Communications Research Centre

Storage specifications for 500-metre-grid terrain data

by

J.H. Whitteker

CRC REPORT NO. CRC-~~CR~~ RP-92-00~~6~~ 2

Ottawa, August 1992

Government of Canada
Department of Communications

Gouvernement du Canada
Ministère des Communications

Canada

Storage specifications for 500-metre-grid terrain data

by
J.H. Whitteker

Radio Propagation Directorate

CRC REPORT NO. CRC-RP-92-002

OTTAWA, AUGUST 1992

CAUTION
This information is furnished with the understanding that proprietary and patent rights will be protected.

TABLE OF CONTENTS

STORAGE SPECIFICATIONS FOR 500-METRE-GRID TERRAIN DATA

by

J.H. Whitteker

ABSTRACT

Terrain data for much of Canada and some adjoining U.S. territory are stored compactly in a binary format. The format was designed to be used for obtaining elevations and surface-type information along radio propagation paths, but it may be used whenever compact, rapidly-accessible terrain data are required. Terrain points are the intersections of a 500-metre grid defined on the UTM (Universal Transverse Mercator) co-ordinate system. At each point, an elevation is stored, and a code indicating surface type (bare ground, tree-covered, lake, sea, buildings) is also stored wherever it is available. The information for each point occupies 16 bits. A record occupies 1024 bytes, and corresponds to a 7.5×15.5-km rectangle. With this arrangement, points that are geographically close together are usually also close together in the data file. A file represents an area 4° of latitude by 6° of longitude. Each file contains its own index in the first six records.

RÉSUMÉ

Les données du terrain d'une grande partie du Canada et de quelque territoires contigus des États-Unis sont entreposées efficacement dans un format binaire. On a conçu le format pour obtenir, le long des trajets suivis par les ondes radioélectriques, l'altitude et les renseignements sur le type de surface. Cependant, on peut également l'utiliser dans n'importe quel autre but, si l'on désire un accès rapide aux données de ce type. Les points sur le terrain sont les intersections d'un quadrillage de 500 metres defini avec le système de coordonnées UTM (Universel Transverse de Mercator). À chaque point de ce quadrillage, une altitude est emmagasinée avec, si disponible, un code indiquant le type de surface (terrain dénudé, lac, mer, forêt, bâtiments). Les renseignements pour chaque point occupent 16 bits. Un enregistrement occupe 1024 octets, et correspond à un rectangle de 7.5×15.5 km. De cette façon, les points qui sont rapprochés géographiquement sont, pour la plupart, aussi rapprochés dans le fichier. Un fichier représente une superficie de 4° de latitude par 6° de longitude. Chaque fichier comprend son propre index dans les premiers six enregistrements.

EXECUTIVE SUMMARY

A terrain data base that covers much of Canada has been developed by the Department of Communications and the Communications Research Centre (CRC) for use in radio propagation predictions. The purpose of this report is to provide an up-to-date detailed description of the storage format of the data, so that users can access the data in whatever way they choose, without depending on software already written for the purpose.

For each terrain point, the CRC data base stores the elevation in metres above sea level and a surface-cover code with eight possible values. Terrain points are arranged in a square array in which adjacent points are separated by 500 metres, and which corresponds to the square grid drawn on many topographic maps.

The 500-metre horizontal spacing has the disadvantage of being coarse for some purposes, but has the advantage of requiring little enough storage space that all the available data for Canada can reasonably be put on the hard disk of a personal computer. The advantages of using a square grid rather than a latitude-longitude grid are that points can easily be located on a gridded map, that the pattern is uniform with latitude, and that great-circle paths are more easily interpolated, particularly at the higher latitudes.

The storage scheme could be used worldwide, with a slight modification to the specification given here. That is, local elevation offsets would have to be introduced to accommodate land below sea level, and the highest Himalayan peaks.

The data are arranged as follows. Each computer disk file includes all the available data for a region 4° of latitude by 6° of longitude. All the territory within this region need not be covered, and storage is not required for those parts not included.

This large region is divided into many small rectangular areas, each one measuring 7.5 km by 15.5 km. Data for each of these rectangular areas are stored together. This arrangement allows for rapid access when data are required for many points that are geographically close together, as, for example, when a great-circle path is being followed. The index for each file is contained in the file itself. For economy of storage, data are stored in binary code.

The data base was compiled for use in radio propagation predictions, but may be useful wherever there is a requirement for a compact and simple representation of terrain elevations and surface-cover types for a large geographical area.

1. INTRODUCTION

1.1 BACKGROUND

A terrain data base that covers much of Canada has been developed by the Department of Communications and the Communications Research Centre (CRC) for use in radio propagation predictions. A description of the method of compilation of the data, some associated software, and the structure of the data files that was used for several years is given in an earlier report [1]. The use of terrain data for radio predictions is outlined in reference [2], and also in [3], which is one of several papers in a special conference session on the subject.

1.2 PURPOSE

The present report is necessary because the indexing system has been changed in order to facilitate the expansion of the data base. In the indexing system described in reference [1], the index was contained in the computer program that accessed the data. As a result, whenever the data coverage outgrew the confines of the index for one of the files, the program would have to be changed, and the data in that file would have to be rewritten. In the present indexing system, the index for each file is contained in the file, and the accessing program never has to be changed. Furthermore, the index structure for a file can accommodate all the data that can ever be written into that file.

1.3 SCOPE

This report does not repeat the material of report [1] on the compilation of the data, nor does it describe the associated software. However, it does give a complete description of the present-day structure of the data base. The most convenient way to extract information from the data base may be to use various FORTRAN subroutines that have been written for use with radio propagation prediction programs. However, with this report in hand, it should be possible to write programs in any desired language to extract data in whatever form they are desired.

1.4 GENERAL FEATURES OF THE DATA BASE

The general features of the CRC topographic data base are as follows: It provides the elevation and a surface-cover code for points in a 500-metre square array in the Universal Transverse Mercator (UTM) co-ordinate system, which is described in references [4] and [5]. The elevation is stored as an integral number of metres in the range 0 to 8190. The surface cover is represented as a 3-bit code (eight possibilities).

The format of the data base therefore has certain limitations. The main one is that the horizontal resolution is fixed at 500 m, which may be too coarse for some purposes. Another is that land below sea level cannot be represented, nor land higher than 8190 m. Thus Mt. Everest, at 8848 m, cannot be represented, nor can K2

(8611 m), Kānchenjunga (8598 m), Lhotse (8511 m), and Makālu (8481 m), all in the Himalayas. Similarly, the Dead Sea, at −395 m, and Death Valley, at −86 m, cannot be represented. (For a particular region, elevations could be offset to accommodate one extreme or the other.) A third limitation is that there is no provision for storing the height of trees or buildings, only their presence.

On the positive side, the format of the data base is simple and compact. At Canadian latitudes, a single file, which covers a geographical region 4° of latitude by 6° of longitude, does not exceed 1.9 Mbytes in size. The 1991 data base (southern B.C. and Alberta, southern and mid Ontario, southern Quebec, the Atlantic provinces excluding Labrador) occupies 20 Mbytes, easily accommodated on the hard disk of a personal computer. Because the data for points that are geographically close are usually stored close together in the file, they can often be all extracted with a single search and read operation, making for rapid access. On scales larger than a few kilometres, there is no restriction on the shape of the area that is covered, and areas not covered do not require reserved storage space. There is a place to record the presence of surface features (trees, buildings, water, etc.).

The UTM co-ordinate system was chosen for two reasons. One is that it is the basis of the grid lines found on many topographical maps, so that data points are easy to locate on these maps. The other reason is that, within a zone, the UTM grid is uniform and nearly square, making it easy, for example, to interpolate points along a great-circle path. The advantage of UTM co-ordinates over geographical co-ordinates (latitude and longitude) increases with distance away from the equator. The reason is that at mid and high latitudes, parallels of latitude are curved and meridians of longitude converge toward the pole.

2. DATA FILES

2.1 FILE NAMES

A file contains whatever data exist for terrain that lies within the area covered by a map sheet of the International Map of the World, a 1:1 000 000 standard that is supported by the Canadian National Topographic System and other nationional mapping systems. Each sheet covers an area 6° in longitude by 4° in latitude, and the area covered by the file matches this closely, but not exactly, as explained later. The files are also named after these maps, as illustrated in Figure 1. Thus the 1:1 000 000-scale map sheet that covers James Bay is called NN-17, and the corresponding file is called NN17. The 1:1 000 000-scale maps do no more that provide a framework for dividing the data into files. The data do not come from such small-scale maps.

2.2 REGION COVERED BY A FILE

The 6° of longitude covered by a file is the east-west extent of a single UTM zone. For example UTM zone 17 lies between 78° W and 84° W. At zone boundaries,

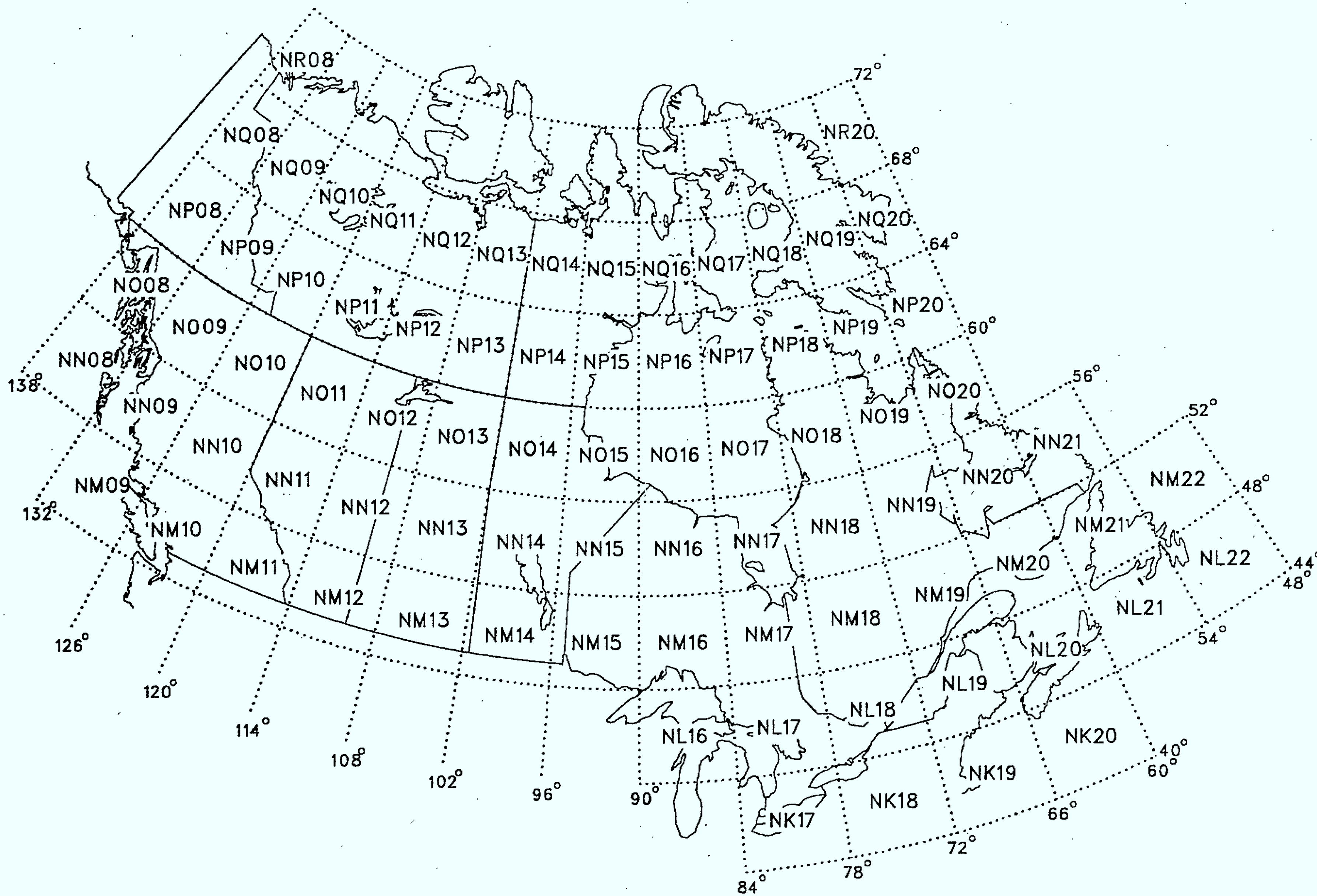

Fig. 1. Data-base file names.

there is a data overlap of at least 500 m on either side of the meridian. This overlap is necessary because each UTM zone has its own coordinate reference, and it would otherwise be difficult to interpolate the elevation of a point very close to the boundary between the two.

The northern and southern boundaries of the data region do not correspond exactly with the 1:1 000 000-scale map boundaries. This is because the data for a UTM zone are arranged in blocks that correspond to certain small rectangular areas on the ground (specified in the next section), and it is convenient to divide the zone along the boundaries of these blocks. An analogy may be found in some types of chocolate bar, in which the chocolate is arranged in squares. It is most convenient to divide the bar along the lines between the squares. The divisions were chosen to be those closest to the 1:1 000 000-scale map boundaries which occur every 4° of latitude. The northings of these divisions are given in Table 1 in the two columns with the heading UTM, along with the corresponding nominal latitude. The fifth column of Table 1 is explained later.

TABLE 1

NORTH-SOUTH EXTENT OF DATA-BASE FILE COVERAGE

Map Designation	Southern Boundary			Rows of Rectangles
	latitude	UTM (km)	UTM (500m)	
A	0°	0.0	0	29
B	4°	449.5	899	28
C	8°	883.5	1767	29
D	12°	1333.0	2666	28
E	16°	1767.0	3534	29
F	20°	2216.5	4433	28
G	24°	2650.5	5301	29
H	28°	3100.0	6200	28
I	32°	3534.0	7068	29
J	36°	3983.5	7967	29
K	40°	4433.0	8866	28
L	44°	4867.0	9734	29
M	48°	5316.5	10633	29
N	52°	5766.0	11532	28
O	56°	6200.0	12400	29
P	60°	6649.5	13299	29
Q	64°	7099.0	14198	29
R	68°	7548.5	15097	28
S	72°	7982.5	15965	29
T	76°	8432.0	16864	29
U	80°	8881.5	17763	29
	84°	9331.0	18662	

3. DIVISION OF DATA WITHIN A FILE

3.1 DIVISION INTO RECTANGULAR AREAS

The geographical region covered by a data file is divided into rectangular areas 7.5 km in the east-west direction by 15.5 km in the north-south direction. The boundaries of these rectangular areas form a grid in the UTM coordinate system, which are drawn at integral multiples of 15.5 km from the equator, and integral multiples of 7.5 km from the central meridian of the UTM zone. Data from each of these geographical rectangles are stored together as a record in a data file.

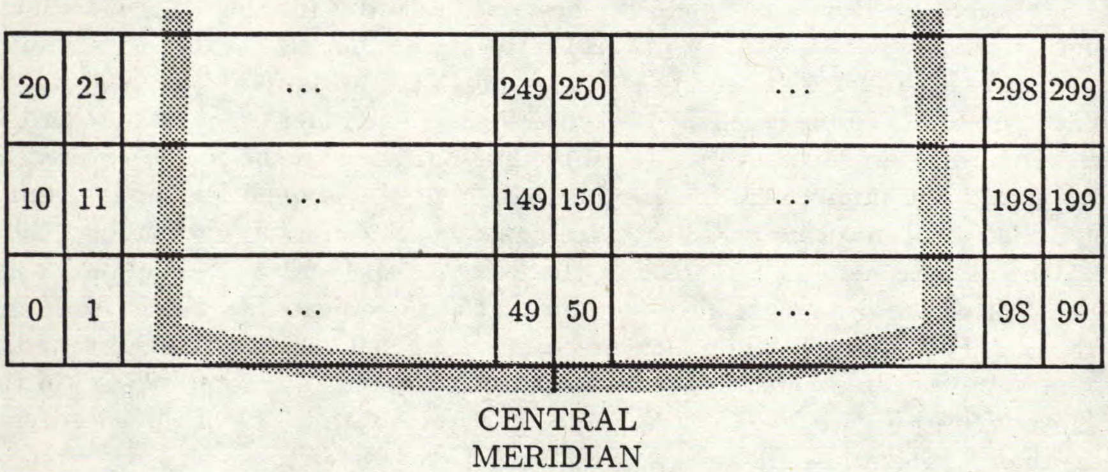

Fig. 2. Numbering scheme of rectangular areas represented in a file (first three rows). Each numbered rectangle measures 7.5×15.5 *km. The central meridian of a UTM zone is assigned an easting of* 500 *km. The shaded outline indicates the boundary of the corresponding* 1 : 1 000 000*-scale map. (The horizontal scale in the regions marked ... is compressed.)*

3.2 NUMBERS THAT IDENTIFY RECTANGULAR AREAS

The locations and numbering of the three most southern rows of these rectangular areas are indicated in Figure 2. For purposes of numbering, each row extends from 375 km west of the central meridian to 375 km east of it, and each such row contains 100 rectangles. As indicated in Table 1, there are either 28 or 29 rows of rectangles in the region covered by a file. The width of the region illustrated in Figure 2 is $100 \times 7.5 = 750$ km. This is larger than the width of a UTM zone, which reaches a maximum value of 668 km at the equator. Therefore the width of the region to be covered by a data file, the southern part of which is indicated by a shaded outline in Figure 2, fits easily within the limits imposed by the numbering scheme.

3.3 INDEX RECORDS AND DATA RECORDS

Each file is composed of a number of records, each 1024 bytes in length. These records are conceptual, rather than physical; there are no end-of-record characters separating them. The first few records in a file are illustrated in Figure 3. The first six records are reserved for the index to the file, and all the following records are data records. The data records may be written in any order, but the record number of each must be entered in the appropriate location in the index records.

Record numbers are indicated on the right-hand side of Figure 3. The index records are numbered from 1 to 6, and the first data record is number 7. In the index, each successive string of 16 bits (two bytes) corresponds to a geographic rectanglular area, in numerical sequence from 1 to 3072, as indicated in Figure 3. The last 2-byte position in the index (number 3072) is occupied by the number of the last record in the file. Since there are at most 29 rows of rectangular areas in the region covered by a file, no identifying number can be greater than 2899. Consequently, the file cannot be larger than 2905 records, or 2.9 Mbytes. In fact, it cannot be as large as 2905 records, because the rectangular areas at the extreme east and west of Figure 2 are always outside of the data area, even at the equator, as indicated in Section 3.2, and the corresponding identifying numbers are never used. (It may be noticed that the identifying number zero, which appears in Figure 2, cannot be accommodated in the index illustrated in Figure 3. This is unimportant, since zero is one of the identifying numbers that are never used.)

1	2	...	511	512	1
513	514	...	1023	1024	2
. . .					
2561	2562	...	3071	3072	6
1	2	...	511	512	7
1	2	...	511	512	8
. . .					

Fig. 3. The records in a file. The first six records contain the index while the remaining records contain data. Each data record corresponds to a numbered geographic rectangle illustrated in Figure 2. Each small box represents a 16-bit string. The numerals in the boxes simply count the boxes. The numerals on the right count the records.

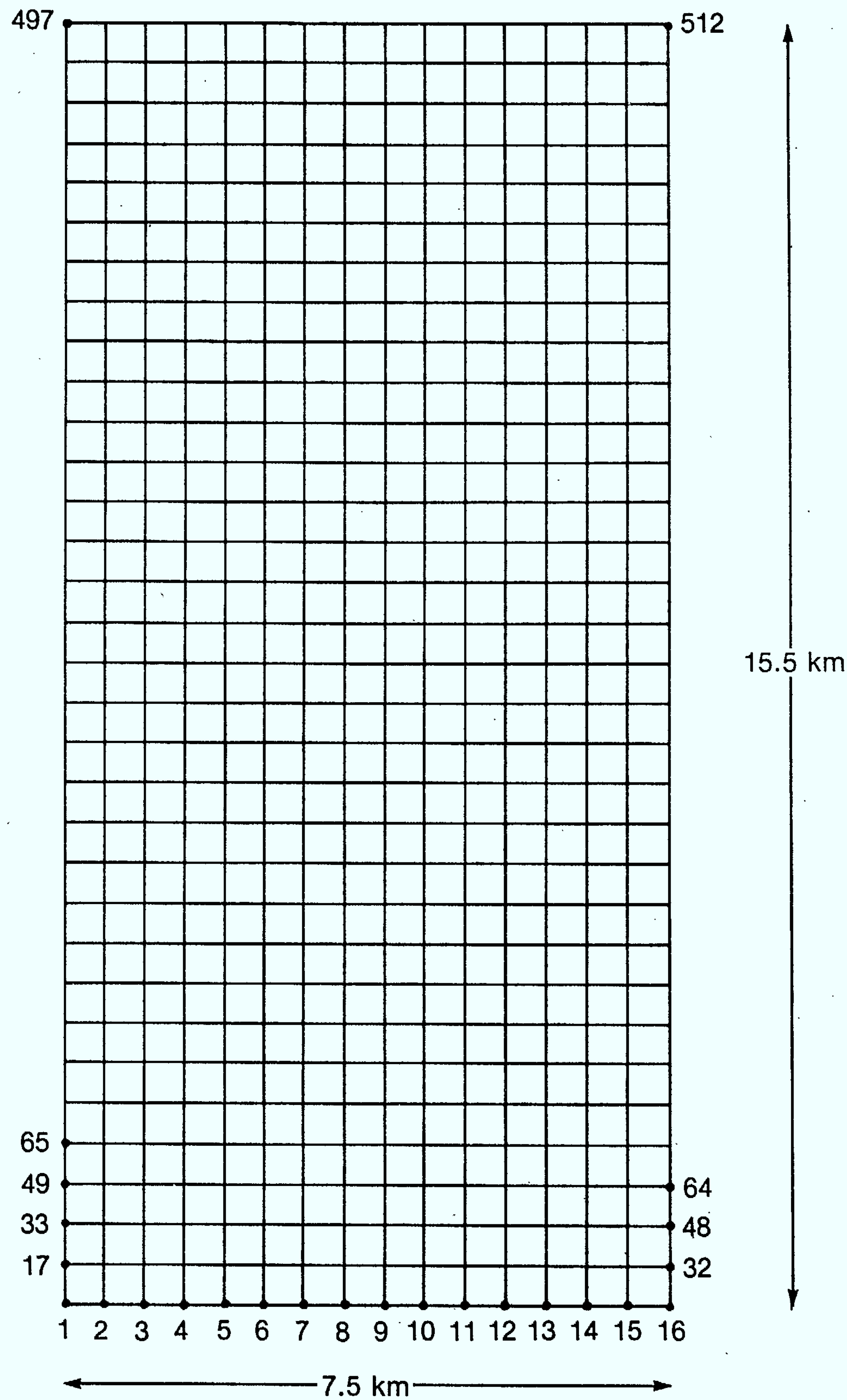

Fig. 4. Geographical area represented by one record. The whole area represented corresponds to a numbered rectangle illustrated in Figure 2.

3.4 POINTS IN A DATA RECORD

Data from 512 geographic points are written into one data record, which is 1024 bytes long. The arrangement is illustrated in Figure 4. The numbers on the diagram correspond to the order of the data in the record. The points form a square array, and are located at integral multiples of 500 m from the equator of the earth, and from the central meridian of the UTM zone in which they are located. The data on the eastern and northern edge of the rectangular area illustrated in the diagram are duplicated on the western and southern edges of adjoining rectangles. Because of this overlap, an elevation anywhere can be interpolated from the four nearest grid points by reading only one data record.

3.5 DATA POINTS

For each point on the 500-metre grid, there is provision for recording a surface type and elevation, as illustrated in Figure 5.

The surface type is a 3-bit code, 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 that 3 storeys high)
5 Marsh
6 Seawater
7 Urban Core (high density, tall buildings)

The elevation in metres above mean sea level is stored in 13 bits, in which it is possible to store numbers in the range 0 to $2^{13}-1 = 8191$. Zero is a valid elevation (sea level), but 8191 (1FFF hex) is used as an 'elevation not known' indicator. Therefore the range of elevations that can be stored is 0 to 8190 metres.

SURFACE	ELEVATION
0 2	3 15

Fig. 5. Storage in a 16-bit word. The surface code occupies the three most significant bits, and the elevation occupies the 13 least significant bits. The data illustrated here would be contained in one of the small boxes in a data record illustrated in Figure 3.

4. LOCATING A POINT IN THE DATA BASE

4.1 UTM CO-ORDINATES

If a geographical point is given in latitude and longitude, the UTM coordinates must be found. This is a common operation in cartography. One algorithm for making the conversion is described in [1]. In the UTM coordinate system, the location of a point is specified by the zone number and by an easting x and a northing y of the desired point in metres. (Although finding x and y is somewhat complex, the zone number is given simply by (186+longitude)/6 truncated to integer, where longitude is in the range $-180°$ to $180°$.) For dealing with a 500-metre grid, it is convenient to express x and y in units of 500 m. When that is done, the points contained in the data base occur at integral values of easting and northing.

4.2 FILE IDENTIFICATION

To identify the file that contains data for the point of interest, its UTM zone number and northing must be taken into account. First, the nearest latitude in the sequence $\lambda = 0°, 4°, 8°, 12°, \ldots, 84°$ is found. If the latitude of the given point is not known, it may be approximated for this purpose as $y/222.222$ where y is expressed in units of 500 m. Then the UTM northing y_B (in units of 500 m) of the nearest file boundary is then found from

$$y_B = 31 \left[7\lambda + \frac{1000 + (246 + \lambda)\lambda}{2000} \right] \tag{1}$$

All numbers in this formula are integers, and the result of the division is truncated to integer. Table 1 may be used instead of equation (1). The quantity y_B just obtained is the northing of the file boundary nearest to y, but what is really required is the southern boundary of the area covered by a file. Therefore, if $y_B > y$, then decrement λ by 4, and use (1) again. Now y_B is the northing of the base (southern boundary) of the area covered by the file, and λ is the nominal latitude of the base of the file.

The required file is now determined. The first letter of the file name is always N in the northern hemisphere, the second letter may be obtained from Table 1, and the numeral is just the UTM zone number. If, for example, $\lambda = 40$, and the UTM zone number is 17, the file name is NK17.

4.3 DATA RECORD WITHIN A FILE

The number identifying the 7.5×15.5-km rectangular area that contains (x, y) may be found with the following formulas:

$$I_y = (y - y_B)/31 \quad \text{truncated to integer} \tag{2}$$

$$I_x = (x - 250)/15 \quad \text{truncated to integer} \tag{3}$$

In (2) and (3), 15 and 31 are the dimensions of the rectangles in 500-metre units, and 250 is an offset that makes rectangle 50 occur at the UTM central meridian. The rectangle number is then

$$n = 100I_y + I_x \tag{4}$$

The number n is used to locate the position in the index that contains the entry for the desired rectangular area. The entry is found in the n'th 16-bit position in the file, where n ranges from 1 to 3072 (Figure 3). The 16-bit number found in that position is the number of the record in which the data for the geographical point in question are located. If that number is zero, then the corresponding geographic rectangle is not represented in the data base. If that number is not zero, then it must be in the range from seven to the number found in 16-bit position 3072.

4.4 DATA WORD WITHIN A RECORD

With the data record located, it remains to find the appropriate data word or words within the record. The elevation and surface code can be interpolated from the nearest data-base points in any desired way. The simplest is to use the four nearest data-base points. Let one of these data-base points have co-ordinates x_p and y_p, measured in 500-metre units. The quantities x_p and y_p are integers. The elevation and surface code for that point are found in the w'th 16-bit word of the record, where

$$w_x = x_p - 250 - 15I_x \tag{5}$$

$$w_y = y_p - y_B - 31I_y \tag{6}$$

$$w = 16w_y + w_x + 1 \tag{7}$$

In any 1024-byte record, the 16-bit words referred to here are numbered from 1 to 512. Once the desired 16-bit word has been extracted, it remains only to decompose it into surface code and elevation, according to the scheme illustrated in Figure 5.

4.5 READING DATA WORDS

Data may be read as characters, 16-bit integers, or 32-bit integers. If a file created according to the specification given in this report is read as characters, which are then converted to integers, then the data will come out correctly.

However, if a file is read as integers, the result depends on the particular computer being used. If the computer stores integers in its registers with the most significant byte first, the data will come out correctly. However, some processors, in particular the Intel series of microprocessors 80286, 80386 etc., invert the byte order within a binary number. Thus if the character string 'ab' is read as a 16-bit integer, the integer is $256b + a$ rather than $256a + b$. That is, the result is the same as reading 'ba' as a 16-bit integer by a processor that does not invert the bytes. Similarly, the string 'abcd' read as a 32-bit integer gives the same result as 'dcba' on other computers.

Data distributed on diskette before June 1, 1992 do not, in fact, follow the specification given here. This is because the they were written as 32-bit integers from a 80386-based microcomputer. Therefore, successive sequences 'abcd' were written as 'dcba'. If these data are used on another Intel-based computer and read as 32-bit words (as is done by the software supplied with the data), all will be well, and the byte inversion will be transparent. Otherwise, some byte re-ordering will be required.

Data distributed after June 1, 1992 follow the specification. Furthermore, in these data the first 32 bits of each file (which would otherwise always be filled with zeros), contain the file name, e.g. NL17. These characters allow the recipient to verify that the characters are in the correct order, simply by printing the first record of one of the files.

4.6 SUBROUTINES FOR ACCESSING DATA

The main purpose of this report is to allow recipients of the CRC terrain data base to extract data using their own methods. However, Appendix B offers a FORTRAN example of how the elevation and surface code for any given point may be extracted.

5. CONCLUSION

This report specifies the structure of the CRC topographic data base. The format allows for the storage of terrain elevation and a ground-cover code on a 500-metre grid. The structure is simple and compact, and is designed for rapid access to successive points that are within a few kilometres of each other. The data were compiled for use in radio propagation predictions, but may be useful wherever terrain data of this type are desired that cover a large geographical area but which have modest storage requirements.

6. REFERENCES

[1] Whitteker, J.H., *The CRC Topographic Data Base*, CRC Report No. 1353, Department of Communications, February 1982.

[2] Palmer, F.H., The Communications Research Centre VHF/UHF propagation prediction program: an overview, *Can. Elec. Eng. J.*, 6, No. 4, pp. 3-9, 1981.

[3] Whitteker, J.H., Propagation Prediction from a topographic data base, *IEEE International Conference on Communications, ICC '83*, pp. A1.1.1-A2.1.5, June, 1983.

[4] 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.

[5] *Grids and Grid References*, U.S. Department of the Army Technical Manual TM-241-1, 14 September 1962.

APPENDIX A

Geographical Coverage

The geographical coverage of the CRC topographic data base changes with time. This appendix can therefore only describe the situation at the time of writing. The general picture is shown in Figure A1. The cross-hatched area is contained in the data base as of June 1992. In addition, the region in Canada west of 115.5° longitude and south of 55° is covered, but is not shown, since it is expected to be replaced with new data. The areas filled with rectangles are available or soon will be available for adding to the data base.

The data have come mainly from three types of source. First are the data scaled by hand by the Department of Communications Regional Offices from 1:50 000-scale maps. The area covered by these data is the cross-hatched area in Canada. Data for a few maps within this area (in eastern Alberta) were provided under contract by the RCMP, and are of the same type. The elevations were interpolated from the contours on 1:50 000-scale maps, which are spaced every 25 or 50 feet or every 10 or 20 metres. They also contain surface-cover codes obtained from the same maps. It is expected that these data will not be superceded by any data from smaller-scale (1:250 000) maps.

Next are the existing data west of 113.5° W longitude, which were obtained from the Gravity Division of the Department of Energy, Mines, and Resources. The latitude range is from the U.S.-Canada border to 55° N. These data are not indicated in Figure A1. Although they were obtained from 1:50 000-scale maps, the distance between points in the source data was 1 km rather that 500 m, and sometimes mountain peaks are not well represented. Therefore it is expected that elevations in this area will be superceded by new data. Surface-cover codes are present, but they were obtained with a resolution much coarser that 500 m. Therefore large uniformly forested areas and large bodies of water will have the correct surface code, but it is not accurate on a small scale.

Finally, there are elevation data obtained from 3-arc-second data obtained in DTED (Digital Terrain Elevation Data, a NATO standard) format or the U.S. Geological Survey format, which is derived from the DTED format. Rectangles enclosed by solid lines indicate existing DTED, obtained from the Department of National Defence via the Department of Energy, Mines, and Resources. Rectangles enclosed on two or more sides by broken lines indicate DTED created under contract in 1992 by the Department of Communications. Cross-hatched areas in the U.S. indicate data from the U.S. Geological Survey. All these data are obtained from 1:250 000-scale maps. Although the source data have fine horizontal resolution (3 arc seconds $\approx$ 100 m), the vertical accuracy is limited by the contour spacing of the 1:250 000-scale maps, which is 100 or 200 feet. There are, at the time of writing, no surface-cover codes for these areas.

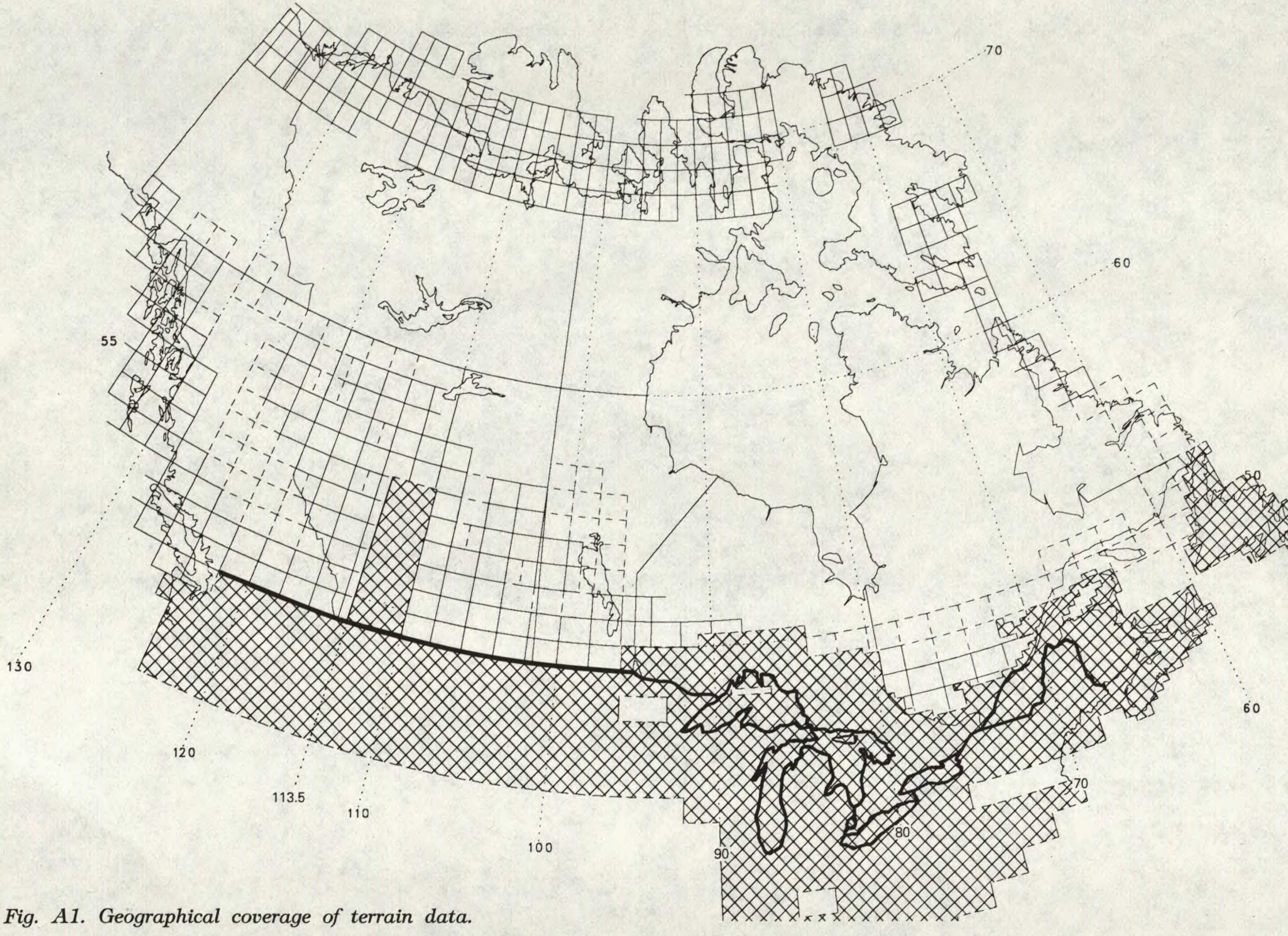

Fig. A1. Geographical coverage of terrain data.

APPENDIX B

FORTRAN subroutines to extract elevation and surface code for any point, given the UTM zone number, and the UTM coordinates easting (x) and northing (y) of the point.

```
* ---------------------------------------------------------------------------
* ---------------------------------------------------------------------------
      SUBROUTINE ELEV(NZONE,X,Y,LAND,H)
*  To find the elevation ELEV and surface code LAND at point NZONE,
*  X (easting, km), Y (northing, km) in UTM coordinates.

*  LAND is the terrain type code, determined by a weighted vote.
*  (LAND = -1 is a flag for data not in database)
*  H is the elevation at X,Y in metres, interpolated     3   4
*  from the 4 nearest grid points, numbered as:
*                                                        1   2
*  To enforce explicit declarations:
      IMPLICIT LOGICAL (A-Z)
*  Subroutines called:
      EXTERNAL INDEX,GETPAG,UNPACK
*  Variables and constants:
      LOGICAL OPENED
      CHARACTER*5 NAMFIL,NOWFIL,NMOPEN,MAPLET*2
      CHARACTER*1 BUFF(1024)
      INTEGER NZONE,LAND,IPAGE,NOWPAG,  NADD(4),LANDG(4),IHG(4)
      INTEGER INFILE, IX,IY, IOS, LL16,IC16,IC8, IDATA, J,L
      REAL  X,Y,H,  W(4),V(8), FX,FY,H1,H2,H3,H4, VMAX
      SAVE  NOWFIL,NOWPAG, BUFF
      PARAMETER (INFILE = 9)                                          I/O
      DATA NOWFIL,NOWPAG /'NOMAP',-1/
      DATA NADD /0, 1, 16, 17/

*  Truncate x,y coordinates:
      IX = 2. * X
      IY = 2. * Y
      CALL INDEX (1,IX,IY,MAPLET,IPAGE,LL16)
      IF (IPAGE .LT. 0)     LAND = -1
      IF (IPAGE .LT. 0)     RETURN
*  Is the page of data that we want already in the buffer?
      IF (IPAGE .NE. NOWPAG .OR. NAMFIL .NE. NOWFIL)    THEN

*     If not, synthesize desired file name:
         IF (NZONE.LT.10)  WRITE(NAMFIL,'(A2,''0'',I1)') MAPLET,NZONE
         IF (NZONE.GE.10)  WRITE(NAMFIL,'(A2,      I2)') MAPLET,NZONE

*        INQUIRE whether unit INFILE is connected to the correct file,
*        to avoid repetitive openings:
*        (It is not sufficient to verify that NOWFIL=NAMFIL
*        because the program may have just started up, in which case
*        no file will be open.)
         INQUIRE (UNIT=INFILE,OPENED=OPENED,NAME=NMOPEN)

         IF (.NOT. OPENED  .OR.  NMOPEN .NE. NAMFIL)     THEN
            OPEN(UNIT=INFILE,FILE=NAMFIL,ACCESS='DIRECT',RECL=1024,
     :      STATUS='OLD',IOSTAT=IOS)
            IF (IOS .NE. 0)     THEN
               PRINT*,' DATA-BASE FILE  ',NAMFIL,'  CANNOT BE OPENED.'
               LAND = -1
               RETURN
            END IF
         END IF

         CALL GETPAG (INFILE,'OLD',IPAGE,BUFF,IOS)
         IF (IOS .NE. 0)     THEN
*           Probably record IPAGE has not been written into the file.
```

```
            LAND = -1
            RETURN
         END IF
*        Map and page now in buffer:
         NOWFIL = NAMFIL
         NOWPAG = IPAGE
      END IF

*  LL16, as obtained from 'INDEX' is a 16-bit word.  It specifies
*  the location of the lower-left corner of the square that encloses
*  the point X,Y.  The corners of this square are:
 10   DO 18  J=1,4
*  16-bit-word address of this corner:
      IC16 = LL16 + NADD(J)
*  Byte address of this corner (second byte of the two):
      IC8 = 2*IC16
*  The surface type and elevation in these two bytes:
      CALL UNPACK(BUFF(IC8-1),BUFF(IC8),LANDG(J),IHG(J))
      IF (IHG(J) .EQ. 8191)     LAND = -1
      IF (IHG(J) .EQ. 8191)     RETURN
*  (8191 m is the highest elevation that can be stored;
*  It is 1FFF hex, and is used as a 'no elevation datum' indicator.)
 18   CONTINUE

*  Calculate weights
      FX = 2.*X - FLOAT(IX)
      FY = 2.*Y - FLOAT(IY)
      W(1) = (1.-FX)*(1.-FY)
      W(2) =      FX*(1.-FY)
      W(3) = (1.-FX)*FY
      W(4) =      FX*FY

*  Find elevations
      H1 = FLOAT (IHG(1))
      H2 = FLOAT (IHG(2))
      H3 = FLOAT (IHG(3))
      H4 = FLOAT (IHG(4))
      H = W(1)*H1 + W(2)*H2 + W(3)*H3 + W(4)*H4

*  Do terrain vote
      LAND = 0
*  Start with all votes = zero:
 30   DO 38  L=1,8
 38   V(L) = 0.
*  Check all 4 corners for the 8 possible surface codes (0 to 7):
*  Sum weighted votes:
 40   DO 48  L=1,8
      DO 48  J=1,4
 48   IF (LANDG(J) .EQ. L-1)    V(L) = V(L) + W(J)
*  Find which code has the greatest vote:
      VMAX = 0.
 50   DO 58  L=1,8
      IF (V(L) .GT. VMAX)  LAND = L - 1
 58   VMAX = V(LAND+1)
      END
```

```
* ------------------------------------------------------------------
* ------------------------------------------------------------------
      SUBROUTINE INDEX(INSTR,IX,IY,MAPLET,IPAGE,IWORD)

*  Universal index to topographic data, February 1990.

*  Arguments always used as input variables:
*     INSTR (stands for the word 'instruct') defines the desired task:
*       if INSTR > 0,   given geographic location, find page and word.
*       if INSTR < 0,   given zone and page, find geographic location.
*       if INSTR = 0,   given zone, find the largest page number.
*  Arguments which may be input or output, depending on 'INSTR':
*     IX is the UTM  easting in units of 500 metres.
*     IY is the UTM northing in units of 500 metres.
*     MAPLET is the first two characters, e.g. NM, of the file name.
*     (if INSTR > 0, IX,IY may be given any integer values.)
*     (if INSTR < 0, IX,IY is at the south-west corner of a page.)
*     IPAGE is the number of the page, or data block.
*  Arguments always used as output variables:
*     IWORD is the word number within a page (1 to 512).

*  To enforce explicit declarations:
      IMPLICIT LOGICAL (A-Z)
      CHARACTER LET1*1,LET2*1,MAPLET*2
      INTEGER INSTR,IX,IY,IPAGE,IWORD, MAXPAG
      INTEGER IYLAT,LAT,IYBASE,EBASE,IXPAGE,IYPAGE,IWX,IWY
      PARAMETER (EBASE = 250, MAXPAG = 2999)

*  The data are stored in pages of 512 words each.
*  Each page covers an area 7.5 km (x) by 15.5 km (y),
*  with points each 0.5 km.
*  The last column and row are reserved for duplicating
*  the elevations of the adjacent (on the map) pages.
*  Pages are ordered in rows on the map,
*  and grid points within pages are ordered in the same way.

*  A function to find the northing of a file boundary that corresponds
*  to a 1:1000000 map boundary (every 4 deg of latitude):
      IYLAT(LAT) = 31   *   ( 7*LAT + (1000 + LAT*(246 + LAT))/2000 )

*  Branch according to instruction:
      IF (INSTR .LT. 0)     GO TO 22
      IF (INSTR .EQ. 0)     GO TO 44

*  >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> Find MAPLET,IPAGE,IWORD
*  To find the page and word number in the terrain data base
*  corresponding to UTM coordinates IX,IY.
*  If the procedure fails, IPAGE=-1 is returned.

      MAPLET = '**'
      IPAGE = -1
      IWORD = 0
      IF (IX .LT. EBASE .OR. IX .GT. 2000-EBASE)   RETURN
      IF (IY .LT. -20000 .OR. IY .GT. 20000)       RETURN
*  Latitude of nearest map boundary:
      LAT = 4 * NINT(IY/888.888)
*  The base northing for this map:
      IYBASE = IYLAT (LAT)
*  The nearest boundary may be north of IY:
      IF (IYBASE .GT. IY)     THEN
```

```
         LAT = LAT - 4
         IYBASE = IYLAT (LAT)
      END IF
*   The map designation:
      LET1 = 'N'
      LET2 = CHAR (ICHAR('A') + LAT/4)
      MAPLET = LET1//LET2
*   Northing relative to the base:
      IYPAGE = (IY - IYBASE) / 31
*   Easting:
      IXPAGE = (IX - EBASE) / 15
*   Page of data base:
      IPAGE = 100*IYPAGE + IXPAGE

*   Find the word count within the page:
      IWX = IX-EBASE   - IXPAGE*15
      IWY = IY-IYBASE - IYPAGE*31
      IWORD = 16*IWY+IWX+1
*   This final + 1 changes the range of IWORD from 0-511 to 1-512.
      RETURN

*   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> Find IX,IY:
*   To find the IX,IY value of the origin of a given page
*   in a file defined by the letters MAPLET

  22  IX = 0
      IY = 0
      IF (IPAGE .LT. 0 .OR. IPAGE .GT. MAXPAG)     RETURN
      IYPAGE = IPAGE / 100
      IXPAGE = IPAGE - 100*IYPAGE
      LET2 = MAPLET(2:2)
      LAT = 4 * (ICHAR(LET2) - ICHAR('A'))
      IF (LAT .LT. 0 .OR. LAT .GT. 90)          RETURN
      IYBASE = IYLAT (LAT)
      IX = EBASE + 15*IXPAGE
      IY = IYBASE + 31*IYPAGE
      RETURN

*   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> Max page number:
*   Return the largest permissible page number.
*   This is independent of zone number, but depends on MAPLET.
  44  LET2 = MAPLET(2:2)
      LAT = 4 * (ICHAR(LET2) - ICHAR('A'))
      IYBASE = IYLAT (LAT)
      IY     = IYLAT (LAT+4)
      IPAGE = ((IY-IYBASE)/31) * 100  -  1
      END
```

```
* ------------------------------------------------------------------
* ------------------------------------------------------------------
      SUBROUTINE GETPAG (NUNIT,OLDNEW,IPAGE,BUFF,IERR)
*  Given unit number NUNIT and data-base page number IPAGE,
*  read the appropriate record into buffer BUFF.
*  IERR not equal to zero indicates an error.
*  If record IPAGE does not exist, and OLDNEW='NEW', a new record is created
*  Otherwise an error indication is returned.
      IMPLICIT LOGICAL (A-Z)
      CHARACTER*1 BUFF(1024),INDX(1024),IND6(1024)
      CHARACTER*1 HNUL1,HNUL2
      CHARACTER*3 OLDNEW
      INTEGER NUNIT,IPAGE,IERR, IREC,NOWPAG
      INTEGER KREC,N1BYTE,N2BYTE,J,I1,I2
      SAVE INDX,KREC,IREC,NOWPAG,N1BYTE
      DATA KREC,IREC,NOWPAG /0,0,0/
*  The buffers are used as follows:
*  INDX is a line of the file index.
*  IND6 is the last line of this index, containing the largest record number
*  BUFF is a line of topographic data.
*  Only BUFF is available externally; the index is hidden in this subroutine

*  Find the physical record number IREC corresponding to IPAGE.
*  The file index is contained in the first 6 records of the file:
      IERR = -1
      KREC = (IPAGE-1)/512 + 1
*  Trap for a page number out of the permissible range:
      IF (KREC .LT. 1 .OR. KREC .GT. 6)      RETURN
*  Read the appropriate record of the 6-record index:
      READ (NUNIT,REC=KREC,IOSTAT=IERR)   INDX
      IF (IERR .NE. 0)      RETURN
*  The 2-byte word containing the record number:
      N2BYTE = IPAGE - (KREC-1)*512
*  The second byte of the two containing the record number:
      N1BYTE = 2*N2BYTE
*  The record number:
      IREC = 256*ICHAR(INDX(N1BYTE-1)) + ICHAR(INDX(N1BYTE))
*  Read the record:
      IF (IREC .GT. 6)       THEN
         READ (NUNIT,REC=IREC,IOSTAT=IERR)   BUFF
      ELSE
*        There is no index entry for this record;
*        If a 'NEW' record is wanted, create a nul record:
         IF (OLDNEW .EQ. 'NEW')      THEN
*           Two succesive characters that make up the decimal number 8191:
            HNUL1 = CHAR(31)
            HNUL2 = CHAR(255)
   10       DO 18  J=1,1023,2
            BUFF(J)   = HNUL1
   18       BUFF(J+1) = HNUL2
         ELSE
            IERR = -1
            RETURN
         END IF
      END IF
      NOWPAG = IPAGE
      END
```