Communications Research Centre

THE CRC TOPOGRAPHIC DATA BASE

by

J.H. WHITTEKER

CRC REPORT NO. 1353

Department of Communications
Ministère des Communications

OTTAWA, FEBRUARY 1982

COMMUNICATIONS RESEARCH CENTRE

DEPARTMENT OF COMMUNICATIONS
CANADA

THE CRC TOPOGRAPHIC DATA BASE

by

J.H. Whitteker

(Radar and Communications Technology Branch)

CRC REPORT NO. 1353

February 1982
OTTAWA

NOTE

Readers who wish only to learn how to plot path profiles are referred to Section 6.

Those who want a summary of the subroutines that are available are referred to Appendix A.

TABLE OF CONTENTS

THE CRC TOPOGRAPHIC DATA BASE

by

J.H. Whitteker

ABSTRACT

A data base of terrain elevations and surface types has been compiled, primarily to provide input data for radio-wave prediction programs. The data have been derived by hand-scaling 1:50000 scale topographic maps. An elevation and a surface code (forest, lake, etc.) is stored for each point in a square array with 500 metre spacing. The data reside on computer disk, and are organized for rapid access. The usual method of obtaining information from the data base is by means of a Fortran subroutine that provides elevations and surface codes at 500 m intervals along any specified great-circle path within the data-base area. The elevations may also be plotted on an ordinary (alphanumeric) computer terminal.

1. INTRODUCTION

1.1 REQUIREMENT FOR A TOPOGRAPHIC DATA BASE

1.1.1 Need for Machine-Readable Topographic Data

The topography of Canada is quite well known. Topographic maps may be obtained from the Canada Map Office of the Department of Energy, Mines, and Resources (EMR) that show terrain elevations and other information for any part of the country (1). However, if a computer is used to make calculations involving the shape of the terrain, there is a serious mismatch between the speed of the calculations and the speed with which a human can obtain elevations from a map and provide them to the computer. Therefore it is a great advantage to have these data already in machine-readable form.

1.1.2 Predictions of Radio-wave Field Strength

In particular, there is a need for a terrain data base to provide input for predictions of radio-wave field strengths at Very High and Ultra High Frequencies (VHF and UHF). A computer program for making such predictions has been developed at CRC (2); the data base described here has been developed for use with this program. However, the data base can be used independently. Potential applications include preliminary surveys of line-of-sight paths, or of sites proposed for satellite earth stations.

1.1.3 Digital Maps

EMR is now making some topographic maps in digital form. If all topographic maps were made this way, creating a data base of topographic information for a certain application would simply be a matter of abstracting the desired information from EMR tapes. This will be the case eventually, perhaps by the year 2000. Meanwhile, there is an immediate need for topographic information in digital form.

1.2 DATA-BASE SPECIFICATIONS

1.2.1 Types of Information

For radio-wave predictions, what is primarily needed is the elevation of the ground (or water) surface. A secondary but sometimes important piece of information is the nature of the ground and ground cover. In particular, the electrical properties of the ground are of interest, and the presence of obstructions such as trees or buildings can be quite important. Therefore, for each geographic point represented, the data base contains a number for the elevation, and a code for surface type.

1.2.2 Horizontal Spatial Resolution

Experience in calculating diffraction attenuation at VHF and UHF suggests that a horizontal resolution of 500 metres is adequate. This depends a great deal on the roughness of the terrain. Nevertheless, for the sake of simplicity, the spacing between data points has been chosen to be 500 metres everywhere in the data base. Figure 1 shows an area 15 x 15.5 km as represented in the data base at this resolution.

1.2.3 Precision of Elevations

Elevations were obtained from topographic maps drawn on the scale 1:50000. On these maps, the contour interval is 25 feet where the terrain is not too rough and 50 feet where it is rough. Therefore an elevation can be uncertain by as much as 50 feet (15 m). However, in reasonably shaped terrain, the data-base values, which are interpolated from the contour lines, are likely to be more precise than that. The other limitation on precision is the fact that in the data base, elevations are stored at points 0.5 km

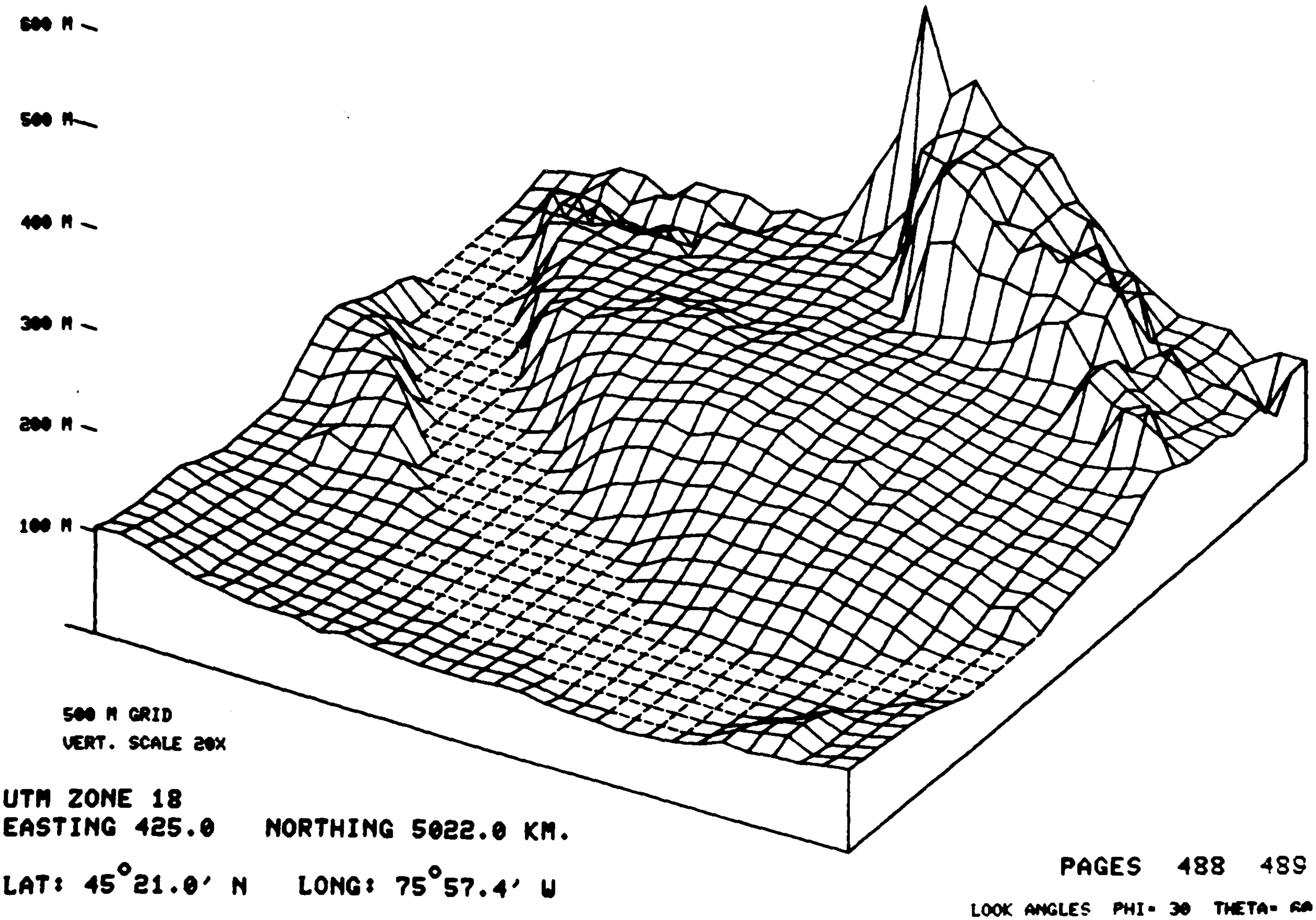


Figure 1. An area 15 x 15.5 km as represented in the data base. The coordinates given are those of the south-west corner. Broken lines indicate water. This view shows part of the Ottawa River just west of Ottawa, with the Gatineau Hills in the north.

apart. It is always possible for hills of small lateral extent to hide in between these points. Again, in reasonably shaped terrain, it is unlikely that hills of any significance will escape detection in this way.

1.2.4 Accuracy of Elevations

A more serious question is whether these elevations are correct. To begin with, they were scaled by hand from topographic maps. In the course of processing, the elevations were subjected to two stages of error detection and correction:

(1) The raw data are examined by a program that checks for sequence of data, for correct format, and for unlikely jumps in elevation. After such errors are corrected, it is fairly certain that the data are in the right place, and that the most spectacular errors are removed. However, less serious errors remain.

(2) After the data base has been formed, its contents are displayed in a large number of pseudo-three-dimensional plots. These plots are examined by eye, allowing the unmatched pattern-recognition ability of the human brain to find errors that the computer could not detect. See Section 4.9 for more detail.

Even after these visible errors have been corrected, there undoubtedly remains a residue of small errors that are difficult to detect. The data base at present contains data for over 3 million points, and will eventually contain many more; it is never going to be possible to guarantee that every one of them is correct. Nevertheless, it is hoped that errors will be small and not too numerous.

1.2.5 Speed of Access

The data base has been designed with real-time use in mind. Therefore it is meant to be placed on disk (rather than tape), and in files allowing direct access to any record in the file (rather than sequential access). A given computer may have several varieties of file organization that allow direct access. Physically contiguous files are presumably the fastest, if they are available, but keyed files serve adequately. The organization of data within the file has also been done with quick access in mind. Usually the profile of even a long path can be read from disk in no more than a few seconds. If a plot is made of the profile, it is mainly the speed of character transmission to the terminal that determines how long the user must wait.

1.2.6 Geographical Coverage

As of January 1982, the data base covers all of Ontario up to about 50° latitude, and a substantial fraction of the Atlantic provinces. This is illustrated in Figure 2. It is hoped eventually to cover most of the populated parts of Canada. The coverage of the data base resident at the Communications Research Centre will increase as data become available (see Section 4). It is expected that copies of the data base issued for use on other computers will be updated annually.

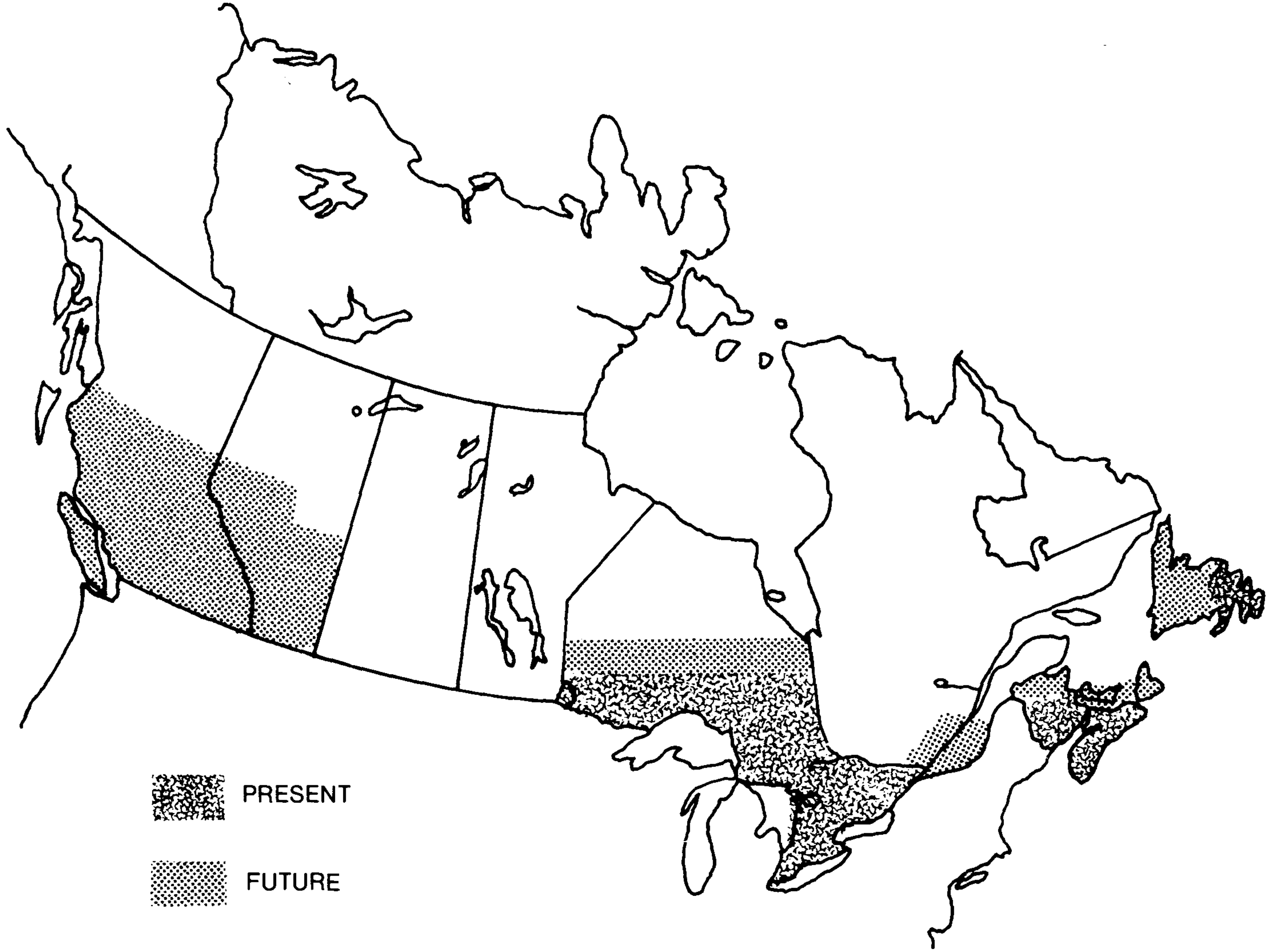


Figure 2. The present and future extent of data-base coverage. The "future" areas are those for which maps have been scaled, but which have not yet been incorporated into the data base. The large "future" area west of Calgary represents a data base held by the Gravity Division of Energy, Mines, and Resources.

1.2.7 Storage Requirements

As of January 1982, about 3.5 million terrain points are represented in the data base, occupying about 8 Mbytes of storage. (Each point occupies two bytes, but storage is not 100% efficient.)

1.2.8 Portability of Software

The programs that use the data base are written in Fortran that is not far removed from the 1966 ANS standard, i.e., a fairly primitive Fortran. It is hoped, therefore, that they can be transplanted to other computers with a minimum of trouble. (They were written on a Honeywell Sigma-9.) However, some changes will be necessary: A Fortran-callable routine is needed to open files. Since file access differs from computer to computer, the statement that reads the data base will have to be re-written. Finally, if the word length of the recipient computer is different from 32 bits, the procedure for unpacking data will have to be changed.

1.3 METHODS OF USING THE DATA BASE

1.3.1 Data Output as Plots

The form of plot usually wanted is a path profile, i.e., a plot of elevation as a function of distance along a great-circle path. A program called PLOTPATH plots a profile on any (printing) computer terminal, for any specified path. The user may choose any effective earth's radius. Another program plots horizon elevation angles as seen from any specified site. More information on both programs is given in Section 6.

1.3.2 Data Available in Computer Memory

For radio predictions, what is wanted is a list of elevations and surface codes along a specified great-circle path. A subroutine called PROFIL provides this. The use of a data-base path profile by the prediction program is illustrated in Figure 3. The plot programs also obtain their elevations from PROFIL. PROFIL, and its supporting subroutines, are outlined in Appendix A.

2. ORGANIZATION OF THE TERRAIN DATA BASE

2.1 THE UTM COORDINATE SYSTEM

The data base uses as a frame of reference the Universal Transverse Mercator (UTM) system of coordinates (3,4). The primary reason for this is that a UTM reference grid (blue lines) is printed on all large-scale topographic maps in Canada. Because of the presence of this grid, maps can be scaled much more easily using UTM coordinates than using latitude and longitude. Another advantage of the UTM coordinate system is that it is rectangular, whereas the latitude-longitude system is curvilinear. A disadvantage of the UTM system is that it is discontinuous in the east-west direction; every 6 degrees of longitude, the system changes.

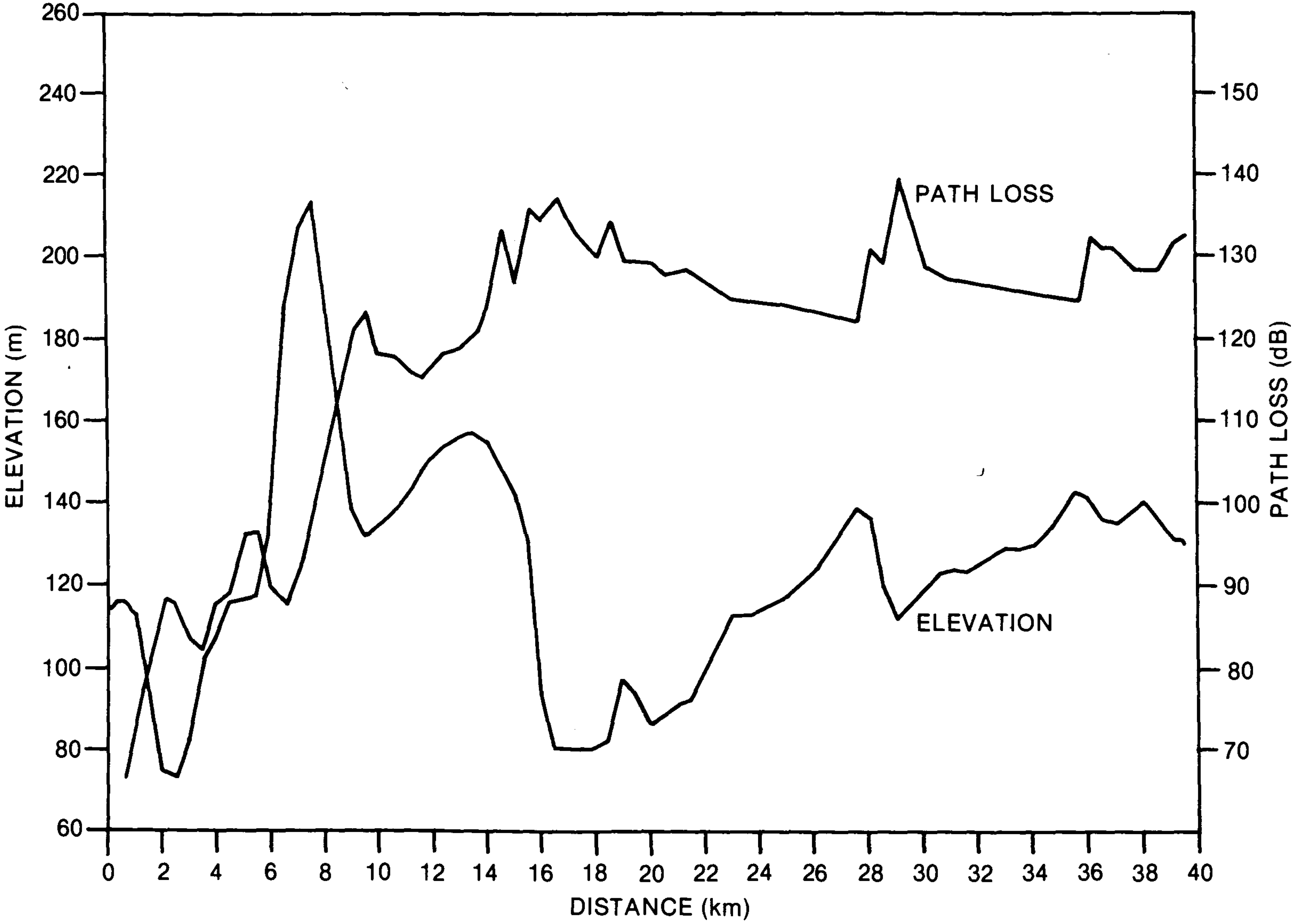


Figure 3. The predicted path loss as a function of distance along a radial from a Channel 6 television transmitter in the Ottawa area. The prediction is based on the path profile, which was obtained from the data base.

2.2 DEFINITION OF UTM COORDINATES

The UTM coordinate system divides the world into 60 zones, each zone covering 6 degrees of longitude (Figure 4). The zone numbers increase eastward, starting at the international date line, so that, for example, Ottawa is in zone #18. The UTM coordinates for a single zone are illustrated in Figure 5. The central meridian of each zone (75° W in the case of zone #18) is the backbone of the system for that zone. Along the central meridian, the y coordinate or 'northing' of a point is its distance from the equator, measured in metres. (In this report, such distances are expressed in kilometres.) Attached to the 'backbone' are 'ribs', the east-west grid lines, which intersect the central meridian at right angles, and follow great-circle paths away from it. These lines are then defined to be lines of constant 'northing'. (They do not follow parallels of latitude, since parallels of latitude are not great circles.) The x coordinate, or 'easting', is measured along these 'ribs', relative to the central meridian which by definition has an easting of 500 km. Finally, the north-south grid lines are drawn as lines of constant easting.

2.3 LATITUDE, LONGITUDE, AND UTM COORDINATES

Figure 6 shows the relationship between latitude, longitude, and UTM coordinates in zone #18. (For other zones, the only change in the diagram would be in the values of longitude.) The north-south line marked 500 (km) is the central meridian of the zone. The east-west UTM grid line marked 5000 (km) crosses the central meridian halfway between the equator and the pole. It may be noted in passing that, because of the ellipsoidal shape of the earth, this crossing point is not at 45° latitude. At the intersection of

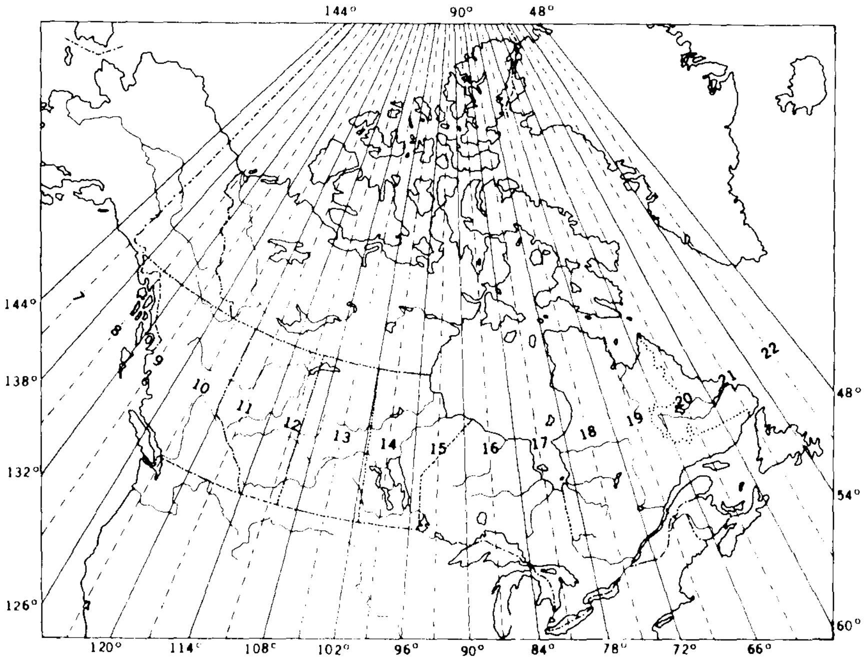


Figure 4. UTM zones and central meridians for Canada. Reproduced from Reference 1.

the 5000 km grid line with the central meridian are outlined the rectangular areas corresponding to a few blocks or 'pages' of the data base.

2.4 BLOCKS OF DATA: PAGES, RECORDS

In Figure 7 is shown an expanded view of the twelve rectangular areas that are shown in Figure 6. These areas measure 7.5 x 15.5 km; the coordinates of UTM grid lines defining the boundaries are indicated in the diagram. These areas will be referred to as 'pages', as will also the blocks of data assigned to these areas. The block of data for one page is written into a record of the data base. The position of the pages in the east-west direction is arranged so that the central meridian is a page boundary. The north-south position of the pages seems arbitrary in this diagram, but in fact it is arranged so that the equator is a page boundary. Each page has a number

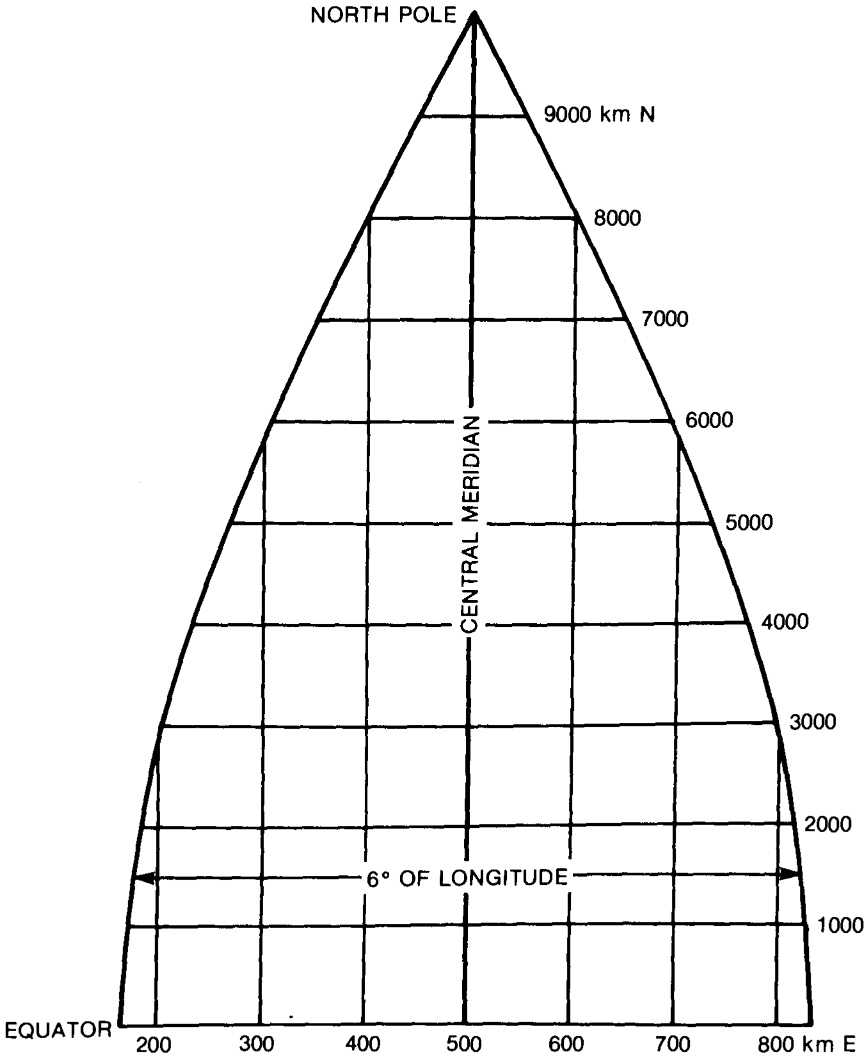


Figure 5. UTM coordinates for a single zone. The east-west scale is exaggerated by a factor of 10 with respect to the north-south scale.

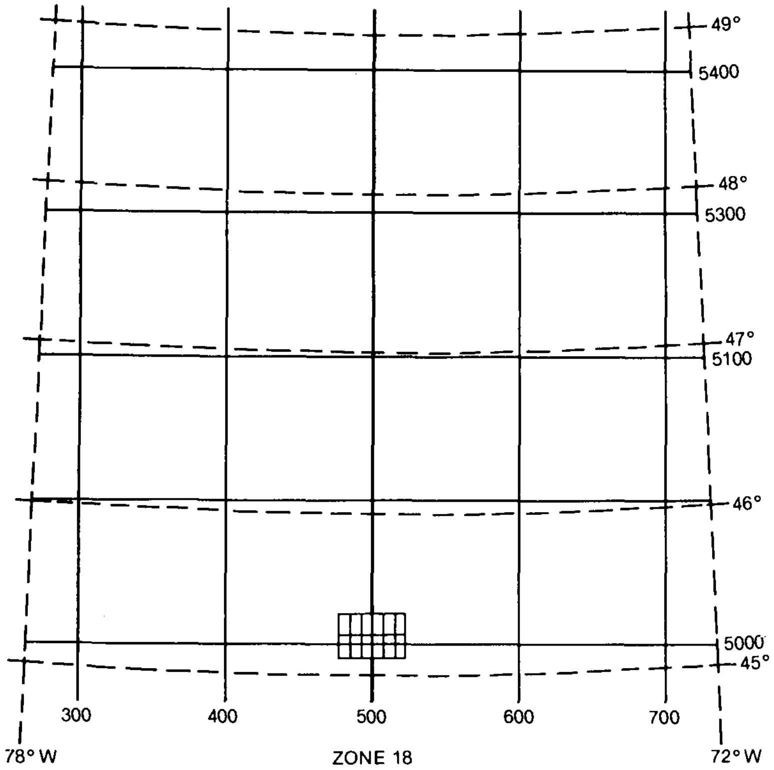


Figure 6. Part of UTM zone 18. The bounding meridians of longitude and several parallels of latitude are shown as broken lines. A few "pages" of the data base are outlined near the bottom of the diagram.

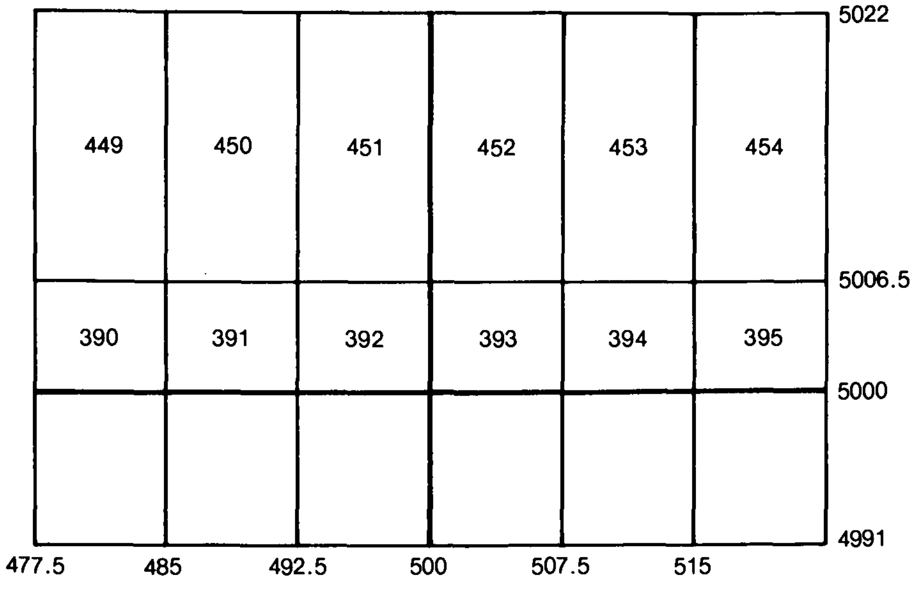


Figure 7. An expanded view of the twelve data-base pages shown in Figure 6. Page numbers (subject to change) are indicated for each page. The numbers around the periphery are UTM coordinates.

associated with it, as indicated in Figure 7. The pages are numbered from west to east and from south to north. The page (record) numbers indicate the positions of the pages in the disk file. If the data base were enlarged or re-organized, the coordinates on Figure 7 would remain the same, but the page numbers would change.

Figure 8 shows the area covered by the file ZONE 18. At the left of each row of pages is the number of the first page in the row. The area for which there are data is shaded. The twelve pages shown in previous figures are outlined in this diagram.

2.5 STRUCTURE OF A PAGE

The geographical area represented by a page of data is shown in Figure 9. The points in the page are numbered from 1 to 512, with numbers increasing from west to east and from south to north. The spacing between the points is 0.5 km. The information for one geographic point is contained in 16 bits. The page size is 512 16-bit words (1024 bytes) to conform, within a few factors of 2, to the sector length of many disk drives. A serious mismatch between record length and disk sector length would lead to wasted read operations.

The dimensions of the area represented in one page of data is 7.5 x 15.5 km rather than 8 x 16 km because the data on the eastern and northern edge of the area are repeated in the pages representing the adjoining areas. This duplication increases the storage requirement by about 9%, but greatly simplifies access, since the elevation of any point can be found by interpolation from points contained in a single page. Without such a duplication, sometimes two or four pages would have to be read to determine the elevation of a point. The shape of the page was chosen to be approximately square. The reason is illustrated in Figure 10. The most common use of the data base is to find elevations along a path. The use of long thin blocks of data would create the need for more read operations than for nearly-square blocks.

2.6 DATA WORDS

Each 16 bits contains data on the terrain surface cover and on its elevation. The format of a 32-bit word is represented in Figure 11. Bits 0 to 2 of each 16 bits contain the terrain surface code (variable 'LAND'), as defined in Appendix A (A1.1).

Bits 3 to 15 of each 16 bits contain the elevation of the point (IH) above sea level, in metres. The greatest elevation that can be represented in this space is 2**13 - 1 = 8191 metres. Only a few Himalayan peaks are higher than this. The exact elevation 8191 (1FFF in hexadecimal) is used as a 'no data' indicator. The elevation zero is not used for this because zero is a valid elevation that is very common in coastal areas. It is not possible in this scheme to represent areas below sea level.

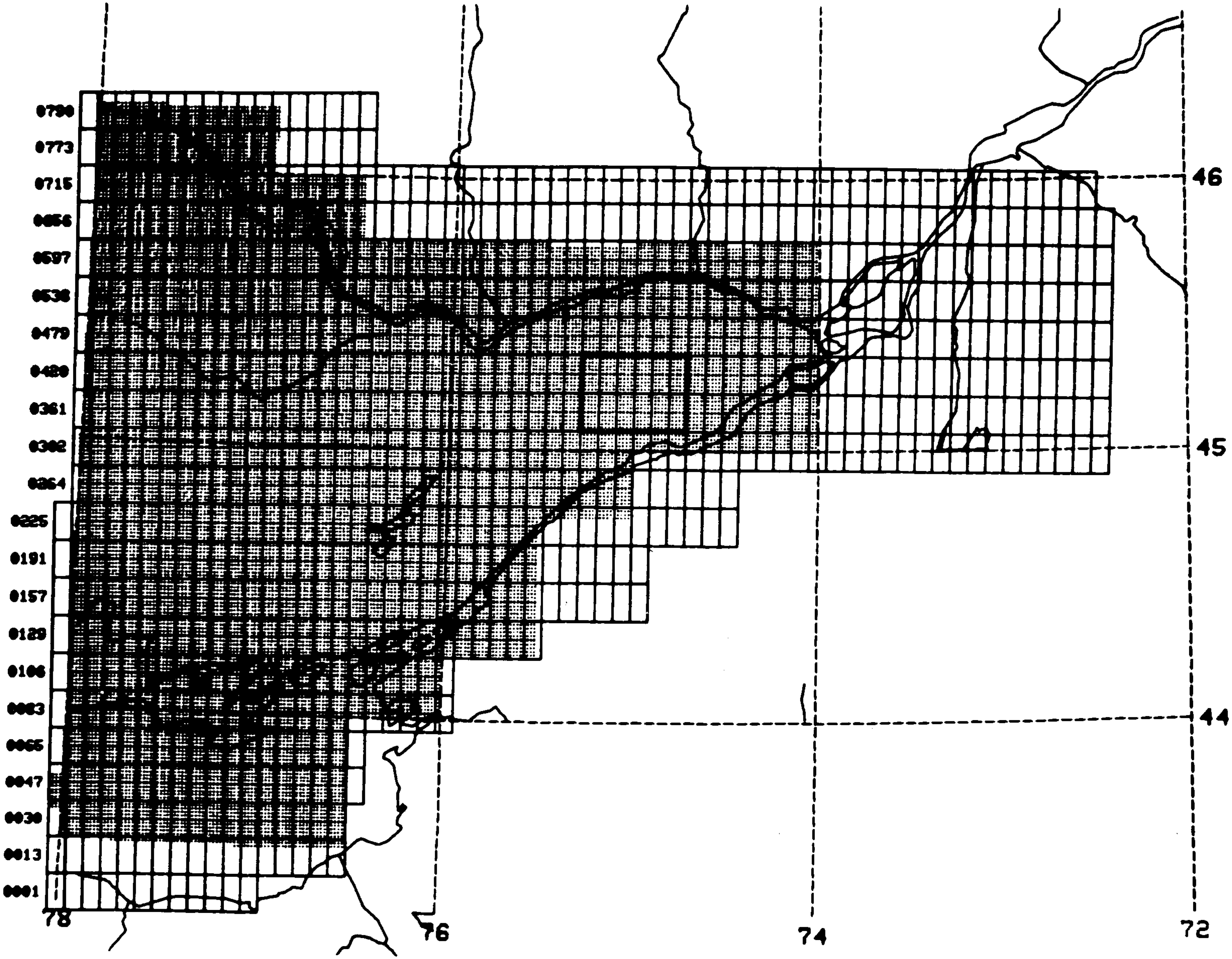


Figure 8. Data-base coverage in zone 18, as of January 1982. The small rectangles are pages; the numbers at the left are page numbers. The shaded region contains data, while the remainder does not.

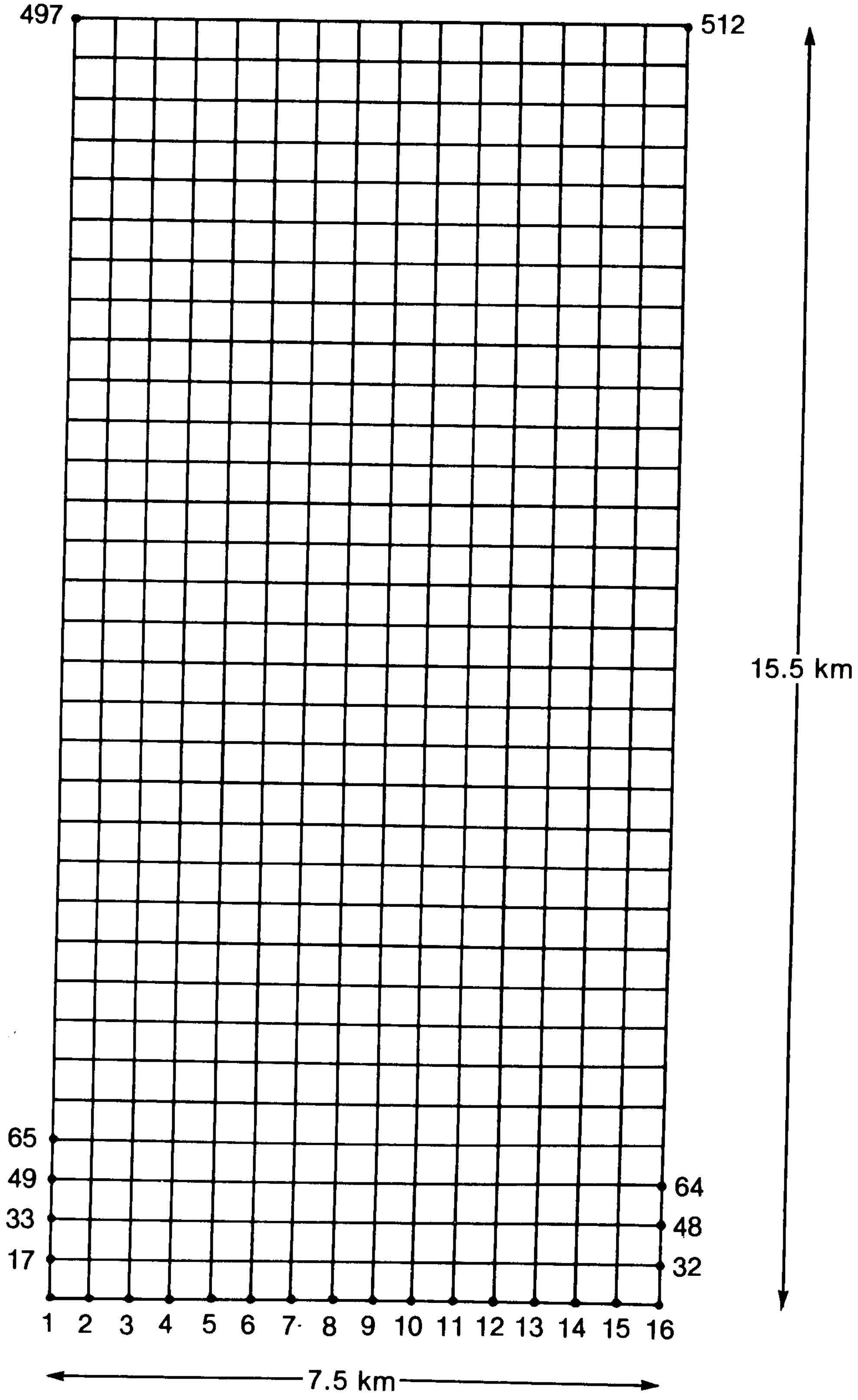


Figure 9. A page of data, showing the ordering of data within the page.

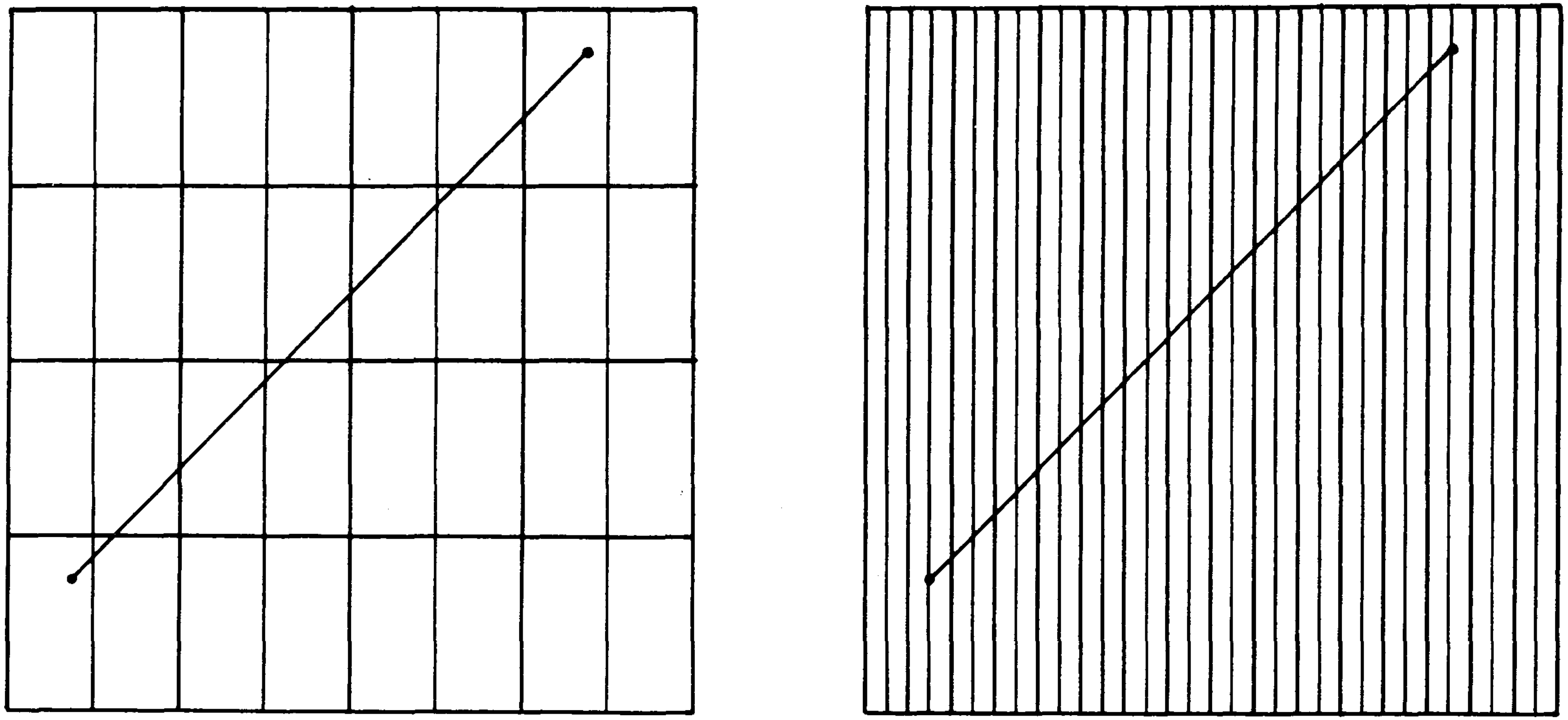


Figure 10. The reason for making data records correspond to almost square areas of terrain rather than long thin strips. Typically, fewer records must be read to obtain a profile.

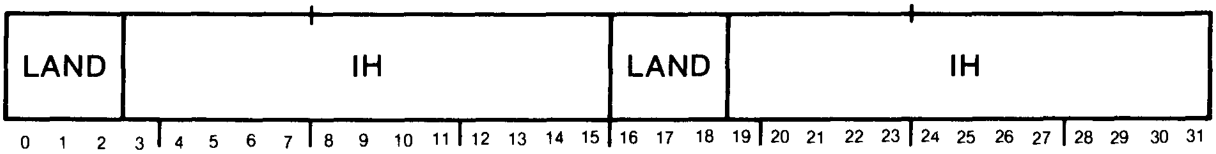


Figure 11. The format of a 32-bit word of data

3. INDEX TO THE DATA BASE

The data base is simply a long string of numbers on a few disk files. An index is required to relate geographic location to position in this string. There is one file for each UTM zone, and each file is divided (conceptually at least) into 1024-byte records (512 16-bit words, or 256 32-bit words). Each record corresponds to a 'page' of data.

3.1 CREATING THE INDEX - PROGRAM 'MAKINDEX'

3.1.1 Choosing Pages for the Data Base

For each UTM zone, there is a file on disk. In order to assign page numbers for a given zone, it is necessary to decide which pages to include and which to omit. We wish to choose only those pages for which data are available. Since topographic data are obtained from 1:50000 maps which cover a latitude range of 0.25 degrees, the area of interest is defined in terms of strips that are 0.25 degrees of latitude wide, but which can cover any longitude range. A set of such strips is represented in Figure 12. Our task is to find a set of pages, as previously defined, that covers all of this area, using no (or at least not many) more pages than necessary.

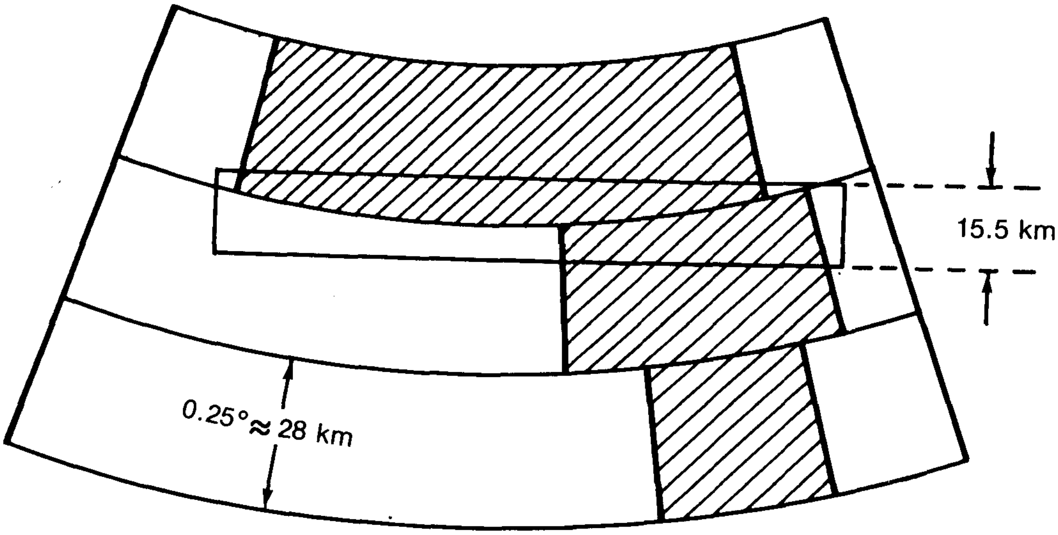


Figure 12. Choosing the eastern and western limits of a row of pages. Each row is 15.5 km in north-south extent. Taken together, such rows of pages must cover the shaded region, which has limits defined in latitude and longitude.

This task is complicated by the fact that the strips are curved. To simplify matters, we consider only the maximum and minimum northing of each strip. The maximum is always at one of the top corners. The minimum may be at one of the bottom corners, or it may be at the intersection of the bottom parallel of latitude with the central meridian (e.g., the top strip in Figure 12.)

Consider, then, a given row of pages, for which we want to choose western and eastern limits. This row may be included in one or two strips. In order to determine which strip(s) is(are) involved, first find the most southern strip that contains any part of the row, i.e., whose maximum northing is north of the southern edge of the row. Then to determine whether the next strip must also be considered, find whether its most southern point lies below the northern edge of the row. Finally, choose the first page in the row to be one whose western edge is west of the most western point of the strip(s) involved, and choose the last page similarly with respect to the most eastern point.

When all of the possible rows of pages in the latitude range of the data have been dealt with in this way, the data-base area has been selected. It remains to make an index to relate geographical position to position in the data-base files.

3.1.2 Tabulating the Index

The index to the data base is essentially a list of the x coordinate and page number at the beginning of each row of pages. A few other numbers are also required, namely the y coordinate of the bottom row, the number of rows, and the total number of pages. This is the information required for the part of a data base in a single zone. But rather than having a separate index for each UTM zone, we combine them, and therefore we must also record the starting position of the table serving each zone. A program called MAKINDEX tabulates all these numbers in the form of Fortran data statements for use in subroutine INDEX: a case of one Fortran program being used to write part of another.

3.2 USING THE INDEX - SUBROUTINE 'INDEX'

Given a point in UTM coordinates, we wish to find its location on disk. All the data for UTM zone #18 are found in a file called ZONE18 (and similarly for other zones). Subroutine INDEX finds the page number (IPAGE), and the (16 bit) word number within the page (IWORD) where the desired data word is located in this file. The subroutine proceeds as follows: First, it directs the search to the section of the index devoted to the UTM zone specified. One such section is represented in Figure 13. Then, the difference between the UTM northing (y) of the point in question and the base of the data gives the row number. The UTM easting (x) and the page number at the western end of the row is found from the index. Then the page number is found from the difference in x between the point in question and the beginning of the row. At various points along the way, tests are made for the given point being in or out of the data base, and if it fails one of the tests, the page number -1 is returned. To find the word number, the location of the point is compared to the location of the page origin.

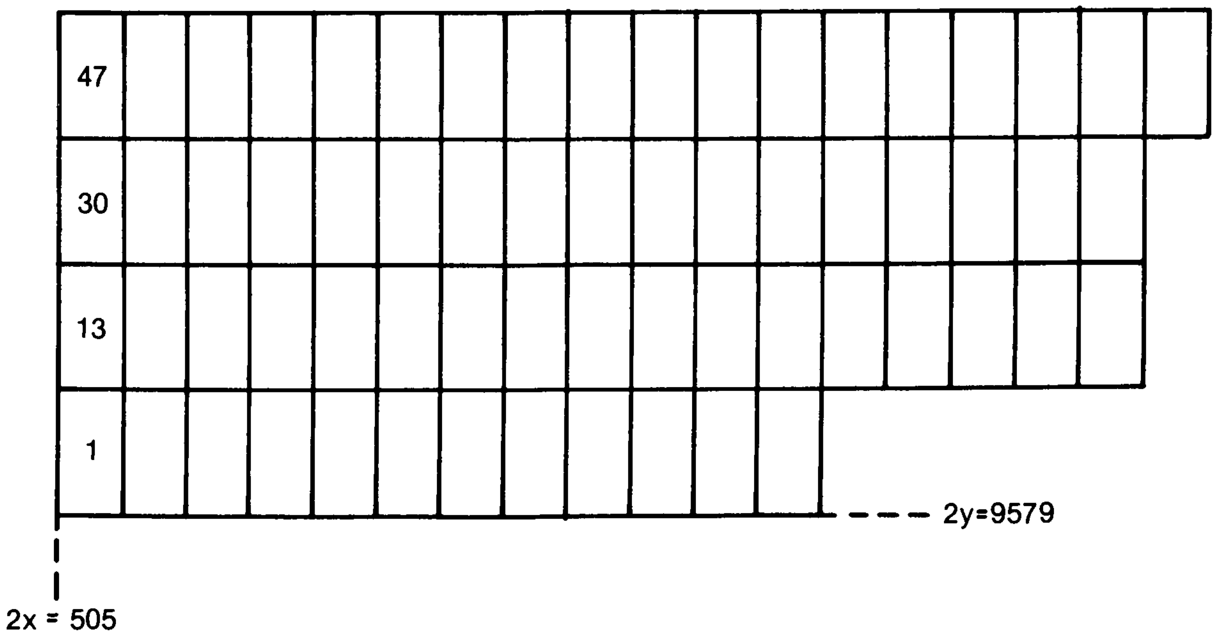


Figure 13. The first few pages of the part of the data base in zone 18. The index contains a single UTM y coordinate: the northing at which the data base begins. Then for each row of pages, it contains a UTM x coordinate and a page number that define the beginning of the row. The x values happen to be all the same in this example, but in general, they may be different.

The converse operation is to find the coordinates of a given page. The row number is found by examining in turn the page numbers at the beginning of the rows. Once the row is identified, the x coordinate (and page number) of its western end is obtained from the index, and the coordinates of the beginning (south-west corner) of the page may be found.

4. CREATING THE DATA BASE

4.1 SCALING TOPOGRAPHIC DATA

4.1.1 Who Does It?

The scaling has been done by students and other temporary workers under contract to various DOC Regional Offices. Each Regional Office has undertaken the responsibility for organizing scaling for its own region. Data are read from the topographic maps, written onto paper forms, and later punched into machine-readable form. Where possible, the same people also do the first level of error correcting (see Section 4.3). As of January 1982, most of Ontario and the Atlantic provinces have been scaled, and a start has been made in Quebec and in the Central Region. The files resulting from the scaling operations are processed into data-base form at CRC, where further error detection and correction are done.

4.1.2 Maps and Grid Lines

The data were scaled from the 1:50000 topographic maps published by Energy Mines and Resources, each of which covers an area of 0.25° in latitude by 0.5° in longitude. The Universal Transverse Mercator (UTM) coordinate system was chosen for this work, the main reason being that the UTM coordinate grid (blue lines) is already printed on the maps. There are two types of data scaled from topographic maps: elevation, and surface type.

4.1.3 Scaling for Elevation

Figure 14 reproduces the south-west corner of a topographic map. Such maps are scaled for elevation by following grid lines from south to north, and recording the elevation and northing for each point at which an elevation contour crosses the grid line. Since 500 m resolution is wanted, and the grid lines are printed on the maps in 1000 m intervals, half of the lines must be drawn in by hand. When a new north-south grid line is started, a heading record is entered which records the easting of that grid line. Some of the data scaled from the map represented in Figure 14 are displayed in Section 4.2.4. The east-west grid lines could also have been scaled in a similar way, but this would have doubled the amount of work for only a modest increase in accuracy; therefore it was not done.

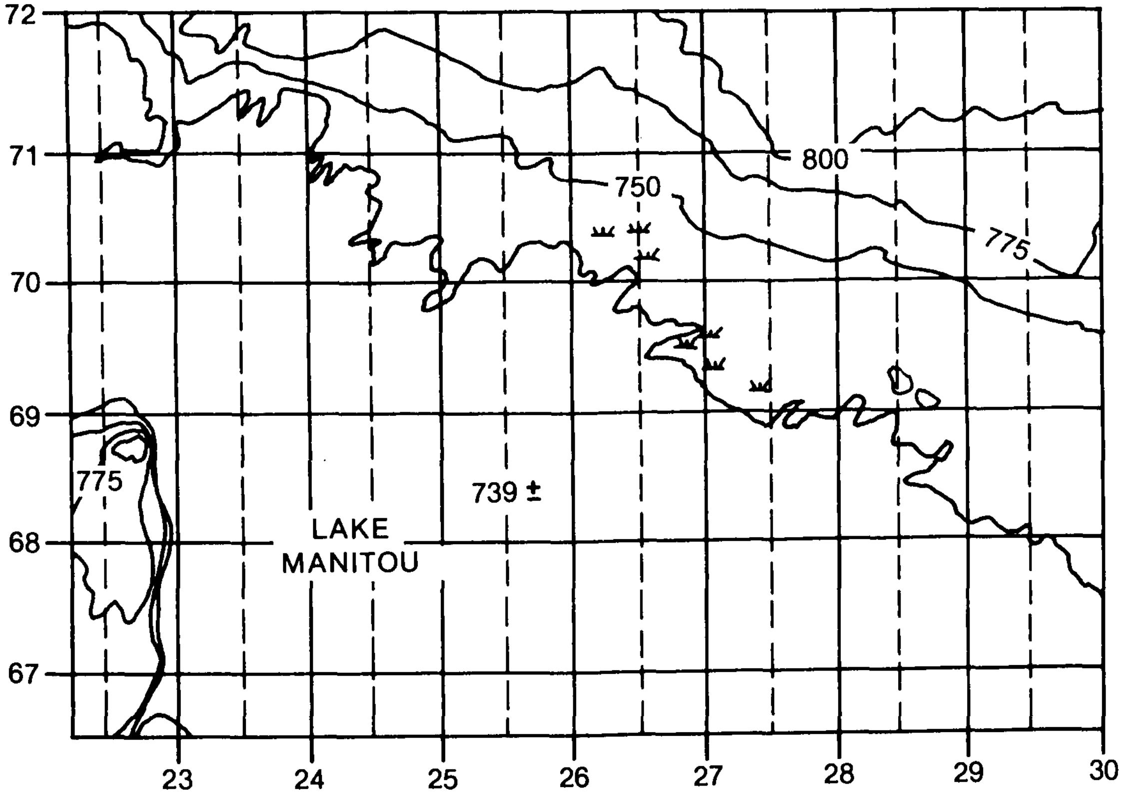


Figure 14. A reproduction of the south-west corner of the Little Current topographic map, map number 41H/13. The grid printed in blue in the original map is shown as solid lines. The lines that are drawn in by hand are shown as broken lines.

4.1.4 Overlap at UTM Zone Boundaries

At the boundaries between UTM zones, some extra scaling is done, in order to provide some overlap between the adjacent zones. Otherwise, there would be points very close to the boundary whose elevations could not be found by interpolation. This involves, for example, scaling elevations near the west side of a map in zone 18 in the coordinate system of zone 17, and also elevations on the east side of the adjacent map in zone 17 in the coordinate system of zone 18. On such maps EMR prints reddish-brown tick marks to indicate the coordinate system of the neighbouring zone. This extra scaling is done only for a narrow strip within about 0.6 km of the edge of the map.

4.1.5 Scaling Surface Type

The scaling for surface type is done along the same grid lines as the scaling for elevation. An entry is made every 500 m, at the intersections of the printed grid lines and at points halfway between. There is a code for tree-covered land, land without trees, fresh water, salt water, marsh, suburban area, urban area. The codes for scaling are letters representing the appearance on the map, as given in Section 4.2.4. The numerical codes in the data base are given in Appendix A (A.1.1).

4.2 STORAGE OF PRIMARY DATA ON DISK FILES

4.2.1 Names and Sizes of Files

Elevation and surface data are stored and processed separately, but they are combined in the final data base. For the Province of Ontario, the elevation data are stored in files which are given the names of topographic maps from which they were scaled, e.g., LITTLECRRNT. Abbreviations are used where necessary to reduce the length of the same to 11 characters or fewer. For all other areas, the file is named after the map number, e.g., 21H-05. (Map numbers have proved to be more convenient than map names.) The name of the corresponding file containing surface-type information is derived by adding an X at the beginning, e.g., XLITTLECRRN, or X21H-05. If the resulting name contains more than 11 characters, the final letter is dropped. The size of the elevation data files varies according to the density of contours on the map, but is typically about 200 kbytes. (This is for a keyed file; if the keys are removed, the space occupied by the file is reduced by almost a factor of 2.) The surface-type files occupy only 8 kbytes.

4.2.2 Lists of File Names

For Ontario, lists are kept of file names in order to retrieve and process the data from the files. An alphabetic listing of file names is stored on a file called MAPSA (A for alphabetic). A map number, latitude, longitude, and sort key are given along with each name. This list, sorted according to UTM zone number, latitude, and longitude, becomes the file MAPSN (N for numeric). There is also a file called MAPSM, sorted according to map number. In other regions, the use of map numbers avoids the necessity of all such lists.

4.2.3 Generating File Names

A program called SEARCH uses the file MAPSN to find the name of a map, given its geographic location. If no name for map at the given location is found on the list, SEARCH determines the map number (this can be done without a list), and uses this number as a name.

4.2.4 Format of Primary Data Files

Files containing elevation data have the following format:

```
NS LINE 22.5
0775 67.5
0775 68.7
0750 68.9
0739 69.1
0739 72.0
0750 72.1
0775 75.6
0775 76.8
etc.
NS LINE 23.0
0750 66.5
0739 66.6
0739 71.0
0750 71.8
0775 75.6
0800 76.7
0800 78.5
0775 78.6
0750 79.2
0675 79.3
0655 79.5
etc.
```

NS stands for north-south. The number after NS LINE is the easting of the UTM grid line being followed. Subsequent entries are the elevation (4 digits) and the northing (in kilometres) of each contour encountered along that grid line.

Files containing surface data have the following format:

```
22.5 67.0 GGGGWBBBBBBWGGGGGWGGGGWWGGWGGGGGGGBBGGGGWGGGGW
23.0 67.0 BBBBBBBBGGGWGGGGGGGWGGGMWWBBGGGWGGGBGGGGWGGGGGGGGBBBBBB
23.5 67.0 BBBBBBBBBGWWWWWGGGGGGGMGWBBBBGWGGGGGGGGWWGGGGGGGWWGBBBB
etc.
```

Each record contains the data for one UTM grid line running from south to north. The first number in the record is the easting of the grid line being followed. The second number is the northing of the first point in the sequence that follows. The letters following are the surface codes, every 0.5 km, as follows: G (green), W (white), R (red), M (marsh), S (sea water), F (fresh water). B (blue) may also be used for fresh water. Green, white, and red designate forest, open land, and city, respectively. Another designation, U (urban core), cannot be obtained from topographic maps, but is added later from aerial photographs of cities.

4.3 CHECKING INPUT FILES - PRIMARY ERROR CORRECTION

Data entered into the computer by hand are subject to a large variety of errors. A program called CHECK makes various consistency checks on the files of elevation data. Such a program cannot, of course, verify that the data are correct, but it detects any format errors, sequence errors in either easting or northing, and unreasonable excursions in elevation. Another program, XCHECK, checks surface-type data files for sequence and format errors. No attempt is made to check the surface-type designations themselves, except to verify that permissible letters are used. When all errors detected by the appropriate program have been corrected, the file is ready to be processed.

4.4 PROCESSING ELEVATION DATA

4.4.1 Creation of File With Tagged Records

One map is dealt with at a time. Processing begins with a program called HANDOUT. (It reads HAND-scaled data and creates an OUTput file.) This program uses elevation data files (as described in Section 4.2.4) as input, and creates a file in condensed, binary format, with position tags. In the new file, there are no headings; all records are self contained. Each record has the following form (64 bits):

<X TAG-><Y TAG->----<---ELEV---><-------X-------><------Y------>

The X TAG and Y TAG (8 bits each) identify the (500 m) grid intersection nearest to the point represented by the record. The next 4 bits are not used for hand-scaled data. ELEV is a 12-bit representation of the elevation of the contour to a resolution of 5 feet. X and Y are the coordinates of the point with 5-metre resolution.

A few map parameters are output to another file for later use (by NSGRID).

4.4.2 Use of Neighbouring Maps

The program HANDOUT reads data from the 8 adjoining maps as well as from the one being processed. This is done in order to solve the 'boundary problem'. That is, since the elevations to be entered into the data base are found by interpolation, there are always points at or near the edge of the map that cannot be correctly handled if data from neighbouring maps are not used. All of the points of all 8 maps are not written into the output file, since this would require an unnecessarily large amount of storage space. Instead, data are selected only if they lie within the area shown in Figure 15, e.g., within 10 km north and south of the map of primary interest, and within 2 km east and west. The appropriate map names are generated by subroutine SEARCH.

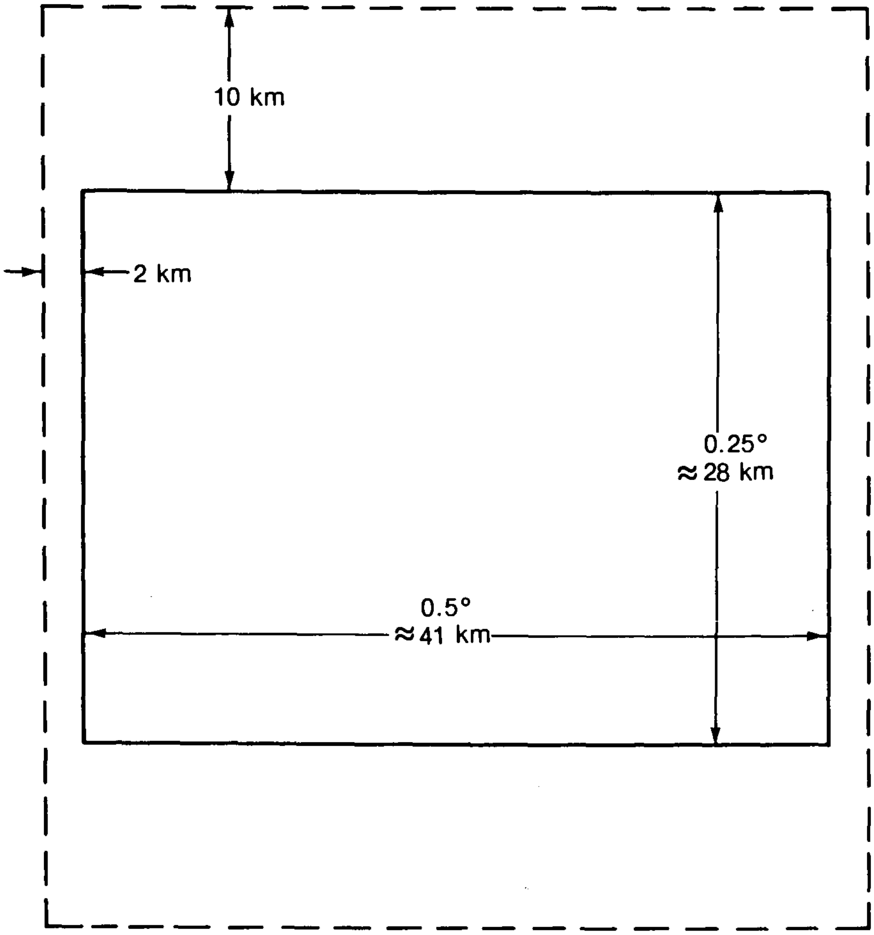


Figure 15. A 1:50000 topographic map and the area for which data from neighbouring maps are added

4.4.3 Sorting Elevation Data

Since data have been taken from several maps, their orderly sequence from south to north and from east to west has been lost. Therefore they are rearranged with the SORT processor provided by the computer operating system. The sort is done on the second word of each record, i.e., characters 5, 6, 7, 8, which contain the x,y coordinates. The file of sorted data now looks as if it came for a single large map with boundaries indicated by the outer rectangle of Figure 15.

4.4.4 Defining the Map Area

The program NSGRID uses the file of sorted data to interpolate the elevations at the grid intersections represented in the data base. First, the coordinates of the map corners are read from the file of parameters that was created by HANDOUT. NSGRID finds the coefficients of the equations of the straight lines that form the edges of the map. If the map happens to be located at one edge of a UTM zone, the edge is extended 0.6 km into the adjoining zone in order to accommodate the extra scaling that was done on the adjoining map to provide an overlap between zones. The limits of each north-south grid line are then found from these equations, and translated into local integer coordinates, leaving a boundary of from 0.5 to 1.0 km at both ends. The grid points within these limits are the ones for which elevations are to be found. These 'grid points' are 500 m apart as in Figure 9.

4.4.5 Interpolating Elevations

The sorted elevation data are read into memory one cell at a time, a cell being the 0.5 km interval whose centre is a grid point. Cells are identified by the tags mentioned in Section 4.4.1. At any given time, the data from 2 cells are present in memory. These cells, denoted A and B, are either adjacent, or, if not, any intervening cells contain no data. A possible situation is depicted in Figure 16, in which it is supposed that there is one empty cell between A and B. We wish to find the elevation at points 1,2 and 3.

Begin with point #1. Usually, there will be data points to the south of cell A. (If A is the most southern cell on the map, these data points will be from an adjacent map.) Then the elevation at A is found by interpolation between the nearest such point and the data point just to the north of A. If there are no data to the south of A, as in Figure 16, the elevation at A is taken to be the same as that of the first data point to the north of A. This occurs either when the surface is so flat that there are no contours for at least 10 km south of A (most likely a lake or ocean), or when we are at the edge of the data base.

The elevation at point #2 is found by interpolating between the most northern point of cell A and the most southern point of cell B. The eleva-

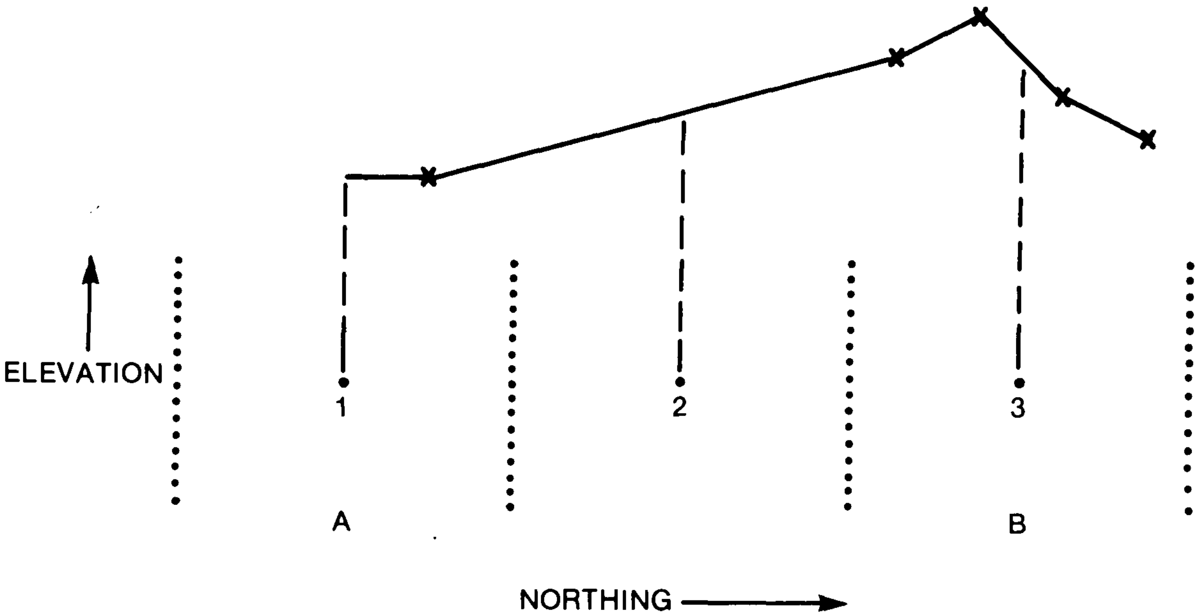


Figure 16. Calculation of elevations at grid intersections 1, 2, and 3. The data points used for this calculation are represented by x's.

tion at point #3 is found by interpolating between the nearest points in cell B. In order to find the elevation at point #4 (not shown), a read operation must be done, in which the cell containing point #3 is re-labelled A, and the next non-empty cell is called B.

In order to make certain that no points are missed, the program begins with the most southern point on the most western grid line, and deals with them in order, one at a time. The reading of data points from disk is done as necessary so that, where possible, there is at least one data point north and south of the grid point for which an elevation is wanted.

4.4.6 Writing into the Data Base

The elevation found in this way are entered into the data base by a subroutine called OUTDATA, which, when not provided with a valid surface-type code as input, preserves any surface-type data already present in the data base. The CPU time for processing one map ranges from 2 to 10 minutes.

4.5 PROCESSING SURFACE-TYPE DATA

The program for entering surface-type data into the data base is called SURFACE. No interpolation is required; therefore maps are processed without reference to adjacent ones. However, since the CPU time for processing one map is only a fraction of a minute, several are usually done in one run. The surface-type data are entered in such a way as to preserve any elevation data already there.

4.6 FILLING EXPANSES OF WATER

For map-size regions containing nothing but water, it is possible to enter a single elevation and a single surface-type code over the area that would be covered by a map if there were one. The short program that does this is called FILLIN.

4.7 DATA-BASE FILES

The data base is contained in several files, each one corresponding to a UTM zone. Thus, file ZONE15 contains data from UTM zone 15, and similarly with ZONE16, ZONE17, etc. When processing data, the program NSGRID assigns the output to the appropriate file. Each record of a data-base file contains 256 32-bit words, corresponding to 512 geographical points (one page).

4.8 DUPLICATION OF PAGE EDGES

The programs mentioned above do not enter data into the northern and eastern edges of the data-base pages. These are supposed to be duplicates of the southern and western edges of the adjacent pages. (See Section 2.5.) A program called REPLICATE goes through any chosen data-base file, and makes the necessary duplications.

4.9 EDITING THE DATA BASE - SECONDARY ERROR CORRECTION

The main method of checking the data base is by making pseudo-3-dimensional plots on a graphics computer terminal using program 3DPLOT. This program plots, in one picture, the elevations over the approximately square area covered by two pages of the data base. An example is shown in Figure 17. Errors tend to appear as unnatural-looking features. Because of the way the scaling is done, they often take the form of straight, narrow ridges or valleys running exactly north-south. In case a plot is difficult to interpret, the contents of any chosen page of the data base may be examined in detail by printing its contents onto a line printer with a program called DUMP. When an error has been detected, the source file is corrected and reprocessed.

5. EXTERNAL SPECIFICATION OF DATA-BASE COVERAGE

5.1 REASON FOR DOING IT

It is sometimes useful to be able to determine whether a point or a path is in the data-base area without actually accessing the data base. There are two reasons: (1) A long profile can take more than a few seconds to obtain from the data base, and a user might not want to wait this long, only to be told that it's not all there. (2) While the data base is in a state of development, there are sometimes areas outside of its previous coverage which contain errors. An external specification of the extent of the data base can be used to prevent use of these data until they have been corrected.

5.2 METHODS NOT USED

One can conceive of several ways of specifying the boundary of the region covered by a data base. A polygon with a convex outer boundary would be very convenient, since it has the property that a straight line joining any two points interior to the boundary is itself completely contained within the boundary. However, the boundaries of the land masses that we must deal with do not approximate convex polygons; to fill the interior of a convex polygon would entail an inordinate amount of otherwise unnecessary map scaling, and would increase the size of the resulting data base. It is possible to deal with a general polygon by dividing it into a number of convex polygons. However, if the number of divisions is not very small, determining whether a line is interior can be quite complicated.

5.3 METHOD USED - RECTANGULAR STRIPS

The method chosen here is to represent the region as a series of rectangular strips, 0.25° of latitude in width, and any desired longitude interval in length. The boundary of such a region is a polygon, and not a convex one, but not a general one either, since its form is restricted. The reason for this choice is that it can usually be made to fit the region

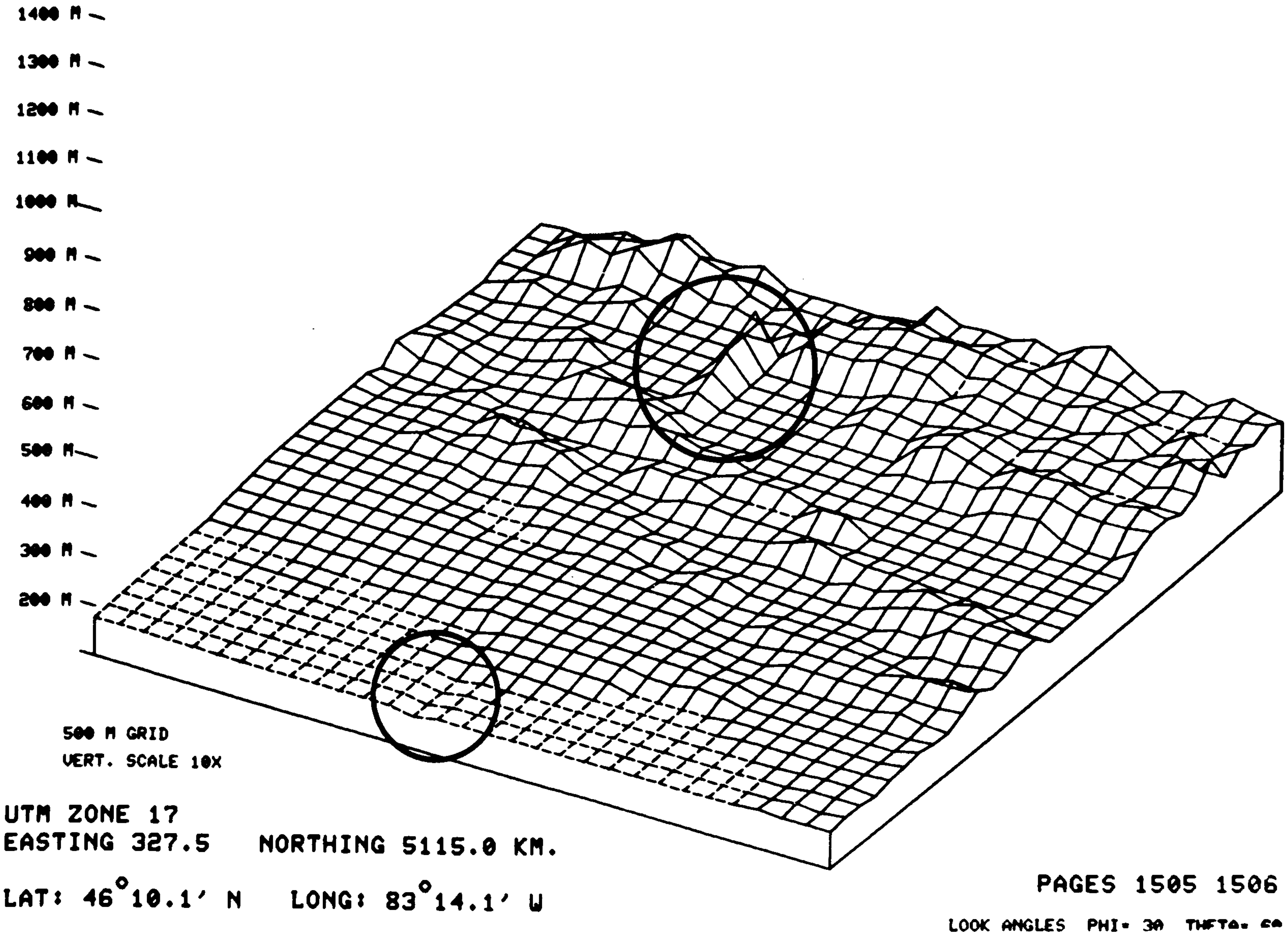


Figure 17. One of a great many plots used for error detection. Errors tend to give rise to unnatural-looking features in the terrain, such as the two enclosed by circles in this diagram.

actually scaled from 1:50000 topographic maps, which usually cover 0.25° in the north-south direction. Furthermore it is not too difficult to make the region more general by allowing interior 'holes', i.e., each strip can consist of two or more disjoint segments. A program called MAKIMAGE writes the data statements that specify these strips, and which are used in subroutine 'IMAGE'.

5.4 DATA-BASE 'IMAGE'

5.4.1 Is a Point in the Data Base?

A subroutine called IMAGE (see also Appendix A) determines whether a point is in the data-base area. (It is called IMAGE because it does not access the data base, but contains an image of its extent.) This subroutine contains data statements that define the extent of constant-latitude strips, 0.25 degrees in width, which contain data. These strips are different in form from those used in INDEX in two respects: They are not confined to a single zone, and they may contain gaps. (They are also different in content. The fact that a page number can be found in the INDEX is no guarantee that the page contains data.) A few such strips are shown in Figure 18. Finding whether a single point is in the data base is quite simple. The appropriate 0.25 degrees range of latitudes is found, and then the longitude limits of each segment of the strip are examined to find whether the point lies between them.

5.4.2 Is a Great-circle Path in the Data Base?

The procedure for determining whether a great-circle path is in the data base is more complicated. First, the path is divided into segments that are no more than 2 degrees in longitude. For each segment, we can approximate

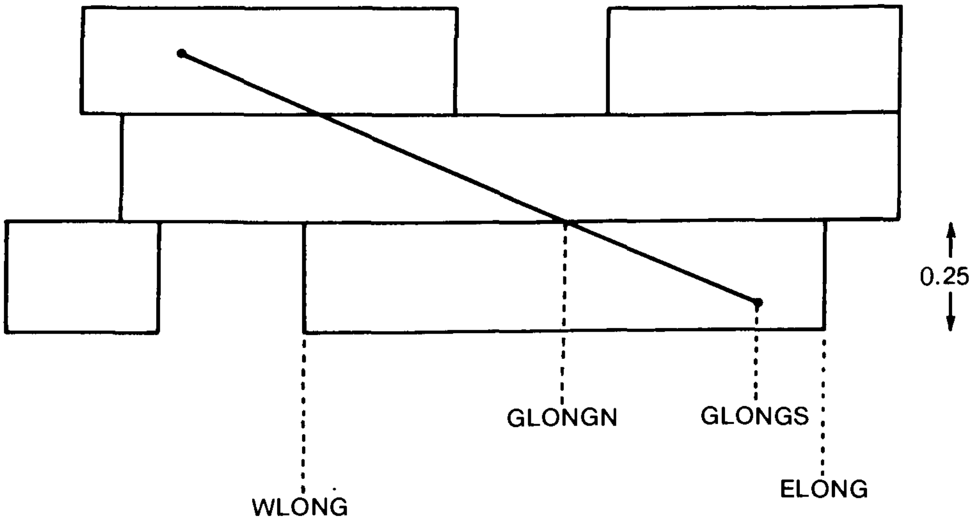


Figure 18. A great-circle path (oblique line), and a region that contains data (rectangualr strips). The words WLONG etc. are the names of variables representing longitudes in the Fortran program that determines whether a given path lies within the data-base area.

the parallels of latitude as straight lines. (The deviation from a straight line is less than 0.5 km.) We subdivide the problem further by considering separately each strip that contains any part of the segment in question. We find the longitudes GLONGS,GLONGN of the limits of the path within this strip (see Figure 18). Such a limit may be the intersection with a bounding parallel of latitude, or it may be the end point of the line segment previously defined. Now if both limits of the line segment lie between the longitude limits WLONG,ELONG of the same strip segment, then that part of the path is in the data-base area. If the results of all such tests are positive, then the great-circle path is wholly within the data-base area. Otherwise it is not.

6. PATH PROFILES - PLOTS

6.1 PROGRAM 'PLOTPATH'

This is a program that plots on an ordinary (alphanumeric) computer terminal the elevation profile between any two points specified by the user. The program finds out what the user wants by asking questions. A flow chart for these questions is shown in Figure 19.

6.1.1 How to Use the Program

The first question is: '# SPACES ACROSS PAGE?'. By knowing the width of the paper in the computer terminal, the program can scale the plot to fit. Typical answers would be 80 or 100; the maximum permissible number is 132.

The program then prints:

```
TRANSMITTER COORDINATES:

 COORD SYSTEM:   1 LAT,LONG   2 UTM CIV   3 UTM MIL
?
```

and waits for a response. The reply is 1, 2, or 3, depending on whether the user wants to express the transmitter coordinates in geographic, civilian UTM, or military UTM units. The program then asks for the coordinates in the units selected.

If option 1 is selected, the request takes the form:

```
 INPUT LATITUDE AND LONGITUDE(WEST)
       LAT         LONG
 DEG MIN SEC  DEG MIN SEC
?
```

The user is here invited to enter the coordinates of the transmitter end of the path. The response may be of the form

```
        45 57 19   78 4 23
```

or of the equivalent form

```
        45.955 0 0 78.073 0 0
```

The spacing between the numbers is unimportant, provided that there is at least one space between adjacent numbers. The computer then prints out the

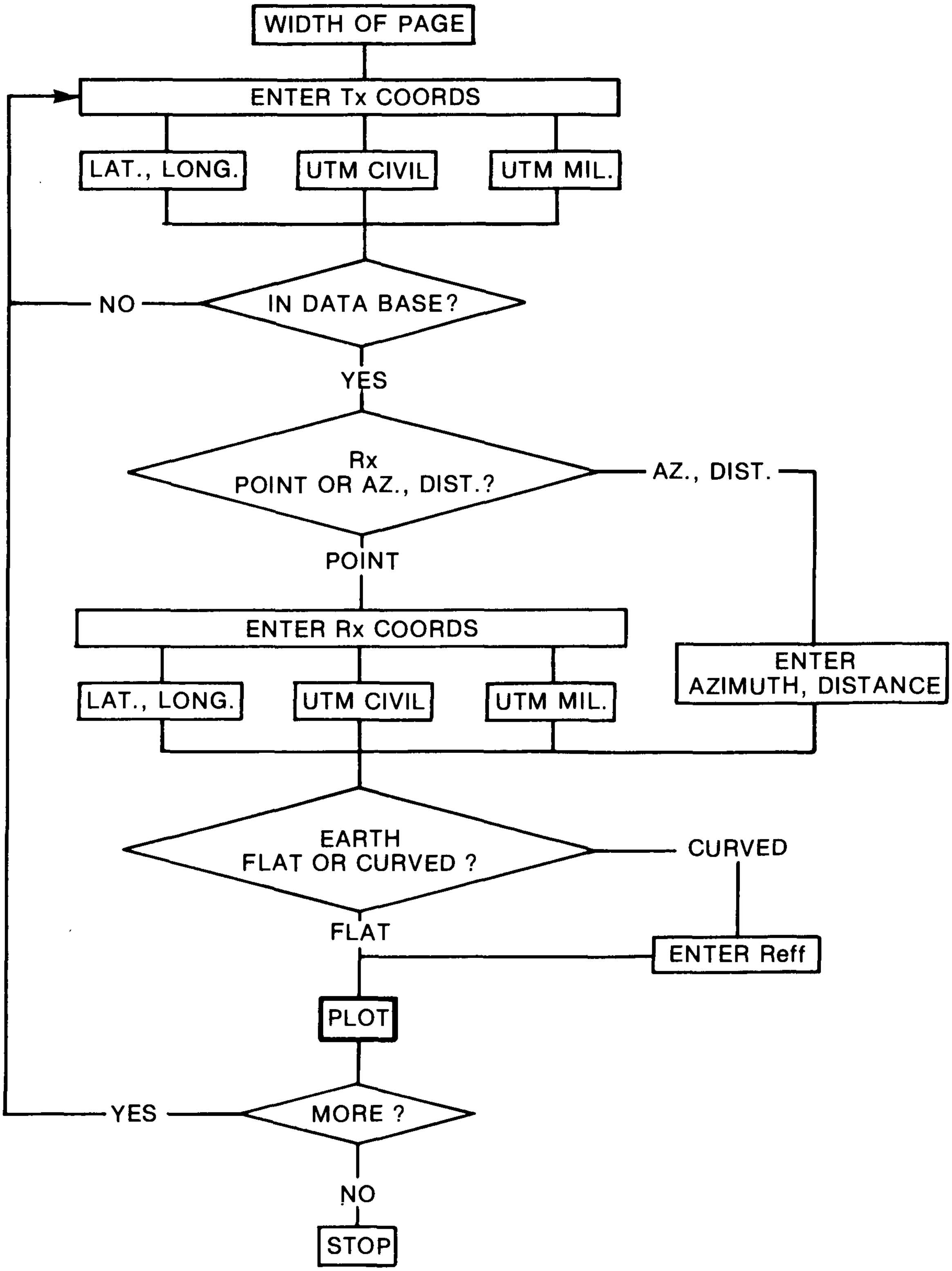


Figure 19. A flow chart for the interactive part of program PLOTPATH

transmitter coordinates in both forms, and asks for assurance that it has understood correctly. If the response is 'N' (for 'no'), the coordinates are requested again.

If option 2 is selected, the program prints:

```
 UTM ZONE NUMBER, EASTING, NORTHING:,      E.G.:
  18  445.2  5030.2
?
```

and awaits a response. Again, the spacing between numbers and the number of figures after the decimal point are optional. The easting and northing are assumed to be given in kilometres.

If option 3 is chosen, the program prints:

```
 ZONE DESIGNATION, SQUARE IDENT, GRID REF:      E.G.:
 22T CH703712
?
```

In this case, the format is not optional; it must be exactly as specified.

Once the transmitter location has been entered correctly, the computer informs the user whether it is within the area covered by the data base. If the transmitter is outside of this area, no profile can be plotted, and the user is asked to try another location.

Provided that the transmitter was found to be within the data-base area, the program asks for the coordinates of the receiver. If the receiver is not in the data-base area, the part of the path that is available will be plotted anyway.

The computer then obtains the profile from the data base, a procedure that may take a few seconds. When finished, it tells the user the length of the path and how many points will be plotted.

```
 DO YOU WANT A FLAT-EARTH, OR CURVED-EARTH PLOT?
 ANSWER 1 OR 2:
```

If the answer is 2, the computer requests:

```
 ENTER K FACTOR FOR EFFECTIVE EARTH:
 (NUL RESPONSE --> K=1.3333)
```

This is the user's opportunity to specify the effective earth's radius that he wants used for the plot. If he is happy with the standard 4/3 earth, he presses carriage return. Otherwise he can enter whatever value he likes. This feature provides much more flexibility than 4/3 earth graph paper, on which only a small number of values of K can be used, e.g., K=4/3 or K=2/3. The number must be entered in decimal form; the program will not accept a fraction such as '4/3'.

Immediately after receiving a value for K, the terminal will begin the plot. Therefore if the user wishes the plot to start on a new page, he may roll the paper up to the desired place before pressing the carriage-return key.

After the plot is finished, the user is asked

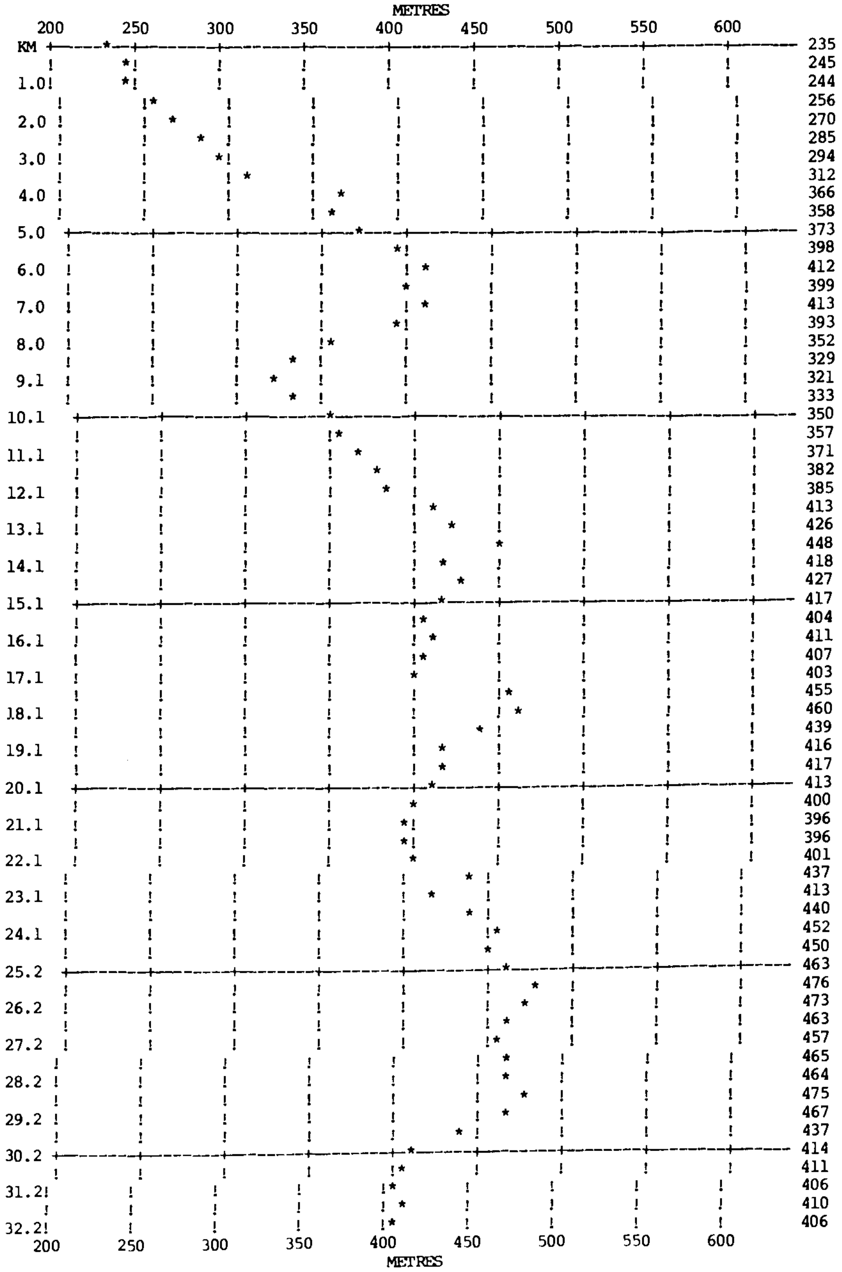
```
 MORE PLOTS?   Y OR N
```

If the answer is Y, the program goes back to ask for new coordinates. Otherwise, it stops.

An example of a profile plot is shown in Figure 20.

K FACTOR: 1.25000

TX: 45.955 N 78.073 W
45 57' 19" 78 4' 23"



RX: 45.704 N 78.280 W
45 42' 16" 78 16' 49"

Figure 20. A path profile plotted from the data base. The path is a radial from the Algonquin Radio Onservatory extending 32.2 km along an azimuth of 210°.

6.1.2 The Distance Axis of the Plot

The path specified by the user is divided into equal intervals in such a way that the interval length is as close as possible to being 0.5 km. Why not make the interval exactly 0.5 km? That would make the distance scale easier to read. But if the path length were not an exact multiple of 0.5 km, the receiving antenna site would not be one of the points of the plot. Or, if it were included, its distance from nearby terrain points would be incorrect. By making the distance intervals all equal, rather than making them exactly 0.5 km, the program can place both ends of the path on the plot in a correct relationship to the rest of the terrain. If it happens that the path length is an exact multiple of 0.5 km, then the distance between points is exactly 0.5 km.

6.1.3 The Height Axis of the Plot

The height scale varies from plot to plot, in order to obtain the greatest height resolution possible for each plot. In an alphanumeric plot, the resolution is the width of one character. The plot program therefore makes the height scale as large as possible for the given width of paper and range of height data.

The plots do not quite fill the paper, however, in order to leave room for drawing antennas and ray paths. At least 200 m are left clear above the end points of the path, and at least 50 m are left clear above all points.

6.1.4 The Curved Earth

If a curved-earth plot is specified, the lines of constant elevation are curved (except for very short paths), just as with 4/3 earth graph paper. The difference is that the computer-plotted 'curved-earth paper' is infinitely variable, depending on the K value specified, and on the vertical scale adopted for a given plot. Just as on ordinary '4/3 earth' paper, radio ray paths may be drawn as straight lines, under the assumption that the effective earth is as specified by the user.

6.2 PROGRAM 'HORIZON'

This is a program that plots on an ordinary (alphanumeric) computer terminal the elevation angle of the horizon as a function of azimuth as seen from any given point. The user may specify any effective earth radius. The program may be used, for example, to do preliminary surveys of proposed sites for satellite earth stations. Information is obtained from the user in much the same way as in the case of PLOTPATH.

A printed table that accompanies the plot lists the horizon angles, and also the distance to the horizon and the distance searched. If the horizon is very close, say within one kilometre, the elevation angle is not likely to be very accurate, because of the limited resolution of the data base. If the horizon is very distant, the program may not have found it, since it can only search to the edge of the data base, or to some predetermined distance (now fixed at 100 km).

Figure 21 shows an example of a plot made by HORIZON. (The accompanying table of values is not shown.) It represents the horizon around the Algonquin Radio Observatory.

```
K FACTOR: 1.33333
ANTENNA HEIGHT:   30.0 METRES

SITE:   45.955 N          78.073 W          235 METRES
        45  57' 19"       78   4' 23"
```

ELEVATION ANGLE OF HORIZON - DEGREES

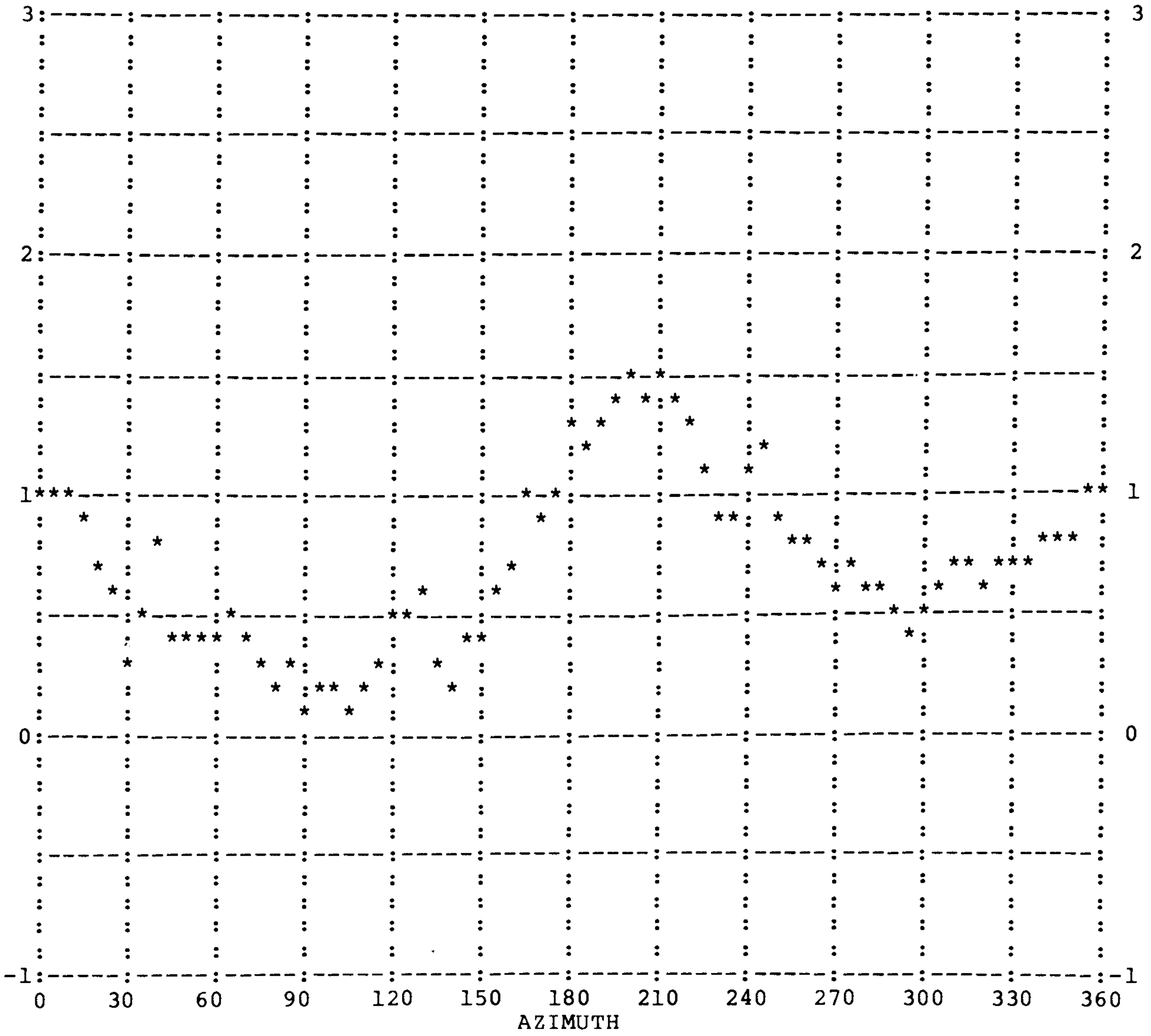


Figure 21. A plot of the elevation angle of the horizon as seen from the Algonquin Radio Observatory

7. ALGORITHMS FOR OBTAINING PATH PROFILES

7.1 SUBROUTINE 'PROFIL'

This subroutine obtains, from the terrain data base, a profile of elevation as a function of distance along a great-circle path. The path is defined by its end points given in geographic coordinates. The calling sequence is given in Appendix A. The main job of the PROFIL subroutine itself is to divide the great-circle path into segments such that each segment can be regarded as a straight line in the UTM coordinate system. In order to do this, it is necessary to examine the extent of the path in both the east-west and north-south directions. See Figure 22.

A great-circle path that goes approximately in the east-west direction will be a straight line in UTM coordinates because the east-west UTM grid lines are themselves great circles. However, since the coordinate system changes every time a zone boundary is crossed, the path must be divided into as many segments as there are UTM zones along the path. Usually, this will be no more than two, but the program can accommodate several zones. It finds the zone numbers of the end points of the path, and then uses subroutine FINDLA (find latitude) to locate the end points of each segment.

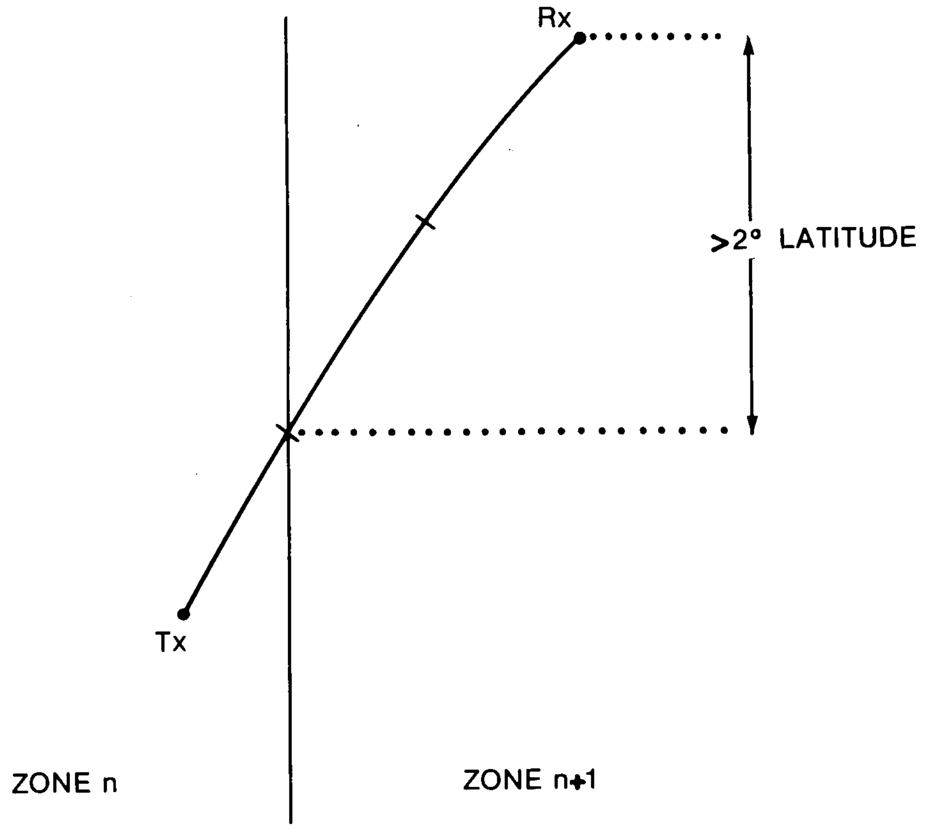


Figure 22. A great-circle path for which a terrain profile is to be found. The path is broken into segments if it occupies more than one UTM zone, or if within one zone it covers more than 2° of latitude.

Paths that go a long distance in the north-south direction present an additional problem, in that they are not quite straight lines in UTM coordinates. It happens that a path that traverses 2 degrees of latitude along the edge of a zone near the equator (the worst case) deviates from a straight line by 51 metres. Since this is approximately 1/10 of the data-base interval, such a deviation is not considered excessive. The program examines the path segments closest to the end points of the path, and if the latitude range of these segments exceeds 2 degrees in latitude, a subdivision is made. If the latitude range is less than 4 degrees, the segment is divided in two. Otherwise, it is divided so that the new segment closest to the end point is approximately 2 degrees in latitudinal range. The ends of the path are favoured in these procedures because for such long paths, the terrain profile at the centre of the path is relatively unimportant for VHF/UHF predictions.

Having found the end points of all the segments, the subroutine then calls subroutine INTERP, which does the work of finding the profile in each segment.

7.2 SUBROUTINE 'INTERP'

This subroutine interpolates distances within each segment of the propagation path, selecting points at approximately 1/2 km intervals, and calls ELEV to provide the data at these points. The path segment is divided into equal intervals such that the interval is as close as possible to 0.5 km in length. The interpolation procedure assumes that each segment is a straight line in UTM coordinates.

7.3 SUBROUTINE 'ELEV'

Given a point, whose location is expressed in UTM coordinates, this subroutine accesses the data base to find the elevation and surface code for the given point. The subrountine calls INDEX to find the appropriate page in the data base, opens the appropriate file, and reads the page. However, if the page is already in the buffer, it is not read again. The program then selects the four points that lie at the corners of the 0.5 km square that encloses the given point, and calls UNPAKH to decode the data words from these points. Elevations are then interpolated from these four points, and the terrain surface code is obtained by a weighted vote.

7.4 SUBROUTINE 'UNPAKH'

'UNPAKH' unpacks a word from the data base into elevation and surface code, according to the format given in Section 2.6.

8. ACKNOWLEDGEMENTS

The raw data for the data base have been and continue to be provided by the Regional Offices of the Department of Communications. They have organized and supervised map-scaling projects for this purpose. The work

of processing these data into data-base form, and especially of finding and correcting the many errors and irregularities that are inevitable in such a huge volume of data generated by hand, has been done at the Communications Research Centre. This work has been pursued with diligence, at various times, by Ann Lalumiere, Fran Brousseau, and Joan Thomas. Joan Thomas has also tested and repaired some of the programs. Eleanor Bachand has keypunched data that were left unfinished from one of the map-scaling projects.

9. REFERENCES

1. Sebert, L.M., *Every Square Inch,* Catalogue No. M52-2974, Information Canada (1970).

2. Palmer, F.H., *The Communications Research Centre VHF/UHF Propagation Prediction Program: An Overview,* Can. Elec. Eng. J., Vol. 6, No. 4, 3-9, 1981.

3. Sebert, L.M., *The Mercator and Transverse Mercator Projections,* Technical Report No. 69-1, Surveys and Mapping Branch, Dept. of Energy, Mines, and Resources (February 1975).

4. *Grids and Grid References,* U.S. Department of the Army Technical Manual TM5-241-1 (14 September 1962).

5. Danjon, Andre, *Astronomie General,* J.&.R. Sennac, Paris 2nd Edition (1959), pp. 16-18.

6. Peterson, A.E., *Program GSRUG (B60433), Geodetic Survey of Canada,* Dept. of Energy, Mines, and Resources, 615 Booth St., Ottawa, Ontario, K1A 0E9, September, 1973.

APPENDIX A

Outline of Subroutines

This appendix summarizes the subroutines associated with the data base, in each case giving the purpose of the subroutine and defining its arguments. The algorithms used by these subroutines are described in Section 7 and in Appendices B and C.

The relationship among the subroutines is shown in Figure A1. Of these subroutines INTERP is not outlined here (although it is mentioned in Section 7) because it is really an extention of PROFIL and cannot reasonably be called independently. OPEN and CLOSE are not outlined because they are machine dependent.

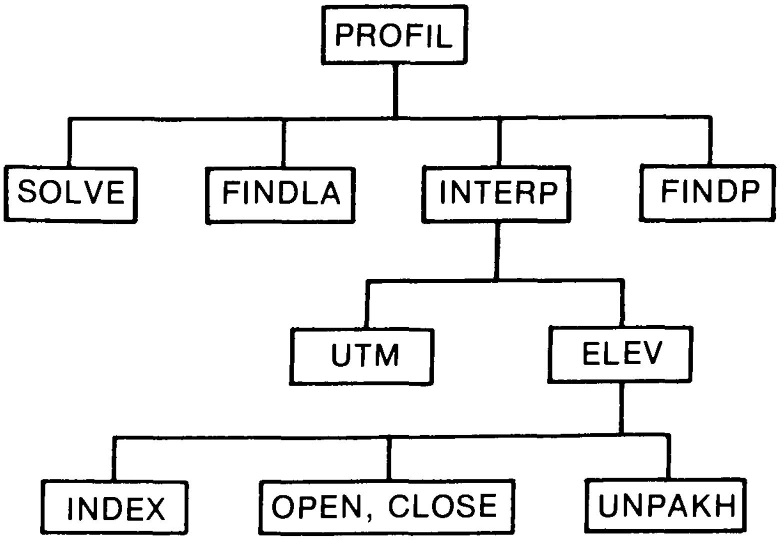


Figure A1. Block diagram of the relationship among the subroutines required to obtain a path profile from the data base.

A1. FINDING AND OBTAINING DATA

A1.1 PROFIL - Elevations and Surface Codes Along a Great-Circle Path

CALL PROFIL (GLAT1,GLONW1,GLAT2,GLONW2,NMAX,N,S,D,H,LAND)

-INPUT-

GLAT1	Latitude of transmitter (degrees)
GLONW1	Longitude of transmitter (degrees west)
GLAT2	Latitude of receiver (degrees)
GLONW2	Longitude of receiver (degrres west)
NMAX	The dimension of the arrays that will receive D,H,LAND

-OUTPUT-

N	The number of points in the profile
S	The length of the path in kilometres
D	Array containing N values of distance in km from transmitter location, D(1) = 0. (always); D(N) = S if the profile is complete. D(N) < S if the profile is not complete (because the end point is outside data-base area). Difference between successive D(i) is approximately 0.5 km.
H	Array of elevations in metres above sea level. H(i) is the elevation at D(i).
LAND	Array of surface-type codes; LAND(i) corresponds to D(i). Code is as follows:

0 Surface type unknown
1 Tree cover
2 Bare ground (no trees)
3 Fresh water (lakes and rivers)
4 Suburban (most buildings no more than 3 stories in height)
5 Marsh
6 Seawater
7 Urban core (high density, tall buildings)

A1.2 ELEV - Elevation and Surface Code at One Point

CALL ELEV (NZONE,X,Y,LAND,H)

-INPUT-

NZONE	The UTM zone number of the point in question
X	The UTM easting (in km) of the point
Y	The UTM northing (in km) of the point

-OUTPUT-

LAND	The surface code of the point in question
H	The elevation in metres above sea level

A1.3 UNPAKH - Unpack Data from One Word of Data Base

CALL UNPAKH (IDATA,IPART,LAND,IH)

-INPUT-

IDATA	32-bit word from data base representing two terrain points
IPART	0 or 1, meaning take data from left to right part of IDATA

-OUTPUT-

LAND	The surface code
IH	The elevation in metres above sea level

A1.4 INDEX - Find the Page and Word for a Geographic Point

CALL INDEX (1,NZONE,IX,IY,IPAGE,IWORD)
-INPUT-
NZONE The UTM zone number of the point in question
IX Twice the UTM easting in units of km (i.e., IX has units of 0.5 km)
IY Twice the UTM northing in units of km
-OUTPUT-
IPAGE The page number in the data-base file for zone NZONE
IWORD The location of the point within page IPAGE in 16-bit units.

A1.5 INDEX - Find the Geographic Point at the Origin of a Page

CALL INDEX (-1,NZONE,IX,IY,IPAGE,IWORD)
-INPUT-
IPAGE The page number
NZONE The UTM zone number, which is used for file selection
-OUTPUT-
IX Twice the UTM easting in units of km (i.e., IX has units of 0.5 km) of the origin (south-west corner) of page IPAGE in zone NZONE
IY Twice the UTM northing in units of km of the origin (south-west corner) of page IPAGE in zone NZONE
The variable IWORD is not used.

A1.6 INDEX - Find the Largest Page Number in a File

CALL INDEX (0,NZONE,IX,IY,MAXPAGE,IWORD)
-INPUT-
NZONE The UTM zone number, which is used for file selection
-OUTPUT-
MAXPAGE The largest permissible page number in that file
The other variables are not used.

A2. FINDING WHETHER DATA WILL BE AVAILABLE

A2.1 IMAGE - Is a Point Within the Data-Base Area?

CALL IMAGE (GLAT,GLONGW,0.,0.,IN)
-INPUT-
GLAT Latitude of the point (degrees)
GLONGW Longitude of the point (degrees west)
-OUTPUT-
IN If 0, point is not in the data-base area
If 1, point is in the data-base area

A2.2 IMAGE - Is a Great-Circle Path Within the Data-Base Area?

CALL IMAGE (GLAT1,GLONW1,GLAT2,GLONW2,IN)

-INPUT-

GLAT1	Latitude of transmitter (degrees)
GLONW1	Longitude of transmitter (degrees west)
GLAT2	Latitude of receiver (degrees)
GLONW2	Longitude of receiver (degrees west)

-OUTPUT-

IN	If 0, the path is not wholly within the data-base area If 1, the path is wholly within the data-base area

A3. TRANSFORMATION BETWEEN GEOGRAPHIC AND UTM COORDINATES

A3.1 UTM - Transform from Geographic to UTM Coordinates

CALL UTM (INSTR,GLAT,GLONG,NZONE,X,Y)

-INPUT-

INSTR	'Instruct'; will usually be put = 0 for this operation. If INSTR = 1, NZONE will be treated as an input variable, and IX,IY will be found in the coordinate system of UTM zone NZONE, whether GLAT,GLONG is in that zone or not.
GLAT	Latitude in degrees
GLONG	Longitude in degrees east of Greenwich. This is different from the convention used for PROFIL. Here, the longitude of Ottawa may be expressed as -76 or +284 degrees.

-OUTPUT-

NZONE	The UTM zone number of the given point
X	The UTM easting of the given point in units of km
Y	The UTM northing of the given point in units of km

A3.2 UTM - Transform from UTM to Geographic Coordinates

CALL UTM (-1,GLAT,GLONG,NZONE,X,Y)

-INPUT-

NZONE	The UTM zone number
X	The UTM easting
Y	The UTM northing

-OUTPUT-

GLAT	The latitude of the given point in degrees
GLONG	The longitude of the given point in degrees east of Greenwich

A4. SPHERICAL GEOMETRY

A4.1 SOLVE - Solve a Triangle, given 2 Sides and Included Angle

CALL SOLVE (TH1,TH2,PHI,S,A,B)

-INPUT-

TH1	Side #1 of triangle in units of radians
TH2	Side #2 of triangle in units of radians
PHI	The angle between side #1 and side #2, in units of radians

-OUTPUT-

S	The third side of the triangle, in units of radians
A	The angle between TH1 and S, in units of radians
B	The angle between TH2 and S, in units of radians

A4.2 FINDP - Find the Point a Given Distance Along a Radial

CALL FINDP (GLAT1,GLONG1,A,S,GLAT,GLONG)

-INPUT-

GLAT1	The latitude of the starting point, in degrees
GLONG1	The longitude of the starting point, in degrees east of Greenwich
A	The azimuth of the radial, in radians (from true north, clockwise)
S	The distance from the starting point, in radians

-OUTPUT-

GLAT	The latitude of the desired point, in degrees
GLONG	The longitude of the desired point, in degrees east of Greenwich

A4.3 FINDLA - Find Where a Given Radial Crosses a Meridian

CALL FINDLA (GLAT1,GLONG1,A,GLONG,GLAT)

-INPUT-

GLAT1	The latitude of the starting point, in degrees
GLONG1	The longitude of the starting point, in degrees east of Greenwich
A	The azimuth of the radial, in radians (from true north, clockwise)
GLONG	The longitude of the given meridian, in degrees east of Greenwich

-OUTPUT-

GLAT	The latitude of the desired point, in degrees

APPENDIX B

Spherical Geometry and Great-Circle Paths

Several subroutines are required to deal with great-circle paths: Their arguments are defined in Appendix A.

B1. SUBROUTINE 'SOLVE'

To solve a spherical triangle given two sides and the included angle. This procedure will solve any spherical triangle, but the primary purpose of this subroutine is to find the length and azimuth of the great circle path between two points whose latitudes and longitudes are given. See Figure B1. The two given points are labelled A and B, and C is the north pole. These letters also designate the angles at these points. The sides a and b are the co-latitudes of the points, and c is the path length to be found. Some quantities are given alternate names to suggest their geographical roles.

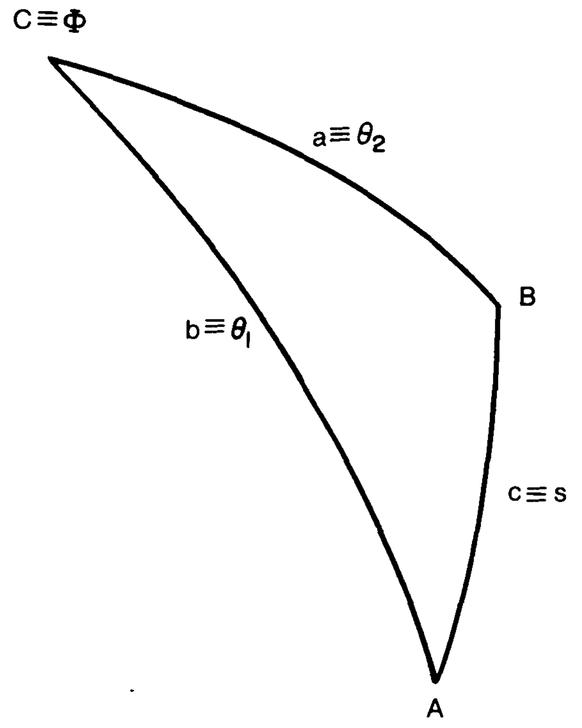


Figure B1. A spherical triangle. A, B, and C are the angles at the vertices, and a, b, and c are the arc lengths of the sides. If vertex C is at the north pole, then Φ is a difference in longitude, θ_1 and θ_2 are co-latitudes, and s is a great-circle path length.

In the following, equation numbers refer to 'Astronomie General' (5). We start with

$$\cos b \cos C = \sin b \cot a - \sin C \cot A \qquad \text{IV-3}$$

Solving For A,

$$\tan A = \sin C \,/\, (\sin b \cot a - \cos b \cos C)$$

or, translating to the other symbols,

$$\tan A = \sin \Phi \,/\, (\sin \theta_1 \cot \theta_2 - \cos \theta_1 \cos \Phi)$$

By reversing the foles of θ_1 and θ_2, we have also

$$\tan B = \sin \Phi \,/\, (\sin \theta_2 \cot \theta_1 - \cos \theta_2 \cos \Phi)$$

As Figure B1 is drawn, all angles are in the range o to π. However, when the subroutine is used to find the great-circle path from point P1 to P2, angle Φ is taken to be $\Phi_2 - \Phi_1$, where Φ_1 and Φ_2 are the longitudes of P1 and P2, eastward from Greenwich. Φ may therefore lie anywhere in the range $-\pi$ to π. SOLVE obtains angles A and B from their tangents by the 2-argument ATAN function, so that A and B have the sign of sin Φ, and also occupy the range $-\pi$ to π. Angle A is always the azimuth of the great circle path from P1 to P2, with the usual convention, i.e., clockwise rotation with respect to true north. Angle B is the azimuth from P2 to P1, but with its sign reversed.

Next, find the path length: We find the sine from

$$\sin s = \sin \Phi \sin \theta_2 \,/\, \sin A \qquad \text{II-3}$$

and the cosine from

$$\cos s = \cos \theta_1 \cos \theta_2 + \sin \theta_1 \sin \theta_2 \cos \Phi \qquad \text{I-3}$$

The path length s is then found from the tangent. In principle, we could have obtained s directly from equation I-3, but in practice, this would have given imprecise results for small values of s, since in this case cos s is insensitive to changes in s. This is important, because for VHF radio paths, s is usually small. If sin Φ happens to be zero, it is set equal to an arbitrary small value, in order to avoid a 0/0 in equation II-3. This gives the correct result, namely $s = |\theta_1 - \theta_2|$.

B2. SUBROUTINE 'FINDP'

Given the location of a transmitter, an azimuth A, and a path length s, we wish to find the latitude and longitude of the end of the path. This problem has the same geometry as the one just discussed (SOLVE), except that the known sides are θ_1 and s, and the included angle is A. To find the point, after converting all variables to radians, FINDP calls SOLVE with the appropriate calling sequence (see Appendix A4.2).

B3. SUBROUTINE 'FINDLA'

Given the location of a transmitter, an azimuth A, and a longitude, find the latitude at which the path crosses the given longitude. In this triangle (see Figure B1), we know one side (side b, the given co-latitude), and the two adjacent angles. We use again the equation

$$\cos b \cos C = \sin b \cot a - \sin C \cot A \qquad \text{IV-3}$$

this time solving for a:

$$\tan a = \sin b \;/\; (\cos b \cos C + \sin C \cot A)$$

Translating symbols,

$$\tan \theta_2 = \sin \theta_1 \;/\; (\cos \theta_1 \cos \Phi + \sin \Phi \cot A)$$

Since $\sin \theta_1$ will always be positive, θ_2 will be in the range 0 to π. If A is zero, and Φ is non-zero, the result is $\theta_2 = 0$, which is correct. If both are zero, the result is indeterminate, which is also correct, since the lines that are supposed to intersect lie along each other.

B4. PROBLEM 'FINDLONG'

Given the location of a transmitter, an azimuth A, and a latitude, find the longitude at which the path crosses the given latitude. The solution to this problem has not been implemented, because it is more difficult than the others, and not really needed. The solution is contained in equation IV-3. Solving for sin C leads to a quadratic whose double-valued solution is given by a complicated expression.

APPENDIX C

Transformation Between Geographic and UTM Coordinates

C1. PRECISION

The algorithm described here (subroutine UTM; see also Appendix A) is based on the documentation of a program by Peterson (6) which is used for geodetic work. In order to have a compact, fast-running subroutine, the series used for the transformations have been severely truncated. The resulting precision is nevertheless more than adequate for use with the data base. The subroutine has been tested with the 24 test points provided by Peterson. When it was compiled in single (32-bit) precision (the current practice for on-line work), the maximum error was found to be 24 metres. The maximum at Canadian latitudes would be about 15 metres. The dominant limitation here is the precision with which longitude can be specified in a 32-bit word. When the subroutine was compiled in double precision, the maximum error was found to be 11 cm.

C2. CONSTANTS AND NOMENCLATURE

The constants used in the transformations are based on the semi-major and semi-minor axes a and b of the Clarke 1866 spheroid. The eccentricity e and other derived constants are pre-calculated by a short separate program which creates the necessary data statements.

Peterson's nomenclature is adopted here, except that the roles of x and y are reversed:

- e eccentricity
- ϕ latitude
- ω longitude difference from the central meridian
- x easting
- y northing
- ρ radius of curvature in the meridian
- ν radius of curvature perpendicular to the meridian

C3. SUBROUTINE 'UTM' - GEOGRAPHIC TO UTM

C3.1 Meridian Length

The meridian length M to the given latitude is found from

$$M = a(1 - e^2)\ [E_0\phi - E_2 \sin 2\phi + E_4 \sin 4\phi] \tag{C1}$$

which is a truncated series correct to order e^4. The coefficients E_0, E_2, E_4 are functions of e, as follows:

$$E_0 = 1 + \frac{3}{4} e^2 + \frac{45}{64} e^4 + \frac{175}{256} e^6$$

$$E_2 = \frac{3}{8} e^2 + \frac{15}{32} e^4 + \frac{525}{1024} e^6$$

$$E_4 = \frac{15}{256} e^4 + \frac{105}{1024} e^6$$

The largest term neglected is

$$a(1 - e^2) \frac{35}{3072} e^6 \sin 6\phi$$

which has a maximum value of 2.3 cm.

C3.2 Zone Number

The zone number is selected according to the longitude given, provided that the input variable INSTR is not set to 1. If INSTR = 1, then the zone number specified by the user is accepted. (The user might wish to specify the zone number if he is working close to the boundary between two zones, and wants to use the grid system of only one of these zones.) Once the zone number is specified, the longitude relative to the central meridian can be found.

C3.3 Northing

The northing of the given point is found from

$$y = M + \omega^2 \frac{\nu}{2} \sin\phi \cos\phi\ [1 + \frac{\omega^2}{12} \cos^2\phi\ (5 - \tan^2\phi)] \tag{C2}$$

The expression $5 - \tan^2\phi$

is an approximation of $4 \frac{\nu^2}{\rho^2} + \frac{\nu}{\rho} - \tan^2\phi$

Since the term to which the expression belongs has a maximum value of 10 metres, the difference caused by the neglect of $\frac{\nu}{\rho} - 1$, which is of order $e^2 = 0.00677$, is only about 7 cm.

C3.4 Easting

The easting is found from

$$x = \omega \nu \cos \phi + \omega^3 \frac{\nu}{6} \cos^3 \phi \left(\frac{\nu}{\rho} - \tan^2 \phi\right) \tag{C3}$$

Here, the largest neglected term is approximately

$$\omega^5 \frac{\nu}{120} \cos^5 \phi \cdot 5$$

Since the maximum value of ω is 3° = 0.052 radians, the maximum value of the term is about 10 cm.

C3.5 Scale Factor

Finally, the scale factor 0.9996 shrinks the cylinder on which the projection is made by a standard amount to reduce the maximum scale error of the projection.

C4. SUBROUTINE 'UTM' - UTM TO GEOGRAPHIC

C4.1 Scale Factor

The scale factor is applied in reverse. From the zone number, the longitude of the central meridian is found.

C4.2 Zero-Order Latitude

The latitude is found which corresponds to a meridian length equal to the given northing. This involves inverting equation C1, reproduced here:

$$M = a(1-e^2) \left[E_0 \phi - E_2 \sin 2\phi + E_4 \sin 4\phi\right] \tag{C1}$$

E_0 is of order unity, E_2 of order e^2, E_4 of order e^4. Write

$$\phi = \frac{M}{E_0 \; a(1 - e^2)} + \frac{E_2}{E_0} \sin 2\phi - \frac{E_4}{E_0} \sin 4\phi$$

A zero'th order solution is

$$\phi_0 = \frac{M}{E_0 \; a(1 - e^2)}$$

This leads to a second order solution:

$$\phi_2 = \phi_0 + \frac{E_2}{E_0} \sin 2\phi_0$$

which in turn leads to a 4'th order solution:

$$\phi_4 = \phi_0 + \frac{E_2}{E_0} \sin 2\phi_2 - \frac{E_4}{E_o} \sin 4\phi_0$$

Now the second term of the expression is

$$\frac{E_2}{E_0} \sin 2(\phi_0 + \delta)$$

where
$$\delta = \phi_2 - \phi_0 = \frac{E_2}{E_0} \sin 2\phi_0$$

But
$$\sin 2(\phi_0 + \delta) = \sin 2\phi_0 \cos 2\delta + \cos 2\phi_0 \sin 2\delta$$

$$= \sin 2\phi_0 + 2\delta \cos 2\phi_0 \text{ to first order in } \delta$$

$$= \sin 2\phi_0 + \frac{E_2}{E_0} \sin 4\phi_0$$

to first order in E_2/E_0.

Therefore to order e^4, we have

$$\phi_4 = \phi_0 + \frac{E_2}{E_0}\left[\sin 2\phi_0 + \frac{E_2}{E_0} \sin 4\phi_0\right] - \frac{E_4}{E_0} \sin 4\phi_0$$

$$\phi_4 = \phi_0 + \frac{E_2}{E_0} \sin 2\phi_0 + \left[\left(\frac{E_2}{E_0}\right)^2 - \frac{E_4}{E_0}\right] \sin 4\phi_0 \qquad \text{(C4)}$$

This is the expression used in the program. However, in the program and in the next two subsections of this report, the result is called ϕ_1 to agree with Peterson's notation.

C4.3 Latitude

The latitude is found from

$$\phi = \phi_1 - \frac{x^2 \tan \phi_1}{2\rho\nu} + \frac{x^4 \tan \phi_1}{24 \nu^3 \rho} [5 + 3 \tan^2 \phi_1] \tag{C5}$$

The factor $5 + 3 \tan^2 \phi_1$ is an approximation to

$$- 4 \frac{\nu^2}{\rho^2} + 9 \frac{\nu}{\rho} (1 - \tan^2 \phi_1) + 12 \tan^2 \phi_1$$

The difference $\nu/\rho - 1$, of order e^2, is ignored. The maximum error in doing this is less than 4 cm. The largest term which is ignored is approximately

$$\frac{x^6 \tan \phi_1}{720 \rho \nu^5} \cdot 180 \cdot 5 \cdot \tan^2 \phi_1$$

which, at 45° latitude, is about 3 cm. (Since the maximum value of x varies as $\cos \phi$, this term is greatest at 45°.)

C4.4 Longitude

The longitude is found from

$$\omega = \frac{x}{\nu \cos \phi_1} - \frac{x^3}{6 \nu^3 \cos \phi_1} \left(\frac{\nu}{\rho} + 2 \tan^2 \phi_1\right) \tag{C6}$$

The largest term that is ignored is approximately

$$\frac{x^5 \sec \phi_1}{120 \nu^5} (28 \tan^2 \phi_1 + 24 \tan^4 \phi_1)$$

At 45°, this has a maximum value of about 33 cm, and will not be substantially greater for other latitudes (since maximum x varies as $\cos \phi$).

APPENDIX D

Curvature of an Ellipsoid of Revolution

The derivation of these results is presented here because it does not appear to be readily available outside of the specialized field of cartography, and because it represents an interesting exercise in three-dimensional geometry. Refer to Figure D1.

We wish to find the curvature of an ellipsoid

$$\frac{x^2 + y^2}{a^2} + \frac{z^2}{b^2} = 1 \tag{D1}$$

at any co-latitude θ. There are two curvatures to be considered, namely the curvatures parallel and perpendicular to the meridian. The most convenient way of approaching the problem is to rotate the ellipsoid about the y axis by angle θ, and to consider the point P_1 (x_1, y_1, z_1), $(y_1 = 0)$ where the tangent plane is horizontal. At P_1, the curvatures are given by the second derivatives

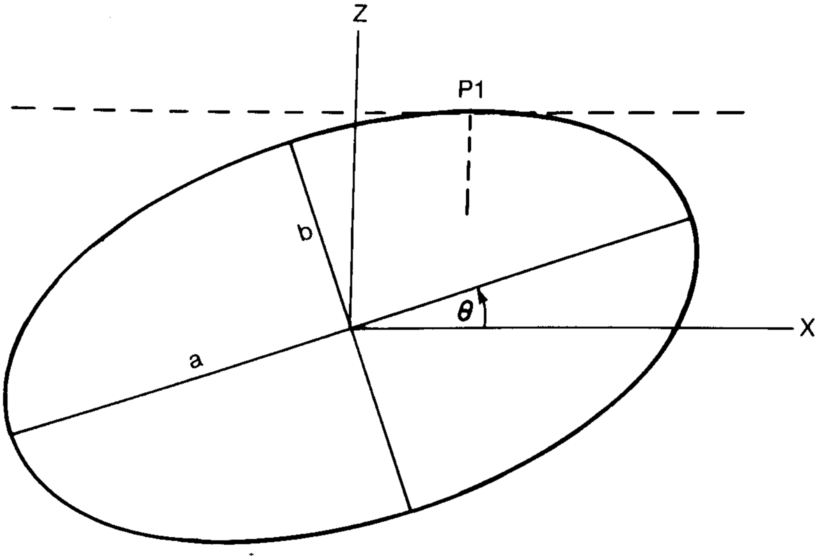


Figure D1. An ellipsoid of revolution with semimajor and semiminor axes a and b, inclined at angle θ from the xy plane. The broken line represents a plane which is parallel to the xy plane, and which is tangent to the ellipsoid at point P1.

$$\frac{\partial^2 z}{\partial x^2} \quad \text{and} \quad \frac{\partial^2 z}{\partial y^2}$$

since both $\frac{\partial z}{\partial x}$ and $\frac{\partial z}{\partial y}$ are zero at this point.

The equation of the rotated ellipsoid is

$$\left(\frac{\cos^2\theta}{a^2} + \frac{\sin^2\theta}{b^2}\right) x^2 + \frac{y^2}{a^2} - 2 \sin\theta \cos\theta \left(\frac{1}{b^2} - \frac{1}{a^2}\right)$$

$$+ \left(\frac{\sin^2\theta}{a^2} + \frac{\cos^2\theta}{b^2}\right) z^2 = 1 \tag{D2}$$

With obvious substitutions, this may be abbreviated to

$$A\, x^2 + \frac{y^2}{a^2} - 2B\, xz + Cz^2 = 1 \tag{D3}$$

The first partial derivatives of z are found from equation D3.

$$2Ax - 2Bx - 2Bx \frac{\partial z}{\partial x} + 2Cz \frac{\partial z}{\partial x} = 0$$

$$\frac{2y}{a^2} - 2Bx \frac{\partial z}{\partial y} + 2Cz \frac{\partial z}{\partial y} = 0$$

from which

$$\frac{\partial z}{\partial x} = \frac{Bz - Ax}{Cz - Bx} \tag{D4}$$

$$\frac{\partial z}{\partial y} = \frac{-y/a^2}{Cz - Bx} \tag{D5}$$

We next take second derivatives, putting immediately $\frac{\partial z}{\partial x} = 0$ and $\frac{\partial z}{\partial y} = 0$ in the results.

$$\left.\frac{\partial^2 z}{\partial x^2}\right|_{P_1} = \frac{-A}{Cz_1 - Bx_1} \tag{D6}$$

$$\left.\frac{\partial^2 z}{\partial y^2}\right|_{P_1} = \frac{-1/a^2}{Cz_1 - Bx_1} \tag{D7}$$

Now from equation D4, and the condition that $\partial z/\partial x = 0$ at P_1, we have

$$Bz_1 = Ax_1$$

$$\therefore \qquad Cz_1 - Bx_1 = C\frac{A}{B}x_1 - Bx_1 = \frac{x_1}{B}(AC - B^2) \qquad \text{(D8)}$$

It remains to find x_1. This is done by substituting $x = x_1$, $y = 0$, $z = (A/B)\,x_1$ in equation D3.

The result is

$$x_1^{\,2} = \frac{B^2/A}{AC - B^2} \qquad x_1 = \frac{B}{\sqrt{A}\sqrt{AC - B^2}} \qquad \text{(D9)}$$

Then equation D8 becomes

$$Cz_1 - Bx_1 = \frac{\sqrt{AC - B^2}}{\sqrt{A}} \qquad \text{(D10)}$$

and the second derivatives in equations D6 and D7 become

$$\left.\frac{\partial^2 z}{\partial x^2}\right|_{P_1} = \frac{-A^{\frac{3}{2}}}{\sqrt{AC - B^2}} \qquad \text{(D11)}$$

$$\left.\frac{\partial^2 z}{\partial y^2}\right|_{P_1} = \frac{-\sqrt{A}}{a^2\sqrt{AC - B^2}} \qquad \text{(D12)}$$

Now from the definitions implicit in equations D2 and D3, it may be found by carrying out the multiplications and combining like terms, that

$$\sqrt{AC - B^2} = \frac{1}{ab} \qquad \text{(D13)}$$

The coefficient A may be expressed as

$$A = \frac{\cos^2\theta}{a^2} + \frac{\sin^2\theta}{b^2} = \frac{1}{b^2}\left[1 - \left(1 - \frac{b^2}{a^2}\right)\cos^2\theta\right] = \frac{1}{b^2}\left[1 - \varepsilon^2\cos^2\theta\right] \qquad \text{(D14)}$$

where ε is the eccentricity of the ellipse.

If we now define the radii of curvature

$$\rho = -1/\frac{\partial^2 z}{\partial x^2} \quad \text{and} \quad \nu = -1/\frac{\partial^2 z}{\partial y^2} \quad \text{at } P_1,$$

and use the results just obtained (D13, D14) in equations D11 and D12, the radius of curvature in the meridian is

$$\rho = \frac{b^2/a}{(1 - \varepsilon^2 \cos^2\theta)^{\frac{3}{2}}} \tag{D15}$$

and the radius of curvature perpendicular to the meridian is

$$\nu = \frac{a}{\sqrt{1 - \varepsilon^2 \cos^2\theta}} \tag{D16}$$

where θ is the co-latitude of the point in question.

The ratio of the radii, used in some calculations, is

$$\frac{\nu}{\rho} = \frac{a^2}{b^2} (1 - \varepsilon^2 \cos^2\theta)$$

The product is

$$\nu\rho = \frac{b^2}{(1 - \varepsilon^2 \cos^2\theta)^2}$$

CRC DOCUMENT CONTROL DATA

1. **ORIGINATOR:** Department of Communications/Communications Research Centre

2. **DOCUMENT NO:** CRC Report No. 1353

3. **DOCUMENT DATE:** February 1982

4. **DOCUMENT TITLE:** The CRC Topographic Data Base

5. **AUTHOR(s):** J.H. Whitteker

6. **KEYWORDS:**
   (1) Topographic
   (2) Data
   (3) Base

7. **SUBJECT CATEGORY (FIELD & GROUP: COSATI)**

   20 Physics
   20 14 Wave Propagation

8. **ABSTRACT:**

A data base of terrain elevations and surface types has been compiled, primarily to provide input data for radio-wave prediction programs. The data have been derived by hand-scaling 1:5000 scale topographic maps. An elevation and a surface code (forest, lake, etc.) is stored for each point in a square array with 500 metre spacing. The data reside on computer disk, and are organized for rapid access. The usual method of obtaining information from the data base is by means of a Fortran subroutine that provides elevations and surface codes at 500 m intervals along any specified great-circle path within the data-base area. The elevations may also be plotted on an ordinary (alphanumeric) computer terminal.

9. **CITATION:**

Government of Canada Gouvernement du Canada