N85-23216 SQT D1
NASA-CR-175615
E85-10100

AN INVESTIGATION OF THE CRUSTAL PROPERTIES OF AUSTRALIA AND SURROUNDING REGIONS DERIVED FROM INTERPRETATION OF MAGSAT ANOMALY FIELD DATA

FINAL REPORT

JUNE 1983

B. David Johnson, Centre for Geophysical Exploration Research
Macquarie University
North Ryde NSW 2113, Australia

C.N.G. Dampney, Centre for Geophysical Exploration Research
Macquarie University
North Ryde NSW 2113, Australia

B.J.J. Embleton, CSIRO Division of Mineral Physics
P.O. Box 136
North Ryde NSW 2113, Australia

(E85-10100 NASA-CR-175615) [MAGSAT ANOMALY FIELD DATA OF THE CRUSTAL PROPERTIES OF AUSTRALIA] (MacQuarie Univ.) 88 p
HC A05/MF A01 CSCL 05B

N85-23215
THRU
N85-23220
Unclas
G3/43 00100

CONTENTS

1.0 PROJECT SUMMARY

This report comprises the investigations completed by the Australian Magsat Investigators at the end of the formal period of their investigations. A number of projects have received continuing development and the results of some of these are included as appendices.

1.1 The original objectives

The following objectives were stated in the original proposal to NASA.

a. To produce a magnetic anomaly map at constant elevation covering the whole of Australia and the surrounding oceans.

b. To produce a map of surface bulk magnetization of Australia.

c. To produce a crustal model of Australia and surrounding regions based on Magsat data and supported by available correlative geophysical data.

d. To develop an efficient array data base management system for manipulating, retrieving and displaying the Magsat data.

e. To produce maps of the geomagnetic field intensity, inclination and declination for the Australian region from global models of the geomagnetic field derived from Magsat.

1.2 Progress

The investigation of the Magsat data has not yet been finished and the stated objectives have not been completely met. Work is continuing and it is expected that all of the objectives will be met in the near future.

A magnetic anomaly map of the Australian continental region has been obtained by filtering the 2° averaged data set to remove effects of along-profile noise and between-profile leveling errors. This smoothed version of the published global 2° averaged Magsat map (Langel, et al., 1982) has been overlain on a tectonic map of Australia (Fig. 6.1). There are clear correlations between the large positive anomaly in south-central Australia with the Precambrian Gawler block and the less positive anomaly in south-western Australia with the Precambrian Yilgarn block. The anomaly signal over the eastern Australian Paleozoic fold-belts is very subdued due probably to relatively shallow Curie point depths. The anomalies in the north of Australia are probably distorted by processing effects in the 2° average data set which tended to enhance east-west contours. A spectral analysis of this 2° data shows that, for Australia, there is a processing bias giving rise to greater power at higher frequencies in the north-south direction than in the east-west direction (this effect shows up as an "east-west stripping" in the global anomaly map).

A selection of good-quality quiet-time passes has been made for the Australian region with a preference for passes that have their perigee (lowest elevation) within the Australian region. This produced a data set that was dominated by relatively low elevation profiles with small changes of elevation within the Australian area. Thus we were able to maximize the crustal anomaly field signal and to reduce the effect of variation in elevation of the satellite. An even geographic distribution of profiles was sought aiming for a mean profile interval of the order of half a degree of longitude. These selected data were reduced by removing the field model (MGST 4/81) and Dst corrections and then leveled to the filtered 2° average map. Maps of the dawn and dusk and combined profiles were made by color coding the anomaly values and plotting each profile as a colored swath. These maps show a combination of the location of the satellite profiles and of the relative values along adjacent profiles. The dawn and dusk maps show good local consistency but the combined map shows some differences. A more detailed comparison of the selected profiles and the interpolated filtered 2^{o} averaged map reveals the difference in resolution between the profiles and the averaged map.

Interpretation of the satellite anomaly map beyond the "arm-waving" stage has recently been initiated by studing the Broken Ridge satellite anomaly. Considerable effort has been expended to correct and modify the modeling program and to fully understand the problems inherent in the data when used for crustal anomaly interpretation.

Final anomaly and equivalent source magnetization maps are awaiting recent improvements in field models and in external field estimation.

The data base and graphics software development has been extraordinarily successfull and is being applied in a number of other applications. The reason behind this software development was based on earlier experience with POGO data in the Australian region (Mayhew, Johnson and Langel, 1980). The philosophy that developed out of this earlier work and experience in other fields lead us to seek an interactive environment for data selection and manipulation. We were thus able to examine every profile of data in the Investigator B quiet time data set and efficiently carry out the data selection procedures. The manipulation of subsets of the selected profiles becomes relatively simple due to the data base organization. It has also been our intent to build software systems that are state-of-the-art and useful building blocks for future work.

1.3 Public presentations (* denotes appended material at end of this report)

*December 1979: presentation of paper on the data base system to the Australian Association for Computer Aided Design, mini-conference on Computer Graphics and Spatial Analysis, held in Sydney.

August 1981: A preliminary presentation of Magsat anomalies over Australia to the Adelaide meeting of the Australian Society of Exploration Geophysicists.

March 1982: Workshop on processing of Magsat data with R.A. Langel at Macquarie University.

*October 1982: paper on data selection techniques to the 52nd Annual Meeting of the Society of Exploration Geophysics held in Dallas, Texas.

*May 1983: paper on the interpretation of the Broken Ridge anomaly to the American Geophysical Union meeting held in Baltimore.

*May 1983: paper on precedency control in data bases to the Association of Computing Machinery Data Base week meeting at San Jose, California.

August 1983: invited paper to be given to the International Union of Geodesy and Geophysics in Hamburg on the status and future of crustal anomaly investigations (with Mayhew and Wasilewski).

*September 1983: paper to be given on the graphics package to the meeting of the Australian Institute of Engineers in Canberra.

September 1983: paper to be given on the data base at the CSIRO workshop on Data Bases in Brisbane.

2.0 DATA UTILIZED IN THIS PROJECT

Four sets of data have been acquired for this project, mainly through the World Data Center at Goddard Space Flight Center. These are:

a. The Investigator-B quiet-time selected data tapes containing the observed vector data, the observed scalar data, the location data for the satellite in geocentric coordinates, and the field model values at each satellite location using the MGST 4/81 field model. The data set is decimated from the original chronicle data set, the resulting sampling interval being of the order of 35 kms along profile. Three geographic selections were made for Australia, the Indian Ocean and Antarctica--although in hindsight it would have been easier to deal with a global set.

b. The global 2 degree averaged data set containing averages of the computed scalar field over 2^o by 2^o latitude-longitude bins for all elevations. The profiles had been corrected for field model using MGST 4/81, Dst corrections. Linear fits were also removed to reduce the between profile differences. The data set included the averaged computed scalar field, the standard deviation for each average, the average elevation and the number of data values within each average bin.

c. Supplementary data sets including topography, gravity and heat flow have been obtained from various sources but have not yet played an important role in the investigation.

d. The chronicle data tapes containing all the original data obtained from spacecraft with the orbit and altitude calculations completed. The shear quantity of the data tapes (of the order of 100) have caused several problems which have lead us not to use these tapes. The first problem that arose was the damage caused to some of the tape reels due to inadequate packaging between and around the cannisters. Some of the reels were damaged beyond repair and one tape arrived without its reel! It is recommended that tapes be shipped in smaller quantities and special attention be given to protecting the reel hub.

The second problem that arose was due to the volume of tapes involved and our inadequate resources for handling and storing such quantities of tapes. This problem is not likely to be resolved until Macquarie University acquires 6250 bpi tape drives.

We have received the data tapes requested in the agreement and have found the World Data Center to be most helpful in meeting our requests. The tapes that have been utilized have been read successfully and without difficulty--some minor problems we encountered were apparently due to tape head misalignment.

3.0 SYSTEMS SUPPORT

This section is documented in this report in order that readers may benefit from some of the experience we had in developing a suitable combination of hardward and software. More detailed reports of the software systems briefly described here are appended.

The shear volume of Magsat data and the diverse nature of the influences effecting it requires a rational approach to the development of computer-based systems support. The fundamental specification that guided our systems development was the need to interact with the data on a pass-by-pass basis. This interactive requirement demands that the computing environment be capable of supporting on-line processes which reduce, manipulate, display and interpret the data interactively.

Four major software supporting systems have been developed. These are for:

1. data conversion from NASA supplied tapes,
2. active control parameter input,
3. versatile active graphics displays, and
4. the data base system.

3.1 Hardware - main frame

Due to the interactive requirements of the project we chose to use the VAX 11/780 rather than the Univac 1106. The VAX is equipped with 4 megabytes of core, 4 disc drives, 80 terminals, a tape drive and a line-printer. The computer serves a large part of the computing needs within the university and we did experience some access problems. However, these were offset by the high degree of suitability of the VAX to interactive computing.

A key factor in the success of an interactive computing project is an adequate amount of disc space. Considerable effort was required to maintain our disc space and also to obtain sufficient terminal access. With this in mind, we were able to carry out a complex data processing task on a heavily used small computer system.

3.2 Hardware - graphics facility

The graphics hardware we used in this project was part of a research facility largely dedicated to Magsat. The equipment comprises a Tektronix 4027 color graphics terminal together with a Tektronix 4663 flat-bed digital plotter. The connection to the main frame was via a standard RS232C connection with transmission rates varying between 2400 and 9600 baud.

The Tektronix 4027 terminal served our purposes very well as it is a useful and versatile terminal in addition to its color graphics capability. The color graphics display is of sufficient resolution (order 700 x 700) for most purposes and has up to 8 user definable colors. There is considerable flexibility in the color choice and in the construction of patterns for mixing colors. The screen can be split into a region for graphics and another for communication with the main frame. We have found that the 4027 provides a satisfactory mix between a highly versatile terminal and high resolution graphics. The use of color in the display of information is an interesting area of research in its own right.

We also photographed the screen using a Nikon FE camera with 50-200 mm zoom telephoto lens. The camera was placed at a distance of about 2 meters to reduce distortion and as close to the axis of the screen as possible. We were able to produce presentation quality transparencies simply and cheaply by this method. Some of the figures in this report are obtained from these slides.

3.3 Software - data base system (GADB)

The basic concept of the data base system that we have developed is that profiles of geophysical data should be stored as arrays or sequences of data. Each variable, dependent and independent, is placed in a separate array. Thus, we can store all the information related to a single profile by storing a small number of arrays. The length of the arrays are the same and must be determined at the start but the number of different variables may be changed very easily at any stage. In geophysical data processing, the lengths of the data sets are normally predetermined but the choice of additional parameters is often made at a late stage in the processing.

The advantages of such a data base system are particularly great when implemented on a computer supporting a hierarchical file structure. In these systems disc storage is controlled by a tree structure directory or catalogue. There are basically 3 levels within our data base hierarchy (although this could be increased). The uppermost level contains the names of projects in the data base. The middle level contains the names of the profiles within each project. The lowest level contains the data associated with each profile. A data dictionary is used to define the set of variables for each project and their data types.

The net result of such an organization is that a particular variable of a particular profile may be readily accessed by name. Thus, it is very simple to compare two profiles by retrieving the appropriate variables for each profile and then displaying them. There is no longer any need for long sequential searches through the entire data set just to pick up one or two profiles.

The complexity of our data base system is due to the incorporation of various features which help to protect the data base from inadvertent corruption. There is also a "roll-back" capability in which we can revert the data base to its situation at a given time in the past.

We have been careful to adopt a useful naming convention for our profile names. The profile names include the pass number used by NASA and the mean longitude of the profile. Directory searches can, therefore, be made to scan for particular pass numbers and ranges of longtitudes. In hindsight, we should also have incorporated an indicator showing if the profile was ascending or descending to assist in separating these profiles.

(Appended: Dampney (1983) GADB - A data base facility for modeling naturally occuring geophysical fields)

3.4 Software - graphics system (GP-GRAPH)

The use of the VAX for our computing requirements has forced us into a position where we had to build a complete graphics software system as there was none available. (In any case we would have had some graphics development work to do for the Univac.) The basic philosophy that we employed was a requirement that programs using the existing Calcomp-style plotting routines be run without modification. These basic calls were then to be extended to provide access to the more advanced features available on the terminals and plotters that we could use. In order to comply with programming standards we adhered to the recommendations of the SIGGRAPH Core Graphics standards.

The GP-GRAPH system essentially comprises three sets of subroutines: core, device and applications. The core module contains those subroutines which are common to all devices and carry out much of the "leg-work" of plotting. These may be accessed in a high-level Calcomp-style form (e.g. call plot, axis, line, etc.) or at a low-level Tektronix-style form (e.g. call move, draw, etc.). The device module contains those subroutines which are specific to each device and provide an implementation of the graphic call on that device. In some cases a software emulation is carried out where the call is essentially a hardware feature of a different device (e.g. in the filling of a polygonal outline by a color or pattern).

The core module and the user application program are gathered together with the required device module at run time. Some other general purpose graphics packages are implemented by collecting together a set of routines containing all the possible device drivers which tends to make even simple graphic programs very large in executable form. Our implementation requires the user to define the graphic device that is required. A null device may be configured for bypassing the graphics routines.

A number of application routines have been developed to perform some of the more common graphic applications. At present we have application routines for contouring, surface display (fish-net) and for multivalue profile plotting. These will be augmented as they are required.

The range of output devices that are currently implemented includes the Tektronix 4027 color graphics terminal, the 4010-series Tektronix terminals, the Tektronix 4662 and 4663 plotters, various character display terminals with and without full screen addressing, printer plotters including the Diablo and the Printronix. Skeleton device drivers are also provided to enable new devices to be configured in a simple manner.

We have implemented all the capabilities of the Tektronix 4027 as emulations for other devices. Care needs to be exercised in some of these emulations as the screen overwrites any existing plotting whereas hard-copy devices superimpose. Segmentation has been implemented whereby frequently plotted subjects may be stored and replayed at future occasions.

Three-dimensional graphics routines are currently under development but as they were not essential to this project, they have been a low-priority development.

(Appended: Gillings, Johnson and Dampney (1983) Design and Implementation of a Device Independent Active Graphics Package)

3.5 Software - data conversion and selection

The program MSTP2GA was written to convert the NASA supplied Investigator-B tapes into VAX internal format, gather the data into passes, select passes that satisfied certain criteria and store them in our data base system.

3.5.1 Data conversion module

This module carries out the following operations:

a. reads physical blocks of data from the Magsat Investigator-B tapes;

b. converts data fields from internal IBM form to VAX internal form according to whether the items are real, integer or character;

c. checks and reports errors in header information, block length and representation errors;

d. flags data fields containing representation errors;

e. generates time of every data point;

f. logs summary information for each data block;

g. writes the data block into an internal buffer which is accessible by the data gathering and selection module.

In general the major source of data errors was dropped frames during tape reading. This can be caused by a variety of factors including poor tape transportation, storage and handling. Relative alignment problems between writing and reading can also produce errors. See Table 3.1 for a summary report on data errors.

Recovery from these tape reading problems was automatic and usually successful. In the future we would recommend that alignment fields be included in each data block. These alignment fields would contain known information and would assist data checking and recoverability

3.5.2 Pass gathering and selection module

The basic unit of data used in our processing was an array containing all the measurements of a given field along a specified portion of a pass. Data provided by NASA was presented in blocks containing various single scalar fields plus 25 arrays each 30 elements long.

Passes commence and terminate near the south pole. If necessary data blocks are padded so that each new pass starts on a block boundary, the data blocks containing data in the required geographic region were accepted and combined together. Once all the data blocks had been gathered together the individual data points lying outside the goegraphic area were deleted. Data selection up to this point in the processing included quality control and geographic search criteria. Following this a number of other search criteria were available including:

a. by altitude--either by selecting data obtained below a given maximum altitude or by selecting the pass if containing a perigee point (lowest altitude of orbit) within the geographic region;

b. by data length--if the number of data points was greater than a given minimum number;

c. by day of pass; and

d. by number of pass-either singly, multiply or as a specified set.

Having passed these automatic selection criteria the data is further processed to remove the field model values provided and to correct for the Dst (ring current) correction. Data fields containing representation errors are interpolated across. Noise spikes above a certain amplitude are detected and eliminated by replacing them with an interpolated value.

Interactive selection techniques were then employed using the active graphics package GP-GRAPH.

The graphics module allows the user to select the field(s) to be plotted and includes some editing capabilities including further despiking and detrending. Every pass of data is viewed using the graphics module before it is finally accepted or rejected by the user for entry into the data base. Fig. 3.1 shows examples of two of the displays available to the user.

All essential information, including interactive input, is logged for each pass so that a permanent record of the selection procedure is available. An example of the queries and responses of an actual selection run is shown in Table 3.2

Table 3.1 - Summary of data errors encountered during processing Magsat investigator tapes

Best case

Tape OF8023-1 File 1
Blocks 1 to 2206
Pass # within range 0012 to 1170
No errors

Worst case

Tape OF0513; 14 1 of 2
Blocks 1001 to 3549
Pass # 1735 to 3089, 1205 to 1250

Unnormalized data	Overflow VAX 11/780	Underflow VAX 11/780
4103	14580	12579

Length of Header block errors	28
Length of data block errors	188

Summary

Best case error rate - 0%
Worst case error rate - 1%

The cause of all errors was most likely an error in alignment of physical block to its fields caused by 1 or more dropped bytes.

Table 3.2 - Control data accepted by the Magsat data processing system

```
What is the name of the Magsat file? >MTAO:
Is this a restart? (Y/N) >Y
Number of files to skip >0
Number of blocks to skip >830
Dump of every block? (Y/N) >N
Record of the data errors? (Y/N)>Y
Log processing? (Y/N) >Y
Input from tape? (Y=tape/N=disc) (Y/N) >Y
  Enter tape label  OF8023-1 FILE 1
Enter data into data base? (Y/N) >Y
What is the Data Base name? >MS
What is the data dictionary name? >MS
Clear the data base every how many profiles? >4
Check termination of run every how many profiles? >25
What is the time increment tolerance factor? >3.0
Specify selection criteria? (Y/N) >Y
  Select (and chop) by latitude? (Y/N) >Y
    Latitude minimum >-50.
    Latitude maximum >0.
  Select by longitude? (Y/N) >Y
    longitude minimum >90.
    longitude maximum >180.
    Select another longitude strip? (Y/N) >N
  Select by altitude? (Y/N) >Y
    Select bottoming profiles? (Y/N) >Y
      Maximum altitude for bottoming profiles >450.
    Select other profiles?  (Y/N) >Y
      Maximum altitude for other profiles >400.
  Select on pass number? (Y/N) >N
  Select on year and day? (Y/N) >N
  Select only profiles with more than a minimum of points (Y/N) >Y
    Enter minimum number of points >30
  Examine and select profiles yourself? (Y/N) >Y
Specify processing parameters? (Y/N) >Y
  Remove external field? (Y/N) >y
  User specified automatic despiking of DMAGTVEC? (Y/N) >y
    Enter DMAGTVEC minimum value >-100.
    Enter DMAGTVEC maximum value >100.
    Size of smallest spike to remove >5.
Have you made a mistake? (Y/N) >N
```

4.0 DATA SELECTION

The data, provided by NASA/Goddard Space Flight Center, is taken from the Magsat Investigator "quiet-day" data set for the Australian region, from 90°E to 180°E and from 50°S to the equator. This Australian data set consists of some thousands of near north-south profiles of 3-component magnetic field measurements together with position information consisting of latitude, longitude and altitude for each data sample. The altitude range of the satellite was from 300 to 500 kms above the earth's surface except for the last few orbits which were lower.

The data sample interval of the Investigator data set is approximately 35 kms. Since most of the orbits were no lower than 300 kms this sample rate was deemed sufficient to recover all the crustal anomaly information. The effective data sample interval perpendicular to orbit direction is given by the number of profiles and the longitude range of the study area. Provided that profiles are chosen that are spread evenly over the longitude range, then it should be sufficient to reduce the number of profiles required to about 200.

This apparent redundancy of the data (by an order of magnitude) enables the rejection of profiles which are not long enough, have gaps in the data, have high noise levels or contain time-varying effects. Since the signal-to-noise ratio decreases with increasing altitude, profiles obtained at lower altitudes are preferred to those obtained from higher altitudes.

The initial search through the data tapes was for passes having their perigee (lowest point in orbit) located within the geographic region. This resulted in 45 passes which formed a surprisingly good basis for our final data set. Subsequent searches were made for other good quality passes which also lay below altitudes of 400 kms. This resulted in a total of 179 passes. In order to achieve a more uniform geographic coverage these have been culled by logically deleting those passes which essentially duplicate other passes.

Fig. 4.1 shows some plots of near duplicate passes falling within 1° of each other in longitude. The pass numbers shown in this figure are:

"west trending 126 profiles" - 2000, 1860, 2679

"east trending 131 profiles" - 1076, 2381, 1463

"west trending 133 profiles" - 1953, 2062

"west trending 139 profiles" - 0621, 1875, 0498, 0575

Table 4.1 shows a list of 113 passes which have finally selected for the purposes of map generation. This contains 65 ascending and 48 descending passes. The * denotes those passes that perigee within the area and A and D denote ascending and descending passes respectively. Table 4.2 contains those that were initially selected (and therefore containing good data) but not used.

The passes plotted in Fig. 4.1 show that after field model and Dst correction there remains a time varying effect which shows up as level differences between near coincident passes. The source of these differences is due to incomplete external field removal and is the subject of a detailed study at NASA. Improvements are being made in both the field model and the Dst correction procedure and it is to be hoped that these level differences will not remain a problem. The standard procedure is to adjust each pass by removing a linear or quadratic polynomial. However, these procedures appear to give rise to unwanted features and have no physical basis. It

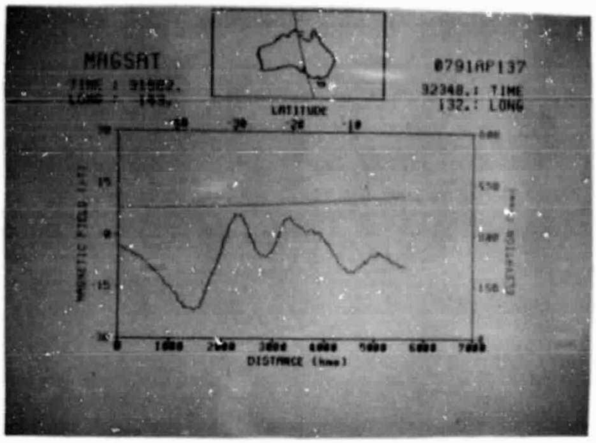
Fig. 3.1

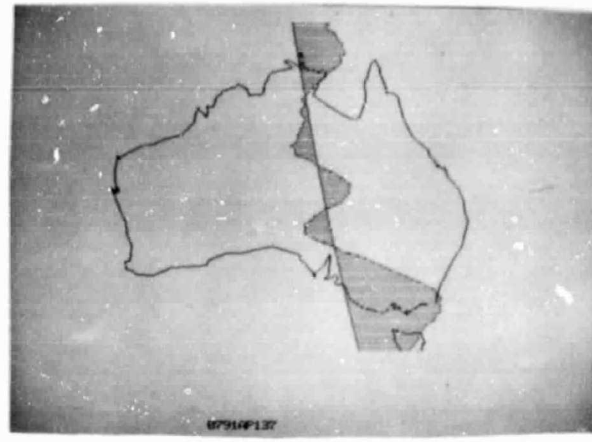
Examples of profile displays available to the user during data selection.

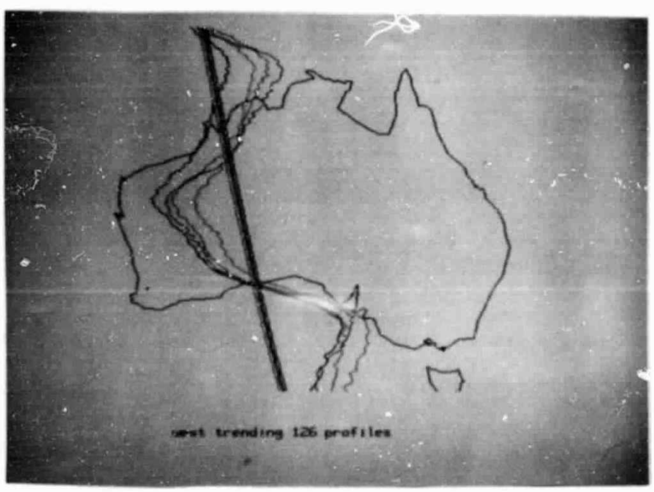
Fig. 4.1

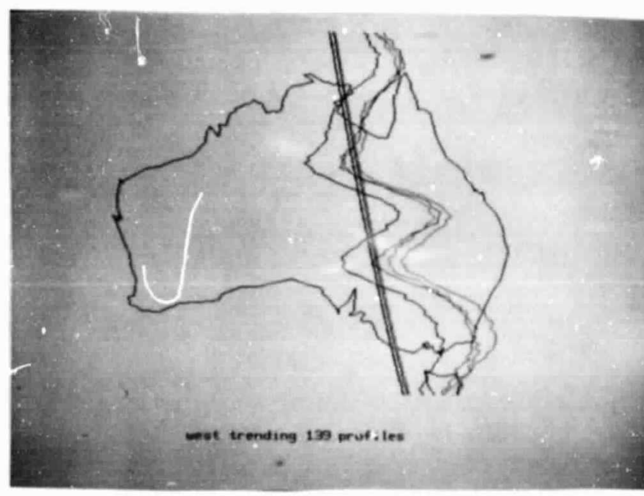
Examples of near duplicate track comparisons after field model and Dst corrections have been made.

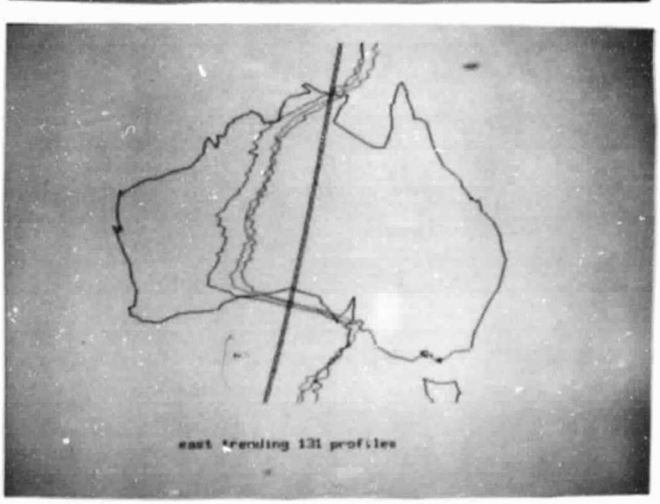
ORIGINAL PAGE
COLOR PHOTOGRAPH

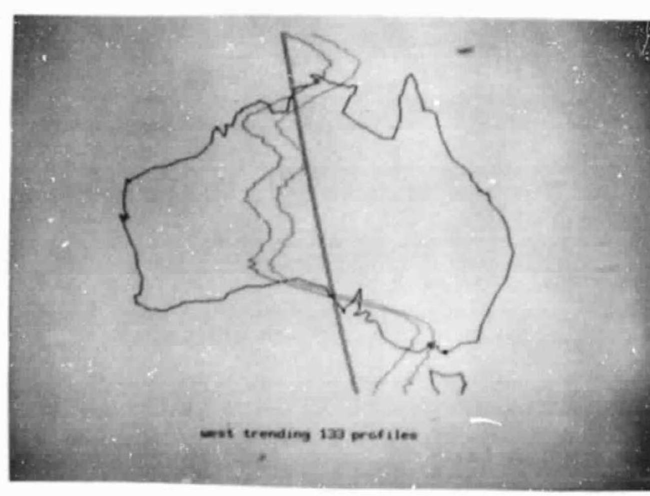










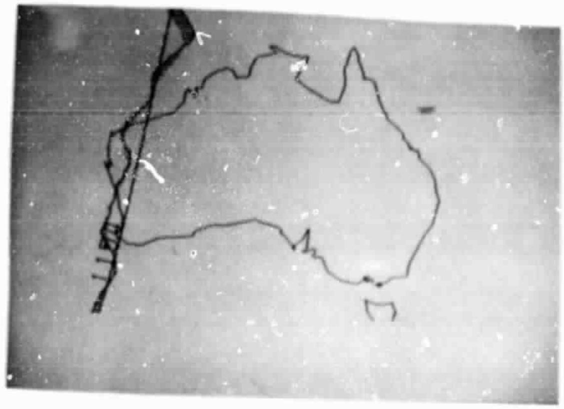


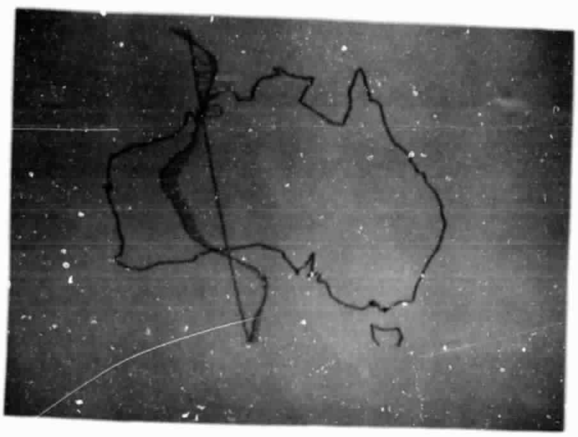
was therefore decided, as a interim measure, to adjust the passes to the 2° average data set after it had been filtered (see section 5). Fig. 4.2 shows a number of passes which have been linearly adjusted to the 2° data--the solid line is the adjusted anomaly profile while the east west bars join these to the interpolated value from the 2° average map. It can readily be seen that the 2° average map contains a surprising amount of the information present in the anomaly profiles. Some differences can be seen which may be due to elevation effects, small scale anomalies and auroral effects to the south. Some profiles show large differences to the north but these are probably due to the linear fitting effect, described in section 5, in the 2° average data. For gross correlation with geological occurences the 2° average data is sufficient but for detailed geological interpretation the profiles contain critical information.

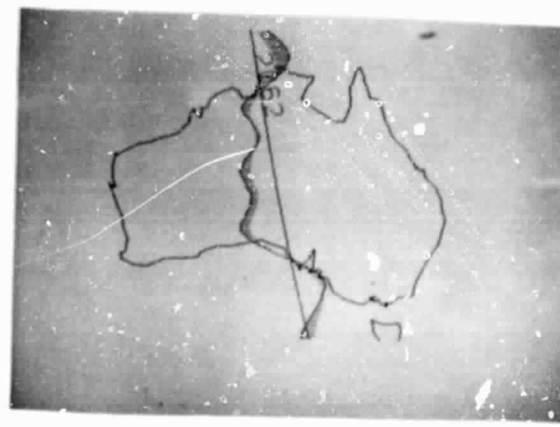
After adjusting the individual passes to the 2° average data the passes were displayed as colored swaths (Fig. 4.3). Separate plots of the descending (dawn) and ascending (dusk) passes were made and it can readily be seen that there is good local coherence between adjacent passes. The combined plot of ascending as descending passes is less good and probably reflects the differences between the dusk and dawn fields.

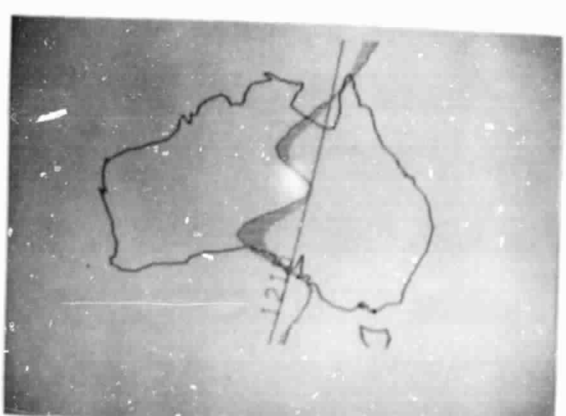
Fig. 4.2

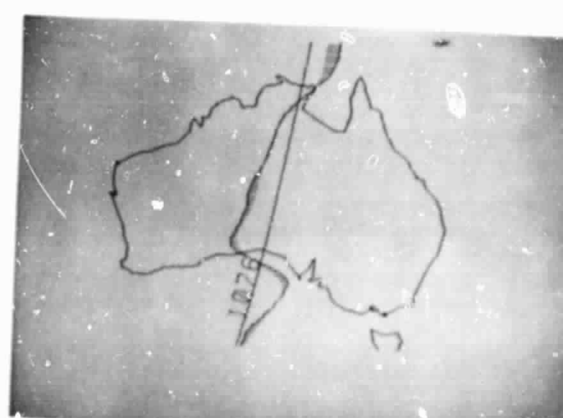
Examples of individual passes adjusted to the 2^o averaged data interpolated to the satellite location.











2° averaged data filtered

Swath map of dawn
(descending) passes

Swath map of dusk
(ascending) passes

Swath map of all
selected passes

Fig. 4.3

Table 4.1

Magsat Passes Accepted Over Australian Region

Pass Number	Per.	Average Long.	A/D	Pass Number	Per.	Average Long.	A/D	Pass Number	Per.	Average Long.	A/D
0374		159	A	0376		112	A	0389		168	A
0390		144	A	0406		130	A	0420		163	A
0421		139	A	0576	*	116	A	0590		149	A
0591	*	125	A	0592	*	102	A	0605	*	158	A
0606	*	135	A	0607	*	111	A	0622	*	121	A
0637	*	131	A	0638	*	107	A	0653	*	116	A
0730		120	A	0744	*	153	A	0759		163	A
0760		140	A	0761		117	A	0790		160	A
0791		137	A	0792		114	A	0805		170	A
0807		124	A	0821		157	A				
0913		172	A	0951		163	D	0952		140	D
0953		117	D	1014		135	D	1015		112	D
1044		157	D	1045		133	D	1046		110	D
1059		167	D	1060		144	D	1061		121	D
1075		155	D	1076		131	D	1106	*	153	D
1107	*	130	D	1108	*	106	D	1138	*	128	D
1139	*	105	D	1152	*	162	D	1153	*	139	D
1154	*	115	D	1168	*	150	D	1169	*	127	D
1170	*	104	D	1185	*	115	D	1214	*	160	D
1215	*	137	D	1229		170	D	1230	*	148	D
1245		159	D	1293		123	D	1294		099	D
1323		145	D	1400		155	D	1401		132	D
1433		108	D	1446		166	D	1448		120	D
1449		097	D	1478		143	D	1781		156	A
1859		149	A	1861		103	A	1874		162	A
1875		139	A	1891		128	A	1892		105	A
1905		164	A	1952		156	A	1953		133	A
1970		100	A	1998		173	A	2000		126	A
2001		103	A	2030		153	A	2032		106	A
2063		110	A	2077		147	A	2078		123	A
2092		160	A	2124	*	141	A	2155	*	145	A
2156	*	122	A	2157	*	099	A	2217		154	A
2264		150	A	2265		127	A	2266		104	A
2270		169	D	2271		146	D	2281		118	A
2301		174	D	2342		152	A	2350		125	D
2357		167	A	2360		098	A	2364		163	D
2365		140	D	2381		131	D	2490		143	D
2545		162	A	2642		096	A	2924		124	

Table 4.2

Magsat Passes Selected But Not Used

Pass Number	Per.	Average Long.	A/D	Pass Number	Per.	Average Long.	A/D	Pass Number	Per.	Average Long.	A/D
0375		135	A	0405		153	A	0497		162	A
0498		139	A	0575	*	139	A	0608	*	088	A
0652	*	140	A	0654	*	094	A	0745		130	A
0837		144	A	0838		121	A	0914		149	A
0915		126	A	1013		159	D	1123	*	116	D
1155	*	093	D	1186	*	091	D	1247		112	D
1384		167	D	1385		144	D	1462		155	D
1463		131	D	1767		120	A	1843		160	A
1858		172	A	1860		126	A	1876		115	A
1954		110	A	1969		123	A	1999		149	A
2031		130	A	2061		156	A	2062		133	A
2076		170	A	2079		100	A	2093	*	137	A
2094	*	114	A	2107	*	173	A	2108	*	150	A
2123	*	164	A	2125	*	118	A	2126	*	095	A
2154	*	168	A	2236		122	A	2263		173	A
2272		122	D	2282		095	A	2287		137	D
2344		106	A	2359		121	A	2366		117	D
2413		115	D	2489		166	D	2547		116	A
2595		094	A	2608		156	A	2610		110	A
2616		130	D	2623		172	A	2624		149	A
2625		126	A	2656		136	A	2679		126	D
2687		144	A	2907		153	A	3019		120	A

5.0 ANALYSIS OF THE MAGSAT TWO-DEGREE AVERAGE DATA SET FOR THE AUSTRALIAN REGION

The published map of the Magsat scalar magnetic anomalies (Langel, et al., 1982) was constructed from a selection of quiet day profiles. The geomagnetic field model MGST(4/81) (Langel, et al., 1981) was removed from the data. Linear fits were made to segments of the data and these were removed. The resulting data were averaged over 2^o x 2^o bins and these values were then contoured.

The resulting map shows many of the same features as the equivalent maps for the POGO data (Langel, et al., 1982). Some additional features show up in the Magsat data due to the better quality of the data set and lower altitude of Magsat.

Figure 5.1 is a contour plot of the 2^o average data set for the Australian region (90^oE to 180^oE and 50^oS to 0^oN). No projection has been applied to the geographic coordinates. The contouring method uses linear interpolation and hence, tends to enhance irregularities in the data.

A two-dimensional Fourier transform (2DFFT) was applied to the Australian region data. A plot of the contours of the logarithm of power is shown in Fig. 5.2, plotted in wave number space. Zero wave number is in the center and maximum wave numbers (Nyquist frequencies) are at the boundaries. The spatial frequencies of the maximum wave numbers are equal since the sampling interval is the same in both directions.

Two significant bands of noise appear in the transform adjacent to the zero wave number axes. There is relatively high power in a band close to the zero east-west wave number axis extending to maximum wave number (Nyquist frequency) along that axis (at the top and bottom of the figure). This power is interpreted as representing the presence of high frequency noise in the profile data. This "point-to-point" noise appears as wave forms of high frequency in the north-south direction (along the profile) and constant in the east-west direction.

The second band of noise is less well defined and lies along the zero north-south wave number axis. The source of this noise is interpreted as being due to the incomplete removal of external field effects which gives rise to small level differences between profiles. This "profile-to-profile" noise appears as wave forms of high frequency in the east-west direction (perpendicular to profile) and constant in the north-south direction.

A filter was applied to the two-dimensional transform in order to reduce the effects of these two types of noise as much as possible. The filter was defined as being circular in wave number space and accepted all power within a wave number corresponding to 80% of the maximum wave number in the north-south direction. This is actually an elliptical filter in frequency space. The maximum frequencies correspond to wavelengths of about 2.5° in the north-south direction and about 4° in the east-west direction. An inverse 2DFFT was then applied to the filtered transform. Fig. 5.3 is a plot of the filtered map (interpolated to 1 degree spacing for smooth contouring). Figs. 5.4 and 5.5 are also presented for comparison and represent enlargements of the central portions of Figs. 5.1 and 5.3 respectivley. Much of the irregularity in the original data has been removed although the same basic characteristics remain.

The number of data values averaged in each $2^{\circ} \times 2^{\circ}$ bin is plotted in pixel form in Fig. 5.6. A prominent band of high numbers appears across the map between latitudes 8°S and 16°S. The numbers within this band are approximately double the numbers elsewhere. The source of this relatively high data density band is due to the overlapping linear fits that have been removed from thedata. The linear fits were made in three bands: 50°S to 0°S; 25°S to 25°N; and 50°N to 0°N. Thus, the Australian data contains a region from which two quite different linear trends have been removed. Dominant east-west trends exist in the contour map in northern Australia largely due to this effect. Presumably the same effect occurs in other parts of the world at the same distances from the equator. It is difficult to see how this effect may be removed from the data without repeating the averaging process.

Fig. 5.7 shows a contour plot of the mean altitude of the data with each 2° x 2° bin. This shows prominent stripping parallel to the ascending orbit track direction. The variation in the mean altitude is of the order of 40-50 kms and is presumably due to the inclusion of a few late-mission orbits which were relatively low in altitude. The two holes in the transform plot just to the east and west of the center appear to correspond to the wavelength of the stripping in the mean altitude plot.

The transform of the data when viewed in frequency space is elongated in the north-south direction. The ellipticity varies between 2:1 and 3:1 depending on the contour level chosen. This ellipticity in the transform is a measure of the directional bias in the data set which contains considerably more information in the north-south direction than in the east-west direction. This data bias is a result of many different effects included the orientation of the satellite tracks, the spherical harmonic analysis procedure and the removal of linear fits. At this stage, it is assumed that the anomaly field due to crustal sources does not contain this directional bias.

It is necessary to look more closely at the effect of the processing procedures in terms of the above effects.

Fig. 5.1

Magsat 2^o average data set for the Australian region (90^oE to 180^oE, 50^oS to 0^oN).

Fig. 5.3

Filtered map of 2^o average data set interpolated to 1^o.

Fig. 5.2

Two-dimensional fourier transform of 2^o average data set showing contours of log(power).

Fig. 5.4

Expanded section of Fig. 5.1 for Australia (110^oE to 160^oE, 40^oS to 10^oS).

Fig.5.5

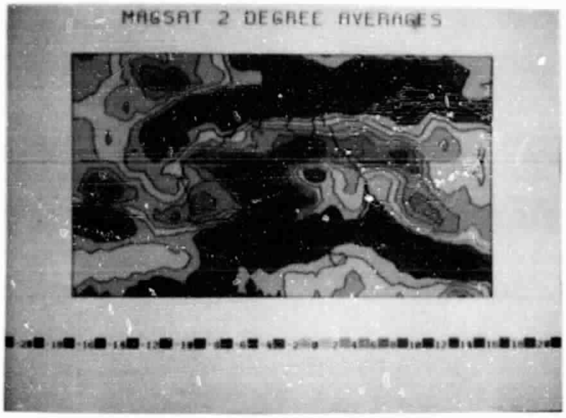
Expanded section of Fig. 5.3 for Australia.

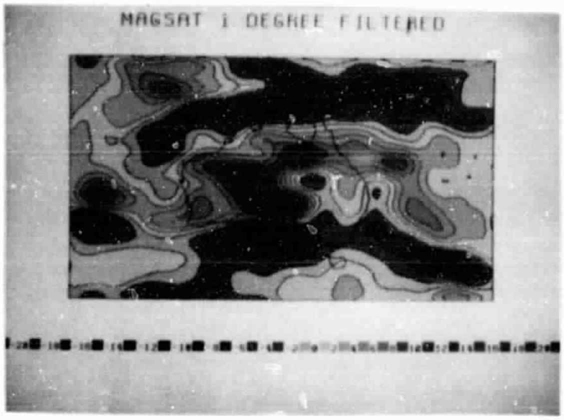
Fig. 5.6

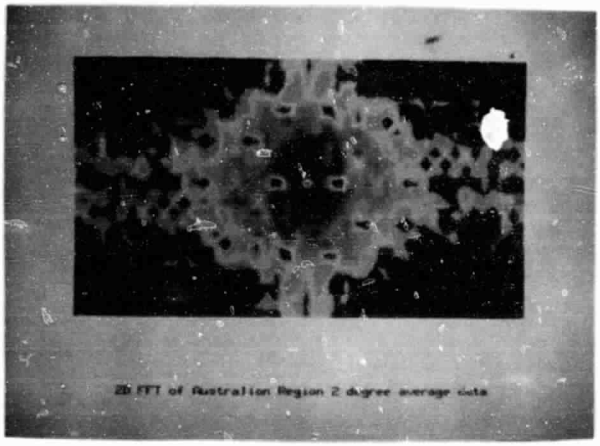
Number of data points per $2^o \times 2^o$ bin in the 2^o average data set.

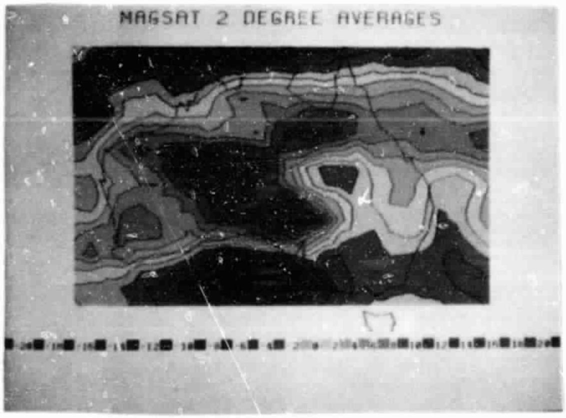
Fig. 5.7

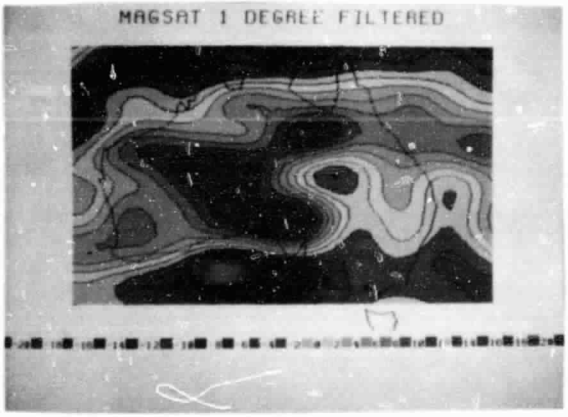
Mean altitude of data points in each $2^o \times 2^o$ bin in the 2^o average data set.

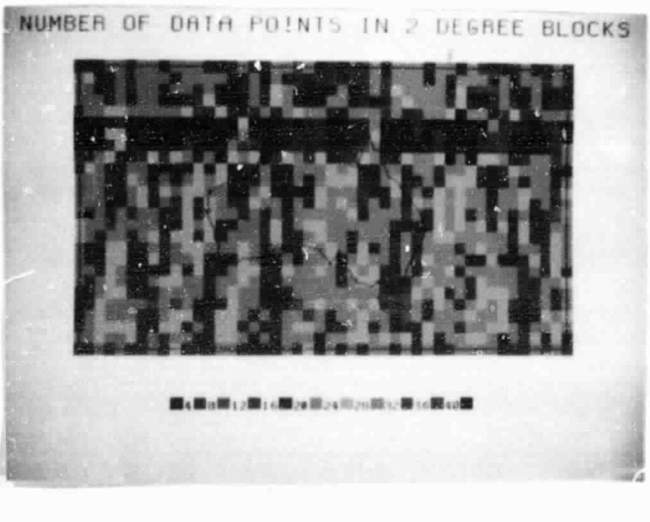

MAGSAT 2 DEGREE AVERAGES

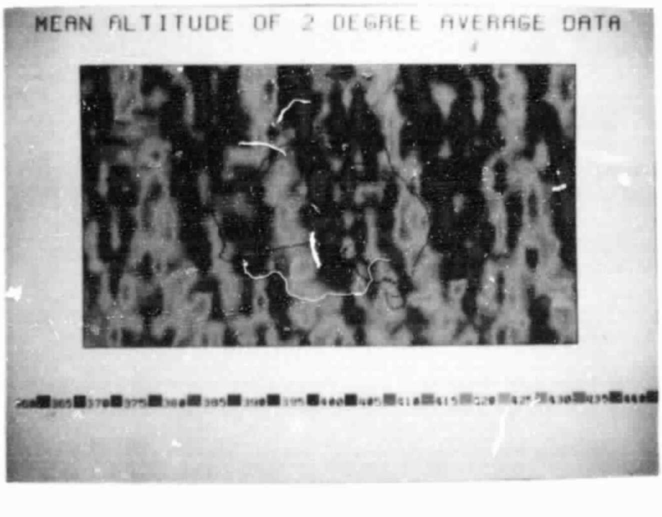

MAGSAT 1 DEGREE FILTERED


2D FFT of Australian Region 2 degree average data


MAGSAT 2 DEGREE AVERAGES


MAGSAT 1 DEGREE FILTERED


NUMBER OF DATA POINTS IN 2 DEGREE BLOCKS


MEAN ALTITUDE OF 2 DEGREE AVERAGE DATA

6.0 CRUSTAL ANOMALY MAP GENERATION

At the timing of writing this report only preliminary crustal anomaly maps have been generated. The first is the result of filtering the 2^{o} average data (as described in section 5). This is presented in contour form overlain on a geological map of Australia (Fig. 6.1). The anomalies can be correlated with known geological boundaries provided that it is remembered that there is a displacement of anomaly peaks as a function of geomagnetic latitude. There are also a number of processing effects present which make detailed interpretation unjustifiable.

The second form of crustal anomaly map that has been formed is the colored swath form shown in Fig. 4.3. The longer wavelength features of this map are identical to those of the 2^{o} average map since it was used as a fitting base. The display shows well the local consistency of the data set but more subtle influences are not easily seen.

We are now awaiting improvement in the field model and Dst (ring current) corrections before producing a final map. The form of this map will probably be an equivalent source calculation in order to remove altitude variation effects and to act as an interpolater.

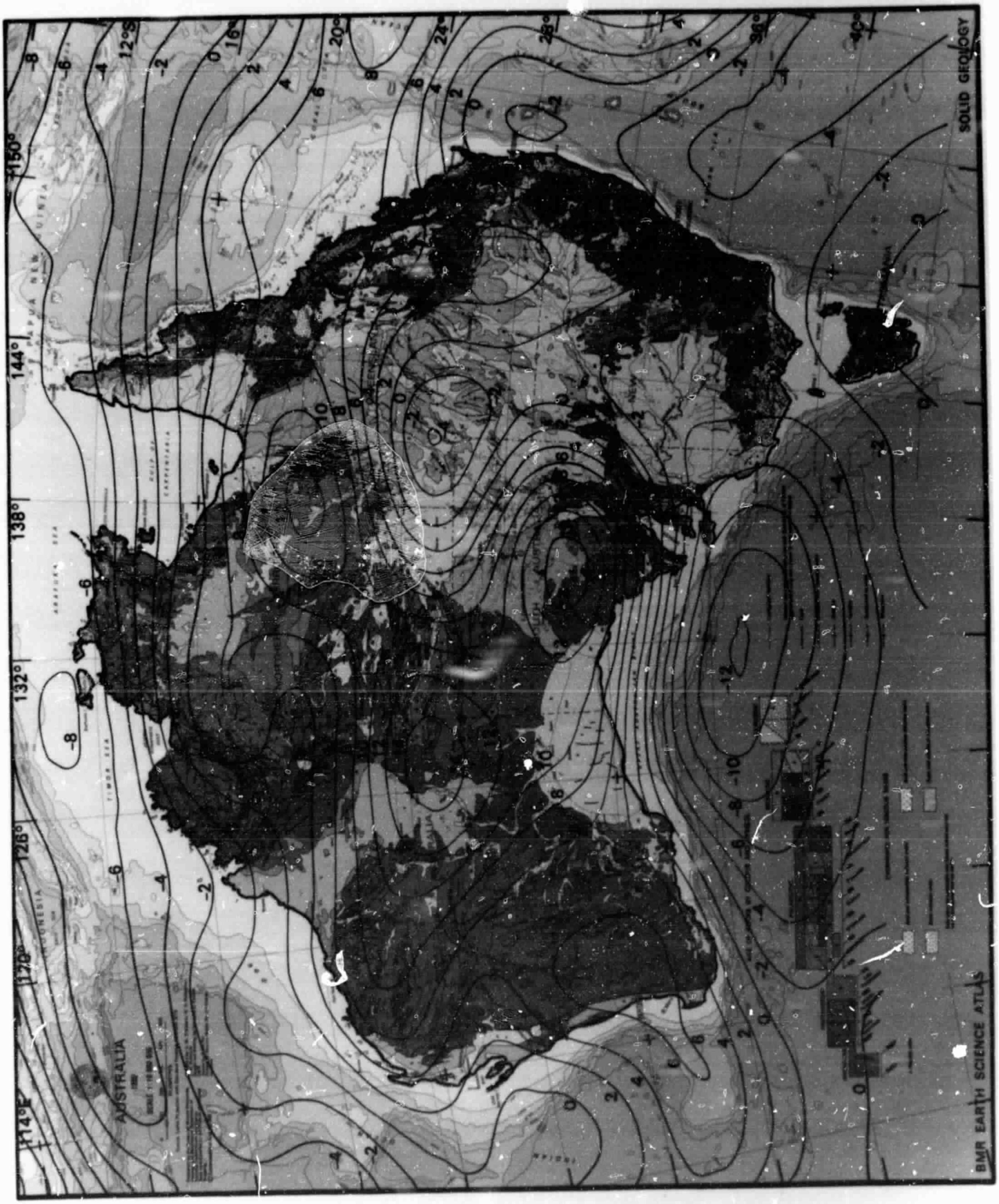
7.0 CONCLUSIONS

The demands for an interactive data selection environment have been fully met by the systems developed at Macquarie University. The development of the data base and graphics packages are clearly important contributions.

We feel that our approach to the data selection problem has been amply justified in that we have been able to critically examine large quantities of data in an efficient manner.

Fig. 6.1

Contour map of filtered 2^{o} average data overlain on a geological map of Australia.


SOLID GEOLOGY
BMR EARTH SCIENCE ATLAS
AUSTRALIA
114°E
120°
126°
132°
138°
144°
150°
12°S
16°
20°
24°
28°
36°
40°

The interpretation of the data has just begun. Qualitatively the Australian region shows extremely good correspondence between geology and the magnetic anomaly field. As has been seen many times the Precambrian shield regions are dominated by large positive magnetic anomalies. There is a curious contrast between the Yilgarn Shield in south-west Australia and the Gawler Block in southern Australia. The boundary between the two anomalies, which are of different amplitude, lies parallel to the structural trend of the region and appears to be of fundamental importance.

Most of the Australian continent is marked by positive anomalies except for the Paleozoic fold belt region in the east which shows up as an area of subdued anomalies. The large positive anomaly over the Gawler block in south Australia is matched by a similar anomaly in Antarctica. In addition the Trans-Antarctic mountains show a similar low magnetic relief to that in south-east Australia. Petrologic evidence and seismic investigations indicate that south-east Australia has a relatively shallow depth to the magnetic Curie point. The boundary between the Precambrian and Paleozoic fold belts is well correlated with change from large positive magnetic anomalies over the shield regions and the negative (and more subdued) anomalies over the Paleozoic rocks. There has been controversy concerning the continuation of Precambrian rocks found in Queensland (the Georgetown Inlies) and the Mt. Isa Block just south of the Gulf of Carpentaria. The magnetic anomaly map indicates that there is a continuous Precambrian shield joining these areas.

Modelling of the various anomalies has been initiated by a study of the Broken Ridge anomaly. The reasons for choosing this anomaly is that the relationship between anomaly and source is unambiguous and the anomaly is relatively isolated. A preliminary paper describing this interpretation is appended.

The Magsat project has been extremely fruitful in bringing together many scientists with varying backgrounds and specializations. The work that has been completed to date should be regarded as an initial phase of a much longer investigation. Those involved in Magsat have only recently begun to understand the nature of the data, the physical fields being measured, the correct approach to obtaining the crustal anomaly signal and production of geologically valid interpretations.

We believe that we have contributed in these areas and are looking forward to a continuing association with the Magsat data and perhaps that obtained from future satellite investigations.

8.0 ACKNOWLEDGMENTS

We have obtained much encouragement and advice from the Magsat team. We would like to thank all involved and in particular to Dr. R.A. Langel. The project has also benefited from an extensive visit to Goddard by one of us (B.D. Johnson) who would like to take this opportunity to thank NASA and especially the Geophysics Branch for providing facilities and an opportunity for uninterrupted research. Numerous discussions with many people including H. Frey, C.C. Schnetzler, P.J. Wasilewski, and M.A. Mayhew have materially improved the current level of understanding of the data and its interpretation.

The funding for our investigation has been from research grants from Macquarie University and the Australian Research Grants Scheme. Professor K. Vozoff at Macquarie University and Dr. K. McCracken at CSIRO Mineral Physics have been very helpful and encouraging throughout.

A number of programmers have assisted us in our software development at Macquarie University. We are grateful to Lyn Brady, John Gillings and Jim Duncan who have enabled the project to proceed at a rapid rate.

N85
23217

UNCLAS

GADB - A DATABASE FACILITY FOR MODELLING NATURALLY OCCURRING GEOPHYSICAL FIELDS

C.N.G.Dampney

Computing Discipline
School of Mathematics and Physics
Macquarie University
North Ryde
N.S.W. 2113
AUSTRALIA

Abstract

The unit record is the basic building block of most database systems in widespread use. The concept works well for many data processing applications, but its use is cumbersome when applied to field data collected in certain kinds of geophysical surveys. The fields are continua*, but measured at discrete points referenced by their position or time of measurement. Systems of this kind are better modelled by databases built from basic data structures attuned to representing traverses across continua that are not of pre-defined fixed length.

The General Array DataBase is a consequence of this requirement. It is built on arrays (ordered sequences of data) with each array holding data elements of one type. The arrays each occupy their own physical data set, in turn inter-related by a hierarchy to other arrays over the same space/time reference points.

The GADB illustrates the principle that a data facility should reflect the fundamental properties of its data, and support retrieval based on the application's view. The GADB is being tested by its use in project MAGSAT, a NASA sponsored geophysical experiment involving ~ 10**7 measurements of the geomagnetic field at altitudes of about 350Km.

* Continuum. n., a whole, the structure of whose parts is continuous and not atomic.

1. INTRODUCTION

The General Array DataBase was built to support the processing and interpretation of geophysical data. Section 2 gives a more detailed description of the application environment and its requirements.

The essential point is that we have a requirement to process interactively data with rather variable properties. It is not practical to define either:-

Figure 1. DATA CENTRED PROCESSING

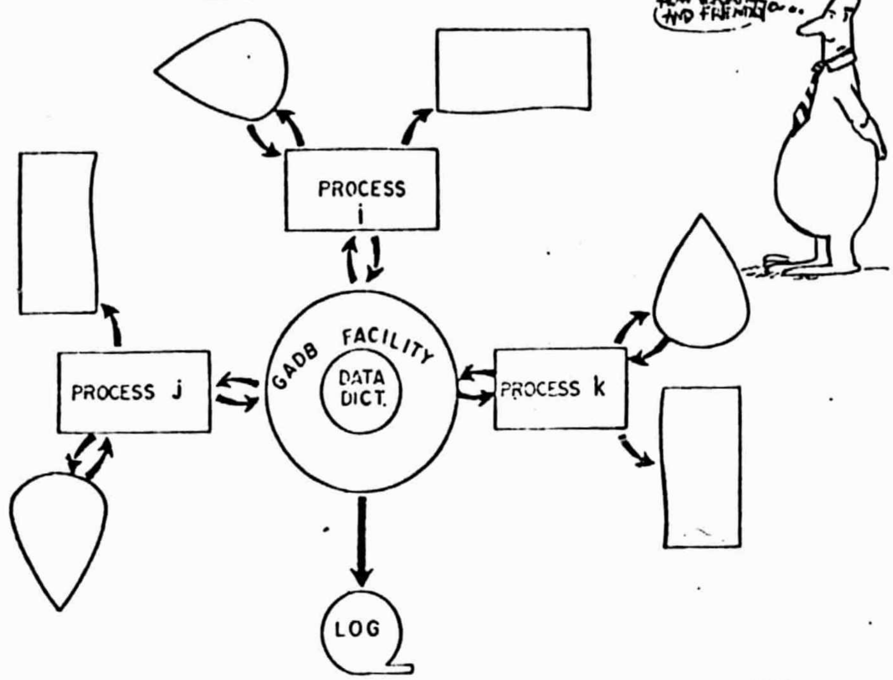
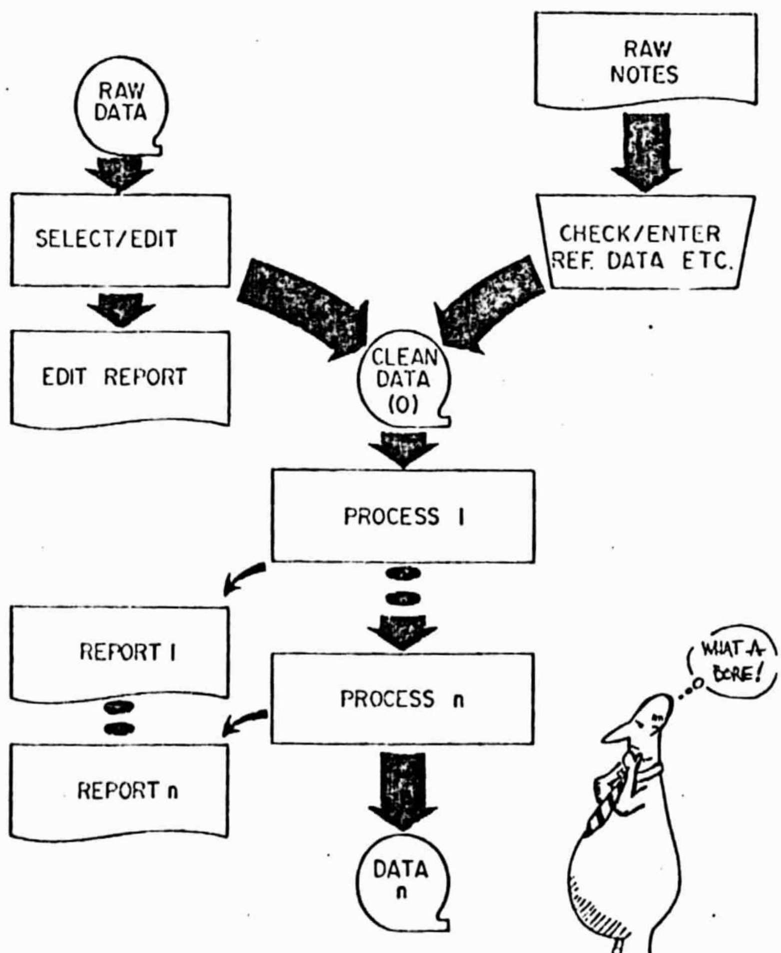


Figure 2.

CONVENTIONAL SEQUENTIAL PROCESSING



(1) all the data types with their properties before the database is created; or

(2) the actual size of the data structures until they are written.

As well the exact sequence of processing cannot be defined. This requirement, of course, dictated in the first place that a database be used (figure 1) rather than the old fashioned batch processing method of a sequence of processes connected by various intermediate files (figure 2).

In section 3 the detailed requirements are analysed and derived. Important amongst these requirements are:-

(1) a built in data dictionary (British Computer Society, 1977);

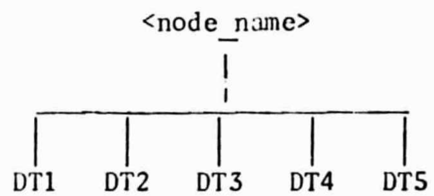
(2) a dynamic rollback facility (Fernandez et al, 1981); and

(3) various integrity checks to give some control over data structure length.

We also found that the data itself fitted readily into a two level hierarchical scheme within a database where data is located by

<database_name>.<node_name>.<data_type_name>

that is schematically:-



where DTn are <data_type_name>'s.

Implementing the database proved an interesting exercise in software architecture. As detailed in section 4 the software layers provide the necessary facilities ranging from those close to the application view down to those concerned with physical storage. The requirement to support data structures that do not have pre-defined fixed length marks a major difference with the unit record concept (Kent, 1980) of the conventional data bases that support business data processing.

2. A PERSPECTIVE OF THE APPLICATION AREA

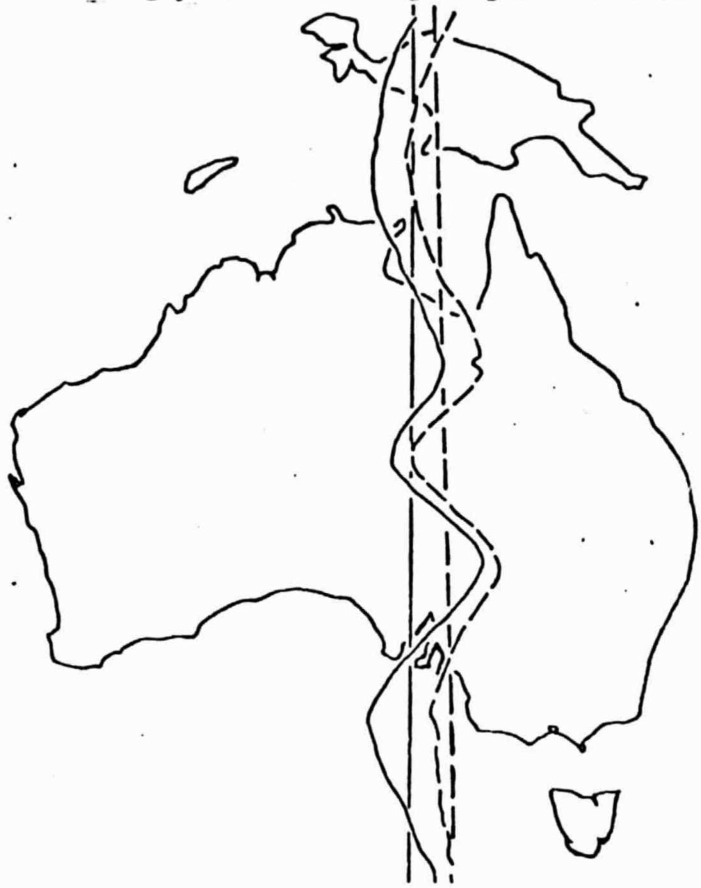
Geophysical exploration is the application of physical methods to discovering geological structure. Generally large quantities of varied data are collected during the measurement or survey phase. This data is reduced to remove unwanted influences and then interpreted in terms of geology.

Because a geological system is open, that is subject to outside influences thot cannot be measured or prevented, there is always an element of uncertainty in interpretation. Two consequences follow:-

(1) There is considerable incentive to collect and compare a variety of different physical measurements, and

(2) Interpretation itself tends to be exploratory as various ad hoc geological hypotheses suggested by the data are tested. These tests are best done in an interactive environment.

Figure 3. Typical project MAGSAT traverses showing path and total geomagnetic field.



The typical geophysical survey we are concerned about here is conducted by measuring geophysical fields* along traverses across a survey area. This enables both systematic coverage by the traverses and more detailed measurement within each traverse.

A somewhat exotic example of this kind of survey is project MAGSAT (American Geophysical Union, 1982), a recent space-borne vector magnetometry survey run by the U.S. National Aerospace Administration. Figure 3 illustrates a typical traverse.

Size is important in database applications. Associated with each traverse were 25 different kinds of measurement along the traverse and quite a number of scalar values associated with the traverse as a whole. 750 traverses were across the Australian region of which 200 were selected for detailed interpretation. Each traverse has about 180 points of measurement. We are therefore dealing with a survey containing ~ 10**6 measured and reduced data elements. However, the database facility developed could easily have handled ~ 10**7 to ~ 10**8 data elements.

One aspect of the data collected is important. The various measurements at each point within a traverse are conventionally considered a single record. However this conflicts with the nature of the measurements. The fields measured are continua. Individual points are simply artifacts of the discretising necessary for digital recording. The measurements are better thought of as a set of arrays along the entire traverse. The elements of each array can be matched one for one with a special "fiducial" array containing the reference values of the points of measurement. This reference could be time or spacial location.

* field as used in Physics - "a region of space influenced by some agent: electric field, magnetic field, gravitational field."

The database facilities described in this paper should be suitable for any application that is concerned with analysing time or spacially varying fields.

3. THE GENERAL ARRAY DATABASE FACILITIES

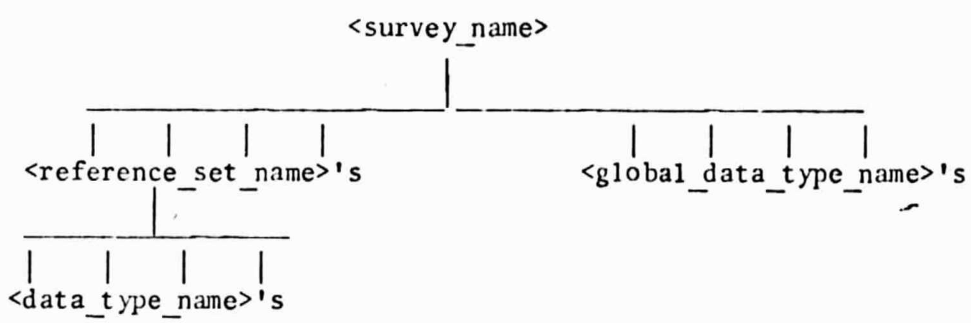
Figure 4 lists the features of the GADB.

Figure 4 - Features of the GADB

* Hierarchical reference scheme to data.
* Basic unit of stored data is an array which is not of pre-defined fixed length.
* Built-in Data Dictionary.
* Any scalar data type is supported.
* All incoming arguments are checked for correctness.
* Dynamic rollback capability from DB or application failure, automatically invoked if necessary.

The basic conceptual model of the database is simply a tree of data entities. The data entities themselves are arrays with the special case that a scalar is an array of unit length. The basic unit of stored data is a variable length array.

Data is referenced through a two level hierarchy described in application terms as



which is represented as

<survey_name>.<reference_set_name>.<data_type_name>

OR

<survey_name>.<global_data_type_name>

which translated to data base terms is

<database_name>.<node_name>.<data_type_name>

with the possibility that <node_name> may be null.

The data may be made up from scalars of any type, for example, REAL, INTEGER, LOGICAL, COMPLEX. These basic scalars may also be a tuple such as the triple

(GMAGX,GMAGY,GMAGZ)

corresponding to the vector components of the geomagnetic field. This facility to support data of all types is provided by a built-in data dictionary. This holds the data type properties necessary to calculate the storage requirement for each element of a given data type. It follows that this in turn requires the facility to enter the various properties, the data type attributes (figure 5), of the data type into the dictionary. These properties are referenced by

<database_name>.<data_type_name>.<data_type_attribute_name>

corresponding to the reference structure already described. Thus the data dictionary maps nicely into the GADB system itself.

Figure 5. The data type attributes

* PTYPE - The primal scalar type {REAL, INTEGER, etc.}
* PSIZE - Tuple size, generally 1 except for such data as fixed length character strings, vectors and tensors.
* STYPE - Structure type {SCALAR, ARRAY}
* ATYPE - Array structure type, whether fixed length within a given node (FIXED) or variable length within a given node (VARIABLE).

The nature of the data processing environment described in section 2 is inherently interactive. A variety of processes may be applied to the data. One possibility is that a human interpreter using the system will need to abnormally terminate a running process. Alternatively a process may fail. In both these cases a database needs to be kept consistent. An automatic dynamic roll-back facility is provided by the database. We call this *failure control*. When failure control is activated it restores the database to the state it was in before the process started.

Actually failure control is somewhat more powerful. The database provides the facility for a process to COMMIT all data entities written so far. The actions between two consecutive commits within a process form a *success unit*. If a process fails the database rolls back to the most recent COMMIT point.

Various integrity checks are desirable in such an environment. In particular it can be seen from section 2 that the various arrays within a given traverse will all have the same number of elements. When an array is stored or retrieved its length is checked to ensure it is consistent with the length of the "fiducial" array for that traverse. Put in database terms the number of elements in the array are checked against the "array length" attribute associated with the node.

The order in which data is stored (PUT) or retrieved (GET) can also be subject to integrity checks. Within a success unit the database only permits actions allowed by the transition table in figure 6. This enforces some discipline onto the application processes.

Figure 7. Example program accessing the GADB.

```
C
C** Data declarations
C
   PARAMETER MAXLEN=1000
      REAL XLONG(MAXLEN), YLAT(MAXLEN)      ! Long, Lat
      REAL GMAG(3,MAXLEN)                   ! Geomag field components
      REAL COREMAG(3,MAXLEN)                ! Earth's core geomag field
      REAL CUSTMAG(3,MAGLEN)                ! Crust geomag field
C
      CHARACTER*12 SURVNAME                 !  Survey name
      CHARACTER*8  TRAVERSE                 ! Traverse name
   .  .  .  .  .  .  .
C
C** Procedure
C
      OPEN database SURVNAME
C
      FOR each traverse DO
         Determine traverse name and place in TRAVERSE
         GET array TRAVERSE.GMAG
         GET array TRAVERSE.COREMAG
         Calculate CRUSTMAG from GMAG and COREMAG
         PUT array TRAVERSE.CRUSTMAG
         CLEAR database   ! COMMIT updates
      END
C
      CLOSE database
C
   .  .  .  .  .  .  .
      END
```

The example shows the generality and versatility of the system. Further arrays are easily added. The old problem of adding one more item to the record of a conventional approach just disappears. A new array is added as a new dataset instead. All arrays are directly accessible and are read in from file to central processor storage in one operation.

Apart from the actions indicated in the program outline other actions of INQUIRE, DELETE, and UNDELETE are also provided.

4.2 The architecture of the GADB

Figure 8 shows the organisation of the GADB system into 4 layers:-

(1) The GADB application interface;

(2) GADBX - The translation of the application view to the storage view and vice versa. This includes various integrity checks.

(3) The storage level interface;

(4) The storage management level which includes the dynamic rollback facility.

Each of the various database actions:-

OPEN, INQUIRE, GET, PUT, DELETE, UNDELETE, CLEAR, and CLOSE

propogate across the various layers.

Figure 8. The software layers of the GADB.

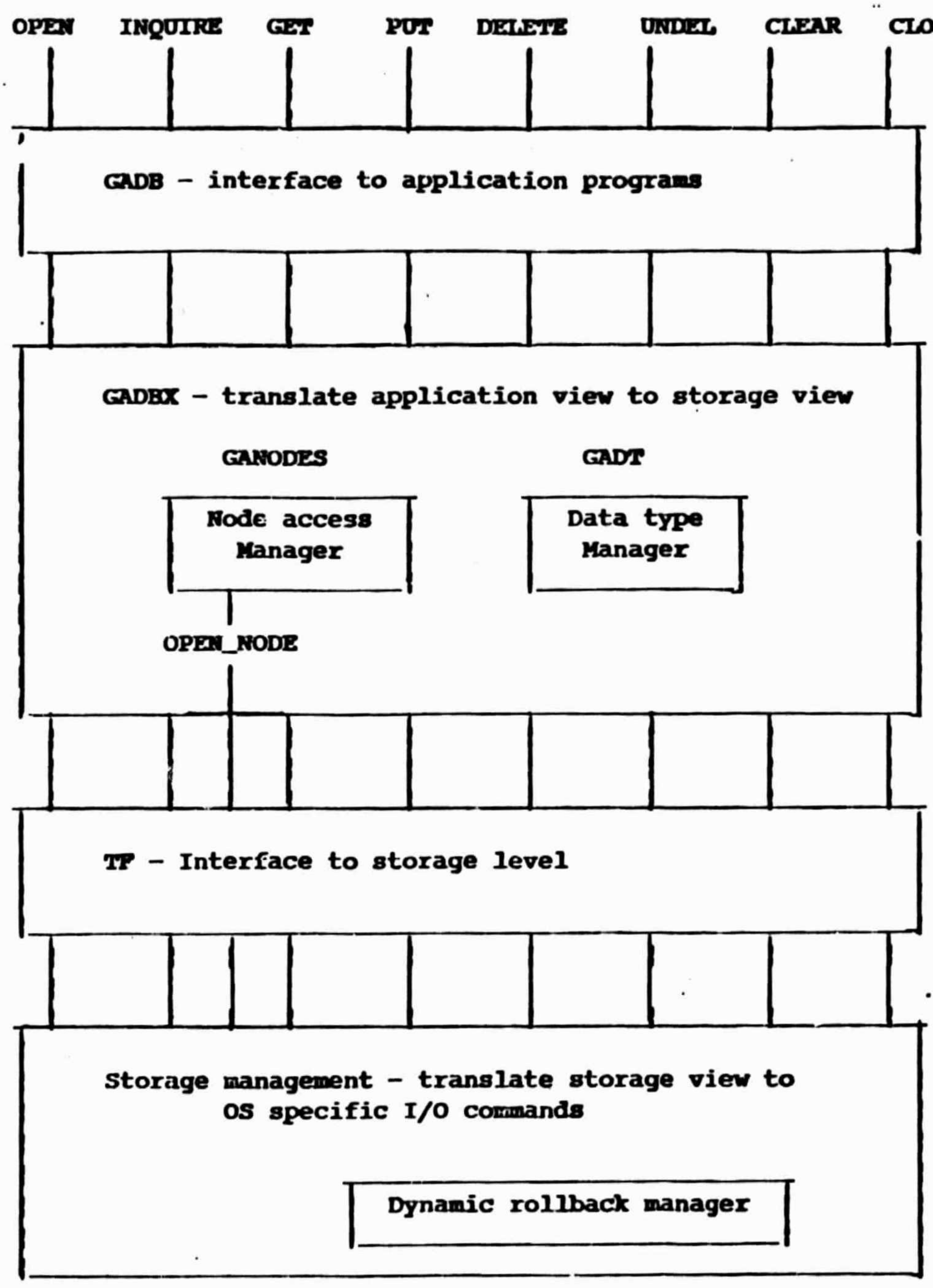


Figure 9. The GADB calling sequences.

The logical form of the calling sequences from the application program are:-

```
INQUIRE ( IN: <node_name>,<data_type_name>
          OUT: <exist>  )

GET( IN: <node_name>,<data_type_name>,<max_length>,<variable>,
     OUT: <actual_length>,<result>  )

PUT( IN: <node_name>,<data_type_name>,<variable>,<actual_length>,
     OUT: <result>  )

DELETE( IN: <node_name>,<data_type_name>,
        OUT: <result>  )

UNDELETE( IN: <node_name>,<data_type_name>,
          OUT: <result>  )
```

where

```
Incoming arguments:
    <node_name>       ! Node name
    <data_type_name>  ! Data type name
    <max_length>      ! Maximum number of elements in array
    <variable>        ! Location in application program into
                      !  which array is be stored or retrieved
    <actual_length>   ! Actual number of elements in array
Outgoing arguments:
    <exist>           ! Whether data entity exists
    <result>          ! Whether action was successful
```

OR

```
OPEN( IN: <database_name>,<access>,
      OUT: <result>  )

CLEAR( OUT: <result> )

CLOSE_COMMIT

CLOSE_ABORT
```

where

```
<database_name>     ! The database name
<access>            ! Whether read or write access
```

Neither CLOSE can return an unsucessful result because in the event of failure during closing the database closes down in an inconsistent state.

4.2.1. Application to Storage Translation. The logical form of the calling sequences are given in figure 9.

Integrity checks are applied to:-

(1) ensure that <node_name> exists,

(2) ensure that <data_type_name> is defined within the system, and

(3) check that <actual_length> is consistent with the <array_length> attribute of the node <node_name> if ATYPE is FIXED (see figure 5).

If the action is a PUT and if the node <node_name> does not already exist, then a new node is created and the <array_length> attribute is entered.

These integrity checks dictated that:-

(1) a list of existing node names and their attributes, and

(2) a list of data type names and their attributes

be held in central processor storage.

Several detailed aspects of integrity are not dealt with here, and are covered instead in Dampney (1983). One very important aspect of integrity is validating data when it is first entered - see Brady and Dampney (1983).

The storage level interface is then called with the calling sequence given in figure 10.

Figure 10. The storage level calling sequence.

```
     The logical form of the calling sequence is:-
     IN:
       <action>            ! INQUIRE, GET, PUT, DELETE, or UNDELETE
       <node_name>
       <data_type_name>
       <structure_type>    ! SCALAR or ARRAY
       <array_type>        ! FIXED or VARIABLE
       <maximum_length_in_bytes>  ! required for all but variable arrays
       <variable>          ! See figure 8
    OUT:
       <actual_length_in_bytes>
       <result>            ! Whether action was successful
                                   OR

     IN:
       NODE_OPEN           ! Required when a new node is created
    OUT:
       <result>
                                   OR

     IN:
       OPEN
       <database_name>
       <access>            ! READ or WRITE access
    OUT:
       <result>
                                   OR
       CLEAR <result>      ! Clear database
       CLOSE_COMMIT
       CLOSE_ABORT
```

4.3 The storage management level

An earlier implementation of the database storage level was built directly on the hierarchical file system provided by VAX/VMS (Digital Equipment Corporation). A number of Operating systems provide such facilities. Therefore it is easy, although rather space inefficient, to implement this database by storing the contents of each array within its own file <data_type_name>: which belongs to a directory corresponding to <reference_set_name> in turn belonging to a directory <survey_name>.

A more recent implementation maps this conceptual storage scheme onto a small number of multiply indexed files. While much more space efficient we must now have specialised utilities for the database, rather than make use of the generalised utilities available for standard system files.

The important point is that the architecture allows the storage management system to be replaced without disturbing the application programs.

Dynamic rollback is implemented by logging datasets changed during a success unit and rolling the changes out if failure occurs. The system uses condition signallers and handlers and exit handlers (Digital Equipment Corporation) to ensure that any error within the GADB itself or the application program automatically causes dynamic rollback. Particular effort was made to implement both CLOSE_COMMIT and CLOSE_ABORT (dynamic rollback) to only release resources. This helps to ensure that they themselves do not fail.

5. CONCLUSION

The GADB system is being used successfully in the MAGSAT project.

It demonstrates the principle that a database facility should reflect the fundamental properties of its data and support retrieval based on the applications view. In particular it supports an interactive environment where the user is able to follow ad hoc his various process options with little hindrance.

6. REFERENCES

American Geophysical Union, 1982. Special issue "MAGSAT preliminary results". Geophysical Research Letters, vol. 9, No.4, pp239-376.

Brady, L.I. and C.N.G. Dampney. "The Architecture and Capabilities of a Screen Processor". Technical Report, School of Mathematics and Physics, Macquarie University.

British Computer Society, 1977. Data Dictionary Systems Working Party Report. DATA BASE, vol.9, No.2,pp1-24.

Dampney, C.N.G., 1983. Precedency Control and other Semantic Issues in a Workbench Database. Proceedings of the ACM SIGMOD83.

Digital Equipment Corporation. VAX/VMS Command Language User's Guide.

Digital Equipment Corporation. VAX/VMS System Services Reference Manual.

Digital Equipment Corporation. VAX-11 Common Runtime Procedure Library Reference Manual.

Fernandez, et al, 1981. Database Security and Integrity. Addison Wesley.

Kent, W., 1980. Data and Reality.

N85
23218

UNCLAS

MACQUARIE UNIVERSITY

CENTRE FOR GEOPHYSICAL EXPLORATION RESEARCH

NORTH RYDE NEW SOUTH WALES 2113 AUSTRALIA

TELEPHONE: 888 8000 EXTN: 9220

TELEGRAPHS & TELEX: MACQUNI AA22377

Director: Professor K. Vozoff
Assistant Director: Dr A.P. Raiche

IN REPLY PLEASE QUOTE: CD:MM

4th May, 1982.

EXTENDED ABSTRACT OF A PAPER SUBMITTED FOR THE 52ND ANNUAL MEETING OF THE SOCIETY OF EXPLORATION GEOPHYSICISTS, DALLAS, TEXAS, OCTOBER 17-21, 1982.

Data selection techniques in the interpretation of MAGSAT data over Australia

B. David Johnson and C.N.G. Dampney,
Centre for Geophysical Exploration Research,
Macquarie University, North Ryde, N.S.W. 2113, Australia.

Summary

The Magsat data require critical selection in order to produce a self-consistent data set suitable for map construction and subsequent interpretation. The interactive data selection techniques, described in this paper, involve the use of a special-purpose profile-oriented data base and a colour graphics display.

Search criteria are employed to select profiles that lie within the survey area, have a minimum altitude and satisfy various data quality indicators. An initial scan selects profiles which have their perigee (lowest altitude of orbit) within the survey region. Subsequent scans through the data are made to select profiles that contain data below a given altitude.

Each profile selected by these automatic search criteria is displayed for visual validation. Various interactive procedures are available to remove data spikes, trim the ends of the profile and to detrend the data values. This corrected profile data may then be compared with other profiles already in the data base and then finally stored in the data base if selected. The large degree of redundancy in the Magsat data enables the rejection of noisy or bad profiles and the detection of time-varying effects in the data.

The use of colour in the graphics has greatly assisted the presentation and appreciation of the data. The three components of the vector magnetic field may be plotted together with the elevation of the satellite in the same display.

For profile comparison, the anomaly values may be projected from the track of the satellite orbit superimposed on a map of Australia. This display is useful for only a relatively small number of profiles and retains the fidelity of the data along each profile.

For larger numbers of profiles the anomalies are plotted along the satellite track by representing each anomaly value by appropriately coloured pixels. This display tends to resemble the final colour contour map and at the same time gives information regarding the distribution of the profiles. The pixel display is particularly good at highlighting level errors between adjacent and intersecting profiles.

The careful application of these data selection techniques is enabling us to validate every data value and ensure that we use the best possible self-consistent data set to construct the maps of the magnetic field measured at satellite altitudes over Australia.

Introduction

The purpose of this study is to obtain a map of the magnetic field at satellite altitudes, due to crustal sources, as an aid to an investigation of larger scale lithospheric structures in the Australian region.

The Magsat Data

The data, provided by NASA/Goddard Space Flight Center, is taken from the Magsat Investigator "quiet-day" data set for the Australian region, from 90°E to 180°E and from 50°S to the equator. This Australian data set consists of some thousands of near north-south profiles of 3-component magnetic field measurements together with position information consisting of latitude, longitude and altitude for each data sample. The altitude range of the satellite was from 300 to 500 kms above the earth's surface except for the last few orbits which were lower.

The data sample interval of the Investigator data set is approximately 50 kms. Since most of the orbits were no lower than 300 kms this sample rate was deemed sufficient to recover all the crustal anomaly information. The effective data sample interval perpendicular to orbit direction is given by the number of profiles and the longitude range of the study area. Provided that profiles are chosen that are spread evenly over the longitude range, then it should be sufficient to reduce the number of profiles required to about 200.

This apparent redundancy of the data (by an order of magnitude) enables the rejection of profiles which are not long enough, have gaps in the data, have high noise levels or contain time-varying effects. Since the signal-to-noise ratio decreases with increasing altitude, profiles obtained at lower altitudes are preferred to those obtained from higher altitudes.

.../3

Selection techniques

The approach that we have employed is to devise an interactive system for selecting the data that uses a combination of automatic search criteria and user-selected data manipulation techniques. For this purpose we have designed a data base system specifically oriented towards storing and transferring arrays of data. The system is disc based and makes use of hierarchical file directories present in many modern computer systems. Profiles are stored under directories whose names correspond to the profile names. Each position-dependant parameter is stored as an array in a file whose name is related to the name of that parameter. Profiles and selected parameters of profiles may be readily retrieved by their names. Parameters summarising the properties of each profile may be quickly scanned to check the validity of the profile (e.g. to check that the profile lies within the geographic region).

The automatic search criteria are applied to a series of scans through the data starting with those criteria most likely to give rise to a reasonable data set. A search criterion may be either a numerical test, in which a parameter is tested against a range of possible values, or a logical test, in which some property of a parameter is tested. Only those profiles which satisfy a specified set of search criteria are automatically selected for interactive validation. Profiles may currently be selected on the basis of:

1) satisfying various data quality indicators,

2) passing through a specified geographic region,

3) containing data obtained below a given altitude, and

4) the orbit perigee (lowest altitude point of the satellite orbit) lying within the survey area.

The search strategy, that we have employed, starts by accepting all profiles that have their perigee within the Australian region and also satisfy the data quality indicators. The reasons for selecting these profiles first are that they have a high signal-to-noise expectation and that they have a relatively constant altitude. The number of profiles resulting from this initial pass through the data is not great enough or sufficiently well distributed to give all the profiles that are needed. Subsequent scans through the data are then made by seeking profiles that contain data below a given altitude, satisfy the data quality indicators and have not been previously selected. The scan may also be restricted to searching for profiles that pass through a selected region in order to fill in gaps in the data distribution.

In testing whether a new profile is required a comparison may be made with the 2 degree average data set that NASA use to construct their global maps. This comparison is carried out by plotting, on the profile display, a profile interpolated from the 2 degree values. The new profile is also plotted against a number of the profiles already accepted

.../4

into the data base. This comparison between profiles for the same region but taken at different times enables us to identify the time varying features in the profiles. The profiles containing such features are then discarded from the data base.

Data display

We have used extensively a Tektronix 4027 colour graphic terminal in the course of this project. Four basic types of display have been developed.

The first is a display of the individual profile values against distance along the profile. In addition to the magnetic field values (either total or component data) the altitude of the orbit is also plotted. In more recent versions of this display more complete information about the data is provided including a small orbit track map. The use of colour in this display is not absolutely essential but it does help the perception of the data.

The second type of display is a fairly standard plot of the anomaly values projected from the track of the satellite orbit. The projection is made in an east-west direction as the orbits are all near north-south. This form of display is very good for comparing a small number of profiles but becomes too complex when this number gets large.

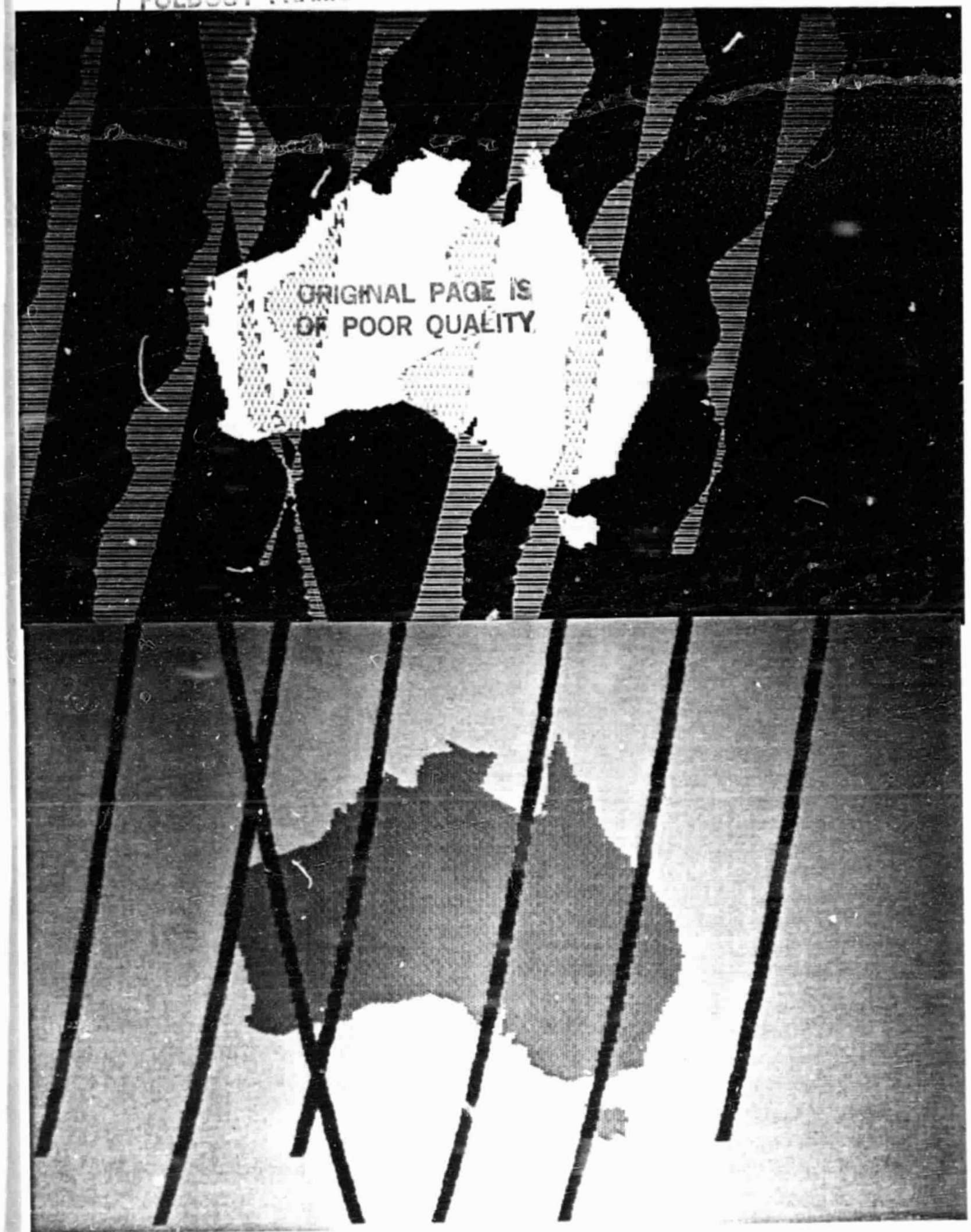
The third type of display complements the second and is again a plot of the anomaly values along the orbit track. This time the anomaly values are represented by coloured rectangles (pixels) of the appropriate colour. Large numbers of profiles may be represented in this display. Information regarding the distribution and comparison between adjacent and intersecting profiles can be readily perceived. It is in this display that the use of colour makes the most dramatic impact in conveying complex forms of information.

The fourth type of display that has been developed is a colour contour display in which the values between contour levels are assigned different colours. There is also an option to bypass the contouring algorithm and provides a very rapid pixel presentation of the same data.

Conclusion

This paper describes techniques which we are currently employing to derive a satellite altitude magnetic field map for the Australian region. The interpretation of this map is currently underway and we propose to comment on some of its more interesting aspects.

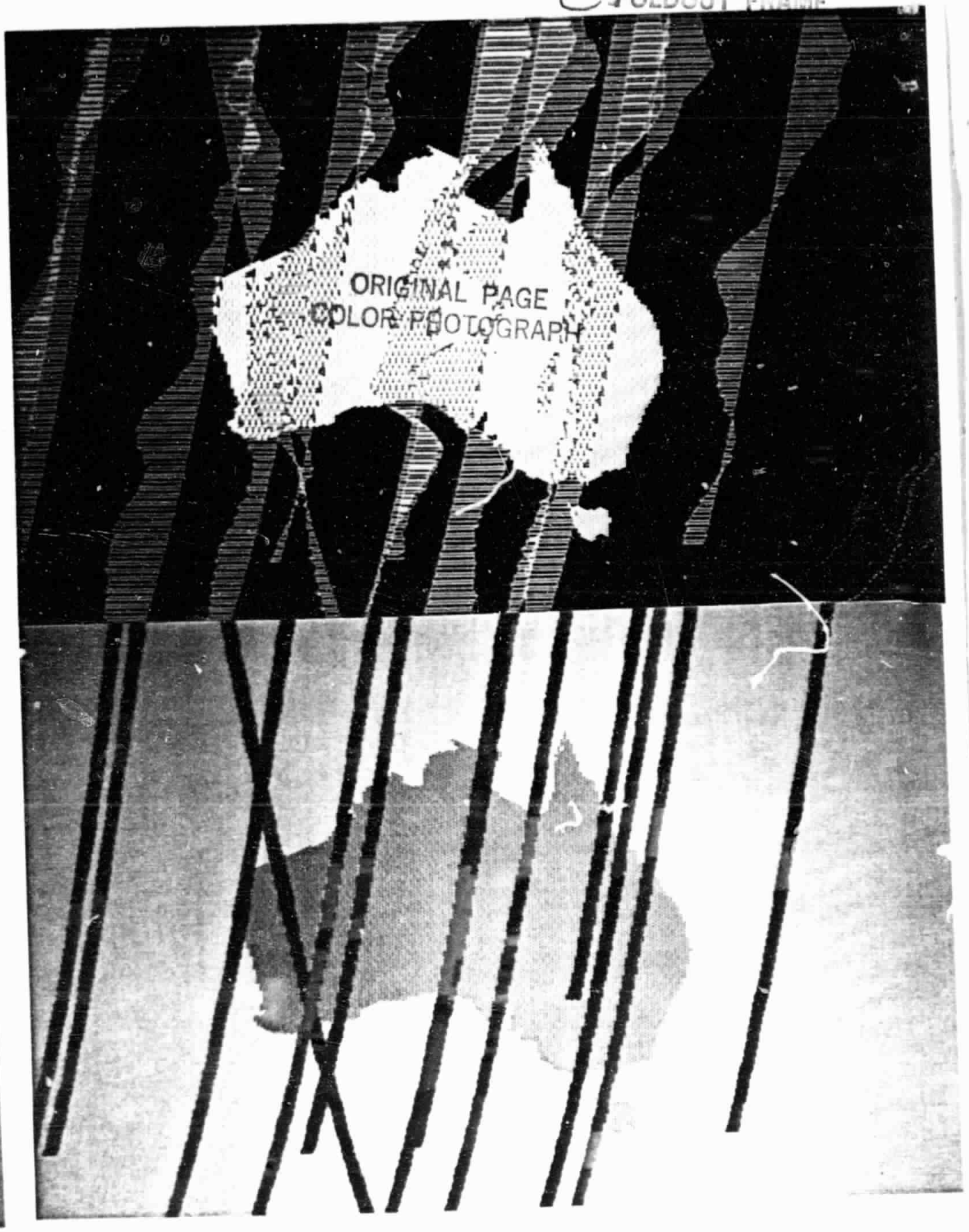
FOLDOUT FRAME



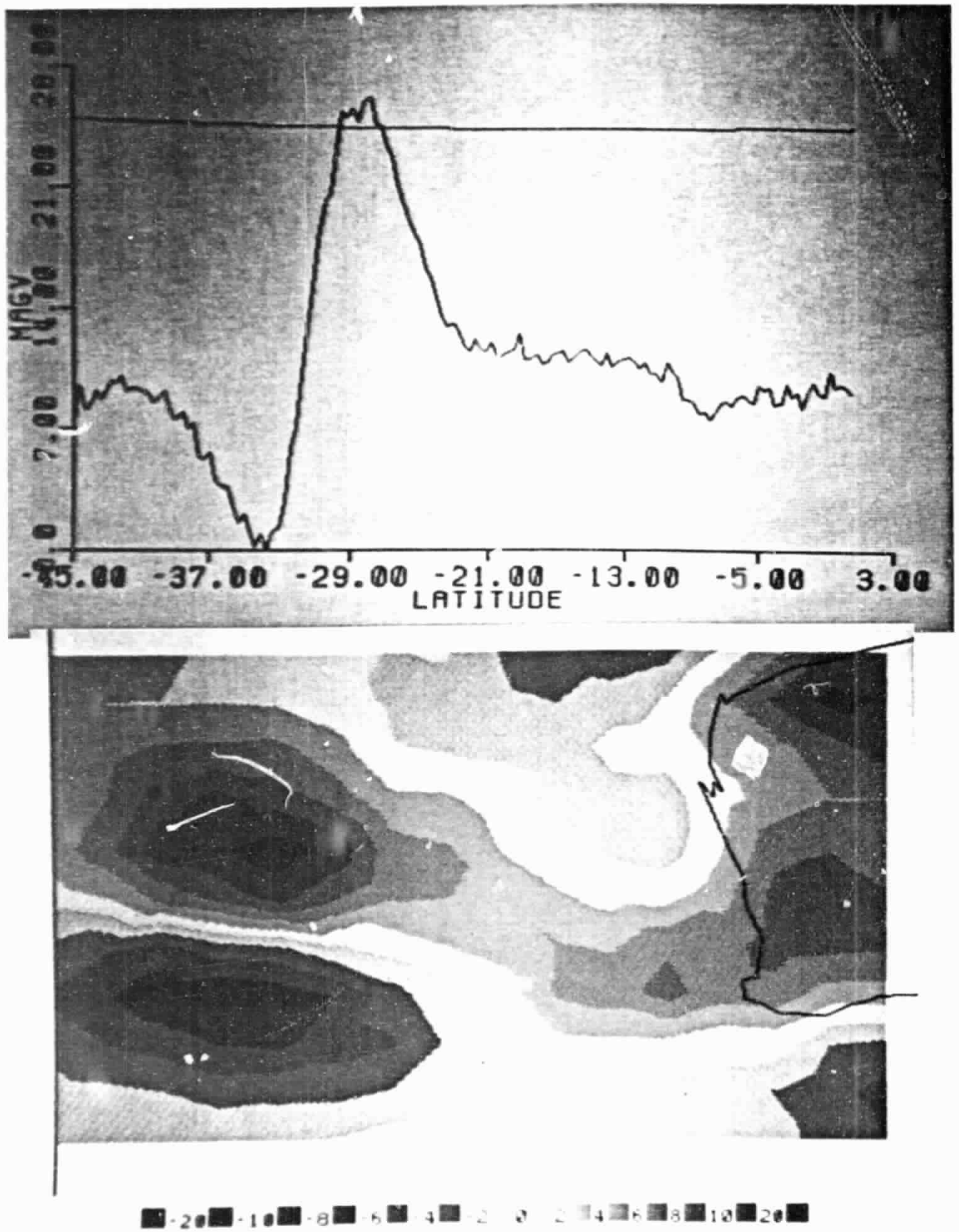
1. Comparison of stacked profile plot and coloured strip plot for 7 profiles.

2. Comparison of stacked plot for 13 profile

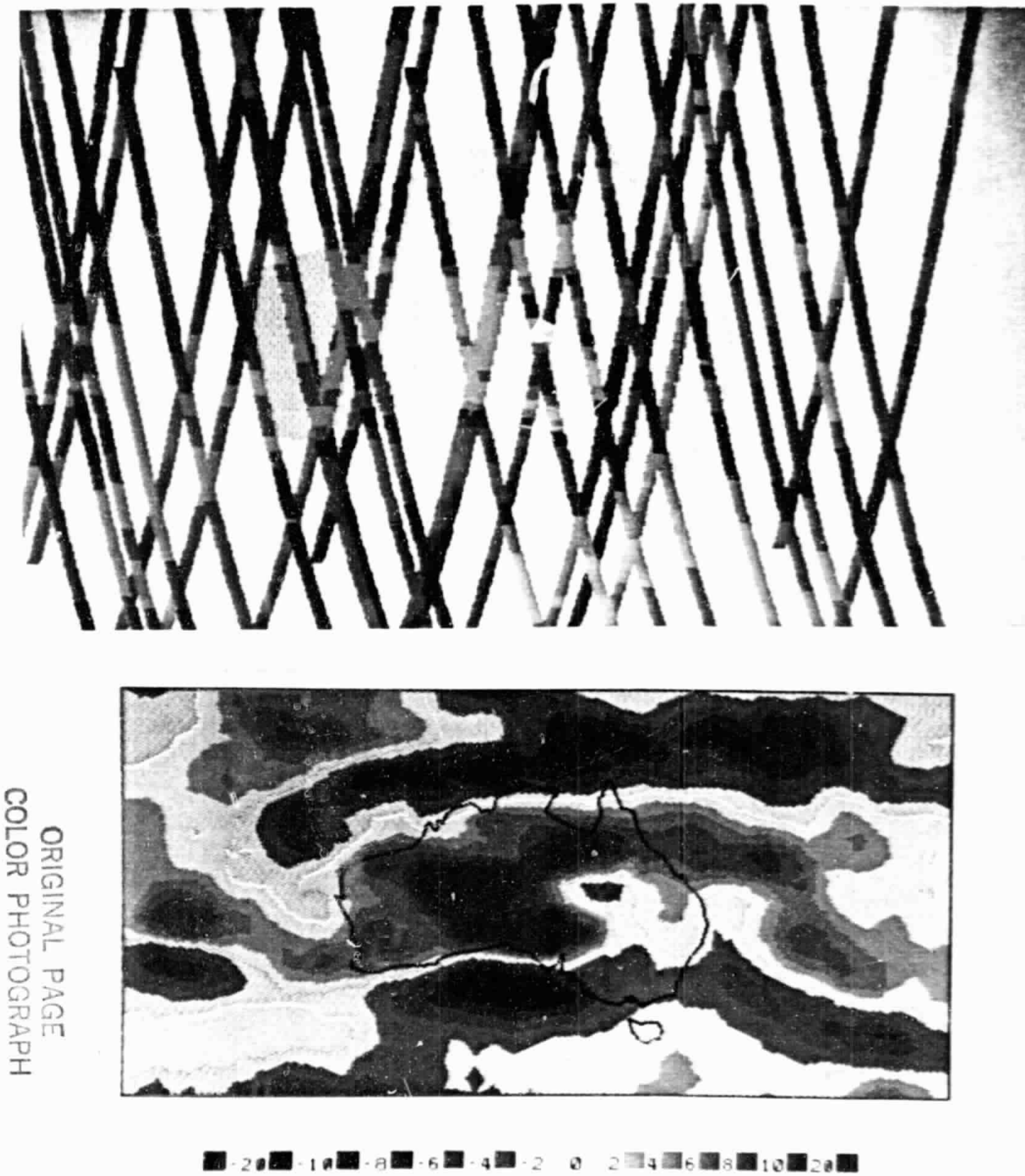
...oured strip



2. Comparison of stacked profile plot and coloured strip plot for 13 profiles.



3. Comparison of profile across Broken Ridge with the NASA 2 degree average map of the area.



4. Comparison of coloured strip plot of 34 profiles with the NASA 2 degree average map of the Australian region.

N85
23219

UNCLA

N85-23219

REMANENT MAGNETIZATION MODEL FOR THE BROKEN RIDGE SATELLITE MAGNETIC ANOMALY

B. David Johnson
NASA/Goddard Space Flight Center, Code 922
Greenbelt MD 20771, USA

(on leave from the Centre of Geophysical Exploration Research, School of Earth Sciences, Macquarie University, North Ryde NSW 2113, Australia)

ABSTRACT

A crustal model for the interpretation of the Broken Ridge satellite magnetic anomaly has been constructed from bathymetric data assuming an Airy-type isostatic compensation. This is in accord with seismic refraction data which gives a maximum depth to the Moho of about 20 kms under the ridge.

An average crustal magnetization of 6 $A.m^{-1}$ is required to account for the observed anomaly amplitudes provided that the whole crust is homogeneously magnetized. In contrast, a model representing only the topographic expression of the Broken Ridge, above the surrounding sea floor, requires a magnetization of the order of 40 $A.m^{-1}$. Since this latter figure is much higher than is to be expected from studies of magnetic properties of oceanic rocks, it is concluded that the majority of the crustal volume of Broken Ridge is magnetized relatively uniformly.

The form of the observed anomaly curve, when compared with model anomalies assuming induced magnetization, shows that the source magnetization has an inclination shallower than that of the present day field which is -65°. There is some uncertainty in the zero-level of the anomaly particularly towards the south where it may be contaminated by auroral anomalies. However, the source location is well determined and hence the locations of the anomaly and of the

steep gradient between its peaks can be used to determine the source inclination.

The source inclination corresponding closest to the observed anomaly curve is close to -50° which indicates a source latitude for Broken Ridge further north than it is at present. Paleomagnetic data for eastern Gondwanaland together with plate tectonic models for the evolution of the eastern Indian Ocean combine to indicate that the Broken Ridge has never been further north than its present location.

It is concluded that the magnetization giving rise to the Broken Ridge satellite anomaly is essentially parallel to the axial dipole field and represents a viscous magnetization which averages out the present day field over time periods long enough to remove secular variation effects.

INTRODUCTION

The author has been involved with a project to interpret the satellite magnetic field over the Australian region. An equivalent source model solution was obtained from the POGO data (Mayhew, Johnson and Langel, 1980) and showed that the satellite anomalies could be related to geological features in Australia. The processing and selection of the Magsat data over the Australian region (Johnson and Dampney, 1982) has progressed to the point where interpretation procedures can be initiated. In order to fully understand the interpretation process, it was decided to start by attempting to model the Broken Ridge satellite anomaly. This anomaly is one of the very few relatively isolated anomalies with an unambiguous source region.

The Broken Ridge is an elevated ridge or plateau standing some 2.5 kms above the normal sea floor. It is elongated east-west (and hence an ideal target for Magsat) having a north-south extent of some 400 kms with an east-west extent a little over 1000 kms. Seismic refraction data indicates that the topographic relief is approximately compensated and that Broken Ridge has a maximum depth to the Moho of around 20 kms (Francis and Raitt, 1967). The southern margin of the ridge is steep and is accompanied by the presence of the Ob Trench, a partially developed trough in the ocean floor, the nature of which is obscure. The relatively high seismic velocities in the lower part of the crust of Broken Ridge have been interpreted as being indicative of an oceanic origin (Carlson, Christensen, and Moore, 1980). Plate tectonic models of the region assume that the Broken Ridge was formed together with the Kerguelen plateau prior to the initiation of spreading between the Australian and Antarctic plates (Sclater and Fisher, 1976; Johnson, Powell and Veevers, 1976; Luyendyck and Rennick, 1977).

THE DATA

The Broken Ridge satellite magnetic anomaly can be observed on the POGO map of Regan, Cain and Davis (1975) and is more clearly defined on the Magsat map (Langel, Phillips and Horner, 1982). The anomaly is relatively isolated and is situated directly over the bathymetric feature of the Broken Ridge. It is a typical dipole type anomaly having a peak-to-peak anomaly of about 20 nT. The positive peak is somewhat larger in amplitude than the negative peak, the positive being to the north of the negative. This pattern is characteristic of southern hemisphere mid-latitude anomalies.

The initial interpretation was carried out with respect to the Magsat map data which has been averaged over $2^{\circ} \times 2^{\circ}$ bins after field model, Dst and linear fit removal. Modelling work was also carried out for a number of selected passes, from the Investigator B data set, by computing the model fields at the satellite observation points.

THE METHOD

A forward modelling technique was used to compute the magnetic anomaly due to a specified model. The modelling technique used was the Gauss-Legendre quadrature program developed at the University of Purdue (von Frese, et al., 1981) and subsequently modified by the author (Johnson, 1983). The three-dimensional model is defined by an upper and lower boundary and is bounded horizontally by a polygonal boundary. A three-dimensional quadrature is carried out first in longitude, then in latitude and finally in the vertical direction. The anomaly is calculated by integrating the weighted sum of the dipoles (or masses for gravity calculations) located at the quadrature nodes. The calculations take into account the spherical geometry of the earth and the varying inclination and declination of the geomagnetic field. The computations can be carried out for a grid of locations in latitude and longitude at a constant elevation or along an individual satellite orbit.

TOPOGRAPHIC MODEL

The topographic expression of the Broken Ridge was modelled by defining the horizontal extent of the ridge and the elevation of the ridge on a 1^{o} grid within that boundary. The upper surface of the model was defined as elevations above the mean sea floor surface which is at about 4.5 kms below the sea surface. The lower surface of the topography model was flat at a depth of 4.5 kms.

Trail-and-error adjustments of the model magnetization were made until the model anomaly peak-to-peak amplitudes matched the observed peak-to-peak amplitudes. The required magnetization was of the order of 40 $A.m^{-1}$.

AIRY MODEL

The model was then modified to include a larger volume of material as the above value of magnetization is too high. A simple Airy-type isostatic model was made assuming a density contrast ratio of 3:1 for the crust against sea water and the crust against mantle. Hence, topographic expressions above 4.5 kms below sea level were compensated for by roots of three times their extent protruding downwards from 10 kms below sea level. This simplistic approach yields maximum depths to the Moho which are in agreement with the seismic refraction results. It should be noted that the volume of the model includes the slab between 4.5 and 10 kms depth within the boundary.

The magnetization was adjusted to fit the observed anomaly magnitude, the required magnetization being of the order of 5 $A.m^{-1}$. This value is well within the range of values that can be expected for oceanic rocks and lies within the range of values obtained from other studies of long wavelength anomalies (Wasilewski and Mayhew, 1982).

INCLINATION TEST

The form of the modelled anomaly curves for the Airy and Topographic models above show a much larger magnitude positive peak than the negative. The geomagnetic field parameters for the Broken Ridge region are an inclination of -65° and a declination of -15°. Thus, the high ratio of the magnitudes of the positive and negative anomaly peaks is caused by the relatively high geomagnetic latitude, due to the proximity of the Broken Ridge to the south geomagnetic pole.

A set of models for inclinations varying from -65° to -5° were computed at 15° intervals. A comparison of this suite of curves with the 2° average data shows that the observed data corresponds to an inclination of about -40°. In this computation some care was taken to simulate the smoothing of the observed data by averaging model results over a 2° x 2° x 100 km box. Hence, the characteristics of the model and observed anomalies can be more easily compared.

COMPARISON WITH PALAEOMAGNETIC DATA

Paleomagnetic data for India, Australia and Antarctica when combined with plate tectonic models for the evolution of eastern Gondwanaland give reliable estimates of the paleolatitude of the Broken Ridge region prior to the separation of Australia and Antarctica. This data indicates that Broken Ridge has never been further north than it is at present and it appears to have been formed at least 20° further south (Schmidt and Embleton, 1981).

MODELLING MAGSAT PROFILES

Since the paleomagnetic data appears to contradict the inferred direction of magnetization it was decided to look more closely at the problem of estimation of inclination of the source magnetization. The 2° average data has a number of problems associated with it due to the use of linear fits to remove the between-track differences (Johnson and Dampney, 1982) and the averaging process itself. The profile of 2° average data used in the earlier comparisons can be seen to have a zero-level error of the order of 1 nT.

The selected Magsat passes (ibid) for the Broken Ridge region were modelled by computing the model anomalies, for the inclination suites, at the locations of the satelllite observations. These can then be directly compared with the observed data along each satellite profile. The satellite data have had no further corrections applied to them other than the removal of the Magsat 4/81 field model (Langel et al., 1981).

The result is that the best fitting inclination is close to -50° for these passes. This is not as shallow as inferred earlier and is indistinguishable from the direction for an axial dipole field. In addition it has been necessary to increase the source magnetization to 6 $A.m^{-1}$ in order to better match the profile data.

CONCLUSIONS

It is concluded that the Broken Ridge satellite magnetic anomaly is due to a magnetization involving the whole (or nearly so) volume of the crust under the Broken Ridge topographic feature and that the magnetization is relatively uniform in direction. Any departure from these situations would increase the inferred magnetization of 6 $A.m^{-1}$ still further.

The direction of the source magnetization is consistent with an inclination shallower than the present geomagnetic field and close to that of an axial dipole. Since a more northerly source location for Broken Ridge is contrary to the paleolatitude data it is thought that the magnetization represents a magnetization obtained by averaging the geomagnetic field direction over a sufficient time to remove secular variation effects. This pattern is indicative of viscous magnetization.

REFERENCES

Johnson, B.D. and C.N.G. Dampney, 1982, Data selection techniques in the interpretation of Magsat data over Australia. (Paper presented at the 52nd Annual Society of Exploration Geophysicists meeting in Dallas, Texas, October 1982).

Schmidt, P.W. and B.J.J. Embleton, 1981, J. Geophys. Res., 86, 3998-4008.

Wasilewski, P.J. and M.A. Mayhew, 1982, Crustal xenolith magnetic properties and long wavelength anomaly source requirements, Geophys. Res. Letts., 9, 329-332.

von Frese, R.R.B., W.J. Hinze, L.W. Braile and A.J. Luca, 1981, Spherical earth gravity and magnetic anomaly modeling by Gauss-Legendre quadrature integration, J. Geophys., 49, 234-242.

Johnson, B.D., 1983, Sphere2 - notes for the use of this revised version of the program SPHERE for modeling spherical earth gravity and magnetic anomalies (in preparation).

Regan, R.D., J.C. Cain and W.M. Davis, 1975, A global magnetic anomaly map, J. Geophys. Res., 80, 794-802.

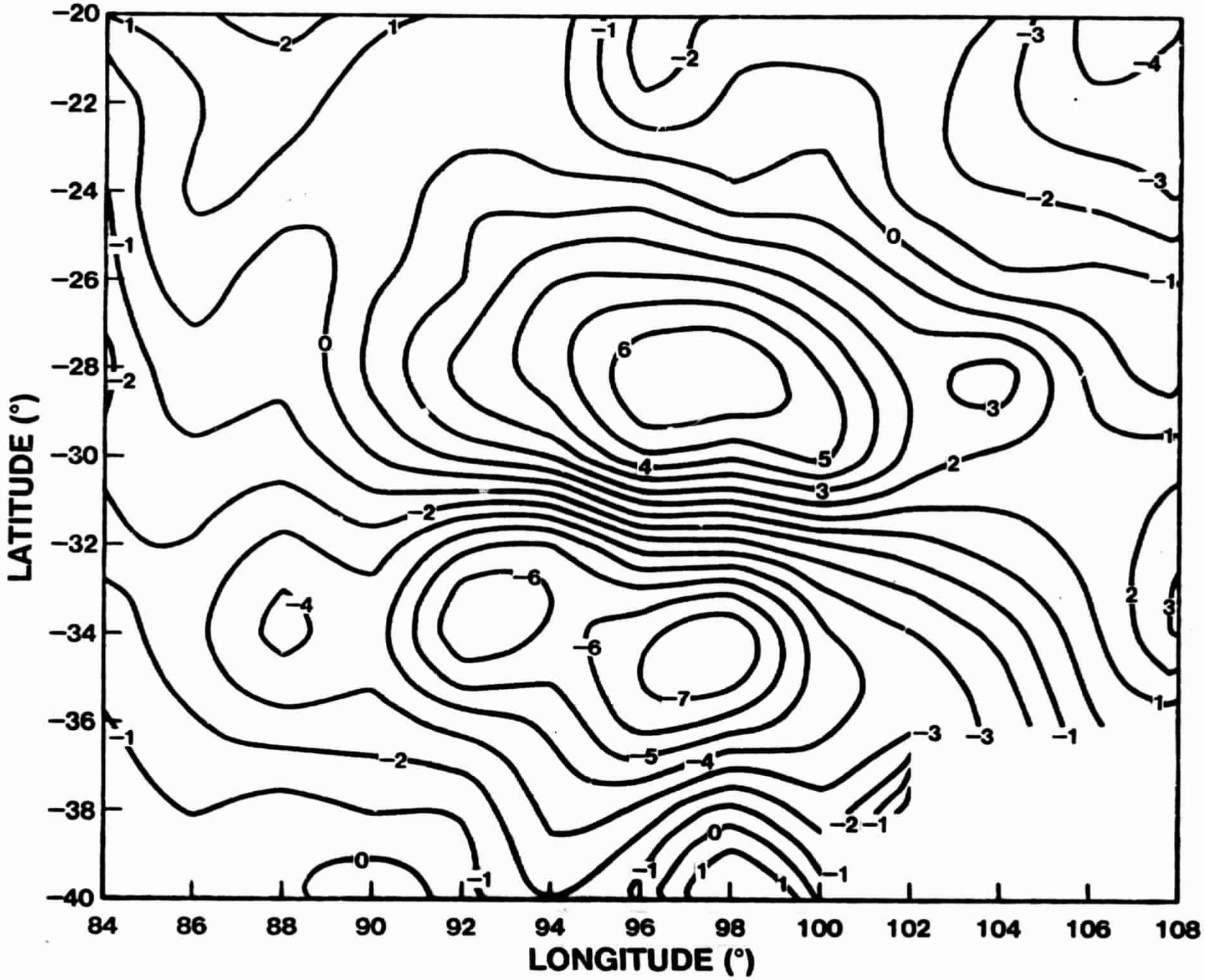
Langel, R.A., J.D. Phillips and R.J. Horner, 1982, Initial scalar magnetic anomaly map from Magsat, Geophys. Res. Letts., 9, 269-272.

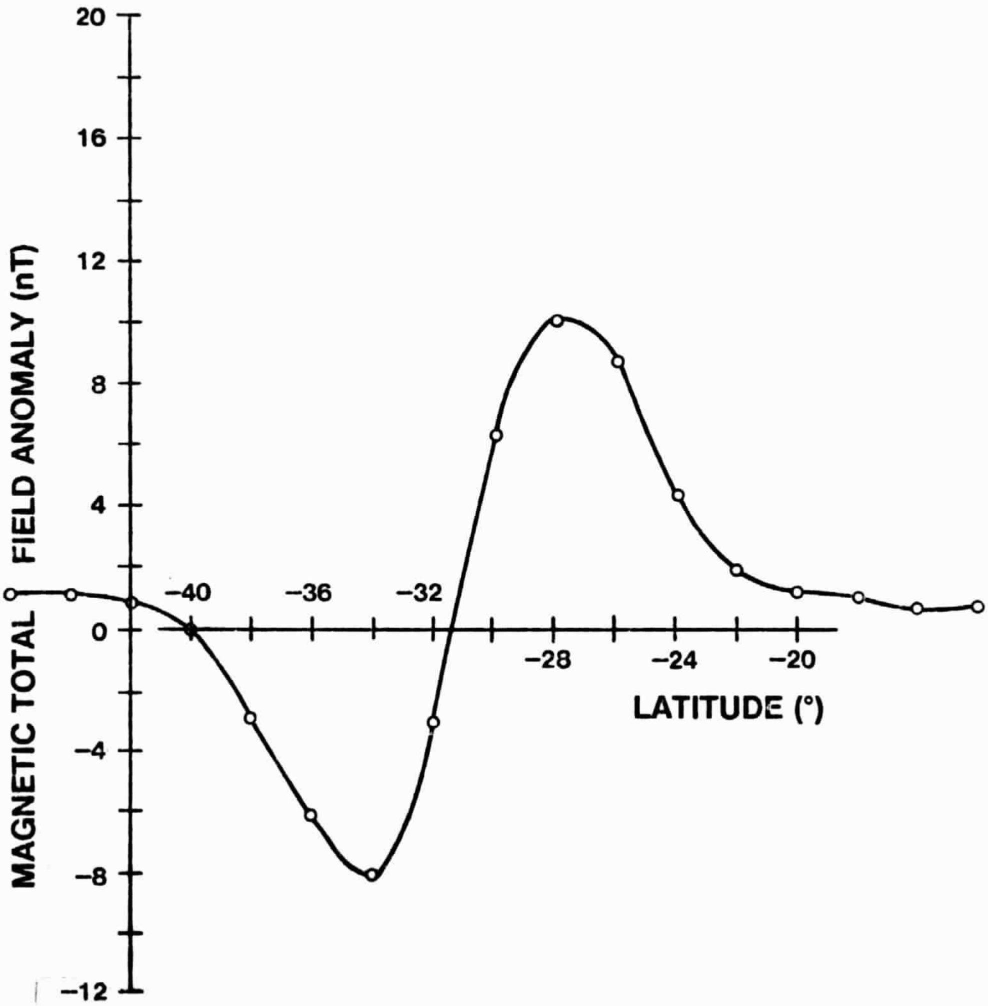
Langel, R.A., J. Berbert, T. Jennings and R. Horner, 1981, Magsat data processing: an interim report for investigators, NASA TM 82160.

Sclater, J.G. and R.L. Fisher, 1976, Evolution of the east central Indian Ocean, with emphasis on the tectonic setting of the Ninety East Ridge, Geol. Soc. Amer. Bull., 85, 683-702.

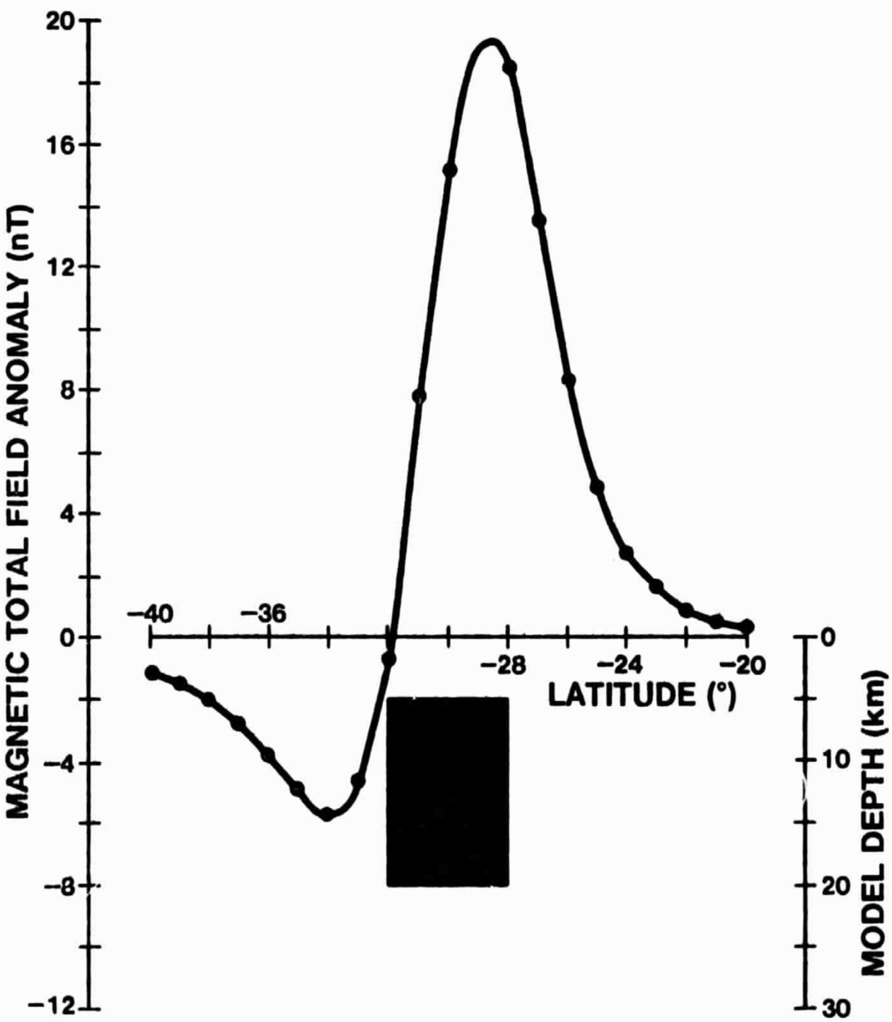
Luyendyck, B.P. and W. Rennick, 1977, Tectonic history of aseismic ridges in the eastern Indian Ocean, Geol. Soc. Amer. Bull., 88, 1347-1356.

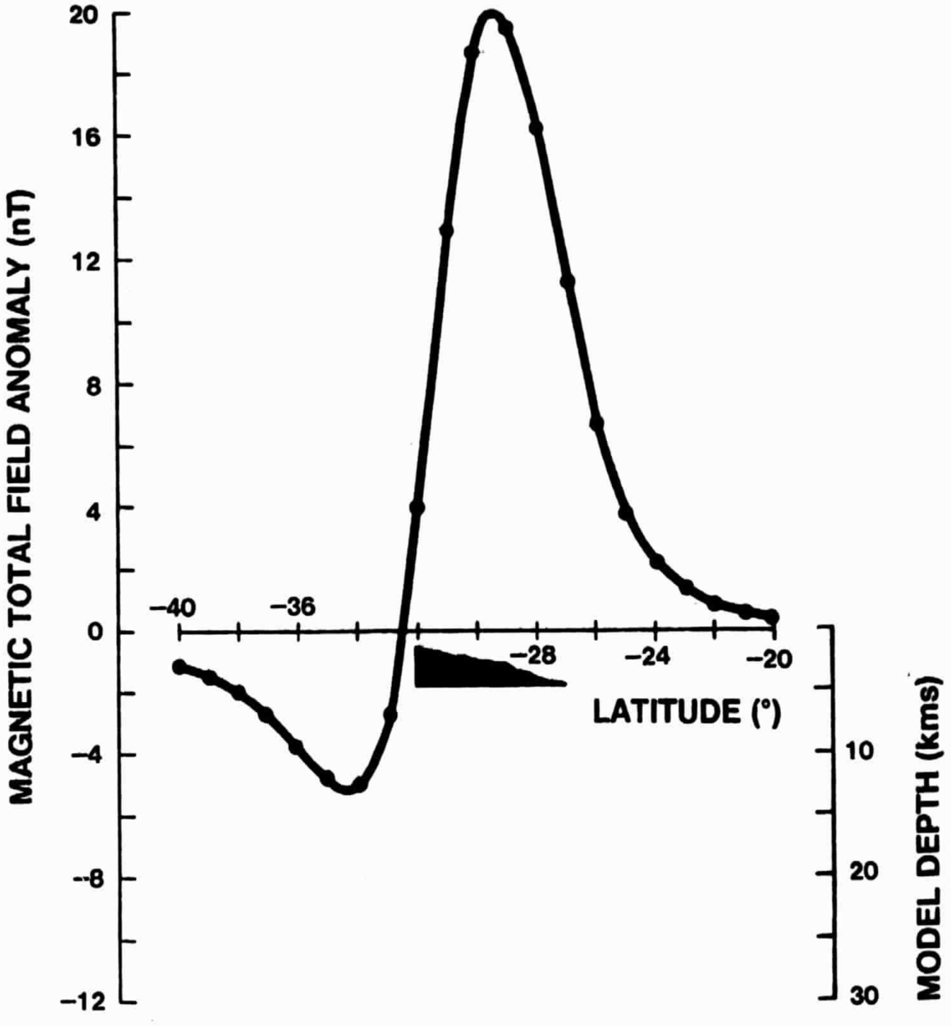
BROKEN RIDGE
MAGSAT 2 DEGREE AVERAGE DATA
TOTAL MAGNETIC FIELD ANOMALY



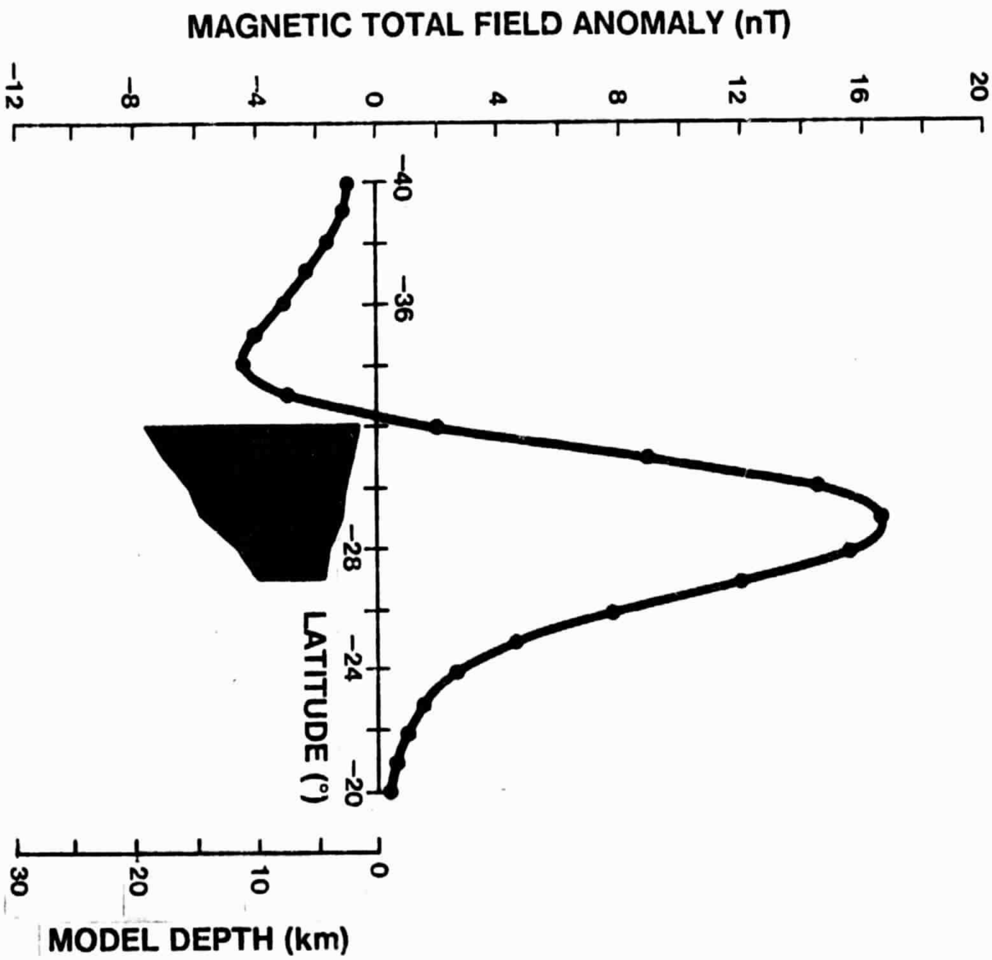


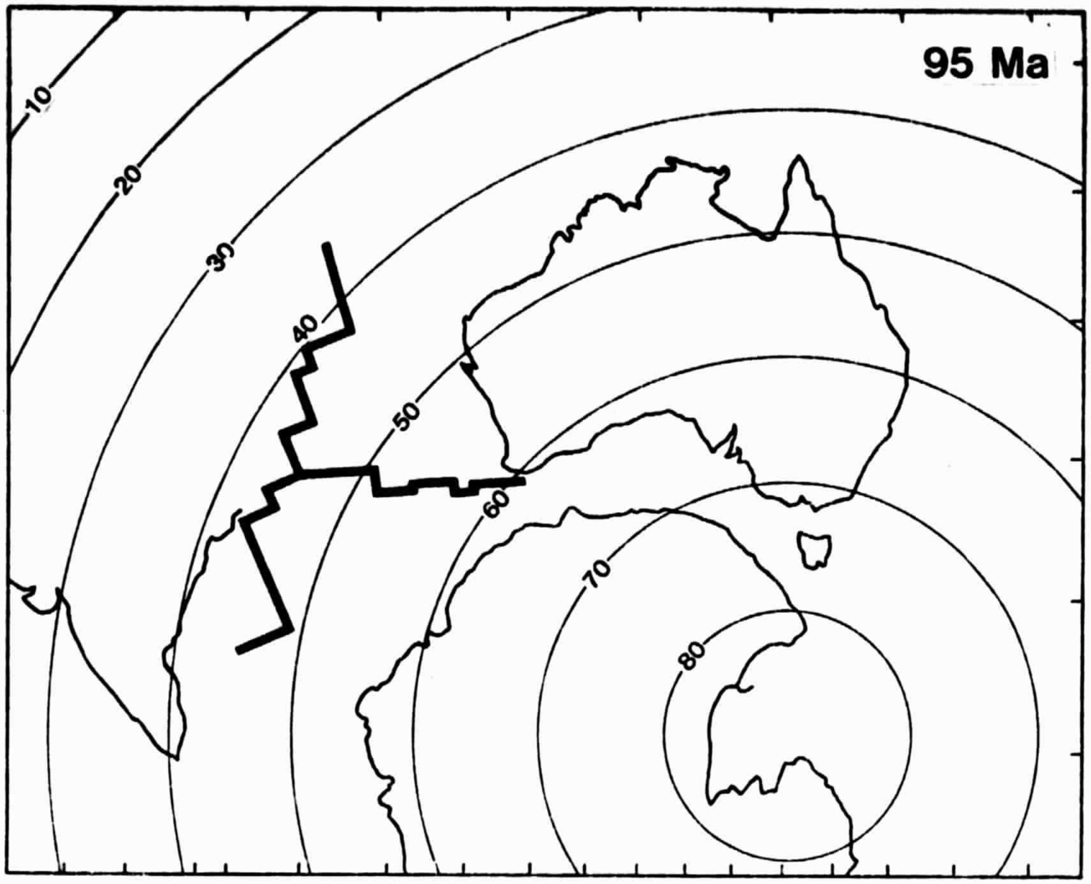
20
16
12
8
4
0
-4
-8
-12
MAGNETIC TOTAL FIELD ANOMALY (nT)
-40
-36
-32
-28
-24
-20
LATITUDE (°)

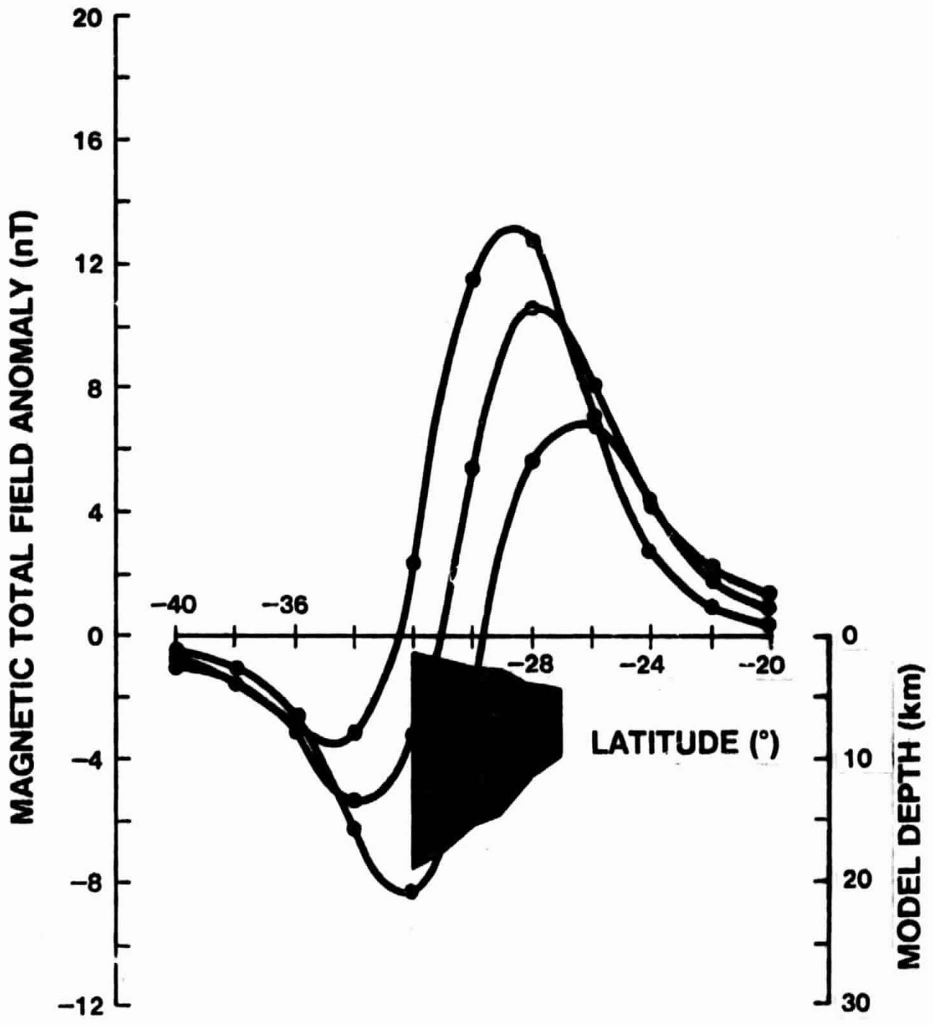

MAGNETIC TOTAL FIELD ANOMALY (nT)
20
16
12
8
4
0
−4
−8
−12
−40
−36
−28
−24
−20
LATITUDE (°)
MODEL DEPTH (km)
0
10
20
30

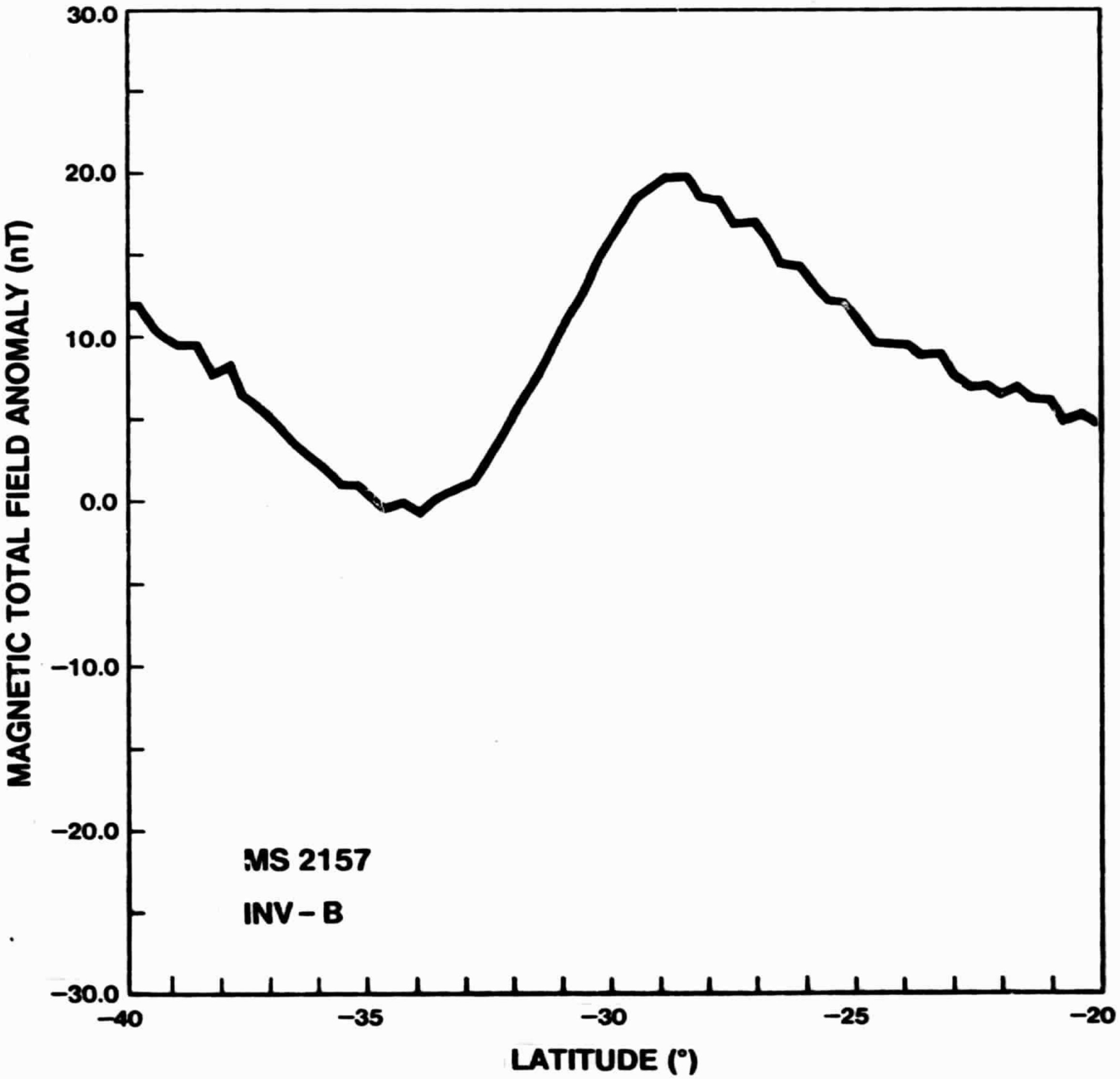


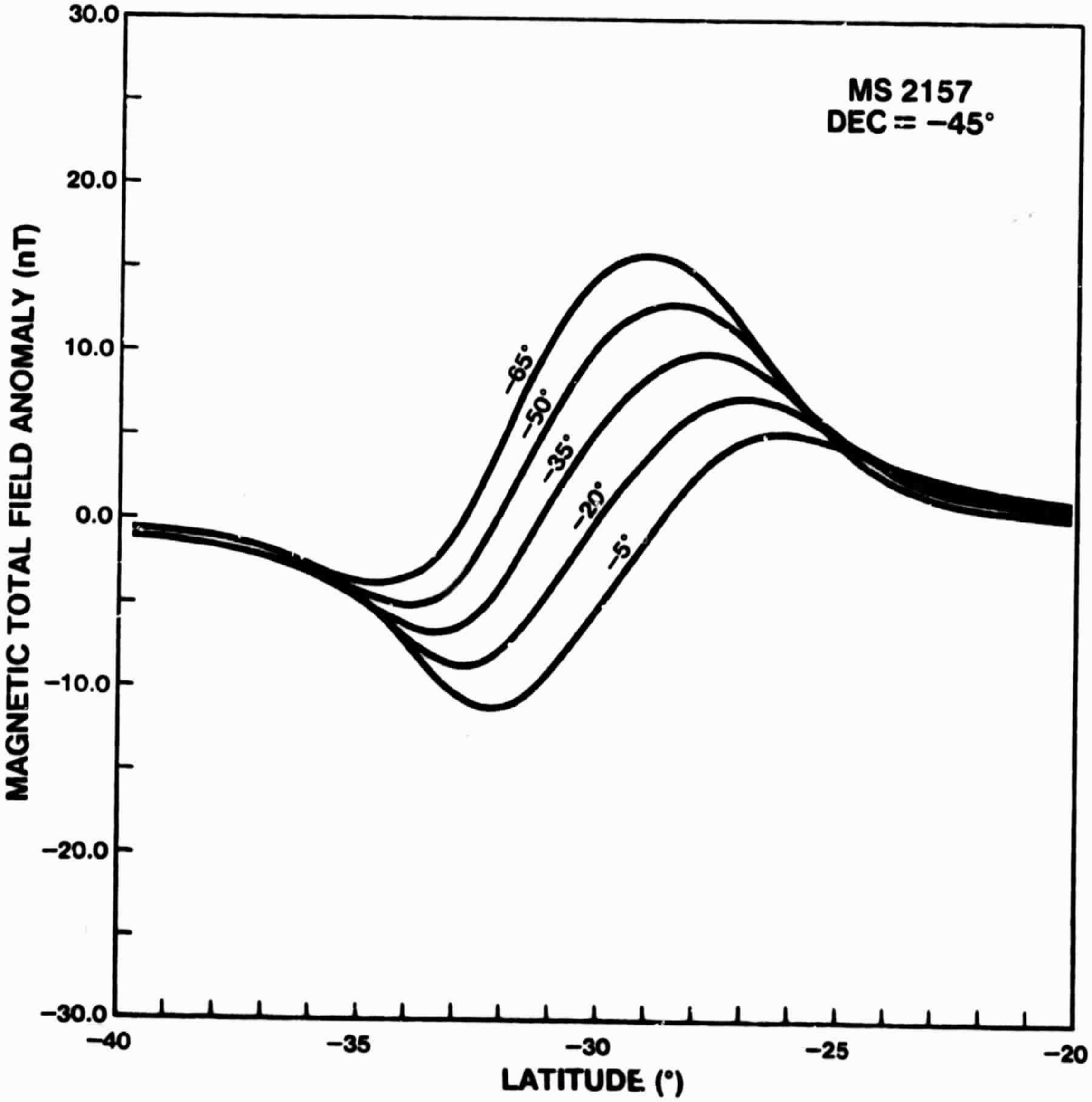
MAGNETIC TOTAL FIELD ANOMALY (nT)
20
16
12
8
4
0
-4
-8
-12
-40
-36
-28
-24
-20
LATITUDE (°)
10
20
30
MODEL DEPTH (kms)


MAGNETIC TOTAL FIELD ANOMALY (nT)
-12
-8
-4
0
4
8
12
16
20
-40
-36
-28
-24
-20
LATITUDE (°)
0
10
20
30
MODEL DEPTH (km)

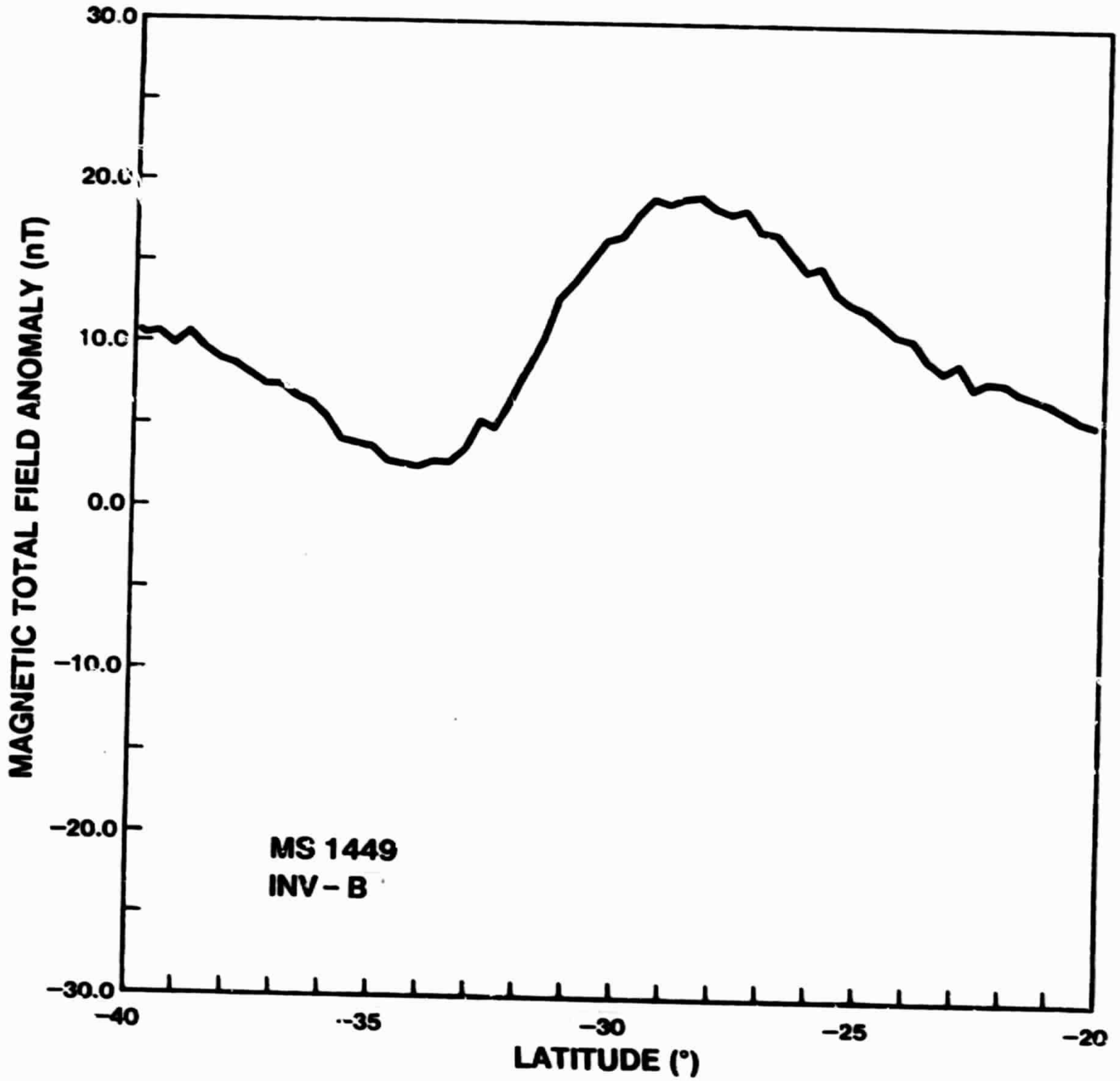

95 Ma
10
20
30
40
50
60
70
80

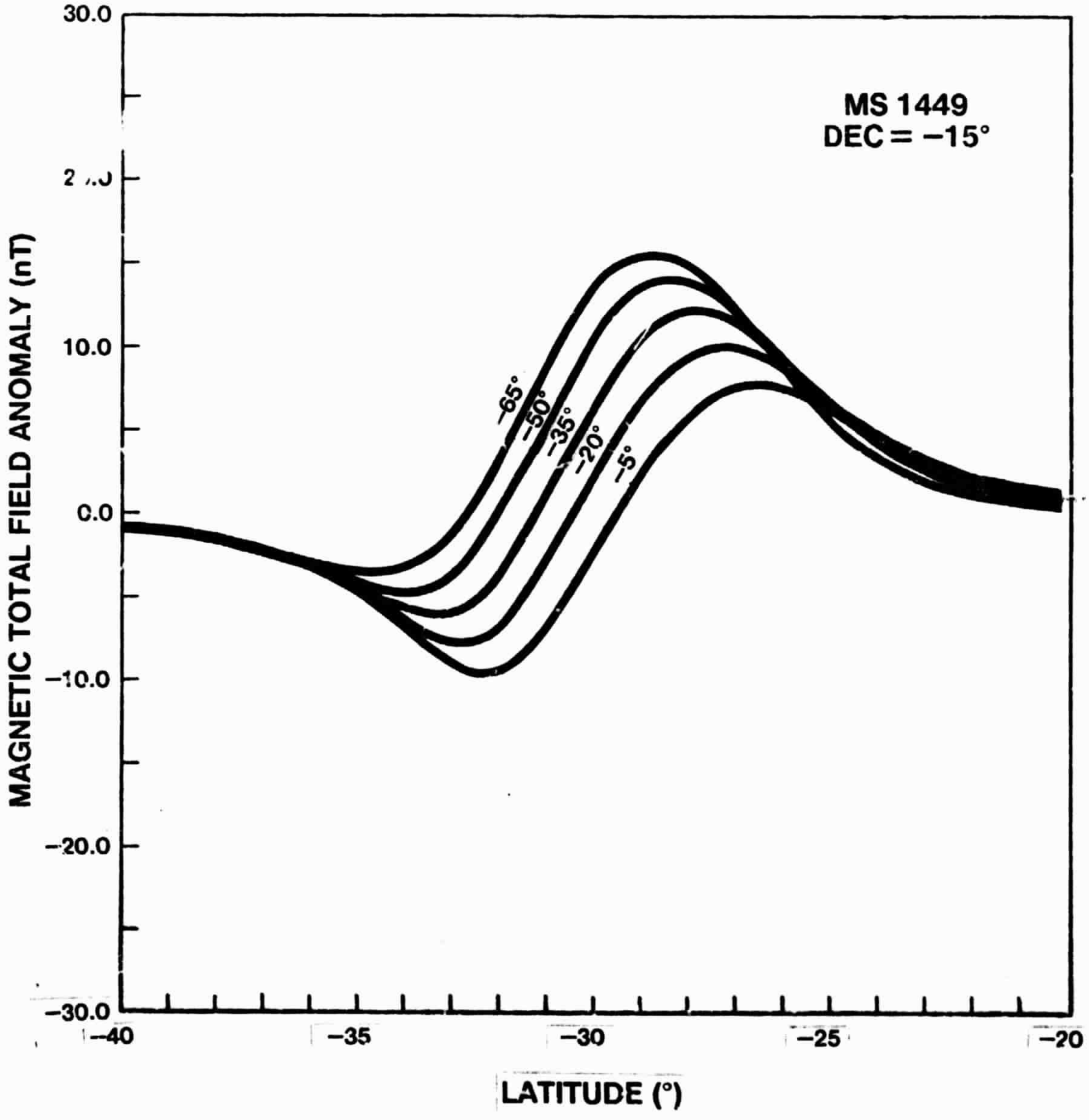

MAGNETIC TOTAL FIELD ANOMALY (nT)
20
16
12
8
4
0
−4
−8
−12
−40
−36
−28
−24
−20
LATITUDE (°)
MODEL DEPTH (km)
0
10
20
30


30.0
20.0
10.0
0.0
−10.0
−20.0
−30.0
MAGNETIC TOTAL FIELD ANOMALY (nT)
MS 2157
INV – B
−40
−35
−30
−25
−20
LATITUDE (°)

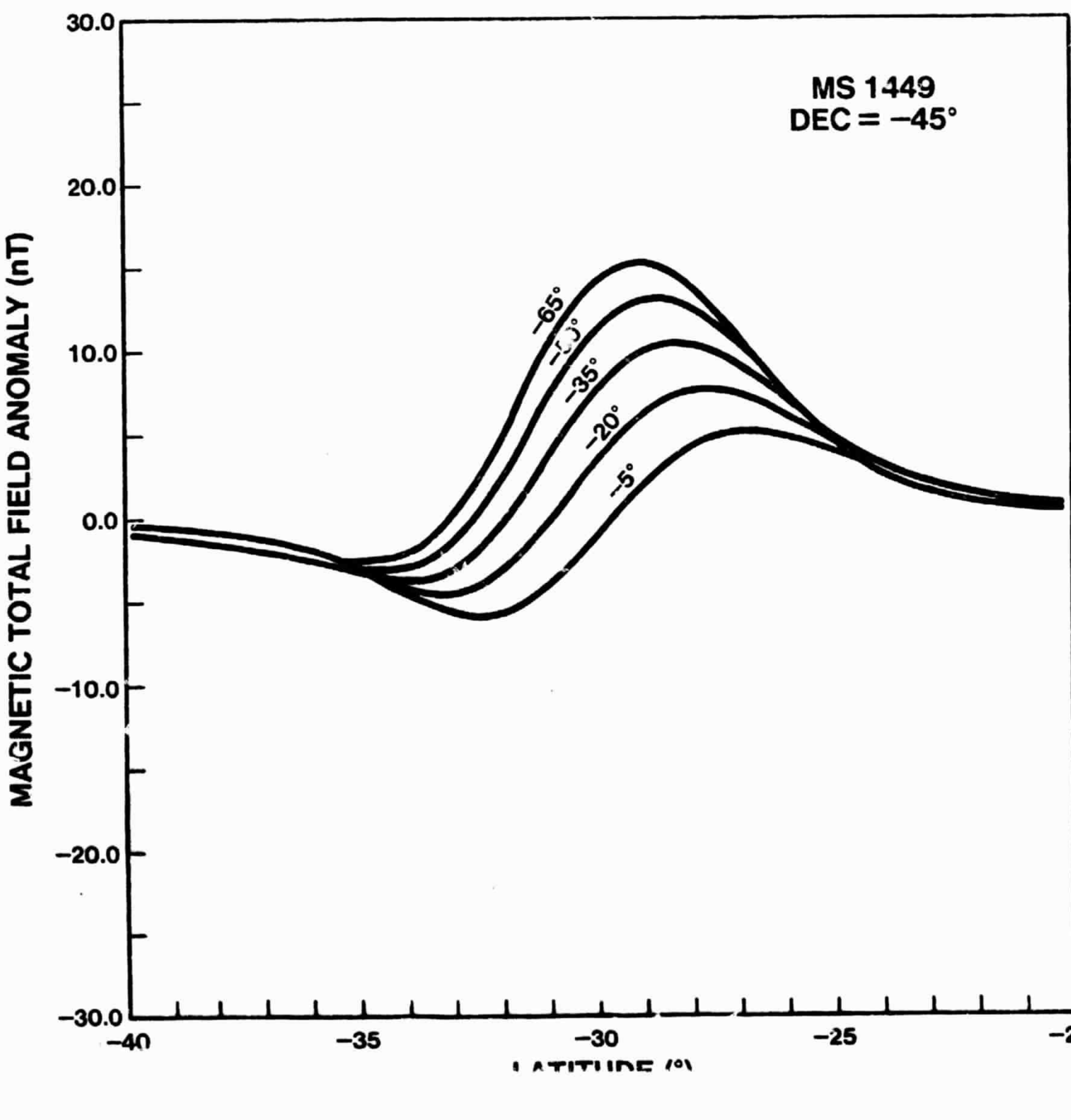


MS 2157
DEC = −45°
MAGNETIC TOTAL FIELD ANOMALY (nT)
30.0
20.0
10.0
0.0
−10.0
−20.0
−30.0
−40
−35
−30
−25
−20
LATITUDE (°)
−65°
−50°
−35°
−20°
−5°


30.0
20.0
10.0
0.0
−10.0
−20.0
−30.0
MAGNETIC TOTAL FIELD ANOMALY (nT)
MS 1449
INV – B
−40
−35
−30
−25
−20
LATITUDE (°)



MS 1449
DEC = −15°
30.0
2 .0
10.0
0.0
−10.0
−20.0
−30.0
MAGNETIC TOTAL FIELD ANOMALY (nT)
−65°
−50°
−35°
−20°
−5°
−40
−35
−30
−25
−20
LATITUDE (°)



MS 1449
DEC = −45°
MAGNETIC TOTAL FIELD ANOMALY (nT)
30.0
20.0
10.0
0.0
−10.0
−20.0
−30.0
−65°
−35°
−20°
−5°
−40
−35
−30
−25
−20

N85
23220

UNCLAS

N85-23220

PRECEDENCY CONTROL AND OTHER SEMANTIC INTEGRITY ISSUES IN A WORKBENCH DATABASE

by C.N.G. Dampney

Computing Discipline, School of Mathematics and Physics
Macquarie University, North Ryde, N.S.W. 2113, Australia.

ABSTRACT

Most database systems model the current state of a system of real world discrete and simple entities together with their relationships. By examining instead a database system that is a workbench and models more complicated entities, a fresh perspective is gained. Specifically, semantic integrity is analysed. Four aspects distinct from physical integrity are identified, namely - access, failure, concurrency and precedency. Access control is shown to be the consequence of semantic interdependency between data and its matching semantic routines. Failure, concurrency and precedency controls are concerned with preventing processes interfering with each other. Precedency is a new concept in the database context. It expresses a constraint between processes that act on the database. As processes create, update and delete entities they in general obey a partial ordering imposed by the semantics of their actions. Precedency control ensures that data remains consistent with respect to this partial order.

GAINING A NEW PERSPECTIVE OF DATABASE SYSTEMS

Information systems provide the context and rationale for most current database management systems. Amongst database systems that are commercially available, most cater for the needs of business and administrative information systems. These systems are used to track the current state of inter-related and dynamically changing discrete entities of interest to their business or administrative organisation.

In this paper we examine a database set up for a different purpose. We gain fresh insight by changing our perspective to a database that;

1) supports a *workbench* environment, and
2) models more complicated entities, in this case *continuous entities*.

Figure 1 suggests a classification of database contexts.

Figure 1. Database contexts.

Database environments

Tracking database - a database tracking the dynamic changes in a system of real world entities. All changes are permanently applied.

Workbench database - a database supporting a workbench environment. Changes are tentative and may be undone or superceded. Several versions of data are kept.

Entities represented

Discrete (simple) entity database - a database representing discrete (and simple) entities - basic storage unit of data is a fixed length segment.

Continuous entity database - a database representing continua*. Basic storage unit of data is a variable length segment.

The property of "data independence" is fundamental to databases, yet it is often misused. The independence only applies to the "immunity of applications to change in storage structure and access strategy" [1, page 13]. It does not go further and mean that data and application are independent of each other - quite to the contrary so far as semantic integrity is concerned.

As database facilities have improved the emphasis has shifted from providing access to data to providing access to information. Not only is the data itself made available independent of physical storage and access concerns, but the meaning of the data is protected and kept consistent within the context of the real world objects it models. This requires more refined integrity control.

Analogous to operand and operator, data is incomplete without the routines that modify it. Data alone merely represents one state of the real world object it models. To be complete, the routines that modify data according to the properties of its real world counterpart are also necessary. The term *semantic integrity* is used to designate "the correctness of database information in the presence of user modifications" [2, page 7].

Footnote

** "continuum", n., a whole, the structure of whose parts is continuous and not atomic. [Pocket Oxford]. To emphasis the distinction from* **discrete** *entities, we use "continuous entities" instead.*

Such modifications are effected through *semantic routines* [3] which are generally, but not necessarily, within the application programs. If semantics are simple, they may instead be at least partly expressed within the database [4-6]. Thus semantic integrity is concerned with maintaining the integrity of the connections between data and its semantics. This has three aspects:-

(1) *Access control* which controls data flow between the DBMS interface and the semantic routines within application programs lying outside the database system. This may be implemented using sub-schemas specified by the database administrator [2, page 85] or possibly a capability mechanism [7];

(2) *Datastore control or physical integrity* which ensures that data is not corrupted within the DBMS between the interface and the datastore; and

(3) *Connection sequencing* which constrains as necessary the order in which data flows between the datastore and the application programs.

Our objective in this paper is to analyse the connection sequencing aspects of semantic integrity. An experimental workbench database for modelling continua, the General Array DataBase [8], was used as a practical point of reference. An important result of this analysis is a new aspect of semantic integrity, namely *precedency*. It has been given little if any attention in the conventional database environment. Some work has been done in the CAD environment [9].

As is seen in section 2 process (or transaction) sequencing is more explicit in a system of continuous entities compared to discrete entities. In contrast to requirements for modelling discrete entities, more complicated entities dictate some basic differences in the databases supporting them.

Sequencing problems are more pronounced in a workbench environment. In section 3 interactions between transactions are analysed. The concept of *precedency control* as an important aspect of semantic integrity emerges. We show that all possible sequencing interactions between transactions can be correctly constrained by concurrency, precedency and failure control.

An architecture for such a database becomes evident and is briefly summarised in the conclusion.

CONTINUOUS ENTITIES AND THEIR RELATIONSHIPS

Databases for representing discrete entities [10] dominate. Whether they correspond to the hierarchical, network or relational model, their purpose is to represent discrete entities. Their elemental physical data components are simple scalars of various types gathered into records or segments of fixed length.

If things other than discrete entities need to be represented within a database system then these databases may well be inappropriate. Text, pictures, images and maps are examples of objects that are not naturally represented within discrete (simple) entity databases. Databases specialised to these types of objects have been developed [11-14].

In our case a need arose to represent continuous entities sampled and measured in project MAGSAT [15]. The continua were magnetic fields caused by the Earth's core, crust and ionisphere and measured along space-craft traverses orbiting about 350 kilometres above the Earth's surface. The kind of data collected along each traverse is illustrated in figure 2.

Figure 2. MAGSAT measurements collected along a traverse.

Scalars

Traverse: 0746
Perigee altitude: 352

(plus more than 10 other scalars)

Arrays

Magnetic field: 23416.1, 23418.1, 23419.5. ..
Altitude: 322.1, 322.3, 323.1, ..

(plus another 23 arrays of various other types)

These continuous entities were measured at a series of discrete points along space-craft traverses. The measurements for each continuous entity i is held in an array $\underset{\sim}{a}_i$ of values. Each array for a given traverse $\underline{P}$ has the same number of elements ℓ_p as each continuum is measured at the same set of points. It is emphasized that these arrays are logically atomic wholes because their elements are meaningless outside their context - they are simply the artifacts of the discretising process necessary for digital recording.

The set of arrays MAGSAT_MEASUREMENTS M_0 holding all the measurements along a given traverse is thus defined as the tuple

$$M_0: \ \underline{P}, \underset{\sim}{a}_1, \underset{\sim}{a}_2 \dots \underset{\sim}{a}_n \qquad (1)$$

where $M_0 \in \mathcal{M}_0$ the set of all tuples for the traverses over the surveyed region. $\mathcal{M}_0$ is a relation in the context of relational theory. It is important to note that the individual attributes within a tuple are arrays rather than the scalars more familiar in conventional application of relational theory. There are also scalar attributes associated with each traverse P.

Thus (1) is extended to become

$$M_0: \ \underline{P}, \underset{\sim}{a}_1, \underset{\sim}{a}_2 \dots \underset{\sim}{a}_{n_1}, s_1, \dots s_{n_2} \qquad (2)$$

The database therefore consists of relations with tuples of arrays and scalars. The arrays are not of predefined fixed length, so that database systems which depend heavily on predefined fixed length basic data components are unsuitable.

However there is a more significant distinction between continuous entity and discrete entity databases.

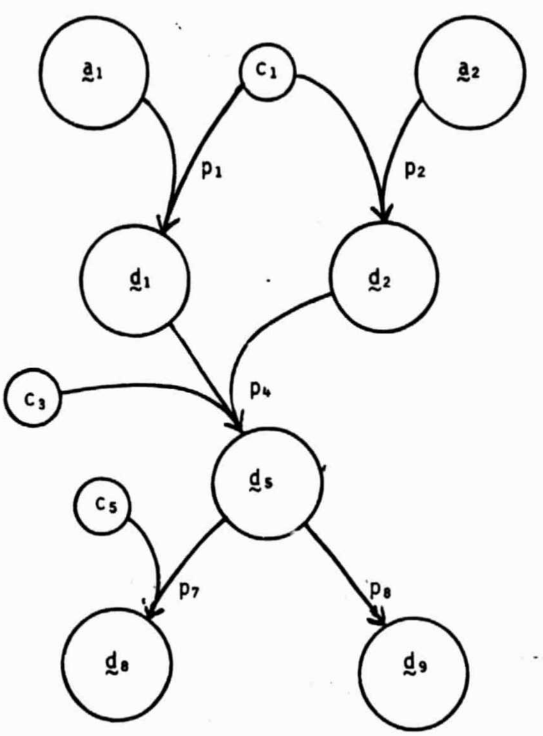


Figure 3. Typical functional dependency for a system of continuous entities. This figure can also be considered as a process flow diagram or a data precedency graph.

We first need to describe the nature of the processing applied to data such as that gathered for project MAGSAT. The processing has as its objective the reduction of the data so that it can be understood more easily and thus interpreted. These processes form a system. They can be modelled as suggested in figure 3. For each process a subset of the data entities $\{\underset{\sim}{a}\}$ or $\{\underset{\sim}{d}\}$, and $\{s\}$ are input together with control values $\tilde{c}$ determined interactively by the human interpreter. Further arrays $\underset{\sim}{d}$ are output. Representing the process by a function, say $F_{k\ell}$, we have for a simple case that

$$\underset{\sim}{d}_k, \underset{\sim}{d}_\ell = F_{k\ell}(\underset{\sim}{d}_i, c_j) \qquad (3)$$

More generally there may be a subset $k(\ell)$ of derived continuous entities $\{d\}_{k(\ell)}$ produced by a process $f_{k(\ell)}$ from a subset of control values and previously produced continuous entities and scalars.

To avoid cluttering the following discussion with unnecessary generalization for the purpose at hand, we keep the simple form of (3).

In terms of set theory (3) represents the intension of $F_{k\ell}$. The extension is represented by the relation $F_{k\ell}$ containing tuples of the form

$$f_{k\ell}: \underline{\underset{\sim}{d}_i, c_j}, \underset{\sim}{d}_k, d_\ell \qquad (4)$$

where $\underline{\underset{\sim}{d}_i, c_j}$ is the primary key in this relation. As well $\underset{\sim}{d}_i$ is a foreign key in some other relation.

A system of continuous entities can therefore be represented by the relation (2), together with many relations like (4). Such a representation would suffer the serious disadvantage that the keys such as $\underline{\underset{\sim}{d}_i, \underset{\sim}{c}_j}$ are huge. Duplicating them by maintaining separate relations and comparing them to check which tuples in the two relations are connected would be impractical.

Thus the derived continuous entities and control data are joined to relation M_0 (2) to give tuples of the form:-

$$M: \underline{P}, \underset{\sim}{a}_1, \ldots \underset{\sim}{a}_{n_1}, \underset{\sim}{d}_1, \ldots \underset{\sim}{d}_{n_2}, s_1 \ldots s_{n_3}, c_1, \ldots c_{n_4} \qquad (5)$$

Loosing the means to check the link between two relations does cause a problem. The functional dependencies represented in the separate relations and connected by a common key value (primary in one relation, foreign in the other) defining a partial order between the data entities. The partial order can be controlled by checking whether or not the connection defined through the key value is intact or broken. The partial order, as evident in figure 3 for example, is a *precedency* constraint.

With the relation in the form of (5) an alternative method must be found for what we now identify as *precedency control*. This is considered further in section 3. Precedency control performs a role similar to maintaining referential integrity [1, page 89] in the discrete entity case.

In relational terms (5) is not fully normalised. Functional dependencies not on the primary key such as in (4) are evident. It is emphasized again that the arrays $\underset{\sim}{a}$ and $\underset{\sim}{d}$ are regarded as atomic wholes. In a sense this bends the property of the presumed scalar nature of attributes in relation theory. But really, the essential property is that the attributes are logically indivisible atoms. Thus, in this slightly extended sense, <u>the relation consisting of tuples such as (5) is in second normal form</u>.

Representing Continuous Entities

Each of the various attributes that form a continuous entity are represented in the General Array DataBase as

<data-type-name> , *<value>* (6)

where *<data-type-name>* identifies the particular array ($\underset{\sim}{a}_i$ or $\underset{\sim}{d}_i$) or scalar (c_i or s_i), and *<value>* contains the array or scalar value itself. As described further in [8], a data dictionary holds the relevant properties of a data type necessary to calculate the storage size of *value*. If it is an array value the length of the array ℓ_p is held in the database associated with the key P.

A *<node-name>* corresponds to each key P. Thus the set of continuous entities are represented by a set of sets:-

{*<node-name>*,{*<date-type-name><value>*}*}* (7)

Figure 4 shows how the various data entity members:-

$$(\langle node\text{-}name \rangle, \langle data\text{-}type\text{-}name \rangle, \langle value \rangle) \qquad (8)$$

are managed by the GADB system. Each data entity could without semantic integrity checks in place, be independently created (PUT), retrieved (GET) and destroyed (DELETE). Functional dependencies must be explicitly enforced by precedency constraints.

Figure 4. Logical form of GADB database call.

<GADB database call>::=<request> <result>

where

<request> ::=	*<Process-name>,<action>, <node-name>,<data-type-name>, <variable name>,<length>*

and

<Process name>	is the program name of the calling process
<action>	GET|PUT|DELETE|UNDELETE|INQUIRE
<node-name>	is the key of the continuous entity
<data-type-name>	is the name of the attribute
<variable-name>	is the process's storage location into which values are to be got (GET) from the database, or from which values are to be PUT into the database
<length>	is the number of value elements
<result>	is the process's storage location into which the result of the action is recorded.

THE WORKBENCH ENVIRONMENT - IMPACT ON CONCURRENCY, PRECEDENCY AND FAILURE CONTROL

Software development, engineering design and geophysical data interpretation use computer systems as a workbench. These applications have a common need for facilities for building things - software, engineering structure or a geological interpretation.

An essential feature of such workbench databases is that changes to their contents are tentative. In contrast, changes to a tracking database, are permanent. For example, depostis and withdrawals into a bank account cannot be lost.

Process Flow Graphs With Multiple Versions of Data

An example of a process flow graph showing a single version only of each data entity is given in figure 5. The processes are represented by the multi-arcs connecting data entities. The direction of the arrows define input and output data.

Multiple versions of data are represented by overlaying nodes. For example Figures 8 and 9 show multiple versions of data entities. The most recent data version is on top with other versions partially obscured. Processes that have produced data that is now overwritten are shown in dotted form.

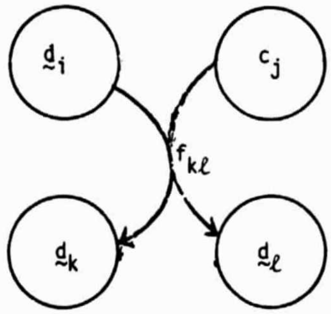


Figure 5. Process flow.

The Nature of the Workbench Environment

A typical process in the workbench environment is modelled by figure 5. Its control data c_j represents information provided by the user. Suppose the user is responsible for interpreting the geological causes that produce the geophysical effects measured and stored in a database. It is his task to make sense of the data confronting him by recognising the components of it that can be associated with identifiable causes. It is a task that by its very nature is not completely defined. He must make several attempts with various values of control data to process the measurements. In this way the interpreter interacts with the system. He sometimes also directs that a process be aborted and its effects wiped from the system. Various versions of data elements may be generated and sometimes deleted and possibly later "undeleted". The database must remain globally and locally consistent under the impact of these changes which may occur concurrently if several interpreters are at work.

The work of interpretation requires that new versions can be destroyed or undone. As interpretation involves many separate steps that are individually committed, it is desirable that once the consequences of several steps become evident, the interpreter can undo his work. This work may occur over several days.

While all these experimental steps are made and withdrawn the functional dependencies between the data elements must remain intact. The facility to maintain the precedency constraints over the functional dependencies is called precedency control. The facility to wipe out all the effects of an aborted process is called failure control.

These same requirements arise also in software development and engineering design.

Interactions Between Processes

A process may have more than one cycle of inputting data, processing it and outputting. Providing a process commits at the end of each cycle it can be modelled as a series of transactions. A transaction is a sequence

$$T_j : readset_j : Process : writeset_j$$

An interaction occurs when two (or more) transactions have overlapping readsets or writesets.

The following is an informal derivation of the controls necessary to maintain consistency. A more detailed analysis, which includes loops in the process flow chart, is given elsewhere [16]. The basic concept is *version consistency*.

Definition 1: Two data entities are *version consistent* if they were derived from the same versions of their common predecessors. For example in figure 3, the same version of c_1 should be used to calculate d_1 and d_2, before they in turn are used to calculate d_5.

Definition 2: A database is consistent if no data entity has been derived from data entities that are version inconsistent with respect to each other.

Note that with this definition a database is consistent even if it contains version inconsistent data, but such data cannot be used further.

Consider a single transaction T

$$d_1, d_2, c \xrightarrow{T} d_3, d_4$$

noting, of course, that the control data c is provided by the user when the transaction is run. Providing d_1 and d_2 are version consistent and no other transactions interfere, then d_3 and d_4 are version consistent.

We analyse interference between pairs of transactions T and T' where

$$d_1', d_2', c' \xrightarrow{T'} d_3', d_4'.$$

Suppose T and T' interact on a data entity X. There are four different kinds of interaction as shown below. We examine whether X remains version consistent with all data entities involved while T and T' interact in all possible ways. As version consistency is an equivalence relationship* this covers version consistency between all data entities in T and T'. Applying equivalence again it covers version consistency between all data entities in all transactions.

In cases 1, 2 and 3 below T and T' interact on X only. In case 4 we extend the analysis to the various possibilities with increasingly overlapped readsets and writesets. These possibilities include cases 1, 2 and 3 so extended.

In each case the conclusion is evident by inspecting the process flowcharts. Providing each transaction is restricted to reading everything before it writes anything then the cases given below include every possible interaction involving X. Between each action given below any number of actions not involving X, including *Terminate* (except for T in case 2b), may occur.

Footnote

* *Version consistency is reflexive, symmetric and transitive.*

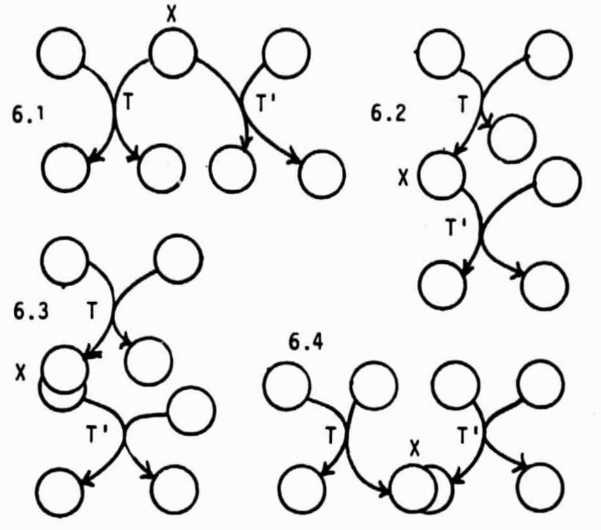


Figure 6. Interactions between two transactions with only one overlapping data entity.

Case 1 (figure 6.1)

T : READ X ; T' : READ X

Conclusion - no possible inconsistency.

Case 2 (figure 6.2)

T : WRITE X ; T' : READ X

Conclusion - no possible inconsistency.

Case 2b (figure 7)

T : WRITE X ; T' ; READ X ; T : Abort

A value written by T, is read by T', then recognised as erroneous by T' which then aborts. Conclusion - X is erroneous and inconsistent values are written by T'.

Case 3 (figures 6.3 and 9)

T : READ X ; T' : WRITE X

Conclusion - inconsistent values are written out by T relative to value in X whether written before or after X is written.

Case 4 (figures 6.4 and 8)

T : WRITE X ; T' : WRITE X

Conclusion - value written by T is lost. If readsets and or writesets of T and T' overlap then inconsistencies relative to X between other values read in and written out can occur.

Failure, Concurrency and Precedency Controls

Version inconsistency can be prevented in cases 2b and 4 by concurrency control. However in case 3, even if T and T' are not concurrent, version inconsistency can still occur. Version inconsistency can occur even with strict serializability enforced [17]. Additional precedency constraints are necessary to maintain semantic integrity. It is therefore apparent that version consistency is a stronger requirement than transaction consistency as defined in [18].

<u>Failure Control</u>. Failure control ensures that the effects of an aborted process on the database are completely wiped out (Figure 7). Concurrency control ensures that a failed transaction does not propogate erroneous data through the system.

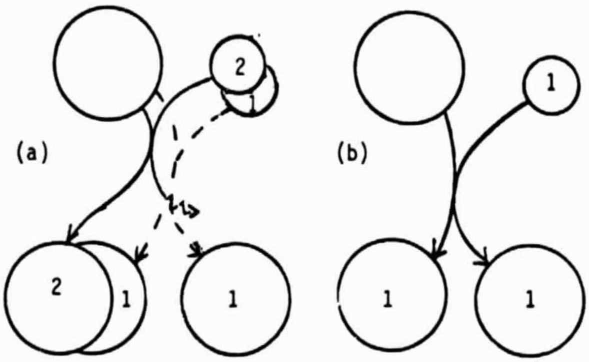


Figure 7. Failure control eliminates all the effects of aborting a transaction (a) back to the previous state (b). While a transaction is in progress access to writesets could be blocked by concurrency control.

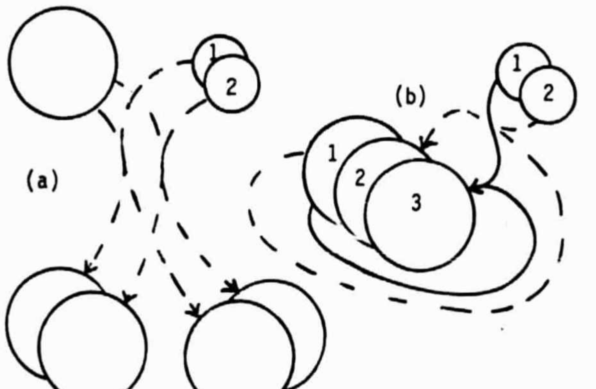


Figure 8. Concurrency control would prevent these lost updates and inconsistency. This represents the situation in Case 4 where a data entity is (a) in either the readset or the writeset or (b) in both the readset and the writeset.

<u>Concurrency control</u>. Concurrency control is well covered elsewhere [17], [19].

Consider case 4 in section 3.3. If T and T' are the same (figure 8) then concurrent interference between them can cause data to be effectively lost. With one data entity output the effect of the most recent value of control data may be lost. With more than one data entity output the effects of all control data used in the interfering transactions can be lost because output data is version inconsistent with each other. This may still be acceptable in a workbench environment where no guarantee is given that the consequences of all updates are subsequently used.

Precedency control, which ensures that version inconsistent data is not used, would be sufficient to maintain database consistency in the sense we have defined it. <u>The one aspect of concurrency control still necessary is to ensure that data written by a failed transaction is not propagated</u>. This could be ensured by preventing access to data written since any currently executing transaction started. Version inconsistent data may still result (cases 3 or 4), but this problem could be left to precedency control.

Alternatively concurrency control could reduce the burden on precedency control. Apart from preventing version inconsistent data in case 4, concurrency control would also prevent values read by a transaction being made obsolete, and hence the values written version inconsistent, before the transaction completes in case 3.

Methods for concurrency control by locking to prevent conflict or logging so that roll back can occur on detection are well known [19].

<u>Precedency Control</u>

Precedency control is concerned with maintaining the consistency of data with respect to the partial order constraint imposed by functional dependencies. Figure 9 shows a precedency control failure.

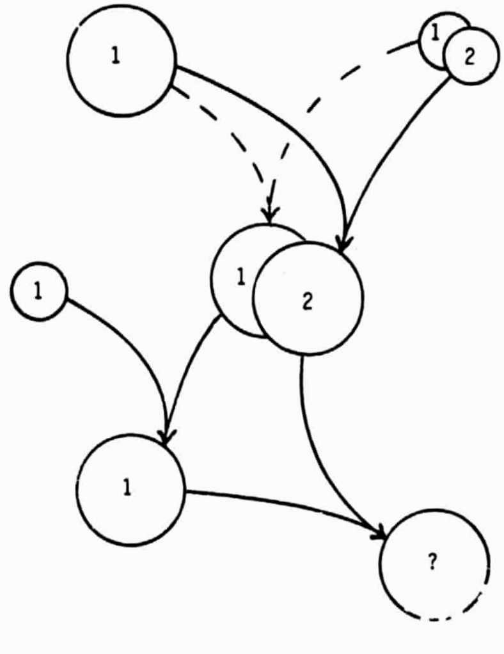


Figure 9. Inconsistency caused by lack of precedency control.

The problem is caused by different versions of a common predecessor for two or more distinct members of a transaction's readset.

Overview of an Algorithm for Precedency Control.

For a given Transaction T_i with readset $R_i = (r_1^{(i)} \ldots r_n^{(i)})$ and writeset $W_i = (w_1^{(i)} \ldots w_m^{(i)})$ the basis common predecessor set $P(R_i)$ is found. Each $p_j \in P(R_i)$ satisfies the property that it is a common predecessor to more than one member of the readset R_i if

$$p_j \in P(r_a^{(i)}) \text{ and } p_j \in P(r_b^{(i)}) \text{ for some } a \neq b$$

and has at least one immediate successor that does not satisfy

$$s_k \in P(r_a^{(i)}) \text{ and } s_k \in P(r_b^{(i)}) \text{ for the same } a \neq b.$$

Precedency can then be checked by ensuring that the same version of each p_j is the predecessor to all its successors in the read set R_i.

One technique is to maintain transaction histories with each transaction logged with entries

Transaction name
Readset - all data entities input by data name and timestamp
Writeset - all data entities output by data name and timestamp.

From this a process flow graph can be constructed and used to ensure that the versions of common predecessors of the readset are consistent.

Precedency control could be activated with each transaction. Alternatively, if some database inconsistency is tolerable for a period of time then precedency could be checked periodically and inconsistent data flushed out.

Which specific techniques are effective in the various interactive workbench environments has yet to be examined in detail. It is clear that concurrency and precedency control techniques need to be compatible. One issue is whether some kind of automated assistance should be provided to the user to generate required version consistent data. Should precedency constraints be registered with the database system when it is set up, or is the transaction history log the best way to determine it? The trade-off in this last question is between a strictly controlled environment which requires the user to register constraints initially or a more free wheeling situation with higher overheads where the system gathers the information automatically to determine the constraints.

Other Guises of Precedency Control. Precedency constraints are present in tracking databases modelling discrete entities. Virtual data [1, page 16] is a form of it. When a virtual data item is requested, a virtual transaction is run to produce it. This ensures consistency with the current version of its predecessors. The reverse situation occurs when successors are automatically generated by consequential updates.

In the CODASYL database model, the membership class [1, page 412], specifically RETENTION and INSERTION clauses, is a precedency control mechanism. For example, if a particular entity is erased from the database then its successors (dependents in CODASYL terms) are automatically erased as well.

It can be seen that neither the virtualising nor the membership class mechanisms are suitable for the workbench environment described here. Virtualising is impractical as the computations invoked are lengthy and require further information, the control data, from the user. Membership class is unsuitable as older versions need to be retained in case newer versions are discarded.

It is also apparent that if control data is required from the user then automatically generating successors is impossible. In this sense the need for precedency control is a consequence of an interactive workbench environment.

CONCLUSION

The various controls necessary for maintaining semantic integrity have been identified.

Figure 10 summaries the purpose of the various controls necessary to maintain integrity and correct sequencing of connections between data and semantic routines. Physical integrity is maintained by data store control. It is concerned with ensuring that data entities stored and linked with other data entities are not corrupted by the physical storage and I/O environment.

Semantic Control	Purpose
Access control	semantic routine - data connection
Precedency control	process to data to process sequencing
Concurrency control	data to process to data sequencing
Failure control	process integrity
Data store control	data integrity

Figure 10. Summary of semantic integrity controls.

Figure 11 shows the various layers of control implemented for the GADB. At this stage the details of the precedency control mechanism are being investigated. Access as the outer most control layer and data store as the innermost is evident from their purpose in section 1. Precedency, concurrency and failure control layers are concerned with increasingly restricted scope and length of transaction history.

One is finally left with a computational model of the entire system of transactions and data which is logically the same as the data flow graph [20] of a single program. The one distinction is that the system dynamics and control data is firmly under human rather than automated control. We therefore also have a model of man-computer interaction!

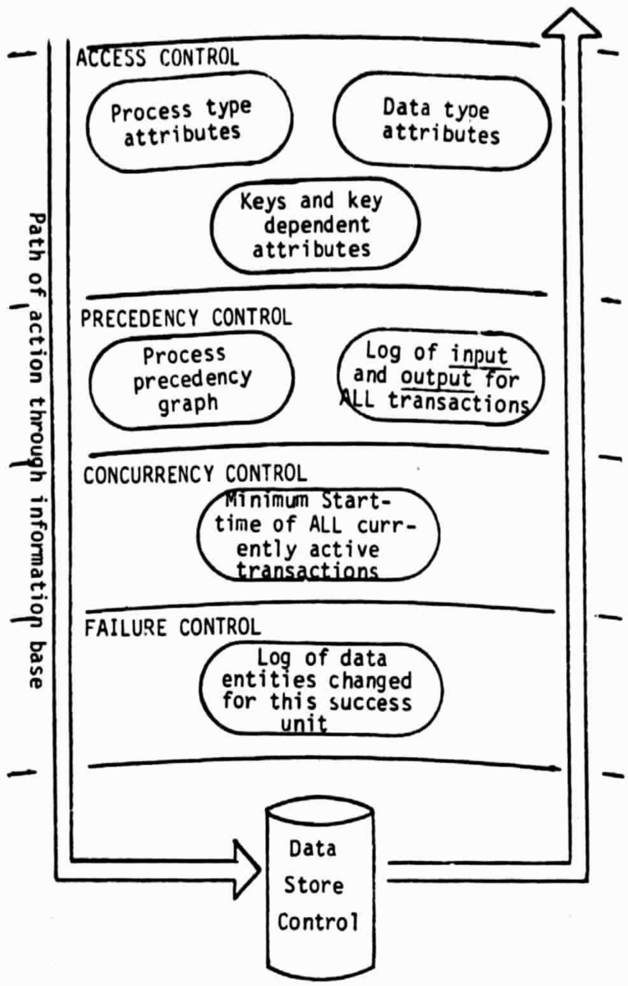


Figure 11. Layers of control supporting semantic integrity in a database.

ACKNOWLEDGEMENTS

I gratefully acknowledge the constructive criticisms offered by the referees who reviewed the earlier version of this paper. One referee in particular offered detailed comments on several sections of the paper. I also thank Professor J.B. Hext for comments that helped clarify some points.

This work was supported by a 1982 Australian Computer Research Board grant and partly by Australian Research Grants Scheme Grant E8015330R.

REFERENCES

[1] Date, C.J., 1981. An Introduction to Database Systems. 3rd edition Addison-Wesley.

[2] Fernandez, E.E., E.C. Summers and C. Wood 1981. Database Security and Integrity. Addison-Wesley, New York.

[3] Keedy, J.L. and I. Richards, 1982. "A software engineering view of files". The Australian Computer Journal, Vol.14, No.2, pp 56-74.

[4] Theerachetmangkol, A. and A.Y. Montgomery, 1980. "Semantic Integrity constraints in the Query by Example Data Base Management Language". The Australian Computer Journal, Vol.12, pp 28-42.

[5] Stonebraker, et al., 1976. "The design and implementation of INGRES". ACM Trans. on Database systems, Vol.1, No.3, pp 189-222.

[6] Nijsson, G.M., 1981. "An architecture for knowledge based software". Unpublished paper presented to the Australian Computer Society, July 30, 1981, Melbourne.

[7] Fabry, R.S., 1974. "Capability based addressing". Communications of the ACM, Vol.17, No.7, pp 403-412.

[8] Dampney, C.N.G. and B.D. Johnson, 1980. "GADB - a database facility for modelling time/space referenced continua". Technical report 79-006, School of Mathematics and Physics, Macquarie University.

[9] Newmann, T. and C. Horning, 1982. "Consistency and Transactions in CAD database". Proceedings of the 8th VLDB Conference, Sept. 1982, pp 181-188.

[10] Chen, P.P.S., 1976. "The entity-relationship model - towards a unified view of data". ACM Transactions on DataBase Systems, Vol.1, No.1, pp 9-36.

[11] IBM. STAIRS (Storage and Information Retrieval System) Reference Manual. IBM publication GH12-5114.

[12] Westlake, B.J. and M.N. Groth, 1980. "IVA - a technical database for the 80's and beyond". Symposium - CAD/CAM in the 80's and Integration through Databases Melbourne, 20-22, October 1980.

[13] Chang, S.K. and T.L. Kunii. 1981. "Pictorial database systems". Computer Vol.14, No.11, pp 13-22.

[14] Nagy, G. and S. Wagle, 1979. "Geographic data processing". ACM Computing Surveys, Vol.11, pp 139-81.

[15] American Geophysical Union, 1982. Special issue "MAGSAT preliminary results". Geophysical Research Letters, Vol.9, No.4, pp 239-376.

[16] Dampney, C.N.G., 1983. "Version consistency and database integrity", Technical report 83, Computing Discipline, School of Mathematics and Physics, Macquarie University.

[17] Bernstein, P.A., et al, 1979. "Formal aspects of serializability in database concurrency control". IEEE Trans. on Software Engineering Vol. SE-5, No.3 pp 203-216.

[18] Traiger, I.L., et al, 1982. "Transactions and Consistency in Distributed Database Systems". ACM Trans. on Database Systems, Vol.7, No.3, pp 323 342.

[19] Gray, J.N., 1978. "Notes on database operating systems". In Operating Systems - An Advanced Course. R.Bayer et al (Eds.) Springer-Verlag, pp 393-481.

[20] Karp, R.M., and R.E. Miller, 1966. "Properties of a model for parallel computations: determinacy, termination, queueing". SIAM J.Applied Mathematics, Vol.14, No.6, Nov. 1966, pp 1390-1141.

END

DATE

FILMED

JUN 7 1985