

**stichting
mathematisch
centrum**



AFDELING NUMERIEKE WISKUNDE
(DEPARTMENT OF NUMERICAL MATHEMATICS)

NN 26/82

JUNI

J. VAN DE LUNE & H.J.J. TE RIELE

RIGOROUS HIGH SPEED SEPARATION OF ZEROS OF RIEMANN'S ZETA FUNCTION, 11

kruislaan 413 1098 SJ amsterdam

Printed at the Mathematical Centre, 413 Kruislaan, Amsterdam.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O.).

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

by

J. van de Lune & H.J.J. te Riele

ABSTRACT

This is an intermediate report for the purpose of announcing that the first 250,000,000 zeros of the Riemann zeta function $\zeta(s)$ in the critical strip are simple and lie on the line $\text{Re}(s) = \frac{1}{2}$. This extends our previous result that the first 200,000,001 zeros have this property. A listing of the FORTRAN/COMPASS program is included.

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

1. INTRODUCTION

This is an intermediate report for the purpose of announcing that *the first 250,000,000 zeros of the Riemann zeta function $\zeta(s)$ in the critical strip are simple and lie on the line $\text{Re}(s) = \frac{1}{2}$* . This result was obtained by extending the computations (up to zero # 200,000,001) described in report NW 113 ([3]).

In section 2 we present tables similar to those given in section 4 of NW 113, this time for the range $[g_{200,000,000}, g_{249,999,999})$. A listing of the complete FORTRAN/COMPASS program by which the zeros in this range were separated is given in section 3. Although there are only a few changes (most of them being explained in section 2) compared with the program given in NW 113, we have decided to give the full listing for the sake of completeness.

We hope to extend our computations up to zero # 300,000,000 in the near future.

2. THE TABLES

In Table 2.1 we list the number of Gram blocks of type (L, k) , $1 \leq L \leq 8$, $1 \leq k \leq L$, in the interval $[g_{200,000,000}, g_{249,999,999})$, as actually counted by our program. These counts are *exact* now as opposed to the counts given in NW 113. This was realized by immediately computing $Z_B(g_n)$ in case $|Z_A(g_n)|$ was too small (rather than computing $Z_A(g_n - \delta)$ for a few small values of δ , as we did in NW 113).

On the lines with $L=2$ and $L=3$ in Table 2.1 we also list the number of exceptions to Rosser's rule of length 2 (with "0 0"-zero pattern) and of length 3 (with "0 1 0"-zero pattern), respectively. Moreover, on the line with $L=2$ we also mention the two blocks with "2 2"-zero pattern found in relation to the exceptions of type 5 and 6 (cf. Table 2.3). Note the two exceptions of length 3. Previously, exceptions of length 3 were implicitly given by KARKOSCHKA & WERNER [2] (viz. $B_{1,089,751,985}$ and $B_{10,008,051,629}$), although these are far beyond the range covered by our systematic search. The entries in parentheses are the approximate percentages with respect to

the total number of blocks of length L , given in the final column.

For $2 \leq L \leq 5$ our strategy of finding the missing two zeros was the same as in NW 113. However, in view of the counts in Table 4.1 of NW 113, this strategy was adapted for Gram blocks of length $L \geq 6$. (The search for the missing two was started now in the Gram interval (g_{n+1}, g_{n+2}) or (g_{n+L-2}, g_{n+L-1}) of the Gram block $B_n = [g_n, g_{n+L})$ rather than in (g_n, g_{n+1}) or (g_{n+L-1}, g_{n+L}) .) Table 2.1 is our justification of this slightly adapted strategy.

Table 2.1

Number of Gram blocks of type (L, k) , $1 \leq L \leq 8$, $1 \leq k \leq L$, in the interval $[g_{200,000,000}, g_{249,999,999})$

$k \rightarrow$									Total
$\downarrow L$	1	2	3	4	5	6	7	8	
1	34,837,572								34,837,572
2	2,642,082 (50)	2,640,298 (50)	63 blocks with 0 0 zero-pattern 2 blocks with 2 2 zero-pattern						5,282,445
3	560,044 (47)	62,216 (5)	560,871 (47)	2 blocks with 0 1 0 zero-pattern					1,183,133
4	104,629 (46)	9,870 (4)	9,670 (4)	104,341 (46)					228,510
5	9,815 (39)	2,093 (8)	1,054 (4)	2,151 (9)	9,757 (39)				24,870
6	204 (14)	424 (28)	126 (8)	133 (9)	411 (27)	205 (14)			1,503
7	0	29	21	5	14	33	0		102
8	0	0	1 ^{*)}	0	0	1 ^{*)}	0	0	2

^{*)} viz. B_n , for $n = 218,164,689$ and $211,643,971$

In Table 2.2 we present the 65 newly found exceptions to Rosser's rule in the range $[g_{200,000,000}, g_{249,999,999})$, including the local extreme values of $S(t)$. (The definitions of the various types are implicitly given in Table 2.3.)

Table 2.2

The exceptions to Rosser's rule of length 2 and 3

in the interval $[g_{200,000,000}, g_{249,999,999})$

Exceptions of length 2, zero pattern "0 0"

notation: n (type) extreme $S(t)$,

where n is the index of the Gram block $B_n = [g_n, g_{n+2})$

containing no zeros

201007375(1)	-2.002900	216274605(2)	2.052279	231810024(1)	-2.026611
201030606(2)	2.111895	216957121(2)	2.032421	232838063(2)	2.022488
201184291(2)	2.001518	217323208(1)	-2.013607	234389089(2)	2.106429
201685414(5)	-2.016715	218799264(1)	-2.040304	235588194(1)	-2.001915
202762876(2)	2.011439	218803558(2)	2.013448	236645695(1)	-2.089639
202860958(2)	2.018888	219735146(1)	-2.026815	236962877(2)	2.023259
203832578(2)	2.063611	219830063(2)	2.015232	237516725(4)	2.108817
205880544(1)	-2.017679	219897904(1)	-2.081132	240004911(1)	-2.000249
206357111(1)	-2.031216	221205545(1)	-2.014535	240221307(2)	2.096293
207159768(2)	2.033954	223601929(1)	-2.101580	241549003(1)	-2.036151
207167344(2)	2.029320	223907077(2)	2.007094	241729717(1)	-2.025503
207669541(2)	2.020740	223970397(1)	-2.028754	241743685(2)	2.070155
208053426(1)	-2.073357	224874046(6)	2.022804	243780201(2)	2.025648
208110028(2)	2.031212	225291157(1)	-2.152675	243801317(1)	-2.020358
209513827(2)	2.023920	227481734(1)	-2.018298	244122072(1)	-2.035325
212623522(1)	-2.010194	228006443(2)	2.023042	244691225(2)	2.018927
213841715(1)	-2.024334	228357900(1)	-2.022758	244841577(1)	-2.053021
214012333(1)	-2.010937	228386399(1)	-2.008899	245813461(1)	-2.035731
214073567(1)	-2.009287	228907446(1)	-2.018338	246299475(1)	-2.001039
215170601(2)	2.007728	228984553(2)	2.032004	246450177(2)	2.116655
215881040(2)	2.021267	229140286(2)	2.000109	249069349(1)	-2.020698

Exceptions of length 3, zero pattern "0 1 0"

$B_{207,482,540}$ preceded by a Gram block of length 1 with 3 zeros, extreme $S(t) = 2.000431$

$B_{241,389,213}$ followed by a Gram block of length 1 with 3 zeros, extreme $S(t) = -2.010430$

Table 2.3 contains the frequencies of occurrence of the various types of exceptions to Rosser's rule in the range $[g_{200,000,000}, g_{249,999,999})$.

Table 2.3

Various types of exceptions to Rosser's rule and their frequencies in

$[g_{200,000,000}, g_{249,999,999})$

Exceptions of length 2

Gram block of length 2 without any zeros								type	frequency
g_{n-2}	g_{n-1}	g_n	g_{n+1}	g_{n+2}	g_{n+3}	g_{n+4}			
		0	0	3				1	32
	3	0	0					2	28
		0	0	4	0			3	0
0	4	0	0					4	1
		0	0	2	2			5	1
2	2	0	0					6	1

Exceptions of length 3

Gram block of length 3 containing 1 zero						frequency
g_{n-1}	g_n	g_{n+1}	g_{n+2}	g_{n+3}	g_{n+4}	
	3	0	1	0		1
		0	1	0	3	1

Finally, from Tables 2.1 and 2.2 we have counted the following numbers of Gram intervals in $[g_{200,000,000}, g_{249,999,999})$ containing exactly m ($0 \leq m \leq 4$) zeros. Note the occurrence of the *second* Gram interval containing *four* zeros, viz., $G_{237,516,724}$.

m	0	1	2	3	4
#	6,808,880	36,470,555	6,632,249	88,312	1
%	13.6	72.9	13.3	0.2	0.0

3. THE(NEW)PROGRAM

In this section we present the source text of our new program by which the zeros in the range $[g_{200,000,000}, g_{249,999,999})$ were separated. The *eco-rate* (i.e. the average number of Z-evaluations needed to separate one zero) varied between 1.18 and 1.19. Roughly 100 Z-evaluations per second CPU-time were done on the CYBER 175. The separation of the zeros in the whole range $[g_{200,000,000}, g_{249,999,999})$ required about 175 hours CPU-time.

```

PROGRAM RHCHECK (OUTPUT,TAPE1=OUTPUT,STATIN,TAPE2=STATIN,          90
$          STATOU,TAPE3=STATOU)                                     100
C.....BY MEANS OF THIS PROGRAM ONE MAY VERIFY THE RIEMANN HYPOTHESIS 110
C.....IN A GIVEN RANGE. IT WAS WRITTEN IN OCTOBER 1981 AT THE        120
C.....MATHEMATICAL CENTRE AT AMSTERDAM BY: J.VAN DE LUNE, H.J.J.TE RIELE 130
C.....AND D.T.WINTER. IT IS AN IMPROVED VERSION OF THE PROGRAM PUBLISHED 140
C.....IN THE MATHEMATICAL CENTRE REPORT NW 113/81.                   150
C*****                                                             160
C          *****                                                             170
C          *****                                                             180
C.....SET THE DIMENSIONS OF DLN(.) AND SQRTINV(.) PROPERLY !          190
C.....THESE TWO ARRAYS MUST BE SUFFICIENTLY LONG FOR THE EVALUATION 200
C.....OF Z(.) IN THE RANGE TO BE INVESTIGATED.                       210
C.....THESE DIMENSIONS MUST BE AT LEAST INT(SQRT(TMAX/TWOPI)).        220
      IMPLICIT DOUBLE (D)                                              230
      COMMON/MZDZ1/DLN(4025)                                           240
      COMMON/MZDZ2/SQRTINV(4025),PREPCOS(8200),PREPDIF(8200)          250
C.....THROUGHOUT THIS PROGRAM THE ARRAYS PREPCOS(.) AND PREPDIF(.)    260
C.....MUST HAVE DIMENSIONS AT LEAST 8194 AND 8193 RESPECTIVELY.        270
C          *****                                                             280
C          *****                                                             290
C*****                                                             300
      COMMON/S2PARA/DARRT(513),ARRZ(513),NEXT(512),ACCEPT(512)        310
      COMMON/BLOCA/NBLOCL(10)                                          320
      COMMON/BLOC0/INTRVAL(10,10)                                       330
      COMMON/BLOC1/DPI,DPIINV,DTWOPI,DTWOPIN,DPISL8                   340
      COMMON/BLOC2/PI,TWOPI                                             350
      COMMON/BLOC3/GRID,GRIDIN                                          360
      COMMON/BLOC4/DCNST1,DCNST2,DCNST3                                 370
      COMMON/BLOC5/EPS,EPSDBLE,ZTIME,NZEVALU                          380
      COMMON/BLOC6/NSHIFTS                                              390
      COMMON/BLOC7/DC0(30),DC1(30),DC2(30),DC3(30)                   400
      COMMON/BLOC8/FCONCAV                                              410
      COMMON/BLOC9/DCNST0                                              420
      DIMENSION DTDIM(11),ZDIM(11)                                     430
      LOGICAL ACCEPT,STAT                                              440
      TOTTIME=SECOND(CP)                                               450
C AS TO THE VALUE OF NPREP, SEE THE COMMENT LINES AFTER THE DECLARATION 460
C OF THE ARRAYS PREPCOS(8200) AND PREPDIF(8200).                     470
      NPREP = 8 200                                                    480
C*****                                                             490
C          *****                                                             500
C          *****                                                             510
C          EPS = .000 100 000                                           520
C          EPSDBLE= .000 000 250                                         530
C          FCONCAV= 1.425 000 000                                         540
C          LASTN = 250 000 000                                           550
C          LPRINT = 7                                                    560
C          LSWITCH= 6                                                    570
C          MDIMENS= 4 025                                                580
C          NRANGE = 2 000 000                                           590
C          STAT = .TRUE.                                                600
C          ZMAXTST= 75.                                                  610
C          *****                                                             620
C          *****                                                             630
C*****                                                             640
      PI=DPI=4.DO*DATAN(1.DO) $ DPIINV=1.DO/DPI $ DPISL8=DPI*.125D0    650
      TWOPI=DTWOPI=2.DO*DPI $ DTWOPIN=.5D0*DPIINV                     660
      WRITE(1,10)LASTN,NRANGE                                           670

```

```

10  FORMAT(*1THIS RUN STARTS WITH LASTN=*,I14,/,/,
    $* AND NRRANGE                      =*,I14,/,/,/)
C.....DCNSTO IS USED IN DGRAM(N,DSTART)
DCNSTO=1.DO/96.DO/DPI/DPI
NMAX=LASTN+NRANGE $ TMAX=DGRAM(NMAX,DTWOPI*NMAX/ALOG(FLOAT(NMAX)))
LASTM=1.+SQRT(TMAX/TWOPI)
IF(LASTM.LE.MDIMENS)GOTO 30
WRITE(1,20)MDIMENS,LASTM
20  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 DCI( 30 ) CAN EASILY BE EXTENDED TO DCI( >30 ).
C.....SEE: F.D.CRARY AND J.B.ROSSER, HIGH PRECISION COEFFICIENTS
C.....RELATED TO THE ZETA FUNCTION, UNIV. OF WISCONSIN, REPORT # 1344.
C.....ALSO SEE: W.GABCKE, DISSERTATION, GOETTINGEN, 1979.
30  DC0(01)=+.3826834323 6508977172 8459984030D+00
    DC0(02)=+.4372404680 7752044936 0296467371D+00
    DC0(03)=+.1323765754 8034352332 4035267392D+00
    DC0(04)=-.1360502604 7674188654 9831887091D-01
    DC0(05)=-.1356762197 0103580887 9156705835D-01
    DC0(06)=-.1623725323 1444652828 5462529413D-02
    DC0(07)=+.2970535373 3379690783 1272833995D-03
    DC0(08)=+.7943300879 5214695880 1639026488D-04
    DC0(09)=+.4655612461 4504505037 0634021603D-06
    DC0(10)=-.1432725163 0955105754 0824631206D-05
    DC0(11)=-.1035484711 2312946075 0074156774D-06
    DC0(12)=+.1235792708 3861738056 1257626231D-07
    DC0(13)=+.1788108385 7954904985 6667814071D-08
    DC0(14)=-.3391414389 9270359069 4062189788D-10
    DC0(15)=-.1632663390 2565905101 3740529710D-10
    DC0(16)=-.3785109318 5412203828 5464720019D-12
    DC0(17)=+.9327423259 2017248456 6232063987D-13
    DC0(18)=+.5221843015 9781368553 1389314785D-14
    DC0(19)=-.3350673072 7442637895 1509035795D-15
    DC0(20)=-.3412426522 8117264940 8098710456D-16
    DC0(21)=+.5751203341 4323991603 3950179516D-18
    DC0(22)=+.1489530136 3211505454 7562777573D-18
    DC0(23)=+.1256537271 7021416853 3042817661D-20
    DC0(24)=-.4721295250 1434256689 5398813667D-21
    DC0(25)=-.1326906936 3039619992 7354130926D-22
    DC0(26)=+.1105343999 5121418344 5378225423D-23
    DC0(27)=+.5499646377 5274655111 4010449998D-25
    DC0(28)=-.1823137650 2318026280 6410898095D-26
    DC0(29)=-.1568940373 7720880146 8682982319D-27
    DC0(30)=+.1583963508 8238011610 6597605378D-29
C .....
DC1(01)=+.2682510262 8375347029 9914039557D-01
DC1(02)=-.1378477342 6351853049 8704525899D-01
DC1(03)=-.3849125048 2235082228 7364153632D-01
DC1(04)=-.9871066299 0620764720 1214704619D-02
DC1(05)=+.3310759760 8584043329 0907695130D-02
DC1(06)=+.1464780857 7954150824 9779656198D-02
DC1(07)=+.1320794062 4876963675 1614474944D-04
DC1(08)=-.5922748701 8471413232 2349952819D-04
DC1(09)=-.5980242585 3734485877 1083507452D-05
DC1(10)=+.9641322456 1698263526 7298532985D-06
DC1(11)=+.1833473372 2714411760 0167936578D-06
DC1(12)=-.4467087562 7178335995 6079422715D-08
DC1(13)=-.2709635082 1772743216 9262839871D-08

```

	DC1(14)=-.7785288654	3158510462	9482308521D-10	1280
	DC1(15)=+.2343762601	0893688532	4845504871D-10	1290
	DC1(16)=+.1583017278	9987521642	1622264263D-11	1300
	DC1(17)=-.1211994157	3723791246	6463447380D-12	1310
	DC1(18)=-.1458378116	1108307017	5828548170D-13	1320
	DC1(19)=+.2878630525	8131917504	5582128002D-15	1330
	DC1(20)=+.8662862902	1237241225	2825288793D-16	1340
	DC1(21)=+.8430722727	1370412715	6002253146D-18	1350
	DC1(22)=-.3630807223	0973462001	7324618110D-18	1360
	DC1(23)=-.1162669821	2838296719	4138886292D-19	1370
	DC1(24)=+.1097548671	1527531815	9018328340D-20	1380
	DC1(25)=+.6157399020	4684271038	8147079097D-22	1390
	DC1(26)=-.2290928006	7678471513	9638263100D-23	1400
	DC1(27)=-.2203281174	8848795343	7959827044D-24	1410
	DC1(28)=+.2476025180	0402785082	8527421518D-26	1420
	DC1(29)=+.5954277215	5836578022	7268286395D-27	1430
	DC1(30)=+.3261202074	6795952615	3375631907D-29	1440
C				1450
	DC2(01)=+.5188542830	2931684937	8458151923D-02	1460
	DC2(02)=+.3094658388	0634746033	4567436096D-03	1470
	DC2(03)=-.1133594107	8229373382	1824352559D-01	1480
	DC2(04)=+.2233045741	9581447720	5712552758D-02	1490
	DC2(05)=+.5196637408	8623302051	1692695307D-02	1500
	DC2(06)=+.3439914407	6208336694	6559135799D-03	1510
	DC2(07)=-.5910648427	4705828217	3225230308D-03	1520
	DC2(08)=-.1022997254	7935857454	4278675227D-03	1530
	DC2(09)=+.2088839221	6992755408	0732961742D-04	1540
	DC2(10)=+.5927665493	0965359578	9199648498D-05	1550
	DC2(11)=-.1642383836	2436275977	6903028478D-06	1560
	DC2(12)=-.1516119970	0940682861	7346053972D-06	1570
	DC2(13)=-.5907803698	2066679629	2279025398D-08	1580
	DC2(14)=+.2091151485	9478188977	7455551897D-08	1590
	DC2(15)=+.1781564958	3292351053	7997018788D-09	1600
	DC2(16)=-.1616407245	5353830752	8557694445D-10	1610
	DC2(17)=-.2380696249	6667615707	2107403801D-11	1620
	DC2(18)=+.5398265295	5425949181	8200414834D-13	1630
	DC2(19)=+.1975014219	6969515273	3087335885D-13	1640
	DC2(20)=+.2333286873	2882634831	0481530059D-15	1650
	DC2(21)=-.1118751761	0048080208	2004838090D-15	1660
	DC2(22)=-.4164009488	8837671885	0112283643D-17	1670
	DC2(23)=+.4446081109	2918830289	0304350093D-18	1680
	DC2(24)=+.2854611478	3637144545	7338742698D-19	1690
	DC2(25)=-.1191323143	0037894304	9718475053D-20	1700
	DC2(26)=-.1298163436	0736498946	7099023133D-21	1710
	DC2(27)=+.1612376317	8033262338	7796586632D-23	1720
	DC2(28)=+.4382497519	8873440596	5525842464D-24	1730
	DC2(29)=+.2718638957	6555759138	8203562714D-26	1740
	DC2(30)=-.1145889650	6774580369	7439455793D-26	1750
C				1760
	DC3(01)=+.1339716090	7194569042	6983572995D-02	1770
	DC3(02)=-.3744215136	3793937046	6416186446D-02	1780
	DC3(03)=+.1330317891	9321468120	3185472240D-02	1790
	DC3(04)=+.2265466076	5471787114	7603199052D-02	1800
	DC3(05)=-.9548499998	5067304151	1225515765D-03	1810
	DC3(06)=-.6010038458	9636039120	7580587580D-03	1820
	DC3(07)=+.1012885828	6776621953	3443494181D-03	1830
	DC3(08)=+.6865733449	2998256424	5742836487D-04	1840
	DC3(09)=-.5985366791	5385981593	0593385329D-06	1850
	DC3(10)=-.3331659851	2399471290	4355366984D-05	1860
	DC3(11)=-.2191928910	2435081057	1848421923D-06	1870

```

DC3(12)=+.7890884245 6814944105 5524826157D-07      1880
DC3(13)=+.9414685081 2952621516 5246515671D-08      1890
DC3(14)=-.9570116210 8834803018 8072284774D-09      1900
DC3(15)=-.1876313745 3470662796 8129705778D-09      1910
DC3(16)=+.4437837679 3233993274 6470898497D-11      1920
DC3(17)=+.2242673850 5617353248 4110685731D-11      1930
DC3(18)=+.3627686865 7352436894 0825563792D-13      1940
DC3(19)=-.1763980955 0821581607 8311214981D-13      1950
DC3(20)=-.7960765246 7867777572 9034517928D-15      1960
DC3(21)=+.9419651490 5896907639 1489502569D-16      1970
DC3(22)=+.7133103854 5696578245 5666792464D-17      1980
DC3(23)=-.3289910584 5546243211 7966525849D-18      1990
DC3(24)=-.4180730374 8984592913 6292487056D-19      2000
DC3(25)=+.5550542071 6463337897 8211640266D-21      2010
DC3(26)=+.1787044190 6260123858 7176363531D-21      2020
DC3(27)=+.1331280396 4656094286 2973430146D-23      2030
DC3(28)=-.5818610611 0909875161 7921659609D-24      2040
DC3(29)=-.1401903608 8526555374 3649670980D-25      2050
DC3(30)=+.1464132021 1626254148 9977525019D-26      2060
C .....                                2070
DO 50 I=1,10                                2080
C.....NBLOCL(I):= THE NUMBER OF GRAM BLOCKS OF LENGTH I. 2090
NBLOCL(I)=0                                2100
DO 40 J=1,10                                2110
C.....INTRVAL(I,J):= THE NUMBER OF GRAM BLOCKS OF LENGTH I 2120
C.....CONTAINING THE MISSING TWO IN THEIR J-TH GRAM INTERVAL. 2130
INTRVAL(I,J)=0                                2140
40 CONTINUE                                2150
50 CONTINUE                                2160
DO 60 I=1,MDIMENS                            2170
DI=I $ DLN(I)=DLOG(DI) $ SQRTINV(I)=DSQRT(1.DO/DI) 2180
60 CONTINUE                                2190
C....."8192" IS AN ABSOLUTE CONSTANT FOR THIS PROGRAM ! 2200
GRID=DGRID=DTWOPI/8192.DO $ GRIDIN=8192.DO/DTWOPI 2210
PREPCOS(1)=1. $ DPC1=1.DO                    2220
DO 70 I=2,NPREP                              2230
DPC2=DCOS((I-1)*DGRID)                      2240
PREPCOS(I)=DPC2 $ PREPDIF(I-1)=DPC2-DPC1 $ DPC1=DPC2 2250
70 CONTINUE                                2260
NZEVALU=NEW00=NSHIFTS=0 $ ZTIME=0. $ NFIRST=LASTN 2270
C.....NZEVALU:= NUMBER OF Z-EVALUATIONS. 2280
C.....NZEVALU IS SET TO 0 SO THAT "WASTE" RELATES TO THIS RUN ONLY. 2290
C.....THE FOLLOWING THREE CONSTANTS ARE USED IN DTHETA(DT). 2300
DCNST1=1.DO/48.DO $ DCNST2=7.DO/5760.DO $ DCNST3=31.DO/80640.DO 2310
C .....                                2320
C..... END OF PREPARATIONS 2330
C *****                                2340
C *****                                2350
C ..... ESSENTIAL START OF THE PROGRAM 2360
C *****                                2370
C *****                                2380
C .....                                2390
C.....LASTN IS THE INDEX OF THE LAST KNOWN ZERO. 2400
C.....TAKE LASTN FROM THE OUTPUT OF THE PREVIOUS RUN. 2410
80 N=LASTN-1                                2420
DGO=DGRAM(N,DTWOPI*N/ALOG(FLOAT(N)))          2430
TIME=SECOND(CP) $ ZO=Z(DGO)                  2440
ZTIME=ZTIME+(SECOND(CP)-TIME) $ NZEVALU=NZEVALU+1 2450
IF(ABS(ZO).GT.EPS)GOTO 100                    2460
TIME=SECOND(CP) $ ZO=DZ(DGO)                  2470

```

```

      ZTIME=ZTIME+(SECOND(CP)-TIME) $ NZEVALU=NZEVALU+1      2480
      IF(ABS(ZO).GT.EPSDBLE)GOTO 100      2490
      WRITE(1,90)LASTN,DGO,ZO      2500
90    FORMAT(/,/,*,* BAD START WITH LASTN=*,I14,/,      2510
      $* ZO(*,F26.12,*)=*,F26.12,/,*,* THIS STARTING VALUE IS TOO SMALL.*,      2520
      $/,*,* LASTN IS INCREASED BY 1 DUE TO THIS "UNCLEAR" VALUE OF Z(T).*)      2530
      GOTO 130      2540
100   IF(ZO*(-1)**N).LT.0.)GOTO 110      2550
      IF(ABS(ZO).LT.ZMAXTST)GOTO 140      2560
      ZMAXTST=ZMAXTST+.5*(ABS(ZO)-ZMAXTST)      2570
      WRITE(1,200)N,DGO,ZO      2580
      GOTO 140      2590
110   WRITE(1,120)N,DGO      2600
120   FORMAT(/,/,*,* LASTN IS INCREASED BY 1 DUE TO THE BAD INITIAL      2610
      $ GRAM POINT G(*,I14,*)=*,F26.12,/,/)      2620
130   LASTN=LASTN+1 $ NRANGE=NRANGE-1 $ GOTO 80      2630
140   IF(NEW00.LT.NRANGE)GOTO 190      2640
C.....FINAL OUTPUT      2650
150   DGN=DGRAM(N,DGO) $ M=DSQRT(DGN/DTWOPI)      2660
      TOTTIME=SECOND(CP)-TOTTIME $ WASTE=FLOAT(NZEVALU)/NEW00-1.      2670
C.....N+1 IS THE "LASTN" FOR THE NEXT RUN.      2680
      WRITE(1,160)NFIRST,N+1,DGN,NZEVALU,WASTE,FCONCAV,NSHIFTS,      2690
      $ZTIME/NZEVALU,ZTIME,TOTTIME,TOTTIME/NZEVALU,M,EPS,EPSDBLE      2700
160   FORMAT(1H1,/,      2710
      $* NFIRST ( WAS INPUT FOR THIS RUN )      2720
      $* LASTN ( INPUT FOR NEXT RUN )      2730
      $* THE GRAM POINT G(LASTN-1)      2740
      $* NZEVALU      2750
      $* "WASTE" ( FOR THIS RUN )      2760
      $* FCONCAV      2770
      $* NSHIFTS      2780
      $* AVERAGE TIME FOR ONE Z - EVALUATION      2790
      $* TOTAL TIME USED FOR ALL Z - EVALUATIONS      2800
      $* TOTAL TIME USED IN THIS RUN      2810
      $* AVERAGE TOTAL TIME FOR ONE Z - EVALUATION      2820
      $* LAST M (= FINAL SUMMATION RANGE IN Z(T))      2830
      $* EPS      2840
      $* EPSDBLE      2850
      INTRVAL(1,1)=NBLOCL(1)      2860
      DO 180 I=1,10      2870
      WRITE(1,170)I,NBLOCL(I),(INTRVAL(I,J),J=1,I)      2880
170   FORMAT(* I=*,I2,*; # BLOCKL=*,I8,* $ *,2I8,I7,I6,I5,I4,4(I3),/)      2890
180   CONTINUE      2900
      IF(STAT)CALL STATIST(N+1)      2910
C.....      2920
      END OF JOB.      2930
      STOP      2940
C.....WE ARE GOING TO SET UP A GRAM BLOCK OF LENGTH LBLOC.      2950
190   DG1=DGRAM(N+1,DGO)      2960
      DREFPNT=DGO $ CALL COMPZ(Z1,DREFPNT,DG1,.99,N+1)      2970
      IF(ABS(Z1).LT.ZMAXTST)GOTO 210      2980
C.....LARGE VALUES OF ABS(Z(T)) ARE INTERESTING ANYHOW.      2990
C.....WE DON'T INCREASE ZMAXTST TOO DRASTICALLY.      3000
      ZMAXTST=ZMAXTST+.5*(ABS(Z1)-ZMAXTST)      3010
      WRITE(1,200)N+1,DG1,Z1      3020
200   FORMAT(/,*,* LARGE VALUE FOR Z ..... *,      3030
      $* G(*,I14,*)=*,F24.12,* CORRESPONDING Z=*,F18.6,/)      3040
210   IF(ZO*Z1.GT.0.)GOTO 220      3050
C.....WE ENCOUNTER A GRAM INTERVAL ( GRAM BLOCK OF LENGTH LBLOC = 1 ).      3060
      NEW00=NEW00+1 $ N=N+1 $ DGO=DG1 $ ZO=Z1      3070
      NBLOCL(1)=NBLOCL(1)+1 $ GOTO 140

```

C.....	EXCEPTION TO GRAM'S LAW. LBLOC WILL BE > 1.	3080
220	LBLOC=1 \$ DTDIM(1)=DGO \$ ZDIM(1)=Z0	3090
230	LBLOC=LBLOC+1 \$ IF(LBLOC.LT.11)GOTO 250	3100
	WRITE(1,240)N+1,DTDIM(1)	3110
240	FORMAT(* WE ENCOUNTER A GRAM BLOCK OF LENGTH > 10 STARTING AT *,/,	3120
	\$* G(*,I14,*)=*,F26.12,/,	3130
	\$*INCREASE THE DIMENSIONS OF DTDIM(.) AND ZDIM(.) *)	3140
	GOTO 150	3150
250	DTDIM(LBLOC)=DG1 \$ ZDIM(LBLOC)=Z1	3160
	DG2=DGRAM(N+LBLOC,DG1)	3170
	DREFPNT=DG1 \$ CALL COMPZ(Z2,DREFPNT,DG2,.99,N+LBLOC)	3180
	IF(ABS(Z2).LE.ZMAXTST)GOTO 260	3190
	ZMAXTST=ZMAXTST+.5*(ABS(Z2)-ZMAXTST)	3200
	WRITE(1,200)N+LBLOC,DG2,Z2	3210
260	IF(Z1*Z2.GT.0.)GOTO 270	3220
	DG1=DG2 \$ Z1=Z2 \$ GOTO 230	3230
270	DTDIM(LBLOC+1)=DG2 \$ ZDIM(LBLOC+1)=Z2	3240
	NBLOCL(LBLOC)=NBLOCL(LBLOC)+1	3250
C.....	THE GRAM BLOCK IS READY AND PRESERVED IN TDIM(.) AND ZDIM(.)	3260
C.....	NOTE THAT IN THIS GRAM BLOCK WE HAVE ALREADY DETECTED THE	3270
C.....	PRESENCE OF LBLOC-2 ZEROS. IF LBLOC <= LSWITCH THE MISSING TWO ARE	3280
C.....	FIRSTLY SOUGHT IN THE TWO OUTER GRAM INTERVALS OF THE BLOCK.	3290
	IF(LBLOC.GT.2)GOTO 280	3300
C.....	LBLOC=2.	3310
	IF(ABS(ZDIM(2)).GT.AMIN1(ABS(ZDIM(1)),ABS(ZDIM(3))))GOTO 280	3320
C.....	ABS(Z(T)) LOOKS CONVEX AND THEREFORE WE CALL SRCH2D.	3330
	CALL SRCH2D(DTDIM(1),ZDIM(1),DTDIM(2),ZDIM(2),DTDIM(3),ZDIM(3),	3340
	\$NFOUND,INTVAL)	3350
C.....	IF NFOUND=0 WE STOP IN SRCH2D.	3360
C.....	WE MAINTAINED NFOUND AS A PARAMETER FOR EXPERIMENTAL REASONS.	3370
	INTRVAL(2,INTVAL)=INTRVAL(2,INTVAL)+1 \$ GOTO 310	3380
280	IF(LBLOC.GE.LSWITCH)GOTO 390	3390
C.....	IF LBLOC < LSWITCH, WE FIRST SEARCH IN THE OUTER INTERVALS.	3400
C.....	IF LBLOC >= LSWITCH, WE FIRST SEARCH IN THE INNER INTERVALS.	3410
290	I1=1 \$ I2=LBLOC	3420
	IF(ABS(ZDIM(I1)+ZDIM(I1+1)).LT.ABS(ZDIM(I2)+ZDIM(I2+1)))GOTO 300	3430
	I1=LBLOC \$ I2=1	3440
300	CALL SRCH2A(DTDIM(I1),DTDIM(I1+1),ZDIM(I1),ZDIM(I1+1),NFOUND)	3450
	IF(NFOUND.LT.2)GOTO 330	3460
	INTRVAL(LBLOC,I1)=INTRVAL(LBLOC,I1)+1	3470
310	NEW00=NEW00+LBLOC	3480
	IF(LBLOC.GE.LPRINT)GOTO 550	3490
320	N=N+LBLOC \$ DGO=DTDIM(LBLOC+1) \$ Z0=ZDIM(LBLOC+1) \$ GOTO 140	3500
330	CALL SRCH2A(DTDIM(I2),DTDIM(I2+1),ZDIM(I2),ZDIM(I2+1),NFOUND)	3510
	IF(NFOUND.LT.2)GOTO 340	3520
	INTRVAL(LBLOC,I2)=INTRVAL(LBLOC,I2)+1 \$ GOTO 310	3530
340	IF(LBLOC.GT.2.AND.LBLOC.LT.LSWITCH)GOTO 390	3540
C.....	EXPERIMENTS INDICATE THAT IT IS USEFULL TO CALL SRCH2B HERE.	3550
	CALL SRCH2B(DTDIM(1),ZDIM(1),DTDIM(3),ZDIM(3),NFOUND,INTVAL,2,0)	3560
	IF(NFOUND.LT.2)GOTO 350	3570
	INTRVAL(2,INTVAL)=INTRVAL(2,INTVAL)+1 \$ GOTO 310	3580
350	DA=DTDIM(1) \$ DB=DTDIM(3)	3590
	WRITE(1,360)DA,DB,N,N+LBLOC	3600
360	FORMAT(/,40(* !*),/,*, LBLOC=2. WE DID NOT FIND ANY ZEROS BETWEEN	3610
	\$*,/,*, T1=*,F26.12,* AND T2=*,F26.12,/,	3620
	\$* WE LIST SOME VALUES OF Z(T) IN THIS RANGE *,/,	3630
	\$* WE ADD TWO ZEROS TO NEW00 AND CONTINUE *,/,	3640
	\$* THIS HAPPENED BETWEEN N=*,I14,* AND N+LBLOC=*,I14,/))	3650
	DH=(DB-DA)/11.DO	3660
	DO 380 I=1,12	3670

```

TIME=SECOND(CP) $ ZA=Z(DA)
ZTIME=ZTIME+(SECOND(CP)-TIME) $ NZEVALU=NZEVALU+1
WRITE(1,370)DA,ZA
370 FORMAT(* T=*,F26.12,*      Z(T)=*,F28.14)
DA=DA+DH
380 CONTINUE
GOTO 310
C.....SEARCH IN THE INNER INTERVALS OF THE GRAM BLOCK.
390 LASTK=LBLOC-2 $ LASTKP1=LASTK+1 $ IDEPTH=2
NUM1=INTRVAL(LBLOC,2) $ NUM2=INTRVAL(LBLOC,LBLOC-1)
IF(NUM1.LT.NUM2)GOTO 520
DO 410 II=2,7
ISIGN=-1 $ IP=LASTKP1 $ IDEPTH=2*IDEPTH
DO 400 K=1,LASTK
C.....THE NEXT LINE CONTROLS THE ZIG-ZAG SEARCH OF SRCH3.
IP=IP+ISIGN*(LASTKP1-K) $ ISIGN=-ISIGN
CALL SRCH3(DTDIM(IP+1),ZDIM(IP+1),DTDIM(IP+2),ZDIM(IP+2),
$NFOUND,IDEPTH)
IF(NFOUND.EQ.3)GOTO 510
400 CONTINUE
410 CONTINUE
C.....(LBLOC > 2).
C.....THE MISSING TWO WERE NOT FOUND IN ONE OF THE INNER INTERVALS.
C.....WE SEARCH AGAIN IN THE MOST SUSPICIOUS OUTER GRAM INTERVAL OF
C.....THE GRAM BLOCK BY MEANS OF SRCH2C ( = LEHMAN'S SEARCH ).
C.....IN STEAD OF SRCH2C ONE MAY ALSO EXPERIMENT WITH SRCH2B HERE.
420 IF(ABS(ZDIM(1)+ZDIM(2)).GT.ABS(ZDIM(LBLOC)+ZDIM(LBLOC+1)))GOTO 430
DA=DTDIM(1) $ DB=DTDIM(2) $ INDEX=1
ZA= ZDIM(1) $ ZB= ZDIM(2) $ GOTO 440
430 DA=DTDIM(LBLOC) $ DB=DTDIM(LBLOC+1) $ INDEX=LBLOC
ZA= ZDIM(LBLOC) $ ZB= ZDIM(LBLOC+1)
C.....FROM THE ABOVE 5 LINES IT IS CLEAR ON WHICH INTERVAL WE BET.
440 IF(ABS(ZA).LT.ABS(ZB))GOTO 465
DAUX=DA $ DA=DB $ DB=DAUX
C450 WRITE(1,460)DA,DB,LBLOC,N
C460 FORMAT(/,* WE CALL SRCH2C BETWEEN *,/,
C $* T1=*,F22.12,* AND T2=*,F22.12,* LBLOC=*,I2,* N=*,I11,/)
465 CALL SRCH2C(DA,DB,ZA,NFOUND,4)
C.....THE LAST PARAMETER IN SRCH2C REGULATES THE SEARCH DEPTH.
IF(NFOUND.EQ.2)GOTO 500
C.....IT APPEARS TO BE USELESS TO SEARCH IN THE OTHER OUTER INTERVAL.
C.....WE PRINT SOME VALUES OF Z(T).
DA=DTDIM(1) $ DB=DTDIM(LBLOC+1)
WRITE(1,470)DA,DB,LBLOC,LBLOC-2,LBLOC,N,N+LBLOC
470 FORMAT(/,40(* !*),/,* ? VIOLATION OF ROSSER'S RULE ? BETWEEN *,/,
$* T1=*,F26.12,* AND T2=*,F26.12,/,
$* # OF SIGN CHANGES SHOULD BE LBLOC=*,I3,* WE FOUND ONLY L=*,I3,/,
$* WE ADD LBLOC=*,I3,* ZEROS TO NEW00 *,/,
$* THE POSSIBLE ERROR SHOULD BE DETECTED LATER "BY HAND" *,/,
$* THIS HAPPENED BETWEEN N=*,I14,* AND N+LBLOC=*,I14,/,
$* WE LIST SOME VALUES OF Z(T) IN THIS RANGE AND CONTINUE *,/,/)
NPRINT=7*LBLOC+1 $ DH=(DB-DA)/(NPRINT-1)
DO 490 KK=1,NPRINT
DARG=DA+(KK-1)*DH $ TIME=SECOND(CP) $ ZARG=Z(DARG)
ZTIME=ZTIME+(SECOND(CP)-TIME) $ NZEVALU=NZEVALU+1
WRITE(1,480)DARG,ZARG
480 FORMAT(* T=*,F26.12,*      Z(T)=*,F28.14)
490 CONTINUE
C.....WE PRETEND (!) THAT THE MISSING TWO WERE FOUND !
C.....CLASSIFICATION OF THE BLOCK IS IMPOSSIBLE AT THIS INSTANT.

```

```

      GOTO 310
      ....THE MISSING TWO WERE FOUND. WE CLASSIFY THE GRAM BLOCK.
20  INTRVAL(LBLOC,INDEX)=INTRVAL(LBLOC,INDEX)+1 $ GOTO 310
10  INTRVAL(LBLOC,IP+1) =INTRVAL(LBLOC,IP+1)+1 $ GOTO 310
      ....SEARCH IN THE INNER INTERVALS OF THE GRAM BLOCK.
20  DO 540 II=2,7
      ISIGN=+1 $ IP=0 $ IDEPTH=2*IDEPTH
      DO 530 K=1, LASTK
      ....THE NEXT LINE CONTROLS THE ZIG-ZAG SEARCH OF SRCH3.
      IP=IP+ISIGN*(LASTKP1-K) $ ISIGN=-ISIGN
      CALL SRCH3(DTDIM(IP+1),ZDIM(IP+1),DTDIM(IP+2),ZDIM(IP+2),
      $NFOUND,IDEPTH)
      IF(NFOUND.EQ.3)GOTO 510
30  CONTINUE
40  CONTINUE
      ....THE MISSING TWO WERE NOT FOUND. WE TRANSFER THE SEARCH BACK
      ....TO ONE OF THE OUTER INTERVALS.
      GOTO 420
      ....IF LBLOC >= NPRINT WE PRINT SOME INTERMEDIATE STATISTICS.
50  WASTE=FLOAT(NZEVALU)/NEW00-1.
      WRITE(1,560)LBLOC,N,DG0,DG2,WASTE,NSHIFTS,(NBLOCL(I),I=1,10)
50  FORMAT(/,* GRAM-BLOCK OF LENGTH *,I3,* FOR N=*,I14,* BETWEEN *,/,
      $* T1=*,F27.12,* AND T2=*,F27.12,/,
      $* WASTE=*,F8.4,/,
      $* NSHIFTS=*,I14,/,
      $* BLOCKLENGTHS *,I9,I8,8(I6),/,/,/)
      INTRVAL(1,1)=NBLOCL(1)
      DO 580 I=1,10
      WRITE(1,570)I,(INTRVAL(I,J),J=1,I)
70  FORMAT(* I=*,I3,10I7)
30  CONTINUE
      GOTO 320
      END

      DOUBLE FUNCTION DGRAM(N,DSTART)
      ....DGRAM(N,DSTART) RETURNS THE GRAM POINT G(N).
      IMPLICIT DOUBLE (D)
      COMMON/BLOC1/DPI,DPIINV,DTWOPI,DTWOPIN,DPISL8
      COMMON/BLOC9/DCNSTO
      DGI=DSTART/DTWOPI
      DGI1=(DGI+N+.125DO-DCNSTO/DGI)/DLOG(DGI)
      IF(DABS((DGI-DGI1)/DGI1).LT.1.D-20)GOTO 20
      DGI=DGI1 $ GOTO 10
      DGRAM=DGI1*DTWOPI
      RETURN
      END

      SUBROUTINE COMPZ(ZT1,DREFPNT,DT1,SQZFCTR,K)
      IMPLICIT DOUBLE (D)
      COMMON/BLOC5/EPS,EPSDBLE,ZTIME,NZEVALU
      COMMON/BLOC6/NSHIFTS
      TIME=SECOND(CP) $ ZT1=Z(DT1)
      ZTIME=ZTIME+(SECOND(CP)-TIME) $ NZEVALU=NZEVALU+1
      IF(ABS(ZT1).GT.EPS)RETURN
      ....WE TRY TO FIND A CLEAR Z - VALUE BY MEANS OF DZ(DT).
      ....WE FIRST TRY DZ(DT1).
      TIME=SECOND(CP) $ ZT1=DZ(DT1)
      ZTIME=ZTIME+(SECOND(CP)-TIME) $ NZEVALU=NZEVALU+1
      IF(ABS(ZT1).GT.EPSDBLE)RETURN
      IF(K.LT.1)GOTO 20

```

```

WRITE(1,10)DT1,ZT1,K
10  FORMAT(/,* WE ENTER DZ(.) IN A GRAM POINT *,/,
    $* T1=*,F26.12,* ZT1=*,F15.12,* N=*,I14)
C.....NOW WE START SHIFTING THE ARGUMENT OF DZ(.).
20  DO 30 NZ=1,12
    CALL PRSHIF(DT1,ZT1)
    DT1=DREFPNT+(DT1-DREFPNT)*SQZFCTR $ NSHIFTS=NSHIFTS+1
    TIME=SECOND(CP) $ ZT1=DZ(DT1)
    ZTIME=ZTIME+(SECOND(CP)-TIME) $ NZEVALU=NZEVALU+1
    IF(ABS(ZT1).GT.EPSDBLE)RETURN
30  CONTINUE
    WRITE(1,40)
40  FORMAT(/,* SERIOUS DIFFICULTIES IN FINDING A CLEAR Z. WE STOP. *)
    STOP
    END

SUBROUTINE PRSHIF(DT,ZT)
C.....ANNOUNCEMENT OF SHIFTS.
    IMPLICIT DOUBLE (D)
    WRITE(1,10)DT,ZT
10  FORMAT(* .... WE MAKE A SHIFT AT T=*,F30.14,* ..... ZT=*,F15.12)
    RETURN
    END

DOUBLE FUNCTION DTHETA(DT)
    IMPLICIT DOUBLE (D)
    COMMON/BLOC1/DPI,DPIINV,DTWOPI,DTWOPIN,DPISL8
    COMMON/BLOC4/DCNST1,DCNST2,DCNST3
    DTINV=1.D0/DT $ DTINVSQ=DTINV*DTINV
    DTHETA=DT*.5D0*(DLOG(DT*DTWOPIN)-1.D0)-DPISL8+
    $((DCNST3*DTINVSQ+DCNST2)*DTINVSQ+DCNST1)*DTINV
    RETURN
    END

SUBROUTINE SRCH2A(DA,DB,ZA,ZB,NFOUND)
C.....SEARCH FOR TWO ZEROS BETWEEN A AND B.
C.....SRCH2A IS CALLED ONLY IF ZA*ZB > 0.
    IMPLICIT DOUBLE (D)
    COMMON/S2PARA/DARRT(513),ARRZ(513),NEXT(512),ACCEPT(512)
    COMMON/BLOC5/EPS,EPSDBLE,ZTIME,NZEVALU
    COMMON/BLOC8/FCONCAV
    LOGICAL ACCEPT
    NFOUND=0
    IF(ABS(ZA).GT.ABS(ZB))GOTO 10
    DARRT(1)=DA $ DARRT(513)=DB
    ARRZ(1)=ZA $ ARRZ(513)=ZB $ GOTO 20
10  DARRT(1)=DB $ DARRT(513)=DA
    ARRZ(1)=ZB $ ARRZ(513)=ZA
20  DT=(DA+DB)*.5D0 $ DREFPNT=DA $ CALL COMPZ(ZT,DREFPNT,DT,.99,0)
    IF(ZA*ZT.GT.0.)GOTO 30
    NFOUND=2 $ RETURN
C.....IF ABS(Z(T)) LOOKS "QUITE" CONCAVE WE DON'T CONTINUE THE SEARCH.
30  IF(ABS(ZT).GT.FCONCAV*(ABS(ZA)+ABS(ZB)))RETURN
    IF(ABS(ZT).GT.AMIN1(ABS(ZA),ABS(ZB)))GOTO 40
C.....WE HAVE DETECTED CONVEXITY AND HENCE CALL SRCH2D.
    CALL SRCH2D(DA,ZA,DT,ZT,DB,ZB,NFOUND,INTVAL) $ RETURN
C.....IF NFOUND=0 WE STOP IN SRCH2D.
C.....WE MAKE THE NECESSARY PREPARATIONS FOR CALLING PARA.
40  DARRT(257)=DT $ ARRZ(257)=ZT
    ACCEPT( 1)=.FALSE. $ NEXT( 1)=256

```

```

ACCEPT(257)=.FALSE. $ NEXT(257)=256 5480
INDEX=1 5490
50 CALL PARA(INDEX,INDOUT,NFOUND) 5500
C.....NOTE THAT PARA IS CALLED ONLY IN SRCH2A. 5510
IF(NFOUND.GT.0)RETURN 5520
IF(INDOUT.LT.513)GOTO 60 5530
C.....THE WHOLE INTERVAL [ A,B ] HAS BEEN SCANNED NOW BY PARA. 5540
RETURN 5550
C.....TRANSFER THE SEARCH TO THE NEXT INTERVAL. 5560
60 INDEX=INDOUT 5570
70 IF(.NOT.ACCEPT(INDEX))GOTO 50 5580
C.....SINCE ACCEPT(INDEX)=.TRUE., WE SKIP THE NEXT INTERVAL 5590
C.....[ DARRT(INDEX),DARRT(INDEX+NEXT(INDEX)) ]. 5600
INDEX=INDEX+NEXT(INDEX) 5610
IF(INDEX.LT.513)GOTO 70 5620
RETURN 5630
END 5640
5650
SUBROUTINE PARA(INDEX,INDOUT,NFOUND) 5660
C.....PARA ( SHORT FOR PARABOLA ) IS CALLED ONLY IN SRCH2A SO THAT IT 5670
C.....IS APPLIED ONLY TO INTERVALS [ A,B ] WITH ZA*ZB > 0. 5680
C.....DEFINE: N=NEXT(INDEX), T1=AT(INDEX), T3=AT(INDEX+N), N2=N/2. 5690
C.....PARA TRIES TO FIND TWO ZEROS IN THE INTERVAL [ T1,T3 ] BY MEANS 5700
C.....OF A CLOSE LOOK AT THE GRAPH OF ABS(Z(T)) ON THIS INTERVAL. 5710
C.....THIS ROUTINE SAVES MANY EVALUATIONS OF Z(T) IN CASE THE GRAPH 5720
C.....OF ABS(Z(T)) IS CONCAVE. 5730
C.....IF FOR SOME K THE INTERVAL [ DARRT(K),DARRT(K+NEXT(K)) ] IS JUDGED 5740
C.....AS CONTAINING NO ZEROS WE SET ACCEPT(K)=.T., ELSE ACCEPT(K)=.F.. 5750
IMPLICIT DOUBLE (D) 5760
COMMON/S2PARA/DARRT(513),ARRZ(513),NEXT(512),ACCEPT(512) 5770
COMMON/BLOC5/EPS,EPSDBLE,ZTIME,NZEVALU 5780
LOGICAL ACCEPT 5790
N=NEXT(INDEX) $ N2=N/2 5800
IF(N2.GE.1)GOTO 10 5810
INDOUT=INDEX+1 $ RETURN 5820
10 NEXT(INDEX)=N2 $ NEXT(INDEX+N2)=N2 5830
ACCEPT(INDEX)=.TRUE. $ ACCEPT(INDEX+N2)=.TRUE. 5840
DT1=DARRT(INDEX) $ DT3=DARRT(INDEX+N) 5850
Z1=ARRZ(INDEX) $ Z3=ARRZ(INDEX+N) 5860
DT2=(DT1+DT3)*.5D0 $ DREFPNT=DT1 $CALL COMPZ(Z2,DREFPNT,DT2,.99,0) 5870
IF(Z1*Z2.GT.0.)GOTO 15 5880
NFOUND=2 $ RETURN 5890
15 DARRT(INDEX+N2)=DT2 5900
IF(ABS(Z2).GT.AMIN1(ABS(Z1),ABS(Z3)))GOTO 20 5910
C.....SINCE CONVEXITY HAS BEEN DETECTED WE CALL SRCH2D. 5920
C.....THE PARAMETER INTVAL IS IRRELEVANT HERE. 5930
CALL SRCH2D(DT1,Z1,DT2,Z2,DT3,Z3,NFOUND,INTVAL) 5940
C.....IF NFOUND=0 WE STOP IN SRCH2D. 5950
RETURN 5960
20 ARRZ(INDEX+N2)=Z2 5970
IF(Z2*Z1.GT.0.)GOTO 30 5980
C.....TWO ZEROS FOUND. 5990
NFOUND=2 $ RETURN 6000
C.....IN THE FOLLOWING LINES OF THIS ROUTINE WE INSPECT THE 6010
C.....GRAPH OF ABS(Z(T)) BY MEANS OF A PARABOLIC APPROXIMATION. 6020
30 ABSZ1=ABS(Z1) $ ABSZ2=ABS(Z2) $ ABSZ3=ABS(Z3) 6030
40 IF(ABSZ2.LT.(ABSZ1+ABSZ3)*.5)GOTO 60 6040
50 INDOUT=INDEX+N $ RETURN 6050
60 DX2=DT2-DT1 $ DX3=DT3-DT1 6060
DC1=(DX3*(ABSZ2-ABSZ1)-DX2*(ABSZ3-ABSZ1))/DX2/DX3/(DX2-DX3) 6070

```

```

DC2=(ABSZ2-ABSZ1)/DX2-DC1*DX2                                6080
DMU=-DC2*.5D0/DC1 $ DXMIDP=DX3*.5D0 $ DHALF=DABS(DXMIDP)      6090
DXDIST=DABS(DMU-DXMIDP)                                       6100
IF(DXDIST.GT.DHALF)GOTO 50                                    6110
IF(DABS(DMU).LE.DABS(DX2))GOTO 70                             6120
ACCEPT(INDEX+N2)=.FALSE. $ INDOUT=INDEX+N2 $ RETURN           6130
70 ACCEPT(INDEX)=.FALSE. $ INDOUT=INDEX $ RETURN               6140
END                                                            6150
                                                                6160
SUBROUTINE SRCH2B(DA,ZA,DB,ZB,NFOUND,INTVAL,NCYCLES,IOPTION)  6170
C.....WE TRY TO FIND 2 SIGN CHANGES OF Z(T) IN THE INTERVAL (A,B). 6180
C.....SRCH2B IS CALLED ONLY IF A < B AND Z(A)*Z(B) > 0.        6190
IMPLICIT DOUBLE (D)                                           6200
COMMON/BLOC5/EPS,EPSTBLE,ZTIME,NZEVALU                      6210
SIGN=-1. $ IF(ABS(ZA).GT.ABS(ZB))SIGN=1.                     6220
DH=(DB-DA)*.125D0 $ DTMIDPT=(DA+DB)*.5D0                    6230
C.....NOTE THAT IN THIS ROUTINE THE ARGUMENT OF Z(T) ZIGZAGS    6240
C.....ACCORDING TO A PATTERN SUCH AS ... 9 7 5 3 1 0 2 4 6 8 10... 6250
IF(IOPTION.EQ.0)GOTO 10                                       6260
C.....IF(IOPTION.NE.0) WE ALSO COMPUTE Z(DTMIDPT). ELSE WE DON'T. 6270
DT=DTMIDPT                                                    6280
DREFPNT=DT+SIGN*DH $ CALL COMPZ(ZT,DREFPNT,DT,.99,0)         6290
IF(ZA*ZT.GT.0.)GOTO 10                                       6300
C.....TWO ZEROS FOUND.                                          6310
NFOUND=2 $ INTVAL=1                                           6320
IF(DT.GT.DTMIDPT)INTVAL=2 $ RETURN                            6330
10 IDEPTH=4 $ DH=2.D0*DH                                       6340
DO 40 I=1,NCYCLES                                           6350
IDEPTH=2*IDEPTH $ LASTJ=IDEPTH/2-1 $ DH=DH*.5 $ DHSIGN=DH*SIGN 6360
DO 30 J=1,LASTJ,2                                           6370
DSTEP=SIGN*J*DH $ DT=DTMIDPT+DSTEP                          6380
DREFPNT=DT+DHSIGN $ CALL COMPZ(ZT,DREFPNT,DT,.99,0)         6390
IF(ZA*ZT.GT.0.)GOTO 20                                       6400
C.....TWO ZEROS FOUND.                                          6410
NFOUND=2 $ INTVAL=1                                           6420
IF(DT.GT.DTMIDPT)INTVAL=2 $ RETURN                            6430
20 DT=DTMIDPT-DSTEP                                           6440
DREFPNT=DT-DHSIGN $ CALL COMPZ(ZT,DREFPNT,DT,.99,0)         6450
IF(ZA*ZT.GT.0.)GOTO 30                                       6460
C.....TWO ZEROS FOUND.                                          6470
NFOUND=2 $ INTVAL=1                                           6480
IF(DT.GT.DTMIDPT)INTVAL=2 $ RETURN                            6490
30 CONTINUE                                                  6500
40 CONTINUE                                                  6510
NFOUND=0                                                       6520
RETURN                                                         6530
END                                                            6540
                                                                6550
SUBROUTINE SRCH2C(DA,DB,ZA,NFOUND,NCYCLES)                   6560
C.....WE SEARCH FOR TWO ZEROS IN [ A,B ]. SEARCH DIRECTION: A --> B. 6570
C.....IN THIS ROUTINE WE ALWAYS HAVE Z(A)*Z(B) > 0.          6580
IMPLICIT DOUBLE (D)                                           6590
COMMON/BLOC5/EPS,EPSTBLE,ZTIME,NZEVALU                      6600
DH=(DB-DA)*.5D0 $ K=2                                         6610
DO 20 I=1,NCYCLES                                           6620
DH=DH*.5D0 $ K=2*K $ LASTJ=K-1                               6630
DO 10 J=1,LASTJ,2                                           6640
DT=DA+J*DH                                                    6650
DREFPNT=DT-DH $ CALL COMPZ(ZT,DREFPNT,DT,.99,0)             6660
IF(ZT*ZA.GT.0.)GOTO 10                                       6670

```

```

10      NFOUND=2 $ RETURN                                6680
20      CONTINUE                                          6690
20      CONTINUE                                          6700
      NFOUND=0                                           6710
      RETURN                                             6720
      END                                                6730
                                                    6740
      SUBROUTINE SRCH3(DA,ZA,DB,ZB,NFOUND,IDEPTH)         6750
C.....WE TRY TO FIND 3 SIGN CHANGES OF Z(T) IN THE INTERVAL (A,B), 6760
C.....WHERE A AND B ARE GRAM POINTS. SRCH3 IS CALLED ONLY IN CASE 6770
C.....Z(A)*Z(B) < 0. THIS ROUTINE IS ESSENTIALLY DUE TO S.R. LEHMAN. 6780
      IMPLICIT DOUBLE (D)                                6790
      COMMON/BLOC5/EPS,EPSSDBLE,ZTIME,NZEVALU           6800
      IF(ABS(ZA).LT.ABS(ZB))GOTO 10                      6810
      DAA=DB $ ZAA=ZB $ DBB=DA $ ZBB=ZA $ GOTO 20        6820
10      DAA=DA $ ZAA=ZA $ DBB=DB $ ZBB=ZB               6830
20      DH=(DBB-DAA)/IDEPTH                              6840
      ZO=ZAA $ L=0 $ JJ=IDEPTH-1                        6850
      DO 30 J=1,JJ,2                                     6860
      DT=DAA+J*DH                                         6870
      DREFPNT=DT-DH $ CALL COMPZ(ZT,DREFPNT,DT,.99,0)    6880
      IF(ZO*ZT.GT.0.)GOTO 30                             6890
C.....ONE ZERO FOUND.                                   6900
      L=L+1 $ ZO=ZT                                       6910
      IF(L.LT.2)GOTO 30                                   6920
      NFOUND=3 $ RETURN                                  6930
30      CONTINUE                                          6940
      NFOUND=L                                           6950
      RETURN                                             6960
      END                                                6970
                                                    6980
      REAL FUNCTION Z(DT)                                6990
C.....THE RIEMANN-SIEGEL FORMULA (ON SIGMA=1/2) IN SINGLE-PRECISION. 7000
C*****                                                    7010
C*****                                                    7020
C*****                                                    7030
C*****                                                    7040
      IMPLICIT DOUBLE (D)                                7050
      COMMON/MZDZ1/DLN(4025)                             7060
      COMMON/MZDZ2/SQRTINV(4025),PREPCOS(8200),PREPDIF(8200) 7070
C*****                                                    7080
C*****                                                    7090
      COMMON/BLOC1/DPI,DPIINV,DTWOPI,DTWOPIN,DPI8        7100
      COMMON/BLOC3/GRID,GRIDIN                          7110
C.....GRIDIN IS USED ONLY IN CASE COMPASS-ZFUNC IS REPLACED BY ITS 7120
C.....FORTRAN EQUIVALENT GIVEN BELOW ( IN COMMENT LINES ). 7130
      DTAU=DT*DTWOPIN                                    7140
C.....WE ASSUME DTAU TO BE EXACT. THE ERROR IN DT IS ACCOUNTED FOR 7150
C.....IN THE ERROR ANALYSIS.                            7160
      DRHO=DSQRT(DTAU) $ RHOINV=1.DO/DRHO               7170
      M=IDINT(DRHO)                                       7180
C.....WE NOW DETERMINE M SUCH THAT M**2 <= DTAU < (M+1)**2. 7190
      IF(M*M.GT.DTAU)M=M-1                                7200
C.....THIS MIGHT HAPPEN IF DTAU = (K**2)*(1-EPSILON) AND M=K. 7210
      IF(DTAU.GE.(M+1)*(M+1))M=M+1                      7220
C.....THIS MIGHT HAPPEN IF DTAU = (K**2)*(1+EPSILON) AND M=K-1. 7230
      DP=DRHO-M $ RKSI=1.DO-2.DO*DP $ RKSIQ=RKSI*RKSI    7240
C.....HERE WE HAVE USED THE NOTATION OF HASELGROVE AND MILLER. 7250
      DTH=DTHETA(DT)                                     7260
      DH=DTH*DTWOPIN                                     7270

```

```

TETAMOD=DH-IDINT(DH)
SUM=DCOS(DTH)+ZFUNC(M,DT,TETAMOD)
C.....THE PRECEDING "COMPASS EVALUATION" OF THE MAIN SUM IN Z(T) IS
C.....EQUIVALENT TO THE FOLLOWING LOOP:
C.....DO 10 I=2,M
C.....D1=DT*DLN(I)
C.....TLNIMOD=D1-IDINT(D1*DTWOPIN)*DTWOPI
C.....X=ABS(TETAMOD-TLNIMOD) $ XGRIDIN=X*GRIDIN
C.....J=INT(XGRIDIN) $ Q=XGRIDIN-J
C.....ZSUM=ZSUM+(PREPCOS(J+1)+Q*PREPDIF(J+1))*SQRTINV(I)
C10 CONTINUE
C..... ERROR TERM C0
E0=+.000000000000009
E0=(E0*RKSISQ-.00000000000038)*RKSISQ-.00000000001633
E0=(E0*RKSISQ-.00000000003391)*RKSISQ+.00000000178811
E0=(E0*RKSISQ+.00000001235793)*RKSISQ-.00000010354847
E0=(E0*RKSISQ-.00000143272516)*RKSISQ+.00000046556125
E0=(E0*RKSISQ+.00007943300880)*RKSISQ+.00029705353733
E0=(E0*RKSISQ-.00162372532314)*RKSISQ-.01356762197010
E0=(E0*RKSISQ-.01360502604767)*RKSISQ+.13237657548034
E0=(E0*RKSISQ+.43724046807752)*RKSISQ+.38268343236509
C..... ERROR TERM C1
E1=-.000000000000001
E1=(E1*RKSISQ-.00000000000012)*RKSISQ+.00000000000158
E1=(E1*RKSISQ+.00000000002344)*RKSISQ-.00000000007785
E1=(E1*RKSISQ-.00000000270964)*RKSISQ-.00000000446709
E1=(E1*RKSISQ+.00000018334734)*RKSISQ+.00000096413225
E1=(E1*RKSISQ-.00000598024259)*RKSISQ-.00005922748702
E1=(E1*RKSISQ+.00001320794062)*RKSISQ+.00146478085780
E1=(E1*RKSISQ+.00331075976086)*RKSISQ-.00987106629906
E1=(E1*RKSISQ-.03849125048224)*RKSISQ-.01378477342635
E1=(E1*RKSISQ+.02682510262838)*RKSISQ
ERROR=(E1*RHOINV+E0)*SQRT(RHOINV)*((-1)**(M-1))
C..... END OF ERROR COMPUTATION
Z=2.*SUM+ERROR
RETURN
END

DOUBLE FUNCTION DZ(DT)
IMPLICIT DOUBLE (D)
C.....THIS IS OUR DP VERSION OF THE RIEMANN-SIEGEL FORMULA FOR Z(T).
C.....DZ(DT) IS COMPLETELY IN DOUBLE PRECISION ARITHMETIC.
COMMON/MZDZ1/DLN(4025)
COMMON/BLOC1/DPI,DPIINV,DTWOPI,DTWOPIN,DPISL8
COMMON/BLOC7/DCO(30),DC1(30),DC2(30),DC3(30)
DTAU=DT*DTWOPIN $ DTAUINV=1.DO/DTAU
C.....WE ASSUME DTAU TO BE EXACT. THE ERROR IN DT IS ACCOUNTED FOR
C.....IN THE ERROR ANALYSIS
DRHO=DSQRT(DTAU) $ DRHOINV=1.DO/DRHO
M=IDINT(DRHO)
C.....WE NOW DETERMINE M SUCH THAT M**2 <= DTAU < (M+1)**2.
IF(M*M.GT.DTAU)M=M-1
C.....THIS MIGHT HAPPEN IF DTAU = (K**2)*(1-EPSILON) AND M=K.
IF(DTAU.GE.(M+1)*(M+1))M=M+1
C.....THIS MIGHT HAPPEN IF DTAU = (K**2)*(1+EPSILON) AND M=K-1.
DKSI=2.DO*(DRHO-M)-1.DO $ DKSISQ=DKSI*DKSI
DTETADT=DTHETA(DT)
DSUM=0.DO
DO 10 I=1,M
DI=I $ DSUM=DSUM+DCOS(DTETADT-DT*DLN(I))/DSQRT(DI)

```

```

10  CONTINUE
C.....COMPUTATION OF THE ERROR TERM IN THE RIEMANN-SIEGEL FORMULA.
C.....HERE WE USE THE NOTATION OF CRARY AND ROSSER.
DPHI0=DC0(30) $ DPHI1=DC1(30) $ DPHI2=DC2(30) $ DPHI3=DC3(30)
DO 20 I=1,29
K=30-I
DPHI0=DPHI0*DKSISQ+DC0(K)
DPHI1=DPHI1*DKSISQ+DC1(K)
DPHI2=DPHI2*DKSISQ+DC2(K)
DPHI3=DPHI3*DKSISQ+DC3(K)
20  CONTINUE
DPHI1=DPHI1*DKSI*DRHOINV
DPHI2=DPHI2*DTAUINV
DPHI3=DPHI3*DKSI*DTAUINV*DRHOINV
DSIGN=1.DO $ IF(M/2*.EQ.M) DSIGN=-1.DO
DERROR=DSIGN*(DPHI0-DPHI1+DPHI2-DPHI3)/DSQRT(DRHO)
C.....  END OF ERROR COMPUTATION
DZ=2.DO*DSUM+DERROR
RETURN
END

IDENT      ZFUNC
ENTRY      ZFUNC

*
*      SUM = ZFUNC(M, T, TETAMOD)
*

ZFUNC      ENTRY      ZFUNC
BSS        1
SB1        1
SA5        X1          M
SB2        B1+B1
SB3        13          AUX SHIFT FOR NGRID=8192
SB5        PREPCOS
SB6        SQRTINV+1
SA2        A1+B1
SA3        A2+B1
SB4        X5-2        I, INITIALLY M-2
SA1        X2          T.HEAD
SA2        X3          TETAMOD/TWOPI
BX7        X2          COPY
SA2        A1+B1        T.TAIL
LX5        1          M*2
SA3        DTWOPIN
SA4        A3+B1
FX2        X2*X3        T.T*PIINV.H
FX4        X1*X4        T.H*PIINV.T
FX4        X4+X2        T.T*PIINV.H+T.H*PIINV.T
DX2        X1*X3        (T.H*PIINV.H).T
FX0        X1*X3        (T.H*PIINV.H).H
FX6        X2+X4        ADD TAIL PARTS
NX6        X6          NORMALIZE
DX1        X0+X6        T*PI HEAD
FX0        X0+X6        T*PI TAIL
MX6        0          INITIALIZE SUM
SA4        X5+DLN-2     INITIALIZE A4
SA5        A4+B1        PREFETCH X5

*
*      IN THE LOOP THE FOLLOWING REGISTERS ARE DEFINED:
*      B1 = 1
*      B2 = 2

```

```

*      B3 = 13 (BASE2LOG(GRID))      8480
*      B4 = I   COUNTS DOWN          8490
*      B5 = PREPCOS FIRST ELEMENT    8500
*      B6 = SQRTINV+1 B6+B4 = SQRTINV(I) 8510
*      A4 = DLN INDEX, A4-1 IS TAIL OF NEXT DLN 8520
*      X0,X1 = T/TWOPI                8530
*      X6 = SUM                        8540
*      X7 = TETAMOD/TWOPI             8550
*                                     8560
* FIRST WE GIVE THE INTENDED LOOP    8570
*                                     8580
* 1. PICK UP NEXT ELEMENTS FROM DLN  8590
*      SA4      A4-B2                8600
*      SA5      A4+B1                8610
*      NOTE: THESE ELEMENTS ARE ALREADY PREFETCHED AT THE START
*      OF THE ACTUAL LOOP AND THE NEXT ELEMENTS ARE FETCHED NEAR
*      THE END.                      8620
*                                     8630
*                                     8640
*                                     8650
* 2. DOUBLE PRECISION MULTIPLICATION WITH T/TWOPI 8660
*      FX2      X4*X1                8670
*      DX3      X4*X0                8680
*      FX4      X4*X0                8690
*      FX5      X5*X0                8700
*      FX3      X3+X2                8710
*      FX2      X5+X3                8720
*      DX5      X2+X4                8730
*      FX4      X2+X4                8740
*      NOTE: THE LAST TWO INSTRUCTIONS ARE OMITTED IN THE ACTUAL
*      LOOP, AND THE LAST INSTRUCTION BUT TWO IS CHANGED INTO FX5.
*      THESE INSTRUCTIONS ARE NOT NEEDED SINCE IT IS NOT NECESSARY
*      THAT THE VALUE MOD 1 IS ACTUALLY COMPUTED, AS LONG AS WE
*      FIND TWO VALUES, J AND Q WITH J INTEGER AND J+Q IS THE TRUE
*      VALUE. HOWEVER, WE NEED A SLIGHT EXTENSION OF THE ARRAY WITH
*      PRECOMPUTED COSINES, SINCE THE INDEX MAY BE OUT OF THE RANGE
*      [ 0,8191 ]. HOWEVER, ONLY A FEW ELEMENTS MORE ARE NEEDED.
*                                     8750
*                                     8760
*                                     8770
*                                     8780
*                                     8790
*                                     8800
*                                     8810
*                                     8820
*                                     8830
* 3. COMPUTE FRACTION MODULO 1.      8840
*      UX2      B7,X4                8850
*      LX2      B7,X2                8860
*      AX2      B7,X2                8870
*      BX4      X4-X2                8880
*      THE LAST OPERATION IS CORRECT BECAUSE X4 >= 0. 8890
*                                     8900
* 4. NORMALIZE, ADD HEAD AND TAIL.  8910
*      NX3      X5                    8920
*      NX4      X4                    8930
*      FX2      X4+X3                8940
*      NOW X2 SHOULD BE < 1. HOWEVER, SEE THE NOTE IN STEP 2. 8950
*                                     8960
* 5. SUBTRACT FROM TETAMOD.          8970
*      FX2      X7-X2                8980
*      NORMALIZATION IS NOT YET NEEDED. 8990
*                                     9000
* 6. COMPUTE INTEGER PART AND FRACTIONAL PART OF 9010
*      ABSOLUTE VALUE * 8192.        9020
*      UX5      B7,X2                9030
*      SB7      B7+B3                9040
*      LX5      B7,X5                9050
*      BX4      X2                    9060
*      AX4      60                    9070
*                                     SIGN EXTENDED

```

```

*          BX2          X2-X4          ABSOLUTE VALUE          9080
*          BX5          X5-X4          ABSOLUTE VALUE OF INTEGER PART 9090
*          PX2          B7,X2          STORE NEW EXPONENT          9100
*          AX4          B7,X5          REPOSITION INTEGER PART      9110
*          BX4          X2-X4          SUBTRACT                      9120
*          NX4          X4            AND NORMALIZE                  9130
*      IN THIS CODE WE DON'T FIRST COMPUTE ABSOLUTE VALUE.          9140
*      IF WE DID SO WE MIGHT SAVE ONE INSTRUCTION (TAKING          9150
*      ABSOLUTE VALUE OF INTEGER PART IS NOT NEEDED). HOWEVER,      9160
*      IT IS NOT POSSIBLE TO DO PARALLEL COMPUTATIONS, SO THIS      9170
*      CODE IS FASTER.                                              9180
*
*      7. PICK UP COSINE AND COSINE DIFFERENCE                      9190
*          SA3          X5+CC          COSINE DIFFERENCE            9200
*          SA5          X5+B5          COSINE                        9210
*
*      8. COMPUTE INTERPOLATED COSINE.                               9220
*          FX3          X3*X4          9230
*          FX3          X3+X5          9240
*          NX3          X3            9250
*
*      9. PICK UP INVERSE OF SQRT(I) AND MULTIPLY.                 9260
*          SA2          B4+B6          9270
*          FX2          X2*X3          9280
*
*      10. ADD TO ACCUMULATIVE SUM.                                  9290
*          FX6          X6+X2          9300
*          NX6          X6            9310
*
*      11. COUNT AND POSSIBLY JUMP BACK.                             9320
*          SB4          B4-B1          9330
*          GE          B4,LOOP         9340
*
*      THE ACTUAL LOOP CONTAINS THE INSTRUCTIONS ABOVE BUT JUGGLED 9350
*      AROUND TO GET THE UTMOST FROM THE PARALLEL PROCESSING OF THE 9360
*      FUNCTIONAL UNITS OF THE CYBER-175.                            9370
*
*      LOOP          BSS          0                                  9380
*      INSTRUCTION    CYCLE        BUSY REGISTERS (CYCLES STILL BUSY) 9390
*      FX2 X4*X1       0           X2( 5),X5( 4)                      9400
*      *              1           X2( 4),X5( 3)                      9410
*      DX3 X4*X0       2 X3( 5),X2( 3),X5( 2)                        9420
*      *              3 X3( 4),X2( 2),X5( 1)                        9430
*      FX4 X4*X0       4 X3( 3),X2( 1),X5( 1),X4( 5)                 9440
*      *              5 X3( 2),X4( 4)                                9450
*      FX5 X5*X0       6 X3( 1),X5( 5),X4( 3)                        9460
*      FX3 X3+X2       7 X3( 4),X5( 4),X4( 2)                        9470
*      *              8 X3( 3),X5( 3),X4( 1)                        9480
*      UX2 B7,X4       9 X3( 2),X5( 2),X2( 2),B7( 2)                 9490
*      *             10 X3( 1),X5( 1),X2( 1),B7( 1)                 9500
*      LX2 B7,X2      11 X2( 2)                                     9510
*      FX5 X5+X3      12 X2( 1),X5( 4)                               9520
*      AX2 B7,X2      13 X2( 2),X5( 3)                               9530
*      *             14 X2( 1),X5( 2)                               9540
*      BX4 X4-X2      15 X4( 2),X5( 1)                               9550
*      NX3 X5,B0      16 X4( 1),X3( 3)                               9560
*      NX4 X4,B7      17 X4( 3),X3( 2),B7( 3)                        9570
*      *             18 X4( 2),X3( 1),B7( 2)                        9580
*      *             19 X4( 1),B7( 1)                               9590
*      FX2 X4+X3      20 X2( 4)                                     9600
*
*      *** WORD 1
*      **MULTIPLY BUSY
*      *** WORD 2
*      *** WORD 3
*      *** WORD 4

```

*	21	X2(3)	.	9680
*	22	X2(2)	.	9690
*	23	X2(1)	.	9700
FX2 X7-X2	24	X2(4)	.	9710
*	25	X2(3)	.	9720
*	26	X2(2)	.	9730
*	27	X2(1)	.	9740
UX5 B7,X2	28	X5(2),B7(2)	.	9750
BX4 X2	29	X5(1),B7(1),X4(2)	.	9760
SB7 B7+B3	30	B7(2),X4(1)	*** WORD 5	9770
AX4 60	31	B7(1),X4(2)	.	9780
LX5 B7,X5	32	X5(2),X4(1)	.	9790
BX2 X2-X4	33	X5(1),X2(2)	.	9800
BX5 X5-X4	34	X5(2),X2(1)	*** WORD 6	9810
PX2 B7,X2	35	X5(1),X2(2)	.	9820
SA3 X5+CC	36	X3(23),X2(1),A3(2)	.	9830
AX4 B7,X5	37	X3(22),X4(2),A3(1)	*** WORD 7	9840
SA5 X5+B5	38	X3(21),X4(1),X5(23),A5(2)	.	9850
BX4 X2-X4	39	X3(20),X4(2),X5(22),A5(1)	.	9860
SA2 B4+B6	40	X3(19),X4(1),X5(21),X2(23),A2(2)	.	9870
NX4 X4,B0	41	X3(18),X4(3),X5(20),X2(22),A2(1)	*** WORD 8	9880
*	42	X3(17),X4(2),X5(19),X2(21)	.	9890
*	43	X3(16),X4(1),X5(18),X2(20)	.	9900
*	44	X3(15),X5(17),X2(19)	.	9910
*	45	X3(14),X5(16),X2(18)	.	9920
*	46	X3(13),X5(15),X2(17)	.	9930
*	47	X3(12),X5(14),X2(16)	.	9940
*	48	X3(11),X5(13),X2(15)	.	9950
*	49	X3(10),X5(12),X2(14)	.	9960
*	50	X3(9),X5(11),X2(13)	.	9970
*	51	X3(8),X5(10),X2(12)	.	9980
*	52	X3(7),X5(9),X2(11)	.	9990
*	53	X3(6),X5(8),X2(10)	.	10000
*	54	X3(5),X5(7),X2(9)	.	10010
*	55	X3(4),X5(6),X2(8)	.	10020
*	56	X3(3),X5(5),X2(7)	.	10030
*	57	X3(2),X5(4),X2(6)	.	10040
*	58	X3(1),X5(3),X2(5)	.	10050
FX3 X3*X4	59	X3(5),X5(2),X2(4)	.	10060
SA4 A4-B2	60	X3(4),X4(23),X5(1),X2(3),A4(2)	.	10070
*	61	X3(3),X4(22),X2(2),A4(1)	.	10080
*	62	X3(2),X4(21),X2(1)	.	10090
*	63	X3(1),X4(20)	.	10100
FX3 X5+X3	64	X3(4),X4(19)	.	10110
SA5 A4+B1	65	X3(3),X4(18),X5(23),A5(2)	*** WORD 9	10120
SB4 B4-B1	66	X3(2),X4(17),X5(22),A5(1),B4(2)	.	10130
*	67	X3(1),X4(16),X5(21),B4(1)	.	10140
NX3 X3,B0	68	X3(3),X4(15),X5(20)	.	10150
*	69	X3(2),X4(14),X5(19)	.	10160
*	70	X3(1),X4(13),X5(18)	.	10170
FX2 X2*X3	71	X2(5),X4(12),X5(17)	.	10180
*	72	X2(4),X4(11),X5(16)	.	10190
*	73	X2(3),X4(10),X5(15)	.	10200
*	74	X2(2),X4(9),X5(14)	.	10210
*	75	X2(1),X4(8),X5(13)	.	10220
FX6 X6+X2	76	X6(4),X4(7),X5(12)	*** WORD 10	10230
*	77	X6(3),X4(6),X5(11)	.	10240
*	78	X6(2),X4(5),X5(10)	.	10250
*	79	X6(1),X4(4),X5(9)	.	10260
NX6 X6,B0	80	X6(3),X4(3),X5(8)	.	10270

```

      GE B4,LOOP 81 X6( 2),X4( 2),X5( 7) . 10280
      *          82 X6( 1),X4( 1),X5( 6) . 10290
      *          83* X5( 5) . 10300
      * 10310
      *      END LOOP 10320
      * 10330
      *      EQ ZFUNC 10340
      * 10350
      *      USE /MZDZ1/ 10360
      DLN BSS 4025*2 10370
      * 10380
      *      USE /MZDZ2/ 10390
      SQRINV BSS 4025 10400
      PREPCOS BSS 8200 10410
      CC BSS 8200 10420
      * 10430
      *      USE /BLOC1/ 10440
      DPI BSS 2 10450
      DPIINV BSS 2 10460
      DTWOPI BSS 2 10470
      DTWOPIIN BSS 2 10480
      DPISL8 BSS 2 10490
      * 10500
      *      END 10510
      * 10520
      SUBROUTINE STATIST(K) 10530
      COMMON/BLOC0/INTRVAL(10,10) 10540
      COMMON/BLOCA/NBLOCL(10) 10550
      DIMENSION NAUX(10) 10560
      WRITE(1,10) 10570
      10 FORMAT(*1CUMULATIVE STATISTICS (ALSO WRITTEN ON PF RIESTAT) *) 10580
      REWIND 2 10590
      READ(2,20)N1,N2 10600
      20 FORMAT(2I12) 10610
      WRITE(3,20)N1,K 10620
      WRITE(1,30)N1,K 10630
      30 FORMAT(* RANGE:*,2I12) 10640
      DO 60 I=1,10 10650
      READ(2,50)IH,NTOT,(NAUX(J),J=1,I) 10660
      NTOT=NTOT+NBLOCL(I) 10670
      DO 40 L=1,I 10680
      NAUX(L)=NAUX(L)+INTRVAL(I,L) 10690
      40 CONTINUE 10700
      WRITE(3,50)IH,NTOT,(NAUX(J),J=1,I) 10710
      WRITE(1,50)IH,NTOT,(NAUX(J),J=1,I) 10720
      50 FORMAT(I3,3I10,I9,I8,I7,I6,4I4) 10730
      60 CONTINUE 10740
      RETURN 10750
      END 10760
      * 10770
      SUBROUTINE SRCH2D(DA,ZA,DT,ZT,DB,ZB,NFOUND,INTVAL) 10780
      C.....WE "MUST" FIND TWO ZEROS BETWEEN DA AND DB. 10790
      C.....SRCH2D IS CALLED ONLY IN CASE ABS(Z(T)) IS CONVEX IN THE 10800
      C.....INTERVAL [ DA,DB ] WITH ZA*ZB > 0. 10810
      C.....WE FIRST COMPUTE Z(.) IN THE EXTREMUM OF THE PARABOLIC FIT. 10820
      IMPLICIT DOUBLE (D) 10830
      COMMON/BLOC5/EPS,EPSDBLE,ZTIME,NZEVALU 10840
      DTT=DT $ ZTT=ZT 10850
      C.....IF CALLED BY PARA WE MAY HAVE DA > DB. INTVAL IS IRRELEVANT THEN. 10860
      IF(DA.LT.DB)GOTO 10 10870

```

```

      DAA=DB $ ZAA=ZB                                10880
      DBB=DA $ ZBB=ZA $ GOTO 20                      10890
10     DAA=DA $ ZAA=ZA                                10900
      DBB=DB $ ZBB=ZB                                10910
20     KOUNT=0                                         10920
30     KOUNT=KOUNT+1 $ IF(KOUNT.GT.12)GOTO 100        10930
C.....COMPUTATION OF THE PARABOLIC FIT  $F(X)=C1*X**2+C2*X+C3$ . 10940
      Q1=(ZAA-ZTT)/(DAA-DTT) $ Q2=(ZTT-ZBB)/(DTT-DBB) 10950
      C1=(Q2-Q1)/(DBB-DAA) $ C2=Q1-C1*(DAA+DTT)       10960
C.....THE PARABOLIC FIT HAS ITS EXTREMUM IN TNEW.    10970
      DTNEW=-.5*C2/C1                                10980
C.....THE NEXT LINE IS INSERTED FOR SAFETY REASONS. 10990
      IF((DTNEW.LE.DAA).OR.(DTNEW.GE.DBB))GOTO 120    11000
      IF(DTNEW.NE.DTT)GOTO 50                        11010
      IF(ABS(ZAA).GT.ABS(ZBB))GOTO 40                 11020
      DTNEW=(DAA+DTT)*.5DO $ GOTO 50                 11030
40     DTNEW=(DTT+DBB)*.5DO                          11040
50     DREFPNT=DTT $ CALL COMPZ(ZTNEW,DREFPNT,DTNEW,.99,0) 11050
      IF(ZTNEW*ZA.LT.0.)GOTO 90                      11060
      IF(DTNEW.GT.DTT)GOTO 70                        11070
C.....WE ARE IN THE LEFT-HAND INTERVAL [ DAA,DTT ]. 11080
      IF(ABS(ZTNEW).GT.ABS(ZTT))GOTO 60              11090
      DBB=DTT $ ZBB=ZTT                              11100
      DTT=DTNEW $ ZTT=ZTNEW $ GOTO 30                 11110
60     DAA=DTNEW $ ZAA=ZTNEW $ GOTO 30               11120
C.....WE ARE IN THE RIGHT-HAND INTERVAL [ DTT,DBB ]. 11130
70     IF(ABS(ZTNEW).GT.ABS(ZTT))GOTO 80             11140
      DAA=DTT $ ZAA=ZTT                              11150
      DTT=DTNEW $ ZTT=ZTNEW $ GOTO 30                 11160
80     DBB=DTNEW $ ZBB=ZTNEW $ GOTO 30               11170
90     NFOUND=2                                       11180
C.....NOTE THAT INTVAL IS RELEVANT ONLY IF DA < DB. 11190
      INTVAL=1 $ IF(DTNEW.GT.DT)INTVAL=2 $ RETURN    11200
100    WRITE(1,110)DAA,DBB                          11210
110    FORMAT(/,* DIFFICULTIES IN SRCH2D BETWEEN *,/, 11220
      $* AA=*,F26.12,* AND BB=*,F26.12,/)           11230
      STOP                                           11240
120    WRITE(1,130)DAA,DBB                          11250
130    FORMAT(* IN SRCH2D TNEW LIES OUTSIDE THE INTERVAL *,/, 11260
      $* AA=*,F26.14,* BB=*,F26.14,* WE STOP *)     11270
      STOP                                           11280
      END                                           11290

```

4. REFERENCES

- [1] BRENT, R.P., J. VAN DE LUNE, H.J.J. TE RIELE & D.T. WINTER, *On the zeros of the Riemann zeta function in the critical strip, II*, to appear in Math. Comp., 1982.
- [2] KARKOSCHKA, E. & P. WERNER, *Einige Ausnahmen zur Rossumerschen Regel in der Theorie der Riemannschen Zetafunktion*, Computing, 27(1981) pp. 57-69.
- [3] LUNE, J. VAN DE, H.J.J. TE RIELE & D.T. WINTER, *Rigorous high speed separation of zeros of Riemann's zeta function*, Report NW 113/81, October 1981, Mathematical Centre, Amsterdam.