The ASDEX Integrated Data Analysis System AIDA

Kai Graßie, Otto Gruber, Otto Kardaun,
Michael Kaufmann, Karl Lackner, Peter Martin,
Karl-Friedrich Mast, Patrick J. McCarthy, Vitus Mertens,
Daniela Pohl, Ursula Rang, Reinhard Wunderlich,
and the ASDEX Team

IPP 5/31 November 1989

MAX-PLANCK-INSTITUT FÜR PLASMAPHYSIK

8046 GARCHING BEI MÜNCHEN

The ASDEX Integrated Data Analysis System AIDA

K. Graßie, O. Gruber, O. Kardaun, M. Kaufmann, K. Lackner,
P. Martin, F. Mast, P.J. McCarthy, V. Mertens, D. Pohl,
U. Rang, R. Wunderlich,
the ASDEX Team

Abstract:

Since about two years, the ASDEX integrated data analysis system (AIDA), which combines the database (DABA) and the statistical analysis system (SAS), is successfully in operation. Besides a considerable, but meaningful, reduction of the 'raw' shot data, it offers the advantage of carefully selected and precisely defined datasets, which are easily accessible for informative tabular data overviews (DABA), and multi-shot analysis (SAS). Even rather complicated, statistical analyses can be performed efficiently within this system. In this report, we want to summarise AIDA's main features, give some details on its set-up and on the physical models which have been used for the derivation of the processed data. We also give short introduction how to use DABA and SAS.

1. Introduction:

About three years ago, the decision has been made to develop a unique database for processed ASDEX shot data. The main goal of this project is not only to reduce the large amount of raw data per shot to a carefully selected subset of processed data, but also to provide experimentalists with a tool that allows for advanced statistical analysis. Furthermore, it plays a role in unifying the sometimes ambiguous (i.e., user dependent) definitions and derivations of processed shot data.

From the several possible set-ups of a database we have chosen a realisation which consists essentially of two parts. One is the shot-data archive DABA containing, in a number of subfiles, a carefully selected set of processed as well as of raw ASDEX shot data. For an optimum correspondence of archive and ASDEX data structure, this part of the database has been devoloped entirely at IPP. It's structure, therefore, ideally fits to the ASDEX requirements, however at the expense of flexibility.

The second part of the system consists of the commercial statistical package SAS [1], which is coupled via interface routines to DABA (see Fig. 1). This package allows for advanced statistical multi-shot analysis, such as multiple (non-)linear regression, multivariate analysis, analysis of variance, discriminant analysis, etcetera. All these applications can easily be programmed by invoking SAS routines. Also available are flexible data management- and plot routines.

In the present status of ASDEX operation, about 4 Mbyte of raw data are produced per discharge. After processing and 'filtering' these, about 0.05 Mbyte of information is stored in DABA. Initially, our aim was to keep the data of $\sim$ 200 discharges available on disk. However, the requirements grew during time, so that now $\sim$ 1000 shots are permanently available on disk, corresponding to roughly half of our present maximum storage capacity.

In this report, we want to give an overview of what has been done so far, which data are available and how they are defined. In section 2 we discuss the global structure of the subfiles and give a brief description of the internal setup. The derivation of the corresponding processed data will be discussed in section 3. The direct and interactive use of DABA will be explained – on the basis of simple examples – in section 4. Finally, we give in section 5 a brief description of how to use SAS. In the Appendix A, an annotated list is given of the plasma variables available in the various subfiles of DABA/SAS.

2. File and program structure:

All DABA data are stored in shot files on one CMS disk (200 cylinders of type 3350). Each shot file is labelled by a prefix, from A to Z, and the shot number. The prefix denotes the version number of the data file. If data in a shot file are modified, a new version of the shot file with a different prefix is automatically created. The user always addresses the latest version only. Every shot file contains five logically separated subfiles, each of which holds physically related data. MKSA units are used as standard (temperatures and beam particle energies are in eV). In many cases, data from different subfiles do depend on each other. In these cases, if a modification of a certain quantity is performed, the DABA program either updates or deletes also the corresponding dependent data. The following information is stored (for a detailed description see Appendix A) :

Subfile 0 :

here we store time independent survey data and diagnostic information in six subgroups.
group 1 – survey data
group 2 – system information
group 3 – heating and pellet information
group 4 – diagnostic information
group 5 – characteristic parameters at three time points
group 6 – correction factors for YAG/HCN laser channels
Most of these subgroups are divided into two parts, one of which contains data that are automatically set (e.g., taken from other subfiles), whereas the other part contains data that have to be entered 'by hand'. For example, $BTMAX$ (from group 1) is automatically taken from subfile 1a, whereas $RLIM$ (also from group 1) is to be introduced by hand.

Subfile 1a :

This subfile contains time-dependent geometrical data and basic (equilibrium) data, like plasma current, toroidal magnetic field, etc. Most of this information is calculated with the function parameterisation (FP) equilibrium code $FPSTD1$ [2]. Note that subfile 1a is the first subfile to be created, as it contains the general definition of the time points at which all data (for all subfiles) are calculated. These time points are taken from the program $YAGTIM$ and correspond to the times when the YAG laser has fired into the plasma. Thus, according to the YAG-laser repetition frequency, we have time intervals of $\sim 17ms$. The time-base then consists of an uninterrupted sequence of YAG-laser time points for which $FPSTD1$-results can be determined.

Subfile 1b :

In this subfile, time-dependent scalar quantities are stored. In contrast to the data in subfile 1a, these data have a higher degree of sophistication, i.e. they are in general derived from profiles. Typical subfile 1b quantities are: volume-averaged densities, (density weighted) volume-averaged temperatures, energy confinement times, etc.

Subfile 2 :

Subfile 2 contains time dependent profiles for electron density, electron temperature, bolometrically measured radiated power densities as well as their radial derivatives. Densities and temperatures are given at nine equidistant radial positions; the radial dependence in the boundary layer region is assumed to be exponential, with fall-off coefficients that are also stored in subfile 2. The latter assumption does not hold for the radiated power density profiles, so that here we have included a tenth radial point at about 3 cm outside the separatrix. Electron temperatures and densities (HCN-corrected) are usually calculated from FP-fits to YAG data. Optionally, also profiles from $MULTIPRO$, $HCEPRO$ or combinations of these separate fitting programs can be stored. Note that the corresponding correction factors are kept in subfile 0, i.e. they are assumed to be time independent. Before storing the profiles, some checks are made whether this requirement is satisfied.

Subfile 3 :

For statistical purposes, and for comparison with the fitted profiles of subfile 2, we store in subfile 3 the original YAG data for electron temperature, electron density and their experimental errors, corresponding to the 16 YAG-laser channels.

A block diagram symbolising the above described structure is shown in Fig. 2. In addition, the programs that calculate, from the raw data, the quantities to be stored, are shown together with their main interrelationships. It can be seen that programs filling the subfiles

1a and 2 need ASDEX raw data. Hence, the discharge under consideration has to be online (or else retrieved with '$SHOTS$'). In contrast, $SF1BGEN$ only requires data which are already stored in DABA. A description of the various programs and the definition of the pertaining quantities is given in the next section.

3. Definition and calculation of processed data:

With few exceptions, data for subfile 1a are determined from the function parameterisation equilibrium code $FPSTD1$ [2]. For shot numbers below #23392 - NBI powers have to be taken form the shot journals and inserted 'by hand', whereas for later discharges the shot files directly contain the beam power. LH powers always have to be taken from the shot journals, whereas ICRH powers can be read directly from the shot file.

An important quantity is $IMODE$, indicating whether the discharge is in the Ohmic- , L- or H-phase. In the latter case, a distinction is made between 'large ELM', 'small ELM' and 'ELM-free' discharge phases. Since as yet no reliable automatic determination of $IMODE$ exists, its values have to be taken from visual inspection of characteristic H_α light traces. As a preliminary estimate, however, $IMODE$ is set automatically according to:
$IMODE = 1$, if no additional heating is applied (Ohmic phase);
$IMODE = 2$, if additional heating power below 1 MW is applied (L-phase);
$IMODE = 0$, if additional heating power above 1 MW is applied (undefined).

The quantity ILD specifies whether the machine operates in limiter ($ILD = 1$), single null top ($ILD = 2$), single null bottom ($ILD = 3$) or double null ($ILD = 4$) configuration. Usually this variable is determined automatically from the vertical shift $ZGEO$ of the geometrical axis (i.e. the center of the outermost 'circular' flux-surface) with respect to the midplane. For some critical value z_{crit},
$ILD = 2$, if $ZGEO > z_{crit}$;
$ILD = 4$, if $ZGEO < -z_{crit}$;
$ULD = 3$, for other divertor discharges.
The value of z_{crit} chosen is 0.005 m.

The calculation of the Ohmic heating power POH requires the knowledge of the loop voltage $ULOOP$ at the plasma surface, which is determined according to :

$$ULOOP = ULOOP(30) - \Delta_0 + \Delta_1 - \Delta_2 - \frac{\partial}{\partial t} FLMBR, \qquad (1)$$

where Δ_0 and Δ_1 are averaged values of the signal $UL(31)$ in the 'prehistoric' phase of the plasma discharge (before the toroidal field is switched on) and in the interval $t\epsilon[0.7s, 0.9s]$ (corresponding to the ohmic phase), respectively. Similarly Δ_2 denotes the averaged value of $ULOOP(30)$ for $t\epsilon[0.7s, 0.9s]$. The numbers in brackets correspond to the diagnostic number in the ASDEX raw data shot-file. To diminish the noisy effects of the thyristor spikes on the final signal, the raw data $ULOOP(30)$ and $UL(31)$ are smoothed by applying a simple three-point moving average

$$< X(t) >= \Big[X(t - \Delta t) + 2X(t) + X(t + \Delta t) \Big] / 4, \qquad (2)$$

where Δt is some smoothing time which is small with respect to $17ms$. To determine the time derivative of the flux difference, we fit a spline to the $FLMBR$ time trace, which is subsequently differentiated. Note that the smoothing procedure affects neighbouring time points. In order not to introduce fake time dependences on our basic $17ms$ time scale, we perform all smoothing manipulations with the raw data on a $4ms$ time scale, determine $ULOOP$ according to Eqs. 1 and 2, and then put the result back to the $17ms$ scale.

The diamagnetic β is calculated from the diamagnetic loop measurement (corrected for base-line shifts), according to

$$\beta_{dia} = \beta_{\perp} = 1 - f * \Big[DIA(31) + t_{vvc} \cdot \frac{\partial}{\partial t} DIA(31) \Big] \qquad (3)$$

where $t_{vvc} = 0.0155s$ is the vacuum vessel time constant, and

$$f = 7.958 \cdot 10^9 * \frac{\Big[AMIN^2 + BMIN^2 \Big]}{AMIN * BMIN} * \frac{BT(RGEO)}{CUR^2}. \qquad (4)$$

For the determination of confinement times in subfile 1b, we will need the time derivative of $BETDIA$. Because of causality considerations, $DBETDT$ is calculated from $BETDIA$ on a faster time scale ($1ms$) and then transferred back to our $60Hz$ base frequency.

$PMASS$ and $PTIME$ denote the pellet 'mass' and the pellet injection time. These values are stored at the time point following the injection of the pellet. At all other time points, these variables set 'undefined'. Strictly speaking, $PMASS$ denotes the particle number per pellet and is determined from the signal $PELL3(30)$ and the empirical proportionality constants :

$PMASS = 1.63 \times 10^{19} * PELL3(30)$ for small hydrogen pellets,
$PMASS = 4.13 \times 10^{19} * PELL3(30)$ for large hydrogen pellets,
$PMASS = 1.90 \times 10^{19} * PELL3(30)$ for small deuterium pellets,
$PMASS = 4.04 \times 10^{19} * PELL3(30)$ for large deuterium pellets.

Pellet times are determined from bursts in the H_α light from the main chamber, i.e. the diagnostic $FDASOT(32)$ and the pellet μ-wave mass detector signal $PELL3(30)$. Since neither the pellet mass $PELL3(30)$ nor the H_α light intensity $FDASOT(32)$ unambiguously determine the injection of a pellet, the simultaneous occurrence of a characteristic behaviour of both signals is required for the indication of an injected pellet. A typical example is shown in Fig. 3, where both signals are plotted and accepted pellets are labeled by crosses at the corresponding time values. Nevertheless, in roughly 10 % of the pellet discharges complications do arise and non existing 'ghost' pellets are erraneously counted. At present we do not have a procedure for determining the sawtooth period $SZPER$ automatically. Thus $SZPER$ is undefined in most of the cases.

After creation of subfile 1a, plots of all important quantities versus time are produced. This not only allows for a consistency check of the data stored in DABA, but also provides the user a quick overview of all relevant information. A typical example is shown in Fig. 4.

Subfile 1b contains data which are based on profile information of subfile 2. Some of these quantities, like line- or volume-averaged densities and temperatures are calculated straightforward from the profile data. For the q value on axis we give four values

$Q0SD, Q0ND$: q_0 - values where the current diffusion is taken into account for classically and neoclassically determined current densities. These quantities require the solution of a differential equation and are still undefined at present.

$Q0S, Q0N$: q_0 - values calculated from classically and neoclassically determined current densities, neglecting the current diffusion. Here, q_0 is related to the current density $j(r)$ via

$$q(0) = \frac{2B_t}{\mu_0 R j(0)}, \tag{5}$$

where R denotes the position of the magnetic axis, i.e. $R = RGEO + DRBET$, B_t the toroidal field and $j(0)$ is the axis value of the toroidal current density [3]

$$j(r) = 5.63 \cdot 10^6 \frac{1}{Z_{eff}} T_e(r)^{3/2} \frac{ULOOP}{RGEO} \cdot G_1 \cdot G_2 \cdot G_3. \tag{6}$$

The numerical constant applies to MKSA units, but T_e in keV. The coefficient G_3 is

$$G_3|_{Spitzer} = 1 \tag{7}$$

in the Spitzer model and

$$G_3|_{neo} = \left[1 - \frac{f_T}{1+\xi\nu_e^*}\right]\left[1 - \frac{C f_T}{1+\xi\nu_e^*}\right] \tag{8}$$

in the neoclassical model, where

$$f_T = 1 - \frac{(1-r/R)^2}{\sqrt{1-r^2/R^2}(1+1.46\sqrt{r/R})} \tag{9}$$

$$C = \frac{0.56}{Z_{eff}}\left[\frac{3 - Z_{eff}}{3 + Z_{eff}}\right] \tag{10}$$

$$\xi = 0.58 + 0.2 \cdot Z_{eff} \tag{11}$$

$$\nu_e^* = 1.186 \cdot 10^{-22} \frac{R^{5/2}}{r^{3/2}} \frac{n_e}{T_e^2} \frac{q(r)}{G_2}. \tag{12}$$

The coefficients G_1 and G_2 are the same in both models and read :

$$G_1 = \frac{0.51}{\frac{0.457}{1.077+Z_{eff}} + 0.29} \tag{13}$$

and

$$G_2 = 17/\left[39.2 + ln(\frac{T_e}{Z_{eff}}) - 0.5 \cdot ln(n_e)\right]. \tag{14}$$

Taking the above definitions for j_S and j_{NC}, respectively, values for $ZEFFS$ and $ZEFFN$ can be determined by varying, for given $ULOOP$, Z_{eff} iteratively until the relation

$$CUR = 2\pi \int_0^a j_{S,NC}(Z_{eff}) r dr \tag{15}$$

is satisfied. The safety factor profile $q(r)$ in Eq. 12 is calculated according to

$$q(r) = \frac{2\pi}{\mu_0} \frac{B_t r^2}{R_{geo} I(r)}, \tag{16}$$

where $I(r) = 2\pi \int_0^r j(r')r'dr'$ denotes the current up to radius r. In the neoclassical model, the q-profile and the current density profile are iterated until convergence is reached. Assuming the ideal case of a pure hydrogen plasma, i.e. $Z_{eff} = 1$, the loop voltages $URESS$ and $URESN$ are calculated from Eq. 6.

The four quantities $ALISD, ALIND, ALIS, ALIN$ denote l_i-values corresponding to the different definitions of the current density and are calculated according to

$$l_i = \frac{< B_p^2 >}{B_p^2}|_a = 2\left[I(a)^{-2} \int_0^a (I(r)^2/r) dr \right]. \tag{17}$$

Two different expressions for the electron energy content are given in subfile 1b. $EKINE$ denotes the kinetic energy carried by the electrons as calculated from the electron temperature profiles of subfile 2:

$$EKINE = \frac{3}{2} \int_0^{V_p} n_e T_e dV. \tag{18}$$

The energy content of the plasma perpendicular to the magnetic field is determined from $BETDIA = \beta_\perp$, which is measured by the diamagnetic loop. Since $\beta_\perp = 2\mu_0 < p_\perp > /B_{pol}^2(a)$, where the index $\perp$ denotes the perpendicular contributions, and $< . >$ denotes the volume average, we find for the thermal energy

$$E_\perp = \frac{3}{8} \mu_0 R I(a)^2 \beta_\perp. \tag{19}$$

The calculated energy

$$EDIA = 4.7 \cdot 10^{-7} * RGEO * CUR^2 * BETDIA. \tag{20}$$

corresponds to the total (beam + particle) plasma energy under the assumption of isotropic energy distribution of the thermal particles, and a relatively small fraction of beam particles (which is not true for all discharges). The derivative $\partial EDIA/\partial t$ of the total plasma energy is calculated from the high resolution diamagnetic signal $DIA(31)$. With these definitions for the energy, two confinement times are calculated :

$$TAUEE = 1.9 * EKINE/\left[POH + CA * PNBI + 0.6 * PICRH + 0.3 * PLH - \frac{\partial}{\partial t} EDIA\right], \tag{21}$$

and

$$TAUE = EDIA/\left[POH + CA * PNBI + 0.6 * PICRH + 0.3 * PLH - \frac{\partial}{\partial t} EDIA\right]. \tag{22}$$

The factor 1.9 in Eq. 21 accounts approximately for the ion contributions to the plasma energy content. The factors in front of the heating powers take care of the fact that only part of the nominally injected heating powers are absorbed by the plasma. The value for CA is chosen to be $CA = 0.85$ for discharges with co-injected neutral beams and $CA = 0.65$ for cases with counter-injection. Obviously, these absorption factors are only crude approximations of reality. More sophisticated models, however, are often too complicated and/or of preliminary nature and therefore not yet suitable to be included in DABA.

Similar as with subfile 1a, we also plot the quantities of subfile 1b versus time, as can be seen in Fig. 5.

Subfile 2 contains profiles for electron temperature, electron density and bolometrically measured radiated power densities, as well as their corresponding radial derivatives. Temperatures and densities are determined at 9 equidistant radial grid points $r_i = i * \sqrt{(AMIN * BMIN)}/8, i = 0, ..., 8$ by the function parameterisation technique [4] from YAG-laser measurements. The behaviour in the scrape-off region is determined by fall-off lengths $(RDECT, RDECD)$ for temperature and density and their corresponding coefficients $(TECO3, DENCO3)$ according to

$$T_e(x > RSEP) = \sum_{j=1}^{3} TECO3(j) * exp\left[-(x - RSEP)/(j * RDECT)\right], \tag{23}$$

and

$$n_e(x > RSEP) = \sum_{j=1}^{3} DENCO3(j) * exp\left[-(x - RSEP)/(j * RDECD)\right]. \qquad (24)$$

This exponential behaviour in the scrape-off region does not hold for the radiated power densities – here a tenth radial grid point at $RLAST = \sqrt{(}AMIN * BMIN) + 0.03m$ is added. For a rough check of the consistency of the processed power density data with the bolometer channel raw data, we plot, for the first six and last six time points of each discharge, the measured intensity (raw signal) versus channel number (solid line in Fig. 6a) and the corresponding result calculated from the determined radiated power density profile (dashed line in Fig. 6a). Furthermore we also plot the power density profile at some other time points, as shown in Fig. 6b.

4. How to use DABA:

Access of DABA requires a read-password, which will be installed by P. Martin. Since the DABA program is kept on P. Martin's minidisk, one has to link 'PEM'. Now the direct and/or interactive communication with DABA can be started by 'DABAR'. The user will be guided by a comfortable menu, which consists of several functions:

E - end	R	ends DABA session
H - help	R	general DABA description
N - shows names of variables	R	shows all variable names and gives their description
R - releases locked shotfiles	W	a shotfile is locked whenever a program generating DABA data crashes
0 - write subfile 0	W	for entering subfile 0 data by hand
D - deletes subfiles/groups	W	for deleting subfiles and datagroups
V - verify	W	sets verification flags for subfiles or data groups
L - list data	R	menu for listing data either as a function of time in form of a table or the full subfile at one chosen time point
S - short information	R	gives short information of shot file, i.e. time points in subfiles, file version and availability of survey data

U - subfile descriptions	R/W	lists descriptions of subfiles
A - lists online shots	R	lists all online shots and their prefix numbers
M - show memory file	R	lists all shots, date of creation, name of creator, number of time points in subfiles, sorting according to different keys is possible
2 - info : origin of subfile 2	R	gives information for subfile 2

The third column indicates whether the pertaining command is for a user reading from DABA (R), or for a user writing to DABA (W), or for both (R/W). In order to write to DABA, one needs a special write password.

Tables which show variables as a function of time can be written into a file on the users machine by specifying a corresponding option. Similary, it is possible to produce plot output on a lasergraphics printer as well as on the terminal. In many applications, it is necessary to address DABA interactively from a FORTRAN program. This can easily be done with interface routines, which are contained in the DABA program and are described in the file REFER DABA on the machine PEM. Some simple examples are :

DBSET$	initialises DABA
DBMES$	gives error messages
DBNAM$	lists all DABA variables
DBCL$	closes open DABA files
DBSKO$	spline routine interpolates temperature, density and radiated power density profiles
DBR1T$	reads variables from specified subfile at specified time

etc.

An example for a FORTRAN program reading data from subfile 2 and subfile 1b is given in Appendix B.

5. How to use SAS:

Now we will give a short introduction how to access and analyse the data with SAS under CMS on the IBM mainframe computer. Statemements you have to type in, are written in bold-face. In the example SAS programs, /* and */ are used to delineate comments. The first step is to access the various disks. This is done in CMS as follows:

giveme sas /* links the user to the disk containing the sas system */

```
giveme ojk 193                  /* links to the disk containing the DABA/SAS data files */
```

The second step is to read the large DABA/SAS data files, but to write only relevant sub-sets, of the observations (shots, timepoints) and as well as of the variables, to the user's disk for further analysis. For simplicity, we consider first the case that only the data of one subfile, for instance subfile 1b, are needed. Look with the CMS command **q disk** to the letter that is assigned to the linked disk 'ojk 193'. We assume here it is the letter c. Write the following program on a file with filetype 'sas':

```
cms filedef benc
 disk dummy dummy c;            /* this tells the sas system to look on disk c, whenever
                                a reference is made to a file with filetype 'benc' */
data bin.myfile;                /* makes a new sas data set, with fn ft fm = myfile bin a */
 set benc.subf1b;               /* using the file 'subf1b benc c' as input file */
 if (25000 ≤ shotno ≤ 27000)
  and (1.2 ≤ time ≤ 1.35);      /* reads all data of benc.subf1b, but copies only the data
                                satisfying this selection criterion to bin.myfile */
 keep denvav taue ... ;         /* selects the variables of which you typed the names */
run;                            /* this slice of the SAS program (from the
                                'data myfile.bin' statement onwards) is carried out */
proc means data=bin.myfile;     /* makes an overview of the file you just created */
endsas;                         /* to finish the entire sas program */
```

Notice that a SAS statement ends with a semicolon (;). It is of no concern how many blanks or linefeeds there are between the words within a sentence. SAS can read, but does not distinguish between, upper-case and lower-case characters. After you have typed in your SAS program, save the file on which it was written, under the name 'shots1 sas' for instance (the filetype 'sas' is obligatory). If you now type under CMS: **sas shots1**, the program will run. When it is finished (you get the prompt 'ready' from CMS) you will have an additional file 'shots1 saslog a', which contains the text of the submitted SAS program, interspersed with comments and errors diagnostics of the SAS compiler. Normally, if not aborted by an early error, you will also get the file 'shots1 listing a', on which the output is written from SAS procedures you requested in your program, such as data overviews and regression results. Note that you created the SAS file 'bin.myfile' on your A disk. In subsequent SAS programs you can directly use this file. For instance, the following few lines can be written on a new file ('shots2 sas a'):

```
proc contents data=bin.myfile;/* gives an overview of the variables stored in the file */
proc print data=bin.myfile;    /* prints the data on 'shots1 listing a' */
proc fsprint data=bin.myfile; /* prints data file on the screen for interactive browsing */
endsas;
```

The new program is run under CMS by typing: **sas shots2**.

Now, we consider the slightly more complicated example of how to merge the various subfiles (0,1a,1b,2,3) into one file. The SAS program is as follows.

```
data bin.myfile;
  merge benc.subf1a benc.subf1b
benc.subf2;                          /* 'merges' the indicated files */
  by shotno time;                    /* for each value of (shotno,time) */
  if (25000 ≤ shotno ≤ 27000)
   and (1.2 ≤ time ≤ 1.35);          /* only this selection */
  keep denvav taue ... ;             /* is written to your disk ! */
run;
data bin.myfile2;                    /* creates bin.myfile2;*/
  merge bin.myfile benc.subf0;/* by merging the indicated files */
  by shotno;                         /* for each value of shotno */
  if (25000 ≤ shotno ≤ 27000);/* now, only the selection with respect to shotno ! */
run;
proc fsprint data=bin.myfile2;/* for an interactive look at the data; exit with PF3 */
endsas;
```

In order that a merge 'by shotno time' can be carried out by SAS, the dataset needs to be ordered by shotno and time. As the 'benc'-files are routinely ordered in this way, there is no problem. For other instances, sorting a file in SAS is done as follows:

```
proc sort data=bin.myfile;
by shotno time;
```

For equal values of shotno, the observations are sorted with respect to time. All sorting is done in ascending order, unless the option 'descending' is used. The merging of the files ('matched' by shotno and time) is visually represented in Fig. 7. Note that, since subfile0 does not contain time as a variable, one cannot merge subfile0 with the other subfiles 'by shotno time'.

The variable names in the SAS files benc.subf0, benc.subf1a, . . ., benc.subf3 are the same as those in the corresponding DABA files. The array variables, such as te5 and den5 in subf1b, are denoted by the variables TE5_1, TE5_2, ... TE5_5, DEN5_1, DEN5_2, ... DEN5_5, etc. (In the keep statement one can briefly put **TE5_1-TE5_5** to include 5 such temperature variables.) One can always get an overview of the variable names by including the SAS statement **proc contents data=benc.subf1a**, etc.

Creating the selected user file 'bin.myfile2' from the large database may require some time (presently, about half a minute CPU time, 3 to 7 minutes response time, as the large database consists of nearly 100.000 observations and, in total, about 300 variables.) In a typical investigation, by making a sensible initial selection, one usually does not need to do this more often than once a week or a fortnight, however.

Finally, using directly the specific SAS user file 'bin.myfile2', one has entire powerful spectrum of the SAS procedures at one's disposal. We give only a brief example. For more information, one it is enlightening to read the somwhat bulky, but clearly written SAS documentation, momentarily present at the Benutzer Beratung of the RZG, and the computer rooms 272 (L6) and 214 (L5): SAS BASE for data management procedures and simple statistics, SAS STATISTICS for more advanced statistics, SAS GRAPH for graphical procedures, FSP for full screen interactive applications, ETS for time series analysis, and IML for using the Interactive Matrix Language. There also exists a thin SAS Introductory Guide (in English as well as German) to provide the novice user a quick start.

```
data myfile3;
set myfile2;
/* data transformations that are deemed useful: */
/* (the new variables are added to the variables already present) */
ltaue=log(taue);
lcur=log(cur);
...
run;
/* to perform a regression analysis: */
proc reg data=myfile3;
model ltaue = lcur lbt ldenlav;
```

```
run;
/* to make a plot: */
proc plot data=myfile3;
plot ltaue * lcur;
run;
```

An investigation is often started with the idea that only very simple statistical procedures are needed, but after a while, the requirements typically grow. Using SAS instead of a more superficially 'canned' statistical package has the advantage that in a later stage also more complex research questions can be attacked with it, without leaving the environment. For that reason, it was decided some time ago at JET to use SAS (with the same modules as we have presently at Garching) as the standard system for all multi-shot statistical analysis and data management of the processed pulse data.

Figure Captions:

Fig. 1 : Block diagram symbolising the coupling of DABA to periphery routines and to the statistical analysis package SAS.

Fig. 2 : Block diagram symbolising the structure of DABA and corresponding programs for processing raw data. Parts which are still being constructed are bounded by dashed lines.

Fig. 3 : Determination of $PTIME$ and $PMASS$. Plotted are $PELL3(30)$ (upper figure) and $FDASOT(32)$ lower figure. The accepted pellets – where both signals are above certain background levels – are marked by crosses. Determined pellet times are listed on the right-hand side. Similar plots are automatically produced for all pellet discharges.

Fig. 4 : Important quantities of subfile 1a versus time. These plots are produced for each discharge to be stored in DABA.

Fig. 5 : Important quantities of subfile 1b versus time. These plots are produced for each discharge to be stored in DABA.

Fig. 6a : Intensity measured by the bolometer channels versus channel number (solid line) and the corresponding quantity calculated from the power density profiles (dashed lines). Results are shown at the first six and last six time points of a discharge.

Fig. 6b : Calculated radiated power density versus radius at twelve different time points. The times are as in Fig. 6a.

Fig. 7 : Merging of two SAS datasets by shotno and time.

Appendix A:

We list the quantities stored in the various subfiles of DABA. Variables ending on a number denote FORTRAN vectors, those not ending on a number denote FORTRAN scalars. Variables beginning with the letter I denote integers, the other variables denote real quantities.

DABA Quantities of subfile 0:

All data are in MKSA units, temperatures and beam-particle energies are in eV.

subfile 0: survey data, subgroup 1

Maxima of time-dependent quantities, many of which are in subfile 1a or subfile 1b. Unless stated otherwise, all maxima are taken over the interval 0.5 < t < TEND-0.1, where TEND denotes the end of the discharge

BTMAX	max. BT
CURMAX	max. CUR
QAMIN	min. QA
BETMAX	max. BETDIA
EKIEMA	max. EKINE
EDIAMA	max. EDIA
DENMAX	max. DENLAV
DTOTMA	max. DTOT
TEOMAX	max. electrontemp. on axis
TIOMAX	max. iontemp. on axis
ZEFMAX	max. ZEFF in range
TEND	max. time in subfile 1a
PTMAX	max. PTOT
IHREG	(0 = undef. 1 = H regime not reached. 2 = H regime reached)
RGEOM	averaged RGEO
AGEOM	averaged $\sqrt{(AMIN * BMIN)}$
ZGEOM	averaged ZGEO

subfile 0: survey data, subgroup 2

RLIM	radius of limiter closest to plasma
IPLAS	working gas (0=undef. 1=H. D=2. He=4).

subfile 0: system information

(standard coding: 0=undef, 1=no, 2=yes)

ICARB	carbonisation
ITITAN	Ti-puffing
INBI	NBI
IICRH	ICRH
ILH	LH
IPEL	pellet (1=no pellet, 11=H_2 small, 12=H_2 large, 21=D_2 small, 22=D_2 large)

subfile 0: heating and pellet information, subgroup 1

POHMAX	max. POH
PNBIO	max. PNBI
TBON	time where NBI sets in
TBOFF	time where NBI is switched off
PICRHO	max ICRH power
TICON	time where ICRH sets in
TICOFF	time where ICRH is switched off
PLHO	max. LH power
TIHON	time where LH sets in
TIHOFF	time where LH is switched off

subfile 0: heating and pellet information, subgroup 2

PBEAM	injected power in 3 comp. E, E/2 and E/3
FICRH	frequency of ICRH
TPEON	begin of injection
TPEOFF	end of injection
FPEL	frequency
VPEL	velocity
EBEAM	beam energy
IBEAM	beam gas (1=H, 2=D)
ISPEC	LHH spectrum (1=asymmetric, 2 = symmetric)

subfile 0: diagnostic availability

(standard coding: 0=undefined, 1=no, 2=yes)

ITHOMS	thomson scattering
IHCN1	HCN1
IHCN2	HCN2
IHCN3	HCN3
IHCN4	HCN4
IECE	ECE
IBOL	bolometry
ISXR	soft X-ray cameras
IMIRN	Mirnov coils
ICXPAS	passive neutral particle diagnostic
ICXACT	active neutral particle diagnostic
ICXREC	Ti-measurement with CX recombination line
INEUT	neutral flux measurement
INEUTS	neutron spectroscopy
ICHPAR	measurement of charged fusion products
ISPECT	spectroscopic measurements
ILI	Li-beam diagnostic

<u>subfile 0: characteristic parameters at 3 time points</u>

The quantities of this sub-section are vectors with 3 components, corresponding to 3 characteristic time points

TIME3	the 3 time points
CUR3	plasma current
BT3	toroidal field
DENLA3	line averaged density
DTOT3	$\int n_e dV$
PTOT3	tot. heating power
POH3	OH power
PNBI3	NBI power
PICRH3	ICRH power

PLH3	LH power
TI03	ion temperature in center
TE03	electron temperature in center
BETDI3	BETDIA, see subfile 1b
EKINE3	EKINE, see subfile 1b
EDIA3	EDIA, see subfile 1b
TAUE3	TAUE, see subfile 1b

subfile 0: correction factors

EICHF	correction factors: EICHF(1:17) for MULTIPRO, EICHF(18:23) for HCEPRO, and EICHF(24) for FPYAG

DABA variables of subfile 1a:

BT	toroidal field at 1.65 m
CUR	plasma current
FLMBR	flux difference in loop at plasma boundary
ULOOP	loop voltage at plasma boundary $(UL - \partial(FLMBR)/\partial t)$
POH	$\|ULOOP * CUR\|$
BETLI2	β_{pol} + li/2 from FP
BETDIA	β_{pol} from diamagnetic measurement
PNBI	neutral injection power (nominally injected)
PICRH	ICRH power launched
PLH	LHH power launched
RGEO	major radius of geometric center
DRBET	shift of the magnetic axis versus geometrical axis
ZGEO	vertical displacement of geometrical axis
AMIN	horizontal plasma radius
BMIN	vertical plasma radius
QA	cylindrical q at plasma boundary
Q95	Shafranov q at 95% of separatrix flux
SZPER	sawtooth period
RINPL	innermost r coordinate of plasma boundary
ROUTPL	outermost r coordinate of plasma boundary

PMASS	pellet mass
PTIME	time point of latest injected pellet
DBETDT	time derivative of BETDIA
ILD	plasma configuration (1 = limiter, 2 = SN top, 3 = DN, 4 = SN bottom)
IMODE	L-/H-mode (1 = OH, 2 = L, 3=H)
IFPREL	index for FP-version (1 = version 1 (1987))

DABA variables of subfile 1b:

DENVAV	volume averaged electron density, $\int n_e dV / \int dV$
DENLAV	line averaged electron density, $\int n_e dl / \int dl$
DTOT	electron content, $\int n_e dV$
TEAV	volume averaged electron temperature, $\int T_e dV / \int dV$
TEDEAV	$\int n_e T_e dV / \int n_e dV$
DEN5	DEN5(i)=n_e((i-1)*RSEP/4), i=1,5, with RSEP = $\sqrt{AMIN * BMIN}$
TE5	TE5(i)=T_e((i-1)*RSEP/4), i=1,5
Q0SD	q_0 according to Spitzer (from diff. eq., incl. current diffusion)
Q0ND	q_0 according to neoclass. (from diff. eq., incl. current diffusion)
Q0S	q_0 according to Spitzer (neglecting current diffusion)
Q0N	q_0 according to neoclass. (neglecting current diffusion)
URESS	resistive loop voltage, assuming ZEFF=1 according to Spitzer
URESN	resistive loop voltage, assuming ZEFF=1 according to neoclass.
ZEFFS	ZEFF according to spitzer
ZEFFN	ZEFF according to neoclass.
ALIS	li according to Spitzer (neglecting current diffusion)
ALISD	li according to Spitzer (from diff. eq., incl. current diffusion)
ALIN	li according to neoclass. (neglecting current diffusion)
ALIND	li according to neoclass. (from diffeq.) (from diff. eq., incl. current diffusion)
EKINE	energy content determined from electron profiles, $3/2 \int n_e T_e dV$
EDIA	$4.7 \cdot 10^{-7} * RGEO * CUR^2 * BETDIA$
BETORD	$BETDIA/(QA * RGEO/AMIN)^2$
TAUEE	$1.9 * EKINE/(POH + CA * PNBI + 0.6 * PICRH + 0.3 * PLH - \partial EDIA/\partial t)$,
TAUE	$EDIA/(POH + CA * PNBI + 0.6 * PICRH + 0.3 * PLH - \partial EDIA/\partial t)$ where CA = 0.85 (for co-injection) or CA = 0.65 (for counter-injection).

PBOLTO total bolometrically measured radiation

FLNEUT neutron flux

ZEFFB0 ZEFF on magnetic axis, from infra-red Bremsstrahlung

ZEFFBF form factor (volume averaged ZEFF)/ZEFFB0

DABA variables of subfile 2:

RDECD basic fall-off lengths of density fit (fall-off lengths are k*RDECD with k=1,2,3)

RDECT basic fall-off lengths of temp. fit (fall-off lengths are k*RDECT with k=1,2,3)

DEN9 DEN9(9): DEN9(i)=density at points (i-1)*(RSEP/8),i=1,9, with $RSEP = \sqrt{(AMIN * BMIN)}$

DENDR9 DENDR9(9): DENDR9 is derivative of DEN9, storage as for DEN9

TE9 TE9(9): TE9(i)=temperature in points (i-1)*(RSEP/8), i=1,9

TEDR9 TEDR9(9): TEDR9 ist derivative of TE9, storage as for TE9

DENCO3 DENCO3(3): 3 coefficients of density falloffs in scrape off region

TECO3 TECO3(3): 3 coefficients of temperature falloffs in scrape off region

PBOL9 PBOL9(10): bolometrically measured radiated power density in the points: (i-1)*(AMIN/8),i=1,9 and RLAST

PBOLD9 PBOLD9(10): derivative of PBOL9 in the points (i-1)*(RSEP/8), i=1,9 and RLAST

RLAST last radial grid point for PBOL9, PBOLD9

DABA variables of subfile 3:

The vectors in this subfile all have length 16. The i^{th} component corresponds to the i^{th} radial channel of the YAG-laser Thomson scattering experiment ($i = 1, 16$).

TER electron temperature

DER electron density

The following 4 variables indicate approximately 67% confidence intervals from an error-propagation analysis:

TERMIN minimum electron temperature

TERMAX maximum electron temperature

DERMIN minimum electron density

DERMAX maximum electron density

Appendix B:

```
*************************************************************************
*
*     Program example to read a DABA file.
*
*     Following DABA fortran-interface routines are used:
*
*        initialize DABA once                       :sr. DBSET$
*        open a DABA file for read                  :sr. DBOPR$
*        get subfile data                           :sr. DBRAT$
*        close DABA file opened                     :sr. DBCL$$
*        convert error index to a text message      :sr. DBMES$
*
*     For detailed description of subroutines used see REFER DABA.
*************************************************************************
*
      PROGRAM EXAMPLE
*
*     NDIM2    = max. number of time steps
*     NDIMR    = leading dimension of DABA output field RVAL
*     NDIMI    = leading dimension of DABA output field IVAL
*
      PARAMETER ( NDIM2 = 200, NDIMR = 40, NDIMI = 4 )
      DIMENSION   RVAL (NDIMR,NDIM2), IVAL (NDIMI,NDIM2), TT (NDIM2)
      CHARACTER   TEXT*70, PNAME*10, VRSN*25, FNAME*6
*
      PRINT *,'HALLO,this is an example how to use DABA read routines'
      PRINT *,'Program written by P. Martin, IPP, Tel. 600'
*
      PNAME = 'BOSS' ! sets DABA password : for your use replace ??????
*
* Initialise DABA by subroutine DBSET$.
*    If an error returns : no access to DABA is possible, the reason
*    is explained by the text string returned from sr. DBMES$.
* NOTE: initialisation is required only once for a job.
* Short parameter description for subroutine DBSET$:
*    PNAME    -   input,character*(*) string containing the DABA password
*    VRSN     -   output, character*25 string containing an information
*                         about the DABA version
*    IER      -   output, integer, DABA error index, IER=0 means okay
*
*
      CALL DBSET$ (PNAME, VRSN, IER)
      IF (IER.NE.0)   THEN
         CALL DBMES$ (IER, TEXT)
         PRINT *,'error message from DABA initialiser DBSET$ :'
         PRINT *, TEXT
         GO TO 5000
      END IF
      PRINT *,'DABA correctly initialised', '    /     ', VRSN
*
*
* Open DABA file for read by subroutine DBOPR$.
* Short parameter description for subroutine DBOPR$:
*    ISHOT -   input , integer, ISHOT defines the shotnumber
*    FNAME -   output, character*6, name of DABA file opened
*    IER   -   output, integer, DABA error index, IER=0 means okay
*    IDUM  -           integer, dummy parameter.
* After sucessful open you can use following subroutines
* in arbitrary order and number to read DABA data:
*    DBRST$, DBRRS$, DBRIS$, DBRSO$, DBR1T$, DBRAT$
*
*
```

```
      ISHOT = 27770              ! Define shot number
*
      CALL DBOPR$ (ISHOT, FNAME, IER, IDUM)
      IF (IER.NE.0)  THEN
         CALL DBMES$ (IER, TEXT)
         PRINT *,'error message from DABA routine DBOPR$ :'
         PRINT *, TEXT
         GO TO 5000
      END IF
      PRINT *,'DABA file opened:', FNAME
*
* Read subfile 1A data for all existing time steps.
* short parameter description for subroutine DBRAT$:
* Input:
*   IDUM     - integer, dummy parameter
*   ISUBF    - integer, index for subfile to read.
*              ISUBF=1 : subfile 1A    ,  ISUBF=2 : subfile 1B
*              ISUBF=3 : subfile 2     ,  ISUBF=4 : subfile 3
*   NDIMR    - integer, leading dimension of output parameter RVAL
*              The required size of NDIMR is subfile dependend :
*              for subfile 1A   NDIMR >= 23
*              for subfile 1B   NDIMR >= 36
*              for subfile 2    NDIMR >= 65
*              for subfile 3    NDIMR >= 96.
*   NDIMI    - integer, leading dimension of output parameter IVAL
*              The required size of NDIMI is subfile dependend :
*              for subfile 1A    NDIMI >= 3
*              for all other subfiles  NDIMI >= 1
*   NDIM2    - integer, first dimension of output parameter TT  and
*              second dimension of output parameters RVAL, IVAL.
*              Suggested size for NDIM2 : 200
* Output:
*   TT       - real vector of length NDIM2 containing the time steps
*              found. The time steps are sorted from low to high.
*   NTIME    - integer, number of time steps found.
*   RVAL     - 2-dimensional real field. Leading dimension is NDIMR,
*              the second dimension corresponds to the time steps and
*              is NDIM2. Elements of RVAL not indicated are meaningless
*              at present.
*                  ISUBF=1 : subfile 1A :
*                   RVAL (1,)..RVAL (23,) =
*                    BT, CUR, FLMBR, ULOOP, POH, BETLI2, BETDIA, PNBI,
*                    PICRH, PLH, RGEO, DRBET, ZGEO, AMIN, BMIN, QA,
*                    Q95, SZPER, RINPL, ROUTPL, PMASS, PTIME, DBETDT.
*   IVAL     - 2-dimensional integer-field, leading dimension
*               is NDIMI. The second dimension corresponds to the
*               time steps and is NDIM2. Elements of IVAL
*               not described below are meaningless.
*                  ISUBF=1 : subfile 1A :
*                     IVAL (1,)..IVAL (3,) = ILD, IMODE, IFPREL
*   IMES       - integer, message index.
*                  IMES = 1 unable to transfer all elements of
*                           a time step (=group). Increase dimensioning
*                           with NDIMR  resp. NDIMI.
*                  IMES = 2 unable to transfer all time steps. Increase
*                           dimensioning with NDIM2.
*   IER        - integer, error index, IER=0  means okay.
*
*
      ISUBF = 1
      CALL DBRAT$
```

```
     |    (IDUM,ISUBF,NDIMR,NDIMI,NDIM2,TT,NTIME,RVAL,IVAL,IER,IMES)

      IF (IER.NE.0)   THEN
         CALL DBMES$ (IER, TEXT)
         PRINT *,'error message from DABA routine DBRAT$ :'
         PRINT *, TEXT
         GO TO 5000
      END IF
*
* Close DABA file opened.
* Short parameter description for DBCL$$ :
*    FNAME   -   output, character*6, name of DABA file closed
*    IER     -   output, integer, DABA error index, IER=0 means okay.
*
      CALL DBCL$$ (FNAME,IER)
      IF (IER.NE.0)   THEN
         CALL DBMES$ (IER, TEXT)
         PRINT *,'error message from DBCL$$ routine DBCL$$ :'
         PRINT *, TEXT
         GO TO 5000
      END IF
*
* Print BT, CUR, RGEO AND ZGEO within time range from TMIN to TMAX
*
      TMIN = 1.0  ! sec
      TMAX = 1.4  ! sec
      PRINT *,'from DABA file ', FNAME
      WRITE (6,100)
      DO  1  I = 1, NTIME
        IF(TT(I).GE.TMIN  .AND.  TT(I).LE.TMAX)
     |   WRITE (6,200)  TT(I),RVAL(1,I),RVAL(2,I),RVAL(11,I),RVAL(13,I)
    1 CONTINUE
*
  100 FORMAT(1X,'     TIME     BT         CUR        RGEO        ZGEO')
  200 FORMAT(1X, F11.4, 1P, 4E11.3)
*
 5000 PRINT *, 'EXAMPLE program ended'
      STOP
      END
```

References:

1) SAS Institute Inc., SAS User's Guides (Introductory Guide, SAS/BASE, SAS/STAT, SAS/GRAPH SAS/ETS, SAS/IML, SAS/FSP), Release 5, Cary, North Carolina (1985).

2) See FP online HELP of the ASDEX library of the VAX or CMS systems.

3) R. Hawryluk, 'Physics of Plasmas Close to Thermonuclear Conditions', Vol. I, Varenna (1979).

4) P.J. Mc Carthy and M.C. Sexton, 'Plasma Profile Recovery by Function Parameterisation', IPP report 5/12, (1986).

DABA/SAS INTERFACES

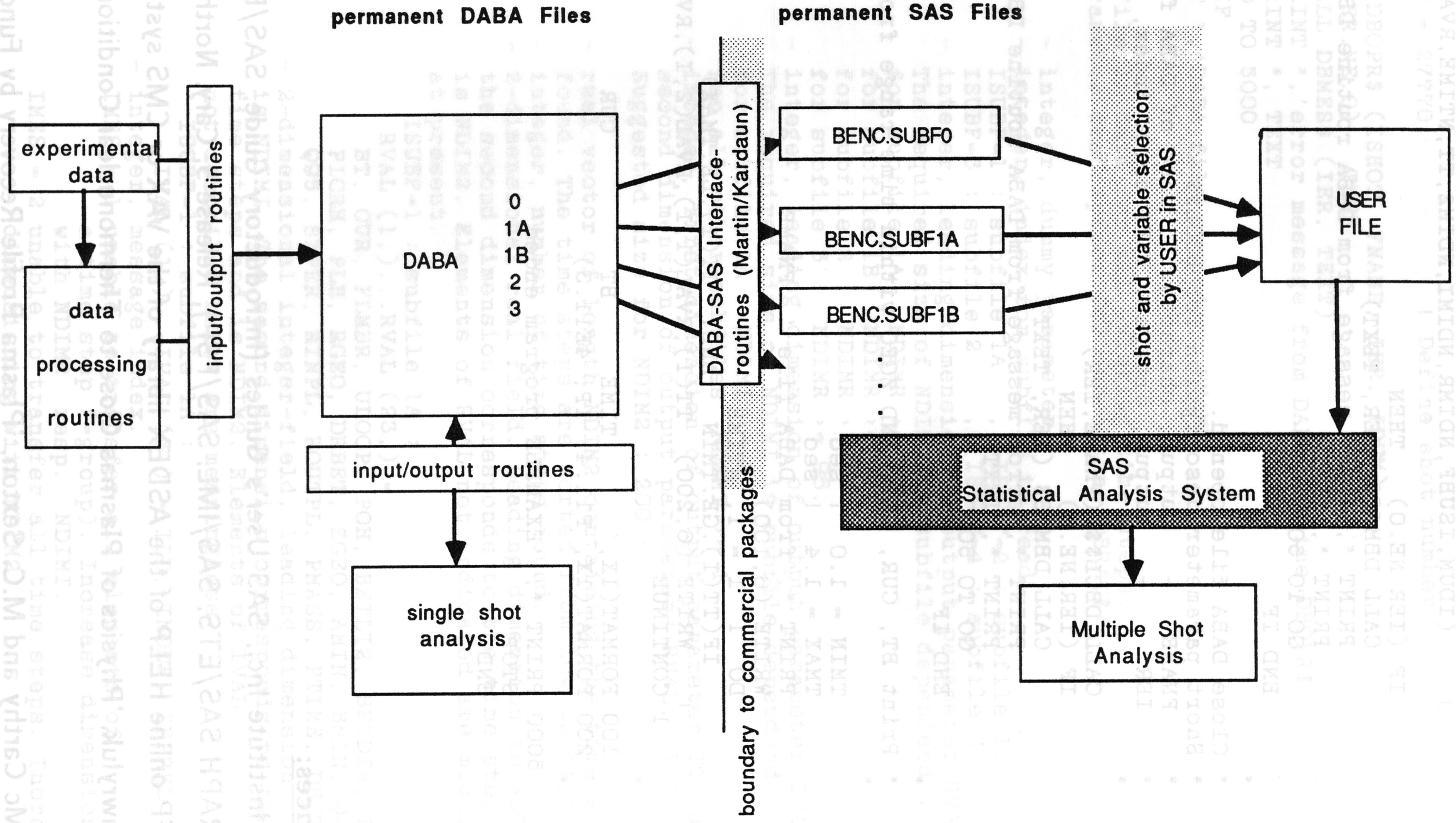


Figure 1

ASDEX INTEGRATED DATA-ARCHIVE (AIDA)

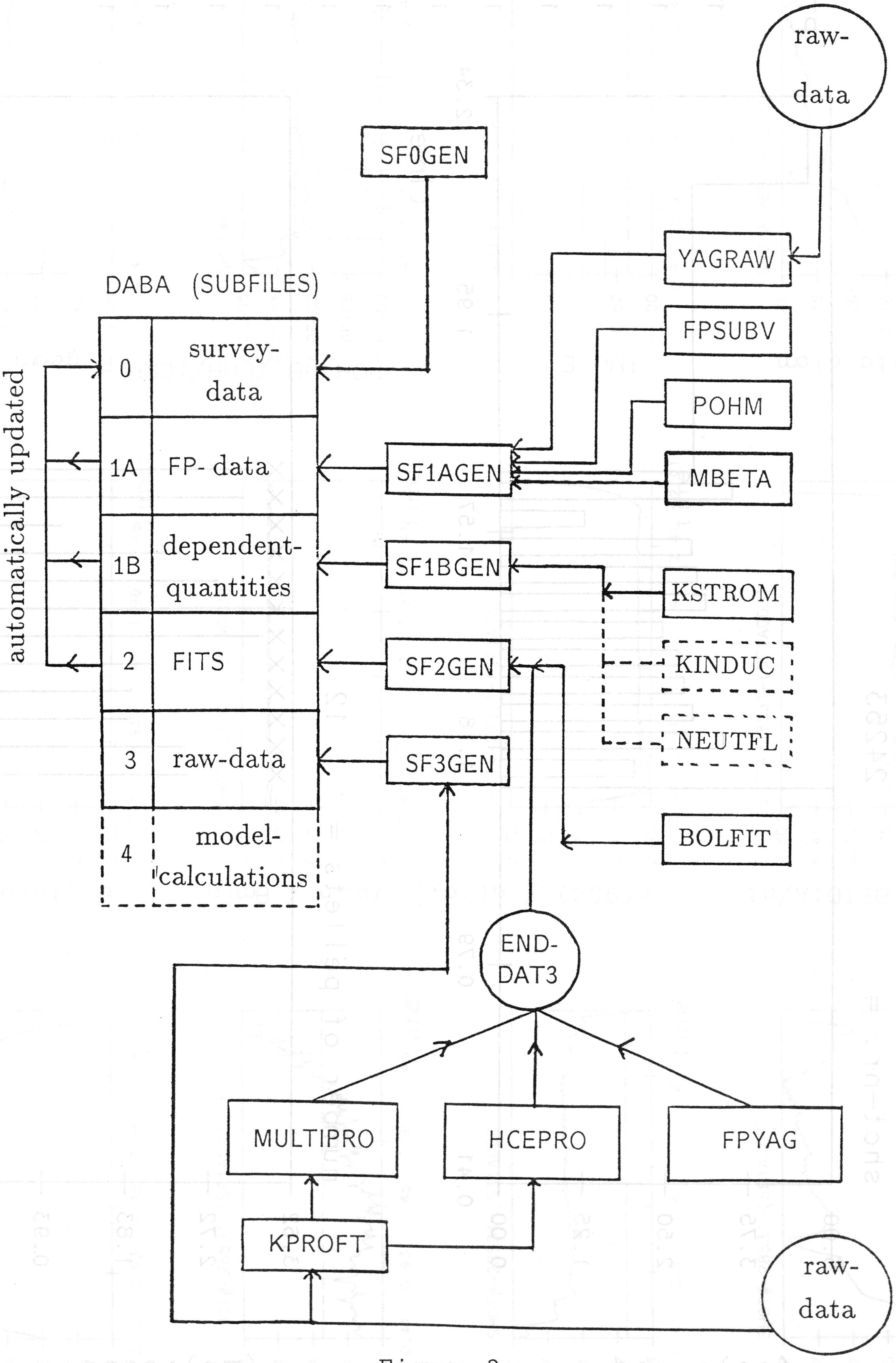


Figure 2

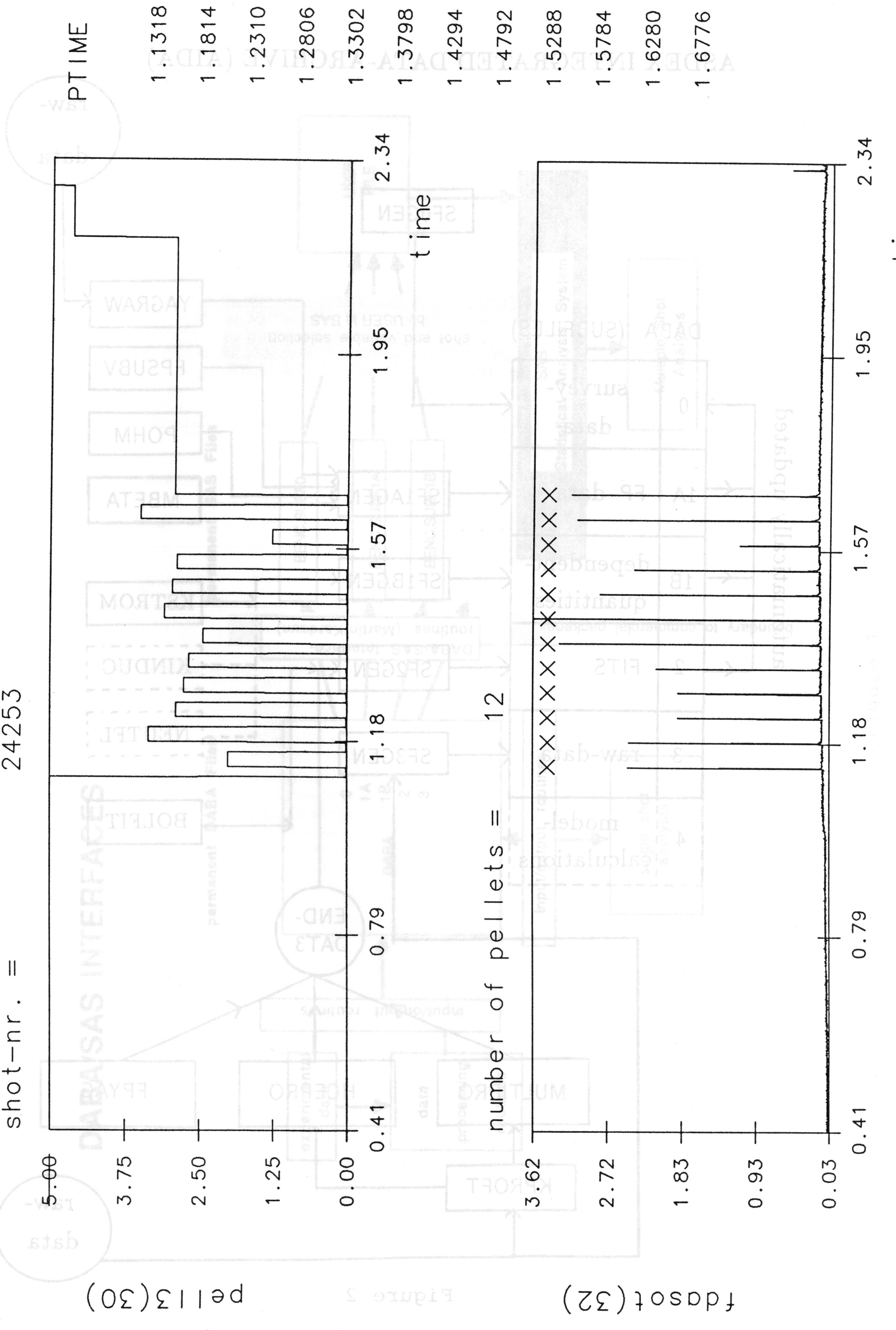
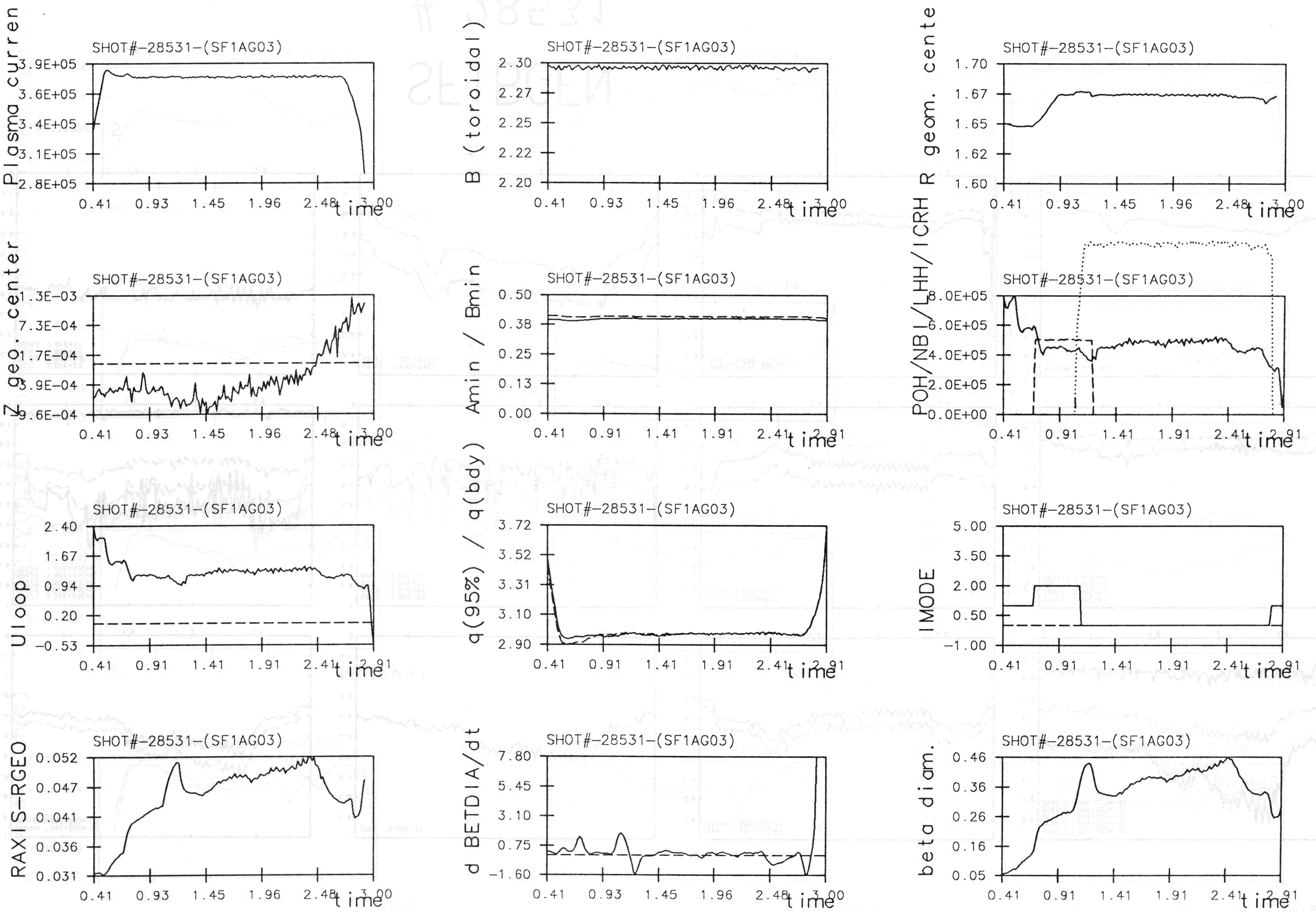

shot-nr. = 24253
number of pellets = 12
PTIME
1.1318
1.1814
1.2310
1.2806
1.3302
1.3798
1.4294
1.4792
1.5288
1.5784
1.6280
1.6776
pel13(30)
5.00
3.75
2.50
1.25
0.00
fdasot(32)
3.62
2.72
1.83
0.93
0.03
0.41
0.79
1.18
1.57
1.95
2.34
time

Figure 3



SHOT#-28531-(SF1AG03)
Plasma curren
3.9E+05
3.6E+05
3.4E+05
3.1E+05
2.8E+05
B (toroidal)
2.30
2.27
2.25
2.22
2.20
R geom. cente
1.70
1.67
1.65
1.62
1.60
0.41
0.93
1.45
1.96
2.48
3.00
time
Z geo. center
1.3E-03
7.3E-04
1.7E-04
-3.9E-04
-9.6E-04
Amin / Bmin
0.50
0.38
0.25
0.13
0.00
POH/NBI/LHH/ICRH
8.0E+05
6.0E+05
4.0E+05
2.0E+05
0.0E+00
0.41
0.91
1.41
1.91
2.41
2.91
Uloop
2.40
1.67
0.94
0.20
-0.53
q(95%) / q(bdy)
3.72
3.52
3.31
3.10
2.90
IMODE
5.00
3.50
2.00
0.50
-1.00
RAXIS-RGEO
0.052
0.047
0.041
0.036
0.031
d BETDIA/dt
7.80
5.45
3.10
0.75
-1.60
beta diam.
0.46
0.36
0.26
0.16
0.05

Figure 4

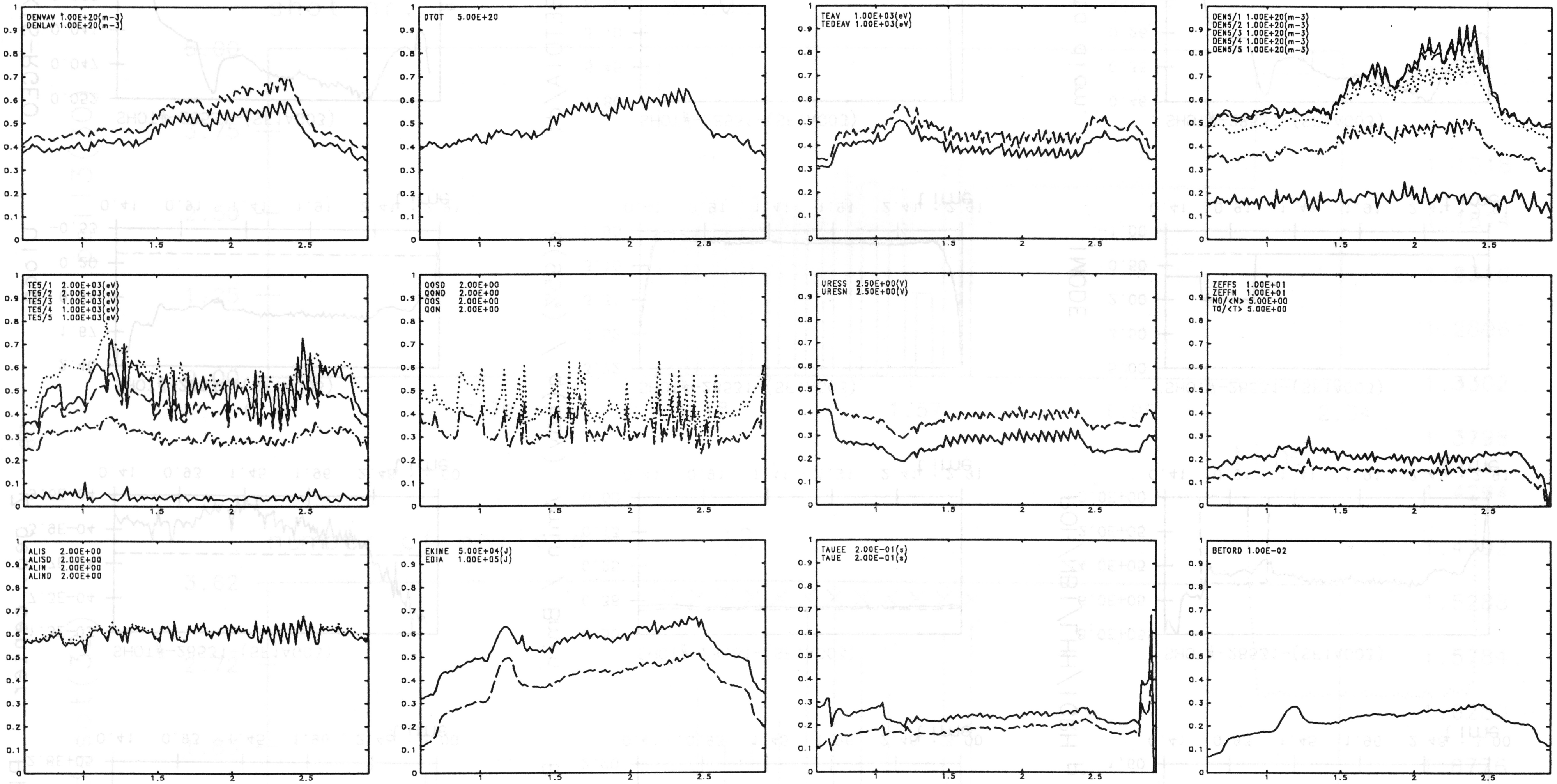


Figure 5

SF1BGEN

28531

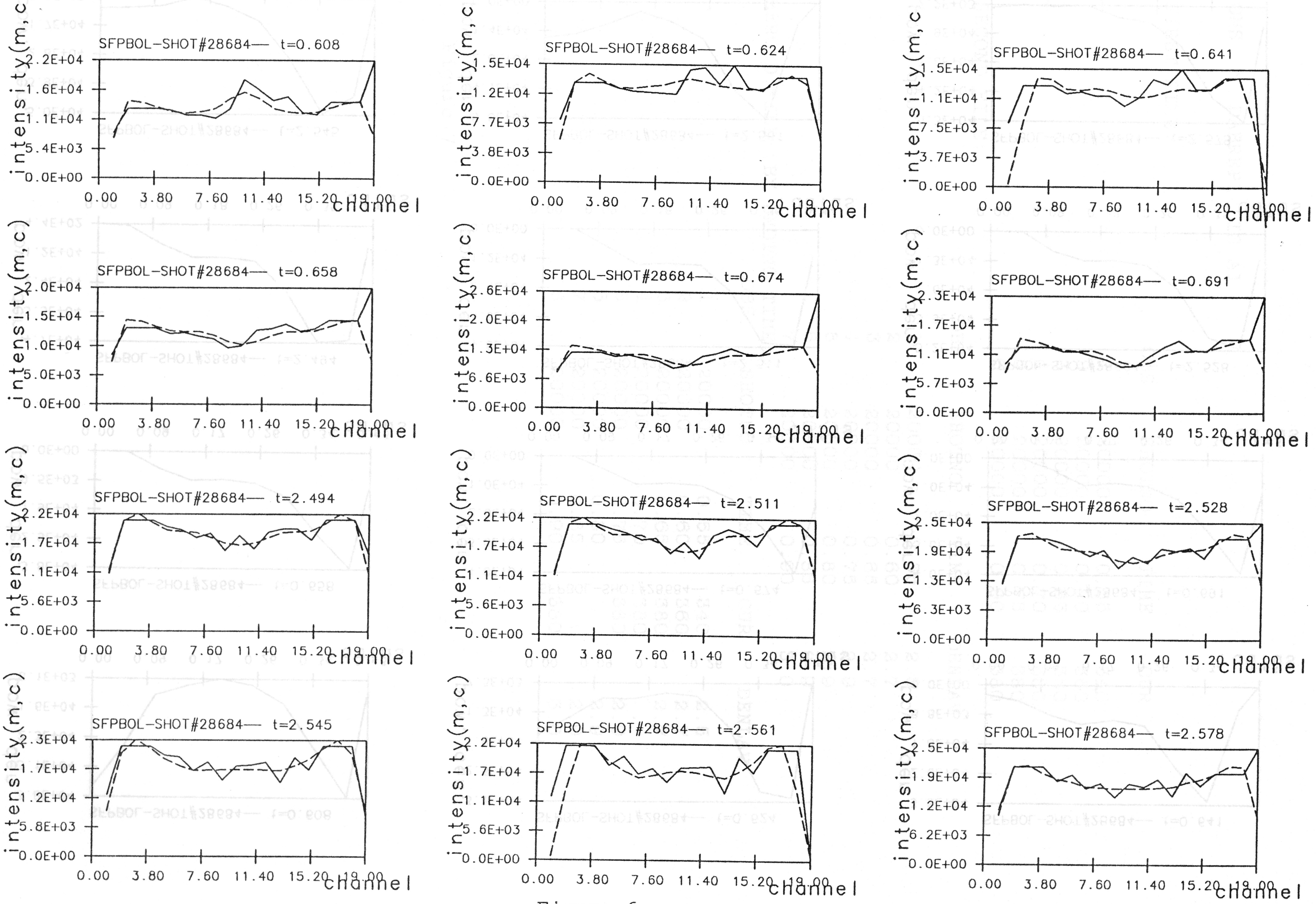

SFPBOL-SHOT#28684— t=0.608
2.2E+04
1.6E+04
1.1E+04
5.4E+03
0.0E+00
intensity(m,c
SFPBOL-SHOT#28684— t=0.624
1.5E+04
1.2E+04
7.7E+03
3.8E+03
0.0E+00
intensity(m,c
SFPBOL-SHOT#28684— t=0.641
1.5E+04
1.1E+04
7.5E+03
3.7E+03
0.0E+00
intensity(m,c
SFPBOL-SHOT#28684— t=0.658
2.0E+04
1.5E+04
1.0E+04
5.0E+03
0.0E+00
intensity(m,c)
SFPBOL-SHOT#28684— t=0.674
2.6E+04
2.0E+04
1.3E+04
6.6E+03
0.0E+00
intensity(m,c)
SFPBOL-SHOT#28684— t=0.691
2.3E+04
1.7E+04
1.1E+04
5.7E+03
0.0E+00
intensity(m,c)
SFPBOL-SHOT#28684— t=2.494
2.2E+04
1.7E+04
1.1E+04
5.6E+03
0.0E+00
intensity(m,c)
SFPBOL-SHOT#28684— t=2.511
2.2E+04
1.7E+04
1.1E+04
5.6E+03
0.0E+00
intensity(m,c)
SFPBOL-SHOT#28684— t=2.528
2.5E+04
1.9E+04
1.3E+04
6.3E+03
0.0E+00
intensity(m,c)
SFPBOL-SHOT#28684— t=2.545
2.3E+04
1.7E+04
1.2E+04
5.8E+03
0.0E+00
intensity(m,c)
SFPBOL-SHOT#28684— t=2.561
2.2E+04
1.7E+04
1.1E+04
5.6E+03
0.0E+00
intensity(m,c)
SFPBOL-SHOT#28684— t=2.578
2.5E+04
1.9E+04
1.2E+04
6.2E+03
0.0E+00
intensity(m,c)
0.00
3.80
7.60
11.40
15.20
19.00
channel

Figure 6a

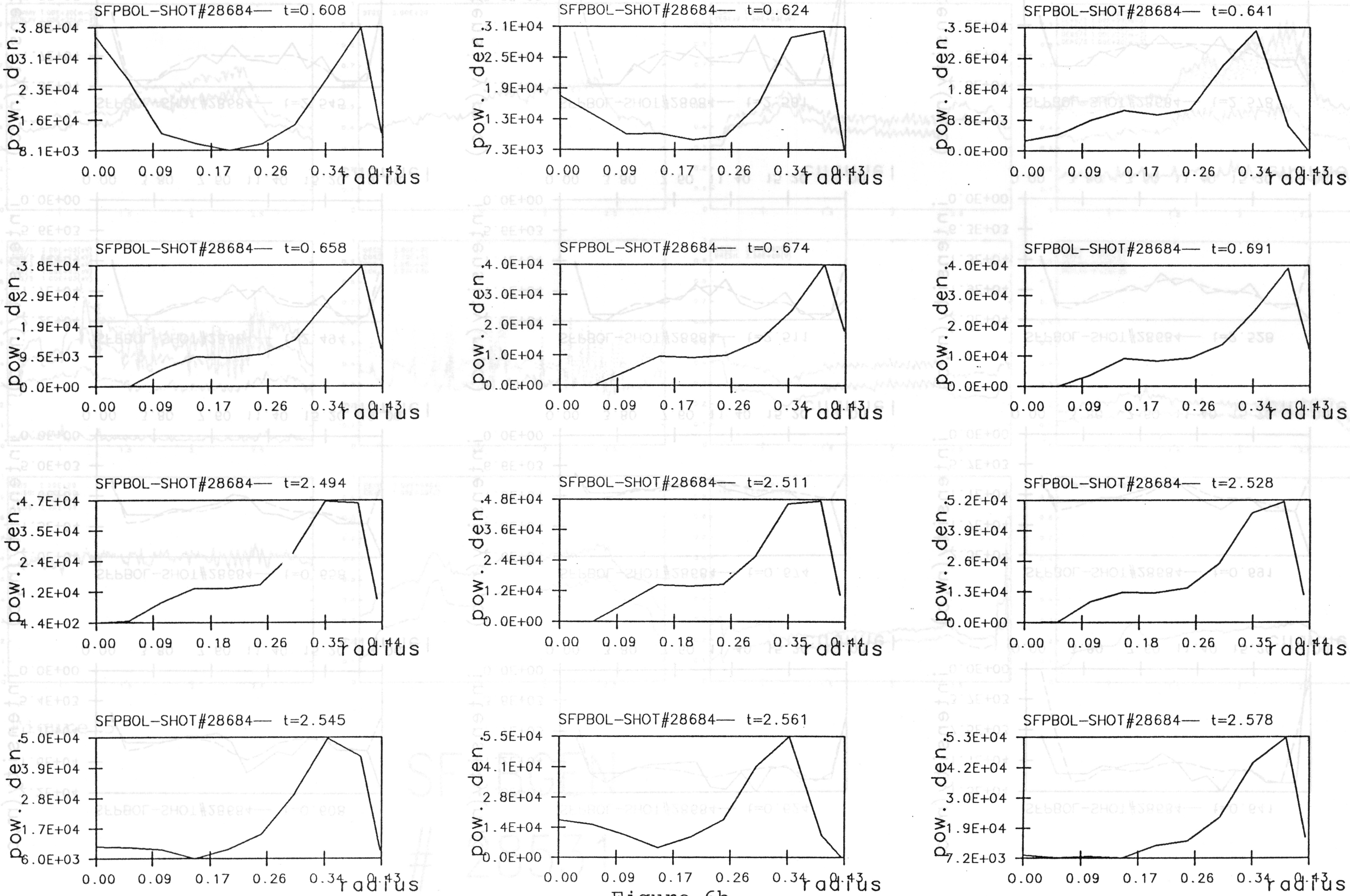

SFPBOL-SHOT#28684— t=0.608
pow. den.
3.8E+04
3.1E+04
2.3E+04
1.6E+04
8.1E+03
0.00 0.09 0.17 0.26 0.34 0.43
radius
SFPBOL-SHOT#28684— t=0.624
pow. den.
3.1E+04
2.5E+04
1.9E+04
1.3E+04
7.3E+03
0.00 0.09 0.17 0.26 0.34 0.43
radius
SFPBOL-SHOT#28684— t=0.641
pow. den.
3.5E+04
2.6E+04
1.8E+04
8.8E+03
0.0E+00
0.00 0.09 0.17 0.26 0.34 0.43
radius
SFPBOL-SHOT#28684— t=0.658
pow. den.
3.8E+04
2.9E+04
1.9E+04
9.5E+03
0.0E+00
0.00 0.09 0.17 0.26 0.34 0.43
radius
SFPBOL-SHOT#28684— t=0.674
pow. den.
4.0E+04
3.0E+04
2.0E+04
1.0E+04
0.0E+00
0.00 0.09 0.17 0.26 0.34 0.43
radius
SFPBOL-SHOT#28684— t=0.691
pow. den.
4.0E+04
3.0E+04
2.0E+04
1.0E+04
0.0E+00
0.00 0.09 0.17 0.26 0.34 0.43
radius
SFPBOL-SHOT#28684— t=2.494
pow. den.
4.7E+04
3.5E+04
2.4E+04
1.2E+04
4.4E+02
0.00 0.09 0.18 0.26 0.35 0.44
radius
SFPBOL-SHOT#28684— t=2.511
pow. den.
4.8E+04
3.6E+04
2.4E+04
1.2E+04
0.0E+00
0.00 0.09 0.18 0.26 0.35 0.44
radius
SFPBOL-SHOT#28684— t=2.528
pow. den.
5.2E+04
3.9E+04
2.6E+04
1.3E+04
0.0E+00
0.00 0.09 0.18 0.26 0.35 0.44
radius
SFPBOL-SHOT#28684— t=2.545
pow. den.
5.0E+04
3.9E+04
2.8E+04
1.7E+04
6.0E+03
0.00 0.09 0.17 0.26 0.34 0.43
radius
SFPBOL-SHOT#28684— t=2.561
pow. den.
5.5E+04
4.1E+04
2.8E+04
1.4E+04
0.0E+00
0.00 0.09 0.17 0.26 0.34 0.43
radius
SFPBOL-SHOT#28684— t=2.578
pow. den.
5.3E+04
4.2E+04
3.0E+04
1.9E+04
7.2E+03
0.00 0.09 0.17 0.26 0.34 0.43
radius

Figure 6b

```
CMS  *  DABREP1 LIS A1

FIRST FILE:
                              OBS    SHOTNO    TIME    CUR

                               1     20000     0.55    340
                               2     20000     0.60    360
                               3     20000     0.65    380
                               4     20000     0.70    380
                               5     20000     0.75    380
                               6     20000     0.90    380
SECOND FILE:
                            OBS    SHOTNO    TIME    DENLAV

                             1     20000     0.55     2.5
                             2     20000     0.60     2.7
                             3     20000     0.65     2.7
                             4     20000     0.75     2.9
                             5     20000     0.80     2.9
                             6     20000     0.85     2.9
                             7     20000     0.90     3.0
MERGED FILE (BY SHOTNO TIME):
                         OBS    SHOTNO    TIME    CUR    DENLAV

                          1     20000     0.55    340     2.5
                          2     20000     0.60    360     2.7
                          3     20000     0.65    380     2.7
                          4     20000     0.70    380      .
                          5     20000     0.75    380     2.9
                          6     20000     0.80     .      2.9
                          7     20000     0.85     .      2.9
                          8     20000     0.90    380     3.0
```

Figure 7