BY MEANS
5310=C.....OF A CLOSE LOOK AT THE GRAPH OF ABS(Z(T)) ON THIS INTERVAL.
5320=C.....THIS ROUTINE SAVES MANY EVALUATIONS OF Z(T) IN CASE THE GRAPH
5330=C.....OF ABS(Z(T)) IS CONCAVE.
5340=C.....IF FOR SOME K THE INTERVAL [ AT(K),AT(K+NEXT(K)) ] IS JUDGED
5350=C.....AS CONTAINING NO ZEROS WE SET ACCEPT(K)=.T., ELSE ACCEPT(K)=.F..
5360=       COMMON/S2PARA/AT(513),AZ(513),NEXT(512),ACCEPT(512)
5370=       COMMON/BLOC5/EPS,EPSDBLE,NZEVALU
5380=       COMMON/BLOC6/ZTIME
5390=       LOGICAL ACCEPT,CONVEX
5400=       N=NEXT(INDEX) $ N2=N/2
5410=       IF(N2.GE.1)GOTO 10
5420=       INDOUT=INDEX+1 $ RETURN
5430= 10    NEXT(INDEX)=N2 $ NEXT(INDEX+N2)=N2
5440=       ACCEPT(INDEX)=.TRUE. $ ACCEPT(INDEX+N2)=.TRUE.
5450=       T1=AT(INDEX) $ T3=AT(INDEX+N)
5460=       Z1=AZ(INDEX) $ Z3=AZ(INDEX+N)
5470=       T2=(T1+T3)*.5 $ AT(INDEX+N2)=T2
5480=       TIME=SECOND(CP) $ Z2=Z(T2)
5490=       ZTIME=ZTIME+(SECOND(CP)-TIME) $ NZEVALU=NZEVALU+1
5500=       IF(ABS(Z2).GT.EPS)GOTO 30
5510=       CALL SRCH2B(T1,Z1,T3,NFOUND,INTVAL,7,0)
5520=       IF(NFOUND.EQ.2)RETURN
5530=       WRITE(1,20)T1,T3
5540= 20    FORMAT(* WE ARE IN PARA BETWEEN T1=*,F20.6,* AND T2=*,F20.6,/,
5550=      $* WE FOUND A DOUBTFULL ZT AT T2=(T1+T3)/2 *,/,
5560=      $* WE CALLED SRCH2B FOR HELP BUT EVEN THEN NO ZERO WAS FOUND *,/,
5570=      $* WE ADD 2 ZEROS TO NEWOO AND CONTINUE *,/)
5580=       GOTO 35
5590= 30    AZ(INDEX+N2)=Z2
5600=       IF(Z2*Z1.GT.0.)GOTO 40
5610=C.....TWO ZEROS FOUND.
5620= 35    NFOUND=2 $ RETURN
5630=C.....IN THE FOLLOWING LINES OF THIS ROUTINE WE INSPECT THE
5640=C.....GRAPH OF ABS(Z(T)) BY MEANS OF A PARABOLIC APPROXIMATION.
5650= 40    Z1=ABS(Z1) $ Z2=ABS(Z2) $ Z3=ABS(Z3)
5660=       IF(Z2.GE.AMIN1(Z1,Z3))GOTO 70
5670=       CONVEX=.TRUE.
5680=C.....ALTHOUGH CONVEX=.TRUE. NOW, WE DON'T CALL SRCH2B YET !
5690=C.....WE EXPECT PARA TO FIND THE MISSING TWO.
5700=       IF(Z1.LT.Z3)GOTO 60
5710= 50    ACCEPT(INDEX+N2)=.FALSE. $ INDOUT=INDEX+N2 $ RETURN
5720= 60    ACCEPT(INDEX)=.FALSE. $ INDOUT=INDEX $ RETURN
5730= 70    IF(Z2.LT.(Z1+Z3)*.5)GOTO 90
5740= 80    INDOUT=INDEX+N $ RETURN
5750= 90    X2=T2-T1 $ X3=T3-T1
5760=       A=(X3*(Z2-Z1)-X2*(Z3-Z1))/X2/X3/(X2-X3)
```

```
5770=       B=(Z2-Z1)/X2-A*X2
5780=       RMU=-B*.5/A $ XMIDP=X3*.5 $ HALF=ABS(XMIDP)
5790=       XDIST=ABS(RMU-XMIDP)
5800=       IF(XDIST.GT.HALF)GOTO 80
5810=       IF(ABS(RMU).GT.ABS(X2))50,60
5820=       END
5830=
5840=       SUBROUTINE SRCH2B(A,ZA,B,NFOUND,INTVAL,NCYCLES,IOPTION)
5850=C.....WE TRY TO FIND 2 SIGN CHANGES OF Z(T) IN THE INTERVAL (A,B).
5860=C.....SRCH2B IS CALLED ONLY IF A < B AND Z(A)*Z(B) > 0.
5870=       COMMON/BLOC5/EPS,EPSDBLE,NZEVALU
5880=       COMMON/BLOC6/ZTIME
5890=       H=(B-A)*.25 $ TMIDPNT=(A+B)*.5
5900=C.....NOTE THAT IN THIS ROUTINE THE ARGUMENT OF Z(T) ZIGZAGS
5910=C.....ACCORDING TO A PATTERN SUCH AS ... 9 7 5 3 1 0 2 4 6 8 10...
5920=       IF(IOPTION.EQ.0)GOTO 40
5930=C.....IF(IOPTION.NE.0) WE ALSO COMPUTE Z(TMIDPNT). ELSE WE DON'T.
5940=       T=TMIDPNT $ TIME=SECOND(CP) $ ZT=Z(T)
5950=       ZTIME=ZTIME+(SECOND(CP)-TIME) $ NZEVALU=NZEVALU+1
5960=       IF(ABS(ZT).GT.EPS)GOTO 30
5970=       TREFER=T-H $ CALL COMPZ(ZT,TREFER,T)
5980= 30    IF(ZA*ZT.GT.0.)GOTO 40
5990=C.....TWO ZEROS FOUND.
6000=       NFOUND=2 $ INTVAL=1 $ RETURN
6010= 40    IDEPTH=2 $ H=2.*H
6020=       DO 100 I=1,NCYCLES
6030=       IDEPTH=2*IDEPTH $ LASTJ=IDEPTH/2-1 $ H=H*.5
6040=       DO 90 J=1,LASTJ,2
6050=       STEP=J*H $ T=TMIDPNT-STEP $ TIME=SECOND(CP) $ ZT=Z(T)
6060=       ZTIME=ZTIME+(SECOND(CP)-TIME) $ NZEVALU=NZEVALU+1
6070=       IF(ABS(ZT).GT.EPS)GOTO 50
6080=       TREFER=T-H $ CALL COMPZ(ZT,TREFER,T)
6090= 50    IF(ZA*ZT.GT.0.)GOTO 70
6100=C.....TWO ZEROS FOUND.
6110=       NFOUND=2 $ INTVAL=1 $ RETURN
6120= 70    T=TMIDPNT+STEP $ TIME=SECOND(CP) $ ZT=Z(T)
6130=       ZTIME=ZTIME+(SECOND(CP)-TIME) $ NZEVALU=NZEVALU+1
6140=       IF(ABS(ZT).GT.EPS)GOTO 80
6150=       TREFER=T-H $ CALL COMPZ(ZT,TREFER,T)
6160= 80    IF(ZA*ZT.GT.0.)GOTO 90
6170=C.....TWO ZEROS FOUND.
6180=       NFOUND=2 $ INTVAL=2 $ RETURN
6190= 90    CONTINUE
6200= 100   CONTINUE
6210=       NFOUND=0
6220=       RETURN
6230=       END
6240=
6250=       SUBROUTINE SRCH2C(A,B,ZA,NFOUND,NCYCLES)
6260=C.....WE SEARCH FOR TWO ZEROS IN [ A,B ]. SEARCH DIRECTION: A --> B.
6270=C.....IN THIS ROUTINE WE ALWAYS HAVE Z(A)*Z(B) > 0.
6280=       COMMON/BLOC5/EPS,EPSDBLE,NZEVALU
6290=       COMMON/BLOC6/ZTIME
6300=       H=(B-A)/2. $ K=2
6310=       DO 30 I=1,NCYCLES
6320=       H=H/2. $ K=2*K $ LASTJ=K-1
6330=       DO 20 J=1,LASTJ,2
6340=       T=A+J*H $ TIME=SECOND(CP) $ ZT=Z(T)
6350=       ZTIME=ZTIME+(SECOND(CP)-TIME) $ NZEVALU=NZEVALU+1
6360=       IF(ABS(ZT).GT.EPS)GOTO 10
6370=       TREFER=T-H $ CALL COMPZ(ZT,TREFER,T)
6380= 10    IF(ZT*ZA.GT.0.)GOTO 20
6390=       NFOUND=2 $ RETURN
6400= 20    CCNTINUE
```

```
6410= 30   CONTINUE
6420=      NFOUND=0
6430=      RETURN
6440=      END
6450=
6460=      SUBROUTINE SRCH3(A,ZA,B,NFOUND,IDEPTH)
6470=C.....WE TRY TO FIND 3 SIGN CHANGES OF Z(T) IN THE INTERVAL (A,B),
6480=C.....WHERE A AND B ARE GRAM POINTS. SRCH3 IS CALLED ONLY IN CASE
6490=C.....Z(A)*Z(B) < 0. THIS ROUTINE IS ESSENTIALLY DUE TO SH.LEHMAN.
6500=      COMMON/BLOC5/EPS,EPSDBLE,NZEVALU
6510=      COMMON/BLOC6/ZTIME
6520=      H=(B-A)/IDEPTH
6530=      ZO=ZA $ L=0 $ JJ=IDEPTH-1
6540=      DO 10 J=1,JJ,2
6550=      T=A+J*H $ TIME=SECOND(CP) $ ZT=Z(T)
6560=      ZTIME=ZTIME+(SECOND(CP)-TIME) $ NZEVALU=NZEVALU+1
6570=      IF(ABS(ZT).GT.EPS)GOTO 5
6580=      TREFER=T-H $ CALL COMPZ(ZT,TREFER,T)
6590= 5    IF(ZO*ZT.GT.0.)GOTO 10
6600=C.....ONE ZERO FOUND.
6610=      L=L+1 $ ZO=ZT
6620=      IF(L.LT.2)GOTO 10
6630=      NFOUND=3 $ RETURN
6640= 10   CONTINUE
6650=      NFOUND=L
6660=      RETURN
6670=      END
6680=
6690=      REAL FUNCTION Z(T)
6700=      IMPLICIT DOUBLE (D)
6710=C.....THE RIEMANN-SIEGEL FORMULA (ON SIGMA=1/2) IN SINGLE-PRECISION.
6720=C*************************************************************************
6730=C                    **************************
6740=C                          *****
6750=      COMMON/MCHZDZ1/DLN(3700)
6760=      COMMON/MCHZDZ2/SQRTINV(3700),PREPCOS(8200),PREPDIF(8200)
6770=C.                         *****
6780=C                    **************************
6790=C*************************************************************************
6800=      COMMON/BLOC1/DPI,DPIINV,DTWOPI,DTWOPIN,DPISL8
6810=      COMMON/BLOC3/GRID,GRIDIN
6820=C.....GRIDIN IS USED ONLY IN CASE COMPASS-ZFUNC IS REPLACED BY ITS
6830=C.....FORTRAN EQUIVALENT GIVEN BELOW ( IN COMMENT LINES ).
6840=      DT=T $ DTAU=DT*DTWOPIN
6850=C.....WE ASSUME DTAU TO BE EXACT. THE ERROR IN T IS ACCOUNTED FOR
6860=C.....IN THE ERROR ANALYSIS.
6870=      DRHO=DSQRT(DTAU) $ RHOINV=1.D0/DRHO
6880=      M=IDINT(DRHO)
6890=C.....WE NOW DETERMINE M SUCH THAT M**2 <= DTAU < (M+1)**2.
6900=      IF(M*M.GT.DTAU)M=M-1
6910=C.....THIS MIGHT HAPPEN IF DTAU HAS THE FORM (K**2)*(1-EPSILON),M=K.
6920=      IF(DTAU.GE.(M+1)*(M+1))M=M+1
6930=C.....THIS MIGHT HAPPEN IF DTAU HAS THE FORM (K**2)*(1+EPSILON),M=K-1.
6940=      DP=DRHO-M $ RKSI=1.D0-2.D0*DP $ RKSISQ=RKSI*RKSI
6950=C.....HERE WE HAVE USED THE NOTATION OF HASELGROVE AND MILLER.
6960=      DTH=DTHETA(DT) $ TETAMOD=DTH-IDINT(DTH*DTWOPIN)*DTWOPI
6970=      SUM=COS(TETAMOD)
6980=      SUM=ZFUNC(M,T,TETAMOD)+SUM
6990=C.....THE PRECEDING "COMPASS EVALUATION" OF THE MAIN SUM IN Z(T) IS
7000=C.....EQUIVALENT TO THE FOLLOWING LOOP:
7010=C.....DO 10 I=2,M
7020=C.....D1=DT*DLN(I)
7030=C.....TLNIMOD=D1-IDINT(D1*DTWOPIN)*DTWOPI
7040=C.....X=ABS(TETAMOD-TLNIMOD) $ XGRIDIN=X*GRIDIN
```

```
7050=C.....J=INT(XGRIDIN) $ Q=XGRIDIN-J
7060=C.....ZSUM=ZSUM+(PREPCOS(J+1)+Q*PREPDIF(J+1))*SQRTINV(I)
7070=C10   CONTINUE
7080=C.....           ERROR TERM C0
7090=      E0=+.00000000000009
7100=      E0=(E0*RKSISQ-.00000000000038)*RKSISQ-.00000000001633
7110=      E0=(E0*RKSISQ-.00000000003391)*RKSISQ+.00000000178811
7120=      E0=(E0*RKSISQ+.00000001235793)*RKSISQ-.00000010354847
7130=      E0=(E0*RKSISQ-.00000143272516)*RKSISQ+.00000046556125
7140=      E0=(E0*RKSISQ+.00007943300880)*RKSISQ+.00029705353733
7150=      E0=(E0*RKSISQ-.00162372532314)*RKSISQ-.01356762197010
7160=      E0=(E0*RKSISQ-.01360502604767)*RKSISQ+.13237657548034
7170=      E0=(E0*RKSISQ+.43724046807752)*RKSISQ+.38268343236509
7180=C.....           ERROR TERM C1
7190=      E1=-.00000000000001
7200=      E1=(E1*RKSISQ-.00000000000012)*RKSISQ+.00000000000158
7210=      E1=(E1*RKSISQ+.00000000002344)*RKSISQ-.00000000007785
7220=      E1=(E1*RKSISQ-.00000000270964)*RKSISQ-.00000000446709
7230=      E1=(E1*RKSISQ+.00000018334734)*RKSISQ+.00000096413225
7240=      E1=(E1*RKSISQ-.00000598024259)*RKSISQ-.00005922748702
7250=      E1=(E1*RKSISQ+.00001320794062)*RKSISQ+.00146478085780
7260=      E1=(E1*RKSISQ+.00331075976086)*RKSISQ-.00987106629906
7270=      E1=(E1*RKSISQ-.03849125048224)*RKSISQ-.01378477342635
7280=      E1=(E1*RKSISQ+.02682510262838)*RKSI
7290=      ERROR=(E1*RHOINV+E0)*SQRT(RHOINV)*((-1)**(M-1))
7300=C.....       END OF ERROR COMPUTATION
7310=      Z=2.*SUM+ERROR
7320=      RETURN
7330=      END
7340=
7350=      DOUBLE FUNCTION DZ(DT)
7360=      IMPLICIT DOUBLE (D)
7370=C.....THIS IS OUR DP VERSION OF THE RIEMANN-SIEGEL FORMULA FOR Z(T).
7380=C.....DZ(DT) IS COMPLETELY IN DOUBLE PRECISION ARITHMETIC.
7390=      COMMON/MCHZDZ1/DLN(3700)
7400=      COMMON/BLOC1/DPI,DPIINV,DTWOPI,DTWOPIN,DPISL8
7410=      COMMON/BLOC8/DC0(30),DC1(30),DC2(30),DC3(30)
7420=      DTAU=DT*DTWOPIN $ DTAUINV=1.D0/DTAU
7430=C.....WE ASSUME DTAU TO BE EXACT. THE ERROR IN DT IS ACCOUNTED FOR
7440=C.....IN THE ERROR ANALYSIS
7450=      DRHO=DSQRT(DTAU) $ DRHOINV=1.D0/DRHO
7460=      M=IDINT(DRHO)
7470=C.....WE NOW DETERMINE M SUCH THAT M**2 <= DTAU < (M+1)**2.
7480=      IF(M*M.GT.DTAU)M=M-1
7490=C.....THIS MIGHT HAPPEN IF DTAU HAS THE FORM (K**2)*(1-EPSILON),M=K.
7500=      IF(DTAU.GE.(M+1)*(M+1))M=M+1
7510=C.....THIS MIGHT HAPPEN IF DTAU HAS THE FORM (K**2)*(1+EPSILON),M=K-1.
7520=      DKSI=2.D0*(DRHO-M)-1.D0 $ DKSISQ=DKSI*DKSI
7530=      DTETADT=DTHETA(DT)
7540=      DSUM=0.D0
7550=      DO 200 I=1,M
7560=      DI=I $ DSUM=DSUM+DCOS(DTETADT-DT*DLN(I))/DSQRT(DI)
7570= 200  CONTINUE
7580=C.....COMPUTATION OF THE ERROR TERM IN THE RIEMANN-SIEGEL FORMULA.
7590=C.....HERE WE USE THE NOTATION OF CRARY AND ROSSER.
7600=      DPHI0=DC0(30) $ DPHI1=DC1(30) $ DPHI2=DC2(30) $ DPHI3=DC3(30)
7610=      DO 500 I=1,29
7620=      K=30-I
7630=      DPHI0=DPHI0*DKSISQ+DC0(K)
7640=      DPHI1=DPHI1*DKSISQ+DC1(K)
7650=      DPHI2=DPHI2*DKSISQ+DC2(K)
7660=      DPHI3=DPHI3*DKSISQ+DC3(K)
7670= 500  CONTINUE
7680=      DPHI1=DPHI1*DKSI*DRHOINV
```

```
7690=       DPHI2=DPHI2*DTAUINV
7700=       DPHI3=DPHI3*DKSI*DTAUINV*DRHOINV
7710=       DSIGN=1.D0 $ IF(M/2*2.EQ.M) DSIGN=-1.D0
7720=       DERROR=DSIGN*(DPHI0-DPHI1+DPHI2-DPHI3)/DSQRT(DRHO)
7730=C.....         END OF ERROR COMPUTATION
7740=       DZ=2.D0*DSUM+DERROR
7750=       RETURN
7760=       END
7770=
7780=            IDENT     ZFUNC
7790=            ENTRY     ZFUNC
7800=*
7810=*           SUM = ZFUNC(M, T, TETAMOD)
7820=*
7830=            ENTRY     ZFUNC
7840= ZFUNC      BSS       1
7850=            SB1       1
7860=            SA5       X1              M
7870=            SB2       B1+B1
7880=            SB3       13              AUX SHIFT FOR NGRID=8192
7890=            SB5       PREPCOS
7900=            SB6       SQRTINV+1
7910=            SA2       A1+B1
7920=            SA3       A2+B1
7930=            SB4       X5-2            I, INITIALLY M-2
7940=            SA1       X2              T
7950=            SA2       X3              TETAMOD
7960=            LX5       1               M*2
7970=            SA3       DTWOPIN
7980=            SA4       A3+B1
7990=            FX0       X3*X1           T*PI.H
8000=            DX6       X3*X1           T*PI.H
8010=            FX7       X4*X1           T*PI.T
8020=            FX6       X6+X7
8030=            NX6       X6              NORMALIZE
8040=            DX1       X0+X6           T*PI HEAD
8050=            FX0       X0+X6           T*PI TAIL
8060=            FX7       X2*X3           TETAMOD/TWOPI
8070=            MX6       0               INITIALIZE SUM
8080=            SA4       X5+DLN-2        INITIALIZE A4
8090=            SA5       A4+B1           PREFETCH X5
8100=*
8110=*           IN THE LOOP THE FOLLOWING REGISTERS ARE DEFINED:
8120=*           B1 = 1
8130=*           B2 = 2
8140=*           B3 = 13  (BASE2LOG(GRID))
8150=*           B4 = I   COUNTS DOWN
8160=*           B5 = PREPCOS  FIRST ELEMENT
8170=*           B6 = SQRTINV+1  B6+B4 = SQRTINV(I)
8180=*           A4 = DLN INDEX, A4-1 IS TAIL OF NEXT DLN
8190=*           X0,X1 = T/TWOPI
8200=*           X6 = SUM
8210=*           X7 = TETAMOD/TWOPI
8220=*
8230=* FIRST WE GIVE THE INTENDED LOOP
8240=*
8250=* 1. PICK UP NEXT ELEMENTS FROM DLN
8260=*           SA4       A4-B2
8270=*           SA5       A4+B1
8280=*     NOTE: THESE ELEMENTS ARE ALREADY PREFETCHED AT THE START
8290=*     OF THE ACTUAL LOOP AND THE NEXT ELEMENTS ARE FETCHED NEAR
8300=*     THE END.
8310=*
8320=* 2. DOUBLE PRECISION MULTIPLICATION WITH T/TWOPI
```

```
8330=*          FX2       X4*X1
8340=*          DX3       X4*X0
8350=*          FX4       X4*X0
8360=*          FX5       X5*X0
8370=*          FX3       X3+X2
8380=*          FX2       X5+X3
8390=*          DX5       X2+X4
8400=*          FX4       X2+X4
8410=*     NOTE: THE LAST TWO INSTRUCIONS ARE OMITTED IN THE ACTUAL
8420=*     LOOP, AND THE LAST INSTRUCTION BUT TWO IS CHANGED INTO FX5.
8430=*     THESE INSTRUCTIONS ARE NOT NEEDED SINCE IT IS NOT NECESSARY
8440=*     THAT THE VALUE MOD 1 IS ACTUALLY COMPUTED, AS LONG AS WE
8450=*     FIND TWO VALUES, J AND Q WITH J INTEGER AND J+Q IS THE TRUE
8460=*     VALUE. HOWEVER, WE NEED A SLIGHT EXTENSION OF THE ARRAY WITH
8470=*     PRECOMPUTED COSINES, SINCE THE INDEX MAY BE OUT OF THE RANGE
8480=*     [ 0,8191 ]. HOWEVER, ONLY A FEW ELEMENTS MORE ARE NEEDED.
8490=*
8500=*  3. COMPUTE FRACTION MODULO 1.
8510=*          UX2       B7,X4
8520=*          LX2       B7,X2          INTEGER PART
8530=*          AX2       B7,X2          REPOSITION
8540=*          BX4       X4-X2
8550=*     THE LAST OPERATION IS CORRECT BECAUSE X4 >= 0.
8560=*
8570=*  4. NORMALIZE, ADD HEAD AND TAIL.
8580=*          NX3       X5
8590=*          NX4       X4
8600=*          FX2       X4+X3
8610=*    NOW X2 SHOULD BE < 1. HOWEVER, SEE THE NOTE IN STEP 2.
8620=*
8630=*  5. SUBTRACT FROM TETAMOD.
8640=*          FX2       X7-X2
8650=*     NORMALIZATION IS NOT YET NEEDED.
8660=*
8670=*  6. COMPUTE INTEGER PART AND FRACTIONAL PART OF
8680=*     ABSOLUTE VALUE * 8192.
8690=*          UX5       B7,X2          UNPACK
8700=*          SB7       B7+B3          NEW EXPONENT
8710=*          LX5       B7,X5          INTEGER PART
8720=*          BX4       X2
8730=*          AX4       60             SIGN EXTENDED
8740=*          BX2       X2-X4          ABSOLUTE VALUE
8750=*          BX5       X5-X4          ABSOLUTE VALUE OF INTEGER PART
8760=*          PX2       B7,X2          STORE NEW EXPONENT
8770=*          AX4       B7,X5          REPOSITION INTEGER PART
8780=*          BX4       X2-X4          SUBTRACT
8790=*          NX4       X4             AND NORMALIZE
8800=*     IN THIS CODE WE DON'T FIRST COMPUTE ABSOLUTE VALUE.
8810=*     IF WE DID SO WE MIGHT SAVE ONE INSTRUCTION (TAKING
8820=*     ABSOLUTE VALUE OF INTEGER PART IS NOT NEEDED). HOWEVER,
8830=*     IT IS NOT POSSIBLE TO DO PARALLEL COMPUTATIONS, SO THIS
8840=*     CODE IS FASTER.
8850=*
8860=*  7. PICK UP COSINE AND COSINE DIFFERENCE
8870=*          SA3       X5+CC          COSINE DIFFERENCE
8880=*          SA5       X5+B5          COSINE
8890=*
8900=*  8. COMPUTE INTERPOLATED COSINE.
8910=*          FX3       X3*X4
8920=*          FX3       X3+X5
8930=*          NX3       X3
8940=*
8950=*  9. PICK UP INVERSE OF SQRT(I) AND MULTIPLY.
8960=*          SA2       B4+B6
```

```
8970=*          FX2         X2*X3
8980=*
8990=* 10. ADD TO ACCUMULATIVE SUM.
9000=*          FX6         X6+X2
9010=*          NX6         X6
9020=*
9030=* 11. COUNT AND POSSIBLY JUMP BACK.
9040=*          SB4         B4-B1
9050=*          GE          B4,LOOP
9060=*
9070=* THE ACTUAL LOOP CONTAINS THE INSTRUCTIONS ABOVE BUT JUGGLED
9080=* AROUND TO GET THE UTMOST FROM THE PARALLEL PROCESSING OF THE
9090=* FUNCTIONAL UNITS OF THE CYBER-175.
9100=*
9110= LOOP      BSS         0
9120=* INSTRUCTION    CYCLE     BUSY REGISTERS (CYCLES STILL BUSY)
9130=   FX2 X4*X1      0          X2( 5),X5( 4)                    .** WORD 1
9140=*                 1          X2( 4),X5( 3)                    ..MULTIPLY BUSY
9150=   DX3 X4*X0      2  X3( 5),X2( 3),X5( 2)                     .
9160=*                 3  X3( 4),X2( 2),X5( 1)                     ..MULTIPLY BUSY
9170=   FX4 X4*X0      4  X3( 3),X2( 1),X5( 1),X4( 5)              .
9180=*                 5  X3( 2),               X4( 4)             ..MULTIPLY BUSY
9190=   FX5 X5*X0      6  X3( 1),X5( 5),        X4( 3)             .
9200=   FX3 X3+X2      7  X3( 4),X5( 4),        X4( 2)             .** WORD 2
9210=*                 8  X3( 3),X5( 3),        X4( 1)             .
9220=   UX2 B7,X4      9  X3( 2),X5( 2),X2( 2),B7( 2)              .
9230=*                10  X3( 1),X5( 1),X2( 1),B7( 1)              .
9240=   LX2 B7,X2     11  X2( 2)                                    .
9250=   FX5 X5+X3     12  X2( 1),X5( 4)                             .
9260=   AX2 B7,X2     13  X2( 2),X5( 3)                             .** WORD 3
9270=*                14  X2( 1),X5( 2)                             .
9280=   BX4 X4-X2     15  X4( 2),X5( 1)                             .
9290=   NX3 X5,B0     16  X4( 1),X3( 3),B0( 3)                      .
9300=   NX4 X4,B7     17  X4( 3),X3( 2),B0( 2),B7( 3)               .
9310=*                18  X4( 2),X3( 1),B0( 1),B7( 2)               .
9320=*                19  X4( 1),               B7( 1)              .
9330=   FX2 X4+X3     20  X2( 4)                                    .** WORD 4
9340=*                21  X2( 3)                                    .
9350=*                22  X2( 2)                                    .
9360=*                23  X2( 1)                                    .
9370=   FX2 X7-X2     24  X2( 4)                                    .
9380=*                25  X2( 3)                                    .
9390=*                26  X2( 2)                                    .
9400=*                27  X2( 1)                                    .
9410=   UX5 B7,X2     28  X5( 2),B7( 2)                             .
9420=   BX4 X2        29  X5( 1),B7( 1),X4( 2)                      .
9430=   SB7 B7+B3     30         B7( 2),X4( 1)                      .** WORD 5
9440=   AX4 60        31         B7( 1),X4( 2)                      .
9450=   LX5 B7,X5     32  X5( 2),       X4( 1)                     .
9460=   BX2 X2-X4     33  X5( 1),X2( 2)                            .
9470=   BX5 X5-X4     34  X5( 2),X2( 1)                            .** WORD 6
9480=   PX2 B7,X2     35  X5( 1),X2( 2)                            .
9490=   SA3 X5+CC     36  X3(23),X2( 1),A3( 2)                     .
9500=   AX4 B7,X5     37  X3(22),X4( 2),A3( 1)                     .** WORD 7
9510=   SA5 X5+B5     38  X3(21),X4( 1),X5(23),A5( 2)              .
9520=   BX4 X2-X4     39  X3(20),X4( 2),X5(22),A5( 1)              .
9530=   SA2 B4+B6     40  X3(19),X4( 1),X5(21),X2(23),A2( 2)       .
9540=   NX4 X4        41  X3(18),X4( 3),X5(20),X2(22),A2( 1),B0(.** WORD 8
9550=*                42  X3(17),X4( 2),X5(19),X2(21),    B0( 2).
9560=*                43  X3(16),X4( 1),X5(18),X2(20),    B0( 1).
9570=*                44  X3(15),       X5(17),X2(19)              .
9580=*                45  X3(14),       X5(16),X2(18)              .
9590=*                46  X3(13),       X5(15),X2(17)              .
9600=*                47  X3(12),       X5(14),X2(16)              .
```

```
9610=*                48  X3(11),       X5(13),X2(15)                .
9620=*                49  X3(10),       X5(12),X2(14)                .
9630=*                50  X3( 9),       X5(11),X2(13)                .
9640=*                51  X3( 8),       X5(10),X2(12)                .
9650=*                52  X3( 7),       X5( 9),X2(11)                .
9660=*                53  X3( 6),       X5( 8),X2(10)                .
9670=*                54  X3( 5),       X5( 7),X2( 9)                .
9680=*                55  X3( 4),       X5( 6),X2( 8)                .
9690=*                56  X3( 3),       X5( 5),X2( 7)                .
9700=*                57  X3( 2),       X5( 4),X2( 6)                .
9710=*                58  X3( 1),       X5( 3),X2( 5)                .
9720=  FX3 X3*X4      59  X3( 5),       X5( 2),X2( 4)                .
9730=  SA4 A4-B2      60  X3( 4),X4(23),X5( 1),X2( 3),A4( 2)         .
9740=*                61  X3( 3),X4(22),       X2( 2),A4( 1)         .
9750=*                62  X3( 2),X4(21),       X2( 1)                .
9760=*                63  X3( 1),X4(20)                              .
9770=  FX3 X5+X3      64  X3( 4),X4(19)                              .
9780=  SA5 A4+B1      65  X3( 3),X4(18),X5(23),A5( 2)                .** WORD 9
9790=  SB4 B4-B1      66  X3( 2),X4(17),X5(22),A5( 1),B4( 2)         .
9800=*                67  X3( 1),X4(16),X5(21),       B4( 1)         .
9810=  NX3 X3         68  X3( 3),X4(15),X5(20),B0( 3)                .
9820=*                69  X3( 2),X4(14),X5(19),B0( 2)                .
9830=*                70  X3( 1),X4(13),X5(18),B0( 1)                .
9840=  FX2 X2*X3      71  X2( 5),X4(12),X5(17)                       .
9850=*                72  X2( 4),X4(11),X5(16)                       .
9860=*                73  X2( 3),X4(10),X5(15)                       .
9870=*                74  X2( 2),X4( 9),X5(14)                       .
9880=*                75  X2( 1),X4( 8),X5(13)                       .
9890=  FX6 X6+X2      76  X6( 4),X4( 7),X5(12)                       .** WORD 10
9900=*                77  X6( 3),X4( 6),X5(11)                       .
9910=*                78  X6( 2),X4( 5),X5(10)                       .
9920=*                79  X6( 1),X4( 4),X5( 9)                       .
9930=  NX6 X6         80  X6( 3),X4( 3),X5( 8),B0( 3)                .
9940=  GE  B4,LOOP    81  X6( 2),X4( 2),X5( 7),B0( 2)                .
9950=*                82  X6( 1),X4( 1),X5( 6),B0( 1)                .
9960=*                83*               X5( 5)                       .
9970=*
9980=*          END LOOP
9990=*
10000=          EQ        ZFUNC
10010=*
10020=          USE       /MCHZDZ1/
10030= DLN      BSS       3700*2
10040=*
10050=          USE       /MCHZDZ2/
10060= SQRTINV  BSS       3700
10070= PREPCOS  BSS       8200
10080= CC       BSS       8200
10090=*
10100=          USE       /BLOC1/
10110= DPI      BSS       2
10120= DPIINV   BSS       2
10130= DTWOPI   BSS       2
10140= DTWOPIN  BSS       2
10150= DPISL8   BSS       2
10160=*
10170=          END
10180=
10190=      SUBROUTINE STATIST(K)
10200=      COMMON/MCH1/INTRVAL(10,10)
10210=      COMMON/MCH2/NBLOCL(10)
10220=      DIMENSION NAUX(10)
10230=      WRITE(1,10)
10240= 10   FORMAT(*1CUMULATIVE STATISTICS (ALSO WRITTEN ON PF RIESTAT) *)
```

```
10250=       REWIND 2
10260=       READ(2,20)N1,N2
10270= 20    FORMAT(2I12)
10280=       WRITE(3,20)N1,K
10290=       WRITE(1,30)N1,K
10300= 30    FORMAT(* RANGE:*,2I12)
10310=       DO 60 I=1,10
10320=       READ(2,50)IH,NTOT,(NAUX(J),J=1,I)
10330=       NTOT=NTOT+NBLOCL(I)
10340=       DO 40 L=1,I
10350=       NAUX(L)=NAUX(L)+INTRVAL(I,L)
10360= 40    CONTINUE
10370=       WRITE(3,50)IH,NTOT,(NAUX(J),J=1,I)
10380=       WRITE(1,50)IH,NTOT,(NAUX(J),J=1,I)
10390= 50    FORMAT(I3,3I10,I9,I8,I7,I6,4I4)
10400= 60    CONTINUE
10410=       RETURN
10420=       END
```

6. REFERENCES

[1] BACKLUND, R., *Sur les zeros de la fonction* $\zeta(s)$ *de Riemann*, C.R. Acad. Sci. Paris, 158 (1914) pp. 1979-1982.

[2] BRENT, R.P., *A Fortran multiple precision arithmetic package*, ACM Trans. Math. Software, 4 (1978) pp. 57-70.

[3] BRENT, R.P., *On the Zeros of the Riemann Zeta Function in the Critical Strip*, Math. Comp., 33 (1979) pp. 1361-1372.

[4] BRENT, R.P., & L. SCHOENFELD, *Private communications*, March 5, 1981 and July 17, 1981.

[5] BRENT, R.P., *Private communication*, March 26, 1981.

[6] CONTROL DATA, *Fortran common library mathematical routines reference manual*, Revision C, 1978.

[7] CRARY, F.D. & J.B. ROSSER, *High Precision Coefficients Related to the Zeta Function*, MRC Technical Summary Report #1344, Univ. of Wisconsin, Madison (1975).

[8] EDWARDS, H.M., *Riemann's Zeta Function*, Academic Press, New York (1974).

[9] GABCKE, W., *Neue Herleitung und explizite Restabschätzung der Riemann-Siegel-Formel*, Dissertation, Universität Göttingen, 1979.

[10] GRAM, J., *Sur les zéros de la fonction* $\zeta(s)$ *de Riemann*, Acta Math., 27 (1903) pp. 289-304.

[11] HASELGROVE, C.B. & J.C.P. MILLER, *Tables of the Riemann Zeta Function*, Roy. Soc. Math. Tables No. 6, Cambridge Univ. Press, New York (1960).

[12] HUTCHINSON, J.I., *On the roots of the Riemann zeta-function*, Trans. Amer. Math. Soc., 27 (1925) pp. 49-60.

[13] LEHMAN, R.S., *Separation of zeros of the Riemann zeta-function*, Math. Comp., 20 (1966) pp. 523-541.

[14] LEHMER, D.H., *On the roots of the Riemann zeta function*, Acta Math., 95 (1956) pp. 291-298.

[15] LEHMER, D.H., *Extended computation of the Riemann zeta function*, Mathematika, 3 (1956) pp. 102-108.

[16] MELLER, N.A., *Computations connected with the check of Riemann's Hypothesis*, Doklad. Akad. Nauk SSSR, 123 (1958) pp. 246-248. (Russian)

[17] PARLETT, B.N., *The symmetric eigenvalue problem*, Prentice-Hall, 1980.

[18] RIEMANN, B., *Gesammelte Werke*, Dover, New York, 1953.

[19] ROSSER, J.B., J.M. YOHE & L. SCHOENFELD, *Rigorous computation and the zeros of the Riemann zeta-function*, Proc. IFIP Congress, Edinburgh, 1968.

[20] TITCHMARSH, E.C., *The zeros of the Riemann zeta function*, Proc. Roy. Soc. London, 151 (1935) pp. 234-255, 157 (1936) pp. 261-263.

[21] TITCHMARSH, E.C., *The theory of the Riemann Zeta-function*, Oxford, Clarendon Press, 1951.

[22] WILKINSON, J.H., *Rounding errors in algebraic processes*, Prentice-Hall, 1963.

E R R A T A in R E P O R T NW 113/81

(Rigorous high speed separation of zeros of Riemann's zeta function,
by J. van de Lune, H.J.J. te Riele and D.T. Winter)

page	line(s)	instead of	read	remark
3	7↓	definitions.	definitions. Any interval $[g_j, g_{j+1})$ is called a Gram interval.	
3	9-10↓	For L = 1 ... interval.		skip
3	12↓	(g_j)	$Z(g_j)$	
4	1↓	(	[	
4	12↑	(	[	
4	11↑	(	[	
5	12↓	:=	=:	
6	7↓	=	:=	
6	15↓	) cos	)-cos	
6	11↑	'S	's	
7	6↑	(3.3),	(3.3)),	
8	1↓	[5]	[3]	
10	4↓	value-	value	
10	10↓	(t	t	
10	16↓	(t	t	
10	19↓	$1+^{(130)}$	$1+\varepsilon^{(130)}$	
10	12↑	$\frac{\pi}{8}($	$\frac{\pi}{8})($	
11	13↓	$0 \le k \le 8192$	$0 \le k \le 8193$	
12	4↓	10^{15}	10^{-15}	
14	2↓	In Table 4.1	We call a Gram block B_j of type (L,k) if its length is L and if the "missing two zeros" are located in $[g_{j+k-1}, g_{j+k})$. In Table 4.1	
14	2-3↓	j	L	3 x
14	3↓	]	)	
14	4↓	j	L	
14	8↓	j	L	
14	14↓	g_{j+l-1}	g_{j+L-1}	
14	18↑	j	L	3 x
14	17↑	]	)	
14	15↑	j	L	
15	1↑	(	[	2 x
16	3↓	(	[	
16	9↑	(	[	
17	14↓	8193	8194	