~~00391D~~

Program Description I

00315

Program Title CURVE FITTING USING THE CUBIC SPLINE METHOD

Contributor's Name Bruce K. Murdock

Address 6875 Sabado Tarde Rd.

City Goleta State Calif Zip Code 93017

Program Description, Equations, Variables

This program will fit a cubic spline interpolating curve through 2 to 9 equally spaced points. The cubic spline represents the shape of the curve that would be generated if a clock spring were threaded through the data points. The shape of such a curve looks natural, and generally is the same shape that one would attempt to draw by hand.

Let the ordinates y_i be given at $x_i = x_1 + (i-1)*h$, where $i = 1,2,\ldots,n$. Let $y(x)$ be the interpolating curve fitted to these points, and let y'_i and y''_i represent the first and second derivatives of $y(x)$ evaluated at $x = x_i$. $y(x)$ may be represented as pieces where the first and second derivatives of the various pieces are equal at the boundaries. The first and last segments of the interpolating curve may have their second derivatives specified by the user. The individual cubic polynomial $f_i(x)$ can be expressed using the ordinates y_i, y_{i+1}, or their first or second deravitives to represent the cubic coefficients. All forms will provide the same $y(x)$, but the second derivative form has simpler calculations. Assume that the third derivative, $y'''(x)$, is constant in each interval, h. This assumption implies that $y''(x)$ is linear in x, i.e.

$$(1) \qquad f_i''(x) = y_i'' \left\{1 - \frac{x - x_i}{h}\right\} + y_{i+1}'' \left\{\frac{x - x_i}{h}\right\}$$

Operating Limits and Warnings $2 \le n \le 9$ due to register storage location limitations

Equation (1) may be integrated twice with respect to x and the constants of integration chosen so the boundary conditions are met to the extent that $f_i(x_i) = y_i$ $(i = 1, 2, \cdots, n-1)$ and $f_{i-1}(x_i) = y_i$ $(i = 2, 3, \cdots, n)$. The results of this integration yield:

$$(2)\quad f_i(x) = y_i\left\{1 - \frac{x-x_i}{h}\right\} + y_{i+1}\left\{\frac{x-x_i}{h}\right\} - \frac{h^2}{6}y_i''\left\{1 - \frac{x-x_i}{h} - \left(1 - \frac{x-x_i}{h}\right)^3\right\} - \frac{h^2}{6}y_{i+1}''\left\{\frac{x-x_i}{h} - \left(\frac{x-x_i}{h}\right)^3\right\}$$

The above boundary conditions imply that the curve segments line up at the boundaries. In addition, the first derivatives (and second derivatives) must also match at the boundaries. The first derivative condition may be evaluated by differentiating (2) with respect to x to yield:

$$(3)\quad f_i'(x) = \frac{y_{i+1} - y_i}{h} - \frac{h}{6}\left\{2y_i'' + y_{i+1}''\right\}$$

and

$$(4)\quad f_{i-1}'(x) = \frac{y_i - y_{i-1}}{h} + \frac{h}{6}\left\{y_{i-1}'' + 2y_i''\right\}$$

Since $f_i'(x)$ must equal $f_{i-1}'(x)$ at the boundary, equations (3) and (4) may be equated to yield

$$h(y_{i-1}'') + 4h(y_i'') + h(y_{i+1}'') = 6\left\{\frac{y_{i+1} - 2y_i + y_{i-1}}{h}\right\}$$

$$i = 2, 3, \cdots, n-1$$

These equations represent $n-2$ equations in n unknowns.

Program Description I

Specifing y_1'' and y_n'' reduces the equation set to $n-2$ equations in $n-2$ unknowns. This equation set may be expressed in matrix notation

$$\begin{bmatrix} 4 & 1 & 0 & \cdots & 0 \\ 1 & 4 & 1 & 0 & \cdots 0 \\ 0 & 1 & 4 & 1 & \cdots 0 \\ \vdots & & \ddots & \ddots & \ddots \\ 0 & \cdots & 1 & 4 & 1 \\ 0 & \cdots & 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} y_2'' \\ y_3'' \\ \vdots \\ y_{n-1}'' \end{bmatrix} = \begin{bmatrix} \frac{6}{h^2}(y_1 - 2y_2 + y_3) - \frac{y_1''}{h} \\ \frac{6}{h^2}(y_2 - 2y_3 + y_4) \\ \vdots \\ \frac{6}{h^2}(y_{i-1} - 2y_i + y_{i+1}) \\ \vdots \\ \frac{6}{h^2}(y_{n-2} - 2y_{n-1} + y_n) - \frac{y_n''}{h} \end{bmatrix}$$

Because of the off diagonal zeroes, a Gauss reduction may be used to solve for the various second derivatives. Let $d_i = \frac{6}{h^2}(y_{i-1} - 2y_i + y_{i+1})$, and $y_1'' = y_n'' = 0$. (another common selection is $y_1'' = y_2''/2$ and $y_n'' = y_{n-1}''/2$). The Gauss reduced matrix becomes

$$\begin{bmatrix} 4 & 1 & 0 & \cdots & 0 \\ 0 & 4-\frac{1}{4} & 1 & 0 & \cdots 0 \\ 0 & 0, & 4-\frac{1}{4-1/4}, & 1 & \cdots 0 \\ \vdots & & & & \\ 0 & 0 & \cdots & 0, & 4-\frac{1}{4-\frac{1}{\cdots}} \end{bmatrix} \begin{bmatrix} y_2'' \\ y_3'' \\ \vdots \\ y_{n-1}'' \end{bmatrix} = \begin{bmatrix} d_2 \\ d_3 - \frac{d_2}{4} \\ d_4 - \frac{1}{4-\frac{1}{4}}\left(d_3 - \frac{d_2}{4}\right) \\ \vdots \end{bmatrix}$$

If an algorithm is defined thus

$$ {}_n4 = \cfrac{1}{4_0 - \cfrac{1}{4_1 - \cfrac{1}{4_2 - \cfrac{1}{\ddots - \cfrac{1}{4_n}}}}} $$

i.e.

$$ {}_04 = \frac{1}{4} = .25 $$

$$ {}_14 = \frac{1}{3.75} = .266\cdots $$

$$ {}_24 = \cfrac{1}{4 - \cfrac{1}{4 - \cfrac{1}{4}}} = .267857143 $$

$$ \vdots $$

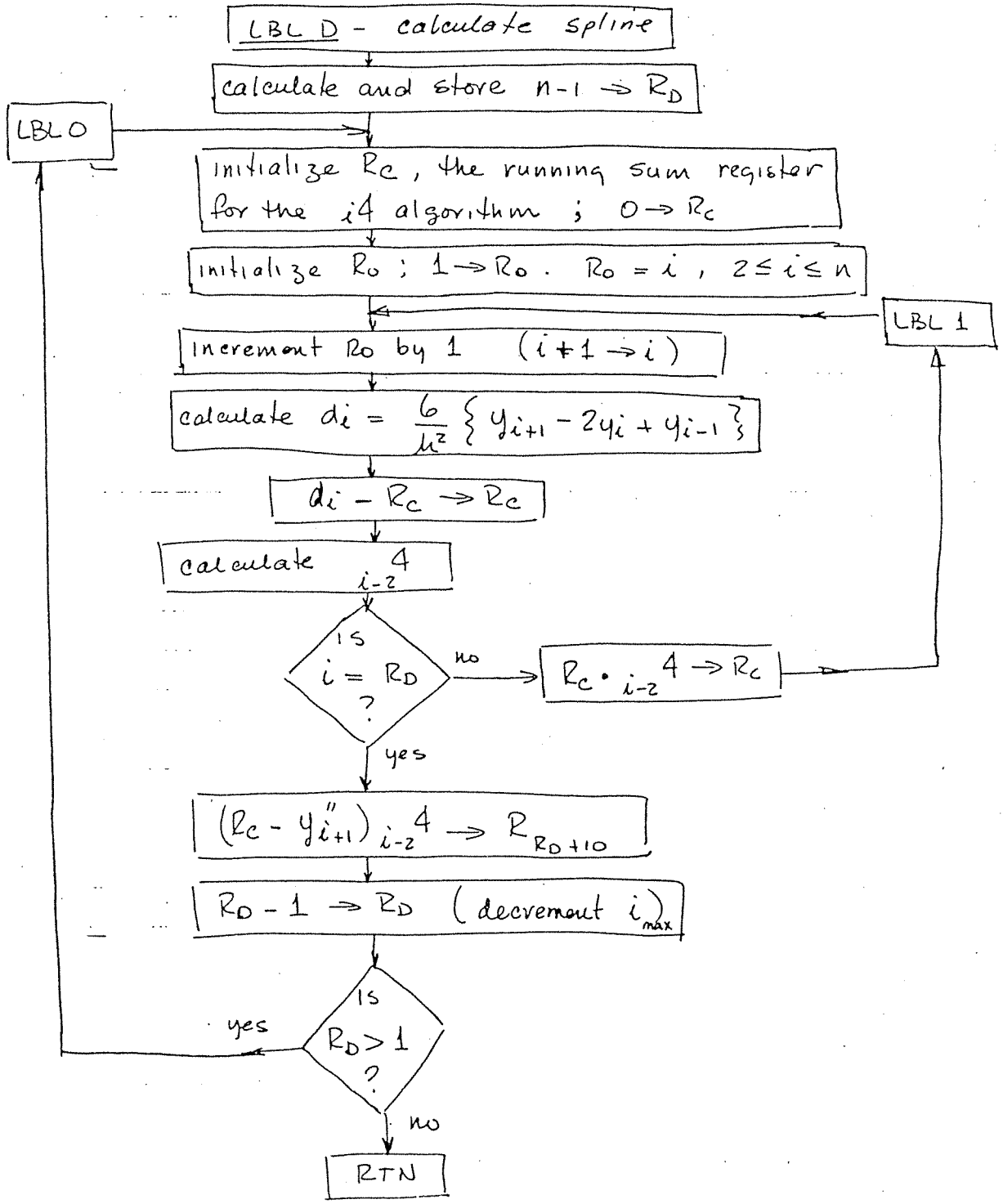
$$ {}_74 = {}_84 = \cdots = .2679491924 $$

Then the Gauss reduced matrix becomes:

$$ \begin{bmatrix} \frac{1}{{}_04} & 1 & 0 & \cdots & 0 \\ 0 & \frac{1}{{}_14} & 1 & \cdots & 0 \\ 0 & 0 & \frac{1}{{}_24} & 1 & \cdots \\ \vdots & & & & \vdots \\ 0 & \cdots & \cdots & 0 & \frac{1}{{}_{n-1}4} \end{bmatrix} \begin{bmatrix} y_2'' \\ \vdots \\ y_{n-1}'' \end{bmatrix} = \begin{bmatrix} d_2 \\ d_3 - {}_04\, d_2 \\ d_4 - {}_14(d_3 - {}_04\, d_2) \\ \vdots \end{bmatrix} $$

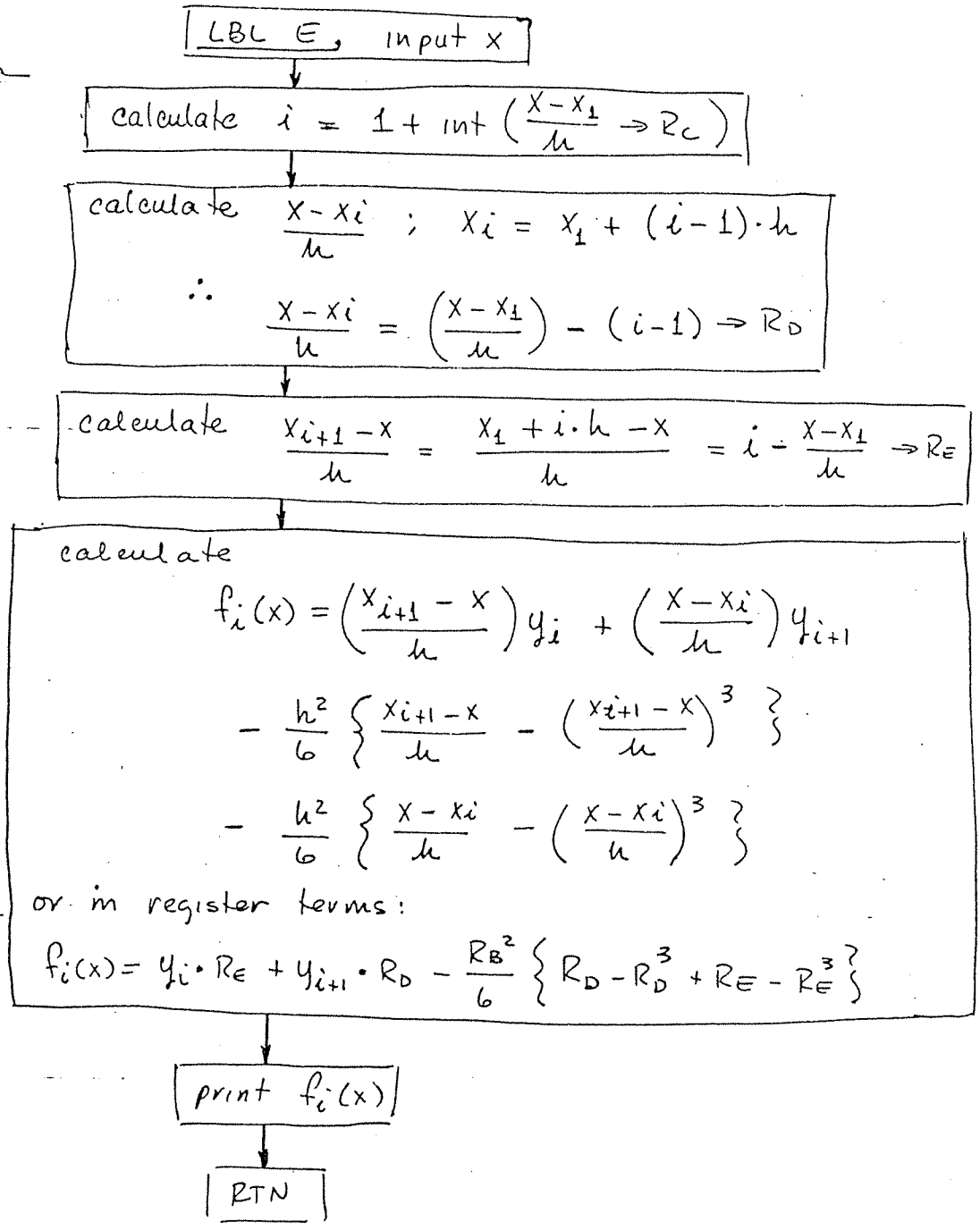
Program Description I

The solution of the gauss matrix is shown in this flowchart: (Refer to program listing for register contents)



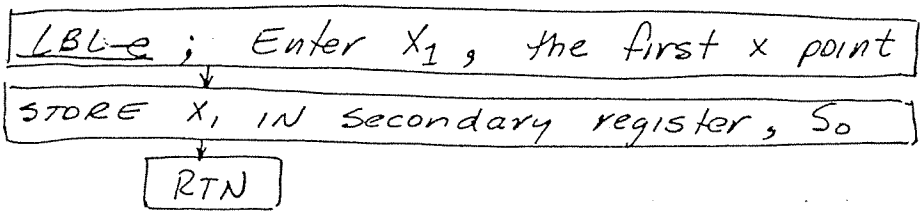
Program Description I

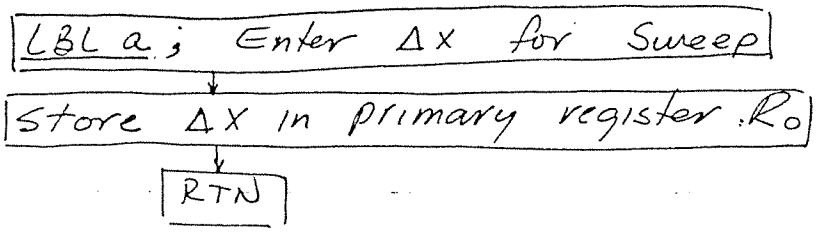
Once the second derivatives have been calculated and stored, equation (2) may be used to calculate $f_i(x)$. In flowchart form, this output section of the program becomes:

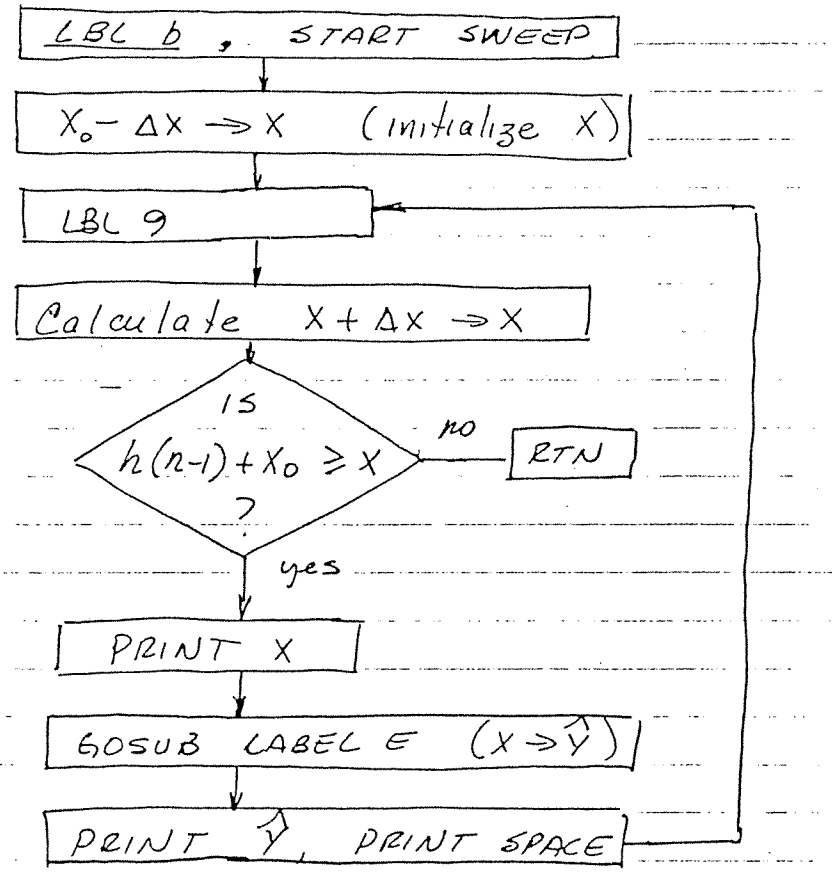


Program Description 1

In addition to the main routines shown on the previous pages, several smaller routines are required to enter the problem parameters, the data, and to output a sweep of values in x and $\hat{y}$. These routines are flowcharted here





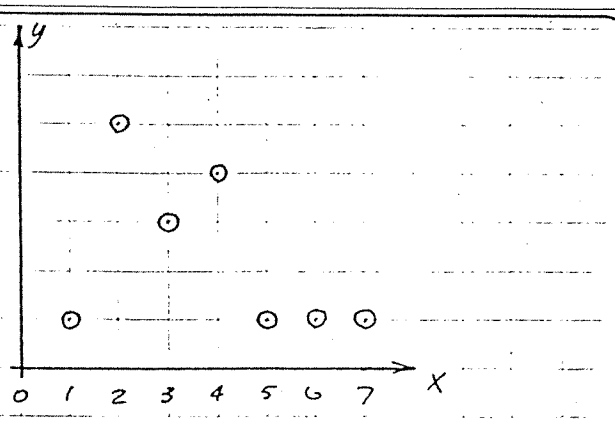


Program Description II

Sketch(es)

FIT A CUBIC SPLINE TO THESE POINTS

X	Y
1	1
2	5
3	3
4	4
5	1
6	1
7	1



Sample Problem(s) FIT A CUBIC SPLINE TO THE ABOVE DATA POINTS (7 POINTS)

Solution(s)

SEE HP-97 PRINTOUT ON NEXT PAGE

Reference(s) CARNAHAN, LUTHER, WILKES, "APPLIED NUMERICAL METHODS" JOHN WILEY © 1969

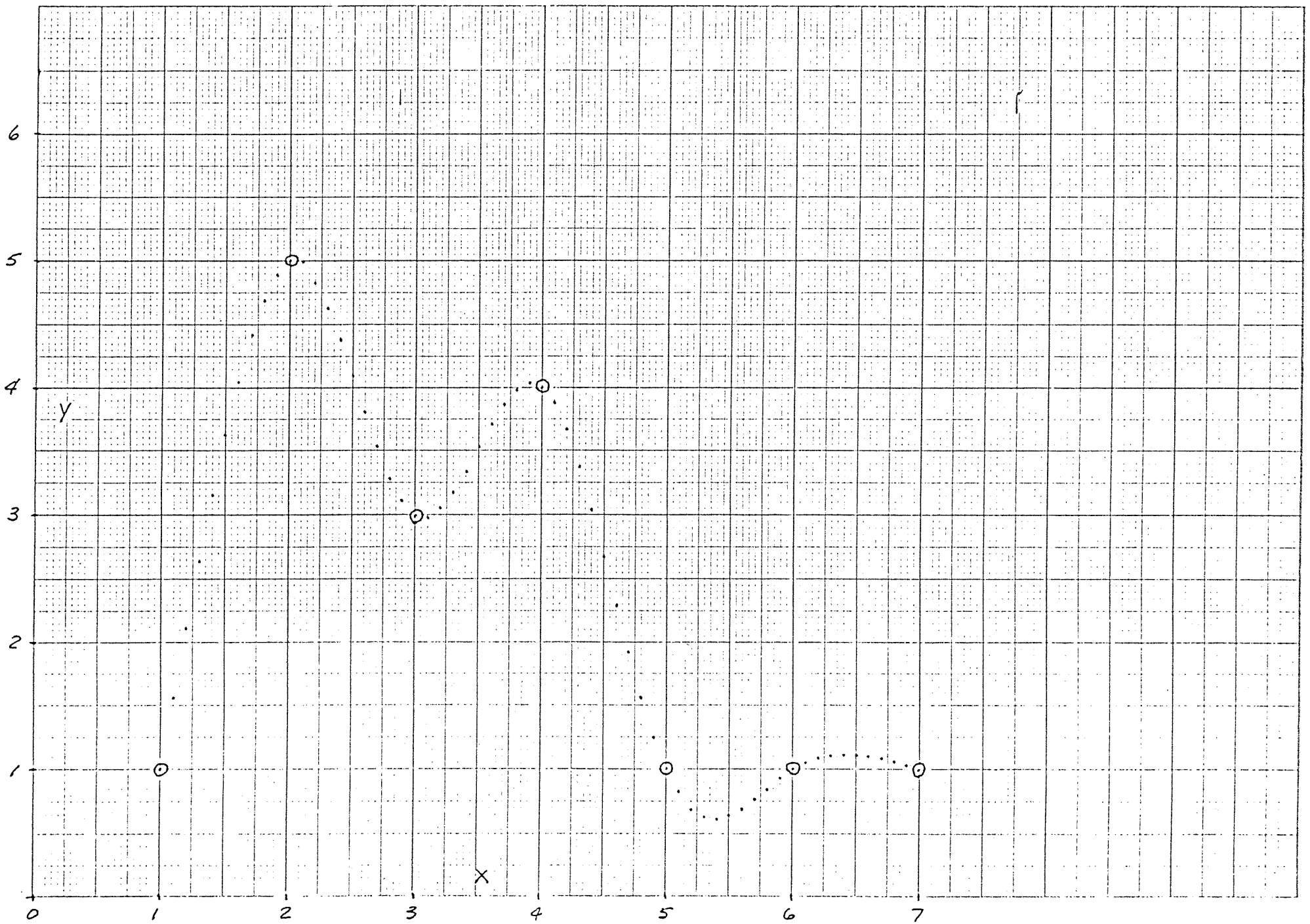
HAMMING, "NUMERICAL METHODS FOR SCIENTISTS AND ENGINEERS" MC GRAW-HILL © 1973

PROGRAM INPUT

n =	7.000	GSBA
h =	1.000	GSBB
y_1 =	1.000	GSBC
y_2 =	5.000	GSBC
y_3 =	3.000	GSBC
y_4 =	4.000	GSBC
y_5 =	1.000	GSBC
y_6 =	1.000	GSBC
y_7 =	1.000	GSBC
CALCULATE SPLINE		GSBD
X_1 =	1.000	GSBe
ΔX =	.100	GSBa
START SWEEP		GSBb

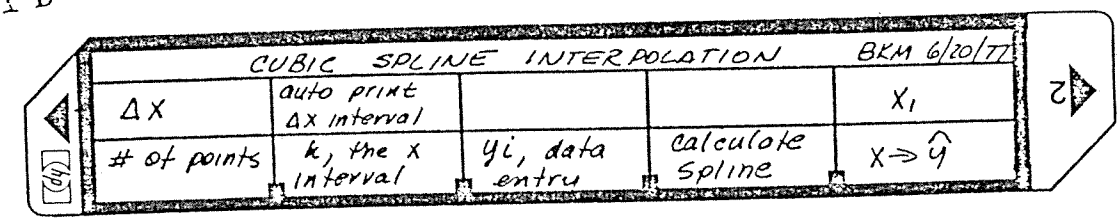
PROGRAM OUTPUT

X	Y
1.000	1.000
1.100	1.550
1.200	2.166
1.300	2.723
1.400	3.244
1.500	3.718
1.600	4.136
1.700	4.484
1.800	4.752
1.900	4.928
2.000	5.000
2.100	4.963
2.200	4.833
2.300	4.630
2.400	4.377
2.500	4.095
2.600	3.806
2.700	3.530
2.800	3.289
2.900	3.106
3.000	3.000
3.100	2.987
3.200	3.054
3.300	3.179
3.400	3.344
3.500	3.526
3.600	3.706
3.700	3.864
3.800	3.979
3.900	4.031
4.000	3.000
4.100	3.871
4.200	3.656
4.300	3.373
4.400	3.041
4.500	2.676
4.600	2.297
4.700	1.922
4.800	1.569
4.900	1.256
5.000	1.000
5.100	0.815
5.200	0.697
5.300	0.636
5.400	0.622
5.500	0.645
5.600	0.697
5.700	0.768
5.800	0.849
5.900	0.929
6.000	1.000
6.100	1.054
6.200	1.091
6.300	1.113
6.400	1.121
6.500	1.118
6.600	1.106
6.700	1.086
6.800	1.061
6.900	1.031
7.000	1.000



User Instructions

00139-1D



STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA UNITS
1	LOAD PROGRAM, BOTH SIDES			
2	ENTER # OF DATA POINTS	n	A	
3	ENTER h, THE X INTERVAL	h	B	
4	ENTER y_1	y_1	C	z
	y_2	y_2	C	3
		⋮		
		y_{n-1}	C	n
		y_n	C	
5	CALCULATE SPLINE		D	
6	ENTER FIRST X POINT	X_1	f E	
7	FOR SINGLE POINT INTERPOLATION	X	E	$\hat{y}$
8	FOR INTERPOLATION SWEEP			
	a) ENTER PRINT INTERVAL	ΔX	f A	
	b) START SWEEP		f B	X_1
				y_1
				$X_1 + \Delta X$
				y
				$X_1 + 2\Delta X$
				y
				⋮

```
001  *LBLA   ENTER # OF POINTS
002   STOA
003     1    }
004     0    }
005     +    }  Y_n'' = 0
006   STOI   }
007     0    }
008   STOi   }
009     1    }  INITIALIZE INDEX REGISTER
010   STOI   }
011    RTN
012  *LBLB   ENTER h, THE X POINT
013   STOB   SEPARATION
014    RTN
015  *LBLC   LOAD REGISTERS 1 THROUGH 9
016   STOi   WITH DATA (Y_i)
017   ISZI
018   RCLI
019    RTN
020  *LBLD   CALCULATE SPLINE
021   RCLA
022     1
023     -
024   STOD
025  *LBL0
026     0    }  INITIALIZE RUNNING SUM
027   STOC   }
028     1    }  INITIALIZE INDEX
029   STO0   }
030  *LBL1
031     1    }
032   ST+0   }
033   RCL0   }  INCREMENT & STORE INDEX
034     1    }
035     +    }
036   STOI   }
037   RCLi   }
038   DSZI   }
039   RCLi   }
040     2    }
041     x    }
042     -    }
043   DSZI   }  CALCULATE d_i
044   RCLi   }
045     +    }
046     6    }
047     x    }
048   RCLB   }
049    X²    }
050     ÷    }
051   RCLC   }
052     -    }  d_i - R_C → R_C
053   STOC   }
054   RCL0   }
055     2    }
056     -    }  i - 2 = 0 ?
057   X=0?   }
058   GTO2   }
059   STOI
060     4
061  *LBL3   CALCULATE (i-2 4)^-1
062    1/X
063    CHS
064     4
065     +
066   DSZI
067   GTO3
068   GTO4
069  *LBL2   INITIALIZE (i-2 4)^-1
070     4
071  *LBL4
072    1/X
073   STOE   STORE (i-2 4)
074   RCL0
075   RCLD
076   X=Y?
077   GTO2
078   RCLC
079   RCLE
080     x
081   STOC
082   GTO1   RETURN TO MINOR LOOP START
083  *LBL2
084   RCL0
085     1
086     1
087     +
088   STOI   STORE n+10 AS INDEX
089   RCLC
090   RCLi   RECALL Y_{i+1}''
091     -
092   RCLE
093     x
094   RCLD
095     1
096     0
097     +
098   STOI
099    R↓
100   STOi   (R_C - y_{i+1}'')(x_{i-2} 4) → R_{R_D+10}
101     1
102   RCLD   }
103     1    }  DECREMENT INDEX
104     -    }
105   STOD   }
106   X>Y?   }  IF R_D > 1 GTO MAJOR LOOP START
107   GTO0   }
108   SPC
109   RTN
```

REGISTERS

0 Δx for sweep or scratch	1 Y_1	2 Y_2	3 Y_3	4 Y_4	5 Y_5	6 Y_6	7 Y_7	8 Y_8	9 Y_9
S0 X_1, the first x point	S1 current x for sweep	S2 Y_2''	S3 Y_3''	S4 Y_4''	S5 Y_5''	S6 Y_6''	S7 Y_7''	S8 Y_8''	S9 $Y_9'' = 0$

A n, the # of points	B h, the x point separation	C scratchpad	D scratchpad	E scratchpad	I index

```
110  *LBLe   STORE FIRST X POINT
111    P≠S
112   STO0
113    P≠S
114    RTN

115  *LBLE   CALCULATE Y ESTIMATE, X → ◊
116    P≠S   } RECALL X1
117   RCL0
118    P≠S   } X - X1
119      -
120   RCLE   } CALC (X - X1)/h → Rc
121      ÷
122   STOC
123    INT   } CALC. 1 + INT((X - X1)/h) → I
124      1
125      +
126   STOI
127   RCLC   } (Xi+1 - X)/h → R0
128      -
129   STOD
130   RCLi   } yi (Xi+1 - X)/h
131      x
132   RCLC   } CALC. (X - XI)/h → RE
133   RCLI
134      1
135      -
136      -
137   STOE
138   ISZI   } (yi+1)((X - XI)/h) + yi((Xi+1 - X)/h) → Rc
139   RCLi
140      x
141      +
142   STOC
143   RCLE   } (X - Xi)/h - ((X - Xi)/h)^3
144   RCLE
145      3
146     Y^x
147      -
148   RCLI   } (yi'')·{(X - Xi)/h - ((X - Xi)/h)^3}
149      1
150      0
151      +
152   STOI
153     R↓
154   RCLi
155      x
156   RCLD   } (Xi+1 - X)/h - ((Xi+1 - X)/h)^3
157   RCLD
158      3
159     Y^x
160      -
161   DSZI   } (yi'')·{(Xi+1 - X)/h - ((Xi+1 - X)/h)^3}
162   RCLi
163      x
164   RCLI
```

```
165      1   } 0 IF 1ST INTERVAL, OTHERWISE 1
166      1   } (ALLOWS USE OF S1 FOR
167      -   } PRESENT X VALUE STORAGE
168   X≠0?   } IN SWEEP)
169   GSB7
170      x
171      +
172      6   } MULTIPLY RUNNING SUM BY
173      ÷   } h²/6 TO FORM ŷ
174   RCLB   }
175     X²   }
176      x   }
177    CHS   }
178   RCLC   }
179      +   }
180   PRTX   PRINT ◊
181    SPC
182    RTN

183  *LBL7   SUBROUTINE USED WITH
184    CLX   FIRST INTERVAL TEST
185      1
186    RTN

187  *LBLa   STORE Δx FOR SWEEP
188   STOB
189    RTN

190  *LBLb   EXECUTE SWEEP
191    SPC
192   RCL0
193    CHS
194    P≠S
195   RCL0
196      +
197   ST01
198    P≠S

199  *LBL9   SWEEP LOOP
200   RCL0
201    P≠S
202   ST+1
203   RCL1
204   RCL0
205    P≠S
206   RCLA
207      1
208      -
209   RCLB
210      x
211      +
212    X≠Y   } TEST FOR LOOP EXIT
213   X≤Y?   }
214   GTO8
215    RTN

216  *LBL8   PRINT PRESENT X
217   PRTX
218   GSBE   CALCULATE & PRINT ◊
219   GTO9
```

LABELS					FLAGS	SET STATUS		
A enter # of points	B enter h, the x interval	C Data entry	D Calculate spline	E X → ◊	0	FLAGS	TRIG	DISP
a enter Δx for sweep	b start sweep	c	d	e enter X1, the first x	1	ON OFF	DEG ■	FIX ■
0 loop destination	1 loop dest.	2 Initialize i4 & scratch	3 Calculate i4	4 gauss reduction	2	0	GRAD	SCI
5	6	7 used w/ output routine	8 Jump destination	9 loop destination	3	1	RAD	ENG
						2		n 3
						3		