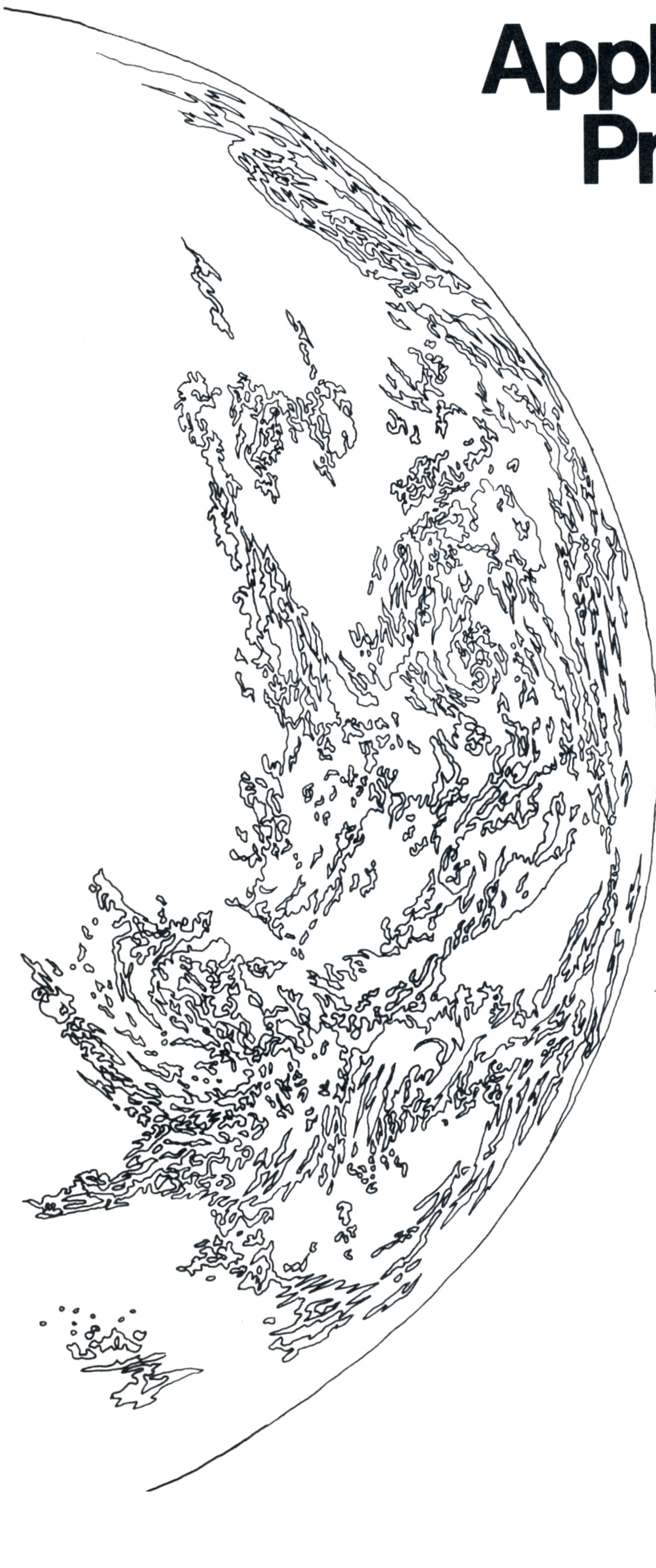
Chapter 5 —"Numerical Methods"

HEWLETT-PACKARD

HP-25 Applications Programs



00025 - 90011 Rev. B 7/75

CHAPTER 5 NUMERICAL METHODS

NEWTON'S METHOD SOLUTION TO f(x) = 0

One of the most common and frustrating problems in algebra is the solution of an equation like

$$\ln x + 3x = 10.8074,$$

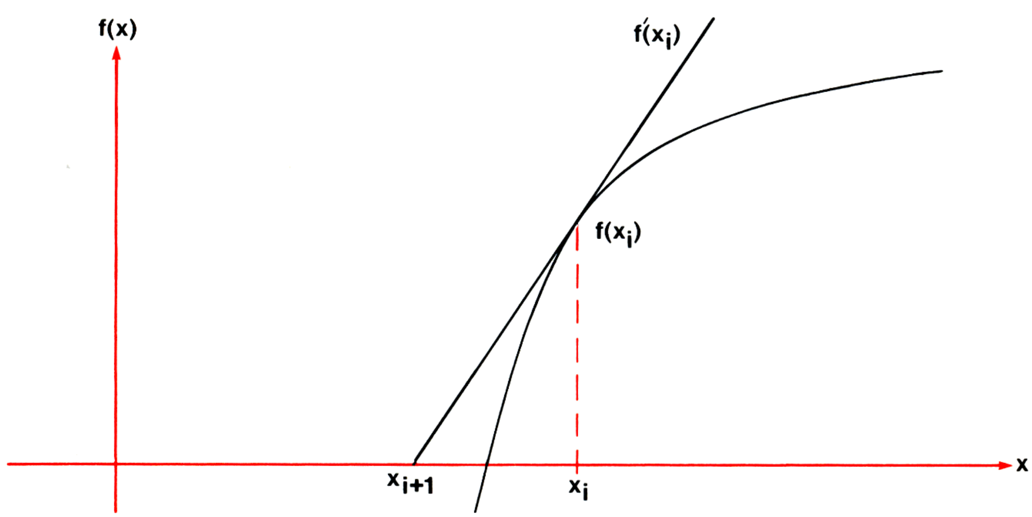
in which the x's refuse to conveniently migrate to one side of the equation and isolate themselves. That is, there is no simple algebraic solution. In this case, one of several root-finding algorithms may be employed to solve the equation $f(x) = 0$, where $f(x) = \ln x + 3x - 10.8074$. The following program uses Newton's method to find a solution for $f(x) = 0$, where $f(x)$ is specified by the user.

The user must define the function f(x) by keying into program memory the keystrokes required to find f(x), assuming x is in the X-register. Fourteen program steps are available for defining f(x); the stack registers and storage registers R_5 through R_7 are also available to the user. In addition, the user must provide the program with an initial guess, x_1, for the solution. The closer the initial guess is to the actual solution, the faster the program will converge to an answer. The program will halt when two successive approximations for x, say x_i and x_{i+1}, are within a tolerance ϵ, i.e., when $|x_{i+1} - x_i| < \epsilon$. The value for ϵ must be input by the user. In general a reasonable value for ϵ might be $10^{-6}\, x_1$.

Equations:

The basic formula used by Newton's method to generate the next approximation for the solution is

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



This program makes a numerical approximation for the derivative $f'(x)$ to give the following equation:

$$x_{i+1} = x_i - \delta_i \left[\frac{f(x_i + \delta_i)}{f(x_i)} - 1 \right]^{-1}$$

where $\delta_i = 10^{-5}\, x_i$

Notes:

1. After the routine has finished calculating, the last value of f(x) may be displayed by pressing RCL 4. If this value is not close enough to zero, the program may be run again with a smaller value for ϵ.

2. The user can watch the function converge to zero by making a slight change in the program. If the g NOP in line 07 is replaced by an f PAUSE, the program will pause during each iteration, displaying successive values of f(x) which should be converging to zero. To make this change to a program that has already been keyed in, perform the following operations:

    1. Press GTO 0 6
    2. Switch to PRGM
    3. Press f PAUSE
    4. Switch to RUN
    5. Press f PRGM

Programming Remarks

This is one of the more complex programs in the book. The main difficulty is that at each iteration both f(x) and $f(x + \delta)$ need to be calculated, but the function f is keyed in in only one place in program memory. Large computers handle this problem by the use of a subroutine. This program simulates that technique by a number stored in R_0 known as a flag. The flag is set to 0 to indicate that f(x) is to be calculated, or to 1 if $f(x + \delta)$ is to be found. After the calculation of f, a test is made on the flag. If it is 0, the program will branch to an instruction which will store f(x); if it is 1, the program will go on to calculate a derivative based on $f(x + \delta)$. All operations connected with the flag occupy a total of 9 program steps.

DISPLAY LINE	DISPLAY CODE	KEY ENTRY	X	Y	Z	T	COMMENTS
00							
01	34	CLX	0				Set flag to 0 for f(x)
02	23 00	STO 0	0				
03	24 01	RCL 1	x	0			Recall x and branch to
04	13 17	GTO 17	x	0			calculate f(x)
05	22	R↓	f(x)				Roll down to remove flag
06	23 04	STO 4	f(x)				
07	15 22	g NOP	f(x)				May Pause to see convergence
08	01	1	1	f(x)			Set flag to 1 for f(x + δ)
09	23 00	STO 0	1	f(x)			
10	24 01	RCL 1	x	1	f(x)		
11	24 01	RCL 1	x	x	1	f(x)	
12	33	EEX	1. 00	x	x	1	
13	05	5	1. 05	x	x	1	
14	71	÷	10^{-5} x	x	1	1	
15	23 03	STO 3	δ	x	1	1	
16	51	+	x + δ	1	1	1	
17							Lines 17 through 30 are
18							reserved for user
19							to define f(x)
20							
21							This section of pgm is
22							used to find f(x) and
23							f(x + δ). Flag in R_0 is
24							0 for f(x), 1 for
25							f(x + δ)
26							
27							
28							
29							
30							
31	15 71	g x = 0	f(x)/(x + δ)				Is function value = 0?
32	13 49	GTO 49	f(x)/(x + δ)				Yes, output solution
33	24 00	RCL 0	Flag	f(x)/(x + δ)			No, check flag
34	15 71	g x = 0	Flag	f(x)/(x + δ)			Flag = 0?
35	13 05	GTO 05	Flag	f(x)			Yes, have f(x)
36	22	R↓	f(x + δ)			Flag	No, flag = 1, have f(x + δ)
37	24 04	RCL 4	f(x)	f(x + δ)			
38	71	÷	R				R = f(x + δ)/f(x)
39	01	1	1	R			
40	41	–	R – 1				R–1 = [f(x+δ) – f(x)] /f(x)
41	15 22	g 1/x	$(R - 1)^{-1}$				Approximate:
42	24 03	RCL 3	δ	$(R - 1)^{-1}$			f′(x) = [f(x+δ) – f(x)] /δ
43	61	x	δ/(R – 1)				Δ = f(x)/f′(x)
44	23 41 01	STO – 1	Δ				$x_{i+1} = x_i - \Delta$
45	15 73	g ABS	|Δ|				
46	24 02	RCL 2	ε	|Δ|			
47	14 41	f x<y	ε	|Δ|			$\|x_{i+1} - x_i\| > \epsilon$?
48	13 01	GTO 01	ε	|Δ|			Yes, iterate again
49	24 01	RCL 1	x	ε	|Δ|		No, display x and halt

REGISTERS	
R_0	Flag
R_1	x
R_2	ε
R_3	δ
R_4	f(x)
R_5	
R_6	
R_7	

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in lines 1-16 of program						16 51
2	Key in function f(x)						
3	Key in a branch to line 31		GTO	31			
4	Press SST until display shows						
	line 30						
5	Key in lines 31-49 of program						
6	Switch to RUN						
7	Store initial guess for solution	x_1	STO	1			
8	Store tolerance	ϵ	STO	2			
9	Compute solution		f	PRGM	R/S		x_0
10	To change x_1 or ϵ go to appro-						
	priate step and store new value.						

Example:

An equation often solved by gear designers is

$$\tan x - x - I = 0$$

where x is an angle in radians and I is the *involute* of x. Find the angle x_0 corresponding to an involute of 0.0324.

Note:

Since a gear designer might want to calculate x for several values of I, it will be simpler to store I in R_7 for use by the function f(x).

Solution:

Example User Instructions

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in lines 1-16 of program						16 51
2	Key in steps for f(x) = tan x −						
	x − I		f	TAN			17 14 06
			f	LASTx			18 14 73
			−				19 41
			RCL	7			20 24 07
			−				21 41
3	Key in branch to 31		GTO	31			22 13 31
4	Press SST 8 times, until display						
	shows line 30						
5	Key in lines 31-49						49 24 01
6	Switch to RUN						
7	Set angular mode		g	RAD			
8	Store I	.0324	STO	7			
9	Guess x_1 = 1	1	STO	1			
10	Set tolerance $\epsilon = 10^{-6}$	10^{-6}	STO	2			
11	Compute solution x_0		f	PRGM	R/S		0.45
12	Convert the angle to degrees		180	x	g	π	
			÷				25.62
13	Display last value of f(x)		RCL	4			2.30 −09

$x_0 = 25.62°$

Last $f(x) = 2.30 \times 10^{-9}$

NUMERICAL INTEGRATION, SIMPSON'S RULE

Let x_0, x_1, ..., x_n be equally spaced points such that $x_i = x_0 + ih$ for $i = 0, 1, 2, ..., n$ at which corresponding values $f(x_0)$, $f(x_1)$, ..., $f(x_n)$ of a function f(x) are known. This function need not be known explicitly but if it is, these values can be found previously by writing the function into memory and evaluating at the various points. n must be an even positive integer.

Simpson's Rule is:

$$\int_{x_0}^{x_n} f(x)\,dx \cong \frac{h}{3}[f(x_0) + 4f(x_1) + 2f(x_2) + \ldots + 4f(x_{n-3}) + 2f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)].$$

Let the solution be indicated by I.

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
00		
01	24 00	RCL 0
02	03	3
03	71	÷
04	23 00	STO 0
05	61	x
06	23 01	STO 1
07	74	R/S
08	24 00	RCL 0
09	61	x
10	24 01	RCL 1
11	51	+
12	23 01	STO 1
13	74	R/S
14	24 00	RCL 0
15	61	x
16	04	4
17	61	x
18	24 01	RCL 1
19	51	+
20	23 01	STO 1
21	74	R/S
22	24 00	RCL 0
23	61	x
24	02	2
25	61	x
26	24 01	RCL 1
27	51	+
28	23 01	STO 1
29	13 13	GTO 13
30		
31		
32		
33		
34		
35		
36		
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		

REGISTERS
R_0 h/3
R_1 Σ
R_2
R_3
R_4
R_5
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Store increment	h	STO	0			
3	Enter first function value	$f(x_0)$	f	PRGM	R/S		Partial sum
4	Enter last function value	$f(x_n)$	R/S				Partial sum
5	Enter values i = 1, 2,..., n – 2	$f(x_i)$	R/S				Partial sum
6	Enter value i = n – 1	$f(x_{n-1})$	R/S				I

Example

Compute $\int_0^{\pi} \sin^2 x \, dx$ using Simpson's rule with $h = \pi/8$.

The following data must be found first:

i	0	1	2	3	4	5	6	7	8
x_i	0	$\pi/8$	$\pi/4$	$3\pi/8$	$\pi/2$	$5\pi/8$	$3\pi/4$	$7\pi/8$	π
$f(x_i)$	0	0.1464	0.5	0.8536	1	0.8536	0.5	0.1464	0

Solution:

$$\int_0^{\pi} \sin^2 x \, dx \cong 1.5708$$

The exact solution is $\pi/2$.

NUMERICAL SOLUTION TO DIFFERENTIAL EQUATIONS

This program may be used to solve a wide variety of first order differential equations of the form

$$y' = f(x, y)$$

with initial values x_0, y_0.

The solution is a numerical solution which calculates y_i for $x_i = x_0 + ih$, where h is an increment specified by the user and $i = 1, 2, \ldots$.

The program uses a modified Euler method (predictor – corrector):

$$\hat{y}_{i+1} = y_i + h\, f(x_i, y_i) \qquad y_{i+1} = y_i + \frac{h}{2}\left[f(x_i, y_i) + f(x_{i+1}, \hat{y}_{i+1})\right]$$

f(x, y) is keyed into memory starting at line 18. The user has 13 program steps to write f(x, y); registers R_5, R_6, and R_7 are also available. The user should assume that x and y will be in the X- and Y-registers, respectively. The routine should return with the value of f(x, y) in the X-register and should end with a GTO 31.

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
00		
01	34	CLX
02	23 04	STO 4
03	24 02	RCL 2
04	24 01	RCL 1
05	13 18	GTO 18
06	22	R↓
07	23 03	STO 3
08	24 00	RCL 0
09	61	x
10	24 02	RCL 2
11	51	+
12	24 01	RCL 1
13	24 00	RCL 0
14	51	+
15	01	1
16	23 04	STO 4
17	22	R↓
18		
19		
20		
21		
22		
23		
24		

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
25		
26		
27		
28		
29		
30		
31	24 04	RCL 4
32	15 71	g x=0
33	13 06	GTO 06
34	22	R↓
35	24 03	RCL 3
36	51	+
37	24 00	RCL 0
38	61	x
39	02	2
40	71	÷
41	24 02	RCL 2
42	51	+
43	23 02	STO 2
44	24 01	RCL 1
45	24 00	RCL 0
46	51	+
47	23 01	STO 1
48	14 74	f PAUSE
49	22	x⇄y

REGISTERS
R_0 h
R_1 x
R_2 y
R_3 y′
R_4 Flag
R_5
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in lines 1-17 of program						17 22
2	Key in function f(x, y)						
3	Key in branch to line 31		GTO	31			
4	Press SST repeatedly until dis-						
	play shows line 30						
5	Key in lines 31-49 of program						49 13 01
6	Switch to RUN						
7	Store increment	h	STO	0			
8	Store initial conditions	x_0	STO	1			
		y_0	STO	2	f	PRGM	
9	Display next x-value and cor-						
	responding y-value		R/S				(x_k)
							y_k
10	Repeat step 9 as often as desired						

Example:

Solve numerically the differential equation $y' = x\sqrt{y}$ with initial conditions $x_0 = 1, y_0 = 1$. Use a step size of $h = 0.1$.

Solution:

Key the function in as x⇄y f √x ×

x	1.0	1.1	1.2	1.3	1.4	1.5
y (by prgm)	1.0	1.1077	1.2319	1.3745	1.5372	1.7221
y (exact)	1.0	1.1078	1.2321	1.3748	1.5376	1.7227

LINEAR INTERPOLATION

If $(x_1, f(x_1))$ and $(x_2, f(x_2))$ are two points of a function $f(x)$, then the function at x_0 can be approximated by the following formula:

$$f(x_0) \cong \frac{(x_2 - x_0)\, f(x_1) + (x_0 - x_1)\, f(x_2)}{(x_2 - x_1)}$$

This is called the linear interpolation formula. Of course, x_2 cannot equal x_1.

DISPLAY		KEY ENTRY	DISPLAY		KEY ENTRY
LINE	CODE		LINE	CODE	
00			25		
01	23 04	STO 4	26		
02	24 00	RCL 0	27		
03	41	–	28		
04	24 03	RCL 3	29		
05	61	x	30		
06	24 02	RCL 2	31		
07	24 04	RCL 4	32		
08	41	–	33		
09	24 01	RCL 1	34		
10	61	x	35		
11	51	+	36		
12	24 02	RCL 2	37		
13	24 00	RCL 0	38		
14	41	–	39		
15	71	÷	40		
16	13 00	GTO 00	41		
17			42		
18			43		
19			44		
20			45		
21			46		
22			47		
23			48		
24			49		

REGISTERS
R_0 x_1
R_1 $f(x_1)$
R_2 x_2
R_3 $f(x_2)$
R_4 x_0
R_5
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Store first point	x_1	STO	0			
		$f(x_1)$	STO	1			
3	Store second point	x_2	STO	2			
		$f(x_2)$	STO	3	f	PRGM	
4	Key in x_0, find $f(x_0)$	x_0	R/S				$f(x_0)$
5	Repeat step 5 for as many x-						
	values as desired.						

Example:

Given

$$f(7.3) = 1.9879$$

$$f(7.4) = 2.0015,$$

find by linear interpolation f(7.37).

Solution:

$$f(7.37) = 1.9974$$