CÁLCULO DAS FUNÇÕES DE BESSEL PROGRAMAÇÃO EM FORTRAN II PARA COMPUTADOR DIGITAL IBM 1620 MODÊLO 2

LÚCIA FARIA SILVA

PUBLICAÇÃO IEA N.º 138

Abril — 1967

INSTITUTO DE ENERGIA ATÔMICA
Caixa Postal 11049 (Pinheiros)
CIDADE UNIVERSITÁRIA "ARMANDO DE SALLES OLIVEIRA"
SÃO PAULO — BRASIL

C Á L C U L O   D A S   F U N Ç Õ E S   D E   B E S S E L

PROGRAMAÇÃO EM FORTRAN II PARA COMPUTADOR DIGITAL

I B M 1620 MODÊLO 2

Lúcia Faria Silva

Serviço de Cálculo Analógico e Digital
Instituto de Energia Atômica
São Paulo - Brasil

Publicação IEA nº 138
Abril - 1967

Comissão Nacional de Energia Nuclear

Presidente: Prof. Uriel da Costa Ribeiro

Universidade de São Paulo

Reitor: Prof. Dr. Luiz Antonio da Gama e Silva

Instituto de Energia Atômica

Diretor: Prof. Rômulo Ribeiro Pieroni

Conselho Técnico-Científico do IEA

Prof. Dr.	José Moura Gonçalves	pelo USP
Prof. Dr.	José Augusto Martins	
Prof. Dr.	Rui Ribeiro Franco	pela CNEN
Prof. Dr.	Theodoreto H. I. de Arruda Souto	

Divisões Didático-Científicas

Divisão de Física Nuclear -
Chefe: Prof. Dr. Marcello D. S. Santos

Divisão de Radioquímica -
Chefe: Prof. Dr. Fausto Walter de Lima

Divisão de Radiobiologia -
Chefe: Prof. Dr. Rômulo Ribeiro Pieroni

Divisão de Metalurgia Nuclear -
Chefe: Prof. Dr. Tharcísio D. S. Santos

Divisão de Engenharia Química -
Chefe: Lic. Alcídio Abrão

Divisão de Engenharia Nuclear -
Chefe: Engº Pedro Bento de Camargo

Divisão de Operação e Manutenção de Reatores
Chefe: Engº Azor Camargo Penteado Filho

Divisão de Física de Reatores -
Chefe: Prof. Paulo Saraiva de Toledo

Divisão de Ensino e Formação -

C Á L C U L O   D A S   F U N Ç Õ E S   D E   B E S S E L

PROGRAMAÇÃO EM FORTRAN II PARA COMPUTADOR DIGITAL

I B M 1620 MODÊLO 2

Lúcia Faria Silva

RESUMO

Êste trabalho apresenta uma série de programas tendo como objetivo o cálculo das funções de Bessel; êles podem ser ainda utilizados como sub-programas em outros cálculos, ou mesmo como programas isolados.

Todos os programas foram escritos em linguagem Fortran II, e foram processados num Computador IBM 1620 Mod. 2, utilizando o Sistema Monitor I.

São apresentados também comentários sôbre as expressões escolhidas, listagens Fortran, diagramas-bloco, e exemplos numéricos.

RESUMÉ

On présente une série de programmes en language Fortran II pour le calcul des Fonctions de Bessel. Ils peuvent être utilisés soit comme sous-programmes dans d'autres calculs, soit comme des programmes isolés. Tous les programmes sont adaptés à l'ordinateur IBM 1620 - Mod 2, utilisant le sistème Monitor I.

Ils sont aussi presentés commentaires sur les éxpressions choisies, "lists Fortran", diagrammes bloc, et éxemples numériques.

RESUME

This report outlines a set of programs developed for Bessel functions calculation. The programs may also be used like subprograms in many others calculations, as well as single programs.

They are developed in Fortran II language, and were processed in a IBM 1620 - Mod. 2 Digital Computer, provided with Monitor I System.

Comments about accuracy of expressions chosen, Fortran lists, flow charts, as well as numerical examples, are presented.

I - INTRODUÇÃO

Diversas são as aplicações das funções de Bessel, quer sejam em problemas astronômicos, telecomunicações, ou em simples circuitos eletrônicos.

Em particular, são muito utilizadas em Física de reatores, no cálculo do fator de utilização térmica, probabilidade de escape à ressonância, distribuição do fluxo (em função da distância radial) em arranjos multiplicadores e meios moderadores, e em muitos outros cálculos.

Êste trabalho apresenta um programa para o cálculo destas importantes funções de Bessel, visando sua utilização como subrotinas em outros programas que as requerem, ou como um programa isolado independente de qualquer programa principal.

Como o cálculo das funções de Bessel requer a utilização de outras funções, como por exemplo a função Gama, e cujo cálculo justifica o uso do computador, também são apresentados os programas para o cálculo dessas funções auxiliares; estas funções , por sua vez, além de serem usadas para calcular funções de Bessel, têm grande aplicação em outros cálculos.

Todos os programas foram escritos em linguagem FORTRAN II e foram processados num computador Digital IBM 1620 modêlo 2 utilizando o Sistema Monitor 1.

II - PRELIMINARES

Primeiramente apresentamos, em forma sintética, o desenvolvimento matemático das funções de Bessel e algumas considerações sôbre os polinômios de Laguerre, e em seguida os programas necessários para os cálculos destas funções.

Na apresentação dos programas consideramos o método desenvolvido, maneira de utilizar o programa, posições de memória requeridas, problema amostra e tempo de processamento. Para os programas principais acrescentamos diagramas-bloco.

Muitos programas são desdobrados em outros, ou para utilização dêstes diretamente, ou para melhor aproveitar a capacidade do computador.

De acôrdo com o que será exposto posteriormente os cálculos das funções de Bessel são desenvolvidos por série de potências, quando a convergência é mais ou menos rápida; no caso de x e ν serem tais que a convergência é lenta ($x \gg 1$), os cálculos são feitos usando expansão assintótica ou fórmulas de recorrência. As funções modificadas de ordem 0 e 1 são desenvolvidas por aproximações polinomiais. No caso particular de $\nu = m + 1/2$ ou $\nu = -(m + 1/2)$ (m inteiro) usamos funções circulares. Para calcular a função de Bessel de primeira classe e ordem n (n inteiro), adaptamos basicamente o programa de C. E. Grosch (1) e transformamos em sub-programa, introduzindo algumas modificações.

As funções de Bessel foram "identificadas" (para programação) de acôrdo com a ordem, em dois grupos: Funções de ordem inteira (FBEJ, FBEY, FBEI, FBEK) e funções de ordem fracionária ... (BEJN, BEYN, BEIN, BEKN). (Diagramas bloco são feitos para êstes grupos). Cada programa de um grupo, por sua vez, utiliza uma série de outros, para valores particulares de ν e x. Desta forma, para calcular uma determinada função, basta "utilizar" o programa de acôrdo com o grupo (ν inteiro ou fracionário) ou, caso se saiba o valor de ν e os valores extremos de x, o programa particular para o caso. Para cada grupo são feitos problemas-amostra que exemplificam os programas utilizados. Assim, por exemplo, no programa FBEY (cálculo da função de Bessel de segunda classe e ordem inteira n) o problema-amostra exemplifica não só o uso do programa ... FBEY, pròpriamente dito, mas também dos programas utilizados por êle, FBSEY e FBYN. Nota-se, com os exemplos dados posteriormente, que o tempo de processamento varia muito dependendo dos valores de x e ν ; isto ocorre porque êsses valores determinam o programa a ser usado e êstes programas usam desenvolvimentos matemáticos bem diferentes entre si, tornando mais, ou menos, simples a computação.

Seguimos o seguinte esquema na apresentação dos programas:

Em primeiro lugar apresentamos o programa AINT, pois o programa que calcula a função Gama é uma adaptação dêste. Depois os programas utilizados pelo AINT: RALAG, COLAG, VPOL, DPOL, LAGG, DELAG, DIPOL e FAT. A seguir o programa GAMA e os das funções de Bessel: FBEJ, BEJN, êste último auxiliado pelos programas FBSEJ, FBJN e BJMEIO; posteriormente os programas FBEY, BEYN e os que êles utilizam: FBSEY e FBYN; em continuação os programas FBEI , BEIN para os quais são necessários os programas FBIO, FBI1, FBSEI e FBIN; por último os programas FBEK, BEKN que utilizam os programas FBKO, FBK1, FBSEK, FBKN, FSENH e FCOSH.

III - FUNÇÕES DE BESSEL - DESENVOLVIMENTO MATEMÁTICO

A forma clássica da equação diferencial de Bessel é

$$t^2 \frac{d^2y}{dx^2} + t \frac{dy}{dx} + (\lambda^2 t^2 - n^2) y = 0$$

equação de Bessel de ordem n com parâmetro λ. Fazendo uma mudança de variável, $x = \lambda t$ a equação transforma-se em

$$x^2 \frac{d^2y}{dx^2} + x \frac{dy}{dx} + (x^2 - n^2)\ y = 0$$

ou

$$\frac{d^2y}{dx^2} + \frac{1}{x} \frac{dy}{dx} + (1 - \frac{n^2}{x^2})\ y = 0$$

A primeira solução desta equação denomina-se função de Bessel de la. classe e ordem n, $J_n(x)$, e uma segunda solução, devida a Weber, função de Bessel de 2a. classe e ordem n, $Y_n(x)$. Quando n é inteiro, desenvolvendo-se em séries de potências, essas funções são dadas por

$$J_n(x) = \sum_{k=o}^{\infty} (-1)^k \frac{(x/2)^{n+2k}}{(n+k)!\ k!}$$

$$Y_n(x) = \frac{2}{\pi} J_n(x) \left[\ln \frac{x}{2} + \gamma \right] - \frac{1}{\pi} \sum_{k=o}^{\infty} \frac{(n-k-1)!}{k!} (\frac{2}{x})^{n-2k} -$$

$$- \frac{1}{\pi} \sum_{k=o}^{\infty} \frac{(-1)^k (\frac{x}{2})^{n+2k}}{k!\ (n+k)!} \left[1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{k} + 1 + \frac{1}{2} + \ldots + \frac{1}{n+k} \right]$$

Quando k=o êste último têrmo é $\left[1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{n} \right]$

γ = constante de Euler = .57772

Quando n é qualquer, $n = \nu$, introduzimos a função Gama. Lembrando que $(\nu + k)! = \Gamma(\nu + k + 1)$

a função $J_\nu(x)$ fica

$$J_\nu(x) = \sum_{k=0}^{\infty} (-1)^k \frac{(\frac{x}{2})^{\nu + 2k}}{k!\ \Gamma\ (\nu + k + 1)}$$

A 2a. solução $Y_\nu(x)$, neste caso é dada por:

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

Para grandes valores de x as funções de Bessel são desenvolvidas por expansão assintótica. Assim, para $x \gg 1$ e $x \gg \nu^2$ temos

$$J_\nu(x) = \sqrt{\frac{2}{\pi x}} \left[\zeta_\nu(x)\ \cos\psi - \xi_\nu(x)\ \operatorname{sen}\psi \right]$$

$$Y_\nu(x) = \sqrt{\frac{2}{\pi x}} \left[\zeta_\nu(x)\ \operatorname{sen}\psi + \xi_\nu(x)\ \cos\psi \right]$$

onde

$$\zeta_\nu(x) \sim 1 - \frac{(4\nu^2-1^2)(4\nu^2-3^2)}{2!\ (8x)^2} + \frac{(4\nu^2-1^2)(4\nu^2-3^2)(4\nu^2-5^2)(4\nu^2-7^2)}{4!\ (8x)^4} - \ldots$$

$$\xi_\nu(x) \sim \frac{4\nu^2-1^2}{1!\ 8x} - \frac{(4\nu^2-1^2)(4\nu^2-3^2)(4\nu^2-5^2)}{3!\ (8x)^3} + \ldots$$

O menor têrmo (p-ésimo) ocorre aproximadamente quando $(p-1)^2 = x^2$ ou $p = 1 + x$.

As funções de Bessel guardam entre si uma certa relação de recorrência; assim, para x não muito menor que ν temos:

$$J_{\nu+1}(x) = \frac{2\nu}{x} J_\nu(x) - J_{\nu-1}(x)$$

$$Y_{\nu+1}(x) = \frac{2}{x} Y_{\nu}(x) - Y_{\nu-1}(x)$$

No caso particular de $\nu = n+1/2$ e $\nu = -n - 1/2$, n inteiro e positivo, as funções de Bessel podem ser dadas por funções circulares:

$$J_{n+\frac{1}{2}}(x) = \sqrt{\frac{2}{\pi x}} \left[\operatorname{sen}(x - \frac{1}{2} n\pi) \sum_{k=o}^{\leq \frac{1}{2} n} \frac{(-1)^k (n+2k)!}{(2k)!(n-2k)!(2x)^{2k}} + \right.$$

$$\left. + \cos(x - \frac{1}{2} n\pi) \sum_{k=o}^{\leq \frac{1}{2}(n-1)} \frac{(-1)^k (n+2k+1)!}{(2k+1)!(n-2k-1)!(2x)^{2k+1}} \right]$$

$$J_{-n-\frac{1}{2}}(x) = \sqrt{\frac{2}{\pi x}} \left[\cos(x + \frac{1}{2} n\pi) \sum_{k=o}^{\leq \frac{1}{2} n} \frac{(-1)^k (n+2k)!}{(2k)!(n-2k)!(2x)^{2k}} - \right.$$

$$\left. - \operatorname{sen}(x + \frac{1}{2} n\pi) \sum_{k=o}^{\leq \frac{1}{2}(n-1)} \frac{(-1)^k (n+2k+1)!}{(2k+1)!(n-2k-1)!(2x)^{2k+1}} \right]$$

$$Y_{n+\frac{1}{2}}(x) = (-1)^{n+1} J_{-n-\frac{1}{2}}(x)$$

$$Y_{-n-\frac{1}{2}}(x) = (-1)^{n} J_{n+\frac{1}{2}}(x)$$

FUNÇÕES DE BESSEL MODIFICADAS

Se na equação $\frac{d^2y}{dx^2} + \frac{1}{x} \frac{dy}{dx} + (1 - \frac{n^2}{x^2}) y = 0$

fizermos $x = j\,x$ teremos:

$$\frac{d^2y}{dx^2} + \frac{1}{x}\frac{dy}{dx} - (1 + \frac{n^2}{x^2}) = 0$$

Considerando a função $J_\nu(x)$ e substituindo x por jx, teremos

$$J_\nu(jx) = j^\nu \sum_{k=o}^{\infty} \frac{(\frac{x}{2})^{\nu+2k}}{k!\ \Gamma\ (\nu+k+1)}$$

Esta equação serve para definir a função modificada de 1a. classe e ordem ν

$$I_\nu(x) = j^{-\nu} J_\nu(jx) = \sum_{k=o}^{\infty} \frac{(\frac{x}{2})^{\nu+2k}}{k!\ \Gamma\ (n+k+1)}$$

A função de Bessel modificada de 2a. classe $K_\nu(x)$ é definida para $\nu = n$ como

$$K_n(x) = \frac{1}{2} \sum_{k=o}^{n-1} \frac{(-1)^k\ (n-k-1)!}{k!} (\frac{2}{x})^{n-2k} + (-1)^{n+1} \sum_{k=o}^{\infty} \frac{(\frac{1}{2}x)^{n+2k}}{k!\ (n+k)!}\ x$$

$$x \left[\ln\ (\frac{1}{2}\ x) - \frac{1}{2} \left[\psi(r+1) + \psi(n+r+1) \right] \right] \text{, onde}$$

$$\psi(r+1) = (1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{r}) - \gamma \qquad \psi(1) = -\gamma$$

$$\psi(n+r+1) = (1 + \frac{1}{2} + \ldots + \frac{1}{n+r}) - \gamma$$

Para ν qualquer

$$K_\nu(x) = \frac{\frac{1}{2}\pi \left[I_{-\nu}(x) - I_\nu(x) \right]}{\text{sen}\ \nu\ \pi}$$

Analogamente às funções generalizadas temos as seguintes relações:

Expansão assintótica:

$$I_\nu(x) \sim \frac{e^x}{\sqrt{2\pi x}} \left[1 - \frac{4\nu^2-1^2}{1!\ 8x} + \frac{(4\nu^2-1^2)(4\nu^2-3^2)}{2!\ (8x)^2} - \dots \right]$$

$$K_\nu(x) \sim \sqrt{\frac{\pi}{2x}}\ e^{-x} \left[1 + \frac{4\nu^2-1}{1!\ 8x} + \frac{(4\nu^2-1^2)(4\nu^2-3^2)}{2!\ (8x)^2} + \dots \right]$$

Recorrência: $I_{\nu+1}(x) = -\frac{2\nu}{x}\ I_\nu(x) + I_{\nu-1}(x)$

$$K_{\nu+1}(x) = \frac{2\nu}{x}\ K_\nu(x) + K_{\nu-1}(x)$$

$\nu = n + \frac{1}{2}$, $\nu = -n - \frac{1}{2}$:

$$I_{n+\frac{1}{2}}(x) = \sqrt{\frac{1}{2\pi x}} \left[e^x \sum_{k=o}^{n} \frac{(-1)^k\ (n+k)!}{k!\ (n-k)!(2x)^k} + \right.$$

$$\left. + (-1)^{n+1}\ e^{-x} \sum_{k=o}^{n} \frac{(n+k)!}{k!\ (n-k)!(2x)^k} \right]$$

$$I_{-n-\frac{1}{2}}(x) = \sqrt{\frac{1}{2\pi x}} \left[e^x \sum_{k=o}^{n} \frac{(-1)^k\ (n+k)!}{k!\ (n-k)!(2x)^k} + \right.$$

$$\left. + (-1)^{n}\ e^{-x} \sum_{k=o}^{n} \frac{(n+k)!}{k!\ (n-k)!(2x)^k} \right]$$

$$K_{n+\frac{1}{2}} = K_{-n-\frac{1}{2}} = \sqrt{\frac{\pi}{2x}} \; e^{-x} \sum_{k=o}^{n} \frac{(n+k)!}{k!\,(n-k)!\,(2x)^k}$$

As funções de Bessel de ordem n=0 e n=1, podem ser também representadas por aproximações polinomiais, sem êrros muito grandes.

Utilizamos êste desenvolvimento para as funções modificadas. Assim,

para $-3.75 < x < 3.75$ temos a seguinte relação:

$$x^{-n} I_n(x) = \sum_{k=o}^{6} e_{kn}(x/3.75)^{2k} + \varepsilon_n(x) \qquad n=0,\ n=1$$

onde

$$|\varepsilon_0(x)| \leq 1 \times 10^{-7}$$
$$|\varepsilon_1(x)| \leq 1 \times 10^{-8}$$

e os valores de k (0, 1, 2, 3, 4, 5 e 6), e_{k0} e e_{k1} são dados pela tabela seguinte:

k	e_{k0}	e_{k1}
0	1.0000000	0.50000000
1	3.5156229	0.87890594
2	3.0899424	0.51498869
3	1.2067492	0.15084934
4	0.2659732	0.02658733
5	0.0360768	0.00301532
6	0.0045813	0.00032411

para $3.75 \leqslant x \leqslant \infty$ temos

$$x^{1/2}\, e^{-x}\, I_n(x) = \sum_{k=o}^{8} g_{kn}\, (x/3.75)^{-k} + \varepsilon_n(x) \qquad n=0,\ n=1$$

onde

$$|\varepsilon_0(x)| \leqslant 11 \times 10^{-9}$$
$$|\varepsilon_1(x)| \leqslant 11 \times 10^{-9}$$

e k, g_{k0} e g_{k1} são:

k	g_{k0}	g_{k1}
0	0.398942280	0.398942280
1	0.013285917	-0.039880242
2	0.002253187	-0.003620183
3	-0.001575649	0.001638014
4	0.009162808	-0.010315550
5	-0.020577063	0.022829673
6	0.026355372	-0.028953121
7	-0.016476329	0.017876535
8	0.003923767	-0.004200587

para $0 \leqslant x \leqslant 2$ temos

$$x^n(K_n(x) + (-1)^n\, (\ln\, (x/2))I_n(x)) = \sum_{k=o}^{6} f_{kn}(x/2)^{2k} + \eta_n(x) \qquad n=0, n=1$$

onde

$$|\eta_0(x)| \leqslant 1 \times 10^{-8}$$
$$|\eta_1(x)| \leqslant 4 \times 10^{-9}$$

e k, f_{k0} e f_{k1} são:

k	f_{k0}	f_{k1}
0	-0.57721566	1.00000000
1	0.42278420	0.15443144
2	0.23069756	-0.67278579
3	0.03488590	-0.18156897
4	0.00262698	-0.01919402
5	0.00010750	-0.00110404
6	0.00000740	-0.00004686

para $2 < x$ temos

$$x^{1/2}\ e^{x}\ K_n(x) = \sum_{k=o}^{6} h_{kn}(x/2)^{-k} + \eta_n(x) \qquad n=0,\ n=1$$

onde

$$|\eta_0(x)| \leqslant 7 \times 10^{-8}$$
$$|\eta_1(x)| \leqslant 10 \times 10^{-8}$$

e k, h_{k0} e h_{k1} são:

k	h_{k0}	h_{k1}
0	1.25331414	1.25331414
1	-0.07832358	0.23498619
2	0.02189568	-0.03655620
3	-0.01062446	0.01504268
4	0.00587872	-0.00780353
5	-0.00251540	0.00325614
6	0.00053208	-0.00068245

Antes de entrarmos no cálculo pròpriamente dito das funções de Bessel, começamos com uma série de programas relacionados com os polinômios de Laguerre, que são usados para o cálculo da função Gama ($\Gamma(n) = \int_0^\infty e^{-x} \, x^{n-1} \, dx$), empregada nas fórmulas de desenvolvimento em série; só depois dêstes programas preliminares calculamos as funções de Bessel, pròpriamente ditas.

Os programas que utilizam polinômios de Laguerre, devido à limitação da capacidade do computador, consideram polinômios de grau, no máximo, 20. Êles se aplicam, no entanto, para graus maiores.

Os polinômios de Laguerre de ordem n, são definidos pela expressão

$$L_n(x) \equiv e^x \frac{d^n}{dx^n} (x^n e^{-x})$$

e são soluções polinomiais da equação diferencial de 2a. ordem:

$$\frac{x \, d^2 L_n}{dx^2} + (1 - x) \frac{d \, L_n}{dx} + n \, L_n = 0$$

onde n é 0 ou inteiro positivo.

Pelo método de integração por séries, essas soluções podem ser escritas da seguinte forma:

$$L_n(x) = n! \sum_{j=0}^{n} \binom{n}{n-j} \frac{(-x)^j}{j!} \; ;$$

essa indicação é a que usamos nos nossos cálculos.

IV - <u>PROGRAMA AINT</u>

<u>Cálculo da integral</u> $I = \int_0^\infty e^{-x} \, f(x) \, dx$

Para o cálculo desta integral utilizamos o método de Gauss - Laguerre.

O valor da integral I é dado por:

$$I = \sum_{j=1}^{n} H_j \, f(a_j) \qquad \text{(a)}$$

onde

$$H_j = \frac{(n!)^2}{a_j \, L_n'(a_j)^2}$$

e $f(a_j)$ é o valor da função $f(x)$ no ponto a_j. Iniciamos o programa com o polinômio de Laguerre de grau n, $L_n(x)$. Determinamos as raízes $a_j (j = 1,\ldots n)$ dêste polinômio, utilizando o programa RALAG. Em seguida, utilizando o programa DELAG, calculamos as derivadas $L_n'(a_j)$, derivadas de $L_n(x)$ nos pontos a_j. O valor aproxima do I_n, da integral I é dado por (a). Incrementando n de 1 ... $(n \to n + 1)$, fazemos os mesmos cálculos anteriores, determinando I_{n+1}. Vamos repetindo o raciocínio até encontrarmos uma relação entre I_n e I_{n+1} menor ou igual a uma determinada precisão. Nestas condições, I_{n+1} é o valor da integral I.

<u>Entrada</u>: EM - precisão desejada

N - grau do polinômio inicial

<u>Saída</u>: AINT - valor da integral

<u>Posições requeridas</u>: 1702 posições de memória

<u>Problema amostra</u>: foi resolvida a integral:

$$I = \int_0^\infty y^{2m} e^{-y^2} dy \quad (m=4)$$

que é da forma

$$I = \int_0^\infty \frac{x^m}{2\sqrt{x}} e^{-x} dx$$

Precisão: 10^{-2}

$N = 7$

Resultado: .58158464E+01

Tempo de processamento: 18 seg

Diagrama-bloco: pág. I-1 - Apêndice I

Listagem Fortran: pág. II-1 - Apêndice II

IV-1 - PROGRAMA RALAG

Determinação das raízes dos polinômios de Laguerre

Para o cálculo da primeira raiz de $L_n(x)$ utilizamos o método de Newton - Raphson, como segue:

Primeiramente escolhemos um valor aproximado x_0 da menor raiz de $L_n(x)$. Calculamos o valor do polinômio e sua derivada neste ponto, $L_n(x_0)$ e $L_n'(x_0)$.

(Utilizamos os programas LAGG e DELAG). Os valores aproximados seguintes da raiz foram calculados pela equação de recorrência

$$x_{i+1} = x_i - \frac{L_n(x_i)}{L_n'(x_i)}$$

Depois de cada iteração comparamos x_{i+1} com x_i e continuamos com as iterações até obtermos uma relação entre x_i e x_{i+1} menor que uma determinada precisão; nestas condições x_{i+1} é a primeira raiz (x_1) de $L_n(x)$. Para calcularmos as restantes n-1 raízes dividimos $L_n(x)$ por $(x - x_1)$ (utilizamos o programa DIPOL), obtendo um polinômio $P_m(x)$ de grau $m = n-1$. As m raízes de $P_m(x)$ são as n-1 raízes de $L_n(x)$.

Para pesquisar as raízes de $P_m(x)$ utilizamos o mesmo método anterior, sendo que neste caso consideramos x_1 como raiz aproximada da primeira raiz (x_2) de P_m; seguimos êste procedimento (diminuindo o grau do polinômio sempre de 1), até calcularmos a raiz x_n de $P_1(x)$. Neste ponto temos tôdas as raízes x_1, x_2,... x_n de $L_n(x)$.

Para n $\leqslant$ 15 introduzimos os valores das raízes tirados diretamente de uma tabela (6)

Argumentos de entrada:

NEGRAP - grau de $L_n(x)$
XO - valor aproximado da la. raiz
EPS - precisão desejada

Saída: R - Arranjo R_1, R_2, ... R_N, raízes de $L_n(x)$.

Posições requeridas: 11808 posições de memória

Problema amostra: Determinou-se as raízes do polinômio de grau 16

XO = .1 E-04
EPS = .1 E-02

Saída:

RN 1 = .87649399E-01
RN 2 = .46269626E+00
RN 3 = .11410628E+01
RN 4 = .21291833E+01
RN 5 = .34378037E+01
RN 6 = .50747062E+01
RN 7 = .70793882E+01
RN 8 = .94391741E+01
RN 9 = .12104902E+02
RN10 = .16059829E+02
RN11 = .18084699E+02
RN12 = .25718067E+02
RN13 = .26470434E+02

RN14 = .35283366E+02

RN15 = .41696680E+02

RN16 = .51730405E+02

Tempo de processamento: 1m 27 seg

Diagrama-bloco: pág. I-2 - Apêndice I

Listagem Fortran: pág. II-1 - Apêndice II

IV-2 - PROGRAMA COLAG

Determinação dos coeficientes dos polinômios de LAGUERRE

Calculamos os coeficientes dos polinômios de Laguerre diretamente, como são definidos. Neste programa utilizamos o programa FAT.

Argumento de entrada: NPOLIN- grau de $L_n(x)$

Argumento de saída: A-arranjo A_1, A_2, ... A_{n+1}, coeficientes de $L_n(x)$

A_{n+1} - coeficiente de x^n
A_1 - termo independente

Posições requeridas: 1060 posições de memória.

Problema amostra: Determinou-se os coeficientes do polinômio $L_n(x)$ de grau 6.

Saída: A_1 = .72000000E+03
A_2 = -.43200000E+04
A_3 = .54000000E+04
A_4 = -.24000000E+04
A_5 = .45000000E+03
A_6 = -.36000000E+02
A_7 = .100000000E+01

Tempo de processamento: aproximadamente .5 seg.

Listagem Fortran: pág. II-5 - Apêndice II

IV-3 - PROGRAMA VPOL

Determinação de valor numérico de um polinômio $P_n(x)$

Argumentos de entrada: X - valor do ponto em que se quer calcular $P_n(x)$

N - grau de $P_n(x)$

A - arranjo A_1, A_2, ... A_{n+1} coeficientes de $P_n(x)$

Argumento de saída: VPOLIN - valor numérico de $P_n(x)$ para $x = X$

Posições requeridas: 864 posições de memória

Problema amostra: Determinou-se o valor numérico do polinômio $P(x) \equiv 5x^4+2x^3-x^2+x-2$ para $x=2$

Saída: .92000000E+02

Tempo de processamento: aproximadamente .5 seg.

Listagem Fortran: pág. II-5 - Apêndice II

IV-4 - PROGRAMA DPOL

Determinação do valor da derivada de um polinômio $P_n(x)$

Argumentos de entrada: X - valor do ponto em que se quer calcular $P_n'(x)$

N - grau de $P_n(x)$

A - arranjo A_1, A_2, ... A_{n+1}, coeficientes de $P_n(x)$

Argumento de saída: DPOLIN - derivada de $P_n(x)$ para $x=X$

Posições requeridas: 856 posições de memória

Problema amostra: Determinou-se o valor da derivada do polinômio

$$P(x) = 5x^4 + 2x^3 - x^2 + x - 2 \quad \text{no ponto } x=2$$

Saída: .18100000E+03

Tempo de processamento: aproximadamente .5seg.

Listagem Fortran: pág. II-6 - Apêndice II

IV-5 - PROGRAMA LAGG

Determinação do valor numérico do polinômio $L_n(x)$

Êste programa é adaptado do VPOL para o caso do polinômio de Laguerre. Utiliza o programa COLAG.

Argumentos de entrada:

X - valor do ponto em que se que calcular $L_n(x)$
NGRAP - Grau de $L_n(x)$

Argumentos de saída: POLLAG - Valor numérico de $L_n(x)$, para x=X

Posições requeridas: 996 posições de memória

Problema amostra: Determinou-se o valor numérico de $L_n(x)$ para x=1

Saída: -.40000000E+01

Tempo de processamento: aproximadamente .5 seg

Listagem Fortran: pág. II-6 - Apêndice II

IV-6 - PROGRAMA DELAG

Determinação do valor da derivada do polinômio $L_n(x)$

Êste programa é adaptado do DPOL para o caso do polinômio de Laguerre. Utiliza o programa COLAG.

Argumento de entrada:

X - Valor do ponto em que se quer calcular $L_n'(x)$.

NNGRAP - grau de $L_n(x)$

Argumento de saída: DEPOL - Derivada de $L_n(x)$ para x=X

Posições requeridas: 1042 posições de memória

Problema amostra: Determinou-se a derivada de $L_n(x)$ de grau 6 para x=2

Saída: DEPOL = .19200000E+03

Tempo de processamento: aproximadamente .5 seg

Listagem Fortran: pág. II-6 - Apêndice II

IV-7 - PROGRAMA DIPOL

Divisão de um polinômio racional e inteiro P(x) por um monômio ($x-\alpha$)

Êste programa determina Q(x) e R, respectivamente quociente e resto da divisão, aplicando a regra de Ruffini, como segue:

O coeficiente do primeiro têrmo de Q(x) é igualado ao coeficiente do primeiro têrmo de P(x). A partir do segundo têrmo obtêm-se os coeficientes de Q(x) somando-se o coeficiente do têrmo de mesma ordem de P(x) ao produto de α pelo coeficiente anterior de Q(x). O resto é obtido somando-se o têrmo independente de P(x) ao produto de α pelo têrmo independente de Q(x).

Argumentos de entrada:

NUGRAP - grau do polinômio P(x)

A - arranjo $A_1, A_2, \ldots A_{n+1}$, coeficientes de P(x)
ALFA - α

Argumentos de saída:

A - arranjo $A_1, A_2, \ldots A_n$, coeficientes de Q(x)
RES - Resto da divisão

Posições requeridas: 1262 posições de memória

Problema amostra: Efetuou-se a divisão do polinômio $P(x) \equiv x^4 + 2x^3 - x - 2$ por (x-1)

Saída: A_1 = .20000000E+01
A_2 = .30000000E+01
A_3 = .10000000E+01
Resto= .00000000E-99

Tempo de processamento: Aproximadamente .5 seg

Listagem Fortran: pág. II-7 - Apêndice II

IV-8 - PROGRAMA FAT

Determinação do fatorial de n

Argumento de entrada: N
Saída: FAT

Posições requeridas: 322 posições de memória

Listagem Fortran: pág. II-7 - Apêndice II

V - PROGRAMA GAMA

Determinação da função Gama

$$\Gamma(n) = \int_0^\infty e^{-x} \; x^{n-1} \, dx$$

Êste programa é adaptado do programa AINT (explicado anteriormente).

Neste programa o grau de $L_n(x)$ inicial é tomado como 10 e a precisão .1E-01.

Para se obter melhores resultados pode-se aumentar a precisão, mas o tempo de processamento aumenta consideravelmente neste caso.

A precisão que consideramos já nos dá um resultado razoável.

Argumentos de entrada: AM - argumento da função Gama

Saída: - GAMA

A função f(x) considerada é x^{n-1}

Posições requeridas: 1966 posições de memória

Problema amostra: Determinou-se Γ (1.5)

Saída: .88906848E+00

Tempo de processamento: 45 segundos

Listagem Fortran: pág. II-7 - Apêndice II

VI - PROGRAMA FBEJ

Cálculo da função de Bessel de 1a. classe e ordem ν $J_\nu(x)$ (ν=n, n inteiro, $.001 \leq x \leq 100$)

Argumentos de entrada: N - Ordem da função
X - Argumento da função

Saída: FBEJ

Posições requeridas: 4050 posições de memória

Problema amostra: Calculou-se $J_n(x)$
n=10
x=6

Saída - $J_{10}(6)$ = .69639820E-02

Tempo de processamento: Aproximadamente 2 seg

Listagem Fortran: pág. II-8 - Apêndice II

VII - PROGRAMA BEJN

Cálculo da função de Bessel de 1a. classe e ordem ν, $J_\nu(x)$ (ν fracionário qualquer $x > 0$)

Êste programa testa o valor de ν e x, e providencia a "chamada" de outros programas conforme sejam êstes valores. Se ν fôr da forma $\nu = m + 1/2$ ou $= -m - 1/2$, o programa "chamado" é BJMEIO.

Para ν diferente dêstes valores e $\nu < 2$ os programas utilizados serão FBSEJ ou FBJN, conforme seja $x \leq 6$ ou $x > 6$, respectivamente.

Para $\nu > 2$ o programa usado é FBSEJ, válido para valores não muito grandes de x.

Argumentos de entrada: AN - ordem da função
X - Argumento da função

Saída: BEJN

Posições requeridas: 724 posições de memória

Problemas - amostra: Foram calculados: $J_\nu(x)$

$\nu = .5$ $x = 8$

$\nu = -1.5$ $x = 6$

$\nu = -1/3$ $x = 4$

$\nu = 2/3$ $x = 10$

Saída:

$J_{.5}(8) = .27909280E+00$

$J_{-1.5}(6) = .38888561E-01$

$J_{-1/3}(4) = -.33308298E+00$

$J_{2/3}(10) = -.80149650E-01$

<u>Tempo de processamento</u>: 1º - aproximadamente .5 seg
2º - " .5 seg
3º - " 55 seg
4º - " 2.5 seg

<u>Diagrama-bloco</u>: pág. I-4 - Apêndice I

<u>Listagem Fortran</u>: pág. II-9 - Apêndice II

VIII - <u>PROGRAMA BJMEIO</u>

<u>Cálculo da função de Bessel de la. classe e ordem ν, $J_\nu(x)$</u>

($\nu = m + 1/2$, $\nu = -(m + 1/2)$, m inteiro positivo, $x \geqslant o$)

O programa testa ν verificando se é da forma $\nu = m+1/2$ ou $\nu = -(m + 1/2)$, e calcula $J_\nu(x)$ pela fórmula adequada ao caso. Êste programa utiliza o programa FAT.

<u>Argumentos de entrada</u>: AN - ordem da função
X - argumento da função

<u>Saída</u>: BJMEIO

<u>Posições requeridas</u>: 2768 posições de memória

<u>Listagem Fortran</u>: pág. II-9 - Apêndice II

IX - <u>PROGRAMA FBSEJ</u>

<u>Cálculo da função de Bessel de la. classe e ordem ν, $J_\nu(x)$, utilizando expansão em série de potências</u>

O programa começa com o número de têrmos da série igual a 10, depois vai acrescentando têrmos à série, e efetuando os cálculos até o ponto em que a contribuição dêstes têrmos seja mínima.

Êste programa utiliza os programas GAMA e FAT. Para diminuir o tempo de processamento utilizamos a função Γ (n) para um só valor de n; para outros valores necessários ao programa, usamos a relação de recorrência: Γ (n + 1) = n Γ (n)

Argumentos de entrada: AN - ordem da função
X - argumento da função

Saída: FBSEJ

Posições requeridas: 1844 posições de memória

Listagem Fortran: pág. II-10 - Apêndice II

X - PROGRAMA FBJN

Cálculo da função de Bessel de 1a. classe e ordem ν $J_\nu(x)$, usando expansão assintótica, ($x \gg 1$, $x \gg |\nu|^2$)

Argumentos de entrada: AN - ordem da função
X - argumento da função

Saída: FBJN

Posições requeridas: 1634 posições de memória

Listagem Fortran: pág. II-11 - Apêndice II

XI - PROGRAMA FBEY

Cálculo da função de Bessel de 2a. classe e ordem ν $Y_\nu(x)$ (ν = n inteiro, n $\leq$ 20, x > 0)

Se n=0 ou n=1 calculamos $Y_n(x)$ por meio do desenvolvimento em série de potências (no próximo programa) ou através do programa FBYN conforme seja x menor (ou igual) ou maior que 6 , respectivamente.

Para n > 1 são testados os valores de x, e de acôrdo com êstes valores $Y_n(x)$ é calculado por recorrência ou através do programa FBSEY.

Argumentos de entrada: N - ordem da função
X - argumento da função

Posições requeridas: 3144 posições de memória

Problema amostra: Determinou-se $Y_n(x)$

n = 0	x = 3
n = 1	x = 8
n = 3	x = 2
n = 5	x = 5

Saída:

$Y_0(3) = .37685001E+00$

$Y_1(8) = -.15806043 E+00$

$Y_3(2) = -.11277960E+01$

$Y_5(5) = -.45369501E+00$

Tempo de processamento:

1º -	aproximadamente	5 seg
2º -	"	2.5 seg
3º -	"	6 seg
4º -	"	8 seg

Diagrama-bloco: pág. I-5 - Apêndice I

Listagem Fórtran: pág. II-12 - Apêndice II

XII - PROGRAMA BEYN

Cálculo da função de Bessel de 2a. classe e ordem ν $Y_\nu(x)$ (ν fracionário qualquer, $x > 0$)

O programa testa o valor de ν; caso ν seja da forma $m + 1/2$ ou $-(m + 1/2)$ (m inteiro) calcula $Y_\nu(x)$ através do programa BJMEIO.

Para outros valores de ν, $Y_\nu(x)$ é calculado por intermédio do programa FBSEJ ou FBYN conforme seja $x \leq 6$ ou $x > 6$, res pectivamente.

Argumentos de entrada: AN - ordem da função
X - argumento da função

Saída: BEYN

Posições requeridas: 1160 posições de memória

Problema amostra: Determinou-se $Y_\nu(x)$

$\nu = -.5 \qquad x = 5$

$\nu = 1.5 \qquad x = 6$

$\nu = 2/3 \qquad x = 2$

$\nu = 1/3 \qquad x = 10$

Saída:

$Y_{1/2}(5) = -.34216797E+00$

$Y_{3/2}(6) = .38888561E-01$

$Y_{2/3}(2) = .11980132E+00$

$Y_{1/3}(10) = .17020107E+00$

Tempo de processamento: 1º - aproximadamente .5 seg
2º - " .5 seg
3º - " 1 m 50 seg
4º - " 2.5 seg

Diagrama-bloco: pág. I-6 - Apêndice I

Listagem Fortran: pág. II-13 - Apêndice II

XIII - PROGRAMA FBSEY

Cálculo da função de Bessel de 2a. classe e ordem ν $Y_\nu(x)$ ($\nu = n$ inteiro, $n > o$), usando desenvolvimento em série de potências.

O programa começa com o número de têrmos M = 20, vai efetuando os cálculos e acrescentando têrmos à série até o ponto em que a contribuição dêstes é mínima.

Êste programa utiliza os programas FBEJ e FAT.

Argumentos de entrada: N - ordem da função
X - argumento da função

Saída: FBSEY

Posições requeridas: 2474 posições de memória

Listagem Fortran: pág. II-13 - Apêndice II

XIV - PROGRAMA FBYN

Cálculo da função de Bessel de 2a. classe e ordem ν $Y_\nu(x)$ usando expansão assintótica ($x \gg 1$, $x \gg |\nu|^2$)

Argumentos de entrada: AN - ordem da função
X - argumento da função

Saída: FBYN

Posições requeridas: 1634 posições de memória

Listagem Fortran: pág. II-14 - Apêndice II

XV - PROGRAMA FBEI

Cálculo da função modificada de Bessel de 1a. classe, ordem ν $I_\nu(x)$ ($\nu = n$, n inteiro, $n \leq 20$, $x > o$)

Primeiramente é testado o valor de n, e de acôrdo com êste valor são "chamados" outros programas. Se n=0 ou n=1, os programas utilizados são FBIO e FBI1, respectivamente. Para $n > 1$, e $x \geq 12$ ou $x \geq n$ calculamos $I_n(x)$ utilizando a fórmula de recorrência, e para isso empregamos também os programas FBIO e FBI1.

Caso seja $x < n$ e $x < 12$, utilizamos o programa FBSEI.

Argumentos de entrada: N - ordem da função
X - argumento da função

Saída: FBEI

Posições requeridas: 1064 posições de memória

Problema amostra: Determinou-se $I_n(x)$

$n = 0 \qquad x = 3$

$n = 4 \qquad x = 3$

$n = 10 \qquad x = 11$

Saída:

$I_0(3) = .48807923E+01$

$I_4(3) = .32570511\ E+00$

$I_{10}(11) = .85053560E+02$

Tempo de processamento: 1º - aproximadamente 5 seg
2º - " 6 seg
3º - " 1.5 seg

Diagrama-bloco: pág. I-7 - Apêndice I

Listagem Fortran: pág. II-15 - Apêndice II

XVI - PROGRAMA BEIN

Cálculo da função modificada de Bessel de la. classe e ordem ν, $I_\nu(x)$ (ν fracionário qualquer, $x > 0$)

Êste programa testa o valor de ν e x e conforme sejam êstes valores, calcula ou "chama" outros programas para calcular $I_\nu(x)$.

Primeiramente é feito um teste para verificar se ν é da forma $m + 1/2$ ou $-(m + 1/2)$ (m inteiro). Caso seja, $I_\nu(x)$ é calculado no próprio programa. Para ν diferente dêsses valores, e $\nu < 2$, os programas utilizados são FBSEI ou FBIN, conforme seja $x \leq 6$ ou $x > 6$, respectivamente.

Para $\nu > 2$, o programa usado é FBSEI, válido para valores não muito grandes de x.

Argumentos de entrada: AN - ordem da função

X - argumento da função

Saída: BEIN

Posições requeridas: 2322 posições de memória

Problema amostra: Determinou-se $I_\nu(x)$

$\nu = .5 \qquad x = 6$

$\nu = -2.5 \qquad x = 8$

$\nu = 1/4 \qquad x = 3$

$\nu = -1/3 \qquad x = 10$

Saída: $I_{.5}(6) = .65704957E+02$

$I_{-2.5}(8) = .28249414E+03$

$I_{1/4}(3) = .48081380\ E+01$

$I_{-1/3}(10) = .27992395\ E+04$

Tempo de processamento: 1º - aproximadamente .5 seg

2º - " .5 seg

3º - " 55 seg

4º - " 1.5 seg

Diagrama-bloco: pág. I-8 - Apêndice I

Listagem Fortran: pág. II-15 - Apêndice II

XVII - PROGRAMA FBSEI

Cálculo da função modificada de Bessel de 1a. classe e ordem ν, $I_\nu(x)$, utilizando expansão em séries de potência

Analogamente ao programa FBSEJ, êste começa com número de têrmos da série igual a 10, depois vai acrescentando mais têrmos.

São válidas para o FBSEI as mesmas considerações feitas para o programa FBSEJ.

Argumentos de entrada: AN - ordem da função
X - argumento da função

Saída: - FBSEI

Posições requeridas: 1766 posições de memória

Listagem Fortran: pág. II-16 - Apêndice II

XVIII - PROGRAMA FBIN

Cálculo da função modificada de Bessel de la. classe e ordem ν , $I_\nu(x)$, usando expansão assintótica. $(x >> 1 , x >> |\nu|^2)$

Argumentos de entrada: AN - ordem da função
X - argumento da função

Saída: FBIN

Posições requeridas: 1446 posições de memória

Listagem Fortran: pág. II-17 - Apêndice II

XIX - PROGRAMA FBIO

Cálculo da função modificada de Bessel, de la. classe e ordem 0, usando expansão polinomial. $x > -3.75$

Argumento de entrada: X - argumento da função

Saída: FBIO

Posições requeridas: 1838 posições de memória

Listagem Fortran: pág. II-18 - Apêndice II

XX - PROGRAMA FBI1

Cálculo da função modificada de Bessel de 1a. classe e ordem 1, usando expansão polinomial ($x > -3.75$)

Argumento de entrada: X - argumento da função

Saída: FBI1

Posições requeridas: 1850 posições de memória

Listagem Fortran: pág. II-18- Apêndice II

XXI - PROGRAMA FBEK

Cálculo da função modificada de Bessel de 2a. classe e ordem ν ($\nu = n$, n inteiro, $n \leq 20$, $x > 0$)

Valem as mesmas considerações feitas para o programa FBEI

Argumentos de entrada: AN - ordem da função
X - argumento da função

Saída: FBEK

Posições requeridas: 1010 posições de memória

Problema amostra: Determinou-se $K_n(x)$

$n = 1 \quad x = 4$

$n = 3 \quad x = 4$

$n = 8 \quad x = 3$

Saída: $K_1(4) = .12483499$ E-01

$K_3(4) = .29884925$ E-01

$K_8(3) = .71867626$E+02

Tempo de processamento: 1º - aproximadamente .5 seg

2º - aproximadamente 1 seg
3º - " 6.5 seg

Diagrama-bloco: pág. I-9 - Apêndice I

Listagem Fortran: pág. II-19 - Apêndice II

XXII - PROGRAMA BEKN

Cálculo da função modificada de Bessel de 2a. classe e ordem ν, $K_\nu(x)$ (ν fracionário qualquer, $x > 0$)

Quando $\nu = m + 1/2$ ou $\nu = -(m + 1/2)$, $K_\nu(x)$ é calculado no próprio programa, utilizando a fórmula adequada.

Para outros valores de ν, $K_\nu(x)$ é calculado utilizando ou FBSEI ($x < 6$), ou FBKN ($x > 6$)

Argumentos de entrada: AN - ordem da função
X - argumento da função

Posições requeridas: 1648 posições de memória

Problema amostra: Determinou-se $K_\nu(x)$

$\nu = 1.5$ $x = 3$
$\nu = 1/3$ $x = 2$
$\nu = 1/3$ $x = 9$
$\nu = -2/2$ $x = 7$

Saída: $K_{1.5}(3) = .12008661E-01$
$K_{1/3}(2) = .11604125E+00$
$K_{1/3}(9) = .51180590E-04$
$K_{-2/3}(7) = .43762316E-03$

Tempo de processamento: 1º - aproximadamente .5 seg
2º - " 1 m 50 seg
3º - " 1.5 seg
4º - " 1.5 seg

Diagrama-bloco: pág. I-10 - Apêndice I

Listagem Fortran: pág.II-19 - Apêndice II

XXIII - PROGRAMA FBSEK

Cálculo da função modificada de Bessel de 2a. classe e ordem ν , $K_\nu(x)$ ($\nu = n$, inteiro, $n > 0$), usando desenvolvimento em série de potências

Para êste cálculo o número de têrmos da série primeiramente é tomado como 20. Depois vão sendo acrescentados têrmos à série, e feitos os cálculos correspondentes, até que a contribuição dêstes seja mínima.

Êste programa utiliza o programa FAT.

Argumentos de entrada: AN - ordem da função
X - argumento da função

Saída: FBSEK

Posições requeridas: 2508 posições de memória

Listagem Fortran: pág. II-20 - Apêndice II

XXIV - PROGRAMA FBKN

Cálculo da função modificada de Bessel, de 2a. classe e ordem ν $K_\nu(x)$, usando expansão assintótica. ($x \gg 1$, $x \gg |\nu|^2$)

Argumentos de entrada: AN - ordem da função
X - argumento da função

Saída: FBKN

Posições requeridas: 1308 posições de memória

Listagem Fortran: pág. II-21 - Apêndice II

XXV - PROGRAMA FBKO

Cálculo da função modificada de Bessel de 2a. classe e ordem 0, usando expansão polinomial (x > 0)

Argumento de entrada: X - argumento da função

Saída: FBKO

Posições requeridas: 1704 posições de memória

Listagem Fortran: pág. II-22 - Apêndice II

XXVI - PROGRAMA FBK1

Cálculo da função modificada de Bessel de 2a. classe, e ordem 1, usando expansão polinomial (x > 0)

Argumento de entrada: X - argumento da função

Saída: FBK1

Posições requeridas: 2028 posições de memória

Listagem Fortran: pág. II-22 - Apêndice II

XXVII - PROGRAMA FSENH

Determinação do seno hiperbólico

Argumento de entrada: X

Saída: FSENH

Posições requeridas: 274 posições de memória

Listagem Fortran: pág. II-23 - Apêndice II

XXVIII - PROGRAMA FCOSH

Determinação do cosseno hiperbólico

Argumento de entrada: X

Saída: FCOSH

Posições requeridas: 274 posições de memória

Listagem Fortran: pág. II-23 - Apêndice II

BIBLIOGRAFIA

(1) C. E. Grosch - Computation of Bessel Functions of integral order

(2) C. R. Wyllie, Jr - Advanced Engineering Mathematics

(3) Juan P. Arnaud - Teoria de las Telecomunicaciones

(4) Mc Lachlan - Bessel functions for engineers

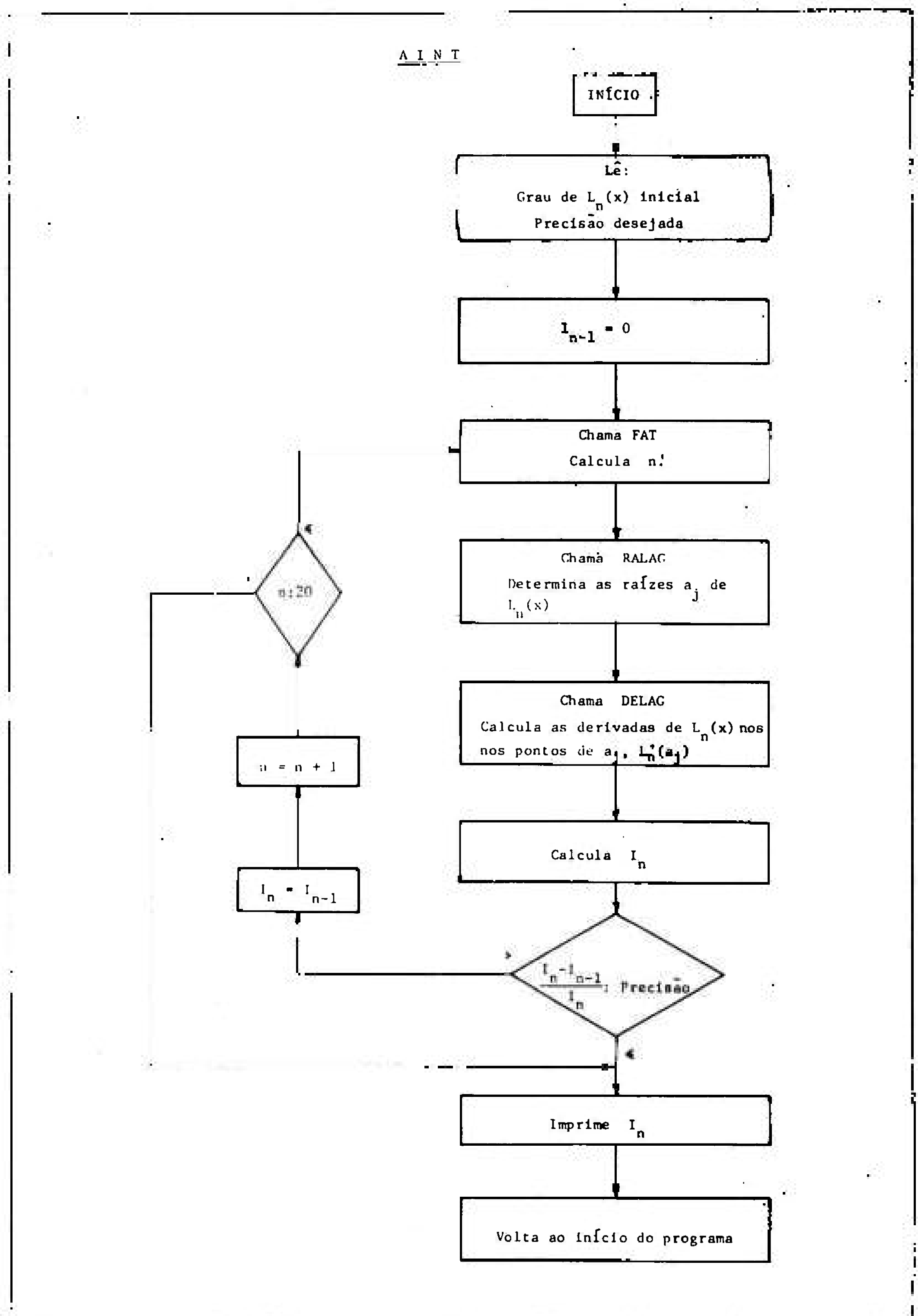
(5) Yudell L. Luke - Integrals of Bessel functions

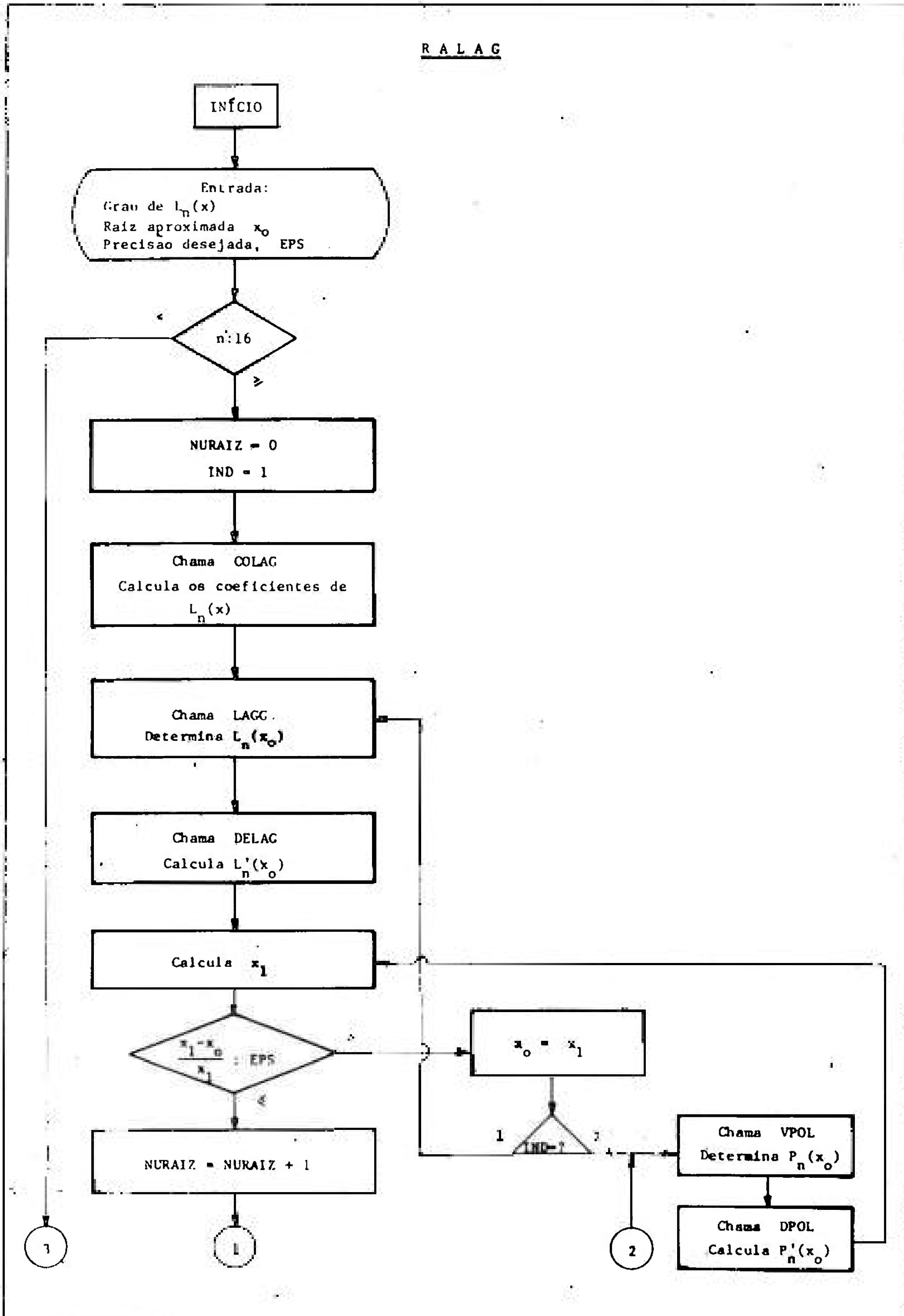
(6) Zdenek Kopal - Numerical Analysis

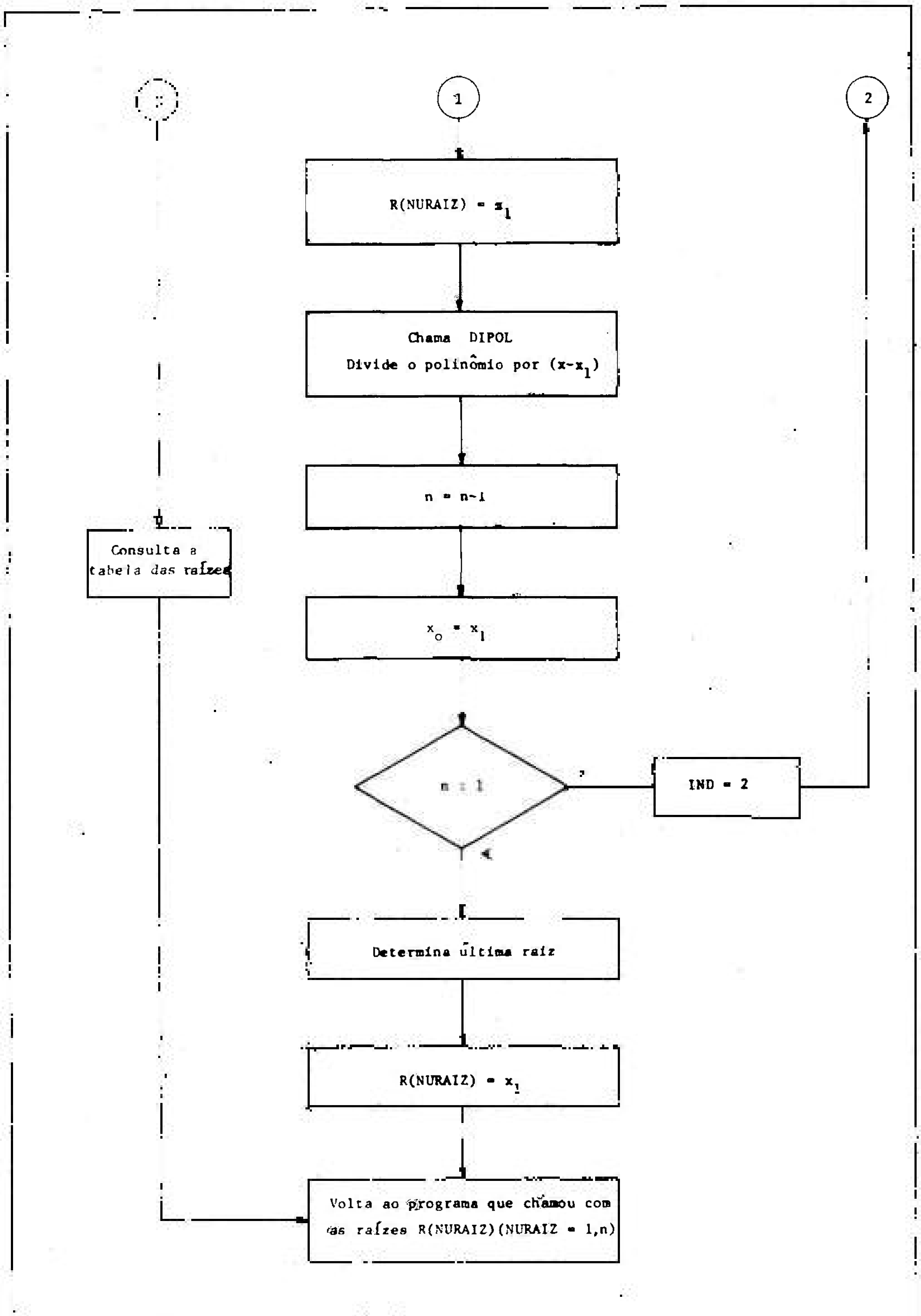
****

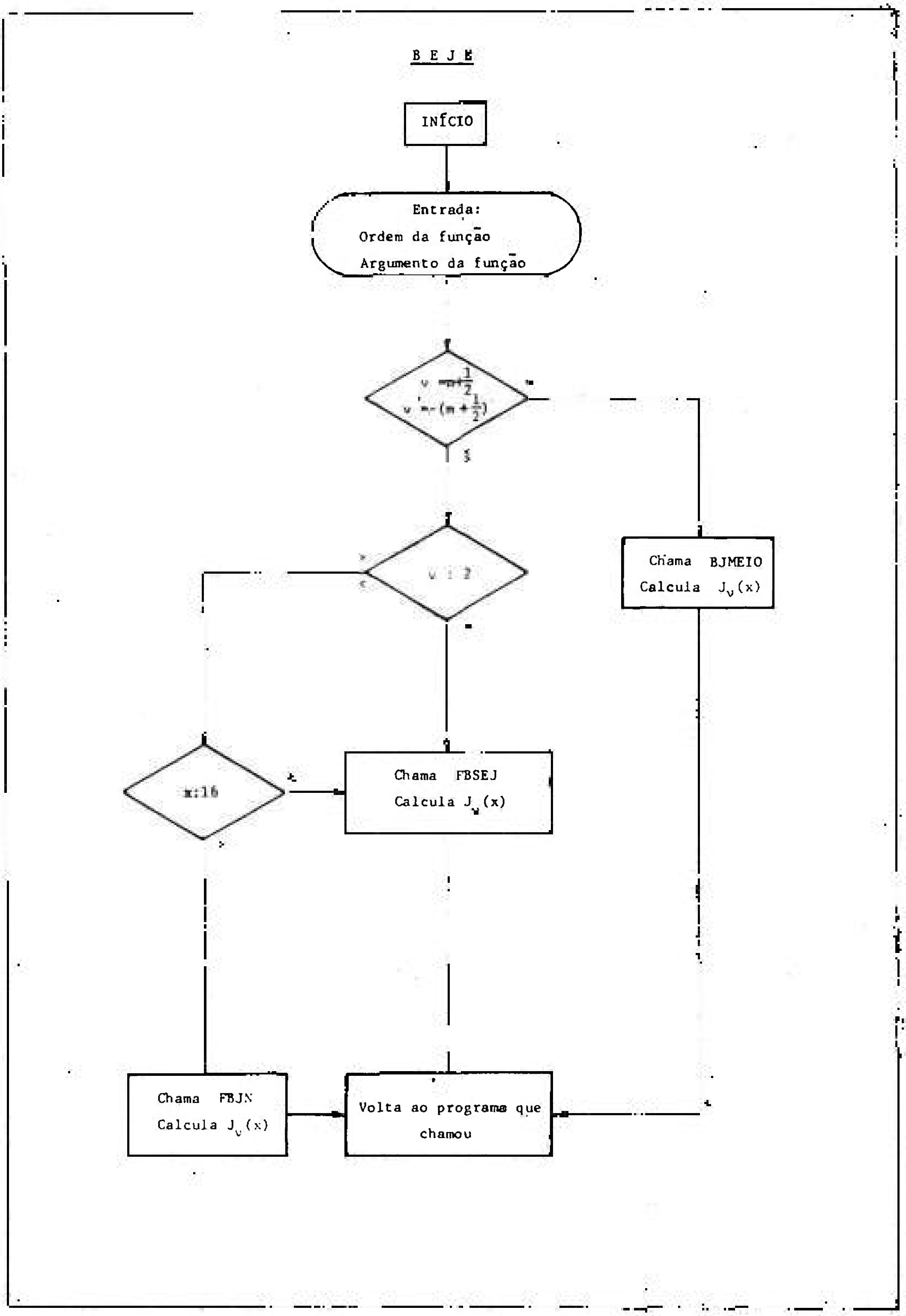
APÊNDICE I

DIAGRAMA-BLOCO

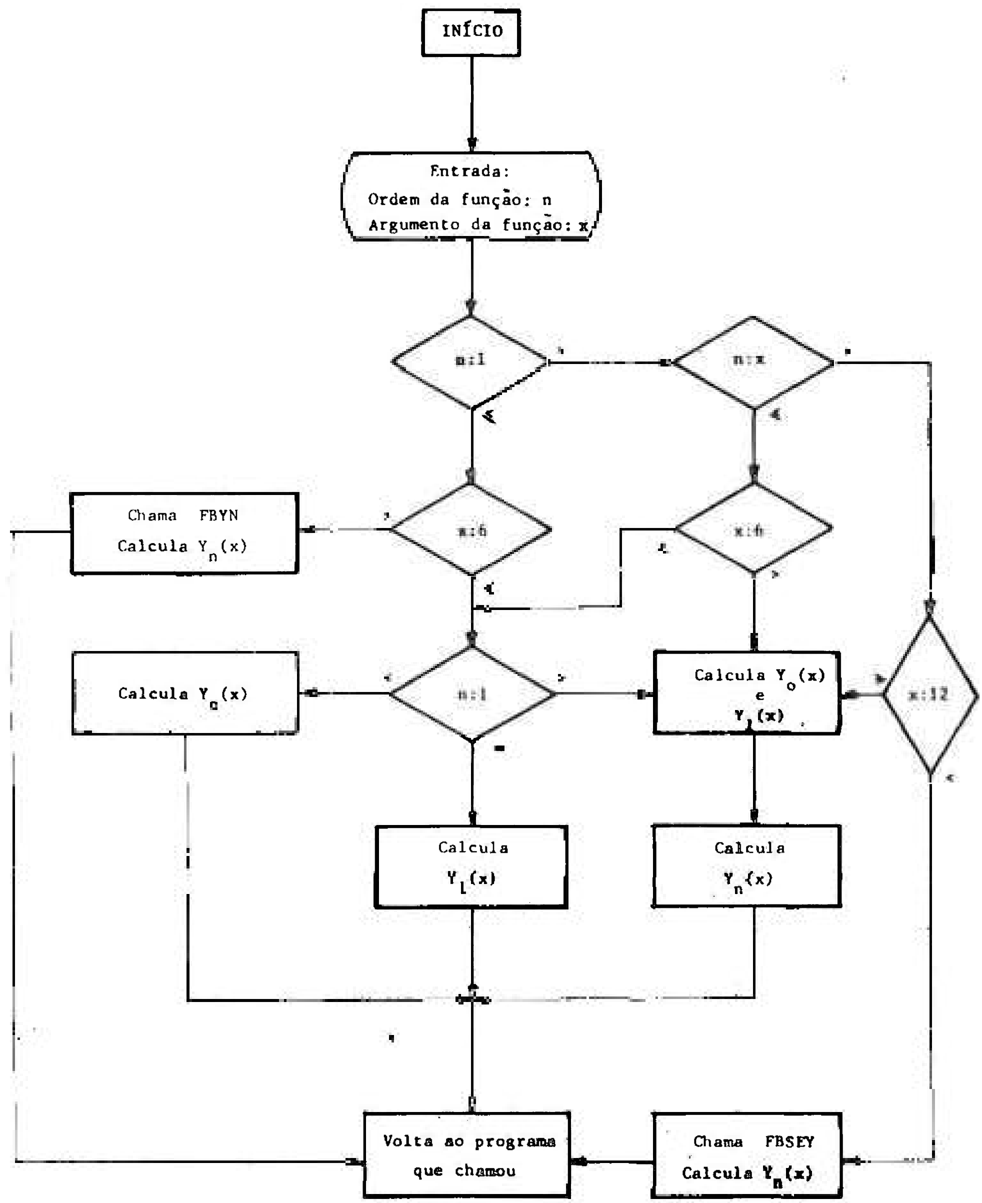

A I N T
INÍCIO
Lê:
Grau de $L_n(x)$ inicial
Precisão desejada
$I_{n-1} = 0$
Chama FAT
Calcula n!
Chama RALAG
Determina as raízes a_j de $L_n(x)$
Chama DELAG
Calcula as derivadas de $L_n(x)$ nos nos pontos de a_j, $L_n'(a_j)$
Calcula I_n
$\frac{I_n - I_{n-1}}{I_n}$: Precisão
$I_n = I_{n-1}$
n = n + 1
n:20
Imprime I_n
Volta ao início do programa


R A L A G
INÍCIO
Entrada:
Grau de $L_n(x)$
Raiz aproximada x_o
Precisao desejada, EPS
<
n:16
≥
NURAIZ = 0
IND = 1
Chama COLAG
Calcula os coeficientes de $L_n(x)$
Chama LAGG
Determina $L_n(x_o)$
Chama DELAG
Calcula $L'_n(x_o)$
Calcula x_1
$\frac{x_1-x_o}{x_1}$: EPS
>
≤
$x_o = x_1$
1
IND=?
2
Chama VPOL
Determina $P_n(x_o)$
Chama DPOL
Calcula $P'_n(x_o)$
NURAIZ = NURAIZ + 1
3
1
2


1
2
R(NURAIZ) = x_1
Chama DIPOL
Divide o polinômio por $(x-x_1)$
n = n-1
$x_o = x_1$
n : 1
IND = 2
Consulta a tabela das raízes
Determina última raiz
R(NURAIZ) = x_1
Volta ao programa que chamou com as raízes R(NURAIZ)(NURAIZ = 1,n)

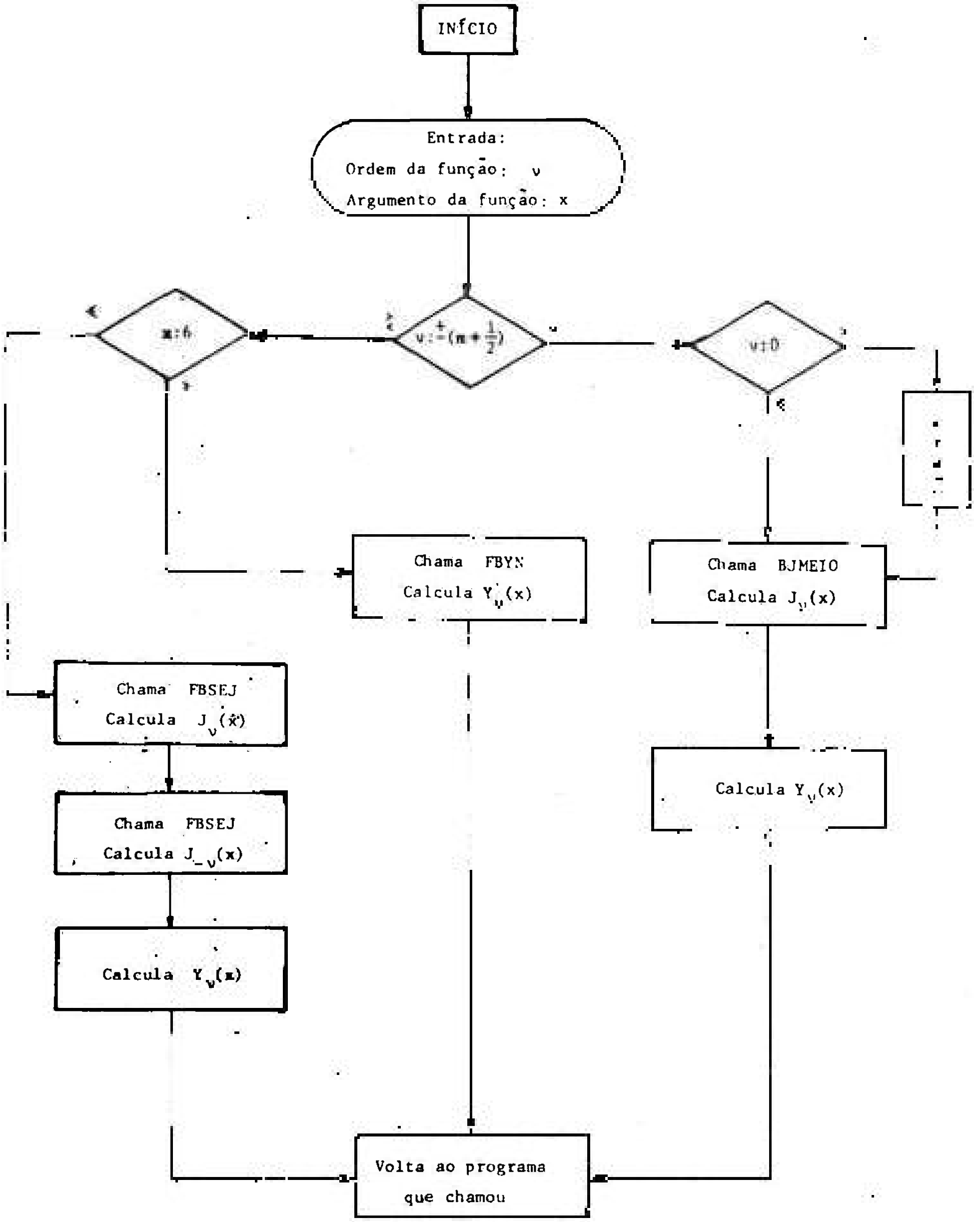

B E J E
INÍCIO
Entrada:
Ordem da função
Argumento da função
v = m + 1/2
v = -(m + 1/2)
v : 2
Chama BJMEIO
Calcula $J_v(x)$
m:16
Chama FBSEJ
Calcula $J_v(x)$
Chama FBJN
Calcula $J_v(x)$
Volta ao programa que chamou

F B E Y

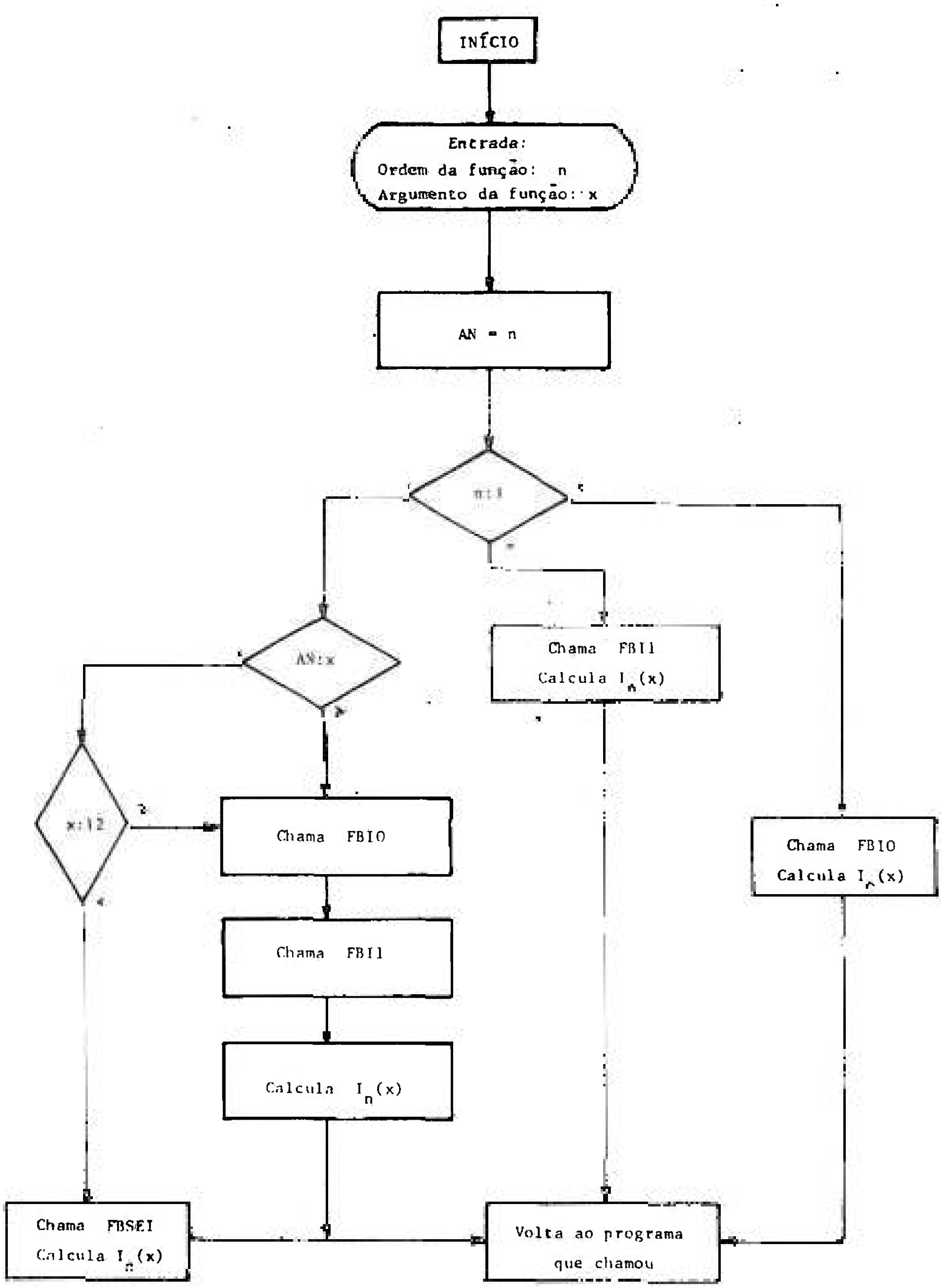


INÍCIO
Entrada:
Ordem da função: n
Argumento da função: x
n:1
n:x
Chama FBYN
Calcula $Y_n(x)$
x:6
x:6
Calcula $Y_o(x)$
n:1
Calcula $Y_o(x)$
e
$Y_1(x)$
x:12
Calcula
$Y_1(x)$
Calcula
$Y_n(x)$
Volta ao programa
que chamou
Chama FBSEY
Calcula $Y_n(x)$

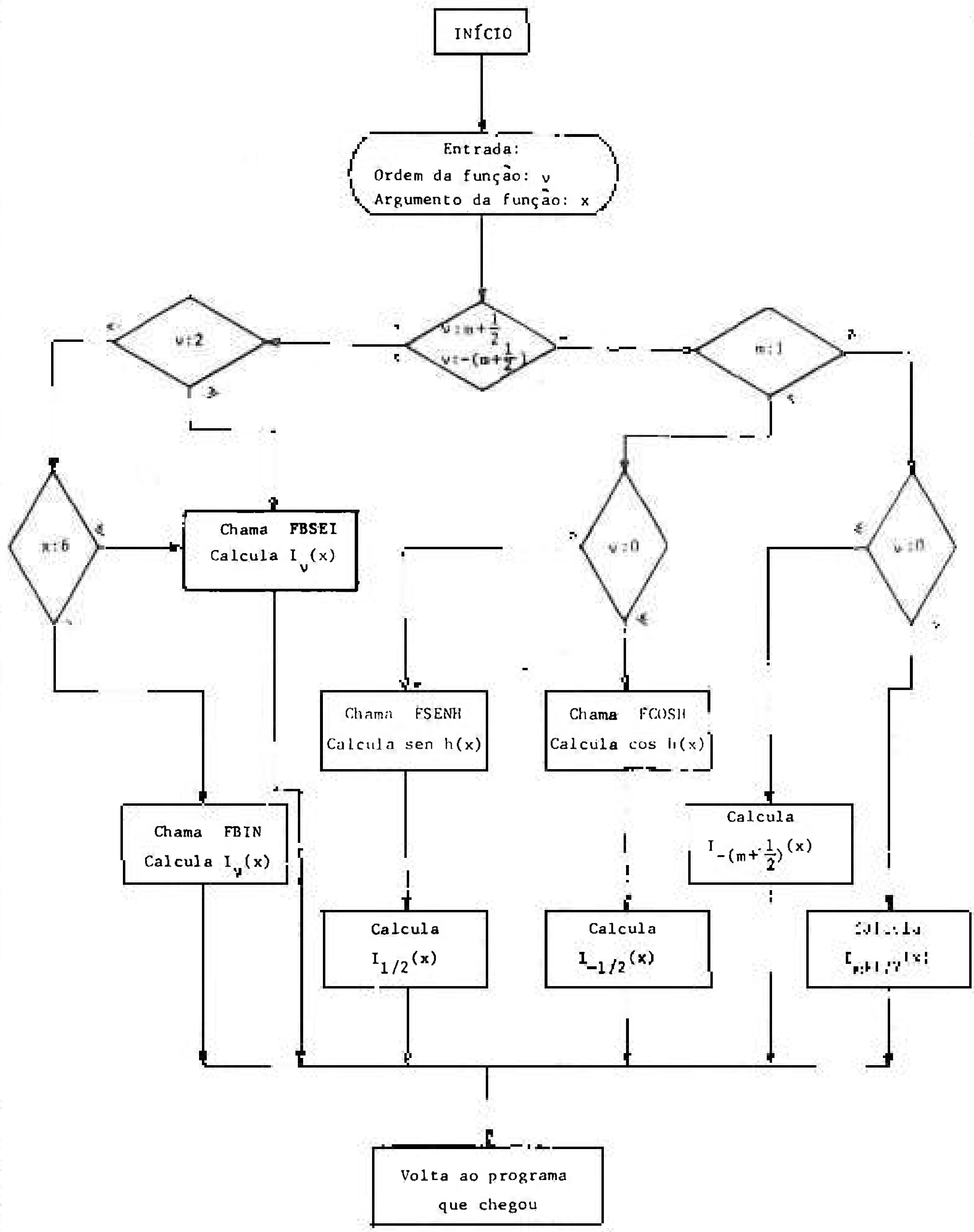
HFYN


INÍCIO
Entrada:
Ordem da função: ν
Argumento da função: x
Chama FBYN
Calcula $Y_\nu(x)$
Chama BJMEIO
Calcula $J_\nu(x)$
Calcula $Y_\nu(x)$
Chama FBSEJ
Calcula $J_\nu(x)$
Chama FBSEJ
Calcula $J_{-\nu}(x)$
Calcula $Y_\nu(x)$
Volta ao programa
que chamou

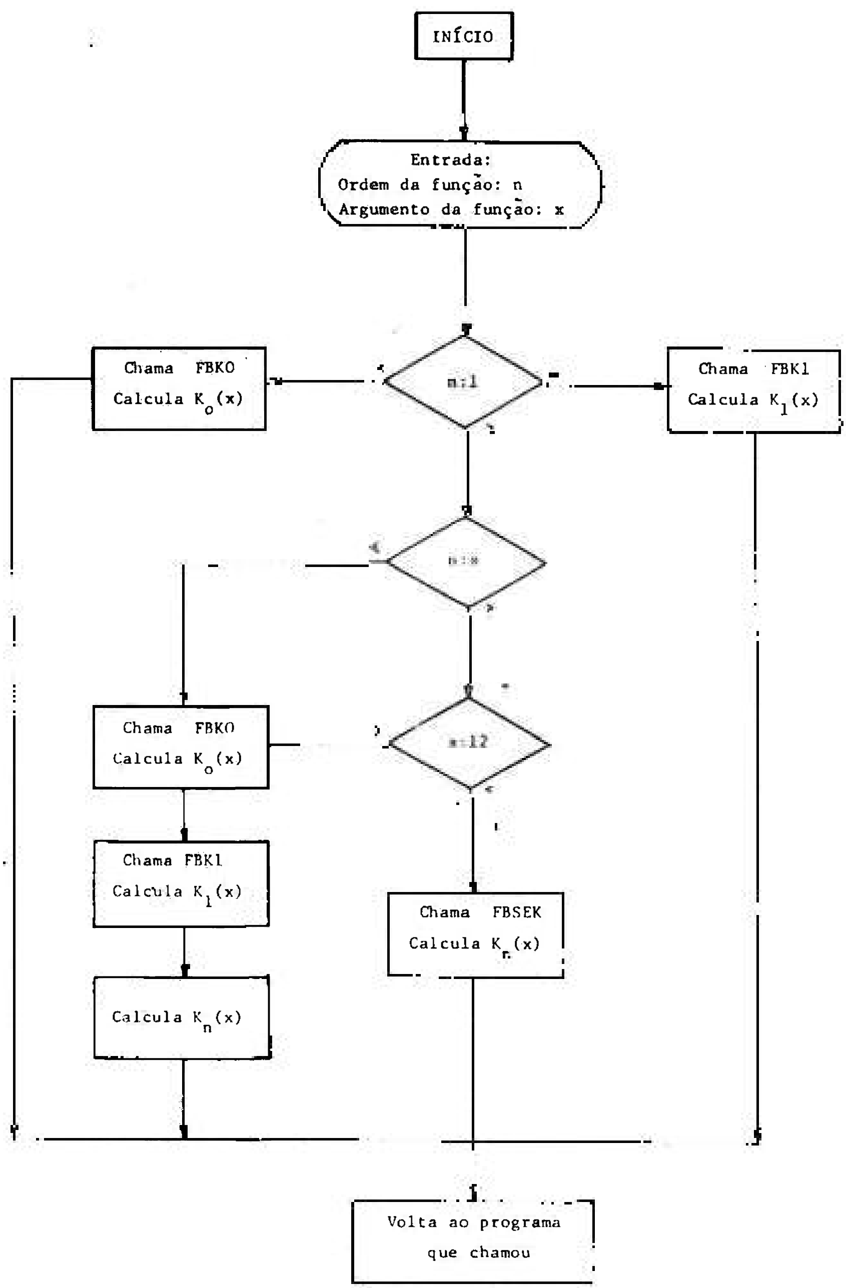
F B E I

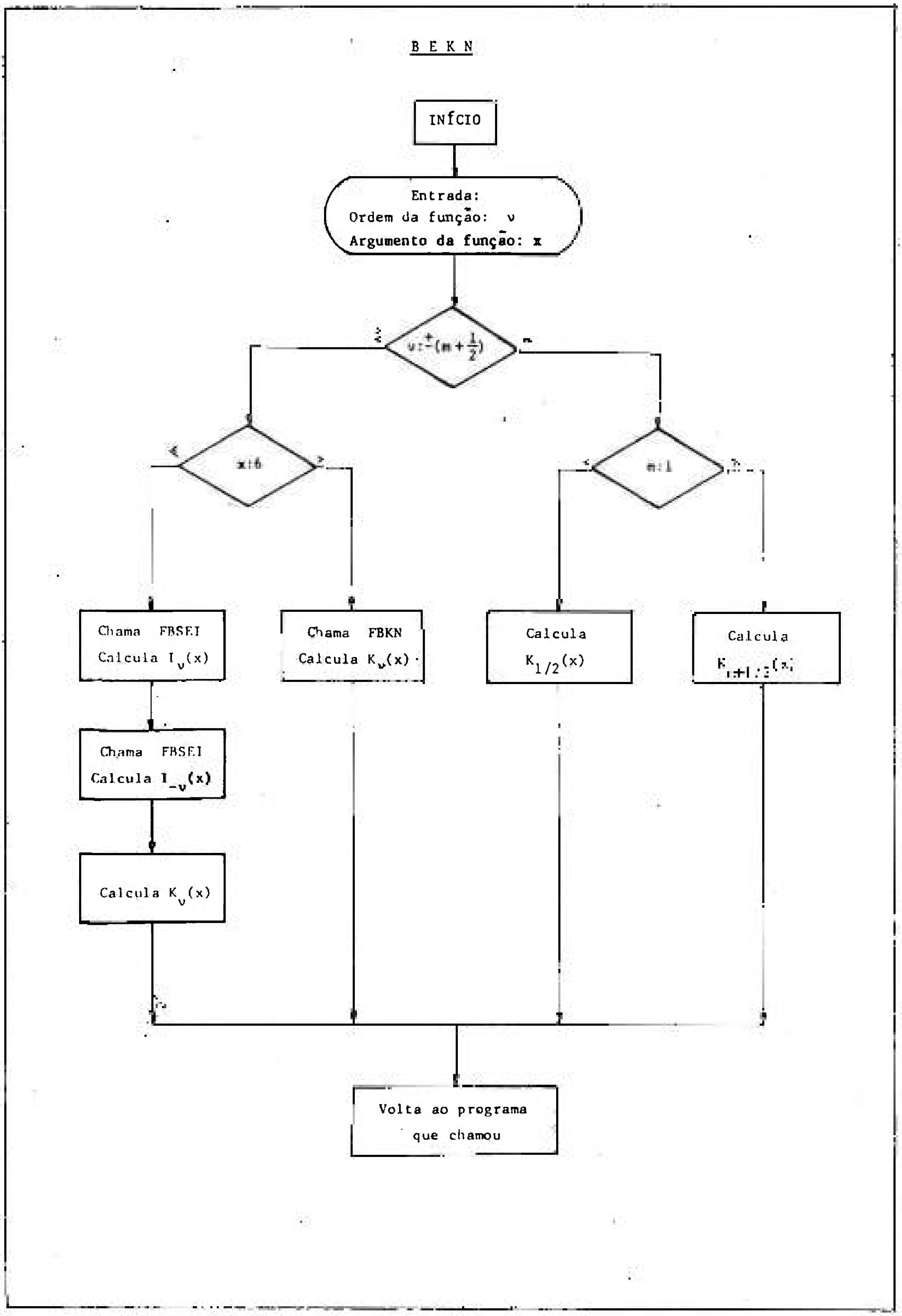

INÍCIO
Entrada:
Ordem da função: n
Argumento da função: x
AN = n
n:1
AN:x
Chama FBI1
Calcula $I_n(x)$
x:12
Chama FBIO
Chama FBI1
Calcula $I_n(x)$
Chama FBIO
Calcula $I_n(x)$
Chama FBSEI
Calcula $I_n(x)$
Volta ao programa
que chamou

B E I N


INÍCIO
Entrada:
Ordem da função: ν
Argumento da função: x
ν:2
Chama FBSEI
Calcula $I_\nu(x)$
Chama FSENH
Calcula sen h(x)
Chama FCOSH
Calcula cos h(x)
Chama FBIN
Calcula $I_\nu(x)$
Calcula
$I_{-(m+\frac{1}{2})}(x)$
Calcula
$I_{1/2}(x)$
Calcula
$I_{-1/2}(x)$
Volta ao programa
que chegou

F B E K


INÍCIO
Entrada:
Ordem da função: n
Argumento da função: x
n:1
Chama FBKO
Calcula $K_o(x)$
Chama FBK1
Calcula $K_1(x)$
n:12
Chama FBKO
Calcula $K_o(x)$
Chama FBK1
Calcula $K_1(x)$
Calcula $K_n(x)$
Chama FBSEK
Calcula $K_n(x)$
Volta ao programa
que chamou


B E K N
INÍCIO
Entrada:
Ordem da função: ν
Argumento da função: x
$\nu : \pm(m+\frac{1}{2})$
≠
=
x:6
<
>
m:1
<
≥
Chama FBSEI
Calcula $I_\nu(x)$
Chama FBKN
Calcula $K_\nu(x)$
Calcula
$K_{1/2}(x)$
Calcula
$K_{i+1/2}(x)$
Chama FBSEI
Calcula $I_{-\nu}(x)$
Calcula $K_\nu(x)$
Volta ao programa
que chamou

A P Ê N D I C E   II

LISTAGEM FORTRAN

AINT

```
C       PROGRAMA AINT
        DIMENSION R(20)
  120   READ 1, EM, N
        AINT2=0.
        FATAN=FAT(N-1)
  505   AN=N
        AINT=0
        FATAN=AN*FATAN
        N=AN
        CALL RALAG(N, 0., EM, R)
        DO 900 NURAIZ=1, N
        X=R(NURAIZ)
        CALL DELAG(X,N,DEPOL)
  900   AINT=FUNC(R(NURAIZ))/(R(NURAIZ)*DEPOL*DEPOL)+AI
           NT
        AINT=FATAN*FATAN*AINT
        TESTE=ABS(AINT-AINT2)/ABS(AINT)
        IF(TESTE-EM) 821, 821, 822
  822   AINT2=AINT
        N=N+1
        IF(N-20) 505, 505, 506
  506   TYPE 2, TESTE
        GO TO 120
    2   FORMAT (64HPRECISAO DESEJADA NAO ATINGIDA (DENT
           RO DOS LIMITES DO P
       1ROGRAMA)./26HINTEGRAL CALCULADA COM EM=E14.8)
  821   TYPE 1, AINT
    1   FORMAT (E14.8)
        GO TO 120
        END
```

****

RALAG

```
        SUBROUTINE RALAG (NEGRAP, X0, EPS, R)
        DIMENSION A(21), R(20)
        IF(NEGRAP-15) 160, 160,86
  160   GO TO (71,72,73,74,75,76,77,78,79,80,81,82,83,8
           4,85,86),NEGRAP
   71   R(1)=1.
        RETURN
   72   R(1)=.58578643
        R(2)=3.4142136
```

```
      RETURN
   73 R(1)=.41577456
      R(2)=2.2942804
      R(3)=6.2899451
      RETURN
   74 R(1)=.32254769
      R(2)=1.7457611
      R(3)=4.5366203
      R(4)=9.3950709
      RETURN
   75 R(1)=.26356032
      R(2)=1.4134031
      R(3)=3.5964258
      R(4)=7.0858100
      R(5)=12.640801
      RETURN
   76 R(1)=.22284660
      R(2)= 1.1889321
      R(3)=2.9927363
      R(4)=5.7751436
      R(5)=9.8374674
      R(6)=15.982874
      RETURN
   77 R(1)=.19304368
      R(2)=1.0266649
      R(3)=2.5678767
      R(4)=4.9003531
      R(5)=8.1821534
      R(6)=12.734180
      R(7)= 19.395728
      RETURN
   78 R(1)=.17027963
      R(2)=.90370178
      R(3)= 2.2510866
      R(4)=4.2667002
      R(5)=7.0459054
      R(6)=10.758516
      R(7)=15.740679
      R(8)=22.863132
      RETURN
   79 R(1)=.15232223
      R(2)=.80722002
      R(3)=2.0051352
      R(4)=3.7834740
      R(5)=6.2049568
      R(6)=9.3729853
      R(7)=13.466237
```

```
      R(8)=18.833598
      R(9)=26.374072
      RETURN.
80    R(1)= .13779347
      R(2)= .72945455
      R(3)=1.8083429
      R(4)=3.4014337
      R(5)=5.5524961
      R(6)=8.3301527
      R(7)=11.843786
      R(8)=16.279258
      R(9)=21.996586
      R(10)=29.920698
      RETURN
81    R(1)=.12579644
      R(2)=.66541826
      R(3)=1.6471505
      R(4)=3.0911381
      R(5)=5.0292844
      R(6)=7.5098879
      R(7)=10.605951
      R(8)=14.431614
      R(9)=19.178857
      R(10)=25.217709
      R(11)=33.497193
      RETURN
82    R(1)=.11572212
      R(2)=.61175748
      R(3)=1.5126103
      R(4)=2.8337513
      R(5)=4.5992276
      R(6)=6.8445254
      R(7)=9.6213168
      R(8)=13.006055
      R(9)=18.116855
      R(10)=22.151090
      R(11)=28.487967
      R(12)=37.099121
      RETURN
83    R(1)=.10714239
      R(2)=.56613190
      R(3)=1.3985643
      R(4)=2.6165971
      R(5)=4.2388459
      R(6)=6.2922563
      R(7)=8.8150019
      R(8)=11.861404
```

```
      R(9)=15.510762
      R(10)=19.884636
      R(11)=25.185264
      R(12)=31.800386
      R(13)=40.723009
      RETURN
   84 R(1)=.099747507
      R(2)=.52685765
      R(3)=1.3006291
      R(4)=2.4308011
      R(5)=3.9321028
      R(6)=5.8255362
      R(7)=8.1402401
      R(8)=10.916499
      R(9)=14.210805
      R(10)=18.104892
      R(11)=22.723382
      R(12)=28.272982
      R(13)=35.149444
      R(14)=44.366082
      RETURN
   85 R(1)=.093307812
      R(2)=.49269174
      R(3)=1.2155954
      R(4)=2.2699495
      R(5)=3.6676227
      R(6)=5.4253366
      R(7)=7.5659162
      R(8)=10.120229
      R(9)=13.130282
      R(10)=16.654408
      R(11)=20.776479
      R(12)=25.623894
      R(13)=31.407519
      R(14)=38.530683
      R(15)=48.026086
      RETURN
   86 NN=NEGRAP
      NURAIZ=0
      AUX=X0
      IND=1
      CALL COLAG (NEGRAP, A)
  320 CALL LAGG (X0, NEGRAP, POLLAG)
      VPOLIN=POLLAG
      GO TO 60
   20 CALL VPOL (X0, NEGRAP, A, VPOLIN)
      GO TO 70
```

```
 60 CALL DELAG (X0, NEGRAP, DEPOL)
    DPOLIN=DEPOL
    GO TO 90
 70 CALL DPOL (X0, NEGRAP, A, DPOLIN)
 90 X1=X0-VPOLIN/DPOLIN
    IF(ABS(X1-X0)/ABS(X1)-EPS) 201,201,200
200 X0=X1
    GO TO (320, 20), IND
201 NURAIZ=NURAIZ+1
    R(NURAIZ)=X1
    CALL DIPOL(NEGRAP,A,X1,RES)
    NEGRAP=NEGRAP-1
    IF(NEGRAP-1) 680, 680, 206
206 IND=2
    GO TO 20
680 X1=-A(1)/A(2)
    NURAIZ=NURAIZ+1
    R(NURAIZ)=X1
    NEGRAP=NN
    X0=AUX
    RETURN
    END
```

****

COLAG

```
    SUBROUTINE COLAG (NPOLIN,A)
    DIMENSION A(21)
    AUGRAP=NPOLIN
    A(1)=FAT(NPOLIN)
    A(2)=-AUGRAP*A(1)
    FATJ=1
    DO 20 J=2, NPOLIN
    AJ=J
    FATJ=AJ*FATJ
 20 A(J+1)=A(1)*A(1)/(FAT(NPOLIN-J)*FATJ)*((-1.)**J
        )/FATJ
    RETURN
    END
```

****

VPOL

```
    SUBROUTINE VPOL (X, N, A, VPOLIN)
    F(B, J, Y)=B*Y**J
    DIMENSION A(21)
```

```
      VPOLIN =A(2)*X+A(1)
      DO 20 J=2, N
   20 VPOLIN =F(A(J+1), J, X)+VPOLIN
      RETURN
      END
```

****

DPOL

```
      SUBROUTINE DPOL (X, N, A, DPOLIN)
      DF(B, Y)=AK*B*Y**(K-1)
      DIMENSION A(21)
      DPOLIN=A(2)
      DO 25 K=2,N
      AK=K
   25 DPOLIN  =DF(A(K+1), X)+DPOLIN
      RETURN
      END
```

****

LAGG

```
      SUBROUTINE LAGG (X,  NGRAP, POLLAG)
      F(B, J, Y)=B*Y**J
      DIMENSION A(21)
      CALL COLAG( NGRAP, A)
      POLLAG  =A(2)*X+A(1)
      DO 20 J=2, NGRAP
   20 POLLAG =F(A(J+1),J,X)+POLLAG
      RETURN
      END
```

****

DELAG

```
      SUBROUTINE DELAG(X, NNGRAP, DEPOL)
      DF(B,Y)=AK*B*Y**(K-1)
      DIMENSION A(21)
      CALL COLAG (NNGRAP, A)
      DEPOL=A(2)
      DO 25 K=2, NNGRAP
      AK=K
   25 DEPOL=DF(A(K+1),X)+DEPOL
      RETURN
      END
```

DIPOL

```
      SUBROUTINE DIPOL(NUGRAP,A,ALFA,RES)
      DIMENSION A(21), B(21)
      B(NUGRAP)=A(NUGRAP+1)
      N1=NUGRAP-1
      DO 100 J=1, N1
      NJ=NUGRAP-J
      NJ1=NJ+1
      B(NJ)=B(NJ1)*ALFA+A(NJ1)
  100 A(NJ1)=B(NJ1)
      RES=B(1)*ALFA+A(1)
      A(1)=B(1)
      RETURN
      END
```

****

FAT

```
      FUNCTION FAT(N)
      FAT=1.
      DO 10  I=1,N
      AI=I
   10 FAT=AI*FAT
      RETURN
      END
```

****

GAMA

```
      FUNCTION GAMA(AM)
      FUM(X,HM)=X**(HM-1.)
      DIMENSION R(20)
      M=AM
      EM=M
      IF(AM-EM) 700, 999, 700
  999 GAMA=FAT(M-1)
      RETURN
  700 EM=.1E-01
      N=10
      AINT2=0.
      FATAN=FAT(N-1)
  505 AN=N
      AINT=0
```

```
      FATAN=AN*FATAN
      N=AN
      CALL RALAG(N, 0., EM, R)
      DO 900 NURAIZ=1, N
      X=R(NURAIZ)
      CALL DELAG (X, N, DEPOL)
  900 AINT=FUM(R(NURAIZ),AM)/(R(NURAIZ)*DEPOL*DEPOL)+
     .   AINT
      AINT=FATAN*FATAN*AINT
      IF(ABS(AINT-AINT2)/ABS(AINT)-EM) 821, 821, 822
  822 AINT2=AINT
      N=N+1
      GO TO 505
  821 GAMA=AINT
      RETURN
      END
```

****

FBEJ

```
      FUNCTION FBEJ( N,X)
      DIMENSION C(140)
      SOMA=0
      DO 3 I=1, 140
    3 C(I)=0.
      IF(X-10.)4, 4, 5
    4 NN=(35.0/(3.5-LOG(X)))+1.
      GO TO 6
    5 NN=1.05*X+26.0
    6 IEND=NN+2
      FLN=NN
      F=X/FLN
      A=SQRT(1.-F*F)
      A2=(F**FLN)*((FLN*A)**(-.5))*((1.+A)**(-FLN))
      C(2)=.3989240*A2*EXP(FLN*A)
      DO 7 I=1,NN
      M=NN-I+1
      FLM=M
    7 C(I+2)=FLM*2./X*C(I+1)-C(I)
      D=.5*FLN
      J=D
      FLJ=J
      IF(D-FLJ) 8,8,9
    8 K=2
      GO TO 30
    9 K=1
   30 DO 31 I=K,NN,2
```

```
 31 SOMA=SOMA+C(I)
    DO 32 I=K, IEND, 2
 32 SOMA=SOMA+C(I)
    I=0
 33 I=I+1
    C(I)=1./SOMA*C(I)
    NP=NN-I+2
    FBEJ=C(I)
    IF(N-NP) 33, 34, 34
 34 RETURN
    END
```

****

BEJN

```
    FUNCTION BEJN(AN, X)
    N=AN
    EN=N
    DIF=AN-EN
    IF(ABS(DIF)-.5) 666, 92, 666
666 IF(AN-2.) 640, 690, 690
640 IF(X-6.) 690, 690, 830
830 BEJN=FBJN(AN,X)
    RETURN
690 BEJN=FBSEJ(AN,X)
    RETURN
 92 BEJN=BJMEIO(AN,X)
    RETURN
    END
```

****

BJMEIO

```
    FUNCTION BJMEIO.(AN,X)
    R2PIX=SQRT(2./(.31415926E+01*X))
    N=ABS(AN)
    HN=N
    XV2=2.*X
 93 IF(N-1) 300, 301, 302
302 JJ=N/2
    SOMAT1=1
    DO 100 K=1, JJ
    KV2=2*K
100 SOMAT1=(-1.)**K*FAT(N+KV2)/(FAT(KV2)*FAT(N-KV2)
       *(XV2 )**(KV2))+SOM
```

```
    1AT1
     SOMAT2=FAT(N+1)/(FAT(N-1)*XV2)
     II=(N-1)/2
     IF(II) 650, 650, 651
651  DO 200 K=1, II
     KV2=2*K
200  SOMAT2=(-1.)**K*FAT(N+KV2+1)/(FAT(KV2+1)*FAT(N-
        KV2-1)*(XV2 )**(KV2
    1+1))+SOMAT2
650  IF(AN) 800, 800, 801
800  BJMEIO=R2PIX*(COS(X+HN*1.5707960)*SOMAT1-SIN(X+
        HN*1.5707960)*SOMAT
    12)
     RETURN
801  BJMEIO=R2PIX*(SIN(X-HN*1.5707960)*SOMAT1+COS(X-
        HN*1.5707960)*SOMAT
    12)
     RETURN
300  IF(AN) 400, 400, 401
400  BJMEIO=R2PIX*COS(X)
     RETURN
401  BJMEIO=R2PIX*SIN(X)
     RETURN
301  IF(AN) 500, 500, 501
500  BJMEIO= -R2PIX*(SIN(X)+ COS(X)/X)
     RETURN
501  BJMEIO=R2PIX*(SIN(X)/X-COS(X))
     RETURN
     END
```

****

FBSEJ

```
     FUNCTION FBSEJ(AN,X)
     M=10
     FBEJ=0
     AJN=0
     L=1
     AM=M
     GAMAN=GAMA(AN+AM+2.)
     GAMAN1=GAMAN
     FATM=FAT(M)
     FATK=FATM
600  DO 777 I=L, M
     K=M-(I-L)
     AK=K
     GAMAN=GAMAN/(AK+AN+1.)
     IF(K-1) 888, 940,888
940  GAMAL1= GAMAN/(AN+1.)
```

```
 888 AJN=(-1.)**K*(X**(AN+2.*AK)/(2.**(AN+2.*AK)*FAT
        K*GAMAN))+AJN
 777 FATK=FATK/AK
     IF(FBEJ-AJN) 110, 109, 110
 110 FBEJ=AJN
     L=M+1
     M=M+2
     AM=M
     FATK=FATM*(AM-1.)*AM
     FATM=FATK
     GAMAN=GAMAN1*(AN+AM)
     GAMAN1=GAMAN
     GO TO 600
 109 AJ0=X**AN/(2.**AN*GAMAL1)
     FBSEJ=AJN+AJ0
1000 RETURN
     END
```

****

FBJN

```
     FUNCTION FBJN(AN,X)
     QUAN2=4.*AN*AN
     OITOX=8.*X
     R2PIX=SQRT(2./(.31415926E+01*X))
     ALFA=X-.78539816-.15707963E+01*AN
     A2=1
     N=X+1.
     NN=4*N-3
     AL=1
     FBJN=0
     J=0
     DO 600 I=1,NN,4
     AI=I
     A1=A2*(QUAN2-AI*AI)/(AL*OITOX)
     A2=A1*(QUAN2-(AI+2.)*(AI+2.))/((AL+1.)*OITOX)
     FBJN=(-1.)**(J+1)*A2*COS(ALFA)-(-1.)**J*A1*SIN(
        ALFA)+FBJN
     J=J+1
 600 AL=AL+2.
     FBJN=R2PIX*(FBJN+COS(ALFA))
     RETURN
     END
```

****

FBEY

```
      FUNCTION FBEY (N,X)
      AUX=0
      AN=N
      IF(N-1) 333, 333, 335
  333 IF(X-6.) 690, 690, 580
  580 FBEY=FBYN(AN,X)
      RETURN
  690 SOMATO=0
      SOMAT1=0
      L=1
      M=20
      XS2=X/2.
      FLOG=LOG(XS2)
      SOMO=0
      FATOR1=.63661978E+00*(.57721566+FLOG)
      FATK=1
  600 DO 900 K=L,M
      AK=K
      SOMO=1./AK+SOMO
      SOM1=2.*SOMO+1./(AK+1.)
      FATK=AK*FATK
      FATOR2=(-1.)**K*XS2**(2*K)/(FATK**2)
      SOMATO=FATOR2*SOMO+SOMATO
  900 SOMAT1=FATOR2*XS2/(AK+1.)*SOM1+SOMAT1
      IF(AUX-SOMAT1) 110, 109, 110
  110 AUX=SOMAT1
      L=M+1
      M=M+2
      GO TO 600
  109 SOMAT1=SOMAT1+XS2
      IF(N-1) 230, 231, 232
  230 FY0 =FBEJ(0, X)*FATOR1-.63661978E+00*SOMATO
      FBEY=FY0
      RETURN
  231 FY1 =FBEJ(1, X)*FATOR1-1./3.141592*SOMAT1-.6366
         1978E+00*1./X
      FBEY=FY1
      RETURN
  232 FY0 =FBEJ(0, X)*FATOR1-.63661978E+00*SOMATO
      FY1 =FBEJ(1, X)*FATOR1-1./3.141592*SOMAT1-.6366
         1978E+00*1./X
      GO TO 700
  335 IF(AN-X) 520, 520, 519
  519 IF(X-12.) 300, 850, 850
  520 IF(X-6.) 690, 690, 850
  300 FBEY=FBSEY( N,X)
```

```
      RETURN
  850 FY0=FBYN(0.,X)
      FY1=FBYN(1.,X)
  700 DO 980 I=2,N
      AI=I
      FY2=2.*(AI-1.)/X*FY1-FY0
      FY0=FY1
  980 FY1=FY2
      FBEY=FY2
      RETURN
      END
```

****

BEYN

```
      FUNCTION BEYN(AN,X)
      N=AN
      EN=N
      DIF=AN-EN
      IF(ABS(DIF)-.5) 666,92, 666
  666 IF(X-6.) 690, 690, 830
  830 BEYN=FBYN(AN,X)
      RETURN
  690 ALFA=AN*.31415926E+01
      FPOS=FBSEJ(AN,X)
      BN=-AN
      FNEG=FBSEJ(BN,X)
      BEYN=(FPOS*COS(ALFA)-FNEG)/SIN(ALFA)
      RETURN
   92 N=ABS(AN)
      IF(AN) 600, 600, 200
  200 N=N+1
  600 BEYN=(-1.)**N*BJMEIO(-AN,X)
      RETURN
      END
```

****

FBSEY

```
      FUNCTION FBSEY(N,X)
      AUX=0
      L=1
      M=20
      AN=N
      XS2=X/2.
```

```
      FLOG=LOG(XS2)
      SOMO=0.
      FATOR1=.63661978E+00*( .57721566+FLOG)
      FATK=1
      S1=0
      P1=0
      SOMATN=0
      PROD=1
      FATAN=FAT(N)
  600 DO 900 K=L,M
      AK=K
      FATK=AK*FATK
      SOMO=1./AK+SOMO
      PROD=(AN+AK)*PROD
      FATOR2=(-1.)**K*XS2**(N+2*K)/(FATK*PROD*FATAN)
      SOMN=0
      DO 800 J=1,N
       AJ=J
  800 SOMN=1./(AK+AJ)+SOMN
      SOMN=2.*SOMO+SOMN
  900 SOMATN=FATOR2*SOMN+SOMATN
      IF(AUX-SOMATN) 110, 109, 110
  110 AUX=SOMATN
      L=M+1
      M=M+2
      GO TO 600
  109 DO 540 I=1,N
      AI=I
      I1=I-1
      S1=1./AI+S1
  540 P1=FAT(N-I1-1)/FAT(I1)*XS2**(2*I1-N)+P1
      SOMATN=SOMATN+XS2**N/FATAN*S1
      FBSEY=FBEJ( N,X)*FATOR1-1./3.14156*(P1+SOMATN)
      RETURN
      END
```

****

FBYN

```
      FUNCTION FBYN(AN,X)
      QUAN2= 4.*AN*AN
      OITOX=8.*X
      R2PIX=SQRT(2./(.31415926E+01*X))
      ALFA=X-.78539816-.15707963E+01*AN
      A2=1
      N=X+1.
```

```
      NN=4*N-3
      AL=1
      FBYN=0
      J=0
      DO 600 I=1,NN,4
      AI=I
      A1=A2*(QUAN2-AI*AI)/(AL*OITOX)
      A2=A1*(QUAN2-(AI+2.)*(AI+2.))/((AL+1.)*OITOX)
      FBYN=(-1.)**(J+1)*A2*SIN(ALFA)+(-1.)**J*A1*COS(
     1ALFA)+FBYN
      J=J+1
  600 AL=AL+2.
      FBYN=R2PIX*(FBYN+SIN(ALFA))
      RETURN
      END
```

****

FBEI

```
      FUNCTION FBEI( N,X)
      AN=N
      IF(N-1) 333, 334, 335
  333 FBEI=FBI0(X)
      RETURN
  334 FBEI=FBI1(X)
      RETURN
  335 IF(AN-X) 520, 520, 519
  519 IF(X-12.) 690, 520, 520
  690 FBEI=FBSEI(AN,X)
      RETURN
  520 FI0=FBI0(X)
      FI1=FBI1(X)
      DO 900 I=2,N
      AI=I
      FI2=-2.*(AI-1.)/X*FI1+FI0
      FI0=FI1
  900 FI1=FI2
      FBEI=FI2
      RETURN
      END
```

****

BEIN

```
      FUNCTION BEIN(AN,X)
      N=AN
```

```
      EN=N
      DIF=AN-EN
      IF(ABS(DIF)-.5) 666, 92, 666
  666 IF(AN-2.) 640, 690, 690
  640 IF(X-6.) 690, 690, 830
  830 BEIN=FBIN(AN,X)
      RETURN
  690 BEIN=FBSEI(AN,X)
      RETURN
   92 R2PIX=SQRT(2./(3.1416*X))
      N=ABS(AN)
      DOISX=2.*X
      IF(N-1) 700, 702, 702
  702 SOM1=1
      SOM2=1
      DO 800 J=1,N
      SOM=FAT(N+J)/(FAT(J)*FAT(N-J)*DOISX**J)
      SOM1=(-1.)**J*SOM+SOM1
  800 SOM2=SOM+SOM2
      IF(AN) 900, 900, 901
  900 BEIN=SQRT(1./(.62831852E+01*X))*(EXP(X)*SOM1+(-
         1.)**N*EXP(-X)*SOM2
     1)
      RETURN
  901 BEIN=SQRT(1./(.62831852E+01*X))*(EXP(X)*SOM1+(-
         1.)**(N+1)*EXP(-X)*
     1SOM2)
      RETURN
  700 IF(AN) 500, 500, 501
  500 BEIN=R2PIX*FCOSH(X)
      RETURN
  501 BEIN=R2PIX*FSENH(X)
      RETURN
      END
```

****

FBSEI

```
      FUNCTION FBSEI(AN,X)
      M=10
      FBEI=0
      AIN=0
      L=1
      AM=M
      GAMAN=GAMA(AN+AM+2.)
      GAMAN1=GAMAN
```

```
      FATM=FAT(M)
      FATK=FATM
  600 DO 777 I=L,M
      K=M-(I-L)
      AK=K
      GAMAN=GAMAN/(AK+AN+1.)
      IF(K-1) 888, 940, 888
  940 GAMAL1=GAMAN/(AN+1.)
  888 AIN=X**(AN+2.*AK)/(2.**(AN+2.*AK)*FATK*GAMAN )+
         AIN
  777 FATK=FATK/AK
      IF(FBEI-AIN) 110, 109, 110
  110 FBEI=AIN
      L=M+1
      M=M+2
      AM=M
      FATK=FATM*(AM-1.)*AM
      FATM=FATK
      GAMAN=GAMAN1*(AN+AM)
      GAMAN1=GAMAN
      GO TO 600
  109 AI0=X**AN/(2.**AN*GAMAL1)
      FBSEI=AIN+AI0
      RETURN
      END
```

****

FBIN

```
      FUNCTION FBIN(AN,X)
      QUAN2= 4.*AN*AN
      OITOX=8.*X
      FATOR=EXP(X)/SQRT(.62831852E+01*X)
      A2=1
      N=X+1.
      NN=4*N-3
      AL=1
      FBEI=0
      J=0
      DO 600 I=1,NN,4
      AI=I
      A1=A2*(QUAN2-AI*AI)/(AL*OITOX)
      A2=A1*(QUAN2-(AI+2.)*(AI+2.))/((AL+1.)*OITOX)
      FBEI=(-1.)**(J+1)*A1+(-1.)**J*A2+FBEI
  600 AL=AL+2.
      FBIN=FATOR*FBEI+FATOR
```

```
      RETURN
      END
```

****

FBI0

```
      FUNCTION FBI0(X)
      X3=X/3.75
      X32=X3*X3
      X33=X3*X32
      X34=X32*X32
      X35=X32*X33
      X36=X32*X34
      X37=X32*X35
      X38=X34*X34
      X310=X32*X38
      X312=X32*X310
      IF(X+3.75)300,10,10
   10 IF(X-3.75)100,100,200
  100 FBI0=1.+3.5156229*X32+3.0899424*X34+1.2067492*X
         36+.2659732*X38+.03
     160768* X310+.0045813*X312
      GO TO 300
  200 FBI0 =(.39894228+.01328592*(1./X3)+.00225319*(1
         ./X32)-.00157565*(1
     1./X33)+.00916281*(1./X34)-.02057706*(1./X35)+.0
         2635537*(1./X36)-.0
     21647633*(1./X37)+.00392377*(1./X38))/(X**.5*EXP
         (-X))
  300 RETURN
      END
```

****

FBI1

```
      FUNCTION FBI1(X)
      X3=X/3.75
      X32=X3*X3
      X33=X3*X32
      X34=X32*X32
      X35=X32*X33
      X36=X32*X34
      X37=X32*X35
      X38=X34*X34
      X310=X32*X38
```

```
      X312=X32*X310
      IF(X+3.75)300,10,10
   10 IF(X-3.75)100,100,200
  100 FBI1=X*(.5+.87890594*X32+.51498869*X34+.1508493
     4*X36+.02658733*X38
     1+.00301532*X310+.00032411*X312)
      GO TO 300
  200 FBI1=(.39894228-.03988024*(1./X3)-.00362018*(1.
     /X32)+.00163801*(1.
     1/X33)-.01031555*(1./X34)+.02282967*(1./X35)-.02
     895312*(1./X36)+.01
     2787653*(1./X37)-.00420059*(1./X38))/(X**.5*EXP(
     -X))
  300 RETURN
      END
```

****

FBEK

```
      FUNCTION FBEK(N,X)
      AN=N
      IF(N-1) 333, 334, 335
  333 FBEK=FBK0(X)
      RETURN
  334 FBEK=FBK1(X)
      RETURN
  335 IF(AN-X) 520, 520, 519
  519 IF(X-12.) 690, 520, 520
  690 FBEK=FBSEK(AN,X)
      RETURN
  520 FK0=FBK0(X)
      FK1=FBK1(X)
      DO 900 I=2,N
      AI=I
      FK2=2.*(AI-1.)/X*FK1+FK0
      FK0=FK1
  900 FK1=FK2
      FBEK=FK2
      RETURN
      END
```

****

BEKN

```
      FUNCTION BEKN(AN,X)
```

```
      N=AN
      EN=N
      DIF=AN-EN
      IF(ABS(DIF)-.5)666, 92, 666
  666 IF(X-6.) 690, 690, 830
  830 BEKN=FBKN(AN,X)
      RETURN
  690 PI=.31415926E+01
      BEKN=PI/2.*(FBSEI(-AN,X)-FBSEI(AN,X))/SIN(AN*PI)
      RETURN
   92 DOISX=2.*X
      N=ABS(AN)
      RAIZ=SQRT(.31415926E+01/DOISX)
      IF(N-1) 700, 702, 702
  702 SIGMA=0
      DO 800 J=1,N
  800 SIGMA=FAT(N+J)/(FAT(J)*FAT(N-J)*DOISX**J)+SIGMA
      BEKN=RAIZ*EXP(-X)*SIGMA
      RETURN
  700 BEKN=RAIZ*EXP(-X)
      RETURN
      END
```

****

FBSEK

```
      FUNCTION FBSEK(AN,X)
      N=AN
      L=1
      M=20
      XS2=X/2.
      AUX=0
      FLOG=LOG(XS2)
      SOMO=0
      FATK=1
      S1=0
      PARC1=0
      PARC2=0
      PROD=1
      FATAN=FAT(N)
  600 DO 1000 K=L,M
      SOMA=0
      AK=K
      FATK=AK*FATK
      SOMO=1./AK+SOMO
      PROD=(AN+AK)*PROD
```

```
      DIV=XS2**(N+2*K)/(FATK*PROD*FATAN)
      DO 999 J=1, N
      AJ=J
  999 SOMA=1./(AK+AJ)+SOMA
      SOMA=SOMO+.5*SOMA
 1000 PARC2=DIV*(FLOG-SOMA+.57721566)+PARC2
      IF(AUX-PARC2) 110, 109, 110
  110 AUX=PARC2
      L=M+1
      M=M+2
      GO TO 600
  109 DO 500 I=1,N
      AI=I
      I1=I-1
      S1=1./AI+S1
  500 PARC1=(-1.)**I1*FAT(N-I1-1)/FAT(I1)*XS2**(2*I1-
         N)+PARC1
      PARC2=(-1.) **(N+1)*(PARC2+XS2**N/FATAN*(FLOG-.
         5*S1  +.57721566))
      FBSEK=.5*PARC1+PARC2
      RETURN
      END
```

****

FBKN

```
      FUNCTION FBKN(AN,X)
      OITOX=8.*X
      RAIZ=SQRT(.31415926E+01/(2.*X))
      QUAN2=4.*AN*AN
      A2=1
      N=X+1.
      NN=4*N-3
      AL=1
      FBEK=0
      J=0
      DO 600 I=1,NN,4
      AI=I
      A1=A2*(QUAN2-AI*AI)/(AL*OITOX)
      A2=A1*(QUAN2-(AI+2.)*(AI+2.))/((AL+1.)*OITOX)
      FBEK=A1+A2+FBEK
  600 AL=AL+2.
      FBKN=RAIZ*EXP(-X)*(FBEK+1.)
      RETURN
      END
```

****

FBKO

```
      FUNCTION FBKO(X)
      X2=X/2.
      X22=X2*X2
      X23=X2*X22
      X24=X22*X22
      X25=X22*X23
      X26=X22*X24
      X28=X24*X24
      X210=X22*X28
      X212=X22*X210
      IF (X-2.)100,100,200
100   AIO=FBIO(X)
      FBKO=-LOG (.5*X)*AIO-.57721566+.42278420*X22+.2
         3069756*X24+.034488
     1 590*X26+.00262698*X28+.00010750*X210+.00000740
         *X212
      GO TO 300
 200  FBKO=(1.25331414-.07832358*(1./X2)+.02189568*(1
         ./X22)-.01062446*(1
     1 ./X23)+.00587872*(1./X24)-.00251540*(1./X25)+
         .00053208*(1./X26))
     2/(X **.5*EXP(X))
300   RETURN
      END
```

****

FBK1

```
      FUNCTION FBK1(X)
      X2=X/2.
      X22=X2*X2
      X23=X2*X22
      X24=X22*X22
      X25=X22*X23
      X26=X22*X24
      X28=X24*X24
      X210=X22*X28
      X212=X22*X210
      IF(X-2.)100,100,200
100   AI1=FBI1(X)
  45  EK1=(1.+.15443144*X22-.67278579*X24-.18156897*X
         26-.01919402*X28-.0
     10110404*X210-.00004686*X212) /X
```

```
      IF(ABS(EK1+ABS(LOG(.5*X)*AI1)-LOG(.5*X)*AI1)-EK
         1) 120, 50, 120
   50 FBK1=ABS(LOG(.5*X)*AI1)+EK1
      GO TO 300
  120 FBK1=ABSF(ABS(LOG(.5*X)*AI1)-EK1)
      GO TO 300
  200 FBK1=(1.25331414+.23498619*(1./X2)-.03655620*(1
         ./X22)+.01504268*(1
     1./X23)-.00780353*(1./X24)+.00325614*(1./X25)-.0
         0068245*(1./X26))/(
     2X**.5*EXP(X))
  300 RETURN
      END
```

****

FSENH

```
      FUNCTION FSENH(X)
      FSENH=(EXP(X)-EXP(-X))/2.
      RETURN
      END
```

****

FCOSH

```
      FUNCTION FCOSH(X)
      FCOSH    =(EXP(X)+EXP(-X))/2.
       RETURN
       END
```

****