PRL

TECHNICAL NOTE

TN-82-31

TN-~~82-03~~

TN-82-31

"AUTOMATIC INTEGRATOR"
A FORTRAN PROGRAM
FOR
NUMERICAL INTEGRATION

by

V.R.Chokshi & K. J. Shah

June, 1982

Physical Research Laboratory
Ahmedabad

C O N T E N T S

' AUTOMATIC INTEGRATOR '

A FORTRAN PROGRAM

FOR

NUMERICAL INTEGRATION

V.R. CHOKSHI & K.J. SHAH

Physical Research Laboratory

Ahmedabad-380009.

ABSTRACT

This note describes an adaptive interative technique based on Simpson's 1/3 modified rule for integrating a function defined analytically in an interval or tabulated at given points of an interval. A general subroutine, named "AUTOMATIC INTEGRATOR" based on the above method has been developed in FORTRAN IV and tested on an IBM 360/44 system. The program can be used to integrate functions defined for both finite and infinite limits (0 to ∞, $-\infty$ to 0, $-\infty$ to $+\infty$) A complete documentation of the program and instructions for execution are provided. Some sample results of various test functions are given for illustration. The subroutine is stored in compiled form in the library 'PDSREL' which resides on the system disk 'PLSYS' on unit ICO, at the computer center of PRL, Ahmedabad.

1. INTRODUCTION

In this note, we describe a general algorithm and a program to integrate functions that are either defined or tabulated over an interval. A brief description of the numerical algorithm is presented in section 1.1. Sections 1.2 and 1.3, describe the accuracy and convergence of the process respectively.

1.1 Numerical Algorithm:

For a given function F (x), the Simpson's Approximation is represented by,

$$\int_a^b F(x)\,dx \approx \frac{b-a}{6}\left(F(a)+4F\left(\frac{a+b}{2}\right)+F(b)\right) - \frac{h^5}{90}F^{iv}(\xi_1) \qquad \ldots\ldots\ldots\ldots(I)$$

where $F^{iv}(\xi_1)$ denotes fourth order .... derivative of function at $x = \xi_1$, $a < \xi_1 < b$, $h=b-a$.

For a given accuracy, this approximation can be modified as follows:

$$\int_a^b F(x)\,dx = F1 + F2$$

where, $F1 = \frac{h}{3}\left[F(a) + 4F(t) + F\left(\frac{a+b}{2}\right)\right]$

$$F2 = \frac{h}{3}\left[F\left(\frac{a+b}{2}\right) + 4F(x_1) + F(b)\right]$$

$$h = \frac{b-a}{4D}, \quad D=2^m, \quad m = \text{increment}$$

$$t = a + (4m+1)h$$

$$x_1 = x + 2h$$

$\ldots\ldots\ldots\ldots(II)$

For a function that is analytically defined, the functional values needed in eqn. (I) are calculated by a user's supplied function subprogram, while in case the function is tabulated, they are calculated by using Langrange's Interpolation formula given below:

$$F(x) = \sum_{i=0}^{n} F(x_i) \prod_{\substack{j=o \\ j \neq i}}^{n} \frac{(x-x_j)}{(x_j-x_i)} \qquad \text{------------(III)}$$

The program evaluates the integral by eqn. (I). Then it computes the integral over the interval (a,a+4h) by eqn. (II). The absolute difference of these two integrals is compared for the given accuracy. If this difference is greater than the desired accuracy, the interval length is halved and the above calculation is repeated. If the above difference is less than or equal to the given accuracy, summation of eqn. (II) is performed.

In case of infinite limits, the program takes into account the convergence. The convergence process is described in section 1.3.

1.2 Accuracy of the process:

The program automatically generates step sizes depending on the number of interations needed to attain the given accuracy. For a wide class of functions the program can compute the values of the integral with an accuracy of the 5th order. If an error occurs for a given accuracy and function, the intermediate computed variables go beyond the storage capacity of the computer. In such a case, the user is advised to decrease the order of the accuracy.

1.3 Convergence of the process:

The program uses the following convergence criteria "the sequence $\{f_n\}$ converges to finite limit if and only if $|F_{n+1} - f_n| \leq \epsilon$ "

In the case of infinite limits, the program generates new lower and upper limits with proper increment The absolute difference of the two consecutive integrals is compared within the tolerence for finite number of iterations . If the tolerence is attained, the current value of the integral and the corresponding limits are returned. Otherwise, the program selects the value of n for which

$$|F_{n+1} - F_n| = d_n$$

is minimum, and returns F_n as the value of the integral as well as the corresponding limits a and b.

DOCUMENTATION OF THE PROGRAM

This section contains a general description of the program.

2.1 Definition of the program:

SUBROUTING ATINT (NP, AP, AQ, NN, A, B, ACC, AINT)

2.2 Parameters of the program:

NP is used as a decision making parameter.

NP = 1 indicates that the function to be integrated is tabulated

NP = 2 indicates that the function is defined for finite limits

NP = 3 indicates that the function is defined for limits 0 to ∞

NP = 4 indicates that the function is defined for limits $-\infty$ to 0

NP = 5 indicates that the function is defined for limits $-\infty$ to $+\infty$

AP : (REAL * 4, ARRAY, IN-GOING, INTACT)
This array represents the abscissa of the tabulated function.

AQ : (REAL * 4, ARRAY, IN-GOING, INTACT)
This array represents the ordinate of the tabulated function.

NN : (INTEGER * 4, SCALAR, IN-GOING, INTACT)
This variable represents the number of points of the tabulated function. In case a function is defined analytically NN = 1.

A : (REAL * 4, SCALAR, IN-GOING, DESTROYED)
This variable represents the lower limit of the function

B : (REAL * 4, SCALAR, IN-GOING, DESTROYED)
This variable represents the upper limit of the function.

ACC : (REAL * 4, SCALAR, IN-GOING, INTACT)
This variable represents the desired accuracy.

AINT : (REAL * 4. SCALAR, OUT-COMING)
This variable represents the value of the integral for a given function.

2.3 General characteristics of the Progam:

(a) Language : FORTRAN IV

(b) Precision : Single, Double

(c) Names of the subprograms Called:

(i) Available in Program:

Subroutine 'PLAGN' which interpolates for needed points.

(ii) User's supplied subprogram:

User should supply function subprogram for a function defined as follows:

FUNCTION F(X)

(d) Total memory requirements:

2788 (HEX) bytes

2.4 Description of the output:

This program gives value of the integral 'AINT' as output for a given input. Lower limit A and upper limit B will remain the same in case of a function that is tabulated or a function that is defined for finite limits. Upper limit B will be changed for the limit $0 \rightarrow \infty$. Lower limit A will be changed for the limit $-\infty \rightarrow 0$. Lower limit A and upper limit B will be change for the limit $-\infty \rightarrow +\infty$

2.5 Special characteristics of the Program :

(a) The program uses objective dimension for arrays.

(b) In case of a function that is tabulated, user can supply an upper limit B, defined as

$$B = (NN-1) * H, \text{ if } A=o.o$$
$$= NN * H, \text{ if } A \# o.o$$

where NN = No. of points

H = constant increment in abcissa

A = Lower limit

(c) User should supply function definition through a function subprogram as follows :

FUNCTION F(X)

(d) User can set lower limit A or upper limit B exactly equal to o.o without worrying about overflow error through division by zero.

(e) If there is an exponential underflow message for given accuracy, user should decrease the order of accuracy.

(f) In case of double precision requirement, user should call the program as 'DATINT' and all real parameters should be defined in double precision.

3. HOW TO EXECUTE THIS PROGRAM :

User can access this program by the following statements :

```
JOB CARD
// SYSREL ACCESS PDSREL
// EXEC FORTCLG
    Main Program
    .
    .
    .
    .
    CALL ATINT (Parameters)
    .
    .
    .
    .
    END
    FUNCTION F (X)
    .                          user's function
    .                          subprogram
    .
    .
    END
    Subroutines (if any)
    /*
    Data card (if any)
    /*
    /&
```

4. SAMPLE PROBLEMS :

We have tested various functions for which numerically computed values can be compared to analytically available values. Results of these test cases are given in Table 1.

5. ACKNOWLEDGEMENT :

We are thankful to Dr. Y.C. Saxena and Dr.S.K.Alurkar for encouragement. Thanks are due to Dr. A. Sen for guidance and sample problems. We also wish to thank Dr. S.V. Deshmukh and Mr. S.K. Goyal who inspired us to develop the general program. Thanks are also due to Mr. Sourabhan for neat typing of the manuscript.

6. <u>BIBLIOGRAPHY</u> :

1. BRAND, L — Advanced Calculus

2. DAVIS, P.J. — Numerical Itegration

3. HILDEBRAND, F.B. — Introduction to Numerical Analysis

4. Krishnamurthy, E.V. and Sen, S.K. — Computer based Numerical Algorithm

5. Villars, D.S. — Simultaneous multiple integration, Floating point" SHARE File Number 0704 - 0240 NOSIG

TABLE - 1

Sr. No.	Function Definition	Lower Limit	Upper Limit	Analylrcal value	Numerical Computed Value Accuracy 10^{-4}	10^{-5}	10^{-6}
1.	$x^2 \cdot \sin x$	o.o	$\pi/2$	1.141593	1.141588 0.11415887 Do1	1.141587 0.11415872 Do1	*
2.	$\sin x \cdot \cos x$	o.o	$\pi/4$	0.2500	0.2500 0.2501 Doo	0.249999 0.249999 Doo	0.249999 0.249999 Doo
3.	$e^{-\cos x} \cdot \cos(\sin x)$	o.o	π	3.141593	3.141582 0.31415821 Do1	3.141588 0.31415881 Do1	*
4.	$\sin(\cos x) \cdot \cos x$	o.o	π	1.382460	1.382449 0.13824493 Do1	1.3824492 1.3824493 Do1	*
5.	$\frac{\sin^6 (x)}{x^6}$	o.o	∞	0.863938	0.863923 0.8639281 Doo	*	*
6.	$e^{-x^2/2}$	$-\infty$	$+\infty$	2.506628	2.506619 0.25066187 Do1	2.506608 0.25066082 Do1	*

* indicates that for the corresponding accuracy expontial underflow occurs.