Apparent to true dip restoration: user's manual for App2truedip versions $\geqslant$ 5.0

—

Version 3.0

Bernard Célérier

2025-04-26

Contents

1 Purpose and methods **4**
- 1.1 Purpose 4
- 1.2 Structural measurements on ODP cores 4
- 1.3 Methods 5
- 1.4 Using App2truedip 6
  - 1.4.1 Input and Output 6
  - 1.4.2 Input 7
  - 1.4.3 Running the program 7
  - 1.4.4 Output 7
- 1.5 About App2truedip 7

2 Input files **8**
- 2.1 Introduction 8
- 2.2 Common conventions for both types of input files 8
- 2.3 Alphanumerical input files 9
  - 2.3.1 Alphanumerical data line case 1 9
  - 2.3.2 Alphanumerical data line case 2 9
- 2.4 Numerical input files 10
- 2.5 Input examples 11
  - 2.5.1 Input data examples 11
  - 2.5.2 Input files samples 11

3 Output files **13**
- 3.1 Introduction 13
- 3.2 Common characteristics of both types of output files 13
- 3.3 Alphanumerical output files 14
  - 3.3.1 Alphanumerical data line case 1 14
  - 3.3.2 Alphanumerical data line case 2 14
- 3.4 Numerical output files 14
- 3.5 Output examples 15

4 Fortran sequential files **16**
- 4.1 Introduction 16
- 4.2 Fortran input/output files 16

4.2.1 FORTRAN free format . . . 16
4.2.2 FORTRAN fixed format . . . 17
4.3 Operating system issues . . . 17
4.3.1 Encoding . . . 17
4.3.2 End of line . . . 17
4.3.3 End of file . . . 18
4.4 Creating input files . . . 18
4.4.1 Text editors . . . 18
4.4.2 Spreadsheets . . . 19
4.4.3 Word processors . . . 19
4.5 Standard input/output files . . . 19
4.5.1 Standard file structure . . . 19
4.5.2 Standard header . . . 19
4.5.3 Standard data line . . . 20
4.5.4 Standard file example . . . 20

5 How to refer to App2truedip and relevant references 21
5.1 How to refer to APP2TRUEDIP . . . 21
5.2 Other relevant references . . . 21

6 App2truedip revision history 22

List of Tables

1.1 Symbols and acronyms used in this document . . . 6

2.1 Alphanumerical input data line case 1 . . . 9
2.2 Alphanumerical input data line case 2 . . . 10
2.3 Numerical input data line . . . 10
2.4 Examples of alphanumerical input data lines . . . 11
2.5 Examples of numerical input data lines . . . 12

3.1 Alphanumerical output data line case 1 . . . 14
3.2 Alphanumerical output data line case 2 . . . 15
3.3 Numerical output data line . . . 15

4.1 End of line coding . . . 18

6.1 App2truedip program versions 22

List of Figures

1.1 ODP core structural measurements 5

Chapter 1

Purpose and methods

1.1 Purpose

App2truedip computes strike and dip of planar structures measured as two apparent dips along two azimuths. Originally designed for structural measurements on cores from the Ocean Drilling Program (ODP, see Table 1.1 for symbols and acronyms used in this document), it can be used in any situation where two apparent dips are available. It is designed for simplicity: it just makes a few queries and returns restored strike and dip. A more complex software, App2truedipG, derived from App2truedip allows to generate graphical representations of the restoration results.

1.2 Structural measurements on ODP cores

ODP cores are longitudinally split into working and archive halves. This results in two directions along which apparent dips of planar structures can easily be measured: one parallel and the other perpendicular to the cut surface (Figs. 1.1-A and 1.1-B) (*Shipboard Scientific Party*, 2003).

Leg 131 structural geologists defined a conventional local frame of reference attached to these directions (*Shipboard Scientific Party*, 1991a); this frame was also used during leg 134 (*Shipboard Scientific Party*, 1992a).

During leg 135 this frame was rotated by 180° so as to be identical to that used in paleomagnetism (*Shipboard Scientific Party*, 1992b); this new frame has been used henceforth, for examples during legs 140, 147, 153, 176, 180, and 206 (*Shipboard Scientific Party*, 1992c, 1993a, 1995, 1999a,b, 2003), so that the usual convention is (Fig. 1.1-A):

- the archive half rounded surface faces South (180°) and
- when facing the upright plane cut surface with the top of the core upward, East (90°) is to the left and West (270°) to the right.

A direct strike measurement can also be made when a structure intersects an horizontal section of the core (Fig. 1.1-C).

True dip and dip direction can thus be computed either from two apparent dip measurements along two directions, or from one apparent dip measurement along one direction coupled with a strike measurement.

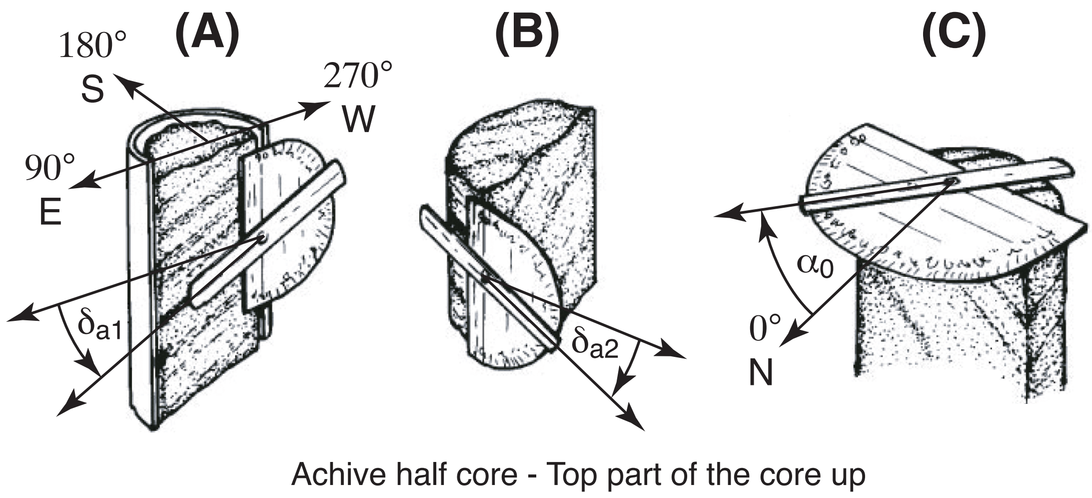

Figure 1.1: ODP core structural measurements on an archive half core piece. The top part of the core is oriented upwards. (A) Conventional reference frame and apparent dip, δ_{a1}, measurement in the East - West vertical plane; (B) Apparent dip, δ_{a2}, measurement in the North - South vertical plane; (S) Strike, α_0, measurement on an horizontal cut plane. Modified after *Shipboard Scientific Party* (1991a, 2003).

1.3 Methods

Given a 1st apparent dip, δ_{a1}, along direction, β_{a1}, and a 2nd apparent dip, δ_{a2}, along direction, β_{a2}, true dip, δ_0, and true dip direction, β_0, are the two unknowns of the system of two equations:

$$tan(\delta_{a1}) = cos(\beta_0 - \beta_{a1})tan(\delta_0) \quad (1.1)$$

$$tan(\delta_{a2}) = cos(\beta_0 - \beta_{a2})tan(\delta_0) \quad (1.2)$$

The algorithm implemented in App2truedip proceeds in three steps:

1. it first validates input;
2. it then checks whether the system of equations (1.1) and (1.2) falls in the case that yields a single solution;
3. if both above verifications are successfull, it finally computes the single solution (δ_0, β_0).

If input validation fails or if the equation system does not yield a single solution, an explanation is given in the output and a message is written in the terminal.

Two particular cases require special treatment: horizontal and vertical planes. Even though there is no need to resort to a program to compute true dip in theses cases, they may occur in a structural database. App2truedip deals with these particular cases, so that it is not necessary to remove them from input files before processing.

Table 1.1: Symbols and acronyms used in this document

Symbol	Comments
DSDP	Deep Sea Drilling Project (1966-1983)
ODP	Ocean Drilling Program (1985-2003); this acronym is used sensu lato in this document to refer to DSDP, ODP and IODP indistinctly
IODP	Integated Ocean Drilling Program (2003-2013) and International Ocean Discovery Program (2013-2024)
N	North
E	East
S	South
W	West
α_0	Strike
β_0	True Dip direction
δ_0	True Dip
β_{a1}	Apparent dip direction 1
δ_{a1}	Apparent dip 1 along direction 1
β_{a2}	Apparent dip direction 2
δ_{a2}	Apparent dip 1 along direction 2

1.4 Using App2truedip

1.4.1 Input and Output

Input can be read either from the keyboard, or from a text file. Similarly output can be directed either to the screen terminal, or to a text file.

The simplest way to run the program when a couple of data need be processed is to choose keyboard input and screen output. Each output result is then displayed after each input is entered. Alternatively, output can be directed to a file to keep a record.

Input and output files become more appropriate as the number of data increases. In that case, the program is designed with the idea of easily creating the input file by exporting the relevant data from a structural spreadsheet and of generating an output file that can also easily be imported back in the spreadsheet.

1.4.2 Input

Two types of input can be used:

- the alphanumerical input file type designed to accommodate apparent dip measurements made along North, East, South and West directions only, and recorded as N, E, S and W in structural spreadsheets;
- alternatively, the numerical input file type is designed to accommodate apparent dip measurements along any direction recorded as a number in the structural spreadsheet.

They are described in detail in Chapter 2.

1.4.3 Running the program

The program is designed to run with minimal interaction. It first issues a few queries, then makes the calculations, writes the results, and exits. It thus processes only one file at a time and need to be launched again if another file need to be processed.

1.4.4 Output

Output is tab delimited characters and numbers so as to facilitate importation into a spreadsheet. There are two different output types that correspond to the two input types and are described in detail in Chapter 3.

1.5 About App2truedip

Versions history of App2truedip is summarized in Chapter 6 and a few relevant references can be found in Chapter 5.

Chapter 2

Input files

2.1 Introduction

App2truedip accepts two types of structural measurements input files:

1. **alphanumerical input files** that were designed for core structural measurements made in ODP legs where apparent dip directions were measured along North, East, South, and West conventional directions and labeled by the characters E, W, N, S;
2. **numerical input files** that are made for measurements of apparent dips along any azimuth and where the azimuth is thus given as a number.

This chapter describes the formats of these two data files.

2.2 Common conventions for both types of input files

The input files are Fortran sequential input files that are read in Fortran free format. They are ASCII text files containing space or tab delimited character strings or numbers, where character strings need further be delimited by single quotes. They can be created in and then exported from text editors, spreadsheets, or word processors as detailed in section 4.4. Further sequential files details and caveats are set out in Chapter 4.

Both input files follow a standard file structure that begins with a standard header which is followed by standard data lines.

The standard header contains two lines with a title in the first line and columns headers in the second line.

The data lines formats are specific for each type of file and cannot be mixed in the same file. They are described below.

2.3 Alphanumerical input files

There are 2 cases for input data lines that correspond to two different couples of measurements:

- case 1: two apparent dips or
- case 2: one apparent dip with one strike direction.

These cases can be mixed at will in the input file.

2.3.1 Alphanumerical data line case 1: two apparent dips

Table 2.1: Alphanumerical input data line case 1.

Column[1]	Parameter	Type	units	Range	Comments
1	apparent dip 1	real	degrees	[0°, 90°]	Dip = 0° not allowed in this column, except for horizontal planes
2	azimuth 1	character		'E','W','N','S'[2]	
3	apparent dip 2	character	degrees	[0°, 90°]	Dip = 0° allowed here
4	azimuth 2	real		'E','W','N','S'[2]	

[1] Column number refers to spreadsheet columns, i.e., to the tab-separated entries in the text file.

[2] N, E, S, W are upper case, single character variables delimited by single quotes. N, E, S, W = North, East, South, West.

One rule must be followed:

- 0° dip measurements must not be in columns 1 and 2, but in columns 3 and 4. The only exception is for horizontal planes where both apparent dips are zero and thus zero dip is accepted in the first column.

Particular cases:

- horizontal planes: both apparent dips = 0° can be entered; this is the only case where dip = 0° is accepted in column 1;
- vertical plane measurements should be entered as dip and strike measurements in alphanumerical data line case 2.

The alphanumerical data line case 1 is described in Table 2.1.

2.3.2 Alphanumerical data line case 2: one apparent dip with one strike direction

Rules to follow:

- Apparent dip data is given in columns 1 and 2
- Strike is given in column 3 followed by 'A' (for Azimuth) in column 4
- Strike should be chosen so that dip is to the right (if not, it will be corrected)

The alphanumerical data line case 2 is described in Table 2.2.

Table 2.2: Alphanumerical input data line case 2.

Column[1]	Parameter	Type	units	Range	Comments
1	apparent dip 1	real	degrees	[0°, 90°]	Dip = 0° not allowed in this column, except for horizontal planes
2	azimuth 1	character		'E','W','N','S'[2]	
3	strike	real	degrees	[-360°, +360°]	
4	strike indicator	character		'A'[3]	

[1] Column number refers to spreadsheet columns, i.e., to the tab-separated entries in the text file.
[2] N, E, S, W are upper case, single character variables delimited by single quotes. N, E, S, W = North, East, South, West.
[3] A is an upper case, single character variable delimited by single quotes that indicates that strike is given in column 3.

2.4 Numerical input files

Table 2.3: Numerical input data line.

Column[1]	Parameter	Type	units	Range	Comments
1	apparent dip 1	real	degrees	[0°, 90°]	Dip = 0° not allowed in this column, except for horizontal planes
2	azimuth 1	real	degrees	[-360°, +360°]	
3	apparent dip 2	real	degrees	[0°, 90°]	Dip = 0° allowed here
4	azimuth 2	real	degrees	[-360°, +360°]	

[1] Column number refers to spreadsheet columns, i.e., to the tab-separated entries in the text file.

Only one restriction must be followed:

- 0 dip measurements must not be in columns 1 and 2, but in columns 3 and 4. The only exception is for horizontal planes where both apparent dips are zero and thus zero dip is accepted in the first column.

The numerical data line is described in Table 2.3.

2.5 Input examples

2.5.1 Input data examples

Examples of acceptable and erroneous orientation data are given for

- alphanumerical input in Table 2.4,
- and for numerical input in Table 2.5.

2.5.2 Input files samples

Samples for each type of input file are available in the 'Sample_files' folder as follows:

- alphanumerical input file: 894G8R1_in1_aln.txt (*Shipboard Scientific Party*, 1993b)
- numerical input files: 1109D47R2_in1_num.txt (*Shipboard Scientific Party*, 1999c)

Table 2.4: Examples of alphanumerical input data lines.

Columns				Comments
1	2	3	4	
41	'E'	56	'S'	OK case 1: 2 dips
41	'W'	56	'S'	OK case 1: 2 dips (*)
41	'E'	56	'N'	OK case 1: 2 dips
41	'W'	56	'N'	OK case 1: 2 dips
60	'W'	0	'S'	OK case 1: dip + strike (**)
0	'W'	0	'N'	OK case 1: horizontal plane
56	'S'	41	'W'	OK case 1: 2 dips, equivalent to (*)
0	'S'	60	'W'	Faulty case 1: dip + strike, dip = 0° should be entered second as in (**), or as strike as in (***)
90	'E'	30	'A'	OK case 2: vertical plane
60	'E'	0	'A'	OK case 2: dip + strike
60	'S'	90	'A'	OK case 2: dip + strike
60	'W'	180	'A'	OK case 2: dip + strike (***)
60	'N'	270	'A'	OK case 2: dip + strike
60	'N'	90	'A'	Flawed case 2: dip + strike, strike will be corrected to 270° (dip to the right)

Table 2.5: Examples of numerical input data lines.

Columns				Comments
1	2	3	4	
41	90	56	180	OK 2 dips
41	270	56	180	OK 2 dips
56	0	41	90	OK 2 dips
56	0	41	270	OK 2 dips
60	270	0	180	OK dip + strike
0	270	0	0	OK 2 dips: horizontal plane
0	270	60	0	Faulty dip + strike: dip = 0° should be entered second as in (*)
90	90	0	30	OK dip + strike: vertical plane
60	90	0	0	OK dip + strike
60	180	0	90	OK dip + strike
60	270	0	180	OK dip + strike
60	0	0	270	OK dip + strike (*)
60	0	0	90	Flawed dip + strike: strike will be corrected to 270° (dip to the right)

Chapter 3

Output files

3.1 Introduction

App2truedip generates two types of output files:

1. **alphanumerical output files** when the input file is alphanumerical;
2. **numerical output files** when the input file is numerical;

This chapter describes the formats of these two output files.

3.2 Common characteristics of both types of output files

The output files are Fortran sequential output files. They are ASCII text files containing tab delimited characters and numbers so as to facilitate importing the results into a spreadsheet. Character strings are delimited by single quotes so that the ouput file can also be used as input file for App2truedip. Further sequential files specifications can be found in Chapter 4.

Both alphanumerical and numerical output files follow a standard file structure that begins with a standard header which is followed by standard data lines.

The standard header contains two lines with a title in the first line and columns headers in the second line.

The data lines contains 7 columns in both types of files:

- Columns 1 to 4 repeat the input data in the same format as in the input file,
- Columns 5 and 6 contain the results of the restoration: strike (with dip to the right convention) and true dip,

- Column 7 contains comments.

Further data line characteristics are specific to each type of output file and are described below.

3.3 Alphanumerical output files

There are 2 cases for output data lines that correspond to the two different input couples of measurements:

1. case 1: two apparent dips or
2. case 2: one apparent dip with one strike direction.

3.3.1 Alphanumerical data line case 1: two apparent dips input

The alphanumerical output data line case 1 is described in Table 3.1.

Table 3.1: Alphanumerical output data line case 1.

Column[1]	Parameter	Type	units	Range
1	apparent dip 1	real	degrees	[0°, 90°]
2	azimuth 1	character		'E','W','N','S'[2]
3	apparent dip 2	real	degrees	[0°, 90°]
4	azimuth 2	character		'E','W','N','S'[2]
5	strike[3]	real	degrees	[0°, +360°]
6	true dip	real	degrees	[0°, 90°]
7	comments[4]	character		

[1] Column number refers to spreadsheet columns, i.e., to the tab-separated entries in the text file.
[2] N, E, S, W = North, East, South, West.
[3] Such that dip is to the right.
[4] Eventual explanation of the input data problem that precluded computation.

3.3.2 Alphanumerical data line case 2: one apparent dip with one strike direction input

The alphanumerical output data line case 2 is described in Table 3.2.

3.4 Numerical output files

The numerical data line is described in Table 3.3.

Table 3.2: Alphanumerical output data line case 2.

Column[1]	Parameter	Type	units	Range
1	apparent dip 1	real	degrees	[0°, 90°]
2	azimuth 1	character		'E','W','N','S'[2]
3	strike[3]	real	degrees	[-360°, +360°]
4	strike indicator	character		'A'[4]
5	strike[5]	real	degrees	[0°, +360°]
6	true dip	real	degrees	[0°, 90°]
7	comments[6]	character		

[1] Column number refers to spreadsheet columns, i.e., to the tab-separated entries in the text file.
[2] N, E, S, W = North, East, South, West.
[3] Eventually corrected so that dip is to the right.
[4] A indicates that strike is given in column 3.
[5] Such that dip is to the right.
[6] Eventual explanation of the input data problem that precluded computation.

Table 3.3: Numerical output data line.

Column[1]	Parameter	Type	units	Range
1	apparent dip 1	real	degrees	[0°, 90°]
2	azimuth 1	real	degrees	[-360°, +360°]
3	apparent dip 2	real	degrees	[0°, 90°]
4	azimuth 2	real	degrees	[-360°, +360°]
5	strike[2]	real	degrees	[0°, +360°]
6	true dip	real	degrees	[0°, 90°]
7	comments[3]	character		

[1] Column number refers to spreadsheet columns, i.e., to the tab-separated entries in the text file.
[2] Such that dip is to the right.
[3] Eventual explanation of the input data problem that precluded computation.

3.5 Output examples

Examples of each type of output file are available in the 'Sample_files' folder as follows:

- alphanumerical output file: 894G8R1_out1_aln.txt which is generated from input file 894G8R1_in1_aln.txt (*Shipboard Scientific Party*, 1993b).
- numerical output files: 1109D47R2_out1_num.txt which is generated from input file 1109D47R2_in1_num.txt (*Shipboard Scientific Party*, 1999c).

Chapter 4

Fortran sequential files

4.1 Introduction

This chapters describes a few common attributes of sequential files used as input for or output from FORTRAN programs.

4.2 Fortran input/output sequential files

- Input or output sequential files are ASCII files, i.e. plain text files without accentuated characters. If input files are prepared within a software other than a pure text editor, such as a word processor or a spreadsheet, they need be exported as text only files.
- During input, each reading statement normally focusses on one input line with the following conventions:
  - if all the data to be read within a line are found, the rest of the line is not read; the next input statement will seek its data in the next line; this implies that extra information can be added AFTER the required data without affecting the input;
  - if all the data to be read are not found within a line, the missing data will be sought in the next line; this implies that an incomplete line will be completed by the probably misinterpreted next line.
- There are two possible formats for FORTRAN sequential files: free or fixed formats.

4.2.1 Fortran free format

- Input reading rules:
  - numbers must be delimited by empty spaces or tabulations;

  - character strings must be delimited by single quotes, ', to be read properly.

- Possible file preparation:
  - in a text editor, spreadsheet, or word processor, and saved as text only;
  - saving as tab separated text is most convenient, as it is easily exchanged with spreadsheets.

Free format is preferred and used as much as possible for input, not only because it is insensitive to exact data placement in the line (it is only sensitive to data order sequence and delimiters), which avoids many causes of input error, but also because it can easily be exchanged with spreadsheets.

4.2.2 Fortran fixed format

- Input reading rules:
  - numbers and character strings are read within a specific column location; column here meaning the number of characters (or spaces) counted from the beginning of the line;
  - no delimiters are necessary: character strings or numbers are delimited by their column location;
  - a column offset in a data line will result in misinterpreted data; for instance, if '1234' or 'abcd' are offset by one column to the left, they will be read in input as '2340' and 'bcd '.

- Possible file preparation:
  - in a text editor (recommended) or a word processor and saved as text only; each character or digit position must be counted from the beginning of each line.

Fixed format thus does not tolerate errors in data placement in the line, but does not require delimiters, which make it convenient in some circonstances.

4.3 Operating system issues

4.3.1 Encoding

Text encoding systems that are compatible with ASCII should work. On MacOS X both Mac OS Roman and UTF-8 work.

4.3.2 End of line

The special character used to mark the end of line (EOL) in the input file must be consistent with the system used to run the software (Table 4.1). If the end of line is not recognized, the whole input file may appear as a single line to the program. Problems tend to arise when the file is transferred from one operating system to another, or when the file is exported from a word processor or spreadsheet. If ftp is used between systems, setting text, instead of binary, transfer of data files should translate the end of line.

It is therefore recommended to use a text editor to check, and eventually correct, end of line characteristics of data files that have been exchanged between systems or that have been exported from a word processor or a spreadsheet.

Table 4.1: End of line (EOL) coding in common operating systems.

Operating system	EOL symbol	EOL description
MacOS X	LF	Line Feed
Unix	LF	Line Feed
Windows	CRLF	Carriage Return + Line Feed
MacOS Classic	CR	Carriage Return

4.3.3 End of file

- Aways terminate the file with an empty line (extra line with no space). This avoids putting the end of file (EOF) tag in the last data line. In some systems, including MacOS X, such a situation can result in the last data line not being read.
- Always check that the number of data stored in the program is exactly the expected number of data. If one datum is missing, the above most likely applies.

4.4 Creating input files

Data files can be created with a text editor (recommended) or exported from a word processor. Free format data files can also be exported from a spreadsheet.

4.4.1 Text editors

Preparing an input file with a text editor has the advantage of directly creating a plain text file. There then only remain two issues to deal with when saving the file:

1. check and eventually modify end of line coding, and
2. make sure the end of file is below the last data line.

Here are a few text editors that allow to verify and alter end of line and text encoding of text files:

- under MacOS X: TextWrangler, BBEdit, Smultron, and Plain Text Editor;
- under Windows: ConTEXT.

4.4.2 Spreadsheets

Input files that are read in free format can be exported from spreadsheet. There are then four issues to deal with:

1. exporting the file as tab delimited text,
2. making sure that character strings are enclosed within quotes for free format files,
3. checking and eventually correcting end of line coding, and
4. making sure the end of file is below the last data line.

The last three issues are best dealt with by importing the file into a text editor.

4.4.3 Word processors

Finally, input files can also be exported from a word processor. Three issues must be dealt with:

1. exporting the file in plain text (tab delimited columns recommended),
2. checking and eventually correcting end of line coding, and
3. making sure the end of file is below the last data line.

Again, the last two issues are best dealt with by importing the file into a text editor.

4.5 Standard input/output files

A file format designed to be easily exchanged with spreadsheets is called 'standard file' in this documents and used as much as possible by the software.

4.5.1 Standard file structure

- Data files are ASCII files.
- They are made of a header followed by data lines.

4.5.2 Standard header

The standard header is made of two lines:

1. Title line: the first line contains the title.
2. Columns headers line: the second line contains the columns headers

The standard header is read in free format: title and column headers are character strings and need be delimited by single quotes ', so as to be read properly.

4.5.3 Standard data line

- All parameters for each datum are given in a single data line.
- Data line are read in free format. Parameters may be separated by empty spaces or tabs. Character strings need be delimited by single quotes '.

4.5.4 Standard file example

Example of a standard data file with 2 reals, 1 integer and 1 character string per data. Here are the 3 first lines of the file with two header lines and the first data line:

1. 'Title'
2. 'Parameter-1' 'Parameter-2' 'Parameter-3' 'Parameter-4'
3. 12000.6 2999.4567 245 'label-of-data1'

Chapter 5

How to refer to App2truedip and relevant references

5.1 How to refer to App2truedip

If you publish results obtained with App2truedip, it would be appreciated that you referred to:

- the software version and its location as:
  Celerier, B., YYYY, App2truedip: Apparent to true dip restoration software, version XX.X,
  http://www.celerier.gm.univ-montp2.fr/software/dcmt/app2truedip/app2truedip.html.
  where XX.X and YYYY are the version and year of the software used that are displayed in the console when the program starts and terminates.

5.2 Other relevant references

- Early ODP conventions for structural measurements in boreholes with reference frame different from that in paleomagnetism: *Shipboard Scientific Party* (1991a, 1992a).
- Usual ODP conventions for structural measurements in boreholes with same reference frame as that in paleomagnetism: *Shipboard Scientific Party* (1992b,c, 1993a, 1995, 1999a,b, 2003).
- Core structures reoriented with paleomagnetism: *Shipboard Scientific Party* (1991b); *Lallemant et al.* (1993).
- Core structures reoriented with imaging logs: *MacLeod et al.* (1992); *MacLeod and Pratt* (1994).

Chapter 6

App2truedip revision history

App2truedip was originally designed in summer 1997 to restore strike and dip of planar structures measured as two apparent dips on ODP Leg 176 cores. Numerical input format was added during Leg 180 in summer 1998. Originally developed on MacOS Classic, it was ported to SunOS workstations, and to MacOS X. Version history is summarized in Table 6.1.

Table 6.1: App2truedip program versions

Version	Date	Comments
5.1	13 April 2025	Additional run information in output files header.
5.0	25 March 2025	Accept N, S, E, W in first measurement, revise output and dialogs, better handling of horizontal and vertical planes.
4.3	15 March 2025	Restructure input validation tests, revise output comments.
4.2	13 March 2025	Revise file output, menus and dialogs, f90.
4.1	11 March 2025	Add run information in output header, revise output, menus, and dialogs, f90.
4.0	28 August 2022	Adapt to upgraded libraries.
3.2	1 January 2018	Adapt to upgraded libraries.
3.1	17 May 2012	Adapt to upgraded libraries.
3.0	17 May 2012	Standard file header, upgraded libraries, last MacOS Classic version.
2.2	16 May 2012	Tidy up source code.
2.1	10 November 2004	Allows input from file or keyborad, output to file or screen; MacOS X version.
2.0	22 June 1998	Add numerical input format.
1.0	30 July 1997	Alphanumerical input format only.

Acknowledgments

The best parts of this work were stimulated by discussions with and suggestions from Chris MacLeod, Benoit Ildefonse, Bernard Le Gall, & Veronique Gardien. I remain sole responsible for the worst parts.

Bibliography

Lallemant, S. J., T. Byrne, A. Maltman, D. Karig, and P. Henry (1993), Stress tensors at the toe of the Nankai accretionary prism: An application of inverse methods to slickenlined faults, in *Proceedings of the Ocean Drilling Program, Scientific Results*, vol. 131, edited by A. Taira, I. Hill, J. V. Firth, and P. J. Vrolijk, pp. 103–122, Ocean Drilling Program, College Station, Texas, doi:10.2973/odp.proc.sr.131.109.1993.

MacLeod, C. J., and C. E. Pratt (1994), In-situ stresses in the Lau basin and Tonga forearc (Southwest Pacific), in *Proceedings of the Ocean Drilling Program, Scientific Results*, vol. 135, edited by J. W. Hawkins, L. M. Parson, J. F. Allan, J. Resig, and P. Weaver, pp. 287–299, Ocean Drilling Program, College Station, Texas, doi:10.2973/odp.proc.sr.135.159.1994.

MacLeod, C. J., L. M. Parson, W. W. Sager, and ODP Leg 135 Scientific Party (1992), Identification of tectonic rotations in boreholes by the integration of core information with formation microscanner and borehole televiewer images, in *Geological Applications of Wireline Logs II*, *Geol. Soc. London Spec. Publ.*, vol. 65, edited by A. Hurst, C. M. Griffiths, and P. F. Worthington, pp. 235–246, doi:10.1144/GSL.SP.1992.065.01.

Shipboard Scientific Party (1991a), Explanatory notes, in *Proceedings of the Ocean Drilling Program, Initial Reports*, vol. 131, edited by A. Taira, I. Hill, J. V. Firth, and Shipboard Scientific Party, chap. 4, pp. 25–60, Ocean Drilling Program, College Station, Texas, doi:10.2973/odp.proc.ir.131.104.1991.

Shipboard Scientific Party (1991b), Site 808, in *Proceedings of the Ocean Drilling Program, Initial Reports*, vol. 131, edited by A. Taira, I. Hill, J. V. Firth, and Shipboard Scientific Party, chap. 6, pp. 71–269, Ocean Drilling Program, College Station, Texas, doi:10.2973/odp.proc.ir.131.106.1991.

Shipboard Scientific Party (1992a), Explanatory notes, in *Proceedings of the Ocean Drilling Program, Initial Reports*, vol. 134, edited by J. Y. Collot, H. G. Greene, L. B. Stokking, and Shipboard Scientific Party, chap. 6, pp. 65–91, Ocean Drilling Program, College Station, Texas, doi:10.2973/odp.proc.ir.134.106.1992.

Shipboard Scientific Party (1992b), Explanatory notes, in *Proceedings of the Ocean Drilling Program, Initial Reports*, vol. 135, edited by L. Parson, J. Hawkins, J. Allan, and Shipboard Scientific Party, chap. 2, pp. 49–79, Ocean Drilling Program, College Station, Texas, doi:10.2973/odp.proc.ir.135.102.1992.

Shipboard Scientific Party (1992c), Explanatory notes, in *Proceedings of the Ocean Drilling Program, Initial Reports*, vol. 140, edited by H. J. B. Dick, J. Erzinger, L. B. Stokking, and Shipboard Scientific Party, chap. 1, pp. 5–33, Ocean Drilling Program, College Station, Texas, doi:10.2973/odp.proc.ir.140.102.1992.

Shipboard Scientific Party (1993a), Explanatory notes, in *Proceedings of the Ocean Drilling Program, Initial Reports*, vol. 147, edited by K. Gillis, C. Mével, J. Allan, and Shipboard Scientific Party, chap. 2, pp. 15–42, Ocean Drilling Program, College Station, Texas, doi:10.2973/odp.proc.ir.147.102.1993.

Shipboard Scientific Party (1993b), Site 894, in *Proceedings of the Ocean Drilling Program, Initial Reports*, vol. 147, edited by K. Gillis, C. Mével, J. Allan, and Shipboard Scientific Party, chap. 3, pp. 45–108, Ocean Drilling Program, College Station, Texas, doi:10.2973/odp.proc.ir.147.103.1993.

Shipboard Scientific Party (1995), Explanatory notes, in *Proceedings of the Ocean Drilling Program, Initial Reports*, vol. 153, edited by M. Cannat, J. A. Karson, D. J. Miller, and Shipboard Scientific Party, chap. 2, pp. 15–42, Ocean Drilling Program, College Station, Texas, doi:10.2973/odp.proc.ir.153.10X.1995.

Shipboard Scientific Party (1999a), Explanatory notes, in *Proceedings of the Ocean Drilling Program, Initial Reports*, vol. 176, edited by H. J. B. Dick, J. H. Natland, D. J. Miller, and Shipboard Scientific Party, chap. 2, pp. 1–42, Ocean Drilling Program, College Station, Texas, doi:10.2973/odp.proc.ir.176.102.1999.

Shipboard Scientific Party (1999b), Explanatory notes, in *Proceedings of the Ocean Drilling Program, Initial Reports*, vol. 180, edited by B. Taylor, P. Huchon, A. Klaus, and Shipboard Scientific Party, chap. 4, pp. 1–75, Ocean Drilling Program, College Station, Texas, doi:10.2973/odp.proc.ir.180.104.2000.

Shipboard Scientific Party (1999c), Site 1109, in *Proceedings of the Ocean Drilling Program, Initial Reports*, vol. 180, edited by B. Taylor, P. Huchon, A. Klaus, and Shipboard

Scientific Party, chap. 6, pp. 1–298, Ocean Drilling Program, College Station, Texas, doi:10.2973/odp.proc.ir.180.106.2000.

Shipboard Scientific Party (2003), Explanatory notes, in *Proceedings of the Ocean Drilling Program, Initial Reports*, vol. 206, edited by D. S. Wilson, D. A. H. Teagle, G. D. Acton, and Shipboard Scientific Party, chap. 2, pp. 1–94, Ocean Drilling Program, College Station, Texas, doi:10.2973/odp.proc.ir.206.102.2003.