ISDAT *igr*
General Purpose Data Manipulation and Display Client
User's Manual

Bjørn Lybekk, University of Oslo
Michael Thomsen, ESTEC
Anders Lundgren, IRF-U
Editor: Gunnar Holmgren, IRF-U

with change bars for version 1.3

Contents

1 Introduction **1**

1.1 Background 1

1.2 Intended readership 1

1.3 Applicability of the manual 1

1.4 Purpose of the software 1

1.5 How to use this document 1

1.6 Related documents 2

1.7 Conventions and acronyms 2

2 Overview **3**

2.1 igr capabilities 3

2.2 Underlying structure and concepts 3

2.2.1 Panels 3

2.2.2 Quantities 4

2.2.3 Plots 4

3 User's Instructions **5**

3.1 Help functions 5

3.2 Starting igr 5

3.3 Main window layout 5

3.4 Exiting from igr 6

3.5 Specifying the time interval 6

3.6 Specifying input to igr 7

3.6.1 Setting current plot and panel 7

3.6.2 Setting quantities to plot 8

3.6.3 Specifying units of the data request 8

3.6.4 Limiting the number of returned samples 10

3.6.5 Specifying the data packing 10

3.7 Simple Plotting 10

3.7.1 Plotting scalars 11

3.7.2 Plotting complex quantities 11
3.7.3 Plotting vectors 11
3.7.4 Plotting tensors 11
3.7.5 Plotting spectra 11
3.8 Advanced Plotting 11
3.8.1 Adjusting scales 11
3.8.2 Setting individual or common scale 13
3.8.3 Setting plot type 14
3.8.4 Setting axis type 14
3.8.5 Setting frame and grids for the panel 14
3.8.6 Setting colours 16
3.8.7 Plotting several parameters in one panel 16
3.8.8 Plotting static spectra 18
3.8.9 Plotting dynamic spectra 18
3.8.10 Adding quantities 20
3.8.11 Deleting quantities 20
3.8.12 Adding plots 21
3.8.13 Deleting plots 21
3.8.14 Adding panels 22
3.8.15 Deleting panels 22
3.8.16 Setting size of a panel 22
3.8.17 Expanding time axis 23
3.8.18 Plotting one parameter versus another parameter 23
3.8.19 Plotting contiguous time intervals in adjacent panels 24
3.8.20 Running in continuous mode 26
3.8.21 Expanding and compressing header text 27
3.9 Manipulating Data 28
3.9.1 Overview 28
3.9.2 Use of the calculator 32
3.9.3 Introducing time shifts between quantities 34
3.9.4 Joining data sets to a common time line 36

3.9.5 Transforming to frequency domain 36
3.9.6 Performing coordinate transformations 38
3.10 Exporting and importing 39
3.10.1 Saving data in a flat File 39
3.10.2 Saving data as a CDF File 41
3.10.3 Printing the graph 42
3.10.4 Saving an igr layout 44
3.10.5 Restoring an igr layout 45
3.10.6 Predefined plot layout 45
3.11 Running igr in batch mode 46
3.11.1 Batch file syntax 46
3.11.2 Useful hints about igr batch processing 48
3.11.3 Igr batch file example 48

4 Errors and probable causes **50**
4.1 Problem reporting 50
4.2 Error messages 50
4.3 Common user errors and probable causes 51

5 Customizing igr **51**

6 Using *igr* for de-bugging **54**

7 Reference Documents **55**

8 Index **56**

1 Introduction

1.1 Background

The *igr* (ISDAT GRaphics) is an ISDAT client of class *general clients* (see [Ref. 6]). It is primarily developed for the WEC detailed analysis software package, but a special version *cuigr* (see [Ref. 7]) is also available within the CSDS User Interface. The main development has been made by Bjørn Lybekk, University of Oslo. Contributions to the data manipulations parts have been made by Michael Thomsen, ESTEC, Joe Zender, ESTEC, and Per Ola Dovner, IRF-U. Anders Lundgren, IRF-U has had the responsibility for the software quality assurance and conformance to the ISDAT conventions.

1.2 Intended readership

This manual is intended for the scientific user of the ISDAT *igr* client within the WEC ISDAT Client package and within other projects.

1.3 Applicability of the manual

This manual is based on the *cuigr* manual [Ref. 7]. The current version of the document applies to the ISDAT version 2.4. It is valid for UNIX, SUN Solaris workstations.

1.4 Purpose of the software

The purpose of the WEC/ISDAT software package, of which *igr* is one component, is to provide the WEC scientific community with software tools to manipulate and display Cluster WEC parameters. The *igr* client is the general purpose data manipulation and display client provided within the package.

1.5 How to use this document

This document consists of an overview of the software in order to familiarize the user with the capabilities provided and introduce the concepts and terminology used. The User Instructions section (3) should be read in connection with the first hands-on encounter with the *igr*. An index section is included for the user's reference.

For the use of this document it is assumed that:

- You are familiar with the X-window and mouse usage. If not, consult the appropriate manuals for your work station.
- You have logged in and have access to an ISDAT server. If not,see [Ref. 4] for instructions.

- The ISDAT client package has been properly installed and configured at your local workstation. If not, see [Ref. 5] for instructions.
- A *time manager* presumably *ctm* [Ref. 8] is running and the user is familiar with its use.

The *igr* is designed to provide an extremely flexible tool for composing a data display as well as manipulating and combining data from different instruments. Therefore, by necessity, the underlying structure is complex, and the concepts, as described in section 2.2 may seem hard to understand. However, a user who intends to use *igr* only for simple plots, do not have to bother about the underlying structure and may jump to section 3 to get started.

1.6 Related documents

Valuable information can be obtained from the CSDS User Interface software manuals. An overview of the CSDS UI ISDAT Client Package is given in [Ref. 6]. The installation of the ISDAT client package is described in [Ref. 5]. The CSDS User Interface is described in [Ref. 3]. The time manager *ctm* is described in [Ref. 8]. A general introduction to the WEC software tools is given in the WWW on URL
http://www.irfu.se/isdat/wec/intro/wec_intro.html.

1.7 Conventions and acronyms

In the following, we will use:

- *italics* to indicate exact names or expressions.
- Courier fonts to give command line expressions or file excerpts.

Acronyms and abbrieviations used are described in Table 1.

Acronym	Meaning
ASCII	American Standard Code for Information Interchange
CDF	Common Data Format
CSDS	Cluster Science data System
CUI	CSDS User Interface
IRF-U	Institutet för Rymdfysik, Uppsalaavdelningen
	Swedish Inst. of Space Phys., Uppsala Division
ISDAT	Interactive Science Data Analysis Tool
UI	User Interface
URL	Universal Resource Locator
WWW	World Wide Web

Table 1: Acronyms and abbrieviations

2 Overview

2.1 igr capabilities

The *igr* graphic client has the following capabilities:

- display up to 30 panels with up to 4 parameters in each panel
- plot line graphic (lines, scatter and hodograms)
- plot colour coded dynamic spectra and superimposed line graphs.
- manipulate input data (arithmetics, FFT, coord. transf. etc)
- customize, save and retrieve plot layout (panels, colors, axis)
- plot parameter versus time
- plot parameter 1 versus parameter 2
- interactively expand time axis (zoom time scale)
- produce hard copies (black & white and colors)
- save data as flat files (ascii) or CDF files
- run in batch mode

2.2 Underlying structure and concepts

In the following we will use the concepts:

- panels
- quantities
- plots

They are defined and described in the sections 2.2.1, 2.2.2 and 2.2.3.

2.2.1 Panels

By *panels* we understand plot areas that may hold zero to four *plots*. The *igr* client can display one or more panels. When the program starts panel 0 is defined.

The menubar *Graphics→Panel→Create* buttons are used to create new panels, as described in 3.8.14.

The numbering of the panels is defined in the panels matrix, shown in Table 2:

The borders between the panels can be changed as described in section 3.8.16.

Panel 0	Panel 10	Panel 20
Panel 1	Panel 11	Panel 21
Panel 2	Panel 12	Panel 22
Panel 3	Panel 13	Panel 23
Panel 4	Panel 14	Panel 24
Panel 5	Panel 15	Panel 25
Panel 6	Panel 16	Panel 26
Panel 7	Panel 17	Panel 27
Panel 8	Panel 18	Panel 28
Panel 9	Panel 19	Panel 29

Table 2: Panel matrix

The *current panel* is the panel containing the *current plot*, (see section 2.2.3). When the user selects the current plot (see below section 2.2.3) he automatically selects the *current panel.*

2.2.2 Quantities

The data structure containing the input data from the ISDAT server is named *quantity.* Each quantity is a data element q[]. One quantity is selected (see section 3.6.2), this is named the *current quantity.* The user specifies what data is read into the *current quantity* (see section 3.8.10). Information about the *current quantity* is always shown in the *current quantity status line* (see section 3.3) and in the *Quantity dialog* window (Figure 8).

When a request about new data from the data base handler appears the *igr* client read data into all *defined quantities* (see section 3.6.2).

The data are then passed trough the *Calculator* function (see section 3.9.2). In the *Calculator* function the data are transferred from the quantity structure into the plot structure.

2.2.3 Plots

The *plot* structure contains the data plotted in the panels. A plot may correspond to a *quantity.* However a *plot* may also require input from several quantities, for example a sum of two *quantities.*

Each of the plots are associated with a pushbutton in the right upper corner of the panel. When pressing one of the pushbuttons this plot is selected as *current plot.* At the same time the information about this plot is written in the *current plot status line.*

When the *igr* starts the default links between the quantities and plots are:

```
q[k] -> plot[k]
```

The data from quantity number k is automatically passed to plot number k. The reason for this is that this client shall also be useful for an unexperienced user.

3 User's Instructions

3.1 Help functions

All major windows have on line help functions. They are accessed via a button in the upper or lower right hand corner of the window.

In addition, there is an on-line *man* page provided that your *igr* client has been properly installed. The on-line man-page adhere to the UNIX standards both with respect to access and text format. The man-page is accessed from a terminal window by the command:

```
>man cuigr
```

The man page is written for the special *CSDS User Interface* version of *igr* and therefore do not cover all the *igr* features. Numerous other ISDAT clients, libraries and functions are described by man-pages.

3.2 Starting igr

An ISDAT server must be running and the *time manager (cuitm)*, (see [Ref. 8]), must have been started at your local workstation. The *igr* client is started from the *time manager*. The user presses the *clients*, *general* and *igr* buttons. Now a window similar to that in Figure 1 will appear.

3.3 Main window layout

When the *igr* starts it has the layout shown in Figure 1. The layout consists of four fields, which are described below. The *menubar* is located at the top of the window. Below this menubar are two text lines, the *current quantity status line* and the *current plot status line*. When the program starts it contains one *graphic panel* (Panel 0).

menubar - program control The menubar consists of the *File, Edit, Inst, Param, Select, Graphics* and *Help* buttons. Each of these buttons is associated with a pulldown menu. These pulldown menues are used for program control. Some pulldown menues can be "teared off" for permanent access. The tear-off menues are recognized by a dashed line at the top of the menu.

current quantity status line In this text field is shown information about *current quantity*. The quantities and the *current quantity* are further explained in section 2.2 page 3.

current plot status line In this text field is shown information about *current plot*. The plots and the current plots are further explained in section 2.2 page 3.

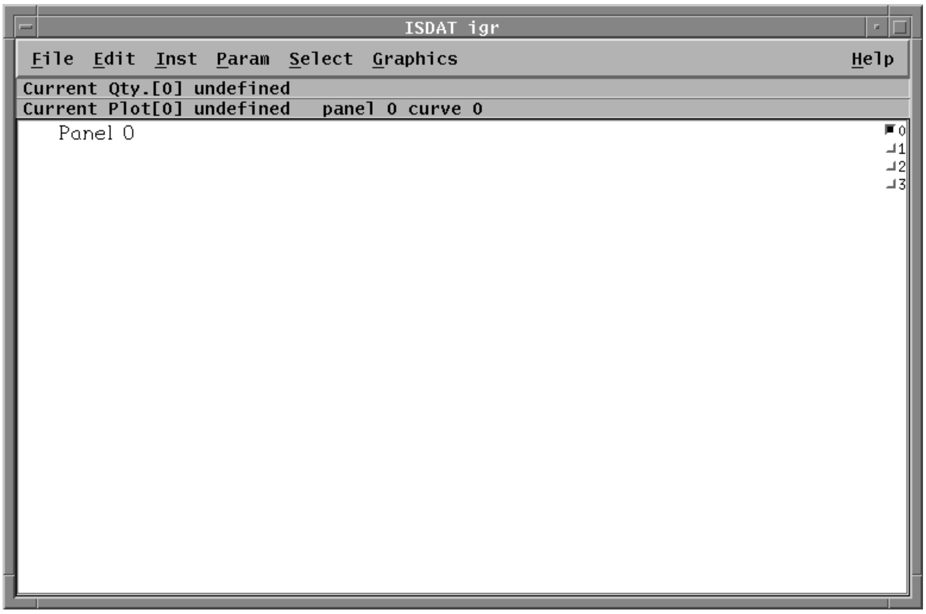

Figure 1: The igr client window

graphic panel The panels are graphic widgets. The panels are used to display graphic data. Panel 0 is the only panel defined when the program starts. Up to 30 panels can be displayed in variable configurations as explained in section 2.2 page 3.

3.4 Exiting from igr

The *igr* client is stopped by pressing *File* in the main menubar and *Exit*. The *igr* will terminate and the window disappear. *Igr* will also terminate when the associated *time manager* is terminating.

3.5 Specifying the time interval

The time interval to analyze or display is set in the ISDAT *time manager ctm* . The times are then, at *update* communicated to all active ISDAT clients including *igr* for each client to take appropriate actions. For the *igr* it means that it normally takes note of the new time interval and then requests data from the ISDAT server for the time interval and for the parameters required. The use of the *ctm* is described in [Ref. 8].

Normally **only** the time manager, notably *ctm* , is allowed to change the time interval, However, *igr* has the special privilege to be allowed to modify the time interval as described in section 3.8.17, page 23.

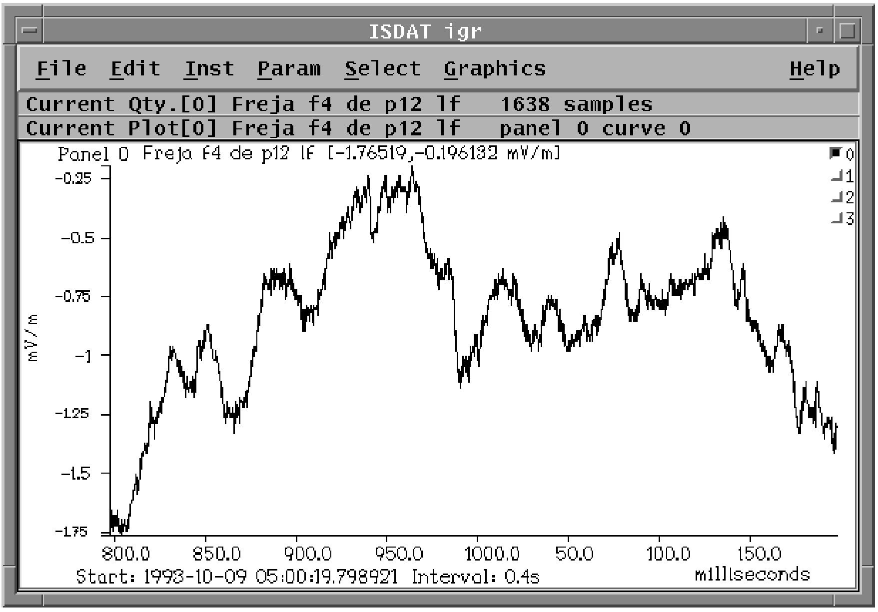

Figure 2: Igr window with one plot panel

3.6 Specifying input to igr

Assuming that the client has been started as explained in 3.2, the user presses the *Param* button and then selects one of the parameters in the pulldown menu. Now he presses the right mouse button when the cursor is located inside the graphic Panel 0. Alternatively, the *update* button of the time manager *ctm* (see [Ref. 8]) is pressed. *igr* follows the time interval specified in *ctm* . The data should now be displayed in the panel similar to that shown in Figure 2. Pressing the right button updates the panel.

When pressing the *forward arrow* button in the *ctm* the subsequent data interval is displayed (see [Ref. 8]).

Igr will automatically be informed about the type of variable returned from the ISDAT server. Depending on the type of variable, *igr* will act in different ways (see the following sections).

3.6.1 Setting current plot and panel

In *igr* all specifications and settings operate on the *current plot, current quantity or current panel* depending on the nature of the setting.

The panel number is printed in the left upper corner of each panel, (e.g. Panel 0). Each panel can contain maximum four plots. In the right upper corner of the panel are four small pushbuttons . Each pushbutton is associated with one plot. The plot numbers are written at the right side of the small pushbutton.

The pushbuttons that is pressed represents the current plot. Information about this plot is written in the second status line (current plot status line).

Example:

```
Current Plot[0] CSDS_PP C1 EFW E_dusk panel 0 curve 0
```

The panel containing the *current plot* is the *current panel.*

The *current quantity* is set in the *Quantity dialog* activated by *Select→quantity.*

3.6.2 Setting quantities to plot

The data are read into *current quantity.* The data from quantity k is as default transferred to plot k.

In ISDAT, data is described as *conceptual instruments* in an hierarchical structure, *project member instrument sensor signal channel parameter* (see [Ref. 9]).

The *Inst* pulldown menu in the main menubar is used for selecting the *project*, *member* and *instrument* for the current quantity. The *Param* pulldown menu is used to select the *sensor, signal, channel* and *parameter* for the current quantity.

Note that the contents of the *Inst* and *Param* menues depend on the data base currently in use. The content of the menues may also depend on the access rights of the user. The *igr* actually queers the ISDAT server about descriptions of the data and dynamically builds the menues depending on the information obtained from the server and ultimately from the database. This is for example illustrated by changing the *Inst* setting. The content of the *Param* menu is then instantly re-built.

3.6.3 Specifying units of the data request

Normally, the user does not have to bother about the units of the data delivered by the ISDAT server. Information about the units comes with the data. In some cases, however, it might be useful to force the ISDAT server to deliver data in telemetry units for example. This can be accomplished by using the *Request specification dialog* activated from the *Options* button in the *Current quantity* dialog which is activated by *Select→Quantity* from the *igr menu bar.* The *Request specification dialog* is shown in Figure 3.

The options are:

TM telemetry units

Phys physical units

corr corrected units. This option applies only to certain quantities and may have special meanings in each case. Usually this is used to show some intermediate level of data processing between raw data and physical units, usually used for de-bugging and verification purposes.

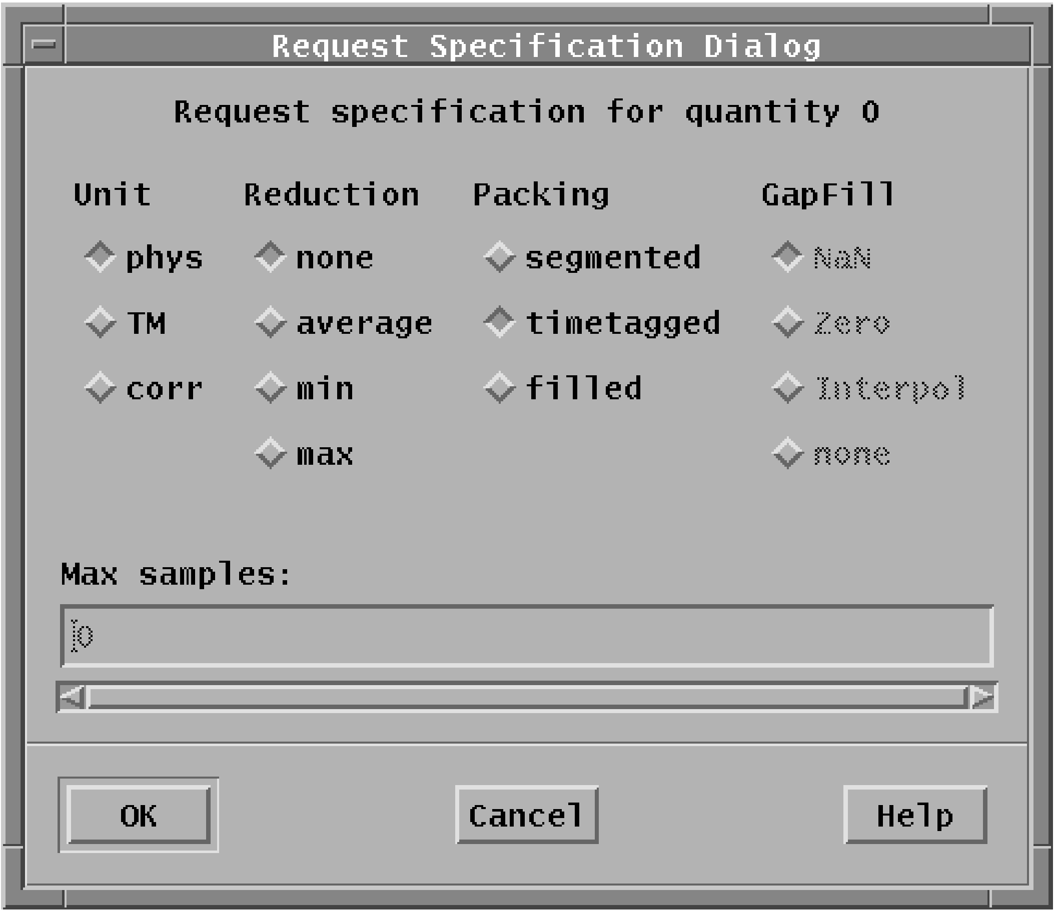

Figure 3: Igr request specification dialog

3.6.4 Limiting the number of returned samples

Normally, the ISDAT server returns all the available samples for the specified quantity within the specified time interval. In some cases, however, it may be useful to limit the number of returned samples by data reduction already in the server, for example to reduce the network traffic. This can be accomplished by using the *Request specification dialog* activated from the *Options* button in the *Current quantity* dialog which is activated by *Select→Quantity* from the *igr menu bar*. The *Request specification dialog* is shown in Figure 3. In order to reduce the number of returned samples select an option other than *none* under the *reduction* set of buttons and type in *Max samples* in the edit field. The ISDAT server will then reduce the number of samples according to yous choice:

Average where a sufficiently large number of adjacent samples are averaged in order not to exceed the specified *Max samples* .

min where the minimum value within an adequately long interval in order not to exceed the specified *Max samples* is returned.

max where the maximum value within an adequately long interval in order not to exceed the specified *Max samples* is returned.

3.6.5 Specifying the data packing

ISDAT knows of three ways of packing the data returned from the ISDAT server:

segmented where data is provided in segments with contiguous data and one time tag per segment.

time tagged where data is returned with each sample time tagged.

filled where data is returned as one segment with all gaps filled according to the user's specification.

The packing mode can be specified by using the *Request specification dialog* activated from the *Options* button in the *Current quantity* dialog which is activated by *Select→Quantity* from the *igr menu bar*. The *Request specification dialog* is shown in Figure 3.

In case *filled* is chosen, the gap filling can be one of:

NaN IEEE "not a number"

zero where gaps are filled with zeroes

interpol where linear interpolation is applied across data gaps.

3.7 Simple Plotting

The ISDAT server will inform the *igr* about the nature of the returned data. The user can specify how to treat different kinds of data by using the *calculator* (see section 3.9.2, page 32) but usually the default actions are satisfactory.

3.7.1 Plotting scalars

No particular action has to be taken by the user in order to plot scalars.

3.7.2 Plotting complex quantities

Default is to plot the *norm* of the complex quantity. See page 30 how to obtain the real and imaginary parts of a complex quantity.

3.7.3 Plotting vectors

When *igr* gets a vector variable to plot as a time series, *igr* will automatically plot the norm of the vector. If plotting of vector components is desired, this has to be specified by using the *calculator* (see "Treating vectors", page 30).

A vector variable will be appear as a gap when one or more of the vector components have *fill values.* Note that the single components can still be plotted or used in expressions. The component fill valuse will then, sometimes, be interpreted as numbers (-1*e31 for the CSDS project).

3.7.4 Plotting tensors

When *igr* gets a tensor to plot, the default action is to compute the trace and plot it as a time series.

3.7.5 Plotting spectra

When the ISDAT server returns several spectra, default action is to plot the average of all returned spectra within the specified time interval. In order to plot all returned spectra, the user must select the *spectrogram* plot (see section 3.8.9, page 18).

3.8 Advanced Plotting

3.8.1 Adjusting scales

When the *Graphics→Plot→Control* button in the main menubar is pressed the *Plot Control Dialog* window appears. The layout is shown in Figure 4 . The *Plot Control Dialog* window is used to customize the *current plot.* The top of the *Plot Control Dialog* window consists of five columns with radio buttons. The plot control settings are effective on the *current plot.* See however in section 3.8.3, page 14 how to make it work on all plots in the *current panel.*

The *Scale* radio buttons determine the maximum and minimum of the ordinate and abscissa scales. When it is on *preset* (default) the program uses the scaleMin and scaleMax

Figure 4: igr plot control window

which was provided by the ISDAT server. The scaleMin and scaleMax are elements that are provided from the data base. If *preset* is requested and no scaleMin and scaleMax are provided by the server, *auto* scaling is used.

When the radio button *auto* is pressed the *igr* client computes the maximum and minimum from the data values. When *manual* is pressed the user inputs the maximum and minimum values by using this dialog.

Only when the *ScaleY* radio button *manual* is on, the two scale windows and two text lines *ordinate scale minimum* and *ordinate scale maximum* are sensitive. The scales are used to set the maximum and minimum of the ordinate for the current plot. If higher accuracy is needed, or if the wanted maximum or minimum is outside the scale range, the user can input the numbers in the text lines below the scales. The *ScaleX* radio button determine the maximum and minimum of the abscissa scale. The radio buttons have similar functions as the *ScaleY* radio buttons explained above.

The scales *abscissa scale minimum* and *abscissa scale maximum* are used to set the maximum and minimum of the abscissa for the current plot. The scales are sensitive when the toggle button *ScaleX* manual is on.

The *Update* or *OK* buttons has to be pressed before the *igr* client will use the new settings.

The *abscissa scale minimum* and *abscissa scale maximum* have no effect when the abscissa scale is the time axis.

In every panel is printed one line with header text for each plot. This header text contains the name of the plotted parameter and the ordinate scale range.

Example:

```
CSDS_PP C1 EFW E_dusk [-20,20 mV/m]
```

The name of the parameter is CSDS_PP C1 EFW E_dusk and the ordinate scale for this parameter is -20 to 20 millivolt per meter.

Usually the different plots in a panel have different ordinate scales. However only one label and tic marks are shown.

If a common scale for all plots in a panel is desired, see section 3.8.2.

3.8.2 Setting individual or common scale

The *Panel Control Dialog* (see Figure 5) as explained in 3.8.5, is selected. Now the *scale* radio buttons are used. The default is: each curve in the panel has it's own scale (*individual*).The scale used for the plot is drawn between the [ ] brackets at the end of the plot name in the top of the panel. When the *common* button is pressed the settings for the current plot will be used for all plots in the current panel. The scales of the last used *current plot* in the panel determines the scale to be applied.

3.8.3 Setting plot type

When the *Graphics→Plot→Control* buttons in the main menubar are pressed the *Plot Control Dialog* window appears, see Figure 4.

The *PlotType* radio buttons are used to set the plot type. The default is normal line plot. In the scatter plot each data point is drawn as a cross or a dot. The selection of dots or crosses in the scatter plot is done using the pull down menus *Graphics→Plot→Marker→Dot* or *Graphics→Plot→Marker→Cross* respectively. The default is cross. In the hodogram an arrow is drawn from each data point to the next.

The default is to label the time axis only in the lowest panel. The time axis can however be labeled in other panels by using the *TimeAxis* radio buttons in the *Panel Control Dialog* (see Figure 5) .

3.8.4 Setting axis type

The *igr* client can display data along a logarithmic or a linear scale. Again the *Plot Control Dialog* window is used, see 3.8.3.

The *LinLogY* and *LinLogX* radio buttons determine if the data is plotted on a linear or logarithmic scale. The *LinLogX* radio button has no effect when the ordinate axis is the time axis. When *preset* (default) is chosen the scale type which was proposed by the data base handler is used. When the *linear* button is pressed the data is always plotted in a linear scale. When the *log* button is pressed the data is plotted in a logarithmic scale.

If the data is plotted in a logarithmic scale *(log)* is written behind the name of the parameter in the plot header text.

If it is impossible to use logarithmic scale (e.g. negative data) the *igr* client automatically switches to linear scale. The user will notify this by the missing *(log)* at the end of the plot header text.

When plotting data at a logarithmic scale, it is not possible to plot data points which are zero or negative. However, it is often useful to plot the positive part of the data and skip the negative part. When igr is plotting data at a logarithmic scale and part of the data is positive and another part is zero or negative, igr computes the smallest positive data value. This data value is automatically selected as the new scale minimum. The user can override this by selecting a new (small) positive scale minimum, see section 3.8.1, page 11. The igr client uses this algorithm both at the ordinate and abscissa axis. When plotting spectra, (1 dimensional data), and the abscissa axis is selected logarithmic, the frequencies being zero (or less) are skipped.

3.8.5 Setting frame and grids for the panel

When the *Graphics→Panel→Control* buttons in the main menubar is pressed the *Panel Control Dialog* appears, see Figure 5. The *Panel Control Dialog* window is used to

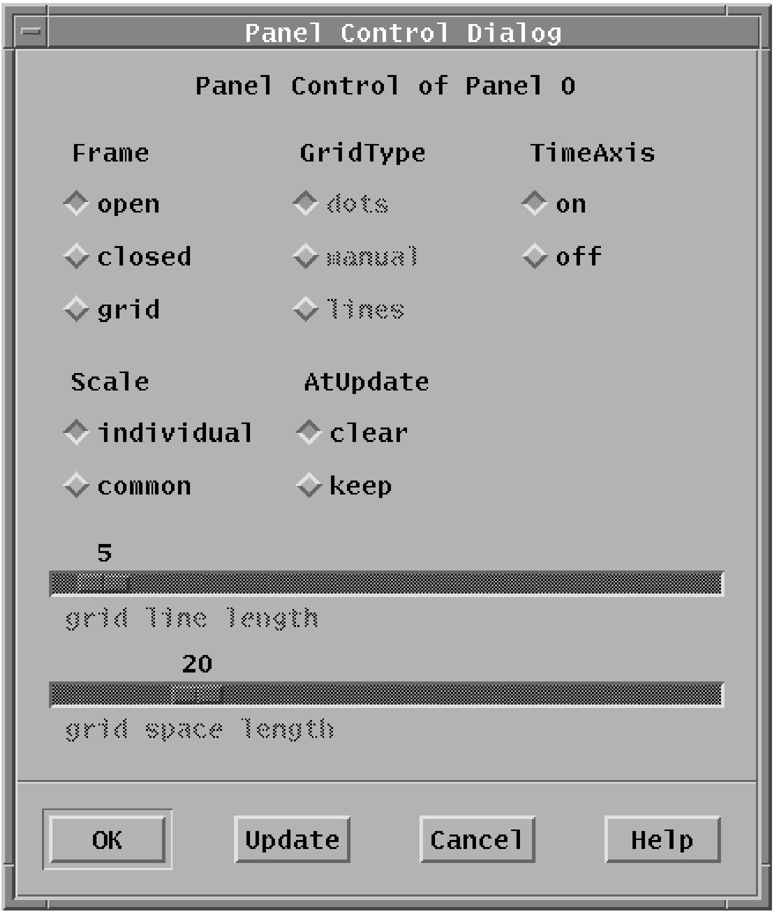

Figure 5: Igr panel control window

customize the current panel. (The current panel is the panel containing the current plot.) The *Panel Control Dialog* window consists of five columns with radio buttons.

The *Frame* radio buttons determine the type of axis used in the panel. When it is on *open* (default) only the y-axis to the left and a x-axis at the bottom of the panel is drawn. When the toggle button *closed* is pressed a frame around the panel with tick marks are drawn. When it is on *grid*, a frame around the panel and grids inside the panel are drawn. When *grid* is selected the *GridType* radio buttons can also be used.

The *GridType* radio buttons are only sensitive when the *grid* radio button is pressed. These radio buttons determines the grid type used. The default is *dots*. In this case the grid is marked with dots. The dots are one pixel wide with a 10 pixels space between each dot. When *manual* is pressed the scales below is used to set the grid line length and the space between each. When the *lines* radio button is pressed the grids are drawn as continous lines.

3.8.6 Setting colours

The *Graphics→Plot→Color* in the main menubar is used to select colour for the current plot. A list of colour names appears and the user selects the colour by clicking at one of the names. Then the colour of the *current plot* is set.

In order to adjust colours for the current panel the *Graphics→Panel→Color* in the main menubar is used.

The *Axis* is used to select colour of the axis and text in the current panel. The *Background* and *Border* is used to set background and border colour of the current panel, respectively. *Grid* is used to select grid (see section 3.8.5) colour of the current panel.

3.8.7 Plotting several parameters in one panel

It is assumed the user has already selected one time series as explained in 3.7 above.

The user presses one of the small pushbuttons in the right upper corner of Panel 0. If he presses the second pushbutton (1) the first status line (current quantity status line) will display :

```
Current Qty.[1] undefined
```

The current plot status line will display:

```
Current Plot[1] undefined panel 0 curve 1
```

(This text is written if plot 1 has not been defined before.) The user has now selected current plot 1 and current quantity 1. Then he uses the *Param* pulldown menu in the main menubar to select the data he wants to read into quantity 1. Now he presses the right mouse button when the cursor is located inside graphic Panel 0 or the *update* button of the time manager. Then the *time series* is displayed. Now in Panel 0 two parameters are displayed, (Plot 0 and Plot 1).

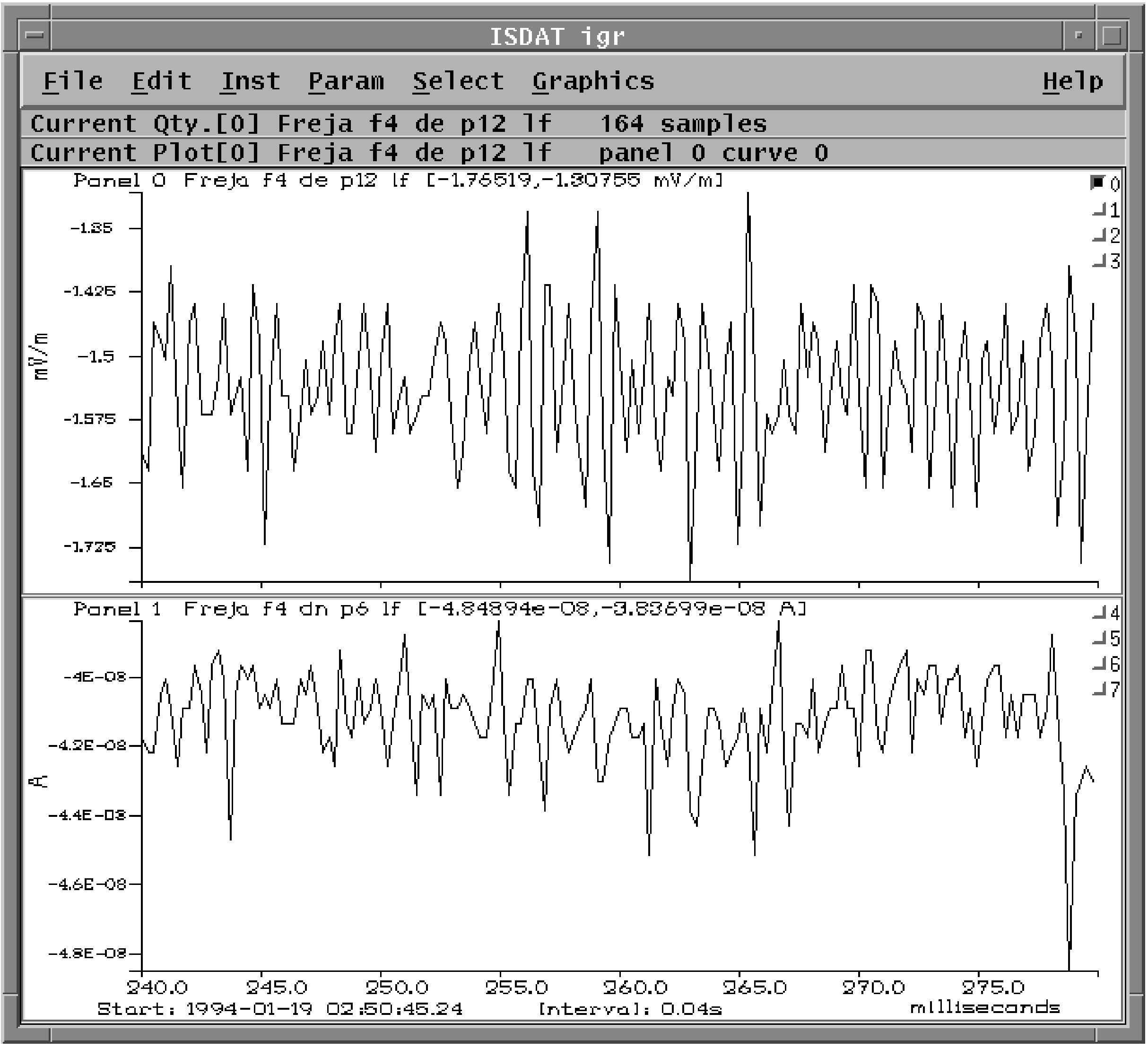

Figure 6: Igr window with two plot panels

This procedure can be repeated for plot 2 and 3. (More information about quantities and plots are given in 3.9.2 below.)

3.8.8 Plotting static spectra

When *igr* obtains spectra (dimension 1) it automatically replaces the time axis by the appropriate axis specified in the data, for example power versus frequency. See also section 3.7.5, page 11. See section 3.9.5, page 36 how to transform from time to frequency domain.

3.8.9 Plotting dynamic spectra

The data in current plot can be plotted as a 3D spectrogram. The spectrogram is controlled from the Spectrogram Dialog window. The Spectrogram Dialog is selected from the *Graphics→Plot→Spectra* in the main menubar. The Spectrogram Dialog is shown in Figure 7.

The *igr* client can currently only show one spectrogram per page at a time. Only spectra (i.e. dimension 1 data) can be plotted as a spectrogram. One example of spectra is the output of the fft function when applied in the Calculator, (eg. p0y = fft(q0)).

The Spectrogram Dialog The Spectrogram Dialog window consists of five columns with radio buttons (see Figure 7).

The Used radio buttons are used to set spectrogram *on* or *off* for the current plot.

The Zaxis radio buttons determine the amplitude scale (i.e. the color scale) maximum and minimum. When the *preset* button is pressed the maximum and minimum of the color scale are given from the data base server. When *auto* is pressed the maximum and minimum is computed from the data. When manual is pressed the user inputs the maximum and minimum values in the input lines Zaxis amplitude, minimum and maximum.

The Size radio button determines the size of the spectrogram, (i.e. rows and columns). When *auto* is selected the number of columns in the spectra is the number of samples/spectra returned from the data base server (or in the case of fft from the Calculator). The number of rows is the number of bins in each spectra. When 'manual' is selected the user inputs the number of rows and columns in the spectrogram using the Spectrogram size, rows and columns, input lines. When selecting *manual* Size the client automatically selects *manual* Zaxis.

The LinLog radio buttons determines if the data is plotted at a linear or logarithmic z-axis (i.e. color) scale. When the preset button is selected the scale type which was delivered by the data base server is used. When the linear button is pressed the data is always plotted at a linear scale. When the log buttons is pressed the data is (if possible) plotted at a logarithmic scale.

Figure 7: igr spectrogram dialog

Some quantities are only sampled in short intervals with gaps between. In the spectrogram these gaps can be closed when the *filled* button in the Gaps radio buttons is selected.

When plotting data from a big time interval the user might use the Continuous utility in the ctm time manager. In this case the user must first select manual size of the spectrogram, see section 3.8.20, page 26 .

It is possible to plot time series and spectrogram in the same panel. This is not possible in continuous mode when plotting to the screen. However, time series and spectrogram can be plotted in the same panel in a postscript file.

When the *OK* button is pressed the spectrogram is updated and the Spectrogram Dialog disappears. When the *Update* button is pressed the spectrogram is updated and when the *Cancel* button is pressed the Spectrogram Dialog disappears.

Spectrogram Colour Selection The color of the spectrogram is selected from *Graphics→ Panel→ Color→ Spectra* in the main menubar. Then the IgrColorSelect window appears. This window consists of four buttons. Below the buttons is a graphic display of the actual spectrogram color scale in use.

The left button in the IgrColorSelect window is used for selecting the number of colours in the spectrogram. When pressed a Select Color Nr window appears. This window shows a list of values for the number of colours to use. The user presses on a number and then on the OK button.

The second button from the left contains a selection for normal (NRM) and inverse (INV) colour scale. The third button from the left contains a selection of different color scales. When pressed the Select Table window appears containing a list of available color scales. The user presses one of the names in the list and then the OK button.

When the rightmost button in the IgrColorSelect window (*Done!*) is pressed the *IgrColorSelect* window disappears.

3.8.10 Adding quantities

See section 2.2.2, page 4 for the definition of the concept *quantity*. When specifying *quantities*, first set the *current quantity* in the *Quantity dialog* window (see Figure 8) activated by menu entry *Select→Quantity*.

Then specify the parameter to be associated with the *current quantity* via the *Inst* and *Param* menu entries.

3.8.11 Deleting quantities

The data is first read into the quantities, q0, q1, ..., from the ISDAT server and then later transferred to the plots, p0, p1, ..., in the *Calculator* (see sections 2.2.2 and 2.2.3). The *Quantity Dialog* is selected by pressing *Select→Quantity* in the main menubar. This dialog

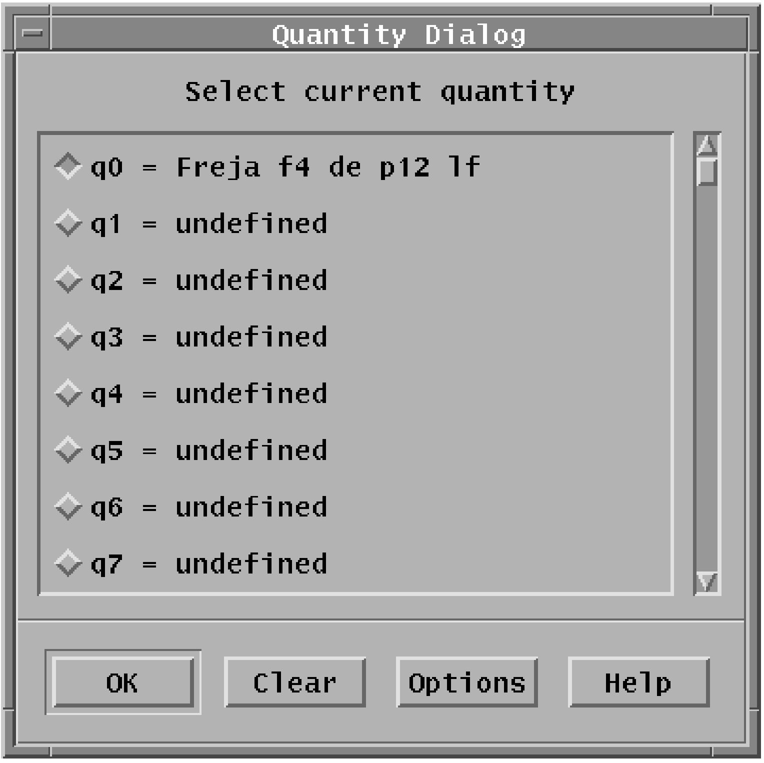

Figure 8: igr quantity dialog

shows all quantities used by the *igr* client. The *undefined* lines indicate the quantities not in use.

The pushbutton *in/selected* marks the current quantity. This quantity can be removed by pressing the *Clear* button at the bottom of the panel. Then the text after the pushbutton changes to *undefined*. No data is then read from the ISDAT server into this quantity any more.

3.8.12 Adding plots

The igr concept *plot* is defined in section 2.2.3, page 4. The specification of new plots is described in section 3.9.1, page 28.

3.8.13 Deleting plots

To delete a *plot* proceed as follows:

1. Press the *Select* → *Calculator* menu item.
2. On the *calculator* mark the *plot* to be deleted.
3. On the *calculator* press the *Edit* → *clear marked plot* menu item.
4. On the *calculator*, press the *Apply* button.

5. On the *time manager* press the *update* button.

See also page 34.

Note that during this process also the default *plot vs. quantity* binding is lost for the particular *plot*.

3.8.14 Adding panels

When the *Graphics→Panel→Create* buttons in the main menubar are pressed a *Create Panel Dialog* window appears. This window consists of a matrix with 10 times 3 push-button. Each pushbutton indicates one panel. The existing panels are in the upper left corner. The possible new panel positions are marked with sensitive (boldface) pushbuttons. When one of the sensitive pushbuttons is pressed a corresponding panel is created. Note that the *current plot* is changed to the first plot in the new panel. See Figure 6.

3.8.15 Deleting panels

When the *Graphics→Panel→Remove* buttons in the main menubar are pressed a *Remove Panel Dialog* window appears. This window consists of a matrix with 10 times 3 push-button. Each pushbutton indicates one panel. The existing panels are in the upper left corner. The possible removable panel positions are marked with sensitive pushbuttons. When one of the sensitive pushbuttons are pressed the corresponding panel is removed. At the same time all plots in this panel are removed (undefined). Note that panel 0 cannot be removed.

3.8.16 Setting size of a panel

The user is expected to have added several panels using the procedure explained in 3.8.14.

You are expected to run a window manager which allows you to resize windows. When the user resizes the main *igr* window all panels inside the window are rescaled with the same factors as the main *igr* window.

In order to change the size of an individual panel without changing the size and shape of the main *igr* window the left mouse button is used.

When the cursor is located on the panel borders the shape of the cursor is changed to a cross. The user now presses the left mouse button. The left mouse button is pressed as the cursor is moved to the new panel border position.

When the cursor is moved to the new horizontal border position the shape of the cursor is an arrow pointing down. When the cursor is moved to the new vertical border position the shape of the cursor is an arrow pointing left.

The user releases the mouse button and the *igr* client resizes the panels.

When a panel is resized all graphics inside the panel is resized to the same scale. An exception is the four small pushbutton associated with plots located in the right upper corner of each panel.

3.8.17 Expanding time axis

If a long time interval is selected (e.g. one orbit) the user probably wants to investigate some interesting features at a finer time scale. One method is to specify new *Start* and *Interval* times in the time manager (*ctm*) and then to press the *Update* button in *ctm* .

However, this can also be accomplished by the *igr* client. Now the middle mouse button is used. When the user presses the middle mouse button and the cursor is located in a graphic panel a start time is recorded. In order to notify the user about the location of the start time a vertical line is drawn. At the same time the shape of the cursor is changed to a cross. The user now moves the mouse in the rightward direction and releases the middle mouse button. The location of the release point is recorded as stop time. The time between the start time and stop time is computed and sent to the time manager. The time manger changes to this start and stop time, and then immediately notifies all clients about this new start and stop times. One of the clients is *igr* and it will now get new data from the data base handler and, *igr* will display this data. *Igr* is an exceptional client in the sence that it has the privilege to change the time manager time.

The previous time interval is recorded in a stack and can be restored with the *Edit*→*Undo*→*TimeScale* buttons in the main menu bar. The time intervals can be restored in 8 levels. When no previous time interval is available the *TimeScale* button is not sensitive.

Note that time axis expansion can only be done when data is explicitely plotted along a time axis, not when y is plotted versus x and not when spectra is plotted.

3.8.18 Plotting one parameter versus another parameter

This section contains some information about the *Calculator*. The *Calculator* is further explained in 3.9.2.

The *igr* client is started and one parameter is plotted in plot 0 and one in plot 1 as explained in 3.8.7.

The *Calculator* is selected by pressing *Select* and *Calculator* in the main menubar. The *Calculator* window now appears, see Figure 10.

The user now can release all default bindings by pressing *Edit* and *Release all bindings* in the *Calculator* window. Now all text in the lower part of the *Calculator* window disappear.

He now writes in the *Calculator* input line (upper):

```
p0y = q0
p0x = q1
```

Each line is ended by the *Enter* key or the *Apply* button. Then these two line is interpreted by the *Calculator* and the:

```
p0y = q0
p0x = q1
```

is written in the lower part of the *Calculator* window. The meaning of these two lines is that quantity 0 is plotted along the ordinate of plot 0, and quantity 1 is plotted along the abscissa of plot 0.

Now the user has to press the *Update* button in the time manager (*ctm*). Then quantity 0 is plotted as function of quantity 1.

The *Calculator* window is removed by pressing *Close* in the *Calculator* window.

3.8.19 Plotting contiguous time intervals in adjacent panels

The igr can plot different time sequences in different panels. In order to accomplish this we must use the igr lock mode. The most useful example is when Panel 0 is showing data from t0 to t1, Panel 1 is showing data from t1 to t2, Panel 2 is showing data from t2 to t3, ...,

How lock mode is started Lock mode is started when the current panel is locked with *Edit→Lock→Panel* in the main menubar. (The current panel is the panel containing the current plot). When a panel is locked further plotting is inhibited in this panel. The 3 lower pushbuttons are grayed out (insensitive) in a locked panel. Thus the user can see that a panel is locked. A panel is unlocked with *Edit→UnLock→Panel.* or *Edit→UnLock→AllPanels* in the main menubar.

Restrictions in lock mode When a panel is locked the client does not plot in the panel. Then when using the lock mode the user should NOT resize the igr window or temporary cover it with other windows. Then the time manager (ISDAT ctm) should not cover any part of the igr window (ISDAT igr) during the lock mode process.

Example on using the lock mode

1. Start the igr client.
2. Define 3 panels with *Graphics→Panel→Create.*
3. Select equal parameters for plots 0, 4 and 8 (i.e. one curve in each panel).
4. Press the 'Update' button in the time manger (ctm). Resize the igr window to a useful size. (Do not resize the window after panels are locked.)
5. Select plot 0 as the current plot, (i.e. press the small pushbutton labeled "0" in the right upper corner).
6. Press *Edit→Lock→Panel*, (panel 0 should now be locked).

7. Press the "→" arrow in ctm . Panels 1 and 2 will now update with the new time from ctm , but Panel 0 stays as before.
8. Select plot 4 as the current plot.
9. Press *Edit→Lock→Panel* (panel 1 should now be locked).
10. Press the "→" arrow in ctm . Panel 2 will now update with the new time from ctm , but Panels 0 and 1 stays as before.

Now Panel 1 should show data from t0 to t1, Panel 1 should show data from t1 to t2, Panel 2 should show data from t2 to t3. If panel 0 does not show any data more it was probably temporary covered with another window.

Hardcopy in lock mode In order to create hardcopy in lock mode the hardcopy must be initiated before the lock mode is started. This is done by pressing the 'lock mode' button in the 'Print Dialog'. Then the postscript file is opened. Each time a new panel is locked this panel is also drawn in the postscript file. When the last panel is locked the postscript file is finished. The last panel is the panel with the higest number (usually the lowest panel). The printing directly to the printer is inhibited in lock mode.

Example with hardcopy in lock mode

1. Start the igr client.
2. Define 3 panels with *Plots→Panels→Create.*
3. Select equal parameters for plots 0, 4 and 8 (i.e. one curve in each panel).
4. Press the *Update* button in the time manger (ctm). Resize the igr window to a useful size. (Do not resize the window after panels are locked.)
5. Select the *Print Dialog* from *File→Print* in the main menubar. Press the *lock mode* (and the 'portrait' button). Then press the 'OK' button.
6. Select plot 0 as the current plot, (i.e. press the small pushbutton labeled "0" in the right upper corner).
7. Press *Edit→Lock→Panel*, (panel 0 should now be locked).
8. Press the "→" arrow in ctm . Panels 1 and 2 will now update with the new time from ctm , but Panel 0 stays as before.
9. Select plot 4 as the current plot.
10. Press *Edit→Lock→Panel* (panel 1 should now be locked).
11. Press the "→" arrow in ctm . Panel 2 will now update with the new time from ctm , but Panels 0 and 1 stays as before.
12. Select plot 8 as the current plot.
13. Press *Edit→Lock→Panel*, (panel 2 should now be locked).

When panel 2 was locked the *igr hardcopy info* window appears with the information:

```
 ---------------------------
| The hardcopy in lock mode |
| is now finished.          |
| File: filename            |
 ---------------------------
```

The filename is replaced with the actual file name.
The filename should contain the data in postscript format.

```
Panel 1 should show data from t0 to t1.
Panel 1 should show data from t1 to t2.
Panel 2 should show data from t2 to t3.
```

3.8.20 Running in continuous mode

When plotting a large time interval and using a quantity which produce a huge number of data values the computer might run out of memory. In this case it is better to read the large time interval in smaller subsequent intervals. The *ctm* time manager can read time intervals in continuous mode. The continuous part is selected from *Options→continuous* in the *ctm* time manager. The end time of the continuous interval is written into the text field and the run button is pressed in *ctm* .

The default behavior of igr is to display data with each of the small time intervals received from the data base server as the length of the time axis. However, by setting igr into continuous mode the whole continuous interval can be used as the interval of the time axis.

The continuous mode is set in the Page Control Dialog. When the *Graphics→Page→Control* buttons are pressed the Page Control Dialog window appears. The Page Control Dialog window is used to customize the igr page (window, i.e. all panels), see Figure 9.

The *Update* or *OK* buttons in this dialog must be pressed before the selections are used.

The Continuous mode toggle button is used to set igr into continuous mode. In this case the user should use the Continuous mode option in the *ctm* time manager. Then the interval displayed in the igr panel is the whole interval from the Start time to the Continuous time in the *ctm* time manager. When the Run button is pressed in *ctm* the plotting start. When the whole continuous interval is displayed the Continuous mode in *ctm* and igr are automatically set off.

Because the data from the whole continuous interval is not stored in memory, a redraw of the window will redraw only the last (small) data interval read from the data base server and not the whole continuous interval. It is important that the user do not cover, resize or do anything else with the igr window and trigger a redraw request when the client is drawing i continuous mode. (The spectrogram are always stored in memory and is redrawn with the actual length after a redraw request.)

Figure 9: igr page control window

For time series the manual ordinate scale must be selected, see 3.8.1 .

The continuous mode can also be used for spectrograms. In this case the manual size of the spectrogram must be selected, see 3.8.9.

In order to print to a postscript file in Continuous mode the user must first press the Continuous mode toggle button (and OK or Update) in the Page Control Dialog. Then press OK in the Print dialog. At last he should press Run in the ISDAT *ctm* . The printing directly to printer is disabled in Continuous mode. The spectrogram is not shown on the screen when printed to a postscript file.

It is possible to plot time series and spectrogram in the same panel. This is not possible in continuous mode when plotting to the screen. However, time series and spectrogram can be plotted in the same panel in a postscript file.

3.8.21 Expanding and compressing header text

When using the Calculator and creating a plot from several quantities, the panel header text often becomes to long to be printed inside the panel. This can be modified from the Page Control Dialog window.

When the *Graphics→Page→Control* buttons are pressed the Page Control Dialog window appears. The Page Control Dialog window is used to customize the igr page (window, i.e. all panels), see figure 9. The *Update* or *Ok* buttons in this dialog must be pressed before the selections are used.

The Expand text toggle button is used to customize the length of the plot names printed in the top of each panel. When the Expand text toggle button is on/in the quantity names are expanded. Example:

```
(Cluster 1 efw E p1 10Hz - Cluster 1 efw E p3 10Hz) [-1.716,1.992]
```

When the Expand text toggle button is off/out the quantity names are not expanded. In the same example as above is now written: (q0 - q1) [-1.716,1.992] In this case the user must use the Quantity Dialog (*Select→Quantity*) to see what parameters each quantity contains.

In the postscript plots a list of all quantity names is written at the bottom of the page. This list is selected/removed by using the with *Print quantity names* pushbutton in the Print Dialog, see section 3.10.3, page 42.

3.9 Manipulating Data

3.9.1 Overview

The builtin *calculator* can do arithmetical operations on the *quantities* defined in *igr*. It can make functions to operate on them, rotate them and "join" quantities. The calculator also defines in which plot in *igr* the result of the operations should be displayed, and on which axis. Note that the quantities are not known to the *calculator* unless an *update* has been performed after the quantity was specified. It should also be noted that unit labels may be missing for manipulated data.

Examples:

- 'p3y = q1 + q2' Which means that the sum of quantities q1 and q2 will be plotted in plot 3 on the y-axis (and time along the x-axis which is default).
- 'p4x = q2' and 'p4y = 4*(q1 - q0)' will plot the expression of the quantities q1 and q0 as a function of the quantity q2.
- 'p1y = {T0*q2}[0]' will plot the first component of the result of the multiplication of tensor the T0 (to be set up by the user) and the quantity q2.

The expressions are typed in the text field at the top of the window (see Figure 10). The plot specifications and quantity codes can alternatively be chosen from the pull-down menus just below the input text field. The result will then be seen in the text field at the bottom of the window, see Figure 10. When an update is performed (either on the *time manager* or in *igr*) the expressions in the lower text field will be used to define what to plot in the different plots of *igr*.

Legal operators: Vector multiplication cross 'x' and dot '*', division '/', addition '+' and subtraction '-'. Division by zero normally introduces a gap in data for integer values.

Legal tokens: In addition to the operators listed above, '(', ')', '[', ']', '{', '}' are legal.

- ( ) are used for overriding precedence rules.
- [ ] are used for indexing vectors

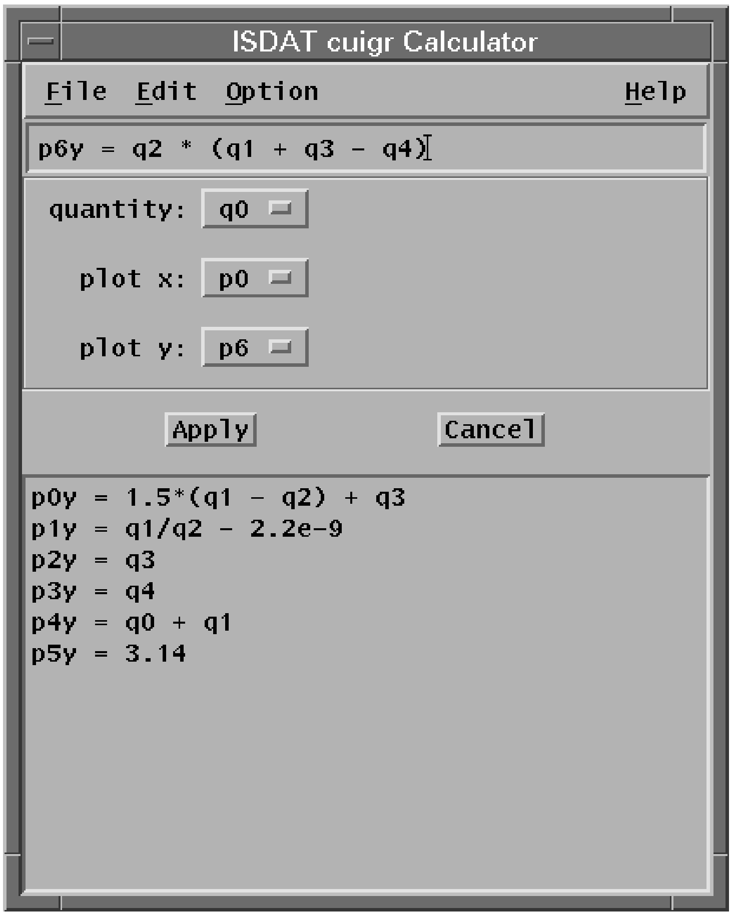

Figure 10: The igr calculator window

- { } are used for indexing the vector results of an operation (see "Expression with tensors", page 34).

Quantity names

```
q0, q1, ......
```

Plot names and axes

```
p0x, p1x, p2x, ......
p0y, p1y, p2y, ......
```

The quantities may have an index

```
q0[0], q0[1], q0[2]
q1[0], q1[1], q1[2]
.
.
.
```

After a data manipulation, the pre-set scale values will be over-written with max and min values found for the resulting value for the current interval. *Manual scaling* will be retained. A plot can be set to a constant, e.g. p0y = 5 provided that at least one *quantity* is defined. This is needed to define the time interval for the *calculator*. Note however that the setting to a constant normally is only meaningful if all *plots* in the *panel* have manually set scales equal for all plots in the panel.

Precedence in arithmetical expressions: Unary minus has the highest precedence. 'x' and '/' have higher precedence than '+' and '-'. To override this precedence scheme use paranthesis.

Treating complex quantities The real and imaginary components of a complex quantity is obtained by using the functions *re()* and *im()*. Example:

```
p0y=re(q0)
p1y=im(q0)
```

where $q0$ is a complex quantity. See also section 3.7.2, page 11 for default treatment of complex quantities.

Treating vectors If the quantity to plot is of type vector, or the result of the expression is of type vector, the norm is computed before it is plotted. If plotting of components is desired it has to be explicitely specified by using brackets, e.g. q[0]. Components are numbered from 0. See also "Expression with tensors", page 34.

An example of vector component plotting is shown in Figure 11.

Transformations to polar coordinates The calculator contains predefined functions to convert from a vector in Cartesian coordinates and associated settings to the components of a polar coordinate system. The functions are: pol_r(), pol_phi() and pol_th().

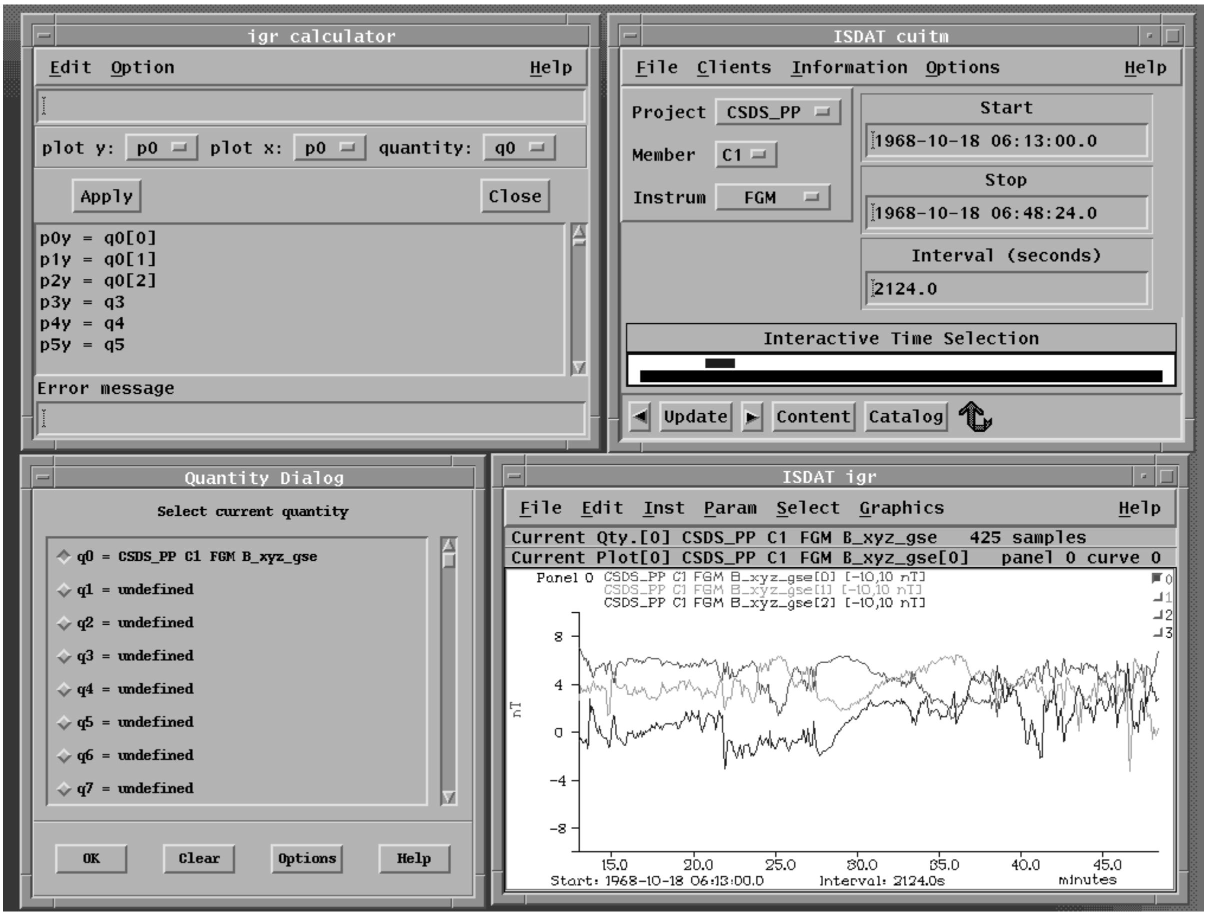

Figure 11: Plotting of vector components

The function pol_r(v) is defined as $\sqrt{x^2 + y^2 + x^2}$.

The function pol_phi() is defined as atan(y/x), the result is given in degrees in the range -180 to 180 degrees.

The function pol_th() is defined as acos(z/r) where r = $\sqrt{x^2 + y^2 + z^2}$, the result is given in degrees in the range 0 to 180 degrees.

To get the length of a vector q0 use the equation p0y = pol_r(q0).

Joining of data Joining will be automatically performed if needed. E.g. p0y = q0 + q1, if q0 and q1 do not have the same number of samples or the sample times are not the same, the *Calculator* will automatically join the least sparse of the two quantities to the same time line as the most sparse one. Fuzzy join is used as described in [Ref. 2]. Joining of data is further described in section 3.9.4.

Word explanations and definitions:

Binding the connection of a quantity or an expression of quantities to a plots x- or y-axis

Setting used in the same meaning as binding

Joining the modification of a quantity so its sampling frequency and "time line" will be in accordance with another quantity

Plot name a 'p' and the plot number, e.g. q3

Tensor name a 'T' or 't' and the tensor number, e.g. T5

3.9.2 Use of the calculator

Starting the Calculator The *Calculator* can be started by choosing *Calculator* in the *Select* menu on the *igr* menu bar.

Exiting from the Calculator To exit the *Calculator* press the *Close* button.

Making quantity to plot bindings To make a binding of an expression of quantities to a plots x- or y-axis, use the input text field a the top of the window. First should the plot name (a 'p', the plot number, e.g. p12) and axis specification (x or y) be typed in. Then an equal sign as a separator, and finally the expression. Examples:

```
p0y = q0
p2x = q1*q2 + 10.0
```

If the axis is not specified, *Calculator* will assume that it is the y-axis. The plot names can alternatively be choosen from the 'plot x' and 'plot y' menus below the input text field. These menus contain the first 20 plot names and axis. Choosing one of these will

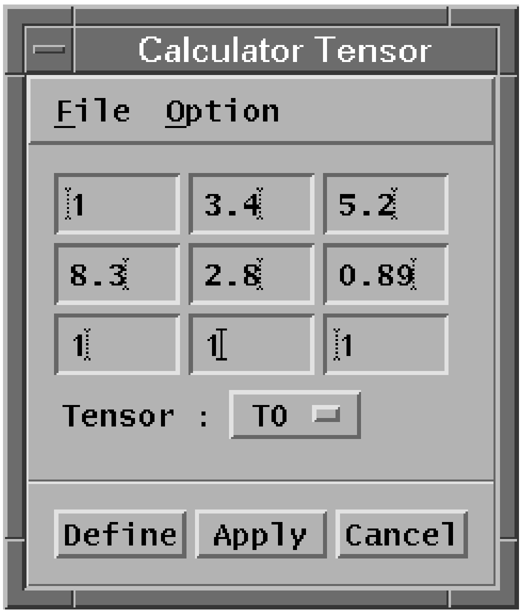

Figure 12: The calculator tensor

clean the input text field and print the plot name and an equal sign. Double-clicking on an expression in the text-field at the bottom of the window will result in copying the expression to the input text field at the top of the window. Syntax- or semantic errors will normally result in an error message in the error message field below the *Apply* button. Some help is usually provided by underlining the character, where the problem was first identified.

Making corrections Point with the mouse on the character just after the part to erase. Press the left button and the cursor is in position for using the 'Back space' button.

Specifying a tensor Under the menu *Options*, choose *Tensor* and a window will pop up, see Figure 12.

That window has a field for a 3x3 tensor where all elements can be typed in separately. First choose a tensor name on the menu below the tensor field. Up to eight different tensors can be defined. Type in the tensors elements and press the *Define* button. This tensor can now be used in expressions in the *Calculator*.

Saving a tensor to a file In the Tensor windows File menu, choose *Save*. A file selector will then pop up. Select the file name where to save the tensor currently seen in the Tensor windows tensor field. Press the *OK* button and the tensor will be written to the file and the file selector window will disappear. Note that tensors are also saved in the normal configuration files, see section 3.10.4, page 44.

Reading a tensor from file In the Tensor windows *File* menu, choose *Read*. A file selector will then pop up. Select the file you desire to read the tensor from. Press the *OK* button and the tensor will be copied to the Tensor windows tensor field, and the file selector window will disappear. Choose a tensor number from the tensor menu. Press the *Define* button on the Tensor window so this tensor can be used in the *Calculator*.

Expression with tensors Tensors can be multiplied by expressions, e.g. p0y = T1*q2 The result of the expression (T1*q2) is of type vector (if q2 is a vector) and the norm will be computed so it can be plotted. If a specific component of the vector should be plotted, it can be obtained by extracting it from the result, e.g. p0y = {T1*q2}[1] where the second component is extracted (the components are numbered 0, 1 and 2). Observe curly braces should be used, not paranthesis. Since vectors are assumed to be "column arrays", only tensors multiplied by a vector, $\mathbf{T0} * \mathbf{v}$, is legal, not $\mathbf{v} * \mathbf{T0}$. A tensor could be multiplied to an expression including tensor expressions, e.g. p2y = T3*(q1+T3*q0).

Erasing all settings To take away all bindings and make the plot text field clean, choose *Release all bindings* under the *Edit* menu.

How to retrieve the default settings To erase all bindings and retrieve the default bindings, choose *Reset to default bindings* under the *Edit* menu.

Erasing a binding To erase one or more bindings, and still retain the other, mark those bindings by clicking on them with the mouse. Then, in the *Edit* menu, select *Clear marked plot*. The marked bindings will then dissappear, and if an update is performed, these plots will no more be displayed. To take away an unwanted marking of a binding, point on that binding with the mouse and press the control key and the left button on the mouse.

How to take away an error message To erase the error message in the error message field (below the *Apply* and *Cancel* buttons) either correct the error and make an *apply* again or select *Clean input line* in the *Edit* menu.

3.9.3 Introducing time shifts between quantities

If the user wants to compare the data from two different satellites or any other quantities it might be useful to shift the data from one satellite relative to the other. One example is the crossing of a boundary when the two (or more) satellites have almost the same orbit.

Two types of time shifts are implemented in igr. They are named *plot time shift* and *quantity time shift*.

When doing *plot time shift* a time interval is added (or subtracted) to one of the plots when drawing the data in the panel. **This is suitable for small time shifts**.

Figure 13: igr time shift dialog

When doing *quantity time shift* a time interval is added (or subtracted) to the quantity time when requesting for data from the data base server. **This is suitable for large time shifts**.

The time shift is specified in the *Time Shift Dialog*. The user presses *Edit→TimeShift→ Plot or Edit→TimeShift→Quantity* to select the *plot time shift* or *quantity time shift* respectively. The *Time Shift Dialog* is shown in Figure 13.

The Time Shift Dialog consist of two radio button columns:

Used : The user set time shift on for this quantity (or plot) by pressing the *on* button. When this button is pressed the *Direction* radio buttons becomes sensitive.

Direction : When the *forward* (default) button is pressed a time is subtracted from the quantity (or plot). In this case the curve moves forward in the panel. When the *backward* button is pressed a time is added to the quantity (or plot). In this case the curve moves backwards in the panel.

The text input line is used for input of the time shift value. The format must either be:

```
    hh:mm:ss.ms
or
    ss.ms
```

`(hh is hour, mm is minute, ss is second, and ms is fraction of a second)`

When the OK button is pressed the time shift settings for the current quantity (or plot) is updated and the Time Shift Dialog disappears. When The Update button is pressed the time shift settings for the current quantity (or plot) is updated. When the Cancel button is pressed the Time Shift Dialog disappears.

The time shift is indicated with ≪ time shift ≪ or ≫ time shift ≫ for backwards or forwards shift behind the name of the parameter in the plot header text. As an example:

```
Cluster 2 efw E p1234 10Hz [0.1,3 V] >>00:00:01.0>>
```

means, this parameter is shifted forward with 1 second.

3.9.4 Joining data sets to a common time line

See also page 32. To force which quantity will be guiding, when we adopt quantities to the same sampling frequency and to the same time line, the syntax is 'Join(qX, qY)'. The result of 'Join(q2,q3)' is q3 adopted to the sampling frequency and time line of quantity q2. 'p0y = q0 + join(q0, q1)' will thus add q0 and q1, and the result will have the same time line as quantity q0. If we don't use this forced joining, the *Calculator* will automatically perform joining (if necessary) with the most sparse quantity as guiding. Joining of several quantities is not yet implemented, at the moment one has to explicitly make a binding for each quantity. E.g., quantities q0, q2, q4, q6 and q8 should be joined with q2 as the quiding quantity:

```
p0y = join(q2,q0)
p1y = q2
p2y = join(q2,q4)
p3y = join(q2,q6)
p4y = join(q2,q8)
```

3.9.5 Transforming to frequency domain

Most of the data provided by the *ISDAT server* is provided in the time domain. There are some exceptions where the server provides data in the frequency domain. The way to display frequency domain data is described in section 3.8.9, page 18. This section describes how to transform data from the time domain to frequency domain and how to eventually display the result.

The *igr calculator* (see section 3.9.2) has a number of builtin *operators*. One is the *FFT* operator. The *FFT operator* is accessed from the *calculator* (section 3.9.2). The syntax is:

```
p0y=fft(q0)
```

The power spectral density of the data contained in the specified time interval is then computed and bound to plot 0. In order to cover the exact time interval, a FFT size less than the number of samples within the time interval is used and in general two or more partially overlapping FFT's are computed and averaged in order to cover exactly the data set. The FFT parameters used can be inspected in the *FFT operator dialog* shown in Figure 14.

The *FFT operator dialog* is activated by marking the *fft* string in the calculator edit field and then select *Edit→Operator* in the *calculator* menubar. Note that the *FFT operator*

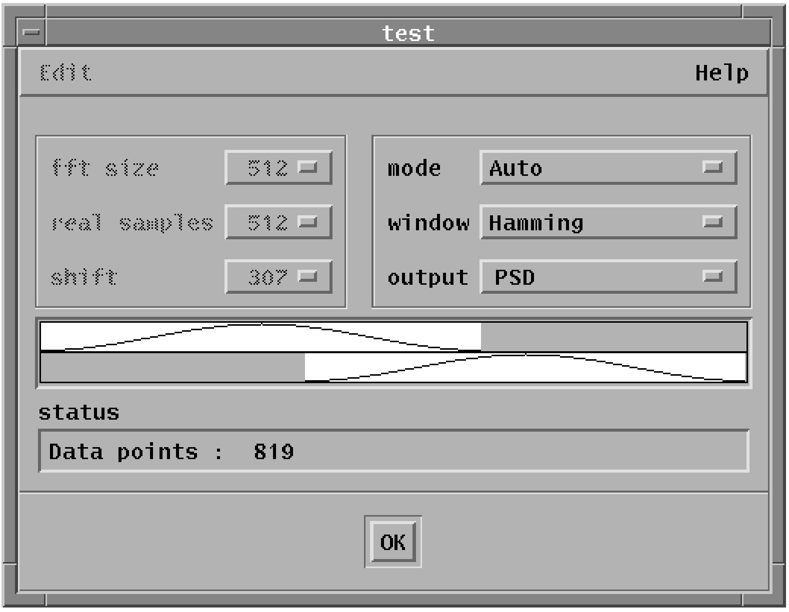

Figure 14: FFT operator dialog

dialog cannot be activated until the operator has been specified in the edit field and the *Apply* button has been pressed and the expression has been accepted by the calculator. If the default settings are not satisfactory, it is possible to manually set the parameters by selecting *mode→manual* and then specify the five parameters:

Window where the following windows are currently available:

1. Rectangular
2. Hamming
3. Balckman Harris
4. Gaussian

Output where the following options are available:

1. PSD - Power Spectral Density
2. Real - Real part of the complex Fourier transform
3. Imaginary - imaginary part of the complex Fourier transform
4. Phase
5. Amplitude

FFT size Select out of the proposed numbers or edit your own number by activating the *Edit→FFT size* edit field. Note that only power of 2 numbers are accepted.

Real samples specifies how many samples from the data to use. Missing samples are zero filled. Select out of the proposed numbers or edit your own number by activating the *Edit→Real samples* edit field.

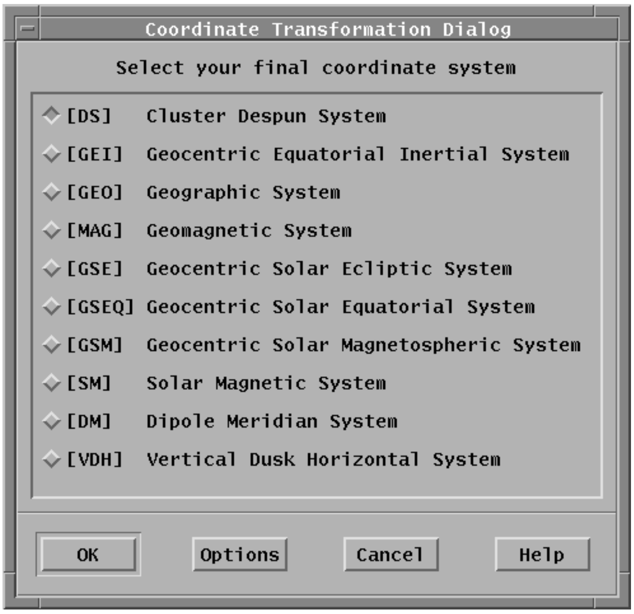

Figure 15: despin operator dialog

Shift Select out of the proposed numbers or edit your own number by activating the *Edit*→ *Shift* edit field.

Static spectra As long as the *igr*→ *Graphics*→ *Panel*→ *Control panelType* is in *Normal* (see Figure 5) the spectra are displayed as one static spectrum based on averages of elementary spectra. See also section 3.8.8, page 18 regarding plotting of static spectra.

Dynamic spectra When *igr*→ *Graphics*→ *Panel*→ *Control panelType* is in position *spectrogram* (see Figure 5) the spectra are colour coded dynamic spectra.

3.9.6 Performing coordinate transformations

The simple Cartesion to polar coordinate transformation is described in section 3.9.1, page 30.

A number of transformations between reference systems is provided by the *despin* operator built into the *calculator*. Note that the *despin* operator currently only works for the Cluster project. The syntax is:

```
p0y=despin(q0)
```

The desired coordinates are then specified in the *despin operator dialog* shown in Figure 15.

The *despin operator dialog* is activated by marking the *despin* string in the calculator edit field and then select *Edit*→*Operator* in the *calculator* menubar. Note that the *despin operator dialog* cannot be activated until the operator has been specified in the edit field and the *Apply* button has been pressed and the expression has been accepted by the calculator.

The following final coordinate systems are available:

DS - Platform de-spun system

GEI - Geocentric Equatorial System

GEO - Geographic system

MAG - Magnetic system

GSE - Geocentric Solar Ecliptic System

GSEQ - Geocentric Solar Equatorial System

GSM - Geocentric Solar Magnetospheric System

SM - Solar Magnetic System

DM - Dipole Meridian System

VDH - Vertical Dusk Horisontal System.

The coordinate transformations are described in more detail in [Ref. 1] and [Ref. 10] .

3.10 Exporting and importing

3.10.1 Saving data in a flat File

The data (in the time interval) displayed in the graphic panels can be stored in an ASCII file. When the *File→SaveData→FlatFile* buttons in the main menubar are pressed the *Flat File Dialog* appears, see Figure 16. This dialog consists of a scrolled panel with 8 buttons at the top and a file selection part at the bottom. The file selection part is not sensitive when the dialog starts.

The user now presses one of the eight pushbuttons (with numbers from 0 to 7). Then the current plot is assigned to one column in the flat file. If the user then wants to save several plots in the same flat file he selects another plot as the current plot (see section 3.6.1).Then he has to press one of the other buttons (with numbers from 0 to 7). If the user wants to undo the selection he presses the *Reset* button. Then all the buttons are marked with *undefined*.

When the plot selection is finished he presses the *Apply* button. Now the file selection part becomes sensitive. The user selects one of the files in the *Files* panel or writes a new file name in the *Selection* line. The default name is *$HOME/igr.flat*.

When the *OK* button is pressed the flat file is created.

The error messages are written in the *Error message* line at the bottom of the dialog.

One example of a flat file is:

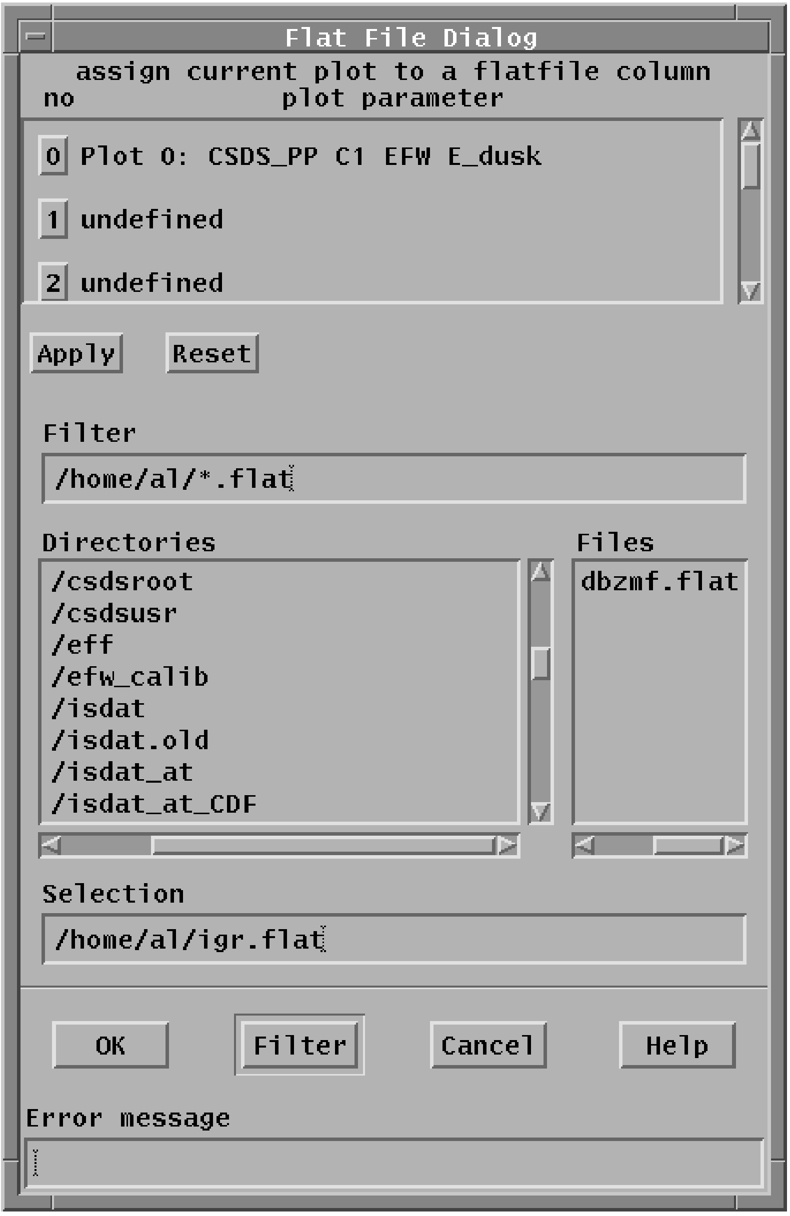

Figure 16: Flat file dialog window

```
# Data from ISDAT igr
#
# Filename:      igr.flat
# Created date:  Sat May 25 18:36:48 1996
# Version:       blybekk@efw 1996-05-25 18:36:48.0
#
#
#
# Time start at: 1996-02-04 16:41:52.304558
#
#   Time from start   , Cluster 1 efw E p2 10Hz
#      [s]               [V]
       0.099708889        -1.458
       0.299708889        -1.114
       0.499708889        -0.702
       0.699708889        -0.242
       0.899708889          0.23
       1.099708889         0.682
       1.299708889          1.09
       1.499708889         1.428
       1.699708889         1.674
       1.899708889         1.816
       2.099708889          1.84
       2.299708889         1.744
       2.499708889         1.542
       2.699708889         1.238
       2.898854555         0.852
       3.098854555          0.41
       3.298854555        -0.054
       3.498854555        -0.524
       3.698854555        -0.958
       3.898854555         -1.33
       4.098854555        -1.626
```

The lines starting with '#' are comment lines. The left column shows seconds from start of the time interval. The second column contains data values. The user can for example use *xgraph* to display the two first columns of the flat files. (*xgraph* is a standard X-tool, not included in the ISDAT package.)

Only data with a common time line can be saved in one file. See however, section 3.9.4, how to force data to a common time line.

3.10.2 Saving data as a CDF File

The data in the time interval displayed in the graphic panel can be stored in a CDF file. When the *File→SaveData→CDF* buttons in the main menubar are pressed the *CDF control* window is displayed.

One to sixteen plots can be stored as a parameter in a CDF file. These plots are associated with the buttons 0 to 15. The user presses one of these buttons to assign the current plot to a parameter in the CDF file. The variable names are set to DP_0 to DP_15 by default but can be changed by editing the names.

The user can add his own entries of the global attributes *Data_version*, *MODS* and *Caveats* by pressing the *Global attributes* button.

All associations between plots and CDF variables can be removed by pressing the *Reset* button.

Pressing *Apply* will do a consistency check of all defined variables verifying that they all have the same time resolution. If the consistency check fails an error message will be printed at the bottom of the window.

If all is good the file selection part will be set sensitive, ie. the greyish look will disappear, a default file name will also be constructed based on the selected start time and time resolution. The default name can be changed by editing the Selection field.

When the *OK* button is pressed the CDF file is created.

3.10.3 Printing the graph

The data displayed in the panel can be copied to a postscript file or directly printed as a hard copy. Note however, that in cases of extremely much data ($\geq$ 10000 samples) it may be more practical to make a screen dump and not produce a postscript file.

When the *File→Print* buttons in the main menubar are pressed the *Print Dialog* window appears, see figure 17.

This window consists of four radio button columns.

Which Data: When the *current plot*, *current panel* or *all panels* radio button is pressed the data from current plot, all plots in current panels or all panels respectively are copied to a postscript file. The *lock mode button* is used when printing in lock mode and is further explained in section 3.8.19, page 24.

Which device: When the *file* radio button is pressed the data is written to a postscript file. The name of this file is written in the *File name:* text field. When the *printer* radio button is pressed the data is transferred directly to the printer. The printer name is now written in the *Printer name:* text field.

B&W or Color: When the *black&white*, *b&w alt* or *color* radio button is pressed the hardcopy will be in black and white, black and white with dotted lines or color respectively.

Orientation: The *landscape* and *portrait* buttons determines if the plot should be orientated in landscape or portrait direction.

Title text: If a text is written in the *Title text field*, this line is printed at the top of the postscript page.

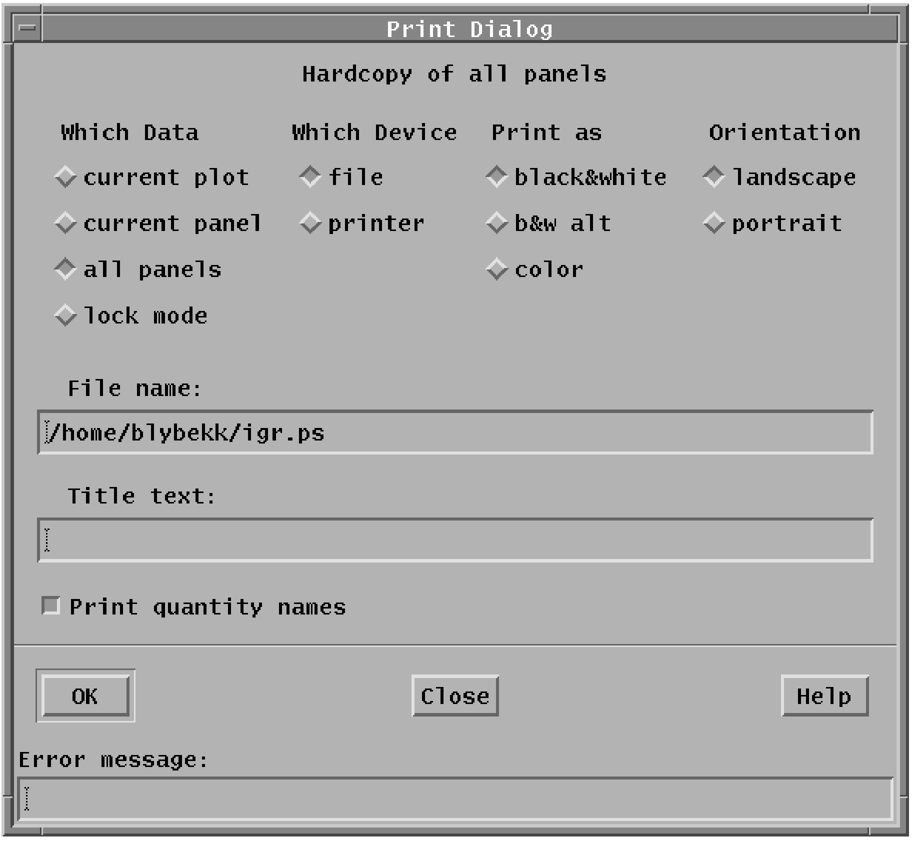

Figure 17: Igr print dialog window (Figure not up to date)

Print quantity names At the bottom of the postscript page a list with the quantities in use and the associated quantity names are printed together with version numbers for software used to produce the quantity in question. The format of the version information is:

```
(a.b.c.d.e)
```

where

a is the ISDAT version

b is the ISDAT revision

c is the main instrument server module, e.g. WEC, software version

d is the minor instrument server module, e.g. EFW, software version

e is the version of the calibration file

. The user can select the printing of this list with the *Print quantity names* push-button in the Print Dialog.

The default printer command is *lp*. The default command can be modified in the igr resource file:

```
Specifies the print command with options
example: Igr*printCommand: lp
         Igr*printOption:  -d
         When nothing is specified in the "Printer name:"
         line in the PrintDialog the printer command
         will be: 'lp'
         When a name of a printer e.g. 'myprinter' is
         written in the "Printer name:" line the printer
         command will be: 'lp -d myprinter'

Igr*printCommand: lp
Igr*printOption:  -d
```

See further section 5 how to modify the environment.

When the *OK* button is pressed the postscript file is created and the dialog window disappears. The postscript file can then be inspected using a postscript preview program, (imagetool, gs, ghostview, pageview, ..).

3.10.4 Saving an igr layout

The complete settings of the *igr* client can be stored in a file. This file is an ASCII file and is named the *configuration file*. When the *File→Config→Save* buttons in the main menubar are pressed the *Config Save Dialog* window appears. This is a *Motif File Selection* dialog.

The filename is either selected in the *Files* panel or written behind the directory name in the *Selection* line. If not present the extension ".igr" is automatically added to the filename. The user presses the *OK* button to save the file. The program checks if the file name is valid (and not a directory). Error messages are written in the *Error message* line in the *Config Save* dialog window.

If the file already exists a *warning* dialog appears. The user then presses *Yes* to overwrite or *No* to cancel.

The configuration files can be interchanged between different users (e.g. with e-mail), running *igr* of the same version.

Note that the time specification is not an *igr* setting and is thus not saved in the igr configuration file. See however [Ref. 8] how to save time in a *ctm* configuration file.

3.10.5 Restoring an igr layout

The *igr* client can read and interpret the configuration files saved as explained in 3.10.4. Now the *File→Config→Load* buttons in the main menubar are pressed. Then the *Config Load Dialog* window appears.

The user selects the correct file and presses the *OK* button. When the configuration file is loaded the *igr* client gets the same configuration as when this file was stored. Note, however, that the time interval is a time manager issue and is not restored. Error messages are written in the *Error message* line in the *Config Load* dialog window.

3.10.6 Predefined plot layout

For the CSDS a number of predefined plot layouts are supplied.They can be retrieved via the *File→Config→Predefined* menu button.

In CSDS User Interface, Release 4, the following pre-defined configuration files are provided:

```
File                        igr setting
---------------------------------------------------------------------------
CSDS-PP-C1.igr              A 24 panel collection of PP from Cluster 1
CSDS-PP-C1_overview.igr     Like the CSDS Summary plot but for PP Cluster 1
CSDS-PP-C2.igr              A 24 panel collection of PP from Cluster 2
CSDS-PP-C2_overview.igr     Like the CSDS Summary plot but for PP Cluster 2
CSDS-PP-C3.igr              A 24 panel collection of PP from Cluster 3
CSDS-PP-C3_overview.igr     Like the CSDS Summary plot but for PP Cluster 3
CSDS-PP-C4.igr              A 24 panel collection of PP from Cluster 4
CSDS-PP-C4_overview.igr     Like the CSDS Summary plot but for PP Cluster 4
CSDS-PP_ASPOC.igr           ASPOC PP from all four satellites
CSDS-PP_CIS.igr             CIS PP from all four satellites
CSDS-PP_DWP.igr             DWP PP from all four satellites
CSDS-PP_EDI.igr             EDI PP from all four satellites
CSDS-PP_EFW.igr             EFW PP from all four satellites
CSDS-PP_FGM.igr             FGM PP from all four satellites
CSDS-PP_PEACE.igr           PEACE PP from all four satellites
```

CSDS-PP_RAPID.igr	RAPID PP from all four satellites
CSDS-PP_STAFF.igr	STAFF PP from all four satellites
CSDS-PP_WHISPER.igr	WHISPER PP from all four satellites
CSDS-SP.igr	A 24 panel plot of summary parameters
CSDS-SP_AUX.igr	A 30 panel plot of all AUX summary parameters
CSDS-SP_comp.igr	Like the CSDS summary plot
CSDS-SP_config.igr	Like the CSDS summary plot
CSDS-SP_fields_1.igr	Like the CSDS summary plot (part 1)
CSDS-SP_fields_2.igr	Like the CSDS summary plot (part 2)
CSDS-SP_overview.igr	Like the CSDS summary plot
white_landscape.igr	An empty plot panel with A sheet proportions, landscape
white_portrait.igr	An empty plot panel with A sheet proportions, landscape

3.11 Running igr in batch mode

The igr client can be run in batch mode. Then it is possible to run the igr client during the night to produce postscript plots of selected events. The user does then not need to be present while the client is working.

The data base server (dbh) and the time manager (ctm) must have been started in advance. The igr client is then started manually in a xterm window. The syntax is:

```
igr filename
```

The filename is a igr batch script file. The filename is an ascii file. The first line in the igr batch file must be

```
IgrBatchFile
or
IgrBatchFileDebug
```

(The reason for this is, the user should not be allowed to force igr to interpret other files than valid igr batch files).

When IgrBatchFileDebug is used the result of the batch file processing is written in a text window.

3.11.1 Batch file syntax

When a # is found in first character position the rest of the line is regarded as a comment line. Empty lines are ignored.

A batch file contains several blocks The block names are:

```
BATCH_CONFIG
BATCH_TIME
BATCH_HARDCOPY
BATCH_EXIT
```

The first block must be BATCH_CONFIG and the last block must be BATCH_EXIT . Each block start with BEGIN and end with END

The BATCH_CONFIG block:

```
BATCH_CONFIG
BEGIN
    CONFIG_DIR directory
    CONFIG_FILE filename
END
```

The BATCH_CONFIG block defines a igr configuration file (see 3.10.4). The directory is the directory where the configuration file is located, and filename is the configuration file name. The configuration defined in this file is loaded before the batch processing starts. This configuration file must be created manually by the user before it can be used in igr batch processing.

The BATCH_TIME block:

```
BATCH_TIME
BEGIN
   START     yyy-mm-dd hh:mm:ss.s
   INTERVAL  sss.s
END
```

The START time is the data interval start and the INTERVAL is the data interval. These are equal and has the same format as the Start and Interval lines in the ctm time manager.

The BATCH_HARDCOPY block:

```
BATCH_HARDCOPY
BEGIN
    DATA plot | panel | all
    DEVICE file | printer
    COLOR bw | bw_diff | color
    ORIENT land | port
    FILE_DIR directory
    FILE_NAME filename
END
```

The BATCH_HARDCOPY block defines the hardcopy output. When $a \mid b \mid c$ is written the user has to write one of a, b or c. The DATA, DEVICE, COLOR, ORIENTATION are equal to the *Which Data*, *Which Device*, *Print as* and *Orientation* toggle columns in the Print Dialog respectively. The data printed (DATA) can be current plot (plot), current panel (panel) or all panels (all). The postscript file (DEVICE) can either be stored in a file (file) or sent directly to the printer (printer). The output (COLOR) can either be in black and white (bw), black and white with different line styles (bw_diff) or color (color). The orientation (ORIENT) can either be landscape (land) or portrait (port).

The (FILE_DIR) directory (directory) where the postscript file is created is specified. The (FILE_NAME) file name (filename) of the postscript file is specified.

If the postscript filename exits it will **not** be overwritten. (The user must remove existing postscript files before starting igr in batch mode if he wants to use existing file names).

The batch file can consist of several BATCH_TIME BATCH_HARDCOPY blocks.

The BATCH_EXIT block:

```
BATCH_EXIT
BEGIN
    SLEEP sec
END
```

The BATCH_EXIT has to be the last block in the batch file. The seconds after SLEEP defines how many seconds the igr client has to wait before it exits.

3.11.2 Useful hints about igr batch processing

- Create a configuration file manually to use.
- Time shift can not be combined with batch processing.
- Lock mode can not be combined with batch processing.
- The igr batch file must be provided by the user and for example written with a text editor or created by a user written C program.
- Postscript files are not overwritten.
- The block sequence in the batch file must be:

```
BATCH_CONFIG
BATCH_TIME
BATCH_HARDCOPY
BATCH_TIME
BATCH_HARDCOPY
BATCH_TIME
BATCH_HARDCOPY
. . . . . .
BATCH_EXIT
```

An example of a batch file igr_1.bat is given in next section. The postscript files igr_a.ps and igr_b.ps (if they exist) at directory /home/blybekk must be removed before starting igr igr_1.bat . The configuration file /home/blybekk/bat/default_1.igr must be a valid configuration file.

3.11.3 Igr batch file example

IgrBatchFile

```
# igr_1.bat
#
#
# created Aug. 14 95, by Bjoern Lybekk

BATCH_CONFIG
BEGIN
    CONFIG_DIR  /home/blybekk/bat
    CONFIG_FILE default_1.igr
END

# first time interval

BATCH_TIME
BEGIN
    START    1996-01-01 00:20:00.0
    INTERVAL 1200.0
END

BATCH_HARDCOPY
BEGIN
    DATA   all
    DEVICE file
    COLOR  bw
    ORIENT portrait
    FILE_DIR  /home/blybekk
    FILE_NAME igr_a.ps
END

# second time interval

BATCH_TIME
BEGIN
    START    1996-01-10 06:15:00.0
    INTERVAL 18000.0
END

BATCH_HARDCOPY
BEGIN
    DATA   all
    DEVICE file
    COLOR  color
    ORIENT land
    FILE_DIR  /home/blybekk
    FILE_NAME igr_b.ps
END
```

```
BATCH_EXIT
BEGIN
    SLEEP 10
END
```

4 Errors and probable causes

4.1 Problem reporting

Problems should be reported to *isdat@irfu.se*. See also URL *http://www.irfu.se/isdat/* or *http://www.irfu.se/isdat/wec/* for detailed instructions on error reporting.

4.2 Error messages

- Load configuration (File Config Load). Warning messages are written at the bottom text line of the *File Selection* window.

  These are:

  ```
  "filename" is a directory
  "filename" No such file or directory
  "filename" Permission denied
  "filename" not a configuration file
  ```

  The filename is the full path of the selected file.

- Save configuration

  Warning message is written at the bottom text line of the *File Selection* window.

  This is:

  ```
  "filename" Permission denied
  ```

- SaveData Flatfile has the error messages:

  ```
  "filename" is a directory
  "filename" Permission denied
  only one xy-plot is allowed in flat file
  ordinate and abscissa timelines differs
  time line in plot #1 and plot#2 differs

  (The '#1' and '#2' are numbers).
  ```

- Print has the error messages:

```
"filename" is a directory
cannot open "filename"
```

(The "filename" is the full path of the selected file.)

- The error messages from the data base handler are written into the relevant panels.

  Examples:

```
DbBAD_TIME        no data is available for the requested instrument
DbBAD_SENSOR      eg. asking for density when only efield available
DbBAD_CHANNEL     eg. asking for V12L when in tape mode 1
DbBAD_ALLOC       malloc() failure (out of memory)
DbBAD_MEMLIMIT    the maximum request size exceeded
DbBAD_INTERNAL    should not happen !
DbBAD_ZONE        requested interval is in between two samples
DbBAD_DROP        no data available due to missing packets
DbBAD_EOF         reached end-of-file before finding any data
                  for the requested instrument
```

4.3 Common user errors and probable causes

The most common errors are explained in [Ref. 6].

5 Customizing igr

The initial igr settings are specified by the list of parameters listed below. A user who wants to modify these settings is recommended to copy the relevant lines to his personal *.Xdefaults* or *.profile* file. For example some systems have an alternative default printer than *lp*. The lines:

```
Igr*printCommand: lp
Igr*printOption:  -d
```

should then be appropriately modified and copied to the user's *.profile*. Another useful modification is to modify the window size:

```
Igr.formWidth: 752
Igr.formHeight: 469

! can be set to smaller size when starting e.g. 600x400
! Igr.formWidth:  600
! Igr.formHeight: 400
```

Note also that the complete igr configuration can be saved by using *configuration files*, see sections 3.10.4 and 3.10.5.

Default igr environment:

```
!# generated file - do not edit
!# generated from: resources.res calc_gui.res iff_gui.res


! $Id: resources.res,v 1.3 1995/10/14 18:53:31 al Exp al $
!
! General resources
!

! workaround for a bug in motif 1.2.3 causing Cascade button to
! sometimes not popup its sub-menu, motif CR #8238 #8172
Igr*mappingDelay:     0

Igr*fontList: \
    *-b&h-lucidatypewriter-bold-r-normal-*-14-*-75-75-m-*-iso8859-1
Igr*background: grey70
Igr*foreground: black
Igr*keyboardFocusPolicy: pointer

Igr.formWidth: 752
Igr.formHeight: 469

! can be set to smaller size when starting e.g. 600x400
! Igr.formWidth:  600
! Igr.formHeight: 400

Igr.qtyMax: 120
Igr.plotMax: 120
Igr.rowMax: 10
Igr.columnMax: 3
Igr.curveMax:            4
Igr.colorMax: 32
Igr.undoTimeLevels: 8

! Special setup
! Not all systems support grey72 & grey77
Igr*PanelWidget*bottomShadowColor: grey70
Igr*PanelWidget*background: grey70
Igr*PanelRowColumnW*background:        grey70

Igr*igr_status*.background: grey70
Igr*MenuBar*.background: grey70

! Resources for the pushbutton "linked" to the plots
!  fontList:   font type for numbers
```

```
!   foreground: color for numbers

Igr*PushButton*.fontList: 6x13

Igr*PushButton*.foreground: black
Igr*PushButton*.selectColor: black
Igr*PushButton*.topShadowColor:         white

Igr*PushButton.data*.fontList: \
-adobe-courier-bold-r-normal--12-120-75-75-m-70-iso8859-1

! Position of the pushbuttons
!  0 : upper left corner (default)
! 1 : upper right corner
! 2 : lower right corner
! 3 : lower left  corner
Igr*pushButtonPosition: 1

! Specifies if plot numbers should be printed to the right of the pushbuttons
!  1: plot numbers is printed  (default)
! 0: plot numbers is not printed
Igr*pushButtonTextOn: 1


! Specifies the scatter plot:
! scatterCrossSize  is the size * 1e3 of the cross at screen
! scatterCrossSizeP is the size * 1e6 of the cross in postscript
!      scatterDotSizeP   is the size * 1e6 of the dot   in postscript (radius of circle)
Igr*scatterCrossSize:  2000
Igr*scatterCrossSizeP: 2000
Igr*scatterDotSizeP:   720

! Specifies the print command with options
! example: Igr*printCommand: lp
!          Igr*printOption:  -d
!          When nothing is specified in the "Printer name:"
!          line in the PrintDialog the printer command
!          will be: 'lp'
!          When a name of a printer e.g. 'myprinter' is
!          written in the "Printer name:" line the printer
!          command will be:  'lp -d myprinter'
!
Igr*printCommand: lp
Igr*printOption:  -d


! Fonts and colours for menues

Igr*CurrentQuantity*.background: grey70
```

```
Igr*debug_text*fontList: \
    *-b&h-lucidatypewriter-medium-r-normal-*-12-*-75-75-m-*-iso8859-1
```

6 Using *igr* for de-bugging

Via the *Help→Debuglevel→* menu entry there is a set of debug tools available. When used the debug information is printed into the *Debug Dialog window*. The user writes a number into the debug dialog. The help menu window (Help about Debug Level) shows the interpretation of these numbers. In order to turn the debug print off the user inputs a 0 (zero). The debug tools are primarily intended for the development team and are therefore not documented in detail here. There is, however, one facility that might be useful for the scientific user also in regular use:

The characteristics of the data object of current quantity or current plot can be printed in the Debug Dialog. This is avaiable from the *Help→Debug→CurrentQty* or *Help→Debug→CurrentPlot* respectively. The *current quantity* and *current plot* represent the data object before and after the Calculator respectively. The information defined in the data definition file (see [Ref. 9]) is printed.

7 Reference Documents

[1] WEC/ISDAT Coordinate Transformations. Technical Report CWD-OBSPM-DD-002, OBSPM, February 1995. Issue 1.0.

[2] The Joining of Cluster WEC Data. Technical Report CWD-OBSPM-TN-0001, Observatoire de Paris a Meudon, September 1994.

[3] CSDS-UI external interface control document. Technical Report DS-ESR-ID-0001, ESRIN, October 1994. Issue 1.1.

[4] CSDS-UI software user manual. Technical Report DS-ESR-SM-0001, ESRIN, August 1994.

[5] CSDS User Interface, ISDAT Installation Manual. Technical Report DS-IRF-IM-0001, IRF-U, October 1995. Issue 2.1.

[6] CSDS User Interface, ISDAT User Manual. Technical Report DS-IRF-UM-0001, IRF-U, October 1995. Issue 2.1.

[7] CSDS User Interface, ISDAT cuigr Client User Manual. Technical Report DS-IRF-UM-0003, IRF-U, October 1995. Issue 2.1.

[8] CSDS User Interface, ISDAT cuitm User Manual. Technical Report DS-IRF-UM-0004, IRF-U, October 1995. Issue 2.1.

[9] WECdata structure working group. Editor C. Harvey. The structure of the WEC/ISDAT data. Technical Report CWD-OBSPM-DD-001, OBSPM, March 1995.

[10] Joe Zender. ISDAT coordinate transformation module, software description. Technical report, ESTEC, May 1996.

8 Index

Subject	Section	Page
Acronyms	1.7	2
Arithmetic operations	3.9.2	32
Batch mode	3.11	46
Bad alloc	4.2	51
Bad channel	4.2	51
Bad drop	4.2	51
Bad eof	4.2	51
Bad interval	4.2	51
Bad mem limit	4.2	51
Bad sensor	4.2	51
Bad time	4.2	51
Bad zone	4.2	51
Bug reporting	4.1	50
Calculator	3.9.2	32
Colours	3.8.6	16
Complex quantities	3.9.1	30, 11
Conceptual instrument	3.6.2	8
Configuration		
- saving	3.10.4	44
- restoring	3.10.5	45
Continuous mode	3.8.20	26
Coordinate transformation		
- Cartesian to polar	3.9.1	30
- to DS	3.9.6	38
- to GEI	3.9.6	38
- to GEO	3.9.6	38
- to MAG	3.9.6	38
- to GSE	3.9.6	38
- to GSEQ	3.9.6	38
- to GSM	3.9.6	38
- to SM	3.9.6	38
- to DM	3.9.6	38
- to VDH	3.9.6	38
Current		
- panel	2.2.1	3
- panel status line	3.3	5, 7
- plot	2.2.3	4, 7
- plot status line	3.3	5
- quantity	2.2.2	4, 8
Customizing the environment	5	51

Subject	Section	Page
Debugging	6	54
Deleting		
- panels	3.8.15	22
- plots	3.8.13	21
- quantity	3.8.11	20
Error messages	4	50
Error reporting	4.1	50
Exit	3.4	6
Exporting data		
- as a flat file	3.10.1	39
- as a CDF file	3.10.2	41
FFT	3.9.5	36
- Power spectral density, PSD	3.9.5	37
- Real component	3.9.5	37
- imaginary component	3.9.5	37
- phase	3.9.5	37
- amplitude	3.9.5	37
Hard copy	3.10.3	42
Help	3.1	5
IGR version	1.3	1
Input parameters	3.6.2	8
Joining to common time line	3.9.4	36, 32
Linear axis	3.8.4	14
Locking a panel	3.8.19	24
Logarithmic axis	3.8.4	14
man pages	3.1	5
Page control dialog	9	27
Panel	2.2.1	3
- adding	3.8.14	22
- changing size of	3.8.16	22
- deleting	3.8.15	22
- grid in	3.8.5	14
- frame type	3.8.5	14
Printing		
- in batch mode	3.11	46
- interactively	3.10.3	42
- device	3.10.3	44
Plot	2.2.3	4
- add	3.8.12	21
- delete	3.8.13	21
- specify	3.8.12, 3.9.1	21, 28
Plot layout		
- predefined	3.10.6	45
- restoring from a file	3.10.5	45
- saving in a file	3.10.4	44

Subject	Section	Page
Plotting		
- colour coded spectra	3.9.5	38
- complex quantities	3.9.1	30, 11
- dynamic spectra	3.9.5	38
- in batch mode	3.11	46
- long time intervals	3.8.19, 3.8.20	24, 3.8.20
- multiple curves in one panel	3.8.7	16
- time series	3.7	10
- scalars	3.7.1	11
- spectra	3.7.5	11
- spectrograms	3.8.9	18, 11
- static spectra	3.8.8, 3.7.5, 3.9.5	18, 11, 38
- tensors	3.7.4	11, 33
- vectors	3.7.3	11
- vector components	3.9.1	30
- x vs. y	3.8.18	23
Postscript	3.10.3	42
Setting		
- axis type	3.8.4	14
- colours	3.8.6	16
- common scales	3.8.2	13
- continuous mode	3.8.20	26
- current plot	3.6.1	7
- data packing mode	3.6.5	10
- default calculator	3.9.2	34
- erasing calculator	3.9.2	34
- FFT parameters	3.9.5	36
- frame type	3.8.5	14
- lock mode	3.8.19	24
- max samples to return	3.6.4	10
- panel size	3.8.16	22
- parameters to plot	3.6.2	8, 16
- plot style	3.8.3	14
- print options	3.10.3	42
- quantity	3.8.10,2.2.2	20,4
- scales	3.8.1	11, 13
- spectrum colour scale	3.8.9	20
- time interval	3.5	6, 23
- time shift between plots	3.9.3	34
- units	3.6.3	8

Subject	Section	Page
Starting igr	3.2	5
Time shifts	3.9.3	34
Quantity	2.2.2	4. 16
- adding	3.8.10	20
- deleting	3.8.11	20
- time shifting	3.9.3	34
- version	3.10.3	44
Version	1.3, 3.10.3	1, 44
Zooming in time	3.8.17	23