

TPOINT — Telescope Pointing Analysis System (v4.4)

Contents

1	INTRODUCTION	2
1.1	System Overview	3
1.2	Getting Started	4
1.3	Syntax	6
2	MODELLING	7
2.1	The Pointing Terms	7
2.2	Order of Terms	10
2.3	Sizes of Coefficients	11
2.4	Saving Models	11
3	GRAPHICS	12
3.1	Using the G... Commands	14
4	POINTING DATA	16
4.1	Reference Frames	16
4.2	INDAT Data Formats	17
4.3	Star Catalogues	19
4.4	Saving and Combining Processed Data	20
4.5	Sample Data	21
4.6	Simulated Pointing Tests	22
5	ADVANCED MODELLING	22
6	MAINTENANCE	24
6.1	Directories and Files	24
6.2	Compiling and Linking	26
6.3	New Commands	26
6.4	Local Pointing Terms	27
7	HISTORY AND ACKNOWLEDGMENTS	27
8	COMMAND REFERENCE	29
9	COMMAND SUMMARY	51

1 INTRODUCTION

TPOINT is an interactive telescope pointing analysis system. It allows data from pointing tests to be input and fitted to various models. The residuals from the fits can be displayed in a variety of graphical formats. If systematic errors are visible, the pointing model can be adjusted by adding and removing terms.

The system is available for PC/MS-DOS, VAX/VMS, DECstation/Unix and Sun SPARCstation/SunOS. The present document concentrates on the VAX and Unix versions. For information on running the PC version please refer to the file READ.ME.

TPOINT can deal with data from either equatorial or altazimuth mounts, and could easily be extended to alt/alt and other designs. It has been applied to many telescopes, including optical, IR, mm and radio, and is in routine use at the AAO, UKST, UKIRT, LPO and several other observatories.

Many observatories have their own programs for fitting models to pointing data, and in a numerical sense TPOINT cannot be expected to produce results that are different or better. However, TPOINT is unusual (a) in that the telescope model can be changed during the interactive session, and (b) in the variety of its graphical displays. These two features allow rapid exploration of the pointing possibilities of any given telescope and may lead to a better model than the one previously used; this has already happened in several cases.

The general approach taken by TPOINT is that as far as possible the telescope model should describe real effects (geometrical misalignments, well understood flexures, *etc.*), and empirical functions should be used only to mop up any remaining systematic errors. There is a school of thought which advocates using empirical functions (for example polynomials or spherical harmonics) for the whole job. However, the TPOINT approach has some advantages:

- Simple geometrical misalignments – for example a miscentred instrument on a mount which is located at a Nasmyth focus but is not coincident with the elevation axis – might require very complicated empirical functions but are simple to deal with analytically.
- Direct manipulation of certain geometrical terms while the telescope is in operation can be very useful. An important application of this technique is where a star is switched from one instrument aperture to another simply by changing the collimation parameters. Another example is where the polar axis of a wide field telescope is routinely raised and lowered as a function of declination, to minimize the field rotation effects of differential refraction; the polar axis elevation parameter in the pointing model can simply be changed by the same amount and accurate pointing is maintained.
- A realistic model of a telescope frequently exposes mechanical deficiencies which can then be diagnosed and cured.
- A realistic model is likely to require fewer terms, and the number of stars observed in a pointing test can be correspondingly smaller.
- Physically-based models are less likely to misbehave when extrapolating outside the area covered by the available test data.

In any case, the models available with TPOINT allow considerable empiricism to be employed when called for. A good example of this is the AAT model. Though firmly based on well-understood geometrical misalignments and plausible tube flexure corrections, there are also

complicated harmonics describing what are believed to be flexures in the horseshoe and centre section. These flexure models are encapsulated in two terms called HFX and HFD, which are always left at their nominal sizes and are not fitted when individual pointing tests are reduced. HFX and HFD were determined by empirical fits to a very large sample (well over 1000 stars) formed by superimposing the residuals from tests carried out over several years.

1.1 System Overview

TPOINT is a monolithic program which accepts simple commands from a terminal. There is a rudimentary ‘command procedure’ capability, and a log file is produced. Pointing test data are input from ordinary ASCII files (as produced by a text editor) in flexible formats. On the VAX and the Unix platforms, graphical output is via the SGS/GKS package. On the PC a simplified SGS-like interface to the Microsoft graphics library is used. A small star catalogue is provided, and some sample data. There is a HELP system.

On all platforms, the command which starts the TPOINT system is this:

```
tpoint logfile proclib starcat
```

The three parameters – the log file, the initial procedure library, and the initial star catalogue – are all optional and are normally allowed to default, respectively, to the following files:

	PC	VAX	Unix
<i>logfile</i>	NUL	LOG.LIS	tpoint.log
<i>proclib</i>	PROCS.DAT	TPDIR:PROCS.DAT	/star/etc/tpoint/procs
<i>starcat</i>	STARS.DAT	TPDIR:STARS.DAT	/star/etc/tpoint.stars

On termination (via the END command) the specified log file is PURGED but not printed; this is left to the user. The log contains a full record of the session, and includes extra information (*e.g.* correlations between pointing terms) too extensive to be displayed on the terminal.

Each TPOINT command line consists of one or more fields separated by spaces. The first field, of up to 6 characters, is the **command name**, and specifies the action to be performed. The subsequent fields, if any, are the **arguments**. Where the arguments are numeric, free-format decoding is performed, and numbers may be entered in a wide variety of formats. The precise way the arguments are interpreted depends on the command in question.

TPOINT deals with all forms of input – both commands and the various sorts of data file – in a standardized way which allows for comments and blank lines to be used and provides consistent handling of lowercase information. Details of these conventions are given in the section *Syntax*, later.

To abort a command, type CTRL/C. The command will terminate cleanly as soon as it can, and control will be returned to the terminal ready for the next command. To exit normally from TPOINT, use the END command.

A full list of all TPOINT commands, in alphabetical order, is given later. A quick reference list can be found at the end.

Sequences of TPOINT commands can be executed from a library. A procedure library file is a sequential file of up to 80-character records, searched from the start each time. Standard TPOINT syntax applies to such libraries, which may thus contain blank lines and comments (beginning with '!') to enhance readability. In using procedures from a library, the following commands are involved:

CALL <i>x</i>	calls procedure <i>x</i> from the library
PROC <i>x</i>	specifies the start of procedure <i>x</i>
RETURN	returns from and ends a procedure

Of these commands, CALL alone is permissible from the terminal; PROC and RETURN are used only in library procedures. The standard procedure library file is automatically loaded when TPOINT is started unless a different one is specified on the command line as described above. After loading the library, TPOINT automatically executes a 'CALL INIT'; the INIT procedure supplied in the standard library does nothing, but one in a private version can contain any required preliminary operations – for instance preparations for plotting on a favourite graphics device. All procedure libraries should contain an INIT procedure, otherwise an error message will occur during TPOINT setup. A private library can be loaded at any time during the TPOINT session by using the INPRO command. The INIT procedure is not called under these circumstances. An INPRO command without arguments restores the original library. By default, procedure commands are not echoed on the terminal; they can be made to appear by means of the ECHO ON command.

1.2 Getting Started

The best way to learn how TPOINT works is to reduce some real data, using one of the sample files. Proceed as follows. First start the system:

```
tpoint
```

There will be various announcements, followed by a '*' prompt. Except on the PC, select a suitable plotting device:

```
PLTON device
```

(Consult your system manager for local SGS device names.) Read in the sample UKST data:

```
INDAT TPDIR:UKST (VAX)
or INDAT UKST (PC)
or INDAT /star/bin/examples/tpoint/ukst.dat (Unix)
```

The observations will be listed on the terminal. Specify the standard geometrical model for an equatorial mount, and then fit the model to the data:

```
USE IH ID NP CH ME MA
```

```
FIT
```

The values of the coefficients are reported, and the RMS and population standard deviation. Plot the residuals:

```
CALL E9
```

The axis labelling will be described later. In this particular case, conspicuous systematic effects are present in the dD (declination residual) versus H (hour angle) plot. The term FO (fork flexure) is a standard one for fork mounted telescopes and produces this characteristic pattern of residuals. Include the term in the model and fit again:

```
USE FO
```

```
FIT
```

Note that the RMS and population standard deviation are much improved. Plot the residuals again:

```
CALL E9
```

There are no obvious remaining systematic effects.

This illustrates the general strategy for modelling a telescope from scratch. Begin by suitably preparing the TPOINT system and inputting the data. Then create a preliminary model consisting of the basic set of geometrical terms, and perform a fit:

```
INDAT file
```

```
USE IH ID NP CH MA ME      (for an equatorial)
or USE IA IE NPAE CA AW AN  (for an altazimuth)
```

```
FIT
```

(Procedures for setting up the basic equatorial and altaz models are also provided in the standard library: use CALL EQUAT or CALL ALTAZ.) The fit can be repeated until the solution settles down. Make a note of the population standard deviation, which is an indication of the general quality of the telescope before any modelling of flexure *etc.* The next step is to display the residuals graphically. A useful way to start is procedure E9 or A9 in the standard library:

PLTON <i>device</i>	to prepare the plotting device
CALL E9	9 favourite plots for an equatorial
or CALL A9	9 favourite plots for an altazimuth

(The standard library also includes procedures E6 and A6 for the six most useful plots, to reduce plotting time on slow terminals.)

By means of the USE and FIT commands, try out extra terms to reduce any systematic errors. The simple Hooke's-law tube flexure term TF is frequently required; for fork mounted equatorials try the fork flexure term FO, and for mounts with a declination axis supported at one end only (the English cross-axis and the German mountings are examples) try the declination axis flexure term DAF. When adding new terms, pay attention to the standard deviation of the new coefficient, the effect on other terms, and whether the population standard deviation has been reduced. Remove new terms if they are reported as being indistinguishable from existing ones, or if they lead to the appearance of messages warning that the fit is ill-conditioned. With the exception of the six standard geometrical terms, it is wise to reject any terms which are not at least 2 sigma in size. For each FIT, a table of the correlations between every pair of terms is output to the LOG file, for inspection after the TPOINT session; values close to unity show terms which are not easily distinguished given the current data.

More advanced information on modelling strategy is given in a later section.

The current settings of various internal parameters can be displayed as follows:

SHOW

The list produced by SHOW includes the command names required to change the parameters, and is a useful guide to some of the facilities available within TPOINT.

1.3 Syntax

All TPOINT commands, procedure library records, star catalogue records, model data records, and pointing data records, are subjected on input to a preliminary vetting and conditioning, as follows:

- Comments – records which are blank, or which have '!' as the first non-space character – are logged and displayed if appropriate but are otherwise ignored.
- Non-printing characters (TABs for instance) are replaced by single spaces.
- Leading blanks are eliminated as appropriate.
- Except within *string arguments*, lowercase characters are converted to uppercase.

String arguments are groups of characters delimited by pairs of either " or ' characters to show that they must stay lowercase. When a procedure library file is being input, string argument delimiters are left intact (to allow library commands to have lowercase arguments). For TPOINT commands, string argument delimiters are interpreted by the command routine itself. When star catalogue files, pointing data files, and model data files are input, string argument delimiters are removed.

2 MODELLING

The pointing model is a sequence of terms selected from an internal repertoire. As well as explicitly formulated terms (geometrical effects for example) there is a generic type covering a wide range of polynomials and harmonics.

Terms can be **added** to the model by means of the USE command. For example, to add the terms for polar axis misalignment:

```
USE ME MA
```

The USE command is also used to re-enable fitting after a FIX command.

Terms can be **removed** from the model by means of the LOSE command. For example, to drop the term PDH2:

```
LOSE PDH2
```

The value of the coefficient of a single term can be **specified** by means of the command:

```
coeff val
```

where *coeff* is the name of the term and *val* the value in arcseconds. Thus to set the HA index error to +20.7 arcsec:

```
IH 20.7
```

You would normally only do this for 'FIXed' coefficients – ones excluded from fitting.

The current value of the coefficient of a single term can be **inquired** by entering a command consisting simply of the name of the term. For example, to display the current value of the declination index error, simply use the command:

```
ID
```

The current model is **fitted** to the observations by means of the FIT command. Individual terms can be frozen at a particular value by means of the FIX command, or reinstated in the fitting by means of the USE command.

2.1 The Pointing Terms

The pointing terms fall into five categories – equatorial, altazimuth, special (to a telescope or type of telescope), polynomial and harmonic. The model for any particular telescope may be a mixture of several of these types.

Some terms are functionally identical but have different names for historical, convenience or efficiency reasons. Most of them can be expressed using generic terms (polynomials and harmonics), though doing so will lead to slightly slower execution of the FIT and UNFIT commands. To find out what a given term means in a geometrical sense, refer to the Fortran source for the tpt_PTERMS and tpt_PTERML modules.

The following terms are most useful when modelling an **equatorial** telescope:

IH	index error in HA
ID	index error in Dec
NP	nonperpendicularity of HA and Dec axes
CH	nonperpendicularity of Dec and pointing axes
ME	polar axis elevation error
MA	polar axis error east-west
FO	fork flexure
DAF	flexure of cantilevered Dec axis
HCES	HA centring error (sin component)
HCEC	HA centring error (cos component)
DCES	Dec centring error (sin component)
DCEC	Dec centring error (cos component)
DNP	dynamic nonperpendicularity
X2HS	$\sin(2h)$ term EW
X2HC	$\cos(2h)$ term EW
D2HS	$\sin(2h)$ effect in Dec
D2HC	$\cos(2h)$ effect in Dec
D2DS	$\sin(2\delta)$ term in Dec
D2DC	$\cos(2\delta)$ term in Dec
D4HS	$\sin(4h)$ effect in Dec
D4HC	$\cos(4h)$ effect in Dec
EHS	$\sin(h)$ effect in polar axis elevation
EHC	$\cos(h)$ effect in polar axis elevation
AUX1H	HA change supplied through auxiliary reading 1
AUX1X	EW change supplied through auxiliary reading 1
AUX1D	Dec change supplied through auxiliary reading 1

The following terms apply to **altazimuth** telescopes; some also describe effects in zenith distance which apply equally well to equatorial telescopes:

TF	tube flexure – $\sin \zeta$ law
TX	tube flexure – $\tan \zeta$ law
IE	index error in elevation
IA	index error in azimuth
CA	nonperpendicularity of elevation and pointing axes
AN	NS misalignment of azimuth axis
AW	EW misalignment of azimuth axis
NP AE	nonperpendicularity of azimuth and elevation axes
ACES	Az centring error (sin component)
ACEC	Az centring error (cos component)
ECES	El centring error (sin component)
ECEC	El centring error (cos component)
NRX	Horizontal displacement of Nasmyth rotator
NRV	Vertical displacement of Nasmyth rotator
AUX1A	Az change supplied through auxiliary reading 1
AUX1S	LR change supplied through auxiliary reading 1
AUX1E	El change supplied through auxiliary reading 1

The following terms are **special** to the AAT:

ZH	AAT HA Z-gear effect in HA
ZE	AAT HA Z-gear effect in polar axis elevation
HF	AAT main east-west horseshoe flexure
HGES	36^m gear error in HA – sin
HGEC	36^m gear error in HA – cos
DGES	9° gear error in Dec – sin
DGEC	9° gear error in Dec – cos
TFP	AAT tube flexure – non-Hooke’s-law term
HFx	AAT residual horseshoe east-west
HFD	AAT residual horseshoe north-south
CD4A	AAT coudé 4 collimation error A component
CD4B	AAT coudé 4 collimation error B component
CD5A	AAT coudé 5 collimation error A component
CD5B	AAT coudé 5 collimation error B component

Polynomial terms have names of the form:

$$Pr*c*[i[c[i]]]$$

One example would be a term PXH2D1, which would model an effect which produced an east-west shift on the sky (X) proportional to $h^2\delta$ (H2D1).

The initial ‘P’ identifies this term as a polynomial.

The *r* field describes the result, and is one of the following:

H	result is in hour angle
X	result is east-west on the sky
D	result is in declination
U	result tilts polar axis up/down
L	result tilts polar axis left/right
P	result changes HA/Dec nonperpendicularity
A	result is in azimuth
S	result is left-right on the sky
Z	result is in zenith distance
N	result tilts azimuth axis north/south
W	result tilts azimuth axis east/west
V	result changes Az/El nonperpendicularity

The two *ci* fields indicate the independent variables and their powers. Each *i* is in the range 0–9; each *c* can be any of:

H	hour angle
D	declination
A	azimuth
Z	zenith distance
Q	parallactic angle

Trailing *ci* or *i* fields, if omitted, default to unity. For example, a change in HA proportional to HA (*i.e.* a scale change, such as might be necessary with a roller-driven encoder) can be produced by using the term PHH.

The same independent variable can be specified twice, allowing powers up to 18 to be used.

Harmonic terms have names of the forms:

Hr*fc*[*i*][*fc*[*i*]]
or Hr*fcii*

One example would be a term HXSH2, which would model an effect which produced an east-west shift on the sky (X) proportional to sine (S) of h^2 (H2).

The initial 'H' identifies this term as a harmonic.

The *r* and *c* fields describe the result and an independent variable respectively, and are as for polynomials, above.

Each *fc*i**, *fcii* and *fciii* field indicates a function of an integer multiple of an independent variable *c*. The *f* field is either S for sine or C for cosine. Each *i* is in the range 0–9, allowing frequencies from zero to 999 cycles per revolution in the case of simple one-coordinate harmonics, and 0 to 9 cycles per revolution in the case of compound two-coordinate harmonics. An omitted *i* defaults to unity, as does an omitted trailing *fc*i**.

2.2 Order of Terms

To match different styles of telescope control system, the model can either start with raw telescope positions and predict the corresponding true¹ position, or alternatively can start with true star positions and predict the required raw telescope position. The selection is made with the ADJUST command. The sign convention for the coefficients is the same for both options, so that the two options should deliver coefficients that are substantially the same. When the ADJUST command is used to change the option, the order of the terms in the model is reversed so that more or less the same coefficient values will be produced if a FIT is carried out (following a RESET to cancel the previous set of corrections).

Normally, each term in the model is *chained* to the previous one, the input position for each term being the output position from the previous term. Sometimes this may not be appropriate, and a set of terms can instead share one input position and their adjustments be added in as a group; in this case the terms are said to be *parallel*. The commands CHAIN and PARAL select which option applies to a given term, the first in a parallel group being marked 'chained' and the rest marked 'parallel' to indicate they share the former's starting point.

In typical cases the coefficients will not be much altered by changing the order of terms, or switching terms between chained and parallel. Significant changes are likely only if some coefficients are large, or if high order polynomials or harmonics are employed. It is nevertheless reassuring if the model being fitted matches the way the pointing corrections are implemented in the telescope control system itself, in direction, order, formulation, and chaining.

¹Ones corrected for everything up to and including diurnal aberration and refraction – the formal term is *observed* but the word *true* has been adopted here as marginally less likely to cause confusion.

2.3 Sizes of Coefficients

The mathematical expressions used by TPOINT are rather simple and designed to give results of adequate accuracy when not too close to the pole of the mounting and *when the pointing errors of the telescope are small*. Flexures and misalignments of up to a few arcminutes are unlikely to give problems; they will not interact significantly with one another, and will be calculated by TPOINT to sufficient accuracy. However, terms with larger coefficients may not be so well-behaved, especially near a pole.

A particular source of problems in practice is collimation error, the non-perpendicularity of the chosen pointing axis with respect to the declination or elevation axis of the mounting. If the measuring device – eyepiece, autoguider probe, CCD, TV camera *etc.* – is offset too far, there may be unmodelled pointing errors due to such effects as optical distortion. The best plan is to ensure that the measuring device is offset no more than a few minutes of arc. If this is impossible it is advisable to pre-process pointing observations to remove the bulk of the offset before the TPOINT run. This might involve rigorous calculation of the tangent-plane projection and optical distortions.

2.4 Saving Models

Commands are provided for writing the current model to a file, and for replacing the current model with one read from such a file:

OUTMOD <i>file</i>	write model to a file
INMOD <i>file</i>	read model from a file

where *file* is the name of the file containing the model. As well as providing long-term storage for models, the file may also be read by programs external to the TPOINT system. In addition to the model information required by INMOD, the file contains the current caption and the basic statistics which relate the model to the current data list. The INMOD command ignores these latter items.

The file consists of variable length formatted character records in ‘list’ format (*i.e.* the records do not contain printer vertical format characters). The order of records within the file is:

- caption record
- method and statistics record
- term records (one per term in the model)
- end record

Here is an example file:

```
AAT_f/15_1979/06/11
T_49_1.1639
IH_+175.2234
=ZH_+3.5100
ID_+20.7048
```

```

&=HFX_____+1.0000
&=HFD_____+1.0000
&_HF_____ -22.4514
&_X2HC_____ -3.0225
&_NP_____ +2.9841
&_CH_____ -19.1780
_ =ZE_____ +0.7000
&_ME_____ +57.9758
&_MA_____ +3.2308
__TF_____ +8.0261
&_TFP_____ -2.1594
END

```

The **caption** record can be up to 80 characters long.

The **method and statistics** record can be read with a format specification of (A1,I5,F9.4). The three fields are:

```

method: T or S
number of active observations
sky RMS (arcsec)

```

A method of 'T' means that the model corrects the telescope readings, whereas 'S' means that the model is applied to star positions to predict the required telescope setting.

Each **term** record can be read with a format specification of (2A1,A8,F10.4). The fields are:

```

chained/parallel flag
fixed/floating flag
term name
coefficient value

```

The chained/parallel flag is either a space (for 'chained' terms - ones which are computed from the position as affected by all previous terms) or an ampersand (for 'parallel' terms - members of a group all computed from the same starting position). The fixed/floating flag is either a space (for coefficients that are to be fitted) or an equals sign (for coefficients of fixed value). The term name and coefficient value have their normal meanings.

The **end** record consists of the string 'END'. Reading programs should also interpret end-of-file as equivalent to an end record.

Note: population standard deviation can be obtained as follows. Count the number of floating coefficients (ones not marked '=') and call it n . With sky RMS ρ and number of active observations o , PSD $\sigma = \sqrt{\rho^2 o / (o - n)}$.

3 GRAPHICS

The TPOINT graphics commands are as follows:

CAPT	enable/disable plotting of caption
CLRNG	select zone clearing option
FRAME	enable/disable plotting of graph frame
G	plot residuals in one coordinate against another
GAM	look for changing polar/azimuth axis misalignment
GCLR	erase the whole display surface
GDIST	plot error distributions
GHYST	look for hysteresis
GMAP	plot error vectors on Cartesian Cylindrical projection
GSCAT	plot residuals as a scatter diagram
GSMAP	plot error vectors on orthographic projection
MARKH	specify marker size
PENS	specify pens for lines and text
PLTOFF	switch back to previous graphics workstation
PLTON	switch to a new graphics workstation
PLTZ	specify plotting zone
TXF	specify text font
TXP	specify text precision

A PLTON command is an essential prelude to plotting on the VAX and Unix platforms but has no action on the PC. The various graph plotting commands are then available – G, GAM, GDIST, GHYST, GMAP, GSCAT and GSMAP. PLTOFF will be needed if switching back and forth between multiple graphics devices is to be performed (not applicable to the PC). PLTZ will be useful for displaying several graphs at one, or keeping graph and text separate.

The remaining commands – CAPT, CLRNG, FRAME, GCLR, MARKH, PENS, TXF, TXP – are used either for reducing plotting time, or for achieving special effects for demonstration and publication purposes. They can be ignored by novice users.

TPOINT uses SGS/GKS for all its plotting, and any graphics device supported by the local GKS can be selected by means of the PLTON command. On the PC, all plotting goes to the screen using the best available graphics mode, and there is no need to use the PLTON command. On the other platforms, before any plotting can take place a PLTON command must be executed, specifying the SGS name of the device to be used. A PLTON command with no argument will produce a list of the devices known to the site's SGS/GNS/GKS system. If the appropriate name tables have not been set up (see SUN/57) the list will be empty, and it will be necessary to specify the GKS *workstation type* directly. Further PLTON commands will switch the plotting successively to different devices, while PLTOFF commands will revert to the previous device. A common pattern is for the initial PLTON to specify a screen or window; whenever a hardcopy version of a plot is subsequently required a further PLTON is issued specifying the hardcopy device; a PLTOFF then reverts to the screen or window.

Multiple plotting zones on the one display surface are supported, which allows several plots to be displayed at once. The current plotting zone is specified by the PLTZ command:

PLTZ	use full display surface
or PLTZ name	use named region
or PLTZ x1 x2 y1 y2	use numerically specified region

The predefined zone names specify regular subdivisions of the display surface:

T
B

L	R
---	---

TLQ	TRQ
BLQ	BRQ

TL	TC	TR
CL	CC	CR
BL	BC	BR

If numeric parameters are supplied they describe directly the region of the display surface which is to contain the plotting zone. The numbers are the two X extremes and the two Y extremes respectively, in units where the corresponding dimension of the display surface is unity. All four numbers are required. Irrespective of the shape of the zone and how it was specified, plotting will actually occur in a zone of the standard aspect ratio centred within this area.

The CLRNG command controls what zone or display surface clearing occurs before each plot is drawn. In conjunction with the GCLR command, CLRNG can be used to speed up plotting in cases where multiple zones are in use and clearing of individual zones is a slow operation on the graphics device concerned. On VT100-compatible terminals, the text scroll region can be specified by means of the VT command. On VT100-compatible terminals with a separate graphics plane, the VT and PLTZ commands can be used in combination to confine the text and plots to separate areas of the screen.

3.1 Using the G... Commands

The commands in TPOINT which plot graphs all have names beginning with G. The most important one for deciding what terms to add to the model is called simply **G**, and plots one component of the residuals against one coordinate. It has the following syntax:

$$G \ ydata \ xdata \ [scale]$$

The argument *ydata* specifies which component of the residuals is to be plotted:

H	in HA
X	EW on the sky
D	in Dec
P	corresponding to HA/Dec nonperpendicularity
A	in azimuth
S	horizontally on the sky
Z	in zenith distance
V	corresponding to Az/El nonperpendicularity
R	total residual

The argument *xdata* specifies what coordinate to plot against:

H	hour angle
D	declination
A	azimuth
Z	zenith distance
Q	parallactic angle
N	observation order

The optional argument *scale* indicates the vertical plotting scale: it is the absolute value of the largest residual to be plotted. In default, a scale is chosen automatically to suit the data.

The general strategy is to plot different components of the residuals against the various coordinates. Terms can then be added to the model (USE followed by FIT) to target any systematic effects that are seen. Some plots are more likely than others to have a mechanical interpretation, and attention should be confined to these, at least to start with. For example, an apparent relationship between polar axis elevation and azimuth, on the evidence of a G U A plot, should be treated with great scepticism (especially on an altazimuth mount), whereas systematic residuals on G N A and G W A plots (for an altazimuth) or on G U H and G L H plots (for an equatorial) are much more credible.

Because files of pointing observations are normally written in time order, G *ydata* N plots are useful for exposing shifts or drifts that have happened during the pointing run. Other x-axis meanings can be contrived by appropriately sorting the observation file prior to input.

The other plots have a variety of presentational and diagnostic roles. **GSCAT** plots residuals as a scatter diagram, like a shooting target; as well as being a good way of presenting the overall pointing performance, it exposes errant observations and abnormal distributions. **GSMAP** displays the residuals as error vectors on an orthographic projection (a distant view of the celestial sphere looking down on the zenith from above). The vectors (lines projecting from the square symbols which mark the stars) are the great-circle continuations of the pointing residuals. **GMAP** plots error vectors on a Cartesian Cylindrical projection, either HA/Dec or Az/El. **GDIST** plots histograms showing various sorts of error distribution, and is useful for testing whether the residuals are reasonably normal. **GAM** looks for changing polar/azimuth axis misalignment. Each residual is interpreted as being due to a misalignment of the specified “roll” axis – either polar or azimuth as specified. A histogram is plotted to show if the residuals do, in fact, favour one particular direction. The mean direction is calculated and reported, and then the component of each residual in the direction of this misalignment is plotted. **GHYST** looks for hysteresis, by drawing the error vector at a position which indicates the direction from

which the telescope has come, assuming that the original data file contains all the observations and in the correct order, and that any hysteresis is related to the direction of the large scale movements in cylindrical coordinates (either HA/Dec or Az/El). A numerical estimate of the hysteresis is made, by summing the residuals oriented relative to the telescope movements.

Many of the plotting commands have arguments, for specifying scales and selecting HA/Dec and Az/El options *etc.* Full details are given in the section *Commands*, later.

Finally, the command **GCLR** erases the whole display surface.

4 POINTING DATA

At present, only one command is provided within TPOINT for reading a set of pointing observations: **INDAT**. Several different formats are supported; all are converted into a standard internal form, called the **data list**. The command is:

INDAT *file*

where *file* is the name of the file of pointing observations.

The new observations may either replace or be appended to any previous ones. This is controlled by a flag armed by the **APPEND** command. The flag is reset once the new data have been appended. In the append case the caption is reset and the one supplied in the data file is ignored. The new latitude replaces the previous one; it would, however, be very unusual to combine the data from several telescopes.

The **MASK** and **UNMASK** commands allow specified subsets of the observations to be suppressed temporarily. They are useful for verifying the independence of the solution in these different subsets. The most common application is where observations from only a specified region of the sky are included. Another use of **MASK** and **UNMASK** is to suppress observations whose total residual, on the sky, is greater than some specified amount; however, this is a dubious practice unless there are *a priori* reasons for suspecting the data are invalid – telescope malfunction or star misidentification for example. **MASK** and **UNMASK** can also remove and re-instate individual observations or sequences of observations. Genuinely erroneous observations (for example misidentified stars) should be removed outside TPOINT simply by using a text editor.

4.1 Reference Frames

In the internal form produced by the **INDAT** command, each observation consists of the raw mechanical HA/Dec of the telescope and what is called in TPOINT the ‘true’ HA/Dec of the star (*i.e.* as affected by both diurnal aberration and refraction).

The raw telescope HA/Dec normally means either the mechanical HA/Dec directly read from dials or encoders, or, in the case of non-equatorial mounts, the HA/Dec obtained by applying the appropriate canonical transformation to the actual readings. Az/El data, for example, are transformed by the standard text-book Az/El to HA/Dec rotation, using the longitude and geodetic latitude of the observatory as affected by polar motion. For telescope control systems which can only conveniently log mean places, an option is provided to convert this back to ‘true’

HA/Dec, assuming that all the telescope pointing corrections proper have been disabled but that refraction and diurnal aberration (as well as precession/nutation and annual aberration) have been included.

In order to avoid having to log anything which changes rapidly, the equatorial coordinates accepted by INDAT are in the form of RA/Dec rather than HA/Dec. The telescope HA is deduced simply by subtracting the given RA from the given sidereal time. This applies to all but one of the data formats, where star and telescope azimuths and elevations are supplied directly and must all correspond to one moment in time.

The star 'true' HA/Dec positions are obtained by allowing as necessary for proper motion, precession, nutation, annual aberration, and light deflection to obtain geocentric apparent place, followed by corrections for diurnal aberration and atmospheric refraction. It is also possible to supply the star data as 'true' RA/Dec by (i) including an optional record which disables the diurnal aberration corrections and (ii) omitting the temperature, pressure, *etc.* to disable the refraction corrections. This would be done in cases where the telescope control system which generated the pointing data has already applied these corrections (either correctly, or at least in a way which will be consistent with any subsequent implementation of the pointing model obtained from TPOINT).

The modelling procedures available elsewhere in this package have the job of expressing the relationship between these two sets of coordinates. It is thus the responsibility of the telescope control system itself to perform accurate mean to apparent place transformations, and to allow for Earth polar motion, diurnal aberration, refraction, *etc.*, before applying the pointing model as determined by using this package.

4.2 INDAT Data Formats

The INDAT input file consists of free-format records up to 120 characters in length, as follows:

CAPTION record
 OPTION records (optional)
 RUN PARAMETERS record
 OBSERVATION records
 'END' or end-of-file

Blank lines and comments (which begin with a '!' character) can be included anywhere.

CAPTION record:

First 80 characters used, trailing blanks eliminated

OPTION records (optional):

' : NODA'	disables correction for diurnal aberration
' : ALLSKY'	disables horizon/zenith/pole checks
' : equinox'	required if telescope positions are mean places

The correction for diurnal aberration (maximum 0.3 arcsec) should be disabled if the telescope control system itself omits it. The ALLSKY option is useful in cases where, for one reason or another, the observations are scattered all over the celestial sphere. Where the telescope positions are mean places, only one *equinox* (probably 'B1950' or 'J2000') may be specified, and applies to the whole file.

RUN PARAMETERS record:

latitude ($^{\circ},',''$)
 UTC date (y,m,d)
 temperature ($^{\circ}\text{C}$)
 pressure (mB)
 height above sea level (metre, default determined from pressure)
 relative humidity (range 0–1, default 0.5)
 observing wavelength (micrometre, default 0.55)
 tropospheric lapse rate ($^{\circ}\text{K}/\text{metre}$, default 0.0065)

The latitude is mandatory. If the UTC date is omitted, the star and telescope data must not be mean places. If the refraction data (temperature and so on) are omitted, no corrections for refraction are made. Optical refraction is computed for wavelengths below 100 micrometres, and radio refraction for wavelengths longer than this figure. In the radio case, it is important to specify the humidity accurately.

OBSERVATION records – there are four different ways of expressing each observation, as follows:

Observation record format 1:

star geocentric apparent RA/Dec (h,m,s, $^{\circ},',''$)
 telescope raw RA/Dec (h,m,s, $^{\circ},',''$)
 local apparent sidereal time (h,m)
 [up to 2 auxiliary readings]

Observation record format 2:

star mean RA/Dec (h,m,s, $^{\circ},',''$, $\mu_{\alpha}, \mu_{\delta}$, equinox)
 telescope raw RA/Dec (h,m,s, $^{\circ},',''$)
 local apparent sidereal time (h,m)
 [up to 2 auxiliary readings]

Observation record format 3:

'*' identifier from star catalogue
 telescope raw RA/Dec (h,m,s, $^{\circ},',''$)
 local apparent sidereal time (h,m)
 [up to 2 auxiliary readings]

Observation record format 4:

```

star 'true' Az/El (°,°)
telescope raw Az/El (°,°)
[up to 2 auxiliary readings]

```

Note that format 4 requires exact times to be extracted from the telescope control system, whereas formats 1-3 use instead the telescope RA/Dec, which is unlikely to vary rapidly during tracking of a star and is therefore easier to get right. Moreover, precisely consistent Az/El and time information may simply be unavailable if special provision has not been made in the control software. However, in the case of formats 1-3, if sidereal time of high precision is available, this information can, and should, be supplied, by appending decimals to the sidereal time minutes.

Following the normal convention, the azimuths in format 4 are reckoned from north through east (though internally within the TPOINT software they go from south through east).

Two approaches are available for reading 'foreign' data formats. The first is to implement a local TPOINT command which accepts the foreign format and converts it directly into the internal form. The second approach, which does not involve modifications to TPOINT, is to write a freestanding program which translates the foreign format into one of the INDAT formats. Such translation programs are apt to be laborious to write, and frequently involve applying positional astronomy adjustments that will all be undone again when INDAT reads the file. Nevertheless, this approach is probably the best in most cases.

4.3 Star Catalogues

A small star catalogue is supplied with the TPOINT system and is loaded automatically when the system is initialized. The standard catalogue is a compilation of several lists of stars used for pointing tests on various telescopes. The first part of the catalogue contains the selection of Perth70 stars used on the AAT; all the stars are magnitude 7-8 and lie approximately on a 20° grid, skewed to give uniform coverage in both HA and Dec.

A private star catalogue can be substituted, either at startup time via the command line or during a TPOINT session via the INST command. The original catalogue can be restored at any time by means of an INST command with no arguments.

A star catalogue is a sequential file of up to 80-character records. Here is an example record:

```
073716 00 00 51.781 +20 23 22.02 +0.0051 -0.146 1950.0
```

The first six characters of each record are an alphanumeric star identifier. The remainder of the record is the mean RA/Dec (hours, minutes, seconds, degrees, arcminutes, arcseconds), RA/Dec proper motions (seconds and arcseconds per year) and equinox (optionally preceded by 'B' or 'J' to distinguish between the FK4 and FK5 systems). Free-format input decoding is used. The file is terminated by 'END' or end-of-file. See the INCLUDE file STARS for the current maximum allowed number of records.

The standard TPOINT syntax rules apply to such catalogue files, which may thus contain blank lines and comments (beginning with '!') to enhance readability. The INST command merely reads the records; actual decoding of the information in the catalogue is deferred until a star is to be looked up, and errors will not come to light until then.

4.4 Saving and Combining Processed Data

When the modelling possibilities of any one set of pointing data have been exhausted it will be necessary to combine the results of several runs if further systematic effects are to emerge from the noise. In general, the different runs cannot simply be concatenated (which is done with the APPEND command) because some of the terms will have changed – the collimation corrections in particular. The UNFIT command allows this difficulty to be overcome.

UNFIT applies a pointing model in reverse, taking the current adjusted positions and predicting the observations that produce those adjusted positions when the model is applied. Thus a canonical model can be applied to a set of pointing residuals to yield new ‘observations’ which can be combined for further analysis. Two modes of operation are provided:

	UNFIT	to apply the current model
or	UNFIT N	to apply a null model (coeffs all zero)

If the star to telescope modelling option has been chosen (via the ADJUST command), the star positions are set and the telescope positions left alone. If the telescope to star option has been chosen, the reverse is true.

To understand UNFIT, think first of what the FIT command does. Assume that the modelling option ‘telescope to star’ has been chosen (the default). Under these circumstances, the FIT command takes the raw telescope positions and computes what coefficient values in the current model will give the smallest residuals, then applies the model to the ‘raw telescope’ positions to give the ‘adjusted telescope’ positions. The differences between these ‘adjusted telescope’ positions and the ‘star true’ positions are the pointing residuals. UNFIT leaves these pointing residuals at their current values, but applies the pointing model in reverse to generate fictitious ‘observations’ (in this case raw telescope positions) which are consistent with both the current model and the current residuals.

If the current residuals and coefficient values are both simply the result of fitting, UNFIT will not have any significant effect. However, if one or more of the coefficients is set to a different value, UNFIT will produce a changed set of observations which, if fitted with the same model, will reproduce the changed coefficient values. This is how different pointing runs can be combined. First, you fit each run to the full model in the normal way, to determine the residuals; then you set all the coefficients, one by one, to whatever standard values are thought suitable; then you invoke the UNFIT command, to make the residuals and the model match; and finally you write out the amended data by using the OUTDAT command:

OUTDAT *file*

where *file* is the name of the file to be written. Two or more amended runs can then be combined (by using APPEND followed by INDAT), and will match as long as the other coefficients were consistent from run to run. If the combined file is fitted with the normal model, the arbitrary coefficients will have approximately the values that were chosen for them, and the other coefficients will be intermediate between the the values produced by each of the two runs separately.

Very often, a null model (all coefficients zero) is required, and here the UNFIT N command is useful. UNFIT N has essentially the same effect as setting all the coefficients to zero by hand

and then executing an UNFIT command, except that it leaves the original coefficient values intact. UNFIT N is useful for recording pure residuals and eliminating the need for fitting with the original model when new terms are being investigated.

Before using OUTDAT you should set an appropriate caption with SETCAP.

4.5 Sample Data

Four examples of authentic pointing data are provided. The file UKST contains data from the UK Schmidt Telescope (1.2 metre aperture, fork equatorial). AAT15 contains data from the Anglo-Australian Telescope (3.9 metre, horseshoe equatorial) in the f/15 Cassegrain configuration. HALE contains data from the Palomar 5 metre telescope (horseshoe equatorial) in the f/17 IR configuration. MMT contains data from the central reference telescope of the Multiple Mirror Telescope (4.5 metre equivalent aperture, altazimuth). The UKST file refers to the standard star catalogue; the other three files use explicit star positions and are therefore self contained. The filenames are as follows:

	PC	VAX	Unix
UK Schmidt	UKST	TPDIR:UKST	/star/bin/examples/tpoint/ukst.dat
AAT	AAT15	TPDIR:AAT15	/star/bin/examples/tpoint/aat15.dat
Palomar	HALE	TPDIR:HALE	/star/bin/examples/tpoint/hale.dat
MMT	MMT	TPDIR:MMT	/star/bin/examples/tpoint/mmt.dat

The recommended analysis procedure, described in the earlier section *Getting Started*, begins as follows:

```

PLTON device

INDAT file

CALL EQUAT (equatorial)
or CALL ALTAZ (altazimuth)

FIT

CALL E6 (equatorial)
or CALL A6 (altazimuth)

```

It is then necessary to start trying obvious extra terms via the USE command – tube flexure TF, fork flexure FO, *etc.*, as appropriate. Example models (which in the case of the UKST and AAT are the ones actually in service) are supplied as procedures, yielding the results shown in the following table:

datafile	procedure	σ ''	terms
UKST	UKST	6.17	7
AAT15	AAT	1.26	10 fitted + 4 fixed
HALE	HALE	2.39	10
MMT	MMT	0.73	11

To reproduce, for example, the AAT result using a PC you would simply enter:

```
INDAT AAT15
```

```
CALL AAT
```

It is important to understand that these standard deviations are *a posteriori* and may not be realized in practice. They do not take into account long term stability, the effectiveness of any pre-observing calibration procedures, the correctness of the implementation of the model within the telescope control software, or the ability of the user interface to give the astronomer access to the available pointing performance.

4.6 Simulated Pointing Tests

When developing the pointing model of part of a telescope control system, refer to the source code for the `tpt_PTERMS` module for information on the mathematical form of `TPOINT`'s various pointing terms. The operational implementation of the terms may well involve techniques which differ in some way from `TPOINT`'s – perhaps by being more elaborate and general. Any substantial inconsistencies between `TPOINT` and the telescope software will manifest themselves as pointing errors, and telescope users will not experience the levels of performance which the `TPOINT` results appear to offer.

In order to check that the model as implemented in the telescope system is consistent with `TPOINT`, a “dummy” pointing test is recommended. This is like a real pointing test except that the telescope is left where it is after each blind setting rather than being guided onto the star. When the dummy pointing test is reduced with `TPOINT`, a perfect fit should be reported, and the coefficients should exactly match the values used by the control software. Without a test of this sort, errors of scaling and (especially) sign may go unnoticed.

Note that the dummy pointing test just described is a check on the software alone, with the telescope itself playing no part. Consequently, it is usually best to arrange that the control software can operate in a “simulation” mode, capable of generating a dummy pointing test offline. It is **strongly** recommended that dummy pointing tests be carried out from time to time, to verify that any revisions of the telescope software and `TPOINT` have not led to incompatibilities.

5 ADVANCED MODELLING

`TPOINT` allows a straightforward model for a telescope to be generated easily and without requiring great cunning or insight, but also provides the tools more experienced and determined users will need when conducting more exhaustive analyses.

As already described, the general strategy for modelling a telescope from scratch is to input a data file with `INDAT`, to set up a basic geometrical model using `CALL EQUAT` or `CALL ALTAZ`, and to fit using the `FIT` command, repeating as necessary until the coefficients settle down. With the aid of the `G...` commands, plots of residuals can then be made and inspected for signs of systematic error. Plausible flexure terms can be tried (`USE TF etc.`), and any others that the particular telescope design suggests.

When adding new terms, pay attention to the standard deviation of the new coefficient, the effect on other terms, and whether the population standard deviation has gone down; a new term will almost always reduce the RMS but population σ is a better indication of whether the improvement is real. If an attempt is made to use two terms which are more or less indistinguishable (for example CH and PXH0, which are identical), the condition will be reported; if this happens, get rid of one of them via the LOSE command. In such ill-conditioned cases, the singular value decomposition (SVD) methods used by FIT will normally return a meaningful model, with the “true” value of a duplicated term being shared out between the two (or more) coefficients involved. Two common cases are (i) if two inseparable terms are present from the start, they each receive an equal share of the value, and (ii) where a new term is added to an already fitted model, and that term is inadequately distinguished from something already present, then the new term will be left close to zero. This behaviour can be controlled by means of the FITTOL command, which allows the SVD ill-conditioned criterion to be set. A command FITTOL 0 will switch off the detection of ill-conditioning, whereupon FIT will return a conventional least squares fit, and any highly correlated terms will head off towards cancelling infinities. A FITTOL argument of between 10^{-3} and 10^{-2} will give a useful degree of control over ill-conditioning; the value is set to 10^{-2} initially. With FITTOL arguments much bigger than 10^{-2} , there is increasing danger that respectable terms will be excluded from full fitting. However, whenever FIT decides to take action over ill-conditioning, it reports that this has occurred.

Do not try to extract more information from a given sample of data than the size of the sample justifies. A preliminary assessment of a telescope’s capabilities can be made using perhaps 20-30 stars. Routine fitting of a typical operational telescope model, consisting of six geometrical terms, two or three flexures, and perhaps a few scale adjustments, centring corrections, *etc.*, calls for 50-100 stars, well-distributed over the whole observable sky. A campaign to expose high-frequency flexures and bumps should not be embarked upon without a sample of 500-1500 stars, obtained by combining multiple tests by means of the UNFIT and APPEND facilities. Even with this large number of stars, fitting a full set of polynomials or harmonics up to the required degree in one go is impractical, and new terms must be tried out in groups of no more than about 20.

When first analysing a new telescope, it is not uncommon to see obvious systematic residuals on the simple graphs of error in one coordinate plotted against that or another one – plotting error in zenith distance versus against zenith distance (G Z Z) for example. However, once these straightforward effects have been treated, systematic errors can still hide in what looks like noise. Some of these can be exposed by using more specialized plots – zenith distance errors against parallactic angle (G Z Q), errors in HA/Dec nonperpendicularity plotted against HA (G P H), and so on. However, certain effects which depend simultaneously on two coordinates may prove more stubborn; given a data set of adequate size, the solution is to use the MASK and UNMASK commands to select bands in one coordinate, allowing the residuals for that one band to be plotted.

At various stages, the GHYST command should be tried, to assess whether the current pointing accuracy seems to be limited by hysteresis.

When pursuing low-level flexures and irregularities, difficult questions arise. To what extent is the model physically meaningful? Has adding empirical terms affected the supposedly well-understood underlying model? Is it worth trying to keep a grip on physical reality, or wouldn’t it be just as good to fit a massive set of polynomials or harmonics?

An approach that seems to preserve the considerable advantages of having a model firmly based on mechanical reality, yet which allows all the power of empirical modelling to be unleashed, is to construct a model which includes both physical and empirical terms, but at no stage to leave both sets free to float at once during fitting. The following procedure is one way to do this:

1. Fit many pointing tests with the best available model; reject any obviously bad runs; UNFIT the rest onto a model with a fixed set of typical coefficient values; OUTDAT to a file; use APPEND to concatenate all these files; INDAT the result.
2. FIX everything except maybe one or two zero points.
3. USE a selection of empirical terms.
4. FIT; reject any disruptive or poorly-determined terms and FIT again. Repeat until only well-determined terms (say 2σ or better) remain.
5. Plot the residuals to see what extra terms might help.
6. Repeat from step 3 until further improvements seem unlikely.
7. FIX everything, un-fix the original model from step 1 by means of USE, and FIT.
8. Repeat from step 2 until neither the physical model nor the empirical terms are changing significantly.
9. Repeat from step 1 inserting into to the standard model the newly determined empirical terms **FIXed at the values just determined**. Depending on how the terms will be coded operationally, decide whether to flag them CHAINED or PARALLEL.
10. Once everything has stabilized, code the empirical model into one or more encapsulated local terms, using TRMLOC and PTERML. Include these terms in the standard model, with their coefficients FIXed at unity.

Such advanced modelling will require repeated fitting of large datasets. This will take time and be prone to mistakes. A good way to proceed is to pre-program the key steps using a private procedure library. Such a library can be edited during the TPOINT run by means of the SPAWN command, and then re-input with the INPRO command. A private library is also an option when running in batch mode; instead of specifying TPOINT commands in the batch control file, the latter can simply start TPOINT, read the procedure library with INPRO, and execute the required library routine by means of an appropriate CALL.

6 MAINTENANCE

6.1 Directories and Files

The TPOINT development system resides on the VAX and consists of a main directory and a subdirectory. The main directory contains the development system, and includes separate Fortran modules, command procedures to build the libraries, and so on. The subdirectory is called [.RELEASE] and contains just those files required to run the system, plus source in text libraries, and the L^AT_EX source for the Starlink User Note you are currently reading.

The executable system requires the SGS, GKS, GNS, GWM, PSX, CNF, CHR and EMS libraries. To relink TPOINT also requires the SLALIB and HELP libraries.

The following files are in the **development** directory:

READ.ME	general information
TPOINT.NEWS	news item on latest release
*.DAT	copied to the runnable system (see below)
SUN100.TEX	copied to the runnable system (see below)
*.FOR	TPOINT Fortran source modules common to all platforms
*.IND	platform-independent TPOINT Fortran source modules
*.VAX	VAX/VMS-dependent TPOINT Fortran source modules
*.PCM	PC/Microsoft-dependent TPOINT Fortran source modules
*.ASM	PC/Microsoft Assembler TPOINT source modules
*.SUN4	Sun SPARCstation/SunOS-dependent TPOINT Fortran source modules
*.HLP	HELP text
CREATE.COM	DCL procedure to create the TPOINT system from source
LTP.COM	DCL procedure to link TPOINT
LTP.OPT	For linking TPOINT on VAX
PUT.COM	DCL procedure to update a module in the libraries
TPOINT.COM	DCL procedure for starting TPOINT
TPOINT	C-shell script for starting TPOINT
MK	mk script for building Unix versions
MAKEFILE	makefile for building Unix versions
TPOINT.OLB	object library
PCTOVAX.COM	tailor PC/VAX release for VAX
*.BAT	for PC
LTP.LB	for PC
LTP.LNK	for PC
VAX_TO_UNIX.USH	C-shell script to produce a Unix archive file

The following files are in the **runnable system** directory:

AAT15.DAT	sample data – Anglo-Australian Telescope
UKST.DAT	sample data – UK Schmidt
HALE.DAT	sample data – Palomar 5 metre
MMT.DAT	sample data – Multiple Mirror Telescope
TPOINT.TLB	Fortran source modules
TPOINT.COM	command procedure to run the TPOINT program
PROCS.DAT	library of TPOINT procedures
STARS.DAT	star catalogue
TPOINT.SHL	HELP library – root
CMNDS.SHL	HELP library – “Commands” tree
MODEL.SHL	HELP library – “Model” tree
LOCAL.SHL	HELP library – “Local_Enhancements” tree
TPOINT.EXE	the TPOINT executable program
SUN100.TEX	L ^A T _E X source for SUN/100

See the file READ.ME for information about recent changes and instructions for setting up the SGS graphics system.

SUN100.TEX contains alternative page layout commands to suit either European A4 or American Quarto paper sizes. The file supplied is set up for A4; to produce the Quarto version simply make a private copy of SUN100.TEX, comment out the A4 line, and remove the comment character from the Quarto line.

6.2 Compiling and Linking

The procedure for modifying the TPOINT system is as follows:

1. SET DEFAULT to the development directory.
2. Modify the appropriate Fortran source module(s).
3. Use @PUT *module* to compile each module and update the source and object libraries.
4. Relink with @LTP.
5. Update the HELP source, and use @HELPPDIR:HLIB *.HLP [.RELEASE]*.SHL to create the help libraries.
6. Update the documentation.

After a series of changes it is wise to execute the CRE procedure in order to recreate the whole system from source.

6.3 New Commands

Special versions of TPOINT containing extra commands (for example to read observation data in local formats) can easily be generated. The essential details are as follows:

A command is implemented as an argument-less subroutine, conventionally of the same name as the command, prefixed with `tpt_`. Code to recognize the command name and to execute the appropriate subroutine must be added to the `tpt_CMDLOC` module. This is straightforward, and the `tpt_CMDLOC` implementation supplied with TPOINT contains example coding. Information flow in and out of the routine is by COMMON blocks, defined through a set of INCLUDE files. These files should be studied to see what information is available – command line, observations, model tables, *etc.* Prior to the command subroutine being called, the variable IOKF (defined in INCLUDE file CMD) will have been set to `'?'`, and the subroutine should either reset IOKF to a space on successful completion or left alone to indicate failure. The flag ABORT (also defined in CMD.INC) should be examined during lengthy applications; if ABORT becomes `.TRUE.` at any time, the application should be brought to a speedy but clean finish. This may involve making sure that tables are internally consistent and properly endmarked, and that the graphics system has been left in a tidy state. The best plan is to take one of the existing applications and to use it as a template for the new one.

n.b. Locally-implemented commands have to make assumptions about the location and meaning of the information in the COMMON blocks, and may also call TPOINT internal routines. The stability of these “unpublished” interfaces cannot formally be guaranteed.

6.4 Local Pointing Terms

Where it is found that the existing library of pointing terms is not enough to eliminate some observed or predicted systematic effect, new pointing terms can be added to the system. This can be done without altering any of the standard TPOINT modules, minimizing the work of re-implementing locally-defined terms each time a new release of the system occurs. The modules set aside for local terms are `tpt_TRMLOC`, which defines their names, and `tpt_PTERML`, which defines their form. (The module `tpt_TRMLOC` can also perform any special initialization actions required by a locally-implemented `tpt_PTERML` module.) The supplied versions of `tpt_TRMLOC` and `tpt_PTERML` contain examples, and they are both consistent with their counterparts `tpt_TRMSTD` and `tpt_PTERMS`, which perform the same role for TPOINT's standard set of terms. It is permissible to have local terms which are identical with standard ones in form and/or name; in the case of identical names, the locally defined term takes precedence over the standard one.

To introduce a new pointing term, the following changes to `tpt_TRMLOC` and `tpt_PTERML` are required:

Changes to `tpt_TRMLOC`:

1. Update the value given in the `NCOEFF` parameter statement.
2. Add the name of the new coefficient to the `CONAME` data statement.

Changes to `tpt_PTERML`:

3. Add a further destination to the large computed `GO TO` near the start of the module.
4. Insert, at the appropriate place, a `CONTINUE` with the label chosen in the previous step, followed by the expressions which calculate the pointing change per unit coefficient. Use existing terms (in `tpt_PTERML` and `tpt_PTERMS`) as a guide.

After these changes, follow the compiling and linking instructions given in the previous section.

Some implementors of telescope control systems have incorporated `tpt_PTERML` into their operational software, thereby ensuring that the model encountered by users is the same as the one that generated the impressive RMS pointing performance quoted for the telescope (a weak link at many observatories). To facilitate this practice, `tpt_PTERML` uses neither `COMMON` nor `INCLUDE` files, and carries a command and status argument which effectively does nothing in the standard version of the TPOINT system. TPOINT calls `tpt_PTERML` with a command of zero, telling private versions that normal TPOINT operation is required. The standard version of `tpt_PTERML` ignores the command and always returns a zero status, indicating success. A private version of `tpt_PTERML` should return a status which is zero or positive, to indicate success. The routines within TPOINT which use `tpt_PTERML` check the status and, if less than zero, flag the observation inactive, excluding it from further use until re-activated by means of the `UNMASK` command. The two TPOINT commands which do this are `FIT` and `OUTMOD`.

7 HISTORY AND ACKNOWLEDGMENTS

TPOINT grew out of the work John Straede and I did at the AAT between 1974 and 1980 using Interdata 70 computers. Most of the early least-squares software was written by John Straede,

culminating in the very successful IBORL2 program. I then developed a new program based on the IBORL2 fitting algorithm, but with improved interactive model building capabilities, extra graphics options, *etc.*; this program was the direct ancestor of TPOINT. In the early 1980s the program was ported from the Interdata to the VAX and then refined and expanded, exploiting the Starlink SGS/GKS and SLALIB packages as they became available. TPOINT underwent a major refurbishment in 1986/87 to prepare it for release through Starlink. During the period 1990-92 a version which could be run on both VAX/VMS and PC/MS-DOS was developed, followed by further revisions to include Unix platforms.

TPOINT would probably not exist had Peter Gillingham, Joe Wampler and Don Morton not encouraged continual refinement of the AAT pointing. Recent development of TPOINT has been greatly assisted by the many valuable comments and suggestions I have received from Steve Lee, Robert Laing, Ken Elliott, Hilton Lewis, Bob Kibrick, and Fang Yanling. I have worked closely with Russell Owen of the Apache Point 3.5 metre project on facilities for local additions to, and use of, TPOINT code, and on extensions to the modelling and graphical tools.

8 COMMAND REFERENCE

This section contains a description of every TPOINT command, in alphabetical order. A quick reference list is given at the end of the document.

<i>coeff</i>	Set/Inquire Coefficient Value	<i>coeff</i>
--------------	-------------------------------	--------------

FUNCTION :

The value of any of the coefficients in the current model may be set or inquired by using its name as the command.

COMMAND :

coeff value

where *coeff* is the name of the coefficient, and *value* is the value.

NOTES :

If no argument is supplied, the coefficient is unaltered – merely logged.

In most cases *value* is in arcseconds; a few terms have dimensionless coefficients.

ADJUST	Select Model Direction	ADJUST
---------------	------------------------	---------------

FUNCTION :

ADJUST allows one of two methods of applying the pointing model to be selected – whether the telescope positions are to be adjusted to fit the star positions or *vice versa*.

COMMAND :

ADJUST T	to adjust the telescope positions
or ADJUST S	to adjust the star positions

NOTES :

If no argument is supplied, the method is unaltered – merely logged.

If the ADJUST command results in the direction being changed, the order of the terms in the pointing model is reversed to match.

A given set of data will not produce precisely the same coefficients if the model is reversed. The differences will be small unless one or more of the coefficients is very large – hundreds of arcseconds or more. Under these circumstances there is in any case a danger that TPOINT's simple formulation of the pointing terms does not match exactly the way the model is implemented in the control software of the telescope concerned. Large collimation corrections, in particular, may require more rigorous geometry than TPOINT can conveniently offer, and detailed knowledge of field distortions.

The results of any previous FIT are not changed when ADJUST is used and, if a fresh FIT is to be carried out, the RESET command should be used first in order to eliminate the previous corrections.

ECHO Enable/Disable Command Echoing **ECHO**

FUNCTION :

ECHO enables or disables the echoing of commands to the screen during execution of library procedures.

COMMAND :

 ECHO ON enable command echoing
or ECHO OFF disable command echoing

NOTES :

The argument defaults to ON. However, echoing is disabled when TPOINT is started. Echoing is useful for debugging, but can clutter the screen on some graphics terminals.

END Terminate the Session **END**

FUNCTION :

END terminates TPOINT.

NOTE :

Additionally, QUIT, QU and CTRL/Z are all interpreted as END commands.

FIT Fit Model to Data **FIT**

FUNCTION :

FIT computes the coefficient values for the current pointing model which fit the observations best, and updates either the adjusted telescope coordinates or the adjusted star coordinates depending on which of these two options is in force (see the ADJUST command).

COMMAND :

 FIT to perform a normal fit
or FIT N to apply the current model without fitting

NOTES :

If two terms in the model are too highly correlated to be reliably distinguished, this is reported by FIT so that one of them can be eliminated from the model. The FITTOL command provides control over the treatment of such ill-conditioned cases, allowing the acceptance criterion used by the “singular value decomposition” fitting method used in FIT to be specified. If this criterion is set to zero, by means of a FITTOL 0 command, the fit is simply a conventional least-squares approximation, and any highly correlated terms

will be awarded meaningless, large and mutually cancelling values, accompanied as a rule by a poor overall RMS result. With a FITTOL parameter of (for example) 0.01, such correlated terms are kept under control and the model is sound.

If all the coefficients have been FIXed, the result of the FIT command is simply to recompute the residuals, and in this case FIT and FIT N are equivalent.

FITTOL	Set ill-conditioning tolerance	FIT
---------------	--------------------------------	------------

FUNCTION :

FITTOL specifies and reports the acceptance criterion for the singular value decomposition algorithm used in the FIT command.

COMMAND :

FITTOL *v*

where *v* is the tolerance used by the FIT command to decide whether action is to be taken to keep a poorly determined solution under control.

NOTES :

A tolerance value *v* between zero and 0.01 is recommended. The initial value is 0.01.

If no argument is supplied, the current tolerance is reported but not changed.

FIX	Fix the Specified Terms	FIX
------------	-------------------------	------------

FUNCTION :

FIX excludes one or more terms from the fit but not from the model.

COMMAND :

FIX *coeff1 coeff2 etc.*

where the arguments are the names of the coefficients to be frozen at their current values.

NOTES :

If no arguments are supplied, the whole model is fixed.

If any argument is unrecognized, the entire command is rejected.

GDIST Plot Error Distributions **GDIST**

FUNCTION :

GDIST plots distributions of the pointing residuals.

NOTE :

The histograms labelled X, D, S and Z are the components east-west, north-south, Left-Right and Up-Down; the histogram labelled R is for the total pointing errors.

GHYST Hysteresis Plot **GHYST**

FUNCTION :

GHYST attempts to expose hysteresis by assuming that the direction (and perhaps distance) from the current observation to the previous observation determine the direction (and perhaps size) of the effect. The format is a polar plot, with north at the top if the equatorial option has been selected, or up at the top if the altazimuth option has been selected. Each observation (except the first) is plotted as a square marker, from the centre of which comes a line indicating the pointing residual. The position of the marker shows in what direction the telescope had to move in order to travel from the previous star, and how far (on a logarithmic scale).

COMMAND :

GHYST *type scale*

The *type* is E or Q for equatorial and A or Z for Az/El. The equatorial plot has north at the top, whereas the Az/El plot has up at the top. The Q and Z options suppress an informational message which is normally sent to the command terminal, and are intended for use where the command terminal is of a type which does not have independent alphanumeric and graphic planes. This argument defaults to E.

The *scale* is the residual in arcsec to give a vector of nominal maximum length. This defaults to the largest actual residual providing it is reasonable.

NOTES :

A numerical estimate of the hysteresis is made and reported. It is obtained by rotating each pointing residual by the orientation of the preceding telescope movement, accumulating vectorially, and finally dividing by the number of active observations.

Note that various important but dubious assumptions are being made:

- The order of observations in the data list is the same as in reality.
- The telescope travelled directly from one star to the next, along a straight line in Cartesian Cylindrical coordinates.
- Any hysteresis comes from the large scale movements between observations and is unaffected by small scale adjustments during acquisition.

MASK Flag Selected Observations Inactive **MASK**

FUNCTION :

MASK flags selected pointing observations inactive. The selection is made by area of sky, total residual, or sequence number.

COMMAND :

MASK *quantity condition value*
or MASK *n1 n2*

where the arguments are as follows:

<i>quantity</i>	H (HA), D (Dec), A (Az), Z (ZD), R (radial error) or N (observation number)
<i>condition</i>	L (less than) or G (greater than)
<i>value</i>	cutoff value (degrees for H, D, A or Z; arcseconds for R; number for N)
or	
<i>n1</i>	first observation number
<i>n2</i>	last observation number (defaults to <i>n1</i>)

NOTES :

If the arguments are omitted, all observations are MASKed.

The inverse command is UNMASK.

Azimuths lie within the range 0 to 360 degrees, reckoned from north through east.

Observation numbers:

- start at 1 and refer to all observations, whether or not currently active;
- can be given in either order;
- can be less than 1 or greater than the current number of observations without ill effect.

EXAMPLES :

MASK H L 60	makes inactive all observations where hour angle is less than 60 degrees (4 hours)
MASK Z G 20	makes inactive all observations where zenith distance is greater than 20 degrees (elevation less than 70 degrees)
MASK N G 20	makes inactive all but the first 20 observations
MASK 3	makes inactive the 3rd observation

PARAL Apply Terms in Parallel **PARAL**

FUNCTION : PARAL is used to indicate which terms are to be evaluated in parallel to the immediately preceding term in the model. This allows several consecutive terms to use the same position as their starting point, rather than each successively adjusting the position before it is used by the next term.

COMMAND :

PARAL *coeff1 coeff2 etc.*

where the arguments are the names of the terms to share starting points with their predecessors.

NOTES :

If no arguments are supplied, all the terms in the model use the same starting point – the initial, unadjusted position.

If any argument is unrecognized, the entire command is rejected.

The starting position for a term named in this command will be the adjusted position produced by the last term which is ‘chained’ rather than ‘parallel’. Thus the first term in a group of terms sharing a common starting point will, in fact, be ‘chained’.

The converse command is CHAIN.

PENS Specify Pens **PENS**

FUNCTION :

PENS specifies and reports the pens used for plotting on the current workstation.

COMMAND :

PENS *f c a l p o*

where the arguments are the SGS pen numbers for:

f	frame
c	caption
a	axes
l	labels
p	points which are valid
o	points which are off-scale

NOTES :

Arguments default to the existing value. If any argument is invalid none of the pens are changed.

The actual colours and line styles are dependent on the implementation of the GKS graphics package being used. Note that although coloured pens may be used to produce both coloured lines and text, pens giving dotted or broad lines may not produce analogous effects on text.

PERFCT Create Ideal Observations **PERFCT**

FUNCTION :

PERFCT creates a list of artificial error-free pointing observations.

COMMAND :

PERFCT ϕ

where ϕ is the telescope latitude to which the artificial observations will correspond, given as degrees, arcminutes, arcseconds.

NOTE :

The arcminutes and/or arcseconds may be omitted. If the entire argument is omitted the latitude found in any existing data list is used. If the data list is empty, latitude zero is used.

PLTOFF Close Graphics Device **PLTOFF**

FUNCTION :

PLTOFF stops plotting on the current graphics workstation and reverts to the previous one (if any).

PLTON Open Graphics Device **PLTON**

FUNCTION :

PLTON stops plotting on the current graphics workstation (if any) and prepares to plot on a new one.

COMMAND :

PLTON *device*

where *device* is the SGS workstation name.

NOTES :

To obtain a list of the SGS workstation names supported on your machine, type PLTON with no argument.

A GKS workstation type is acceptable in lieu of an SGS name.

RETURN	Return from a Procedure	RETURN
---------------	-------------------------	---------------

FUNCTION :

RETURN returns from a library procedure.

NOTE :

A RETURN command issued from the command device results in an error message.

SETCAP	Set the Caption	SETCAP
---------------	-----------------	---------------

FUNCTION :

SETCAP sets the caption string.

COMMAND :

SETCAP *string*

where *string* is the required caption, and consists of either the whole of the command line following the command name but with leading and trailing blanks eliminated or, if the first non-space character following the command name is a ' or " delimiter, the string between that delimiter and the next appearance of the same one (unless the end of the command is reached first).

NOTES :

If no argument is supplied the caption is set to a single space.

If the caption contains lowercase characters, enclose it in single or double quotes.

SHOW	Display Parameters	SHOW
-------------	--------------------	-------------

FUNCTION :

SHOW displays and logs the current values of various TPOINT internal parameters.

NOTE :

The output from SHOW is a useful guide to some of the features of TPOINT, and includes the names of the commands used for changing the parameters listed.

SLIST	List the Observations	SLIST
--------------	-----------------------	--------------

FUNCTION :

SLIST lists the pointing observations and the current residuals.

NOTES :

The SLIST output includes the residuals in various directions rather than simply in radius.

If just the RMS residuals are needed the command 'REPLEN S' may be used to suppress reporting the individual observations.

The azimuths are reckoned from north through east.

SPAWN	Execute a Shell Command	SPAWN
--------------	-------------------------	--------------

FUNCTION :

SPAWN executes one or more shell commands (DCL on VAX/VMS).

COMMAND :

SPAWN *string*

where *string* is the required shell command, and consists of either the whole of the TPOINT command line following the 'SPAWN' name but with leading and trailing blanks eliminated or, if the first non-space character following the command name is a ' or " delimiter, the string between that delimiter and the next appearance of the same one (unless the end of the command is reached first).

NOTE :

If no argument is supplied a new shell is spawned, allowing a sequence of shell commands then to be entered. To return to TPOINT, issue a "logout" command.

TXF	Specify Text Font	TXF
------------	-------------------	------------

FUNCTION :

TXF specifies and reports the fonts for text plotting on the current workstation.

COMMAND :

TXF *c l*

where the arguments are the SGS font numbers for:

c	caption
l	labels

NOTES :

Arguments default to the existing value. If any argument is invalid (the font numbers must be greater than zero) none of the fonts are changed.

The actual fonts depend on the implementation of the GKS graphics kernel being used. See local GKS documentation for details.

TXP	Specify Text Precision	TXP
------------	------------------------	------------

FUNCTION :

TXP specifies and reports the text precision for the current workstation.

COMMAND :

TXP *n*

where *n* is the SGS precision:

	<i>quantity</i>	H (HA), D (Dec), A (Az), Z (ZD), R (radial error) or N (observation number)
	<i>condition value</i>	L (less than) or G (greater than) cutoff value (degrees for H, D, A or Z; arcseconds for R; number for N)
or	<i>n1</i>	first observation number
	<i>n2</i>	last observation number (defaults to <i>n1</i>)

NOTES :

If the arguments are omitted, all observations are UNMASKed.

The inverse command is MASK.

Azimuths lie within the range 0 to 360 degrees, reckoned from north through east.

Observation numbers:

- start at 1 and refer to all observations, whether or not currently active;
- can be given in either order;
- can be less than 1 or greater than the current number of observations without ill effect.

EXAMPLES :

UNMASK H L 60	makes active all observations where hour angle is less than 60 degrees (4 hours)
UNMASK Z G 20	makes active all observations where zenith distance is greater than 20 degrees (elevation less than 70 degrees)
UNMASK N G 20	makes active all but the first 20 observations
UNMASK 3	makes active the 3rd observation
UNMASK 1 25	makes active the first 25 observations
UNMASK	makes all the observations active

USE

Include Terms in Model

USE**FUNCTION :**

USE includes one or more terms in the pointing model, and flags it or them to be fitted.

COMMAND :

USE *coeff1 coeff2 etc.*

9 COMMAND SUMMARY

COMMAND	FUNCTION	DEFAULT
<i>coeff value</i>	set and report coefficient value	report only
ADJUST T or S	select model direction	report only
APPEND ON or OFF	next observations will [not] be appended	will be appended
CALL <i>proc</i>	call library procedure <i>proc</i>	-
CAPT ON or OFF	enable/disable plotting of captions	captions enabled
CHAIN <i>c1 c2 etc.</i>	selected terms are applied sequentially	whole model chained
CLIST	list current coefficients	-
CLRNG ON or OFF	enable/disable screen/zone clearing	clearing enabled
ECHO ON or OFF	enable/disable procedure command echo	echo on
END	exit	-
FIT N	fit [or apply model without fitting]	fit
FITTOL	specify and report ill-conditioning tolerance	report only
FIX <i>c1 c2 etc.</i>	exclude selected terms from fit	fix all terms
FRAME ON or OFF	enable/disable plotting of frames	frames enabled
G <i>y x s</i>	plot residuals	autoscale
GAM <i>t s</i>	look for axis misalignment	equatorial, autoscale
GCLR	clear display surface	-
GDIST	plot distributions of residuals	-
GHYST <i>t s</i>	look for hysteresis	equatorial, autoscale
GMAP <i>t s</i>	Cartesian Cylindrical plot	equatorial, autoscale
GSCAT <i>t r</i>	scatter plot	equatorial, autoscale
GSMAP <i>s</i>	orthographic map	autoscale
HELP <i>topic</i>	enter HELP session	enter at top level
INDAT <i>file</i>	read file of observations	-
INMOD <i>file</i>	read model from a file	-
INPRO <i>file</i>	read library file	original library
INST <i>file</i>	read star catalogue	original catalogue
LOSE <i>c1 c2 etc.</i>	remove selected terms from model	discard all terms
MARKH <i>h</i>	set marker height	0.2
MASK <i>q c v</i> or <i>n1 n2</i>	flag selected observations inactive	mask all observations
OUTDAT <i>file</i>	write file of observations	-
OUTMOD <i>file</i>	write model to a file	-
PARAL <i>c1 c2 etc.</i>	terms <i>c1 c2 etc.</i> are applied in parallel	whole model parallel
PENS <i>f c a l p o</i>	specify and report pens	report only
PERFCT <i>° ' "</i>	create ideal observations	ϕ from data list
PLTOFF	revert to previous graphics device	-
PLTON <i>device</i>	open new graphics device	list of devices displayed
PLTZ <i>z</i>	select plotting zone	use whole display surface
PROC <i>x</i>	start of library procedure <i>x</i>	-
REPLEN S or L	specify report length option	full length reports
RESET	zero coefficients and cancel corrections	-
RETURN	return from a procedure	-
SETCAP <i>string</i>	specify caption	caption blank
SHOW	display system parameters	-
SLIST	list the observations	-
SPAWN <i>string</i>	execute shell command	spawn new shell
TXF <i>c l</i>	specify text fonts	report only
TXP <i>n</i>	specify text precision	report only
UNFIT N	apply pointing model in reverse	current model applied
UNMASK <i>q c v</i> or <i>n1 n2</i>	flag selected observations active	unmask all observations
USE <i>c1 c2 etc.</i>	include selected terms in model	unfix all terms
VT <i>t b</i> or N	declare whether terminal VT100	VT100, whole screen
CTRL/C	abort current command	