

LONDON, METEOROLOGICAL OFFICE.

Met.O.19 Branch Memorandum No.66.

Automatic grid-coastline overlay for APT  
images: a FORTRAN program for the use in  
AUTOSAT. By EYRE, J.E. and CHAPMAN, D.E.

London, Met.Off., Met.O.19 Branch Mem.No.66,  
1982, 31cm.Pp.15.1 Ref.

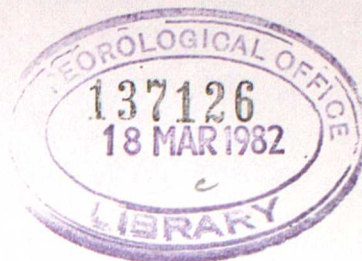
An unofficial document - restriction on  
first page to be observed.

FGZ

National Meteorological Library  
and Archive

Archive copy - reference only





MET O 19 BRANCH MEMORANDUM NO. 66

AUTOMATIC GRID-COASTLINE OVERLAY FOR APT IMAGES:

A FORTRAN PROGRAM FOR USE IN AUTOSAT

by

EYRE J.R. and CHAPMAN D.E.

March 1982

Met O 19 (Satellite Meteorology Branch)  
Meteorological Office  
London Road  
Bracknell  
Berkshire RG12 2SZ

NOTE:

This paper has not been published. Permission to quote from it should be obtained from the Assistant Director of the above Meteorological Office Branch.



AUTOMATIC GRID-COASTLINE OVERLAY FOR APT IMAGES:  
A FORTRAN PROGRAM FOR USE IN AUTOSAT

J.R. Eyre

D.E. Chapman

March 1982

1. Introduction

Satellites in the TIROS-N series carry the imaging instrument AVHRR (Advanced Very-High Resolution Radiometer) which senses radiation in the visible and infrared window regions with a ground resolution at the sub-satellite point of about 1 km (see Schwalb, 1978). Digital AVHRR data are processed on board the satellite (as explained in section 3) in order to:

- a) reduce the horizontal resolution and hence the data rate, and
- b) "linearize" each scan line, making the ground resolution more uniform and thus aiding interpretation of the images.

These data are then transmitted by the satellite in analogue form - the APT (Automatic Picture Transmission) signal.

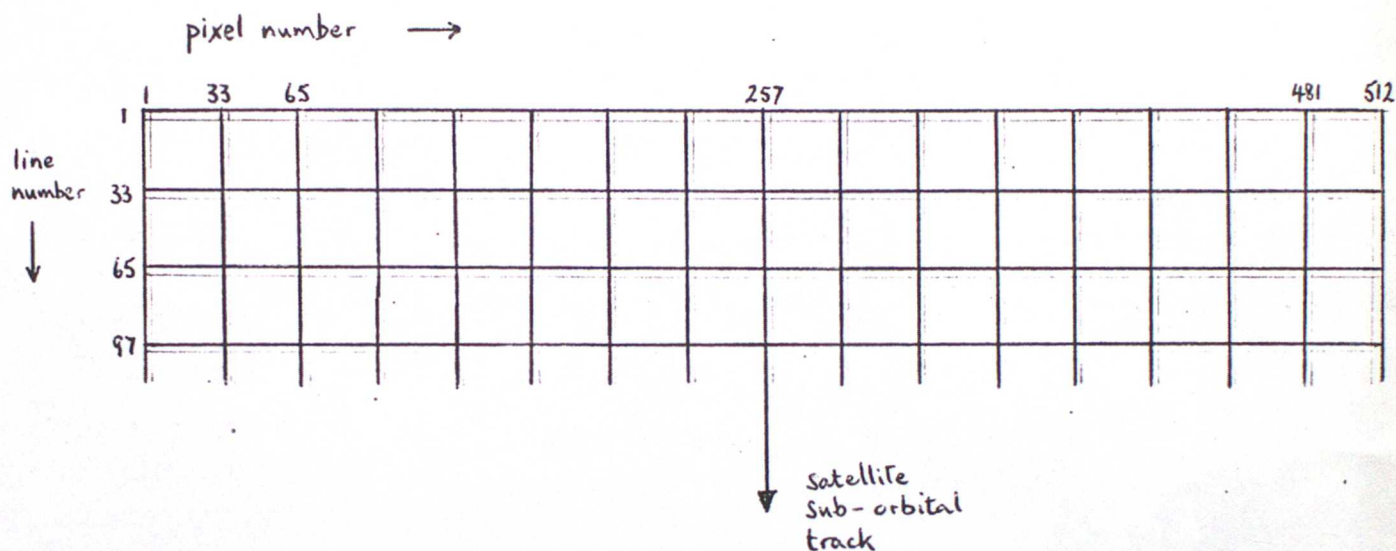
The data are received at Lasham and transmitted directly to the AUTOSAT computer where each line is re-digitized and stored as 512 8-bit values. A typical APT image is about 1800 lines long.

A FORTRAN program has been written to create a data set containing a mask of latitude-longitude grid and/or coastline information for automatic overlay on the APT images. Corresponding to each line of image data, the program provides 512 1-bit values - 1 for the presence of a grid-coastline point and 0 for its absence. An example of the output data set, displayed in graphical form, is shown in figure 1. The program has been designed to run sufficiently quickly to be operationally useful.



The following sections give a functional description of the program. The appendices contain additional details on the program structure, the function of each subroutine, program input/output requirements and information needed for running the program on COSMOS. The program is described in the form delivered to Met 0 5 in November 1981. In February 1982 the coastline data set used by the program was replaced by one which also included data for labelling selected latitude and longitude lines.

## 2. The box structure of the overlay mask



In order to save time, latitude-longitude co-ordinates are not calculated for each pixel in the image. Instead the image is divided into boxes, and earth-location calculations are only performed for the "top left-hand" pixel (as shown above) in each box. Three adjacent boxes provide information at the other three corners. A box size of 32 x 32 APT pixels has been found to be suitable. For each row of boxes, the latitude and longitude of the corners are calculated and then the 16 boxes are processed sequentially as described below. The program then outputs the overlay for these 32 lines and proceeds to the next row of boxes.

Inside each box the known latitude-longitude co-ordinates of points on the grid and coastline are converted to line-pixel co-ordinates using the transformations:

$$\begin{aligned} \text{pixel number} = p &= u_1 + u_2 \cdot \text{lat} + u_3 \cdot \text{long} + u_4 \cdot \text{lat} \cdot \text{long}, & \dots 1 \\ \text{line number} = l &= v_1 + v_2 \cdot \text{lat} + v_3 \cdot \text{long} + v_4 \cdot \text{lat} \cdot \text{long}. & \dots 2 \end{aligned}$$



The coefficient vectors,  $\underline{u}$  and  $\underline{v}$ , are found for the box by fitting the above equations exactly at the 4 corners.

$$\underline{p} = \underline{A} \cdot \underline{u}, \quad \dots 3$$

$$\underline{l} = \underline{A} \cdot \underline{v}, \quad \dots 4$$

where, at the  $i$ th corner of the box,

line number =  $l_i$ , pixel number =  $p_i$ ,

$A_{i1} = 1$ ,  $A_{i2} = \text{lat}_i$ ,  $A_{i3} = \text{long}_i$ ,  $A_{i4} = (\text{lat} \times \text{long})_i$ .

$\underline{u}$  and  $\underline{v}$  are found by inverting these equations:

$$\underline{u} = \underline{A}^{-1} \cdot \underline{p} \quad \dots 5$$

$$\underline{v} = \underline{A}^{-1} \cdot \underline{l} \quad \dots 6$$

This method is used except where rounding errors involved in the matrix inversion procedure are too great (when the determinant of  $A$  is small). This occurs when the latitude-longitude grid intersects the line-pixel grid at or very close to  $45^\circ$ . Empirically it was found that, for this particular application (with latitude and longitude expressed in degrees and a grid size of  $32 \times 32$  APT pixels), such errors can be important if the determinant is less than about 4. Therefore, when  $\det A < 4$ ,  $u_4$  and  $v_4$  are set to zero and the other coefficients are found by fitting equations 1 and 2 at only 3 corners of the box. This leads to small errors near the 4th corner but they are found to be insignificant.

In general the accuracy of the usual approach (fitting exactly at all 4 corners) is fundamentally limited by the way in which equations 1 and 2 approximate the true relationship between the co-ordinate systems. The maximum error at points away from box corners arises when the transformation coefficients underestimate the curvature of a latitude line (projected on to line-pixel co-ordinates) a  $32 \times 32$  APT pixel box the error is less than 0.3 km at  $50^\circ\text{N}$  and less than 1.1 km at  $80^\circ\text{N}$ . Therefore the errors from this source are not only smaller than those expected from satellite location errors but also less than the APT pixel resolution ( $\sim 4$  km).



### 3. Earth location

In order to calculate the earth location of a point in the APT image, the processing carried out on the satellite and by AUTOSAT must be taken into account. AVHRR generates its images using a scan mirror rotating at 360 r.p.m. However only one in three scan lines generated is used in the APT image (thus reducing the horizontal resolution along the satellite track by a factor of three). Along each scan line used the following averaging is performed to produce an image of more uniform ground resolution. For each half of the scan line, either side of nadir, 5 regions are used with a different averaging process in each:

Region	Range of nadir angles	Averaging process	AVHRR samples	APT samples
1	$0^{\circ} \rightarrow 16.98^{\circ}$	average 4 contiguous samples	314	78.5
2	$16.98^{\circ} \rightarrow 34.83^{\circ}$	average 2 samples; skip one; repeat	330	110
3	$34.83^{\circ} \rightarrow 43.81^{\circ}$	average 2 samples	166	83
4	$43.81^{\circ} \rightarrow 48.84^{\circ}$	average 2 samples to give a resolution of $1\frac{1}{2}$ AVHRR samples as follows: $\frac{1}{2}(a+b)$ , $\frac{1}{2}(b+c)$ , $\frac{1}{2}(d+e)$ , $\frac{1}{2}(e+f)$ , etc.	93	62
5	$48.84^{\circ} \rightarrow 55.34^{\circ}$	retain original resolution	121	121
Totals: $\frac{1}{2}$ line			1024	454.5
1 line			2048	909

Therefore 2048 AVHRR samples are reduced to 909 "linearized" APT samples. These are transmitted from the satellite in analogue form as the APT signal. The image section of each APT line, when received, is sampled and digitized at 524 points. Of these only 512 are stored, losing 6 samples off each edge of the picture.

Using the above information, one can construct a look-up table which relates pixels in the 512-length line held in AUTOSAT to original AVHRR scan angle. Such a table is used to calculate the scan angles for the pixels at the box corners, i.e. at pixel numbers: 1, 33, 65, ..., 449, 481, 512.



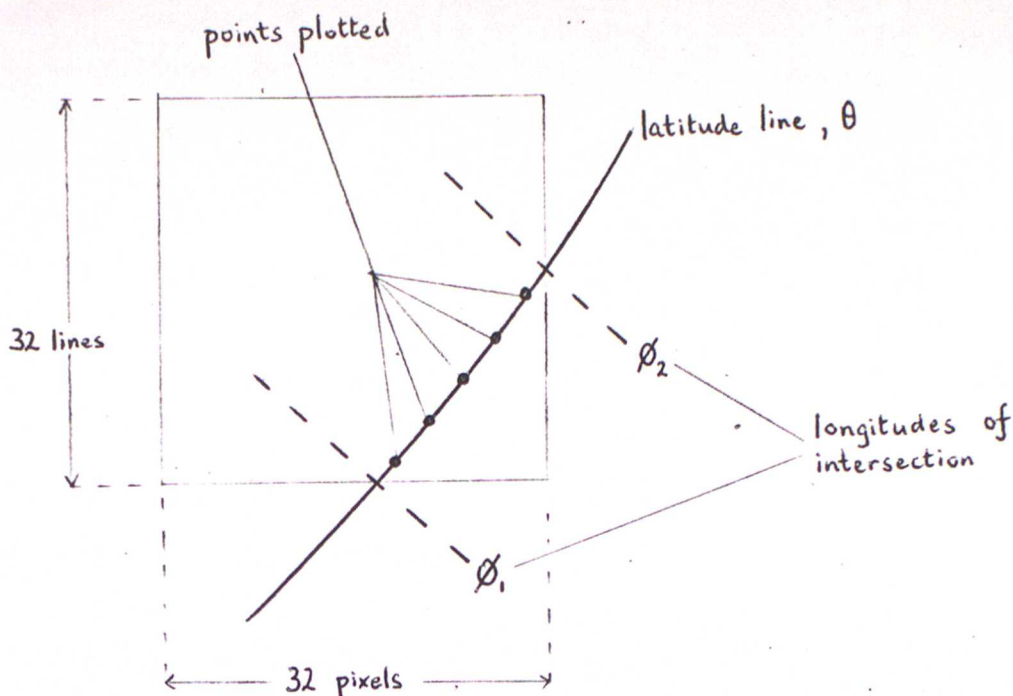
The scan angle, time of observation and parameters describing the satellite's orbital geometry are then used to calculate the latitude and longitude of the point on the earth viewed by the radiometer at this time and angle. (The program uses the subroutine LOCAT to perform this conversion. The parameters of the satellite's orbital geometry are input parameters to the program and are obtained from TBUS messages received via the Global Telecommunications System from the U.S.A. Problems with the accuracy of LOCAT have not yet been fully overcome or understood. However LOCAT could easily be replaced by a more accurate routine if necessary).

#### 4. Calculation of latitude - longitude overlay

For each 32 x 32 pixel box the following process is used to find the latitude and longitude lines which cross the box and to set appropriate bits in the overlay bit-plane to represent them. Taking each side of the box in turn, the latitude-longitude co-ordinates at the corners are used to identify which latitude lines (at  $5^\circ$  intervals) intersect the side. If such a latitude line is found:

- a) the transformation coefficients for the box are calculated as described in section 2,
- b) the 2 sides of the box intersected by the line are identified,
- c) the longitudes of the points of intersection are found using equation (1) or (2) as appropriate (in which, for a given box side, all variables are known except longitude),
- d) line-pixel co-ordinates in the overlay grid are then calculated using the transformation coefficients. Points are spaced along the latitude line between the 2 longitudes of intersection at intervals of  $k \cdot \sec \theta$  degrees of longitude, where  $k$  is a constant,  $\theta$  is the latitude and the factor,  $\sec \theta$ , is included to make the spacing of the points on the picture independent of latitude.  $k$  was chosen empirically as 0.13 degrees to give a spacing  $\sim 3$  pixels between points.





Longitude lines are calculated in a similar way, with a spacing between points chosen as 0.1 degrees of latitude.

##### 5. Calculation of coastline overlay

The coastline for the European-N. Atlantic area is represented by  $\sim 3000$  points in a form suitable for a "point-join" coastline plotting routine, i.e. each point carries information to indicate whether it should be "joined" to the previous point in the data set. The method used involved searching this data set to find those points which lie in the 32 x 32 pixel box being processed. To minimise searching time, the coastline data set is stored in sections which refer to  $5^\circ$  latitude x  $5^\circ$  longitude "squares". The co-ordinates of the box corners are examined in order to establish which "squares" are to be searched. Line-pixel co-ordinates in the overlay grid are calculated for each point found (within the box) in the coastline data set and also for additional points linearly interpolated between them. The spacing of interpolated points could be chosen to suit requirements but has been set at 3 APT pixels. Points are converted from latitude-longitude co-ordinates to line-pixel co-ordinates using the coefficients pre-computed for the box (see section 2).

The coastline data set has been extended to include data for annotation of selected lines on the latitude-longitude grid, the numbers being expressed in the form of "pseudo-islands".



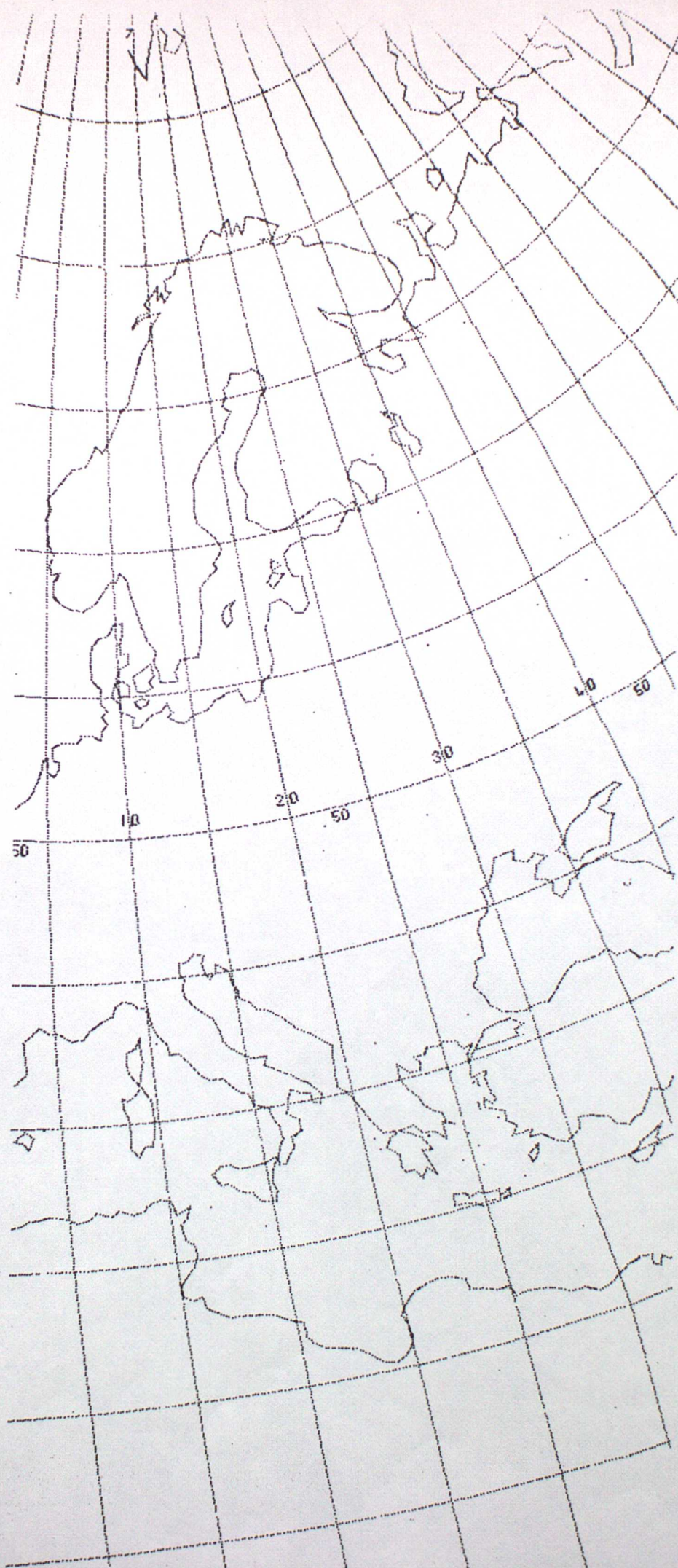
# REFERENCES

Schwalb A.      1978      NOAA Technical Memorandum NESS 95.  
The TIROS-N/NOAA A-G Satellite Series.



FIGURE 1.

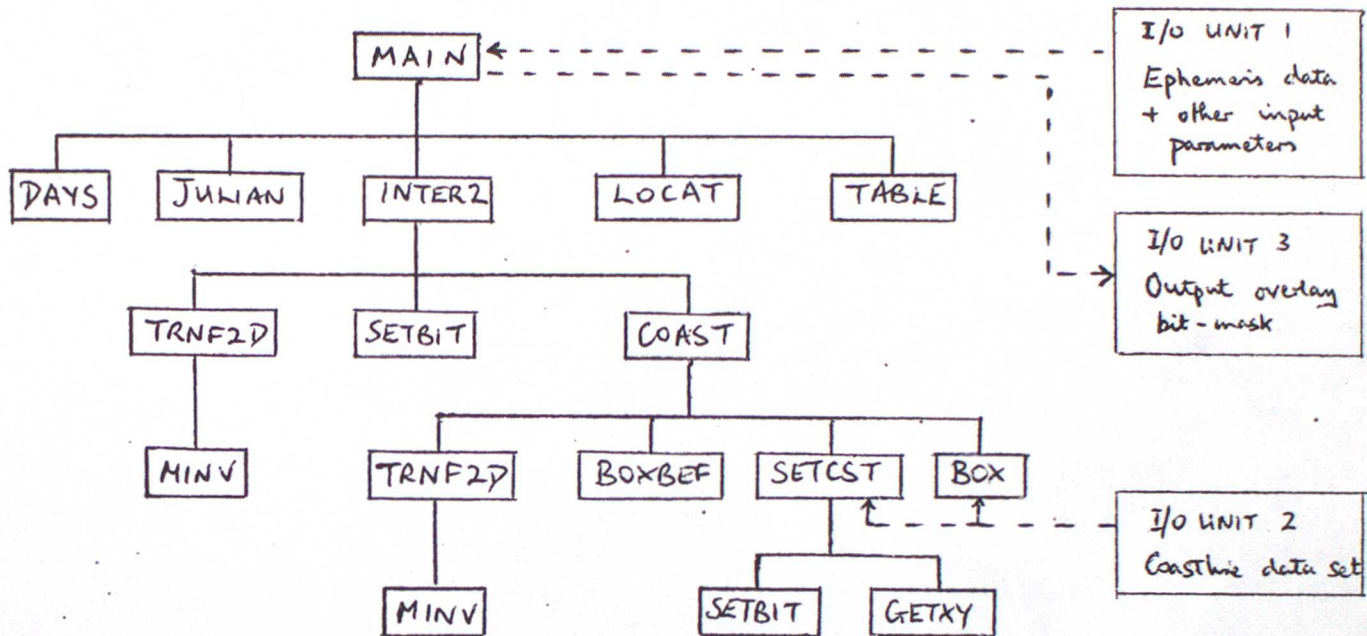
AN EXAMPLE OF A COMPUTED OVERLAY





## Appendix A

### Structure of APT overlay program





## APPENDIX B

### Subroutine descriptions

#### MAIN program

1. Calls TABLE to create look-up table for conversion of position on scan line to scan angle (for box corners).
2. Reads in ephemeris data and other input parameters on I/O unit 1.
3. Converts start time of pass into time suitable for input to LOCAT using subroutines DAYS and JULIAN.
4. For each set of 32 lines:
  - calculates earth location of corners using LOCAT,
  - calls INTER2 to calculate the overlay for this line of 16 boxes,
  - writes out 32 lines of overlay bit-mask to I/O unit 3.

Routines required: see appendix A

#### Subroutine TABLE

Creates look-up table for conversion of position on scan line to scan angle.

#### Subroutine JULIAN

Converts the observation time (in days and decimal parts of a day) to Julian days minus 2,400,000.5 for use by LOCAT.

#### Subroutine DAYS

Converts month and days into days from beginning of year.

#### Subroutine LOCAT

Calculates the latitude and longitude of a point on the earth given the parameters of the satellite's orbital geometry, the observation time and the observation nadir angle (in a plane perpendicular to the orbital plane). A full discussion of the methods used in this routine are outside the scope of this note.



### Subroutine INTER 2

For each of 16 boxes containing 32 x 32 pixels:

- Calls COAST to calculate the bit-mask for the coastline points.
- Then calculates the bit-mask for the latitude-longitude grid using the method described in section 4.

Requires routines: COAST, TRNF2D, MINV, BOX, BOXBEF, SETCST, SETBIT, GETXY.

### Subroutine COAST

Controlling subroutine for the calculation of the coastline overlay bits for a box. Processes the corners of the box sequentially and for each:

- a) calls BOX to find which "square" it is in,
- b) calls BOXBEF to check whether this "square" has already been processed,
- c) if the transformation coefficients for the box have not already been calculated, calls TRNF2D to calculate them,
- d) calls SETCST to set the required bits in the array storing the bit-mask.

Requires routines: BOX, BOXBEF, TRNF2D, MINV, SETCST, GETXY, SETBIT.

### Subroutine BOX

Calculates the  $5^{\circ} \times 5^{\circ}$  latitude-longitude "square" to which a box corner belongs and sets pointers to the appropriate section of the coastline data set.

Reads coastline data set from I/O unit 2.

### Subroutine BOXBEF

Checks if this "square" of the coastline data set has already been processed for this box.



### Subroutine TRNF2D

A general routine to calculate the transformation coefficients, u and v, for transforming a point in one frame to co-ordinates in another given four points specified in both frames. Transformation coefficients stored in COMMON/UVCOF/. The routine is explained in section 2 in terms of its use in this program for finding the transformation coefficients for each box.

Requires routine: MINV

### Subroutine MINV

A general purpose matrix inversion routine.

### Subroutine SETCST

A routine to set the required bits in the array storing the bit-mask to value = 1 to represent coastline. Sets bits at points corresponding to those in the coastline data set and those linearly interpolated between them with a spacing not greater than ISTEP (value set externally).

Requires routines: GETXY, SETBIT

Reads coastline data set from I/O unit 2.

### Subroutine GETXY

Transforms a point given in latitude-longitude co-ordinates to line-pixel co-ordinates using coefficients stored in COMMON/UVCOF/ (see section 2).

First checks if latitude < 0. If it is, this signifies a break in the coastline data set (i.e. a "lift-pen" instruction in a point-join coastline plotting routine). In this case, adds 100° to give the correct latitude and sets flag to indicate break in coastline (flag used in SETCST).

### Subroutine SETBIT

Sets to 1 the Jth bit of an array.

Also, Entry GETBIT

Finds the value of the Jth bit of an array and puts it in an I\*4 variable.



## APPENDIX C

### Input/Output description

I/O UNIT 1:      Ephemeris data and other input parameters: sequential, unformatted

a) Ephemeris Data:      EPH(1) - EPH(24)      48 byte block

EPH(1)	-	EPH(4)	Days from 0.0 January 1976 to epoch	
EPH(5)	-	EPH(8)	Mean motion	(revs. day <sup>-1</sup> )
EPH(9)	-	EPH(12)	0.5 x acceleration of mean motion	(revs. day <sup>-2</sup> )
EPH(13)	-	EPH(14)	Eccentricity	
EPH(15)	-	EPH(16)	Right ascension of ascending node	(degrees)
EPH(17)	-	EPH(18)	Argument of perigee	(degrees)
EPH(19)	-	EPH(20)	Orbital inclination	(degrees)
EPH(21)	-	EPH(22)	Mean anomaly at epoch	(degrees)
EPH(23)			Data set number	} not used by program
EPH(24)			Orbit number	

b) Header information, etc:      TDAT(1) - TDAT(15)      30 byte block

TDAT(1)	}	Spare	}	Observation time for start of pass
TDAT(2)				
TDAT(3)		Year		
TDAT(4)		Month		
TDAT(5)		Day		
TDAT(6)		Hour		
TDAT(7)		Minute		
TDAT(8)		Second		
TDAT(9)		Ticks		
TDAT(10)		Ticks per second		
TDAT(11)		Number of APT lines (usually about 1920)		
TDAT(12)		Satellite letter and overlay indicator (two INTEGER*1 bytes).		
		Overlay indicator: 1 = grid, 2 = coastline, 3 = both		
TDAT(13)	}	Spare		
TDAT(14)				
TDAT(15)				

EPH and TDAT are INTEGER\*2



I/O UNIT 2: Coastline data set

Direct access: 31 records of 512 bytes

RECORD 1: 28 bytes used: NTOT, MAX, IYSTEP, IXSTEP, L1, L2, NX.

NTOT: I\*4: total number of lat-long. points in data set

MAX: I\*4: total number of "squares" (= 32 x 11 = 352)

IYSTEP: I\*4: length of a "square" in degrees of lat. (=5)

IXSTEP: I\*4: length of a "square" in degrees of long. (=5)

L1: I\*4: reference lat. in degrees E (= -80)

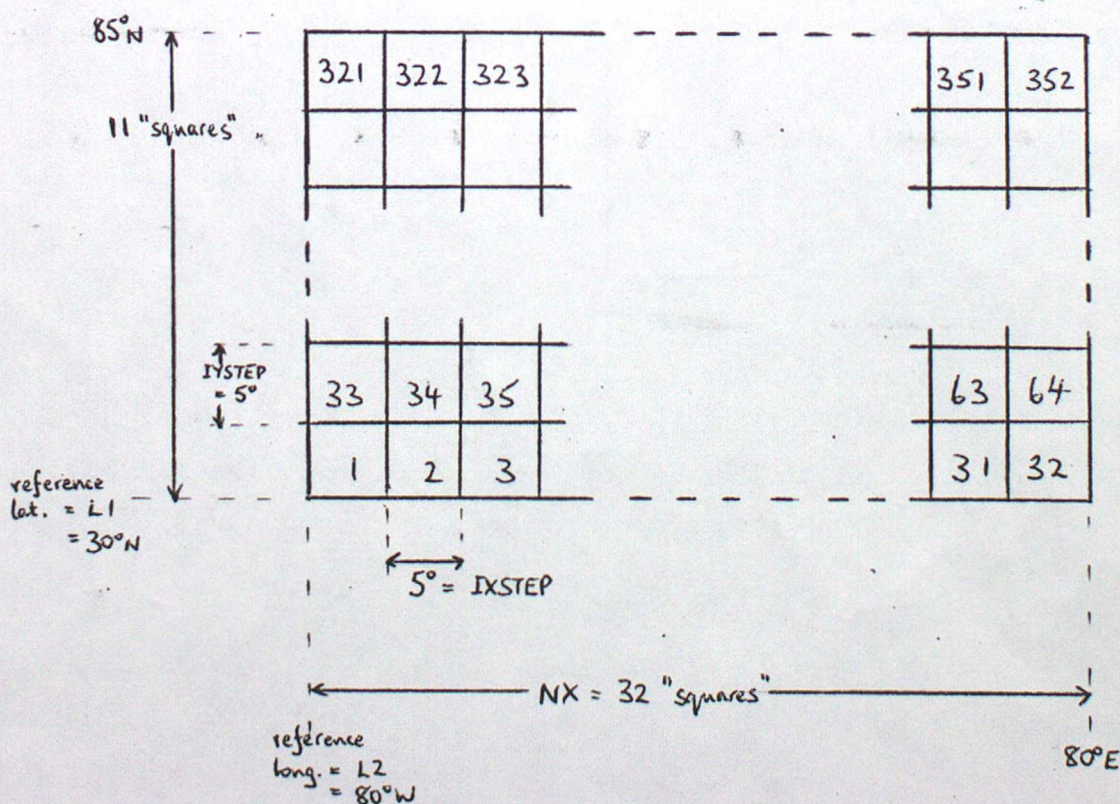
L2: I\*4: reference long. in degrees N (=30)

NX: I\*4: number of "squares" in E-W direction (=32)

RECORDS 2-3: EL(512): I\*2

The nth element (n=1→MAX) contains the relative address of the first I\*2 word of the coastline in the nth "square", where the "square" numbering is shown below and the address of the start of "square" 1 is the start of record 4.

EL(MAX+1) → EL(512) are unused.





RECORDS 4-31: NTOTx2x2 bytes. Contains NTOT pairs of lat.-long points for  
MAX "squares".  
Each record contains 128 x (lat., long.) in I\*2 form.



I/O UNIT 3:      Output data set for grid-coastline overlay  
sequential, unformatted.

RECORD 1:          TDAT(15): I\*2  
Header information - see I/O UNIT 1 for description.

RECORD 2 onwards: Each record contains a LOGICAL\*1 array,  
BITMSK (64,32) = 1024 bytes.  
This is 32 x 512 bits representing the grid-coastline overlay  
for 32 APT lines. The ordering of points is matched to that  
of the APT image, i.e. along the direction of the orbital track  
and scanning from right to left.



## APPENDIX D

### Running the program on COSMOS:

#### Guide to data sets required - March 1982

1. The FORTRAN source code is on M19.SRCCELIB in packed form, members:

EAUTOST#, EMETO5#, ETRIN#, EMINV#, SSS#.

The program can be submitted from TSO using T19PB.JCL.CNTL(EAPT).

2. I/O data sets (see also appendix C)

I/O UNIT:

- 1 Ephemeris data, start time and other input parameters. Must be created as an unformatted data set of two logical records.
- 2 Coastline data set.  
Use either M19.SCOAST.APT (coastline only)  
or M19.SCOAST.APTNUM (coastline plus annotation of latitude-longitude grid)
- 3 The program writes unformatted to the grid-coastline overlay output data set. Typically requires  $\approx$  100K bytes of storage.

3. Ancillary programs

- a) Program to create input data set (for I/O unit 1). Source code on M19.SRELIB (EPHEM#). Submitted from TSO using T19PB.CNTL (EPHEM), suitably edited.
- b) Program to create plot of grid-coastline overlay on CALCOMP. Source code on M19.SRCCELIB (EFILM#). Also requires object module MET.CALCOMP (OPENON35). Submitted from TSO using T19PB.JCL.CNTL(EFILM).
- c) Program to create coastline data set (with added points for the annotation of the lat.-long. grid). Source code on M19.SRCCELIB(SCSTNUM). Requires input data sets (on SYS006):  
I/O unit 27: M19.SGRIDNUM (grid annotation data set)  
I/O unit 28: M19.UCHART (master coastline data set)  
Creates M19.SCOAST.APTNUM on SYS006 (SPACE=(512,(31)) ).
- d) Program to create lat.-long. grid annotation data set, M19.SGRIDNUM, on SYS006. Source code on M19.SRCCELIB (SGRIDNUM).