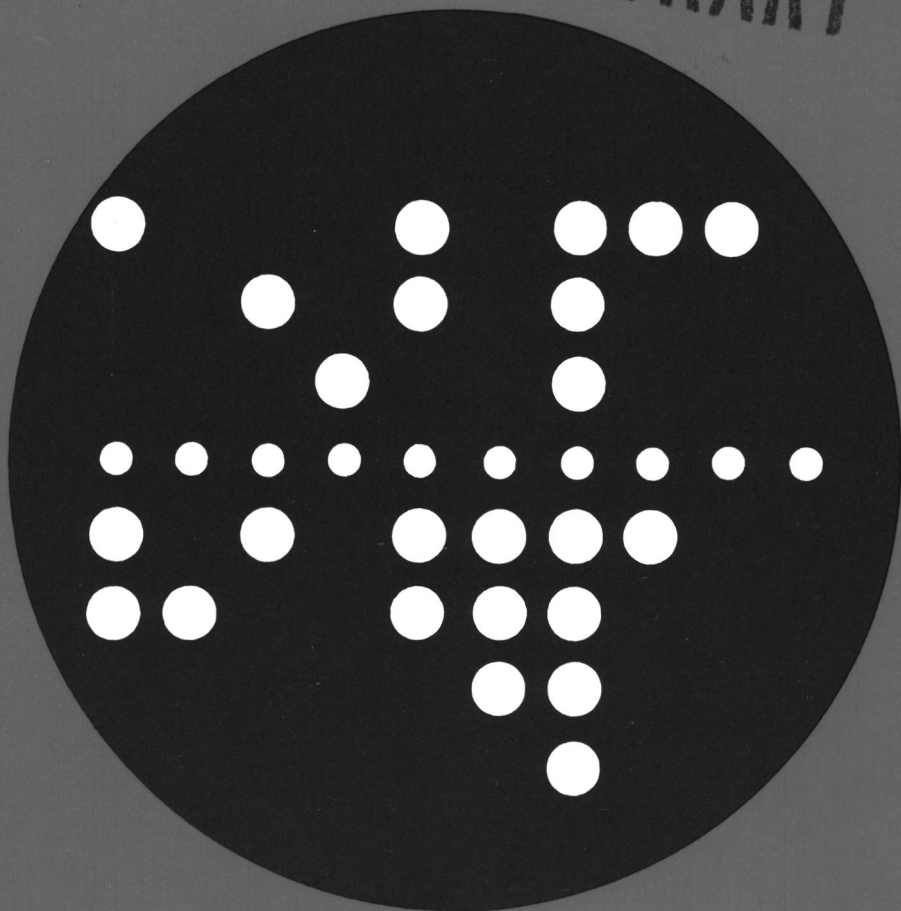


COMPUTING CENTRE NEWSLETTER

May 1980 - N. 41

LIBRARY



Commission of the European Communities



JOINT
RESEARCH
CENTRE

Ispra Establishment

CONTENTS

Editorial Note	2
Easy Graphics	3
NAG Note	9
Use of Partitioned Data Sets	10
Statistics of Computing Installation, April	16
Utilisation by Objectives & Accounts, April	17
Statistics of Batch Processing, April	18
Histogram of Equivalent Time Usage	18
List of Personnel	19

EDITORIAL NOTE.

The Computing Centre Newsletter is published monthly except for August and December.

It describes developments, modifications and specific topics in relation to the use of the computing installations of the Joint Research Centre, Ispra Establishment.

The aim of the Newsletter is to provide information of importance to the users of the computing installations, in a form which is both interesting and readable.

The Newsletter also includes articles which are of intellectual and educational value in order to keep the users informed of new advances in computer science topics.

The Editorial Board is composed as follows:

J. Pire.	Responsible Editor.
M. Dowell.	Technical Editor.
C. Pigni.	Editors.
H. de Wolde.	

Administration and contact address:

Ms. A. Cambon (tel. 730)
Support to Computing
Building 36
J.R.C. Ispra Establishment
21020-ISPRA (Varese)

LEGAL NOTICE:

Neither the Commission of the European Communities nor any person acting on behalf of the Commission is responsible for the use which might be made of the information in this Newsletter.

EASY GRAPHICS

Herman I. de Wolde

This article has been written for programmers who wish to start using the graphic facilities of the Computer Centre. It describes the actions and the related documents which are necessary for a smooth entry into this field. Secondly it gives a description of the X-collection of graphic subroutines which allow for easy 2D (two dimensional) plotting of functions and histograms in A4 and A5 format.

Graphic devices

The following graphic devices are available:

- Tektronix 4015 terminals.
These are mainly to be used during the development of programs for the testing of the graphic output.
- Gould 5200 electrostatic plotter.
This plotter is used for the final drawing of large production.
The device is very quick but not highly accurate.
- Benson penplotter.
This penplotter is used for final drawings where high accuracy is requested.

Intermediate File

To allow for a complete free choice of the output device, without any modification of the application program, the graphic output is stored, numerically, in an Intermediate File. This Intermediate File may be scanned by a Tektronix 4014 or 4015 terminal, or may be converted to magnetic tape format for an off-line interpretation by the Gould - or Benson plotter. The tape formats of these two plotters are not identical. Your personal Intermediate File may be created, under TSO, by:

```
CREATES name USEPOX RF1(V) RF2(S) LRFCL(300) BLKSIZE(804)
```

followed by the reservation procedure.

name is second part of the fully qualified name of your file.
(ie the data set created will be named TSOxxx.name)
USEROX is the volume on which the data set is to be created.

Reference: Green Book "CPADHIT".

The application program

The application program may contain in the most simple form:

- Some function calculation
- CALL GSTART as the first call for graphic output
- Use of graphics routines (see section "The Graphics Library")
- CALL GEND to close the intermediate file

Execution of the program in batch

For the execution of the program in batch, the deck composition is as follows:

```
//      JOB(your job card)
//      EXEC FTG1CLG,PRN=ERTY,VLB=COPICB,ULB=DISK
//CHP.SVSIN DD *
//      FOPTRAN deck
/*
//GO.FT16F001 DD DSN=name,UNIT=DISK,DISP=(OLD,KEEP)
//GO.SYSIN   DD *
//      input data
/*
```

in which name is now the fully qualified name of the intermediate file.

Note. It is not necessary to specify the volume if the data set has been created catalogued by the previously defined CREARES procedure. If the data set has been created in a different way and not catalogued, it will be necessary to specify the "VOL=SER=....." parameter.

Execution in TSO foreground

For the execution of a FOPTRAN program with graphic output, you may prepare the following procedure, plot.CLIST:

```
PROC 0
FREEALL
ALLOC DA(filename) FILE(FT16F001)
ALLOC DA(*) FILE(SYSPRINT)
LOADGO progr FORTLIB LIB('SYS1.LIBEPTY')
FREEALL
END
```

in which:

plot.CLIST is the name of the procedure
filename is the name of your intermediate file (without the TSOxxx qualifier)
progr is the object deck of the program

The plot and progr data sets must have the respective CLJST and OBJ qualifiers.

Once this procedure has been prepared you may produce the graphic output for this intermediate file simply by the command:

```
EXEC plot
```

Inspection of the Intermediate File on a Tektronix terminal

The available commands for the screening of your graphic output on a Tektronix terminal, have been described in the Newsletters no. 31 (May 1979) and no. 37 (January 1980)

Output on the Gould plotter

The output on the Gould plotter may be obtained by the following batch job:

```
//      JOB(your job card)
$OC TP9=GRxxxx,SL,Y
$OC M=GOULD T=GRxxxx ABEND=NO
//      EXEC GOULD,TAPE='GPxxxx'
//GO.FT15F001 DD DSN=filename,
//      DISP=(OLD,KEEP),UNIT=DISK
//GO.SYSIN DD *
      1.0
/*
```

in which:

GRxxxx is your graphic output tape number
filename is the intermediate file (fully qualified name)

X-collection

The X-collection is a series of subroutines, especially developed for easy graphics production. In the future this series will be extended according to the needs.

Images are numbered automatically and origin replacements are supplied by this system.

The routines from the X-collection may be mixed with the other routines out of SVS1.LIBERTY.

Presently only 2D graphs and histograms, in A4 and A5 format are possible.

Specification of X-collection Subroutines

GSTART Mandatory first call for graphic applications

XGRAF4 (X,Y,N,K,L)
X and Y are two arrays containing the coordinates
N is total number of points
K=-1 X axis logarithmic, Y axis logarithmic
K=0 X axis normal, Y axis normal
K=1 X axis logarithmic, Y axis normal
K=2 X axis normal, Y axis logarithmic
L=0 horizontal A4 format
L=1 vertical A4 format

XGRAF5 (X,Y,N,K,L) Equivalent to XGRAF4 but for A5 format

XGRAF (X,Y,N)
X and Y are two arrays, N points, of a curve which will be added to the last image by a XGRAF4 or XGRAF5 call

XHIST4 (Y,N,K,F,XL,XH,L) This routine produces a histogram
Y is an array containing the column lengths
N is the number of columns
K=0 Y axis is normal
K=1 Y axis is logarithmic
F is the width of the column on the interval; (F=1 will give a normal bar-graph)
XL X value low, minimum X value
XH X value high, maximum X value
L=0 horizontal A4 format
L=1 vertical A4 format

XHIST5 (Y,N,K,F,XL,XH,L) Equivalent to XHIST4 but with A5 format

GEND Mandatory last call for graphic applications

The following example gives an illustration of the use of the X-collection.

```
C      DEMONSTRATION PROGRAM LIBRARY LIBERTY, SUB-SET X-COLLECTION
C      DIMENSION X(100),Y(100)
C      THE CALL TO GSTART IS MANDATORY FOR GRAPHICS APPLICATIONS
C      CALL GSTART
C      CONSTRUCT A FUNCTION
C      NPOINT=50
C      DO 100 I=1,NPOINT
C      X(I)=FLOAT(I)
100  Y(I)=X(I)**2+5.0
C      DRAW A GRAPH A4 FORMAT HORIZONTAL, LINEAR X, LINEAR Y
C      CALL XGRAF4(X,Y,NPOINT,0,0)
C      DRAW A GRAPH A5 FORMAT VERTICAL, LINEAR X, LOGARITHMIC Y
C      CALL XGRAF5(X,Y,NPOINT,2,1)
C      CONSTRUCT A SECOND FUNCTION
C      DO 120 I=1,NPOINT
120  Y(I)=0.5*X(I)
C      ADD THE SECOND FUNCTION TO THE LAST GRAPH A5 VERTICAL
C      CALL XGRAF(X,Y,NPOINT)
C      DRAW A HISTOGRAM A4 FORMAT HORIZONTAL, LINEAR Y
C      CALL XHIST4(Y,10,0,0.7,0.0,10.0,0)
C      DRAW A HISTOGRAM A5 FORMAT VERTICAL, LOGARITHMIC Y
C      CALL XHIST5(Y,20,1,1.0,0.0,20.0,1)
C      THE CALL TO GEND IS MANDATORY TO END GRAPHICS APPLICATION
C      CALL GEND
C      STOP
C      END
```

See Annex 1 for example output.

The Graphics Library

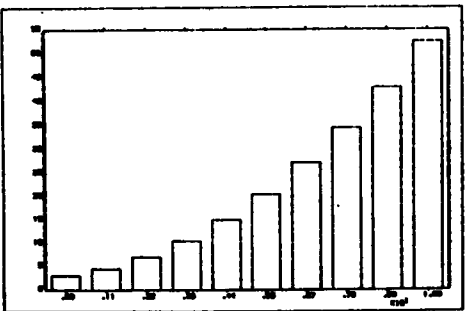
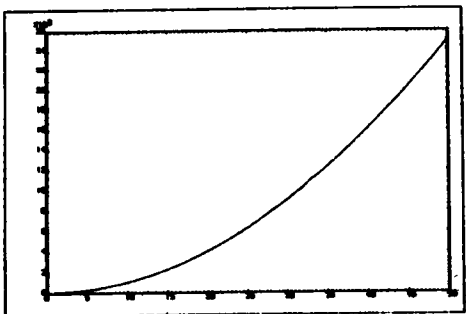
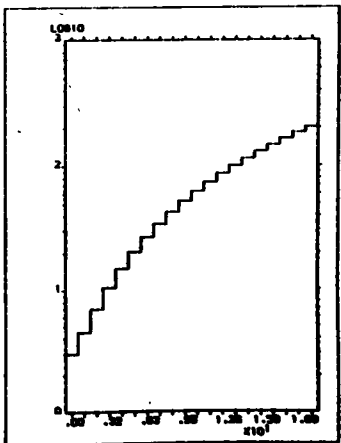
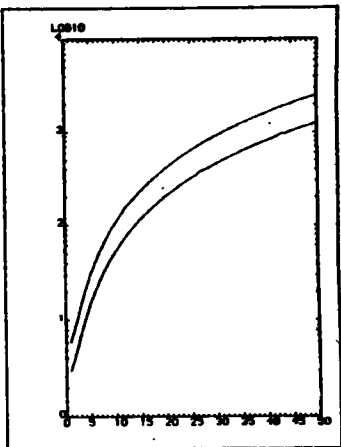
The SYS1.LIBERTY library, residing on volume COPIC3, which forms the collection of all the graphics subroutines, contains element from various sources:

- GINO-F, for 2D and 3D plotting with transformations
- GINOGRAF, for 2D curve plotting, histograms, barcharts, piecharts, etc.
- SYS1.LIBGRAPH, the GRAPHIT system with routines to write to the intermediate file
- Several interesting routines from various sources (CARPET, GREYSI, etc.)
- X-collection, a group of routines with very simple application aspects

For the use of GINO-F and GINOGRAF, the programmers may consult the manuals in the Computing Support Library.

Annex 1

Sample output of the four pictures in the previous example as produced on the Tektronix 4015 hard copy (not to scale).



```
GRAPHIT VERSION NOVEMBER 1979
summary
NUMBER OF COMMANDS      3841
FILE      XMIN  XMAX  YMIN  YMAX
100      -0.01  91.11 -0.01  21.11
101      -0.01  30.21 -0.01  21.11
150      30.29  45.56  0.09  21.11
151      45.64  75.76  0.09  21.11
151      75.84  91.11  0.09  21.11
noframe
enhanced
screen 4
display 1 100 n
display 2 101 n
display 3 102 n
SYNTAX ERROR, RE-ENTER GO
display 3 150 n
display 4 151 n
end
```

NAG NOTE

M. Dowell

In the article "The NAG Library is Available" in Newsletter no. 38. (February 1980) information was given concerning the use of the NAG Library.

One piece of information was omitted from this document concerning the use of the single-precision library. This information is now given in the following section.

Use of the NAG Single-precision library

Users of the single-precision library should note that, when calling a NAG single-precision routine, the name that should be used is not the same as specified in the NAG Library manual. For single precision users the final character of the routine name should be changed from F (which is for the double precision library) to E.

Example

For double precision:

```
CALL E04CGF(N,X,F,IW,LIW,W,LW,IFAIL)
```

For single precision:

```
CALL E04CGE(N,X,F,IW,LIW,W,LW,IFAIL)
```

USE OF PARTITIONED DATA SETS

M. Dowell

Partitioned data sets are sets of sequential data sets residing on a direct access device (disk) which have an internal organization structure (partitioned organization). This organization structure enables independent groups of sequentially organized data sets (each called a member) to be stored within the data set, each of which is identified by a member name. At the beginning of the data set there is a directory which contains a list of the member names and the starting addresses for each member in the directory. This directory is maintained in alphabetic order. Member names may be 1-8 characters, the first character must be alphabetic or a national (@, \$, #) character, the remaining characters may be alphanumeric or national or a hyphen. Just as sequential data sets are referred to by their data set name, for example:

XYZ or ABCDEFGH.FORT or TSOXXX.AB-D.COBOL

also the members of a partitioned data set are referred to by using the data set name followed by the member name in brackets, for example:

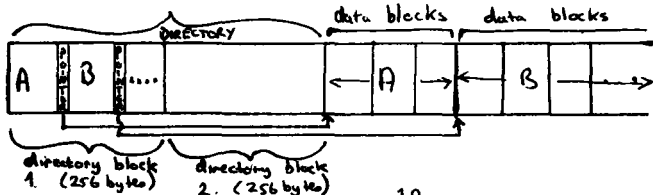
PQR.FORT(TEST-1) or TSOXXX.ABE.COBOL(A1234567)

All members within a partitioned data set automatically have the same attributes (record format, block length, record length) as that which was set up for the whole data set when it was created.

The directory of a partitioned data set is stored in directory blocks at the beginning of the data set. These directory blocks do not, in general, have the same length as the data blocks in the data set but have a fixed block length of 256 bytes.

When a partitioned data set is created it is necessary to specify the number of directory blocks which are to be allocated. Unlike the actual data area, it is not possible to extend this directory when all of the space allocated has been utilized. Therefore, it is necessary to initially allocate sufficient directory blocks to enable all relevant information for all members to be stored.

The following schematic diagram shows a partitioned data set with 3 directory blocks containing two members A & B



For each directory there is a pointer to the start of the actual data blocks.

The amount of space required for storing information about a member is not fixed, but depends on the type of data being stored in the partitioned data set. The following two examples give the most commonly used formats:

- 1) For a partitioned data set containing members which are load modules (i.e. a "load module library") with the following characteristics:

Recording format: U
block size: 13030 bytes

It is possible to store 7 entries in each directory block (but only 6 in the final block). Therefore, in a partitioned data set, containing a load module library, with 10 directory blocks it is possible to have a maximum of 69 members.

- 2) For a partitioned data set containing source programs/data with the following characteristics:

Recording format: FB
block size: 3120 bytes
logical record length: 80 bytes

It is possible to store 21 entries in each directory block (but only 20 in the final blocks). Therefore, in a partitioned data set containing source text with 3 directory blocks it is possible to have a maximum of 62 members. If your data set members have alternate names (called aliases) [1] then records containing information about the aliases will be stored in the directory and consequently there will be less space for actual member records.

Recommended Use of Partitioned Data Sets

Partitioned data sets offer a useful way of storing a logically grouped set of data sets in a manner which enables them to be maintained and used as one unit.

* A library of load modules is the use which is most common and this is obviously a practical way of grouping together a set of load modules which are, for instance, various different programs which together form a suite of programs.

* Another valid use of partitioned data sets is that of grouping together libraries of source program subroutines.

- * On TSO partitioned data sets may be used to store libraries of command procedures and also libraries of jobs suitable for use with the SUBMIT command.

An example of the use in this way is as follows:

Suppose we have three TSO command procedures for:

- a) executing the IEHLIST utility,
- b) deleting block record in a data set,
- c) executing a load module to perform some statistical analysis

then it is possible to create a partitioned data set on TSO named PROG,CLIST by using the CREARES command.

```
CREARES PROG,CLIST USER0X DIR(1)
```

This will create (and reserve for 6 months for the proper user) a data set on disk volume USER0X (this should be one of the disks USER0A/.../USER0F) with partitioned organization and:

- * 1 directory block (sufficient to hold 20 entries)
- * recording format = FB (fixed blocked)
- * block size = 3120 bytes
- * logical record length = 80 bytes

It is then possible to set the members (for example):

```
IEHLIST      DELBLK      STATS
```

by using the EDIT(QED) command or the COPY command.

To execute the command procedure it is necessary to type (for example):

```
EXEC PROG(DELBLK) 'list of parameters'
```

Thus, the data sets have been grouped together in a sensible logical arrangement.

It is, also, much easier to make a magnetic tape back-up (using IEHMOVE) of a partitioned data set and all of its members than to back-up individually a number of separate sequential data sets.

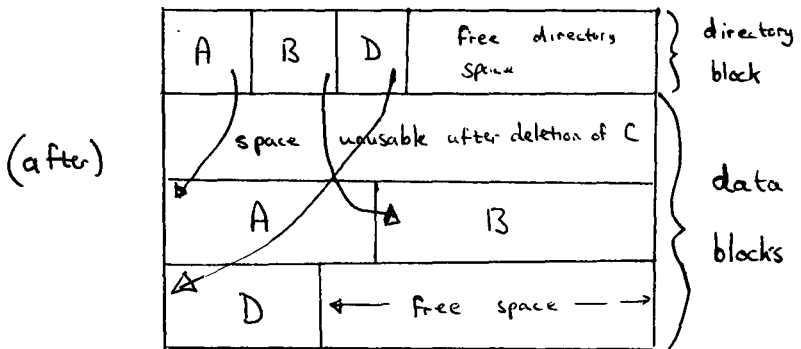
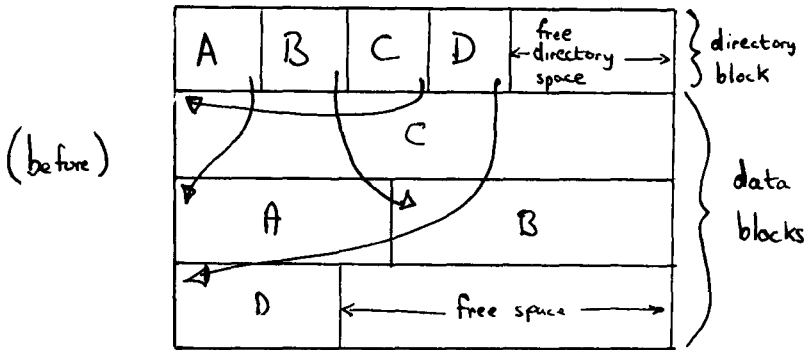
Therefore, the use of partitioned data sets in this way gives an organizational ability to the user which is especially suitable for TSO users.

However, it is not recommended to use partitioned data sets for very large collections of information. (For example a group of 100 FORTRAN subroutines). For such large groups of information users should consider the use of the LIBRARIAN package [2].

Also, as previously stated, it is not possible to group together as member of a partitioned data set information which must be stored in different formats. The format of all of the members is defined when the data set is created.

COMPRESSING Your Partitioned Data Sets

When a member of a partitioned data set is deleted the directory record for the member is nullified and the directory is reorganized to eliminate the nullified record. However, there is no housekeeping process performed at that time to recuperate the data blocks associated with the deleted member. The following schematic diagram shows the situation before and after member C is deleted:



The directory has been modified to eliminate the entry for C, but the data blocks occupied by C have not been recovered for free space and are at present unusable.

On TSO, when a member of a partitioned data set is EDITed the mechanism for SAVEing an updated version of the member is performed by inserting the new copy and then deleting the old version. Thus, this system will generate unusable data blocks as described in the example above.

It is obviously necessary to have some mechanism for performing a housekeeping action on a partitioned data set to recuperate such data blocks. This action is performed by COMPRESSing the data set. It is possible to perform the compress action either in batch or during a TSO session.

COMPRESS during a TSO Session

The action is performed by using the COMPRESS command procedure. This has the general form:

```
COMPRESS [dsname]
```

where: dsname is the name of the partitioned data set to be COMPRESSED. Full details of the use of this command on TSO may be found in [3].

COMPRESS in Batch

In batch the action is performed by executing the COMPRESS procedure. An example of the use is:

```
//          JOB(your job card)
$          TIME 002
//STEP1 EXEC COMPRESS,DSN='dsname',VOL=user0x
```

where:

dsname is the fully qualified data set

user0x is one of the user disk volumes (USER0A/.../USER0F)

Full details of the use of COMPRESS in batch may be found in [4].

USERS SHOULD REGULARLY COMPRESS ANY PARTITIONED DATA SETS WHICH THEY ARE USING (I.E. THEY ARE MODIFYING) BECAUSE:

- * SPACE RECUPERATED IN THIS WAY ENABLES THEM TO INSERT EXTRA INFORMATION INTO THE DATA SET.
- * MORE ECONOMICAL USE OF SPACE ENABLES SPARE SPACE WHICH HAS BEEN RECUPERATED BY COMPRESS TO BE MADE AVAILABLE TO OTHER USERS. (BY USE OF A SPECIAL UTILITY RUN EVERY WEEK BY THE SYSTEM).

References

- [1] IBM System/360 Operating System
"Data Management Services" (GC26-3746-0)
- [2] JRC "green book" - The LIBRARIAN
- [3] JRC Newsletter, no. 36 (November 1979), page 5
- [4] Installation Notes, UTIL - page A.2-1

Statistics of computing installation utilization.
 Report of computing installation exploitation
 for the month of April 1980.

YEAR 1979 YEAR 1980

General

Number of working days	18 d	19 d
Work hours from 8.00 to 24.00 for	16.00h	16.00h
Duration of scheduled maintenance	18.00h	15.68h
Duration of unexpected maintenance	4.00h	22.53h
Total maintenance time	22.00h	38.21h
Total exploitation time	266.00h	290.64h*
CPU time in problem mode	122.35h	192.80h

Batch Processing

Number of jobs	6817	7526
Number of cards input	1363000	1197000
Number of lines printed	21895000	23687000
Number of cards punched	108000	145000
CPU time	107.22h	170.66h
Number of I/O (Disk)	19761000	19270000
Number of I/O (Magnetic tape)	3303000	3924000

T.S.O

Number of LOGON's	2663	3427
Number of messages sent by terminals	139600	221000
Number of messages received by terminals	684000	1271000
CPU time	13.25h	20.30h
Number of I/O (Disk)	2031000	3047300
Connect time	1781.47h	2429.88h

IMS

Total time service is available	129.50h	104.90h
CPU time	1.88h	1.84h
Number of I/O (Disk)	405000	388500

* This figure includes 24.50h of overtime.

Utilisation of computer centre by objectives and appropriation
accounts for the month of April 1980.

IBM 370/165
equivalent time in hours

1.20.2	General Services - Administration - Ispra	44.98
1.20.3	General Services - Technical - Ispra	0.40
1.30.3	Central Workshop	2.97
1.30.4	L.M.A.	-
1.90.0	ESSOR	32.27
1.92.0	Support to the Commission	5.37
2.10.1	Reactor Safety	246.72
2.10.2	Plutonium Fuel and Actinide Research	5.80
2.10.3	Nuclear Materials	8.77
2.20.1	Solar Energy	0.01
2.20.2	Hydrogen	0.37
2.20.4	Design Studies on Thermonuclear Fusion	9.22
2.30.0	Environment and Resources	18.87
2.40.0	METRE	2.10
2.50.1	Informatics	29.27
2.50.2	Training	-
2.50.3	Safeguards	8.89
	TOTAL	416.01
1.94.0	Services to External Users	14.10
	TOTAL	430.11

BATCH PROCESSING DISTRIBUTED BY REQUESTED CORE MEMORY SIZE

	100	200	300	400	600	800	1000	1200	1400	>1400
No. of jobs	2276	1969	1403	1002	345	63	8	86	18	-
Elapsed time	70	145	144	231	115	13	5	73	10	-
CPU time	3.1	19.5	20.8	39.4	31.6	3.8	2.8	45.1	3.1	-
"Equiv" time	23	43	48	88	43	7	3	45	5	-
"Turn" time	1.2	3.2	3.5	5.1	5.3	11.1	7.5	14.4	15.1	-
I/O (disk)	2010	3246	3799	6416	1582	439	33	158	277	-
I/O (tape)	1981	338	257	1227	27	4	11	44	4	-

NOTE.

All times are in hours.

"Equiv" means equivalent.

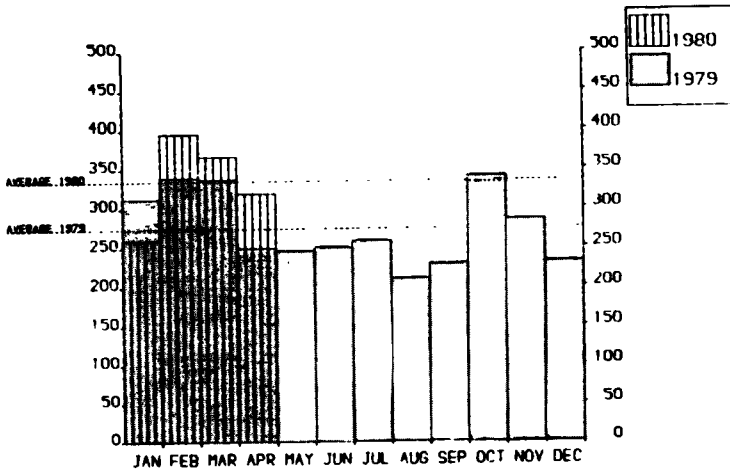
"Turn" means turn around.

All I/O transfers are measured in 1000's.

PERCENTAGE OF JOBS FINISHED IN LESS THAN

TIME	15mn	30mn	1hr	2hrs	4hrs	8hrs	1day	2day	3day	6day
%year 1979	28	42	58	73	88	97	98	100	100	100
%year 1980	23	36	51	64	78	89	97	100	100	100

HISTOGRAM OF TOTAL EQUIVALENT TIME(HRS)



Projected total for 1980 = 4018 hours (using average)

Total for 1979 was = 3292 hours

REFERENCES TO THE PERSONNEL/FUNCTIONS OF THE COMPUTING CENTRE.

<u>Manager of The Computing Centre</u>		J.Pire	
Responsible for User Registration	Ms. J.Rambs		
 <u>Operations Sector</u>			
Responsible for the Computer Room	E.Binda-Rossetti		
Substituted in case of absence by:			
Responsible for Peripherals	G.Nocera		
 <u>Systems Group</u>			
Responsible for the group	J.Konig		
Substituted in case of absence by:	P.A.Moinil		
Responsible for TSO Registration	C.Daolio		
		Room	Tele.
 <u>Informatics Support Sector</u>			
Responsible for the Sector	(f.f.) H.de Wolde	1883	1259
Secretary	Mrs. G.Huđry	1873	787
Responsible for User Support	H.de Wolde	1883	1259
General Inf./Support Library	Mrs. A.Cambon	1871	730
 <u>Advisory Service/List of Consultants(See Note 1)</u>		1870	730
A.Inzaghi	A.A.Pollicini		
	H.I. de Wolde		
R.Meelhuysen	M.Dowell		

NOTE 1. The advisory service is available in the same room as the Computing Support Library(room 1870). Exact details of the advisory service times for a specific week can be found at the head of any output listing(for that week).

Any informatics problem may be raised. However, the service is not designed to help users with problems which are their sole responsibility. For example, debugging of the logic of programs and requests for information which can easily be retrieved from available documentation.

If necessary, other competent personnel from the informatics division may be contacted by the consultant but not directly by the users.

The users should only contact the person who is the consultant for that specific day and only during the specified hours. Outside the specified hours general information may be requested from Mrs. A. Cambon in the Computing Support Library.

HOW TO OBTAIN COMPUTING CENTRE DOCUMENTATION.

Persons interested in receiving copies of the Computing Centre "green books" or in receiving regularly the "Computing Centre Newsletter" are requested to complete the appropriate part of the following form and send it to :-

Ms. A. Cambon
Support To Computing
Building 36
Tel. 730.

Indicate with a (✓) which options are required.

Please add my name to Newsletter mailing list ()

Please send me copies of the following "green books":

JRC-TSO Primer ()

GRAPHIT ()

Towards a New Programming Style ()

LIBRARIAN ()

NAME

ADDRESS

.....

.....

TELEPHONE

