Includes easy-entry keystroke listings.

HEWLETT-PACKARD

HP-75

USERS' LIBRARY SOLUTIONS

Math III

NOTICE

The program material contained herein is supplied without representation or warranty of any kind. Hewlett-Packard Company therefore assumes no responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of this program material or any part thereof.

HEWLETT PACKARD

HP-75 SOLUTIONS BOOK/MEDIA ORDER FORM

All HP-75 Solutions Books are available recorded from the Users' Library* or from any Hewlett-Packard Full Service Dealer. The individual Solutions Books are $12.50 each; recorded mini-data cassettes are $12.00 for each Solutions Book requested.

Telephone the Users' Library directly (503-757-2000, ext. 3371) with your Solutions Book request, your order will ship that SAME DAY; or, call our TOLL FREE number (800-547-3400, Continental U.S. only), and orders ship within 48 hours. Mail requests also ship within 48 hours of receipt.

HP-75 Users' Library Solutions Books

00075-90061	Math I
00075-90062	Math II
00075-90063	Math III
00075-90065	Games
00075-90066	Electronics
00075-90067	Statistics
00075-90070	Real Estate
00075-90071	Finance
00075-90076	Test Statistics

Solutions Book Part Number	Title	Book Only	Mini-Data Cassettes	Price
00075-90_____				
00075-90_____				
00075-90_____				
Please add Handling Charge on orders including Solutions Books				**3.50**
State and Local Taxes				
TOTAL				

All payments must be in U.S. dollars and drawn on a U.S. bank.
Send orders (with payment) directly to: HEWLETT-PACKARD USERS' LIBRARY
1000 N.E. Circle Blvd.
Corvallis, Oregon 97330

☐ Payment enclosed ☐ Master Card ☐ VISA ☐ Purchase order

Card Number/Company Purchase Order Exp

Name

Company (if applicable)

Address Telephone

City State/Country Zip

*Recorded Solutions Book programs shipped to U.S. customers only.

TABLE OF CONTENTS

1. FAST FOURIER TRANSFORM . . . . . by Frank Flaherty 1

This program computes the direct or inverse discrete Fourier transform of a function or data points using the Cooley-Tukey algorithm.

2. FAST FOURIER SERIES/TRIGONOMETRIC INTERPOLATION . . . . . by Frank Flaherty 14

This program takes as input data points or a periodic square-integrable function on [0, 2pi] and estimates its Fourier coefficients, using the Cooley-Tukey algorithm.

3. ATTENUATED FOURIER SERIES . . . . . by Frank Flaherty 26

This program estimates the Fourier coefficients of a piecewise-linear approximation to a periodic square integrable function on [0, 2pi] using the Cooley-Tukey algorithm.

4. SPHERICAL HARMONICS . . . . . by Frank Flaherty 35

This program computes the spherical harmonics (eigen functions of the Laplace operator on the unit sphere) Y_n^m ( ) using a recurrence relation.

5. ELLIPTIC INTEGRALS . . . . . by Frank Flaherty 43

This program computes complete and incomplete elliptic integrals of the first and second type. The method of computation is based on iteration using the Landen transformation.

6. BESSEL FUNCTION: ASYMPTOTIC EXPANSION . . . . . by Frank Flaherty 55

This program computes Bessel functions of the first and second kind for arbitrary order and large positive arguments, using an asymptotic series.

7. BESSEL FUNCTIONS: BACKWARD RECURRENCE . . . . . by Frank Flaherty 60

This program computes the Bessel functions $J_m(z)$ for positive z and for non-negative integral order m.

8. GAMMA FUNCTION . . . . . by Frank Flaherty 65

This program extends the factorial function to arbitrary values using an asymptotic approximation to the analytic continuation.

TABLE OF CONTENTS (continued)

9. ERROR FUNCTION . . . . . by Frank Flaherty 71

This program computes the integral.

10. LEGENDRE POLYNOMIALS . . . . . by Frank Flaherty 76

Orthogonalizing the polynomials $1, X, X^2, \ldots, X^n$ on the interval [-1, 1] with respect to the inner product $\langle f,g \rangle = f(t), g(t)\, dt$ yields the Legendre polynomials. Using a recurrence relation this program computes the value of the Legendre polynomials.

PROGRAM DESCRIPTION

FAST FOURIER TRANSFORM

Given a complex vector $\alpha=(\alpha_0,\ldots,\alpha_{n-1})$, this program computes the Fourier transform coefficients

$$\beta_m = \frac{1}{n}\sum_k \alpha_k \, e^{-2\pi \frac{m}{n}k\sqrt{-1}} \qquad 0\le k\le n-1$$

The inverse coefficients are

$$\beta_m = \sum_k \alpha_k \, e^{2\pi \frac{m}{n}k\sqrt{-1}} \qquad 0\le k\le n-1$$

Algorithm: Input n, ($m=\log_2(n)$), data points a(*), b(*) or input z(t)

For i=0 to n-1 step 1

Input $\alpha(i) = a(i) + \sqrt{-1}\, b(i)$ or evaluate z(i)

Next i

Def bit reversing function r(k)

Set flag f=1 for inverse, f=-1 for direct

For k=m-1 to 0 step -1

For p=0 to n-1 step -2^{k+1}

$s=r(p/2^k)$

$\xi_1=\cos(2\pi s/n) + \sqrt{-1}\,\sin(2\pi s/n)f$

For j=0 to 2^k-1 step 1

$\xi_2=\xi_1\,\alpha(j+p+2^k)$

$\alpha(j+p) = \alpha(j+p) + \xi_2$

$\alpha(j+p+2^k) = \alpha(j+p) -2\xi_2$

Next j

Next p

Next k

PROGRAM DESCRIPTION

FAST FOURIER TRANSFORM (continued)

```
For i=0 to n-1 step 1
If r(i)>i, swap α(i), α(r(i))
Next i
For i=0 to n-1 step 1
If f=-1 disp β(i)=α(i)/n
If f=1 disp β(i)=α(i)
Next i
Stop
```

SAMPLE PROBLEM

A) Compute the Fourier transform (direct) of the following data pairs

$\alpha(0) = (1,2)$
$\alpha(1) = (4,1)$
$\alpha(2) = (-2,0)$
$\alpha(3) = (3,-1)$

SOLUTION

STEP	INSTRUCTIONS	DISPLAY	INPUT
A 1	Run program	*** FOURIER TRANSFORM ***	
2	Enter number of data points:	Number of data points?	4 [RTN]
3	Select data as input:	Function or Data points?F	D [RTN]
4	Enter data:	Real part of $\alpha(0)$?	1 [RTN]
		Imag part of $\alpha(0)$?	2 [RTN]
		Real part of $\alpha(1)$?	4 [RTN]
		Imag part of $\alpha(1)$?	1 [RTN]
		Real part of $\alpha(2)$?	-2 [RTN]
		Imag part of $\alpha(2)$?	0 [RTN]
		Real part of $\alpha(3)$?	3 [RTN]
		Imag part of $\alpha(3)$?	-1 [RTN]
5	Continue:	Recheck data or Continue?R	C [RTN]
6	Select direct transform:	Direct/Inverse transform?D	[RTN]
		Computing...	
7	View each point by pressing	Real part of $\beta(0)$=1.5	[RTN]
	[RTN]. Pressing [BACK] will	Imag part of $\beta(0)$=.5	[RTN]
	goto the previous point.	Real part of $\beta(1)$=1.25	[RTN]

SOLUTION

STEP	INSTRUCTIONS	DISPLAY	INPUT
		Imag part of β(1)=.249999999998	[RTN]
		Real part of β(2)=-2	[RTN]
		Imag part of β(2)= .5	[RTN]
		Real part of β(3)= .25000000003	[RTN]
		Imag part of β(3)=.730000000003	[RTN]
8	End program.	Run again, View again, or End?R	E [RTN]

SAMPLE PROBLEM

B) Find the inverse Fourier transform of the data pairs $\alpha(i) = i + \sqrt{-1}\ i$ $i = 0,\ldots,3$

SOLUTION

STEP	INSTRUCTIONS	DISPLAY	INPUT
B 1	Run program	*** FOURIER TRANSFORM ***	
2	Enter number of data points:	Number of data points?	4 [RTN]
3	Select data as input:	Function or Data points?F	D [RTN]
4	Enter data:	Real part of $\alpha(0)$?	0 [RTN]
		Imag part of $\alpha(0)$?	0 [RTN]
		Real part of $\alpha(1)$?	1 [RTN]
		Imag part of $\alpha(1)$?	1 [RTN]
		Real part of $\alpha(2)$?	2 [RTN]
		Imag part of $\alpha(2)$?	2 [RTN]
		Real part of $\alpha(3)$?	3 [RTN]
		Imag part of $\alpha(3)$?	3 [RTN]
5	Continue:	Recheck data or Continue?R	C [RTN]
6	Select inverse transform:	Direct/Inverse transform?D	I [RTN]
		Computing...	
7	View each point by pressing	Real part of $\beta(0)$=6	[RTN]
	[RTN]. Pressing [BACK] will	Imag part of $\beta(0)$=6	[RTN]
	goto the previous point.	Real part of $\beta(1)$= .0000000001	[RTN]
		Imag part of $\beta(1)$=-3.999999999	[RTN]
		Real part of $\beta(2)$=-2	[RTN]
		Imag part of $\beta(2)$=-2	[RTN]

SOLUTION

STEP	INSTRUCTIONS	DISPLAY	INPUT
		Real part of β(3)=-4.0000000001	[RTN]
		Imag part of β(3)=-.00000000001	[RTN]
8	End program.	Run again, View again, or End?R	E [RTN]

SAMPLE PROBLEM

C) Estimate inverse Fourier transform of

$$z(t) = t + \sqrt{-1}\ t$$

with four data points. In this case

Real part $z(t) = x(t) = t$,

Imag part $z(t) = y(t) = t$.

SOLUTION

STEP	INSTRUCTIONS	DISPLAY	INPUT
C 1	Enter x(t) in 510	510 DEF FNX(T)=T	
	Enter y(t) in 520	520 DEF FNY(T)=T	
1	Run program	*** FOURIER TRANSFORM ***	
2	Enter number of data points:	Number of data points?	4 [RTN]
3	Select function as input:	Function or Data points?F	F [RTN]
		Evaluating function...	
6	Select inverse transform:	Direct/Inverse transform?D	I [RTN]
		Computing...	
7	View each point by pressing	Real part of β(0)=6	[RTN]
	[RTN]. Pressing [BACK] will	Imag part of β(0)=6	[RTN]
	goto the previous point.	Real part of β(1)= .0000000001	[RTN]
		Imag part of β(1)=-3.999999999	[RTN]
		Real part of β(2)=-2	[RTN]
		Imag part of β(2)=-2	[RTN]
		Real part of β(3)=-4.0000000001	[RTN]
		Imag part of β(3)=-.00000000001	[RTN]
8	End program.	Run again, View again, or End?R	E [RTN]

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
1	Run program	*** FOURIER TRANSFORM ***	
2	Enter number of data points:	Number of data points?	n [RTN]
	(Must be a power of 2)		
3	Select function or data	Function or Data points?F	F or D [RTN]
4	If data, enter the real and		
	imaginary part of each point:	Real part of α(i)?	Re α(i) [RTN]
		Imag part of α(i)?	Im α(i) [RTN]
	If function, enter the x(t) in	510 DEF FNX(T)=	
	line 510 and y(t) in 520	520 DEF FNY(T)=	
	before starting program.		
	Goto step 6.		
5	Check data:	Recheck data or Continue?R	R or C [RTN]
a	When checking the data (R),		
	press [RTN] to proceed to		
	each point. If an error is		
	found, press [BACK] and		
	re-enter:	Imag part of α(i) =	[BACK]
		New real part of α(i)?	Re α(i) [RTN]
	Then continue through the data		
b	Enter C to continue to step 6.		
6	Select direct or inverse	Direct or Inverse transform?D	D or I [RTN]
	transform:		
7	Computation begins:	Computing...	
8	View each point by pressing	Real part β(i) =	[RTN] or
	[RTN]. Pressing [BACK] will	Imag part β(i) =	[BACK]

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
9	Select program options:	Run again, View again, or End?R	R,V, or E [RTN]
	To run the program again,		
	enter 'R' and goto step 2.		
	To view the answers again,		
	enter 'V' and goto step 8.		
	To end the program enter 'E'.		

VARIABLE NAMES

NAME	DESCRIPTION	NAME	DESCRIPTION
FNR(X)	Bit reversing function	X_2	Imag part of prim. root of unity to some power multiplied by F
X	Dummy variable for FNR(X)		
A	Temporary storage for FNR(X)	X_2, Y2	Temp storage for main calc.
U	Temp storage for FNR(X)	X_3, Y_3	Temp storage for swap
FNA	Answer function	S	Temp storage for bit reversing function FNR
B	Temp storage for FNA	K	Simplifying power
FNY(T)	Real part of object function	Q$	String for RVE module
FNY(T)	Imag part of object function	R$	String for Direct/Inverse module
A(I)	Real part of object sequence	S$	String for Data/Function module
B(I)	Imag part of object sequence	T$	String for checking data-module
F	Flag for inverse/direct transform	N	Number of subdivision points
X_1	Real part of primitive root of unity to some power	M	$Log_2(n)$

NOTES AND REFERENCES

Notes: 1. The program will work with up to 1024 points if approx. 20K bytes of RAM are available. To alter the number of points that are available, change the dimension in line 80. The number of points used must be a power of 2.

2. Execution time increases significantly with the number of data points.

Reference: Stoer, J. and R. Bulirsch, INTRODUCTION TO NUMERICAL ANALYSIS, (Spring-Verlag, New York, 1980), Section 2.3.2.

PROGRAM LISTING

```
 10 ! FAST FOURIER TRANSFORM
 20 ! Computes the direct or
 30 ! inverse Fourier transform
 40 ! of data points or functions
 50 ! using the Cooley-Tukey
 60 ! algorithm
 70 ! Revision 11/01/1982
 80 DIM A(512),B(512)
 90 DISP '* * *  FOURIER  TRANSFORM * *
    *' @ OPTION ANGLE RADIANS
100 INPUT 'Number of data points?'; N @
    M=IP(LOG(N)/LOG(2))
110 IF FP(LOG(N)/LOG(2))#0 THEN BEEP @      -Number of data points must be
    DISP 'Incorrect number of points!'       a power of two
    @ GOTO 100
120 ! Bit reversing function                -Reverse binary expansion of
                                             integer
130 DEF FNR(X)
140 U=0
150 FOR P=0 TO M-1
160 A=RMD(X,2)
170 U=U+A*2^(M-1-P)
180 X=IP(X/2)
190 IF X<=1 THEN 210
200 NEXT P
210 U=U+X*2^(M-2-P)
220 FNR=U @ END DEF
230 ! Define answer function
240 DEF FNA
250 B=NUM(KEY$) @ IF B#13 AND B#8 THEN      -Wait for 'RTN' or 'BACK' keys
    250
260 FNA=B @ END DEF
270 ! Transform data points or func?
280 DISP CHR$(198);'unction or ';CHR$(1
    96);
290 INPUT 'ata points?','F'; S$ @ S$=UP     -Select data points or function
    RC$(S$[1,1])                             as input
300 ON POS('FD',S$)+1 GOTO 280,500,320
310 ! Input data points                     -Data point entry
320 FOR I=0 TO N-1
330 DISP USING "'Real part of ',a,'(',k
    " ; CHR$(4),I;
340 INPUT ')? ';A(I)
350 DISP USING "'Imag part of ',a,'(',k
    " ; CHR$(4),I;
360 INPUT ')? ';B(I)
370 NEXT I
380 ! Check data points
390 DISP CHR$(210);'echeck data,or ';CH     -Option to check input data
    R$(195);
400 INPUT 'ontinue?','R'; T$ @ T$=UPRC$
    (T$[1,1])
410 ON POS('RC',T$)+1 GOTO 390,420,590
420 FOR I=0 TO N-1
```

PROGRAM LISTING

```
430 DISP USING "'Real part of ',a,'(',k
    ,')=',k" ; CHR$(4),I,A(I)
440 IF FNA=8 THEN DISP USING "'New real
     part of ',a,'(',k" ; CHR$(4),I; @
    INPUT ')? ';A(I)
450 DISP USING "'Imag part of ',a,'(',k
    ,')=',k" ; CHR$(4),I,B(I)
460 IF FNA=8 THEN DISP USING "'New imag
     part of ',a,'(',k" ; CHR$(4),I; @
    INPUT ')? ';B(I)
470 NEXT I
480 GOTO 390
490 ! Function to be transformed
500 DISP 'Evaluating function...'
510 DEF FNX(T) = T                          -User defined functions
520 DEF FNY(T) = T
530 FOR I=0 TO N-1
540 A(I)=FNX(I) @ B(I)=FNY(I)
550 NEXT I
560 ! Direct/inverse transform               -Select direct or inverse
                                              transform
570 ! from keyboard
580 ! Set flag for inverse
590 F=1
600 DISP CHR$(196);'irect/';CHR$(201);
610 INPUT 'nverse transform?','D'; R$ @
     R$=UPRC$(R$[1,1])
620 DISP 'Computing...'
630 ON POS('DI',R$)+1 GOTO 600,650,670
640 ! Set flag for direct
650 F=-1
660 ! Main loops
670 FOR K=M-1 TO 0 STEP -1
680 K1=2^K
690 FOR L=0 TO N-1 STEP 2^(K+1)
700 S=FNR(L/K1)
710 ! With f=-1,computing direct
720 ! Fourier transform
730 X1=COS(2*PI*S/N) @ Y1=SIN(2*PI*S/N)
    *F
740 FOR J=0 TO K1-1
750 X2=A(J+L+K1)*X1-B(J+L+K1)*Y1
760 Y2=A(J+L+K1)*Y1+B(J+L+K1)*X1
770 A(J+L)=A(J+L)+X2 @ B(J+L)=B(J+L)+Y2
780 A(J+L+K1)=A(J+L)-X2-X2
790 B(J+L+K1)=B(J+L)-Y2-Y2
800 NEXT J
810 NEXT L
820 NEXT K
830 ! Unscramble remainders
840 FOR I=0 TO N-1
850 IF FNR(I)<=I THEN 890
860 X3=A(I) @ Y3=B(I)
870 A(I)=A(FNR(I)) @ B(I)=B(FNR(I))
```

PROGRAM LISTING

```
 880 A(FNR(I))=X3 @ B(FNR(I))=Y3
 890 NEXT I
 900 ! Test for inverse transform
 910 ! which is the case f=1
 920 IF F=1 THEN 980
 930 ! Compute direct transform
 940 FOR I=0 TO N-1
 950 A(I)=A(I)/N @ B(I)=B(I)/N
 960 NEXT I
 970 ! View the transform
 980 FOR I=0 TO N-1
 990 DISP USING "'Real part of ',a,'(',k
     ,')=',k" ; CHR$(5),I,A(I)
1000 IF FNA=8 THEN I=MAX(I-1,0)
1010 DISP USING "'Imag part of ',a,'(',k
     ,')=',k" ; CHR$(5),I,B(I)
1020 IF FNA=8 THEN I=MAX(I-1,0) @ GOTO 9
     90
1030 NEXT I
1040 ! Run,view, or end
1050 DISP CHR$(210);'un again,';CHR$(214
     );'iew again,or';CHR$(197);
1060 INPUT 'nd?','R'; Q$ @ Q$=UPRC$(Q$[1
     ,1])
1070 ON POS('RVE',Q$)+1 GOTO 1050,100,98
     0,1080
1080 STOP
```

—Program options

PROGRAM DESCRIPTION

FAST FOURIER SERIES/TRIGONOMETRIC INTERPOLATION

This program finds a trigonometric interpolation of the Fourier series. Given a function $z(t) = x(t)+\sqrt{-1}y(t)$ which is peroidic on $[0,2\pi]$ and whose modulus squared is integrable, the Fourier expansion is

$$z(t) = \sum_m \hat{z}(m)e^{mt\sqrt{-1}} \qquad m=0,\pm1,\pm2,\ldots$$

The Fourier expansion is approximated by trigonometric polynomials of the form

$$p(t) = \sum_m \beta_m \, \mathrm{EXP}(mt\sqrt{-1}) \qquad m=0,1,\ldots,n-1$$

To obtain a two-sided polynomial, β_m is extended with period n :(i.e., $\beta_{m+n}=\beta_m$, where n is the number of points or terms in the approximation).

An important special case is z(t), a real valued function (y(t)=0 for all t). The coefficients $\hat{z}(m)+\hat{z}(-m)$, $i(\hat{z}(m) -\hat{z}(-m))$ are real in that case, so putting

$$A_m = 2\,\mathrm{Re}(\beta_m)$$

$$B_m = -2\,\mathrm{Im}(\beta_m) \qquad m=1,\ldots,n/2-1$$

$$A_o = 2\,\beta_o$$

$$A_{n/2} = 2\,\beta_{n/2}$$

yields a sin, cos - polynomial with real coefficients:

$$A(t) = \frac{A_o}{2} + \sum_m (A_m \,\mathrm{COS}(mt) + \beta_m \,\mathrm{SIN}(mt)) + \frac{A_{n/2}}{2}\,\mathrm{COS}(\tfrac{n}{2}t) \quad m=1,2,\ldots,\tfrac{n}{2}-1$$

Note that n must be a power of two.

PROGRAM DESCRIPTION

FAST FOURIER SERIES/TRIGONOMETRIC INTERPOLATION (continued)

Algorithm: Input n, $m=\log_2(n)$, data points a(*),b(*) or input z(t)

For i=0 to n-1 step 1

Input $\alpha(i) = a(i) + \sqrt{-1}\, b(i)$ or evaluate $z\left(\frac{2\pi\sqrt{-1}}{n}\right)$

Next i

Define bit reversing function r(k)

For k = m-1 to 0 step -1

For p = 0 to n-1 step 2^{k+1}

$s = r(p/2^k)$

$\xi_1 = \mathrm{COS}\,(2\pi s/n) + \sqrt{-1}\,\mathrm{SIN}(2\pi s/n)$

For j=0 to 2^k-1 step 1

$\xi_2 = \xi_1\, \alpha(j+p+2^k)$

$\alpha(j+p) = \alpha(j+p) + \xi_2$

$\alpha(j+p+2^k) = \alpha(j+p) - 2\xi_2$

Next j

Next p

Next k

For i=0 to n-1 step 1

Swap $\alpha(i)$, $\alpha(r(i))$ if $r(i)>i$

Next i

For i = 0 to n-1 step 1

Disp $\beta(i)=\alpha(i)/n$

Next i

Stop

SAMPLE PROBLEM

A) Estimate the Fourier series of the data points

$\alpha(0)=(0,0)$, $\alpha(1)=(1,1)$, $\alpha(2)=(2,2)$, $\alpha(3)=(3,3)$.

SOLUTION

STEP	INSTRUCTIONS	DISPLAY	INPUT
A1	Run program	*** FAST FOURIER SERIES ***	
2	Enter number of data points:	Number of data points?	4 [RTN]
3	Select data as input:	Function or Data points?F	D [RTN]
4	Enter data:	Real part of $\alpha(0)$?	0 [RTN]
		Imag part of $\alpha(0)$?	0 [RTN]
		Real part of $\alpha(1)$?	1 [RTN]
		Imag part of $\alpha(1)$?	1 [RTN]
		Real part of $\alpha(2)$?	2 [RTN]
		Imag part of $\alpha(2)$?	2 [RTN]
		Real part of $\alpha(3)$?	3 [RTN]
		Imag part of $\alpha(3)$?	3 [RTN]
5	Continue:	Recheck data or Continue?R	C [RTN]
		Computing...	
6	Select complex series:	Select series: Real or Complex?R	C [RTN]
7	View each point by pressing	Real part of $\beta(0)$= 1.5	[RTN]
	[RTN]. Pressing [BACK] will	Imag part of $\beta(0)$= 1.5	[RTN]
	goto the previous point.	Real part of $\beta(1)$=-.99999999998	[RTN]
		Imag part of $\beta(1)$= 2.5E-12	[RTN]
		Real part of $\beta(2)$=-.5	[RTN]

SOLUTION

STEP	INSTRUCTIONS	DISPLAY	INPUT
		Imag part of β(2)=-.5	[RTN]
		Real part of β(3)=-2.5E-12	[RTN]
		Imag part of β(3)=-1	[RTN]
8	End program	Run again, View again, or End?R	E [RTN]

SAMPLE PROBLEM

B) Compute the polynomial approximation to $EXP(t\sqrt{-1}) = COS(t+\sqrt{-1})*SIN(t)=x(t)+\sqrt{-1}\ y(t)$

SOLUTION

STEP	INSTRUCTIONS	DISPLAY	INPUT
B1	Enter x(t) on line 530	530 FNX=COS(T)	
	Enter y(t) on line 560	560 FNY=SIN(T)	
2	Run program	*** FAST FOURIER SERIES ***	
3	Enter number of data points:	Number of data points?	2 [RTN]
4	Select function as input:	Function or Data points?F	F [RTN]
		Evaluating function...	
		Computing...	
5	Select complex series:	Select series: Real or Complex?R	C [RTN]
6	View each point by pressing	Real part of β(0)= 0	[RTN]
	[RTN]. Pressing [BACK] will	Imag part of β(0)=-1.04E-13	[RTN]
	goto the previous point.	Real part of β(1)= 1	[RTN]
		Imag part of β(1)=1.04E-13	[RTN]
7	End program.	Run again, View again, or End?R	E [RTN]

SAMPLE PROBLEM

C) Estimate $J_0(17.5), J_1(17.5), J_2(17.5), J_3(17.5)$

using $z(t) = \mathrm{COS}(17.5\ \mathrm{SIN}(t) + \sqrt{-1}\ \mathrm{SIN}(17.5\ \mathrm{SIN}(t)$.

Read off real part of B(0),B(1),B(2),B(3).

SOLUTION

STEP	INSTRUCTIONS	DISPLAY	INPUT
C 1	Enter x(t) on line 530	530 FNX=COS(17.5*SIN(T))	
	Enter y(t) on line 560	560 FNY=SIN(17.5*SIN(T))	
2	Run program	*** FAST FOURIER SERIES ***	
3	Enter number of data points:	Number of data points?	64 [RTN]
4	Select function as input:	Function or Data points?F	F [RTN]
		Evaluating function...	
		Computing...	
5	Select complex series:	Select series: Real or Complex?R	C [RTN]
6	View each point by pressing	Real part of β(0)=-.103110398225	[RTN]
	[RTN]. Pressing [BACK] will	Imag part of β(0)= 5.755625E-12	[RTN]
	goto the previous point.	Real part of β(1)=-.163419969425	[RTN]
	In this case, ignore the	Imag part of β(1)=-7.412512652E-12	[RTN]
	imaginary part of (i).	Real part of β(2)=8.44338302963E-2	[RTN]
		Imag part of β(2)= 8.411875E-12	[RTN]
		Real part of β(3)= .182719130625	[RTN]
7	End program, ignoring rest		
	of answer.		[ATTN]

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
1	Run program	*** FAST FOURIER SERIES ***	
2	Enter number of data points:	Number of data points?	n [RTN]
	(Must be a power of 2)		
3	Select function or data	Function or Data points?F	F or D [RTN]
	as input:		
4	If data, enter the real and		
	imaginary parts of each point:	Real part of $\alpha(i)$?	Re $\alpha(i)$ [RTN]
		Imag part of $\alpha(i)$?	Im $\alpha(i)$ [RTN]
	If function, enter the x(t)	530 FNX=	
	in line 510 and y(t) in 560	560 FNY=	
	before starting program.		
	Goto step 6.		
5	Check data:	Recheck data or Continue?R	R or C [RTN]
a	When checking the data (R),		
	press [RTN] to proceed to		
	each point. If an error is		
	found, press [BACK] and		
	re-enter:	Imag part of $\alpha(i)$ =	[BACK]
		New real part of $\alpha(i)$?	Re $\alpha(i)$ [RTN]
	Then continue through the data.		
b	Enter C to continue to step 6.		
6	Select real or complex series:	Select series: Real or Complex?R	R or C [RTN]
7	Computation begins:	Computing...	
8	View each point by pressing		
	[RTN]. Pressing [BACK] will		

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
	goto the previous point.		
	If real:	A(i)=	[RTN] or
		B(i)=	[BACK]
	If complex:	Real part β(i)=	[RTN] or
		Imag part β(i)=	[BACK]
9	Select program options:	Run again, View again, or End?R	R,V, or E [RTN]
	To run the program again,		
	enter 'R' and goto step 2.		
	To view the answers again,		
	enter 'V' and goto step 8.		
	To end the program enter 'E'.		

VARIABLE NAMES

NAME	DESCRIPTION	NAME	DESCRIPTION
FNR(X)	Bit reversing function	X_1, Y_1	Real and imag part of reciprocal of primitive root of unity
X	Dummy variable for FNR		
A	Temporary storage for FNR	X_2, Y_2	Temp storage for real and imag part of calculation
U	Temporary storage for FNR	X_3, Y_3	Temp storage for swap
FNA	Answer function	S	Temp storage for value of FNR
B	Dummy variable for FNA	K_1	Simplifies power
A(I)	Real part of input (I)	N	Number of subdivisions
B(I)	Imag part of input (I)	M	Log_2 (n)
FNX	Real part of object function	Q$	Test string for RVE
FNY	Imag part of object function	P$	Test string for real/ complex data
T	Dummy variable for FNX,FNY	R$	Test string for data points/function
		T$	Test string for rechecking data input

NOTES AND REFERENCES

References: 1. Stoer, J., and R. Bulirsch, INTRODUCTION TO NUMERICAL ANALYSIS, (Springer-Verlag, New York, 1980).

2. Henrici, P., FAST FOURIER METHODS IN COMPUTATIONAL COMPLEX ANALYSIS, (SIAM Review, V. 21, 1979, p.481-527.)

PROGRAM LISTING

```
 10 ! FAST FOURIER SERIES/TRIGON-
 20 ! OMETRIC INTERPOLATION
 30 ! Computes the finite Fourier
 40 ! series (trig interpolant)
 50 ! at points supplied by func-
 60 ! tions or input data,using
 70 ! Cooley-Tukey algorithm
 80 ! Revision 11/01/1982
 90 DIM A(512),B(512)
100 DISP '* * * FAST FOURIER SERIES * *
     *'
110 INPUT 'Number of data points?'; N @
     M=IP(LOG(N)/LOG(2))
120 IF FP(LOG(N)/LOG(2))#0 THEN BEEP @
    DISP 'Incorrect number of data poin
    ts!' @ GOTO 110
130 ! Bit reversing function
140 DEF FNR(X)
150 U=0
160 FOR P=0 TO M-1
170 A=RMD(X,2)
180 U=U+A*2^(M-1-P)
190 X=IP(X/2)
200 IF X<=1 THEN 220
210 NEXT P
220 U=U+X*2^(M-2-P)
230 FNR=U @ END DEF
240 ! Define answer function
250 DEF FNA
260 B=NUM(KEY$) @ IF B#13 AND B#8 THEN
    260
270 FNA=B @ END DEF
280 ! Interpolate func or data
290 DISP CHR$(198);'unction or ';CHR$(1
    96);
300 INPUT 'ata points?','F'; R$ @ R$=UP
    RC$(R$[1,1])
310 ON POS('FD',R$)+1 GOTO 290,510,330
320 ! Input data points
330 FOR I=0 TO N-1
340 DISP USING "'Real part of ',a,'(',k
    " ; CHR$(4),I;
350 INPUT ')? ';A(I)
360 DISP USING "'Imag part of ',a,'(',k
    " ; CHR$(4),I;
370 INPUT ')? ';B(I)
380 NEXT I
390 ! Check data points
400 DISP CHR$(210);'echeck data,or ';CH
    R$(195);
410 INPUT 'ontinue?','R'; T$ @ T$=UPRC$
    (T$[1,1])
```

-Number of points must be a power of two.

-Reverses binary expansion of integers

-Wait for 'RTN' or 'BACK' keys

-Select function or data as input

-Check data points

-Option to check data points

PROGRAM LISTING

```
420 ON POS('RC',T$)+1 GOTO 400,430,620
430 FOR I=0 TO N-1
440 DISP USING "'Real part ',a,'(',k,')
    =',k" ; CHR$(4),I,A(I)
450 IF FNA=8 THEN DISP USING "'New real
     part of ',a,'(',k" ; CHR$(4),I; @
    INPUT ')? ';A(I)
460 DISP USING "'Imag part ',a,'(',k,')
    =',k" ; CHR$(4),I,B(I)
470 IF FNA=8 THEN DISP USING "'New imag
     part of ',a,'(',k" ; CHR$(4),I; @
    INPUT ')? ';B(I)
480 NEXT I
490 GOTO 400
500 ! Function interpolates
510 DISP 'Evaluating function...'
520 DEF FNX(T)                          -User defined functions
530 FNX=COS(17.5*SIN(T))
540 END DEF
550 DEF FNY(T)
560 FNY=SIN(17.5*SIN(T))
570 END DEF
580 FOR I=0 TO N-1
590 A(I)=FNX(2*PI*I/N)
600 B(I)=FNY(2*PI*I/N)
610 NEXT I
620 DISP 'Computing...'
630 ! Main loops
640 FOR K=M-1 TO 0 STEP -1
650 K1=2^K
660 FOR L=0 TO N-1 STEP 2^(K+1)
670 S=FNR(L/K1)
680 X1=COS(2*PI*S/N) @ Y1=-SIN(2*PI*S/N
    )
690 FOR J=0 TO K1-1
700 X2=A(J+L+K1)*X1-B(J+L+K1)*Y1
710 Y2=A(J+L+K1)*Y1+B(J+L+K1)*X1
720 A(J+L)=A(J+L)+X2 @ B(J+L)=B(J+L)+Y2
730 A(J+L+K1)=A(J+L)-X2-X2
740 B(J+L+K1)=B(J+L)-Y2-Y2
750 NEXT J
760 NEXT L
770 NEXT K
780 ! Unscramble remainders
790 FOR I=0 TO N-1
800 IF FNR(I)<=I THEN 840               -Prevent double swap
810 X3=A(I) @ Y3=B(I)
820 A(I)=A(FNR(I)) @ B(I)=B(FNR(I))
830 A(FNR(I))=X3 @ B(FNR(I))=Y3
840 NEXT I
850 ! Last computation
860 FOR I=0 TO N-1
870 A(I)=A(I)/N @ B(I)=B(I)/N
880 NEXT I
```

PROGRAM LISTING

```
 890 ! Review the polynomial
 900 DISP 'Select series: ';CHR$(210);'e    -Select real or complex series
     al or ';CHR$(195);
 910 INPUT 'omplex?','R'; P$ @ P$=UPRC$(
     P$[1,1])
 920 ON POS('RC',P$)+1 GOTO 900,940,1030
 930 ! Real case
 940 FOR I=0 TO N/2
 950 IF I=0 THEN DISP 'A(0)=';A(0) @ Z=F
     NA @ GOTO 1000
 960 DISP USING "'A(',k,')=',k" ; I,A(I)
     +A(N-I) @ IF FNA=8 THEN I=MAX(I-1,0
     )
 970 IF I=0 THEN 950
 980 IF I=N/2 THEN 1000
 990 DISP USING "'B(',k,')='k" ; I,-(B(I
     )-B(N-I)) @ IF FNA=8 THEN 950
1000 NEXT I
1010 GOTO 1100
1020 ! Complex case
1030 FOR I=0 TO N-1
1040 DISP USING "'Real part ',a,'(',k,')
     =',k" ; CHR$(5),I,A(I)
1050 IF FNA=8 THEN I=MAX(I-1,0)
1060 DISP USING "'Imag part ',a,'(',k,')
     =',k" ; CHR$(5),I,B(I)
1070 IF FNA=8 THEN 1040
1080 NEXT I
1090 ! Run,view, or end                    -Program options
1100 DISP CHR$(210);'un again,';CHR$(214
     );'iew again,or';CHR$(197);
1110 INPUT 'nd?','R'; Q$ @ Q$=UPRC$(Q$[1
     ,1])
1120 ON POS('RVE',Q$)+1 GOTO 1100,110,90
     0,1130
1130 STOP
```

PROGRAM DESCRIPTION

ATTENUATED FOURIER SERIES

This program provides an approximation by a trigonometric polynomial with a finite number of terms to the Fourier expansion of a given function. A complex function $z(t) = x(t)+\sqrt{-1}y(t)$ which is periodic on $[0,2\pi]$ with period 2π and whose modulus squared is integrable, has a Fourier expansion

$$z(t) = \sum_m \hat{z}(m)e^{mt\sqrt{-1}} \qquad m=0, \pm1, \pm2...$$

The Fourier series is approximated by the trigonometric polynomial

$$p(t) = \sum_m \beta_m e^{mt\sqrt{-1}} \qquad m=0,...,n-1$$

in which

$$\beta_m = \frac{1}{n}\sum_k z\left(\frac{2\pi k}{n}\right) e^{-2\pi \frac{m}{n}k\sqrt{-1}} \quad 0\le k\le n-1$$

where n is the number of points in the approximation.

Roughly, the β_m approximate the $\hat{z}(m)$ where:

$$\hat{z}(m)= \frac{1}{2\pi}\int_0^{2\pi} z(t)\mathrm{EXP}(-mt\sqrt{-1})dt$$

Now the coefficients $\hat{z}(m)$ approach zero as m grows large. Since the only way to extend β_m is by being periodic in n, there is difficulty in approximation which is resolved by interpolating the points $Z\left(\frac{2\pi k}{n}\right)$. A simple linear approximation leads to coefficients c_m defined by:

$$c_o = \beta_o$$

$$c_m = (n/\pi m)^2(\mathrm{SIN}(\pi m/n))^2\beta_m \qquad m=1,...,n-1$$

The multipliers of the β_m are called attenuation factors and the resulting coefficients c_m now approach zero as m grows large.

PROGRAM DESCRIPTION

ATTENUATED FOURIER SERIES (continued)

Algorithm

```
Input n, m=log_2 (n), arrays a(512), b(512), or z(t)
For i=0 to n-1 step 1
Input α(i)=a(i) + √-1 b(i) or evaluate z(^(2π√-1)/n)
Next i
Def bit reversing function r(k)
For k=m-1 to 0 step -1
For p=0 to n-1 step 2^(k+1)
s = r(p/2^k)
ξ_1 = COS(2πs/n) + √-1 SIN(2πs/n)
For j=0 to 2^k-1 step 1
ξ_2 = ξ_1 ·α(j+p+2^k)
α(j+p) = α(j+p) + ξ_2
α(j+p+2^k) = α(j+p) - 2ξ_2
Next j
Next p
Next k
For i=0 to n-1 step 1
Swap α(i), α(r(i)) if r(i)>i
Next i
α(0) = α(0)/n
For i=1 to n-1
α(i)=(n/πi * sin(πi/n))^2 α(i)
Next i
For i=0 to n-1 step 1
Disp α(i)
Next i
Stop
```

SAMPLE PROBLEM

A) Estimate the first eight coefficients in the Fourier series for $X(t) = (\pi-t)^2/4$

SOLUTION

STEP	INSTRUCTIONS	DISPLAY	INPUT
A_1	Enter x(t)	290 DEF FNX(t)=(PI-T)^2/4	
	Enter y(t)	300 DEF FNY(t)=0	
2	Run program	* ATTENUATED FOURIER SERIES *	
3	Enter number of data points:	Number of data points?	16 [RTN]
		Evaluating function...	
		Computing...	
4	Select real series:	Attn. series: Real or Complex?R	[RTN]
5	View each point by pressing	A(0)=.828892557125	[RTN]
	[RTN]. Pressing [BACK] will	A(1)=.999999999995	[RTN]
	goto the previous point.	B(i)=	[RTN]
	In this example the B(i) can	A(2)=.24999999999	[RTN]
	be considered insignificant..	A(3)=.11111111111	[RTN]
		A(4)=6.25000000009E-2	[RTN]
		A(5)=4.00000000009E-2	[RTN]
		A(6)=2.77777777768E-2	[RTN]
		A(7)=2.04081632641E-2	[RTN]
		A(8)=1.56250000004E-2	[RTN]
6	End program.	Run again, View again, or End?R	E [RTN]

SAMPLE PROBLEM

B) Estimate the first eight coefficients of the Fourier expansion of $t(t-\pi)(t-2\pi)/12$ on $[0,2\pi]$

(Note the exact series is $\Sigma\frac{1}{k}3^{\sin(kt)}$ k=1,2,...)

SOLUTION

STEP	INSTRUCTIONS	DISPLAY	INPUT
B1	Enter x(t)	290 DEF FNX(T)=T*(T-PI)*(T-2*PI)/12	
	Enter y(t)	300 DEF FNY(T)=0	
2	Run program	* ATTENUATED FOURIER SERIES *	
3	Enter number of data points:	Number of data points?	16 [RTN]
		Evaluating function...	
		Computing...	
4	Select real series:	Attn series: Real of Complex?R	[RTN]
5	View each point by pressing	A(i)=	[RTN]
	[RTN]. Pressing [BACK] will	B(1)=.987115800974	[RTN]
	goto the previous point. In	B(2)=.118507431121	[RTN]
	this example the A(i) can be	B(3)=3.26508727055E-2	[RTN]
	considered insignificant.	B(4)=1.22718463038E-2	[RTN]
		B(5)=5.24786275016E-3	[RTN]
		B(6)=2.25918452119E-3	[RTN]
		B(7)=7.97068408518E-4	[RTN]
6	End program.	Run again, View again, or End?R	E [RTN]

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
1	Enter the function x(t) in 290	290 DEF FNX(T)=	
	Enter the function y(t) in 300	300 DEF FNY(T)=	
2	Run program	* ATTENUATED FOURIER SERIES *	
#	Enter number of data points:	Number of data points?	n [RTN]
	(Must be a power of 2)		
		Evaluating function...	
		Computing...	
4	Select attenuated series	Attn. series Real or Complex?R	R or C [RTN]
	real or complex:		
5	View each point by pressing		
	[RTN]. Pressing [BACK] will		
	goto the previous point.		
	If real:	A(i)=	[RTN] or
		B(i)=	[BACK]
	If complex:	Real part c(i)=	[RTN] or
		Imag part c(i)=	[BACK]
6	Select program options:	Run again, View again, or End?R	R,V, or E [RTN]
	To run the program again,		
	enter 'R' and goto step 3.		
	To view the answers again,		
	enter 'V' and goto step 4.		
	To end the program enter 'E'.		

VARIABLE NAMES

NAME	DESCRIPTION	NAME	DESCRIPTION
A(I),B(I)	Storage for vectors to be transformed	Y_1	For direct Fourier transform
N	Number of subdivision parts	X_2,Y_2	In place storage for computation
M	Log_2 (n)	X_3,Y_3	In place storage for swapping
R(X)	Bit reversing function	Q	Attenuation factors
A(X)	Answer function	Q$	String for RVE module
X(T)	Real part of object function	X	Dummy variable for R(X)
Y(T)	Imaginary part of object function	A,U	Temporary storage for K(X)
K_1	Simplifying index	B	Temporary storage for A(X)
X_1	Primitive root of unity	S	Temporary storage for value of R(X)

NOTES AND REFERENCES

Note: The attenuation factors correspond to linear interpolation. Another popular choice is found by interpolation with cubic splines. In this case the factors for lines 670 and 760 are:

$$Q_i = (n/\pi i SIN(\pi i/n))^4 * 3/(2+COS(2\pi i/n)),\ i\#0$$

For line 730 the change is: (2/π*SIN(π/2))^4*3/(2+COS(π/2))

References: 1. Henrici, P., FAST FOURIER METHODS IN COMPUTATIONAL COMPLEX ANALYSIS, (SIAM Review, V. 21, 1979, p. 481-527).

2. Eagle, A., ON THE RELATION OF THE FOURIER CONSTANTS ..., (Phil. Mag. S. 7, Vol. 5, 1928, p. 113-132).

PROGRAM LISTING

```
 10 ! ATTENUATED FOURIER SERIES
 20 ! Computes the Fourier series
 30 ! of a complex-valued function
 40 ! periodic on [0,2pi],using
 50 ! Cooley-Tukey and linear
 60 ! interpolation
 70 ! Revision 11/01/1982
 80 DIM A(512),B(512)
 90 DISP '*  ATTENUATED FOURIER SERIES
    *'
100 INPUT 'Number of data points?'; N @
    M=IP(LOG(N)/LOG(2))
110 IF FP(LOG(N)/LOG(2))#0 THEN BEEP @      -The number of points must be a
    DISP 'Incorrect number of data poin      power of 2
    ts!' @ GOTO 100
120 ! Bit reversing function                -Function to reverse the binary
                                             expansion of X
130 DEF FNR(X)
140 U=0
150 FOR P=0 TO M-1
160 A=RMD(X,2)
170 U=U+A*2^(M-1-P)
180 X=IP(X/2)
190 IF X<=1 THEN 210
200 NEXT P
210 U=U+X*2^(M-2-P)
220 FNR=U @ END DEF
230 ! Define answer function                -Wait for 'RTN' or 'BACK' key
240 DEF FNA
250 B=NUM(KEY$) @ IF B#13 AND B#8 THEN
    250
260 FNA=B @ END DEF
270 ! Periodic function
280 DISP 'Evaluating function...'
290 DEF FNX(T) = T*(T-PI)*(T-2*PI)/12       -User defined functions
300 DEF FNY(T) = 0
310 FOR I=0 TO N-1
320 A(I)=FNX(2*PI*I/N)
330 B(I)=FNY(2*PI*I/N)
340 NEXT I
350 DISP 'Computing...'
360 ! Main loops
370 FOR K=M-1 TO 0 STEP -1
380 K1=2^K
390 FOR L=0 TO N-1 STEP 2^(K+1)
400 S=FNR(L/K1)
410 X1=COS(2*PI*S/N) @ Y1=-SIN(2*PI*S/N
    )
420 FOR J=0 TO K1-1
430 X2=A(J+L+K1)*X1-B(J+L+K1)*Y1
440 Y2=A(J+L+K1)*Y1+B(J+L+K1)*X1
450 A(J+L)=A(J+L)+X2 @ B(J+L)=B(J+L)+Y2
460 A(J+L+K1)=A(J+L)-X2-X2
470 B(J+L+K1)=B(J+L)-Y2-Y2
```

PROGRAM LISTING

```
480 NEXT J
490 NEXT L
500 NEXT K
510 ! Unscramble remainders
520 FOR I=0 TO N-1
530 IF FNR(I)<=I THEN 570
540 X3=A(I) @ Y3=B(I)
550 A(I)=A(FNR(I)) @ B(I)=B(FNR(I))
560 A(FNR(I))=X3 @ B(FNR(I))=Y3
570 NEXT I
580 FOR I=0 TO N-1
590 A(I)=A(I)/N @ B(I)=B(I)/N
600 NEXT I
610 ! Attenuate and view                    -Select real or complex series
620 DISP 'Attn. series ';CHR$(210);'eal
     or ';CHR$(195);
630 INPUT 'omplex?','R'; P$ @ P$=UPRC$(
    P$[1,1])
640 ON POS('RC',P$)+1 GOTO 620,650,740
650 DISP 'A(0)=';A(0) @ IF FNA=13 THEN
    660 ELSE 650
660 FOR I=1 TO N/2-1
670 Q=(N/(PI*I)*SIN(PI*I/N))^2
680 DISP USING 690 ; 'A',I,(A(I)+A(N-I)
    )*Q @ IF FNA=8 THEN I=MAX(I-1,1)
690 IMAGE a,'(',k,')=',k
700 Q=(N/(PI*I)*SIN(PI*I/N))^2
710 DISP USING 690 ; 'B',I,-(B(I)-B(N-I
    ))*Q @ IF FNA=8 THEN 670
720 NEXT I
730 DISP USING 690 ; 'A',N/2,2*A(N/2)*(
    2/PI*SIN(PI/2))^2 @ IF FNA=8 THEN 6
    50 ELSE 850
740 FOR I=0 TO N-1                         -Complex answer
750 IF I=0 THEN Q=1
760 IF I>0 THEN Q=(N/(PI*I)*SIN(PI*I/N)
    )^2
770 DISP USING "'Real part c(',k,')=',k
    " ; I,A(I)*Q
780 IF FNA=8 THEN I=MAX(I-1,0) @ GOTO 7
    50
790 IF I=0 THEN Q=1
800 IF I>0 THEN Q=(N/(PI*I)*SIN(PI*I/N)
    )^2
810 DISP USING "'Imag part c(',k,')=',k
    " ; I,B(I)*Q
820 IF FNA=8 THEN I=MAX(I-1,0) @ GOTO 7
    90
830 NEXT I
840 ! Run,view, or end                     -Program options
850 DISP CHR$(210);'un again,';CHR$(214
    );'iew again,or';CHR$(197);
860 INPUT 'nd?','R'; Q$ @ Q$=UPRC$(Q$[1
    ,1])
```

PROGRAM LISTING

```
870 ON POS('RVE',Q$)+1 GOTO 850,100,620
    ,880
880 STOP
```

PROGRAM DESCRIPTION

SPHERICAL HARMONICS

This program computes the values of the spherical harmonics $Y_n^m(\theta,\phi)$ on the three dimensional sphere of unit radius and angles $0\leq\theta\leq\pi$ $0\leq\phi\leq 2\pi$. The functions Y_n^m satisfy the following relations:

$$Y_o^o = 1/\sqrt{4\pi}$$

$$Y_m^m = [(-1)^m/2^m m!] \quad \sqrt{(2m+1)!/4\pi} \quad (\mathrm{SIN}\ \theta)^m \mathrm{EXP}(m\phi\sqrt{-1})$$

$$Y_{m+1}^m = \sqrt{2m+3} \quad \mathrm{COS}\ \theta\ Y_m^m$$

$$\mathrm{COS}\ \theta\ Y_n^m = \sqrt{\frac{(n+m+1)(n-m+1)}{(2n+1)(2n+3)}} \quad Y_{n+1}^m + \sqrt{\frac{(n+m)(n=m)}{(2n+1)(2n-1)}} \quad Y_{n-1}^m$$

Algorithm: Input n, m, θ, ϕ; let $f_1=2$

For i = 1 to m

$f_1 = 2if_1/(2i+1)$

Next i

$f_1 = (-1)^m/\sqrt{2\pi f_1}$

$w=f_1\ \mathrm{SIN}(\theta)^m$

$u = w\ \mathrm{COS}(m\phi)$, $v = w\ \mathrm{SIN}(m\phi)$

If n=m then finish and display u,v

$u_1 = \sqrt{2m+3}\ \mathrm{COS}(\theta)\ u$, $v_1 = \sqrt{2m+3}\ \mathrm{COS}(\theta)\ v$

If n = m+1 then finish and display u_1, v_1

PROGRAM DESCRIPTION

SPHERICAL HARMONICS (continued)

For i = 1 to n-m-1

$$u_2 = \left[\mathrm{COS}(\theta)\ u_1 - u\sqrt{\frac{(m+i)^2-m^2}{4(m+i)^2-1}}\right]\sqrt{\frac{(2(m+i)+1)(2(m+i)+3)}{(i+2m+1)(i+1)}}$$

$$v_2 = \left[\mathrm{COS}(\theta)v_1 - v\sqrt{\frac{(m+i)^2-m^2}{4(m+i)^2-1}}\right]\sqrt{\frac{(2(m+i)+1)(2(m+i)+3)}{(i+2m+1)(i+1)}}$$

Update u, u_1, v_1, v

NEXT i

Disp u_2, v_2

Stop

SAMPLE PROBLEM

A) Calculate $Y_2^0\ (1,2) = Y_n^m(\theta,\phi)$

B) Calculate $Y_3^1\ (2,2) = Y_n^m(\theta,\phi)$

SOLUTION

STEP	INSTRUCTIONS	DISPLAY	INPUT
1	Run program	*** SPHERICAL HARMONICS ***	
2	Enter subscript	Subscript of harmonic?	2 [RTN]
3	Enter superscript	Superscript of harmonic?	0 [RTN]
4	Continue	Recheck sub/super or Continue?R	C [RTN]
5	Enter θ	First angular argument?	1 [RTN]
6	Enter ϕ	Second angular argument?	2 [RTN]
7	View result	Re(Y(0,2,1,2)) =	
		-3.91780206022E-2	[RTN]
		Im (Y(0,2,1,2)) =	
		0	[RTN]
8	Run again	Run again, View again, or End?R	[RTN]
9	Enter subscript	Subscript of harmonic?	1 [RTN]
10	Enter superscript	Superscript of harmonic?	3 [RTN]
	Sub/super reversed:	Superscript out of range!	
11	Enter subscript	Subscript of harmonic?	3 [RTN]
12	Enter superscript	Superscript of harmonic?	1 [RTN]
13	Check entries	Recheck sub/super or Continue?R	[RTN]
14		Subscript = 3	[RTN]
		Superscript = 1	[RTN]

SOLUTION

STEP	INSTRUCTIONS	DISPLAY	INPUT
15	Continue		C [RTN]
16	Enter θ	First angular argument?	2 [RTN]
17	Enter ϕ	Second angular argument?	2 [RTN]
18	View result	Re(Y(1,3,2,2)) =	
		-1.64004353583E-2	[RTN]
		Im (Y(1,3,2,2)) =	
		3.5835605032E-2	[RTN]
19	End program	Run again, View again, or End?R	E [RTN]

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
1	Run program	*** SPHERICAL HARMONICS ***	
2	Enter subscript, n	Subscript of harmonic?	n [RTN]
	a non-negative integer		
3	Enter superscript, m,	Superscript of harmonic?	m [RTN]
	a non-negative integer,		
	$\leq$n; for negative m,		
	$-n \leq m \leq 0$ see comments.		
4	Check sub/super or continue	Recheck sub/super or Continue?R	R or C [RTN]
	To check entries enter 'R':	Subscript = n	[RTN]/[BACK]
	Press [BACK] to change entry:	New subscript =	n [RTN]
		New superscript =	m [RTN]
5	Enter θ, $0 \leq \theta \leq \pi$	First angular argument?	θ [RTN]
6	Enter ϕ, $0 \leq \phi < 2\pi$	Second angular argument?	ϕ [RTN]
7	View result:	Re(Y(m,n,θ,ℓ)) =	
		number	[RTN]
		Im (Y(m,n,θ,ℓ)) =	
		number	[RTN]
8	Select program options	Run again, View again, or End?R	R,V, or E [RTN]
	To run again, enter 'R' and		
	goto step 2.		
	To review the result, enter 'V'		
	and goto step 7.		
	To end the program, enter 'E'.		

VARIABLE NAMES

NAME	DESCRIPTION	NAME	DESCRIPTION
FNA	Answer function	W	Factor
A	Dummy variable for FNA	F_1	Coefficient for Y
N	Subscript	U,V	Real/Imaginary parts of final values
M	Superscript	U_1,V_1	Real/Imaginary parts in recurrence, Y^m_{m+1}
T	The argument θ		
F	The argument ϕ	U_2,V_2	Real/Imaginary parts in R recurrence, Y^m_n
T$	String for sub/super check		
		Q$	String for RVE module

NOTES AND REFERENCES

References: 1. Sommerfeld, A., PARTIAL DIFFERENTIAL EQUATIONS IN PHYSICS, (Academic Press, New York, 1949).

2. Cohen-Tannondji, C., B. Diu, and F. Laloë, QUANTUM MECHANICS, (Vol. 1, John Wiley & Sons, New York, 1977).

Note: To compute Y^{-m}_n for m>0 use the relation:

Complex conjugate $(Y^m_n(\theta,\phi)) = (-1)^m Y^{-m}_n(\theta,\phi)$

PROGRAM LISTING

```
 10 ! SPHERICAL HARMONICS
 20 ! Compute the values of the
 30 ! spherical harmonics by
 40 ! recursion
 50 ! Revision 11/01/1982
 60 DISP '* * * SPHERICAL  HARMONICS *
    * *' @ OPTION ANGLE RADIANS
 70 ! Wait for return,back keys
 80 DEF FNA
 90 A=NUM(KEY$) @ IF A#13 AND A#8 THEN
    90
100 FNA=A @ END DEF
110 INPUT 'Subscript of harmonic?';N
120 IF FP(N)#0 OR N<0 THEN BEEP @ DISP
    'Invalid subscript!' @ GOTO 110
130 INPUT 'Superscript of harmonic?';M
140 IF FP(M)#0 THEN BEEP @ DISP 'Invali
    d superscript!' @ GOTO 130
150 IF M>N THEN BEEP @ DISP 'Superscrip
    t out of range!' @ GOTO 110
160 IF M<0 THEN BEEP @ DISP 'Enter abs(
    m) and see comments!' @ GOTO 130
170 ! Correct sub-superscript?
180 DISP CHR$(210);'echeck sub/super or
     ';CHR$(195);
190 INPUT 'ontinue?','R'; T$ @ T$=UPRC$
    (T$[1,1])
200 ON POS('RC',T$)+1 GOTO 180,210,240
210 DISP 'Subscript=';N @ IF FNA=8 THEN
     DISP 'New subscrip'; @ INPUT 't=';
     N @ GOTO 220
220 DISP 'Superscript=';M @ IF FNA=8 TH
    EN DISP 'New superscrip'; @ INPUT '
    t='; M @ GOTO 230
230 IF ABS(M)>N THEN BEEP @ DISP 'Super
    script out of range!' @ GOTO 110 EL
    SE GOTO 180
240 INPUT 'First angular argument?';T
250 IF T<0 OR T>PI THEN BEEP @ DISP 'An
    gle out of range!' @ GOTO 240
260 INPUT 'Second angular argument?';F
270 IF F<0 OR F>2*PI THEN BEEP @ DISP '
    Angle out of range!' @ GOTO 260
280 ! Compute Y(m,m)
290 IF M=0 THEN W=1/(2*SQR(PI)) @ GOTO
    360
300 F1=2
310 FOR I=1 TO M
320 F1=2*I*F1/(2*I+1)
330 NEXT I
340 F1=(-1)^M/SQR(2*PI*F1)
350 W=F1*SIN(T)^M
360 U=W*COS(M*F) @ V=W*SIN(M*F)
370 ! Compute harmonics by
```

-Enter subscript, superscript

-Option to correct entries

-Enter angular arguments

PROGRAM LISTING

```
380 ! recursion relation

390 ! Initalize
400 IF N=M THEN 520
410 U1=SQR(2*M+3)*COS(T)*U @ V1=SQR(2*M
    +3)*COS(T)*V
420 IF N=M+1 THEN U=U1 @ V=V1 @ GOTO 52
    0
430 ! Main loop for spherical harmonics
440 FOR I=1 TO N-M-1
450 U2=COS(T)*U1-U*SQR(((M+I)^2-M^2)/(4
    *(M+I)^2-1))
460 U2=SQR((2*(M+I)+1)*(2*(M+I)+3)/((I+
    M+M+1)*(I+M-M+1)))*U2
470 V2=COS(T)*V1-V*SQR(((I+M)^2-M^2)/(4
    *(I+M)^2-1))
480 V2=SQR((2*(I+M)+1)*(2*(I+M)+3)/((I+
    M+M+1)*(I+M-M+1)))*V2
490 U=U1 @ U1=U2 @ V=V1 @ V1=V2
500 NEXT I
510 U=U2 @ V=V2
520 DISP 'Re(Y(';M;',';N;',';T;',';F;')
    )=' @ IF FNA=13 THEN DISP U ELSE 52
    0                                    -Display results
530 IF FNA=8 THEN 520
540 DISP 'Im(Y(';M;',';N;',';T;',';F;')
    )=' @ IF FNA=13 THEN DISP V ELSE 54
    0
550 IF FNA=8 THEN 520
560 ! Run,view,or end                    -Program options
570 DISP CHR$(210);'un again,';CHR$(214
    );'iew again,or';CHR$(197);
580 INPUT 'nd?','R'; Q$ @ Q$=UPRC$(Q$[1
    ,1])
590 ON POS('RVE',Q$)+1 GOTO 570,110,520
    ,600
600 STOP
```

PROGRAM DESCRIPTION

ELLIPTIC INTEGRALS

The general form of the elliptic integral of the second type is given by:

$$el2(x,k',a,b) = \int_0^x \frac{a + bt}{(1+t^2)\sqrt{(1+t^2)(1+k'^2t^2)}}\,dt$$

in which $k' \neq 0$. Using the trigonometric substitution $t=\tan\theta$ the standard elliptic integrals are recovered. With $x=\tan\phi$ and $k=\sqrt{1-k'^2}$ (k is called the modulus) the standard elliptic integrals are

$$F(\phi,k) = \int_0^\phi \frac{d\theta}{\sqrt{1-k^2\sin^2\theta}} = el2(\tan\phi,k',1,1)$$

$$E(\phi,k) = \int_0^\phi \sqrt{(1-k^2\sin^2\theta)}d = el2(\tan\phi,k',1,k'^2)$$

The functions F,E are called incomplete elliptic integrals while

$$F(k) = F(\pi/2,k), \quad E(k) = E(\pi/2,k)$$

are called complete elliptic integrals.

Although the general integral el2 contains the complete integrals as special cases with upper limit $x=10^{22}(\phi\cong\pi/2)$, a separate routine is used to compute F(k) and E(k).

Algorithm: If complete integral goto α
If general integral goto β
Input modulus k and angle ϕ
$x=\tan\phi, k'=\sqrt{1-k^2}$, $a=1$
If $E(\phi,k)$ then $b=k'^2$; If $F(\phi,k)$ then $b=1$; goto β_1

β: Input k', x, a, b

β_1: $m=1$, $n=k$, $x_0=1/|x|$, $c=0$, $d=|x|/(a+x^2)$
$p=\sqrt{((1+(k'x)^2)/(1+x^2))}$, $i=0$, $t=1/2$, $s=0$, $h=a-b$
$m_2=1$, $n_2=1$

PROGRAM DESCRIPTION

ELLIPTIC INTEGRALS (continued)

γ: $m_1=n+m, n_1=2\sqrt{n\ m}, a_1=(b/m+a)/2$

$b_1=a\ n+b, x_1 = -mn/x_o + x_o,\ P_1= mn/p + P$

$c_1=(d/p+2)/2, d_1=m\ n\ c/p+d,$

$s=s+t(1-SGN(x_o))$

If $ABS(1-n_2/m_2)<10^{-6}$ then ψ_i

Update $n_2, m_2, n, m, a, b, x_o, p, d, c, t, i$, goto γ

ψ_i: $f=ATN(m/x_o)+(2^i\ s+(1-SGN(x_o))\pi/2$

$e=a_1\ f/m+h\ c$; disp e

Stop

α: Input k; let $a_2=1, b_2=k', f_1=1,\ g=k^2$

α_1: $a_3=(a_2+b_2)/2,\ c_2=a_3-b_2$

If $c_2^2<10^{-13}$ then ψ_c

Update: $f_1=2f_1,\ g=g+f_1\ c_2^2, b_2=\sqrt{a_2\ b_2}, a_2=a_3$ goto α_1

ψ_c: $i_1=\pi/2a_2$, disp i_1

$i_2= (1-g/2)i_1$, disp i_2

Stop

SAMPLE PROBLEM

A) Compute $F(k) = \int_0^{\pi/2} d\theta/\sqrt{1-k^2 \sin^2\theta}$, with

$k = 1/\sqrt{2} = .707106781188$ and compute

$E(k) = \int_0^{\pi/2} \sqrt{1-k^2 \sin^2\theta}\, d\theta$ and check

against el2(10^{23}, .707106781188,1,1)

and el2(10^{23}, .707106781188,1,.5)

SOLUTION

STEP	INSTRUCTIONS	DISPLAY	INPUT
A 1	Run program	*** ELLIPTIC INTEGRALS ***	
2	Select complete integral	Complete/Incomplete integral?C	[RTN]
3	Enter modulus (1/SQR(2))	Value of modulus?.707106781188	[RTN]
4	Results	F(.707106781188)=1.85407467721	[RTN]
		E(.707106781188)=1.35064388098	[RTN]
5	Run again	Run again, View again, or End?R	[RTN]
6	Select incomplete integral	Complete/Incomplete integral?C	I [RTN]
7	Select general form	Trig or General form?T	G [RTN]
8	Enter modulus (1/SQR(2))	Complementary modulus?.707106781188	[RTN]
9	Enter parameters	Parameter a?	1 [RTN]
		Parameter b?	1 [RTN]
10	Approximate infinite upper	Upper limit of integral?	1E23 [RTN]
	limit		
11	Result	el2(1.E23,.707106781188,1,1)=	[RTN]
		1.8540746773	[RTN]
12	Run again	Run again, View again, of End?R	[RTN]
13	Select incomplete integral	Complete/Incomplete integral?C	I [RTN]

SOLUTION

STEP	INSTRUCTIONS	DISPLAY	INPUT
14	Select general form	Trig or General form?T	G [RTN]
15	Enter modulus (1/SQR(2))	Complementary modulus?.707106781188 [RTN]	
16	Enter parameters	Parameter a?	1 [RTN]
		Parameter b?	.5 [RTN]
17	Approximate infinite upper	Upper limit of integral?	1E23 [RTN]
	limit		
18	Result	e12(1.E23,.707106781188,1,.5)=	[RTN]
		1.35064388105	[RTN]
19	End program.	Run again, View again, or End?R	E [RTN]

SAMPLE PROBLEM

B) Compute el2(10^{12}, 10^{-11},1,1)

SOLUTION

STEP	INSTRUCTIONS	DISPLAY	INPUT
B 1	Run program	*** ELLIPTIC INTEGRALS ***	
2	Select incomplete integral	Complete/Incomplete?C	I [RTN]
3	Select general form	Trig or General form?T	G [RTN]
4	Enter complementary modulus	Complementary modulus?	1E-11 [RTN]
5	Enter parameters	Parameter a?	1 [RTN]
		Parameter b?	1 [RTN]
6	Enter upper limit of integral	Upper limit of integral?	1E12 [RTN]
7	Result	el2(1.E12,.00000000001,1,1)=	[RTN]
		26.614896305	[RTN]
8	Run again	Run again, View again, or End?R	[RTN]
9	Select incomplete integral	Complete/Incomplete integral?C	I [RTN]
10	Select general form	Trip or General form?T	G [RTN]
11	Enter complementary modulus	Complementary modulus?	1E-11 [RTN]
12	Enter parameters	Enter parameter a?	1 [RTN]
		Enter parameter b?	1 [RTN]
13	Enter upper limit	Upper limit of integral?	1E9 [RTN]
14	Result	el2(1000000000,.00000000001,1,1)=	[RTN]
		21.4163880183	[RTN]
15	End program	Run again, View again, or End?R	E [RTN]

SAMPLE PROBLEM

C) Compute the length of the ellipse $x^2/16+y^2/4 = 1$

The modulus k = the eccentricity = $\frac{\sqrt{a^2-b^2}}{a}$

while $k' = b/a, k'^2 = b^2/a^2$. Since a = 4 and

b = 2; we get $k' = .5$ and $k'^2 = .25$. The

length = 4ae(k) = 16 el2(10^{23},.5,1,.25)

SOLUTION

STEP	INSTRUCTIONS	DISPLAY	INPUT
C_1	Run program	*** ELLIPTIC INTEGRALS ***	
2	Select incomplete integral	Complete/Incomplete integral?C	I [RTN]
3	Select general form	Trig or General form?T	G [RTN]
4	Enter complementary modulus	Complementary modulus?	.5 [RTN]
5	Enter parameters	Parameter a?	1 [RTN]
		Parameter b?	.25 [RTN]
6	Approximate infinite upper	Upper limit of integral?	1E23 [RTN]
	limit		
7	Result	el2(1E23,.5,1,.25)=	[RTN]
		1.21105602757	[RTN]
8	End program	Run again, View again, or End?R	E [RTN]
9	Final answer is 16*result:		E*16 [RTN]
		19.3768964411	

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
1	Run program	*** ELLIPTIC INTEGRALS ***	
2	Select complete or incomplete	Complete/Incomplete integral?C	C or I [RTN]
	integral:		
	For incomplete integral,		
	goto 6.		
3	Enter value of modulus K:	Value of modulus?	K [RTN]
4	View results F(K), E(K):	F(K)=	[RTN]
		E(K)=	[RTN]
5	Goto step 16.		
6	Select trig substitution or	Trig or General form?T	T or G [RTN]
	general form:		
	For the general form goto 12.		
7	Enter modulus K:	Value of modulus?	K [RTN]
8	Enter the angle ϕ:	Value of angle?	ϕ [RTN]
9	Select function:	E function or F function?F	E or F [RTN]
10	View result:	E(F2,K)=	[RTN]
11	Goto step 16.		
12	Enter the complementary	Complementary modulus?	K' [RTN]
	modulus:		
13	Enter parameters:	Enter parameter a?	A [RTN]
		Enter parameter b?	B [RTN]
14	Enter the upper limit of the	Upper limit of integral?	X [RTN]
	integrand:		
15	View result:	e12(x,k',a,b)=	[RTN]
		E	[RTN]

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
16	Program options:	Run again, View again, or End?R	R,V, or E [RTN]
	To run the program again		
	enter 'R'.		
	To view the result again		
	enter 'V'.		
	To end the program enter 'E'.		

VARIABLE NAMES

NAME	DESCRIPTION	NAME	DESCRIPTION
FNA	Answer function	S_1	Accumulator
Z1	Dummy for FNA	T	Factor = 2 power
K	Modulus	I	Exponent for 2
K_1	Complementary modulus	E.F	Final computation of el2
F_2	Angle	A2	Iterated variables in complete integral
A	Parameters in el2	B2	
B	Parameters in el2	F1	
X	Upper limit of integral	G	
Z	Flag for various types of integral	C2	
M,M, N,N, XO,X, C,C, D,D, P,P	Variables in iteration	I1,I2	Final answers for E(k), F(k)
		A$	Test string for complete/ incomplete
		B$	Test string for trig/ general form
		C$	Test for E,F
		Q$	Test for RVE
		M2,M2	Save values for stopping test

NOTES AND REFERENCES

References: 1. Bulirsch, R., NUMERICAL CALCULATION OF ELLIPTIC INTEGRALS AND ELLIPTIC FUNCTIONS, (Num. Math., V. 7, 1965, pp. 78-90).

2. Henrici, P., COMPUTATIONAL ANALYSIS WITH THE HP-25 POCKET CALCULATOR, (John Wiley & Sons, New York, 1977).

PROGRAM LISTING

```
 10 ! ELLIPTIC INTEGRALS
 20 ! OF THE FIRST AND SECOND TYPE
 30 ! Computes complete and
 40 ! incomplete elliptic integrals
 50 ! of the first and second
 60 ! type by iteration
 70 ! Revision 11/01/1982
 80 DISP '* * *  ELLIPTIC INTEGRALS  *
    * *'
 90 ! Define answer function                  -Wait for 'RTN' or 'BACK' keys
100 DEF FNA
110 Z1=NUM(KEY$) @ IF Z1#13 AND Z1#8 TH
    EN 110
120 FNA=Z1 @ END DEF
130 ! Test for compl/inc integral             -Select complete or incomplete
                                              integral
140 DISP CHR$(195);'omplete / ';CHR$(20
    1);
150 INPUT 'ncomplete integral?','C'; A$
     @ A$=UPRC$(A$[1,1])
160 ON POS('CI',A$)+1 GOTO 140,650,170
170 ! Trig/general form for inc integra      -Select Trig or General form
    l
180 DISP CHR$(212);'rig or ';CHR$(199);
190 INPUT 'eneral form?','T'; B$ @ B$=U
    PRC$(B$[1,1])
200 ON POS('TG',B$)+1 GOTO 180,220,330
210 ! Input for trig form
220 INPUT 'Value of modulus?';K
230 IF K>=1 OR K<0 THEN BEEP @ DISP 'In
    correct value for modulus!' @ GOTO
    220
240 INPUT 'Value of angle?'; F2 @ IF F2
    =0 THEN DISP 'E=0 and F=0' @ GOTO 8
    00
250 IF F2>=PI/2 OR F2<0 THEN BEEP @ DIS
    P 'Incorrect angle!' @ GOTO 240
260 X=TAN(F2) @ K1=SQR(1-K*K) @ A=1 @ B
    =1 @ Z=1
270 ! Test for E or F                        -Select E or F function
280 DISP CHR$(197);' function or ';CHR$
    (198);
290 INPUT ' function?','F'; C$ @ C$=UPR
    C$(C$[1,1])
300 ON POS('FE',C$)+1 GOTO 280,390,310
310 Z=0 @ B=K1*K1 @ GOTO 390
320 ! Input paramters                        -Enter modulus, parameters
330 INPUT 'Complementary  modulus?';K1
340 INPUT 'Parameter a?'; A @ A4=A
350 INPUT 'Parameter b?'; B @ B3=B
360 INPUT 'Upper limit of integral?'; X      -Enter upper limit
     @ IF X=0 THEN E=0 @ Z=2 @ GOTO 590
370 Z=2
380 ! Initialize variables
```

PROGRAM LISTING

```
390 M=1 @ N=ABS(K1) @ X0=1/ABS(X) @ C=0
     @ D=ABS(X)/(1+X*X) @ P=SQR((1+(K1*
    X)^2)/(1+X*X))
400 I=0 @ T=1/2 @ S=0 @ H=A-B @ M2=1 @
    N2=2
410 ! Main iteration
420 M1=N+M
430 N1=2*SQR(N*M)
440 A1=(B/M+A)/2
450 B1=A*N+B
460 X1=-M*N/X0+X0
470 P1=M*N/P+P
480 C1=(D/P+C)/2
490 D1=M*N*C/P+D
500 S=S+T*(1-SGN(X0))
510 ! Test for completion
520 IF ABS(1-N2/M2)<.000001 THEN 570
530 ! Update variables
540 N2=N @ M2=M
550 N=N1 @ M=M1 @ A=A1 @ B=B1 @ X0=X1 @
     P=P1 @ D=D1 @ C=C1 @ T=T/2 @ I=I+1
     @ GOTO 420
560 ! Final calculation
570 F=ATN(M/X0)+(2^I*S+(1-SGN(X0))/2)*P
    I/2
580 E=A1*F/M+H*C1
590 IF Z=2 THEN DISP USING "'el2(',k,',
    ',k,',',k,',',k,')='" ; X,K1,A4,B3
    @ GOTO 620                          -Display result
600 IF Z=1 THEN DISP 'F(';F2;',';K;')='
     @ GOTO 620
610 IF Z=0 THEN DISP 'E(';F2;',';K;')='
     @ GOTO 620
620 IF FNA=13 THEN 630 ELSE 620
630 DISP E @ IF FNA=8 THEN 590 ELSE 800
640 ! Complete elliptic integrals        -Complete elliptic integrals
650 INPUT 'Value of modulus?'; K @ Z=3
660 IF K>=1 OR K<0 THEN BEEP @ DISP 'In
    correct value for modulus!' @ GOTO
    650
670 A2=1 @ B2=SQR(1-K*K) @ F1=1 @ G=K*K
680 A3=(A2+B2)/2 @ C2=A3-B2
690 IF ABS(C2*C2)<1.E-13 THEN 750
700 F1=2*F1
710 G=G+F1*C2*C2
720 B2=SQR(A2*B2)
730 A2=A3
740 GOTO 680
750 I1=PI/(2*A2) @ DISP USING "'F(',k,'
    )=',k" ; K,I1                       -Display results
760 IF FNA=8 THEN 750
770 I2=(1-G/2)*I1 @ DISP USING "'E(',k,
    ')=',k" ; K,I2
780 IF FNA=8 THEN 770 ELSE 800
```

PROGRAM LISTING

```
790 ! RVE module
800 DISP CHR$(210);'un again,';CHR$(214
    );'iew again,or';CHR$(197);
810 INPUT 'nd?','R'; Q$ @ Q$=UPRC$(Q$[1
    ,1])
820 ON POS('RVE',Q$)+1 GOTO 800,140,830
    ,840
830 ON Z+1 GOTO 610,600,590,750
840 STOP
```

-Program options

PROGRAM DESCRIPTION

BESSEL FUNCTIONS: ASYMPTOTIC EXPANSION

This program computes the values for large z of the Bessel functions $J_m(z)$ and $Y_m(z)$.

Let $(a)_k := a(a+1)\ldots(a+k-1)$ for k a non-negative integer. The Bessel functions of the first order, $J_m(z)$, and second order, $Y_m(z)$ are defined asymptotically by:

$$J_m(z) = \sqrt{2/\pi z}\ \left(P_m(z)\mathrm{COS}(z-m\pi/2-\pi/4)-Q_m(z)\mathrm{SIN}(z-m\pi/2-\pi/4)\right)$$

$$Y_m(z) = \sqrt{2/\pi z}\ \left(Q_m(z)\mathrm{COS}(z-m\pi/2-\pi/4)+P_m(z)\mathrm{SIN}(z-m\pi/2-\pi/4)\right)$$

in which $P_m(z)$ and $Q_m(z)$ for large values of z are given by:

$$P_m(z) = \sum_n (-1)^n (\tfrac{1}{2}-m)_{2n}(\tfrac{1}{2}+m)_{2n}/((2n)!(2z)^{2m}),\quad n=0,1,\ldots$$

$$Q_m(z) = \sum_n (-1)^n (\tfrac{1}{2}-m)_{2n+1}(\tfrac{1}{2}+m)_{2n+1}/((2n+1)!(2z)^{2m+1}),\quad n=1,2,\ldots$$

```
Algorithm:  Input m,z
            Let n=0, c=1, a=1, b=0
            DEF q(w) = (-½-m+w)(-½+m+w)/(2zw)
        α:  n=n+1, c=q(n) c, q=q(n)
            If MOD(n,2)=0 then a=a+(-1)^(n/2) c, goto β
            b=b+(-1)^((n+1)/2) c
        β:  If q(n)=0 then γ
            If n small enough then α
        γ:  u=z-mπ/2-π/4, v= √(2/(πz))
            r=v(a COS(u) - b SIN (u))
            s=v(b COS(u) + a SIN(u))
            Disp J(m,z) = r
            Disp Y(m,z) = s
            Stop
```

SAMPLE PROBLEM

Estimate $J_0(17)$, $Y_0(17)$

Estimate $J_{11/2}(6)$, $Y_{11/2}(6)$

SOLUTION

STEP	INSTRUCTIONS	DISPLAY	INPUT
A_1	Run program	*** BESSEL FUNCTIONS ***	
2	Enter order	Order of Bessel function?	0 [RTN]
3	Enter argument	Argument of Bessel function?	17 [RTN]
		Computing...	
4	Answer	J(0,17) = .169854252151	[RTN]
		Y(0,17) = -926371984416E-2	[RTN]
5	Run again	Run again, View again, or End?R	[RTN]
6	Enter order	Order of Bessel function	5.5 [RTN]
7	Enter argument	Argument for Bessel function?	6 [RTN]
		Computing...	
8	Answer	J(5.5,6) = .309778760825	[RTN]
		Y(5.5,6) = -.319846110638	[RTN]
9	End program	Run again, View again, or End?	E [RTN]

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
1	Run program	**** BESSEL FUNCTIONS ****	
2	Enter order	Order of Bessel function?	M [RTN]
3	Enter argument, Z_0	Argumant of Bessel function?	Z [RTN]
4	Answer	J(m,z) =	[RTN]
		Y(m,z) =	[RTN]
5	Select program options:	Run again, View again, or End?	R,V, or E [RTN]
	To run again enter 'R' and		
	goto step 2.		
	To view the answer again		
	enter 'V' and goto step 4.		
	To end the program, enter 'E'.		

VARIABLE NAMES

NAME	DESCRIPTION	NAME	DESCRIPTION
FNA	Answer function	N	Counter
D	Dummy variable for FNA	A,B	Accumulators for sum of series P,Q
FNG	Multiplier function	C	Main term of series P,Q
M	Order of Bessel function	U	Simplify argument for sine and cosine
Z	Argument of Bessel function	V	Factor for J,Y
W	Dummy variable for FNQ	R	Final computations
Q$	Input string for RVE module	S	

NOTES AND REFERENCES

References: 1. Henrici, P., APPLIED AND COMPUTATIONAL COMPLEX ANALYSIS, (Vol. 2, John Wiley & Sons, New York, 1977).

2. Henrici, P., COMPUTATIONAL ANALYSIS WITH THE HP-25 POCKET CALCULATOR, (John Wiley & Sons, New York, 1977).

PROGRAM LISTING

```
 10 ! BESSEL FUNCTIONS
 20 ! (Asymptotic expansion)
 30 ! Approximates the values of
 40 ! Bessel functions using an
 50 ! asymptotic expansion.
 60 ! Works best for large
 70 ! values of the argument.
 80 ! Revision 11/01/1982
 90 DISP '* * * * BESSEL FUNCTIONS * *
    * *' @ OPTION ANGLE RADIANS
100 INPUT 'Order of Bessel function?';M      -Enter order, argument
110 INPUT 'Argument of Bessel function?
    ';Z
120 IF Z<=0 THEN BEEP @ DISP 'Invalid a
    rgument' @ GOTO 110
130 DISP 'Computing...'
140 ! Initialize
150 B,N=0 @ A,C=1
160 ! Define multiplier function
170 DEF FNQ(W) = 1/(W*(2*Z))*(-1/2-M+W)
    *(-1/2+M+W)
180 ! Define answer function                 -Waits for 'RTN' or 'BACK' key
190 DEF FNA
200 D=NUM(KEY$) @ IF D#13 AND D#8 THEN
    200
210 FNA=D @ END DEF
220 ! Main loop
230 N=N+1
240 C=FNQ(N)*C @ Q=FNQ(N)
250 IF MOD(N,2)=0 THEN A=A+(-1)^(N/2)*C
     @ GOTO 270
260 B=B+(-1)^IP((N+1)/2)*C
270 IF FNQ(N)=0 THEN 300
280 IF N<21 THEN 230                         -Sets number of terms computed
                                              in the series for p,q
290 ! Actual calculation of values
300 U=Z-M*PI/2-PI/4 @ V=SQR(2/(PI*Z))
310 R=V*(A*COS(U)-B*SIN(U))
320 S=V*(B*COS(U)+A*SIN(U))
330 DISP USING "'J(',k,',',k,')=',k" ;       -Display results
    M,Z,R
340 ! Wait for RTN key
350 IF FNA#13 THEN 350
360 DISP USING "'Y(',k,',',k,')=',k" ;
    M,Z,S
370 IF FNA#13 THEN 350
380 ! Program options                        -Options to run agin, view
                                              again, or end.
390 DISP CHR$(210);'un again, ';CHR$(21
    4);'iew again, or ';CHR$(197);
400 INPUT 'nd?','R'; Q$ @ Q$=UPRC$(Q$[1
    ,1])
410 ON POS('RVE',Q$)+1 GOTO 390,100,330
    ,420
420 DISP @ STOP
```

PROGRAM DESCRIPTION

BESSEL FUNCTIONS: BACKWARD RECURRENCE

This program computed approximate values of the Bessel functions $J_m(z)$, which are solutions of the differential equation:

$$z^2 \frac{d^2}{dz^2} J_m(z) + z \frac{d}{dz} J_m(z) + (z^2-m^2)J_m(z) = 0.$$

We can approximate the $J_m(z)$ by finding a sequence of numbers Y_m that also satisfies the backward recurrence relation:

$$Y_{m+1} - \frac{2m}{Z} Y_m + Y_{m-1} = 0.$$

$J_m(z)$ is approximately proportional to Y_m, with the constant of proportionality c, given by:

$$c = Y_o + 2Y_2 + 2Y_4 + \ldots \quad .$$

Algorithm:

```
Input z, m, k
Initialize y(k+1)=0, y(k)=1, c=0
For i = k to 1 step -1
y(i-1) = 2iy(i)/z-y(i+1)
If i = 1 then c=c+y(0)
If MOD(i,2)=0 then c=c+2y(i)
Next i
For i = 0 to m
y(i) = y(i)/c
Disp J_i(z) = y(i)
Next i
Stop
```

SAMPLE PROBLEM

Compute $J_m(1)$ for m=0, 1, 2, 3, 4, 5

SOLUTION

STEP	INSTRUCTIONS	DISPLAY	INPUT
1	Run program	*** BESSEL FUNCTIONS ***	
2	Enter argument	Argument of Bessel function?	1 [RTN]
3	Enter iterations	Number of iterations?	30 [RTN]
4	Enter largest order	Order of Bessel function?	5 [RTN]
5	Answer - review lower	J(0,1) = .765197686559	[RTN]
	order function	J(1,1) = .440050585745	[RTN]
		J(2,1) = .114903484932	[RTN]
		J(3,1) = 1.95633539827E-2	[RTN]
		J(4,1) = 2.47663896411E-3	[RTN]
		J(5,1) = 2.49757730211E-4	[RTN]
6	End program	Run again, View again, or End?R	E [RTN]

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
1	Run program	**** BESSEL FUNCTIONS ****	
2	Enter argument	Argument of Bessel function?	z [RTN]
3	Enter number of iterations	Number of iterations?	k [RTN]
4	Enter order of function	Order of Bessel function?	m [RTN]
5	View answer	J(m,z) =	[RTN]/[BACK]
	Press [RTN] to view		
	next answer, press [BACK]		
	to view previous answer.		
6	Select program options:	Run again, View again, or End?R	R,V, or E [RTN]
	To run the program again		
	enter 'R' and goto step 2.		
	To view the answer again,		
	enter 'V' and goto step 5.		
	To end the program enter 'E'.		

VARIABLE NAMES

NAME	DESCRIPTION	NAME	DESCRIPTION
FNA	Keyboard function	K	Number of iterates
A	Dummy variable for FNA	Y(65)	Array for recurrence, iterates
Z	Argument for Bessel function	C	Proportionality constant
M	Order of Bessel function	Q$	Input string for RVE module

NOTES AND REFERENCES

Notes:

1. The maximum number of iterations is 64. This can be altered by changing the dimension statement on line 60.

2. The number of iterations must be larger than or equal to the order of the Bessel Function.

References:

1. Gautschi, W., COMPUTATIONAL ASPECTS OF THREE-TERM RECURRENCE RELATIONS, (SIAM Review, Vol. 9, 1967, p.24-82).

2. Henrici, P., COMPUTATIONAL ANALYSIS WITH THE HP-25 POCKET CALCULATOR, (Wiley-Interscience, New York, 1977).

PROGRAM LISTING

```
 10 ! BESSEL FUNCTIONS
 20 ! Approximate Bessel functions
 30 ! using backward recurrence
 40 ! relations.
 50 ! Revision 11/01/1982
 60 DIM Y(65)
 70 DISP '  * * * BESSEL FUNCTIONS * *
    *'
 80 ! Define keyboard function                  -Waits for 'RTN' or 'BACK' key
 90 DEF FNA
100 A=NUM(KEY$) @ IF A#13 AND A#8 THEN
    100
110 FNA=A @ END DEF
120 INPUT 'Argument of Bessel function?        -Enter argument, number of
    ';Z                                         iterations
130 INPUT 'Number of iterations?';K            -Maximum number of iterations
                                                is 64- see line 180
140 INPUT 'Order of Bessel function?';M
150 IF M>K THEN BEEP @ DISP 'Invalid or
    der/iterate relation!' @ GOTO 140
160 IF M<0 OR FP(M)#0 THEN BEEP @ DISP
    'Invalid order!' @ GOTO 140
170 DISP 'Computing...'
180 ! Initialize iterate array
190 ! and proportionality constant.
200 Y(K+1)=0 @ Y(K)=1 @ C=0
210 ! Main loop
220 FOR I=K TO 1 STEP -1
230 Y(I-1)=2*I*Y(I)/Z-Y(I+1)
240 IF I=1 THEN C=C+Y(0) @ GOTO 270
250 IF MOD(I,2) THEN 270
260 C=C+2*Y(I)
270 NEXT I
280 FOR I=0 TO M
290 Y(I)=Y(I)/C
300 NEXT I
310 ! Display lower order values
320 FOR I=0 TO M
330 DISP USING "'J(',k,',',k,')=',k" ;
    I,Z,Y(I)
340 IF FNA=8 THEN I=MAX(I-1,0) @ GOTO 3
    30
350 NEXT I
360 DISP CHR$(210);'un again, ';CHR$(21   -Program options
    4);'iew again, or ';CHR$(197);
370 INPUT 'nd?','R'; Q$ @ Q$=UPRC$(Q$[1
    ,1])
380 ON POS('RVE',Q$)+1 GOTO 360,120,320
    ,390
390 STOP
```

PROGRAM DESCRIPTION

GAMMA FUNCTION

This program computes the values of the Gamma function

$$\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt$$

which extends the definition of factorial n! (the product of the first n integers for n a positive integer)

$$\Gamma(x) = (x-1)!$$

to all numbers x such that x>0.

Integration by parts yields $\Gamma(x+1) = x\Gamma(x)$ and this recursive relation in turn implies $\Gamma(x) = \Gamma(x+n)/x(x+1) \ldots (x+n-1)$.

Stirling's formula yields the asymptotic approximation for $\Gamma(x)$:

$$\log\Gamma(x) = (x-\tfrac{1}{2})\log(x) - x + \log\sqrt{2\pi} + b_2/2x + (b_4/3)4x^3 + (b_6/5)6x^5 + \ldots$$

in which b_2, b_4, b_6,... are the Bernoulli numbers.

Algorithm for $\Gamma(x)$:

If x is a positive integer compute $\Gamma(x)=(x-1)!$
as a product of integers.

If x is a real number ≥8 use Stirling's approximation
to compute $\Gamma(x)$

If x is a real number <8 use the recursive formula and Stirling's
formula to compute $\Gamma(x)$.

Disp $\Gamma(x)$

Stop

SAMPLE PROBLEM

A) The formula for the area of the (n-1) - dimensional unit sphere in n-space is given by the formula:

$$2\pi^{n/2}/\Gamma(n/2)$$

Compute the volume of the unit sphere in 4-space. Note that the sphere is the boundary of the unit ball in n-space, so $B_n=\{X:X\varepsilon\,\mathbb{R}^n,\ ||X||\leq 1\}$ while $S_{n-1} = \{X:X\varepsilon\,\mathbb{R}^n,\ ||X||=1\}$ where $||X|| = \sqrt{X_1{}^2+\ldots+X_n{}^2}$.

To solve this problem note that the program terminates with a STOP statement instead of an END statement. This means that the variables used in the program are available for inspection. The variable G contains the final answer for gamma, and can be used in keyboard calculations.

SOLUTION

STEP	INSTRUCTIONS	DISPLAY	INPUT
A_1	Run program	* * * * GAMMA FUNCTION * * * *	
2	Enter argument:	Arg for gamma function?	2 [RTN]
	n=4 so enter n/2=2		
3	View answer	Γ(2)=1	[RTN]
4	End program	Run again, View again, or End?R	E [RTN]
5	Use variable G to complete the		
	problem	>2*PI^2/G [RTN]	
6	Final answer	19.7392088022	

SAMPLE PROBLEM

B) The volume of the n-dimensional unit ball is 1/n area of (n-1) dimensional unit sphere, hence equals:

$$\Pi^{n/2}/\Gamma(n/2+1)$$

Compute the volume of the unit ball in 3-space.

SOLUTION

STEP	INSTRUCTIONS	DISPLAY	INPUT
B_1	Run program	* * * * GAMMA FUNCTION * * * *	
2	Enter argument:	Arg for gamma function?	2.5 [RTN]
3	View answer	Γ(2.5)=1.32924038819	[RTN]
4	End program	Run again, View again, or End?R	E [RTN]
5	Use variable G to complete the		
	problem:	>PI^(3/2)/G [RTN]	
6	Final answer	4.18879020475	

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
1	Run program	* * * * GAMMA FUNCTION * * * *	
2	Enter argument:	Arg for gamma function?	X [RTN]
3	View answer	Γ(X)=Γ	[RTN]
4	Select program options:	Run again, View again, or End?R	R, V, or E [RTN]
	To run the program again enter		
	'R' and goto step 2. To view		
	the answer again, enter 'V'		
	and goto step 3. To end the		
	program enter 'E'.		

VARIABLE NAMES

NAME	DESCRIPTION	NAME	DESCRIPTION
B(I)	First five Bernonlli numbers ($1 \le i \le 5$)	Y	For small values calculation
X	Argument for gamma function	P	
D	In place calculation Q N!	S	Temporary storage
G	Final value of gamma	Q$	String for RVE module

NOTES AND REFERENCES

Note: The volume of the sphere of radius r is r^{n-1} times the area of the unit sphere, while the volume of the ball of radius r is r^n times the volume of the unit ball.

References:

1. Knopp, K., THEORIE AND ANWENDUNG DER UNENDLICHEN REIHEN, (Springer-Verlag, Heidelberg, 1964).

2. Henrici, P., COMPUTATIONAL ANALYSIS WITH THE HP-25 POCKET CALCULATOR, (Wiley-Interscience, New York, 1977).

3. Flanders, H., DIFFERENTIAL FORMS, (Academic Press, New York, 1963).

PROGRAM LISTING

```
 10 ! GAMMA FUNCTION
 20 ! Approximate the generalized
 30 ! factorial function by an
 40 ! augmented Stirling formula
 50 ! N.B. Gammma(x+1)=x!
 60 ! Revision 11/01/82
 70 DISP ' * * * * GAMMA FUNCTION * * *
     *'
 80 ! Store first few Bernoulli nos
 90 B(1)=1/6 @ B(2)=-1/30 @ B(3)=1/42 @
     B(4)=-1/30 @ B(5)=5/66
100 INPUT 'Arg for gamma functn?';X
110 IF FP(X)=0 AND X<=0 THEN BEEP @ DIS
    P 'Gamma has a pole here,try again!
    ' @ GOTO 100
120 IF X=1 THEN G=1 @ GOTO 310
130 IF FP(X)#0 THEN 220
140 ! Compute integral gamma
```

-Compute integral gamma function as factorial seperatly

```
150 D=1
160 FOR I=1 TO X-1
170 D=I*D
180 NEXT I
190 G=D @ GOTO 310
200 ! Use recursion to compute
210 ! Gammma for small values
220 Y=X @ P=1
230 IF Y>=8 THEN 250 ELSE P=Y*P @ Y=Y+1
     @ GOTO 230
240 ! Stirling's formula
```

-Use Stirlings formula for large values of X

```
250 G=SQR(2*PI/Y)*Y^Y*EXP(-Y) @ S=0
260 ! Correction term
270 FOR K=2 TO 10 STEP 2
280 S=S+B(K/2)/((K-1)*K*Y^(K-1))
290 NEXT K
300 S=EXP(S) @ G=G*S/P
310 DISP CHR$(6);'(';X;')=';G
320 IF NUM(KEY$)#13 THEN 320 ELSE 340
330 ! Run,view,or end
```

-Program options

```
340 DISP CHR$(210);'un again,';CHR$(214
    );'iew again or ';CHR$(197);
350 INPUT 'nd?','R'; Q$ @ Q$=UPRC$(Q$[1
    ,1])
360 ON POS('RVE',Q$)+1 GOTO 340,100,310
    ,370
370 STOP
```

PROGRAM DESCRIPTION

ERROR FUNCTION

This program computes the values of the error function

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \mathrm{EXP}(-t^2)\, dt.$$

The integral is evaluated as a special case of the incomplete gamma function $\Gamma(\alpha,x) = \int_0^x e^{-t} t^{\alpha-1} dt$. As a result, $\mathrm{erf}(x) = (1/\sqrt{\pi})\ \Gamma(1/2,x^2)$.

Algorithm: Input argument x

Disp $\mathrm{erf}(x) = (1/\sqrt{\pi}) g(\frac{1}{2},x^2)$

Stop

Def g(b,y) (incomplete gamma)

$u_o = z_o = 1/b$, $z_1 = y/(b+b^2)$, $u_1 = u_o + z_1$

α: $z_2 = yz_1^2/(z_1+yz_o)$

$u_2 = u_1 + z_2$

$g = \mathrm{EXP}(-y) y^b u_2$

If not complete, update z_o, z_1, u, and goto α

g(b,y) = g, end def.

SAMPLE PROBLEM

Compute erf(.01),erf(1),erf(3),erf(5).

SOLUTION

STEP	INSTRUCTIONS	DISPLAY	INPUT
1	Run program	*** ERROR FUNCTION ***	
2	Enter argument:	Argument for Erf?	.01 [RTN]
		Computing...	
3	View result	Erf(.01)=1.12834155558E-2	[RTN]
3	Run again	Run again, View again, or End?R	[RTN]
4	Enter argument:	Argument for Erf?	1 [RTN]
		Computing...	
5	View result	Erf(1)= .842700792944	[RTN]
6	Run again	Run again, View again, or End?R	[RTN]
7	Enter argument:	Argument for Erf?	3 [RTN]
		Computing...	
8	View result	Erf(3)= .999977909523	[RTN]
9	Run again	Run again, View again, or End?R	[RTN]
10	Enter argument:	Argument for Erf?	5 [RTN]
		Computing...	
11	View result	Erf(5)= .999999999999	[RTN]
12	End program	Run again, View again, or End?R	E [RTN]

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
1	Run program	*** ERROR FUNCTION ***	
2	Enter argument	Argument for Erf?	x [RTN]
3	View answer	Erf(x)=	[RTN]
4	Select program options:	Run again, View again, or End?R	R,V, or E [RTN]
	To run the program again,		
	enter 'R' and goto step 2.		
	To view the answer again,		
	enter 'V' and goto step 3.		
	To end the program, enter 'E'.		

VARIABLE NAMES

NAME	DESCRIPTION	NAME	DESCRIPTION
FNG(B,Y)	Incomplete gamma function	G	Final value of FNG
B,Y	Argument for FNG	F	Final value of erf
U0,U1,U2	Iterates for	Q$	Input string for RVE module
Z0,Z1	computing FNG		

NOTES AND REFERENCES

Reference: Tricomi, F., LECTURES ON THE USE OF SPECIAL FUNCTIONS BY CALCULATIONS WITH ELECTRONIC COMPUTERS, (Inst. for Fluid Dynamics, U. of Md. #47, 1966).

PROGRAM LISTING

```
 10 ! ERROR FUNCTION
 20 ! Compute the integral
 30 ! of 2/sqr(pi)*exp(-t^2)
 40 ! on the interval [0,y]
 50 ! Revision 11/01/1982
 60 DISP '   * * * ERROR FUNCTION * * *
    '
 70 ! Define incomplete gamma function
 80 DEF FNG(B,Y)
 90 U0=1/B @ Z0=1/B @ Z1=Y/(B+B^2) @ U1
    =U0+Z1
100 Z2=Y*Z1^2/(Z1+Y*Z0)
110 U2=U1+Z2
120 G=EXP(-Y)*Y^B*U2
130 IF ABS(U1-U2)<EPS THEN 140 ELSE Z0=      -Test for completion
    Z1 @ Z1=Z2 @ U1=U2 @ GOTO 100
140 FNG=G @ END DEF
150 ! Enter arg and compute Erf
160 INPUT 'Argument for Erf?';Y
170 DISP 'Computing...'
180 F=1/SQR(PI)*FNG(.5,Y*Y)
190 DISP 'Erf(';Y;')=';F
200 ! Wait for return key to continue
210 IF NUM(KEY$)#13 THEN 210 ELSE 230
220 ! Run,view,or end                        -Program options
230 DISP CHR$(210);'un again,';CHR$(214
    );'iew again or ';CHR$(197);
240 INPUT 'nd?','R'; Q$ @ Q$=UPRC$(Q$[1
    ,1])
250 ON POS('RVE',Q$)+1 GOTO 230,160,190
    ,260
260 STOP
```

PROGRAM DESCRIPTION

LEGENDRE POLYNOMIALS

This program computes the values of the Legendre polynomials $P_n(x)$ where $-1 \leq x \leq 1$ and $P_o = 1$

The Legendre polynomials satisfy the recurrence relation:

$$(n+1)P_{n+1}-(2n+1)xP_n+nP_{n-1} = 0$$

Algorithm:

```
Input n,x
Let c=x, b=1
FOR k=1 to n-1
  p=((2k+1)xc-kb)/(k+1)
  Update b,c
NEXT k
Disp Pn(x)=P
Stop
```

SAMPLE PROBLEM

Sample Problem: A) Find the value of the Legendre polynomial

Pn(x) =: P(n,x) For n=5 at x = .5

SOLUTION

STEP	INSTRUCTIONS	DISPLAY	INPUT
A_1	Run program	*** LEGENDRE POLYNOMIALS ***	
2	Enter degree n=5	Degree of polynomial?	5 [RTN]
3	Enter point x=.5	Point of evaluation?	.5 [RTN]
4	Answer	P(5,.5) =	[RTN]
		.08984375	[RTN]
5	Trig form of answer $P_n(\cos\theta)$	P(5, cos(1.0471975512) =	[RTN]
		.08984375	[RTN]
6	End program	Run again, View again, or End?R	E [RTN]

SAMPLE PROBLEM

B) A function f(x) on [-1,1] is approximated by the formula $f(x)=-.5\ P_2(x)+1.3P_3(x)$. Estimate f at x=.5

SOLUTION

STEP	INSTRUCTIONS	DISPLAY	INPUT
B_1	Run program	*** LEGENDRE POLYNOMIALS ***	
2	Enter degree n=2	Degree of polynomial?	2 [RTN]
3	Enter point x=.5	Point of evaluation?	.5 [RTN]
4	Answer	P(2,.5) =	[RTN]
		-.125	[RTN]
5	Trig form of answer	P(2, cos(1.0471975512)) =	[RTN]
		-.125	[RTN]
6	End program	Run again, View again, or End?R	E [RTN]
7	Let F_2 = -5.P	F_2 = -.5 * P	[RTN]
8	Run	*** LEGENDRE POLYNOMIALS ***	
9	Enter degree	Degree of polynomial?	3 [RTN]
10	Enter point	Point of evaluation?	.5 [RTN]
11	Answer	P(3,.5) =	[RTN]
		-.4375	[RTN]
12	Trig form of answer	P(3, cos(1.0471975512)) =	[RTN]
		-.4375	[RTN]
14	End program	Run again, View again, or End?R	E [RTN]
15	Calculate final answer	F_2+1.3*P	[RTN]
		-.50625	[RTN]

USER INSTRUCTIONS

STEP	INSTRUCTIONS	DISPLAY	INPUT
1	Run program	*** LEGENDRE POLYNOMIALS ***	
2	Enter degree of polynomial	Degree of polynomial?	n [RTN]
3	Enter point of evaluation	Point of evaluation?	x [RTN]
4	Answer	P(n,x) =	[RTN]
		P	[RTN]
5	Answer in trig form	P(n, cos θ) =	[RTN]
		P	[RTN]
6	Run again	Run again, View again, or End?	R,V, or E [RTN]
	To run again, enter 'R' and		
	goto step 2.		
	To view the answer again enter		
	'V' and goto step 4.		
	To end the program enter 'E'.		

VARIABLE NAMES

NAME	DESCRIPTION	NAME	DESCRIPTION
FNA	Answer function	C,B	Temporary storage and initial values for computing
N	Degree of polynomial		
X	Point of evaluation	P	Final approximation
		Q$	String for RVE module

NOTES AND REFERENCES

Reference: Conte, S.D., and C. deBoor, ELEMENTARY NUMERICAL ANALYSIS, (McGraw-Hill, New York, 1980).

PROGRAM LISTING

```
 10 ! LEGENDRE POLYNOMIALS
 20 ! Compute the value of the
 30 ! n'th Legendre polnomial
 40 ! at the point x
 50 ! Revision 11/01/1982
 60 DISP '* * * LEGENDRE POLYNOMIALS *
    * *'
 70 ! Wait for return,back keys
 80 DEF FNA
 90 A=NUM(KEY$) @ IF A#13 AND A#8 THEN
    90
100 FNA=A @ END DEF
110 INPUT 'Degree of polynomial?';N        -Enter degree, point of
                                            evaluation
120 INPUT 'Point of evaluation?';X
130 IF ABS(X)>1 THEN BEEP @ DISP 'Inval    -ABS(X) must be < one
    id argument!' @ GOTO 120
140 ! Test for degree 0,1 polys
150 IF N>1 THEN 190
160 IF N=0 THEN P=1 @ GOTO 260
170 P=X @ GOTO 260
180 ! Initialize
190 C=X @ B=1
200 ! Main loop
210 FOR K=1 TO N-1
220 P=1/(K+1)*(2*K+1)*X*C-1/(K+1)*K*B
230 B=C @ C=P
240 NEXT K
250 ! Answer display                        -Display answer
260 DISP 'P(';N;',';X;')=' @ IF FNA=13
    THEN DISP P ELSE 260
270 T=ACOS(X) @ IF FNA=13 THEN 280 ELSE
     270
280 DISP 'P(';N;',cos(';T;'))=' @ IF FN
    A=13 THEN DISP P
290 IF FNA=13 THEN 310 ELSE 290
300 ! RVE module                           -Program options
310 DISP CHR$(210);'un again,';CHR$(214
    );'iew again, or';CHR$(197);
320 INPUT 'nd?','R'; Q$ @ Q$=UPRC$(Q$[1
    ,1])
330 ON POS('RVE',Q$)+1 GOTO 310,110,260
    ,340
340 STOP
```

NOTES

NOTES

NOTES

MATH III

FAST FOURIER TRANSFORM
FAST FOURIER SERIES/TRIGONOMETRIC INTERPOLATION
ATTENUATED FOURIER SERIES
SPHERICAL HARMONICS
ELLIPTIC INTEGRALS
BESSEL FUNCTIONS: ASYMTOTIC EXPANSION
BESSEL FUNCTIONS: BACKWARD RECURRENCE
GAMMA FUNCTION
ERROR FUNCTION
LEGENDRE POLYNOMIALS

ALL HP-75 SOLUTIONS BOOKS ARE AVAILABLE RECORDED ON MINI-DATA CASSETTES FROM EITHER A HEWLETT-PACKARD DEALER OR THE HP USERS' LIBRARY.

HEWLETT PACKARD

00075-90063 Printed in U.S.A.

Reorder No.
00075-13005