Two Analyst-Oriented Computer Languages: EASL, POSE

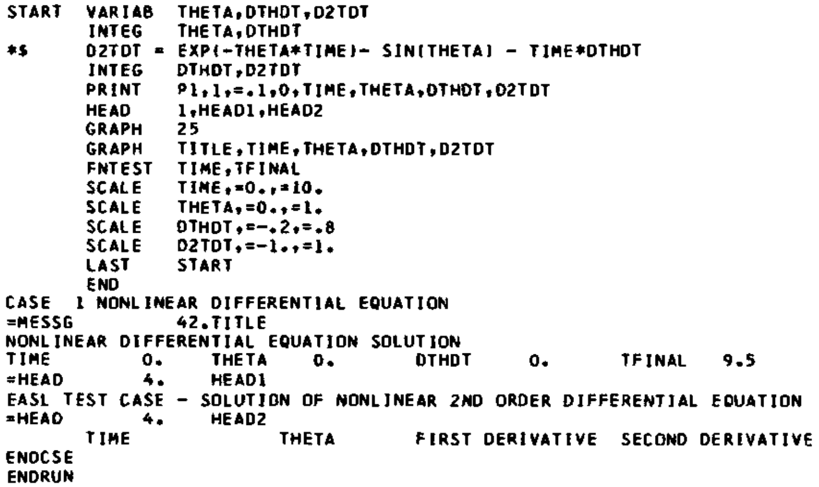
STEWART I. SCHLESINGER,
LAWRENCE SASHKIN, and KENNETH C. REED

Aerospace Corporation, San Bernardino, California

For approximately three and one-half years a continuous modest-level effort has been maintained at Aerospace Corporation to support the development of analyst-oriented computer languages which could be used for both off-line and interactive applications. Initially, this effort produced a block-oriented macro language called EASL (Engineering Analysis and Simulation Language) [1].

All of the EASL operators are macro instructions which make use of the sophisticated features of the IBM 7094 MAP assembler. EASL was made even more useful when the multiple-entry-point feature was added to IBM's FORTRAN language. The second version of the EASL language makes use of a preprocessor which prescans the user's program looking for FORTRAN-type statements and special EASL II operators, and sets up a FORTRAN subroutine with necessary linkages to entry points whenever sets of these statements are found. Each group of FORTRAN statements causes the creation of an ENTRY point. The preprocessor looks for *$ in card columns 1 and 2 of each FORTRAN card to distinguish between EASL operators and FORTRAN statements. Figure 1 illustrates the inclusion of FORTRAN statements in an EASL program. The problem involves integration of a second order differential equation.

From the user's standpoint, EASL simplifies the task of solving a problem using a large-scale computer because it automatically provides for such functions as data input, graphic and printed output, numerical integration, table look-up, and interpolation without the associated burdensome details normally encountered in programming. Even so, the user must state the solution of his problem in a procedural fashion as would any other pro-



```
START   VARIAB  THETA,DTHDT,D2TDT
        INTEG   THETA,DTHDT
*$      D2TDT = EXP(-THETA*TIME)- SIN(THETA) - TIME*DTHDT
        INTEG   DTHDT,D2TDT
        PRINT   P1,1,=.1,0,TIME,THETA,DTHDT,D2TDT
        HEAD    1,HEAD1,HEAD2
        GRAPH   25
        GRAPH   TITLE,TIME,THETA,DTHDT,D2TDT
        FNTEST  TIME,TFINAL
        SCALE   TIME,=0.,=10.
        SCALE   THETA,=0.,=1.
        SCALE   DTHDT,=-.2,=.8
        SCALE   D2TDT,=-1.,=1.
        LAST    START
        END
CASE  1 NONLINEAR DIFFERENTIAL EQUATION
=MESSG          42.TITLE
NONLINEAR DIFFERENTIAL EQUATION SOLUTION
TIME         0.    THETA    0.        DTHDT       0.       TFINAL   9.5
=HEAD        4.    HEAD1
EASL TEST CASE - SOLUTION OF NONLINEAR 2ND ORDER DIFFERENTIAL EQUATION
=HEAD        4.    HEAD2
        TIME              THETA       FIRST DERIVATIVE  SECOND DERIVATIVE
ENOCSE
ENDRUN
```

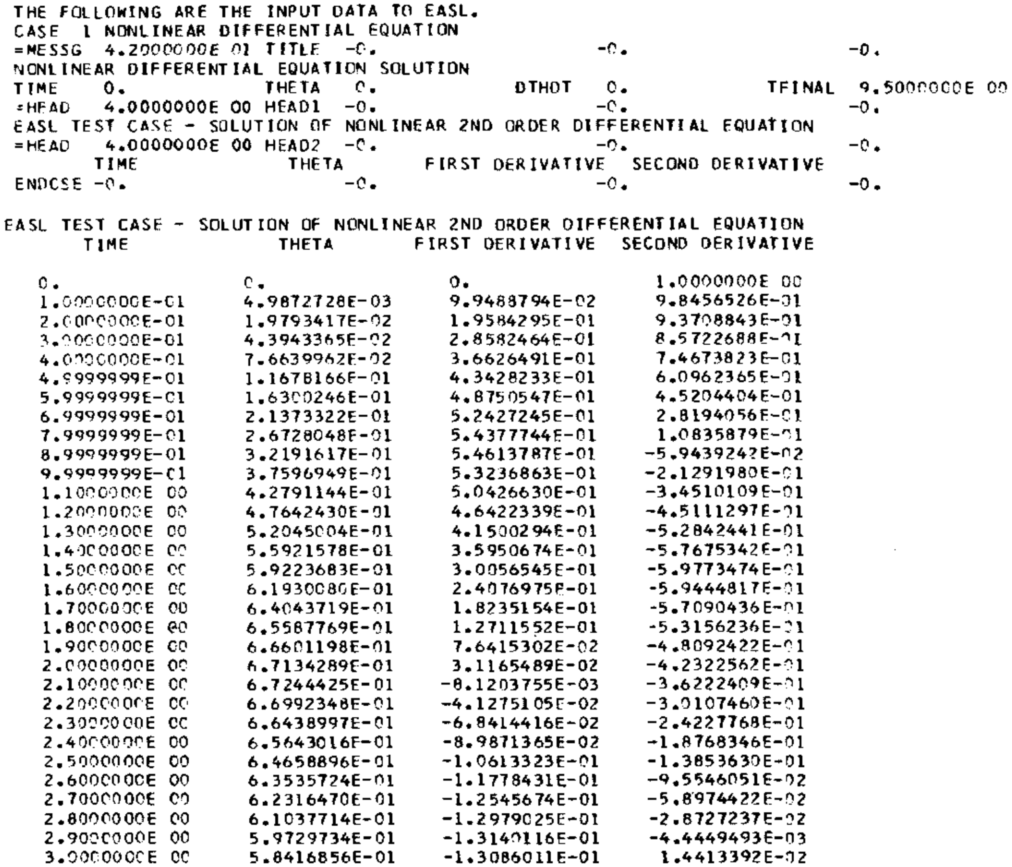
FIG. 1. EASL program and data.

grammer. Figures 2 and 3 illustrate the output capabilities of EASL. (The results are for the problem given in Fig. 1.)

During the time EASL was used on an IBM 7040/7094 II configuration, a sophisticated graphics console (see Fig. 4) was interfaced with the system. A modified version of EASL II provided a tool to permit an engineer at the console to directly control the solution of the problem he had presented to the computer. He could obtain intermediate graphic displays of plots and/or parametric values during computation. He could review previously generated graphs and modify the values of variables to direct the solution progress [2].

Within the past several months a new language has been under development which further simplifies the task of presenting problems to the computer. This new language, called POSE (Processing, Organizing, and Solving Equations) [3], provides the user with declarative as well as procedural capabilities. The operators of this new language more nearly resemble mathematical notation. Employing these operators, the user can state his problem in a general way. A preprocessor translates it into executable FORTRAN statements, setting up the proper subroutine logic, and taking care of the necessary programming details. Again, certain functions such as input and output require no programming detail of the user.

The first model of POSE which has been developed is a prototype for use on the IBM 1800 computer and has limited capabilities. This prototype model is being interfaced with an associated input typewriter and the same graphics console pictured in Fig. 4. In Fig. 5 the problem presented in Fig. 1 is stated in the POSE language.



```
THE FOLLOWING ARE THE INPUT DATA TO EASL.
CASE  1 NONLINEAR DIFFERENTIAL EQUATION
=MESSG  4.2000000E 01 TITLE  -0.                          -0.                         -0.
NONLINEAR DIFFERENTIAL EQUATION SOLUTION
TIME    0.              THETA   0.              DTHOT   0.              TFINAL  9.5000000E 00
=HEAD   4.0000000E 00 HEAD1  -0.                          -0.                         -0.
EASL TEST CASE - SOLUTION OF NONLINEAR 2ND ORDER DIFFERENTIAL EQUATION
=HEAD   4.0000000E 00 HEAD2  -0.                          -0.                         -0.
      TIME               THETA         FIRST DERIVATIVE  SECOND DERIVATIVE
ENDCSE -0.                   -0.                          -0.                         -0.

EASL TEST CASE - SOLUTION OF NONLINEAR 2ND ORDER DIFFERENTIAL EQUATION
      TIME               THETA         FIRST DERIVATIVE  SECOND DERIVATIVE

   0.                 0.                0.                 1.0000000E 00
   1.0000000E-01      4.9872728E-03     9.9488794E-02      9.8456526E-01
   2.0000000E-01      1.9793417E-02     1.9584295E-01      9.3708843E-01
   3.0000000E-01      4.3943365E-02     2.8582464E-01      8.5722688E-01
   4.0000000E-01      7.6639962E-02     3.6626491E-01      7.4673823E-01
   4.9999999E-01      1.1678166E-01     4.3428233E-01      6.0962365E-01
   5.9999999E-01      1.6300246E-01     4.8750547E-01      4.5204404E-01
   6.9999999E-01      2.1373322E-01     5.2427245E-01      2.8194056E-01
   7.9999999E-01      2.6728048E-01     5.4377744E-01      1.0835879E-01
   8.9999999E-01      3.2191617E-01     5.4613787E-01     -5.9439242E-02
   9.9999999E-01      3.7596949E-01     5.3236863E-01     -2.1291980E-01
   1.1000000E 00      4.2791144E-01     5.0426630E-01     -3.4510109E-01
   1.2000000E 00      4.7642430E-01     4.6422339E-01     -4.5111297E-01
   1.3000000E 00      5.2045004E-01     4.1500294E-01     -5.2842441E-01
   1.4000000E 00      5.5921578E-01     3.5950674E-01     -5.7675342E-01
   1.5000000E 00      5.9223683E-01     3.0056545E-01     -5.9773474E-01
   1.6000000E 00      6.1930080E-01     2.4076975E-01     -5.9444817E-01
   1.7000000E 00      6.4043719E-01     1.8235154E-01     -5.7090436E-01
   1.8000000E 00      6.5587769E-01     1.2711552E-01     -5.3156236E-01
   1.9000000E 00      6.6601198E-01     7.6415302E-02     -4.8092422E-01
   2.0000000E 00      6.7134289E-01     3.1165489E-02     -4.2322562E-01
   2.1000000E 00      6.7244425E-01    -8.1203755E-03     -3.6222409E-01
   2.2000000E 00      6.6992348E-01    -4.1275105E-02     -3.0107460E-01
   2.3000000E 00      6.6438997E-01    -6.8414416E-02     -2.4227768E-01
   2.4000000E 00      6.5643016E-01    -8.9871365E-02     -1.8768346E-01
   2.5000000E 00      6.4658896E-01    -1.0613323E-01     -1.3853630E-01
   2.6000000E 00      6.3535724E-01    -1.1778431E-01     -9.5546051E-02
   2.7000000E 00      6.2316470E-01    -1.2545674E-01     -5.8974422E-02
   2.8000000E 00      6.1037714E-01    -1.2979025E-01     -2.8727237E-02
   2.9000000E 00      5.9729734E-01    -1.3140116E-01     -4.4449493E-03
   3.0000000E 00      5.8416856E-01    -1.3086011E-01      1.4413392E-02
```

FIG. 2. EASL printed output.

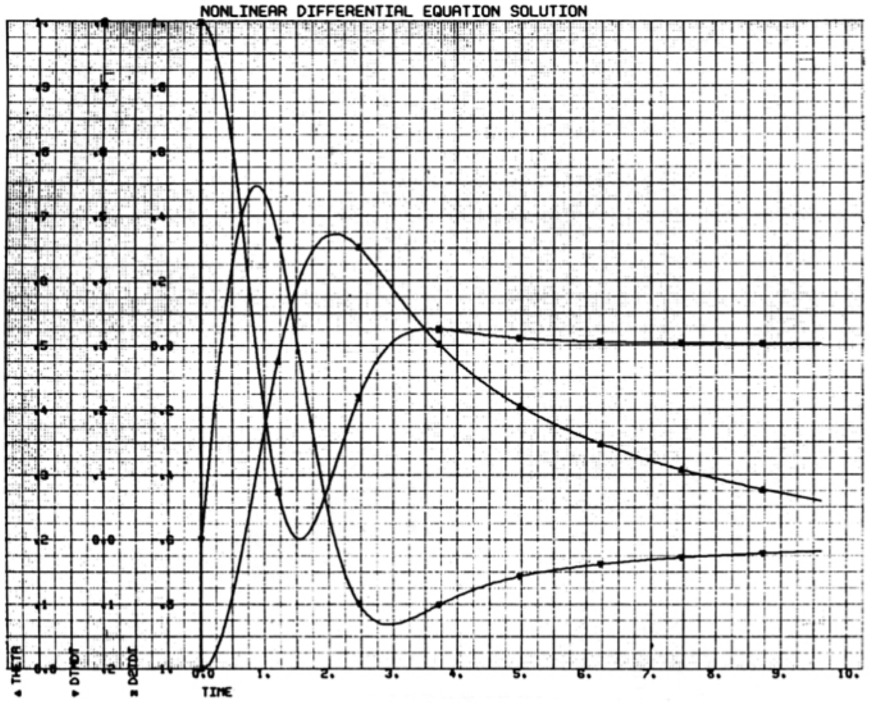


FIG. 3. EASL graphic output.

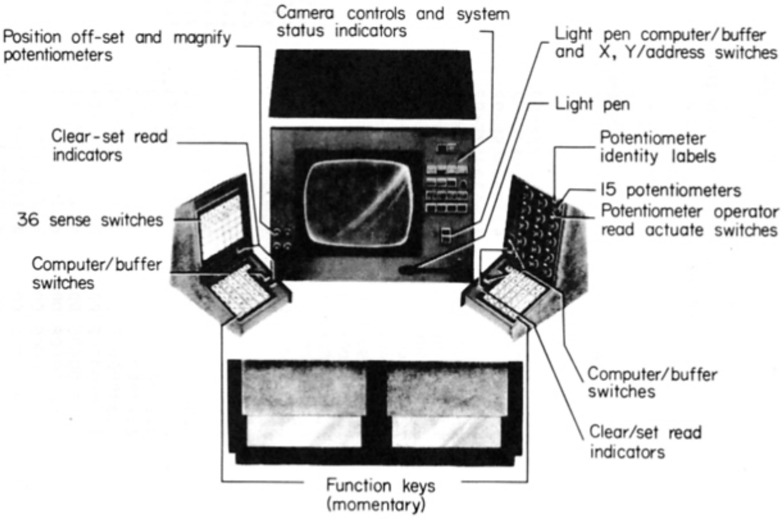
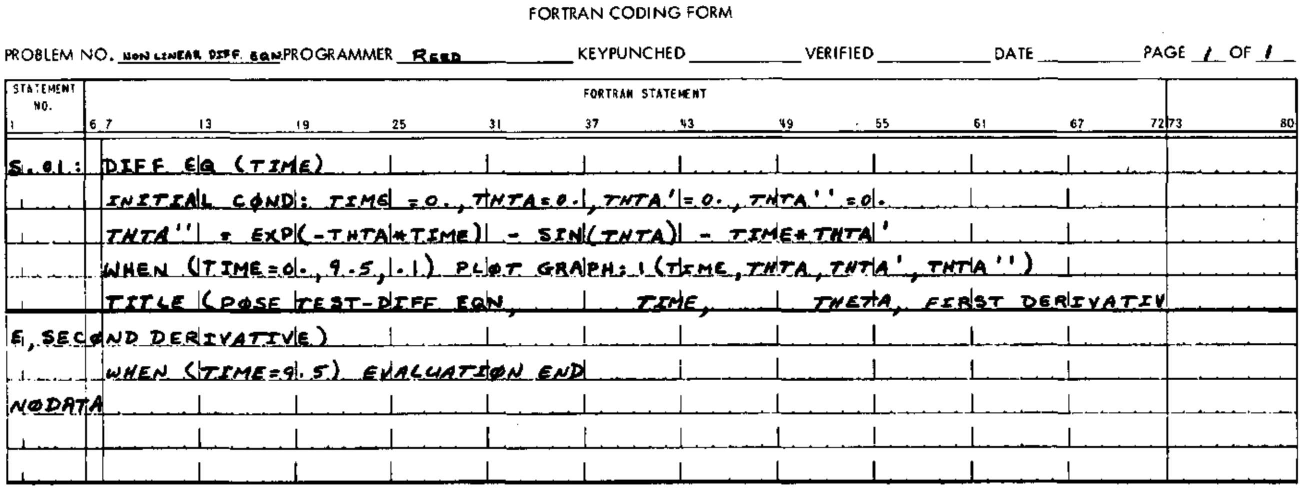

Camera controls and system status indicators
Position off-set and magnify potentiometers
Light pen computer/buffer and X, Y/address switches
Light pen
Clear-set read indicators
Potentiometer identity labels
15 potentiometers
36 sense switches
Potentiometer operator read actuate switches
Computer/buffer switches
Computer/buffer switches
Clear/set read indicators
Function keys (momentary)

FIG. 4. EASL CRT console wave form display/analyzer.



FORTRAN CODING FORM

PROBLEM NO. NON LINEAR DIFF. EQN PROGRAMMER Reed KEYPUNCHED ____ VERIFIED ____ DATE ____ PAGE 1 OF 1

```
STATEMENT
   NO.                          FORTRAN STATEMENT
1     6 7     13    19    25    31    37    43    49    55    61    67    72 73      80
S.01:  DIFF EQ (TIME)
       INITIAL CØND: TIME =0.,THTA=0.,THTA'=0.,THTA''=0.
       THTA'' = EXP(-THTA*TIME) - SIN(THTA) - TIME*THTA'
       WHEN (TIME=0.,9.5,.1) PLØT GRAPH:1(TIME,THTA,THTA',THTA'')
       TITLE (PØSE TEST-DIFF EQN,          TIME,          THETA, FIRST DERIVATIV
E,SECØND DERIVATIVE)
       WHEN (TIME=9.5) EVALUATIØN END
NØDATA
```

FIG. 5. POSE program.

REFERENCES

1. Sashkin, L., and Schlesinger, S. I., Engineering Analysis and Simulation Language (EASL), Aerospace Corporation Tech. Memo. ATM-65(S9990)-1, 1965.
2. Schlesinger, S. I., and Sashkin, L., EASL Engineering Analysis and Simulation Language, *Simulation* **6**, 2 (1966).
3. Schlesinger, S. I., and Sashkin, L., POSE: A Language for Posing Problems to a Computer, *Comm. ACM* **10**, No. 5, 279 (1967).