UNIVERSITY OF ILLINOIS
DIGITAL COMPUTER

LIBRARY ROUTINE F 7 - 312

TITLE:	Integration of ordinary differential equations with automatic selection of elementary interval. (DOI or SADOI)
TYPE:	Closed - one program parameter -- entered with link in Q. Upon being entered, the routine finds an optimum elementary interval, executes <u>one step</u> at that interval and returns control to the link address.
ENTRIES:	Normal entry is at left of (F7) + 1 if (F7) is the location of the first word of this routine. Special "initial entry": To start an integration off, put the initial values of the independent variable and the dependent variables into their data bank locations, put the appropriate program parameter into location 0 and enter at <u>right</u> of (F7) + 0. The initial sense of integration is always such as to increase the independent variable x. Special "reversing entry": In order to effect a reversal of the sense of integration, enter at <u>right</u> of (F7) + 1 instead of at left. Subsequent normal entries will cause x to continue changing in the new sense until another reversing entry is made.
NUMBER OF WORDS:	210 plus data bank of 10n + 2 words.
TEMPORARY STORAGE:	Locations 0, 1, 2; D + 7n to D + 10n - 1 inclusive.
PRESET PARAMETERS:	Locations 3 and 4 must contain the following parameters during input of this routine:

Location	Contents	Significance
3	00F 00 DF	D specifies the location of the data bank. See description of data bank.
4	00F 00 nF	n is the order of the system of differential equations, or the number of dependent variables obeying first order equations. Note that in order to revise D and/or n the F7 program must be read in over again.

PROGRAM PARAMETER: The program parameter must be supplied only at initial entry, when it must be put into location 0 before entering. It is a compound word of the structure:

$$(\ell_o - m + 1)\, 2^{-7} + \alpha\, 2^{-19} + (e + 1)\, 2^{-27} + \ell_o\, 2^{-39}$$

Here the integer ℓ_o specifies the maximum size $h_o = 2^{-\ell_o}$ of elementary interval to be permitted; m is an integer scaling parameter for the derivatives; α is the location of the first word of the auxiliary subroutine; and e is an integer specifying the accuracy asked for. The precise meaning of e is that the e^{th} binary digit in every y_i is to be correct after integrating from x_o to $x_o + 1$.

The allowable ranges of the integer parameters are:

$$0 \leq m \leq 39$$
$$1 \leq \ell_o \leq 39 \text{ and } m \leq \ell_o$$
$$\ell_o - m \leq e \leq \ell_o - m + 39 \text{ and } e \leq 62$$

If the integer parameters violate these restrictions, then either the FF stop or a zero-shift stop will occur. Note that ℓ_o, m, α and e may be revised without reading the F7 program in over again, by making a new initial entry.

DURATION: Normally about (21n milliseconds + $2T_\alpha$) per elementary step, where T_α is the time taken by the auxiliary subroutine to calculate derivatives. About 6n milliseconds additional is required when the interval must be revised. The initial entry, but only the initial entry, requires 24 or more times the above time.

FF STOP DURING OPERATION:

Order with Sexad. Address:	From Location:	Significance:
FF 04J	(F7)+0 left	The tests indicate that an interval $< 2^{-39}$ would have to be used. Either faulty program (e.g., overflow) or too large an e for single precision operation exists.

FORM OF EQUATIONS: In order to use this routine the system of ordinary differential equations to be integrated must be cast into the form

$$\frac{dy_i}{dx} = f_i\ (x, y_1, y_2, \text{--}\ y_n); \quad i = 1, 2\ \text{--}\ n$$

with x and y_i real numbers in machine range and f_i real numbers such that $2^{-m}f_i$ are in machine range. Thus equations of higher than first order must be reformulated by introducing auxiliary dependent variables obeying first order equations; and the variables of the original scientific problem will in general have to be suitably scaled to make the problem conform to the above.

ELEMENTARY INTERVALS WHICH CAN OCCUR:

The computer's choice of elementary intervals is limited to positive and negative inverse powers of 2 (inverse ℓ_o^{th} to inverse 39^{th} power inclusive) because any other choices would much reduce the computing speed. Thus if a solution table is to be output with some constant interval, the independent variable x must be so scaled that the desired interval is represented by a power of 2 in the computer.

DATA BANK: The data bank is arranged as follows:

Location	Contents	Significance
D-2	x_o	initial value of independent variable
D-1	x	current value of independent variable
D	y_1	current rounded values of dependent variables
D+1	y_2	
- - - - - - - - -		
D+n-1	y_n	
D+n	$2^{-m}f_1$	current values of scaled derivatives
- - - - - - - - -		
D+2n-1	$2^{-m}f_n$	
D+2n	$a_1 \equiv 2^{-m}(h/2!)\, d^2y_1/dx^2$	current scaled 2nd derivatives
- - - - - - - - -		
D+3n-1	$a_n \equiv 2^{-m}(h/2!)\, d^2y_n/dx^2$	
D+3n	$b_1 \equiv 2^{-m}(h^2/3!)\, d^3y_1/dx^3$	current scaled 3rd derivatives
- - - - - - - - -		
D+4n-1	$b_n \equiv 2^{-m}(h^2/3!)\, d^3y_n/dx^3$	
D+4n	$c_1 \equiv 2^{-m}(h^3/4!)\, d^4y_1/dx^4$	current scaled 4th derivatives
- - - - - - - - -		
D+5n-1	$c_n \equiv 2^{-m}(h^3/4!)\, d^4y_n/dx^4$	
D+5n	$d_1 \equiv 2^{-m}(h^4/5!)\, d^5y_1/dx^5$	current scaled 5th derivatives
- - - - - - - - -		
D+6n-1	$d_n \equiv 2^{-m}(h^4/5!)\, d^5y_n/dx^5$	
D+6n		guard digits for y_i
- - - - - - - - -		
D+7n-1		
D+7n		"predicted" values of $2^{-m}f_i$
- - - - - - - - -		
D+8n-1		
D+8n		temporary storage
- - - - - - - - -		
D+9n-1		
D+9n		values of y_i at beginning of last step executed
- - - - - - - - -		
D+10n-1		

The second and higher derivatives of y_i alluded to in this description are to be interpreted as higher derivatives of an approximating polynomial for $y_i(x)$ and are developed automatically by the computer. The h which occurs in the description is the signed elementary increment to x which was used in the last elementary step done by the computer.

AUXILIARY SUBROUTINE:

The auxiliary subroutine reads the values of x and y_i from locations D-1 to D+n-1 and computes the scaled derivatives $2^{-m}f_i$ and places them in locations D+n to D+2n-1. This subroutine must be supplied by the user since it depends on the particular system of differential equations he wants solved. It must be designed for standard subroutine entry, with link address in the Q register, and to be entered by transferring control to the left side of location α. It must be capable of calculating the derivatives anywhere in the x-interval over which the equations are to be integrated, since the points at which the F7 routine will call for the derivatives are not known in advance. The accuracy of the auxiliary subroutine must be consistent with the accuracy demanded in the solution (a point sometimes overlooked), and this implies that $2^{-m}f_i$ must be calculated correctly to e + m binary digits or 39 binary digits, whichever is the lesser number.

INFORMATION AVAILABLE UPON EXIT FROM THIS ROUTINE:

Most of the useful information is contained in the data bank, and for such the reader is referred to the above description of the data bank.

The "guard digit" locations D+6n+i-1 are initially (automatically) set to 1/2 so that the registers D+i-1 will contain the rounded current values of the y_i. If the double precision value of y_i is desired, say to improve the accuracy of the auxiliary subroutine arithmetic, it may thus be found from the formula

$$y_i = N(D+i-1) + 2^{-39} [N(D+6n+i-1) - 1/2]$$

The signed increment to x (signed elementary interval h) last used by the routine may be found in location (F7) + 209, which is the last word of F7. It is this value of h, including sign, which obtains in the data bank description.

The recommended procedure for finding y_i at an intermediate point, i.e., a point interior to the elementary interval, is to use the available approximating polynomial to interpolate. This is accomplished as follows:

1.) Integrate until the point of interest lies beyond x-h but not beyond x.
2.) Read out h and data bank locations D-1 to D+6n-1 inclusive.
3.) Let the point of interest be $x-\Theta h$; $0 \leq \Theta < 1$. Then $y_i(x-\Theta h)$ is given with accuracy equivalent to the accuracy of $y_i(x)$, by the interpolation formula:

$$y_i(x-\Theta h) = y_i(x) - 2^m \Theta h \,[2^{-m} f_i - \Theta a_i + \Theta^2 b_i - \Theta^3 c_i + \Theta^4 d_i]$$

Roots and other similar functions of the solution are also best found by using the above interpolation procedure.

Other procedures for treating intermediate points, such as continually integrating with smaller and smaller interval in such a way as to converge on the point of interest, are awkward, time-consuming, and inaccurate compared to the above recommended procedure.

NOTES ON SCALING AND CHOICE OF m, ℓ_o, e:

The parameter m is provided so that the y_i and the f_i may be scaled independently for efficient use of the D+i-1 and D+n+i-1 registers. An obvious requirement is that the $2^{-m} f_i$, as well as the y_i and x, must all be within register capacity throughout the range of integration.

In general, if less than full register accuracy is required it is best to work near the less significant end of the registers, i.e., to scale the y_i so that they are small. The advantage of this is that m may be taken small, which in turn permits ℓ_o to be taken small if desired for increased speed. Round-off errors will not propagate more than a few places into the y_i registers because of the guard digits kept.

ℓ_o should be chosen (a) large enough to guarantee availability of the solution of the printout or display intervals desired, and (b) so that $2^{-\ell_o}$ is just larger than the largest elementary interval the computer will use, if the latter interval can be estimated. The penalty for specifying ℓ_o too small is excessive automatic starting time, which is minor; whereas the penalty for ℓ_o too large is excess time spent by the computer in developing a more accurate solution than asked for.

If some $2^{-m}f_i$ has a relatively sharp temporary fluctuation in value in a narrow range of x (e.g., an approximation to a δ-function) then one must guarantee, by taking ℓ_o large enough or otherwise, that at least one point $x = x_o + (\text{integer})\, 2^{-\ell_o}$ falls within the region of the fluctuation; otherwise, there is some risk of the computer missing the fluctuation entirely.

Discontinuities or "jumps" in $2^{-m}f_i$, if any, must not exceed 1/8 in magnitude so that the higher-derivative registers will not overflow during the process of integratin past the discontinuity.

e should be chosen after the scaling for the y_i has been chosen. If the maximum value of y_i is 2^{-d} then there will be at least about (e-d) correct significant bits of the y_i developed after integrating from x_o to $x_o + 1$. The maximum error tends generally to increase linearly with the range of integration, except that the number of correct significant digits cannot be expected to exceed the initial number of significant digits. For "well-behaved" systems of equations, namely ones such that

$$2^{-m}\left|\frac{df_i}{dx}\right| \cong 1$$

the error is generally less than 2^{-e}, typically 2^{-e-4}. For very ill-behaved equations, such that

$$2^{-m}\left|\frac{df_i}{dx}\right| \cong 2^{36}$$

the error will in general be larger, typically 2^{-e+1}.

A more precise estimate of the error in a particular result may be got by using two different e's in turn and observing which digits in y_i change. e should be changed by at least 3 or 4 units in this process.

The relationship between the value of e specified and the interval automatically chosen by the computer is as follows. The interval h chosen is the largest interval satisfying all of the three conditions:

$$|h| \leq 2^{-\ell_o}; \quad \left| h\frac{\partial f_i}{\partial y_j}\right| \leq 0.38; \; h^6|d^6 y_i/dx^6| \leq 3 \cdot 2^{-e}$$

In the last of these conditions the 6th derivative is to be interpreted as the 6th derivative of an approximating polynomial developed by the computer for $y_i(x)$. The second of the above conditions guarantees the stability of the numerical procedure and minimizes inherited errors in cases where such errors are strongly magnified. The third condition bounds the truncation error in such a way as to supply the final accuracy specified via the accuracy parameter e.

TO FREEZE THE INTERVAL:

The following modifications will transform the F7 routine into a fixed-interval routine operating with elementary interval $h = \pm h_o = \pm 2^{-\ell_o}$. The accuracy parameter e and all interval controlling tests will be ignored.

At:	Overwrite:		With:	
(F7) + 48	32 55L	F5 139L	36 ((F7) + 74) F	F5 ((F7) + 139) F
(F7) + 81	40 110L	40 207L	50	F 40 ((F7) + 207) F

Warning: The accuracy and stability of the procedure may suffer in such a fixed-interval application if the condition

$$\left| h_o \frac{\partial f_i}{\partial y_j} \right| \leq .38$$

is not satisfied.

DESCRIPTION OF ROUTINE:

A detailed description of the method used in this routine may be found in University of Illinois Coordinated Science Laboratory Report R-127, "On Numerical Integration of Ordinary Differential Equations," May, 1961, by Arnold Nordsieck; also to be published in "Mathematics of Computation," Journal of the Division of Mathematics, National Academy of Sciences - National Research Council, Washington, D. C., about October, 1961.

The method works with polynomials of 5th degree approximating to each of the y_i, and these polynomials are specified by their 0th to 5th derivatives in order to facilitate changes of interval. (Actually the derivatives beyond the first are multiplied by 2^{-m} and by a power of the current elementary interval h and divided by a factorial, as indicated in the data bank description, in order to keep them in machine range and to facilitate the arithmetic).

Equations with discontinuous derivatives are integrable by this method with accuracy comparable with the accuracy for the case of continuous derivatives provided only that the jumps in the derivatives are finite and are scaled according to the restriction stated on page 7.

The elementary step from x to (x + h) consists of applying the working equations:

$$y_i(x+h) = y_i(x)$$

$$+ 2^m h\left(2^{-m}f_i(x) + a_i(x) + b_i(x) + c_i(x) + d_i(x) + \frac{95}{288}\left[2^{-m}f_i(x+h) - 2^{-m}f_i^p\right]\right)$$

$$2^{-m}f_i^p \equiv 2^{-m}f_i(x) + 2a_i(x) + 3b_i(x) + 4c_i(x) + 5d_i(x)$$

$$a_i(x+h) = a_i(x) + 3b_i(x) + 6c_i(x) + 10d_i(x) + \frac{25}{24}\left[2^{-m}f_i(x+h) - 2^{-m}f_i^p\right]$$

$$b_i(x+h) = b_i(x) + 4c_i(x) + 10d_i(x) + \frac{35}{72}\left[\quad " \quad\right]$$

$$c_i(x+h) = c_i(x) + 5d_i(x) + \frac{5}{48}\left[\quad " \quad\right]$$

$$d_i(x+h) = d_i(x) + \frac{1}{120}\left[\quad " \quad\right]$$

Here the quantities evaluated at x are known from the previous elementary step (except in the very first step, in which case the "initial entry" must be used and an automatic starting procedure is invoked; see below); and the quantities at (x + h) are to be found. The coefficients 95/288, 25/24, 35/72, 5/48, 1/120 are specially chosen to maximize the stability of the method, to optimize the process of integration across finite discontinuities of the derivatives, and to make the truncation error $O(h^7)$ rather than $O(h^6)$ as it would be for arbitrary coefficients. The truncation error in y_i per elementary step is

$$E_t = + \frac{h^7}{70} \frac{d^7 y_i}{dx^7} + O(h^8)$$

provided the seventh derivative exists.

The solution of the working equations proceeds in three stages. Stage 1 consists of "predicting" all six quantities $y_i \ldots d_i$ at x + h, i.e., applying the working equations without the [ ] terms, using as tentative value of h the value which was accepted in the last previous step, or twice that value if the conditions for doubling h were fulfilled. Stage 2 consists of solving the complete first working equation, which is an implicit equation since the $f_i(x + h)$ depend on the $y_i(x + h)$ in virtue of the differential equations. The equation is solved iteratively by inserting the "predicted" y_i on the right, appealing to the auxiliary subroutine for $f_i(x + h)$, thus producing first improved $y_i(x + h)$, appealing again to the auxiliary subroutine and thus finally getting second improved $y_i(x + h)$. The iterative procedure is always terminated after just these two iterations, the second improved $y_i(x + h)$ and $f_i(x + h)$ being accepted as adequate approximations.

At completion of stage 2 two tests are made to determine whether h is sufficiently small. One test determines whether or not the iterative solution of the implicit equation was sufficiently convergent, and this comes to whether or not

$$\frac{95}{288}\left|h\frac{\partial f_i}{\partial y_j}\right| \leq 1/8$$

because the left side of this inequality is the convergence factor. The second test determines whether the truncation error is small enough to be consistent with the accuracy parameter e, and this comes to whether

$$|f_i(x + h) - f_i^p| \leq (3.03)\ 2^{-e}/|h|$$

If either of these tests is violated we discard the computations of stages 1 and 2, halve h and enter upon stage 1 again. If both tests are satisfied we ascertain whether they are "oversatisfied," i.e., whether a doubled h would likely satisfy them, and if so note this fact for future reference; then proceed to stage 3, which consists of "correcting" a_i, b_i, c_i, d_i by adding the [ ] terms.

The above discussion indicates the conditions for automatic reduction of interval. If the tests are not satisfied even after the interval has been reduced to 2^{-39} the FF stop is invoked. The conditions for increasing the interval, on the other hand, are as follows: If before entering stage 1 we have $|h| < 2^{-\ell_o}$ and both tests were previously "oversatisfied" and the digits of $(x-x_o)$ are such that doubling h will not cause the next point $x = x_o + (\text{integer})\, 2^{-\ell_o}$ to be missed, then h is doubled before stage 1 is entered; but after a reversal of sense of integration h may not be doubled until 4 elementary steps have elapsed. The last condition is necessary to discourage occasional erratic interval behavior after reversal.

Automatic starting is a special procedure automatically invoked before the first step away from the initial value x_o, when neither the correct h nor the quantities $a_i \ldots d_i$ are known. This feature relieves the user of the task of supplying special starting information and requires him to supply only the logically essential initial values of the y_i.

In the automatic starting mode the computer clears the $a_i \ldots d_i$ locations and then repeatedly integrates four steps forward and four steps back to x_o. Because of the high degree of stability of the method the quantities $a_i \ldots d_i$ converge rapidly to the correct values in this process, which is thus a successive approximation method of a special sort for fitting a 5th degree polynomial to $y_i(x)$ near x_o. During starting the interval is also decreased as necessary, so that when the process is complete the correct h as well as the correct $a_i \ldots d_i$ have been established. Thereafter one meaningful forward step is taken and control is returned to the link address. From the point of view of the user the first step is just like any other step except that a) he must supply the initial conditions and the program parameter and enter by the "initial entry," and b) it takes 24 or more times as long as other steps.

Complete details of the starting process and of the whole method in general are given in the paper "On Numerical Integration of Ordinary Differential Equations" referred to above.

DATE	August 9, 1961
PROGRAMMED BY	A. T. Nordsieck
APPROVED BY	J N Snyder

nj

LOCATION	ORDER		NOTES PAGE 1 F 7
	00K(F7)		(This directive to be skipped for DOI use)
0	00 F		Interlude
	00 L		relativizer
1	50 L		
	19 18F		
2	L4 4F	by 10	
	40 202L	by 9	
3	00 20F		
	L4 4F		
4	40 203L		
	80 1F		Construct constants
5	40 204L		depending on D and
	L4 203L		n and place in
6	40 F		202 through 206.
	10 19F		
7	L0 F		
	40 205L		
8	L4 203L		
	L4 3F		
9	42 2L		
	00 20F		
10	46 2L		
	L5 2L		
11	40 206L		
	50 L		resume input
12	26 999F		
	26 1L		execute interlude
	26 1N		

LOCATION	ORDER		NOTES PAGE 2 F 7
0	FF 77F		
	49 207L		← initial entry
1	49 136L		← normal entry
	K5 F	by 2	← reversing entry
2	42 1L		← save link address
	F1 207L		
3	32 16L		transfer unless initial entry
	L5 F		
4	46 45L		auxiliary location
	42 7L		reset ℓ
5	10 12F		
	46 7L		reset $\ell - m + 1$ (digest program parameter)
6	46 10L		reset $\ell_o - m + 1$
	42 9L		reset $e + 1$
7	00 F	by 5',51',69	waste order; address $= \ell - m + 1$
	19 F	by 4',51',69	address $= \ell$
8	00 1F		
	40 209L		reset $\Delta x = +2^{-\ell_o}$
9	09 1F		
	10 F	by 6'	address $= e + 1$
10	00 F	by 6	address $= \ell_o - m + 1$
	40 208L		reset test quantity t
11	L5 1023S3		
	40 1022S3		save x_o
12	50 85L		
	F5 12L		save y_i^o
13	26 27L		
	L5 201L		reset starting code
14	40 207L		
	L5 196L		clear $a_i \ldots d_i$
15	22 26L		
	50 15L		to auxiliary subroutine
16	26 45L		
	L5 136L		
17	36 32L		transfer unless reversing
	L1 209L		$-\Delta x \rightarrow \Delta x$

LOCATION	ORDER		NOTES	PAGE 3 F 7
18	40 209L			↑
	L1 200L			
19	40 200L			Change orders at
	L4 145L			145L and 167L
20	40 145L			from add to
	L5 200L			subtract or vice versa
21	L4 167L			↓
	40 167L			
22	19 3F		reset 4-step	
	40 88L		delay for h-doubling	
23	L7 110L			
	36 39L		transfer unless 24th	starting step
24	L5 208L			↑
	10 1F		halve t with	
25	40 208L		floor 2^{-39}	double h
	50 207L		$2a_i \rightarrow a_i$, $4b_i \rightarrow b_i$	↓
26	L5 197L			
	40 188L		$8c_i \rightarrow c_i$, $16d_i \rightarrow d_i$	↑
27	42 28L		link	
	46 30L		which process subroutine	
28	L1 203L			cycler to cause
	32 F	by 27,142'	transfer if i=n done	successive processing of i=1,2
29	40 2F			--n in data bank
	L4 206L			
30	26 F		to process subroutine	
	L5 2F		← from process subroutine	
31	L4 199L			
	22 28L			↓
32	L5 88L		beginning of h-doubling control	
	36 38L		transfer if delay	
33	L1 111L			
	36 39L		transfer unless tests oversatisfied	
34	L5 10L			
	L0 7L			
35	36 39L		transfer if $\ell = \ell_o$	
	L5 1022S3			

LOCATION	ORDER		NOTES PAGE 4 F 7
36	F0 1023S3		
	00 F	by 41	address = ℓ
37	36 39L		transfer if ℓ - bit of $(x-x_o)$ is 1
	26 24L		to "double h"
38	80 1F		} reduce delay by 1 step
	40 88L		
39	L5 209L		
	L4 1023S3		
40	40 1023S3		advance x
	L5 7L		} plant ℓ- dependent addresses
41	42 36L		
	46 115L		
42	46 167L		
	46 168L		
43	50 91L		} "predict y_i, f_i ... d_i
	F5 43L		
44	26 27L		
	50 44L		} to auxiliary subroutine
45	26 F	by 4	
	41 139L		clear 139L for developing $\|y_i'' - y_i'\|$ max
46	50 153L		} iterate implicit equation
	F5 46L		
47	26 27L		
	L5 112L		
48	32 55L		transfer if 2nd iteration
	F5 139L		} save convergence test information
49	10 4F		
	40 112L		
50	22 44L		back for 2nd iteration
	L5 7L		← from "$2a_i \rightarrow a_i$ etc. — double h (cont.)
51	L0 199L		$\ell - 1 \rightarrow \ell$
	40 7L		
52	L5 209L		
	80 1F		
53	40 209L		$2\Delta x \rightarrow \Delta x$
	22 39L		

LOCATION	ORDER	NOTES PAGE 5 F 7
54	L3 188L	fragment of iteration process subroutine
	10 3F	
55	26 174L	
	L7 207L	
56	32 57L	transfer if in starting mode
	L5 113L	
57	32 61L	transfer if truncation error too large
	L5 207L	
58	36 74L	transfer if {starting step 1 or not starting
	L5 112L	
59	L0 139L	
	36 76L	transfer if convergence test oversatisfied
60	40 111L	veto h-doubling
	F4 112L	
61	36 74L	transfer if convergence test satisfied
	41 133L	undo prediction
62	50 91L	
	F5 62L	
63	26 27L	
	L5 208L	2t → t with ceiling 1 (halve h)
64	80 1F	
	32 65L	
65	40 208L	
	F5 209L	1/2 Δx → Δx and FF stop if underflow (halve h)
66	10 1F	
	40 209L	
67	00 40F	
	36 L	
68	L5 7L	$\ell + 1 \rightarrow \ell$
	L4 199L	
69	40 7L	
	L5 198L	$1/2a_i \rightarrow a_i$, $1/4b_i \rightarrow b_i$ etc.
70	22 26L	
	L5 207L	
71	32 72L	transfer on 16th starting step
	L1 209L	

LOCATION	ORDER		NOTES PAGE 6 F 7
72	22 39L		adjust x
	L5 113L		
73	32 13L		transfer if truncation error too large
	22 15L		
74	50 118L		"correct" a_i, b_i, c_i, d_i
	F5 74L		
75	26 27L		
	L5 207L		beginning of starting mode control
76	80 1F		
	40 136L		to indicate "not reverse"
77	40 207L		
	36 39L		transfer if code digit is 0
78	80 1F		
	40 207L		
79	32 17L		transfer if 10
	80 1F		
80	32 81L		transfer if 110
	80 1F		
81	40 110L		to distinguish between step 16 and step 24
	40 207L		
82	50 86L		reinsert correct initial y_i^o
	F5 82L		
83	26 27L		
	L5 110L		
84	32 63L		transfer if step 16
	22 15L		end of starting mode control
85	L4 205L		← entry for "save"
	26 87L		
86	L0 205L		← entry for "reinsert"
	46 90L		"save y_i^o" or "reinsert y_i^o" process subroutine
87	40 88L		
	11 1F		clear accumulator
88	L4 F	by 22',38', 87	
	40 F		
89	L5 207L		
	32 30L		transfer if starting mode

LOCATION	ORDER		NOTES PAGE 7 F 7
90	49 F	by 86'	set guard digits = 1/2
	22 30L		
91	40 111L		
	L0 204L		
92	46 105L		
	L0 203L		
93	46 104L		
	42 146L		
94	L4 204L		
	46 107L		
95	42 107L		
	40 112L		"prediction"
96	L4 204L		process
	46 108L		subroutine
97	42 108L		(through
	40 110L		117)
98	L4 203L		
	42 109L		
99	L4 204L		
	42 105L		
100	40 113L		
	L4 203L		
101	42 114L		
	L4 203L		
102	42 104L		
	46 146L		
103	L5 133L		
	36 107L		transfer if undoing prediction
104	L5 F	by 93	
	40 F	by 102	save y_i
105	L5 F	by 92	and f_i
	40 F	by 99'	
106	L5 136L		
	32 109L		transfer unless reversing
107	L1 F	by 94'	
	40 F	by 95	$-a_i \rightarrow a_i$ and $-c_i \rightarrow c_i$

LOCATION	ORDER		NOTES	PAGE 8 F 7
108	L1 F	by 96'		
	40 F	by 97		
109	10 1F		waste order	
	L5 F	by 98'		
110	L4 F	by 81		
	40 F			"prediction
111	L4 F	by 60,91		process
	40 F	50,177'	predict first y_i, then	subroutine
112	L4 F	by 49',95',	f_i, ... then c_i	
	40 F	149		
113	L4 F	by 100,147',		
	40 F	175		
114	22 114L	by 117	alternate address 145L	
	40 F	by 101		
115	10 F	by 14'	address = $\ell - m + 1$	
	00 1F			
116	40 1F			
	L5 20L			
117	46 114L			
	22 109L			
118	40 136L			
	L0 203L			"correction"
119	40 139L			process
	L4 204L			subroutine
120	40 133L			
	L4 204L			
121	42 141L			
	L0 203L			
122	40 131L			"correction"
	L0 205L			process
123	46 141L			subroutine
	L0 203L			(through
124	46 125L			145 left)
	42 125L			
125	L1 F	by 124		
	L4 F	by 124'		

LOCATION	ORDER		NOTES	
126	40 F		$2^{-m}[f_i(x+h) - f_i^p]$ → location 0	
	L3 F			
127	36 140L		transfer if correction terms $\equiv$ 0	
	41 1F			
128	L5 F			
	32 143L		transfer if correction terms > 0	
129	L9 134L		1/2 - 16/120, approx., → accumulator	
	50 F			
130	74 192L		$[2^{-m}(f-f^p) \pm 16 \times 2^{-39}] + 1/120$ for d_i	
	50 F		for later 75 order	
131	L4 F	by 122		
	40 F		d_i corrected	
132	75 193L			
	L4 1F			
133	L4 F	by 61', 120		
	40 F		c_i corrected	
134	11 1F		waste order; left function used	
	50 F			
135	75 194L			
	L4 1F			
136	L4 F	by 1, 76', 118		
	40 F		b_i corrected	
137	50 F			
	75 195L			
138	L4 F			
	L4 1F			
139	L4 F	by 45',119, 183		
	40 F		a_i corrected	
140	L7 207L			
	32 30L		transfer if in starting mode	
141	L5 F	by 123		
	40 F	by 121	adjust guard digits	"correction"
142	L5 1L			process
	42 28L		insert exit link address	subroutine
143	22 30L			
	F5 1F			

LOCATION	ORDER		NOTES	
144	40 1F			
	LJ 134L		1/2 + 16/120 approx. → accumulator	
145	22 129L			
	L5 1F	by 20	L1 1F if integrating backwards	
146	L4 F	by 102'		
	40 F	by 93'	y_i predicted	
147	L5 151L			part of
	40 113L		blocking order → 113	"prediction"
148	L5 117L			process
	26 117L			subroutine
149	40 112L		blocking order → 112	
	22 109L			
150	40 111L		blocking order → 111	
	22 109L			
151	L5 152L		constants for blocking	
	26 149L			
152	L5 143L			
	26 150L			
153	L0 205L			
	46 166L			
154	42 178L			
	42 179L			
155	L0 203L			
	46 160L			"iteration"
156	L4 204L			process
	46 161L			subroutine
157	42 161L			(through
	L4 203L			183)
158	46 165L			
	46 171L			
159	L4 203L			
	46 178L			
160	50 F	by 155'		
	00 8F			
161	L1 F	by 156'		
	L4 F	by 157		

LOCATION	ORDER		NOTES	PAGE 11 F 7
162	40 F		$2^{-m}[f_i(x+h) - f_i^p] \to$ location 0	
	S4 F			
163	40 188L		for oscillation test	
	50 F			
164	7J 191L			
	40 F			
165	L4 F	by 158		
	40 1F			
166	51 F	by 153'		
	10 1F			"iteration"
167	00 F	by 42	address = $\ell - m + 1$	process
	L4 1L		LO 1F if going backwards	subroutine
168	10 F	by 42'	address = $\ell - m + 1$	
	00 1F			
169	40 1F		$\Delta y_i \to$ location 1	
	F1 112L			
170	36 178L		transfer if 1st iteration	
	S5 F			
171	40 F	by 158'	save new guard digits	
	L7 F			
172	80 1F			
	L2 188L			
173	36 54L		transfer if oscillation	
	L3 F			
174	LO 208L		small enough	
	36 176L		transfer if truncation error	
175	41 113L		note test failure	
	26 178L			
176	10 6F			
	L2 F			
177	36 178L		transfer if test oversatisfied	
	41 111L		veto h-doubling	
178	L5 F	by 159'		
	50 F	by 154		
179	L4 1F			
	40 F	by 154'	$y_i(x+h)$, 1st or 2nd iterate	

LOCATION	ORDER		NOTES	PAGE 12 F 7
180	S0 F			
	40 F			
181	L5 139L		max change	"interation"
	L2 F		in $y_i \to$ 139L	process
182	32 30L			subroutine
	L7 F			
183	40 139L			
	22 30L			
184	L0 204L			
	42 188L			
185	47 188L			
	L5 188L			"modify
186	L4 202L			a_i, b_i, c_i,
	42 187L			d_i" process
187	40 188L			subroutine
	L5 F	by 186'		
188	11 1F	by 26',163,	alternate functions 10, 00	
	40 F	184',185,187		
189	L5 188L			
	00 17F			
190	32 185L		transfer unless d_i done	
	22 30L			
191	00 F		95/288	
	00 3298 6111 1111J			
192	00 F		1/120	
	00 83 3333 3333J			
193	00 F		5/48	
	00 1041 6666 6667J			
194	00 F		35/72	
	00 4861 1111 1111J			
195	00 F		1/24	
	00 416 6666 6667J			
196	11 184L		for clearing a_i ... d_i	
	40 15L			
197	00 184L		for doubling h	
	40 50L			

LOCATION	ORDER		NOTES
198	10 184L		for halving h
	40 70L		
199	00 1F		increment
	00 1F		
200	00 F	by 19	for changing forward ⟷ backward
	04 F		
201	88 1552F		starting code word
	F1 240F		
	01 210K		causes SADOI to put following tape material immediately after this program
	↑		end of tape
202	00 1		constants depending on n and D, constructed by interlude
	00 n		
203	00 n		
	00 n		
204	00 2n		
	00 2n		
205	00 -3n		
	00 3n		
206	L4 D+4n		
	40 D+4n		
207		by 0',14,77 78',81'	starting code and start/run switch
208		by 10',25, 65	$-t \equiv -2^{-e} + \ell -m$
209		by 8',18, 53,66'	$\Delta x \equiv h = \pm 2^{-\ell}$
	↑		end of F7 program