IBJdis

Version 3.0

June 2007

Janicke Consulting
Environmental Physics
26427 Dunum GERMANY
Alter Postweg 21

Email: info@janicke.de
Internet: www.janicke.de

Contents

1 Data inspection with *IBJdis*

Title: IBJdis
Subject: The graphical user interface *IBJdis* (JAVA program) allows to inspect data files of format DMNA and ARR and to visualize the data in form of tables and graphics.
Date: 2007-05-29

1.1 Introduction

IBJdis is a graphical user interface for the inspection and visualization of files of type DMNA (file extension `.dmna`) and ARR (file extension `.arr`).[1] These files are created for example by the dispersion programs LASAT, AUSTAL2000, and LASPORT.[2]

IBJdis allows the user to read in the files, inspect the file header, select one- or two-dimensional subsets of the data part, and to inspect the data either in tabular or in graphical form.

In a tabular representation, the data is shown in a table with user-defined number formats and scaling factors. Two-dimensional data tables can be depicted as table graphics with optionally coloured background fields.

In a graphical display, a horizontal layer,[3] like the concentration distribution near ground, is shown as a coloured map with a variety of accessories such as colour legend, axis description, and background map. A one-dimensional data set like the time series of the hourly mean velocity of a meteorological time series can be depicted as a one-dimensional function plot. All graphics can be saved in form of PNG (compressed bitmap) or PDF (scalable vector graphics) files.

If information about the statistical uncertainty are provided in a concentration file, *IBJdis* can optionally perform a consistent[4] smoothing of the concentration field.

In a graphical visualization of a wind vector field defined on an Arakawa-C grid, *IBJdis* automatically converts from Arakawa-C components to mean components with respect to a grid cell.

The graphical user interface *IBJdis* is bilingual with the user being able to choose between English and German. Extensive help information for both languages is provided with the online help (button *[Help]* in the main window). All other documents similar to this one are provided in English.

IBJdis requires the auxiliary programs *Lasdap* and *Hglext* in subdirectory `IBJutils`.

IBJdis does not make any changes or entries to the system registry. For the graphical display, a temporary file is written to default temporary directory. This file is deleted when exiting *IBJdis*.

[1] ARR files are compressed binary files that were applied by LASAT before version 3.0.

[2] For further information see www.janicke.de.

[3] The graphically display of vertical layers is not yet implemented.

[4] Values are modified only within their statistical uncertainty.

1.2 Program call

1.2.1 Graphical user interface

IBJdis can be simply started in the Explorer window by a double click on the executable `IBJdis`.[5] `IBJdis` starts `IBJdis.jar` in subdirectory `jar` with increased memory (256 MB) using the local JRE in subdirectory `jre`. Alternatively, and if a JRE is installed, *IBJdis* can be called by a double click on `IBJdis.jar` in subdirectory `jar`.

After loading, the main window of *IBJdis* appears on the screen. With the popup menu (press the right mouse button when the mouse is on top of the window), the *Look & Feel* and the language can be selected. Help information is displayed by pressing the button *[Help]*.

The user interface is terminated by closing the window (press the cross in the upper right window corner). On exit, current user settings like language, window size, and settings for the graphical display are stored to file `.ibjdis` in the users home directory.

The licence information is first searched by *IBJdis* in the directory, where `IBJdis.jar` is located, then in the subdirectory `IBJutils` therein and finally in the parent directory.

1.2.2 Command line mode

IBJdis can be invoked also in command line mode, for example to automatically create and save two-dimensional graphics files:

```
jre\bin\java -jar jar\IBJdis.jar File Option
```

Possible values for *Option* are:

`-g`*Out* Create a graphic from the 2-dimensional input file *File* using the current settings and store it as graphic file *Out*.

This option is useful to create graphic files in a batch procedure. The graphics settings can be adjusted by a prior call of the graphical user interface and with the help of the options `-r`, `-p`, and `-m`.

`-m` Only if option `-g` is applied: display the label with the maximum value.

`-p`*Pal* Only if option `-g` is applied: use palette file *Pal*.

`-r`*Ref* Only if option `-g` is applied: use reference value *Ref*.

`-v` Verbose modus in which additional control informations are written to the DOS shell.

Example:

```
jre\bin\java -jar IBJdis.jar test\so2-j00a.dmna -gtest\so2-j00a.pdf
```

[5] Program `IBJdis.exe` for Windows and shell script `IBJdis` for Linux.

1.2.3 Program call with increased memory

By default, a JAVA application allocates up to 64 MB RAM memory. For large data files or background maps this may not be sufficient in which case a corresponding warning message appears.

In order to provide a larger amount of memory the program must be called with the JAVA option `-Xmx`*Number*`m`. Here, *Number* is the maximum allowed memory in MB, for example

```
jre\bin\java -Xmx256m -jar jar\IBJdis.jar
```

In this example, the user interface is allowed to allocate up to 256 MB memory. This setting and the local JRE in subdirectory `jre` is applied by the executable `IBJdis` which can be called to start the user interface.

1.3 Examples

The following sections contain some selected example applications for *IBJdis*. The example files were taken from the LASAT software package, but the general procedure can be easily adopted to other files.

1.3.1 Scaling of a background map

This section demonstrates how to scale a map so that it can be used later as a background map for the graphical visualization of a concentration field.

The scaling is achieved by specifying three reference points in the map and their corresponding metre coordinates. The program derives from this the transformation matrix that transforms pixel coordinates to metre coordinates, and stores this information together with the name of the graphic file to a LASAT definition file (by default `map.def`). This file, which is conveniently stored in the same directory as the map file, is then later selected for the graphical visualization.

1. Load *IBJdis* by a double click in the Explorer window on the executable `IBJdis`.

2. Move the mouse button on top of the main window of *IBJdis* and press the right mouse button. A popup menu appears.

3. Select menu *Background Map Scaling*. A new window appears.

4. Press the button *[Load Map]* and select the desired map file (JPEG, GIF, or PNG, where PNG is the preferred format), in this case file `plan100.gif` from the LASAT example directory `x\ibjshape`.

5. Press *[Point]* to specify the first reference point.

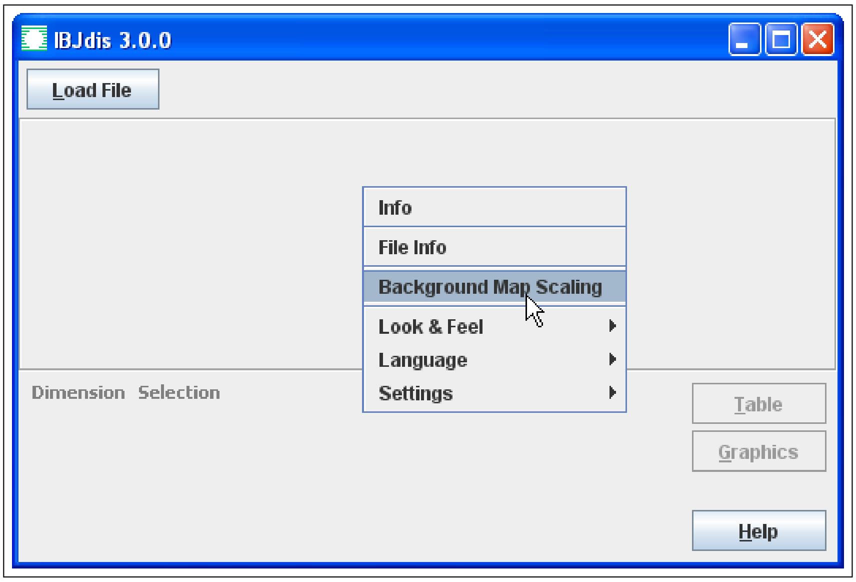


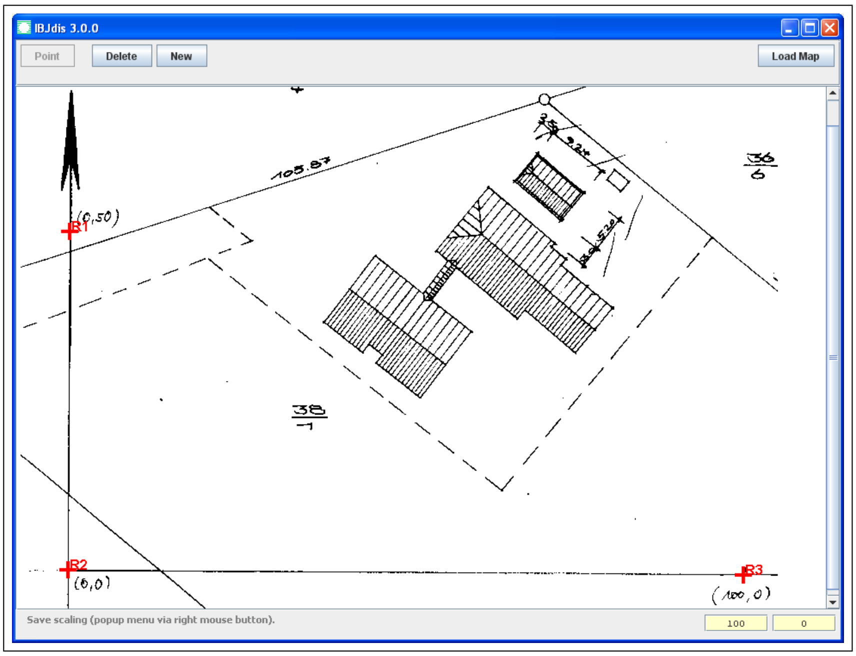
FIGURE 1: Main window of *IBJdis* with popup menu.

6. Move the mouse cursor on top of the reference point, here point (0,50) in the upper left corner of the map, and press the left mouse button. A red cross with label ”R1” is displayed at the chosen location.
7. In the two yellow fields in the bottom right corner of the window insert the x- and y-coordinate of the reference point (in metre with a point as decimal separator).[6]
8. Repeat the last three steps to define the two other reference points at (0,0) and (100,0).
9. Right click on the display window and select menu *Save Scaling...* to write out the scaling file. This stores the scaling information to file `map.def`, here to subdirectory `x\ibjshape`.

> *Note: The scaling file* `map.def` *contains the scaling informations and the name of the graphics file to which it belongs to. If the scaling file is saved to the same directory as the graphics, the name of the graphics file is stored without path information. This can be useful when copying scaling and graphics file to another directory.*

The graphics file is now prepared to be used as background map, as in the next example.

[6] In LASAT and AUSTAL2000 for example, all coordinate specifications are made relative to an overall reference point. The coordinate specifications made here must also be made relative to this point.



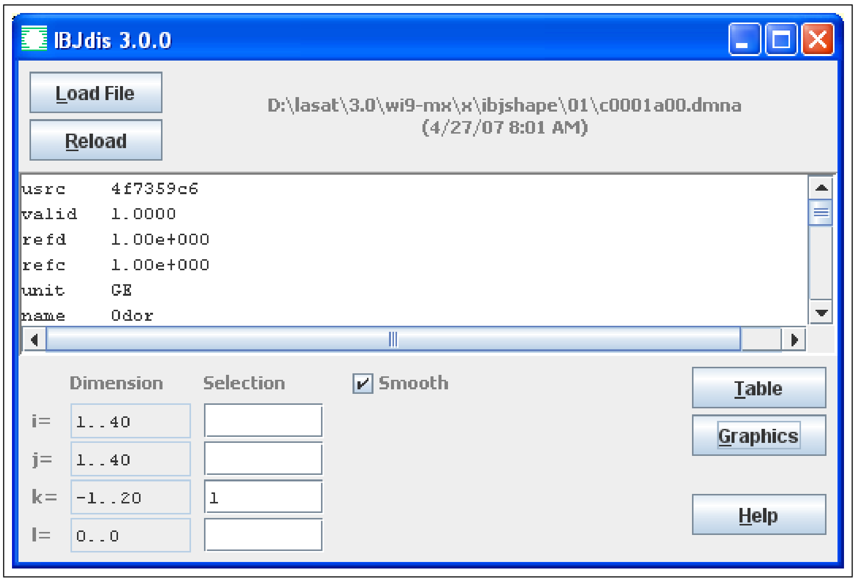
Save scaling (popup menu via right mouse button).

FIGURE 2: Map window after definition of the three reference points.

1.3.2 Graphical display of a horizontal distribution

This section demonstrates how to graphically visualize a horizontal concentration distribution.

Two-dimensional data fields can be visualized as tables or, in case of a horizontal data layer, in form of a map graphics. In this example a horizontal concentration distribution from the LASAT example `ibjshape` is graphically visualized.



D:\lasat\3.0\wi9-mx\x\ibjshape\01\c0001a00.dmna
(4/27/07 8:01 AM)

```
usrc     4f7359c6
valid    1.0000
refd     1.00e+000
refc     1.00e+000
unit     GE
name     Odor
```

	Dimension	Selection
i=	1..40	
j=	1..40	
k=	-1..20	1
l=	0..0	

☑ Smooth

FIGURE 3: Main windows of *IBJdis* after loading the LASAT concentration file.

1. Load *IBJdis*.
2. Load with button *[Load File]* the file `c0001a00.dmna` from the directory `x\ibjshape\01`.

   If the file does not exist, load it from the result directory `rsl` on the LASAT CD and then continue this document.[7]

   After successful loading of the file, its name and date are shown to the right of the button. The buttons *[Table]* and *[Graphics]* are enabled.

   > *Note: A file may also be loaded by dragging the file name from the Explorer window to the main window of IBJdis.*

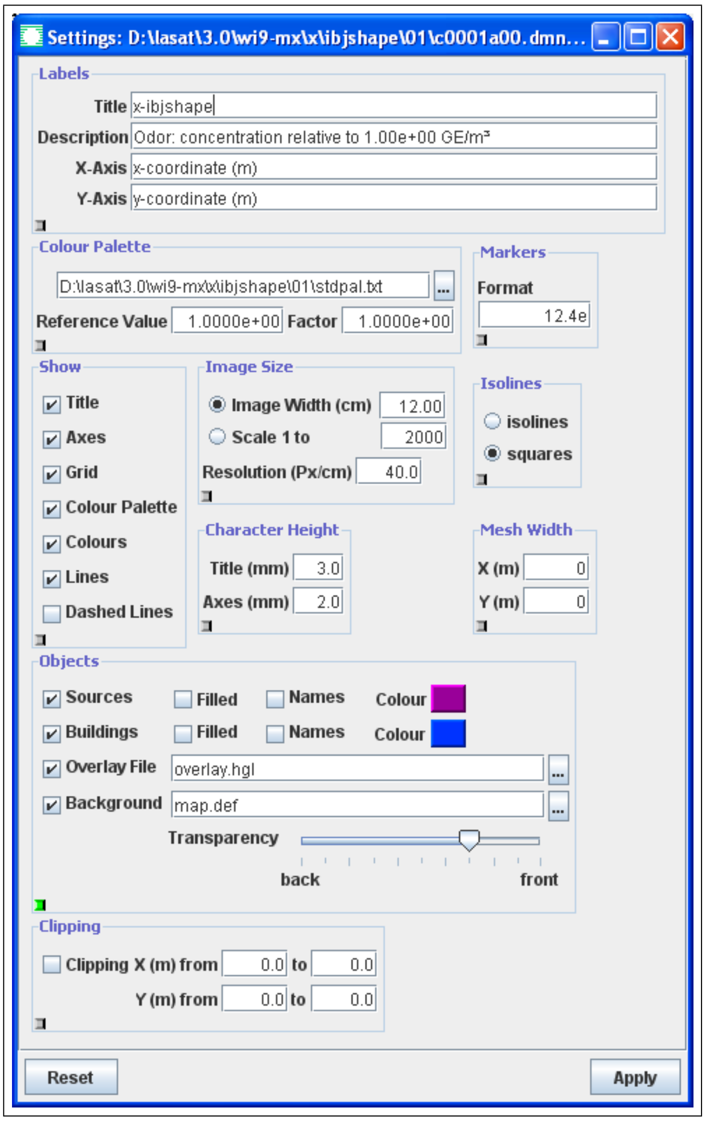
   The centre text panel lists the file header, the text fields below show the indices and ranges of the data part. If the data part refers to a concentration file, i is the index for the grid cells in x-direction, j for the y-direction, k for the vertical (starting with index 1; $k = 0$ refers to dry, $k = -1$ to wet deposition), and l is the index of trace substances (starting with 0).
3. Select index $k = 1$ (type in a ”1” in the k-line) which is the horizontal layer near ground.

[7] Alternatively, run the LASAT example as outlined in the batch procedure `x\runall`, e.g. with the call `x\runall ibjshape`.

4. Check the box ”*Smooth*” to smooth the concentration results (only available for LASAT concentration files).

5. Press the button *[Graphics]*. The horizontal concentration distribution for the trace substance `Odor` is displayed as map graphics in a new window. The colour applied to each cell refers to the concentration value according to the palette shown at the bottom. The palette is defined in file `stdpal.txt` in directory `x\ibjshape\01`.

> *Note: The palette colours and value ranges can be stored in a text file. By default, IBJdis searches for the palette file* `stdpal.txt` *first in the directory of the displayed file, then in the home directory of IBJdis and finally in its subdirectory IBJutils.*



Settings: D:\lasat\3.0\wi9-mx\x\ibjshape\01\c0001a00.dmn...

Labels
Title: x-ibjshape
Description: Odor: concentration relative to 1.00e+00 GE/m³
X-Axis: x-coordinate (m)
Y-Axis: y-coordinate (m)

Colour Palette
D:\lasat\3.0\wi9-mx\x\ibjshape\01\stdpal.txt ...
Reference Value: 1.0000e+00 Factor: 1.0000e+00

Markers
Format: 12.4e

Show
☑ Title
☑ Axes
☑ Grid
☑ Colour Palette
☑ Colours
☑ Lines
☐ Dashed Lines

Image Size
● Image Width (cm): 12.00
○ Scale 1 to: 2000
Resolution (Px/cm): 40.0

Isolines
○ isolines
● squares

Character Height
Title (mm): 3.0
Axes (mm): 2.0

Mesh Width
X (m): 0
Y (m): 0

Objects
☑ Sources ☐ Filled ☐ Names Colour 🟪
☑ Buildings ☐ Filled ☐ Names Colour 🟦
☑ Overlay File: overlay.hgl ...
☑ Background: map.def ...
Transparency: back – front

Clipping
☐ Clipping X (m) from 0.0 to 0.0
Y (m) from 0.0 to 0.0

FIGURE 4: Setting window.

6. Select the popup menu *Settings*. A new window appears which allows the user to change a variety of setting options. The settings are grouped in blocks. If the small rectangle in the lower left corner of a block is changed by a mouse click

from grey to green, the current settings for this block are saved on exit of the program and restored in the next program call.

7. in the block *Objects*, check box *Background* and select with button *[...]* to the right side of the text field the scaling file `map.def` that was created in the previous section to display a background map.

   The map file exists also in the current directory so that it would be sufficient to insert `map.def` into the text field.

8. Check the boxes *Sources* and *Buildings* to draw the source and building outlines onto the graphic.

9. Check the box *Overlay File* and load file `overlay.hgl` that contains an additional description in form of HPGL commands. Again, load the file either by a direct specification of the file name in the text field or by clicking button *[...]* to the right side of the text field.

10. Press *[Apply]*. The graphic is displayed with the current settings.

11. To display the position and value of the maximum concentration, select in the popup menu of the graphics window *Find Maximum*.

12. Press at any location inside the concentration distribution the left mouse button: the concentration value at this location is displayed while the mouse button is pressed down. Double click the left mouse button to permanently display the value label.

13. Select popup menu *Save...* and save the graphics as PDF file `test.pdf` (not possible in demo mode).

1.3.3 Tabular display of a two-dimensional data array

This section demonstrates how to visualize a two-dimensional data array in tabular form.

In this example a tabular display of the horizontal concentration distribution from the LASAT example `ibjshape` is created.

1. Load file `c0001a00.dmna` from directory `x\ibjshape\01`, enable *Smooth*, set the index for k to `1` (ground layer) and for both i and j to `20..40` (x-interval from 60 m to 100 m, y-interval from 40 m to 80 m).[8]

2. Press button *[Table]*. A new window appears in which the format for the elements of the data record (here concentration and statistical uncertainty) are defined.

[8] The calculation area starts in this example in x-direction at 20 m and y-direction at 0 m with a mesh width of 2 m.

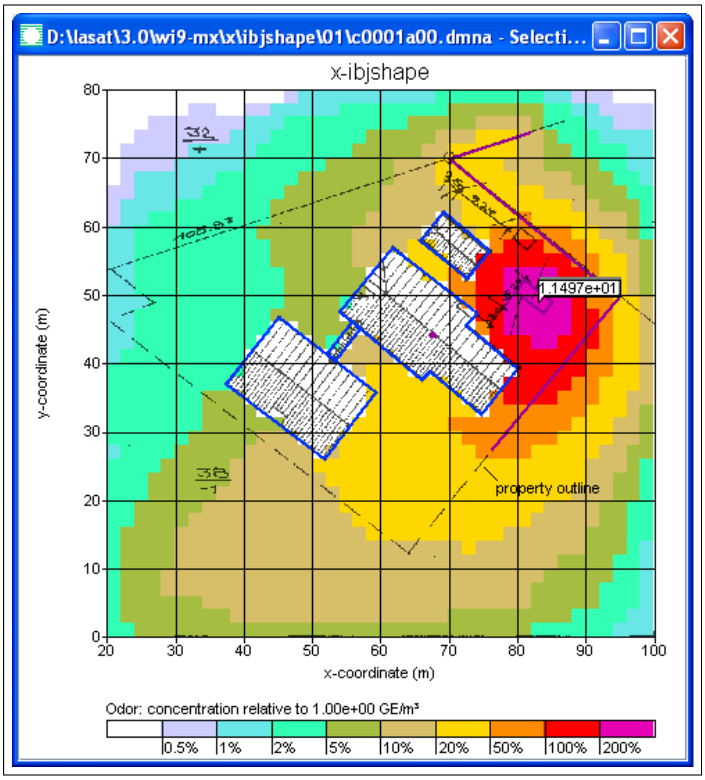


FIGURE 5: Graphics window with the horizontal concentration distribution.

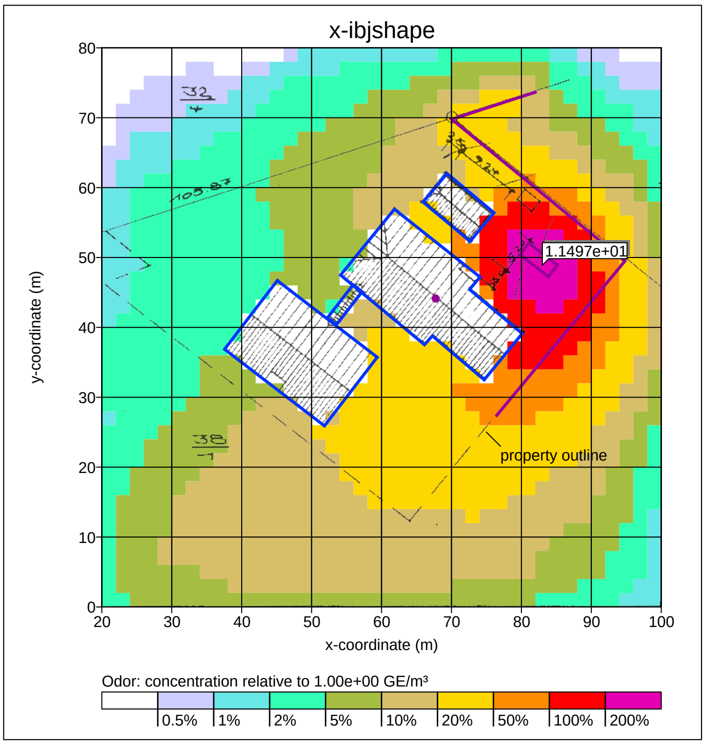
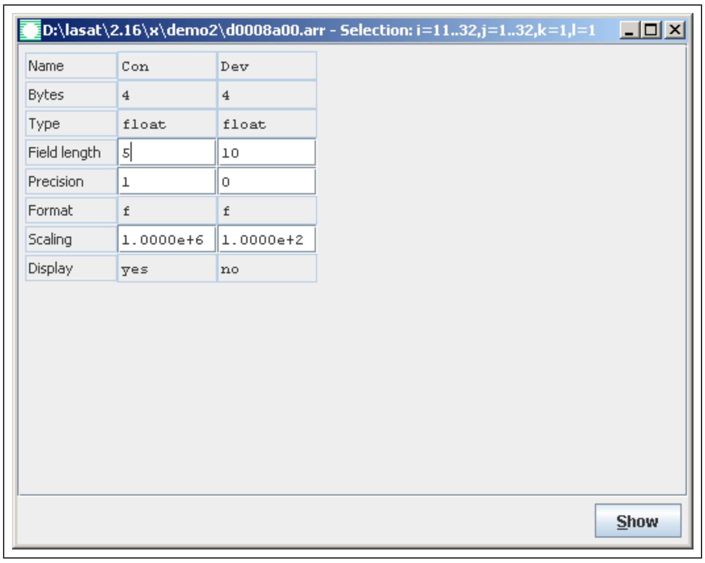


FIGURE 6: The graphics saved as PDF file and inserted into this document without modification.

3. If a two-dimensional data record has more than one element, the last line allows the user to select which element to display in the table. The entries ”yes” and ”no” are altered by a double click on the entry. Leave the ”yes” for the concentration unchanged.

4. Set the format to ”f” (float) by a double click on ”e” (exponential, i.e. scientific



D:\lasat\2.16\x\demo2\d0008a00.arr - Selection: i=11..32,j=1..32,k=1,l=1

Name	Con	Dev
Bytes	4	4
Type	float	float
Field length	5	10
Precision	1	0
Format	f	f
Scaling	1.0000e+6	1.0000e+2
Display	yes	no

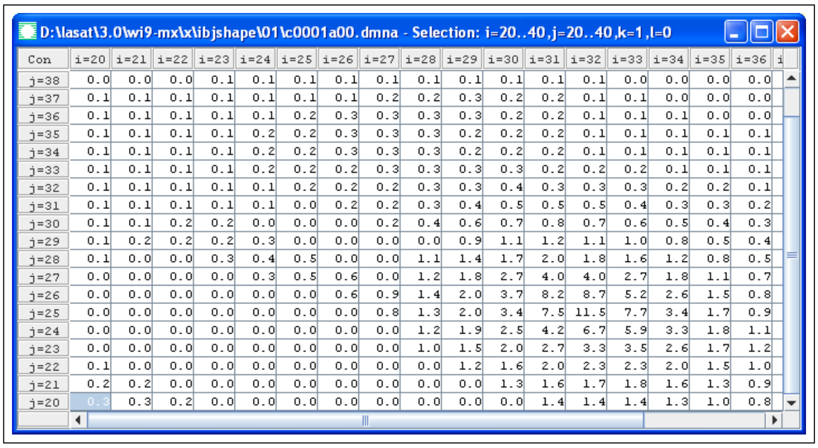
FIGURE 7: Format specification for the tabular display.

notation), set the field length to 5 and the precision to 1.

5. Press *[Show]*. The format table is replaced by the data table (with the popup menu *Back* the format table is shown again). One can scroll through the data table with the scroll bars at the bottom and right side.

6. Select the popup menu *Draw...*. A map table is shown with coloured cell backgrounds according to the cell values (with the popup menu *Back* the data table is shown again). The palette can be modified with the popup menu.

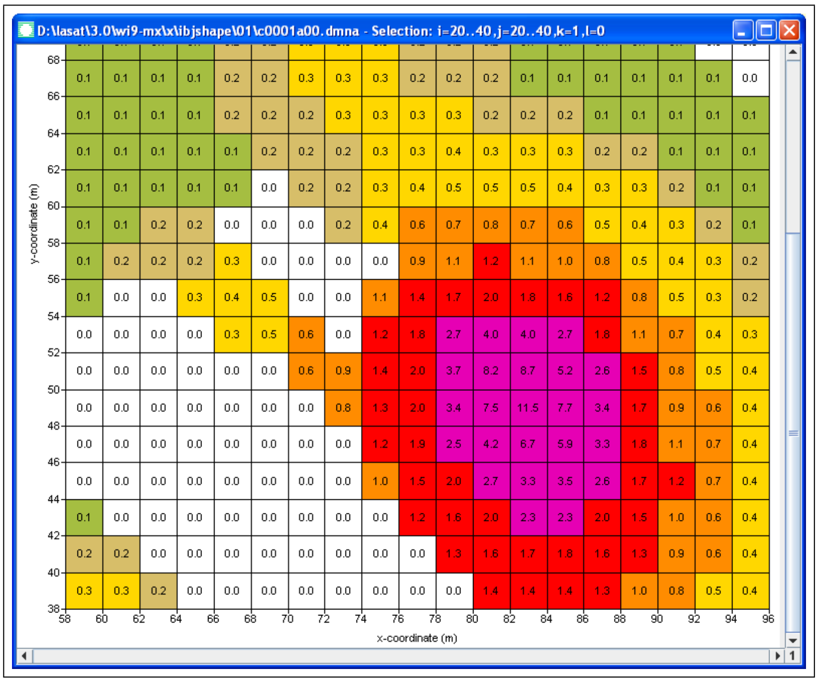
7. Use popup menu *Save image...* to save the map table as PDF file.



D:\lasat\3.0\wi9-mx\x\ibjshape\01\c0001a00.dmna - Selection: i=20..40,j=20..40,k=1,l=0

Con	i=20	i=21	i=22	i=23	i=24	i=25	i=26	i=27	i=28	i=29	i=30	i=31	i=32	i=33	i=34	i=35	i=36	i
j=38	0.0	0.0	0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0	0.0	0.0	0.0	
j=37	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.2	0.2	0.3	0.2	0.2	0.1	0.1	0.0	0.0	0.0	
j=36	0.1	0.1	0.1	0.1	0.1	0.2	0.3	0.3	0.3	0.3	0.2	0.2	0.1	0.1	0.1	0.0	0.0	
j=35	0.1	0.1	0.1	0.1	0.2	0.2	0.3	0.3	0.3	0.2	0.2	0.2	0.1	0.1	0.1	0.1	0.1	
j=34	0.1	0.1	0.1	0.1	0.2	0.2	0.3	0.3	0.3	0.2	0.2	0.2	0.1	0.1	0.1	0.1	0.1	
j=33	0.1	0.1	0.1	0.1	0.2	0.2	0.2	0.3	0.3	0.3	0.3	0.2	0.2	0.2	0.1	0.1	0.1	
j=32	0.1	0.1	0.1	0.1	0.1	0.2	0.2	0.2	0.3	0.3	0.4	0.3	0.3	0.3	0.2	0.2	0.1	
j=31	0.1	0.1	0.1	0.1	0.1	0.0	0.2	0.2	0.3	0.4	0.5	0.5	0.5	0.4	0.3	0.3	0.2	
j=30	0.1	0.1	0.2	0.2	0.0	0.0	0.0	0.2	0.4	0.6	0.7	0.8	0.7	0.6	0.5	0.4	0.3	
j=29	0.1	0.2	0.2	0.2	0.3	0.0	0.0	0.0	0.0	0.9	1.1	1.2	1.1	1.0	0.8	0.5	0.4	
j=28	0.1	0.0	0.0	0.3	0.4	0.5	0.0	0.0	1.1	1.4	1.7	2.0	1.8	1.6	1.2	0.8	0.5	
j=27	0.0	0.0	0.0	0.0	0.3	0.5	0.6	0.0	1.2	1.8	2.7	4.0	4.0	2.7	1.8	1.1	0.7	
j=26	0.0	0.0	0.0	0.0	0.0	0.0	0.6	0.9	1.4	2.0	3.7	8.2	8.7	5.2	2.6	1.5	0.8	
j=25	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.8	1.3	2.0	3.4	7.5	11.5	7.7	3.4	1.7	0.9	
j=24	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.2	1.9	2.5	4.2	6.7	5.9	3.3	1.8	1.1	
j=23	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.5	2.0	2.7	3.3	3.5	2.6	1.7	1.2	
j=22	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.2	1.6	2.0	2.3	2.3	2.0	1.5	1.0	
j=21	0.2	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.3	1.6	1.7	1.8	1.6	1.3	0.9	
j=20	0.3	0.3	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.4	1.4	1.4	1.3	1.0	0.8	

FIGURE 8: Data table.



D:\lasat\3.0\wi9-mx\x\ibjshape\01\c0001a00.dmna - Selection: i=20..40,j=20..40,k=1,l=0

0.1	0.1	0.1	0.1	0.2	0.2	0.3	0.3	0.3	0.2	0.2	0.2	0.1	0.1	0.1	0.1	0.1	0.1	0.0
0.1	0.1	0.1	0.1	0.2	0.2	0.2	0.3	0.3	0.3	0.3	0.2	0.2	0.2	0.1	0.1	0.1	0.1	0.1
0.1	0.1	0.1	0.1	0.1	0.2	0.2	0.2	0.3	0.3	0.4	0.3	0.3	0.3	0.2	0.2	0.1	0.1	0.1
0.1	0.1	0.1	0.1	0.1	0.0	0.2	0.2	0.3	0.4	0.5	0.5	0.5	0.4	0.3	0.3	0.2	0.1	0.1
0.1	0.1	0.2	0.2	0.0	0.0	0.0	0.2	0.4	0.6	0.7	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
0.1	0.2	0.2	0.2	0.3	0.0	0.0	0.0	0.0	0.9	1.1	1.2	1.1	1.0	0.8	0.5	0.4	0.3	0.2
0.1	0.0	0.0	0.3	0.4	0.5	0.0	0.0	1.1	1.4	1.7	2.0	1.8	1.6	1.2	0.8	0.5	0.3	0.2
0.0	0.0	0.0	0.0	0.3	0.5	0.6	0.0	1.2	1.8	2.7	4.0	4.0	2.7	1.8	1.1	0.7	0.4	0.3
0.0	0.0	0.0	0.0	0.0	0.0	0.6	0.9	1.4	2.0	3.7	8.2	8.7	5.2	2.6	1.5	0.8	0.5	0.4
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.8	1.3	2.0	3.4	7.5	11.5	7.7	3.4	1.7	0.9	0.6	0.4
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.2	1.9	2.5	4.2	6.7	5.9	3.3	1.8	1.1	0.7	0.4
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.5	2.0	2.7	3.3	3.5	2.6	1.7	1.2	0.7	0.4
0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.2	1.6	2.0	2.3	2.3	2.0	1.5	1.0	0.6	0.4
0.2	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.3	1.6	1.7	1.8	1.6	1.3	0.9	0.6	0.4
0.3	0.3	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.4	1.4	1.4	1.3	1.0	0.8	0.5	0.4

FIGURE 9: Map table.

A DMN format

Files with extension `.dmna` are plain text files with a file header followed by a data part. For binary or compressed output, the data part is written to a separate file with the same name but extension `.dmnt` (text uncompressed), `.dmnb` (binary uncompressed), `.dmnt.gz` (text compressed), or `.dmnb.gz` (binary compressed).

> *Note: Compressed files can be easily uncompressed with standard programs. To be used in uncompressed form, parameter* `cmpr` *in the file header must be set to 0.*

In a binary file, the table elements are written in their internal order, consecutively without control characters. A table element can be a single number (element) or a sequence of numbers (record).

In every line of the file header one parameter is defined. The name of the parameter appears at the beginning of the line followed by one or several values. Allowed separators are a blank, tabulator or semicolon that can be used separately or in combined form. The line is terminated by LF or CR+LF.

Beside the parameters needed by the program, the file header may contain additional parameters. Parameters not recognized by the program are ignored. The file header ends with a line starting with an asterisk ("*"). The next line is the start of the data part. The formatted table elements in the data part are separated by blanks, tabulators, or semicolons. The data part terminates with a line starting with three asterisks.

The following parameters in the file header are recognized and analysed by the program (the names must be written in lower case):

`cmpr` *integer*(0)
Compression level for the data part (between 0 and 9, default is 0 meaning uncompressed). If larger 0, the data part is stored as a separate file and compressed via GNU zip with the given compression level (file extension "`.dmnt.gz`" for text mode and "`.dmnb.gz`" for binary mode).

`data` *string*(1)
name of the file with the actual data table. If `data` is not specified or if it has the value "*", the data table is written in case of formatted (text) output to the same file as the file header. For unformatted (binary) output, the file name of the header is used but with the extension "`.dmnb`" instead of "`.dmna`". If `data` contains a path specification, it is interpreted as relative to the directory that contains the file header.

`dims` *integer*(1)
Number of dimensions (maximum 5).

`fact` *float*(1)
Factor with which all data elements of type float or double are multiplied before formatted output. When reading in formatted data, the elements are divided by `fact`. The factor acts only on those data elements for which no factor is specified in the format string (see `form`).

`form` *string*(1)

Format used to save data elements in formatted output. If the elements of a table are records that consist of several data elements, a format must be specified for each data element and `form` is the sum of all formats.

Format = *Format*$_1$ *Format*$_2$...
Format$_i$ = *Name*`%(`**Factor*`)`*Length*`.`*PrecisionSpecifier*

Meaning:

Name	name of the data element (optional)
Factor	scaling factor (optional including brackets)
Length	length of the data field
Precision	number of decimal positions (for floating point numbers)
Specifier	type specifier

The scaling factor *Factor* is handled like the parameter `fact`. The specification *Length* is the minimum length of the data field and may be exceeded if necessary for the display of the data field. The numbers are separated by at least one separator character.

The following type specifiers are recognized:

Spec.	Type	Byte Length	Description
`c`	*character*	1	single character
`d`	*integer*	4	decimal number
`x`	*integer*	4	hexadecimal number
`f`	*float*	4	floating point number (without exponent)
`e`	*float*	4	floating point number (with exponent)
`t`	*integer*	4	time specification (without date)

The specifiers `f` and `e` may be preceded by a `l` (double of length 8 bytes). The specifiers `d` and `x` may be proceeded by a `h` (short integer of length 2 bytes).

Time format for binary output: The time specification without date is the number of passed seconds. If the specifier `t` is proceeded by a `l`, the number (double of length 8 bytes) is interpreted as time specification with date: the positions to the left side of the decimal separator represent the number of days passed since 1899-12-30.00:00:00 plus 10^6, the decimal positions represent the fraction of seconds passed at the specified day. Time format for text output: the time is specified in the form `dd.hh:mm:ss` or `hh:mm:ss`, with `lt` in the form `yyyy-mm-dd.hh:mm:ss`.

Similar format specifications can be merged:
`vx%5.2fvy%5.2fvz%5.2f` is equivalent to `vx%[3]5.2f`[9]

`hghb` *integer*(`dims`)

High bound for the different indices.

`lowb` *integer*(`dims`)

Low bound for the different indices.

[9] For merged formats, the specified *Name* applies to the first element only. For the following elements, the last character of *Name* is increased by one alphabetical position.

`mode` *string*(1)
Output mode: `binary` or `text`.

`sequ` *string*(1)
Index sequence for the data output. Usually, the fastest index is the one at the far-right side (C convention). For a three-dimensional field A_{ijk}, this corresponds to the specification `i+,j+,k+`. FORTRAN saves data according to `k+,j+,i+`. A minus sign instead of a plus sign denotes an index running backwards.

Example: For a formatted two-dimensional data table (e.g. terrain heights) with the values oriented like in a map (north at the top etc.), `sequ` must have the value `j-,i+`.

Subranges can be selected as well:
`j=10..1/1,i=5..25/1,k=1`. The optional setting $/n$ specifies the starting value of the corresponding index. If a subrange is selected with `sequ`, the index boundaries `lowb` and `hghb` still refer to the original index definitions.

`size` *integer*(1)
Record size of the data in bytes. For formatted output the sum of record sizes resulting from the format specification must be equal to `size`.

If strings contain blanks they must be enclosed in quotation marks, otherwise quotation marks are optional.

Example:

A field with floating point numbers $A_{ijk} = 100i + 10j + k$, $i = 1..3$, $j = 2..4$, $k = 0..1$ is saved in form of horizontal layers:

```
form  "%4.1f"
mode  "text"
sequ  "k+,j-,i+"
fact  1.000e-001
dims  3
size  4
lowb  1 2 0
hghb  3 4 1
*
  14.0  24.0  34.0
  13.0  23.0  33.0
  12.0  22.0  32.0

  14.1  24.1  34.1
  13.1  23.1  33.1
  12.1  22.1  32.1

***
```

Beside the system header, the following parameters are evaluated by IBJdis depending on the type of data (see as well the IBJdis online help):

`artp` or `arrtype` *string*(1)
: Array type used to derive the palette description (if `mark` is not defined) and to derive other settings.

`axes` *string*(1)
: Axes description. It should be set for example to `xyz` if the data part contains a three-dimensional spatial grid.

`delt` or `delta` *float*(1)
: Mesh width of the data grid in metre.

`gakrx` *float*(1)
: X-coordinate of the reference point to which all coordinate specifications refer to.

`gakry` *float*(1)
: Y-coordinate of the reference point to which all coordinate specifications refer to.

`idnt` *string*(1)
: Graphics title.

`mark` *string*(1)
: Palette description.

`valid` *float*(1)
: A number between 0 and 1 that indicates the valid time fraction (in case of measurement gaps) if the data part contains time-averaged quantities.

`vldf` or `valdef` *string*(1)
: One character per element of the data record specifying whether the values of the element are point values ('P'), volume means ('V'), or area means[10] ('X', 'Y', 'Z', 'S').

`xmin` *float*(1)
: X-coordinate of the left (western) border of the data grid relative to `gakrx` in metre.

`ymin` *float*(1)
: Y-coordinate of the bottom (southern) border of the data grid relative to `gakry` in metre.

[10] Used for wind fields defined on an Arakawa-C grid.