5188 1 Program Description I

Program Title: Fourier Coefficients for Liniar composed functions

Contributor's Name: Soren Videbaek Nielsen
Address: Stenvej 3 Humlum
City: Struer Country: Denmark Postal Code: DK 7600

Program Description, Equations, Variables A function $y=F(x)$ is described by the Fourier-series: $F(x)=a_0+\sum_{n=1}^{\infty}(a_n\cos(2\pi xn/T)+b_n\sin(2\pi xn/T))$ (1)

where:

$$a_0=\frac{1}{T}\int_{x_0}^{x_m}F(t)\,dt \quad (1.1)$$

$$a_n=\frac{2}{T}\int_{x_0}^{x_m}F(t)\cos(Ct)\,dt \quad (1.2)$$

$$b_n=\frac{2}{T}\int_{x_0}^{x_m}F(t)\cos(Ct-\tfrac{1}{2}\pi)\,dt \quad (1.3)$$

$C=2\pi n/T$; $T=x_m-x_0$;

$F(x)=F_j(x)=p_j+q_j(x)$ for: $x_{j-1}\leqslant x\leqslant x_j$. $j=1,2,3,\ldots,m$ and $m\leqslant 10$

$$a_0=\frac{1}{T}\sum_{j=1}^{m}\int_{x_{j-1}}^{x_j}F_j(t)\,dt \;;\; a_0=\frac{1}{T}\sum_{J=1}^{m}\tfrac{1}{2}(x_j-x_{j-1})(y_j+y_{j-1}) \quad (2)$$

$$a_n=\frac{2}{T}\sum_{j=1}^{m}\int_{x_{j-1}}^{x_j}F_j(t)\cos(Ct)\,dt \quad (3)$$

The indefinite integral: $II=\int f(t)\cos(ct)\,dt=\int(q+rt)\cos(ct)\,dt$ has a solution: $II=rc^{-2}\cos(ct)+f(t)c^{-1}\sin(ct)$. (4)

TURN PAGE!

Operating Limits and Warnings Nothing is done to prevent input of more than 10 liniar pieces. The first liniar piece must not be parallel to the Y-axis, i.e. $x_0\neq x_1$. A discontinuety between two periods is still available, see example. If AUR is activated with no available registers (i.e. 10 pieces in the liniar function), an 'Error'-display is produced. If the execution is forced to stop (R/S), then reset the indexregister by pressing: 'f''b'.

51881

Program Description, continued:

Equation (4) is used in (3) to find a_n.

When subtracting $\frac{1}{2}\pi$ from the trigonometric argument, then b_n rather than a_n is found from equation (3).

Compaired to a similar program that uses numeric integration, this program is faster and more accurate. Especially when 'n' grows big this advantage becomes obvious. The sole disadvantage is that only functions composed from liniar pieces may be treated. But approximations of not "Liniar Composed Functions"(LCF) by LCF , may prove usefull.

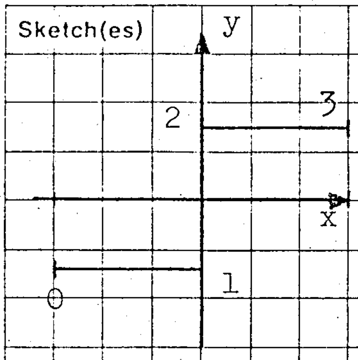
AUR = Absent User Routine

AUR was designed in order to free the user from constant observation of the calculator during execution.
With AUR the calculator can produce and save a whole series of solutions, while the user is busy elsewhere.
All though the AUR is dedicated the HP67, HP97-users may also benefit from it. They may save paper, they can deside whether to print or not, after they have seen the solutions. HP97 users may also save time if the same results are wanted on more prints with different formats(i.e. display-mode).
Some AUR's was used in the programs: 51099D and 51151D.
The present AUR uses the registers left unused after entering the LCF. I.e. registers numbered: 2m+1 through 21. That again is room enough for (10-(number of liniar pieces = m)) pairs of coefficients:(a,b).
The user decides the 'n' on the first pair to be stored. Then the program calculates and stores the following coefficients untill there is no more available registers left.
After that the calculator is prompting:'Crd'. This is the time to:
1. Stop the AUR.
2. Insert a card in order to record data.Execution will continue after that
As many (a,b)'s as wanted may be recorded simply by feeding the calculator with cards untill the number is reached. The 'n' may be monitored during execution.
There is no chance that the number on any (a,b) should be confused. The retrieving part of the AUR provides (a,b) and its number from recorded data.

How to keep track on data in the AUR.

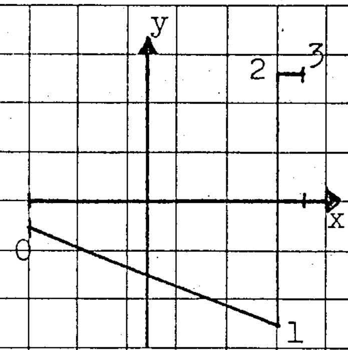
The liniar relation between the register number:'i' and the coefficient-number:'n' is: $i=2n+E$ (5)
i0 is the first and 21 is the last available register. If n0 is the initial 'n' , then $E=i0-2n0$.
Now the connection between i and n is established.
E is different for each card, but since E is stored into RE, its actual value is stored on each card as well.
The retrieving part of the AUR uses equation (5) again to calculate: $n=(i-E)/2$.
The 'i' in equation (5) is the register-number for storing the a-coefficien (i+1) is the index for storing the b-coefficients.

Sketch(es)



j	x	y
0	-π	-1
1	0	-1
2	0	1
3	π	1

Fig. 1



j	x	y
0	-9	-0.5
1	10	-2.5
2	10	2.5
3	11	2.5

Fig. 2

Sample Problem(s) In both examples we have: p= ... ,-2,-1,0,1,2,3, ...

1: y=-1 for $-\pi \leq x < 0$ and y= 1 for $0 \leq x < \pi$ and $F(x)=F(x+2p\pi)=y$

The Fourier coefficients can be expressed exact as:

$a_n=0$ for all values of n. (n is a not,negative integer)

$b_n=0$ for all even values of n.

$b_n=4/(n\pi)$ for all odd values of n.

Verify that: $a_{999}=a_{1000}=b_{1000}=0$.

Verify that: $b_{999}=4/999\pi \sim 0.0013$

2: $y \sim -1.45-x/9.5$ for $-9 \leq x < 10$. and y=2.5 for $10 \leq x < 11$

and $F(x)=F(x+20p)=y$

Generate a table of (a_n, b_n) for all values of n, from 0 through 13 .

Solution(s) 1.

1,CHS,ENTER↑,h,π,CHS,A.
1,CHS,ENTER↑,0,A.
1,ENTER↑,0,A.
1,h,π,A.

~~1. liniar piece~~
~~2. liniar piece~~
~~3. liniar piece~~

If the display does not show: 3.000, then press: h,RTN and start over again. 999,B ===> a_{999}=0.0000 . 1000,B ===> a_{1000}=0.0000

Select the b-coeff.(not a): f,a: display=0.0000

1000,B ===> b_{1000}=0.0000 . 999,B ===> b_{999}=0.0013

TURN PAGE!

Reference(s) Erwin Kreyszig,"Advanced Engineering Mathematics",1962 eddit. Section 8.2 example 1 and section 8.4 .

51881

Sample problem, solution 2.

.,5,CHS,ENTER↑,9,CHS,A.
2,.,5,CHS,ENTER↑,1,0,A. } 1. liniar piece
2,.,5,ENTER↑,1,0,A. } 2. liniar piece
2,.,5,ENTER↑,1,1,A. } 3. liniar piece

If the display does not show: 3.0000, press: h,RTN and start over again.

To start the AUR with:(a_0,b_0), enter 0 and press: C.

(Note that b_0 is defined: $b_0 \equiv 0$ (zero)).

The HP67/97 is now calculating. You may insert a card in the machine. If you do, the sound of the drive will signal you when attention is needed. Along the way the 'n' will be shown (flashing point for 5 sec.). Since F(x) is composed from 3 liniar pieces, the AUR will find (10-3)=7 pairs:(a,b) before prompting:'Crd'. And since n0 was:n0=0 we can store (a_0,b_0) through (a_6,b_6) on the first card. When the second card is needed:(a_7,b_7) through (a_{13},b_{13}) are calculated and are ready to be recorded.
No more (a,b)'s were asked for, so press: R/S, and see: 14.0000 in the display.
Press: D, to verify that the lowest coefficient number present in the machine, is 7 and the higest-, is 13.
For table output press:E, and from the resident data you will have: n(for 1 sec.),a_n(for 5 sec.) and b_n(for 5 sec.).

Now load the recorded data from the card.
Use:D,to verify the coefficient numbers and use:E, to retrieve the tablevalues.

Table:

n	a_n	b_n
0	-1.3000	0.0000
1	-0.2899	-0.7163
2	0.2756	0.4252
3	-0.2526	-0.3530
4	0.2224	0.3305
5	-0.1867	-0.3226
6	0.1475	0.3173
7	-0.1069	-0.3093
8	0.0671	0.2965
9	-0.0302	-0.2782
10	-0.0021	0.2546
11	0.0285	-0.2270
12	-0.0481	0.1966
13	0.0604	-0.1651

5 1 8 8 1 User Instructions

FOURIER COEFFICIENTS . L C F

1	a? $y_i\uparrow x_i\rightarrow$	$n\rightarrow a_n(b_n)$	AUR	LIM	TABLE	2

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1.	LOAD PROGRAM (Card or keys)				
2.	Enter one period of the function in				
	terms of the functions coordinates:	y_0	ENT	ER↑	
		x_0	A		0.0000
3.		y_j	ENT	ER↑	
		x_j	A		j.0000
4.	Repeat 3. for j=1,2,3, ... ,m and $m\leq 10$				m.0000
5.	To modify data, go to 2. after pressing		h	RTN	
6.	For AUR go to 10.				
7.	Select coefficient: for b		f	a	0.0000
	for a		f	a	1.0000
8.	Enter coefficient number	n	B		a_n or b_n
9.	For another coefficient, go to 7.				
10.	AUR: Absent User Routine. Initial 'n'	n0	C		
11.	The 'n' may be monitored during the				
	program execution				- n -
12'	Insert card optionally				
13.	The calculator prompts for card				Crd
14.	Feeding with cards automaticly resumes				
	the program from 11.				
15.	To stop the execution(when 'Crd' is on)		R/S		
16'	Option: read in data from card				
17'	Verify the smallest and the greatest, n		D		n_g , n_G
18.	For table output:(n,a_n,b_n), in that				
	order		E		Table
	The table output stops with:				0.0000
19.	for more table, go to 16.				
20.	for more calculation,go to 10.				
21.	for descrete coefficients, go to 7.				
22.	for another case, go to 2.				
	NOTE:				
	If the program was forced to stop, then				
	the indexregister must be reset:		f	b	

KEY ENTRY	KEY CODE	COMMENTS
LBL A	31 25 11	y_0↑ x_0→
X⇄I	35 24	
CLX	44	
X⇄I	35 24	
LBL 4	31 25 04	
STO(i)	33 24	
ISZ	31 34	
X⇄Y	35 52	
STO(i)	33 24	
RCL I	35 34	
INT	31 83	
1	01	
%	31 82	
+	61	
X⇄I	35 24	
2	02	
÷	81	
INT	31 83	
RTN	35 22	
LBL A	31 25 11	y_i↑ x_i→
ISZ	31 34	
GTO 4	22 04	
LBL a	32 25 11	a ?
0	00	
F1 ?	35 71 01	
e^x	32 52	
SF 1	35 51 01	
X>0	31 81	
CF 1	35 61 01	
RTN	35 22	
LBL B	31 25 12	n→a_n , n→b_n
CF 3	35 61 03	
2	02	
x	71	
π	35 73	
x	71	
GSB 1	31 22 01	
÷	81	
STO C	33 13	
0	00	
STO D	33 14	
X≠Y	32 61	
GTO 0	22 00	
F1 ?	35 71 01	
RTN	35 22	b_0=0 is returned
LBL 2	31 25 02	
SF 2	35 51 02	
GSB 0	31 22 00	
+	61	
R↑	35 54	
x	71	
RCL D	34 14	
+	61	
STO D	33 14	
ISZ	31 34	
F3 ?	35 71 03	

KEY ENTRY	KEY CODE	COMMENTS
GTO 2	22 02	
.	83	
5	05	
GTO b	22 31 12	a_0 out
LBL 0	31 25 00	
F3 ?	35 71 03	
ISZ	31 34	
RCL(i)	34 24	
DSZ	31 33	
RCL(i)	34 24	
DSZ	31 33	
RCL(i)	34 24	
X⇄Y	35 52	
DSZ	31 33	
SF 3	35 51 03	
RCL(i)	34 24	
−	51	
X=0	31 51	
GTO 0	22 00	
R↓	35 53	
F2 ?	35 71 02	
RTN	35 22	
−	51	
R↑	35 54	
÷	81	
GSB 3	31 22 03	
R↓	35 53	
ISZ	31 34	
GSB 3	31 22 03	
DSZ	31 33	
DSZ	31 33	
F3 ?	35 71 03	
GTO 0	22 00	
2	02	
LBL b	32 25 12	Output and reset index-register.
x	71	
X⇄I	35 24	
FRAC	32 83	
1	01	
0	00	
1	01	
x	71	
X⇄I	35 24	
GSB 1	31 22 01	
÷	81	
RTN	35 22	
LBL 3	31 25 03	The indefinite integral: II(x) is calculated
RCL(i)	34 24	
RCL C	34 13	
x	71	
0	00	
F1 ?	35 71 01	
$\cos^{-1}$	32 63	½(pi)
−	51	
RCL C	34 13	
1/X	35 62	

REGISTERS

0 x_0	1 y_0	2 x_1	3 y_1	4 (x_2)	5 (y_2)	6 (x_3)	7 (y_3)	8 ...	9 ...
S0 (x_5)	S1 (y_5)	S2 ...	S3 ...	S4 ...	S5 ...	S6 ...	S7 ...	S8 ...	S9 ...

A (x_{10})	B (y_{10})	C 2πn/T	D SUMMATION	E E (for AUR)	I controll

STEP	KEY ENTRY	KEY CODE	COMMENTS
	R→P	31 72	
	RCL C	34 13	
	÷	81	
	R↑	35 54	
	x	71	
	X⇄Y	35 52	
	ISZ	31 34	
120	RCL(i)	34 24	
	x	71	
	+	61	
	RCL D	34 14	
	-	51	
	STO D	33 14	$T=x_m-x_0$ is
	RTN	35 22	calculated at
	LBL 1	31 25 01	label 1.
	DSZ	31 33	
	RCL(i)	34 24	
130	ISZ	31 34	
	RCL 0	34 00	
	-	51	
	RTN	35 22	
	LBL C	31 25 13	AUR
	2	02	(Absent user
	0	00	routine)
	RCL I	35 34	
	INT	31 83	
	X>Y	32 81	
140	DSP(i)	23 24	'Error'
	1	01	
	+	61	
	R↑	35 54	
	STO B	33 12	
	2	02	
	x	71	
	-	51	
	STO E	33 15	
	RCL B	34 12	
150	GTO 3	22 03	
	LBL 5	31 25 05	
	RCL B	34 12	
	1	01	
	+	61	
	LBL 3	31 25 03	
	STO B	33 12	
	- X -	31 84	
	SPACE	35 84	
	CF 1	35 61 01	
160	GSB B	31 22 12	
	STO A	33 11	
	RCL B	34 12	
	SF 1	35 51 01	
	GSB B	31 22 12	
	RCL B	34 12	
	RCL B	34 12	
	2	02	
	x	71	

STEP	KEY ENTRY	KEY CODE	COMMENTS
	RCL E	34 15	
170	+	61	
	X⇄I	35 24	
	RCL A	34 11	
	STO(i)	33 24	
	R↑	35 54	
	ISZ	31 34	
	STO(i)	33 24	
	+	61	
	R↓	35 53	
	X⇄I	35 24	
180	2	02	
	0	00	
	X>Y	32 81	
	GTO 5	22 05	
	R↑	35 54	
	1	01	
	+	61	
	W/DATA	31 41	
	GTO C	22 13	
	LBL E	31 25 15	TABLE
190	ISZ	31 34	
	2	02	
	2	02	
	RCL I	35 34	
	INT	31 83	
	-	51	
	X=0	31 51	
	GTO b	22 31 12	
	LAST X	35 82	
	GSB 3	31 22 03	
200	RCL(i)	34 24	
	- X -	31 84	
	ISZ	31 34	
	RCL(i)	34 24	
	- X -	31 84	
	GTO E	22 15	
	LBL D	31 25 14	LIM
	RCL I	35 34	
	INT	31 83	
	1	01	
210	+	61	
	GSB 3	31 22 03	
	2	02	
	0	00	
	LBL 3	31 25 03	
	RCL E	34 15	
	-	51	
	2	02	
	÷	81	
	PAUSE	35 72	
220	RTN	35 22	

LABELS

A	B	C	D	E
A yi↑ xi→	B n→	C AUR	D LIM	E TABLE
a a ?	b out, res.	c	d	e
0 used	1 T	2 a_0	3 used	4 input
5 AUR'	6	7	8	9

FLAGS

FLAGS
0
1 not a
2 a_0
3 loop

SET STATUS

FLAGS	ON	OFF	TRIG		DISP	
0	☐	☒	DEG	☐	FIX	☒
1	☐	☒	GRAD	☐	SCI	☐
2	☐	☒	RAD	☒	ENG	☐
3	☐	☒			n 4	