JAERI-M
91-069

BASIC PROGRAM TO COMPUTE URANIUM DENSITY AND VOID VOLUME FRACTION IN LABORATORY-SCALE URANIUM SILICIDE ALUMINUM DISPERSION PLATE-TYPE FUEL

May 1991

Mitsuhiro UGAJIN

日本原子力研究所
Japan Atomic Energy Research Institute

JAERI-Mレポートは，日本原子力研究所が不定期に公刊している研究報告書です。
入手の問合わせは，日本原子力研究所技術情報部情報資料課（〒319-11茨城県那珂郡東海村）あて，お申しこしください。なお，このほかに財団法人原子力弘済会資料センター（〒319-11茨城県那珂郡東海村日本原子力研究所内）で複写による実費頒布をおこなっております。

JAERI-M reports are issued irregularly.
Inquiries about availability of the reports should be addressed to Information Division, Department of Technical Information, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun, Ibaraki-ken 319-11, Japan.

編集兼発行　日本原子力研究所
印　　刷　㈱原子力資料サービス

JAERI-M 91-069

BASIC Program to Compute Uranium Density and Void Volume Fraction in Laboratory-scale Uranium Silicide Aluminum Dispersion Plate-type Fuel

Mitsuhiro UGAJIN

Department of Fuels and Materials Research
Tokai Research Establishment
Japan Atomic Energy Research Institute
Tokai-mura, Naka-gun, Ibaraki-ken

(Received April 1, 1991)

BASIC program simple and easy to operate has been developed to compute uranium density and void volume fraction for laboratory-scale uranium silicide aluminum dispersion plate-type fuel, so called miniplate. An example of the result of calculation is given in order to demonstrate how the calculated void fraction correlates with the microstructural distribution of the void in a miniplate prepared in our laboratory.

The program is also able to constitute data base on important parameters for miniplates from experimentally-determined values of density, weight of each constituent and dimensions of miniplates.

Utility programs pertinent to the development of the BASIC program are also given which run in the popular MS-DOS environment. All the source lists are attached and brief description for each program is made.

Keywords: Uranium, Silicide, Density, Void, Plate-type Fuel, Personal Computer, Utility program, BASIC, FORTRAN 77, C

JAERI－M　91－069

実験室規模ウランシリサイド　アルミニウム分散型板状燃料中のウラン密度及びボイド体積率計算用BASICプログラム

日本原子力研究所東海研究所燃料・材料工学部
宇賀神　光弘

(1991年4月1日受理)

実験室規模のウランシリサイド　アルミニウム分散型板状燃料（ミニプレート）について，燃料ミート中のウラン密度及びボイド体積率を計算するための簡易BASICプログラムを開発した。研究室で調製されたミニプレートについて，ボイド体積率の計算値と燃料ミート中のボイド分布との対応を示す一例をあげた。

本プログラムは，実験的に決定されるミニプレート各成分の密度及び重量・寸法データから，ミニプレートの主要パラメーターに関するデータベースを構築することが可能である。

MS－DOS環境下で，BASICプログラムの開発を容易にするユテリティープログラムを示した。すべてのソースリストが与えられ簡単な説明が加えられている。

東海研究所：〒319－11　茨城県那珂郡東海村白方字白根2－4

Contents

目　　次

1. Introduction

Uranium silicide (U_3Si_2), in the form of aluminum dispersed plate-type fuel, has already been adopted as a next generation fuel for use in research and test reactors in the United States and in Japan after extensive works of R & D[1, 2].
The most important irradiation performance characteristic of a fuel plate is its thickness increase resulting from fuel swelling. Uranium density and void volume fraction in the aluminum-dispersed fuel core are controlling factors of the rate of the swelling.

In general, plate-type fuels suitable and acceptable for use in research and test reactors can be fabricated with U_3Si_2 - Al dispersion compacts with uranium densities up to 4.8 g/cm^3 and with void volume fractions from ca. 4 to 14 %.

This report will deal with a BASIC program to compute uranium density and void volume fraction in laboratory-scale uranium silicide aluminum dispersion plate-type fuel, so called miniplate. An example of the result of calculation will be shown in order to demonstrate how the calculated void fraction correlates with the microstructural distribution of the void in a miniplate prepared in our laboratory.

The program is also able to constitute data base for miniplates from experimentally-determined values of density and weight of each constituent and dimensions of miniplates.

Utility programs pertinent to the development of the BASIC program are also given which run in the popular MS-DOS environment.

2. Programming environments

2.1 Personal computer operating system

Personal computer (NEC PC-9801 series) was employed using MS-DOS Ver. 2.11 or 3.1 as operating system. BASIC compiler (NEC N88-BASIC(86) Compiler version 6.0) was used to generate .EXE file which is able to work in the presence of run-time library ,i.e. N88BASIC.LIB [3]. For MS-DOS version 3.1 , PRINT.SYS should be installed as a device in CONFIG.SYS to avoid run-time error "Feature not available in (source line number)". Also, FILES = 18 or more is recommended to escape run-time error " Too many files ".

Source files can be written by a line editor EDLIN.COM (or .EXE) which is bounded to the MS-DOS system disk or , preferably, by a screen editor MIFES-98, for example. MTTK2 was used as Japanese front-end processor; any other fep may be used instead. Use of the hard disk is not inevitable but recommended.

2.2 Utility programs developed

The batch files ,N8.BAT and QX.BAT, efficient to the compile-edit routine were described as shown in Fig. 1. Unfortunately, the present version of the compiler does not provide a function of "IF ERRORLEVEL" for compile-time errors. But, redirection is valid in this system so that the diagnostic messages can be redirected to the file CLMS. In order to read and interpret the given messages, A1.EXE associated with these two batch files was created from the BASIC source list also shown in Fig.1.

A name of source file to be compiled has to be incorporated into the N8.BAT as a command line parameter without extension (.BAS). The formal argument (%1) is replaced by the filename given and eventually converted to FLNAME$, as seen from the program of A1.BAS. QX.BAT plays a roll of editing to amend errors and recompile the file, on the basis of the judgment by the A1.EXE. If an error message 'Out of memory' is encountered in the course of compile-edit routine, the main batch file N8.BAT should be booted again to resume compilation. The threshold value of FRE(3) is now set 59000 bytes empirically, which should be corrected depending on the system environment adopted.

An attempt has been made to produce another utility program , RL80.EXE, coded by FORTRAN and partly by C language[4,5] as listed in Fig. 2. The aim is to convert the variable-length files UDV.BAS and UPL.BAS (maximum 255 bytes per record with multi-statement) to the fixed-length files (80 bytes per record) . This enables one to print out the BASIC program on the MS-DOS basis by FACOM NLP installed here in JAERI in a format of maximum 80 characters per line (record). PRO FORTRAN - 77 was employed as a compiler and linker to generate RL80.EXE. Directory expressions for input/output filenames can be used within 20 characters as necessary. Selection of the disk drive can be made from the command-line argument for the accommodation of the intermediate (temporal) file XTMP. Source lists given in this report were printed by these means.

Escape sequence was used in these utilities to control character attributes and cursor addressing on the screen.

3. Explanation of the main features of the programs

The BASIC programs developed consists of the main program UDV.EXE and subprogram UPL.EXE (see Fig. 3 and Fig. 4, respectively). Both programs can call each other on line and execute respective job with the aid of F_DELETE.BAT for the removal of any record once saved on a disk file called _DATFILE.UPL.

The calculating procedure in the main program is not complex as explained in the next section ; therefore, principal efforts have been directed to the easy access to and the simple operation of data for input and output through the peripherals. In principle, all the necessary key-operations are designed to be made according to the onscreen message and to avoid hung-up of the machine.

3.1 The main program UDV.BAS

The main program calculates density of sample, uranium density and void volume fraction from temperature-density data for replacement liquid used in the density / volume measurements and from various data input through the keyboard as described in 3.1.1 and 3.1.2.

It has 10 random files of which the following six comprise only one record : IDFUEL.DAT, SAMPLDSP.DAT, PLATE.DAT, VS.DAT, TEMPWSL.DAT and WS.DAT. This means that since prior data is saved in each file automatically, it is unnecessary to repeat the same data input if it remains unchanged. Some flexibility was contemplated to input data in the file PLATE.DAT for the calculation of uranium density and void volume fraction in miniplate.

3. Explanation of the main features of the programs

The BASIC programs developed consists of the main program UDV.EXE and subprogram UPL.EXE (see Fig. 3 and Fig. 4, respectively). Both programs can call each other on line and execute respective job with the aid of F_DELETE.BAT for the removal of any record once saved on a disk file called _DATFILE.UPL.

The calculating procedure in the main program is not complex as explained in the next section ; therefore, principal efforts have been directed to the easy access to and the simple operation of data for input and output through the peripherals. In principle, all the necessary key-operations are designed to be made according to the onscreen message and to avoid hung-up of the machine.

3.1 The main program UDV.BAS

The main program calculates density of sample, uranium density and void volume fraction from temperature-density data for replacement liquid used in the density / volume measurements and from various data input through the keyboard as described in 3.1.1 and 3.1.2.

It has 10 random files of which the following six comprise only one record : IDFUEL.DAT, SAMPLDSP.DAT, PLATE.DAT, VS.DAT, TEMPWSL.DAT and WS.DAT. This means that since prior data is saved in each file automatically, it is unnecessary to repeat the same data input if it remains unchanged. Some flexibility was contemplated to input data in the file PLATE.DAT for the calculation of uranium density and void volume fraction in miniplate.

Name of sample and its detailed information should be addressed for the prompt message entitled "SAMPLE DESCRIPTION" and are temporarily stored in SAMPLDSP.DAT; sample name is read within 8 characters including the space from the first column of the screen . In this case, onscreen editing can be made within 255 characters (N.B.: one 2-bytes Japanese word is equivalent to 2 characters in this case).

The subroutine DATASTORE stores a set of necessary data into the other 3 files (_NAME.UPL, _UDT.UPL, _VOID.UPL) to transfer to the subprogram the calculated values of uranium density, void volume fraction, volume and weight of sample (miniplate), fuel volume fraction and total weight of uranium in the fuel core as well as sample name. Duplication of the sample name is not permitted and checked before the storage is made. Desired data obtained from density calculation can be saved in the last file _UDVFILE.UPL.
At the end of the main program, there is a question whether to continue a series of density/volume calculation, quit the job, to print out the data same as those displayed on the CRT or to tabulate principal data obtained. The last option leads to call the subprogram UPL.EXE.

It is worth noting that the current program can treat 5 types of fuels :

100 wt% U_3Si_2

80 wt% U_3Si_2 + 20 wt% U_3Si

80 wt% U_3Si_2 + 20 wt% USi

99.5 wt% U_3Si_2 + 0.5 wt% Mo

100 wt% U_6Me (Me = Fe, Mn, Ni)

3.1.1 Calculating procedure of density and volume of sample

Density of uranium silicides with variable compositions and volume of miniplate are measured by the Archimedes method i.e. the immersion method using meta-xylene as a replacement liquid[6-8].

Variables to be input from the keyboard are :

WS : Weight of sample in air(g)

WBL : Weight of bucket in replacement liquid(g)

WSBL : Weight of sample & bucket in replacement liquid(g)

WSL : Weight of sample in liquid(g) = WSBL - WBL

ATEMP: Temperature of air(°C)

TEMP : Temperature of replacement liquid (°C)

Then,

$$DS = \frac{WS \cdot DL - WSL \cdot DA}{WS - WSL}$$

$$VS = \frac{WS}{DS - DA}$$

where

DS : Density of sample(g/cm^3)

VS : Volume of sample(cm^3)

DL : Density of replacement liquid(g/cm^3)

DA : Density of air(g/cm^3)[9].

$$DA = \frac{0.001293}{1.000 + 0.00367 \cdot ATEMP}$$

A value of DL is read from an array DL(I) for a given temperature (TEMP) of the replacement liquid.

3.1.2 Calculating procedure of uranium density and void volume fraction in the fuel core of miniplate

Variables to be input from the keyboard are :

WCOMPACT: Weight of fuel-meat compact(g)

= WFUEL + WALUM

WFUEL: Weight of fuel powder(g)

WALUM: Weight of aluminum powder(g)

VP: Volume of fuel plate(cm^3)

The value of VP is normally taken equal to that of VS determined in 3.1.1 and not necessarily input from the keyboard.

$$VMTH = \frac{WFUEL}{DFUEL} + \frac{WALUM}{DALUM}$$

VMTH: Calculated theoretical volume of fuel meat(cm^3)

DFUEL: Density of fuel powder(g/cm^3)

DALUM: Density of aluminum powder= 2.700 (g/cm^3)

Each value of DFUEL may be provided from the subroutine : DU3SI2, DU3SI2USI, DU3SI2U3SI, DU3SI20.5MO or DU6ME depending on the fuel type chosen.

$$VM = VP - \frac{WFC}{DFR}$$

VM: Volume of rolled fuel meat(cm^3)

VP: Volume of miniplate(cm^3)

DFR: Density of frame and cover = 2.710 (g/cm^3) for Al 6061

WFC: Weight of frame and cover(g) = WS - WCOMPACT

And hence,

$$V = \frac{100.0 \cdot (VM - VMTH)}{VM}$$

where V = Void volume fraction in miniplate(%).

Fuel volume fraction is defined throughout the report as follows:

$$FVF = \frac{(100.0 - V)VF}{VF + VA}$$

FVF: Fuel volume fraction(%)

VF : Fuel volume in fuel meat(cm^3)

VA : Aluminum volume in fuel meat(cm^3)

$$UD = \frac{WFUEL \cdot WFU}{VM}$$

$$UW = WFUEL \cdot WFU$$

UD : Uranium density(g/cm^3)

UW : Total uranium weight in fuel core(g)

WFU: Weight fraction of uranium in fuel compound

Parameters such as DS, ATEMP, VMTH and DFR given above have same variable and constant names in the program.

3.2 The subprogram UPL.BAS

The subprogram offers functions of storage of necessary data on disk , data tabulation both for the CRT and for the printer, data insertion/deletion and data search/amendment.
It includes total 12 random files of which two (AMNAME.DAT and INSREC.DAT) have only one record. The former serves as a temporal memory of sample name of which data were previously amended, while the latter stores data of a record inserted before. These prior name and data can be referred ,respectively, for the data correction and for the insertion of new record. The file _DATFILE.UPL is created from data in other 3 files : _NAME.UPL, _UDT.UPL, _VOID.UPL ; filenames are common to those in the program UDV.EXE so as to access data from UPL.EXE directly.

Before the use of UPL.EXE, at least one record should be saved in .UPL files through the start-up of the UDV.EXE.

Actual parameters stored in the file _DATFILE.UPL are as follows ([ ]= input variable names) :

sample (miniplate) name [AP$], weight of miniplate [WS] , volume of miniplate [VP], total weight of uranium contained [UW], uranium density [UD], void volume fraction [V] and fuel volume fraction [FVF] in the fuel meat of a miniplate.

With the program being booted, these data appear in the table on the first screen entitled : ALUMINUM DISPERSION PLATE-TYPE FUEL DATA.

Upon hitting HELP-key at any time, a list of sample names is shown on the CRT and is scrolled by 20 names ; scroll-width ROL can be altered on the program basis. Desired sample name is chosen by using arrow keys to move the highlighting and pressing return key , and then data can be amended as necessary. To abandon data correction, push F・1, F・2 or F・3 key. The subroutines DATAAMEND and NAMESEARCH offer all these functions.

The subroutine XOPTION provides a function of selecting an option from seven ones : to continue or quit the job, to search/amend data, to print out table data, to insert new record number and data, to delete data including record itself or to calculate density and volume of samples. The last option leads to call the main program UDV.EXE automatically.

Thus, new data input may be allowed through the subroutines DATAINSERT and DTMANIP, if data correction or data insertion is anticipated. To insert a new record, a specified record number should be input at first , and then sample name, weight and volume of miniplate, total uranium weight, uranium density, void volume fraction and fuel volume fraction , using a comma between each input. Necessary messages for operation are given onscreen as in the case of the main program. Maximum allowable record number is now set at 80 as AP2$(80) , but it can be extended to the order of 100.

The batch file F_DELETE.BAT built in the subroutine DATAINSERT uses internal DOS commands, COPY and DEL, which makes it possible to remove any record once stored in the files _NAME.UPL, _UDT.UPL, _VOID.UPL, and hence in the file _DATFILE.UPL.

If the program execution is interrupted due to the 'Out of memory ' error during the repeated removal of records, it is recommended to reboot UPL.EXE to ensure the deletion of the records.

As mentioned above, the subprogram UPL.EXE offers the capability of manipulation of records as well as data which have been created through the execution of the main program UDV.EXE.

4. Examples of output data

Fig. 5 and output data list (Table 1) exemplify the results of the execution of UDV.EXE and UPL.EXE, respectively.

Fig. 6 shows a typical example of a metallographic cross-section of a miniplate (same sample as 1PE4B shown in Table 1). It was prepared by the conventional picture-frame method [10] for a purpose of capsule irradiation in JMTR.

Its fuel meat consists of uranium silicide particles with nominal composition U_3Si_2 dispersed in an aluminum matrix.

Void (as-fabricated porosity) appears dark in these photographs.

Within the estimated experimental error of ca. ±10 %, the calculated void volume fraction ,13.1 %, seems to correlate well with the real void distribution in the fuel core.

On the other hand, the calculated uranium density for the miniplate is 4.69 g/cm^3.

5. Conclusion

BASIC program simple and easy to operate has been developed to compute uranium density and void volume fraction for laboratory-scale uranium silicide aluminum dispersion plate-type fuel , so called miniplate. The program is also able to constitute data base for miniplates from experimentally- determined values of density and weight of each constituent and dimensions of miniplates.

Examples of output are shown including a comparison between calculated void volume fraction and actual void distribution in a miniplate.

Utility programs pertinent to the development of the BASIC program are also given which run in the MS-DOS environment.

Acknowledgements

The present author gratefully acknowledges Mr. K. Shiba and other staffs who have provided useful information regarding the software to accomplish this work. He is also grateful to Dr. T. Kondo and Dr. M. Handa ,respectively, Director and Deputy Director of the Department of Fuels and Materials Research for their encouragement.

References

(1) U. S. Nuclear Regulatory Commission, NUREG-1313, "Safety evaluation report related to the evaluation of low-enriched uranium silicide-aluminum dispersion fuel for use in non-power reactors", July, 1988

(2) Y. Futamura et al.,"Status of reduced enrichment program for research and test reactor fuels in Japan" in Proceedings of the 12th international meeting on reduced enrichment for research and test reactors, Berlin, F.R. Germany, Sept. 10-14,1989

(3) NEC N88-BASIC(86) Compiler (MS-DOS Version) 6.0, User's Manual (1988) ; NEC N88-BASIC(86) Ver 6.1 (1989) NEC Corporation

(4) Pro FORTRAN-77 Compiler User's Manual (1988), LIFEBOAT Co., Ltd., Prospero Software Ltd.

(5) Microsoft C Optimizing Compiler Version 5.1, Run-Time Library Reference 2 (1989), Microsoft Corp.

Acknowledgements

The present author gratefully acknowledges Mr. K. Shiba and other staffs who have provided useful information regarding the software to accomplish this work. He is also grateful to Dr. T. Kondo and Dr. M. Handa ,respectively, Director and Deputy Director of the Department of Fuels and Materials Research for their encouragement.

References

(1) U. S. Nuclear Regulatory Commission, NUREG-1313, "Safety evaluation report related to the evaluation of low-enriched uranium silicide-aluminum dispersion fuel for use in non-power reactors", July, 1988

(2) Y. Futamura et al.,"Status of reduced enrichment program for research and test reactor fuels in Japan" in Proceedings of the 12th international meeting on reduced enrichment for research and test reactors, Berlin, F.R. Germany, Sept. 10-14,1989

(3) NEC N88-BASIC(86) Compiler (MS-DOS Version) 6.0, User's Manual (1988) ; NEC N88-BASIC(86) Ver 6.1 (1989) NEC Corporation

(4) Pro FORTRAN-77 Compiler User's Manual (1988), LIFEBOAT Co., Ltd., Prospero Software Ltd.

(5) Microsoft C Optimizing Compiler Version 5.1, Run-Time Library Reference 2 (1989), Microsoft Corp.

(6) K. Kanaya, H. Yamashita, N. Ooka, I. Tanaka
Private communication (March,1987)

(7) N.Ooka, K.Kanaya, Y.Etou, H.Yamashita, "Experiments on compatibility of U_3Si_2 and Aluminum at high temperature in out-pile", Keikinzoku yousetu 339,28(1990)8

(8) Japanese Industrial Standards Committee: Japanese Industrial Standard, "Measuring methods for specific gravity of solid (JIS Z 8807-1976)" , Japanese Standards Association

(9) Rika nenpyo (Chronological Scientific Tables 1990), ed. National Astronomical Observatory, Maruzen Co., Ltd.

(10) M.Ugajin, A. Itoh, M. Akabori, K.Ikawa,"Preparation of silicide fuel for capsule irradiation": presented in the 1989 fall meeting of the Atomic Energy Society of Japan, Oct.17-19,1989 (Tokai-mura, Japan) Abstract No. K57

Table 1 Output Example of UPL. EXE

ALUMINUM DISPERSION PLATE-TYPE FUEL DATA

試料名 SAMPLE	燃料板重量 PLATE WT. (g)	燃料板体積 PLATE VOL. (cc)	総ウラン量 TOTAL U WT. (g)	ウラン密度 U DENSITY (gU/cc)	ボイド率 VOID % (%)	燃料体積率 FUEL VOL.% (%)
1PE4B	2.3059	0.8109	0.1422	4.69	13.10	41.45
88F2A-01	2.2508	0.7907	0.1348	4.38	5.94	38.76
88F2A-02	2.2748	0.8023	0.1247	4.37	5.28	39.83
88F2A-03	2.2447	0.7941	0.1125	4.66	2.82	41.67
88F2A-04	2.2669	0.7989	0.1279	4.66	7.78	39.25
89F1A-01	2.2204	0.7808	0.1408	4.07	14.37	36.00
89F1A-02	2.2518	0.7917	0.1389	4.03	11.03	36.79
89F1A-03	2.2223	0.7789	0.1454	4.15	11.42	37.17
89F1A-04	2.2641	0.7930	0.1515	4.39	13.26	37.03
89F1A-05	2.2784	0.7767	0.2105	6.31	12.84	36.79
89F1A-06	2.3152	0.7929	0.2057	5.98	15.59	34.88

```
100 '  ////////////////////////////////////////////////////
110 '                        A1. BAS
120 '  ////////////////////////////////////////////////////
130 '
140 ' A1.EXE, N8.BAT AND QX.BAT CONSTITUTE A SET OF PROGRAM TO COMPILE, EDIT
150 ' AND RECOMPILE A BASIC SOURCE LIST, EVENTUALLY CREATING EXECUTABLE FILE.
160 ' FILENAME CAN BE GIVEN AS A COMMAND LINE PARAMETER WHEN BOOTING N8.BAT.
170 '
180 '  A1.EXE SHOULD BE BOOTED BY N8.BAT IN ORDER TO INTERPRET COMPILER'S DI-
190 ' AGNOSTIC MESSAGE AND, ON THE BASIS OF THE INTERPRETATION, DECIDES WHETHER
200 ' OR NOT TO BOOT  OTHER BATCH FILE QX.BAT WHICH IS ABLE TO RUN EDITOR &
210 ' COMPILER.
220 '
230   CONSOLE ,,,1
240 '
250   OPEN      "FNAM"     AS #1
260   OPEN      "CLMS"     AS #2
270   IF EOF(1) THEN CLOSE #1 : GOTO  320
280   LINE INPUT #1,  FLNAME$
290 '
300   GOTO  270
310 '              59000 : THE VALUE DEPENDS ON THE SYSTEM ENVIRONMENT ADOPTED.
320   IF FRE(3) < 59000  THEN COLOR 3:PRINT "          [OUT OF MEMORY ]  RE-BOOT
 :  N8(.BAT) " ; FLNAME$ :COLOR 7 : CLOSE: STOP: END
330 '
340   IN$ = INKEY$
350       WHILE NOT EOF(2)
360   LINE INPUT #2,  N$
370 '
380   IF N$="    0000  FATAL ERRORS" OR N$="*** NO ERROR ***"
         THEN  GOTO  390  ELSE GOTO  470
390   FLNAMEX$ = MID$( FLNAME$, 2, LEN(FLNAME$)-3 )  : PRINT : PRINT
400   PRINT " EXECUTABLE  FILE  ";:COLOR 3:PRINT  FLNAMEX$+".EXE"; : COLOR  7 :
PRINT "  CREATED.          RUN   ? ( Y/N ) "
410       WHILE  IN$ <> "Y" OR  IN$ <> "y"
420 '
430   IF  IN$ = "Y" OR IN$= "y"    THEN  RUN   FLNAME$
440   IF  IN$ = "N" OR IN$= "n"    THEN  STOP : END
450        IN$ = INKEY$
460       WEND
470   IF  INSTR( N$, "UNSUCCESSFULLY" ) <> 0  THEN  RUN "QX"+FLNAME$
480 '
490   IF KINSTR( N$, "ソースファイルが正しくない") <> 0 THEN RUN "QX" + FLNAME$
500 '   "ソースファイルが正しくない"  =  "SOURCE  FILE  INVALID"
510       WEND
520   STOP : END
530 '
540 '/////////////  END  OF  FILE  ///////////////
```

Fig. 1 Source Lists of A1. BAS, N8. BAT and QX. BAT

```
ECHO  £>1h  £2K
IF "%1== " GOTO END                                ※  £ indicates Escape Sequence.
CLS
ECHO OFF
CLS
ECHO   %1  >  FNAM
CLS
ECHO         £22;36m    NOW  UNDER  COMPILATION  of  £19;35m  %1.BAS   £m
BASICC %1; > CLMS
TYPE  CLMS|MORE
ECHO     £7;43m  N8.BAT :  HIT  ANY  KEY  TO  CONTINUE.   £m
PAUSE > NUL
A1
:END
CLS
ECHO  £2K £41mWARNING £4;22;36m [USAGE]  DRIVENAME:¥> N8  FILENAME  Without Exte
nsion £m
```

```
ECHO  £2K  £>1h
ECHO OFF
CLS
ECHO      £22;36m AMEND  ERRORS  in   £19;35m  %1.BAS  £22;36m  USING  SCREEN  E
DITOR  £m
MIFES  -N -K -R %1.BAS
ECHO ON
CLS
ECHO  £2K  £>1h
ECHO OFF
CLS
ECHO   £22;36m    RE-COMPILE  OF   £19;35m  %1.BAS  £m
BASICC  %1    ; >  CLMS
TYPE  CLMS|MORE
ECHO        £5;43m  QX. BAT  :  HIT  ANY  KEY  TO  CONTINUE.   £m
PAUSE > NUL
A1
```

Fig.1 (Continued)

```
C ////////////////////////////////////////////////////////////
C                          RL80. FOR
C ////////////////////////////////////////////////////////////
C
C
      PROGRAM    RLCONVERTER
C
C     FORTRAN PROGRAM TO CONVERT TEXT FILE WITH VARIABLE-LENGTH RECORD
C     (MAX. 255 BYTES PER RECORD) TO FIXED-LENGTH FILE WITH 80 BYTES PER
C     RECORD ; EXCLUSIVE USE FOR THE BASIC SOURCE LIST CONVERSION.
C
C  [EXAMPLE OF USAGE]          A:¥>RL80 B:
C  B: OPTION OF DRIVENAME  FOR TEMPORAL
C  FILE GENERATION.
C
C  CURRENT DRIVE AND CURRENT DIRECTORY ARE ASSIGNED FOR THE FILES CHOSEN
C  UNLESS DRIVE & PATH NAMES ARE SPECIFIED.
C
C
      INCLUDE 'SYSREG'
      CHARACTER DRNAME*12, XTMP*60, DRNN*50,XNIN*45,XNOUT*45,XDRNAME*45
      CHARACTER*20  INPNAM, OUTNAM, NIN, NOUT
      CHARACTER*1  SPC, LF
      INTEGER*4 IONUM,IONUM1,IONUM2,IWRITE,TEMP,XCUR
      CHARACTER*1 CARD,CARD2
      DIMENSION CARD(255) !MAXIMUM CHAR NUMBER OF ONE LOGICAL LINE IN BASIC
      DIMENSION CARD2(80)
      DIMENSION SPC(65), LF(80)
      LOGICAL   AFFIRM
      COMMON /DRVID/ NIN, NOUT, DRNAME, XNIN, XNOUT, XDRNAME
C
      DATA SPC /65*1H /
      DATA LF  /80*1H /
      WRITE(*,*) CHAR(27),CHAR(91),CHAR(62),CHAR(49),CHAR(104)
       OPEN ( 8, FILE='XCURDIR',IOSTAT=XCUR, ACCESS='SEQUENTIAL',
     * STATUS='OLD' )
      IF (XCUR .EQ. 0)          CALL   CMND( 'DEL '//'XCURDIR')
                                CALL   CHOME
                                CALL   VIOLET
      WRITE(*,100)
  100 FORMAT(3X,'BASIC可変長ファイル -->>固定長ファイル (RL=80
     1) 変換ユテリティー      ')
      WRITE(*,150)
  150 FORMAT(3X,'RECORD-LENGTH CONVERTER: GIVE INPUT & OUTPUT FILENAMES
     1(EACH WITHIN 20 CHARS) ')
                                CALL  GREEN
      WRITE(*,200)
  200 FORMAT(3X,
     1'入力ファイル名、出力ファイル名 (各20文字以内) インプット:
     2QUIT=CTRL+C',/)
      CALL  CURRENT1
                                CALL  WHITE
      WRITE(*, 10)
   10 FORMAT( 1X,'1---input file----201---output file---20',/)
  400 READ (*, 20 , ERR = 888 ) INPNAM,  OUTNAM
   20 FORMAT( 2A20)
```

Fig. 2 Source Lists of RL80. FOR and CURDIR. C

```
      IF ( AFFIRM ('  この  入力  で宜しいですか。  SURE  (INP'
     *//'UT  Y OR N ) ')) THEN
            GOTO  5656
      ENDIF
            CALL   CADRES
            GOTO   400
5656  OPEN ( 5, FILE= INPNAM ,ACCESS='SEQUENTIAL',
     * STATUS='OLD', ERR = 888 )
                                CALL  GETCOM(DRNN)
       L =LENGTH(DRNN)
       XTMP=DRNN(1:L)//'¥'//'XTMP'
      OPEN (10, FILE=XTMP,IOSTAT=TEMP,ACCESS='SEQUENTIAL',
     * STATUS='OLD', ERR = 11 )
      OPEN (20, FILE=   OUTNAM ,ACCESS='SEQUENTIAL',
     * STATUS='NEW', ERR = 888 )
      INQUIRE(5, NAME = NIN)
      INQUIRE(20, NAME= NOUT)
      INQUIRE(10, NAME = DRNAME)
        CALL  CURRENT2
        CALL  DRSPACE1
        CALL  DRSPACE2
      IF ( NIN .EQ. NOUT )  WRITE (*,88)
      IF ( NIN .EQ. NOUT )  GO TO 3000
1000  READ (5,5001,IOSTAT= IONUM,  END=1001 ) CARD
      IF ( IONUM .GE. 10 )  GO  TO  888
5001  FORMAT( 255A1)
      WRITE (10,5002,IOSTAT=IWRITE,ERR=12) (CARD(J),J=  1,80 )
      WRITE (10,5002,IOSTAT=IWRITE,ERR=12) (CARD(J),J= 81,160)
      WRITE (10,5002,IOSTAT=IWRITE,ERR=12) (CARD(J),J=161,240)
      WRITE (10,5002,IOSTAT=IWRITE,ERR=12) (CARD(J),J=241,255),
     *                                     (SPC(L),L=1,65)
5002  FORMAT(80A1)
      GO TO 1000
1001  CONTINUE
      CALL  BLUE
      WRITE (*,*) '    TEMPORAL(INTERMEDIATE) FILE GENERATED.'
             REWIND  10
      K = 0
1111  READ (10,5002,IOSTAT= IONUM,  END=9009 ) CARD2
      IL = 0
      DO  4477 JJ=1, 80
      IF ( CARD2(JJ)  .EQ. CHAR(32) )
     *  IL = IL + 1
      IF ( IL .EQ. 80 ) GO TO  1111
4477  CONTINUE
      WRITE (20,5002,IOSTAT=IWRITE,ERR=11)  (CARD2(J),J=1,80)
      IF ( IWRITE .EQ. 59 ) GO TO 11
      K = K + 1
      GOTO 1111
9009  CONTINUE
      CLOSE ( 10, STATUS = 'DELETE' )
                          CALL   GREEN
       WRITE (*,*) LF,'    ', XNIN(1:LENGTH(XNIN)),'  SUCCESSFULLY'
     * //' CONVERTED TO  ', XNOUT(1:LENGTH(XNOUT))
       WRITE (*, 99) K
  99  FORMAT(/,'     TOTAL  RECORDS  CONVERTED  =  ', I4)
      IFLAG=1
      GO TO  2001
 888                      CALL   BLUE
      INQUIRE ( 5, NAME=NIN)
      INQUIRE (20, NAME=NOUT)
      WRITE (*, 88)
```

Fig.2 (Continued)

```
   88 FORMAT( //,4X,'CANNOT OPEN OR OVER-WRITE THE FILE : BOOST < RL80 >
     * AND  TRY ONCE AGAIN !! ' )
      CLOSE (20)
      OPEN (20,FILE=OUTNAM,STATUS='OLD', ERR=2323)
      READ ( 20, *, IOSTAT= IONUM1)
      IF (IONUM1 .NE. 0)  CLOSE ( 20, STATUS = 'DELETE')
      GOTO  2001
   11                      CALL  VIOLET
      IF (IWRITE .EQ. 1003) GOTO  3003
      INQUIRE (20,NAME=NOUT)
      CALL CMND ( 'DEL '//  NOUT )
      WRITE(*,3355)
 3355 FORMAT ( //,4X, 'DISK  FULL  !! (OUTPUT OPERATION FAILED)   ')
      GO TO 2001
   12                      CALL  VIOLET
      CALL CMND( 'DEL '//NOUT)
      IF (IWRITE .EQ. 1003) GOTO  3003
      WRITE(6,3366, ERR=7373)
 3366 FORMAT ( //,4X, 'DISK FULL FOR OUTPUT OF TEMPORAL FILE   ')
 2001                    CALL  WHITE
      IF (IFLAG .NE. 1)  CLOSE ( 10, STATUS = 'DELETE', ERR=1234 )
 3003  IF (IWRITE .EQ. 1003)  WRITE(*,*) '    INVALID DRIVENAME FOR TEMP
     *ORAL FILE  !!    '
 3000        CONTINUE
      IF ( NIN .EQ. NOUT )  WRITE (*,*) '   ( OUTPUT FILENAME INVALID :
     * SAME INPUT-FILENAME DESCRIPTION )  '
                              CALL  WHITE
 6363 IF ( NIN .EQ. NOUT )  OPEN (20,FILE=OUTNAM,STATUS='OLD',ERR=2323)
 2323 IF ( NIN .EQ. NOUT )  READ ( 20, *, IOSTAT= IONUM2)
      IF (IONUM2 .LT. 0)  CLOSE ( 20, STATUS = 'DELETE')
      GO TO 7373
 1234 CALL CMND ('DEL '//DRNAME)
      GOTO  3003
 7373 CONTINUE
      END
C
C
      SUBROUTINE  GREEN
      WRITE (*,*)  CHAR(27), CHAR(91), CHAR(50), CHAR(48),CHAR(59),
     1CHAR(51), CHAR(50),CHAR(109),        CHAR(27),CHAR(77)
      RETURN
      END
      SUBROUTINE  VIOLET
      WRITE (*,*)  CHAR(27), CHAR(91), CHAR(52), CHAR(53),
     1CHAR(109), CHAR(27),CHAR(77)
      RETURN
      END
      SUBROUTINE  WHITE
      WRITE (*,*)  CHAR(27), CHAR(91),  CHAR(109),   CHAR(27),  CHAR(77)
      RETURN
      END
      SUBROUTINE  CHOME
      WRITE(*,*) CHAR(27),CHAR(91),CHAR(50),CHAR(74)
      RETURN
      END
      SUBROUTINE  CADRES
      WRITE(*,999) CHAR(27),CHAR(91),CHAR(57),CHAR(59),CHAR(49),CHAR(72)
     1,        CHAR(27), CHAR(91), CHAR(50),CHAR(75)
  999 FORMAT( ' ', 10A1)
      WRITE(*,900) CHAR(27),CHAR(91),CHAR(55),CHAR(59),CHAR(49),CHAR(72)
  900 FORMAT( ' ', 6A1,'1---5---10---15---201---5---10---15---20',/)
      WRITE(*,9898) CHAR(27),CHAR(91),CHAR(56),CHAR(59),CHAR(49),CHAR(72
```

Fig.2 (Continued)

```
     1),CHAR(27), CHAR(91),CHAR(50),CHAR(75)
 9898 FORMAT( ' ',10A1)
      RETURN
      END
      SUBROUTINE  BLUE
      WRITE (*,*)  CHAR(27), CHAR(91), CHAR(50), CHAR(50),CHAR(59),
     1CHAR(51), CHAR(54),CHAR(109),       CHAR(27),CHAR(77)
      RETURN
      END
      SUBROUTINE  RYEL
      WRITE (*,*)  CHAR(27), CHAR(91), CHAR(52), CHAR(51),
     1CHAR(109), CHAR(27),CHAR(77)
      RETURN
      END
C
C
                    SUBROUTINE CURRENT1
C
C     SUBROUTINES 1 & 2 ARE TO GET FULL PATH NAME FOR INPUT FILE
C     AND OUTPUT FILE, USING CURDIR. EXE.
C
C     CURDIR. EXE WRITTEN BY THE C LANGUAGE IS TO READ
C     CURRENT DRIVE & DIRECTORY NAME AND WRITE IT TO THE FILE  "XCURDIR."
C
C
       INTEGER*4 IRET
       CHARACTER*30  XCURDIR
C
       CALL  RYEL
       CALL EXECPG ( ' CURDIR  XCURDIR ' , IRET )
       READ ( 8,3300,ERR = 5757 ) XCURDIR
       WRITE(*,*)
     * ' CURRENT DIRECTORY =  ', XCURDIR
       GOTO  2233
 5757  WRITE(*,*)
     *  ' CURDIR.EXE TO READ CURRENT DIRECTORY NAME IS NOT AVAILABLE.'
       CALL WHITE
       STOP
 3300  FORMAT(30A)
 2233  RETURN
       END
C
C
                    SUBROUTINE CURRENT2
C
C
      CHARACTER  XCURDIR*30, IFN*35, OFN*35,TFN*35
      CHARACTER  XNIN*45, XNOUT*45, XDRNAME*45, DRNN*50
      CHARACTER  NIN*20, NOUT*20, DRNAME*12
      COMMON /DRVID/ NIN, NOUT, DRNAME, XNIN, XNOUT, XDRNAME
C
      IFN= '  INPUT    FILE (入力ファイル名) = '
      TFN= '  TEMPORAL FILE (中間ファイル名) = '
      OFN= '  OUTPUT   FILE (出力ファイル名) = '
                    CALL  VIOLET
       WRITE(*,*) CHAR(27),CHAR(91),CHAR(66)
       REWIND  8
       READ ( 8,3300,ERR = 5757 ) XCURDIR
 3300  FORMAT(30A)
       L=LENGTH(XCURDIR)
       LIN=LENGTH(NIN)
       IF (INDEX(NIN,CHAR(58)//CHAR(92)) .NE. 0 )  THEN
```

Fig.2 (Continued)

```
      XNIN=NIN
      WRITE (*,*) IFN, XNIN
      GOTO 8800
      ENDIF
      IF (NIN(1:1) .EQ. CHAR(92))  THEN
      XNIN=XCURDIR(1:2)//NIN
      WRITE (*,*) IFN, XNIN
      GOTO 8800
      ENDIF
      IF (INDEX(NIN, CHAR(58)) .EQ. 0  .AND. L .EQ. 3) THEN
      XNIN=XCURDIR(1:L)//NIN
      WRITE (*,*) IFN, XNIN
      ENDIF
      IF (INDEX(NIN, CHAR(58)) .EQ. 0  .AND. L .GT. 3)  THEN
      XNIN=XCURDIR(1:L)//'¥'//NIN
      WRITE (*,*) IFN, XNIN
      GOTO 8800
      ENDIF
      IF (NIN(1:1) .NE. XCURDIR(1:1)) GOTO  3232
      IF (INDEX(NIN,CHAR(58)).NE.0 .AND. L .LT. 4 .AND.
     * INDEX(NIN,CHAR(92)).EQ. 0) THEN
      GOTO   8282
      ENDIF
      IF (INDEX(NIN,CHAR(58)).NE.0 .AND.
     * INDEX(NIN,CHAR(92)).EQ. 0) THEN
      XNIN=XCURDIR(1:L)//'¥'//NIN(3:LIN)
      WRITE (*,*) IFN, XNIN
      GOTO 8800
     ELSE IF (INDEX(NIN,CHAR(58)).NE.0 .AND.
     *            NIN(3:3) .NE. CHAR(92) ) THEN
      XNIN=XCURDIR(1:L)//'¥'//NIN(3:LIN)
      ENDIF
      IF (XNIN(3:4) .EQ. '¥¥')  THEN
      XNIN=XNIN(1:3)//XNIN(5:LENGTH(XNIN))
      WRITE (*,*) IFN, XNIN
     ELSE
      WRITE (*,*) IFN, XNIN
      ENDIF
      GOTO  5252
 8282       XNIN=XCURDIR(1:L)//NIN(3:LIN)
     WRITE (*,*) IFN, XNIN
 5252   CONTINUE
     GOTO   8800
 3232   IF (INDEX(NIN, CHAR(58)) .NE. 0  .AND. NIN(3:3) .NE. '¥') THEN
     XNIN=NIN(1:2)//'¥'//NIN(3:LIN)
     WRITE (*,*) IFN, XNIN
     ENDIF
C  ---------------------------------------------------------------
 8800       IF ( INDEX(DRNAME,CHAR(58)) .EQ. 0) THEN
      XDRNAME=XCURDIR(1:2)//DRNAME
      WRITE (*,*) TFN, XDRNAME
      ELSE
      XDRNAME=DRNAME
      WRITE (*,*) TFN, XDRNAME
      ENDIF
C  ---------------------------------------------------------------
      LOUT=LENGTH(NOUT)
      IF (INDEX(NOUT,CHAR(58)//CHAR(92)) .NE. 0 )  THEN
      XNOUT=NOUT
      WRITE (*,*) OFN, XNOUT
      RETURN
      ENDIF
```

Fig.2 (Continued)

```
      IF (NOUT(1:1) .EQ. CHAR(92))  THEN
      XNOUT=XCURDIR(1:2)//NOUT
      WRITE (*,*) OFN, XNOUT
      RETURN
      ENDIF
      IF (INDEX(NOUT, CHAR(58)) .EQ. 0  .AND. L .EQ. 3) THEN
      XNOUT=XCURDIR(1:L)//NOUT
      WRITE (*,*) OFN, XNOUT
      ENDIF
      IF (INDEX(NOUT, CHAR(58)) .EQ. 0  .AND. L .GT. 3) THEN
      XNOUT=XCURDIR(1:L)//'¥'//NOUT
      WRITE (*,*) OFN, XNOUT
      RETURN
      ENDIF
      IF (NOUT(1:1) .NE. XCURDIR(1:1)) GOTO  7272
      IF (INDEX(NOUT,CHAR(58)).NE.0 .AND. L .LT. 4 .AND.
     * INDEX(NOUT,CHAR(92)).EQ. 0) THEN
C
C
      GOTO   9292
      ENDIF
      IF (INDEX(NOUT,CHAR(58)).NE.0 .AND.
     * INDEX(NOUT,CHAR(92)).EQ. 0) THEN
      XNOUT=XCURDIR(1:L)//'¥'//NOUT(3:LOUT)
      WRITE (*,*) OFN, XNOUT
      RETURN
     ELSE IF (INDEX(NOUT,CHAR(58)).NE.0 .AND.
     *           NOUT(3:3) .NE. CHAR(92) ) THEN
      XNOUT=XCURDIR(1:L)//'¥'//NOUT(3:LOUT)
      ENDIF
      IF (XNOUT(3:4) .EQ. '¥¥')  THEN
      XNOUT=XNOUT(1:3)//XNOUT(5:LENGTH(XNOUT))
      WRITE (*,*) OFN, XNOUT
     ELSE
      WRITE (*,*) OFN, XNOUT
      ENDIF
      GOTO  5353
 9292       XNOUT=XCURDIR(1:L)//NOUT(3:LOUT)
       WRITE (*,*) OFN, XNOUT
 5353  CONTINUE
       RETURN
 7272  IF (INDEX(NOUT, CHAR(58)) .NE. 0  .AND. NOUT(3:3) .NE. '¥') THEN
     XNOUT=NOUT(1:2)//'¥'//NOUT(3:LOUT)
     WRITE (*,*) OFN, XNOUT
     ENDIF
     GOTO  7777
 5757 CONTINUE
      WRITE(*,*)
     *  ' CURDIR.EXE TO READ CURRENT DIRECTORY NAME IS NOT AVAILABLE.'
      CLOSE (10,STATUS='DELETE')
      CLOSE (20,STATUS='DELETE')
      STOP
 7777 CONTINUE
      IF (XNOUT(3:4) .EQ. '¥¥') THEN
      XNOUT=XNOUT(1:3)//XNOUT(5:LENGTH(XNOUT))
      WRITE(*,*) OFN, XNOUT
      ENDIF
      RETURN
      END
C
C
                  SUBROUTINE  DRSPACE1
```

Fig.2 (Continued)

```
C
C
C     SUBROUTINES DRSPACE 1 & 2 ARE TO GET INFORMATION ON THE FREE SPACE
C     ON THE DISKS FOR OUTPUT FILE AND TEMPORAL FILE, RESPECTIVELY, USING
C     DOS FUNCTION CALL.
C
C
      INCLUDE 'SYSREG'
      INTEGER*4  FRESPC
      CHARACTER  DRNN*50,XNIN*45,XNOUT*45,XDRNAME*45
      CHARACTER  NIN*20, NOUT*20, DRNAME*12
      COMMON /DRVID/ NIN, NOUT, DRNAME, XNIN, XNOUT, XDRNAME
C
      CALL  WHITE
      AH = $36
      DL = ICHAR(XNOUT(1:1))-64
      CALL  SYS1(SYSREG)
      FRESPC = AX*BX*CX
      WRITE(*,*)' ',XNOUT(1:2),'(DRIVE FOR OUTPUT FILE)   = ',
     *   FRESPC, '  bytes free'
      IF ( FRESPC .EQ. 0) THEN
      CLOSE (20,STATUS='DELETE')
     STOP ': NO FREE SPACE FOR OUTPUT FILE.'
      ENDIF
      RETURN
      END
C
C
                    SUBROUTINE  DRSPACE2
C
C
      INCLUDE 'SYSREG'
      INTEGER*4  FRESPC
      CHARACTER  DRNN*50,XNIN*45,XNOUT*45,XDRNAME*45
      CHARACTER  NIN*20, NOUT*20, DRNAME*12
      COMMON /DRVID/ NIN, NOUT, DRNAME, XNIN, XNOUT, XDRNAME
C
      CALL  WHITE
      AH = $36
      DL = ICHAR(XDRNAME(1:1))-64
      CALL  SYS1(SYSREG)
      IF (AX .EQ. -1)  WRITE(*,*) '    INVALID DRIVENAME FOR TEMP'//
     *'ORAL FILE  !!   '
      FRESPC = AX*BX*CX
      WRITE(*,*)' ', XDRNAME(1:2),'(DRIVE FOR TEMPORAL FILE) = ',
     *   FRESPC, '  bytes free'
      IF ( FRESPC .EQ. 0) THEN
      CLOSE (20,STATUS='DELETE')
      CLOSE (10,STATUS='DELETE')
     STOP': NO FREE SPACE FOR TEMPORAL FILE.'
      ENDIF
      RETURN
      END
C
C ////////////////// END OF FILE //////////////////
```

Fig.2 (Continued)

```
/*              *  *    C U R D I R . C    *  *
*
*  C U R D I R . E X E IS TO READ CURRENT DRIVE & DIRECTORY NAME
*  AND WRITE IT TO ANY FILE WHICH MUST BE INDICATED IN THE COMMAND
*  LINE PARAMETER FOR AVOIDING RUN-TIME ERROR : NULL POINTER ASSIGNMENT.
*/
#include <dos.h>
#include <stdio.h>
#include <direct.h>
#include <process.h>
   char buffer [31] ;     /* Buffer for directory name */
/* - - - - - - - - - - - - - - - - */
   void  main (argc, argv)
   int argc;
   char  *argv [] ;
 {
   FILE  *fp;
   long n, REC;
   REC=31L;
/*   fp=fopen("XCURDIR", "w");    */
   fp=fopen( argv [1] , "w");
   getcwd(buffer,31);
   n=0;
   fseek( fp, n*REC,0 );
   fprintf ( fp,  "%s¥n" , buffer );
/*printf ( "%s¥n", buffer );    */
/*printf ( "¥n%s¥n", buffer ); */
   fclose(fp);
 }
```

Fig.2 (Continued)

```
1000'/////////////////////////////////////////////////////////////////
1010'                          U D V . B A S
1020' MAIN PROGRAM TO COMPUTE MINIPLATE URANIUM DENSITY & VOID VOLUME FRACTION
1030'                ミニプレート ウラン密度・気孔率計算用メインプログラム
1040'/////////////////////////////////////////////////////////////////
1050'
1060            REM  $FILE   13,  295
1070    OPEN    "IDFUEL.DAT"  AS  #1
1080    FIELD   #1, 4 AS IDFF$
1090    OPEN    "_NAME.UPL"  AS  #2
1100    FIELD   #2,   8  AS  DA2$
1110    OPEN    "_VOID.UPL"  AS  #3
1120    FIELD   #3,    4 AS WSS2$, 4 AS VVP2$, 4 AS VV2$, 4 AS  FVFV$
1130    OPEN    "SAMPLDSP.DAT"  AS  #4
1140    FIELD   #4,   255   AS  DA$
1150    OPEN    "PLATE.DAT"  AS  #5
1160    FIELD   #5,4 AS WWFC$,   4 AS WWFUEL$,
                   4 AS WWALUM$, 4 AS VVP$,   4 AS WWCOMPACT$
1170    OPEN    "VS.DAT"  AS  #6
1180    FIELD   #6, 4 AS  VVS$
1190    OPEN    "TEMPWSL.DAT"  AS  #7
1200    FIELD   #7, 4 AS TEMPP$, 4 AS WSLL$
1210    OPEN    "WS.DAT"     AS  #8
1220    FIELD   #8, 4 AS  WSS$
1230    OPEN    "_UDT.UPL"    AS  #9
1240    FIELD   #9,  4  AS UDD$,   4 AS  UWW$
1250    OPEN    "_UDVFILE.UPL"  AS  #10
1260    FIELD   #10,  255 AS  FDA$, 4 AS FVS$, 4 AS  FTEMP$, 4 AS FWSL$,
                      4 AS FWS$, 4 AS FDS$, 20  AS  DT2$
1270 '  OPEN    "FWRITE"  AS  #11
1280 '
1290 '
1300   CONSOLE ,,0,1
1310   WIDTH LPRINT 80
1320 '
1330   ON STOP GOSUB   *XSTOP  :  STOP  ON
1340   CLS 3
1350 '
1360  DIM   TEMP(100),     DL(100)
1370  FOR   I = 1   TO  100  :  READ   TEMP(I),  DL(I)   :  NEXT   I
1380 '
1390  '-----------------------------------------------------------------
1400                          *JOBSTART
1410  '-----------------------------------------------------------------
1420 '
1430 INP1$="INPUT P FOR CALCULATION OF VOID VOLUME FRACTION & URANIUM DENSITY O
F MINIPLATE."
1440 INP2$="1-name-8 - - 15- - - - - - - - - - - - - - - - - - - - - - - - - - -
 - - - - 79"
1450 INP3$="PRINT(印刷)=TURN ON PRN. "
1460 INP4$="RETURN-KEY => DENSITY & VOLUME CALC. "
1470 INP5$="QUIT(中止)=CTRL+C"
1480  COLOR 7
1490 PRINT "【 S A M P L E   D E S C R I P T I O N 】":COLOR 5: PRINT
1500 PRINT "INPUT (入力) or AMEND (訂正) SAMPLE NAME(=8 CHARS), CHARACTERISTIC
S etc.      BELOW THE LINE WITHIN TOTAL 255 CHARS.  FOR THE SAME CONTENT, PUSH R
ETURN-KEY. " : PRINT INP4$+INP3$+INP5$ : COLOR 3 : PRINT INP1$; :COLOR 6: PRINT
```

Fig. 3 Source List of UDV. BAS

```
INP2$: COLOR 7
1510 '
1520 '
1530          IF  LOF(4) =  0   THEN  1570
1540          IF  LOF(4) <> 0   THEN  GET   #4,  1
1550  PRINT  DA$
1560 '
1570 LINE  INPUT  A$
1580 IF A$="" AND  LOF(4) <> 0  THEN  GOTO  1670
1590 IF A$="" AND  LOF(4) = 0 THEN LOCATE 0,CSRLIN-1: GOTO 1570  ELSE GOTO 1600
1600   IF  A$ =  "P"  OR   A$ = "p"   THEN    *PLATE1
1610 '
1620   IF  A$ <> "P"  AND  A$ <> "p" AND  A$ <> "" THEN   GOTO   1640
1630 '
1640      LSET  DA$  =  A$
1650      PUT   #4,  1
1660 '
1670    ON  ERROR  GOTO  *JAMP1
1680   COLOR 6 : PRINT "INPUT  :";:COLOR 5:PRINT "WS";:COLOR 6 :PRINT " (WEIGHT
OF SAMPLE IN AIR, g)   FOR REPEAT, HIT RETURN-KEY ";:COLOR 7 : INPUT  WS
1690  IF WS = 0 AND LOF(8) = 0  THEN  LOCATE  0, CSRLIN - 1 :GOTO  1680
1700  IF WS = 0     THEN  COLOR  7 : GOTO  1750
1710   RSET  WSS$ = MKS$(WS)
1720   PUT   #8,  1
1730 '
1740 '
1750   INPUT "INPUT:WBL(WEIGHT OF BUCKET IN LIQUID, g). If no data available, Pu
sh RETURN-KEY." , WBL
1760 '
1770   IF  WBL = 0 THEN    CLINE  =  CSRLIN  : LOCATE  0 , CLINE-2  :PRINT "
                                                                          "
1780 '
1790 '
1800   IF  WBL  =  0   THEN     GOTO    1840
1810   INPUT " INPUT  :WSBL (WEIGHT OF SAMPLE + BUCKET IN LIQUID , g ) = " ; WSB
L
1820   IF  WBL <>  0   AND  WSBL = 0  THEN  LOCATE 0, CSRLIN - 1 : GOTO 1810
1830              WSL =  WSBL - WBL   :  GOTO   1880
1840  WSBL  = 0
1850    COLOR 6 :    LOCATE 0, CLINE - 2  : INPUT "INPUT  :WSL (WEIGHT OF SAMPLE
 IN LIQUID , g ).  FOR REPEAT,  HIT  RETURN-KEY  "   ; WSL
1860 '
1870  IF WSL= 0 AND  LOF(7) <> 0   THEN    GOTO  1920
1880  IF WSL= 0 AND  LOF(7) =  0   THEN  LOCATE 0, CSRLIN - 2: GOTO  1850
1890            GET  #7,  1 :  TEMP = CVS(TEMPP$)  : LSET  TEMPP$ = MKS$(TEMP)
1900   LSET   WSLL$  =  MKS$(WSL) :  PUT  #7,  1
1910 '
1920  COLOR 4 :INPUT "INPUT:  TEMP (LIQUID TEMPERATURE, ℃).   FOR REPEAT,  HIT
RETURN-KEY " ; TEMP
1930  IF TEMP  =  0   THEN    GOTO    1980
1940  IF TEMP  < 17.0  OR  TEMP > 26.9  THEN  LOCATE  0, CSRLIN-1  : GOTO  1920
1950   GET  #7,1  :   WSL  = CVS(WSLL$)  :   LSET WSLL$ =  MKS$(WSL)
1960   LSET TEMPP$ =  MKS$(TEMP)
1970   PUT    #7,   1
1980   IF LOF(7) = 0    THEN   GOSUB  *TEMPINP     ELSE     GET  #7,  1
1990  TEMP = CVS(TEMPP$) : IF   TEMP  =  0  THEN     GOSUB    *TEMPINP
2000   GET    #7,   1
2010  TEMP  =    CVS(TEMPP$)
2020  WSL   =    CVS(WSLL$)
2030   INPUT "INPUT: ATEMP (AIR TEMPERATURE, ℃). HIT Ret.-KEY for the SAME as A
BOVE " ; ATEMP
2040 '
```

Fig.3 (Continued)

```
2050   IF  ATEMP = 0   THEN  ATEMP = TEMP
2060   IF  TEMP  = 0   THEN  *JOBSTART
2070   IF  TEMP  < 17.0  OR  TEMP > 26.9    THEN   1920
2080 PRINT:PRINT:PRINT
2090 '
2100       FOR  I=1 TO 100
2110  PRINT USING  "##.##       ##.####  ";   TEMP(I);DL(I);
2120 'PRINT #11,  USING  "##.##       ##.####  ";   TEMP(I);DL(I);
2130       NEXT I
2140 PRINT : PRINT :PRINT :PRINT:PRINT
2150 '
2160   RESTORE
2170     FOR   I = 1  TO  100
2180         READ   TEMP(I) ,  DL(I)
2190          IF    TEMP(I) =  TEMP    THEN  GOTO   2250
2200          IF    I = 100    THEN  GOSUB *TEMPERR
2210     NEXT  I
2220 '
2230 '
2240   IF   LOF(4)=0   THEN    2280
2250    GET  #4,    1
2260         ON  ERROR  GOTO  *JAMP1
2270   IF  ERR = 55    THEN   *JAMP1
2280       *JAMP1
2290        RESUME   NEXT
2300 '
2310 '
2320 COLOR 7 : PRINT: PRINT "【 S A M P L E   D E S C R I P T I O N 】 " :PRINT
 : PRINT  DA$ ;
2330 '              DoC = Date of Calculation
2340  COLOR 3:PRINT   " DoC = " + DATE$ + "  "  + TIME$ ;
2350  KK = CSRLIN  :  LL = POS(0)
2360  LOCATE    LL - 3 ,  KK   :    PRINT "   "
2370  DAT$ = DATE$ + "  " + TIME$
2380 '
2390 '
2400  KK = CSRLIN  :  LL = POS(0)
2410 '
2420   DA = 0.001293/(1.00 + 0.00367*ATEMP)
2430  COLOR 4:PRINT :PRINT "LIQUID TEMP (℃)= ";TEMP(I);TAB(25);"  ** DL : DENSI
TY OF LIQUID (g/cc)= ";DL(I)
2440 '
2450 COLOR 4 :PRINT :PRINT "AIR    TEMP (℃)= ";ATEMP  ;TAB(25);"  ** DA : DENSI
TY OF AIR    (g/cc)= ";DA
2460 '
2470     GET   #8,  1
2480                    WS =  CVS(WSS$)
2490    DA  = 0.001293/(1.00 + 0.00367*ATEMP)
2500    DS  = WS*DL(I)/(WS-WSL) + DA*(1.00  -  WS/(WS-WSL))
2510    VS  = WS/(DS - DA)
2520 '
2530 '
2540  PRINT : PRINT   "WS   (WEIGHT OF SAMPLE IN AIR, g )     = " ; WS
2550 '
2560          PRINT   "WBL  (WEIGHT OF BUCKET IN LIQUID, g ) = " ; WBL
2570 '
2580          PRINT   "WSBL (WEIGHT OF SAMPLE + BUCKET IN LIQUID, g ) = " ; WSBL
2590 '
2600          PRINT   "WSL  (WEIGHT OF SAMPLE IN LIQUID, g ) = " ; WSL
2610 '
2620 '
2630                 LSET VVS$ = MKS$(VS)   :  PUT   #6,   1
```

Fig.3 (Continued)

```
2640 '
2650   DS = WS*DL(I)/(WS-WSL)     +    DA*( 1.00 - WS/(WS-WSL))
2660 COLOR 6 : PRINT
2670            PRINT  " DS  (DENSITY OF SAMPLE in g/cc)       =  " ; USING "##.##
#"; DS ;: PRINT TAB(55);" g/cc"
2680 PRINT    : PRINT " VS  (VOLUME OF SAMPLE in cc)          =  " ;  VS ;TAB(55)
;" cc"
2690  COLOR 7 :PRINT:PRINT: PRINT "S-KEY= STORE IN FILE    R-KEY= READ & DISPLAY
    P-KEY= PRINT    C-KEY= CONTINUE"
2700 '
2710  OPT$ = INPUT$(1)
2720  IF OPT$= "C"  OR  OPT$= "c"  THEN  2780 ELSE   2730
2730  IF OPT$= "R"  OR  OPT$= "r"  THEN  CLS:GOSUB  *FREAD   ELSE  2740
2740  IF OPT$= "P"  OR  OPT$= "p"  THEN  CLS:GOSUB  *XLPRINT2 ELSE 2750
2750  IF OPT$= "S"  OR  OPT$= "s"  THEN  CLS:GOSUB  *STORE ELSE  LOCATE 0,CSRLIN
-4  :     GOTO 2690
2760 '
2770 '
2780 CLS :  GOTO   *JOBSTART
2790 IF  INKEY$ = ""  THEN  2790
2800 PRINT :PRINT :
2810 '
2820 '-----------------------------------------------------------------------
2830                          *PLATE1
2840 '-----------------------------------------------------------------------
2850 '
2860  IF  LOF(4) = 0  THEN  GOTO  4710  ELSE  GOSUB  *WSVS
2870  NK = LOF(2) + 1
2880  GET #4, 1
2890  '
2900  DAA$ = LEFT$(DA$, 8)
2910  LSET DA2$ = DAA$
2920 '
2930  IF A$="P" OR A$="p"   THEN   *PLATE2
2940 '
2950 '-----------------------------------------------------------------------
2960                          *PLATE2
2970 '-----------------------------------------------------------------------
2980 '
2990 '
3000 '
3010 '   DETERMINATION OF  " VM : Fuel-Meat Volume after Rolling"
3020 '
3030 '     VM  =  VP - ( WF + WC ) / DFR
3040 '                                  VP : VOLUME OF FUEL-PLATE
3050 '
3060 '                             **** WF  : WEIGHT OF FRAME
3070 '                             **** WC  : WEIGHT OF COVER
3080 '                             **** WFC = WF + WC
3090 '                                  DFR : DENSITY OF FRAME & COVER
3100   PRINT   :  COLOR  7
3110   IF   LOF(5) =   0    THEN    3140
3120   GET  #5, 1
3130 '
3140   PRINT "INPUT";:COLOR 6 :PRINT " WCOMPACT";:COLOR 7:PRINT " ( WEIGHT OF FU
EL COMPACT, g ). IF UNCHANGED, HIT Return-key: " ;"    <CURRENT WCOMPACT = ";CVS
(WWCOMPACT$);"g>        "; : INPUT "", WCOMPACT
3150  IF WCOMPACT = 0 AND LOF(5)= 0 THEN  LOCATE 0, CSRLIN- 2: GOTO 3140
3160  IF   WCOMPACT = 0 AND  LOF(5) <>   0   THEN   GOTO    3200
3170    RSET   WWCOMPACT$ = MKS$(WCOMPACT)
3180   PUT  #5,  1
3190   COLOR  2
```

Fig,3 (Continued)

```
3200   PRINT:PRINT "INPUT";:COLOR 4:PRINT "  VP";:COLOR 7:PRINT" (VOLUME OF FUEL
-PLATE, cc). IF  '";:COLOR 5:PRINT "VS";:COLOR 7 :PRINT "' DETERMINED IN THE DEN
SITY MEASURE-MENT IS USED,   PUSH  Return-key.  ";: INPUT "", VP
3210    IF   LOF(6) = 0  THEN  4710
3220                     IF  VP  =  0   THEN  GET #6,1 :VP = CVS(VVS$)
3230                     IF  VP  =  0   THEN  GOTO 3270
3240 '
3250    LSET  VVP$ = MKS$(VP)
3260    PUT   #5,    1
3270           DFR = 2.71
3280       IF  LOF(8) = 0   THEN  4710
3290    GET #8,   1   :   WS          =  CVS(WSS$)
3300    GET #5,   1   :   WCOMPACT  =  CVS(WWCOMPACT$)
3310       WFC  =  WS  -  WCOMPACT
3320                      LSET WWFC$ = MKS$(WFC)
3330                      PUT  #5,  1
3340                      GET  #5,  1
3350 '
3360      WFC  =  CVS(WWFC$) :  VP  =  CVS(VVP$)
3370       VM  =  VP - WFC/DFR
3380 '
3390 '
3400  COLOR 3 : PRINT "- - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 - - - - - - - - - - -"
3410 '
3420 '
3430 '   V % = (1 - (VMTH/VM))*100
3440 '         V % : Void Volume Fraction
3450 '         VMTH: Theoretical Volume of Fuel Meat
3460 '               i.e. Fuel Meat Volume Excluding Void Volume
3470 '         VM  : Volume of Rolled Fuel Meat Including Void Volume
3480 '
3490    COLOR 7
3500 '
3510 '
3520  IF  LOF(5) <> 0  THEN   GET   #5,  1
3530  PRINT "INPUT";: COLOR 6: PRINT" WFUEL";:COLOR 7:PRINT " ( WEIGHT OF FUEL P
OWDER IN THE FUEL COMPACT, g ):","          <CURRENT WFUEL =";CVS(WWFUEL$);" g>
      IF UNCHANGED,  HIT  Return-key.   "; : INPUT "", WFUEL
3540  IF      WFUEL <> 0      THEN  GOTO  3560
3550  IF   CVS(WWFUEL$)=0.0   THEN  LOCATE 0, CSRLIN- 2: GOTO   3530
3560  IF   WFUEL  =  0  AND   LOF(5) <>   0   THEN   GOTO    3610
3570 '
3580 '
3590                   RSET   WWFUEL$ = MKS$(WFUEL)
3600    PUT   #5,   1
3610    GET   #5,   1
3620 '
3630    PRINT   :  COLOR    2
3640                WFUEL = CVS(WWFUEL$) :  WCOMPACT = CVS(WWCOMPACT$)
3650         WALUM =  WCOMPACT - WFUEL  :   GOTO   3740
3660  PRINT  : COLOR  6  :PRINT "INPUT";:COLOR 5:PRINT " WALUM";:COLOR 6:PRINT "
 ( WEIGHT OF AL POWDER IN THE FUEL COMPACT, g ):","          <CURRENT WALUM =";C
VS(WWALUM$);" g>       IF UNCHANGED,  HIT  Return-key.   ";:INPUT "",WALUM
3670 '
3680  IF CVS(WWALUM$)=0 AND WALUM=0 THEN LOCATE 0,CSRLIN-3: GOTO 3660  :' PRINT
" WALUM (5447) =   "; CVS(WWALUM$)
3690  COLOR 3 : PRINT "- - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 - - - - - - - - - - -"
3700     IF   WALUM <> 0   THEN   GOTO   3720
3710     IF   WALUM =  0   THEN   GOTO   3800
3720    RSET  WWALUM$  =   MKS$(WALUM)
```

Fig.3 (Continued)

```
3730    PUT    #5,   1   :GOTO   3800
3740  COLOR 4 :PRINT "TO ";:COLOR 6: PRINT " CHANGE 'WFUEL'";:COLOR 4:PRINT "  B
Y  RE-INPUT  USING  ";:COLOR 5:PRINT "'WALUM',";:COLOR 7:PRINT "  HIT  Return-ke
y;" ;:COLOR 4:PRINT: PRINT "                                        Otherwise,  H
IT  ANY  KEY.";
3750  AB$ = INPUT$(1)
3760  COLOR 7 :PRINT: PRINT "- - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 - - - - - - - - - - - - - -"
3770  IF AB$ = CHR$(&HD)  THEN   3660
3780            COLOR   3  :  RSET   WWALUM$ = MKS$(WALUM)
3790   PUT  #5, 1
3800   GET  #5, 1
3810  WALUM = CVS(WWALUM$) : WFUEL = CVS(WWFUEL$) : WCOMPACT = CVS(WWCOMPACT$)
3820   DU3SI2 = 12.20 :  DUSI = 10.99 :  DU3SI = 15.58
3830   WFUEL  =  WCOMPACT  -  WALUM
3840                  RSET   WWFUEL$ = MKS$(WFUEL)
3850   PUT  #5, 1
3860  PRINT
3870          IF  LOF(1) <> 0 THEN GET  #1, 1
3880  COLOR 7:PRINT "INPUT ID NUMBER FOR DFUEL (DENSITY OF FUEL POWDER): 1---> U
3Si2, 2---> U3Si2/USi  3---> U3Si2/U3Si, 4---> U3Si2/0.5Mo, 5---> U6ME ";:COLOR
3:PRINT "     【";CVS(IDFF$);"】 " ;IDFUEL$
3890 '
3900 '
3910     IDFUEL = VAL(INPUT$(1))
3920                LSET IDFF$=MKS$(IDFUEL) :  PUT  #1,  1
3930  IF IDFUEL = 1 THEN  4030
3940  IF IDFUEL = 2 THEN  4030
3950  IF IDFUEL = 3 THEN  4030
3960  IF IDFUEL = 4 THEN  4030
3970  IF IDFUEL = 5 THEN  4030
3980  IF IDFUEL <> 1 OR IDFUEL<> 2 OR IDFUEL<> 3 OR IDFUEL <> 4  OR IDFUEL <> 5
         THEN  LOCATE 0,  CSRLIN - 2 :  GOTO  3880
3990 '
4000                IF LOF(1) <> 0 THEN  GET #1, 1
4010                IF LOF(1) <> 0 THEN  PRINT CVS(IDFF$)
4020                LSET IDFF$=MKS$(IDFUEL) :  PUT #1,  1
4030  CLS  3
4040  ON  IDFUEL  GOSUB *DU3SI2,  *DU3SI2USI,  *DU3SI2U3SI,  *DU3SI20.5MO, *DU6M
E
4050   NK  =  LOF(9) + 1
4060  LSET UDD$=MKS$(UD)  :  LSET UWW$=MKS$(UW)
4070 '
4080 '
4090 '            DENSITY OF ALUMINUM  POWDER = DALUM
4100          DALUM = 2.70
4110   GET  #5,  1
4120   WFUEL = CVS(WWFUEL$) : WALUM = CVS(WWALUM$) : WCOMPACT = CVS(WWCOMPACT$)
4130 '
4140    VMTH  =  WFUEL/DFUEL   +   WALUM/DALUM
4150 '
4160      VF  =  WFUEL/DFUEL    :   VA = WALUM/DALUM
4170      MVF =  100*VF/( VF + VA )
4180      MVA =  100*VA/( VF + VA )
4190 '
4200                    GET   #8,  1
4210                                WS = CVS(WSS$)
4220 '        V % : VOID VOLUME FRACTION in PERCENT (%)
4230 '
4240          V   =  ( 1.00 - VMTH/VM )*100.0
4250        FVF   =  (100.0 - V)*VF/(VF+VA)
4260         MK   =  1+LOF(3)
```

Fig.3 (Continued)

```
4270 '
4280   GET #5, 1
4290   GET #6, 1
4300   GET #8, 1
4310 '
4320  LSET  WSS2$= WSS$
4330  LSET  VVP2$= VVP$
4340  LSET  VV2$ = MKS$(V)
4350  LSET  FVFV$=MKS$(FVF)
4360 '
4370 COLOR 7
4380 PRINT "- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - -" : COLOR 6
4390   PRINT " URANIUM DENSITY";:COLOR 3:PRINT " in FUEL MEAT ( UD )        =  "
;: COLOR 6:PRINT USING "##.###"; UD ; : PRINT TAB(57) ; "gU/cc"
4400 PRINT " VOID VOLUME FRACTION";:COLOR 3:PRINT " in FUEL MEAT ( V )    =  "
;: COLOR 6:PRINT USING "##.###" ; V ; : PRINT TAB(57) ;"%"
4410 COLOR  7
4420 PRINT "- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - -"
4430                  COLOR  4
4440 '
4450   IF A$ = "P" OR A$ = "p"   THEN   GET   #4,  1
4460 '
4470 '
4480 PRINT             " VP :  VOLUME OF FUEL-PLATE,  cc = " ; VP
4490 PRINT             " WS :  WEIGHT OF FUEL-PLATE,  g  =          " ;     WS
4500 PRINT             " UW :  WEIGHT OF TOTAL URANIUM IN FUEL-PLATE,  g  =  " ; CVS
(UWW$)
4510 PRINT:   PRINT " WFC      : WEIGHT OF FRAME & COVER,  g =     " ; WFC
4520 '
4530 '
4540 COLOR 7: PRINT " WCOMPACT: WEIGHT OF FUEL COMPACT,   g =     " ; WCOMPACT: C
OLOR 4
4550 PRINT:   PRINT " WFUEL : WEIGHT OF FUEL POWDER,    g  =    " ; WFUEL
4560 PRINT             " WALUM : WEIGHT OF AL POWDER,     g  =    " ;WALUM
4570 PRINT:  PRINT " DALUM : TD. of ALUMINUM,         g/cc =    " ; USING "#.##" ;
 DALUM
4580           PRINT " DFR    : TD. of FRAME(AL 6061),  g/cc =    " ; USING "#.##" ;
 DFR
4590  PRINT " DFUEL : TD. of FUEL( ";:COLOR 3:PRINT TDF$; : COLOR 4:PRINT " ),
g/cc =  " ; DFUEL
4600  COLOR   7
4610 PRINT :   PRINT " VMTH :   THEORETICAL VOLUME OF MEAT,  cc =    " ; VMTH
4620 PRINT             " VM    :   VOLUME of ROLLED FUEL MEAT,  cc =    " ; VM
4630 '
4640  COLOR 6
4650 '
4660  PRINT         " FVF :  FUEL VOLUME FRACTION IN MEAT,          %   =   " ; FVF
4670  COLOR  2 : LOCATE 0,CSRLIN -1
4680  PRINT:PRINT " MVF :  MIXING VOLUME OF FUEL FOR MEAT,         %   =   " ; MVF
4690  PRINT         " MVA :  MIXING VOLUME OF ALUMINUM FOR MEAT,  %   =   " ; MVA
4700  PRINT :  COLOR  7 : IF  LOF(5) <>  0  THEN   4720
4710  COLOR 2:PRINT: PRINT "                    W A R N I N G (注意)": COLOR 7 :P
RINT " ***   Execute  Density  Calculation  First  !   ***": PRINT "
   HIT ANY KEY TO CONTINUE " : WHILE INKEY$ = "" : WEND : CLS : LOCATE 0,  0 :
GOTO  *JOBSTART
4720               GOSUB    *DATASTORE
4730  PRINT:PRINT "CONTINUE, QUIT, PRINT MINI-PLATE DATA or ";:COLOR 6:PRINT "CA
LL U P L.EXE ";:COLOR 7:PRINT "? (C／Q／P／U)" ; :  QUES$ = INPUT$(1)
4740  IF QUES$ = "P" OR QUES$ =  "p"  THEN GOSUB *XLPRINT
4750  IF QUES$ = "Q" OR QUES$ =  "q"  THEN CLOSE :  STOP : END
```

Fig.3 (Continued)

```
4760  IF QUES$ = "C" OR QUES$ =  "c"  THEN   CLS : GOTO  *JOBSTART
4770  IF QUES$ <> "C"  AND  QUES$ <> "Q"  AND  QUES$ <> "c"  AND
         QUES$ <> "q"  AND  QUES$ <> "U"  AND  QUES$ <> "u"  AND
         QUES$ <> "P"  AND  QUES$ <> "p"  THEN LOCATE 0,CSRLIN-1:GOTO 4730
4780   COLOR  7  :  CLOSE
4790 '  THE THRESHOLD FRE-VALUES, 17000 AND 16000 (BYTES), MAY CHANGE
        DEPENDING ON THE SYSTEM ENVIRONMENT ADOPTED.
4800  IF  FRE(0) < 17000  OR  FRE(3) < 16000  THEN  CLS : COLOR 2 :PRINT "OUT OF
 MEMORY :  REBOOST  UDV.EXE OR UPL.EXE " :COLOR 7:PRINT :PRINT: STOP : END
4810   RUN  "UPL" :   STOP  :  END
4820 '
4830 ' - - - - - - - - - - - - -  THEORETICAL DENSITY OF FUEL  - - - - - - - -
4840 '
4850 ' LITERATURE VALUES OF X-RAY DENSITY :
4860 ' DU3SI2 = 12.2 :  DUSI = 10.99 :  DU3SI = 15.58
4870 ' DU3SI2 = 12.2 - - MEASURED & CONFIRMED IN THE PRESENT WORK
4880  *DU3SI2            :    DFUEL =  DU3SI2
4890    PRINT :PRINT  " TD OF U3Si2  = " ;  DFUEL  ; "   g/cc"
4900    UD  =  WFUEL*0.927/VM  : UW = WFUEL*0.927
4910    TDF$ = "U3Si2"   :  RETURN
4920  *DU3SI2USI
4930  ' DFUEL  =  DU3SI2*DUSI/( 0.8*DUSI + 0.2*DU3SI2 )
4940    DFUEL  =  11.92  : '  MEASURED IN THE PRESENT WORK
4950    PRINT :PRINT  " TD OF 80 U3Si2 + 20 USi = " ;  DFUEL  ; "   g/cc"
4960    UD  =  WFUEL*0.92/VM  :  UW = WFUEL*0.92
4970    TDF$ =  "80 U3Si2/20 USi"  :  RETURN
4980  *DU3SI2U3SI
4990  ' DFUEL  =  DU3SI2*DU3SI/( 0.8*DU3SI + 0.2*DU3SI2 )
5000    DFUEL  =  12.70  : '  MEASURED IN THE PRESENT WORK
5010    PRINT :PRINT  " TD OF 80 U3Si2 + 20 U3Si = " ;  DFUEL  ; "   g/cc"
5020    UD  =  WFUEL*0.934/VM :  UW = WFUEL*0.934
5030    TDF$ = "80 U3Si2/20 U3Si"  :  RETURN
5040  *DU3SI20.5MO
5050    DFUEL = 12.12    : '  MEASURED IN THE PRESENT WORK
5060    PRINT :PRINT  " TD OF 99.5 U3Si2 + 0.5 Mo = " ;  DFUEL  ; "   g/cc
5070    UD  =  WFUEL*0.922/VM :  UW =  WFUEL*0.922
5080    TDF$ = "99.5 U3Si2/0.5 Mo"   :  RETURN
5090 '
5100  *DU6ME
5110 '
5120    DFUEL  =  17.80  : '  MEASURED & CONFIRMED FOR U6Mn IN THE PRESENT WORK
5130    PRINT :PRINT  " TD OF U6Me (Me=Mn,Fe,Ni) = " ;  DFUEL  ; "   g/cc"
5140    UD  =  WFUEL*0.963/VM :  UW = WFUEL*0.963
5150    TDF$ = "U6Me" :  RETURN
5160 '
5170 '-----------------------------------------------------------------------
5180                          *TEMPINP
5190 '-----------------------------------------------------------------------
5200 '
5210   CLINE = CSRLIN  :  LOCATE  0 ,  CLINE - 1 : PRINT  "
                                                            "
5220 COLOR 4: LOCATE 0,  CSRLIN - 3  :  PRINT  "INPUT:  TEMP (LIQUID TEMPERATURE
, ℃).   FOR REPEAT,  HIT RETURN-KEY " ;
5230            INPUT      TEMP
5240                   IF  TEMP > 27   OR   TEMP  <  17    THEN   5220
5250  GET  #7,  1    :  WSL     =  CVS(WSLL$)
5260        LSET        WSLL$  =  MKS$(WSL)
5270        LSET        TEMPP$ =  MKS$(TEMP)  :    PUT   #7,  1
5280  GET  #7,  1    :  TEMP    =  CVS(TEMPP$)
5290  IF  TEMP  =  0   THEN  GOTO  5220   ELSE    RETURN
5300 '
5310 '
```

Fig.3 (Continued)

```
5320 ' - - - - - - - - - - - - TEMP - DL - - - - - - - - - - - - - - - - - - - -
5330 '
5340 DATA  17.00,  0.8664, 17.10,   0.8663, 17.20,  0.8662, 17.30,  0.8661
5350 DATA  17.40,  0.8660, 17.50,   0.8659, 17.60,  0.8658, 17.70,  0.8657
5360 DATA  17.80,  0.8657, 17.90,   0.8656, 18.00,  0.8655, 18.10,  0.8654
5370 DATA  18.20,  0.8653, 18.30,   0.8652, 18.40,  0.8651, 18.50,  0.8650
5380 DATA  18.60,  0.8649, 18.70,   0.8648, 18.80,  0.8647, 18.90,  0.8646
5390 DATA  19.00,  0.8646, 19.10,   0.8645, 19.20,  0.8645, 19.30,  0.8644
5400 DATA  19.40,  0.8643, 19.50,   0.8642, 19.60,  0.8641, 19.70,  0.8640
5410 DATA  19.80,  0.8639, 19.90,   0.8638, 20.00,  0.8637, 20.10,  0.8637
5420 DATA  20.20,  0.8636, 20.30,   0.8635, 20.40,  0.8634, 20.50,  0.8633
5430 DATA  20.60,  0.8632, 20.70,   0.8631, 20.80,  0.8631, 20.90,  0.8630
5440 DATA  21.00,  0.8629, 21.10,   0.8628, 21.20,  0.8627, 21.30,  0.8626
5450 DATA  21.40,  0.8625, 21.50,   0.8624, 21.60,  0.8623, 21.70,  0.8622
5460 DATA  21.80,  0.8621, 21.90,   0.8620, 22.00,  0.8620, 22.10,  0.8619
5470 DATA  22.20,  0.8619, 22.30,   0.8618, 22.40,  0.8617, 22.50,  0.8616
5480 DATA  22.60,  0.8615, 22.70,   0.8614, 22.80,  0.8613, 22.90,  0.8612
5490 DATA  23.00,  0.8611, 23.10,   0.8611, 23.20,  0.8610, 23.30,  0.8609
5500 DATA  23.40,  0.8608, 23.50,   0.8607, 23.60,  0.8606, 23.70,  0.8605
5510 DATA  23.80,  0.8605, 23.90,   0.8604, 24.00,  0.8603, 24.10,  0.8602
5520 DATA  24.20,  0.8601, 24.30,   0.8600, 24.40,  0.8599, 24.50,  0.8598
5530 DATA  24.60,  0.8598, 24.70,   0.8597, 24.80,  0.8596, 24.90,  0.8595
5540 DATA  25.00,  0.8594, 25.10,   0.8593, 25.20,  0.8592, 25.30,  0.8591
5550 DATA  25.40,  0.8590, 25.50,   0.8589, 25.60,  0.8588, 25.70,  0.8587
5560 DATA  25.80,  0.8586, 25.90,   0.8585, 26.00,  0.8584, 26.10,  0.8583
5570 DATA  26.20,  0.8582, 26.30,   0.8581, 26.40,  0.8580, 26.50,  0.8579
5580 DATA  26.60,  0.8578, 26.70,   0.8577, 26.80,  0.8576, 26.90,  0.8575
5590 '
5600 ' - - - - - - - - - - - - TEMP - DL - - - - - - - - - - - - - - - - - - - -
5610 '
5620 ' EFFECT OF DIFFERENCE IN ROOM TEMP & LIQUID TEMP :  26.1 <--> LIQUID TEMP
5630 ' R.TMP  DA              DS              VS
5640 ' 26.1   1.17997E-3      12.6183         0.933940
5650 ' 25.1   1.18394E-3      12.6182         0.933945
5660 ' 24.1   1.18793E-3      12.6181         0.933949
5670 '
5680 '
5690     *XSTOP : CLOSE : END
5700 '
5710 '-----------------------------------------------------------------------
5720                          *DATASTORE
5730 '-----------------------------------------------------------------------
5740 '
5750  'ABANDON OR STORE PRESENT DATA IN DA2$(FIELD #2); WSS2$, VVP2$, VV2$,
            FVFV$ (FIELD #3); UDD$,  UWW$(FIELD #9)
5760  '
5770  '
5780  PRINT "STORE (保存) PRESENT MAIN MINI-PLATE DATA IN FILES  ? (Y/N) ";
5790  '
5800  STORE$ = INPUT$(1)
5810  IF STORE$ = "Y" OR  STORE$ = "y"  THEN  GOTO  5840
5820  IF STORE$ = "N" OR  STORE$ = "n"  THEN  RETURN
5830  IF STORE$ <> "Y" OR  STORE$ <> "y" OR STORE$ <> "N" OR  STORE$ <> "n"
      THEN  5800
5840      GOSUB   *NAMECHECK
5850      LSET   DA2$ = DAA$
5860     PUT  #2,NK   : PUT  #3,MK  : PUT  #9,NK
5870     RETURN
5880  '
5890 '-----------------------------------------------------------------------
5900                          *NAMECHECK
5910 '-----------------------------------------------------------------------
```

Fig.3 (Continued)

```
5920 '
5930   IF DAA$ = "          "  THEN  COLOR 3 :PRINT "                                    ANO
NYMOUS  SAMPLE  ?! " : COLOR  7: LOCATE 0, CSRLIN-1: RETURN  4730
5940    DAA$ = LEFT$( DA$, 8 )
5950                   NR = 1
5960 '
5970   IF   NR >  LOF(2)  THEN  RETURN
5980     GET  #2,  NR
5990 '
6000   IF  DA2$ <>  DAA$  THEN  NR = NR + 1 : GOTO  5970 ELSE  GOTO 6010
6010    COLOR  3
6020   PRINT : PRINT "     SAME SAMPLE NAME ALREADY PRESENT  ;  CAN'T STORE !"
            : COLOR  7: LOCATE 0, CSRLIN-1: RETURN  4730
6030   COLOR  7 : RETURN
6040 '
6050 '-----------------------------------------------------------------------
6060                          *TEMPERR
6070 '-----------------------------------------------------------------------
6080 '
6090   GET  #7, 1   :  TEMP = CVS(TEMPP$)
6100 '
6110  COLOR 3:PRINT "AMEND  INPUT-ERROR : TEMP (LIQUID TEMPERATURE, ℃)."; :COLOR
 7:PRINT TEMP
6120  LOCATE 52, CSRLIN - 1
6130  INPUT  "", TEMP
6140  LSET  TEMPP$ = MKS$(TEMP)  :  PUT #7,  1
6150  RETURN  1920
6160 '
6170 '-----------------------------------------------------------------------
6180                          *XLPRINT
6190 '-----------------------------------------------------------------------
6200  '
6210  LPRINT: LPRINT "【 M I N I - P L A T E   D A T A 】": LPRINT :   LPRINT
 DA$  ;
6220  LPRINT "                                DoC = " + DATE$ + "  "  + TIME$
6230 '
6240 '
6250 LPRINT "- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 -"
6260 LPRINT " URANIUM DENSITY in FUEL MEAT ( UD )        =   " ; USING "##.###";
UD ; :LPRINT TAB(57);"gU/cc"
6270 LPRINT " VOID VOLUME FRACTION in FUEL MEAT ( V )    =   " ; USING "##.###";
V  ; :LPRINT TAB(57);"%"
6280 LPRINT "- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 -"
6290 '
6300 LPRINT: LPRINT " VP :  VOLUME OF FUEL-PLATE,  cc =    " ; VP
6310 LPRINT: LPRINT " WS :  WEIGHT OF FUEL-PLATE,  g  =    " ;    WS
6320 LPRINT         " UW :  WEIGHT OF TOTAL URANIUM IN FUEL-PLATE,  g  =  " ; CV
S(UWW$)
6330 LPRINT: LPRINT " WFC      : WEIGHT OF FRAME & COVER,  g =    " ; WFC
6340 '
6350 '
6360 LPRINT         " WCOMPACT: WEIGHT OF FUEL COMPACT,   g =    " ; WCOMPACT
6370 LPRINT: LPRINT " WFUEL : WEIGHT OF FUEL POWDER,  g  =   " ; WFUEL
6380 LPRINT         " WALUM : WEIGHT OF AL POWDER,    g  =   " ;WALUM
6390 LPRINT: LPRINT " DALUM : TD. of ALUMINUM,        g/cc =   " ; USING "#.##"
; DALUM
6400 LPRINT         " DFR   : TD. of FRAME(AL 6061),  g/cc =   " ; USING "#.##"
; DFR
6410 LPRINT " DFUEL : TD. of FUEL( " + TDF$ + " ),  g/cc =  " ; DFUEL
6420 LPRINT:  LPRINT " VMTH : THEORETICAL VOLUME OF MEAT,   cc =  " ; VMTH
```

Fig.3 (Continued)

```
6430 LPRINT            " VM   : VOLUME OF ROLLED FUEL MEAT,   cc =  " ; VM
6440 '
6450 '
6460 LPRINT:  LPRINT " FVF :  FUEL VOLUME FRACTION IN MEAT,        %   =   " ; F
VF
6470 LPRINT:  LPRINT " MVF :  MIXING VOLUME OF FUEL FOR  MEAT,     %   =   " ; M
VF
6480 LPRINT            " MVA :  MIXING VOLUME OF ALUMINUM FOR MEAT, %   =   " ; M
VA
6490 LPRINT:  LPRINT
6500 LOCATE 0,CSRLIN-1:RETURN 4730
6510 '
6520 '-----------------------------------------------------------------------
6530                          *WSVS
6540 '-----------------------------------------------------------------------
6550 '
6560  IF  LOF(6) = 0  OR  LOF(8) = 0  THEN  RETURN
6570  CLS : LOCATE 0, 0
6580   GET  #6, 1  : GET  #8, 1
6590 '
6600   PRINT  : PRINT " WS: WEIGHT  OF  PLATE = "; CVS(WSS$);"g ";:LINE (0,0)-
(319,16) ,3,BF: LOCATE 42, 1 : PRINT "VS: VOLUME  OF  PLATE =" ;CVS(VVS$);"cc"
 :LINE (320,0) - ( 639,16), 1,BF
6610  RETURN
6620 '
6630 '-----------------------------------------------------------------------
6640                          *STORE
6650 '-----------------------------------------------------------------------
6660 '
6670   LSET FDA$ =DA$ : LSET FVS$=MKS$(VS) : LSET FTEMP$=MKS$(TEMP)
6680   LSET FWSL$=MKS$(WSL): LSET FWS$=MKS$(WS) : LSET FDS$=MKS$(DS)
6690   LSET DT2$ = DAT$
6700   PUT #10,  LOF(10)+1
6710   CLS : RETURN  *JOBSTART
6720 '
6730 '-----------------------------------------------------------------------
6740                          *FREAD
6750 '-----------------------------------------------------------------------
6760 '
6770   FOR  K = 1 TO  LOF(10)
6780   GET #10,  K
6790  COLOR 7 : PRINT: PRINT " 【 SAMPLE  DESCRIPTION 】 ";: IF
K = LOF(10) THEN  PRINT  LOF(10)  ELSE  PRINT SPACE$(8)
6800  PRINT  FDA$
6810 '                  DoC = Date of Calculation
6820  COLOR 3:PRINT   " DoC =  " ; DT2$
6830 '
6840  COLOR 4:PRINT "LIQUID TEMP (℃)= ";CVS(FTEMP$)
6850  PRINT   "WS   (WEIGHT OF SAMPLE IN AIR, g )     = " ; CVS(FWS$)
6860  PRINT   "WSL  (WEIGHT OF SAMPLE IN LIQUID, g ) = " ; CVS(FWSL$)
6870  COLOR 6
6880   PRINT  " DS  (DENSITY OF SAMPLE in g/cc)       =  " ; USING "##.###"; CVS(
FDS$) ;: PRINT TAB(55);" g/cc"
6890  PRINT " VS  (VOLUME OF SAMPLE in cc)          =  " ; CVS(FVS$) ;TAB(55);" c
c"
6900   NEXT  K
6910 '
6920  COLOR 7 : PRINT "- - - - - - - - - - - - - - - - - - - - - - - - HIT ANY K
EY TO CONTINUE - - - - "
6930 '
6940  WHILE INKEY$="" : WEND :'  DUMY$ = INPUT$(1)
6950  CLS : RETURN  *JOBSTART
```

Fig.3 (Continued)

```
6960 '
6970 '-----------------------------------------------------------------------
6980                        *XLPRINT2
6990 '-----------------------------------------------------------------------
7000 '
7010  LPRINT:   LPRINT "【 S A M P L E   D E S C R I P T I O N 】": LPRINT :
LPRINT   DA$  ;
7020  LPRINT "                                   DoC = " + DATE$ + "  "  + TIME$  ;
7030  LPRINT:LPRINT :LPRINT "LIQUID TEMP (℃)= ";TEMP(I);TAB(25);"  ** DL : DENS
ITY OF LIQUID (g/cc)= ";DL(I)
7040        LPRINT            "AIR    TEMP (℃)= ";ATEMP  ;TAB(25);"  ** DA : DENSI
TY OF AIR    (g/cc)= ";DA
7050  LPRINT:LPRINT   "WS   (WEIGHT OF SAMPLE IN AIR, g )     = " ; WS
7060          LPRINT   "WBL  (WEIGHT OF BUCKET IN LIQUID, g ) = " ; WBL
7070          LPRINT   "WSBL (WEIGHT OF SAMPLE + BUCKET IN LIQUID, g ) = " ; WSBL
7080          LPRINT   "WSL  (WEIGHT OF SAMPLE IN LIQUID, g ) = " ; WSL
7090  LPRINT : LPRINT  " DS  (DENSITY OF SAMPLE in g/cc)       =  " ; USING "##.#
##"; DS ;: LPRINT TAB(55);" g/cc"
7100  LPRINT   :LPRINT " VS  (VOLUME OF SAMPLE in cc)          =  " ;  VS ;TAB(55
);" cc"
7110  LPRINT  : LPRINT "- - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - " : LPRINT
7120  CLS : RETURN  *JOBSTART
7130 '
7140 ' ////////////////////////// E N D   O F   F I L E  /////
```

Fig.3 (Continued)

```
1000'/////////////////////////////////////////////////////////////
1010'                          U P L. B A S
1020' SUBPROGRAM TO COMPUTE MINIPLATE URANIUM DENSITY & VOID VOLUME FRACTION
1030'                ミニプレート ウラン密度・気孔率計算用サブプログラム
1040'/////////////////////////////////////////////////////////////
1050'
1060    REM     $FILE  13,  295
1070    OPEN    "_DATFILE.UPL"  AS  #1
1080    FIELD   #1, 8 AS DA2C$, 4  AS  WSS2C$, 4 AS VVP2C$, 4  AS  UDDC$, 4 AS
                  VV2C$,  4  AS  UWWC$, 4 AS  FVFVC$
1090    OPEN    "_NAME.UPL"  AS  #2
1100    FIELD   #2,  8  AS   DA2$
1110    OPEN    "_VOID.UPL"   AS  #3
1120    FIELD   #3,  4  AS WSS2$,4 AS VVP2$,  4 AS VV2$, 4 AS FVFV$
1130    OPEN    "INSREC.DAT"  AS  #4
1140    FIELD   #4,8 AS TM$, 4 AS TM1$, 4 AS TM2$, 4 AS TM3$, 4 AS TM4$,4 AS TM5$
                ,4 AS  TM6$
1150    OPEN    "AMNAME.DAT"  AS #5
1160    FIELD   #5,  8  AS  DA$
1170    OPEN    "AVOID.DAT"   AS  #6
1180    FIELD   #6,    4 AS AWSS2$,4 AS AVVP2$,  4 AS AVV2$, 4 AS AFVFV$
1190 '
1200    OPEN    "AUDT.DAT" AS #7
1210    FIELD   #7,  4 AS AUDD$,  4 AS  AUWW$
1220    OPEN    "DAP1.DAT" AS #8
1230    FIELD   #8,  8  AS  ADA2$
1240    OPEN    "_UDT.UPL" AS #9
1250    FIELD   #9,  4 AS  UDD$,    4 AS  UWW$
1260    OPEN    "_VOID.TMP"   AS  #10
1270    FIELD   #10, 4 AS AAWSS2$,4 AS AAVVP2$,4 AS AAVV2$, 4 AS AAFVFV$
1280    OPEN    "_UDT.TMP" AS #11
1290    FIELD   #11, 4 AS AAUDD$, 4  AS  AAUWW$
1300    OPEN    "_NAME.TMP"  AS  #12
1310    FIELD   #12,  8 AS AADA2$
1320 '
1330 '
1340    ON  ERROR GOTO *ERRTREAT
1350    ON  KEY  GOSUB *CANCEL,*CANCEL,*CANCEL: KEY(1) ON :KEY(2) ON :KEY(3) ON
1360    ON  HELP  GOSUB  *NAMESEARCH  :  HELP  ON
1370    OPTION  BASE  1
1380    DIM   AP2$(80)
1390                    *JOBSTART
1400 '
1410  NN = LOF(3)
1420  LPRINT  CHR$(27);  ">" ;
1430  LPRINT  CHR$(27);  "H" ;
1440  LPRINT  CHR$(27);  "T26" ;
1450  LPRINT  CHR$(27);  "P" ;
1460 '
1470  WIDTH  LPRINT  80
1480  JJ =0: FOR   H=1 TO  LOF(2) : JJ= JJ+1: NEXT  H : JJX =  JJ
1490  NNN = LOF(2)
1500 '
1510 '
1520   GOSUB  *DATAFILE
1530  IF FLAG=3  THEN  1550
1540   GOSUB  *NSCROLL
```

Fig. 4 Source Lists of UPL. BAS and F_DELETE. BAT

```
1550 '
1560 '  LPRINT    CHR$(27); "!";
1570 '
1580  IF  FLAG = 3  THEN COLOR 7 : N = NAP : GOTO  1680
1590 ' LPRINT   CHR$(27); CHR$(34);
1600            *JOBSTART1
1610   IL  =  0
1620   FOR  N = 1   TO   NNN
1630   IL = IL + 1
1640   IF  N = 1   THEN   GOTO  1680
1650   IF  IL MOD  5 = 1 AND N <> 1   THEN  GOTO   1680
1660 '
1670    GOSUB  *TABLELINE
1680    GET  #1,   N
1690 '
1700   CONSOLE ,,,,0 : PRINT  DA2C$   ;: GOSUB   *TABLECOLM2
1710 '
1720   PRINT    USING " ###.####"    ; CVS(WSS2C$);
1730   PRINT    USING "    ###.####" ; CVS(VVP2C$);
1740   PRINT    USING "    ###.####" ; CVS(UWWC$);
1750   PRINT    USING "     ###.##"  ; CVS(UDDC$);
1760   PRINT    USING "      ###.##" ; CVS(VV2C$);
1770   PRINT    USING "     ###.##"  ; CVS(FVFVC$)
1780 '
1790 '
1800  PRINT
1810 '
1820  IF  IL  MOD  5 = 0  AND   LOF(2) <>  IL   THEN   GOSUB   *TABLELINE
1830  IF  IL  = LOF(2)    THEN  GOTO   1960  ELSE  COLOR  4 :  GOTO  1840
1840  IF  IL  MOD  5 = 0  THEN  LOCATE 0, CSRLIN   :PRINT "RET.KEY=CONTINUE(次画
面) ";:COLOR 3 : PRINT "E-KEY=DATA END(最終データ) "; :COLOR 5:PRINT "HELP-KEY=S
AMPLE(試料名一覧)"
1850   COLOR  7  : IF  IL MOD 5  <>  0   GOTO  1920
1860   RET.KEY$ = CHR$(&HD) : KS$ = INKEY$
1870  WHILE    KS$ <> RET.KEY$
1880   IF  KS$ = "E" OR KS$= "e"  THEN  N= LOF(2) : CLS : IL =JJX :GOTO 1680
1890 '
1900   KS$ = INKEY$
1910  WEND  : CLS
1920   IF  IL  MOD  5  =  0   AND  INKEY$ <>""  THEN    PRINT
1930  NEXT N
1940  *ERRTREAT : IF ERR=1 THEN  RESUME  NEXT
1950 '               ERR=1 : NEXT without FOR
1960  COLOR 5 : GOSUB  *TABLELINE3  :  COLOR  7
1970   PRINT
1980  LPRINT
1990  '
2000    GOSUB  *XOPTION
2010 '
2020 '
2030   CLOSE :  STOP   :  END
2040 '
2050 '          /////////////////   TABLE   ////////////////////////////
2060 '
2070 *LTABLELINE
2080  CONSOLE ,,,,1
2090  '
2100      FOR  MM = 1  TO  79
2110      IF   MM = 9  THEN   GOSUB   *TABLECROS1
2120      LPRINT CHR$(149);
2130      NEXT  MM  :  RETURN
2140   *TABLELINE
```

Fig.4 (Continued)

```
2150  CONSOLE ,,,,0
2160      FOR  MM = 1  TO  79
2170     IF  MM = 9  THEN    GOSUB   *TABLECROS2
2180     PRINT  CHR$(149);
2190      NEXT  MM
2200  RETURN
2210 '
2220 *TABLECOLM1      :'
2230 CONSOLE ,,,,1
2240 '
2250     LPRINT  CHR$(150);
2260  RETURN
2270 *TABLECOLM2      :'
2280  CONSOLE ,,,,0
2290     PRINT  CHR$(150);
2300 '
2310  RETURN
2320 '
2330 *TABLECROS1     :'
2340 CONSOLE ,,,,1
2350      LPRINT CHR$(143);
2360  RETURN
2370   *TABLECROS2
2380  CONSOLE ,,,,0
2390      PRINT CHR$(143);
2400 RETURN
2410 *TABLECROS3     :'
2420 CONSOLE ,,,,1
2430      LPRINT CHR$(144) ;
2440  RETURN
2450  *TABLECROS4
2460  CONSOLE ,,,,0
2470      PRINT CHR$(144);
2480 RETURN
2490 *TABLECROS5     :'
2500 CONSOLE ,,,,1
2510      LPRINT CHR$(145);
2520 RETURN
2530 *TABLECROS6
2540  CONSOLE ,,,,0
2550    PRINT CHR$(145);
2560 RETURN
2570 *LTABLELINE1
2580 CONSOLE ,,,,1
2590      FOR  MM = 1  TO  79
2600     IF  MM = 9  THEN  GOSUB  *TABLECROS5
2610     LPRINT  CHR$(149);
2620      NEXT  MM : RETURN
2630 '
2640    *TABLELINE1
2650   CONSOLE ,,,,0
2660       FOR  MM = 1  TO  79
2670     IF  MM =  9 THEN  GOSUB  *TABLECROS6
2680      PRINT CHR$(149);
2690      NEXT  MM
2700 RETURN
2710  *TABLELINE2
2720 CONSOLE ,,,,1
2730       FOR  MM = 1  TO  79
2740     IF  MM = 9 THEN  GOSUB  *TABLECROS3
2750     LPRINT CHR$(149);
2760      NEXT  MM :RETURN
```

Fig.4 (Continued)

```
2770   *TABLELINE3
2780 '
2790  CONSOLE ,,,,0
2800       FOR  MM = 1  TO  79
2810     IF  MM =  9  THEN  GOSUB  *TABLECROS4
2820      PRINT CHR$(149);
2830      NEXT  MM
2840 RETURN
2850 '
2860 '-----------------------------------------------------------------------
2870                          *DATAFILE
2880 '-----------------------------------------------------------------------
2890 '
2900 '
2910   FOR  K = 1  TO  NNN
2920     GET  #2,  K
2930     GET  #3,  K
2940     GET  #9,  K
2950 '
2960   LSET  DA2C$  = DA2$  :  LSET WSS2C$ = WSS2$
2970   LSET  VVP2C$ = VVP2$ :  LSET UDDC$  = UDD$
2980   LSET  VV2C$  = VV2$  :  LSET UWWC$  = UWW$
2990                           LSET FVFVC$ = FVFV$
3000     PUT  #1, K
3010  '
3020    NEXT  K
3030    RETURN
3040                      *XOPTION
3050   IF  LOF(2)  MOD  5 = 1  THEN    JM  =  12
3060   IF  LOF(2)  MOD  5 = 2  THEN    JM  =   9
3070   IF  LOF(2)  MOD  5 = 3  THEN    JM  =   6
3080   IF  LOF(2)  MOD  5 = 4  THEN    JM  =   3
3090   IF  LOF(2)  MOD  5 = 0  THEN    JM  =   0
3100   IF  LOF(2)  <=   5      THEN    JM  =   0
3110 '
3120    FOR  HH=1 TO  JM : PRINT : NEXT  HH  : LOCATE 0, CSRLIN
3130 '
3140 '
3150 LOCATE 0,24 :PRINT "CONTINUE,QUIT,PRINT,AMEND,INSERT,DELETE,";
3160 COLOR 6:PRINT " CALL UDV.EXE ";:COLOR 7:PRINT "(C/Q/P/A/I/D/U)";
3170 '
3180 OPT$ = INPUT$(1)
3190 IF OPT$ ="Q"  OR OPT$="q"  THEN  CLS : STOP:END : ELSE 3200
3200 IF OPT$ ="C"  OR OPT$="c"  THEN  CLS : GOTO   *JOBSTART1 : ELSE 3210
3210 IF OPT$ ="P"  OR OPT$="p"  THEN  CLS : GOTO   *XLPRINT :   ELSE 3220
3220 IF OPT$ ="I"  OR OPT$="i"  THEN  FLAG=4:CLS:GOSUB *DTMANIP:ELSE 3230
3230 IF OPT$ ="A"  OR OPT$="a"  THEN        GOSUB  *DATAAMEND : ELSE 3240
3240 IF OPT$ ="D"  OR OPT$="d"  THEN  FLAG=2:    GOSUB *DTMANIP:ELSE 3250
3250 IF OPT$ ="U"  OR OPT$="u"  THEN  RUN  "UDV" : ELSE LOCATE 0, CSRLIN : GOTO
3150
3260 RETURN
3270 '
3280 ' ----------------------------------------------------------------------
3290                          *NSCROLL
3300 ' ----------------------------------------------------------------------
3310 '
3320  CONSOLE 8,18,0,1: SCREEN 3,0
3330 '
3340 PRINT "ALUMINUM DISPERSION PLATE-TYPE FUE
L DATA" : LINE(0,0)-(639,16),3,BF:LOCATE 0, CSRLIN - 1
3350  COLOR 5 : GOSUB *SIGN : COLOR 3
3360 PRINT "試料名 ";:COLOR 5 : PRINT CHR$(150);:COLOR 3 :PRINT " 燃料板重量
```

Fig.4 (Continued)

```
燃料板体積　総ウラン量　　ウラン密度　ボイド率　　燃料体積率"
3370 COLOR 7:PRINT  "SAMPLE  ";:COLOR 5 : PRINT CHR$(150);:COLOR 7 :PRINT " PLAT
E WT.   PLATE VOL.  TOTAL U WT.   U DENSITY   VOID  %   FUEL VOL.%"
3380 COLOR 3 :PRINT  "  AP$   ";:COLOR 5 : PRINT CHR$(150);:COLOR 3 :PRINT "  WS
 (g)      VP (cc)      UW (g)      UD (gU/cc)   V (%)    FVF (%)  "
3390 '
3400  LOCATE  0,  CSRLIN - 1 :COLOR 5 : GOSUB *TABLELINE
3410  COLOR 7 : RETURN
3420    *SIGN
3430  CONSOLE ,,,,0
3440       FOR  MM = 1  TO  58
3450    IF  MM = 9  THEN  GOSUB  *TABLECROS6
3460 '
3470   PRINT  CHR$(149);
3480    NEXT  MM
3490    PRINT " Help-Key=Sample list";
3500   RETURN
3510 '
3520 ' ----------------------------------------------------------------------
3530                        *XLPRINT
3540 ' ----------------------------------------------------------------------
3550 '
3560 LPRINT "ALUMINUM DISPERSION PLATE-TYPE FU
EL DATA" : LOCATE 0, CSRLIN - 1
3570 COLOR  5  : GOSUB  *LTABLELINE1  :LPRINT:  CONSOLE ,,,,0
3580 LPRINT  "試料名  ";:CONSOLE ,,,,1:LPRINT CHR$(150);:CONSOLE ,,,,0:LPRINT "
燃料板重量　燃料板体積　総ウラン量　　ウラン密度　ボイド率　　燃料体積率"
3590 '
3600 LPRINT  "SAMPLE  ";:CONSOLE ,,,,1 :LPRINT CHR$(150);:CONSOLE ,,,,0 :LPRINT
" PLATE WT.   PLATE VOL.  TOTAL U WT.   U DENSITY   VOID  %   FUEL VOL.%"
3610 LPRINT  "        ";:CONSOLE ,,,,1 :LPRINT CHR$(150);:CONSOLE ,,,,0 :LPRINT
"   (g)         (cc)         (g)         (gU/cc)      (%)       (%)  "
3620 '
3630  FOR  N = 1  TO  NNN
3640 '
3650  GOSUB  *LTABLELINE : LPRINT
3660       GET  #1,   N
3670 '
3680   CONSOLE ,,,,0 : LPRINT DA2C$ ;: GOSUB *TABLECOLM1
3690 '
3700 LPRINT    USING " ###.####"    ; CVS(WSS2C$);
3710 LPRINT    USING "    ###.####" ; CVS(VVP2C$);
3720 LPRINT    USING "    ###.####" ; CVS(UWWC$);
3730 LPRINT    USING "     ###.##"  ; CVS(UDDC$);
3740 LPRINT    USING "      ###.##" ; CVS(VV2C$);
3750 LPRINT    USING "     ###.##"  ; CVS(FVFVC$)
3760 '
3770 '
3780  NEXT   N
3790  GOSUB   *TABLELINE2
3800  LPRINT
3810  CONSOLE  ,,,,0 : PRINT :  RETURN  *XOPTION
3820  ON  ERROR  GOTO  *JAMP3
3830  IF  ERR = 82  THEN  *JAMP3
3840  *JAMP3
3850   RESUME  NEXT
3860 ' ----------------------------------------------------------------------
3870                        *DATAAMEND
3880 ' ----------------------------------------------------------------------
3890    KESC = 1
3900    CLS
3910          GOSUB   *UNGAR
```

Fig.4 (Continued)

```
3920 '
3930            KESC = 1
3940            FLAG = 0
3950  COLOR  2
3960 '
3970  LD = 8-LEN(DA$)
3980 '
3990  PNAME$ =   DA$ + SPACE$(LD)
4000  FOR   KK = 1  TO  LOF(2)
4010  GET  #2,   KK
4020  IF  DA2$ = PNAME$  THEN  GOTO  4060
4030   IF  LOC(2) = LOF(2) THEN  PRINT :PRINT "        SUCH SAMPLE NOT PRESENT I
N DATA FILE !": COLOR 7 : LOCATE 0,CSRLIN-4:RETURN   *XOPTION
4040  NEXT  KK
4050 COLOR 6 : IF FLAG = 1 THEN PRINT : PRINT " SAMPLE'S RECORD No. = "; NLOC ;
            :GOTO 4080
4060   COLOR  6
4070   NLOC = LOC(2) : PRINT :PRINT " SAMPLE'S RECORD No. = " ; NLOC ;
4080   GET  #1, NLOC
4090 '
4100  PRINT "  CURRENT  DATA : " ; "      SAMPLE NAME = " ; DA2C$
4110   COLOR  4 : PRINT    " WS=" ;CVS(WSS2C$);" VP=";CVS(VVP2C$); " UW=";CVS(UW
WC$);" UD="; CVS(UDDC$); " V =";  CVS(VV2C$); " FVF=";CVS(FVFVC$)
4120  KESC = 1
4130  COLOR  7
4140  GET #3,  NLOC
4150  GET #9,  NLOC
4160  PRINT :PRINT "INPUT  NEW VALUES OF WS, VP, UW, UD, V, FVF  [QUIT(";:COLOR
2 :PRINT "中止";:COLOR 7:PRINT ")=> HIT ";:COLOR 3:PRINT "F・1-3"; :COLOR 7:P
RINT " KEY] ";
4170  INPUT ": " ,WS, VP, UW, UD, V, FVF
4180 '
4190 '
4200 IF  WS=0 AND VP=0 AND UW=0 AND UD=0 AND V=0 AND FVF=0 THEN
     CLS : RETURN  *XOPTION
4210 '
4220  IF  WS=0   THEN  GOTO  4230  ELSE  LSET WSS2$ =   MKS$(WS)
4230  IF  VP=0   THEN  GOTO  4240  ELSE  LSET VVP2$ =   MKS$(VP)
4240  IF  UW=0   THEN  GOTO  4250  ELSE  LSET UWW$  =   MKS$(UW)
4250  IF  UD=0   THEN  GOTO  4260  ELSE  LSET UDD$  =   MKS$(UD)
4260  IF  V=0    THEN  GOTO  4270  ELSE  LSET VV2$  =   MKS$(V)
4270  IF  FVF=0  THEN  GOTO  4280  ELSE  LSET FVFV$ =   MKS$(FVF)
4280  '
4290  '
4300  PUT  #3,  NLOC
4310  PUT  #9,  NLOC
4320 '
4330   GOSUB  *DATAFILE
4340   KESC = 1
4350   GOTO  3140
4360                *CHECK
4370            LINE INPUT  ANAME$
4380   IF  ANAME$ = ""     THEN  RETURN  *XOPTION
4390 '
4400     LSET  SNAME$ =  ANAME$
4410     PUT  #5,  1   :   RETURN   3960
4420   ON  ERROR  GOTO  *JAMP2
4430  IF  ERR = 55   THEN   *JAMP2
4440  *JAMP2
4450     RESUME  NEXT
4460 ' ------------------------------------------------------------------------
4470                        *UNGAR
```

Fig.4 (Continued)

```
4480 ' ----------------------------------------------------------------------
4490   PRINT
4500 PRINT "INPUT (入力) SAMPLE NAME(WITHIN 8 CHARS) TO AMEND (訂正) DATA or";:C
OLOR 5:PRINT " PUSH HELP-KEY :" :COLOR 7
4510  PRINT "PRIOR SAMPLE NAME (IF PRESENT,SHOWN IN VIOLET) MAY BE USED OR EDITE
D TO NEW ONE"
4520  COLOR 3
4530                    IF  LOF(5) = 0   THEN  4560
4540 IF  LOF(5) <> 0  THEN   GET   #5,  1
4550 PRINT  DA$  : COLOR 7
4560   LINE  INPUT  A$
4570 '
4580 IF A$="" AND LOF(5) <> 0 THEN  GOTO  4670
4590 IF A$="" AND LOF(5) =0   THEN LOCATE 0,CSRLIN-1: GOTO  4560  ELSE   GOTO 46
00
4600 '
4640   LSET  DA$   =   A$
4650   PUT   #5,  1
4660 '
4670   ON  ERROR  GOTO  *JAMP1
4680  *JAMP1
4690 '
4700      RESUME   NEXT
4710      RETURN
4720 '
4730 ' ----------------------------------------------------------------------
4740                          *NAMESEARCH
4750 ' ----------------------------------------------------------------------
4760 '
4770     KESC=3
4780              U = 0 : ROL = 20
4790 '
4800     SCREEN 3,0  : CONSOLE  ,,0,0
4810     LINE (0,0) -( 639, 60) ,5,B
4820     SCREEN ,1 : WIDTH 80,25 :PRINT
4830 '
4840 '
4850   LOCATE 0, CSRLIN - 1
4860 PRINT "ＡＬＵＭＩＮＵＭ　ＤＩＳＰＥＲＳＩＯＮ　ＰＬＡＴＥ－ＴＹＰＥ　ＦＵＥ
Ｌ　ＤＡＴＡ";
4870   PRINT "     SELECT SAMPLE NAME TO AMEND DATA  BY MOVING CURSOR AND HIT RE
TURN-KEY."
4880 PRINT  "     データ確認・修正用サンプル名を選択後、リターン　キーを押して下
さい。"
4890     LINE (0,316) -( 639,360) ,5,B
4900  LOCATE 0,20 : PRINT "       FINAL RECORD NUMBER =  "; LOF(2);: LINE (220,32
0)-(260,337),3,BF :PRINT "       ***       ROLL WIDTH = "; ROL ;:LINE (500,320)-
(540,337),1,BF:PRINT "  RECORDS"
4910 '
4920 '
4930 PRINT  "    PUSH ROLL-UP KEY TO SEE NEXT SCREEN.  EACH RECORD NUMBER IS SHO
WN IN BLUE. "
4940 '
4950   IF LOF(2)=0 THEN  LOCATE 20,10 : PRINT " ＮＯ　　ＤＡＴＡ　　ＡＶＡＩＬＡ
ＢＬＥ " :PRINT :PRINT : STOP : END
4960          CONSOLE 0,25,0,0
4970  FOR  I =1  TO  LOF(2)
4980   GET  #2,  I
4990   AP2$(I) = DA2$
5000  NEXT  I
5010 '
5020 '
```

Fig.4 (Continued)

```
5030  KESC = 3
5040 '
5050  IF LOF(2) <=  ROL THEN GOTO  5060  ELSE  GOTO  5080
5060    FOR CT=0 TO LOF(2)-1 :LOCATE (CT MOD 5)*16+2,CT¥5+5:PRINT AP2$(CT+1) : N
EXT CT
5070  CT = 0 : LCT=LOF(2) : GOTO *CUR1
5080  IF LOF(2) > ROL THEN IF LOF(2) MOD ROL=0 THEN L=LOF(2)¥ROL : KK=L  ELSE L=
LOF(2)¥ROL+1 : KK=L :GOTO  5090
5090  FOR CT=0 TO  ROL -1  :  LOCATE (CT MOD 5)*16+2,CT¥5+5:PRINT AP2$(CT+1) : N
EXT CT
5100  CT = 0 : LCT=ROL      : GOTO *CUR2
5110    LINE (0,0) -( 639, 60) ,5,B
5120 '
5130   *CUR1
5140   CT1=(CT MOD 5)*16+2 :  CT2=CT¥5+5
5150   IF  CT+1 >= 10  THEN  5160  ELSE COLOR 4:LOCATE CT1,CT2:PRINT  AP2$(CT+1)
;CT+1: LINE (8*CT1+72,16*CT2)-(8*CT1+80,16*CT2+16),1,BF : GOTO  5170
5160   COLOR 4:LOCATE  CT1, CT2 : PRINT AP2$(CT+1);CT+1: LINE (8*CT1+72,16*CT2)-
(8*CT1+96,16*CT2+16),1,BF
5170    IK$=INKEY$  :  IF IK$=""   THEN  5170
5180   COLOR 0:LOCATE (CT MOD 5)*16+2,CT¥5+5:  PRINT AP2$(CT+1)
5190   IF IK$=CHR$(13)   THEN   LX=CT+1 : GOTO  5460
5200 IF IK$=CHR$(&H1E) THEN IF CT<5 THEN 5250  ELSE CT=CT-5:GOTO 5250
5210 IF IK$=CHR$(&H1D) THEN IF CT=0 THEN 5250  ELSE CT=CT-1:GOTO 5250
5220 IF IK$=CHR$(&H1C) THEN IF CT>(ROL-2) OR LCT=CT+1  THEN 5250 ELSE CT=CT+1:GO
TO  5250
5230 IF IK$=CHR$(&H1F) THEN IF CT>(ROL-6) OR LCT=<CT+5 THEN 5250 ELSE CT=CT+5:GO
TO  5250
5240 IF IK$=CHR$(&HF8) THEN *CUR1
5250    GOTO *CUR1
5260   *CUR2
5270          IF  KK  = L THEN  LX=CT+1 : GOTO 5300
5280          IF  KK <> 1 AND   KK <> L  THEN LX=ROL+ CT+1 :GOTO 5300
5290          IF  KK  = 1 THEN  LX=ROL*(L-1)+CT+1 :GOTO 5300
5300  CT1=(CT MOD 5)*16+2 : CT2=CT¥5+5
5310   IF   LX  >= 10  THEN  5320  ELSE COLOR 4: LOCATE CT1, CT2:PRINT AP2$(LX);
 LX : LINE (8*CT1+72,16*CT2)-(8*CT1+80,16*CT2+16),1,BF : GOTO 5330
5320   COLOR 4:LOCATE CT1, CT2 :  PRINT AP2$(LX); LX : LINE (8*CT1+72,16*CT2)-(8
*CT1+96,16*CT2+16),1,BF
5330    IK$ = INKEY$ : IF IK$ = ""  THEN  5330
5340   COLOR 0:LOCATE (CT MOD 5)*16+2,CT¥5+5:  PRINT AP2$(LX)
5350   IF IK$=CHR$(13)   THEN  5460
5360 IF IK$=CHR$(&H1E) THEN IF CT<5 THEN 5430  ELSE CT=CT-5:GOTO 5430
5370 IF IK$=CHR$(&H1D) THEN IF CT=0 THEN 5430  ELSE CT=CT-1:GOTO 5430
5380 IF IK$=CHR$(&H1C) THEN IF CT>(ROL-2) OR LCT=CT+1  THEN 5430 ELSE CT=CT+1:GO
TO  5430
5390 IF IK$=CHR$(&H1F) THEN IF CT>(ROL-6) OR LCT=<CT+5 THEN 5430 ELSE CT=CT+5:GO
TO  5430
5400 '
5410   IF IK$=CHR$(&HF8) THEN  CLS 2:LINE (0,0)-(639,60),5,B:LINE (0,0)-(639,16)
,3,BF:  GOTO 5580
5420 '
5430    GOTO *CUR2
5440 '
5450 '
5460       KESC = 3
5470  LSET  DA$ = AP2$(LX)
5480  PUT  #5,  1
5490  CONSOLE  0,25,0,1  :   NLOC = LX
5500 '
5510  CLS 3 : GOSUB   *NSCROLL
5520 '
```

Fig.4 (Continued)

```
5530  FLAG=1 : RETURN  4050
5540   CLS   : PRINT
5550  IF KESC =  3 THEN  CONSOLE,,0,1:SCREEN 3,0 :COLOR 0 :RETURN *NAMESEARCH
5560  IF KESC <> 3 THEN  RETURN 3140
5570 '
5580 '
5590     CLS
5600     LINE (0,0) -( 639, 60) ,5,B
5610  PRINT "ALUMINUM DISPERSION PLATE-TYPE FU
EL DATA";
5620      PRINT "     SELECT SAMPLE NAME TO AMEND DATA  BY MOVING CURSOR AND HIT
 RETURN-KEY"
5630  PRINT "     データ確認・修正用サンプル名を選択後、リターン キーを押して下
さい。"
5640 '
5650     LINE (0,316) -( 639,360) ,5,B
5660  LOCATE 0,20 : PRINT "      FINAL RECORD NUMBER =  "; LOF(2);: LINE (220,32
0)-(260,337),3,BF :PRINT "      ***       ROLL WIDTH = "; ROL ;:LINE (500,320)-
(540,337),1,BF : PRINT "  RECORDS"
5670 PRINT  "    PUSH ROLL-UP KEY TO SEE NEXT SCREEN.  EACH RECORD NUMBER IS SHO
WN IN BLUE. "
5680 '
5690 '
5700                U = U + 1
5710   IF LOF(2) MOD ROL=0 THEN L=LOF(2)¥ROL: GOTO 5730
5720      L = LOF(2)¥ROL+ 1
5730   IF L = 1  THEN  CT=0:LCT=LOF(2): GOTO *CUR2
5740      FOR KK = 1   TO  L
5750   IF ( U+KK ) MOD L = 0 THEN  *AA
5760   NEXT   KK
5770           *AA
5780   CT = 0
5790 '
5800  IF KK <> 1 THEN GOTO 5810  ELSE  GOTO  5860
5810  FOR  LX = ROL*(L-KK)+1  TO  ROL*(L-KK)+ROL
5820    CT = CT + 1
5830  LOCATE ( (CT-1) MOD 5)*16+2,(CT-1)¥5+5 : PRINT AP2$(LX)
5840  NEXT  LX
5850  CT=0  :  LCT=ROL :GOTO  *CUR2
5860  IF KK = 1  THEN  CT=0 :GOTO  5870
5870    FOR  LX = (L-1)*ROL+1   TO  LOF(2)
5880  CT = CT + 1
5890  LOCATE ( (CT-1) MOD 5)*16+2,(CT-1)¥5+5 : PRINT AP2$(LX)
5900    NEXT  LX
5910 '
5920    CT = 0 : LCT = LOF(2)-ROL*(L-1) : GOTO *CUR2
5930 '
5940    RETURN
5950 '
5960 ' ---------------------------------------------------------------------
5970                        *DTMANIP
5980 ' ---------------------------------------------------------------------
5990 '
6000  IF  FLAG = 4 THEN  GOTO  6040  ELSE IF  FLAG = 2  THEN  GOTO  6020
6010   *XDEL
6020  CLS: PRINT " TOTAL RECORDS ［全レコード数］=" ; LOF(2);" ; " ;
6030  PRINT  "INPUT RECORD No. TO";:COLOR 2:PRINT " DELETE";:COLOR 7:PRINT" DATA
 " ;:INPUT NDL  :GOTO  6970
6040  CLOSE #6:  CLOSE #7 :  CLOSE #8
6050  KILL  "AVOID.DAT" : KILL "AUDT.DAT" : KILL "DAP1.DAT"
6060  OPEN  "AVOID.DAT"   AS  #6
6070  FIELD #6,    4 AS AWSS2$,4 AS AVVP2$,  4 AS AVV2$, 4 AS AFVFV$
```

Fig.4 (Continued)

```
6080  OPEN  "AUDT.DAT" AS #7
6090  FIELD  #7,  4 AS AUDD$,     4 AS  AUWW$
6100  OPEN  "DAP1.DAT" AS #8
6110  FIELD #8,  8  AS  ADA2$
6120  PRINT " TOTAL RECORDS [全レコード数] =" ;  LOF(2) ;" ; " ;
6130  PRINT  "INPUT RECORD No. TO";:COLOR 4:PRINT " INSERT";:COLOR 7:PRINT " DAT
A " ;: INPUT   NAP
6140  IF   NAP = 0       THEN CLS : RETURN  3150
6150  IF   NAP > LOF(2) THEN  LOCATE 0, CSRLIN-1:  GOTO  6120
6160  FOR   N = 1  TO  LOF(2)
6170 '
6180 '
6190  GET  #2,    N
6200  GET  #3,    N
6210  GET  #9,    N
6220 '
6230  IF    N < NAP    THEN  GOTO  6270
6240  IF    N >= NAP   THEN  GOTO  6310
6250 '
6260 '
6270  LSET ADA2$   = DA2$
6280  LSET AWSS2$  = WSS2$ :  LSET AVVP2$ = VVP2$ : LSET AVV2$ = VV2$
6290  LSET AFVFV$  = FVFV$ :  LSET AUDD$ = UDD$   : LSET AUWW$ =  UWW$
6300  PUT  #6,N :PUT #7,N :PUT #8, N :  GOTO  6370
6310  LSET ADA2$   = DA2$
6320  LSET AWSS2$  = WSS2$ :  LSET AVVP2$ = VVP2$ : LSET AVV2$ = VV2$
6330  LSET AFVFV$  = FVFV$ :  LSET AUDD$ = UDD$   : LSET AUWW$ =  UWW$
6340 '
6350  PUT  #6,N+1: PUT #7,N+1 : PUT  #8, N+1  : GOTO  6370
6360 '
6370  NEXT  N
6380 '
6390  GOSUB     *DATAINSERT
6400 '
6410  COLOR  3
6420 '
6430  PRINT "FOR ABANDONING TO INSERT DATA, ONLY  PUSH  RETURN- KEY ; Otherwise,
Hit Any Key."
6440  AB$  =  INPUT$(1)
6450  IF  AB$ = CHR$(&HD)  THEN  COLOR 7: CLS  :GOTO  3150
6460 '
6470 '
6480  FOR     N  =  1   TO   LOF(6)
6490           GET  #6,N
6500           GET  #7,N
6510           GET  #8,N
6520  LSET DA2$   = ADA2$
6530  LSET WSS2$  = AWSS2$ : LSET VVP2$ = AVVP2$ : LSET VV2$ = AVV2$
6540  LSET FVFV$  = AFVFV$ : LSET UDD$  = AUDD$  : LSET UWW$ = AUWW$
6550     PUT #2, N : PUT #3,N : PUT #9,N
6560  NEXT    N
6570 '
6580   CLS : FLAG = 3 : RETURN   1480
6590 '
6600 '
6610 ' ------------------------------------------------------------------
6620                        *DATAINSERT
6630 ' ------------------------------------------------------------------
6640 '
6650  NR=1   :  IF  LOF(4)  <>  0  THEN  GET  #4,  1
6660  COLOR 5:PRINT "DATA INSERT: AP$(WITHIN 8 CHARS),WS,VP,UW,UD,V,FVF (PRIOR I
NPUT SHOWN IN VIOLET)->":
```

Fig.4 (Continued)

```
6670  COLOR 3 :LOCATE 0, CSRLIN-1 : PRINT  TM$; ","; CVS(TM1$);",";CVS(TM2$); ",
"; CVS(TM3$); ","; CVS(TM4$);                          ","; CVS(TM5$);","; CVS(TM6$) :
COLOR 7
6680  '
6690  LOCATE 0, CSRLIN - 1: INPUT "", AP$, WS, VP, UW, UD, V, FVF
6700  LSET TM$= AP$:LSET TM1$= MKS$(WS):LSET TM2$= MKS$(VP):LSET TM3$= MKS$(UW)
     :LSET  TM4$=MKS$(UD): LSET TM5$ =MKS$(V) : LSET TM6$= MKS$(FVF)
6710    PUT  #4,  1
6720 '
6730 '
6740  IF LEN(AP$) > 8  THEN  GET  #4,  1 ELSE  GOTO   6820
6750  LOCATE 0, CSRLIN - 2
6760  PRINT "DATA INSERT : AP$(WITHIN 8 CHARS),WS,VP,UW,UD,V,FVF ->
            "
6770  LOCATE 0, CSRLIN - 1: INPUT "", AP$, WS, VP, UW, UD, V, FVF
6780 '
6790  LSET TM$= AP$:LSET TM1$= MKS$(WS):LSET TM2$= MKS$(VP):LSET TM3$= MKS$(UW)
     :LSET TM4$=MKS$(UD): LSET TM5$ =MKS$(V) : LSET TM6$= MKS$(FVF)
6800    PUT  #4,  1
6810  LOCATE 0, CSRLIN  : GOTO  6740
6820  L=LEN(AP$) :  AP$ = AP$ + SPACE$(8-L)
6830  IF  NR > LOF(2) THEN  GOTO 6880 ELSE  GOTO  6840
6840     GET  #2,  NR
6850  IF  DA2$ <> AP$ THEN  NR= NR+1 : GOTO  6830  ELSE  GOTO 6860
6860    COLOR 3 : PRINT "    SAME SAMPLE NAME ALREADY PRESENT ! " : COLOR 7 :
        GOTO 3140
6870 '
6880  LSET ADA2$ = AP$
6890  LSET AWSS2$  = MKS$(WS) :  LSET AVVP2$ = MKS$(VP): LSET AVV2$ = MKS$(V)
6900  LSET AFVFV$  = MKS$(FVF):  LSET AUDD$  = MKS$(UD): LSET AUWW$ = MKS$(UW)
6910  PUT   #8,  NAP: PUT #7, NAP: PUT #6,  NAP
6920 '
6930  RETURN
6940 '
6950 ' ///////////////////////  DATA  DELETE  ////////////////////////
6960 '
6970  IF   NDL = 0  THEN CLS : RETURN  3150 ELSE  GOTO 6980
6980  IF   NDL > LOF(2) THEN  LOCATE 0, CSRLIN-1: GOTO *XDEL
6990  FOR   N = 1  TO  LOF(2)
7000 '
7010  IF FLAG=2 AND N=NDL  THEN  GOTO 7170
7020  GET  #2,    N
7030  GET  #3,    N
7040  GET  #9,    N
7050 '
7060  IF FLAG=2 AND N>NDL THEN GOTO 7110
7070  LSET AADA2$   = DA2$
7080  LSET AAWSS2$  = WSS2$ :  LSET AAVVP2$ = VVP2$ : LSET AAVV2$ = VV2$
7090  LSET AAFVFV$  = FVFV$ :  LSET AAUDD$  = UDD$  : LSET AAUWW$ = UWW$
7100  PUT  #10,N :PUT #11,N :PUT #12, N :  GOTO  7170
7110  LSET AADA2$   = DA2$
7120  LSET AAWSS2$  = WSS2$ :  LSET AAVVP2$ = VVP2$ : LSET AAVV2$ = VV2$
7130  LSET AAFVFV$  = FVFV$ :  LSET AAUDD$  = UDD$  : LSET AAUWW$ = UWW$
7140 '
7150 '
7160      PUT  #10,N-1: PUT #11,N-1 : PUT  #12, N-1
7170  NEXT  N
7180 '
7190  CLOSE :  RUN  "F_DELETE"
7200 '
7210 '   F_DELETE.BAT : A BATCH FILE NEEDED TO HANDLE FILE COPY :
7220 '                  _NAME.TMP --> _NAME.UPL ; _UDT.TMP --> _UDT.UPL ;
```

Fig.4 (Continued)

```
7230 '                    _VOID.TMP --> _VOID.UPL  WITH SUBSEQUENT DELETION
7240 '                    OF _NAME.TMP, _UDT.TMP, _VOID.TMP
7250 '                    AND FINALLY, TO REBOOT  UPL.EXE.
7260 '
7270 '
7280 ' ///////////////////////  DATA  DELETE  ////////////////////////////
7290 '
7300 '
7310 '-------------------------------------------------------------------
7320                          *CANCEL
7330 '-------------------------------------------------------------------
7340 '
7350    CLS  : PRINT
7360  IF KESC =  3 THEN  CONSOLE,,0,1:SCREEN 3,0 :COLOR 0 :RETURN *NAMESEARCH
7370  IF KESC <> 3 THEN  RETURN 3140
7380 '
7390 '
7400 ' ///////////  E N D  O F  F I L E  ////////////////
```

※ £ indicates Escape Sequence.

```
 £16;8m
ECHO  £>1h
CLS
ECHO  £>5h
CLS
COPY _*.TMP  _*.UPL
DEL  _*.TMP
ECHO OFF
ECHO  £m
ECHO RECORD  DELETED  (レコード削除完了)
ECHO  £>5l
UPL
```

Fig.4 (Continued)

```
[ S A M P L E   D E S C R I P T I O N ]

1PE4B    : MINIPLATE => FUEL : SINGLE-PHASED U3Si2, Heat-treated at 850 °C x
2 DAYS. FUEL COMPACT DIMENSION before ROLL = 3mmx5mmx2.14mm,  PRESSED AT 3.7 ton
/cm²   FRAME: AL6061(AL-1.0Mg-0.61Si-0.31Cu), Hot Rolled at 500 °C, PLATE SIZE:
30mmx20mmx1.4mm                                   DoC = 91/03/24  14:47:01

LIQUID TEMP (℃)=  24.3     ** DL : DENSITY OF LIQUID (g/cc)=  .86
AIR    TEMP (℃)=  24.3     ** DA : DENSITY OF AIR    (g/cc)=  1.18713E-03

WS   (WEIGHT OF SAMPLE IN AIR, g )     =  2.3059
WBL  (WEIGHT OF BUCKET IN LIQUID, g ) =  43.8652
WSBL (WEIGHT OF SAMPLE + BUCKET IN LIQUID, g ) =  45.4747
WSL  (WEIGHT OF SAMPLE IN LIQUID, g ) =  1.6095

 DS  (DENSITY OF SAMPLE in g/cc)       =   2.845        g/cc

 VS  (VOLUME OF SAMPLE in cc)          =   .810886      cc
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -


[ M I N I - P L A T E   D A T A ]

1PE4B    : MINIPLATE => FUEL : SINGLE-PHASED U3Si2, Heat-treated at 850 °C x
2 DAYS. FUEL COMPACT DIMENSION before ROLL = 3mmx5mmx2.14mm,  PRESSED AT 3.7 ton
/cm²   FRAME: AL6061(AL-1.0Mg-0.61Si-0.31Cu), Hot Rolled at 500 °C, PLATE SIZE:
30mmx20mmx1.4mm                                   DoC = 91/03/24  14:48:10
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 URANIUM DENSITY in FUEL MEAT ( UD )        =    4.687    gU/cc
 VOID VOLUME FRACTION in FUEL MEAT ( V )    =   13.095    %
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

 VP :  VOLUME OF FUEL-PLATE,  cc =      .810886

 WS :  WEIGHT OF FUEL-PLATE,  g  =      2.3059
 UW :  WEIGHT OF TOTAL URANIUM IN FUEL-PLATE,  g  =    .142174

 WFC      : WEIGHT OF FRAME & COVER,  g =      2.1153
 WCOMPACT: WEIGHT OF FUEL COMPACT,    g =      .1906

 WFUEL : WEIGHT OF FUEL POWDER,  g  =     .15337
 WALUM : WEIGHT OF AL POWDER,    g  =     .03723

 DALUM : TD. of ALUMINUM,          g/cc =   2.70
 DFR   : TD. of FRAME(AL 6061),    g/cc =   2.71
 DFUEL : TD. of FUEL( U3Si2 ),     g/cc =   12.2

 VMTH : THEORETICAL VOLUME OF MEAT,    cc =   .0263602
 VM   : VOLUME OF ROLLED FUEL MEAT,    cc =   .0303323

 FVF :  FUEL VOLUME FRACTION IN MEAT,          %   =    41.4453

 MVF :  MIXING VOLUME OF FUEL FOR  MEAT,       %   =    47.6905
 MVA :  MIXING VOLUME OF ALUMINUM FOR MEAT,    %   =    52.3095
```

Fig. 5 Output Example of UDV. EXE (1)

```
[ S A M P L E   D E S C R I P T I O N ]

89F1A-06 : MINIPLATE => FUEL : U6Fe0.6Mn0.4 with trace amounts of U-PHASE,
Heat-treated at 670°C x 4.5 DAYS. Measured density= 17.6 g/cc, COMPACT DIMENSION
before ROLL = 3mmx5mmx2.02mm, PRESSED at 6.6 ton/cm²,Hot Rolled at 500 °C, SIZE:
30mmx20mmx1.3mm                                   DoC = 91/03/24  13:55:55

LIQUID TEMP (℃)=  25      ** DL : DENSITY OF LIQUID (g/cc)=  .8594
AIR    TEMP (℃)=  25      ** DA : DENSITY OF AIR    (g/cc)=  1.18434E-03

WS   (WEIGHT OF SAMPLE IN AIR, g )     =  2.3152
WBL  (WEIGHT OF BUCKET IN LIQUID, g ) =  43.3774
WSBL (WEIGHT OF SAMPLE + BUCKET IN LIQUID, g ) =  45.0121
WSL  (WEIGHT OF SAMPLE IN LIQUID, g ) =  1.6347

 DS  (DENSITY OF SAMPLE in g/cc)       =   2.921          g/cc

 VS  (VOLUME OF SAMPLE in cc)          =   .792923        cc

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -


[ M I N I - P L A T E   D A T A ]

89F1A-06 : MINIPLATE => FUEL : U6Fe0.6Mn0.4 with trace amounts of U-PHASE,
Heat-treated at 670°C x 4.5 DAYS. Measured density= 17.6 g/cc, COMPACT DIMENSION
before ROLL = 3mmx5mmx2.02mm, PRESSED at 6.6 ton/cm²,Hot Rolled at 500 °C, SIZE:
30mmx20mmx1.3mm                                   DoC = 91/03/24  13:56:35
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 URANIUM DENSITY in FUEL MEAT ( UD )        =    5.980    gU/cc
 VOID VOLUME FRACTION in FUEL MEAT ( V )    =   15.588    %
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

 VP :  VOLUME OF FUEL-PLATE,  cc =      .792923

 WS :  WEIGHT OF FUEL-PLATE,  g  =      2.3152
 UW :  WEIGHT OF TOTAL URANIUM IN FUEL-PLATE,  g  =   .205697

 WFC      : WEIGHT OF FRAME & COVER,  g =      2.0556
 WCOMPACT: WEIGHT OF FUEL COMPACT,    g =      .2596

 WFUEL : WEIGHT OF FUEL POWDER,  g  =    .2136
 WALUM : WEIGHT OF AL POWDER,    g  =    .046

 DALUM : TD. of ALUMINUM,          g/cc =   2.70
 DFR   : TD. of FRAME(AL 6061),    g/cc =   2.71
 DFUEL : TD. of FUEL( U6Me ),      g/cc =   17.8

 VMTH : THEORETICAL VOLUME OF MEAT,    cc =   .029037
 VM   : VOLUME OF ROLLED FUEL MEAT,    cc =   .0343993

 FVF :  FUEL VOLUME FRACTION IN MEAT,          %   =    34.8844

 MVF :  MIXING VOLUME OF FUEL FOR  MEAT,       %   =    41.3265
 MVA :  MIXING VOLUME OF ALUMINUM FOR MEAT,    %   =    58.6735
```

Fig. 5 Output Example of UDV. EXE (2)

(a)

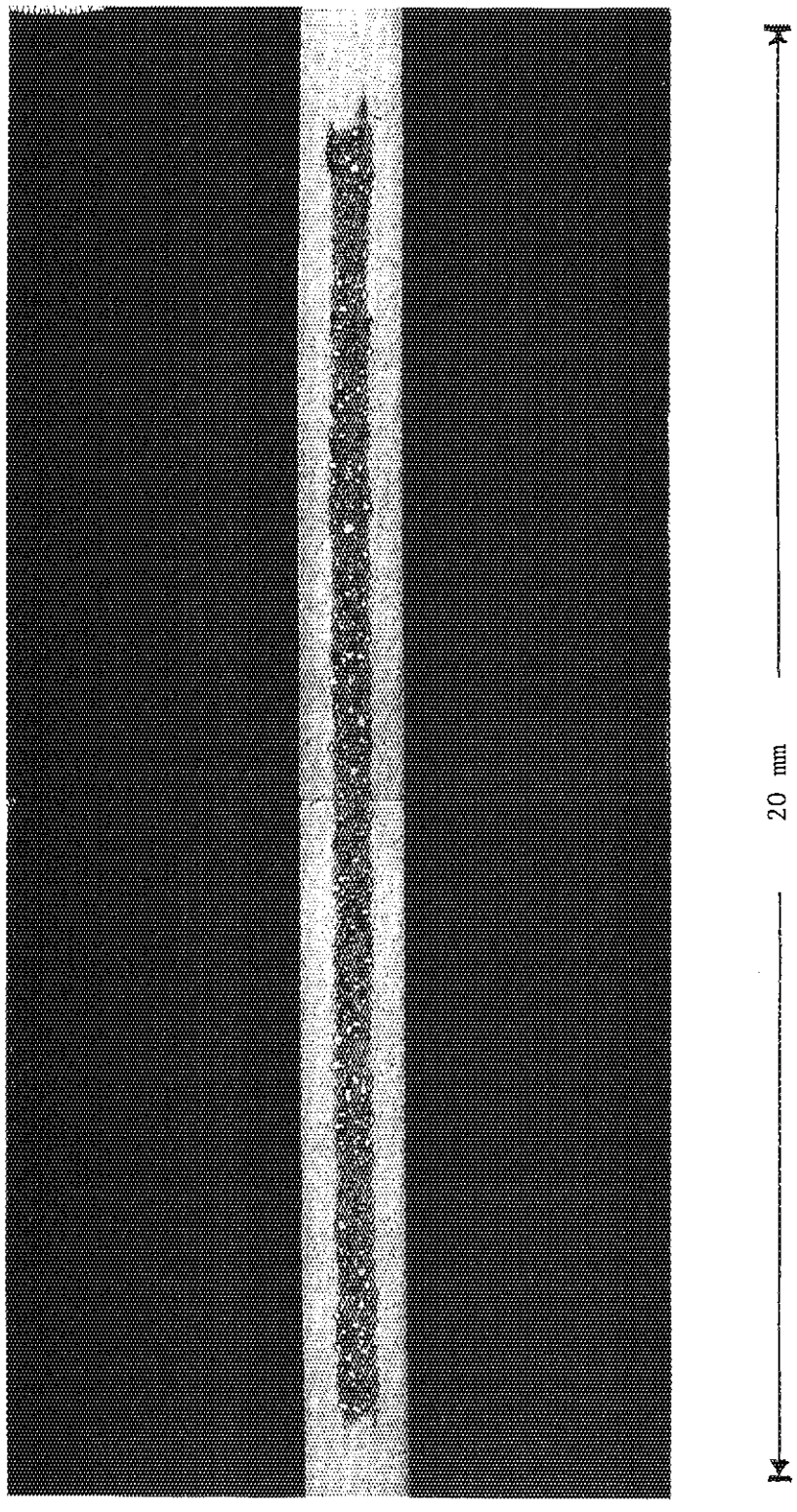

Fig. 6 Metallographic Cross-section (rolling direction) of a Miniplate[10], 1PE4B, same as shown in Fig.5(1)

(b) Partly Enlarged

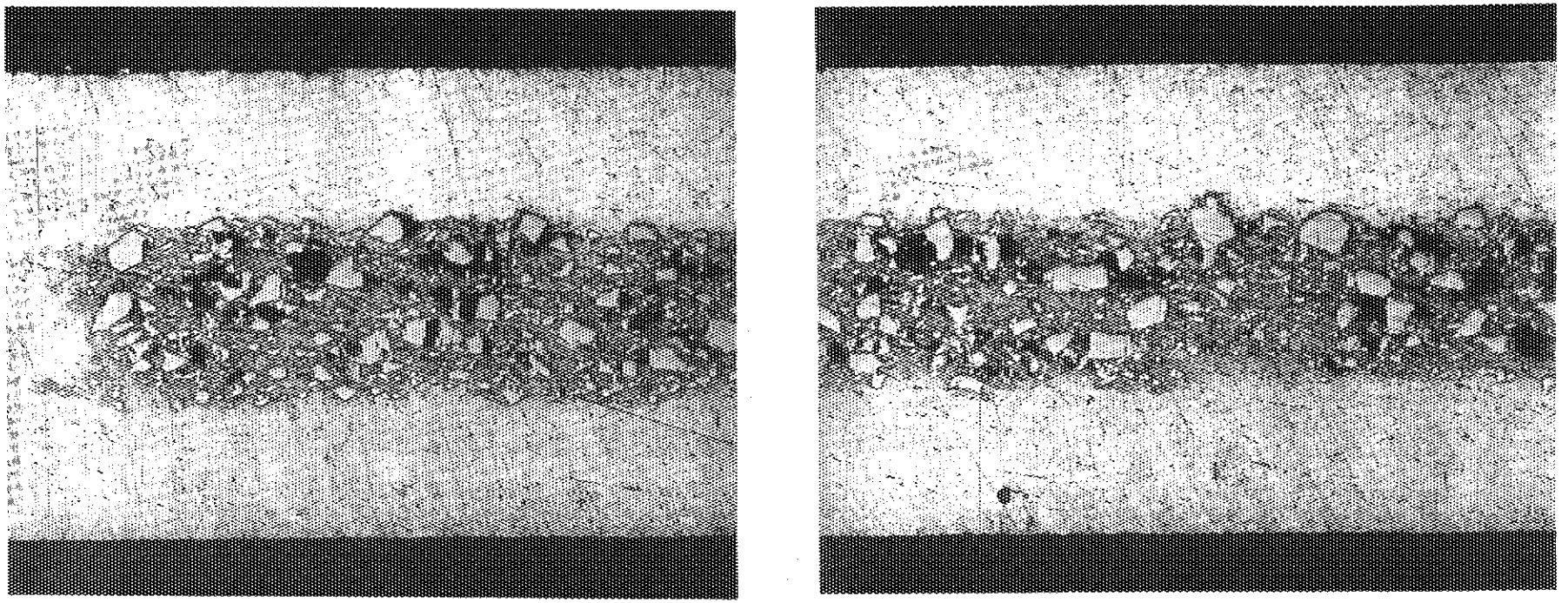

PLATE THICKNESS: 1.4 mm
FUEL MEAT : U_3Si_2 / AL
FRAME & COVER : AL 6061

Fig.6 (Continued)