Abstract

This thesis describes the first high-statistics systematic analysis of JET neutron yield and rate measurements obtained by using data acquired with the Magnetic Proton Recoil (MPR) neutron spectrometer. The neutron yield and rate were determined by using the count-rate from the MPR neutron spectrometer together with neutron profile information from other neutron diagnostic systems. This has previously been done manually for a few pulses. To be able to do this in a more systematic way a part of the neutron spectrum evaluation code was extracted and put into a separate custom-made program and modifications were done to extract sets of MPR data automatically. The codes have been used for analysis of a large set of pulses from the deuterium-tritium campaign at JET in 1997. Several results were obtained, the most significant of which was the clear improvement seen when neutron profile corrections were applied. Neutron yield-rates derived from MPR count-rate are shown to be in excellent agreement with other JET neutron diagnostic data.

Systematic determination of the JET absolute neutron yield using the MPR spectrometer

N. Kronborg Pettersson
Department of Neutron Research
Ångström laboratory
Uppsala University
Uppsala, Sweden

April 17, 2003

Contents

List of Figures

List of Tables

Chapter 1

Introduction

We use more and more energy. With growing population and increased quality of life follows an increased energy consumption. The energy sources we depend on today are limited, and many are also harmful to the environment through pollution and climate change.

Promising research on a new, clean and unlimited energy source, which imitates the reactions inside the sun, has been under development for decades. Inside the sun, where the temperature and pressure are high, a nuclear reaction, fusion, takes place. It is the process by which the sun extracts its energy. In fusion, light atomic nuclei are fused together to form new heavier isotopes and in the process energy is released. The goal of fusion energy research is to be able to harness these nuclear reactions and extract energy from them here on Earth.

Both theoretical and experimental work are necessary for the understanding of the processes in the nuclear fusion reactions. The largest experimental facility for fusion research is the Joint European Torus (JET) at Culham near Oxford in England.

There are a number of possible nuclear fusion reactions from which energy is released. The most suitable one, due to its high reaction probability and high amount of released energy, is that of the two heavy isotopes of hydrogen, deuterium and tritium, creating a helium-4 nucleus (α-particle) and a neutron, Eq. 1.1.

$$D + T \longrightarrow {}^4He + n + energy \tag{1.1}$$

In the center of mass system, neglecting the kinetic energy of the reactants, these neutrons have an energy of 14.03 MeV. As is clear from Eq. 1.1, the total neutron yield, or neutron production, from the plasma carries information on the reaction rate, and it is thereby directly connected to the produced fusion power and energy.

One of the devices that exists for measuring the emitted neutrons at JET is the Magnetic Proton Recoil spectrometer (MPR, JET name KM9). It measures neutron emission spectra from plasmas, which provide information on a range of plasma quantities, such as ion temperature, fuel ion densities, α-particle effects and reaction rate. One important characteristic of the MPR spectrometer is that, in principle, it is ab initio calibrated, both with respect to energy and rate. With no need for calibration, the MPR measures in absolute terms. The aim of this work has been to utilize this important characteristic using MPR data for total neutron yield rate measurements which, as mentioned previously, are directly related to the reaction rate, and hence, the power of the fusion reactor.

The MPR neutron yield determination requires data from the neutron profile cameras situated close to the torus at JET. Data extracted from the neutron cameras can be analyzed to determine neutron emission profiles, which can be used to relate the neutron emission as seen by the MPR to the plasma as a whole. Including profile effects have improved these neutron yield measurements significantly.

In a recent study [1], it has been verified that indeed, the MPR proton count rate can be “converted” into a total JET neutron yield that is in good agreement with what other JET systems give. This thesis is an extension of that work and the intention here has been to estimate the uncertainty in the MPR neutron yield measurements and look for possible systematic effects in the data. For this work a much larger data set has been used, obtained from the first major deuterium-tritium experimental campaign (DTE1) at JET, conducted in 1997. Due to the large data set, codes have been developed to make the analysis more automatic.

The aim has been to show that MPR data can provide neutron yields and rates of the same accuracy or better than the presently used technique, namely the fission chambers (JET name KN1) with associated calibration systems (JET names KN2 and KN4). These measurements are estimated to have an uncertainty of 10%[2].

Chapter 2

JET and its neutron diagnostic systems

The most relevant diagnostics for this work have been the fission chambers (JET name KN1), the neutron cameras (KN3) and the Magnetic Neutron Spectrometer (KM9). A brief description of these systems will be given here. The calibration systems KN2 and KN4 will also be mentioned; they are essential for the operation of the fission chambers.

2.1 Fission chambers

The primary instrument for measuring neutron yields and rates at JET is the time-resolved neutron yield monitor. At JET, this system goes under the name KN1. At present, the fission chambers provide the best estimate of the total neutron yield.

The KN1 system comprises three sets of fission chambers, located on the vertical magnet limbs at the horizontal mid-plane on the outside of the torus. Each set consists of a ^{235}U chamber and a ^{238}U chamber. The chambers are lead shielded and have polythene for neutron moderation.

Through fission events, induced by incoming moderated neutrons, a local neutron flux is measured[3]. The local neutron flux is related to the total neutron yield by the calibration methods described below. The chambers can be operated either in pulse or current mode. In pulse mode, the system records individual fission events. In current mode, the ^{235}U chambers give the integrated current from fission events only, whereas with the ^{238}U chambers the current is the sum of the fission event contribution and a larger gamma-ray contribution. Merging the pulse and current responses from the ^{235}U chambers with the current response from the ^{238}U chambers produces a net signal that is relatively insensitive to neutron energy.

The fission chambers are usable for yields of 10^{10} to 10^{20} neutrons per second[4]. For this study, the time-resolution was set to its default value of 10ms. For special investigations a time-resolution of down to 20μs can be used, provided the neutron emission strength is high enough.

A number of techniques are used to calibrate the fission chambers and thereby to relate the measured local neutron flux to the total number of neutrons coming from the plasma. Techniques used include:

(i) the activation system, KN2

(ii) the delayed neutron system, KN4

(iii) the profile monitor, KN3

(iv) measurement of dose-rate within the tokamak during shutdowns

(v) measurement of activation of sections of the first-wall

The activation system and the delayed neutron system are described in section 2.2 and the profile monitor in section 2.3. Due to a weak sensitivity to the shape and position of the plasma within the torus a geometrical correction factor of 1.14, with small shot-to-shot variations, has been applied to the apparent neutron yield[2].

The raw data from the fission chambers are processed and stored in the JET data base system. Various parameters can be retrieved and are referred to as data nodes. The KN1 data node used for this work has the data base name TIN/RNT. This node has been used as a reference comparing with MPR data. It is the most correct of the TIN data nodes, as it includes the general profile correction factor[1].

2.2 Fission chamber calibration systems

The primary instruments for absolute calibration of the KN1 neutron yield monitor are the KN2 and KN4 time-integrated neutron yield measuring systems. In these systems, samples of selected materials are activated by exposing them to neutrons from the plasma. Radiation emitted from by the samples is then detected.

In the KN2 diagnostic, samples of suitable materials are pneumatically transported to locations near the JET plasma, then retrieved and delivered to either a detector or a dump. The detector measures the sample's gamma radiation, which is proportional to the integrated local neutron flux. This local flux is then related to the total neutron flux using neutron transport calculations[4]. By using high reaction-threshold materials, such as Si and Al, the KN2 system is capable of measuring just the 14 MeV neutrons produced in DT reactions. The results are then used to calibrate the neutron counters, such as the KN1 fission chambers.

Capsules containing the irradiation samples are transported using the pneumatic delivery system that is part of KN2.

The KN4 diagnostic comprises two delayed neutron detection systems for the counting of beta-delayed neutrons from fission events in fissionable materials (^{235}U, ^{238}U and ^{232}Th). Capsules with known quantities of fissile material are occasionally sent to the plasma edge. After exposure to JET neutrons, the capsules are sent back to accurately calibrated neutron counter assemblies where the amount of beta-delayed neutron emission is assayed.

The accuracy of the fluence measurements is estimated to $\pm$10% for 14 MeV neutrons[4].

2.3 Neutron camera

The JET neutron profile monitor (KN3) is referred to as the neutron camera and is used to measure the profile and position of the neutron emissivity. The emission of neutrons from the plasma is given in units of emitted neutrons per m^3 and s.

The JET neutron camera system consists of a pair of fan-like neutron collimator arrays placed in the torus hall. The location of the cameras are shown in Figure 2.1. One camera is located above the torus (the vertical camera) and the horizontal/radial camera is situated on the side of the torus. There are nine vertical viewing lines and ten horizontal ones. Each line of sight is referred to as a channel and each channel consists of a collimator equipped with a detector assembly and some local magnetic shielding[5].

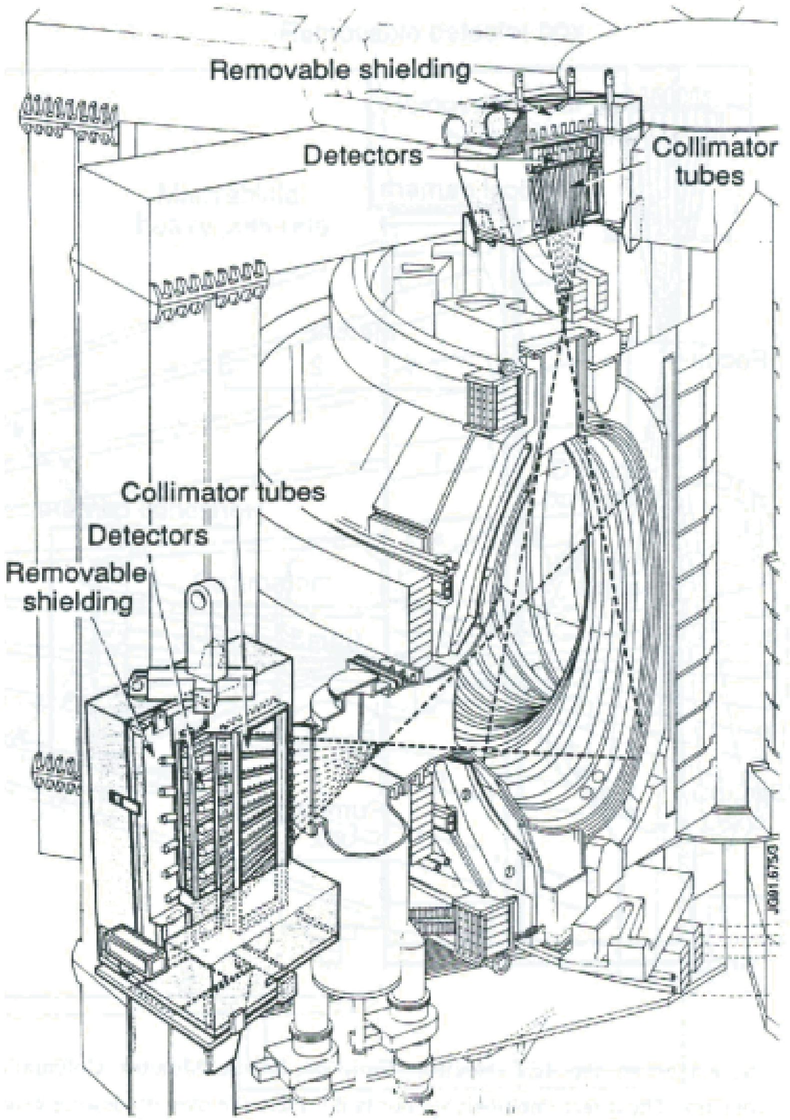

Figure 2.1: KN3 diagnostic.

With the computer code YAPAN (see section 3.3), camera data is processed and time dependent neutron emissivity profiles (NEP) are determined.

2.4 Magnetic Proton Recoil spectrometer

2.4.1 The MPR principle

One of the spectrometers used for detecting 14 MeV neutrons at JET is the magnetic proton recoil spectrometer (MPR), JET name KM9. It combines high sensitivity with good energy resolution and high count rate capability. In this instrument a collimated neutron beam is made to intersect a CH_2 target from which protons are ejected and then momentum separated in a magnetic field, ending at a position sensitive detector, a hodoscope (see Figure 2.2). Detailed response functions based on the well-known characteristics of the MPR such as magnetic field, cross section in target and geometry are used to derive the neutron energy spectrum from the proton distribution measured at the focal plane detector[6]. The scintillator hodoscope consists of 37 detector channels. In addition, three extra scintillators placed behind the hodoscope serve as background monitors. The spectrometer yoke acts as a vacuum chamber to avoid energy loss of the protons over their 3 m flight path.

The MPR neutron energy determination is divided into three steps:

(i) neutron to proton conversion in the scattering foil

(ii) proton momentum separation in the magnetic field

(iii) detection at the focal plane

The first two steps are passive and depend on well-known quantities such as spectrometer geometry, B-field and cross sections[7]. Active measurement is restricted to the last step. The MPR technique allows the instrument to be independently *ab initio* absolutely calibrated with respect to the neutron energy and the neutron flux. This characteristic is of fundamental importance for this work.

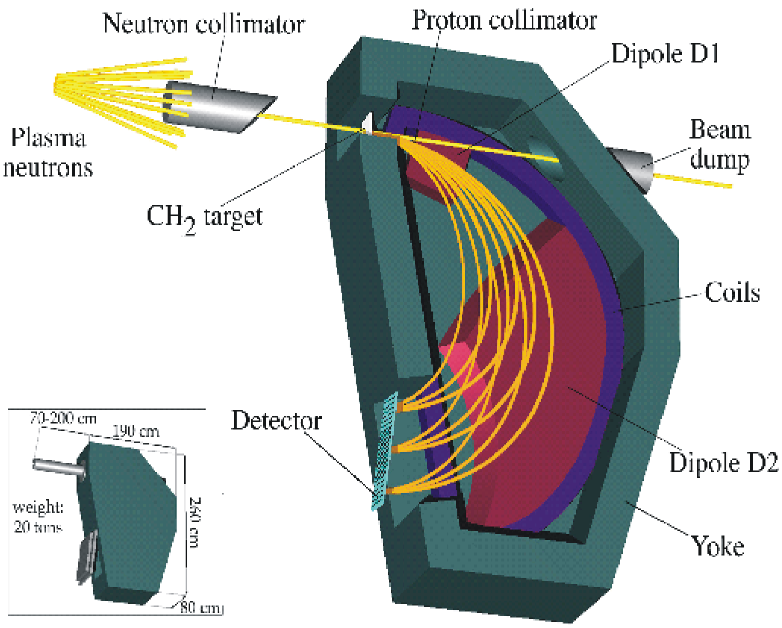

Figure 2.2: MPR overview.

The MPR is a prototype for an ITER[1] diagnostic, aimed at providing information on a range of plasma quantities, such as ion temperature, fuel ion densities and distribution functions, total neutron yield, and α-particle effects.

In order to get good counting statistics, it is important that the MPR "sees" as much as possible of the plasma. The positioning of the MPR is restricted by obstacles, space limitations and interference with other diagnostics. For these reasons, the MPR Line-Of-Sight (LOS) is about $45\,^\circ$ to tangential at the plasma center, and makes a double pass through the plasma center, grazing the central column[8]. The LOS is slightly inclined upwards, by an angle of $4.85\,^\circ$.

The resolution of the spectrometer can be varied by changing the experimental settings for the proton collimator and the foil thickness[8]. Since most of the data during the DTE1 campaign were acquired with a 4% energy resolution setting (using a 18 mg/cm^2 foil and a 52.5 msr proton aperture), this has been the reference case for this analysis. A few pulses with a finer 2.5% resolution (8 mg/cm^2 foil, 40 msr aperture) were also recorded and these have been included for comparison. With a peak amplitude of about 10000 counts (per 10^{18} JET neutrons) and back-

[1]International Thermonuclear Experimental Reactor, the next step tokamak in fusion research. A joint venture between the European Union, Canada, Japan, Russia and recently also China and the USA.

ground uncertainty of 0.2 counts, the sensitivity of the instrument is estimated to be $2{\cdot}10^{-5}$[7].

A more thorough investigation of the efficiency and response of the MPR can be found in ref [1].

2.4.2 The MPR as a neutron yield/rate diagnostic

In a previous study [1], a new method has been developed to enable precise measurements of neutron yields and rates from fusion plasmas using the MPR spectrometer. By using the count-rate from the MPR the number of neutrons in the spectrometer's line of sight has been calculated. To be able to do this, the efficiency and the attenuation (see Figure 2.3) for the instrument have been evaluated. Due to the characteristics of the MPR instrument, this evaluation can be done based only on well-known quantities, such as fundamental corrections, measured B-field maps and geometrical surveys. The number of neutrons in the MPR line of sight (see Figure 2.3) can then be related to the total number of produced neutrons in the plasma by using information on the neutron emission profile from KN3.

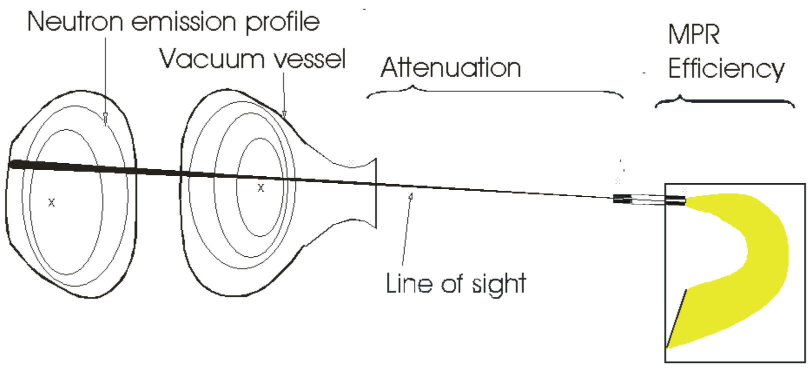

Figure 2.3: MPR line of sight.

Chapter 3

Databases and software

The diagnostics at JET collect enormous amounts of data. These are archived on DAT mass storage and can be retrieved in various ways, through computer code or by interactive software tools.

This section explains the data base system at JET and the software tools that have been used for this analysis. In this work, some codes were modified and new codes were developed. A significant part of this work was devoted to FORTRAN code development.

3.1 JET data base system

Plasma discharges within the JET tokamak device are referred to as shots or pulses. After each pulse, raw data is acquired by the General and Late Acquisition Program (GAP & LAP) and stored in various JET Pulse Files. One of these files is the JPF (Jet Pulse File).

The data stored in the JPF are raw measurement data, together with data acquisition control information[9]. The raw measurement data files are processed and stored in Processed Pulse Files (PPF), which can be re-processed and stored into new PPFs.

The PPF system is a centralized data storage and retrieval system for data derived from raw measurements within the JET Torus and from other sources such as simulation programs[10]. A PPF is a file containing information on one or a group of related diagnostic element(s) or measurements.

Data in the PPF system is arranged by shot number. Each shot contains a number of sequences which may be considered as ’files’, storing data related to the shot. For each shot, the data is grouped into logical areas called DDAs (Diagnostic Data Areas). There are usually one or more DDAs stored in each sequence. Within each DDA are a number of processed measurement results, called DataTypes. The

DataTypes is the basic unit of data stored in the PPF system and contains the actual numerical data associated with the diagnostic or analysis. It is identified using a combination of descriptors, e.g., TIN/RNT.

A given physical quantity within the PPF data is specified by three keywords:

- The shot number (e.g., 42976)
- The DDA name (e.g., TIN)
- The DataType name (e.g., RNT)

There are two types of PPF - Public PPFs and User PPFs (sometimes called 'private' PPFs). Public PPFs are the official repository for the processed data from all the main analysis chains. These PPFs are stored under the PPF user identification "JETPPF".

User PPFs are associated with the individual user identities of their owners. They can be used while developing an application for instance. By writing the data to private PPFs, interference with the public system is avoided - for example the same DDA names may be used in both public and user versions of a PPF, but as the default is to access the public version, other people do not 'see' the development version.

The PPF system is designed to allow easy transfer of data between analysis programs, by providing a unified data storage system and a standard programming interface.

By default when asking for a signal by shot and DDA/DataType name, the most recent version of that signal will be provided, so after re-processing the corrected data will automatically be used. Every time a PPF is reprocessed it gets a new sequence number. Therefore, it is possible to access earlier versions of a given DDA by providing the sequence number of the PPF. In this work, the default has consistently been used.

In this work, PPFs have been accessed with FORTRAN code by using the PPF routines PPFGO, PPFDAT and PPFUID. The purpose of each routine is described below.

- PPFGO - initializes the PPF system
- PPFDAT - reads the requested DataType from a PPF
- PPFUID - sets the user-id when reading or writing user PPFs

For remote access to the JET data base a Citrix ICA[1] Client on a PC with Windows NT has been used to log on to the JAC-system[2].

[1] Independent Computing Architecture

[2] JET Analysis Cluster

Like all JET data, PPFs are migrated automatically from online disks to archive storage tapes, depending on when they were last accessed.

There is now so much PPF data that the vast majority is stored on tape. If a request is made for data on tape it is recalled automatically by the PPF System but may take longer to retrieve than data stored on disk. Once recalled this data will remain on disk until displaced by other recalled data.

The data area for the neutron camera is KN31 and the relevant data nodes have been the neutron peak emission and geometrical parameters for the emission contours. KN3 data are available in time bins of 50 ms, 500 ms and whole shot.

3.2 Display tools

SNAP (Spectroscopic Neutron Analysis Program) is a general tool for retrieval, display and analysis of data, especially developed for the JET neutron diagnostics[11]. It can also display data from other diagnostics, but only limited analysis tools are available for these data. It handles most of the data which are stored in JPF and PPF (both public and private) format. It has also been interfaced to other neutron diagnostic data bases, such as calibration data, nuclear data, computed spectra etc.

Batchplot is another tool for retrieval and display of data. As the name indicates, Batchplot is a data plotting program for use in batch mode. It runs through a list of shots and produces a summary plot of data for each. The format of the summary plot is determined by an input file stored on the JAC. The type of traces which can be included are 1-d plots of PPFs and JPFs. 2-d contour plots and 1-d slices from 2-d profiles are also supported. The default output is a Postscript file which can be viewed with, e.g., Ghostview or just printed out.

3.3 Neutron camera analysis tool - YAPAN

based on a mathematical model of the plasma, the program YAPAN (Yet Another Program Analyzing N3 data) uses data from the KN3 diagnostic to calculate a parameterized form of the neutron emission profile (NEP). These profiles can then be used to calculate neutron emission from any sub-volume of the plasma. The KN3 data are re-binned into 400x50 ms, 40x500 ms and 1x20 s. For the following analysis YAPAN was constantly run in its default mode for each pulse.

YAPAN reconstructs the NEP from camera data by assuming nested elliptical iso-emissivity contours (IEC). The NEP is described by the following parameters:

R_0: The location on the major radius of the innermost IEC.

R_a**:** The location on the major radius of the outermost IEC.

Z_0**:** The distance above the horizontal mid-plane of the innermost IEC.

Z_a**:** The distance above the horizontal mid-plane of the outermost IEC.

a**:** The minor radius of the NEP. The minor radius is defined as half of the distance across the horizontal mid-plane of the NEP as can be seen in Figure 3.1

Ell**:** The ellipticity of the IECs.

α**:** The peaking factor describing how the NEP is peaked.

S_0**:** The central neutron emissivity.

S_a**:** The edge neutron emissivity.

Employing a chi-squared minimization procedure, these parameters are determined by comparing the experimental count rates of the camera channels, with calculated ones, using the profile parameters. It should be noted that the IECs do not need to have the same center and ellipticity. In this work, the following parameters have been fixed: R_a=3.022 m, Z_a=0.3 m and a=1.0 m. The form and location of an IEC between the innermost IEC and the outermost IEC is given by

$$X(\rho) = X_0 + (X_a - X_0) \cdot \rho \tag{3.1}$$

or on a parametric form

$$R = R_0 + (R_a - R_0) \cdot \rho + a \cdot \rho \cdot \cos\theta \tag{3.2}$$

$$Z = Z_0 + (Z_a - Z_0) \cdot \rho + a \cdot \rho \cdot \epsilon \sin\theta \tag{3.3}$$

Here, ρ is the distance to the center normalized to the minor of the outermost IEC radius and X can correspond to either R or Z.

The reconstructed neutron emissivity is a function of the distance to the center. The neutron emissivity is given by the quasi-parabolic function:

$$S(\rho) = (S_0 - S_a)(1 - \rho^2)^\alpha + S_a \tag{3.4}$$

An example of how the IECs are located and the representation of the different variables can be seen in Figure 3.1.

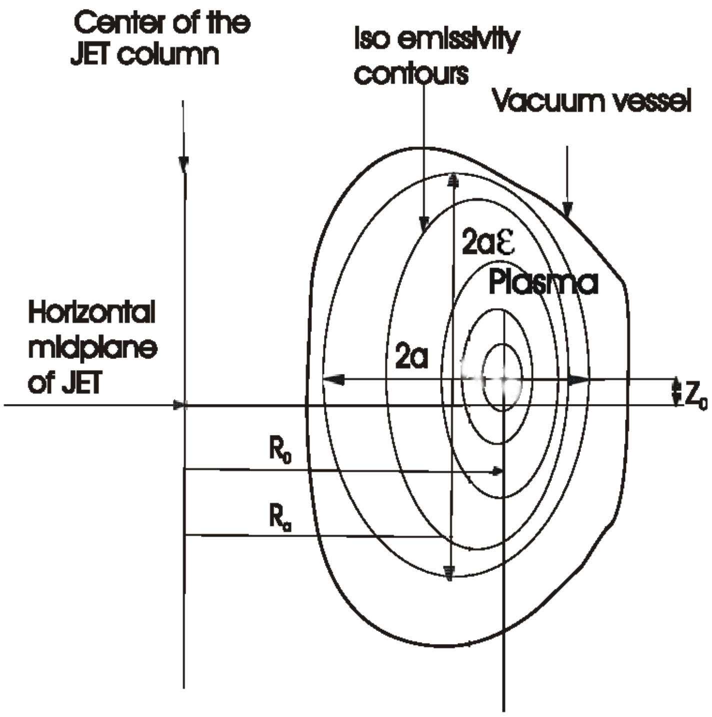

Figure 3.1: Typical IECs for a JET plasma.

The NEPs originally available for the pulses studied in this thesis either did not exist in the JET database, were incomplete or had data that appeared to be wrong (for example too small minor radius), so for the pulses that have been analyses YAPAN has been rerun.

3.4 Line of Sight tool - FPSLOS

3.4.1 General

The FPSLOS code was developed to calculate the collimated neutron flux for different detector positions around JET[12]. Using neutron camera data, analyzed with YAPAN (see section 3.3), it determines the total number of neutrons emitted by the plasma, Y_n(FPSLOS), and the collimated neutron flux entering the LOS of a specific diagnostic.

The input to the code is the NEP from the analysis of the neutron camera data, and geometrical factors relating to physical obstructions. The NEP is extracted from the output of the YAPAN code, see section 3.3.

Using the FPSLOS code together with neutron camera data it is possible to calculate the ratio between the number of neutrons emitted into, for example, the MPR LOS and the total number of neutrons emitted from the plasma.

3.4.2 FPSLOS for the MPR

For this thesis, the flux entering the MPR LOS is referred to as F_n(FPSLOS) and the total yield from the plasma as Y_n(FPSLOS). As was shown in Ref. [1], with the ratio between these quantities together with the MPR neutron flux on target it is possible to determine the total number of neutrons emitted by the plasma on an absolute scale.

For this analysis, the part of the FPSLOS code that determines F_n(FPSLOS) and Y_n(FPSLOS) have been extracted and used in the BANDY code, see section 3.7.

3.5 MPR Analysis Workstation - MAW

3.5.1 General

MAW (MPR Analysis Workstation) is a tool for analysis and presentation of MPR data. MAW is written in FORTRAN and it reads, sorts, corrects and provides interactive graphical presentation and statistical or mathematical analysis. It is also used for the time synchronization between JET and MPR data, which is of great importance here since time resolved JET and MPR data are compared. MAW is a custom written application, but its structure is similar to PAW (Physics Analysis Workstation), which is a system used to analyze and present data at the CERN research laboratory in Switzerland[13]. MAW is based on a hierarchical system of menus and commands using a text-mode user interface. The feature to run files in batch-mode (MUMACS) has been utilized for this systematic analysis.

3.5.2 MAW command ANYD

To extract the necessary data a few MAW functions had to be executed, using the "read" and "extract" commands. These functions were gathered under the new command "ANYD" (Analyze Neutron Yield Data) so that the input file to MAW only consists of one line per pulse and time bin. The new command executed the required sequence of commands and wrote the result to a file.

To match the binning of the YAPAN data, an option for linear interpolation for the endpoints was added in the MAW [14] *Extract* command.

Other optional parameters in the ANYD command are S and B for time synchronization and background correction respectively.

See appendix F for further comments on the code.

3.6 MPR physics analysis

3.6.1 Response functions

The efficiency of the MPR depends on the energy of the incoming neutrons. To calculate the efficiency and to relate the neutron flux on the MPR target with the proton distribution on the hodoscope, the MPR response function must be known. The spectrometer response is dependent on contributions from the target, kinematics and ion-optics as well as the geometry of the MPR and the focal plane detector[15].

The response functions have been calculated using a Monte Carlo simulation code[12]. It relates the MPR recorded proton spectrum to the emitted neutron spectrum. When the response has been determined, it is then used to reconstruct the neutron energy spectrum based on the recorded proton spectrum. The reconstruction procedure is performed in the MS Excel spreadsheet application "MAXL", described in section 3.6.2.

3.6.2 Spectral fitting

After the counts/time bin of the individual channels of the MPR have been extracted and appropriate corrections for time synchronization and background have been applied, a neutron spectrum is fitted to the measured proton spectrum[16]. In principle, this could be done by inverting the response function matrix, but since this is not possible in practice, a minimization routine is carried out in the MAXL spreadsheet. The minimization routine tries to minimize the difference between the calculated and measured proton spectra using the χ^2 quantity as an arbiter. For this work, the most important criteria for a fit has been that the number of "fitted" and measured protons coincide. The exact spectral agreement has been of secondary importance. It could be taken into more consideration, by including components corresponding to different heating modes of the fusion plasma ions.

The MAXL spreadsheet runs with a script interface, which has allowed files with large sets of data to be processed automatically. The input consists of proton counts of individual channels of the MPR and the output contains the calculated neutron rate on the MPR target as well as spectral information such as the goodness of the fit (χ^2) and values of the parameters in the fit.

3.7 Batched Neutron Yield code - BANDY

BANDY is a FORTRAN code that calculates the total JET neutron yield and flux from camera data. The output file from this program contains the important parameters for KN3 and KM9.

Input data for BANDY are pulse number, start- and stop-time, neutrons on target F_n(MPR), MPR-settings, plasma heating components, energy shift and fitted spectrum. A set - batch - of time binned pulses are given as input and processed in one single run.

Output data contains

- pulse number
- start- and stop time
- KN3 derived neutrons on the MPR foil, F_n(FPSLOS)
- MPR measured neutrons on target, F_n(MPR)
- Y_n(FPSLOS)/F_n(FPSLOS) quotient
- KN1 yield rate, Y_n(TIN/RNT)
- KN3 yield rate, Y_n(FPSLOS)
- MPR yield rate, Y_n(MPR)
- MPR and fission chambers yield quotient Y_n(MPR)/Y_n(TIN/RNT)
- χ^2 value for the MPR neutron spectral fit
- MPR proton counts on the hodoscope
- time bin length
- NEP χ^2 fit value from YAPAN

See appendix E for a complete listing of the code.

3.8 Time bin matching tool - MYT

A small FORTRAN code called Match YAPAN Time binning (MYT) matches time binning between JET KN3 data (YAPAN) and MPR data (MAW) and builds a MAW macro (MUMAC). This program takes as input the pulse number, start- and stop times for the selected time interval, and option characters, used to guide the subsequent program execution in MAW and MS Excel. See appendix D for further comments on the code.

Chapter 4

Procedure

The data analysis was performed in a number of steps summarized below and shown graphically in Figure 4.1. The data extraction was divided in two branches, the MPR side and the JET side. The MPR data are stored in Uppsala and at JET, but for this work the analysis was, for the main part, carried out in Uppsala. Therefore, the JET data had to be retrieved remotely as described in section 3.1.

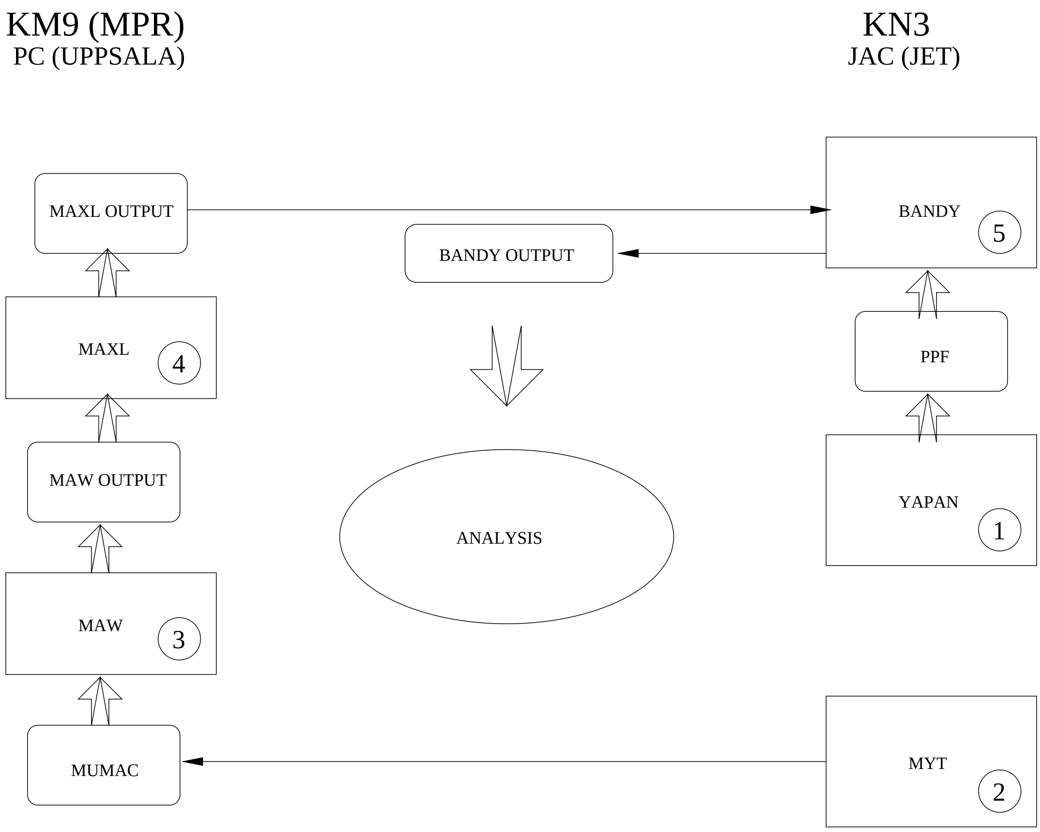

Figure 4.1: Schematic illustration over data analysis.

The steps involved in the analysis are (numbers in parenthesis refer to Figure 4.1):

- pulse selection
- Batchplot - plotting of JET data to select appropriate time windows
- YAPAN - producing the NEP (1)
- MYT - matching KN3 and MPR time binning (2)
- MPR data extraction and corrections using MAW (3)
- Spectral fit of MPR data using an MS Excel script (4)
- BANDY - Combining JET and MPR data (5)
- Statistical analysis of neutron yield results

The MPR branch consists of the extraction of MPR data with the MAW program, a spectral fit of this data with an MS Excel application and the creation of data for use in the BANDY code. On the JET side, fission chamber and neutron camera data were extracted from the JET database and read in to the BANDY program.

A set of pulses from the DTE1 campaign were selected and appropriate time windows found visually with Batchplot. See section 5.1 for further details about the selection process.

YAPAN was run for all individual pulses in the data set.

For the MPR data, the specified time windows were subdivided into time bins of 50, 500 and 20 seconds, matching the time bins of the YAPAN data. This was done with MYT, see section 3.8. The output file (MUMAC) from MYT was read in to MAW.

In MAW, the command ANYD was executed and the mumac read in. The output file from MAW was then read in to the MS Excel application to perform a spectral fit.

Finally BANDY read the file created with the “MAXL” application and its output was then analyzed with MS Excel. The results are presented in chapter 5.

Chapter 5

Analysis and Results

5.1 Data selection and reduction

The pulses studied in this work were produced during the DTE1 campaign. DTE1 contains many pulses with high yields which was necessary for achieving good statistics in this investigation.

Data for pulses taken with malfunctioning data acquisition systems have been avoided by sorting out pulses with missing time synchronization or background correction.

The data analysis was based on a 4% energy resolution (FWHM) setting of the MPR with a magnetic field set to 79.5305% of the maximum current (I_{max}=560A), since during most of the DTE1 campaign it was run in this mode. 67 pulses ranging from JET pulse number 42682 to 43023 were selected as reference. Another eight pulses between 42800 and 42870 with the MPR set to 2.5% energy resolution were also selected for analysis. The duration of each shot has determined the number of bins, or time intervals, that were extracted. A list of the selected pulses is included in appendix C.

Using Batchplot (see section 3.2), suitable time intervals with neutron yield rates of 10^{14} or greater were selected.

NEP fits from YAPAN with χ^2 above 30 (for five degrees of freedom) were filtered out from the data set in the selection process; in this way another two pulses (42887 and 42889) were removed from the dataset. A closer inspection revealed that horizontal camera data were missing for these two pulses.

In the analysis, further selection was carried out. Fits with high χ^2 from the spectral fitting calculations where neglected, as well as time bins with too low proton counts on the MPR hodoscope. Time bins with disruptions were also, in some cases, discarded. However, this latter selection procedure was only partially completed due to time constraints and other priorities.

There are various quality measures to bear in mind when interpreting the results obtained. Firstly, the number of proton counts on the hodoscope of the MPR has a direct correspondence to the statistical error. Generally a larger number gives a smaller error. Secondly, the χ^2 obtained in the fit of the MPR proton counts spectrum can vary, independently of the number of proton counts.

A subdivision of the data sets are shown in Tables 5.1-5.3. The high values of the fitting parameter χ^2 for MPR data seen in column two of Table 5.2 can be due to the simple model with only one thermal heating component included for this analysis. Integrated data were only included in the anlysis for time bin effects comparison with 4% MPR energy resolution for pulses after 42889.

Table 5.1: Subdivision of data set. 4% MPR energy resolution, pulses before 42840.

bin length	χ^2 (MPR)	proton counts	# of bins
50 ms	<2.0	100<8000	463
500 ms	<10.0	4700<70000	51

Table 5.2: Subdivision of data set. 4% MPR energy resolution, pulses after 42889.

bin length	χ^2 (MPR)	proton counts	# of bins
50ms	<5.0	200<1800	136
		1800<31000	714
500ms	<13.0	53000<180000	25
integrated	<26.0	4400<246000	46

Table 5.3: Subdivision of data set. 2.5% MPR energy resolution, pulses after 42889.

bin length	χ^2 (MPR)	proton counts	# of bins
500 ms	<7.0	300<27000	28

An important parameter for the analysis and comparison between MPR and KN1 data has been the neutron quotient Y_n(MPR)/Y_n(FPSLOS), henceforth referred to as the yield-rate quotient Q.

5.2 Bias in camera data

At a quite early stage of the analysis work it was noted that a bias exists in the neutron yield-rate quotient Q for pulses before 42840. Therefore, it was decided to exclude data before a breakpoint at pulse 42870. In this way most of the high yield pulses were included. The origin of this bias is at present unclear. The shift can be seen in Figure 5.1, where data for 50 ms time bin and 4% energy resolution for the MPR were used. Although the scatter of the data points is quite large, a clear distinction exists between "early" pulses (before 42800) and "late" pulses (after 42800), having Q-quotients around 0.9 and 1.0 respectively.

The grouping of data along vertical "lines" is due to the fact that each pulse contains several data points coming from the subdivision into smaller time intervals.

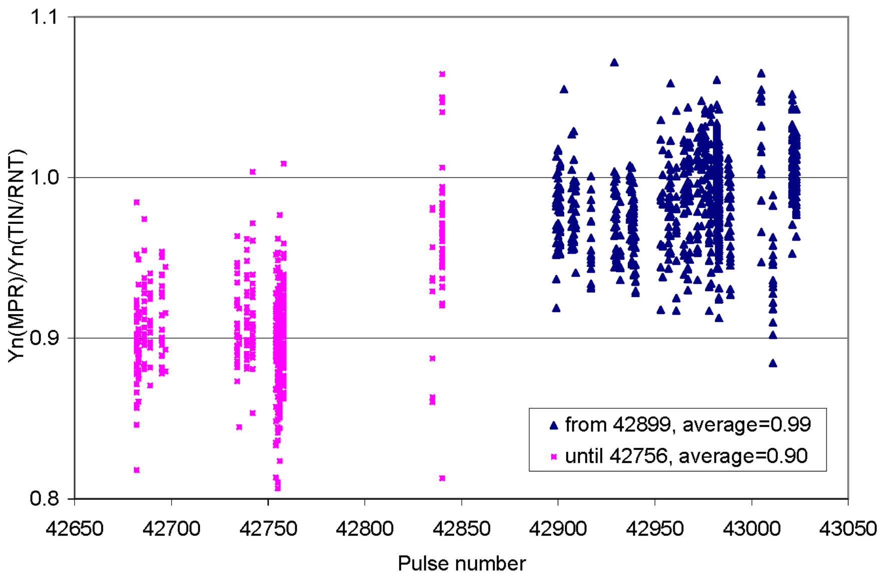

Figure 5.1: Neutron yield-rate quotient for 50 ms bins with the MPR set for 4% energy resolution.

5.3 Normalization in derived camera data

By default, YAPAN normalizes camera data with respect to YRNT from the fission chambers[12]. Since effects of this normalization should cancel out when dividing Y_n(FPSLOS) with F_n(FPSLOS), this normalization was initially left unaltered. However, on closer examination it was found that the normalization was done somewhat differently in F_n(FPSLOS) and Y_n(FPSLOS), resulting in a much larger spread than expected in the values of the neutron yield-rate quotient Q. To correct this normalization discrepancy, it was necessary to restore a consistent way of normalizing the results.

Technically, this was achieved by replacing Y_n(FPSLOS) with TIN/RNT, since both parameters represent the same physical quantity. However, the correction is still not perfect since F_n(FPSLOS) had been normalized with YRNT, and TIN/RNT was used instead. TIN/RNT is based on YRNT, but has a coarser time resolution and accuracy. The reason for using TIN/RNT was that the retrieval is quite straightforward, as opposed to that of YRNT. For some discharges, a coarse approximation has been made where TIN/RNT goes abruptly to zero, which is why data analysis from these regions should be avoided.

In Figure 5.2, the binned data is plotted as function of pulse number, and the larger spread of the data in the set with inconsistent normalization is clearly displayed. Removal of the normalization inconsistency resulted in a marked reduction in the scatter.

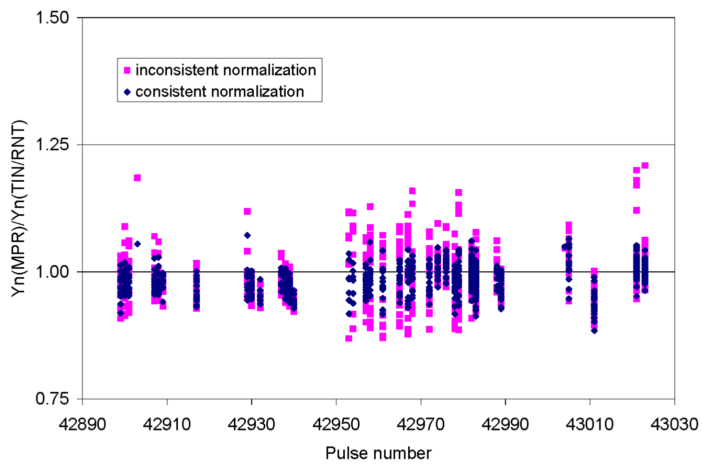

Figure 5.2: Neutron yield-rate quotient with consistent and inconsistent normalization plotted vs pulse number. 50 ms time bin with 4% MPR energy resolution setting.

5.4 Profile effects

As a first step in this systematic study, we want to establish to what degree an improvement is obtained when profile effects are taken into account. In this analysis a reference plasma configuration had to be chosen; the choice is in principle quite arbitrary, but to facilitate comparison between data with and without profile effects it is an advantage to choose a profile that gives the same neutron yield quotient Q as for the full analysis. As will be seen later, this quotient happens to be very close to unity. This fixed plasma shape was then used to analyze all MPR data; in practice this means that the fraction of neutrons seen by the MPR was set as a fixed value.

The result from such a profile comparison, using data with 50 ms time bin and 4% energy resolution, can be seen in Figure 5.3 where the neutron yield quotient Q is plotted against pulse number and in Figure 5.4 where the MPR yield is plotted against TIN/RNT.

In Figure 5.3 the binned data is sorted under each pulse, and the spread is clearly displayed on an individual pulse basis. In Figure 5.4 the binned data is displayed according to MPR and KN1 neutron yield-rates respectively. Already from these plots, it is quite clear that including profile effects generally produce MPR yields with less deviation from TIN/RNT.

The average for the quotient is 0.99 without and 1.02 with profile correction, and the corresponding standard deviations are 10.5% and 2.8% without and with profile correction included, respectively.

A frequency distribution of the yield quotients Q is shown in Figure 5.5. Not only has the inclusion of profile information the effect to reduce the width, but it also removes most of the asymmetry seen in the data using the universal profile.

The full analysis, including bin-by-bin profile corrections, was used for the rest of the analysis presented in this thesis.

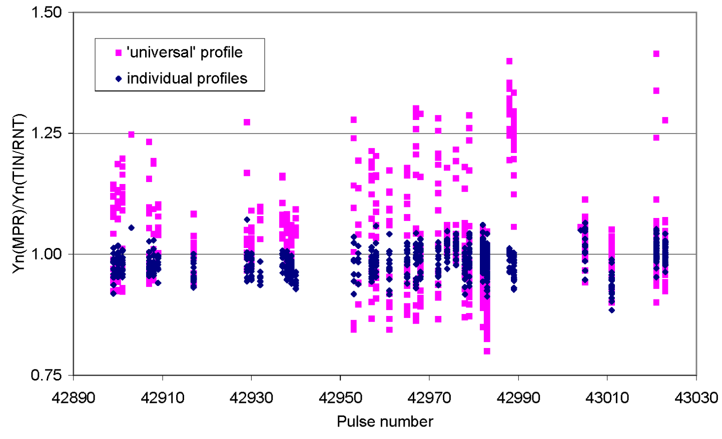

Figure 5.3: Profile correction, neutron yield-rate quotient vs pulse. 50ms time-bin with 4% MPR energy resolution setting and magnetic field set to 79.5305% of I_{max}.

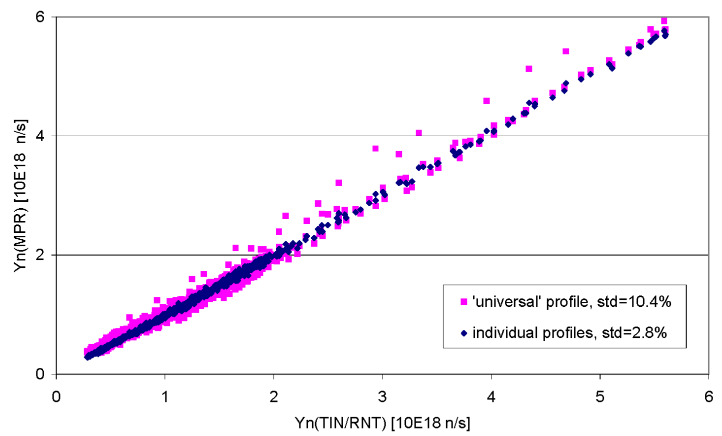

Figure 5.4: Profile correction, MPR vs fission chambers neutron yield-rates. 50 ms time-bin with 4% MPR energy resolution setting and magnetic field 79.5305% of I_{max}. Data series representing with and without profile correction.

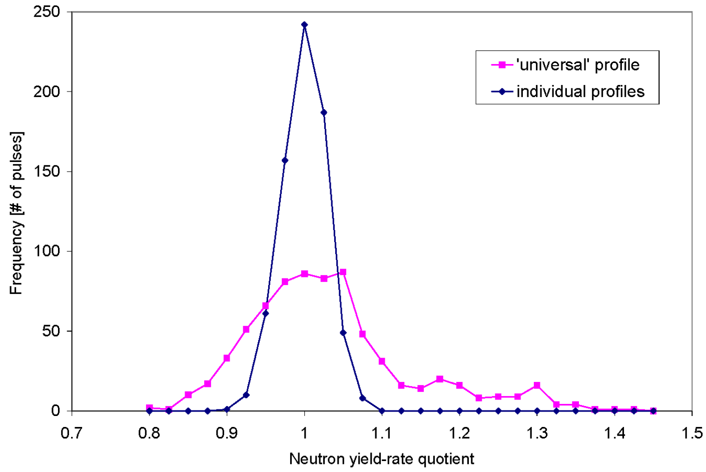

Figure 5.5: Profile correction, frequency distribution histogram. 50ms timebin.

The error estimation made in the previous work[1] gave a systematic error of 3.8% and a statistical error of 2.2% for 3900 proton counts. This estimate includes contributions originating from the MPR efficiency, beam attenuation, the MPR LOS, geometry and the NEPs. In general terms, the present result seems to agree with this estimate. A more detailed comparison of the estimated and observed spread in the data is presented in the next section.

5.5 Count rate effects

To investigate the correlation between actual and estimated uncertainties in the yield data, a proton counts sensitivity analysis was performed, as shown in Figure 5.6. Here, the neutron yield quotient Q is plotted as a function of MPR proton counts in each studied 50 ms time bin. A statistical analysis was then performed on these data, grouped in five different sets (see Table 5.4 for details).

As seen in column two of Table 5.4, the observed standard deviation varies from 9.8% for the group with less than 1000 proton counts to 2.4% for the group with more than 6000 proton counts. The slope of a line fit is slightly positive for the group with proton counts above 3600. For the data with less than 1000 proton counts/bin, the quotient drops 10-15 percent. This latter discrepancy between MPR and KN1 measurements could have been caused by 2.5 MeV neutrons, which are observed by the KN1, but not the MPR.

The expected standard deviation, based on the number of MPR counts alone, is shown in column 3 in Table 5.4. Column 4 shows the difference between the measured (experimental) and the statistical standard deviation. This contribution, originating from other sources of random error, was estimated to 2% in the previous work[1], which, except for the group with the lowest counts, agrees well with the observed 2.0-2.2%.

The columns 5 and 6 of Table 5.4 show a correlation comparison between the MPR and KN1 data with the parameters from a straight-line fit for the data groups. For the group with proton counts between 2000 and 3600, the correlation coefficient, k, deviate slightly more from unity than expected systematic uncertainties of Y_n(TIN/RNT) and Y_n(MPR). An explanation for this has not been found.

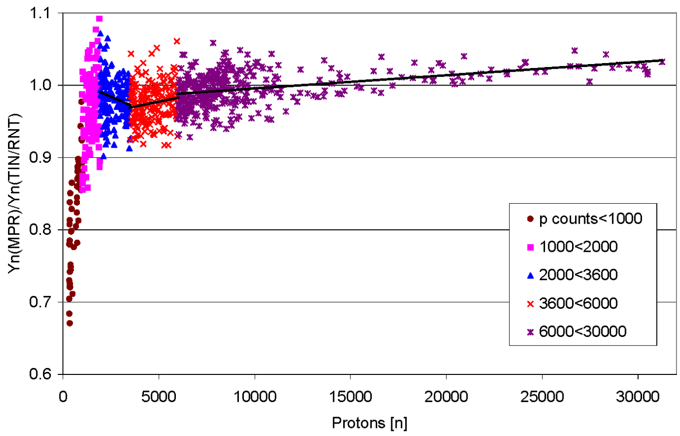

Figure 5.6: Counts comparison. Neutron yield-rate quotient vs hodoscope proton counts, 50 ms binned

Table 5.4: Grouping of data for counts comparison.

# of proton counts	σ exp. (%)	σ stat. (%)	other random contr. (%)	$Y_n(MPR) = k \cdot Y(TIN/RNT) + m$ k	m (10^{15})
<1000	9.8	6.4			
1000<2000	5.0	3.2	3.8	0.98	-4
2000<3600	3.1	2.2	2.2	0.94	20
3600<6000	2.7	1.7	2.1	0.98	-2
6000<30000	2.4	1.3	2.0	1.03	-60

5.6 Time bin effects

A comparison of the results on average yield quotients and widths for data with time binning intervals of 50 ms, 500 ms and 20 seconds are shown in Figure 5.7. Averages and standard deviations are shown in Table 5.5.

Column three of Table 5.5 shows that the average neutron yield-rate quotient Q is near unity for the three time bin lengths. This effect can also be seen in column five and six, which shows the slope of a line fit performed on the data sets. The offset introduced when the fitted line was not forced through zero was insignificant. Column four shows the largest spread for the smallest time bin of 50ms. It should be noted that a large part of the 500 ms data set had high statistics, which may have contributed to the small spread.

The reason for the few number of 500 ms time bins is that half of the time bins were in regions with very low emissivity and were therefore excluded. As this effect was not so pronounced for the integrated time bins in that case, nearly all pulses could be included.

Table 5.5: Neutron yield-rate quotient, time binning comparison.

				$Y_n(MPR) = k \cdot Y(TIN/RNT)$	$Y_n(MPR) = k \cdot Y(TIN/RNT) + m$	
bin-length	# of bins	avg	σ (%)	k	k	m (10^{15})
50 ms	714	0.99	2.8	1.00	1.02	- 40
500 ms	25	0.99	1.5	0.99	0.99	-7
integrated	47	0.99	2.4	0.99	1.00	-10

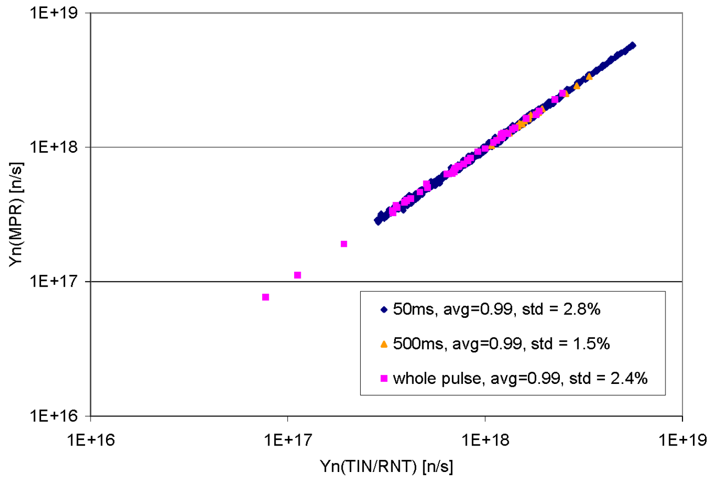

Figure 5.7: Comparison of neutron yield-rate quotient for data using 50 ms, 500 ms and whole-pulse time bins.

5.7 MPR-settings

The 500 ms time bin has been used when studying the effects of varying MPR settings. 25 data points with the MPR 4% resolution setting have been included in this anlysis, with proton counts per time bin ranging from 53000 to 180000. For the 2.5% dataset with 28 data points, the counts per time bin varied between 300 (very low) and 3000. The small number of data points in the latter set is due to the fact that few pulses were actually recorded with this setting. The spread in the quotients for these two data sets can be seen in Figure 5.8.

In columns two and three in Table 5.6 it can be seen that the neutron yield-quotient Q was 0.99 and the standard deviation was 1.5% for the 4% MPR resolution setting. For the 2.5% setting Q was 1.01 and the standard deviation was 7.1%. The small number of high resolution data points can have contributed to the larger spread of the 2.5% data set. Line fits, allowing for an offset, of the 4% and the 2.5% settings gave 0.99 and 1.04 respectively as seen in columns four and five in Table 5.6.

Table 5.6: Neutron yield-quotient Q for 500 ms binned data. 2.5% and 4% MPR energy resolutions.

MPR res. (%)	avg	σ (%)	k	m (10^{14})
2.5	1.01	7.1	1.04	-2
4	0.99	1.5	0.99	-70

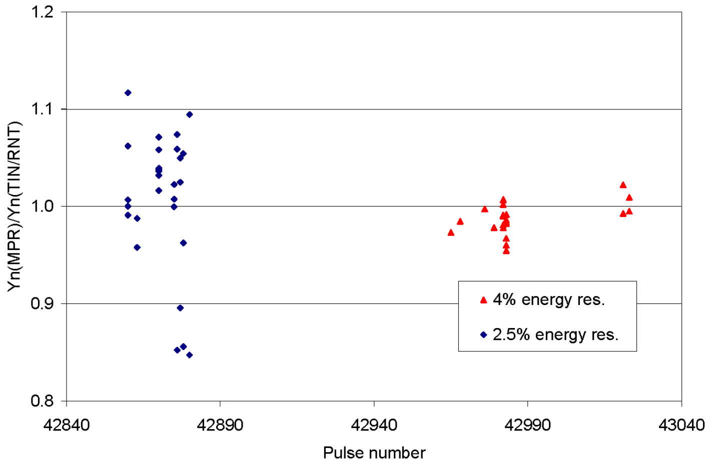

Figure 5.8: MPR energy resolution settings 4% and 2.5%. 500ms binned. Neutron yield-rate quotient on y-axes.

Chapter 6

Discussion

As seen by the bias discussed in section 5.2, the type of analysis performed here depends to a large degree on the availability of correct and consistent camera (KN3) data. Considering the nature of this work, this is an expected result, but it is nevertheless important to emphasize.

Maybe the main results of this study is the clear improvement observed when profile corrections are applied to the MPR data, as discussed in section 5.4. With a significant reduction both in the spread of the data (as compared to the KN1) and the assymetry in this spread, this is clearly an indication that this type of analysis is both necessary and fruitful.

It is interesting to note that the proton counts sensitivity study, section 5.5, gives supporting evidence for the somewhat arbitrary assignment of a 2% uncertainty contribution from “other random effects” than the MPR counting statistics introduced in the previous study[1]. The only deviating result is for the lowest count rates, where possibly effects from 2.5 MeV neutrons could come into play. This should be investigated further.

Regarding the detailed studies in sections 5.6 and 5.7, some comments should be made. In general, there seems to be no clear dependency in the results on the time bin selected. The very small spread in the data for 500 ms with 4% MPR resolution may be because of the few pulses that were used here, all with 50000 or more proton counts on the hodoscope. Due to the limited number of pulses for the 2.5% setting with enough neutron yields it was difficult to draw any conclusions.

In this work the data node TIN/RNT has been used for convenience, but preferably YRNT should have been used. TIN/RNT has coarser time bin resolution and goes abruptly to zero when disruptions in the plasma occur. The value of TIN/RNT then becomes too small and therefore a more thorough investigation of the data set including the removal of time bins near regions of disruptions should be performed. The effect is accentuated at small time-scales where the difference between YRNT and TIN/RNT becomes pronounced[12].

A further development in the systematical analysis would be to write a program which connects remotely to JET. Data has successfully been extracted remotely into programs such as MATLAB, and doing it in MS Excel, as would be the natural choice here, should be possible. The aim should be to run scripts in MS Excel and retrieve parameters from the JET PPF database. The MAW side would probably still need separate execution since it is not prepared to be executed externally.

Chapter 7

Conclusions

This thesis presents a first systematic data analysis of neutron yields and rates derived from data recorded with the MPR neutron spectrometer for a number of high neutron yield pulses of the JET deuterium-tritium campaign (DTE1), performed in 1997. Both time-resolved and time-integrated data were studied.

The analysis has made the importance of applying profile corrections from the neutron profile monitor diagnostic to the MPR data very clear. Both the spread in the data (as compared to the KN1) and the assymetry in this spread were significantly reduced.

The MPR neutron yield and rate measurements are in good agreement with existing (KN1) diagnostic results. The uncertainty in the MPR yield/rate estimates have a firm basis in counting statistics and well-known systematic effects. Rate uncertainties at the level of 5% or lower are obtained in this analysis.

Furthermore, there is no indication in this analysis of a statistically significant dependence on count-rate, time bin or MPR settings.

Based on these results, we conclude that the presented method works and that the MPR is well suited for absolute neutron yield and rate measurements at JET.

Chapter 8

Acknowledgments

I want to express my gratitude to all people at INF, especially those in the fusion diagnostics group. It has been a great help for me to confer with my nearest co-workers Sean, Hans and Henrik. To my supervisor Göran I want to express my greatest thanks having guided me through the entire process. Finally I wish to thank the Swedish Research Council for making this work possible and Jan Källne for giving me the opportunity to do the work at INF.

Bibliography

[1] H. Sjöstrand. Determination of the jet neutron rate and fusion power using the magnetic proton recoil neutron spectrometer. Master's thesis, Department of Neutron Research, Uppsala University, 2003.

[2] JET Radiation dynamics group. *Online description of KN1, time resolved neutron yield monitors*.

[3] O.N. Jarvis et al. Guide to the diagnostics used for fusion reaction studies at JET. Internal Report JET-IR (84) 02, JET Neutron Diagnostics Group, 1984.

[4] O. N. Jarvis and M. J. Loughlin. Guide to the diagnostics used for fusion reaction studies at JET. JET-TN (90) 01, JET Neutron Diagnostics Group, 1990.

[5] J. M. Adams et al. *Nucl. Instrum. Methods Phys. Res.*, Sect A 329 (1993):277, 1993.

[6] S. Forsberg. Neutron emissivity profile camera diagnostics considering present and future tokamaks. Master's thesis, Department of Neutron Research, Uppsala University, 2001.

[7] G. Ericsson et al. Neutron emission spectroscopy at JET - results from the magnetic proton recoil spectrometer (invited). *Rev. Sci. Instr.*, 72(1):759–766, 2001.

[8] S. Conroy, G. Ericsson, and the NPF collaboration. Neutron emission reference characterization of selected JET DT discharges based on MPR spectrometer measurements. UU-NF 99#09, Department of Neutron Research, Uppsala University, 1999.

[9] D. Wilson. *GETDAT User Manual*. JET web user pages, November 2002.

[10] JET web user pages, http://users.jet.efda.org/pages/data-ppf-jpf/ppfuserguide/introduction.shtml. *PPF System User Guide*, 2001.

[11] *JET neutron diagnostics, online description of neutron diagnostics software*. http://w3.ed.jet.efda.org/pages/neutron/software.html/.

[12] S. Conroy. Private communication.

[13] R. Brun et al. *PAW-Physics Analysis Workstation*. CERN Program Library, August 1989.

[14] G. Ericsson. Private communication.

[15] G. Gorini et al. Analytical model for characterization of the MPR spectrometer. UU-NF 95/#1, Department of Neutron Research, Uppsala University, 1995.

[16] H. Henriksson. Private communication.

Appendix A

Program technicalities

A.1 MAW code

The source code of MAW has been written in FORTRAN. For this work *Compaq Visual Fortran version 6.0* was used for developing the code.

The help text from MAW for the “ANYD” command developed for this work is shown in fig A.1.

```
Usage: TOP/NK/ANALYZE  FNAME TMIN TMAX WRITEFILE [ CHOPT ANAL_TYPE ]

Description: Analyzing neutron yield data

Parameters:
 1:  FNAME            char      pulse number
 2:  TMIN             real      start time (default 0.0)
 3:  TMAX             real      stopt time (default 0.0)
 4:  WRITEFILE        char      File to write to
     +
 5:  CHOPT            char      Options
 6:  ANAL_TYPE        int       Excel analyze option. Default=1.

Possible CHOPT values are:

    S Impose time syncronization.
    B Impose background correction.
    I Use interpolation at ends of timeinterval.
    V Verbose mode.
```

Figure A.1: MAW help for anyd command

A.2 Ms Excel script

The custom built MS Excel application for the MPR analysis, MAXL2002e, can read a file that has previously been created with MAW, using the command 'ANYD'. This data in the form of a proton position histogram is fitted when the 'Do analysis' button is pressed. The fit is performed with a number of heating components specified in the file with a certain number (where 0 corresponds to the presently defined model in the MAXL worksheet). This number refers to a global list, stored in the file 'MAXL-Archive.xls'. When data is fitted, the results are stored in a second file with the name specified on the 'Analyse' worksheet. The path should point to the directory where the MAW-generated file is located, which is also where the results file is stored. The format of the MAW-file and the results file can be seen in example files ('MAWout.txt' and 'MAXLout.txt') in 'Inf/Public/AnalysisSheets'.

The analysis models in the file 'MAXL-Archive.xls' can be read by pressing the 'Read model' button. Then all parameters for the fit are copied to the analysis sheet. New models prepared in the 'Analyse' worksheet can be added by entering a name in the Model description cell and pressing 'Store model'.

MAXL2002e uses the files RF and MAXL-Archive, all stored at //Tsl-wnt-server/Inf/Public/AnalysisSheets/

Appendix B

Procedure for MPR neutron yield-rate determination using MYT and BANDY codes

This is a guide explaining how to determine MPR neutron yields and rates using the codes MYT and BANDY together with other relevant tools. The codes have been developed to run through large sets of data. Since codes and tools on different computer systems are needed for the analysis, some file transfer is unavoidable.

The camera data is retrieved from the JET data base system via the JAC (UNIX-environment). The MPR data extraction is performed on a PC with MAW and the MAXL spreadsheet installed and access to MPR data.

These are the relevant codes:

- SNAP - read and visualize pulses
- YAPAN - create NEPs from camera data
- MYT - create files for MAW
- MAW command ANYD - retrieve MPR data
- Excel spreadsheet MAXL - MPR proton to neutron conversion
- BANDY - profile corrections from camera data and output file for analysis

To run the codes “myt” and “bandy”, a directory on the JAC system should include a subfolder “results”, the FORTRAN source code files “myt.f” and “bandy.f”, the executable compilation files “domyt” and “dobandy”, and the executable codes “myt” and “bandy”. Data files are stored in the “results” directory.

On the PC, a current version of MAW needs to be installed and the spreadsheets MAXL2002d and MAXL-Archive placed in a folder.

Below is a step-by-step description of the analysis procedure.

(1) Run pulses of interest in YAPAN to create NEPs.

(2) Find time regions for each pulse with SNAP or BATCHPLOT.

(3) Create a .txt file with pulse number, start- and stop for time regions, MAW parameters option (normally ”bsi” for background, CODAS time synch and integrated) and MAXL parameter option (1 for thermal component only, 0 for manual selection, other options see MAXL). Separate entries with space.

(4) Run MYT to divide time interval into subintervals of 50ms, 500ms and whole pulse.

    ”Enter file to read from:” myt_ex.txt

    ”Enter file to write to:” mum_ex.txt

    ”Enter filename for MAW to write to:” maw_ex.txt

(5) Transfer mum_ex.txt to MAW directory.

(6) Open mum_ex.txt with Notepad and save as mum_ex.mumac (select All files to avoid the .txt extension being added).

(7) In MAW, run ”exec mum_ex.mumac”. This is one of the time-consuming parts. (The .mumac extension is not necessary.)

(8) Open the MAXL spreadsheet. (Dialogs may appear. Choose “Enable Macros” and “Update all linked information”. The RF file should be directly on the C: drive.) Specify the path to the Data and results files. Click on “Do analysis & store reults” to perform fits.

(9) Transfer mum_ex.txt back to the results directory on the JAC.

(10) Run BANDY and read in file and specify output file.

    “Enter file to read from:” ex_ex.txt

    “Binsynched? y/n:” y (n if MYT has not been used.)

    “Enter file to write to:” res_ex.txt

(11) Import BANDY output file into MS Excel or other application for analysis.

The file transfer sequence and suggested file name structure:

```
myt_*.txt -> MYT -> mum_*.txt -> mum_*.mumac -> MAW ->
maw_*.txt -> MAXL -> ex_*.txt -> BANDY -> res_*.txt
```

Appendix C

List of included pulses

2.5% res.	4% res. before 42887	4% res. from 42887	
42800	42682	42887	42967
42801	42683	42889	42968
42802	42686	42899	42972
42808	42689	42900	42974
42811	42691	42901	42976
42812	42695	42903	42978
42814	42697	42907	42979
42817	42734	42908	42982
42818	42735	42909	42983
42820	42739	42917	42988
42821	42742	42929	42989
42822	42754	42930	42991
42823	42755	42932	42992
42824	42756	42937	42993
42825	42757	42938	43002
42826	42758	42939	43003
42827	42835	42940	43004
42828	42840	42953	43005
42829		42954	43011
42830		42957	43021
42831		42958	43023
42832		42961	
42870		42965	
Σ 23	Σ 18	Σ 44	

Table C.1: Pulse list.

Appendix D

Match YAPAN Time bin (MYT) code

```
*     Synchronizes binning after YAPAN, and builds MAW batchfile (MUMAC).
*     Input with pulsenumber, start- and stoptime, option characters (chopts),
*      for MAW and excel (excelpar).
*     Output with MAW command and intervals divided into 50ms, 500ms, 20s bins.
*     Created by Niklas Kronborg Pettersson for INF, Uppsala University
*     ------------------------------------------------------------------------------------

      PROGRAM MYT
      implicit none
      character(len=1) :: rw='R'
      character(len=4) :: dda='KN31',dtype
      character(len=9) :: userid='hsjost',chopts
      character(len=25) :: mawfile
      character(len=80) :: tempname
```

```
character(len=8) :: directory='results/'
integer :: shotn,ierr,nx,nt,tpos,iseq,i,j,k,excelpar
integer,parameter :: ntmax=20000,nxmax=20,ndmax=nxmax*ntmax
real :: d(ndmax),x(nxmax),t(ntmax),tmax,tmin,tdiff
DO                        !Input file must exist.
   write(*,*)'Enter file to read from: '
   read(*,*)tempname
   tempname=directory//tempname
   open (1,file=tempname,status='old',iostat=ierr)
   IF (ierr.eq.0) EXIT
   write(*,*)'The file does not exist.'
END DO
DO                        !Create output file.
   write(*,*)'Enter file to write to: '
   read(*,*)tempname
   tempname=directory//tempname
   open (2,file=tempname,status='unknown',iostat=ierr)
   IF (ierr.eq.0) EXIT
END DO
open (3,file='results/fnb_err_file.txt',status='old'
.     ,position='append',iostat=ierr)
IF (ierr.ne.0) THEN
   open(3,file='results/fnb_err_file.txt',status='unknown')
END IF
write(*,*,advance='no')'Enter filename for MAW to write to: '
read(*,*)mawfile
DO i=1,10000              !Repeated reading from input file.
```

```
      READ(1,*,IOSTAT=ierr,err=100,end=110)(shotn,tmin,tmax
     .     ,chopts,excelpar)
      dda='KN31'
      userid='hsjost'
*     Integrated data.
      dtype='PEMI'
      nx=nxmax               !Specifies maximum of x-array. Neccessary.
      nt=ntmax               !Specifies maximum of t-array. Neccessary.
      CALL PPFUID(userid,rw) !Initialize PPF system.
      CALL PPFDAT(shotn,iseq,dda,dtype,nx,nt,d,x,t,ierr) !Retrieve data.
      IF (ierr.ne.0) THEN
         write(3,*)'Error calling ppfdat',shotn,tmin,tmax,dtype,ierr
         GOTO 990            !Skip and continue with next input line.
      END IF
      write(2,101)'anyd p',shotn,40.0,72.0,mawfile,chopts,excelpar
101   format(a9,i5,2f8.4,a26,a10,i2) !Write integrated bin to file.
      dtype='PEML'
      tdiff=0.505              !Discriminator for last 500ms bin.
*     50ms and 500ms data.
      DO k=1,2
         nx=nxmax            !Specifies maximum of x-array. Neccessary.
         nt=ntmax            !Specifies maximum of t-array. Neccessary.
         CALL PPFUID(userid,rw) !Initialize PPF system.
         CALL PPFDAT(shotn,iseq,dda,dtype,nx,nt,d,x,t,ierr) !Retrieve data.
         IF (ierr.ne.0) THEN
            write(3,*)'Error calling ppfdat',shotn,tmin,tmax,dtype
     .           ,ierr
```

```
            GOTO 990         !Skip and continue with next input line.
         END IF
*     Find position of time in t-vector that is nearest tmax.
         DO j=1,nt           !Condition for last timebin.
            IF (abs(tmax-t(j)).lt.tdiff) THEN
               tpos=j
               GOTO 222
            END IF
         END DO
 222     CONTINUE
         DO                  !Condition for first timebin.
            IF (t(tpos-1).lt.tmin) EXIT
            write(2,102)'analyze p',shotn,t(tpos-1),t(tpos),mawfile
     .           ,chopts,excelpar !Write to file.
 102        format(a9,i5,2f8.4,a26,a10,i2)
            tpos=tpos-1
         END DO
         dtype='PEMS'
         tdiff=0.06          !Discriminator for last 50ms bin.
      END DO
 990  CONTINUE               !Minor error. Read next line of input.
      END DO
 100  write (*,*)'Error reading from file.',ierr
      GOTO 999
 110  write (*,*)'End of input file.'
 999  CONTINUE               !Major error. Finish program.
      rewind(3,iostat=ierr)  !Attempt to continue on old error file.
```

```
 read(3,*,iostat=ierr)
 inquire(3,name=tempname)
 IF (ierr.eq.0) write(*,('a26,a40'))
.     'Error(s) written to file: ',tempname
 close(1)                  !Closing files.
 close(2)
 close(3)
 END PROGRAM
```

Appendix E

BANDY code

```
*      Code for calculating FnFPSLOS, YnFPSLOS and YnMPR.
*      Created by Niklas Kronborg Pettersson for INF, Uppsala University
*      030127 Added function find_chi2 for extracting chi2 parameter, a test for YAPAN fit.
*      Handles timebin synched data and non-synched data for free choice of binlength.
*      Timebin synched data by default from YAPAN consist of 50ms, 500ms and integrated.
*      main program - bandy
*      functions: find_chi2, bin_type, find_binlength, em_profile_fcn, histsum_1d
*      subroutines: FPSLOS_calc, AV_2D
*      ------------------------------------------------------------------------------------

       PROGRAM BANDY              !Main program.
       implicit none
       character(len=1) :: slix='X',slix2,bin_type,synch,find_binlength
       character(len=4) :: dda,dtype
```

```
 character(len=9) :: userid
 character(len=8) :: directory='results/' !File directory.
 character(len=80) :: tempname
 character(len=200) :: string
 integer :: shotn,ierr,iseq=0,i,j,k,ncl,collflag=0,nclprev=0
.      ,tpos=-1
 real :: histsum_1d,startt,stopt,t_lastbin,starttime,binlength
.      ,yapan_chi2=-1.0,find_chi2
 real(8) :: FnMPR,p_counts,chi2,FnFPSLOS,YnTINRNT,YnFPSLOS,RnKN3
.      ,YnMPR,RnFPSLOS
 CALL PPFGO(0,0,0,ierr)     !Initializing PPF system for reading PPF data.
 DO                         !Input file must exist.
    write(*,*)'Enter file to read from: '
    read(*,*)tempname
    tempname=directory//tempname
    open (unit=1,file=tempname,status='old',iostat=ierr)
    IF (ierr.eq.0) EXIT
    write(*,*)'The file does not exist.'
 END DO
 DO                         !Timebins synched with PPF:s?
    write(*,'(a19)',advance='no')'Binsynched ? y/n : '
    read(*,'(a)')synch
    IF ((synch.eq.'y').or.(synch.eq.'n')) EXIT
 END DO
 DO                         !Create outpufile.
    write(*,*)'Enter file to write to: '
    read(*,*)tempname
```

```fortran
         tempname=directory//tempname
         open (unit=2,file=tempname,status='unknown',iostat=ierr)
         IF (ierr.eq.0) EXIT
      END DO                    !Create file to store errors, especially reading PPFs.
      open (unit=3,file='results/error_file.txt',status='replace'
     .     ,iostat=ierr)
      string='shotn startt stopt Fn(FPSLOS) Fn(MPR)  Fn(MPR)/Fn(FPSLOS)
     .     Yn(TIN/RNT) Yn(FPSLOS) Yn(MPR) Yn(MPR)/Yn(TIN/RNT) MPRchi2
     .     p_counts binlength KN3chi2'
      write(2,200) string
200   format(A)
      DO i=1,10                 !Skip 10 lines of input file.
         READ (1,*,IOSTAT=ierr)
         IF (ierr.ne.0) THEN    !If error reading, end program.
            write(*,*)'Error reading header of inputfile.',ierr
            GOTO 999
         END IF
      END DO
      DO i=1,1000000            !Repeated reading from input file.
         READ(1,*,IOSTAT=ierr)(shotn,startt,stopt,NCL,FnMPR,chi2
     .        ,p_counts)
         IF (ierr.gt.0) THEN    !If reading error, end program.
            write (3,*)'Error reading from inputfile.',i
     .           ,shotn,startt,stopt,ierr
            GOTO 999
         ELSEIF (ierr.eq.-1) THEN !If end of input file, end program.
            write (*,*)'End of file EXCEL.'
```

```
      GOTO 999
   END IF
   IF (synch.eq.'y') THEN !Determines binlength.
      slix2=bin_type(startt,stopt)
      tpos=1              !Flag for binsynch, 1=true.
   ELSE                   !Not binsynched.
      IF ((stopt-startt).ge.10.0) THEN !Determine appropriate binning.
         slix2='I'
      ELSE
         slix2=find_binlength(shotn,startt,stopt,ierr)
      END IF
      IF (ierr.ne.0) GOTO 990
      IF (slix2.eq.'L') write(3,*)'No 50ms data, using 500ms, '
.           ,shotn,startt,stopt
   END IF
   write(*,*)'Analyzing pulse ',shotn,startt,stopt !On-screen message.
   IF (NCL.eq.NCLprev) THEN
      CollFlag=1          !Collimator flag. 1 if LOS changed.
   END IF
   NCLprev=NCL
   CALL FPSLOS_calc(shotn,startt,stopt,YnFPSLOS,FnFPSLOS,slix2
.        ,collflag,tpos,ierr) !Calculate FnFPSLOS and YnFPSLOS.
   IF (ierr.ne.0) GOTO 990 !Skip and continue with next input line.
   dda='TIN '
   dtype='RNT '
   userid='JETPPF'
   tpos=-1                !Don't use binsynch in next routine.
```

```
         YnTINRNT=histsum_1d(shotn,startt,stopt,dda,dtype,userid,tpos
     .        ,ierr)
         IF (slix2.ne.'I') YnTINRNT=YnTINRNT/(stopt-startt) !Get YnTINRNT from JETPPF.
         IF (ierr.ne.0) GOTO 990 !If read error, skip and continue.
         RnKN3=YnFPSLOS/FnFPSLOS !Factor relating FnMPR to YnMPR.
         IF (slix2.ne.'I') FnMPR=FnMPR/(stopt-startt)
         YnMPR=RnKN3*FnMPR      !This is YnMPR.
         yapan_chi2=find_chi2(slix2,startt,stopt,shotn) !Get YAPAN chi2 parameter.
         write(2,210)shotn,startt,stopt,FnFPSLOS,FnMPR,FnFPSLOS/FnMPR
     .        ,YnTINRNT,YnFPSLOS,YnMPR,YnMPR/YnTINRNT
     .        ,' ',chi2,p_counts,slix2,yapan_chi2 !Write to file.
210      format(i8,2f7.3,2es15.7,f8.3,3es16.7,f8.3,a1,2es15.7,a4,f8.3)
         call flush(2)          !Secure results in case of program crash.
990      CONTINUE               !Minor error. Read next line of input.
      END DO
999   CONTINUE                  !End of input file or major error. Finish program.
      rewind(3,iostat=ierr)     !Attempt to continue on old error file.
      read(3,*,iostat=ierr)
      inquire(3,name=tempname)
      IF (ierr.eq.0) THEN       !Should notify user only if errors appeared.
         write(*,('a26,a40'))'Error(s) written to file: ',tempname
      ELSE
         write(*,*)'Normal termination of nk1.'
      END IF
      close( 1 )                !Closing files.
      close( 2 )
      close( 3,iostat=ierr )
```

```
      IF (ierr.ne.0) write(*,*)'Could not close error_file.txt ',ierr
      END PROGRAM


*     Gets YAPAN chi2 parameter from PPF.
*     -------------------------------------------------------------------------------------
      REAL FUNCTION find_chi2(slix2,startt,stopt,shotn)
      implicit none
      character(len=1) :: rw='R',slix2
      character(len=4) :: dda='KN31',dtype
      character(len=9) :: userid='hsjost'
      integer :: shotn,ierr,iseq=0,nx,nt,logic=1,i
      integer,parameter :: ntmax=20000,nxmax=20,ndmax=nxmax*ntmax
      real :: d(ndmax),x(nxmax),t(ntmax),startt,stopt,loc_stopt
      find_chi2=-1.0
      dtype='CHI'//slix2
      nx=nxmax                  !Specifies maximum of x-array. Neccessary.
      nt=ntmax                  !Specifies maximum of t-array. Neccessary.
      CALL PPFUID(userid,rw)    !Initializing PPF system.
      CALL PPFDAT(shotn,iseq,dda,dtype,nx,nt,d,x,t,ierr) !Get chi2 parameter.
      IF (ierr.ne.0) THEN       !Write to file, continue with next input line.
         write(3,*)'Error calling PPFDAT in find_chi2',shotn
     .        ,dda,dtype,ierr
         GOTO 999
      END IF
      loc_stopt=(stopt+startt)/2.0 !Time in middle of bin.
      DO i=1,nt-1               !Find index of nearest timebin.
```

```
         IF (t(i)>loc_stopt) EXIT
      END DO
      find_chi2=d(i)            !Get chi2 for indexed bin.
 999  CONTINUE
      END FUNCTION find_chi2


*     Store reference character for binlength.
*     ------------------------------------------------------------------
      CHARACTER(len=1) FUNCTION bin_type(startt,stopt)
      implicit none
      real :: startt,stopt
      IF (stopt-startt.lt.0.06) THEN
         bin_type='S'           !Short bin, 50ms.
      ELSE IF (stopt-startt.lt.0.6) THEN
         bin_type='L'           !Long bin, 500ms.
      ELSE
         bin_type='I'           !Integrated, 20s.
      END IF
      END FUNCTION bin_type


*     Determine binlength based on input data.
*     ------------------------------------------------------------------
      character(len=1) FUNCTION find_binlength(shotn,startt,stopt,ierr)
      character(len=1) :: rw='R'
      character(len=4) :: dda='KN31',dtype='PEMS'
```

```
character(len=9) :: userid='hsjost'
integer :: shotn,ierr,iseq=0,nx,nt,logic=1
integer,parameter :: ntmax=20000,nxmax=20,ndmax=nxmax*ntmax
real :: d(ndmax),x(nxmax),t(ntmax),startt,stopt
find_binlength='S'
CALL PPFUID(userid,rw)    !Initialize PPF system.
DO i=1,2
   nx=nxmax               !Specifies maximum of x-array. Neccessary.
   nt=ntmax               !Specifies maximum of t-array. Neccessary.
   CALL PPFDAT(shotn,iseq,dda,dtype,nx,nt,d,x,t,ierr) !Retrieve data.
   IF (ierr.ne.0) THEN
      write(3,*)'Error calling PPFDAT in find_binlength',shotn
.          ,dda,dtype,ierr !Write error to file.
      GOTO 999            !If PPF read error, skip and continue with next bin.
   END IF
   DO j=2,nt              !Repeat for 50ms and 500ms.
      IF ((stopt-t(j)).lt.(t(j)-t(j-1))) THEN
         IF (d(j).ne.0.0) logic=0
      END IF
   END DO
   IF (logic.ne.0) THEN
      IF (i==1) THEN
         dtype='PEML'
         find_binlength='L'
      END IF
      IF (i==2) find_binlength='I'
   END IF
```

```
      END DO
999   CONTINUE
      END FUNCTION find_binlength


*     Calculates FnFPSLOS and YnFPSLOS.
*     ----------------------------------------------------------------------------------
      SUBROUTINE FPSLOS_calc(shotn,startt,stopt,YnFPSLOS,FnFPSLOS,slix2
     .     ,collflag,tpos,ierr)
      implicit none
      character(len=1) :: slix2
      character(len=4) :: dda,dtype
      character(len=9) :: userid='hsjost'
      integer :: shotn,los_no,i,j,los_valid=-1,collflag,ierr
      integer,parameter :: ntmax=20000,nxmax=20,ndmax=nxmax*ntmax
      real :: losdata(0:600000,8),r,z,scs0,scsa,sca,sce,delr,delz
     .     ,scalp,scr0,scz0,startt,stopt,scdu,test,tpos
     .     ,scra,scza,scroff,scrstep,sczstep,sczoff,p(nxmax)
     .     ,em_profile_fcn,scs0_2,simple_sum_div
      real(8) :: FnFPSLOS,YnFPSLOS,histsum_1d
      common /Drf/ losdata,los_no !Data for line of sight as common block.
      FnFPSLOS=0.0
      YnFPSLOS=0.0
      dda='KN31'
      dtype='PEM'//slix2
*     Central neutron emissivity extracted.
      scs0=histsum_1d(shotn,startt,stopt,dda,dtype,userid,tpos,ierr)
```

```
*      write(*,*)scs0
      IF (ierr.ne.0) GOTO 999   !If PPF error, skip and continue with next bin.
      dtype='PAR'//slix2
      CALL AV_2D(shotn,startt,stopt,dda,dtype,p,ierr) !Profile parameters extracted.
      IF (ierr.ne.0) GOTO 999   !If PPF error, skip and continue with next bin.
      sce=p(2)                  !Ellipsivity.
      scr0=p(4)                 !Major radius of innermost Iso-Emissivity-Contour (IEC).
      scra=p(5)                 !Major radius of outermost IEC.
      scz0=p(6)                 !Distance above mid-plane of innermost IEC.
      scza=p(7)                 !Distance above mid-plane of outermost IEC.
      scalp=p(9)*10.0           !Peaking factor.
      scsa=p(10)*0.01*scs0      !Edge neutron emissivity.
*      write(*,*)'scsa,p(10)',p(10),scsa
      sca=p(18)                 !Minor radius of Nested Emission Profile (NEP).
      delr = SCR0-SCRA          !Difference between major radii.
      delz = SCZ0-SCZA          !Vertical distance differnce.
      SCROFF = SCRA-SCA-0.1     !Width of calculation box
      SCRSTEP = (2.0*SCA+0.2)/200.0 !Horizontal stepsize.
      SCZOFF = SCZA-SCE*SCA-0.1 !Height of calculation box.
      SCZSTEP = (0.2+2.0*SCE*SCA)/200.0 !Vertical stepsize.
*     Calculation of YnFPSLOS.
      DO i=0,200                !Summation of neutron emission from radial slice
         scdu = 0.0             ! of torus.
         DO j = 0,200
            r = SCROFF + i*SCRSTEP
            z = SCZOFF + j*SCZSTEP
            scdu=scdu+em_profile_fcn(r,z,scr0,scz0,delr,delz,sca,sce
```

```
     .           ,scs0,scsa,scalp,los_valid)
         END DO
         YnFPSLOS=YnFPSLOS+scdu*2.0*3.1415926535898*SCRSTEP*SCZSTEP*
     .        (SCROFF+i*SCRSTEP) !Integrating neutron emission 360deg around torus.
      END DO
      IF (collflag.eq.0) THEN   !Read LOS-file if it hasn't been done.
         OPEN(13,FILE='/home/sconroy/codes/fpslos/data/KM9.Drf')
         los_no=-1
 103     los_no=los_no+1
         READ(13,*,ERR=2902,END=2902)(LOSDATA(LOS_NO,j),j=1,8)
         IF(LOSDATA(LOS_NO,4).eq.0.0) LOS_NO = LOS_NO - 1
         IF (LOS_NO.le.600000) GOTO 103
 2902    IF (LOS_NO.eq.0) WRITE(0,*)' No points read in. Bad file?'
         IF (LOS_NO.eq.0) STOP
         CLOSE(13)
         collflag=1             !Line of sight has been changed.
      END IF
      los_valid=-1              !-1 if no valid points in LOS.
*     Calculation of FnFPSLOS.
      DO i=0,LOS_NO - 1
         r=SQRT(losdata(i,1)**2+losdata(i,2)**2)
         z=losdata(i,3)
         FnFPSLOS=FnFPSLOS+em_profile_fcn(r,z,scr0,scz0,delr,delz,sca
     .        ,sce,scs0,scsa,scalp,los_valid)*losdata(i,4)
      END DO
      write(*,*)'Number of valid points in LOS = ',los_valid
*      write(*,*)'FnFPSLOS,YnFPSLOS',FnFPSLOS,YnFPSLOS
```

```
 999  CONTINUE
      END SUBROUTINE FPSLOS_calc


*     Extracting NEP parameters from PPF.
*     ------------------------------------------------------------------------------------
      SUBROUTINE AV_2D(shotn,startt,stopt,dda,dtype,p,ierr)
      implicit none
      character(len=1) :: rw='R'
      character(len=4) :: dda,dtype
      character(len=9) :: userid='hsjost'
      integer :: iseq=0,ierr,nx,nt,shotn,i,j,istart,istop,tempstop
      integer,parameter :: ntmax=20000,nxmax=20,ndmax=nxmax*ntmax
     .     ,npmax=20
      real :: d(ndmax),x(nxmax),t(ntmax),startt,stopt,p(npmax),n
     .     ,loc_stopt
      n=0.0
      nx=nxmax                  !Specifies maximum of x-array. Neccessary.
      nt=ntmax                  !Specifies maximum of t-array. Neccessary.
      istart=nt
      istop=1
      DO i=1,npmax              !Initializing temporary p-array.
         p(i)=0.0
      END DO
      CALL PPFUID(userid,rw)    !Initializing PPF system.
      CALL PPFDAT(shotn,iseq,dda,dtype,nx,nt,d,x,t,ierr) !Retrieve data.
      IF (ierr.ne.0) THEN
```

```
         write(3,*)'Error calling PPFDAT in AV_2D',shotn,dda,dtype,ierr
         GOTO 999               !If PPF read error, skip and continue with next bin.
      END IF
      IF (dtype.eq.'PARI') THEN !Parameters for integrated pulse.
         DO i=1,npmax
            p(i)=d(i)
         END DO
      ELSE                      !Parameters for binned data.
         loc_stopt=(stopt+startt)/2.0 !Time in middle of bin.
         DO i=1,nt-1            !Find index of nearest timebin.
            IF (t(i)>loc_stopt) EXIT
         END DO
         istop=i                !!!NOT COMPLETE, only for binsynched!!!
         istart=i
         DO i=istart,istop      !Include bins inside time interval.
            n=n+1.0             !Remember number of intervals summed over.
            DO j=1,npmax
               p(j)=p(j)+d(i*20-20+j) !Each parameter summed separately.
            END DO
         END DO
         DO j=1,npmax
            p(j)=p(j)/n         !Divide with number of intervals.
*            write(*,*)j,p(j)
         END DO
      END IF
*      write(*,*)startt,stopt,loc_stopt,n
 999  CONTINUE
```

```
      END SUBROUTINE AV_2D


*     Solving emissivity profile function, quasi-parabolic + step.
*     ------------------------------------------------------------------------------------
      REAL FUNCTION em_profile_fcn(r,z,scr0,scz0,delr,delz,sca,sce,scs0
     .      ,scsa,scalp,los_val)
      implicit none
      integer :: los_val
      real :: r,z,scr0,scz0,delr,delz,sce,sca,scs0,scsa,scalp,rho,test
      real :: c1,c2,sol1,sol2,qa,qb,qc
      em_profile_fcn=0.0
      c1=r-scr0
      c2=z-scz0
*     Variables for solving second degree equation.
      qa=delr**2+(delz/SCE)**2-sca**2
      qb=2.0*(c1*delr+c2*delz/sce/sce)
      qc = c1**2+(c2/sce)**2
      test = qb*qb-4.0*qa*qc    !Square root expression.
*     Allow only real valued solutions.
      IF((test.ge.0).and.(qa.ne.0.0)) THEN
         test = SQRT(test)
         sol1=(-qb+test)*0.5/qa
         sol2 = (-qb-test)*0.5/qa
         rho = MAX(sol1,sol2,0.0)
         IF (rho.lt.1.0) THEN
            em_profile_fcn=((SCS0-SCSA)*(1-rho*rho)**scalp+scsa)
```

```
            los_val=los_val+1
         END IF
      END IF
      END FUNCTION em_profile_fcn


*     Summation of bins inside time interval.
*     -------------------------------------------------------------------------------------
      REAL FUNCTION histsum_1d(shotn,startt,stopt,dda,dtype,userid,tpos
     .      ,ierr)
      character(len=1) :: rw='R'
      character(len=4) :: dda,dtype
      character(len=9) :: userid
      integer,parameter :: ntmax=20000,nxmax=20,ndmax=nxmax*ntmax
      integer :: nx,nt,shotn,ierr,iseq=0,i,istart,istop,tpos
      real :: startt,stopt,d(ndmax),x(nxmax),t(ntmax),loc_stopt
      histsum_1d=0.0
      nx=nxmax                   !Specifies maximum of x-array. Neccessary.
      nt=ntmax                   !Specifies maximum of t-array. Neccessary.
      CALL PPFUID(userid,rw)     !Initializing PPF system.
      CALL PPFDAT(shotn,iseq,dda,dtype,nx,nt,d,x,t,ierr) !Retrieve data.
      IF (ierr.ne.0) THEN
         WRITE (3,*)'Error calling PPFDAT in histsum_1d',shotn,dda,dtype
     .         ,ierr
         GOTO 999                !If PPF read error, skip and continue with next bin.
      END IF
      istart=nt
```

```
      istop=1
*     Integrated, same for binsynched or not.
      IF (dtype.eq.'PEMI') THEN
         histsum_1d=d(1)
      ELSE
*     Not binsynched.
         IF (tpos.eq.-1) THEN
            DO i=1,nt-1          !Find end bins for summation.
               IF (t(i).gt.startt) istart=min(istart,i)
               IF (t(i).lt.stopt) istop=max(istop,i+1)
            END DO
            DO i=istart,istop    !Summation of bins inside interval.
               histsum_1d=histsum_1d+d(i)*(t(i)-t(i-1))
            END DO               !Subtraction of superfluous parts of end bins.
            histsum_1d=histsum_1d-(d(istart)*abs(startt-t(istart-1)))
            histsum_1d=histsum_1d-(d(istop)*abs(t(istop)-stopt))
*     Binsynched.
         ELSE
            loc_stopt=(stopt+startt)/2.0 !Time in middle of bin.
            DO i=1,nt-1          !Find index of nearest timebin.
               IF (t(i)>loc_stopt) EXIT
            END DO
            histsum_1d=d(i)
*     *(t(i)-t(i-1)) !Summation of bins.
         END IF
      END IF
 999  CONTINUE
```

```
END FUNCTION histsum_1d
```

Appendix F

MAW command ANYD

```
subroutine execute_NK_menu()
implicit none
character fname*20,writefile*80,extr_chopt*32
integer*4 nch,anal_type
real*4 tmin,tmax
include 'MT-common.for'
include 'MPR-common.for'
include 'SMP-common.for'
include 'WORK-common.for'
include 'MUIP-externals.for'
if( MUIP$command('ANALYZE','Analyzing neutron yield data') ) then
call MUIP$getc('FNAME','*',fname,nch,'pulse number')
call MUIP$getr('TMIN',0.0,tmin,'start time (default 0.0)')
```

```
call MUIP$getr('TMAX',0.0,tmax,'stopt time (default 0.0)')
! call MUIP$geti('NREBIN2',1,nrebin2,'Time rebinning parameter. Default=1.')
call MUIP$getc('WRITEFILE','*',writefile,nch,'File to write to')
call MUIP$optional()
call MUIP$getc('CHOPT','*',extr_chopt,nch,'Options')
        call MUIP$help("Possible CHOPT values are:")
call MUIP$help(" ")
call MUIP$help("    S Impose time syncronization.")
call MUIP$help("    B Impose background correction.")
call MUIP$help("    I Use interpolation at ends of timeinterval.")
call MUIP$help("    V Verbose mode.")
call MUIP$geti('ANAL_TYPE',1,anal_type,'Excel analyze option. Default=1.')
! call MUIP$help("    I   Debug option")
if ( MUIP$do_command() ) then
call NK_analyze(fname,tmin,tmax,writefile,extr_chopt,anal_type)
end if
end if
end
!---------------------------------------------------------------------------
subroutine NK_analyze(fname,tmin,tmax,writefile,extr_chopt,anal_type)
implicit none
character fname*20,writefile*80,charline1*80,charline2*80,charline3*80
character build_chopt*16,extr_chopt*32
integer*4 ierr,dc,anal_type
real*4 tmin,tmax,sync
include 'SMP-common.for'
include 'SCALER-common.for'
```

```
sync=0.0
build_chopt=' n1'
open (unit=1,file=writefile,status='old',position='append',iostat=ierr)
if ( ierr .ne. 0 ) open (unit=1,file=writefile,status='new',iostat=ierr)
call get_KM9_pdb_record
call muip$exec( 'reset' )
! call muip$exec( 'opt liny' )
call muip$exec( 'scaler/mode n' )   ! Work with counts/bin, in comb with EXTRACT below
write(charline1,*) 'mpr/read ',fname,' ! ! lz '
write(*,*)charline1
call muip$exec( charline1 )         ! Read sample file, list processor data, no histograms
if ( index(extr_chopt,'S') .ne. 0 ) build_chopt=build_chopt(1:index(build_chopt,'  ')-1)//'s'
if ( index(extr_chopt,'B') .ne. 0 ) build_chopt=build_chopt(1:index(build_chopt,'  ')-1)//'b2'
if ( index(extr_chopt,'I') .ne. 0 ) build_chopt=build_chopt(1:index(build_chopt,'  ')-1)//'i'
if ( index(extr_chopt,'V') .ne. 0 ) build_chopt=build_chopt(1:index(build_chopt,'  ')-1)//'v'
write(charline2,*) 'extract ',-1,build_chopt,tmin,tmax
write(*,*)charline2
call muip$exec( charline2 )
if ( extract_ok ) then
write(charline3,*) 'sc/fill h2x'
write(*,*)charline3
call muip$exec( charline3 )     ! Single histo (ID=1000) created, with extracted position histogram
write(1,100)file_number,extract_tmin,extract_tmax
     / ,abs(extract_tmax-scl_dt(1)-tmin),abs(tmax-extract_tmax),sync
     / ,build_chopt,anal_type,pdb_itarget,pdb_ipcoll,pdb_incoll
     / ,pdb_Bfield_percent,(scl_nbkgcorr(1,dc)/eps_used(dc),dc=1,36)
     / ,(scl_nbkgcorr_err(1,dc)/eps_used(dc),dc=1,36)
```

```
100 format (i8,5f10.4,a8,4i10,f10.4,72f10.2)
else
write(*,*)'No extracted data'
end if
200 format (a32)
close( unit=1 )
write(*,*)'tmin,tmax',tmin,tmax,scl_dt(1),extract_tmin,extract_tmax
end
```