STICHTING
MATHEMATISCH CENTRUM
2e BOERHAAVESTRAAT 49
AMSTERDAM

REKENAFDELING

Converging Factors for the Weber Parabolic Cylinder Functions of Complex Argument

by

P. Wynn

The Mathematical Centre at Amsterdam, founded the 11th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications, and is sponsored by the Netherlands Government through the Netherlands Organization for Pure Research (Z.W.O.) and the Central National Council for Applied Scientific Research in the Netherlands (T.N.O.),by the Municipality of Amsterdam and by several industries.

Introduction

Numerical values of certain elementary functions (e.g. exp(x), sin(x),cos(x),ln(x))are made available to digital computer users by means of programmed subroutines. The tendency will be to extend this list of "elementary" functions, and considerable interest therefore attaches to general and efficient methods for computing numerical values to great accuracy of the higher functions of Mathematical Physics.One such method is the application of the converging factor.

The Converging Factor

The converging factor is an important numerical device for hastening the convergence of slowly convergent series and increasing the accuracy obtainable by use of an asymptotic series. If the series is

$$S \sim u_0 + u_1 + u_2 + \dots \qquad (1)$$

and the partial remainder R_n is

$$R_n \sim u_n + u_{n+1} + u_{n+2} + \dots \qquad (2)$$

the converging factor C_n is defined by

$$R_n = u_n C_n \qquad (3)$$

The converging factor is most efficiently used, in the case of most applications to asymptotic series, with that value of n which corresponds to the term of smallest modulus in the series (1).

Miller [1] has given a method for developing the converging factor C_n either as series of the form

$$C_n \sim \sum_{r=0}^{\infty} \beta_r z^{-r} \qquad (4)$$

or as a series of the form

$$C_n \sim \sum_{r=0}^{\infty} \delta_r n^{-r} \tag{5}$$

for the cases in which either the function S satisfies a linear differential equation in z or the terms u_r satisfy a linear difference equation in r. He illustrated his method by obtaining converging factors for asymptotic series associated with the Weber parabolic cylinder functions.

In the paper referred to, real values only of the argument are considered. Here the computations are extended into the complex domain. Secondly a convenient recursive technique for obtaining the coefficients in the series for the converging factor is described.

Weber functions

The series which is to be studied is

$$S_1(a;z) \sim e^{-z^2/4} z^{-a-\frac{1}{2}} \left\{ 1 - \frac{(a+\frac{1}{2})(a+\frac{3}{2})}{2.z^2} + \frac{(a+\frac{1}{2})(a+\frac{3}{2})(a+\frac{5}{2})(a+\frac{7}{2})}{2.4.z^4} \cdots \right\} \tag{6}$$

$$\sim u_0 - u_1 + u_2 - \cdots . \tag{7}$$

It formally satisfies the differential equation

$$\frac{d^2y}{dz^2} - \left(a + \frac{1}{4}z^2\right)y = 0 \tag{8}$$

Two linearly independent situations of equation (8) are $S_1(a;z)$ and

$$S_2(a;z) = S_1(-a;iz). \tag{9}$$

The terms u_r of the series (6) satisfy the recursion

$$2rz^2u_r = (a+2r-\tfrac{3}{2})(a+2r-\tfrac{1}{2})u_{r-1} . \tag{10}$$

We wish to determine that value n of r for which $|u_n|$ is a minimum. From (10) this is seen to occur when

$$2nx^2 \doteq (a+2n-\tfrac{3}{2})(a+2n-\tfrac{1}{2}) \qquad (11)$$

where

$$z = xe^{i\theta} \qquad (12)$$

In order to derive an easily usable approximation we ignore the term

$$\mu = (a-\tfrac{1}{2})(a-\tfrac{3}{2}) \qquad (13)$$

independent of n in (11), and obtain

$$x^2 \doteq 2n+\lambda \qquad (14)$$

where

$$\lambda = 2(a-1) \qquad (15)$$

or

$$2n = x^2-\lambda-k \qquad (16)$$

where k is real and may always be chosen so that

$$-1 \leqslant k \leqslant 1. \qquad (17)$$

The integer n having been determined, we define the remainder term R_n and converging factor Γ_n by

$$S_1(a;z) = \sum_{r=0}^{n-1}(-1)^r u_r + R_n \qquad (18)$$

$$R_n = (-1)^n u_n \Gamma_n \qquad (19)$$

We shall obtain a series development of the form

$$\Gamma_n \sim \sum_{r=0}^{\infty} \frac{\beta_r(k)}{2^{r+1} x^{2r}} \tag{20}$$

when $\arg(z) \neq \pi/2$, first using the fact that Γ_n satisfies a differential equation in z and secondly the fact that Γ_n satisfies a recursion in n.

Differential Equation

The converging factor satisfies the differential equation

$$z^2 \frac{d^2\Gamma_n}{dz^2} - z\left(z^2+2a+4n+1\right)\frac{d\Gamma_n}{dz} + \left(a+2n+\tfrac{1}{2}\right)\left(a+2n+\tfrac{3}{2}\right)\Gamma_n + 2nz^2\left(\Gamma_n - 1\right) = 0. \tag{21}$$

This may quite crudely be verified by substituting the series

$$\Gamma_n \sim 1 - \frac{(a+2n+\frac{1}{2})(a+2n+\frac{3}{2})}{2(n+1)z^2} + \frac{(a+2n+\frac{1}{2})(a+2n+\frac{3}{2})(a+2n+\frac{5}{2})(a+2n+\frac{7}{2})}{4(n+1)(n+2)z^4} - \tag{22}$$

in (21). A constructive derivation, based on an idea which is clearly capable of general application to the construction of converging factors, has been given by Miller. He writes

$$u_n = \text{a constant} \times e^{-\frac{1}{4}z^2} z^{-a-2n-\frac{1}{2}} \tag{23}$$

so

$$\frac{\frac{d}{dz}u_n}{u_n} = \left(-\frac{1}{2}z - \frac{a+2n+\frac{1}{2}}{z}\right) \tag{24}$$

and further

$$\frac{\frac{d^2u_n}{dz^2}}{u_n} = \left(\frac{\frac{du_n}{dz}}{u}\right)^2 - \frac{1}{2} + \frac{(a+2n+\frac{1}{2})}{z^2} \tag{25}$$

$$= \frac{1}{4}z^2 + a + 2n + \frac{(a+2n+\frac{1}{2})(a+2n+\frac{3}{2})}{z^2}$$

but

$$\frac{d^2R_n}{dz^2} - (a+\tfrac{1}{4}z^2)R_n = (-1)^n 2n u_n \tag{26}$$

whence

$$\frac{d^2\Gamma_n}{dz^2}u_n + 2\left(\frac{d\Gamma_n}{dz}\right)\left(\frac{du_n}{dz}\right) + \Gamma_n\left(\frac{d^2u_n}{dz^2}\right) - (a+\tfrac{1}{4}z^2)\Gamma_n u_n = 2n u_n \tag{27}$$

Removing u_n and its derivatives from (27) by way of (24) and (25), we arrive at (21).

In this section we shall suppose that a and n are fixed, so that z and k vary together. We have from equations (12) and (16)

$$dz = e^{i\theta}dx, \quad 2x\,dx = dk. \tag{28}$$

By means of equations (12),(16) and (28) we may remove n from equation (21) and transform the result into a differential equation with k as the independent variable. We obtain, after some rearrangement

$$\begin{aligned} &x^4\left\{4\Gamma_n'' - 2(\phi+2)\Gamma_n' + (\phi+1)\Gamma_n - \phi\right\} \\ &+ x^2\left\{2(2k+\lambda-2)\Gamma_n' + (4-\lambda-2k-\phi(\lambda+k))\Gamma_n + \phi(\lambda+k)\right\} \\ &+ \left\{k^2 + (\lambda-4)k + \mu - 2\lambda + 4\right\}\Gamma_n = 0, \end{aligned} \tag{29}$$

where

$$\phi = e^{2i\theta} \tag{30}$$

and dashes denote differentiation with respect to k.

From (20) and (28) we have successively

$$\Gamma_n' \sim \frac{\beta_0'}{2} + \frac{\beta_1'}{2^2 x^2} + \frac{\beta_2' - 2\beta_1}{2^3 x^4} + \dots + \frac{\beta_r' - 2(r-1)\beta_{r-1}}{2^{r+1} x^{2r}} \tag{31}$$

and

$$\Gamma_n'' \sim \frac{\beta_0''}{2} + \frac{\beta_1''}{2^2 x^2} + \frac{\beta_2'' - 4\beta_1'}{2^3 x^4} + \dots + \frac{\beta_r'' - 4(r-1)\beta_{r-1}' + 4(r-1)(r-2)\beta_{r-2}}{2^{r+1} x^{2r}} \tag{32}$$

Substituting the series (20) (31) and (32) in (29) and equating to zero the coefficients of the successive powers of x we obtain a recursion system between the functions $\beta_r(k)$ We have, in succession,

$$x^4: \quad 4\beta_0'' - 2(\phi+2)\beta_0' + (\phi+1)\beta_0 = 2\phi \tag{33}$$

$$x^2: \quad 4\beta_1'' - 2(\phi+2)\beta_1' + (\phi+1)\beta_1 = -4(2k+\lambda-2)\beta_0' - 2\{4-\lambda-2k-\phi(\lambda+k)\}\beta_0 - 4(\lambda+k)\phi \tag{34}$$

and

$$\begin{aligned} x^{-2r+4}: \quad 4\beta_r'' - 2(\phi+2)\beta_r' + (\phi+1)\beta_r = {} & 4\{4r-\lambda-2k-2\}\beta_{r-1}' \\ & +2\{k(\phi+2)+\lambda(\phi+1)-2(r-1)\phi-4r\}\beta_{r-1} \\ & -4\{k^2+k(\lambda-4r+4)k+\mu-2\lambda(r-1)+4(r-1)^2\}\beta_{r-2} \\ & (r=2,3,\dots) \end{aligned} \tag{35}$$

Inspection of equations (33), (34) and (35) reveals that they are formally satisfied by polynomials of the form

$$\beta_r(k) = \sum_{s=0}^{r} p_{r,s} k^s \qquad (r = 0, 1, \ldots), \tag{36}$$

Equation (33) indicates that,

$$\beta_0(k) = 2\phi/(\phi+1), \tag{37}$$

equation (34) yields

$$\beta_1(k) = \frac{4\phi}{(\phi+1)^2} k - \frac{8\phi^2}{(\phi+1)^3}, \tag{38}$$

and from equation (35) we may derive

$$\beta_2(k) = \frac{8\phi k^2}{(\phi+1)^3} - \frac{48\phi^2 k}{(\phi+1)^4} - \left\{ \frac{8\mu(\phi+1)^3 + 32\phi^2(1-2\phi)}{(\phi+1)^5} \right\} \tag{39}$$

and

$$\beta_3(k) = \frac{16\phi k^3}{(\phi+1)^4} - \frac{192\phi^2 k^2}{(\phi+1)^5} + \left\{ \frac{-16\mu\phi(\phi+3)}{(\phi+1)^3} + \frac{64\phi^2(11\phi-4)}{(\phi+1)^6} \right\}$$
$$- \frac{16\phi\mu\lambda}{(\phi+1)^2} + \frac{32\phi^2(\phi+4)}{(\phi+1)^4}\mu - \frac{128\phi^2(6\phi^2-8\phi+1)}{(\phi+1)^7}. \tag{40}$$

We wish, however, to devise some recursive process for determing the coefficients $p_{r,s}$. In principle this can be done since, knowing $\beta_{r-1}(k)$ and $\beta_{r-2}(k)$, $\beta_r(k)$ may be derived from equation (35). Let us examine how this may be accomplished in detail. Substituting polynomial expressions of the form (36) in (35) we obtain, after some rearrangement

$$4\sum_{s=0}^{r-2}(s+2)(s+1)p_{r,s+2}k^s-2(\phi+2)\sum_{s=0}^{r-1}(s+1)p_{r,s+1}k^s+(\phi+1)\sum_{s=0}^{r}p_{r,s}k^s$$
$$=4\{4r-\lambda-2\}\sum_{s=0}^{r-2}(s+1)p_{r-1,s+1}k^s-8\sum_{s=1}^{r-1}s\,p_{r-1,s}k^s$$
$$+2(\phi+2)\sum_{s=1}^{r}p_{r-1,s-1}k^s+2\{\lambda(\phi+1)-2(r-1)\phi-4r\}\sum_{s=0}^{r-1}p_{r-1,s}k^s$$
$$-4\sum_{s=2}^{r}p_{r-2,s-2}k^s-4(\lambda-4r+4)\sum_{s=1}^{r-1}p_{r-2,s-1}k^s$$
$$-4\{\mu-2\lambda(r-1)+4(r-1)^2\}\sum_{s=0}^{r-2}p_{r-2,s}k^s. \qquad (41)$$

Equating to zero the coefficients of k in the order s=r,r-1,...,0, we obtain

$$p_{r,r}=\{2(\phi+2)p_{r-1,r-1}-4p_{r-2,r-2}\}/(\phi+1), \qquad (42)$$

$$p_{r,r-1}=\{2(\phi+2)r\,p_{r,r}-8(r-1)p_{r-1,r-1}$$
$$+2(\phi+2)p_{r-1,r-2}+2\{\lambda(\phi+1)-2(r-1)\phi-4r\}p_{r-1,r-1}$$
$$-4p_{r-2,r-3}-4(\lambda-4r+4)p_{r-2,r-2}\}/(\phi+1) \qquad (43)$$

$$p_{r,s}=\Big[2(s+1)(\phi+2)p_{r,s+1}-4(s+1)(s+2)p_{r,s+2}$$
$$+4\{4r-\lambda-2\}(s+1)p_{r-1,s+1}-8s\,p_{r-1,s}$$
$$+2(\phi+2)p_{r-1,s-1}+2\{\lambda(\phi+1)-2(r-1)\phi-4r\}p_{r-1,s}$$
$$-4p_{r-2,s-2}-4(\lambda-4r+4)p_{r-2,s-1}$$
$$-4\{\mu-2\lambda(r-1)+4(r-1)^2\}p_{r-2,s}\Big]/(\phi+1)$$
$$(s=r-2,r-3,\ldots,2) \qquad (44)$$

$$p_{r,1}=\Big[4(\phi+2)p_{r,2}-24p_{r,3}+8(4r-\lambda-2)p_{r-1,2}$$
$$-8p_{r-1,1}+2(\phi+2)p_{r-1,0}+2\{\lambda(\phi+1)-2(r-1)\phi-4r\}p_{r-1,1}$$
$$-4(\lambda-4r+4)p_{r-2,0}-4\{\mu-2\lambda(r-1)+4(r-1)^2\}p_{r-2,2}\Big]/(\phi+1) \qquad (45)$$

$$p_{r,0}=\Big[2(\phi+2)p_{r,1}-8p_{r,2}+4(4r-\lambda-2)p_{r-1,1}$$
$$+2\{\lambda(\phi+1)-2(r-1)\phi-4r\}p_{r-1,0}$$
$$-4\{\mu-2\lambda(r-1)+4(r-1)^2\}p_{r-2,0}\Big]/(\phi+1). \qquad (46)$$

Thus, if equations (37),(38) and (42)-46) are used in that order the coefficients $p_{r,s}$ (r=0,1,...;s=0,1,...,r) may always be expressed in terms of quantities which have previously been derived.

It will be observed, however, that equations (42)-(46) differ from one another according as to whether certain powers of k do or do not exist in the various sums in (41). This fact may also be expressed by the use of conditional statements, and thus an expression for $p_{r,s}$ which is generally true for r $\geqslant$ 2 may be constructed. The special forms for $p_{0,0}$ and $p_{1,1}$, $p_{1,0}$ may also be incorporated in this expression, and thus we have

$$p_{r,s}=\Big[\ \text{if } s<r \text{ then } 2(\phi+2)(s+1)p_{r,s+1}$$
$$-\text{if } s<r-1 \text{ then } 4(s+2)(s+1)p_{r,s+2}$$
$$+\text{if } s<r-1 \text{ then } 4(4r-\lambda+2)(s+1)p_{r-1,s+1}$$
$$-\text{if } 0<s<r \text{ then } 8s\,p_{r-1,s}$$
$$+\text{if } s>0 \text{ then } 2(\phi+2)p_{r-1,s-1}$$
$$+\text{if } s<r \text{ then } 2\{\lambda(\phi+1)-2(r-1)\phi-4r\}p_{r-1,s}$$

$$
\begin{aligned}
&- \text{if } s>1 \text{ then } 4p_{r-2,s-2} \\
&- \text{if } 0<s<r \text{ then } 4(\lambda-4r+4)p_{r-2,s-1} \\
&- \text{if } s<r-1 \text{ then } 4\{\mu+2(r-1)\{2(r-1)-\lambda\}p_{r-2,s} \\
&+ \text{if } r=0 \text{ then } 2\phi \\
&- \text{if } r=1 \text{ then if } s=0 \text{ then } 4\lambda\phi \text{ and if } s=1 \text{ then } -4\phi\}/(\phi+1).
\end{aligned} \tag{47}
$$

This definition is uniformly valid for r=0,1,... and s=r,r-1,...,0. Its derivation does not, of course, represent an attempt at elegance for its own sake. It will be realised that there is considerable duplication in formulae (42)-(46), so that if we were to write down the formulae for $p_{r,s}$ in some algorithmic language for a digital computer based on formulae (42)-(46), we would in effect be wasting a large number of instructions in needless repetion. Use of formula (47) avoids this at the cost of a few conditional statements, which (in comparison with the complexity of the formulae used) is negligible

Difference Equations

In the notation of equation (18) we have

$$R_{n-1} - R_n = (-1)^{n-1} u_{n-1} \tag{48}$$

and since

$$R_n = (-1)^n u_n \Gamma_n \tag{49}$$

we have

$$u_{n-1}\Gamma_{n-1} + u_n\Gamma_n = u_{n-1} \tag{50}$$

or, using (10)

$$2nz^2\left(\Gamma_{n-1}-1\right)+\left(a+2n-3/2\right)\left(a+2n-1/2\right)\Gamma_n=0 \quad (51)$$

In this section we shall suppose that a and x are fixed, so that when n decreases to n-1,k becomes k+2; thus if

$$\Gamma_n \sim \frac{\beta_0(k)}{2}+\frac{\beta_1(k)}{2^2x^2}+\frac{\beta_2(k)}{2^3x^4}+\cdots \quad (52)$$

then

$$\Gamma_{n-1} \sim \frac{\beta_0(k+2)}{2}+\frac{\beta_1(k+2)}{2^2x^2}+\frac{\beta_2(k+2)}{2^3x^4}+\cdots \quad (53)$$

In equation (51) we write $x^2\phi$ for z^2, substitute for 2n in terms of x and k, and insert the series (52) and (53), finally obtaining

$$\phi x^2\left\{x^2-\lambda-k\right\}\left\{\frac{\beta_0(k+2)}{2}-1+\frac{\beta_1(k+2)}{2^2x^2}+\frac{\beta_2(k+2)}{2^3x^4}+\cdots\right\}$$
$$+\left\{x^4-x^2(\lambda+2k)+k^2+\lambda k+\mu\right\}\left\{\frac{\beta_0(k)}{2}+\frac{\beta_1(k)}{2^2x^2}+\frac{\beta_2(k)}{2^3x^4}+\cdots\right\}=0 \quad (54)$$

By equating to zero the coefficients of the successive powers of x in (54) we shall again obtain a system of recursions between the functions $\beta_r(k)$ r=0,1,.... . We have:

$$x^4: \qquad \phi\beta_0(k+2)+\beta_0(k)=2\phi, \quad (55)$$

$$x^2: \qquad \phi\beta_1(k+2)+\beta_1(k)=2\left\{\phi(\lambda+k)\beta_0(k+2)+(\lambda+2k)\beta_0(k)-2\phi(\lambda+k)\right\}, \quad (56)$$

$$x^{-2r+4}: \quad \phi\beta_r(k+2)+\beta_r(k) = 2\{\phi(\lambda+k)\beta_{r-1}(k+2)+(\lambda+2k)\beta_{r-1}(k) - 2(k^2+\lambda k+\mu)\beta_{r-2}(k)\} \tag{57}$$

Before proceeding further we introduce factorial functions of the form

$$\underset{2}{k}^{(s)} = k(k-2)\ldots(k-2s+2). \tag{58}$$

These quite clearly satisfy a recursion of the form

$$\underset{2}{k}^{(s+1)} = (k-2s)\,\underset{2}{k}^{(s)} \tag{59}$$

and thus

$$k\,\underset{2}{k}^{(s)} = \underset{2}{k}^{(s+1)} + 2s\,\underset{2}{k}^{(s)}. \tag{60}$$

Furthermore

$$\begin{aligned} k^2\,\underset{2}{k}^{(s)} &= k\,\underset{2}{k}^{(s+1)} + 2sk\,\underset{2}{k}^{(s)} \\ &= \underset{2}{k}^{(s+2)} + (4s+2)\,\underset{2}{k}^{(s+1)} + 4s^2\,\underset{2}{k}^{(s)}. \end{aligned} \tag{61}$$

If the difference and displacement operators $\underset{2}{\Delta}$ and $\underset{2}{E}$ are defined by

$$\underset{2}{\Delta}\,g(k) \equiv g(k+2)-g(k), \quad \underset{2}{E} \equiv 1+\underset{2}{\Delta} \tag{62}$$

then

$$\underset{2}{\Delta}\,\underset{2}{k}^{(s)} = 2s\,\underset{2}{k}^{(s-1)} \tag{63}$$

and

$$\underset{2}{(k+2)}^{(s)} = \underset{2}{k}^{(s)} + 2s\,\underset{2}{k}^{(s-1)}. \tag{64}$$

Equipped with these formulae, we see that equations (55)-(57) are formally satisfied by expressions of the form

$$\beta_r(k) = \sum_{s=0}^{r} q_{r,s}\, k_2^{(s)}. \qquad (65)$$

From (55) and (56) we have successively

$$\beta_0(k) = 2\phi/(\phi+1), \qquad (66)$$

$$\beta_1(k) = \frac{4\phi}{(\phi+1)^2} k_2^{(1)} - \frac{8\phi^2}{(\phi+1)^3}, \qquad (67)$$

and (57) may be rearranged to give

$$\begin{aligned}(\phi+1)\sum_{s=0}^{r} q_{r,s}\, k_2^{(s)} &+ 2\phi\sum_{s=0}^{r-1}(s+1)q_{r,s+1}\, k_2^{(s)} \\ = 2\Big\{(\phi+2)\sum_{s=1}^{r} q_{r-1,s-1}\, k_2^{(s)} &+ (\phi+1)\sum_{s=0}^{r-1}(4s+\lambda)q_{r-1,s}\, k_2^{(s)} \\ + 2\phi\sum_{s=0}^{r-2}(s+1)(\lambda+2s)q_{r-1,s+1}\, k_2^{(s)} &- 2\sum_{s=2}^{r} q_{r-2,s-2}\, k_2^{(s)} \\ - 2\sum_{s=1}^{r-1}(4s+\lambda-2)q_{r-2,s-1}\, k_2^{(s)} &- 2\sum_{s=0}^{r-2}(4s^2+2\lambda s+\mu)q_{r-2,s}\, k_2^{(s)}. \end{aligned} \qquad (68)$$

Again a definition of $q_{r,s}$ which is uniformly valid for $r=0,1,\ldots;\quad s=r,r-1,\ldots,0;$ may be given:

$$\begin{aligned} q_{r,s} = 2\{ &\text{ if } s<r \text{ then } -\phi(s+1)q_{r,s+1} \\ &+ \text{if } s>0 \text{ then } (\phi+2)q_{r-1,s-1} \\ &+ \text{if } s<r \text{ then } (\phi+1)(4s+\lambda)q_{r-1,s} \\ &+ \text{if } s<r-1 \text{ then } 2\phi(s+1)(\lambda+2s)q_{r-1,s+1} \\ &- \text{if } s>1 \text{ then } 2q_{r-2,s-2} \\ &- \text{if } 0<s<r \text{ then } 2(4s+\lambda-2)q_{r-2,s-1} \\ &- \text{if } s<r-1 \text{ then } 2\{2s(2s+\lambda)+\mu\}q_{r-2,s} \\ &+ \text{if } r=0 \text{ then } \phi \end{aligned}$$

$$- \text{ if } r=1 \text{ then if } s=0 \text{ then } 2\lambda\phi \text{ and if } s=1 \text{ then } 2\phi \} / (\phi+1) \tag{69}$$

Comparison with the work of Miller and Airey.

As mentioned at the beginning of this paper, Miller has derived relationships similar to equations (33)-(35) and (55)-(57) for the case in which z is real. Allowing for the difference in notation (Miller uses an auxilary variable b defined by b=a-2 as opposed to $\lambda = 2(a-1)$, and derives sets of equations in which the unknown function is $\beta_{r+1}(k)$ and not $\beta_r(k)$), equations (33)-(35) and (55)-(57) reduce to Miller's equations when $\phi=1$. Miller derives explicit formulae for the initial $\beta_r(k)$ rather than a recursive definition of the coefficients $p_{r,s}$ and $q_{r,s}$; nevertheless, since we have derived expressions for $\beta_s(k)(s=0,1,2,3)$ for the purpose of checking, we remark in passing that these expression reduce to those of Miller when $\phi=1$.

We now recall the work of Airey [2]. He is concerned with the asymptotic series

$$\frac{0!}{z'} - \frac{1!}{z'^2} + \frac{2!}{z'^3} - \ldots = \sum_{n=1}^{\infty} u_n \tag{70}$$

where

$$u_n = (-1)^{n-1}(n-1)!\, z'^{-n}, \tag{71}$$

and writes (70) as

$$\sum_{p=1}^{\infty} u_p \sim \sum_{p=1}^{n-1} u_p + u_n C_n \tag{72}$$

where

$$C_n \sim 1 - \frac{n}{z'} + \frac{n(n+1)}{z'^2} - \frac{n(n+1)(n+2)}{z'^3} + \ldots \tag{73}$$

He makes an auxiliary substitution

$$z' = xe^{i\theta'}, \quad \beta = e^{-i\theta'}, \quad x' = n+h, \tag{74}$$

which is similar to our (16), and obtains

$$C_n \sim 1 - \frac{(x-h)\beta}{x'} + \frac{(x-h)(x+1-h)\beta^2}{x'^2} - \frac{(x-h)(x+1-h)(x+2-h)\beta^3}{x'^3} + \dots \tag{75}$$

By formal expansion of each term of (75) in inverse powers of x', and regroupment, he obtains the expansion

$$C_n \sim \frac{1}{1+\beta} + \frac{1}{x'}\left\{\frac{-\beta^2}{(1+\beta)^3} + \frac{-\beta}{(1+\beta)^2}h\right\} + \frac{1}{x'^2}\left\{\frac{-2\beta^3+\beta^4}{(1+\beta)^5} - \frac{\beta^2-2\beta^3}{(1+\beta)^4}h + \frac{\beta^2 h^2}{(1+\beta)^3}\right\}$$
$$+\frac{1}{x'^3}\left\{\frac{6\beta^4-8\beta^5+\beta^6}{(1+\beta)^7} + \frac{2\beta^3-10\beta^4+3\beta^5}{(1+\beta)^6}h + \frac{-3\beta^3+3\beta^2}{(1+\beta)^5}h^2 + \frac{-\beta^3 h^3}{(1+\beta)^4}\right\} + \dots \tag{76}$$

Airey tabulated values of the coefficients of x^{-s} $(s=0,1,..)$ in this expression when $\beta=1$ and h=1. Miller noted that when $a=\frac{1}{2}$ or $\frac{3}{2}$ the constant terms of the polynomial coefficients which he derived for the expansion of Γ_n, were the same as Airey's numbers. We shall later see that, allowing for the difference in notation, the coefficients of x'^{-s} $(s=0,1,\dots)$ in (76) are in agreement with those given by (37)-(40).

At first sight this should seem to be more a cause for bewilderment than reassurance, for the asymptotic series

$$1 - \frac{(a+\frac{1}{2})(a+\frac{3}{2})}{2z^2} + \frac{(a+\frac{1}{2})(a+\frac{3}{2})(a+\frac{5}{2})(a+\frac{7}{2})}{2.4.z^4} - \dots \tag{77}$$

with which the Weber function may in some sense be associated, manifestly does not reduce to (70) when $a=\frac{1}{2}$ or $\frac{3}{2}$. When $a=\frac{1}{2}$ it becomes

$$1 - \frac{(1/2)}{(z^2/2)} + \frac{(1/2)(3/2)}{(z^2/2)^2} - \dots \qquad (78)$$

and when a= $3/2$

$$1 - \frac{(3/2)}{(z^2/2)} + \frac{(3/2)(5/2)}{(z^2/2)^2} - \dots \qquad (79)$$

In order to explain this curious agreement we must first establish the true significance of Airey's converging factor. We consider the asymptotic series development

$$1 - \frac{a}{z'} + \frac{a(a+1)}{z'^2} - \dots \qquad (80)$$

which may be associated with the incomplete Γ- function.

We write this as

$$\sum_{r=0}^{\infty} u_r = \sum_{r=0}^{n-1} u_r + u_n C_n \qquad (81)$$

where

$$u_r = (-1)^r \frac{a(a+1)\dots(a+r-1)}{z'^r} \qquad (82)$$

and the converging factor C_n may be expanded as

$$C_n \sim 1 - \frac{(a+n)}{z'} + \frac{(a+n)(a+n+1)}{z'^2} - \dots \qquad (83)$$

where n is so chosen that if

$$z' = x' e^{i\theta} \qquad (84)$$

and

$$a + n = x' + h \qquad (85)$$

then $0 \leq h \leq 1$.

Now C_n satisfies the differential equation

$$z'\frac{dC_n}{dz} - (a+n+z')C_n = -z'. \tag{86}$$

We may change the independent variable to h, and eliminate n form this equation by means of (85), and obtain

$$x'\frac{dC_n}{dh} + \left\{x'(1+e^{i\theta})+h\right\}C_n = x'e^{i\theta} \tag{87}$$

We may substitute a series development of the form $C_n \sim \sum_{s=0}^{\infty} \beta_s(h)x'^{-s}$ in (87) and obtain a recursion system among the $\beta_s(h)$ $(s=0,1,\ldots)$ as done earlier in this paper. The point to notice about this system of recursions is that the functions $\beta_s(h)$ produced via equation (87), are independent of the parameter a, so that Airey's converging factor (75) is not only the converging factor for the exponential integral, but also for the incomplete Γ-function.

But the series (78) and (79) are special cases of (80). The only outstanding point is that the relationship between z' and h given by (85) is exact, but that that between z^2 and k, given by (16), was derived under the assumption that μ (given by (13)) was negligible compared with h^2. But when a=1/2 or 3/2 μ is not only negligible but zero, and so the correspondence is complete, and the agreement referred to occurs.

It only remains to show how (37)-(40) reduces to (76) when $\mu=0$. Replacing h by the complementary argument h'=h-1 in (76) we obtain

$$\frac{1}{1+\beta} + \left\{\frac{\beta}{(1+\beta)^2}h' - \frac{\beta}{(1+\beta)^3}\right\}\frac{1}{x'} + \left\{\frac{\beta^2}{(1+\beta)^3}h'^2 - \frac{3\beta^2}{(1+\beta)^4}h' + \frac{\beta^2(2\beta-1)}{(1+\beta)^5}\right\}\frac{1}{x'^2}$$
$$+ \left\{\frac{\beta^3}{(1+\beta)^4}h'^3 - \frac{6\beta^3}{(1+\beta)^5}h'^2 + \frac{\beta^3(11-4\beta)h'}{(1+\beta)^6} - \frac{\beta^3(\beta^2-8\beta+6)}{(1+\beta)^7}\right\}\frac{1}{x'^3} + \ldots \tag{88}$$

If, in (88), we put $\emptyset = \beta^{-1}$, $k = 2h'$, and $x^2 = 2x'$ we arrive at the coefficients (37)-(40), and thus again Airey's work serves, to a certain extent, to check our own.

Singular Case

When z^2 is real and negative, $\emptyset = -1$; the formalism of the preceding two sections breaks down completely; we examine the problem afresh.

In the case being considered, equations (33)-(35),(55)-(57) become

$$2\beta_0'' - \beta_0' = -1 \tag{89}$$

$$2\beta_1'' - \beta_1' = -2(2k+\lambda-2)\beta_0' - (4-k)\beta_0 + 2(\lambda+k) \tag{90}$$

$$2\beta_r'' - \beta_r' = 2\{4r-\lambda-2k-2\}\beta_{r-1}' + \{k-2(r+1)\}\beta_{r-1} - 2\{k^2+k(\lambda-4r+4)k+\mu-2\lambda(r-1)+4(r-1)^2\}\beta_{r-2} \tag{91}$$

$$\underset{2}{\Delta}\beta_0(k) = 2 \tag{92}$$

$$\underset{2}{\Delta}\beta_1(k) = -2\{(\lambda+2k)\beta_0(k) - (\lambda+k)\beta_0(k+2) + 2(\lambda+k)\} \tag{93}$$

$$\underset{2}{\Delta}\beta_r(k) = -2\{(\lambda+2k)\beta_{r-1}(k) - (\lambda+k)\beta_{r-1}(k+2) - 2(k^2+\lambda k+\mu)\beta_{r-2}(k)\} \tag{94}$$

Inspection of equations (89)-(94) reveals that at least the possibility exists that they are satisfied by polynomials of the form

$$\beta_r(k) = \sum_{s=0}^{2r+1} p_{r,s}\, k^s = \sum_{s=0}^{2r+1} q_{r,s}\, k_2^{(s)} \tag{95}$$

But it is quite certain, at least, that equations (91) and (94) do not serve to determine $p_{r,0}$ and $q_{r,0}$ respectively, and since, for example $p_{r+1,2}$, $p_{r+1,1}$ are determined from $p_{r,0}$, it would appear that matters become progressively worse.

Let us, however, proceed upon the assumption that everything is known on the right hand sides of equations (91) and (94) except $p_{r-1,0}$ and $q_{r-1,0}$ respectively. Equations (89) and (92) give to begin with

$$p_{0,1} = q_{0,1} = 1. \tag{96}$$

Equation (91) may be rearranged as

$$\begin{aligned}
&2\sum_{s=0}^{2r-1}(s+1)(s+2)p_{r,s+2}k^s - \sum_{s=0}^{2r}(s+1)p_{r,s+1}k^s \\
&= 2(4r-\lambda-2)\sum_{s=0}^{2r-2}(s+1)p_{r-1,s+1}k^s - 4\sum_{s=1}^{2r-1} s\,p_{r-1,s}k^s \\
&\quad + \sum_{s=1}^{2r} p_{r-1,s-1}k^s - 2(r+1)\sum_{s=0}^{2r-1} p_{r-1,s}k^s - 2\sum_{s=2}^{2r-1} p_{r-2,s-2}k^s \\
&- 2(\lambda-4r+4)\sum_{s=1}^{2r-2} p_{r-2,s-1}k^s - 2\{\mu-2\lambda(r-1)+4(r-1)^2\}\sum_{s=0}^{2r-3} p_{r-2,s}k^s.
\end{aligned} \tag{97}$$

This leads to

$$\begin{aligned}
p_{r,s} = -\Big[\; & p_{r-1,s-2} \\
& - \text{if } s \leq 2r \text{ then } 2\{s(s+1)p_{r,s+1} + (r+2s-1)p_{r-1,s-1} + p_{r-2,s-3}\} \\
& + \text{if } s \leq 2r-1 \text{ then } 2\{s(4r-\lambda-2)p_{r-1,s} - (\lambda-4r+4)p_{r-2,s-2}\} \\
& - \text{if } s \leq 2r-2 \text{ then } 2\{\mu - 2\lambda(r-1) + 4(r-1)^2\}p_{r-2,s-1}\Big]/s, \\
& (s = 2r+1\,(-1)\,3),
\end{aligned} \tag{98}$$

a relationship which may be used without difficulty.

Equation (94) may be rearranged to give

$$\sum_{s=0}^{2r-1}(s+1)q_{r,s+1}\,\underset{2}{k}^{(s)} = \sum_{s=1}^{2r} q_{r-1,s-1}\,\underset{2}{k}^{(s)} - 2\sum_{s=0}^{2r-2}(s+1)(\lambda+2s)q_{r-1,s+1}\,\underset{2}{k}^{(s)}$$
$$-2\sum_{s=2}^{2r-1} q_{r-2,s-2}\,\underset{2}{k}^{(s)} - 2\sum_{s=1}^{2r-2}(4s+\lambda-2)q_{r-2,s-1}\,\underset{2}{k}^{(s)}$$
$$-2\sum_{s=0}^{2r-3}(4s^2+2\lambda s+\mu)q_{r-2,s}\,\underset{2}{k}^{(s)}. \qquad (99)$$

This leads to

$$q_{r,s} = -\Big[\, q_{r-1,s-2}$$
$$-\text{ if } s \leqslant 2r \text{ then } 2q_{r-2,s-3}$$
$$-\text{ if } s \leqslant 2r-1 \text{ then } 2\{s(\lambda+2s-2)q_{r-1,s}+(4s+\lambda-6)q_{r-2,s-2}\}$$
$$-\text{ if } s \leqslant 2r-2 \text{ then } 2\{4(s-1)^2+2\lambda(s-1)+\mu\}q_{r-2,s-1}\Big]/s$$
$$(s = 2r+1\,(-1)\,3). \qquad (100)$$

The coefficients of k^1 and k^0 in (97) and $\underset{2}{k}^{(1)}$ and $\underset{2}{k}^{(0)}$ in (99) respectively give

$$p_{r,2} = \{12p_{r,3} - 4(4r-\lambda-2)p_{r-1,2} - p_{r-1,0} + 2(r+3)p_{r-1,1}$$
$$+ 2\{\mu - 2\lambda(r-1) + 4(r-1)^2\}p_{r-2,1} + 2(\lambda-4r+4)p_{r-2,0}\}/2, \qquad (101)$$

$$p_{r,1} = 4p_{r,2} - 2(4r-\lambda-2)p_{r-1,1} + 2(r+1)p_{r-1,0} + 2\{\mu - 2\lambda(r-1) + 4(r-1)^2\}p_{r-2,0} , \quad (102)$$

$$q_{r,2} = -[q_{r-1,0} - 4(\lambda+2)q_{r-1,2} - 2(\lambda+2)q_{r-2,0} - 2(\mu+2\lambda+4)q_{r-2,1}]/2 , \quad (103)$$

$$q_{r,1} = 2(\lambda q_{r-1,1} + \mu q_{r-2,0}) . \quad (104)$$

Now so far we have used the facts that Γ_n satisfies a differential equation and a difference equation quite seperately and developed $\beta_r(k)$ as a polynomial and as a series of factorial functions quite independently. Now we must use these facts in conjunction.

Firstly

$$p_{r-1,0} = q_{r-1,0} \quad (105)$$

and secondly, as may easily be verified (c.f. equation (112) below)

$$q_{r,1} = p_{r,1} + 2p_{r,2} + 4p_{r,3} + \ldots + 2^{2r} p_{r,2r+1} . \quad (106)$$

Equations (101),(102),(104),(105) and (106) may thus be used to derive $p_{r-1,0} = q_{r-1,0}$, and these may be substituted in (101), (102) and (103) to give $p_{r,2}$, $p_{r,1}$ and $q_{r,2}$. $q_{r,1}$ can of course be determined without computing $q_{r-1,0}$.

Writing

$$p_{r,2} = 12p_{r,3} - 4(4r-\lambda-2)p_{r-1,2} + 2\{\mu - 2\lambda(r-1) + 4(r-1)^2\}p_{r-2,1} + 2(r+3)p_{r-1,1} + 2(\lambda-4r+4)p_{r-2,0} \quad (107)$$

and

$$P_{r1} = 2\{\mu - 2\lambda(r-1) + 4(r-1)^2\} p_{r-2,0} - 2(4r-\lambda-2) p_{r-1,1} \tag{108}$$

and using (106) we have

$$p_{r-1,0} = q_{r-1,0} = \{q_{r,1} - 3P_{r2} - P_{r1} - 4p_{r,3} - \dots - 2^{2r} p_{r,2r+1}\}/(2r-1). \tag{109}$$

Subsequently

$$p_{r,2} = \{P_{r2} - p_{r-1,0}\}/2 \tag{110}$$

$$p_{r,1} = 4p_{r,2} + P_{r1} + 2(r+1)p_{r-1,0} ; \tag{111}$$

$q_{r,2}$ is given by (103), and we may proceed to the next value of r. Use of conditional statements enables the anomalous equations (89),(90),(92) and (93) to be brought into this general scheme.

Checking

Since the expressions $\beta_r(k)$, whether derived as a polynomial or as a series of factorials, represent the same function, there exists the possibility of expressing one set of coefficients in terms of the other, and this may be used as a check.

In the non-singular case we have the matrix equations

$$(q_{r,s}) = (p_{r,s})L \quad , \quad (p_{r,s}) = (q_{r,s})L^{-1} \tag{112}$$

where

$$(p_{r,s}) = \begin{pmatrix} p_{0,0} & & \\ p_{1,0} & p_{1,1} & \\ p_{2,0} & p_{2,1} & p_{2,2} \\ \vdots & \vdots & \vdots \end{pmatrix} , \quad (q_{r,s}) = \begin{pmatrix} q_{0,0} & & \\ q_{1,0} & q_{1,1} & \\ q_{2,0} & q_{2,1} & q_{2,2} \\ \vdots & \vdots & \vdots \end{pmatrix} \tag{113}$$

and

$$L = \begin{pmatrix} 1 & & & & \\ 0 & 1 & & & \\ 0 & 2 & 1 & & \\ 0 & 4 & 6 & 1 & \\ 0 & 8 & 28 & 12 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}, \quad L^{-1} = \begin{pmatrix} 1 & & & & \\ 0 & 1 & & & \\ 0 & -2 & 1 & & \\ 0 & 8 & -6 & 1 & \\ 0 & -48 & 44 & -12 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix} \qquad (114)$$

If the elements in L are referred to as $l_{r,s}$ (r,s=0,1,...) and those in L^{-1} as $l^{-1}_{r,s}$ (r,s=0,1,...) then

$$l_{r,0} = 0, \; l_{r,s} = l_{r-1,s-1} + 2s\,l_{r-1,s}, \quad (r = 1,2,\dots;\ s = 1,2,\dots,r)$$

$$l^{-1}_{r,0} = 0, \; l^{-1}_{r,s} = l^{-1}_{r-1,s-1} - 2(r-1)\,l^{-1}_{r-1,s} \quad (r = 1,2,\dots;\ s = 1,2,\dots,r).$$

Use of these formulae (as we shall see in the ALGOL programme to be given) enables the matrix multiplications (112) to be replaced by a system of algebraic relationships.

Application of the ϵ-algorithm.

We have now shown how the converging factor Γ_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 Γ_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 have to state that if from the initial values

$$\epsilon_0^{(0)} = 0, \quad \epsilon_0^{(m)} = \sum_{r=0}^{m-1} \beta_r(k) n^{-r} \qquad (m=1,2,\dots) \qquad (115)$$

$$\epsilon_1^{(m)} = n^m \{\beta_m(k)\}^{-1} \qquad (m=0,1,\ldots) \quad (116)$$

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

$$\epsilon_s^{(m)} = \epsilon_{s-2}^{(m+1)} + \frac{1}{\epsilon_{s-1}^{(m+1)} - \epsilon_{s-1}^{(m)}} \quad (117)$$

then the quantities $\epsilon_{2s}^{(m)}$ are convergents of certain continued functions, and as such provide better estimates of the formal sum of the series whose partial sums are given by (115) than the partial sums. The quantities $\epsilon_s^{(m)}$ may be displayed in the array

$$\begin{array}{llll}
\epsilon_0^{(0)} & & & \\
 & \epsilon_1^{(0)} & & \\
\epsilon_0^{(1)} & & \epsilon_2^{(0)} & \\
 & \epsilon_1^{(1)} & & \epsilon_3^{(0)} \\
\epsilon_0^{(2)} & & \epsilon_2^{(1)} & \cdot \\
 & \epsilon_1^{(2)} & \cdot & \cdot \\
\epsilon_0^{(3)} & & \cdot & \cdot \\
 & \cdot & \cdot & \\
\cdot & \cdot & & \\
\cdot & \cdot & & \\
\cdot & & &
\end{array}$$

Table I.

and it can be seen that the quantities in (117) occur at the vertices of a lozenge in this array. The various numbers 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: $l_0 \equiv \epsilon_0^{(m)}, l_1 \equiv \epsilon_1^{(m-1)}, l_2 \equiv \epsilon_2^{(m-2)}, \ldots, l_m = \epsilon_m^{(0)}$. (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 $\epsilon_0^{(m+1)}$ and replace in succession

τ_0 by $\epsilon_0^{(m+1)}$, τ_1 by $\epsilon_1^{(m)}$, ..., τ_m by $\epsilon_m^{(1)}$, and add $\tau_{m+1} \equiv \epsilon_{m+1}^{(0)}$. The formation of these quantities is carried out by means of (117) and uses one working space and two auxiliary storage locations. In certain singular cases, as occur for example when a term is equal to zero, the latter procedure breaks down. This difficulty may be overcome by certain singular rules [7].

An ALGOL Programme

We now summarise the formalism which has been developed, in the form of an ALGOL programme. It must be borne in mind, however, that application of the converging factor to an asymptotic series is but one of a number of methods by means of which the Weber function may be computed. Thus this programme is not to be regarded as any sort of fool-proof procedure by means of which the Weber function may be computed for any value of argument and parameter. It should be regarded as a basis from which the interested reader if he so desires may, at the cost of an hour or so of somebody else's typing, continue the author's provisional inquiry into the numerical behaviour of the converging factor.

Before giving the programme it is necessary to make a few remarks. The algorithmic language [5] 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 devise 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 (eg. z, $p_{r,s}$) may be recognised throughout the programme by virtue of the fact that they contain the index i (e.g. z [i], p [R,s,i]). i takes two values, zero corresponding to the real part (e.g. Re(z) ≡ z [0], Re($p_{r,s}$) ≡ p [R,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 e.g. (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 [6] .

It will be recalled that $\beta_r(k)$ is determined from $\beta_{r-1}(k)$ and $\beta_{r-2}(k)$, thus we need only store in the machine two vectors of coefficients, since when $\beta_r(k)$ has been computed its coefficients may be written upon the space occupied by those of $\beta_{r-2}(k)$ since the latter are no longer needed. But we also wish to make the programme as comprehensible at a glance as possible. We therefore introduce integers R, Rminus 1, Rminus 2 which take on the values 0,1,0 when r is even and 1,0,1 when r is odd. In this way the mathematical formulae and the algorithmic formulae preserve a close similarity, and the required economy in the use of storage space is achieved.

Having evaluated $\beta_r(k)$ (by a Horner process in both the cases in which $\beta_r(k)$ is expressed as a polynomial and as a series of factorial function) the series $\sum_{r=0} \beta_r(k)\, 2^{-r-1} x^{-2r}$ is summed either as far as a given upper bound rmax, or until

$$|\beta_{r+1}(k)\, 2^{-r-2} x^{-2r-2}| > |\beta_r(k)\, 2^{-r-1} x^{-2r}| \text{ and}$$

$$|\beta_{r+2}(k)\, 2^{-r-3} x^{-2r-4}| > |\beta_{r+1}(k)\, 2^{-r-2} x^{-2r-2}| \qquad (118)$$

when it is assumed that the converging factor series has itself an asymptotic character and has begun to diverge.

As the terms $\beta_r(k)\,2^{-r-1}x^{-2r}$ are produced the ϵ-algorithm is applied immediately. It will be recalled that only the quantities $\epsilon_s^{(m)}$ with even suffix are of interest in the present application. As these are produced they are mapped onto a display vector (di[0, ms, i]), and afterwards picked out and printed in a table which corresponds to the ϵ-arrays (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, a,x, and θ/π, and immediately prints out a,x, θ/π, k and n. It then computes and prints out the terms $u_0, u_1, \ldots, u_{n-1}$ of the asymptotic series, their partial sum, and u_n. It then computes and prints out (real and imaginary parts seperately) the coefficients $p_{r,s}$, the coefficients $q_{r,s}$ derived from them by means of equation (112), the value of the polynomial $\beta_r(k)$ (real part, imaginary part, modulus) and of the term $\beta_r(k)\,2^{-r-1}x^{-2r}$; if condition (118) is not obeyed the term is added in to the converging factor sum. Application of the ϵ-algorithm to the converging factor takes place at the same time. After $r = rmax$ the numerical sum $\Gamma_n = \sum_{r=0} \beta_r(k)\,2^{-r-1}x^{-2r}$, the product $u_n\Gamma_n$, and the modified sum $\sum_{r=0}^{n-1} u_r + u_n\Gamma_n$ are printed out in turn (real part, imaginary part, and modulus). Next the (triangular) even column ϵ-array resulting from the application of the ϵ-algorithm to the converging factor are printed (real and imaginary parts seperately) and two further triangular arrays which correspond to the application of the transformed converging factor are also printed. The whole process is then repeated with the computation of $q_{r,s}$.

In this way one is able to observe the numerical behaviour of the asymptotic series (6), the coefficients $p_{r,s}, q_{r,s}$ and to check these; one is able to observe how rapidly the converging factor series converges, the effect of applying the ϵ-algorithm to it, and the improvement which is to be obtained by applying it.

A seperate programme has been made for the singular case in which $\arg(z) = \pi/2$. Its construction is as above with the exception that all the quantities involved are real, and the computation of $p_{r,s}$ and $q_{r,s}$ proceeds simultaneously.

```
comment Converging factor for Weber function of complex argument ;
begin integer rmax ;rmax:=read ;
begin real a,x,multiple of pi,xsquared,lambda,mu,k,theta,power of x ;
integer i,r,s,n,j,twormax,sanfang,rs,col,R,Rmin1,Rmin2,r1 ;
boolean polynomial,still converging,display converging factor alone ;
array aux0,aux1,aux2,phi,z,zsquared,u,sum,converging factor[0:1],
pq[0:1,0:rmax,0:1],betar,termr[-2:0,0:1],modtermr[-2:0],f[0:rmax],
check[0:rmax,0:1],l[0:rmax+1,0:1],di[0:1,1:((rmax+1)×(rmax+5))÷4,0:1] ;
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 ;
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 ;
real procedure real(variable) ;real variable ;
real:=( if i=0 then variable else 0.0 ) ;
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 ;
```

```
real procedure arg(it) ;real it ;begin real Reit,Imit ;
i:=0 ;Reit:=it ;i:=1 ;Imit:=it ;
arg:=( if Reit>0.0 then arctan(Imit/Reit) else
sign(Imit)×1.57079 63267 949-arctan(Reit/Imit) ) end ;
procedure polar form(res,r,theta) ;real r,theta ;begin real r1,theta1 ;
r1:=r ;theta1:=theta ;i:=0 ;res:=r1×cos(theta1) ;
i:=1 ;res:=r1×sin(theta1) end ;
procedure compln(res,it) ;real res,it ;begin real aux ;
aux:=ln(mod(it)) ;i:=0 ;res:=aux ;aux:=arg(it) ;i:=1 ;res:=aux end ;
procedure compexp(res,it) ;real res,it ;begin real aux1,aux2 ;
i:=0 ;aux1:=exp(it) ;i:=1 ;aux2:=it ;
res:=aux1×sin(aux2) ;i:=0 ;res:=aux1×cos(aux2) end ;
procedure onehochother(res,one,other) ;real res,one,other ;
begin array aux1[0:1] ;compln(aux1[i],one) ;cm(aux1[i],other,aux1[i]) ;
compexp(res,aux1[i]) end ;
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 ;
procedure compprint(it) ;real it ;for i:=0,1 do print(it) ;
procedure druck(it) ;real it ;begin compprint(it) ;print(mod(it)) end ;
procedure printcompvect(it,j,h,k,col) ;value h,k,col ;
integer j,h,k,col ;real it ;begin integer janfang ;
for janfang:=h step col until k do for i:=0,1 do begin NLCR ;
for j:=janfang step 1 until janfang+col-1 do if j ≤ k then print(it) end end ;
boolean procedure even(integer) ;integer integer ;
even:=( integer=(integer:2)×2 ) ;
```

```
procedure addin(one,other) ;real one,other ;
begin array aux3[0:1] ;cm(aux3[i],one,other) ;
eq(aux1[i],aux1[i]+aux3[i]) end ;
procedure NT ;begin NLCR ;NLCR ;TAB ;TAB ;TAB end ;
procedure sum and display converging factor ;
begin NLCR ;druck(betar[-2,i]) ;druck(termr[-2,i]) ;
eq(converging factor[i],converging factor[i]+termr[-2,i]) ;
for s:=-2,-1 do begin eq(betar[s,i],betar[s+1,i]) ;
eq(termr[s,i],termr[s+1,i]) ;modtermr[s]:=modtermr[s+1] end end ;
comment Introduction ;
a:=read ;x:=read ;multiple of pi:=read ;col:=read ;
xsquared:=x×x ;lambda:=2×(a-1) ;mu:=(a-0.5)×(a-1.5) ;
n:=(entier(xsquared-lambda))÷2 ;
if n<0 then begin n:=0 ;k:=xsquared-lambda end else
begin k:=xsquared-lambda-2×n ;if k>1.0 then begin k:=k-2.0 ;n:=n+1 end end ;
NLCR ;print(a) ;print(x) ;print(multiple of pi) ;print(n) ;print(k) ;
twormax:=2×rmax ;theta:=multiple of pi×3.14159 26535 8979 ;
polar form(phi[i],1.0,2.0×theta) ;polar form(z[i],x,theta) ;
eq(zsquared[i],xsquared×phi[i]) ;
comment Evaluation of terms and partial sum of asymptotic series ;
eq(sum[i],0.0) ;compexp(aux1[i],-zsquared[i]/4.0) ;
onehochother(aux2[i],z[i],real(-a-0.5)) ;cm(u[i],aux1[i],aux2[i]) ;
for r:=1 step 1 until n do begin NLCR ;druck(u[i]) ;eq(sum[i],sum[i]+u[i]) ;
cd(u[i],-(a+2×r-0.5)×(a+2×r-1.5)×u[i],2×r×zsquared[i]) end ;
NLCR ;NLCR ;druck(sum[i]) ;NLCR ;NLCR ;druck(u[i]) ;
```

```
comment Computation of converging factor ;
polynomial:=true ;
COEFFICIENTS: eq(l[0,i],0.0) ;power of x:=2.0 ;still converging:=true ;
eq(converging factor[i],0.0) ;R:=0 ;
for r:=0 step 1 until rmax do begin Rmin1:=R ;R:=Rmin2:=1-Rmin1 ;
for s:=r step -1 until 0 do begin
if polynomial then begin
comment Determination of polynomial coefficients ;
eq(aux1[i],(if r=0 then 2×phi[i] else 0)
-(if r=1 ∧ s=0 then 4×lambda×phi[i] else if r=1 ∧ s=1 then 4×phi[i] else 0)
-(if s<r ∧ s>0 then 8×s×pq[Rmin1,s,i] else 0)
-(if s<r-1 then 4×(s+2)×(s+1)×pq[R,s+2,i] else 0)
+(if s<r-1 then 4×(4×r-lambda-2)×(s+1)×pq[Rmin1,s+1,i] else 0)
-(if s>1 then 4×pq[Rmin2,s-2,i] else 0)
-(if s>0 ∧ s<r then 4×(lambda-4×r+4)×pq[Rmin2,s-1,i] else 0)
-(if s<r-1 then 4×(mu+2×(r-1)×(2×(r-1)-lambda))×pq[Rmin2,s,i] else 0) ) ;
if s<r then begin addin(2×(s+1)×(phi[i]+real(2.0)),pq[R,s+1,i]) ;
addin(2×((phi[i]+real(1.0))×lambda+2×phi[i]-2×r×(phi[i]+real(2.0))),pq[Rmin1,s,i])
end ;
if s>0 then addin(2×(phi[i]+real(2.0)),pq[Rmin1,s-1,i]) ;
cd(pq[R,s,i],aux1[i],(phi[i]+real(1.0))) end else begin
```

```
comment Determination of factorial coefficients ;
eq(aux1[i],2×((if r=0 then phi[i] else 0)
-(if r=1 ∧ s=0 then 2×lambda×phi[i] else if r=1 ∧ s=1 then 2×phi[i] else 0)
-(if s>1 then 2×pq[Rmin2,s-2,i] else 0)
-(if s<r ∧ s>0 then 2×(4×s+lambda-2)×pq[Rmin2,s-1,i] else 0)
-(if s<r-1 then 2×(2×s×(2×s+lambda)+mu)×pq[Rmin2,s,i] else 0) ) ) ;
if s<r then begin addin(-2×(s+1)×phi[i],pq[R,s+1,i]) ;
addin(2×(4×s+lambda)×(phi[i]+real(1.0)),pq[Rmin1,s,i]) end ;
if s < r-1 then addin(4×(s+1)×(lambda+2×s)×phi[i],pq[Rmin1,s+1,i]) ;
if s>0 then addin(2×(phi[i]+real(2.0)),pq[Rmin1,s-1,i]) ;
cd(pq[R,s,i],aux1[i],phi[i]+real(1.0)) end ;
eq(check[s,i],0.0) ;f[s]:=0.0 end ;
comment Printing out coefficients ;
printcompvect(pq[R,s,i],s,0,r,col) ;
comment Evaluation of factorial coefficients in terms of polynomial
coefficients and conversely ;
eq(check[0,i],pq[R,0,i]) ;f[1]:=1.0 ;
for s:=1 step 1 until r do for j:=s step -1 until 1 do begin
if s>1 then f[j]:=f[j-1]+(if polynomial then 2×j else -2×(s-1))×f[j] ;
eq(check[j,i],check[j,i]+f[j]×pq[R,s,i]) end ;
printcompvect(check[s,i],s,0,r,col) ;
comment Evaluation of betar(k) and corresponding term in converging factor
series ;
r1:=(if r>2 then 0 else r-2 ) ;
eq(betar[r1,i],0.0) ;for s:=r step -1 until 0 do
eq(betar[r1,i],pq[R,s,i]+( if polynomial then k else k-2×s )×betar[r1,i]) ;
eq(termr[r1,i],betar[r1,i]/power of x) ;modtermr[r1]:=mod(termr[r1,i]) ;
```

```
comment Add in converging factor term if series still converging ;
if r≥2 ∧ still converging then begin if
modtermr[-2] > modtermr[-1] ∧ modtermr[-1] > modtermr[0]
then sum and display converging factor else still converging:=false end ;
comment Application of epsilon algorithm to converging factor series ;
eq(aux1[i],termr[r1,i]+l[0,i]) ;
for s:=0 step 1 until r do
begin compprecip(aux0[i],( if s=0 then termr[r1,i] else aux1[i]-l[s,i] )) ;
if s≠0 then begin eq(aux0[i],aux0[i]+l[s-1,i]) ;eq(l[s-1,i],aux2[i]) end ;
eq(aux2[i],aux1[i]) ;eq(aux1[i],aux0[i]) ;
if even(s) then begin rs:=(s×(twormax+2-s))÷4+r+1 ;eq(di[0,rs,i],aux2[i]) ;
cm(di[1,rs,i],u[i],aux2[i]) end ;
if s=r ∧ ¬even(r) then begin rs:=((r+1)×(twormax-r+5))÷4 ;
eq(di[0,rs,i],aux1[i]) ;cm(di[1,rs,i],u[i],aux1[i]) end end ;
eq(l[r,i],aux2[i]) ;eq(l[r+1,i],aux1[i]) ;
power of x:=2×xsquared×power of x end ;
if still converging ∧ modtermr[-1] ≤ modtermr[-2] then
begin sum and display converging factor ;
sum and display converging factor  end ;
comment Print converging factor,product of converging factor and un,
and modified sum ;
NT ;druck(converging factor[i]) ;cm(aux1[i],u[i],converging factor[i]) ;
NT ;druck(aux1[i]) ;NT ;druck(sum[i]+aux1[i]) ;
```

```
comment Display application of epsilon algorithm to converging factor
and the corresponding modified sums ;
display converging factor 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( if display converging factor alone then di[0,rs,i] else sum[i]+di[1,rs,i] )
end end end end ;
if display converging factor alone then begin
display converging factor alone:=false ;goto TRIANGULAR DISPLAY end ;
if polynomial then begin polynomial:=false ;goto COEFFICIENTS end end end
```

```
comment Converging factor for Weber function of pure imaginary argument ;
begin integer rmax,twormax ;rmax:=read ;twormax:=2×rmax ;
begin real a,k,x,xsquared,lambda,mu,power of x,sum,u,auxsum,rmin1,
coefftofpRmin2smin1,aux0,aux1,aux2,Pr2,Pr1,qR1,converging factor ;
integer n,s,r,i,rs,sanfang,r1,col,R,Rmin1,Rmin2,twor ;
boolean polynomial,still converging,display converging factor alone ;
array check,f[0:twormax+3],betar,termr[-2:0],l[0:rmax+1],
di[1:((rmax+1)×(rmax+5)):4],p,q[0:1,0:twormax+3] ;
boolean procedure even(integer) ;integer integer ;
even:=( integer=(integer:2)×2 ) ;
procedure sum and display converging factor ;
begin NLCR ;print(betar[-2]) ;print(termr[-2]) ;
converging factor:=converging factor+termr[-2] ;
for s:=-2,-1 do begin betar[s]:=betar[s+1] ;termr[s]:=termr[s+1] end end ;
a:=read ;x:=read ;col:=read ;xsquared:=x×x ;
lambda:=2.0×(a-1.0) ;mu:=(a-0.5)×(a-1.5) ;n:=(entier(xsquared-lambda)):2 ;
if n<0 then begin n:=0 ;k:=xsquared-lambda end else
begin k:=xsquared-lambda-2×n ;if k>1.0 then begin k:=k-2.0 ;n:=n+1 end end ;
NLCR ;print(a) ;print(x) ;print(n) ;print(k) ;
comment Evaluation of terms and partial sum of asymptotic series ;
sum:=0.0 ;u:=exp(xsquared/4.0)×x↑(-a-0.5) ;
for r:=1 step 1 until n do begin NLCR ;print(u) ;sum:=sum+u ;
u:=(a+2×r-0.5)×(a+2×r-1.5)×u/(2×r×xsquared) end ;
NLCR ;NLCR ;print(sum) ;NLCR ;NLCR ;print(u) ;
```

```
comment Evaluation of converging factor coefficients ;
p[0,1]:=q[0,1]:=1.0 ;R:=Rmin2:=1 ;Rmin1:=0 ;power of x:=2.0 ;
converging factor:=l[0]:=0.0 ;still converging:=true ;
for r:=1 step 1 until rmax+1 do begin twor:=2×r ;rmin1:=r-1 ;
r1:=( if r>2 then 0 else r-3 ) ;polynomial:=true ;auxsum:=0.0 ;
coefftofpRmin2smin1:=mu+2.0×rmin1×(2.0×rmin1-lambda) ;
COEFFICIENTS: for s:=twor+1 step -1 until 3 do
begin if polynomial then begin
p[R,s]:=-(p[Rmin1,s-2]
+2.0×(-( if s≤twor then s×(s+1)×p[R,s+1]+(r+2×s-1)×p[Rmin1,s-1]+p[Rmin2,s-3]
        else 0.0 )
      +( if s≤twor-1 then s×(4×r-lambda-2)×p[Rmin1,s]
                         -(lambda-4×r+4)×p[Rmin2,s-2] else 0.0 )
      -( if s≤twor-2 then coefftofpRmin2smin1×p[Rmin2,s-1] else 0.0) ) )/s ;
auxsum:=p[R,s]+2.0×auxsum end
else
q[R,s]:=-(q[Rmin1,s-2]
-2.0×(( if s≤twor then q[Rmin2,s-3] else 0.0 )
      +( if s≤twor-1 then s×(lambda+2×s-2)×q[Rmin1,s]
                         +(lambda+2×(2×s-3))×q[Rmin2,s-2] else 0.0 )
      +( if s≤twor-2 then ((s-1)×(4×(s-1)+2×lambda)+mu)×q[Rmin2,s-1]
                    else 0.0 ) ) )/s end ;
if polynomial then begin polynomial:=false ;goto COEFFICIENTS end ;
Pr2:=12.0×p[R,3]-2.0×(( if r>1 then 2.0×(4.0×r-lambda-2.0)×p[Rmin1,2]
     -coefftofpRmin2smin1×p[Rmin2,1]-(lambda-4.0×r+4.0)×p[Rmin2,0]
                         else 1.0 )
-(r+3)×p[Rmin1,1] ) ;
```

```
Pr1:=2.0×(( if r>1 then coefftofpRmin2smin1×p[Rmin2,0] else -lambda )
-(4×r-lambda-2.0)×p[Rmin1,1] ) ;
qR1:=2.0×(lambda×q[Rmin1,1]+( if r>1 then mu×q[Rmin2,0] else -lambda )) ;
p[Rmin1,0]:=q[Rmin1,0]:=(qR1-3×Pr2-Pr1-4.0×auxsum)/(2×r-1) ;
p[R,2]:=(Pr2-p[Rmin1,0])/2.0 ;p[R,1]:=4.0×p[R,2]+Pr1+2.0×(r+1)×p[Rmin1,0] ;
q[R,2]:=( -q[Rmin1,0]
         +( if r > 1 then 2.0×((lambda+2.0)×(q[Rmin2,0]+2.0×q[Rmin1,2])
         +(mu+2.0×lambda+4)×q[Rmin2,1]) else -2.0))/2.0 ;
q[R,1]:=qR1 ;
comment Print p[r-1,0] ;
NLCR ;print(p[Rmin1,0]) ;
comment Add in converging factor term if series still converging ;
betar[r1]:=0.0 ;
for s:=twor-1 step -1 until 0 do betar[r1]:=k×betar[r1]+p[Rmin1,s] ;
termr[r1]:=betar[r1]/power of x ;
if r>3 ∧ still converging then begin
if abs(termr[-2]) > abs(termr[-1]) ∧ abs(termr[-1]) > abs(termr[0]) then
sum and display converging factor else still converging:=false end ;
comment Print Converging factor coefficients ;
polynomial:=true ;
COEFFT PRINT: NLCR ;for s:=0 step 1 until twor+1 do begin
if (s÷col)×col=s then NLCR ;
if s=0 then TAB else print( if polynomial then p[R,s] else q[R,s] ) end ;
if polynomial then begin polynomial:=false ;goto COEFFT PRINT end ;
```

```
comment Evaluation of polynomial coefficients in terms of factorial
coefficients and conversely ;
polynomial:=true ;
CHECK: for s:=1 step 1 until twor+1 do f[s]:=check[s]:=0.0 ;
f[1]:=1.0 ;
for s:=1 step 1 until twor+1 do for i:=s step -1 until 1 do begin
if s>1 then f[i]:=f[i-1]+( if polynomial then -2×(s-1) else 2×i )×f[i] ;
check[i]:=check[i]+f[i]×( if polynomial then q[R,s] else p[R,s] ) end ;
NLCR ;for s:=0 step 1 until twor+1 do begin if (s÷col)×col=s then NLCR ;
if s=0 then TAB else print(check[s]) end ;
if polynomial then begin polynomial:=false ;goto CHECK end ;
comment Application of epsilon algorithm to converging factor series ;
aux1:=termr[r1]+l[0] ;
for s:=0 step 1 until r-1 do begin
aux0:=1.0/( if s=0 then termr[r1] else aux1-l[0] ) ;
if s≠0 then begin aux0:=aux0+l[s-1] ;l[s-1]:=aux2 end ;
aux2:=aux1 ;aux1:=aux0 ;
if even(s) then di[(s×(twormax+2-s))÷4+r]:=aux2 ;
if s=r-1 ∧ even(r) then di[(r×(twormax-r+6))÷4]:=aux1 end ;
l[r-1]:=aux2 ;l[r]:=aux1 ;power of x:=2.0×squared×power of x ;
Rmin1:=R ;R:=Rmin2:=1-Rmin1 end ;
if still converging ∧ abs(termr[-1]) ≤ abs(termr[-2]) then
begin sum and display converging factor ;
sum and display converging factor end ;
```

```
comment Print converging factor,product of converging factor and un,
and modified sum ;
NLCR ;NLCR ;print(converging factor) ;aux0:=u×converging factor ;
NLCR ;NLCR ;print(aux0) ;NLCR ;NLCR ;print(sum+aux0) ;
comment Display application of epsilon algorithm to converging factor
and the corresponding modified sums ;
display converging factor alone:=true ;
TRIANGULAR DISPLAY: 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( if display converging factor alone then di[rs] else sum+u×di[rs] )
end end end ;
if display converging factor alone then begin
display converging factor alone:=false ;goto TRIANGULAR DISPLAY end
end end
```

Numerical Results

The Non-singular Case

Some numerical results which have been produced by means of the preceding ALGOL programmes are summarised in the following tables which relate to the choice of argument $z=3.5e^{i\pi/4}$, a=0.0 (i.e. n=7,k=0.25)

Table I gives the terms (real part, imaginary part and modulus) and the partial sum of the asymptotic series (6)

r	$Re(u_r)$		$Im(u_r)$	$\lvert u_r \rvert$
0	-0.50845 2329	+	0.16489 5465	0.53452 2484
1	-0.00504 7820	-	0.01556 4867	0.01636 2933
2	+0.00277 9441	-	0.00090 1396	0.00292 1952
3	+0.00030 3531	+	0.00093 5934	0.00098 3923
4	-0.00046 5579	+	0.00015 0991	0.00048 9451
5	-0.00009 9531	-	0.00030 6902	0.00032 2638
6	+0.00025 2098	-	0.00008 1758	0.00026 5024
$\sum_{r=0}^{6} u_r$	-0.51073 0190	+	0.14912 7467	0.53205 6696
7	+0.00008 0447	+	0.00024 8056	0.00026 0775

Table I

Tables II and III give the polynomial coefficients $p_{r,s}$ and factorial coefficients $q_{r,s}$ respectively

r \ s	0	1	2	3	5
0	+ 1.0				
	+ 1.0i				
1	- 2.0	+ 2.0			
	- 2.0i	+ 0.0i			
2	+ 1.0	- 12.0	+ 2.0		
	+ 12.0i	+ 0.0i	- 2.0i		
3	+ 60.0	+ 76.0	- 24.0	+ 0.0	
	- 98.0i	+ 38.0i	+ 24.0i	- 4.0i	
4	-1175.5	+ 480.0	+ 336.0	- 0.0	- 4.0
	+ 747.5i	- 872.0i	- 170.0i	+ 80.0i	- 4.0i

Table II

r \ s	0	1	2	3	4
0	+ 1.0				
	+ 1.0i				
1	- 2.0	+ 2.0			
	- 2.0i	+ 0.0i			
2	+ 1.0	- 8.0	+ 2.0		
	+ 12.0i	- 4.0i	- 2.0i		
3	+ 60.0	+ 28.0	- 24.0	0.0	
	- 98.0i	+ 70.0i	+ 0.0i	- 4.0i	
4	-1175.5	+ 160.0	+ 224.0	-48.0	- 4.0
	+747.5i	- 924.0i	+ 198.0i	+32.0i	- 4.0i

Table III

Table IV gives the values of the coefficients β_r (k) and the terms $\beta_r(k)2^{-r-1}x^{-2r}$, the numerical sum of the converging factors series, the product u_nC_n and the modified sum $\sum_{r=0}^{n-1} u_r + u_nC_n$

r	$\mathrm{Re}\{\beta_r(k)\}$	$\mathrm{Im}\{\beta_r(k)\}$	$\lvert\beta_r(k)\rvert$	$\mathrm{Re}\left\{\frac{\beta_r(k)}{2^{r+1}x^{2r}}\right\}$	$\mathrm{Im}\left\{\frac{\beta_r(k)}{2^{r+1}x^{2r}}\right\}$	$\left\lvert\frac{\beta_r(k)}{2^{r+1}x^{2r}}\right\rvert$
0	+1.0	+1.0	1.414214	+0.5	+0.5	0.707107
1	-1.5	-2.0	2.5	-0.030612	-0.040816	0.051020
2	-1.875	+11.0	12.0221	-0.001562	+0.009892	0.010014
3	+77.5	-87.0625	116.560	+0.002635	-0.002960	0.003963
4	-1274.52	+520.109	1376.56	-0.001769	+0.000722	0.001910
			C_n	+0.468692	+0.466837	0.661520
			$u_n C_n$	-0.000078097	+0.000153817	0.000172508
			$\sum_{r=0}^{n-1} u_r + u_n C_n$	-0.510808287	+0.149281284	0.532174791

Table IV

Tables V and VI give the real and imaginary parts respectively of those modified sums which are to be derived by applying the ϵ-algorithm to the converging factor series, and using the members of the resulting even column ϵ-array as approximations to the converging factor

m \ s	0	2	4
1	-0.51081 3995	-0.51080 6941	
2	.51080 6332	.51080 8523	-0.51080 8194
3	.51080 8912	.51080 8171	-0.51080 8220
4	.51080 7966	-0.51080 8223	
5	-0.51080 8287		

Table V

m \ s	0	2	4
1	+ 0. 14929 1718	+ 0.14928 1499	
2	. 14928 0841	.14928 1472	+0.14928 1461
3	. 14928 1250	.14928 1442	+0.14928 1440
4	. 14928 1665	+ 0.14928 1438	
5	+ 0. 14928 1284		

Table VI

The value of $S_1(a;z)$ computed by means of the asymptotic series and converging factor may be checked by use of the convergent ascending power series

$$S_1(a;z) = 2^{-a/2-1/4}\pi^{1/2}e^{-z^2/4}\left\{\frac{{}_1F_1\left(\frac{a}{2}+\frac{1}{4};\frac{1}{2};\frac{1}{2}z^2\right)}{\Gamma\left(\frac{a}{2}+\frac{3}{4}\right)} - 2^{1/2}z\frac{{}_1F_1\left(\frac{a}{2}+\frac{3}{4};\frac{3}{2};\frac{1}{2}z^2\right)}{\Gamma\left(\frac{a}{2}+\frac{1}{4}\right)}\right\} \quad (119)$$

An ALGOL programme which computes the function $S_1(a;z)$ by means of the series (119) is given in [6]. When $a=0$ and $z=3\cdot5e^{i\pi/4}$, the correct value of $S_1(a;z)$ computed by means of formula (119) is
$-0.51080\,8214 + i0.14928\,1449$.

In Tables VII and VIII respectively are given the polynomial factorial coefficients $p_{r,s}$ and $q_{r,s}$ when $\theta=0$ and $a=1/2$. It will be noticed that the constant terms $+1, -1, +1, +1, -13, +47, +73$, are identical with a sequence of numbers computed by Airey and mentioned by Miller

0	+ 1								
1	− 1	+ 1							
2	+ 1	− 3	+ 1						
3	+ 1	+ 7	− 6	+ 1					
4	− 13	− 5	+ 25	− 10	+ 1				
5	+ 47	− 83	− 60	+ 65	− 15	+ 1			
6	+ 73	+ 637	− 203	− 280	+ 140	− 21	+ 2		
7	−2447	− 1425	+ 3710	+ 77	− 910	+ 266	− 28	+ 1	
8	+16811	− 22341	− 21347	+ 13146	+ 2667	−2394	+ 462	−36	+ 1

Table VII

r \ s	0	1	2	3	4	5	6	7	8
0	+ 1								
1	- 1	+ 1							
2	+ 1	- 1	+ 1						
3	+ 1	- 1	0	+ 1					
4	- 13	+ 13	- 7	+ 2	+ 1				
5	+ 47	- 47	+ 30	- 15	+ 5	+ 1			
6	+ 73	- 73	+ 13	+ 20	- 20	+ 9	+ 1		
7	- 2447	- 2447	-1260	+ 413	- 70	- 14	+ 14	+ 1	
8	+16811	- 16811	+9629	-4074	+1323	-294	+ 14	+ 20	+ 1

Table VIII

Numerical experiments indicate that the rate of convergence of the converging factor series is not greatly influenced by the value of a. This is illustrated in Table IX which gives the values of $|\beta_0(0{\cdot}25)|$ and $|\beta_4(0{\cdot}25)|$ when $\arg(z)=\pi/4$ and a=0, 1.5, and 3.0

a	$\lvert\beta_0(0{\cdot}25)\rvert$	$\lvert\beta_4(0{\cdot}25)\rvert$
0	1.41421	1376.56
1.5	1.41421	1403.36
3.0	1.41421	1378.71

Table IX

In contrast with this, the effect of $\arg(z)$ upon the rate of convergence of the converging factor series appears to be very great; the rate of convergence decreases markedly as $\arg(z)$ increases from 0 to $\pi/2$. This is illustrated in Table X which gives the values of $|\beta_0(0{\cdot}25)|$ and $|\beta_4(0{\cdot}25)|$ when $a=0$ and $\arg(z) = 0, \pi/8, \pi/4$ and $3\pi/8$.

$\arg(z)$	$\lvert\beta_0(0.25)\rvert$	$\lvert\beta_4(0.25)\rvert$
0	1.0	73.12109
$\pi/8$	1.08239	131.64265
$\pi/4$	1.41421	1376.55506
$3\pi/8$	2.61313	51129.210

Table X

The Singular Case

The numerical results produced by the preceding ALGOL programmes for the case in which the argument is pure imaginary may be illustrated by the following Tables which relate to the case $a=0,\ z=4.5i,\ n=11,\ k=0.25.$

Table XI gives the terms and partial sum of the asymptotic series

r	
0	+ 74.4748 3638
1	1.3791 6364
2	.1489 8373
3	.0303 4854
4	.0091 3266
5	.0036 4179
6	.0018 0965
7	.0010 7718
8	.0007 4721
9	.0005 9192
10	+ 0.0005 2725
$\sum_{r=0}^{10} u_r$	+ 76.0508 5994
11	+ 0.0005 2163

Table XI

Tables XII and XIII give the polynomial and factorial coefficients $p_{r,s}$ and $q_{r,s}$ respectively

s \ r	0	1	2	3
0	- 0.6666 6667	- 1.2629 6296	+ 12.0902 9982	- 113.7407 9955
1	+ 1.0	- 1.3333 3333	- 3.0518 5185	+ 30.6473 5449
2		+ 1.3333.3333	- 1.2851 8519	- 19.2007 0547
3		- 0.3333.3333	+ 2.2222 2222	- 0.8864 1975
4			- 0.6666 6667	+ 3.5296 2963
5			+ 0.0666 6667	- 1.2444 4444
6				+ 0.1777 7778
7				- 0.0095 2381

Table XII

s \ r	0	1	2	3
0	- 0.6666 6667	- 1.2629 6296	+ 12.0902 9982	- 113.7407 9955
1	+ 1.0	0	- 1.0	+ 2.1055 5556
2		- 0.6666 6667	+ 1.3814 8148	- 6.0451 4991
3		- 0.3333 3333	+ 0.8888 8889	- 0.8419 7531
4			+ 0.6666 6667	- 1.8037 0370
5			+ 0.0666 6667	- 1.2444 4444
6				- 0.2222 2222
				- 0.0095 2381

Table XIII

Table XIV gives the values of the coefficients $\beta_r(k)$ and the terms $\beta_r(k)\,2^{-r-1}\,x^{-2r}$, the numerical sum of the converging factor series, the product $u_n C_n$, and the modified sum $\sum_{r=0}^{n-1} u_r + u_n C_n$

r	$\beta_r(k)$	$\beta_r(k)\,2^{-r-1}x^{-2r}$
0	- 0.4166 6667	- 0.2083 3333
1	- 1.5181 7130	- 0.0187 4286
2	+ 11.2791 96	+ 0.0034 3826
3	- 107.2802 4	- 0.0008 0747
4	+ 1510.9878	+ 0.0002 8081
5	- 27825.923	- 0.0001 2769
	C_{11}	- 0.2242 9228
	$u_{11}C_{11}$	- 0.0001 1670
	$\sum_{r=0}^{10} u_r + u_{11}C_{11}$	+76.0507 4294

Table XIV

Table XV gives the modified sums which are to be derived by applying the ϵ-algorithm to the converging factor series, and using the members of the resulting even column ϵ-array as approximations to the converging factor.

m \ s	0	2	4	6
1	+ 76.0507 5127	+ 76.0507 4154		
2	76.0507 4149	76.0507 4328	+ 76.0507 4328	
3	76.0507 4329	76.0507 4286	76.0507 4286	+ 76.0507 4286
4	76.0507 4286	76.0507 4301	+ 76.0507 4301	
5	76.0507 4301	+ 76.0507 4294		
6	+ 76.0507 4294			

Table XV

When $a=0{\cdot}0$ and $z=4{\cdot}5i$, the modulus of expression (119) is 76.0507 4302.

It would appear that in the singular case the improvement effected by applying the ϵ-algorithm to the converging series is not so marked.

The effect of the parameter a upon the rate of convergence of the converging factor series is illustrated in Table XVI which gives the values of $|\beta_0(0.25)|$ and $|\beta_4(0.25)|$ when $a = 0, 1.5$, and 3.0.

a	$\|\beta_0(0.25)\|$	$\|\beta_4(0.25)\|$
0	0.4166 6667	107.2802 4017
1.5	0.4166 6667	5.0140 1211
3.0	0.4166 6667	151.9949 9718

Table XVI

The effect of non-zero μ appears to be rather strong.

Acknowledgements

A provisional study of the work contained in this paper was made when the author was a member of the Institut für angewandte Mathematik of the University of Mainz, using an ALGOL Computer for the Z-22 computer, constructed by Manfred Paul; The author is grateful to the Deutsche-forschungsgemeinschaft for a grant which enabled this study to be made. The numerical results of this paper have been produced on the X1 computer in Amsterdam using an ALGOL compiler constructed by E.W. Dijkstra and J.A. Zonneveld.

References

1 Miller J.C.P., A Method for the Determination of Converging Factors, Applied to the Asymptotic Expansions for the Parabolic Cylinder Functions, Proc. Camb. Phil. Soc., vol. 48, part 2, 1952, p. 243

2 Airey J.R. The Converging Factor in Asymptotic series and the Calculation of Bessel, Laguerre and other Functions, Phil. Mag., vol. 24, 1937, p. 24

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, Maths. of Comp., vol. 14, 1960, p. 147

5 Backus J.W. et al., Report on the Algorithmic Language ALGOL 60, Num. Math., vol. 2, 1960, p. 106

6 Wynn P., An Arsenal of ALGOL Procedures for Complex Arithmetic, Nord. Tid. för Inf. Behandlung, to appear

7 Wynn P., Singular Rules for Certain Non-Linear Algorithms, to appear.