

UNCLASSIFIED

AD 4 2 1 9 1 3

DEFENSE DOCUMENTATION CENTER

FOR

SCIENTIFIC AND TECHNICAL INFORMATION

CAMERON STATION, ALEXANDRIA, VIRGINIA



UNCLASSIFIED

Mathematics Department Report M-38

FNOL2, A FORTRAN (IBM 7090) SUBROUTINE
FOR THE SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS
WITH AUTOMATIC ADJUSTMENT OF THE INTERVAL OF INTEGRATION

Prepared by:
Jerry S. Linnekin and L. J. Belliveau*

ABSTRACT: FNOL2 is a FORTRAN subroutine used for the numerical integration of a system of up to 30 ordinary differential equations. It uses double precision arithmetic in key locations. FNOL2 has the option of automatically varying the interval of integration, h , to hold the relative or absolute truncation error within bounds fixed by the user. A FORTRAN listing of the subroutine is included in this report.

U. S. NAVAL ORDNANCE LABORATORY
WHITE OAK, MARYLAND

*Bureau of Naval Weapons (RUME-4)

17 July 1963

FNOL2, A FORTRAN (IBM 7090) SUBROUTINE FOR THE SOLUTION OF ORDINARY
DIFFERENTIAL EQUATIONS WITH AUTOMATIC ADJUSTMENT OF THE INTERVAL OF
INTEGRATION

This report describes a FORTRAN (IBM 7090) subroutine for use in the numerical integration of systems of first order, simultaneous, ordinary differential equations. The preparation of this subroutine is part of a larger project which is aimed at reducing the cost and improving the accuracy of the simulation of missile trajectories on the IBM 7090.

This work was done under NOL Task No. 417.

Adequate recognition of all the contributions to the preparation of FNOL2 is difficult if not impossible, however, our sincere thanks are offered to Mr. Charles R. Newman and Mr. Delbert L. Lehto for their cooperation and invaluable advice. Special thanks are also due to Miss Mary Lou Lyons who was responsible for much of the coding and checkout of FNOL2.

R. E. ODENING
Captain, USN
Commander


RICHARD C. ROBERTS
By direction

U. S. NAVAL ORDNANCE LABORATORY

WHITE OAK
SILVER SPRING, MARYLAND



To all holders of NOLTR 63-171
insert change; write on cover 'Change 1 inserted'
Approved by Commander, U.S. NOL

Change 1
12 November 1963
2 pages

Richard C. R. [Signature]
By direction

This publication is changed as follows:

Insert page 24A

Insert this change sheet between the cover and the title page of your copy.

```

FAP
ENTRY DPA
DPA  CLA* 1,4
      FAD* 3,4
      STO TEMP
      XCA
      UFA* 2,4
      UFA* 4,4
      FAD TEMP
      STO* 5,4
      STQ* 6,4
      TRA 7,4
TEMP HTR 0
      END

```

```

FAP
ENTRY DPS
DPS  CLA* 1,4
      FSB* 3,4
      STO TEMP
      XCA
      UFA* 2,4
      UFS* 4,4
      FAD TEMP
      STO* 5,4
      STQ* 6,4
      TRA 7,4
TEMP HTR 0
      END

```

CONTENTS

	Page
I. INTRODUCTION	1
A. Background	1
B. Description of Package	1
C. Objectives of the Report	2
II. Description of the Method	2
A. Form of Equations	2
1. Runge-Kutta Formulas	3
2. Adams-Moulton Formulas	3
B. Automatic Adjustment of Step Size	4
III. Procedure	4
A. Assignment of Variables	4
B. Main Program	5
C. Subroutine DERIV	8
D. Subroutine TERM	8
E. Subroutine OUTPUT	9
IV. Examples	10
A. Programming Examples	10
B. Results of Test Problems	17
V. Summary	19
APPENDIX A - FORTRAN Listing	21

TABLES

Table	Page
1 Accuracy comparison of sine integration	25
2 Accuracy comparison of cosine integration	26
3 Accuracy study for the integration of $dy/dx = 2 - 2y$; $y(0) = 2$	27
4 Error comparison of FNOL1, FNOL2 integration of $da/dt = A\omega\cos(\omega t)$	28

REFERENCES

1. Belliveau, L.J. and Lehto, D.I., FNOL1..a computer subroutine used for the numerical integration of ordinary differential equations with optional control of the integration interval based on truncation error, TN-5247, 19 June 1961
2. Butler, J.F., A FORTRAN II(IBM 704) subroutine for the solution of ordinary differential equations with automatic linkage, termination and output features, NAVORD 6701, 7 Oct. 1959
3. Hildebrand, F.B., Introduction to Numerical Analysis, McGraw-Hill Book Company, Inc. 1956

FNOL2, A FORTRAN (IBM7090) SUBROUTINE
FOR THE SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS
WITH AUTOMATIC ADJUSTMENT OF THE INTERVAL OF INTEGRATION

I. INTRODUCTION

A. Background

Many techniques are available for the approximate solution of ordinary differential equations by numerical methods. The subroutine described in this report exploits the 4th order Runge-Kutta and/or Adams-Moulton methods which have been quite popular at NOL.

Several years ago (1959), NOL1, reference (2), one of the predecessors of the present subroutine, was introduced. It was used with great success at NOL and the techniques it used for automatic linkage, termination and output were carried over in FNOL1, reference (1), the forerunner of FNOL2.

FNOL1 was written to overcome the following two limitations of reference (2): 1. the inability to change step size efficiently, and 2. the difficulty of modifying NOL1 for special purposes. Consequently, FNOL1 was written in FORTRAN with optional automatic adjustment of step size. Unfortunately, the double precision summation technique of reference (2) was not retained.

B. Description of the Package

1. Special Features

- a. The addition of double precision arithmetic in key locations has made the accuracy of FNOL2 comparable to NOL1, reference (2).

- b. The user may specify whether the automatic adjustment of the integration interval (step size) is to be a function of RELATIVE or ABSOLUTE TRUNCATION ERROR.*

2. Coding Information

- a. FNOL2 is a Fortran Program which requires 4004₃ cells of memory.
- b. No COMMON storage is used.

C. Objectives of the Report

- 1. To provide the reader with a complete description of the use of FNOL2. See Sections III and IV.
- 2. To provide the reader with an accuracy and timing study of FNOL1 and FNOL2 on sample problems.

II. Description of the Method

FNOL2 employs the Runge-Kutta and Adams-Moulton fourth order techniques for the integration of a system of first order ordinary differential equations, references (1) and (2).

A. Form of Equations

Let the system of equations to be solved be given in the form:

$$(1) \quad \begin{aligned} y'_i &= f_i(x, y_1, y_2, \dots, y_N) & i &= 1, 2, \dots, N \\ y_i(x_0) &= y_{i0} \end{aligned}$$

*In most cases, adjustment of the step size as a function of RELATIVE error has worked very well. However, in cases where at least one of the integrated variables is much smaller than the others and $\leq 10^{-7}$, this method has selected an unnecessarily small step size. In this event, the use of absolute truncation error adjustment is recommended.

Let y_{in} be the value of y_i at $x = x_n$ and let h be the step size in the independent variable x .

1. Runge-Kutta Formulas

The Runge-Kutta method uses the following formulas with the appropriate initial conditions to go from the n^{th} to the $n+1^{\text{st}}$ step.

$$k_{11} = h f_1 (x_n, y_{in})$$

$$k_{12} = h f_1 (x_n + \frac{1}{2} h, y_{in} + \frac{1}{2} k_{11})$$

$$k_{13} = h f_1 (x_n + \frac{1}{2} h, y_{in} + \frac{1}{2} k_{12})$$

$$k_{14} = h f_1 (x_n + h, y_{in} + k_{13})$$

$$y_{i,n+1} = y_n + \frac{1}{6} (k_{11} + 2k_{12} + 2k_{13} + k_{14}) .$$

2. Adams-Moulton Formulas

The Adams-Moulton predictor-corrector formulas for the system

(1) are:

$$(2) \quad y_{i,n+1}^{(p)} = y_{in} + \frac{h}{24} (55 f_{in} - 59 f_{i,n-1} + 37 f_{i,n-2} - 9 f_{i,n-3})$$

$$(3) \quad y_{i,n+1}^{(c)} = y_{in} + \frac{h}{24} (9 f_{i,n+1}^{(p)} + 19 f_{in} - 5 f_{i,n-1} + f_{i,n-2}) .$$

The corrector formula, (3) is applied only once so that only two derivative evaluations are needed for each Adams-Moulton integration step. The starting values needed in (2) are obtained using the Runge-Kutta method (3 steps).

Both the Runge-Kutta and the Adams-Moulton methods incorporate round-off control features. This is accomplished by keeping x and the y_{in}

in double precision and forming the sums $x + \Delta x$ and $y_{in} + \Delta y_{in}$ in double precision. All FORTRAN statements are evaluated in single precision.

B. Automatic Adjustment of Step Size

FNOL2 has the option of automatically varying the step size, h , to hold the Relative or Truncation error within bounds fixed by the user (Adams-Moulton Mode only).

The Truncation error, T_{E_1} , is approximated by:

$$T_{E_1} \approx \frac{y_1^{(p)} - y_1^{(c)}}{14} \quad \text{and}$$

the Relative error is

$$R_{E_1} = \frac{T_{E_1}}{y_1^{(c)}}$$

III. Procedure

A. Assignment of Variables

In order to integrate $N(1 \leq N \leq 30)$ simultaneous first order ordinary differential equations each of the form:

$$(4) \quad y'_i = \frac{dy_i}{dx} = f_i(x, y_1, y_2, \dots, y_N) \quad i = 1, 2, \dots, N$$

beginning at some initial point and ending when

$$(5) \quad F(x, y_1, y_2, \dots, y_N, y'_1, y'_2, \dots, y'_N) = 0,$$

one proceeds as follows:

For notation purposes, assign the dependent variables, y_i , as elements of a one-dimensional array, Y , and the derivatives, y'_i ,

as elements of a one-dimensional array, D. Let

$$\begin{aligned}
 & Y(1) = y_1 \\
 & Y(2) = y_2 \\
 & \quad \vdots \\
 & Y(N) = y_N \\
 & D(1) = y'_1 \\
 & \quad \vdots \\
 & D(N) = y'_N
 \end{aligned}
 \tag{6}$$

The names of these two arrays must conform to the usual FORTRAN notation, but are otherwise arbitrary. In the MAIN program the DIMENSION of the Y and D arrays must be at least as large as $(L + N + 3)$.^{*} The arrays containing the dependent variables y_1 and its derivatives y'_1 should not be placed in COMMON storage. One or more additional arrays should be placed in COMMON to provide linkage for tables, controls, extraneous calculations, etc.

B. MAIN Program

The MAIN Program must assign initial values to the dependent variables (y_1) and to the independent variable (x). Somewhere in the MAIN program an F card of the following form must be inserted:

F	DERIV, TERM, OUT, TERM 1, etc.
cc 1	78...

^{*}L and N are control variables and are defined on p.6.

The CALLING sequence is:

CALL FNOL2 (J,N,G,L,M,NE,X,Y,D,DERIV,TERM,OUT)

The names on the F card may be any name that the programmer desires so long as they correspond to the CALLING sequence.

The controls in the CALLING sequence for FNOL2 are as follows:

- J = 1 Use Runge-Kutta method of integrating throughout (truncation error is not calculated).
- = 2 Use Runge-Kutta method of integrating for first 4 intervals, Adams-Moulton (predictor corrector) until termination condition in Subroutine TERM is reached -- then terminate with 4 iterations using Runge-Kutta again. Truncation error is automatically calculated in Adams-Moulton procedures.
- = 3 Use Runge-Kutta method calculating the truncation error by repeating every second calculation with a doubled interval of integration - reference (1). The step size is not adjusted.
- N = Number of simultaneous differential equations to be solved. They are set up in the DERIV subroutine and may be any number up to 30.
- G = First interval of integration.
- L = Number of D's greater than N to be printed out when the enclosed OUTPUT routine is followed (see p. 24). If the user's output routine is used, L may be ignored.
- M = Print frequency -- the number of integration cycles between printouts. If M = 0, then printing is determined by values assigned to Y(N + 1) and Y(N + 2), where Y(N + 1) is set equal to some running variable like X, D(1), Y(1) etc., and Y(N + 2) is a constant interval in Y(N + 1) between printing cycles. This feature is particularly important if the variable interval of integration option is to be used since printout frequency can become somewhat meaningless when the interval of integration changes from step to step.
- NE = Control for the Interval of integration. The Interval is increased, let alone, or decreased depending on whether the estimated relative or truncation error is less than 10^{-NE-3} , between 10^{-NE-3} and 10^{-NE} , or greater than 10^{-NE} . Any decrease in the interval of integration which

places the relative or truncation error below the band is always followed by an advance of one integration step. Using this interval before a subsequent increase allows the subroutine to proceed through difficult regions where endless hunting would otherwise result.

NE = 0 No adjustment of G, the original interval of integration takes place, and $Y(N+3)$ is ignored. NE must be zero when $J = 1$.

$Y(N+3)^* = -1$. And IF NE $\neq 0$ automatic adjustment of the interval of integration is a function of truncation error.

= 0. And IF NE $\neq 0$ automatic adjustment of the interval of integration is a function of the Relative truncation error.

X = Independent variable. The initial value must be specified in the main program before calling FNOL2.

Y = The name given to the array which contains the Dependent variables. Initial values must be specified in the main program before calling FNOL2.

D = The name given to the array which contains the derivatives. The first N D's are the computed values of the differential equations evaluated at current values of X and Y, and obtained from subroutine DERIV. The D's after N can be used to print out auxiliary quantities or compute secondary quantities of interest, except in subroutine DERIV.

DERIV The name of the subroutine in which the differential equations are to be found. In multiple phase problems this name can be varied as DERIV1, DERIV2, etc., and so called from the main routine.

TERM The name of the subroutine in which the termination condition is found. Any variable or combination of variables in X, Y, or D may be used as the termination condition. Several termination conditions can be written and named TERM1, TERM2, etc., to correspond to different phases of a given problem.

OUT A separate standard output routine is provided. Any output routine desired can be written in standard FORTRAN procedure however.

The names of the 3 subroutines above are completely arbitrary. The only restriction is that they must correspond to the F card and the CALLING sequence for FNOL2.

*NOTE: $Y(N+3)$ MUST be defined in the executive routine.

C. Subroutine DERIV

Construct a subroutine which evaluates the derivatives, y'_i , and places the results in the D-array. The name of this subroutine is arbitrary. The first statement of this subroutine must be of the form:

```
SUBROUTINE DERIV (X, Y, D)
```

No recursive formulas should be evaluated in DERIV since it may be entered a variable number of times at each step, depending on the mode of integration. No D greater than N should be used in SUBROUTINE DERIV.

D. Subroutine TERM

Construct a subroutine which evaluates the condition and/or conditions of termination. The subroutine must be of the form

```
SUBROUTINE TERM (X, Y, D, F)
```

where F is the variable which determines termination. F may be equal to $f(X, Y, D)$. Termination occurs when $F = 0$ or undergoes a change of sign. Interpolation and/or iteration is automatic.

Before proceeding to a discussion of the output routine we digress here for a moment to give an example of the above two subroutines. Suppose that it is desired to integrate the system

$$\frac{dy_1}{dx} = y_2$$

$$\frac{dy_2}{dx} = -y_1$$

beginning at some initial point (which we ignore for the moment) and terminating when $y'_1 = 0$. The derivative and termination subroutines

for the problem are:

```
SUBROUTINE DERIV (X, Y, D)
  DIMENSION Y(1), D(1)
  D(1) = Y(2)
  D(2) = -Y(1)
  RETURN
```

```
SUBROUTINE TERM (X, Y, D, F)
  DIMENSION Y(1), D(1)
  F = D(1)
  RETURN
```

Note that, since Y and D are arrays, they must be included in a DIMENSION statement in each subroutine.

E. Subroutine OUTPUT

A standard OUTPUT subroutine is included with the binary deck of FNOL2. The FORTRAN listing is in Appendix A. The format is as follows:

X	D(1)	Y(1)	truncation error	H(interval of integration)	
	D(2)	Y(2)	"		
	D(3)	D(4)	D(5)	D(6)	D(7)
	D(8)	D(9)	.	.	D(L+N)

Any quantity can be labeled by a D beyond the last D used as a differential equation (up to D(L+N)) and it will be printed out automatically in the above format. This OUTPUT routine may be replaced by any FORTRAN output routine with the following CALLING sequence and DIMENSION statement:

```
SUBROUTINE OUT (X, Y, D, ERROR, N, L, H)
  DIMENSION Y(1), D(1), ERROR (1)
```

IV. Examples

A. Programming Examples

Example No. 1

Use the Adams-Moulton method with $\Delta x = .01$ to integrate $\frac{d^2y}{dx^2} = -y$ and terminate when $x = \pi/2$.

The second order equation must be reduced to two first order equations of the following form:

$$\frac{dy_1}{dx} = y_2 \quad (1)$$

$$\frac{dy_2}{dx} = -y_1 \quad (2)$$

The initial conditions are:

$$y_1(0) = 0$$

$$y_2(0) = 1$$

$$0 \leq x \leq \pi/2$$

Using the enclosed Output routine (Appendix A) write the following quantities on the output tape at every fifth step of integration:

$$x, y_1, y_2, y_1', y_2', \epsilon(y_1), \epsilon(y_2) .$$

Solution:

For FORTRAN notation we make the following correspondence:

$$Y(1) = y_1, Y(2) = y_2, D(1) = y_1', D(2) = y_2', X = x.$$

The FORTRAN program is:

Example 1:

```

      FOR
C     MAIN PROGRAM
      DIMENSION Y(5),D(5)
      X=0.0
      Y(1)=0.0
      Y(2)=1.0
F     DERIV,TERM,OUTPUT
      CALL FNOL2(2,2,.01,0,5,0,X,Y,D,DERIV,TERM,OUTPUT)
      STOP*
      END

```

```

      FOR
      SUBROUTINE DERIV(X,Y,D)
      DIMENSION Y(1),D(1)
      D(1)=Y(2)
      D(2)=-Y(1)
      RETURN
      END

```

```

      FOR
      SUBROUTINE TERM(X,Y,D,F)
      DIMENSION Y(1),D(1)
      F=X-1.5707963
      RETURN
      END

```

*STOP should be replaced with any legitimate System exit.

Example No. 2

Integrate the equation given in Example 1 with the Runge-Kutta method. Read in all controls as data. Integrate from $0 \leq x \leq \frac{\pi}{2}$ with $h = .01 = \text{constant}$. Terminate at $\pi/2$ and use the Adams-Moulton technique with a variable step size from $\pi/2$ to π to hold the RELATIVE Truncation error $10^{-8} \leq RE_1 \leq 10^{-5}$ (i.e., $NE = 5$). In addition, compute $y_1^2 + y_2^2$ and print out the results at approximately equal increments of 0.1 in y_1 .

The FORTRAN program is:

```

      FOR
C     MAIN PROGRAM
      DIMENSION Y(7),D(7)
      1 READ 2,J,N,G,L,M,NE
      2 FORMAT(2I3,F3.3,3I3)
      3 IF(J) 4,4,5
      4 STOP
      5 X=0.0
        Y(1)=0.0
        Y(2)=1.0
        Y(4)=0.1
F     DERIV,TERM1,TERM2,OUTPUT
      CALL FNOL2(J,N,G,L,M,NE,X,Y,D,DERIV,TERM1,OUTPUT)
      READ 2,NE,J
      Y(N+3)=0.0
      CALL FNOL2(J,N,G,L,M,NE,X,Y,D,DERIV,TERM2,OUTPUT)
      GO TO 1
      END
      FOR
      SUBROUTINE DERIV(X,Y,D)
      DIMENSION Y(1),D(1)
      D(1)=Y(2)
      D(2)=-Y(1)
      Y(3)=Y(1)
      RETURN
      END
      FOR
      SUBROUTINE TERM1(X,Y,D,F)
      DIMENSION Y(1),D(1)
      F=D(1)
      D(3)=D(1)**2+D(2)**2
      RETURN
      END
      FOR
      SUBROUTINE TERM2(X,Y,D,F)
      DIMENSION Y(1),D(1)
      F=X-3.1415927
      D(3)=D(1)**2+D(2)**2
      RETURN
      END

```

THE DATA FOR THE PROBLEM ARE

	J	N	G	L	M	NE
CARD 1	3	2	.01	1	0	0
	NE	J				
CARD 2	5	2				
CARD 3	0					

Example No. 3

Compute the altitude and velocity of a point mass dropped vertically from an altitude H . Assume that DRAG and GRAVITY are the only forces acting on the body. Read in all controls and initial conditions as data. Print out VELOCITY, ALTITUDE and MACH NO. EVERY 10 TIME STEPS. Use an initial step size of 0.1 sec. and hold $10^{-7} \leq RE_1 \leq 10^{-4}$. Terminate when $H \leq H_{TERM}$.

The equation of motion is:

$$\frac{d^2 y}{dt^2} = \ddot{y} = \frac{-dV}{dt} = -g + \frac{C_D \rho_o A}{2M} e^{-\beta y} V^2$$

where

$g = 32.174 = \text{constant} = \text{acceleration of gravity (ft/sec}^2\text{)}$

$C_D = .1 = \text{constant} = \text{drag coeff. dimensionless}$

$\rho_o = .0034 = \text{constant} = \text{density of air lb/ft}^3$

$\beta = \frac{1}{22000} = \text{constant}$

$M = \text{Mass} = \text{constant} = \text{mass of body (slugs)}$

$V = \dot{y} = \text{velocity (ft/sec)}$

$y = H = \text{altitude (ft)}$

$A = 1.0 = \text{area of body (ft}^2\text{)}$

$C = \text{sound speed} = 1116.4 \text{ (ft/sec) constant}$

$H_{term} = \text{Termination Alt. (ft)}$

Let $\dot{y} = p$. Then

$$\dot{p} = \ddot{y} = -g + \frac{C_D \rho_o A}{2M} e^{-\beta y} V^2$$

and $\dot{y} = p$

become our two first order equations.

Use the following notation for programming purposes.

Derivatives

$$D(1) = \ddot{y}$$

$$D(2) = \dot{y}$$

Integrated variables

$$C(1) = \dot{y}$$

$$C(2) = y$$

Data and Intermediate Storage

$$Y(1) = V$$

$$Y(2) = y_0$$

$$Y(3) = \text{Mass}$$

$$Y(4) = C_D$$

$$Y(5) = A$$

$$Y(6) = e^{-\beta y}$$

$$Y(7) = H_{\text{Term}} (\text{termination Alt. ft})$$

$$Y(8) = G = \Delta t_0$$

$$Y(9) = C_D A \rho_0 / 2M$$

Control Variables

$$K(1) = J$$

$$K(2) = N$$

$$K(3) = L$$

$$K(4) = M$$

$$K(5) = NE$$

The FORTRAN program is:

```

      FOR
C     MAIN PROGRAM
      DIMENSION Y(10),D(5),C(5),K(5)
      COMMON Y,K
111   FORMAT(5I5)
10    FORMAT(5E14.5)
11    READ 10,(Y(I),I=1,8)
      IF(Y(2))20,20,30
F     DERIV,TERM,OUTPUT
30    READ 111,(K(I),I=1,5)
      C(1)=Y(1)
      C(2)=Y(2)
      N=K(2)
      T(N+3)=0.0
      T=0.0
      Y(9)=(Y(4)*Y(5)*.0034)/2.*Y(3)
      CALL FNOL2(K(1),K(2),Y(8),K(3),K(4),K(5),T,C,D,DERIV,TERM,OUTPUT)
      GO TO 11
20    STOP
      END

```

```

      FOR
SUBROUTINE DERIV(T,C,D)
DIMENSION Y(10),D(5),C(5),K(5)
COMMON Y,K
Y(6)=EXP(-C(2)/22000.)
D(1)=-32.174+Y(9)*Y(6)*C(1)**2
D(2)=C(1)
RETURN
END

```

```

      FOR
SUBROUTINE TERM(T,C,D,F)
DIMENSION Y(10),D(5),C(5),K(5)
COMMON Y,K
F=C(2)-Y(7)
D(3)=C(1)/1116.4
RETURN
END

```

The data for the problem are:

	V	y_0	Mass	C_D	A
card 1	0.0	100000.	.7	.1	1.
	blank	H_{Term}	G		
card 2	-	10000.	.1		
	J	N	L	M	NE
card 3	2	2	1	10	4

B. Results of Test Problems*

1. The equations in Example 1

$$\frac{dy_1}{dx} = y_2 \quad (1)$$

and

$$\frac{dy_2}{dx} = -y_1 \quad (2)$$

have the solutions

$$y_1 = \sin x$$

and

$$y_2 = \cos x$$

Equations (1) and (2) were integrated numerically from $0 \leq x \leq 3\pi$ using FNOL1 and FNOL2 with a constant step size of .01. The results of the integration and an error analysis are presented in Tables 1 and 2. The FNOL1 (single precision) solution required 0.8 minutes machine time on the IBM 7090. The FNOL2 (double precision) integration took 0.9 minutes.

*Unless otherwise stated, all integrations were performed in the Adams-Moulton mode.

2. The following equation which is given in Hildebrand, reference (3) as an example of instability with Milne's method, is integrated numerically from $0 \leq x \leq 9$, with constant step size of .01.

The equation is:

$$\frac{dy}{dx} = 2 - 2y ; \quad y(0) = 2$$

The results are compared to the true solution $y(x) = e^{-2x} + 1$ in Table 3 and as can be seen the numerical solution converges monotonically to the true solution. The integration required 900 steps and .58 minutes running time on the IBM 7090. The maximum error was 1.49×10^{-8} .

The run was repeated with $NE = 4$ and it required only 51 steps and .03 mins. of machine time. The maximum error was 6.69×10^{-6} .

3. $Y = \tan X$ was differentiated twice and appropriate substitutions were made to arrive at the following second order equation:

$$\ddot{y} = 2 y \dot{y}$$

The above equation was integrated with a step size of .01 from $0 \leq x \leq 1$ with $y(0) = 0$ and $\dot{y}(0) = 1$. The maximum error observed was $1. \times 10^{-8}$. The run required 0.6 min. on the 7090.

The run was repeated with $NE = 4$ and the maximum error was 3.3×10^{-5} . The running time was reduced to .07 min.

4. $\ddot{a} = \omega A \cos(\omega T)$ was integrated from $0 \leq T \leq \frac{\pi}{5}$ using a constant step size of .1 with

$$\omega = 1$$

$$A = 1$$

The maximum absolute error (ϵ_A) observed was 2.9×10^{-5}

$$\epsilon_A = |A \sin \omega T - \int \dot{a}|$$

Using the same Δt and

$$\omega = 8$$

$$A = 15$$

the integration becomes unstable.

The above equation was integrated from 0 to 3π with variable step size using both FNOL1 and FNOL2. The results are presented in Table 4, but may be summarized as follows:

The FNOL1 (single precision) run with NE = 6 required 1.58 min. machine time with a maximum absolute error of 3.95×10^{-3} .

The FNOL2 (double precision) run with NE = 4 took 1.0 min. and had a maximum error of 4.13×10^{-6} .

V. Summary

The primary objectives of this report have been to present the user with a detailed description of the use of FNOL2 and also to provide a comparison of timing and accuracy between FNOL1 and FNOL2.

The test results show that the use of double precision in FNOL2 does not necessarily increase running time when a variable step size is used. Table 4 shows a case where the use of double precision has produced higher accuracy with reduced running time. This example is not intended to be conclusive, since an optimal step size for a given problem may produce equivalent accuracy using only single precision.

The automatic adjustment of step size is a very valuable feature. It

can result in significant savings of machine time with a possible reduction in accuracy. The trade of accuracy for reduced running time is a choice which the user must make. The choice should be based on careful analysis of the problem, experience from previous results, and accuracy requirements for a particular problem.

APPENDIX A

```

      FOR
      SUBROUTINE FNOL2(J,N,G,L,M,NE,X,Y,D,DERIV,TERM,OUTPUT)
C      6-63
      DIMENSION Y(50),D(50),YB(30,6),GI2(30),GI3(30),GI4( 30),EF(30),
1 EF1(30),EF2(30),EF3(30),Y1(30),ERROR(30),HA(31),YA(50),DA(50),
2YC(30),YP(30),YAL(50),YL(50),YCL(30)
      ERASE ((YB(I,J),J=1,5),ERROR(I),YL(I),I=1,N),XU,NA,NF,NG,F,FA,
      FB,FC,FD
5      NB=1
      EC=Y(N+3)
      IF(EC)1000,1,1
1000 IF(EC+1.)1001,1,1001
1001 EC=0.
      1 H=G
      2 HZ=H
      3 LN=N+XMAXOF(L,3)
13      ENE=NE
14      IF(J-2) 16,15,21
15      IF(NE)18,16,18
16      JA=4
17      GO TO 22
18      RE1=10.**(-ENE)
19      RE2=10.**(-ENE-3.0)
20      REM=10.**(-ENE-1.5)
21      JA=1
22      CALL DERIV(X,Y,D)
      DO 300 I=1,N
      GI2(I)=D(I)
      GI3(I)=D(I)
      GI4(I)=D(I)
300 EF(I)=D(I)
27      CALL OUTPUT(X,Y,D,ERROR,N,L,H)
28      FD=Y(N+1)
29      IF(J-2)30,129,30
30      GO TO(31,37,35,37),JA
31      DO 33 I=1,LN
32      YA(I)=Y(I)
      YAL(I) =YL(I)
33      DA(I)=D(I)
34      GO TO 37
35      HB=H
36      H=2.*H
37      HD2=.5*H
      DO 39 I=1,N
38      YB(I,NB)=D(I)
      XL = D(I) * HD2
39      CALL DPA(Y(I),YL(I),XL,0.,Y1(I),XXL)
      CALL DPA (X,XU,HD2,0.,XX,XXL)
40      CALL DERIV (XX,Y1,GI2)
41      DO 42 I=1,N
      XL = GI2(I)*HD2
42      CALL DPA(Y(I),YL(I),XL,0.,Y1(I),XXL)
      CALL DPA (X,XU,HD2,0.,XX,XXL)
43      CALL DERIV (XX,Y1, GI3)
44      DO 45 I=1,N
      XL=GI3(I)*H
45      CALL DPA(Y(I),YL(I),XL,0.,Y1(I),XXL)

```

```

      CALL DPA(X,XU,H,0.,XX,XXL)
46 CALL DERIV(XX,Y1,GI4)
47 HD6 =H/6.
      GO TO(48,55,60,66),JA
48 DO 52 I=1,N
      XL=(D(I) + 2.*(GI2(I) + GI3(I)) +GI4(I))*HD6
49 CALL DPA (Y(I),YL(I),XL,0.,Y(I),YL(I))
50 YC(I)=Y(I)
      YCL(I)=YL(I)
51 Y(I)=YA(I)
      YL(I) =YAL(I)
52 ERROR(I)=0.
53 JA=3
54 GO TO 35
55 DO 57 I=1,N
      XL=(D(I) + 2.*(GI2(I) + GI3(I)) +GI4(I))*HD6
56 CALL DPA (Y(I),YL(I),XL,0.,Y(I),YL(I))
57 ERROR(I)=(Y(I)-YP(I))/15.
58 JA=1
59 GO TO 69
60 DO 62 I=1,N
61 Y(I)=YC(I)
      YL(I) =YCL(I)
      XL=(D(I) + 2.*(GI2(I) + GI3(I)) +GI4(I))*HD6
62 CALL DPA(YA(I),YAL(I),XL,0.,YP(I),XXL)
63 H=HB
64 JA=2
65 GO TO 69
66 DO 68 I=1,N
      XL=(D(I) + 2.*(GI2(I) + GI3(I)) +GI4(I))*HD6
67 CALL DPA (Y(I),YL(I),XL,0.,Y(I),YL(I))
68 ERROR(I)=0.
69 CALL DPA (X,XU,H,0.,X,XU)
70 CALL DERIV(X,Y,D)
71 FC=F
72 CALL TERM(X,Y,D,F)
73 IF(ABSF(F)-1.0E-5 )731,731,733
731 NF=5
732 GO TO 124
733 IF(F)74,124,76
74 FA=1.
75 GO TO 77
76 FB=1.
77 IF(FA-FB)83,78,83
78 NF=NF+1
79 JA=4
80 NB=1
81 H=H*F/(FC-F)
82 IF(NF-4)37,37,124
83 IF(NE)84,117,84
84 IF(JA-1) 117,85,117
85 IF(J-3) 86,117,86
86 DO 95 I=1,N
      IF(Y(I))886,885,886
885 HA(I)=1000.
      GO TO 95
886 IF(EC) 880,890,87
87 IF(ABSF(Y(I))-EC) 880,880,890
880 IF(ABSF(ERROR(I))-RE2) 882,94,881
881 IF(ABSF(ERROR(I))-RE1)94,94,882

```

NOLTR 63-171

```

882 HA(I)=H*(REM/(ABSF(ERROR(I))+.000000001))**(.2)
883 GO TO 95
890 IF(ABSF(ERROR(I)/Y(I))-RE2)892,94,891
891 IF(ABSF(ERROR(I)/Y(I))-RE1)94,94,892
892 HA(I)=H*(REM/(ABSF(ERROR(I)/Y(I))+.000000001))**(.2)
893 GO TO 95
94 HA(I)=H
95 CONTINUE
96 HA(N+1)=HA(N)
HB = HA(1)
97 DO 98 I=2,N
98 HB = MIN1F(HA(I),HB)
99 IF(H-HB)100,117,101
100 IF(HZ-H)101,101,116
101 DO 103 I=1,LN
102 Y(I)=YA(I)
YL(I)=YAL(I)
103 D(I)=DA(I)
104 IF(NB-6)107,105,105
105 CALL DPS (X,XU,H,0.,X,XU)
106 GO TO 109
107 CALL DPS (X,XU,2.*H,0.,X,XU)
108 HZ=H
109 H=HB
CALL DERIV(X,Y,D)
110 NB=1
111 XABS=ABSF(.000001*X)
112 IF(ABSF(H)-XABS)113,113,117
113 NG=NG+1
114 H=SIGNF(XABS,HB)
115 IF(NG-10)124,126,126
116 HZ=H
117 IF(M)118,118,121
118 IF(ABSF((Y(N+1)-FD))-Y(N+2))29,119,119
119 FD=Y(N+1)
120 GO TO 124
121 NA=NA+1
122 IF(M-NA)123,123,29
123 NA=0
124 CALL OUTPUT(X,Y,D,ERROR,N,L,H)
125 IF(NF-4)29,29,126
126 PRINT 127
127 FORMAT(1H0)
128 RETURN
129 NB=NB+1
130 IF(NB-6)30,131,136
131 DO 134 I=1,N
132 EF3(I)=YB(I,3)
133 EF2(I)=YB(I,4)
134 EF1(I)=YB(I,5)
135 GO TO 137
136 NB=10
137 HD24 =H/24.
DO 138 I=1,N
XL =(55.*D(I) -59.*EF1(I) +37.*EF2(I) -9.*EF3(I))*HD24
138 CALL DPA(Y(I),YL(I),XL,0.,YP(I),XXL)
CALL DPA (X,XU,H,0.,XX,XXL)
139 CALL DERIV(XX,YP,EF)
140 DO 142 I=1,LN
141 YA(I)=Y(I)

```

MOLTR 63-171

```

      YAL(I) =YL(I)
142 DA(I)=D(I)
143 DO 148 I=1,N
      XL = (9.*EF(I) +19.*D(I) -5.*EF1(I) +EF2(I))*HD24
144 CALL DPA(Y(I),YL(I),XL,0.,Y(I),YL(I))
145 ERROR(I)=(YP(I)-Y(I))/14.
146 EF3(I)=EF2(I)
147 EF2(I)=EF1(I)
148 EF1(I)=D(I)
149 GO TO 69
      END

```

```

      FOR
      SUBROUTINE OUTPUT(X,Y,D,ERROR,N,L,H)
      DIMENSION Y(50),D(50),ERROR(30)
      PRINT 1,X,D(1),Y(1),ERROR(1),H
      1 FORMAT(1H0,1P5E14.7)
      IF(N-1) 20,20,10
10 PRINT 2,((D(I),Y(I),ERROR(I)),I=2,N)
      2 FORMAT(15X,1P3E14.7)
20 IF(L)50,50,30
30 NZ=N+1
      NL=N+L
      PRINT 3,(D(I),I=NZ,NL)
      3 FORMAT(15X,1P5E14.7)
50 RETURN
      END

```

TABLE 1					
X	FNOL1 COS(X)	FNOL2 COS(X)	TABLE COS(X)	S.P. ERROR	D.P. ERROR
0.0	1.0000000	1.0000000	1.0000000	0.	0.0000000
0.5	0.87758243	0.87758256	0.87758256	0.00000013	0.0000000
1.0	0.54030212	0.54030234	0.54030231	0.00000019	0.00000003
1.5	0.070737214	.070737256	0.070737200	.00000012	.00000005
2.0	-0.41614642	-0.41614677	-0.41614684	.00000042	.00000007
2.5	-0.80114267	-0.80114356	-0.80114362	.00000095	.00000006
3.0	-0.98999119	-0.98999248	-0.98999250	.00000131	.00000002
3.5	-0.93645532	-0.93645673	-0.93645669	.00000137	.00000004
4.0	-0.65364260	-0.65364373	-0.65364362	.00000102	.00000011
4.5	-0.21079557	-0.21079597	-0.21079580	.00000023	.00000017
5.0	0.28366137	-0.28366200	0.28366218	.00000081	.00000018
5.5	0.70866791	0.70866962	0.70866977	.00000186	.00000015
6.0	0.96016772	0.96017022	0.96017029	.00000257	.00000007
6.5	0.97658492	0.97658768	0.97658763	.00000271	.00000005
7.0	0.75390015	0.75390243	0.75390225	.00000210	.00000018
7.5	0.34663444	0.34663559	0.34663532	.00000088	.00000027
8.0	-0.14549924	-0.14549973	-0.14550003	.00000079	.00000030
8.5	-0.60200945	-0.60201164	-0.60201190	.00000145	.00000026
9.0	-0.91112657	-0.91113012	-0.91113026	.00000369	.00000014

X	TABLE 2		S.P. ERROR	D.P. ERROR
	FNOL1 SIN(X)	FNOL2 SIN(X)		
0.0	0.	0.	0.000000000	0.
.5	0.47942542	0.47942552	0.00000012	0.00000002
1.0	0.84147059	0.84147097	0.00000039	.00000001
1.5	0.99749435	0.99749498	0.00000064	.00000001
2.0	0.90929670	0.90929745	0.00000073	.00000002
2.5	0.59847157	0.59847221	0.00000057	.00000007
3.0	0.14111992	0.14112012	0.00000009	.00000011
3.5	-0.35078259	-0.35078310	.00000064	.00000013
4.0	-0.75680106	-0.75680239	.00000144	.00000011
4.5	-0.97752814	-0.97753007	.00000198	.00000005
5.0	-0.95892226	-0.95892433	.00000202	.00000005
5.5	-0.70553880	-0.70554047	.00000153	.00000014
6.0	-0.27941496	-0.27941572	.00000054	.00000022
6.5	0.21511917	0.21511974	.00000082	.00000025
7.0	0.65698439	0.65698639	.00000221	.00000021
7.5	0.93799679	0.93799988	.00000319	.00000010
8.0	0.98935486	0.98935829	.00000339	.00000004
8.5	0.79848440	0.79848731	.00000271	.00000020
9.0	0.41211718	0.41211880	.00000130	.00000032

TABLE 3

$$y = \int \dot{y} = e^{-2x} + 1 ; y(0) = 2$$

X	$Y(X)_C$	$Y(X)_T^*$	$ Y_C - Y_T $
0.0	2.0		
.5	1.3678795	1.3678794	6.00×10^{-8}
1.0	1.1353353	1.13533528	1.2×10^{-8}
1.5	1.0497871	1.04978707	3×10^{-8}
2.0	1.0183156	1.01831564	6×10^{-8}
2.5	1.0067379	1.00673794	4×10^{-8}
3.0	1.0024787	1.00247875	5×10^{-8}
3.5	1.0009119	1.00091188	1.2×10^{-8}
4.0	1.0003355	1.00033546	4×10^{-8}
4.5	1.0001234	1.00012339	1×10^{-8}
5.0	1.0000454	1.0000455	1×10^{-8}
.	.		
.	.		
.	.		
8.39	1.00000000	1.00000000	
9.0	1.0000000	1.00000000	

*Table Values

TABLE 4

$$\epsilon = |A \sin(\omega T) - \int A \omega \cos(\omega t)|$$

t	FNOL 1 ϵ	FNOL2 ϵ
0.0	0.0	0.0
1.0	3.48×10^{-5}	7.08×10^{-7}
2.0	1.12×10^{-4}	1.70×10^{-6}
3.0	3.94×10^{-4}	2.17×10^{-6}
4.0	7.05×10^{-4}	3.0×10^{-6}
5.0	1.25×10^{-3}	3.83×10^{-6}
6.0	1.76×10^{-3}	4.13×10^{-6}
7.0	2.33×10^{-3}	3.80×10^{-6}
8.0	3.24×10^{-3}	3.33×10^{-6}
9.0	3.95×10^{-3}	1.43×10^{-6}
3π	3.41×10^{-3}	3.06×10^{-6}
Machine Time	1.58 Min.	1.0 Min.

EXTERNAL DISTRIBUTION LIST (M1)

	Copies		Copies
Chief, Defense Atomic Support Agency Washington 25, D. C. Attn: Document Library Branch	2	Commanding Officer Harry Diamond Laboratory Washington 25, D. C. Attn: Library, Bldg. 92, Rm. 211	
Commanding General Aberdeen Proving Ground, Md. Attn: Tech. Information Branch	2	Director of Intelligence Headquarters, USAF Washington 25, D. C. Attn: AOIN-3B	
Attn: Director, Ballistic Research Laboratories	2		
National Aeronautics and Space Agency Geo. C. Marshall Space Flight Center Huntsville, Alabama Attn: M-SandS-PT (Mr. H. A. Connell)	3	Langley Research Center Hampton, Virginia Attn: Theoretical Aero- dynamics Division	1
		Attn: Librarian	2
		Attn: Adolf Busemann	1
ASTIA Arlington Hall Arlington, Va.	10	Attn: Robt. L. James	1
Chief, Bureau of Naval Weapons Department of the Navy Washington 25, D. C. Attn: DLI-3	4	Scientific and Technical Information Facility P. O. Box 5700 Bethesda, Maryland Attn: NASA Representative	
Attn: RR-14	1		
Attn: RMMS	1	Ames Research Center Moffett Field, California Attn: Librarian	
Attn: RRRE	1		
Attn: RRMA	1	NASA Flight Research Center Edwards, California Attn: Mr. W. C. Williams	
Attn: RMGA	1		
Attn: RUME-4	5	Lewis Research Center 21000 Brookpark Road Cleveland 11, Ohio Attn: Librarian Attn: Chief, Supersonic Propulsion Div.	
Commanding Officer and Director David Taylor Model Basin Aerodynamics Laboratory Washington 7, D. C. Attn: Library			
Commanding Officer and Director David Taylor Model Basin Applied Mathematics Laboratory Washington 7, D. C. Attn: Dr. H. Polachek		Commander Pacific Missile Range Point Mugu, California Attn: Technical Library	2
Attn: Dr. D. Shanks			
Attn: Mr. T. Walton		National Bureau of Standards Washington 25, D. C. Attn: Applied Math. Div.	
Attn: Mr. L. Schmid			

EXTERNAL DISTRIBUTION LIST (M1)

	Copies		Copies
Commander U. S. Naval Ordnance Test Station China Lake, California Attn: Technical Library Attn: Code 406 Attn: Code 507 Attn: Code 5019		The Avco Manufacturing Corp., Research Laboratories 2385 Revere Beach Pkwy. Everett 49, Massachusetts	
Commander U. S. Naval Weapons Laboratory Dahlgren, Virginia Attn: Technical Library	2	CONVAIR Corp., A Division of General Dynamics Corp., Daingerfield, Texas	
Commanding Officer ONR Branch Office Box 39, Navy 100 Fleet Post Office New York, New York		Cornell Aeronautical Lab.Inc., P. O. Box 235, 4455 Genessee St. Buffalo 21, New York Attn: Librarian	
Office of Naval Research T-3 Building, Room 2715 Washington 25, D. C. Attn: Head, Mathematics Branch Attn: Dr. T. L. Saaty		Douglas Aircraft Company, Inc. Santa Monica Division 3000 Ocean Park Blvd., Santa Monica, California Attn: Chief Engineer	
Naval Research Laboratory Washington 25, D. C. Attn: Dr. H.M.Trent, Code 6230 Attn: Code 2027		Navy Ordnance Division Eastman Kodak Company 50 West Main Street Rochester 14, New York Attn: Mr. W. B. Forman	2
Commander Aeronautical Systems Division Air Force Systems Command Wright-Patterson AF Base, Ohio Attn: WCOSI-3	2	Missile and Ordnance Systems Dept., General Electric Co., 3198 Chestnut Street Philadelphia 4, Pennsylvania Attn: L. Chasen, Mgr.Library	2
Oak Ridge National Laboratory Oak Ridge, Tennessee Attn: Dr. A. S. Householder		Commander U. S. Naval Weapons Laboratory Dahlgren, Virginia Attn: Mr. Dave Brown Attn: Mr. Dorritt Reinsdorf	
AER, Inc. 871 E. Washington Street Pasadena, California Attn: Dr. A.J.A.Morgan		Mathematics Research Center U. S. Army University of Wisconsin Madison, Wisconsin Attn: Prof. R. E. Langer Attn: Dr. Donald Greenspan	
Allied Research Associates, Inc. 43 Leon Street Boston 15, Massachusetts Attn: Mis Joyce Larkin			

EXTERNAL DISTRIBUTION LIST (M1)

<u>No. of Copies</u>		<u>No. of Copies</u>	
	Guggenheim Aeronautical Laboratory California Institute of Technology Pasadena 4, California		Case Institute of Technology Cleveland 6, Ohio
1	Attn: Aeronautics Library	1	Attn: G. Kuerti
	Hughes Aircraft Co. Florence Avenue at Teale Street Culver City, California		Cornell University Graduate School of Aero. Eng. Ithaca, New York
1	Attn: Mr. Dana H. Johnson R & D Tech Library	1	Attn: Prof. W. R. Sears
	McDonnell Aircraft Corp. P. O. Box 516 St. Louis 3, Missouri		Applied Physics Laboratory Johns Hopkins University 8621 Georgia Avenue Silver Spring, Maryland
1	Attn: Engineering Library, Dept. 644	2	Attn: Dr. N. H. Choksy
	United Aircraft Corporation 400 Main Street East Hartford 8, Connecticut		The Johns Hopkins University Charles and 34th Streets Baltimore 18, Maryland
1	Attn: Chief Librarian	1	Attn: Dr. Francis H. Clauser
	Superintendent U. S. Naval Postgraduate School Monterey, California		Mathematics Department Harvard University Cambridge 39, Massachusetts
1	Attn: Library, Technical Reports Section	1	Attn: Prof. G. Birkhoff
	University of California Berkeley 4, California	2	Director Institute for Fluid Dynamics and Applied Mathematics University of Maryland College Park, Maryland
1	Attn: Dr. S. A. Schaaf		Massachusetts Institute of Tech. Cambridge 39, Massachusetts
	University of California Los Angeles 24, California	1	Attn: Prof. Joseph Kaye, Room 1-212
1	Attn: Prof: M.R.Hestenes		University of Michigan Randall Laboratory Ann Arbor, Michigan
	California Institute of Technology Pasadena 4, California	1	Attn: Prof. Otto Laporte
1	Attn: Aeronautics Department		
	Jet Propulsion Laboratory California Institute of Technology Pasadena 4, California	1	Supervisor Technical Documents Services Willow Run Laboratories University of Michigan Willow Run Airport Ypsilanti, Michigan
1	Attn: Library		
1	Attn: Mr. J. F. Scott		
1	Attn: Mr. D. B. Holdridge		
1	Attn: Mr. G. N. Gianopulos		

EXTERNAL DISTRIBUTION LIST M1

No. of
Copies

1	University of Minnesota Minneapolis 14, Minnesota Attn: Mechanical Eng. Dept.	2	Lockheed Missile and Space Co. Sunnyvale, California Attn: Mr. R. Donati Dept. 5924
	The Ohio State University Research Foundation Columbus 10, Ohio	1	Attn: Mr. C. A. Louis III Dept. 5340
1	Attn: Security Officer Carlos Brewer	1	Attn: Mr. D. L. McGough
1	Attn: Dr. Paul M. Pepper		Goddard Space Flight Center Greenbelt, Maryland
1	Applied Math. and Statistics Lab. Stanford University Stanford, California	1	Attn: Mr. T. Michels
		2	Attn: Mr. R. Groves Code 520.5
	Defense Research Laboratory The University of Texas P. O. Box 8092 Austin 12, Texas		Attn: Mr. R. Danek Code 543
1	Attn: H. D. Krick, Asst. Director		
	Department of Civil Engineering Colorado State University Fort Collins, Colorado		
1	Attn: Prof. J. E. Germak		
	General Electric Company Missile and Space Division 3198 Chestnut Street Philadelphia, Pa.		
1	Attn: Mr. Jerome Persh		
1	Attn: Mr. John F. Butler		
	University of Michigan Serials and Documents Section Ann Arbor, Michigan		
	Lebanon Valley College Annville, Pennsylvania		
1	Attn: Dr. B. Bissinger		
	McDonnell Aircraft Corp., P. O. Box 516 St. Louis 3, Missouri		
1	Attn: Mr. Robert C. Brown		
1	Attn: Mr. Robert V. Brulle		
1	Attn: Mr. Gerald D. Griffin		

CATALOGING INFORMATION FOR LIBRARY USE

BIBLIOGRAPHIC INFORMATION

	DESCRIPTORS		CODES		DESCRIPTORS		CODES	
SOURCE	NOL technical report		NOLTR		Unclassified - 28		up28	
REPORT NUMBER	63-171		63171					
REPORT DATE	17 July 1963		9763					

SUBJECT ANALYSIS OF REPORT

	DESCRIPTORS		CODES		DESCRIPTORS		CODES	
FMOL2			FMOL2		Mathematics		MATH	
Subroutine			SUBO		Double		DOUB	
Numerical			NUMB		Precision		PREI	
Integration			INTG		Arithmetic		ARIT	
Fortran			FOER		Automatic		AUTO	
IBM			IBMA		Variables		VARB	
Computer			COMP		Truncation		TRCT	
Program			PROG		Errors		ERRO	
Differential			DIFE		Code		CODE	
Equations			EQUA		Range		RNG	
Solution			SOLG		Knitts		KUTI	
Internal			INTV		Adams		ADAS	

Naval Ordnance Laboratory, White Oak, Md. (NOL technical report 63-171) FNOL2, A FORTRAN (IBM 7090) SUBROUTINE FOR THE SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS WITH AUTOMATIC ADJUSTMENT OF THE INTERVAL OF INTEGRATION (U), by Jerry S. Linnekin and L.J. Belliveau. 17 July 1963. 28p. tables. (Math-enatics Dept. report M-38). NOL task 417. UNCLASSIFIED FNOL2 is a Fortran subrouting used for the numerical integration of a system of up to 30 ordinary differential equations. It uses double precision arithmetic in key locations. FNOL2 has the option of automatically varying the interval of integration, h, to hold the relative or absolute truncation error within bounds fixed by the user. A Fortran listing of the subroutine is included in this report. Abstract card is unclassified	Codes - Fortran Differential equations Computers - IBM 7090 Title I. Linnekin, II. Belliveau, III. Louis J., IV. jt. author Project	Codes - Fortran Differential equations Computers - IBM 7090 Title I. Linnekin, II. Belliveau, III. Louis J., IV. jt. author Project
Naval Ordnance Laboratory, White Oak, Md. (NOL technical report 63-171) FNOL2, A FORTRAN (IBM 7090) SUBROUTINE FOR THE SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS WITH AUTOMATIC ADJUSTMENT OF THE INTERVAL OF INTEGRATION (U), by Jerry S. Linnekin and L.J. Belliveau. 17 July 1963. 28p. tables. (Math-enatics Dept. report M-38). NOL task 417. UNCLASSIFIED FNOL2 is a Fortran subrouting used for the numerical integration of a system of up to 30 ordinary differential equations. It uses double precision arithmetic in key locations. FNOL2 has the option of automatically varying the interval of integration, h, to hold the relative or absolute truncation error within bounds fixed by the user. A Fortran listing of the subroutine is included in this report. Abstract card is unclassified	Codes - Fortran Differential equations Computers - IBM 7090 Title I. Linnekin, II. Belliveau, III. Louis J., IV. jt. author Project	Codes - Fortran Differential equations Computers - IBM 7090 Title I. Linnekin, II. Belliveau, III. Louis J., IV. jt. author Project
Naval Ordnance Laboratory, White Oak, Md. (NOL technical report 63-171) FNOL2, A FORTRAN (IBM 7090) SUBROUTINE FOR THE SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS WITH AUTOMATIC ADJUSTMENT OF THE INTERVAL OF INTEGRATION (U), by Jerry S. Linnekin and L.J. Belliveau. 17 July 1963. 28p. tables. (Math-enatics Dept. report M-38). NOL task 417. UNCLASSIFIED FNOL2 is a Fortran subrouting used for the numerical integration of a system of up to 30 ordinary differential equations. It uses double precision arithmetic in key locations. FNOL2 has the option of automatically varying the interval of integration, h, to hold the relative or absolute truncation error within bounds fixed by the user. A Fortran listing of the subroutine is included in this report. Abstract card is unclassified	Codes - Fortran Differential equations Computers - IBM 7090 Title I. Linnekin, II. Belliveau, III. Louis J., IV. jt. author Project	Codes - Fortran Differential equations Computers - IBM 7090 Title I. Linnekin, II. Belliveau, III. Louis J., IV. jt. author Project

Naval Ordnance Laboratory, White Oak, Md. (NOL technical report 63-171) FNOL2, A FORTRAN (IBM 7090) SUBROUTINE FOR THE SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS WITH AUTOMATIC ADJUSTMENT OF THE INTERVAL OF INTEGRATION (U), by Jerry S. Linnekin and L.J. Belliveau. 17 July 1963. 28p. tables. (Mathematics Dept. report M-38). NOL task 417. FNOL2 is a Fortran subroutine used for the numerical integration of a system of up to 30 ordinary differential equations. It uses double precision arithmetic in key locations. FNOL2 has the option of automatically varying the interval of integration, h, to hold the relative or absolute truncation error within bounds fixed by the user. A Fortran listing of the subroutine is included in this report. Abstract card is unclassified	Codes - Fortran Differential equations - Computers - IBM 7090 Title I. Linnekin, II. Belliveau, III. Louis J., IV. Project	Codes - Fortran Differential equations - Computers - IBM 7090 Title I. Linnekin, II. Belliveau, III. Louis J., IV. Project
Naval Ordnance Laboratory, White Oak, Md. (NOL technical report 63-171) FNOL2, A FORTRAN (IBM 7090) SUBROUTINE FOR THE SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS WITH AUTOMATIC ADJUSTMENT OF THE INTERVAL OF INTEGRATION (U), by Jerry S. Linnekin and L.J. Belliveau. 17 July 1963. 28p. tables. (Mathematics Dept. report M-38). NOL task 417. FNOL2 is a Fortran subroutine used for the numerical integration of a system of up to 30 ordinary differential equations. It uses double precision arithmetic in key locations. FNOL2 has the option of automatically varying the interval of integration, h, to hold the relative or absolute truncation error within bounds fixed by the user. A Fortran listing of the subroutine is included in this report. Abstract card is unclassified	Codes - Fortran Differential equations - Computers - IBM 7090 Title I. Linnekin, II. Belliveau, III. Louis J., IV. Project	Codes - Fortran Differential equations - Computers - IBM 7090 Title I. Linnekin, II. Belliveau, III. Louis J., IV. Project
Naval Ordnance Laboratory, White Oak, Md. (NOL technical report 63-171) FNOL2, A FORTRAN (IBM 7090) SUBROUTINE FOR THE SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS WITH AUTOMATIC ADJUSTMENT OF THE INTERVAL OF INTEGRATION (U), by Jerry S. Linnekin and L.J. Belliveau. 17 July 1963. 28p. tables. (Mathematics Dept. report M-38). NOL task 417. FNOL2 is a Fortran subroutine used for the numerical integration of a system of up to 30 ordinary differential equations. It uses double precision arithmetic in key locations. FNOL2 has the option of automatically varying the interval of integration, h, to hold the relative or absolute truncation error within bounds fixed by the user. A Fortran listing of the subroutine is included in this report. Abstract card is unclassified	Codes - Fortran Differential equations - Computers - IBM 7090 Title I. Linnekin, II. Belliveau, III. Louis J., IV. Project	Codes - Fortran Differential equations - Computers - IBM 7090 Title I. Linnekin, II. Belliveau, III. Louis J., IV. Project

Naval Ordnance Laboratory, White Oak, Md. (NOL technical report 63-171) FNOL2, A FORTRAN (IBM 7090) SUBROUTINE FOR THE SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS WITH AUTOMATIC ADJUSTMENT OF THE INTERVAL OF INTEGRATION (U), by Jerry S. Linnekin and L.J. Belliveau. 17 July 1963. 28p. tables. (Mathematics Dept. report M-38). NOL task 417. UNCLASSIFIED FNOL2 is a Fortran subroutine used for the numerical integration of a system of up to 30 ordinary differential equations. It uses double precision arithmetic in key locations. FNOL2 has the option of automatically varying the interval of integration, h, to hold the relative or absolute truncation error within bounds fixed by the user. A Fortran listing of the subroutine is included in this report. Abstract card is unclassified	Codes - Fortran Differential equations - Computers - IBM 7090 Title I. Linnekin, II. Belliveau, III. Louis J., IV. jt. author Project	Codes - Fortran Differential equations - Computers - IBM 7090 Title I. Linnekin, II. Belliveau, III. Louis J., IV. jt. author Project
Naval Ordnance Laboratory, White Oak, Md. (NOL technical report 63-171) FNOL2, A FORTRAN (IBM 7090) SUBROUTINE FOR THE SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS WITH AUTOMATIC ADJUSTMENT OF THE INTERVAL OF INTEGRATION (U), by Jerry S. Linnekin and L.J. Belliveau. 17 July 1963. 28p. tables. (Mathematics Dept. report M-38). NOL task 417. UNCLASSIFIED FNOL2 is a Fortran subroutine used for the numerical integration of a system of up to 30 ordinary differential equations. It uses double precision arithmetic in key locations. FNOL2 has the option of automatically varying the interval of integration, h, to hold the relative or absolute truncation error within bounds fixed by the user. A Fortran listing of the subroutine is included in this report. Abstract card is unclassified	Codes - Fortran Differential equations - Computers - IBM 7090 Title I. Linnekin, II. Belliveau, III. Louis J., IV. jt. author Project	Codes - Fortran Differential equations - Computers - IBM 7090 Title I. Linnekin, II. Belliveau, III. Louis J., IV. jt. author Project
Naval Ordnance Laboratory, White Oak, Md. (NOL technical report 63-171) FNOL2, A FORTRAN (IBM 7090) SUBROUTINE FOR THE SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS WITH AUTOMATIC ADJUSTMENT OF THE INTERVAL OF INTEGRATION (U), by Jerry S. Linnekin and L.J. Belliveau. 17 July 1963. 28p. tables. (Mathematics Dept. report M-38). NOL task 417. UNCLASSIFIED FNOL2 is a Fortran subroutine used for the numerical integration of a system of up to 30 ordinary differential equations. It uses double precision arithmetic in key locations. FNOL2 has the option of automatically varying the interval of integration, h, to hold the relative or absolute truncation error within bounds fixed by the user. A Fortran listing of the subroutine is included in this report. Abstract card is unclassified	Codes - Fortran Differential equations - Computers - IBM 7090 Title I. Linnekin, II. Belliveau, III. Louis J., IV. jt. author Project	Codes - Fortran Differential equations - Computers - IBM 7090 Title I. Linnekin, II. Belliveau, III. Louis J., IV. jt. author Project