MATHEMATICS

CONVERGING FACTORS FOR THE WEBER PARABOLIC CYLINDER FUNCTIONS OF COMPLEX ARGUMENT, Ib

BY

P. WYNN

(Communicated by Prof. A. van Wijngaarden at the meeting of June 29, 1963)

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. $\mathrm{Re}(z) \equiv z\,[0]$, $\mathrm{Re}\,(p_{r,s}) \equiv p[R, s, 0]$) 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 *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 [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, $Rminus1$, $Rminus2$ 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

$$(118) \qquad \begin{cases} |\beta_{r+1}(k)2^{-r-2}x^{-2r-2}| > |\beta_r(k)2^{-r-1}x^{-2r}| \text{ and} \\ |\beta_{r+2}(k)2^{-r-3}x^{-2r-4}| > |\beta_{r+1}(k)2^{-r-2}x^{-2r-2}| \end{cases}$$

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

As the terms $\beta_s(k)2^{-r-1}x^{-2r}$ 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 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 separately) 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 separately) 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 separate 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 ;
```

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

$$z = 3.5e^{i\pi/4}, \ a = 0.0 \ \text{(i.e. } n = 7, \, k = 0.25\text{)}.$$

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

TABLE I

r	$\mathrm{Re}(u_r)$	$\mathrm{Im}(u_r)$	$\mid u_r \mid$
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

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

TABLE II

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

TABLE III

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

Table IV 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 factors series, the product u_nC_n and the modified sum $\sum_{r=0}^{n-1} u_r + u_nC_n$

TABLE IV

r	$\mathrm{Re}\,\{\beta_r(k)\}$	$\mathrm{Im}\,\{\beta_r(k)\}$	$\lvert\beta_r(k)\rvert$	$\mathrm{Re}\left\{\dfrac{\beta_r(k)}{2^{r+1}x^{2r}}\right\}$	$\mathrm{Im}\left\{\dfrac{\beta_r(k)}{2^{r+1}x^{2r}}\right\}$	$\left\lvert\dfrac{\beta_r(k)}{2^{r+1}x^{2r}}\right\rvert$
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_nC_n	− 0.000078097	+ 0.000153817	0.000172508
			$\sum_{r=0}^{n-1} u_r + u_nC_n$	− 0.510808287	+ 0.149281284	0.532174791

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 ai approximations to the converging factor

TABLE V

m	0	2	4
1	− 0.51081 3995	− 0.51080 6941	
2	− 0.51080 6332	− 0.51080 8523	− 0.51080 8194
3	− 0.51080 8912	− 0.51080 8171	− 0.51080 8220
4	− 0.51080 7966	− 0.51080 8223	
5	− 0.51080 8287		

TABLE VI

m	0	2	4
1	+ 0.14929 1718	+ 0.14928 1499	
2	− 0.14928 0841	+ 0.14928 1472	+ 0.14928 1461
3	+ 0.14928 1250	+ 0.14928 1442	+ 0.14928 1440
4	+ 0.15928 1665	+ 0.14928 1438	
5	+ 0.14928 1284		

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

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

...ogramme which computes the function $S_1(a; z)$ by means ...19) is given in [6]. When $a=0$ and $z=3.5e^{i\pi/4}$, the correct ...z) computed by means of formula (119) is

$$-0.51080\ 8214 + i0.14928\ 1449.$$

...and VIII respectively are given the polynomial and factorial ...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

TABLE VII

0	+1								
1	−1	+1							
2	+1	−3	+1						
3	+1	+7	−6	+1					
4	−13	−5	+25	−10	+1				
5	+47	−93	−60	+65	−15	+1			
6	+73	+637	−203	−280	+140	−21	+1		
7	−2447	−1425	+3710	+77	−910	+266	−28	+1	
8	+16811	−22341	−21347	+13146	+2667	−2394	+462	−36	+1

TABLE VIII

r^s	0	1	2	3	4	5	7	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

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 $|\beta_0(0.25)|$ and $|\beta_4(0.25)|$ when $\arg(z)=\pi/4$ and $a=0$, 1.5, and 3.0

TABLE IX

a	$\lvert\beta_0(0.25)\rvert$	$\lvert\beta_4(0.25)\rvert$
0	1.41421	1376.56
1.5	1.41421	1403.36
3.0	1.41421	1378.71

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 $|\beta_0(0.25)|$ and $|\beta_4(0.25)|$ when $a=0$ and arg $(z)=0$, $\pi/8$, $\pi/4$ and $3\pi/8$.

TABLE X

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

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

TABLE XI

r	
0	+ 74.4748 3638
1	+ 1.3791 6364
2	+ 0.1489 8373
3	+ 0.0303 4854
4	+ 0.0091 3266
5	+ 0.0036 4179
6	+ 0.0018 0965
7	+ 0.0010 7718
8	+ 0.0007 4721
9	+ 0.0005 9192
10	+ 0.0005 2725
$\sum_{r=0}^{10} u_r$	+ 76.0508 5994
11	+ 0.0005 2163

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

TABLE XII

s \ r	0	1	2	3
0	− 0.6666 667	− 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 XIII

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 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_nC_n, and the modified sum $\sum_{r=0}^{n-1} u_r + u_nC_n$

TABLE XIV

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 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.

TABLE XV

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			

When $a = 0.0$ and $z = 4.5i$, 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

TABLE XVI

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

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

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., 48, part 2, 243 (1952).

AIREY, J. R., The Converging Factor in Asymptotic series and the Calculation of Bessel, Laguerre and other Functions, Phil. Mag., 24, 24 (1937).

WYNN, P., On a Device for Computing the $e_m(S_n)$ Transformation, M.T.A.C., 10, 91 (1956).

———, The Rational Approximation of Functions which are Formally Defined by a Power Series Expansion, Maths. of Comp., 14, 147 (1960).

BACKUS, J. W. *et al.*, Report on the Algorithmic Language ALGOL 60, Num. Math., vol. 2, 106 (1960).

WYNN, P., An Arsenal of ALGOL Procedures for Complex Arithmetic, Nord. Tid. för Inf. Behandlung, vol. 2, 232 (1962).

———, Singular Rules for Certain Non-Linear Algorithms, to appear.