



ELSEVIER

Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Computer Physics Communications 157 (2004) 32–38

Computer Physics
Communications

www.elsevier.com/locate/cpc

f1: a code to compute Appell's F_1 hypergeometric function [☆]

F.D. Colavecchia ^{a,b,*,1}, G. Gasaneo ^c

^a Department of Physics and Astronomy, The University of Oklahoma, Norman, OK 73019, USA

^b Theoretical Division, Group T-12, MS B268, Los Alamos National Laboratory, Los Alamos, NM 87544, USA

^c Departamento de Física and Consejo Nacional de Investigaciones Científicas y Técnicas, Universidad Nacional del Sur, 8000 Bahía Blanca, Buenos Aires, Argentina

Received 11 July 2003

Abstract

In this work we present the FORTRAN code to compute the hypergeometric function $F_1(\alpha, \beta_1, \beta_2, \gamma, x, y)$ of Appell. The program can compute the F_1 function for real values of the variables $\{x, y\}$, and complex values of the parameters $\{\alpha, \beta_1, \beta_2, \gamma\}$. The code uses different strategies to calculate the function according to the ideas outlined in [F.D. Colavecchia et al., Comput. Phys. Comm. 138 (1) (2001) 29].

Program summary

Title of the program: f1

Catalogue identifier: ADSJ

Program summary URL: <http://cpc.cs.qub.ac.uk/summaries/ADSJ>

Program obtainable from: CPC Program Library, Queen's University of Belfast, N. Ireland

Licensing provisions: none

Computers: PC compatibles, SGI Origin2*

Operating system under which the program has been tested: Linux, IRIX

Programming language used: Fortran 90

Memory required to execute with typical data: 4 kbytes

No. of bits in a word: 32

No. of bytes in distributed program, including test data, etc.: 52 325

Distribution format: tar gzip file

External subprograms used: Numerical Recipes hypgeo [W.H. Press et al., Numerical Recipes in Fortran 77, Cambridge Univ. Press, 1996] or chyp routine of R.C. Forrey [J. Comput. Phys. 137 (1997) 79], rkf45 [L.F. Shampine and H.H. Watts, Rep. SAND76-0585, 1976].

Keywords: Numerical methods, special functions, hypergeometric functions, Appell functions, Gauss function

[☆] This paper and its associated computer program are available via the Computer Physics Communications homepage on ScienceDirect (<http://www.sciencedirect.com/science/journal/00104655>).

* Corresponding author.

E-mail address: flavioc@lanl.gov (F.D. Colavecchia).

¹ On leave from División Colisiones Atómicas, Centro Atómico Bariloche and Consejo Nacional de Investigaciones Científicas y Técnicas, 8400 S.C. de Bariloche, Argentina.

Nature of the physical problem: Computing the Appell F_1 function is relevant in atomic collisions and elementary particle physics. It is usually the result of multidimensional integrals involving Coulomb continuum states.

Method of solution: The F_1 function has a convergent-series definition for $|x| < 1$ and $|y| < 1$, and several analytic continuations for other regions of the variable space. The code tests the values of the variables and selects one of the precedent cases. In the convergence region the program uses the series definition near the origin of coordinates, and a numerical integration of the third-order differential parametric equation for the F_1 function. Also detects several special cases according to the values of the parameters.

Restrictions on the complexity of the problem: The code is restricted to real values of the variables $\{x, y\}$. Also, there are some parameter domains that are not covered. These usually imply differences between integer parameters that lead to negative integer arguments of Gamma functions.

Typical running time: Depends basically on the variables. The computation of Table 4 of [F.D. Colavecchia et al., Comput. Phys. Comm. 138 (1) (2001) 29] (64 functions) requires approximately 0.33 s in a Athlon 900 MHz processor.

© 2003 Elsevier B.V. All rights reserved.

PACS: 02.60.Gf; 02.30.Gp; 34.50.Fa

Keywords: Numerical methods; Special functions; Hypergeometric functions; Appell functions; Gauss function

1. Introduction

In this work we present the Fortran code to compute the $F_1(\alpha, \beta_1, \beta_2, \gamma, x, y)$ hypergeometric function of Appell for real values of the variables x and y , and complex parameters $\{\alpha, \beta_1, \beta_2, \gamma\}$. The procedure was sketched in [1], and we will here devote our attention to the code.

The F_1 functions are usually defined in terms of the double series [5]

$$F_1(\alpha, \beta_1, \beta_2, \gamma, x, y) = \sum_{m,n} \frac{(\alpha)_{m+n} (\beta_1)_m (\beta_2)_n}{(\gamma)_{m+n} m! n!} x^m y^n. \quad (1)$$

This series is convergent when:

$$|x| < 1 \quad \text{and} \quad |y| < 1. \quad (2)$$

From a numerical point of view the double summation poses different problems to analyse its convergence. Besides, the numerical region of convergence is smaller than the mathematical one, as usually happens with special functions defined by series summations [2]. Of course, the series (1) cannot be used outside the convergence region (2), and the function has to be extended to the complete $\{x, y\}$ plane by proper analytic continuations.

In our previous work [1], hereafter referenced as paper 1, we presented the basis of a numerical scheme to compute the hypergeometric function F_1 for all values of the real variables $\{x, y\}$. To this end, we

divided the real plane of variables into the convergent and into the divergent region, according to (2). To compute the function outside the convergence region, we use eight different analytic continuations, given by Eqs. (21) through (30) of [1]. These analytic continuations are written in terms of the Appell F_1 and the G_2 function [7,8] given by

$$G_2(\beta_1, \beta_2, \gamma_1, \gamma_2, x, y) = (1+x)^{-\beta_1} (1+y)^{-\beta_2} \times F_2\left(1-\gamma_1-\gamma_2, \beta_1, \beta_2, 1-\gamma_1, 1-\gamma_2, \frac{x}{x+1}, \frac{y}{y+1}\right), \quad (3)$$

which is an independent solution of the hypergeometric system of equations for F_1 . The aim of the analytic continuations is to map the point $\mathbf{P} = (x, y)$ from outside to inside the convergence region. To determine which analytic continuation we need to use, we transform the point (x, y) to the point (u, v) , according to Table 1 of [1]. These transformations represent the mapping of each analytic continuation from outside to inside the convergence region. Then we measure the effective distance in each possible region as

$$t_{\max} = \sqrt{w^2 + u^2} \quad (4)$$

and choose the analytic continuation that minimizes t_{\max} .

When the point is in the convergence region, whether originally or from an analytic continuation,

Table 1
Files in the distribution

check_parms.f90	Checks the parameters $\alpha, \beta_1, \beta_2, \gamma$.
driver.f90	Main program.
flbnl.f90	F_1 function, Eq. (5).
flconv.f90	F_1 function in the convergence region.
flv3.f90	F_1 function.
fl.data	Input file for test 1.
f2lr.f	Gauss function wrapper.
f2.f90	F_2 and G_2 functions.
f2gam.f90	Some special cases of F_2 function.
gamma.f90	Γ function.
get_transformation2.f90	Gets the analytical continuation to be used.
hypdrvfl.f90	Computes the functions and derivatives in Eq. (6).
hypgeofl.f90	Integration of Eq. (6).
hypstartfl.f90	Compute initial values for hypgeofl.
isneg.f90	Checks whether an integer is negative or not.
writefl.f90	F_1 function output routines.
table2new.dat	Output of test 2, see Fig. 1.
table3.dat	Output of test 3, see Fig. 2.
table4.dat	Output of test 4, see Fig. 3.

we check its distance from the origin. For distances smaller than $t_0 = 0.5$ we use the single-index series [10]

$$\begin{aligned}
 &F_1(\alpha, \beta_1, \beta_2, \gamma, x, y) \\
 &= \sum_r \frac{(\alpha)_r (\beta_1)_r (\beta_2)_r (\gamma - \alpha)_r}{(\gamma + r - 1)_r (\gamma)_{2r} r!} (xy)^r \\
 &\quad \times {}_2F_1(\alpha + r, \beta_1 + r, \gamma + 2r, x) \\
 &\quad \times {}_2F_1(\alpha + r, \beta_2 + r, \gamma + 2r, y), \quad (5)
 \end{aligned}$$

while for larger distances we solve the third-order differential equation for the parametric function $z(xt, yt)$ [9]

$$\begin{aligned}
 &z\delta(-2 + \gamma + \delta)(-1 + \gamma + \delta) \\
 &\quad - z(\alpha + \delta)(-1 + \gamma + \delta)t(x(\beta_1 + \delta) + y(\delta + \beta_2)) \\
 &\quad + t^2xyz(\alpha + \delta)(1 + \alpha + \delta)(\beta_1 + \delta + \beta_2) = 0, \quad (6)
 \end{aligned}$$

where the differential operator is defined as $\delta = \frac{d}{dt}$. In this way we can compute the F_1 function confidently in the whole of the convergence region (2). This means that the mapping of the analytic continuations is complete and covers the whole two-dimensional real space of variables x and y .

2. The code

Most of the code was written with a minimal use of features of Fortran 90, so can be easily transformed to Fortran 77. Table 1 summarizes the files contained in the distribution. Note again that the computation of the Gauss hypergeometric function [2,3] and the integration routines [4] are not included in the package, but can be downloaded from the Web. The entry point for the calculation is the complex function `fl` and receives the arguments $\{\alpha, \beta_1, \beta_2, \gamma, x, y\}$. The routine first checks if the function reduces to one of the simple cases or polynomial transformation and it calls `get_transformations`. This routine performs a sequential minimization of the t_{\max} , Eq. (4) and returns the integer `flag`. This value is set to zero if we are going to use the single series (5) or to one if the integration of the differential equation is needed. Any other value of `flag` correspond to an analytic continuation. To ease the tracing of the code we set the value of `flag` to the equation numbers for each analytic continuation of paper 1. The logical variable `impossible` is set true if the code is actually able to perform the computation; it also helps in the debugging process.

Once the `flag` value is set, the function `fl` executes each of the different possibilities. The calculation in the convergence region is performed by

```

-----
                        Select the test
-----
Test 1  : read parameters and variables
          from fl.data
Test 2-4: Generate Tables 2-4 of
          Comp. Phys. Comm. 138 (2001) 29
-----
2
Table 2 of paper 1, corrected
-0.95 -0.95 0.0965781451527413 0.0965781451527413 2.4E-17
-0.95 -0.57 0.1336845747524817 0.1336845746751630 5.8E-10
-0.95 -0.19 0.2025864225056421 0.2025864224726571 1.6E-11
-0.95 0.19 0.3607474325479348 0.3607474326056157 1.6E-10
-0.95 0.57 0.9326701852895512 0.9326701853591102 7.5E-11
-0.95 0.95 23.5220825750274300 23.5220826035481400 1.2E-09
-0.53 -0.95 0.1568791477977473 0.1568791477394587 3.7E-10
-0.53 -0.57 0.2171539131856024 0.2171539131113278 3.4E-10
-0.53 -0.19 0.3290763687047047 0.3290763686839584 6.3E-11
-0.53 0.19 0.5859891975056243 0.5859891975235394 3.1E-11
-0.53 0.57 1.5150063567439340 1.5150063564560710 1.9E-10
-0.53 0.95 38.2086886717265800 38.2086885813113600 2.4E-09
-0.11 -0.95 0.2980589214080558 0.2980589213077664 3.4E-10
-0.11 -0.57 0.4125765727105700 0.4125765726826616 6.8E-11
-0.11 -0.19 0.6252210627245672 0.6252210627808445 9.0E-11
-0.11 0.19 1.1133366710500700 1.1133366711166740 6.0E-11
-0.11 0.57 2.8784014114113450 2.8784014120834480 2.3E-10
-0.11 0.95 72.5937173118998400 72.5937173118998400 2.9E-09
0.31 -0.95 0.7713471895595728 0.7713471895469418 1.6E-11
0.31 -0.57 1.0677076142869480 1.0677076143715760 7.9E-11
0.31 -0.19 1.6180106519768290 1.6180106520736790 6.0E-11
0.31 0.19 2.8812058642488370 2.8812058653284050 3.7E-10
0.31 0.57 7.4490199081276280 7.4490199114220050 4.4E-10
0.31 0.95 187.8654037683292000 187.8654045368447000 4.1E-09
0.73 -0.95 5.0375637491459680 5.0375637440781740 1.0E-09
0.73 -0.57 6.9730534420866100 6.9730534321304130 1.4E-09
0.73 -0.19 10.5670078566551200 10.5670078388515500 1.7E-09
0.73 0.19 18.8167642518767800 18.8167642315892100 1.1E-09
0.73 0.57 48.6485372835603900 48.6485374462004900 3.3E-09
0.73 0.95 1226.9234320172550000 1226.9234444443310000 1.0E-08

```

Fig. 1. Output of `fl` for test number 2, Eq. (8). Columns are: the variables x, y ; the computed functions $|F_1|$ and the reduction (8); and the relative error of the calculation. See text for details.

`flconv`. This function decides whether a summation of the series is needed (through `flbnl`) or if it is required to integrate the differential equation, using `hypgeofl`. The summation of the series checks the relative convergence against a predefined value `eps` and, even though it has set a maximum of 300 terms, we have found that only a few terms are needed to get convergence in our tests. The integration of (6) is performed by a Fehlberg four–fifth order Runge–Kutta method, provided by the routine `rkf45` [4], while `hypdrvfl` computes the function and the derivatives needed by the integrator.

The calculation can be traced by setting the logical flag `debug` to `true`. A complete tracing tells which analytic continuation has been selected, and the different intermediate values of the hypergeometric functions involved. The computation of the Gauss function in the code is performed with the `hypgeo` function

of Numerical Recipes [2] through a wrapper function, called `f21`. This provides a convenient way of checking different codes for computing the ${}_2F_1$ function. We have tested the `chyp` function of Forrey [3]. The latest one is faster and gives accurate results. But, we found that it fails to converge when the variable of the Gauss function is near 1. This is the case for some of the calculation of the series with the `flbnl` routine in Table 3 of [1]. Note however that `flconv` would not have this problem, because it would choose to integrate the partial differential equation in this case. Moreover, the case $\beta_1 + \beta_2 = \gamma$ is exactly treated in `fl`, reducing the function to an analytical expression.

Besides the main stream of the code, there are some auxiliary routines that are used in the computation of the Appell function. We include a `gamma` and `gammar` routines that calculate the Gamma $\Gamma(z)$ and the reciprocal Gamma $1/\Gamma(z)$ functions. The

```

-----
Select the test
-----
Test 1 : read parameters and variables
        from fl.data
Test 2-4: Generate Tables 2-4 of
        Comp. Phys. Comm. 138 (2001) 29
-----
3
Table 3 of paper 1
-0.95 -0.95      0.51282051      0.51282051      0.51282051  0.0E+00 0.0E+00
-0.95 -0.57      0.55535479      0.55535479      0.55535479  1.2E-14 2.9E-12
-0.95 -0.19      0.60791691      0.60791691      0.60791691  8.5E-15 3.1E-12
-0.95  0.19      0.67545323      0.67545323      0.67545323  2.1E-14 2.4E-12
-0.95  0.57      0.76755518      0.76755518      0.76755518  1.8E-14 1.8E-12
-0.95  0.95      0.90428984      0.90428984      0.90428984  5.5E-14 4.1E-12
-0.57 -0.95      0.58342152      0.58342152      0.58342152  6.4E-15 6.4E-12
-0.57 -0.57      0.63694268      0.63694268      0.63694268  0.0E+00 0.0E+00
-0.57 -0.19      0.70425421      0.70425421      0.70425421  1.6E-15 8.3E-13
-0.57  0.19      0.79279973      0.79279973      0.79279973  5.7E-15 5.0E-13
-0.57  0.57      0.91786341      0.91786341      0.91786341  2.3E-14 6.0E-13
-0.57  0.95      1.11675226      1.11675226      1.11675226  4.7E-14 3.7E-12
-0.19 -0.95      0.67982327      0.67982327      0.67982327  4.3E-15 3.8E-12
-0.19 -0.57      0.75004065      0.75004065      0.75004065  8.8E-16 3.6E-13
-0.19 -0.19      0.84033613      0.84033613      0.84033613  0.0E+00 0.0E+00
-0.19  0.19      0.96273876      0.96273876      0.96273876  1.0E-15 4.1E-15
-0.19  0.57      1.14366890      1.14366890      1.14366890  6.2E-15 9.4E-13
-0.19  0.95      1.45953734      1.45953734      1.45953734  7.2E-15 1.8E-12
 0.19 -0.95      0.82196313      0.82196313      0.82196313  5.5E-15 8.4E-13
 0.19 -0.57      0.92026724      0.92026724      0.92026724  2.4E-16 1.6E-13
 0.19 -0.19      1.05054372      1.05054372      1.05054372  7.2E-19 1.8E-13
 0.19  0.19      1.23456790      1.23456790      1.23456790  0.0E+00 0.0E+00
 0.19  0.57      1.52444536      1.52444536      1.52444536  7.6E-15 1.4E-11
 0.19  0.95      2.10562436      2.10562436      2.10562436  6.8E-15 3.2E-11
 0.57 -0.95      1.06293284      1.06293284      1.06293284  7.0E-15 4.8E-12
 0.57 -0.57      1.21816169      1.21816169      1.21816169  3.5E-15 8.5E-12
 0.57 -0.19      1.43365813      1.43365813      1.43365813  5.9E-15 1.0E-11
 0.57  0.19      1.75843151      1.75843151      1.75843151  1.3E-15 2.0E-11
 0.57  0.57      2.32558140      2.32558140      2.32558140  0.0E+00 0.0E+00
 0.57  0.95      3.77963837      3.77963837      3.77963837  2.3E-14 7.0E-12
 0.95 -0.95      1.66982335      1.66982335      1.66982335  4.0E-14 3.3E-11
 0.95 -0.57      2.02668669      2.02668669      2.02668669  1.2E-14 6.4E-12
 0.95 -0.19      2.58135132      2.58135132      2.58135132  1.3E-14 1.4E-10
 0.95  0.19      3.56764083      3.56764083      3.56764083  2.0E-14 1.6E-10
 0.95  0.57      5.85737543      5.85737543      5.85737543  1.8E-16 4.9E-11
 0.95  0.95      20.00000000      20.00000000      20.00000000  0.0E+00 0.0E+00

```

Fig. 2. Output of `fl` for test number 3, Eq. (9). Columns are: the variables x , y ; the $|F_1|$ function computed by series summation, the $|F_1|$ function computed by the differential equation integration, and the exact value (9); and the relative errors of each calculation respect to the exact result.

latter takes care of the possible poles of the Gamma function in such a way that over/underflows and the appearances of numerical infinities in the computation are avoided. The G_2 function is obtained in terms of the hypergeometric F_2 . The routine `fl2` uses the single index sum [10]

$$\begin{aligned}
 F_2(a, \beta_1, \beta_2, \gamma_1, \gamma_2, x, y) &= \sum_r \frac{(a)_r (\beta_1)_r (\beta_2)_r}{(\gamma_1)_r (\gamma_2)_r r!} (xy)^r \\
 &\times {}_2F_1(a+r, \beta_1+r, \gamma_1+2r, x) \\
 &\times {}_2F_1(a+r, \beta_2+r, \gamma_2+2r, y)
 \end{aligned} \quad (7)$$

to compute F_2 . Although this series has a small region of convergence, it is only used after an analytic continuation of the F_1 function. This guarantees the numerical convergence of (7).

The main program contained in `driver.f90` allows the user to select from four different tests. Test number one computes only one F_1 function, reading the input variables and parameters from the file `fl.data`. The next three tests correspond to each of the three tables of [1]. Test two checks the computation of the F_1 function in the convergence region for $\alpha = \gamma$. In this case the hypergeometric reduces to:

$$F_1(\alpha, \beta_1, \beta_2, \alpha, x, y) = (1-x)^{-\beta_1} (1-y)^{-\beta_2}. \quad (8)$$

```

-----
                Select the test
-----
Test 1  : read parameters and variables
          from f1.data
Test 2-4: Generate Tables 2-4 of
          Comp. Phys. Comm. 138 (2001) 29
-----
4
Table 4 of paper 1

```

-3.50	-3.50	2.12132034	2.12132034	2.12132034	0.0E+00	0.0E+00
-3.50	-2.50	2.04040981	2.04040981	2.04040992	5.2E-08	5.2E-08
-3.50	-1.50	1.95401179	1.95401179	1.95401190	5.2E-08	5.2E-08
-3.50	-0.50	1.86035462	1.86035559	1.86035559	5.8E-07	5.2E-08
-3.50	0.50	1.75598179	1.75598184	1.75598193	7.8E-08	5.2E-08
-3.50	1.50	1.62917838	1.62917839	1.62917847	5.9E-08	5.2E-08
-3.50	2.50	1.48548531	1.48548533	1.48548541	6.3E-08	5.2E-08
-3.50	3.50	1.35499406	1.35499412	1.35499419	9.5E-08	5.2E-08
-2.50	-3.50	1.95697834	1.95697834	1.95697845	5.2E-08	5.2E-08
-2.50	-2.50	1.87082869	1.87082869	1.87082869	0.0E+00	0.0E+00
-2.50	-1.50	1.77824921	1.77824921	1.77824930	5.2E-08	5.2E-08
-2.50	-0.50	1.67700137	1.67700355	1.67700363	1.3E-06	5.2E-08
-2.50	0.50	1.56257050	1.56257057	1.56257065	9.8E-08	5.2E-08
-2.50	1.50	1.41872399	1.41872400	1.41872407	5.9E-08	5.2E-08
-2.50	2.50	1.25857059	1.25857062	1.25857068	7.1E-08	5.2E-08
-2.50	3.50	1.12513362	1.12513374	1.12513380	1.6E-07	5.2E-08
-1.50	-3.50	1.77471785	1.77471785	1.77471794	5.2E-08	5.2E-08
-1.50	-2.50	1.68182193	1.68182193	1.68182202	5.2E-08	5.2E-08
-1.50	-1.50	1.58113883	1.58113883	1.58113883	0.0E+00	0.0E+00
-1.50	-0.50	1.46969385	1.46969385	1.46969392	5.2E-08	5.2E-08
-1.50	0.50	1.34118577	1.34118592	1.34118599	1.6E-07	5.2E-08
-1.50	1.50	1.17126079	1.17126080	1.17126086	5.9E-08	5.2E-08
-1.50	2.50	0.99247161	0.99247166	0.99247171	1.1E-07	5.2E-08
-1.50	3.50	0.86922669	0.86922699	0.86922703	4.0E-07	5.2E-08
-0.50	-3.50	1.56577390	1.56578852	1.56578860	9.4E-06	5.2E-08
-0.50	-2.50	1.46350099	1.46351853	1.46351861	1.2E-05	5.2E-08
-0.50	-1.50	1.35127918	1.35127918	1.35127925	5.2E-08	5.2E-08
-0.50	-0.50	1.22474487	1.22474487	1.22474487	0.0E+00	0.0E+00
-0.50	0.50	1.07407680	1.07407680	1.07407686	5.2E-08	5.2E-08
-0.50	1.50	0.85764535	0.85764536	0.85764540	6.3E-08	5.2E-08
-0.50	2.50	0.67330010	0.67330033	0.67330036	4.0E-07	5.2E-08
-0.50	3.50	0.61598961	0.61599062	0.61599065	1.7E-06	5.2E-08
0.50	-3.50	1.30813932	1.30814755	1.30814761	6.3E-06	5.2E-08
0.50	-2.50	1.19046476	1.19047256	1.19047262	6.6E-06	5.2E-08
0.50	-1.50	1.05833564	1.05834321	1.05834326	7.2E-06	5.2E-08
0.50	-0.50	0.90400842	0.90400842	0.90400847	5.2E-08	5.2E-08
0.50	0.50	0.70710678	0.70710678	0.70710678	0.0E+00	0.0E+00
0.50	1.50	0.38872999	0.38873013	0.38873015	4.0E-07	5.2E-08
0.50	2.50	0.42426224	0.42426407	0.42426409	4.4E-06	5.2E-08
0.50	3.50	0.60368908	0.60369234	0.60369237	5.5E-06	5.2E-08
1.50	-3.50	0.92086392	0.92086915	0.92086920	5.7E-06	5.2E-08
1.50	-2.50	0.77207798	0.77208232	0.77208236	5.7E-06	5.2E-08
1.50	-1.50	0.60368908	0.60369234	0.60369237	5.5E-06	5.2E-08
1.50	-0.50	0.42426219	0.42426407	0.42426409	4.5E-06	5.2E-08
1.50	0.50	0.38872999	0.38873013	0.38873015	4.0E-07	5.2E-08
1.50	1.50	0.70710678	0.70710678	0.70710678	0.0E+00	0.0E+00
1.50	2.50	0.90399998	0.90400842	0.90400847	9.4E-06	5.2E-08
1.50	3.50	1.05833564	1.05834321	1.05834326	7.2E-06	5.2E-08
2.50	-3.50	0.73484376	0.73484692	0.73484696	4.4E-06	5.2E-08
2.50	-2.50	0.65183291	0.65183502	0.65183505	3.3E-06	5.2E-08
2.50	-1.50	0.61598958	0.61599062	0.61599065	1.7E-06	5.2E-08
2.50	-0.50	0.67330009	0.67330033	0.67330036	4.0E-07	5.2E-08
2.50	0.50	0.85764535	0.85764536	0.85764540	6.3E-08	5.2E-08
2.50	1.50	1.07407624	1.07407680	1.07407686	5.8E-07	5.2E-08
2.50	2.50	1.22474487	1.22474487	1.22474487	0.0E+00	0.0E+00
2.50	3.50	1.35127918	1.35127918	1.35127925	5.2E-08	5.2E-08
3.50	-3.50	0.79693132	0.79693290	0.79693294	2.0E-06	5.2E-08
3.50	-2.50	0.80691372	0.80691456	0.80691460	1.1E-06	5.2E-08
3.50	-1.50	0.86922668	0.86922699	0.86922703	4.0E-07	5.2E-08
3.50	-0.50	0.99247161	0.99247166	0.99247171	1.1E-07	5.2E-08
3.50	0.50	1.17126079	1.17126080	1.17126086	5.9E-08	5.2E-08
3.50	1.50	1.34118577	1.34118592	1.34118599	1.6E-07	5.2E-08
3.50	2.50	1.46969385	1.46969385	1.46969392	5.2E-08	5.2E-08
3.50	3.50	1.58113883	1.58113883	1.58113883	0.0E+00	0.0E+00

Fig. 3. Output of f1 for test number 4, Eq. (10). Columns are: the variables x, y ; the $|F_1|$ function computed by the code, the $|_2F_1|$ function, and the exact value (10). The last two columns show the relative errors of each calculation respect to the exact result. The same relative error in the last column shows that the $|_2F_1|$ code is actually using the reduction (10).

We also set $\alpha = \gamma = 1$, $\beta_1 = 2 + i$ and $\beta_2 = \frac{3}{2} - \frac{i}{2}$. The output of the run is in Fig. 1. Test three compares the direct summation and the integration routine with the exact reduction of F_1 for $\gamma = \beta_1 + \beta_2$:

$$\begin{aligned} F_1(\alpha, \beta_1, \beta_2, \beta_1 + \beta_2, x, y) \\ = (1 - y)^{-\alpha} \\ \times {}_2F_1\left(\alpha, \beta_1, \beta_1 + \beta_2, (y - x)/(y - 1)\right). \end{aligned} \quad (9)$$

In this case, we set $\alpha = 1$, $\beta_1 = 3 + i$, $\beta_2 = 2 - \frac{i}{2}$ and $\gamma = \beta_1 + \beta_2 = 5 + \frac{i}{2}$. The output of the code is shown in Fig. 2. Finally, test four computes a particular case of (9). When $\alpha = -\frac{1}{2}$, $\beta_1 = 2$, $\beta_2 = 1$ and $\gamma = \beta_1 + \beta_2$, the Gauss function of (9) further reduces to

$$\begin{aligned} {}_2F_1\left(-\frac{1}{2}, 2, 3, z\right) \\ = \frac{4}{15}(1 - y)^{-\alpha} [2 - (2 + 3z)(1 - z)^{3/2}]. \end{aligned} \quad (10)$$

The Appell function is computed in a grid $-3.5 \leq \{x, y\} \leq 3.5$ with unit stepsize in each variable, testing all the analytical continuations included in the code (see Fig. 3). We take this opportunity to note that Table 2 of [1] is not correct for positive values of x : the values of the F_1 and the analytic expression $(1 - x)^{-\beta_1}(1 - y)^{-\beta_2}$ were computed for different x, y pairs that those printed in that table.

3. Discussion and outlook

The `f1` code presented was thoroughly used in the calculation of differential and total ionization cross section in ion–atom collisions, using a correlated wave function proposed a few years ago [6]. The Appell F_1 function appears in these calculations as an analytical result of a tridimensional integration of products of Coulomb functions. The code has proven to be more efficient and has given more accurate results in these calculations than the direct numerical integration.

We tried to introduce in the code, most of the special cases that arise for particular sets of variables

and/or parameters. However, the number of possible ways to reduce the F_1 function is huge, and there is a chance that the code could not give the correct result for a particular set of parameters $\{\alpha, \beta_1, \beta_2, \gamma\}$. We have been careful to add comments and structure the code in such a way that these exceptional cases can be introduced easily in the routines.

We are currently working on the extension of these routines to complex variables. This requires the introduction of other analytic continuations and improved mapping techniques to the convergence region.

Acknowledgements

FDC thanks Prof. P. Abbott and Dr. E. Agol for testing the code and for fruitful discussions. This work was partially supported by ANCYT grant PICT99 03-06249. The work at LANL was performed under the auspices of the U.S. Department of Energy under contract W-7405-ENG-36.

References

- [1] F.D. Colavecchia, G. Gasaneo, J.E. Miraglia, *Comput. Phys. Comm.* 138 (1) (2001) 29.
- [2] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, *Numerical Recipes in Fortran 77*, Cambridge Univ. Press, New York, 1996, Chapters 5 and 6.
- [3] R.C. Forrey, *J. Comput. Phys.* 137 (1997) 79; The code can be found at <http://cfa-www.harvard.edu/~rcf/codes/codes.html>.
- [4] L.F. Shampine, H.A. Watts, Rep. SAND76-0585, Sandia National Laboratories, Albuquerque, NM, 1976; The routine `rkf45` can be downloaded from <http://www.netlib.org>.
- [5] P. Appell, J. Kampé de Fériet, *Fonctions Hypergéométriques et Hypersphériques; Polynômes d'Hermite*, Gauthier-Villars, Paris, 1926.
- [6] F.D. Colavecchia, G. Gasaneo, C.R. Garibotti, *J. Phys. B: At. Mol. Opt. Phys.* 33 (2000) L467.
- [7] A. Erdelyi, *Acta Math.* 83 (1950) 131.
- [8] P.O.M. Olsson, *J. Math. Phys.* 5 (1964) 420.
- [9] J.L. Burchinal, *Quart. J. Math. (Oxford)* 13 (1942) 90.
- [10] J.L. Burchinal, T.W. Chaundy, *Quart. J. Math. (Oxford)* 11 (1940) 248.