STICHTING
MATHEMATISCH CENTRUM
2e BOERHAAVESTRAAT 49
AMSTERDAM

MR 60

A Numerical Study of a Result of Stieltjes

P. Wynn

1963

REVUE FRANÇAISE DE TRAITEMENT DE L'INFORMATION. 1963-3

A NUMERICAL STUDY OF A RESULT OF STIELTJES (*)

par P. WYNN

Centre Mathématique d'Amsterdam

Summary. — « This paper deals, paying special attention to details of programming, with the asymptotic expansion of a certain class of integrals. The steps to be taken in obtaining a Laplace type asymptotic expansion by the method of steepest descents are described. The theory is illustrated by obtaining such an expansion for the remainder term of the exponential integral of complex argument, a similar treatment of which has been given by Stieltjes for real argument. »

I. — INTRODUCTION

Asymptotic Expansions by the Method of Steepest Descents

In this paper we shall consider, paying particular attention to the computational details, the expansion by the method of steepest descents of a certain class of integrals. The integrals which we consider are of the form

$$F(\rho) = \int_{\alpha^*}^{\alpha^{**}} G(t)\, e^{\rho H(t)}\, dt \tag{1}$$

We suppose for the purposes of exposition that $H(t)$ has one maximum point in the whole complex plane (the more general case may easily be treated by an extension of the following work) at $t = \alpha_H$ and that the integral (1) may be so transformed that the path of integration lies along the contour (the steepest path)

$$\operatorname{Im}\{ H(t) \} = \text{constant} = \operatorname{Im}\{ H(\alpha_H) \}. \tag{2}$$

(*) Communication MR 60 of the Computation Department of the Mathematical Centre, Amsterdam.

We suppose that (1) then evolves to the form

$$F(\rho) = \int_{\alpha'}^{\alpha''} g(t)\, e^{\rho h(t)}\, dt \tag{3}$$

(t being real) in which $h(t)$ has a single maximum at the point $t = \alpha$, where $\alpha' \leqslant \alpha \leqslant \alpha''$ (1).

We write (3) as

$$\begin{aligned} F(\rho) &= \left[\int_{\alpha}^{\alpha''} - \int_{\alpha}^{\alpha'}\right] g(t)\, e^{\rho h(t)}\, dt \\ &= I_1 - I_2 \end{aligned} \tag{4}$$

Consider the asymptotic expansion of the integral

$$I_1 = \int_{\alpha}^{\alpha''} g(t)\, e^{\rho h(t)}\, dt \tag{5}$$

this may be carried out as follows (the presentation is freely adapted from Erdélyi [1]). We change the variable of integration from t to u, where

$$u = h(\alpha) - h(t) \sim \sum_{s=0}^{\infty} a_s t'^{\nu+s} \tag{6}$$

and derive the inverse expansion

$$t - \alpha = t' = \sum_{s=0}^{\infty} b_s u^{(s+1)/\nu} \tag{7}$$

We expand $-g(t)/h'(t)$ as an ascending power series in $u^{1/\nu}$ and obtain

$$-\frac{g(t)}{h'(t)} \sim \sum_{s=0}^{\infty} \gamma_s u^{(\lambda+s-\nu)/\nu} \tag{8}$$

say. The integral (3) then has the formal asymptotic expansion

$$I_1 \sim e^{\rho h(\alpha)} \sum_{s=0}^{\infty} \gamma_s\, \Gamma\left(\frac{\lambda+s}{\nu}\right) \rho^{-(\lambda+s)/\nu} \tag{9}$$

(1) The dummy variable t in (1) and (3) does not necessarily have the same meaning in both cases.

The integral I_2 may be treated in the same way.

Thus it is required that we should be able to do three things. We must be able to derive the coefficients $b_s (s = 0, 1, \ldots)$ in (5) by means of some recursive scheme. We must be able to do likewise for the coefficients $c_s (s = 0, 1, \ldots)$ in the expansion

$$- \frac{g(t)}{h'(t)} \sim \sum_{s=0}^{\infty} c_s t'^{(\lambda - \nu + s)} \tag{10}$$

We must be able to obtain the coefficients $b_{r,s}$ in the successive expansions

$$t'^{r} \sim \sum_{s=0}^{\infty} b_{r,s} u^{(r+s)/\nu} \qquad (r = -\lambda, -\lambda + 1, \ldots) \tag{11}$$

and finally we must be able to mechanise the substitution of the expansions (11) in (10) to obtain the coefficients $\gamma_s (s = 0, 1, \ldots)$ in (6).

In many cases, function-theoretic properties of $h(t)$ and $g(t)$ greatly facilitate the execution of the process mentioned above. We consider a simple, but in principle a typical, example.

II. — THE EXPONENTIAL INTEGRAL

The integral

$$\int_0^{\infty} \frac{e^{-zt}}{1+t} \, dt \tag{12}$$

may, with the help of the formula

$$\frac{1}{1+t} = 1 - t + t^2 - t^3 + \ldots + (-t)^{n-1} + \frac{(-t)^n}{1+t} \tag{13}$$

be transformed into

$$\int_0^{\infty} \frac{e^{-zt}}{1+t} \, dt = \sum_{r=0}^{n-1} u_r + R_n, \tag{14}$$

that is to say the sum of n terms and a remainder, where

$$u_r = \frac{(-1)^r r!}{z^{r+1}} \tag{15}$$

and

$$(16) \qquad (-1)^n R_n = \int_0^\infty \frac{e^{-zt}}{1+t} t^n \, dt.$$

We now make the substitution

$$(17) \qquad \rho = n + \eta$$

where

$$(18) \qquad z = \rho \, e^{i\theta}, \; \beta = e^{i\theta}$$

in (16) and obtain

$$(19) \qquad (-1)^n R_n = \int_0^\infty \frac{e^{-\beta\eta t}}{1+t} e^{\rho \ln(te^{-\beta t})} \, dt.$$

A. — The Non-Singular case

We suppose initially that in (16) and (19) $\theta \neq \pi$.
In the notation of the preceding section

$$(20) \qquad H(t) = \ln \{ t \, e^{-\beta t} \}$$

We have

$$(21) \qquad H'(t) = (1 - \beta t) / t$$

and $H(t)$ has a maximum at the point $t = \beta^{-1}$.
The steepest path is given by

$$(22) \qquad \mathrm{Im} \{ \ln (t) - \beta t \} = \mathrm{Im} \{ \ln (\beta^{-1}) - 1 \} = -\theta.$$

Suppose that

$$(23) \qquad t = r \, e^{i\varphi}$$

then (22) becomes

$$(24) \qquad \Phi + \theta = r \sin (\Phi + \theta),$$

the appropriate solution of which is

$$(25) \qquad \theta = -\Phi.$$

Thus the steepest path is the ray from the origin through the point β^{-1}. The integral (19) may thus be transformed into

$$(-1)^n e^{in\theta} R_n(z) = \int_0^\infty \frac{e^{-\eta t}}{\beta + t} e^{n\{\ln(t)-t\}} dt; \tag{26}$$

in the notation of the equation (5)

$$h(t) = \ln(t) - t, \tag{27}$$

$$h'(t) = \frac{1}{t} - 1, \tag{28}$$

and

$$\alpha = 1. \tag{29}$$

We consider the integral

$$\int_1^\infty \frac{e^{-\eta t}}{\beta + t} e^{n\{\ln(t)-t\}} dt, \tag{30}$$

write

$$t' = t - 1, \tag{31}$$

and note that

$$u = t' - \ln(1 + t') \tag{32}$$

$$\sim \frac{t'^2}{2} - \frac{t'^3}{3} + \ldots \tag{33}$$

We obtain the coefficients in the expansion of t' in powers of $u^{\frac{1}{2}}$ [for evidently from (33) $\nu = 2$] from the differential equation which t' satisfies as a function of u. For, from (32),

$$1 + t' = t' \, dt'/du \tag{34}$$

Writing

$$t' = \sum_{s=0}^\infty b_s u^{(s+1)/2} \tag{35}$$

then, from (34)

$$\left\{1 + \sum_{s=0}^\infty b_s u^{(s+1)/2}\right\} = \left\{\sum_{s=0}^\infty b_s u^{(s+1)/2}\right\}\left\{\sum_{s=0}^\infty \left(\frac{s+1}{2}\right) b_s u^{(s-1)/2}\right\} \tag{36}$$

Multiplying the power series on the right hand side of (36) and equating coefficients of corresponding powers of u we have

(37) $$b_0 = \sqrt{2}$$

(38) $$b_s = \left\{ b_{s-1} - \sum_{r=0}^{s-2} b_{r+1} \binom{r+2}{2} b_{s-r+1} \right\} \frac{\sqrt{2}}{s+2} \qquad (s = 1, 2, \ldots)$$

In particular

(39) $$b_1 = \frac{2}{3},\ b_2 = \frac{\sqrt{2}}{18},\ b_3 = -\frac{2}{135},\ b_4 = \frac{\sqrt{2}}{1\,080}, \ldots$$

In the notation of equation (5)

(40) $$-\frac{g(t)}{h'(t)} = -\frac{e^{-\eta t}}{\beta + t} \frac{t}{1-t}$$

Letting

(41) $$-e^{\eta} \frac{g(t)}{h'(t)} = \sum_{s=0}^{\infty} c_s t'^{s-1},$$

we have

(42) $$\{ \beta t' + t' + t'^2 \} \sum_{s=0}^{\infty} c_s t'^s = (1 + t')\, e^{-\eta t'}$$

i. e.

(43) $$(1+\beta) c_0 + \sum_{s=1}^{\infty} \{ (1+\beta) c_s + c_{s-1} \} t'^s$$
$$= \sum_{s=1}^{\infty} \left\{ \frac{(-\eta)^s}{s!} + \frac{(-\eta)^{s-1}}{(s-1)!} \right\} t'^s + 1$$

Thus the coefficients c_s may be determined from

(44) $$c_0 = (\beta + 1)^{-1},$$
$$c_{s+1} = \left\{ (-1)^s \eta^s \frac{(s+1-\eta)}{(s+1)!} - c_s \right\} / (1+\beta) \qquad (s = 0, 1, \ldots)$$

In particular

(45) $$c_1 = \left\{ -\eta + \frac{\beta}{1+\beta} \right\} / (1+\beta),$$

(46) $$c_2 = \left\{ \frac{\eta^2}{2} - \frac{\beta}{1+\beta} \eta - \frac{\beta}{(1+\beta)^2} \right\} / (1+\beta).$$

From (41) it is evident that in the notation of equation (10) $\lambda = 1$, and for this reason we require the coefficients $b_{-1,s}$ in the expansion

$$(47) \qquad \left\{ \sum_{s=0}^{\infty} b_{-1,s} u^{(s-1)/2} \right\}^{-1} = \left\{ \sum_{s=0}^{\infty} b_s u^{(s+1)/2} \right\}$$

and obtain, by multiplying these two series together,

$$(48) \quad b_{-1,0} = (\sqrt{2})^{-1},\ b_{-1,s} = \left(-\sum_{r=1}^{s} b_r b_{-1,s-r} \right) / \sqrt{2} \quad (s = 1, 2, \ldots)$$

In particular

$$(49) \quad b_{-1,1} = -\frac{1}{3},\ b_{-1,2} = \frac{\sqrt{2}}{12},\ b_{-1,3} = -\frac{4}{125},\ b_{-1,4} = \frac{\sqrt{2}}{432}, \ldots$$

Finally we wish to obtain the coefficients of $u^{\frac{1}{2}}$ in t'^r given by

$$(50) \qquad \sum_{s=0}^{\infty} b_{r,s} u^{(s+2)/2} = \left\{ \sum_{s=0}^{\infty} b_s u^{(s+1)/2} \right\}^r$$

We have of course

$$(51) \qquad b_{1,s} = b_s \qquad (s = 0, 1, \ldots)$$

and thereafter

$$(52) \quad b_{r,s} = \sum_{m=0}^{s} b_m b_{r-1,s-m} \qquad (r = 2, 3, \ldots ;\ s = 0, 1, \ldots)$$

In particular

$$(53) \qquad \begin{array}{llll} b_{2,0} = 2, & b_{2,1} = \frac{4}{3}\sqrt{2}, & b_{2,2} = \frac{2}{3}, & \ldots \\ & b_{3,0} = 2\sqrt{2}, & b_{3,1} = 4, & \ldots \\ & & b_{4,0} = 4, & \ldots \end{array}$$

The integral I_2 may be treated in the same way. We assemble the results and obtain

$$(54) \quad R_n(z) \sim (-1)^n\, e^{-in\theta}\, e^{-\rho}\, 2\sqrt{\frac{\pi}{n}} \sum_{r=0}^{\infty} \gamma_{2r} \left(\frac{1}{2}\right)\left(\frac{3}{2}\right) \cdots \left(r - \frac{1}{2}\right) n^{-r}$$

where

$$(55) \qquad \gamma_{2r} = \sum_{s=-1}^{2r} c_{s+1} b_{s,2r-s} \qquad (r = 0, 1, \ldots)$$

Substituting the initial values of the various coefficients which we have derived as a check, we obtain, when $\beta = 1$,

$$(56) \qquad R_n(z) = (-1)^n e^{-z} \sqrt{\frac{2\pi}{n}} \left\{ \frac{1}{2} + \left(\frac{\eta^2}{4} - \frac{\eta}{4} - \frac{12}{1} \right) n^{-1} + \dots \right\}$$

in agreement with Stieltjes [2] who considered the expansion of the integral (19) when $\beta = 1$ and $\beta = -1$.

B. — The Singular Case

When $\theta = \pi$ we find that $g(t)$ ($\equiv e^{-\eta t}/\beta + t$) has a pole at the point $t = 1$. Thus the analysis of the preceding section must be modified. The function with which we are now dealing is

$$(57) \qquad \int_0^\infty \frac{e^{-z't}}{1-t}\, dt$$

for real positive $z' (\equiv -z)$. This may be given a meaning (the Cauchy principal value) by regarding it as the limit as δ tends to zero of the sum

$$(58) \qquad \int_0^{1-\delta} \frac{e^{-z't}}{1-t}\, dt + \int_{1+\delta}^{\infty} \frac{e^{-z't}}{1-t}\, dt$$

This decomposition of the integral (57) into the sum of two integrals as in (58) corresponds precisely to the decomposition (4).

The cancellation of the odd powers of $n^{-r/2}$ to form the analogue of expression (54) removes those components of the sum (58) which become formally infinite as δ tends to zero.

In the notation of equation (5) we have

$$(59) \qquad -\frac{g(t)}{h'(t)} = -\frac{e^{-\eta t}}{(t-1)^2}$$

and in the notation of equation (7)

$$(60) \qquad t' = t - 1 \;;$$

thus equation (10) becomes in this case

$$(61) \qquad -\frac{g(t)}{h'(t)} = e^{-\eta} t'^{-2} \left\{ 1 + \sum_{s=1}^{\infty} \left\{ \frac{(-\eta)^s}{s\,!} + \frac{(-\eta)^{s-1}}{(s-1)\,!} \right\} t'^s \right.$$

i.e.

$$(62)\quad c_0 = 1,\ c_1 = -\eta + 1,\ c_2 = \frac{\eta^2}{2} - \eta,\ c_3 = -\frac{\eta^3}{6} + \frac{\eta^2}{2}, \ldots$$

Examination of expansion (61) reveals that in the notation of equation (10), $\lambda = 0$. Thus we require the coefficients $b_{-2,s}$ ($s = 1, 2, \ldots$) in the expansion

$$(63)\qquad \sum_{s=0}^{\infty} b_{-2,s} u^{(s-2)/2} = \left\{ \sum_{s=0}^{\infty} b_s u^{(s+1)/2} \right\}^{-2},$$

where the coefficients $b_s (s = 0, 1, \ldots)$ are those of equations (35)-(39).

We obtain the coefficients $b_{-2,s} (s = 1, 2, \ldots)$ from the relationship

$$(64)\qquad \sum_{s=0}^{\infty} b_{-2,s} u^{(s-2)/2} = \left\{ \sum_{s=0}^{\infty} b_{-1,s} u^{(s-1)/2} \right\}^{2}$$

where the $b_{-1,s} (s = 0, 1, \ldots)$ are given by (48). Thus, from (64)

$$(65)\qquad b_{-2,s} = \sum_{m=0}^{s} b_{-1,m} b_{-1,s-m} \quad (s = 1, 2, \ldots)$$

In particular

$$(66)\quad b_{-2,1} = -\frac{\sqrt{2}}{3},\quad b_{-2,2} = \frac{5}{18},\quad b_{-2,3} = -\frac{23\sqrt{2}}{270}, \ldots$$

Forming the sum (55) as far as the first two terms, we have in the singular case

$$(67)\quad R_n(z) = e^{-z} \sqrt{\frac{2\pi}{n}} \left\{ \eta - \frac{1}{3} + \left(\frac{1}{6}\eta^3 - \frac{1}{2}\eta^2 + \frac{1}{12}\eta + \frac{1}{540} \right) \frac{1}{n} \cdots \right\}$$

in agreement with Stieltjes ([2] p. 213) who considered this case in detail.

C. — Application of the ε-algorithm

We have now shown how the remainder term R_n may be expressed formally as the sum of a series. But it is a matter of numerical experience that in many cases a continued fraction which may in a certain sense be associated with a given power series, converges far more rapidly than the series. We wish, therefore to transform the series for R_n into such a continued fraction. This may conveniently be done by application of the ε- algorithm [3], the theory of which has been described elsewhere [4] ; it will suffice here to say that if from the initial values

$$\varepsilon_0^{(0)} = 0, \quad \varepsilon_0^{(m)} = \sum_{s=1}^{m-1} t_r \qquad (m = 1, 2, \ldots) \tag{68}$$

$$\varepsilon_1^{(m)} = t_m^{-1} \qquad (m = 0, 1, \ldots) \tag{69}$$

where

$$t_r = \mathrm{e}^{-\rho}\, \mathrm{e}^{-in\theta}\, 2\sqrt{\frac{\pi}{n}}\, \gamma_{2r} \left(\frac{1}{2}\right)\left(\frac{3}{2}\right) \cdots \left(r - \frac{1}{2}\right) \eta^{-r} \tag{70}$$

further quantities $\varepsilon_s^{(m)} (m = 0, 1, \ldots ; s = 2, 3, \ldots)$ are constructed by means of the relationship

$$\varepsilon_s^{(m)} = \varepsilon_{s-2}^{(m+1)} + \frac{1}{\varepsilon_{s-1}^{(m+1)} - \varepsilon_{s-1}^{(1)}} \tag{71}$$

then the quantities $\varepsilon_{2s}^{(m)}$ are convergents of certain continued fractions, and as such provide better estimates of the formal sum of the series whose partial sums are given by (61) than these partial sums (see for example [5] and [6]).

The quantities $\varepsilon_s^{(m)}$ may be displayed in the array

TABLE I

$$\begin{array}{ccccc}
 & \varepsilon_0^{(0)} & & & \\
\varepsilon_{-1}^{(1)} & & \varepsilon_1^{(0)} & & \\
 & \varepsilon_0^{(1)} & & \varepsilon_2^{(0)} & \\
\varepsilon_{-1}^{(2)} & & \varepsilon_1^{(1)} & & \varepsilon_3^{(0)} \\
 & \varepsilon_0^{(2)} & & \varepsilon_2^{(1)} & . \\
\varepsilon_{-1}^{(3)} & & \varepsilon_1^{(2)} & & . \\
. & \varepsilon_0^{(3)} & . & . & . \\
. & . & . & . & .
\end{array}$$

and it can be seen that the quantities in (64) occur at the vertices of a lozenge in this array. The various members of this array are most economically (with regard to storage space) computed by retaining a vector l which at a given stage contains the following quantities :

$$l_0 \equiv \varepsilon_0^{(m)},\ l_1 \equiv \varepsilon_1^{(m-1)},\ l_2 \equiv \varepsilon_2^{(m-2)},\ \ldots,\ l_m = \varepsilon_m^{(0)}.$$

(This corresponds to what, in a table of a function and its differences, would be a line of backward differences). We arrive with a new partial sum $\varepsilon_0^{(m+1)}$ and replace in succession l_0 by $\varepsilon_0^{(m+1)}$, l_1 by $\varepsilon_1^{(m)}$, ..., l_m by $\varepsilon_m^{(1)}$, and add $l_{m+1} \equiv \varepsilon_{m+1}^{(0)}$. The formation of these quantities is carried out by means of (71) and uses one working space and two auxiliary storage locations.

III. — AN ALGOL PROGRAMME

We now summarise, in the form of an ALGOL Programme, the formalism which has been developed. The programme to be given computes the terms in the series

$$(65)\quad R_n(z) = e^{-\rho}\, e^{-in\theta}\, 2\sqrt{\frac{\pi}{n}} \sum_{r=0}^{r\max} \gamma_{2r} \left(\frac{1}{2}\right)\left(\frac{3}{2}\right) \cdots \left(r-\frac{1}{2}\right) n^{-r};$$

it is assumed that r max is given.

Before giving the programme it is necessary to make a few remarks. The algorithmic language [7], [8] in which this programme is written, does not immediately cater for arithmetic operations upon complex numbers. It is therefore necessary to construct an arsenal of procedures for doing this and to derive a convention which governs their use. We therefore stipulate that all complex numbers are to be represented by arrays containing at least two members. There is an integer i which is defined globally throughout the block in which the complex arithmetic takes place, and all complex numbers (e.g. z, c_s) may be recognized throughout the programme by virtue of the fact that they contain the index i (e.g. $z[i]$, $c[s, i]$). i takes two values, zero-corresponding to the real part (e.g. Re $(z) \equiv z[0]$, Re $(c_s) \equiv c[s, 0]$) and unity-corresponding to the imaginary part. The integer i may not, therefore (except in circumstances which are difficult to envisage) be used for any other purpose.

Referring to the ALGOL programme, there is a procedure *eq* (*one*, *other*) which carries out an instruction analogous to the operation *one* : = *other* for real numbers. Similarly *seqeq* (*third*, *second*, *first*) carries out an assignment similar to *third* : = *second* : = *first*. The procedure *cm* (*res*, *one*, *other*) carries out an assignment similar to *res* : = *one* × *other*, and *cd* (*res*, *one*, *other*) one similar to *res* : = *one*/*other*. It is however necessary to ensure that numbers which occur in the arithmetic as real numbers are treated as such (i.e. with their imaginary parts put equal to zero), and for this purpose the procedure *real* (*variable*) is used. The function of further procedures, such as *mod* (*it*), is obvious. The input to all these procedures can either take the form of a complex number or a linear combination of complex numbers in which the coefficients are real. Further details are to be found in [9].

The coefficients $b_{r,s}$ ($r = 1, 2, ..., r$ max ; $s = 0, 1, ..., 2r$ max $- r + 1$) are members of a triangular array, and such arrays are not defined in ALGOL. This may be overcome by constructing a mapping function (the integer procedure $mf(m_1, m_2)$) which maps the $b_{r,s}$ onto a linear array.

Having evaluated the terms t_r defined by (63) the series $\sum_{r=0} t_r$ is summed either as far as the given upper bound r max, or until

$$|t_{r+1}| > |t_r| \quad \text{and} \quad |t_{r+2}| > |t_{r+1}|$$

when it is assumed that the series for the remainder $R_n(z)$ has itself an asymptotic character and has begun to diverge.

As the terms t_r in (61) are produced the ε-algorithm is applied immediately. It will be recalled that only the quantities $\varepsilon_s^{(m)}$ with even suffix are of interest in the present application. As they are produced they are mapped onto a display vector (di [i, ms]), and afterwards picked out and printed in a table which corresponds to the ε-array (Table I) with the columns of odd order missing.

With these remarks in mind and the comments to guide him the following ALGOL programme may be read without difficulty.

It reads, as data, ρ and θ/π, and immediately prints out ρ, θ/π, and n. It then computes and prints out the terms $u_0, u_1, ..., u_{n-1}$ of the asymptotic series (15) and their sum. It then computes and prints out the values of the coefficients γ_{2r} and of the terms t_r ; if the condition

$$|t_{r+1}| > |t_r| \quad \text{and} \quad |t_{r+2}| > |t_{r+1}|$$

is not obeyed the term is added to the numerical sum for the remainder $R_n(z)$. Application of the ε-algorithm to the series (61) takes place at the same time. After $r = r$ max the numerical sum for the remainder

$$\left(\sum_{r=0} t_r\right)$$

and the complete sum

$$\left(\sum_{r=0}^{n-1} u_r + \sum_{r=0} t_r\right)$$

are printed out in turn. Next the (triangular) even column ε arrays resulting from the application of the ε-algorithm to the series (61) are printed (real and imaginary parts separately), and two further triangular arrays which correspond to the addition of the transformed remainder term to the partial sum of the asymptotic series are printed.

In this way one is able to observe the numerical behaviour of the asymptotic series in (14) and of the series (54) for the remainder term, and one is able to observe the effect of applying the ε-algorithm to the latter.

Finally it is remarked that in order to make the programme as easily comprehensible as possible there is at certain points a slight wastage of storage (for example, in the singular case, c_0 is stored and used as a real variable, although its value is known to be unity).

```
comment Numerical study of a result of Stieltjes ;
begin integer max, twormax, fourrmax, col ;
    real rho, multiple of pi ; boolean non singular ;
    rmax : = READ ; rho : = READ ; multiple of pi : = READ ;
    col : = READ ; twormax : = 2 × rmax ; fourrmax : = 4 × rmax ;
    non singular : = (multiple of pi ≠ 1.0) ;
    begin real pi, theta, eta, gammafnquot, au, ratio, power of n ;
        integer i, n, s, m, r, r 1, sanfang, rs, twor ;
        boolean still converging, display Laplace series alone ;
        array aux 0, aux 1, aux 2, z, beta, sum, u,
        factor, remainderterm, gamma [0 : 1],
        b[(if non singular then — 1 — twormax else — 2 — fourrmax) :
        (rmax — 1) × (twormax + 1)],
        di[0 : 1,1 : ((rmax + 1) × (rmax + 5)) ÷ 4, 0 : 1],
        c[0 : 1, 0 : (if non singular then twormax else twormax + 1)],
        l[0 : rmax + 1, 0 : 1],
        Laplace term, termr [— 2 : 1, 0 : 1], modtermr [— 2 : 0] ;
        procedure eq (one, other) ; real one, other ;
        for i : = 0, 1 do one : = other ;
        procedure seqeq (third, second, first) ;
        real third, second, first ;
        for i : = 0,1 do third : = second : = first ;
        procedure cm (res, one, other) ; real res, one, other ;
        begin real Reone, Imone, Reother, Imother ;
            i : = 0 ; Reone : = one ; Reother : = other ;
            i : = 1 ; Imone : = one ; Imother : = other ;
            res : = Reone × Imother + Imone × Reother ;
            i : = 0 ; res : = Reone × Reother — Imone × Imother
        end cm ;
        procedure cd (res, one, other) ; real res, one, other ;
        begin real Reone, Imone, Reother, Imother, denom ;
            i : = 0 ; Reone : = one ; Reother : = other ;
            i : = 1 ; Imone : = one ; Imother : = other ;
            denom : = Reother × Reother + Imother × Imother ;
            res : = (Imone × Reother — Reone × Imother) / denom
            i : = 0 ;
```

```
      res : = (Reone × Reother + Imone × Imother) /denom
end cd ;
real procedure real (variable) ; real variable ;
real : = (if i = 0 then variable else 0.0) ;
real procedure imaginary (variable) ; real variable ;
imaginary : = (if i = 0 then 0.0 else variable) ;
real procedure mod (it) ; real it ;
begin real Reit, Imit ;
      i : = 0 ; Reit : = it ; i : = 1 ; Imit : = it ;
      mod : = sqrt (Reit × Reit + Imit × Imit)
end mod ;
procedure polar form (res, r, theta) ; real res, r, theta ;
begin real r 1, theta 1 ;
      r 1 : = r ; theta 1 : = theta ;
      i : = 0 ; res : = r 1 × cos (theta 1) ;
      i : = 1 ; res : = r 1 × sin (theta 1)
end polar form ;
procedure comprecip (res, it) ; real (res, it) ;
begin real Reit, Imit, denom ;
      i : = 0 ; Reit : = it ; i : = 1 ; Imit : = it ;
      denom : = Reit × Reit + Imit × Imit ;
      res : = — Imit /denom ; i : = 0 ; res : = Reit /denom
end comprecip ;
procedure compexp (res, it) ; real res, it ;
begin real aux 1, aux 2 ;
      i : = 0 ; aux 1 : = exp (it) ; i : = 1 ; aux 2 : = it ;
      res : = aux 1 × sin (aux 2) ;
      i : = 0 ; res : = aux 1 × cos (aux 2)
end compexp ;
procedure compprint (it) ; real it ;
for i : = 0, 1 do PRINT (it) ;
procedure druck (it) ; real it ;
begin compprint (it) ; PRINT (mod (it))
end druck ;
boolean procedure even (integer) ; integer integer ;
even : = (integer = (integer ÷ 2) × 2) ;
integer procedure mf (m 1, m 2) ; value m 1 ; integer m 1, m 2 ;
mf : = ((m 1 — 1) × (fourrmax — m 1)) ÷ 2 + m 2 ;
procedure NT ;
comment This procedure uses the (non ALGOL) real procedures
NLCR and TAB : the first gives a Newline Carriage Return
and the second shifts the typewriter carriage to the next tabu-
lator stop ;
```

```
begin NLCR ; NLCR ; TAB ; TAB ; TAB
end NT ;
procedure cma (res, one, other, it) ; real res, one, other, it ;
begin array aux 3 [0 : 1] ;
      cm (aux 3 [i], one, other) ; eq (res, aux 3 [i] + it)
end cma ;
procedure sum and display remainder term ;
begin NLCR ; druck (Laplace term [— 2, i]) ;
      druck (termr [— 2, i]) ;
      eq (remainder term [i], remainder term [i] + termr [— 2, i]);
      for s : = — 2, — 1 do
      begin eq (Laplace term [s, i], Laplace term [s + 1, i]) ;
            eq (termr [s, i], termr [s + 1, i]) ;
            modtermr [s] : = modtermr [s + 1]
      end s
end sum and display remainder term ;
comment Introduction ;
pi : = 3.14159 26535 89793 ;
n : = entier (rho) ; eta : = rho — n ; NLCR ;
PRINT (rho) ; PRINT (multiple of pi) ; PRINT (n) ;
PRINT (eta) ; theta : = multiple of pi × pi ;
polar form (beta [i], 1.0, theta) ;
eq (z [i], rho × beta [i]) ;
comment Evaluation of terms and partial sum of asymptotic
        series ;
eq (sum [i], 0.0) ;
for s : = 0 step 1 until n — 1 do
begin NLCR ;
      cd (u[i], (if s = 0 then real (1.0) else — s × u[i]), z[i]) ;
      druck (u[i]) ; eq (sum[i], sum[i] + u[i])
end computing terms ;
NLCR ; NLCR ; druck (sum[i]) ;
comment Determination of b[s] ;
b[0] : = sqrt (2.0) ;
for s : = 1 step 1 until twormax do
begin au : = 0.0 ;
      for m : = 0 step 1 until s — 2 do
      au : = au + b[m + 1] × (m + 2) × b[s — m — 1] /2.0 ;
      b[s] : = (b[s — 1] — au) × b[0] /(s + 2)
end computing b[s] ;
comment Determination of b[— 1, s] ;
b[— 1 — twormax] : = 1.0 /b[0] ;
for s : = 1 step 1 until twormax do
begin au : = 0.0 ;
      for m : = 1 step 1 until s do
```

```
      au : = au + b[m] × b[s — m — twormax — 1] ;
      b[s — 1 — twormax] : = — au / b[0]
end computing b[— 1, s] ;
comment Determination of b[r, s] ;
for r : = 2 step 1 until twormax — 1 do
for s : = 0 step 1 until twormax — r — 1 do
begin au : = 0.0 ;
      for m : = 0 step 1 until s do
      au : = au + b[m] × b[mf(r — 1, s — m)] ;
      b[mf(r, s)] : = au
end computing b[r, s] ;
if¬ non singular then
begin comment Determination of b[— 2, s] ;
      for s : = 0 step 1 until twormax do
      begin au : = 0.0 ;
            for m : = 0 step 1 until s + 1 do
            au : = au + b[— 1 — twormax + m]
                                    × b[— twormax + s — m]
            b[— 2 — fourrmax + s] : = au
      end s
end computing b[— 2, s] ;
comment Determination of c[s] ;
if non singular then
   comprecip (c[i, 0], beta[i] + real (1.0))
else eq (c[i, 0], real (1.0)) ;
au : = 1.0 ;
for s : = 0 step 1 until twormax — (if non singular then 1 else 0) do
begin if non singular then
         begin cd (c[i, s + 1],
               real (au × (s + 1 — eta)) — c[i, s],
               beta[i] + real (1.0)) ;
               if s ≠ twormax — 1 then au : = — eta × au /(s + 2
         end non singular case
     else
         begin ratio : = — eta /(s + 1) ;
               eq (c[i, s + 1], real (au × (1.0 + ratio))) ;
               if s ≠ twormax then au : = au × ratio
         end singular case
end computing c[s] ;
comment Computation of remainder term ;
compexp (aux 1[i], — real (rho) — imaginary (n × theta)) ;
eq (factor[i],
(if even(n) then 1.0 else — 1.0) × 2.0 × sqrt(pi /n) × aux 1[i])
gammafnquot : = power of n : = 1.0 ;
```

```
still converging : = true ;
eq (l[0, i], 0.0) ; eq (remainder term[i], 0.0) ;

for r : = 0 step 1 until rmax do
begin twor : = 2 × r ;
      eq (gamma[i]),
         if non singular then c[i, 0] × b[— 1 — twormax + twor]
         else c[i, 0] × b[— 2 — fourrmax + twor]
                    + c[i, 1] × b[— 1 — twormax + twor])) ;
      for s : = 2 step 1 until twor do
      eq (gamma[i], gamma[i] + b[mf(s — 1, twor — s)]
                    × c[i, (if non singular then s else s + 1)]) ;
      r 1 : = (if r ⩾ 2 then 0 else r — 2) ;
      eq (Laplace term[r 1, i], gamma[i]
                                   × gammafnquot/power of n) ;
      cm (termr[r 1, i], factor[i], Laplace term[r 1, i]) ;
      modtermr[r 1] : = mod (termr[r 1, i]) ;
      if r ⩾ 2 ∧ still converging then
      begin if (modtermr[— 2] > modtermr [— 1])
                             ∧ (modtermr[— 1] > modtermr[0])
         then sum and display remainder term
         else still converging : = false
      end adding in remainder series term (or not) ;
      gammafnquot : = gammafnquot × (r + 0.5) ;
      power of n : = n × power of n ;

      comment Application of epsilon algorithm to series for
         remainder term ;
      eq (aux 1[i], Laplace term[r 1, i] + l[0, i]) ;
      for s : = 0 step 1 until r do
      begin comprecip (aux 0[i], (if s = 0 then Laplace term [r 1, i]
         else aux 1[i] — l[s, i])) ;
         if s ≠ 0 then
         begin eq (aux 0[i], aux 0[i] + l[s — 1, i]) ;
               eq (l[s — 1, i], aux 2[i])
         end s non zero ;
         eq (aux 2[i], aux 1[i]) ; eq (aux 1[i], aux 0[i]) ;
         if even(s) then
         begin rs : = (s × (twormax + 2 — s)) ÷ 4 + r + 1 ;
               eq (di[0, rs, i], aux 2[i]) ;
               cma (di[1, rs, i], factor[i], aux 2[i], sum[i])
         end even s ;
         if s = r ∧ ¬ even(r) then
         begin rs : = ((r + 1) × (twormax — r + 5)) ÷ 4 ;
               eq (di[0, rs, i], aux 1[i]) ;
               cma (di[1, rs, i], factor[i], aux 1[i], sum[i])
         end s = r
```

```
            end s ;
            eq (l[r, i], aux 2[i]) ; eq (l[r + 1, i], aux 1[i])
        end r ;
        if still converging ∧ modtermr[— 1] ⩽ modtermr[— 2] then
        begin sum and display remainder term ;
              sum and display remainder term ;
        end adding in last two terms ;
        comment Print remainder term and complete sum ;
        for r 1 : = 0, 1 do
        begin NT ; druck (remainder term[i] + r 1 × sum[i])
        end printing remainder term and modified result ;
        comment Display application of epsilon algorithm to
        Laplace series and corresponding complete sums ;
        display Laplace series alone : = true ;
        Triangular display : for i : = 0, 1 do
        begin for sanfang : = 0 step 2 × col until rmax + 1 do
           begin NLCR ;
                 for r : = 1 step 1 until rmax + 1 — sanfang ÷ 2 do
                 begin NLCR ;
                       for s : = sanfang step 2 until
                                sanfang + 2 × (col — 1) do
                       if (s ÷ 2 ⩽ r) ∧ (r ⩽ rmax + 1 — (s ÷ 2)) then
                       begin rs : = (s × (twormax + 4 — s)) ÷ 4 + r ;
                             PRINT (di[ if display Laplace series alone
                                        then 0 else 1, rs, i])
                       end s
                 end r
           end sanfang
        end real and imaginary parts ;
        if display Laplace series alone then
        begin display Laplace series alone : = false ;
              goto Triangular display
        end returning to Triangular display
    end inner block
end whole programme
```

IV. — NUMERICAL RESULTS (1)

A. — The Non-singular Case

Some numerical results which have been produced by means of the preceding ALGOL programme are summarised in the following tables

(1) The numerical results of this paper were produced on the X1 computer at Amsterdam using an ALGOL translator constructed by J. A. Zonneveld and E. W. Dijkstra.

which relate to the choice of argument $z = 5{,}5i$ (i.e. $\rho = 5{,}5$, $\theta = 0{,}5$, $n = 5$, $\eta = 0{,}5$).

TABLE I gives the terms (real part, imaginary part and modulus) and the partial sum of the asymptotic series (15).

TABLE I

r	$Re(u_r)$	$I_m(u_r)$	$\mid u_r \mid$
0	0,0	— 0,1818 1818	0,1818 1818
1	+ 0,0330 5785	0,0	0,0330 5785
2	0,0	+ 0,0120 2104	0,0120 2104
3	— 0,0065 5693	0,0	0,0065 5693
4	0,0	— 0,0047 6868	0,0047 6868
$\sum_{r=0}^{4} u_r$	+ 0,0265 0092	— 0,1745 6582	0,1765 6592

TABLE II gives the value of the coefficients γ_{2r} and of the terms

$$(-1)^n e^{-in\theta} e^{-\rho} 2\sqrt{\frac{\pi}{n}}\gamma_{2r}\left(\frac{1}{2}\right)\left(\frac{3}{2}\right)\cdots\left(r-\frac{1}{2}\right)n^{-r} \quad (\equiv t_r)$$

($r = 0, 1, \ldots$), the numerical sum of these terms, and the modified sum $\sum_{r=0}^{n-1} u_r + R_n(z)$.

TABLE II

r	$\mathrm{Re}(\gamma_{2r})$	$\mathrm{Im}(\gamma_{2r})$	$\mid\gamma_{2r}\mid$	$\mathrm{Re}(t_r)$	$\mathrm{Im}(t_r)$	$\mid t_r \mid$
0	+ 0,3535 5339	— 0,3535 5339	0,5	+ 0,0022 9063	+ 0,0022 9063	0,0032 3944
1	— 0,0559 7929	+ 0,0206 2395	0,0596 5759	— 0,0001 3362	— 0,0003 6268	0,0003 8651
2	+ 0,0162 1681	+ 0,0035 2326	0,0165 9512	— 0,0000 2283	+ 0,0001 0507	0,0001 0752
3	— 0,0054 8122	— 0,0050 9715	0,0074 8496	+ 0,0000 3302	— 0,0000 3551	0,0000 4849
4	+ 0,0013 8464	+ 0,0043 7594	0,0045 8978	— 0,0000 2835	+ 0,0000 0897	0,0000 2974
5	+ 0,0007 5148	— 0,0035 0770	0,0035 8729	+ 0,0000 2273	+ 0,0000 0487	0,0000 2324
$\sum_{r=0}^{5} t_r = R_5(z)$				+ 0,0021 6158	+ 0,0020 1134	+ 0,0029 5262
$\sum_{r=0}^{4} u_r + R_5(z)$				+ 0,0286 6251	— 0,1725 5448	0,1749 1880

TABLES III and IV give the real and imaginary parts respectively of those modified sums wich are to be derived by applying the ε-algorithm to the series expansion of the remainder term, and using the members of the resulting even column ε-array as approximations to the remainder term.

TABLE III

m/s	0	2	4	6	8	10
1	+ 0,02879 1554	+ 0,02865 6507				
2	02865 7934	02864 9860	+ 0,02865 3523			
3	02863 5107	02865 4137	02865 2108	+ 0,02865 2633		
4	02866 8131	02865 1444	02865 2704	02865 2501	+ 0,02865 2552	
5	02863 9780	02865 3266	02865 2460	02865 2559	02865 2537	+ 0,02865 2541
6	02866 2505	02865 2067	02865 2584	02865 2532	+ 0,02865 2541	
7	02864 5958	02865 2813	02865 2516	+ 0,02865 2541		
8	02865 3899	02865 2466	+ 0,02865 2545			
9	02866 0268	+ 0,02865 2349				
10	+ 0,02862 8662					

TABLE IV

m/s	0	2	4	6	8	10
1	— 0,17227 5189	— 0,17259 6330				
2	17263 7872	17255 1797	— 0,17256 0646			
3	17253 2806	17256 1545	17255 9388	— 0,17255 9644		
4	17256 8318	17255 9180	17255 9667	17255 9614	— 0,17255 9605	
5	17255 9347	17255 9553	17255 9616	17255 9600	17255 9607	— 0,17255 9603
6	17255 4478	17255 9866	17255 9573	17255 9608	— 0,17255 9602	
7	17256 9279	17255 9212	17255 9640	— 0,17255 9599		
8	17254 5504	17256 0096	— 0,17255 9567			
9	17257 7531	— 0,17255 9040				
10	— 0,17254 1127					

The correct value of $\int_0^\infty \frac{e^{-5.5it}}{1+t} dt$ computed independently (by means of an ascending power series and by means of a continued fraction) is + 0,02865 2539 — i 0,17255 9604.

Numerical experiments indicate that the rate of convergence of the series for the remainder term decreases as the argument of z increases from 0 to π. This is illustrated in Table V which gives the values of γ_0 and γ_8 (with $\eta = 0{,}5$) when arg $(z) = 0, \pi/4, \pi/2$ and $3\pi/4$.

TABLE V

Arg (z)	$\lvert\gamma_0\rvert$	$\lvert\gamma_8\rvert$
0	0,35355	0,09148
$\pi/4$	0,38268	0,55350
$\pi/2$	0,5	5,31446
$3\pi/4$	0,92388	151,28530

B. — The Singular Case

When arg $(z) = \pi$, the formulae involved are, it will be recalled, slightly more complicated. We illustrate their use by giving numerical results relating to the case $z = -4{,}0$, i.e. $n = 4$, $h = 0{,}0$. The terms and partial sum of the asymptotic series are given in Table VI.

TABLE VI

r	u_r
0	— 0,25
1	0625
2	03125
3	— 0,0234375
$\sum_{r=0}^{3} u_r$	— 0,3671875

Table VII gives those modified sums which are to be derived by applying the ε-algorithm to the series expansion of the remainder term, and using the members of the resulting even column ε-array as approximations to the remainder term.

TABLE VII

m/s	0	2	4
1	— 0,35953 5750	— 0,35954 6363	
2	35954 6378	35955 9796	— 0,35955 2079
3	35955 2308	35955 2084	— 0,35955 2003
4	35955 2075	— 0,35955 1960	
5	— 0,35955 1998		

The correct value of the (Cauchy) integral in question is

$$-0{,}35955\ 2008$$

It can thus be seen the asymptotic series alone yields an absolute error of $0{,}367 - 0{,}360 = 0{,}007$, that use of the series expansion of the remainder term yields an absolute error of

$$0{,}35955\ 2008 - 0{,}3955\ 1998 = 0{,}00000\ 0010,$$

and that application of the ε-algorithm to the remainder term expansion yields an absolute error of 0,00000 0005 (these last two figures relate, of course, to the use of five terms of the remainder term expansion).

Thus, in the singular case, application of the ε-algorithm to the remainder term expansion does not seem to be so favorable.

REFERENCES

(1) Erdélyi (A.), *Asymptotic Expansions*, Dover (1956), ch. II.

(2) Stieltjes (T.), *Recherches sur quelques séries semi-convergentes*, Ann. de l'Ec. Normale, 3e série, tome III (1886), p. 26.

(3) Wynn (P.), *On a Device for Computing the $e_m(S_n)$ Transformation*, M.T.A.C., vol. 10 (1956), p. 91.

(4) Wynn (P.), *The Rational Approximation of Functions which are Formally Defined by a Power Series Expansion*, Math. of Comp., vol. 14 (1960), p. 147.

(5) Wynn (P.), *On Repeated Application of the ε-Algorithm*, Chiffres, vol. 1 (1961), p. 19.

(6) Wynn (P.), *A Comparison between the Numerical Performances of the Euler Transformation and the Epsilon Algorithm*, Chiffres, vol. 1 (1961), p. 23.

(7) Backus (J. W.) et al., *Report on the Algorithmic Language Algol 60*, Num. Math., vol. 2 (1960), p. 180.

(8) Genuys (M. F.), Poyen (J.) et Vauquois (M. B.), *Rapport sur le langage algorithmique Algol 60*, Chiffres, vol. 3 (1960), p. 1.

(9) Wynn (P.), *An Arsenal of Algol Procedures for Complex Arithmetic*, Nord. Tids. Inf. Behandling.