Programa Computacional para Cálculo de Funções Especiais

(Computational program for special functions calculation)

Nilo Makiuchi
Departamento de Física, Universidade de Brasília
70910-900 Brasília, DF

Trabalho recebido em 23 de julho de 1994

Abstract

FORTRAN program developed for a Mathematical-Physics undergraduate course level is presented. This program can be used to obtain high accuracy values of special functions with computational time relatively small. Students who now how to program can be use this article and develop their own program to calculate special functions or modify our program to improve their knowledge about special functions properties. The programs presented are based on definitions, recurrence relations or mathematical relations as simple as possible, associated to the functions, to be useful as an auxiliary educational tool. Programming procedure to avoid numerical convergence problems on Bessel and Neumann function calculation are also discussed.

Resumo

Apresenta-se um programa, em linguagem FORTRAN, desenvolvido para disciplina de Física Matemática a nível de graduação. Este programa pode ser usado para obter valores numéricos das funções especiais com grande precisão, gastando-se um tempo computacional consideravelmente pequeno. Os estudantes que dispõem de algum conhecimento de programação também podem usar as discussões aqui apresentadas para criar seus próprios programas para cálculo de funções especiais, ou modificar o programa apresentado para aumentar seus conhecimentos em relação às propriedades dessas funções. Os programas aqui apresentados são baseados nas próprias definições ou em fórmulas matemáticas mais simples possíveis associadas às funções, a fim de servir de material complementar ao ensino. No decorrer do artigo, discute-se alguns cuidados que devem ser tomados durante a programação de funções de Bessel e de Neumann, de modo a evitar problemas de convergência numérica.

1. Programa Principal

O programa discutido neste artigo é formado por uma série de funções-subrotinas acessadas por um programa principal. O programa principal é usado basicamente para gerenciar o uso das diferentes funções especiais, de forma interativa, semelhante a uma calculadora científica.

Neste programa, apresenta-se um breve cabeçalho, e em seguida, a lista de funções disponíveis. Uma vez escolhida a função, são solicitadas as informações específicas da função, e o valor do argumento. Após o cálculo da função, é dada a opção de novo cálculo de função ou de encerrar a execução do programa.

A primeira função listada no programa é a função fatorial, que discutiremos juntamente com as funções de Bessel.

2. Funções de Bessel

Em problemas envolvendo soluções da "Equação da Onda" (v. ref.[1], p. 336)

$$\nabla^2\psi = -\frac{1}{c^2}\frac{\partial^2\psi}{\partial t^2} \qquad (1)$$

e "Equação da Difusão" (v. ref. [2] p. 437)

$$\nabla^2\psi = -\frac{1}{a^2}\frac{\partial\psi}{\partial t}. \tag{2}$$

Em coordenadas cilíndricas, temos as equações diferenciais de Bessel

$$\rho\frac{d}{d\rho}\left(\rho\frac{dP}{d\rho}\right) + \left(n^2\rho^2 - m^2\right)P = 0 \tag{3}$$

como resultado da técnica de separação de variáveis, com:

$$\psi(\rho,\theta,z) = P(\rho)\Theta(\theta)Z(z) \tag{4}$$

As soluções da equação diferencial de Bessel podem ser escritas como uma combinação linear de funções de Bessel de primeira espécie e de segunda espécie, esta última, também conhecida como funções de Neumann.

3. Funções de Bessel de Primeira Espécie

Para programar as funções de Bessel, podemos fazer uso da definição sob a forma de expansão em série de potências, (v. ref. [1], p. 360 ou ref. [2] p. 575)

$$J_n(x) = \sum_{s=0}^{\infty}\frac{(-1)^s}{s!(n+s)!}\left(\frac{x}{2}\right)^{n+2s} \tag{5}$$

na medida que, em princípio, tal definição pode ser usada para funções de Bessel de qualquer ordem, tanto positiva quanto negativa, tanto inteira quanto não-inteira. O uso desta definição é particularmente útil para valores pequenos de x.

3.1 Estratégias de Programação

O primeiro cuidado que se deve ter antes de começar a programar um algorítmo qualquer, é evitar a divergência numérica em cálculos intermediários. No nosso caso específico, discutimos algumas formas de evitar certos problemas que podem surgir durante a programação dos algorítmos.

Função fatorial

Ao calcular uma função a partir da expansão em série, muitas vezes é possível identificar a presença de funções que, isoladamente, divergem rapidamente, como é o caso da função fatorial. Mas o programador mais atento perceberá que a divergência numérica devido ao fatorial na maioria das vezes pode ser evitada programando diretamente o inverso do valor da função fatorial, no lugar do fatorial, propriamente dito. Sendo assim, podemos definir uma função subrotina que calcula o inverso do valor da função fatorial.

Função x^n

A presença de funções do tipo x^n na expansão em série também podem acarretar problemas de convergência numérica. Mas, com um pouco mais de atenção, pode-se observar que na maioria das vezes na expansão de função em série de potência, tem-se a razão entre a função x^n e a função fatorial. Sendo assim, uma forma de se evitar divergência numérica, é calcular a razão entre as duas funções diretamente, sem a necessidade de calcular cada função isoladamente. Isso pode ser feito, se observarmos que a função fatorial (de um número inteiro) pode ser escrito como o produto de n termos e a função x^n também pode ser expresso como o produto de n termos, de modo que podemos escrever:

$$\frac{x^n}{n!} = \prod_{i=1}^{n}\left(\frac{x_i}{i}\right) \tag{6}$$

onde $x_i = x$, para qualquer valor de i entre 1 e n.

No caso específico da função de Bessel, tem-se a presença de fatorial na forma: $(n+s)!$. Aplicando a regra de recorrência:

$$z! = z(z-1)! \tag{7}$$

que é válido para qualquer valor de z real, podemos escrever:

$$(n+s)! = n!\prod_{i=1}^{s}(i+n) \tag{8}$$

Sendo assim, a expansão em série para a função de Bessel eq.(5) pode ser escrita na forma:

$$J_n(x) = \frac{(x/2)^n}{n!}\sum_{s=0}^{\infty}(-1)^s\prod_{i=0}^{s}\frac{w_i}{i}\frac{w_i}{n+i} \tag{9}$$

onde w_i no produtório é igual a $x/2$ para qualquer valor de i.

Usando a equação (9), pode-se programar as funções de Bessel de ordem inteira não negativa reduzindo problema de divergência numérica. Para as funções de ordem inteira estritamente negativa, pode-se adotar uma das duas estratégias a seguir; uma é usar a definição de que o valor do inverso da função fatorial de números inteiros estritamente negativos é identicamente nulo, e a outra é, usar a relação existente entre as funções de Bessel de ordem inteira positiva e negativa.

$$J_{-n}(x) = (-1)^n J_n(x) \tag{10}$$

Adotamos a segunda estratégia neste programa.

Table 1: $b_0(x)$ - Teste de convergência para $x = 6$

n	s_n	$\sum(-1)^n s_n$
1	9.0000000000000E+00	-8.0000000000000E+00
2	2.0250000000000E+01	1.2250000000000E+01
3	2.0250000000000E+01	-8.0000000000000E+00
4	1.1390625000000E+01	3.3906250000000E+00
5	4.1006250000000E+00	-7.1000000000000E-01
6	1.0251562500000E+00	3.1515625000000E-01
7	1.8829400510204E-01	1.2686224489796E-01
8	2.6478844467474E-02	1.5334108936543E-01
9	2.9420938297194E-03	1.5039899553571E-01
10	2.6478844467474E-04	1.5066378398039E-01
11	1.9695008281593E-05	1.5064408897211E-01
12	1.2309380175995E-06	1.5064531991013E-01
13	6.5552912179857E-08	1.5064525435721E-01
14	3.0100827021363E-09	1.5064525736730E-01
15	1.2040330808545E-10	1.5064525724689E-01
16	4.2329287998792E-12	1.5064525725113E-01
17	1.3182131210696E-13	1.5064525725099E-01
18	3.6617031140823E-15	1.5064525725100E-01
19	9.1288997303991E-17	1.5064525725100E-01

Um último ponto que merece discussão refere-se ao fato que na expansão em série o número de termos é infinito, mas em programação numérica torna-se necessário fazer um truncamento, levando-se em consideração, um compromisso entre a precisão numérica e a performance computacional. A solução que adotamos foi o de estabelecer um número superestimado de termos na expansão (no caso, 10000 termos), mas com um teste do valor de cada termo da expansão, durante o processo de soma. Assim que o valor do n-ésimo termo da expansão atinge um certo valor a série é truncada automaticamente. No programa colocado à disposição do leitor, foi fixada uma precisão da ordem de 10^{-17}. O número de termos necessários para atingir tal precisão é consideravelmente pequeno (da ordem de 20 termos ou menos como pode-se ver na tabela 1).

No caso das funções de Bessel de ordem não-inteira, também é possível usar a mesma expansão em série. Por conveniência, definimos uma nova função subrotina para essas funções de Bessel. Agora a dificuldade é a programação da função fatorial (ou do inverso do valor dessa função) para argumentos reais.

3.2 Função Fatorial

Uma forma simples e razoavelmente eficiente de programar o fatorial de um número real é através da fórmula de Stirling. (v. ref. [2] p. 556)

$$\ln(z!) = \frac{1}{2}\ln 2\pi + \left(z+\frac{1}{2}\right)\ln z - z - \sum_{n=1} \frac{B_{2n}}{(2n)(2n-1)} z^{1-2n} \qquad (11)$$

onde B_{2n} são os números de Bernoulli, definidos a partir da expressão(v. ref.[2] p. 327)

$$/\frac{x}{e^x - 1} = \sum_{n=0}^{\infty} \frac{B_n x^n}{n!} \qquad (12)$$

que podem ser obtidos por comparação dos termos das expansões em série. Na tabela 2, apresentamos alguns

valores dos números de Bernoulli.

Os números de Bernoulli também podem ser obtidos a partir da expressão(v. ref.[2] p. 328)

$$B_{2n} = \frac{(-1)^{n-1}\,2\,(2n)!}{(2\pi)^{2n}} \sum_{p=1}^{\infty} p^{-2n}, \qquad n = 1, 2, \cdots \tag{13}$$

A fórmula (11) permite o cálculo do fatorial com pelo menos 10 dígitos significativos, se truncarmos a expansão no vigésimo termo da série, como colocado no nosso programa. Quanto à forma de programar, é conveniente levar em consideração, a possibilidade de agrupar as duas funções exponenciais dentro de um mesmo argumento, de modo a evitar erros numéricos que podem ser introduzidos ao multiplicar um número muito grande por um número muito pequeno.

Um ponto importante na programação da função fatorial é assegurar que os valores obtidos tenham o mesmo grau de precisão. A estratégia para isso, é o de se calcular a função dentro de um intervalo conveniente (entre 9 e 10), e obter o valor da função fora desse intervalo, através da regra de recorrência. Para valores inteiros, pode-se aproveitar o programa discutido anteriormente.

3.3 Teste da expansão da em série

Para testar o programa desenvolvido, pode-se comparar os resultados numéricos com os valores apresentados em manuais de tabelas e funções especiais (v. ref. [3]). O programa discutido neste artigo reproduz os resultados numéricos do manual citado acima, em todos os quinze digitos apresentados na tabela, para a função de ordem zero quando se usa programa em precisão dupla, para valores de argumentos pequenos, até valores próximos a 6 (v. tabela 1).Mas o erro numérico torna-se extremamente elevado para argumentos superiores a 40. Cabe salientar que tal erro não decorre do truncamento da expansão em série, mas sim do fato que para tais valores de argumento, termos intermediários da série assumem valores superiores a 10^{15}, ao passo que o resultado da soma da série é da ordem do valor unitário, o que provoca erros nuéricos da ordem de cem porcento, quando se usa programa em precisão dupla. Para confirmar que tal erro realmente decorre da precisão dos cálculos, apresentamos na tabela 3 os valores de alguns termos da expansão em série para $x = 40$.

4. Limite Assintótico

Como vimos, para valores grandes do argumento, o problema de erros numéricos no cálculo da função através da expansão em série torna-se cada vez mais acentuados devido ao fato de ser necessário realizar soma e subtração de termos muito grandes com termos muito pequenos. Felizmente as funções de Bessel atingem um comportamento assintótico muito rapidamente (para valores de argumento da ordem de 10 ou menos). Sendo assim, é possível usar a fórmula para o limite assintótico da função de forma satisfatória. (v. ref.[2] p. 619)

$$J_\nu(z) = \sqrt{\frac{2}{\pi z}}\,[P_\nu(z)\cos\xi - Q_\nu(z)\mathrm{sen}\xi] \tag{14}$$

onde:

$$\xi = z - \left(\nu + \frac{1}{2}\right)\frac{\pi}{2} \tag{15}$$

sendo $P_\nu(z)$ e $Q_\nu(z)$, as séries: (v. ref.[2] p. 345)

$$P_\nu(z) \sim 1 + \sum_{n=1}^{\infty} (-1)^n \frac{\prod_{s=1}^{2n}\left[4\nu^2 - (2s-1)^2\right]}{(2n)!\,(8z)^{2n}} \tag{16}$$

e

$$Q_\nu(z) \sim \sum_{n=1}^{\infty} (-1)^{n+1} \frac{\prod_{s=1}^{2n-1}\left[4\nu^2 - (2s-1)^2\right]}{(2n-1)!\,(8z)^{2n-1}} \tag{17}$$

com

$$\mu = 4\nu^2$$

$$-\pi < \arg z < \pi$$

de fato as séries $P_\nu(z)$ e $Q_\nu(z)$ são divergentes, se consideradas como séries infinitas, mas para valores suficientemente elevados de z, pode-se obter o valor assintótico da função com um grau de precisão desejado, para uma expansão limitada das séries. (v. ref.[2] p. 343)

Testes comparativos entre a forma assintótica e valores tabelados (v. ref. [3] p.392), nos leva a concluir que já em valores de argumento da ordem de 10, o erro cometido ao se cálcular o valor da função a partir da fórmula para o limite assintótico é inferior a 10^{-6}, atingindo-se a precisão maior que 10^{-15}, para valores do argumento superiores a 17.

Table 3: Teste de convergência para $x = 40$

n	s_n	n	s_n
3	1.7777777777778E+06	6	7.9012345679012E+09
9	1.9907368525828E+12	12	7.3121647477790E+13
15	6.2791469610199E+14	18	1.6764777865938E+15
21	1.6848917145936E+15	24	7.3118698351168E+14
27	1.5193372438453E+14	30	1.6386250509952E+13
33	9.7860704946731E+11	36	3.4126314559543E+10
39	7.2638941691798E+08	42	9.7985814096454E+06
45	8.6511907430736E+04	48	5.1411705074504E+02
51	2.1075091099382E+00	54	6.0898064306054E-03
57	1.2645392620733E-05	60	1.9197725646046E-08
63	2.1642409341068E-11	66	1.8374310395737E-14
69	1.1899409498225E-17		

5. Fórmula de recorrência

Para o leitor descontente com a baixa precisão dos cálculos na faixa intermediárias de valores do argumento (entre 6 e 18), existe a opção de calcular as funções de Bessel de ordem inteira usando a fórmula de recorrência (v. ref. [2] p. 576)

$$J_{n-1}(x) + J_{n+1}(x) = \frac{2n}{x} J_n(x) \qquad (18)$$

no caso, escolhe-se um valor para n suficientemente elevado (para a precisão de 10^{-15}, $n \cong 30$ é o suficiente), ao qual associa-se o valor $J_{n+1}(x) = 0$ e um valor α, consideravelmente pequeno para $J_n(x)$. Em nossos testes $\alpha = 10^{-30}$ é o suficiente para evitar problemas de divergência numérica. O valor de $J_{n-1}(x)$ pode ser calculado usando-se a regra de recorrência e os valores das funções de Bessel de ordens inferiores também podem ser obtidas aplicando-se a mesma fórmula de recorrência. Como α é arbitrário, todos os valores de $J_n(x)$ calculados estarão multiplicados por um fator arbitrário em comum. Este fator pode ser determinado usando-se a condição(v. ref.[2] p. 577)

$$J_0(x) + 2\sum_{m=1}^{\infty} J_{2m}(x) = 1 \qquad (19)$$

Para obter a precisão numérica do cálculo realizado, basta repetir o mesmo procedimento acima, tomando-se um valor $n' = n - 3$, de modo que $J_{n'+1}(x) = 0$. Uma vez obtido o valor da função de Bessel desejada, compara-se a diferença entre os dois cálculos (v. ref.[2] p. 577). Os resultados apresentados na tabela 4 foram obtidos a partir do procedimento descrito acima.

Tabela 4: $b_0(x)$ - Teste de convergência para $x = 12$ usando regra de recorrência

n	$J_0(x)\|_{x=12}$	erro
4	4.3750000000000E-01	-8.4169307654803E-01
8	-5.7040656080963E-01	4.3771589294820E-01
12	-1.1703864997278E-02	6.0093491580376E-02
16	4.8008227153057E-02	-2.8899000191402E-04
20	4.7694583835821E-02	-5.0812178547660E-06
24	4.7689334053994E-02	-2.2658044912527E-08
28	4.7689310850157E-02	-5.2295917596368E-11
32	4.7689310796904E-02	-6.8972605404838E-14
36	4.7689310796834E-02	-8.3266726846887E-17

Para finalizar as discussões referentes aos cálculos da função de Bessel a partir das diferentes fórmulas disponíveis, apresentamos na tabela , uma comparação dos resultados obtidos a partir da fórmula de recorrência, expansão em série e limite assintótico. Como esperado, podemos perceber uma boa precisão dos resultados obtidos a partir da expansão em série para valores pequenos do argumento da função e boa precisão dos resultados obtidos a partir do limite assintótico, no caso de valores elevados do argumento da função.

Em termos de tempo de processamento, a programação aqui apresentada pode ser considerada ineficiente quando comparado com algoritmos especialmente desenvolvidos para cálculo das funções de Bessel, tais como os programas que expandem as funções de Bessel em forma polinomial. Alguns testes realizados

Tabela 5: $b_0(x)$ usando a regra de recorrência e comparação com o erro cometido ao usar a expansão em série e o limite assintótico

x	$J_0(x)$	erro: exp. em série	erro: lim. assintótico
1	7.6519768655797E-01	1.1102230246E-16	-3.1635462905E-02
4	-3.9714980986385E-01	0.0000000000E+00	-1.8954901574E-05
7	3.0007927051956E-01	5.7731597280E-15	-2.3658848380E-08
10	-2.4593576445135E-01	-1.7347234759E-14	-3.4129643555E-11
13	2.0692610237707E-01	1.5097090244E-12	-5.0154325137E-14
16	-1.7489907398363E-01	1.3805345755E-12	1.1102230246E-16
19	1.4662943965965E-01	-3.5890734828E-12	-5.5511151231E-17
22	-1.2065147570487E-01	9.7139852689E-09	8.3266726846E-17
25	9.6266783275958E-02	1.4987325538E-07	-4.1633363423E-17
28	-7.3157010549000E-02	-2.4772830062E-06	1.9428902930E-16
31	5.1208145304542E-02	1.0726715355E-04	-2.0122792321E-16
34	-3.0421191021793E-02	4.4309265498E-04	-4.3021142204E-16
37	1.0862369724900E-02	7.6088389931E-03	3.4174052476E-16
40	7.3668905842372E-03	-4.8656427167E-01	-2.8796409701E-16

por nós indicam que nossos algoritmos demoram cerca de duas vezes mais do que os programas que usam a expansão polinomial (v. ref. [4] p. 232). Mesmo assim, nosso programa pode ser considerado competitivo quando usado em microcomputadores 486 ou mesmo 386, na medida que cada resultado solicitado demora uma fração de tempo consideravelmente pequena (da ordem de milésimo de segundo ou menos), com a vantagem adicional de ser possível obter o resultado com o grau de precisão desejado, sem a necessidade de uma pesquisa aprofundada em bibliotecas de subrotinas. De qualquer maneira, nosso programa não tem a finalidade de competir com os melhores programas desenvolvidos para cálculo de funções especiais, mas sim o de servir de material para estudo das propriedades das funções especiais.

6. Função de Neumann

A programação das função de Neumann, pode ser feita a partir da definição (v. ref.[4] p. 230)

$$N_\nu(x) = \frac{\cos \nu\pi J_\nu(x) - J_{-\nu}(x)}{\operatorname{sen}\nu\pi} \qquad (20)$$

no caso de funções de ordem não-inteira ou a partir da equação

$$N_n(x) = A_n(x) - B_n(x) - C_n(x) \qquad (21)$$

para o caso de funções de ordem inteira, onde: (v. ref.[2] p. 598)

$$A_n(x) = \frac{2}{\pi}\left[\ln\left(\frac{x}{2}\right) + \gamma - \frac{1}{2}\sum_{p=1}^{n} p^{-1}\right] J_n(x) \qquad (22)$$

$$B_n(x) = \frac{1}{\pi}\sum_{r=0}^{\infty}(-1)^r \frac{(x/2)^{n+2r}}{r!(n+r)!}\sum_{p=1}^{r}\left[\frac{1}{p} + \frac{1}{p+n}\right] \qquad (23)$$

$$C_n(x) = \frac{1}{\pi}\sum_{r=0}^{n-1}\frac{(n-r-1)!}{r!}\left(\frac{x}{2}\right)^{-n+2r} \qquad (24)$$

onde γ é a constante de Euler-Mascheroni (v. ref.[2] p. 284)

A programação das funções de Neumann de ordem inteira segue basicamente a mesma estratégia de programação adotada durante a programação das funções de Bessel de primeira espécie e, naturalmente, a programação das funções de Neumann de ordem não inteira é realizada aproveitando-se das funções de Bessel previamente progamadas. Mais uma vez, tal procedimento de programação "esbarra" na limitação de precisão numérica no limite de grandes valores do argumento da função. Neste último caso, podemos usar a expressão para o limite assintótico (v. ref.[2] 619)

$$N_\nu(z) = \sqrt{\frac{2}{\pi z}}\left[P_\nu(z)\operatorname{sen}\xi + Q_\nu(z)\cos\xi\right] \qquad (25)$$

onde ξ, P_ν e Q_ν, são os mesmos das equações (15), (16) e (17), respectivamente, apresentados durante as discussões relativas ao limite assintótico das funções de Bessel.

7. Funções esféricas de Bessel e de Neumann

Para programar a função esférica, optamos em utilizar as formas polinomial-trigonométrica, para a função esférica de Bessel, (v. ref.[2] p. 630)

$$j_n(z) = \frac{1}{z}\left[\operatorname{sen}\left(z - \frac{n\pi}{2}\right) p_{1,n}\left(\frac{1}{z}\right) + \cos\left(z - \frac{n\pi}{2}\right) p_{2,n}\left(\frac{1}{z}\right)\right] \tag{26}$$

e

$$n_n(z) = \frac{(-1)^{n+1}}{z}\left[\cos\left(z + \frac{n\pi}{2}\right) p_{1,n}\left(\frac{1}{z}\right) + \operatorname{sen}\left(z + \frac{n\pi}{2}\right) p_{2,n}\left(\frac{1}{z}\right)\right] \tag{27}$$

onde

$$p_{1,n}\left(\frac{1}{z}\right) = \sum_{s=0}^{[n/2]} \frac{(-1)^s (n+2s)!}{(2s)!\,(2z)^{2s}\,(n-2s)!} \tag{28}$$

e

$$p_{2,n}\left(\frac{1}{z}\right) = \sum_{s=0}^{[(n-1)/2]} \frac{(-1)^s (n+2s+1)!}{(2s+1)!\,(2z)^{2s+1}\,(n-2s-1)!} \tag{29}$$

8. Funções de Bessel modificadas

As funções de Bessel modificadas, particularmente útil em problemas de difusão, pode ser obtida a partir da função de Bessel, multiplicando-se o argumento real por i(v. ref.[2] p. 610)

$$I_\nu(k\rho) = J_\nu(ik\rho). \tag{30}$$

Em termo da expansão em série temos:

$$I_\nu(x) = \sum_{s=0}^{\infty} \frac{1}{s!\,(s+\nu)!}\left(\frac{x}{2}\right)^{2s+\nu} \tag{31}$$

válido para qualquer valor de ν.

A segunda solução para o problema de difusão pode ser dada a partir da equação de Hankel modificada, na forma (v. ref.[2] p. 612 ou ref.[4] p. 236)

$$K_\nu(x) = \frac{\pi}{2} i^{\nu+1} H_\nu^{(1)}(ix) \tag{32}$$

$$K_\nu(x) = \frac{\pi}{2} i^{\nu+1}\left[J_\nu(ix) + iN_\nu(ix)\right] \tag{33}$$

Para x real, podemos escrever

$$K_\nu(x) = \frac{\pi}{2}\frac{I_{-\nu}(x) - I_\nu(x)}{\operatorname{sen}\nu\pi} \tag{34}$$

para ν inteiro, devemos tomar o valor limite

$$K_n(x) = F_n(x) + G_n(x) + H_n(x) \tag{35}$$

onde

$$F_n(x) = \frac{1}{2}\left(\frac{x}{2}\right)^{-n} \sum_{k=0}^{n-1} \frac{(n-k-1)!}{k!}\left(-\frac{x^2}{4}\right)^k \tag{36}$$

$$G_n(x) = (-1)^{n+1} \ln\left(\frac{x}{2}\right) I_n(x) \tag{37}$$

e

$$H_n(x) = \frac{(-1)^n}{2}\left(\frac{x}{2}\right)^n \sum_{k=0}^{\infty}\left[-2\gamma + \sum_{p=1}^{k} p^{-1} + \sum_{q=1}^{n+k} q^{-1}\right] \frac{1}{k!\,(n+k)!}\left(\frac{x^2}{4}\right)^k \tag{38}$$

9. Funções de Bessel esféricas modificadas

A programação das funções esféricas modificadas podem ser obtidas a partir das relações: (v. ref. [2] p. 633)

$$i_n(x) = \sqrt{\frac{\pi}{2x}} I_{n+1/2}(x) \tag{39}$$

e

$$k_n(x) = \sqrt{\frac{2}{\pi x}} K_{n+1/2}(x) \tag{40}$$

10. Polinômios de Legendre

Dados os polinômios de Legendre de ordem mais baixa,

$$P_0(x) = 1 \tag{41}$$

e

$$P_1(x) = x\,, \tag{42}$$

pode-se calcular os polinômios de ordem superior, usando-se a fórmula de recorrência, (v. ref.[4] p. 646 ou ref.[1] p. 349)

$$P_{n+1}(x) = \frac{1}{n+1}\left[(2n+1)\,x P_n(x) - n P_{n-1}(x)\right] \tag{43}$$

11. Polinômios associados de Legendre

Para os polinômios associados de Legendre, devemos tomar alguns cuidados de modo a preservar a precisão dos resultados. Para efeito de cálculo, um bom procedimento é (ver ref.[4] p. 253 ou ref.[1] p. 378)

$$(l-m)\,P_l^m = x\,(2l-1)\,P_{l-1}^m - (l+m-1)\,P_{l-2}^m \tag{44}$$

Os valores iniciais podem ser calculados com auxilio da expressão

$$P_m^m = (-1)^m\,(2m-1)!!\,(1-x^2)^{m/2} \tag{45}$$

onde

$$(2m-1)!! = (2m-1)\cdot(2m-3)\cdot\ldots\cdot 3\cdot 1 \tag{46}$$

usando a equação (44) para $l = m+1$ e considerando que $P_{m-1}^m = 0$, temos

$$P_{m+1}^m = x\,(2m+1)\,P_m^m \tag{47}$$

12. Polinômios de Hermite

Dados os polinômios de Hermite

$$H_0(x) = 1 \tag{48}$$

e

$$H_1(x) = 2x\,, \tag{49}$$

podemos calcular os polinômios de ordem superior usando a regra de recorrência (v. ref.[2] p. 712 ou ref.[1] p. 475)

$$H_{n+1}(x) = 2xH_n(x) - 2nH_{n-1}(x) \tag{50}$$

13. Polinômios de Laguerre

Os polinômios de Laguerre também podem ser calculados a partir de uma regra de recorrência (v. ref.[2] p. 724)

$$L_{n+1}(x) = 2L_n(x) - L_{n-1}(x) - \frac{\left[(1-x)\,L_n(x) - L_{n-1}(x)\right]}{(n+1)} \tag{51}$$

na medida em que se conheça os polinômios de ordem mais baixa,

$$L_0(x) = 1 \tag{52}$$

e

$$L_1(x) = -x + 1 \tag{53}$$

14. Polinômios associados de Laguerre

Os polinômios associados de Laguerre podem ser calculados diretamente da expressão (v. ref. [2] p. 725)

$$L_n^k(x) = \sum_{m=0}^{n} (-1)^m\,\frac{(n+k)!}{(n-m)!\,(k+m!)\,m!}\,x^m\,, \qquad k > -1. \tag{54}$$

ou então, a partir da regra de recorrência

$$L_{n+1}^k(x) = \frac{(2n+k+1-x)\,L_n^k(x) - (n+k)\,L_{n-1}^k(x)}{(n+1)} \tag{55}$$

usando o fato que

$$L_0^k(x) = 1 \tag{56}$$

e

$$L_1^k(x) = -x + k + 1 \tag{57}$$

15. Considerações finais

O uso de computadores pessoais de boa performance permite obter resultados numéricos para as funções especiais com grande precisão. No programa colocado à disposição dos leitores, tanto o programa-fonte, em linguagem FORTRAN, quanto o modo executável podem ser obtidos por carta endereçada ao autor. O programa-fonte também pode ser obtido através de correio eletrônico, enviando a solicitação ao endereço eletrônico: nilo@fis.unb.br

Cópias deste programa também estão à disposição na secretaria geral da SBF. Cabe salientar que o programa-fonte pode ser compilado em grande parte dos computadores, desde microcomputadores a sistemas de grande porte sem grandes problemas de compilação. Porém, deve-se levar em consideração as diferenças intrinsecas dos compiladores, o que pode levar a resultados com precisão diferente em máquinas diferentes. Por exemplo, ao compilar o programa usando o compilador f77 do sistema VAX/ultrix, deve-se levar em consideração o fato que a definição de valores das variáveis de dupla precisão deve ser feito com um certo cuidado. Por exemplo, a definição de valor $x = 0$ pode não acarretar um valor exatamente igual a "zero", já que nesta definição de valor, o compilador assume o valor "zero" apenas à primeira palavra reservada à variável. Para que o compilador ássuma o valor zero em ambas as palavras reservadas à variável, é necessário definir o valor da variável na forma x =0.D0 .

References

[1] Butkov, E. *Física Matemática*, Guanabara Dois, Rio de Janeiro, 1978

[2] Arfken, G. *Mathematical Methods for Physicists*, Academic Press, San Diego, California, 1985

[3] Abramowitz, M. & Stegun, I. A., *Handbook of Mathematical Functions*, Dover, New York, 1968

[4] Press, W. H., Teukolsky, S. A., Vetterling, W. T. & Flannery, B. P. *Numerical Recipes in C*, Cambridge University Press, Cambridge, 1992