GL01346

USER'S IDENTIFIATION CODE: AFDB46
PASSWORD: LOAD

SUBJ
COMP
AFBD

USER'S GUIDE TO THE

ALTERNATIVE FUELS DATA BANK

APRIL 1979

DEPARTMENT OF ENERGY

BARTLESVILLE ENERGY TECHNOLOGY CENTER

BARTLESVILLE, OK

Problems — call Jerry Allsup (mechanical engineer, Div. of utilizatio
commercial: (918) 336-2400 ext. 258
FTS: 735-4258

TABLE OF CONTENTS

ALTERNATIVE FUELS DATA BANK

Introduction

The Department of Energy's Bartlesville Energy Technology Center (BETC) with the support of the Transportation Energy Conservation Division has established an information system that is designed to provide rapid access to information concerning alternative fuels for transportation uses. This information system is in the form of a data base which operates on the PDP 11-70 computer located at BETC. This document is intended to provide the user with sufficient instructions to effectively use the system via remote terminal/commercial telephone access.

Data Base Description

The Alternative Fuels Data Bank (AFDB) contains bibliographical information of publications and summaries of current research activities that deal with the use of fuels from non-petroleum sources and nonconventional fuels from petroleum sources in transportation applications. The information that is to be stored in the data base is collected by BETC personnel; screened to assure the material pertains to the subject area; coded by assigning keyword statements that characterize the information; and entered into the central computer at BETC. The stored information can then be retrieved by matching the descriptive keyword statements, supplied by the individual requesting the information, with the keyword statements stored with each item of data in the data base. The response from the information system is in the form of printed reports that list bibliographical information, and a short synopsis of pertinent publications, and/or summaries of current research activities. The information in the data base is divided into three major areas: publications, current research, and topics. The method for coding and retrieving the information in each of these areas is slightly different. The information in the "publications" file is coded and retrieved by selecting the appropriate keywords in the following categories:

a. Type of fuel (methanol, ethanol, etc.--listed in Table 1 of Appendix I)

b. Type of application or investigation (engine tests, fleet study, etc.--listed in Table 2 of Appendix I)

c. Object of investigation (fuel economy, emissions, etc.--listed in Table 3 of Appendix I)

d. Degree of detail (review, bibliographical survey, etc.--listed in Table 4 of Appendix I)

e. Last name of primary author (Adelman, Adt, Allsup, etc.--open to search for any author specified)

f. Year of printing (00-99--open to any year specified)

g. Author's affiliation (ARCO, Argonne, APIUC, etc.--listed in Table 5 of Appendix I)

The information in the "current research" file is coded and retrieved by selecting the appropriate keywords in the following categories:

a. Type of fuel (ethanol, broadcut, methanol, etc.--listed in Table 1 of Appendix I)

b. Type of investigation (engine tests, fleet study, etc.--listed in Table 2 of Appendix I)

c. Objective of investigation (fuel economy, emissions, combustion characteristics, etc.--listed in Table 3 of Appendix I)

d. Performing organization (ARCO, Argonne, BETC, etc.--listed in Table 5 of Appendix I)

e. Last name of principal investigator (Adt, Adelman, Bernhardt, etc.--open, no listing)

f. Sponsoring organization (DOE-TEC, State of California, APIUC, etc.--listed in Table 6 of Appendix I)

g. Level of funding (dollar amount listed for the current fiscal year; e.g., $400,000, $200,000--open, no listing)

The third information file "topics" is comprised of a collection of information on a particular subject. The information is not coded or retrieved using keywords, but rather once the program to select a topic is called, a display of topic titles with serial numbers is displayed. The user selects the serial number associated with the particular topic of interest, and the computer will produce a copy of the preloaded discussion of that particular subject.

AFDB Access Procedure

Remote terminal access to the data base can be made using a computer terminal that will operate at 300 baud rate (30 characters per second) and is equipped with an acoustical coupler. The access procedure is outlined as follows:

a. Turn terminal on and set the terminal for 300 baud rate, full duplex.

b. Dial 918-336-1051.

c. Check for carrier signal tone after your call is answered.

d. Place receiver into the acoustical coupler cradle; carrier signal light should be on, signifying that your terminal is on line.

e. Sign onto the computer using your assigned user identification code by typing "Hello XXXXXX" (substitute the assigned user identification code for "XXXXXX").

f. The computer will then prompt the user for a password. (The password and appropriate user identification code will be supplied upon request.)

g. Call the appropriate program for the particular file you wish to access. (i.e., "@BILDIT" for publication and current research, or "@PRTSELTOP" for topics)

h. Conduct your search according to the procedure illustrated in Appendix II using keyword codes specified in Appendix I. The computer will prompt the user to select keywords in each subject area.

i. Once the search is complete, sign off by typing "BYE" and pressing the carriage return key.

j. Remove the telephone receiver from the acoustical coupler and replace the receiver on the telephone.

Access to the data base via remote terminal access is available 24 hours a day, seven days a week, except for Monday morning from 8:00 a.m. to 12:30 p.m. CST (or CDT), or in the event of a computer outage. Any difficulties encountered when trying to access the system or when making a search, should be referred to Ken Stamper at 918-336-2400, ext. 258 (FTS: 735-4258).

Source Document Coverage

Data, announcements of new publications, announcements of new contracts, publications, and information describing research activities pertinent to the Alternative Fuels Data Bank are gathered from a number of diverse sources. Those source documents that are reviewed on a regular basis include:

a. Periodicals

1. Chemical and Engineering News
2. Commerce Business Daily
3. Automotive Engineering
4. Oil and Gas Journal
5. Progress in Energy Combustion Science
6. Fuel
7. Combustion

b. Abstract Service Publications

1. Energy Conservation Update
2. Fossil Energy Update
3. SAE Transactions
4. Petroleum Refining and Petro Chemicals Section of API Abstracts/ Literature

c. Technical Society Papers

1. Society of Automotive Engineering Papers
2. American Chemical Society Papers

d. Symposium and Conference Proceedings

1. International Alcohol Fuel Technology Symposia
2. DOE Highway Vehicle Systems Contractor Coordination Meetings

e. Newsletters

1. Air and Water News
2. Energy Digest
3. Energy Research Digest

f. Project Progress and Final Reports

1. DOE-TEC Contractors
2. Co-operative Contractors working in related areas

APPENDIX I

AFDB GLOSSARY OF KEYWORDS

TABLE 1: Type of Fuel - Keywords for Category "a" of the Publications File and Category "a" of the Current Research File.

1. Broadcut — Keyword Code: BROADCUT. Any petroleum derived, or non-petroleum derived hydrocarbon liquid with a boiling range wider than conventional distillate fuels; i.e., 150° F to 600° F.

2. Coal — Keyword Code: COAL. Pulverised coal suspended in a hydrocarbon liquid.

3. Emulsion — Keyword Code: EMULSION. A liquid fuel system containing a hydrocarbon liquid and a water or alcohol (not normally miscible), one of which is dispersed as globules in the other.

4. Ethanol — Keyword Code: ETOH. Neat (pure) ethyl alcohol or fuel grade ethyl alcohol.

5. Ethanol-Blend — Keyword Code: ETOH BLEND. A blend of petroleum derived or non-petroleum derived hydrocarbon fuel and ethyl alcohol; i.e., either blended and stored in a single tank or dynamically blended from two tanks.

6. Ethers — Keyword Code: ETHER. Any ether used in pure form or blended with hydrocarbon liquid; typically, methyl-tertiary butyl ether.

7. Hydrogen — Keyword Code: HYDROGEN. Hydrogen used in pure form.

8. Methanol — Keyword Code: MEOH. Neat (pure) methyl alcohol.

9. Methanol-Blend — Keyword Code: MEOH BLEND. A blend of petroleum derived or a non-petroleum derived hydrocarbon fuel and methyl alcohol; either blended and distributed as a single liquid or delivered to the engine as two separated fuels and dynamically blended in the intake system or combustion chamber.

10. Methyl Fuel — Keyword Code: METHYL FUEL. A commercial process grade methyl alcohol; typically, the major constituent is methyl alcohol with smaller concentrations of higher alcohols either blended with a hydrocarbon fuel or used in the pure form.

11. Synthetic Fuels from Coal — Keyword Code: SYNFUEL/COAL. Any liquid hydrocarbon fuel derived from coal either in the unrefined form or refined form, straight, or blended with a petroleum derived stock; i.e., Sasol and Fischer-Tropsch liquids.

12. Synthetic Fuels from Shale — Keyword Code: SYNFUEL/SHALE. Any liquid hydrocarbon fuel derived from oil shale, either in the unrefined or refined form, straight, or blended with a petroleum derived stock.

13. Synthetic Fuels from Waste — Keyword Code: SYNFUEL/BIOMASS. Any liquid fuel produced from renewable sources; i.e., animal or plant wastes, agricultural or forest materials used in blends with hydrocarbon liquids or in neat form.

14. Tertiary Butyl Alcohol — Keyword Code: TBA. Tertiary butyl alcohol either blended with a petroleum derived liquid or in the neat form.

APPENDIX I
(cont'd)
AFDB GLOSSARY OF KEYWORDS

TABLE 2: Type of Investigation - Keywords for Category "b" of the Publications File and Category "b" of the Current Research File.

1.	Combustion Characteristics	Keyword Code: COMBUSTION. Combustion bomb studies.
2.	Engine Tests	Keyword Code: ENGINE. Engine tests from an engine dynamometer, including tests on multicylinder spark ignition and compression ignition engines, stratified charge engines, gas turbines, single cylinder (CFR, CLR) engines, distinguished from those tests made with the engine in a vehicle.
3.	Fleet Study	Keyword Code: FLEET STUDY. Chassis dynamometer or road tests in which one or more vehicles are used regardless of engine type.
4.	Large Scale Utilization Plans	Keyword Code: UTILIZATION. Studies in which scenarios of large scale marketing of alternative fuels are considered; i.e., methods for introducing the alternative fuels into commerce, long term or large scale economics, environmental impact of large scale production, and utilization.
5.	Miscellaneous	Keyword Code: MISC. Health, safety, toxicological, and environmental impact studies.
6.	Physical Properties	Keyword Code: PROPERTIES. Alternative fuel chemical and physical properties studies; i.e., solubility in a petroleum blend, corrosion characteristics, sulfur and/or nitrogen content, and boiling range.
7.	Production	Keyword Code: PRODUCTION. Manufacture and/or processing of alternative fuels.
8.	Refinery Schemes	Keyword Code: REFINERY. Investigations of the adjustments that must be made in the petroleum processing that will allow addition of the alternative fuel to the finished refinery product. Also, those adjustments which must be made to the processing equipment in the refinery to accomodate the addition of an alternative fuel to the refinery feedstock.

APPENDIX I
(cont'd)
AFDB GLOSSARY OF KEYWORDS

TABLE 3: Objective of Investigation - Keywords for Category "c" of the Publications File and Category "c" of the Current Research File.

1.	Combustion Characteristics	Keyword Code: COMBUSTION. Basic combustion study results from tests on alternative fuels; i.e., flame speeds, peak temperatures, flame luminosity, heating value, and combustion-chamber-pressure measurements.
2.	Development	Keyword Code: DEVELOPMENT. Engine, fuel or fuel production, or fuel processing developments or innovations that promote the effective utilization of alternative fuels.
3.	Distillation	Keyword Code: DISTILLATION. Distillation data or distillation curves generated from tests on alternative fuels or alternative fuel blends.
4.	Distribution	Keyword Code: DISTRIBUTION. Fuel distribution in the intake system of a particular engine or bulk fuel distribution in shipping fuels from manufacturers and processors to distribution points.
5.	Driveability	Keyword Code: DRIVEABILITY. User assessment vehicle performance, both those from CRC driveability tests and routine vehicle operation, using alternative fuels.
6.	Durability	Keyword Code: DURABILITY. Assessments of engine, vehicle, or emission control component life when using alternative fuels.
7.	Economics	Keyword Code: ECONOMICS. Relative costs associated with the production, processing, and/or utilization of alternative fuels.
8.	Efficiency	Keyword Code: EFFICIENCY. Engine fuel consumption and power characteristics, engine thermal efficiencies, or manufacturing and process efficiencies.
9.	Emissions	Keyword Code: EMISSIONS. Engine or vehicle exhaust pollutant emissions.
10.	End-Use Influences	Keyword Code: END-USE. Effects of using alternative fuels in engines or vehicles on vapor lock tendencies, cold start difficulties, and engine deposits.

11. Environmental Impact — Keyword Code: ENV IMPACT. Assessments of the environmental impact of processing and using alternative fuels; i.e., the impact of fuel spills, or pollutant emissions associated with use of alternative fuels.

12. Evaporative Emissions — Keyword Code: EVAP. Evaporative hydrocarbon emission test results from EPA shed tests.

13. Fuel Economy — Keyword Code: ECONOMY. In use fuel economy or fuel economy data from chassis dynamometer tests of one or more vehicles operating on alternative fuels.

14. Hydrocarbon Characterization — Keyword Code: HC. A detailed analysis of the hydrocarbon composition of unburned fuel in the exhaust of an engine or vehicle operating on alternative fuels.

15. Materials Compatability — Keyword Code: MATERIALS. Assessments of corrosion, swelling, hardening, etc. of materials in the fuel supply and distributing systems that come into contact with the alternative fuel.

16. Octane Numbers — Keyword Code: OCTANE. Octane ratings (research and/or motor) of neat alternative fuels or alternative fuels blended with petroleum base stocks.

17. Parametric Studies — Keyword Code: PARAMETRIC. A study of the influence of engine operating parameters (i.e., for a spark ignition, spark timing, A/F and/or exhaust gas recirculation rate) on fuel consumption and emissions characteristics of an engine or vehicle operating on alternative fuels.

18. Road Octane — Keyword Code: ROAD OCTANE. Octane measurements made with the vehicle operating either on a chassis dynamometer or on a test track on an alternative fuel (usually using the modified Uniontown technique).

19. Safety — Keyword Code: SAFETY. Studies of special precautions and procedures necessary to insure the safe handling and experimentation of alternative fuels.

20. Solubility — Keyword Code: SOLUBILITY. A study of the solubility of alternative fuels in petroleum based fuels; i.e., methanol/gasoline/water solutions, etc.

21. Temperature Effects — Keyword Code: TEMPERATURE. A study of the influence of ambient temperature on the fuel economy and emissions consequence of using alternative fuel in vehicles. Also a study of the temperature drop in the intake air fuel charge due to evaporative cooling of the fuel.

22. Toxicology — Keyword Code: TOX. A study of the health hazards associated with exposure to alternative fuels; i.e., TLV's, comparative toxicity values, etc.

23. Vapor Pressure — Keyword Code: PRESSURE. A study of the vapor pressure characteristics of alternative fuels blended with petroleum based stocks.

APPENDIX I
(cont'd)
AFDB GLOSSARY OF KEYWORDS

TABLE 4: Degree of Detail - Keywords for Category "d" of the Publications File.

1.	Review	Keyword Code: 1. A review of published material from a limited number of sources.
2.	Bibliographical Survey	Keyword Code: 3. A comprehensive review of published material on a particular subject area from a wide range of sources.
3.	Analytical Study	Keyword Code: 5. Computer modeling or simulation of, or prediction of test results from models.
4.	Original Experiment	Keyword Code: 7. Original data taken in an investigation.
5.	Original In-Depth Experiment	Keyword Code: 9. Original data taken in an investigation in which the experimental techniques were well defined and established, the equipment used, and the analyses of the data reflect the state of the art.

APPENDIX I
(cont'd)
AFDB GLOSSARY OF KEYWORDS

TABLE 5: Author's Affiliation or Performing Organization - Keywords for Category "g" of Publications File and Category "d" of the Current Research File.

AFFILIATION OR PERFORMING ORGANIZATION	KEYWORD CODE
1. Chrysler Corp.	CHRYSLER
2. Ford Motor Co.	FORD
3. General Motors	GM
4. Volvo	VOLVO
5. Volkswagen	VW
6. Cummins	CUMMINS
7. Carter Corp.	CARTER
8. AMOCO	AMOCO
9. Chevron	CHEVRON
10. Continental Oil Co.	CONOCO
11. Exxon	EXXON
12. Getty	GETTY
13. Gulf	GULF
14. Mobil	MOBIL
15. Phillips Petroleum Co.	PHILLIPS
16. Shell	SHELL
17. Sun Oil Co.	SUN
18. Suntech	SUNTECH
19. Texaco	TEXACO
20. Union Oil Co.	UNION
21. Arco	ARCO
22. Ashland	ASHLAND
23. Agricultural Products Industrial Utilization Committee	APIUC
24. Marathon	MARATHON
25. University of California, Berkley	UCB
26. University of California, Livermore	UCL
27. University of California, Los Angeles	UCLA
28. University of Nebraska	UN
29. Harvard	HARVARD
30. Massachusetts Institute of Technology	MIT

AFFILIATION OR PERFORMING ORGANIZATION	KEYWORD CODE
31. University of Miami	MIAMI
32. University of Michigan	MICHIGAN
33. University of Missouri	MISSOURI
34. Santa Clara University	SCU
35. Stanford University	STANFORD
36. Texas A & M	TEXAS A & M
37. Pennsylvania State University	PENN STATE
38. University of Minnesota	MINNESOTA
39. Society of Automotive Engineers	SAE
40. American Petroleum Institute	API
41. Battelle Laboratories	BATTELLE
42. Engineering Foundation	EF
43. Institute of Gas Technology	IGT
44. Math Science Northwest	MSNW
45. Southwest Research Institute	SWRI
46. Southern Research Institute	SRI
47. U.S. DOE Argonne National Laboratory	ARGONNE
48. California Air Resources Board	CARB
49. U.S. Environmental Protection Agency	EPA
50. Brookhaven National Lab	BNL
51. Mueller Associates	MUELLER
52. Other	OTHER

Work done in countries other than the U.S.A. by corporations that have not been specified above can be retrieved by the name of the country; i.e., keyword codes: BRAZIL, GERMANY, NEW ZEALAND, SWEDEN, JAPAN, ENGLAND, etc.

APPENDIX I
(cont'd)
AFDB GLOSSARY OF KEYWORDS

TABLE 6: Sponsoring Organization - Keyword Code for Category "f" of the Current Research File.

1. Agricultural Products Industrial Utilization Committee	Keyword Code:	APIUC
2. American Petroleum Institute	Keyword Code:	API
3. Department of Energy, Transportation Energy Conservation Division	Keyword Code:	DOE-TEC
4. Electric Power Research Institute	Keyword Code:	EPRI
5. Federal Republic of Germany	Keyword Code:	FRG
6. State of California	Keyword Code:	CAL

APPENDIX II

SAMPLE SEARCH OF PUBLICATIONS FILE

Once the link from the remote terminal is made with the computer (see AFDB access procedure) in Bartlesville, the information retrieval process for the publication file proceeds by the following:

1. Signing on the computer using the assigned user identification code
2. Typing the assigned password.
3. Calling the information retrieval program for the publications file "@BILDIT".

Once the retrieval program has been called, the user is asked to code his/her request using the keyword codes outlined in Appendix I. For example, if the user were looking for original published information on the subject of fuel economy of vehicles operating on methanol/gasoline blends, the coding process would go as follows:

Category a: Type of Fuel - Methanol blends implies MEOH BLEND.

Category b: Type of Application or Investigation - Vehicles implies FLEET STUDY.

Category c: Objective of Investigation - Fuel economy implies ECONOMY.

Category d: Degree of Detail - Original data implies 7 or 9; use 7.
(since only one degree of detail can be used at one time, this keyword statement will use 7, the following will use 9)

Category e: Last Name of Primary Author - Unspecified.

Category f: Year of Publication - Unspecified.

Category g: Author's Affiliation - Unspecified.

OR (since two degree of detail classes were specified, the search will be made using two keyword statements linked with the Boolean logic expression "OR" which will collect records that satisfy either keyword statement)

Category a: Type of Fuel - Methanol blends implies MEOH BLEND.

Category b: Type of Investigation - Vehicle implies FLEET STUDY.

Category c: Objective of Investigation - Fuel economy implies ECONOMY.

Category d: Degree of Detail - Original experimental data; use 9.

Category e: Last Name of Primary Author - Unspecified.

Category f: Year of Publication - Unspecified.

Category g: Author's Affiliation - Unspecified.

The information retrieved with this coded request would appear as follows:
(the underscored portions denote the user's response)

```
/HELLO XXXXXX  (use the assigned user identification code in place of "XXXXXX")
PASSWORD:____

        RSX-11M BL22    MULTI-USER SYSTEM

GOOD MORNING
20-APR-79 10:11 LOGGED ON TERMINAL TT25:

/@BILDIT
/PIP RETRIEVE.CMD/PU
/!

/* RETRIEVE PUBLICATIONS [1] OR ONGOING RESEARCH [2]? [0 R:1-2]: 1
/RUN [225,1]BLDPUB
Enter TYPE OF FUEL:  MEOH BLEND
Enter TYPE OF INVESTIGATION:  FLEET STUDY
Enter OBJECTIVE OF INVESTIGATION:  ECONOMY
Enter DEGREE OF DETAIL:  7
Enter LAST NAME OF PRIMARY AUTHOR:
Enter YEAR OF PUBLICATION:                          (if category unspecified, depress
Enter AUTHOR'S AFFILIATION:                          carriage return key)
  ENTER LOGICAL CONNECTOR FOR NEXT LINE.
  ENTER BLANK IF FINISHED.
OR
Enter TYPE OF FUEL:  MEOH BLEND
Enter TYPE OF INVESTIGATION:  FLEET STUDY
Enter OBJECTIVE OF INVESTIGATION:  ECONOMY
Enter DEGREE OF DETAIL:  9
Enter LAST NAME OF PRIMARY AUTHOR:
Enter YEAR OF PUBLICATION:
Enter AUTHOR'S AFFILIATION:
  ENTER LOGICAL CONNECTOR FOR NEXT LINE.
  ENTER BLANK IF FINISHED.

DO YOU WANT THE BIBLIOGRAPHIES PRINTED
IMMEDIATELY AFTER THEY ARE FOUND?  [Y/N]:Y
/!


/DTR @RETRIEVE
FINISH;READY FESR-PUBLICATIONS-DATA
FIND FESR-PUBLICATIONS-DATA WITH SERIAL NE "9999P" AND ((
((FUEL-01="MEOH BLEND" AND TYPE-01="FLEET STUDY" AND OBJ-01="ECONOMY" AND DEG-01="7") O
(FUEL-02="MEOH BLEND" AND TYPE-02="FLEET STUDY" AND OBJ-02="ECONOMY" AND DEG-02="7") OR
(FUEL-03="MEOH BLEND" AND TYPE-03="FLEET STUDY" AND OBJ-03="ECONOMY" AND DEG-03="7") OR
(FUEL-04="MEOH BLEND" AND TYPE-04="FLEET STUDY" AND OBJ-04="ECONOMY" AND DEG-04="7") OR
(FUEL-05="MEOH BLEND" AND TYPE-05="FLEET STUDY" AND OBJ-05="ECONOMY" AND DEG-05="7") OR
(FUEL-06="MEOH BLEND" AND TYPE-06="FLEET STUDY" AND OBJ-06="ECONOMY" AND DEG-06="7") OR
(FUEL-07="MEOH BLEND" AND TYPE-07="FLEET STUDY" AND OBJ-07="ECONOMY" AND DEG-07="7") OR
(FUEL-08="MEOH BLEND" AND TYPE-08="FLEET STUDY" AND OBJ-08="ECONOMY" AND DEG-08="7")))
```

```
((FUEL-01="MEOH BLEND" AND TYPE-01="FLEET STUDY" AND OBJ-01="ECONOMY" AND DEG-01="9") OR
(FUEL-02="MEOH BLEND" AND TYPE-02="FLEET STUDY" AND OBJ-02="ECONOMY" AND DEG-02="9") OR
(FUEL-03="MEOH BLEND" AND TYPE-03="FLEET STUDY" AND OBJ-03="ECONOMY" AND DEG-03="9") OR
(FUEL-04="MEOH BLEND" AND TYPE-04="FLEET STUDY" AND OBJ-04="ECONOMY" AND DEG-04="9") OR
(FUEL-05="MEOH BLEND" AND TYPE-05="FLEET STUDY" AND OBJ-05="ECONOMY" AND DEG-05="9") OR
(FUEL-06="MEOH BLEND" AND TYPE-06="FLEET STUDY" AND OBJ-06="ECONOMY" AND DEG-06="9") OR
(FUEL-07="MEOH BLEND" AND TYPE-07="FLEET STUDY" AND OBJ-07="ECONOMY" AND DEG-07="9") OR
(FUEL-08="MEOH BLEND" AND TYPE-08="FLEET STUDY" AND OBJ-08="ECONOMY" AND DEG-08="9"))))
[12 records found]
SORT BY SERIAL:
! END OF FIND/SORT COMMANDS
:PRINT-SELECTED-BIBLIOGRAPHIES;
```

ALTERNATE FUELS DATA BANK
BIBLIOGRAPHIES

SERIAL # 0077P

BERNHARDT, W.E., AND LEE, W., "ENGINE PERFORMANCE AND EXHAUST EMISSIONS CHARACTERISTICS OF A METHANOL FUEL AUTOMOBILE," CHAPTER IN FUTURE AUTOMOTIVE FUELS, COLUCCI, J.M., AND GALLOPOULOS, N.E., ED., PLENUM PRESS, NEW YORK, 1977, PP 214-234.
SYNOPSIS: A SERIES OF ENGINE TESTS AND VEHICLE TESTS WERE RUN TO EVALUATE THE POTENTIAL OF METHANOL AS AN AUTOMOTIVE FUEL. THE ENGINE TESTS (WITH SPECIAL INTAKE MANIFOLD AND CARBURETOR) SHOW INCREASED POWER, THERMAL EFFICIENCY, REDUCED NOx EMISSIONS AND INCREASED ALDEHYDE EMISSIONS, WITH NEAT METHANOL COMPARED TO GASOLINE. THE COMPRESSION RATIO OF THE ENGINE WAS INCREASED FROM 9.7 TO 14 TO 1. THE RESULTS SHOWED INCREASED EFFICIENCY BUT ALSO REDUCED NOx AND ALDEHYDE EMISSIONS.

SERIAL # 1004P

RICHARDSON, D., "EXHAUST EMISSIONS AND FUEL ECONOMY FROM VEHICLES USING ALCOHOL/GASOLINE BLENDS UNDER HIGH ALTITUDE CONDITIONS," U.S. EPA, OCT. 1978, 25 PP.
SYNOPSIS: THIS REPORT PRESENTS THE RESULTS OF CHASSIS DYNAMOMETER TESTS OF 10 CURRENT PRODUCTION AUTOMOBILES OPERATING ON GASOLINE AND ALCOHOL/GASOLINE BLENDS. THE TEST FUELS INCLUDED INDOLENE, 10% ETHANOL/INDOLENE BLEND, 20% ETHANOL/INDOLENE BLEND, AND 10% METHANOL/INDOLENE BLEND. EVAPORATIVE EMISSIONS TESTS SHOWED INCREASED EVAPORATIVE LOSSES DURING THE HOT-SOAK PORTION OF THE TEST.

```
BYE
/@ [EOF]
/BYE
/
HAVE A GOOD MORNING
20-APR-79 10:16 TT25: LOGGED OFF
/
  TOTAL TIME : :  1644 MINUTES
CONNECT TIME : :     5 MINUTES
   I/O COUNT : : 1482
    CPU TIME : : 13S-43T
 MEM. DEMAND : : 499E+3 KW*TICKS
```

APPENDIX II
(cont'd)
SAMPLE SEARCH OF CURRENT RESEARCH FILE

The information retrieval process for the current research file proceeds in a manner very similar to that for the publications file. The retrieval program for the current research file is called by typing "@BILDIT" after the user has established the link and has properly logged onto the system. As is the case with the publications file, the user is asked to code his/her request using the keyword codes outlined in APPENDIX I.

For instance, if the user were interested in knowing of projects that were developing information on fuel economy of vehicles operating on methanol/gasoline blends, the coding process would proceed as follows:

Category a: Type of Fuel - Methanol/gasoline blends implies MEOH BLEND.

Category b: Type of Investigation - Vehicles inplies FLEET STUDY.

Category c: Objective of Investigation - Fuel economy implies ECONOMY.

Category d: Performing Organization - Unspecified.

Category e: Principal Investigators - Unspecified.

Category f: Sponsoring Organization - Unspecified.

Category g: Level of Funding - Unspecified.

[If the request had been more complicated, this keyword statement could have been linked with other keyword statements (the program allows for a maximum of three keyword statements per search) using the Boolean logic expressions "and" or "or". Linking the keyword statements with "and" produces an intersection of the two sets of records satisfying the requirements of the first keyword statement and the second keyword statement (using "or" would produce a union of the two sets of records).]

The retrieval request and resulting information from the search would appear on the remote terminal as follows:

/HELLO XXXXXX (use the assigned user identification code in place of "XXXXXX")
PASSWORD:_____

RSX-11M BL22 MULTI-USER SYSTEM

GOOD MORNING
20-APR-79 10:11 LOGGED ON TERMINAL TT25:

/@BILDIT
/PIP RETRIEVE.CMD/PU
/!

```
/*RETRIEVE PUBLICATIONS [1] OR ONGOING RESEARCH [2]? [0 R:1-2]: 2
/RUN [225,1]GENRES
Enter TYPE OF FUEL:  MEOH BLEND
Enter TYPE OF INVESTIGATION:  FLEET STUDY
Enter OBJECTIVE OF INVESTIGATION:  ECONOMY
Enter PERFORMING ORGANIZATION:
Enter PRINCIPAL INVESTIGATOR:                    (if category unspecified, depress
Enter SPONSORING ORGANIZATION:                    the carriage return key)
Enter LEVEL OF FUNDING:
     ENTER LOGICAL CONNECTOR FOR NEXT LINE.
     ENTER BLANK IF FINISHED.

DO YOU WANT THE DESCRIPTIONS PRINTED
IMMEDIATELY AFTER THEY ARE FOUND?  [Y/N]:Y
/!


/DTR @RETRIEVE
FINISH;READY ONGOING-RESEARCH-DATA
FIND ONGOING-RESEARCH-DATA WITH SERIAL NE "999R" AND ((
((FUEL-01="MEOH BLEND" AND PURP-01="FLEET STUDY" AND OBJ-01="ECONOMY") OR
(FUEL-02="MEOH BLEND" AND PURP-02="FLEET STUDY" AND OBJ-02="ECONOMY") OR
(Fuel-03="MEOH BLEND" AND PURP-03="FLEET STUDY" AND OBJ-03="ECONOMY"))))
[2 records found]
SORT BY SERIAL;
! END OF FIND/SORT COMMANDS
:PRINT-SELECTED-RESRCH-OBJCTVS;


                    ALTERNATE FUELS DATA BANK
                   ONGOING RESEARCH ACTIVITIES

SERIAL #     004R

PROJECT TITLE:  METHANOL BLEND FLEET STUDY

RESEARCH ORGANIZATION:  FUEL/ENGINE SYSTEMS RESEARCH BETC, DOE

PRINCIPAL INVESTIGATORS:  R. W. HURN

SUPPORTING ORGANIZATION:  DOE TRANSPORTATION ENERGY CONSERVATION

CONTRACT NUMBER:
PERIOD OF RESEARCH FROM:  7/77
                      TO:  PRESENT

LEVEL OF FUNDING:  100k

PROJECT OBJECTIVE

IN AN EFFORT TO EVALUATE THE EFFECTS OF USING ALCOHOL/GASO-
LINE BLENDS IN CURRENT PRODUCTION AUTOMOBILES, THE FUEL/EN-
                                 |
                                 |
                                 |
                                 |
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

SERIAL # 005R

PROJECT TITLE: METHANOL/GASOLINE BLEND DEMONSTRATION

RESEARCH ORGANIZATION: DEPARTMENT OF MECHANICAL ENGINEERING, UNIVERSITY OF SANTA CLARA

PRINCIPAL INVESTIGATORS: RICHARD PEFLEY (JIM MCDONALD)

SUPPORTING ORGANIZATION: CALIFORNIA STATE ENERGY COMMISSION

CONTRACT NUMBER: UNKNOWN
PERIOD OF RESEARCH FROM: 2/24/78
TO: PRESENT

LEVEL OF FUNDING: UNKNOWN

PROJECTIVE OBJECTIVE

THE MECHANICAL ENGINEERING DEPARTMENT OF SANTA CLARA UNIVERSITY IS CONDUCTING A FLEET STUDY INVOLVING FOUR HONDA

: PRINT-SELECTED-RESRCH-STATUS;

****

Serial # 004R

PROJECT STATUS

FOUR 1977 MODEL YEAR VEHICLES WERE INVOLVED IN THE FLEET STUDY FROM ITS BEGINNING IN JULY 1977. EACH OF THESE VEHI-

SERIAL # 005R

PROJECT STATUS

THE FOUR VEHICLES HAVE BEEN PURCHASED AND BASELINE TESTING OF THE FOUR VEHICLES HAS BEEN COMPLETED. THE SANTA CLARA

THAT THE THREE METHANOL/GASOLINE VEHICLES WILL BE OPERATING ON A 10% METHANOL BLEND. THE CYCLE WILL REPEAT AGAIN TO ALLOW THE EVALUATION OF A 15% METHANOL BLEND.

```
BYE
/@ [EOF]
/BYE
/
HAVE A GOOD MORNING
20-APR-79 10:15 TT25: LOGGED OFF
/
  TOTAL TIME : :  1649 MINUTES
CONNECT TIME : :     5 MINUTES
   I/O COUNT : : 704
    CPU TIME : : 4S-56T
 MEM. DEMAND : : 414E+3 KW*TICKS
```

APPENDIX II
(cont'd)
SAMPLE SEARCH OF TOPICS FILE

The information retrieval process for the topics file differs considerably from that of the publications and current research files. The user will establish a link with the computer in Bartlesville, sign on using the proper user identification code, and give the password. The user will then call the topic file program by typing in "@PRTSELTOP". The response to the user will be in the form of an index of topics, giving the topic serial number and corresponding descriptive title. The user will then type in the serial of the desired topic. The response from the computer will be to produce a copy of text associated with the selected topic.

A sample search of the topics file from a remote terminal would appear as follows: (underscored portions denote the user's response)

HELLO XXXXXX (use the assigned user identification code in place of "XXXXXX")
PASSWORD:_____

RSX-11 BL22 MULTI-USER SYSTEM

GOOD MORNING
20-APR-79 10:11 LOGGED ON TERMINAL TT25:

/@PRTSELTOP
/:PRINT-SELECTED TOPICS;

TOPIC SERIAL	TOPIC TITLE
01T	EFFECTS OF FUEL NITROGEN LEVEL ON NOX EMISSIONS FROM STRATIFIED CHARGE ENGINES
02T	SOURCES OF 200 PROOF ETHYL ALCOHOL
03T	VALUE OF ETHANOL AND METHANOL AS OCTANE IMPROVING ADDITIVES
99T	OPEN

/IF YOU WISH TO REVIEW A TOPIC PLEASE SUPPLY THE APPROPRIATE SERIAL NUMBER OTHERWISE SPACE ONCE AND DEPRESS THE CARRIAGE RETURN KEY. 01T

ALTERNATIVE FUELS DATA BANK
TOPICS

SERIAL 01T EFFECTS OF FUEL NITROGEN LEVEL ON NOX EMISSIONS FROM STRATIFIED CHARGE ENGINES

EXPERIMENTS WERE CONDUCTED WITH A DIRECT-INJECTION STRATIFIED-CHARGE ENGINE USING FUELS OF DIFFERENT NITROGEN LEVELS. THE ENGINE,

THE RESULTS OF ESTIMATES ARE SHOWN IN FIGURE 2. THE DIFFERENCE BETWEEN NOX EMISSION RATES FOR THE TWO FUELS IS ON THE ORDER OF 3 GRAMS/LB. FUEL. THIS CORRESPONDS TO A 20 PCT EFFICIENCY FOR CONVERSION OF FUEL-BOND NITROGEN TO NOX.

IF YOU WISH TO REVIEW ANOTHER TOPIC PLEASE SUPPLY THE APPROPRIATE SERIAL NUMBER OTHERWISE SPACE ONCE AND DEPRESS THE CARRIAGE RETURN KEY.

```
BYE
/@ [EOF]
/BYE
/
HAVE A GOOD MORNING
20-APR-79 10:15 TT25:  LOGGED OFF
/
  TOTAL TIME ::   1649 MINUTES
CONNECT TIME ::      5 MINUTES
   I/O COUNT ::  704
    CPU TIME ::  4S-56T
 MEM. DEMAND ::  414E+3 KW*TICKS
```

SUBJ
Comp
CBM

The American Association of Petroleum Geologists Bulle
V. 53, No. 11 (November, 1969), P. 2324-2340, 16 Fig

Contouring by Machine: A User's Guide[1]

RICHARD F. WALTERS[2]
Davis, California 9561

Abstract Several general purpose contouring programs are available. Their effective use requires a knowledge of ways in which the programs operate and the types of applications most suited to machine-contouring techniques.

Machine contouring is not applicable to most one-pass jobs involving the use of considerable scientific judgment or combinations of different types of data; e.g., gradient and point values derived from data that have not been put in machine-readable form. By contrast, machine contouring is a valuable technique in analyzing large masses of data or in routine updating, in analysis of data that do not require a scientific bias for interpretation, in making a suite of maps that is to be made internally consistent; and in further analysis of resulting surfaces.

Most general purpose contour programs follow similar steps in the generation of a map. These steps include (1) specification of map size and accuracy desired, (2) entering raw data to be contoured, (3) generation of a grid from irregularly spaced points, (4) creation of contour lines, and (5) special annotation and other additional features in certain programs. Grid generation is the most important component of the program and there are several techniques for generating grids.

Familiarity with these principles will enable the user to interact more effectively with the program at his disposal, and to create, store, combine, and analyze the surfaces generated in a way which will enhance his scientific investigation.

Introduction

Contours have been used by scientists for years to represent geographic, biologic, political, and other data. Geologists in particular have developed interpretive contouring into an important and valuable predicting tool. In petroleum geology, contour maps are used to depict structure, isopachs, gravity data, and many geochemical and other geological parameters. Much of a geologist's time is spent collecting these data and maintaining a current suite of maps.

The advent of computers has led to attempts to duplicate interpretive contouring techniques by machine methods. Considerable progress has been made in developing an understanding of the human logic of contouring; several computer programs have been written which are capable of making "reasonable" contour maps of data points with an even distribution. An extensive review of the literature describing the development of such programs can be found in Harbaugh and Merriam (1968). Some of the available programs are described in McIntyre *et al.* (1968), Assiter (1967), and California Computer Products, Inc. (1968). Key discussions of basic principles are contained in IBM (1965), Tobler (1965), and McIntyre *et al.* (1968).

To be effective, such a program must allow the user to make judgments during the preparation or modification of the contoured map to enable the specialist to introduce his background into generation of the final contours. The process of interaction between man and machine requires a general purpose contour program and a trained user to derive optimum results. This paper provides the potential user with an understanding of the basic steps incorporated in several contour-programs, in hopes that knowledge of these principles will increase the effectiveness of such programs.

Applications

Before reviewing contour program design, one should first consider what situations best lend themselves to automatic contouring.

Cases Not Lending Themselves to Machine Contouring

Many "one shot" interpretations are not suitable for machine preparation, partly because the problems of initial data preparation in machine form may be excessive for a single interpretation. A second example arises if the

[1] Manuscript received, October 26, 1968; revised and accepted, May 5, 1969.

[2] School of Medicine, University of California at Davis.

The writer thanks the designers of several general purpose programs for their assistance in discussing the techniques used in each program. These include Donald B. McIntyre, Edward Assiter, George Batten, and representatives of California Computer Products, Inc., particularly David Balkenbush. Permission to use illustrations generated by these programs was generously furnished.

The manuscript was reviewed critically by J. W. Harbaugh and R. P. Walters. Typing was done by Sylvia King and assistance in preparation of illustrations was provided by W. E. Renner of the School of Medicine, University of California at Davis, and California Computer Products, Inc.

Computer time was supplied by the Computer Center at the University of California at Davis, operating under support of Public Health Service Grant FR-00009-06.

basic data control points are so sparse that realistic mechanical interpretation is impossible, and the investigator's judgment is essential to produce a meaningful interpretation.

Frequently, more information is available than the location and values of each control point. This knowledge, both subjective and objective, appears in the researcher's judgment decisions as he contours such maps. It would be difficult to introduce all such information into a program; and the labor involved in creating a machine interpretation might not be justified.

Often factual information exists in varied forms which complicate its correlation into map form. For example, geologic trends, gravity or aerial photographic gradients, and magnetic data all require skilled interpretation in the contouring process.

Applications of Single-Purpose Machine Contouring

There are certain single purpose jobs for which machine contouring might well be justified.

First Approximations of Voluminous Data

Frequently, the researcher is confronted with a large mass of data to which he can add little supplemental information. Only a few of these data sets may prove to be revealing, and it is good practice to examine a preliminary machine-contoured version to see whether the results warrant additional investigation. Geochemical data, for example, can best be analyzed in map form. Commonly, however, several geochemical maps must be reviewed to find one with significant parameters. Manual preparation of many such maps would be a waste of the researcher's time. Similarly, a few lithologic mappable relations usually suffice to depict the critical factors that the investigator wishes to make in stating his conclusions.

Other situations where machine contouring might be useful concern tightly controlled surfaces in which the data points are too dense to permit latitude in human interpretation. In such cases the process becomes one that is largely "mechanical"; hence, it should be left to a machine.

A third situation calling for machine contouring arises when the user wishes to assure himself that no conversion errors exist in the data. Automatic contouring will provide instantly a visual check for transcription errors, giving assurance of their absence in areas showing expected contour values.

Contouring for Subsequent Analysis

The most important potential applications for machine contouring involve the subsequent analysis of the completed graph. Some typical examples of potentially fruitful application follow.

Updating.—When new information is obtained that affects part of an existing map, it is convenient to correct only the part of the map directly affected by the new control points; the map can then be redrawn by machine for future reference. Updating oil-field maps during active development programs is so time-consuming that manual map suites are rarely kept current.

Consistency of a suite of maps.—Frequently, when three or more interrelated contour maps of the same area are needed for geologic interpretation, two maps determine a third surface. A deep structure map (Fig. 1c), for example, can be drawn by adding the appropriate isopach (Fig. 1b) to a shallow structural surface (Fig. 1a). Manual "cross contouring" of such maps is tedious, particularly if updating is required, and deep control is less reliable for structure than stratigraphy (Fig. 1d). Hence, the most logical approach would be for the user to create the independent surfaces, and then allow the contour program to generate for him those values which are dependent.

Analysis.—Contour maps are created commonly as a vehicle for future analysis, such as isolating residuals by removing regional gradients (Fig. 2a, b, c) or calculating volumetric relations (Pruner, 1966). In such applications, an automatic contour map can facilitate analysis. The full potential of this approach cannot be appreciated until the user has had experience with the method. Once a user becomes accustomed to this approach, however, he is likely to become increasingly dependent on the computer as a major support for his research.

Basic Requirements

A general purpose program should contain several elements enabling the user to adapt the program to a variety of situations. Although programs differ in the features provided, some of the options listed below should be contained in any general purpose package.

Acceptance of Irregularly Spaced Data

Many programs accept regularly spaced data points for contour generation. Usually, however, geologic data observations are irregularly spaced, and judgment is required to extend

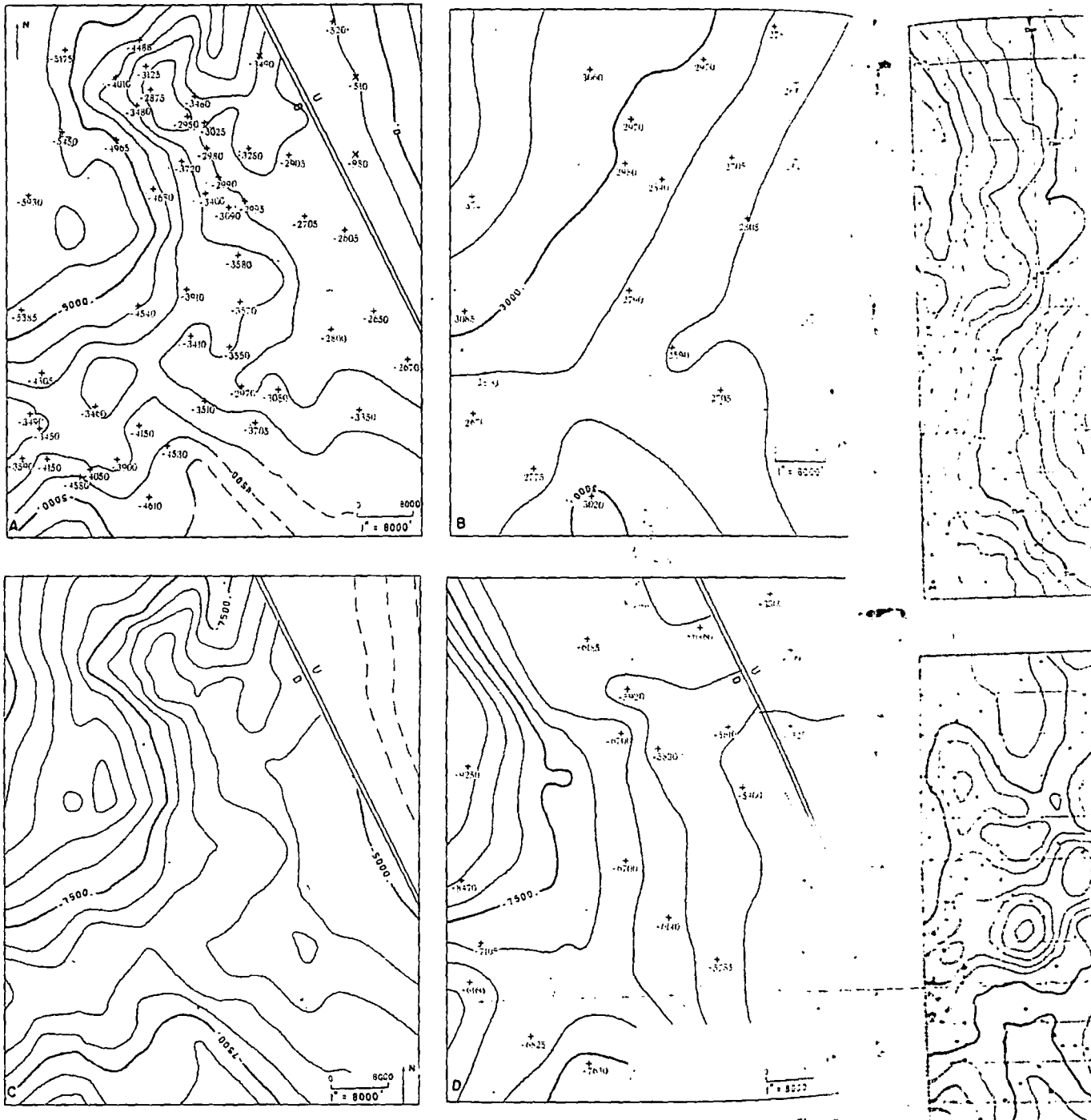


FIG. 1.—Use of machine-generated maps to create deep structure map (hypothetical example). **A.** Sh— structure map. Structure has been affected by post-depositional faulting. **B.** Isopach, showing limited ... and generally smooth northwesterly thickening. **C.** Machine-generated deep structure map, made by combi... maps shown in **A** and **B**. **D.** Deep structure contoured only from available structural points at base of inter...

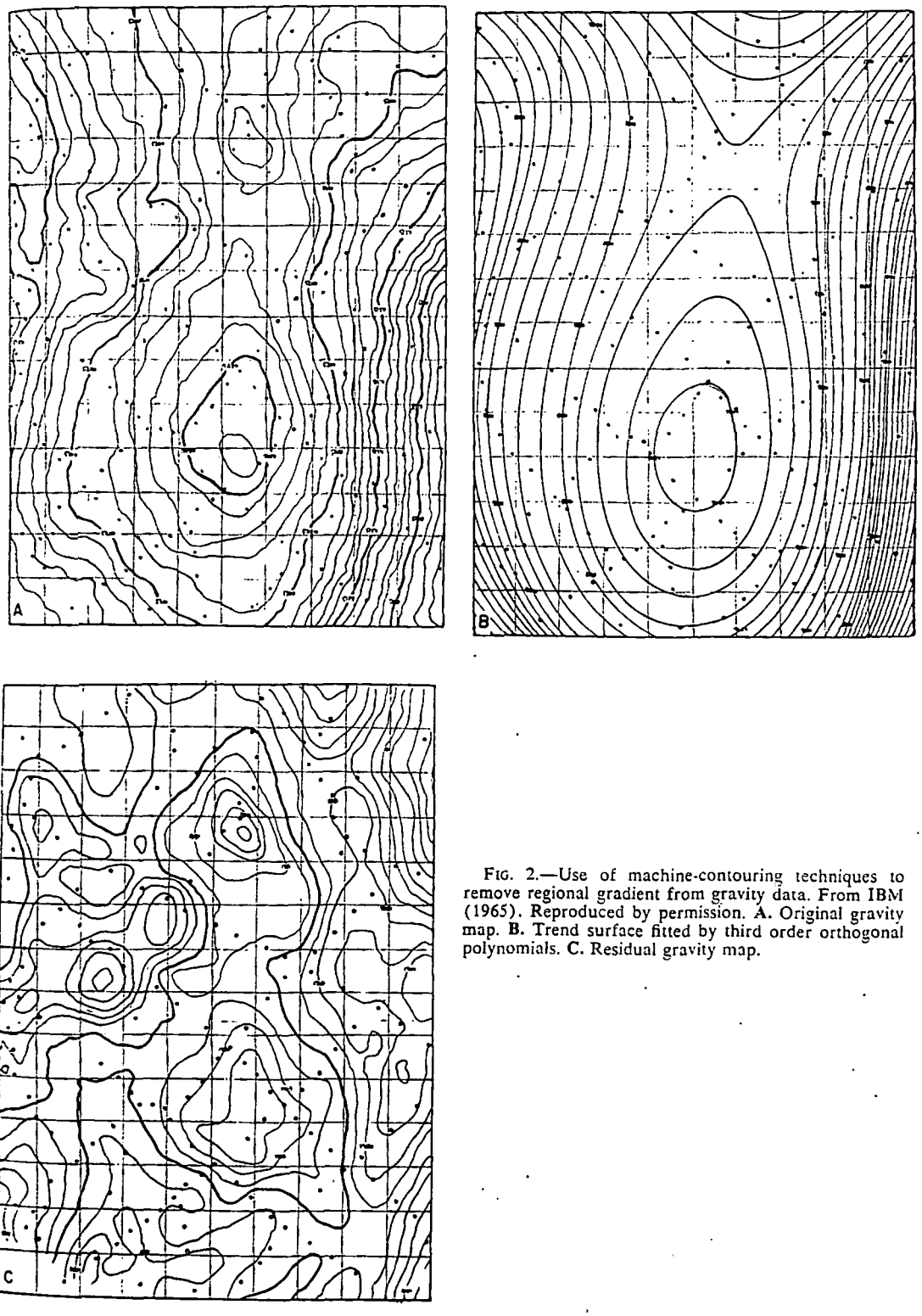


FIG. 2.—Use of machine-contouring techniques to remove regional gradient from gravity data. From IBM (1965). Reproduced by permission. **A.** Original gravity map. **B.** Trend surface fitted by third order orthogonal polynomials. **C.** Residual gravity map.

contours between the points. Irregular data spacing problems include: 1) irregular boundaries to the area covered by data; 2) significant gaps in control points within the general area of coverage; and 3) unusual clusters of data points in an otherwise less densely controlled configuration. These spacing problems are partially resolved in all general purpose programs. Certain distribution problems require additional special programming effort to generate reasonable maps.

Simplicity of Utilization

A fundamental assumption made by most contour program designers is that the average user is not a specialist in computers. For such a user, the control cards should be simple, data input requirements should be straightforward, and the program should include routine checks, so that major errors in sequencing, keypunching, or specifications will be automatically detected and brought to the user's attention, and the computer run terminated or adjusted accordingly.

Simplicity is helpful in overcoming the communications problem between man and machine, and this barrier is reduced as a user becomes familiar with the program. Many programs will offer variation in complexity, and the seasoned user can specify complex options that give him increased flexibility, whereas the novice can use the same program without being aware of the options.

Basic (Minimum) Flexibility

The user should have considerable latitude in adapting the general program to his particular needs. Among the choices is the description of the map—its vertical and horizontal dimensions, its scale, and, if need be, the position where it should be spliced. It is also useful to be able to specify the degree of precision required. This option enables a user to make quick estimates for preliminary examination or detailed analyses where economic or other factors justify additional computational costs. A related option in contour programs requires selection of appropriate contour spacing. Many programs provide for special annotation, such as the ability to post original data points, to indicate special scale or other reference marks, and to add other lines or symbols that contribute to the clarity of the finished map.

Economy

The widespread operational use of machine contouring will occur only if programs can be run at a reasonable cost. Computer costs for contour generation are affected by the computer, the program, and the plotter used for the operation. The figures produced for this report, for example, were made on an IBM 7044 computer. This machine rents in commercial operations for approximately $180.00 per hour. Figures 3–5 required approximately 2.5, 3, and 5 minutes, respectively, to generate a tape for offline plotting, at a cost ranging from $7.50 to $12.00 for machine time. These times are representative for many "second generation" computers. On most modern high-speed digital computers, however, the time for grid generation would be reduced to seconds, and the cost cut substantially despite the larger hourly rental for such machines. The time taken to write the offline plotter magnetic tape would not be reduced significantly by introduction of a high-speed computer because the speed at which a tape is written depends on the speed of the tape drive.

The plotter used in Figures 3–6 was a Calcomp 563, driven by a 750 magnetic tape drive. Plotting Figure 3 required approximately 8 minutes at a cost of $4.00 assuming plotter rental of $30.00 per hour; the total cost of this map was thus $11.50 for both generation and plotting. Figures 4 and 5 required only 6 minutes ($3.00) for plotting because data points were not spotted. Newer plotting devices and tape drives have been manufactured which greatly reduce the time required to create a plot tape.

In summary, costs for an average map might run between $5.00 and $25.00 run on modern equipment at a commercial service bureau. These costs could be reduced by inhouse operations or by stacking several maps in the same run.

"Smoothness" of Contours

Acceptance of automatic contours also depends in part on the production of natural smooth lines. Early machine contour programs tended to be angular and were unacceptable to the average user despite their overall accuracy. Experience has shown that in all but the roughest first appproximations it is essential that the output be pleasing to the specialist's eye. Initial user acceptance of smooth lines is often greater, even though a map with angular lines may agree more closely with the original data points. Generation of smooth lines can be achieved by straightforward interpolative techniques and should be available in any program.

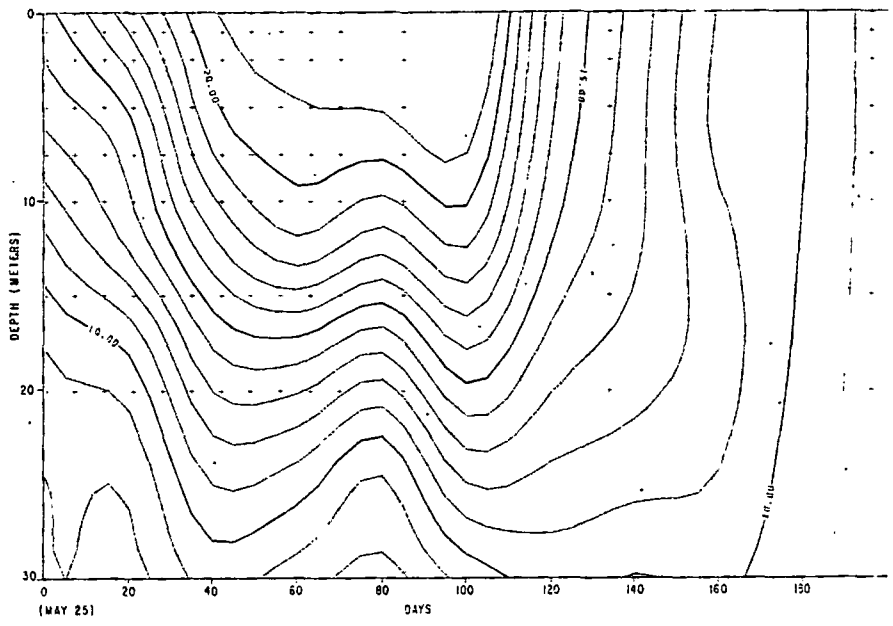


FIG. 3.—Temperature variations with time and depth, Lake Maggiore, Italy. Information obtained from Goldman *et al.* (1968). Contour grid 11 × 7.

SEQUENCE OF PROGRAM OPERATION

Basic organization of contouring routines is dictated by operation logic and machine efficiency. An understanding of this sequence is useful in optimal utilization of program options. The basic steps include (1) map specification, (2) data input, (3) generation of a grid, (4) contour-line generation, (5) annotation, and other special optional operations. A

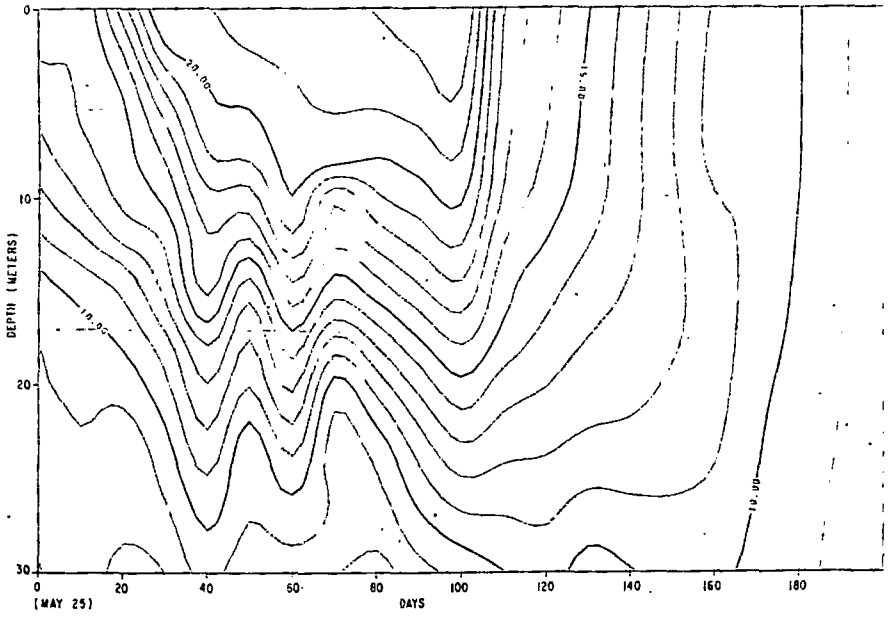


FIG. 4.—Same map as Figure 3, contoured on 21 × 16 grid.

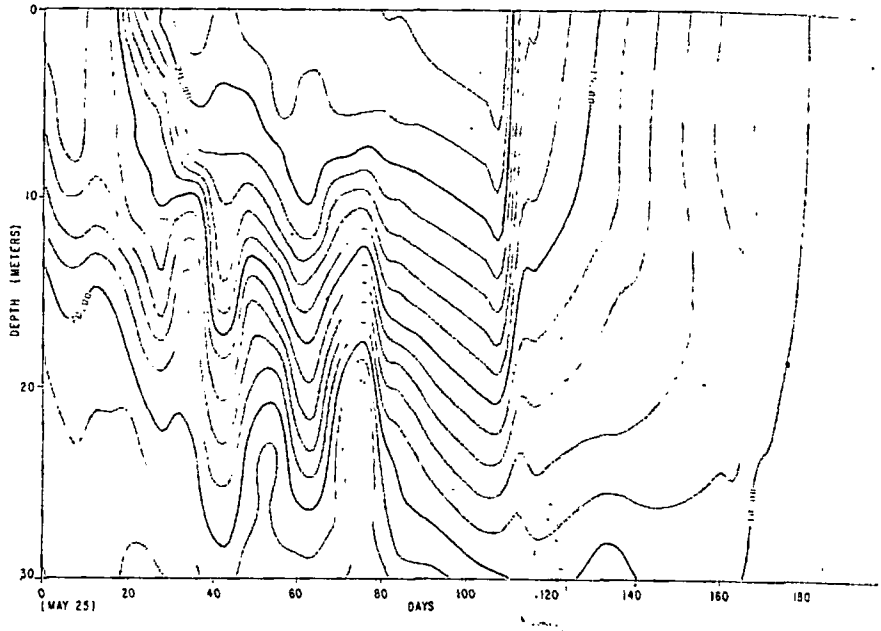


FIG. 5.—Same map as Figure 3, contoured on 41 × 61 grid.

few programs which allow several separate jobs to be run together have an additional step which recycles for the next job.

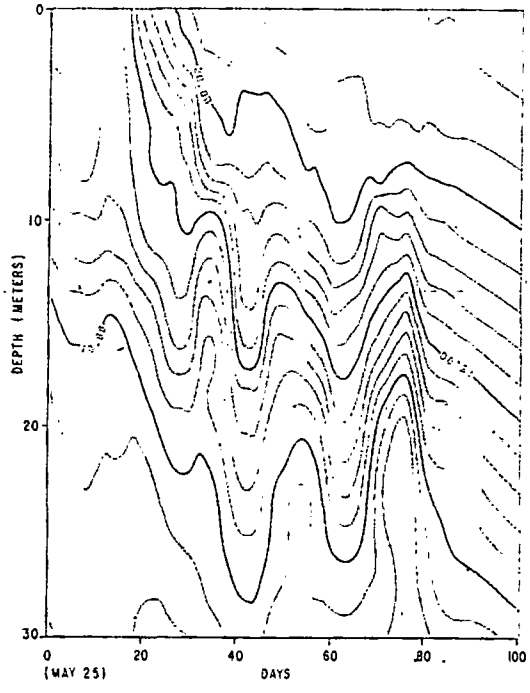


FIG. 6.—Same map as Figure 3, contoured on 61 × 101 grid.

Map Specifications

Job Title Information

Most programs begin the control sequence with identifying information, such as the name of area, the date, or other title information. These facts are usually copied directly from the control card to the output.

Map Dimensions

Just as a draftsman selects his paper for the illustration, so the computer program must know the map dimensions and scale in terms of the original coordinate units. In some cases the vertical and horizontal scales may be different. Figure 3, for example, depicts variations of water temperature with respect to time and depth. It is usually necessary, therefore, to specify scale values for both X and Y axes. The values required to provide this information are the minimum, maximum, and scale values for both the X (horizontal) and Y (vertical) dimensions. The program uses these parameters to allocate memory for certain tasks and to set the scales for plotting.

Grid Specification

Automatic contour generation can be achieved by several techniques, such as linear interpolation, polynomial surface fitting, harmonic analysis, *etc.* It is necessary first to generate a series of regularly spaced values

ie function to be contoured. This "grid" provides a consistent framework for the generation of contour lines. The use of gridding techniques roughly approximates the interpretive approach used by a specialist contouring the same data points, but human interpretive techniques are more flexible.

It is necessary in most programs to allocate computer memory space for the retention of the gridded values. For example, a map 20 in. square with grid values every ¼ in. would require 6,651 words of memory (81 × 81) for the gridded values. A ½-in. grid for the same map would take up only 1,681 locations (41×41). In practice, the limitations of available memory can be overcome by subdividing the area into separate sections.

An example of the effect of grid spacing is shown in Figures 3–6, in which temperature data collected at different depths from Lake Maggiore, Italy, during a period of several months were used (Goldman *et al.*, 1968). Water temperatures (°C) are contoured to show changes with depths (Y axis) and time (X axis). Figure 3 shows the location of the original data points and the contours generated by using an 11×7 grid. Figures 4, 5, and 6 depict the same values contoured on grid spacings of 21×16, 21×61, and 61×101 respectively. The contour details increase markedly between Figures 3 and 4, but remain essentially constant in Figures 5 and 6. Selection of a grid is a matter of individual need, and each problem must be analyzed separately for grid selection. Grid generation times can range from a few seconds of computer time to 30 minutes or more, depending on the computer, the program, and the grid spacing selected.

The opportunity to specify different grid spacing is allowed in many programs. This provision can be used to accommodate different scales in the X and Y directions (Figs. 3–6) or to achieve special effects whose description lies beyond the scope of this report.

Special Instructions

Some contour programs offer special conveniences to the user, such as (1) allowing for oversized maps, (2) relocating, rotating, or skewing the plotted output on the page (Fig. 7), and (3) constructing a stereoscopic pair of illustrations for three-dimensional viewing. It is necessary to specify these options at the inception of the job. If these options are not used, the program should follow certain pre-set standard specifications by default, unless an overriding option is specified.

Data Definition

Irregularly Spaced Data

Three general parameters are required as minimum input for each data point: the x coordinate, the y coordinate, and the "z" value to be contoured. Some programs also allow for certain qualifying information, such as the dip at a single data point. Some observations may be questionable and should be weighted less heavily than others. Certain programs do accept weighting and/or slope qualifications for individual data points.

Raw data must be converted to machine-readable form. If the data are new, it may be necessary to keypunch each point. Retrieval of one data set from an information system requires special programs which may include specific screening options. It is usually best to preprocess such data by means of a separate pro-

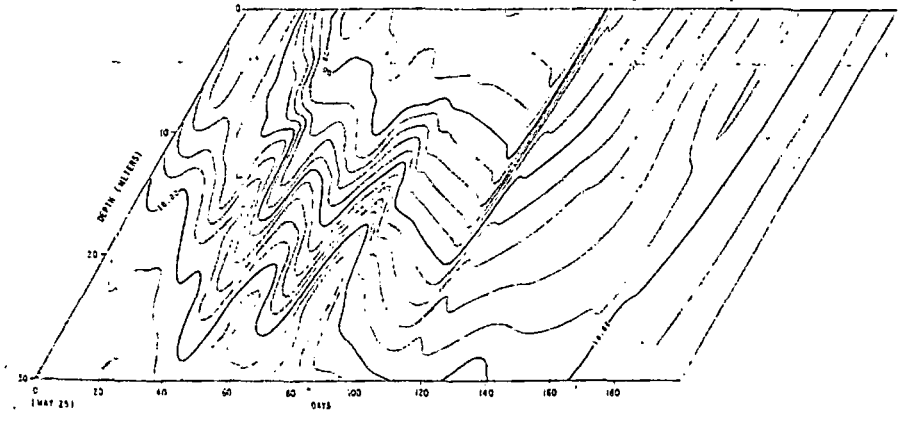


FIG. 7.—Isometric projection of Figure 5.

gram; in certain applications, however, the contour program itself may be used as a screening program.

Most programs permit latitude in the position of the x, y, and z coordinates on data cards. Each job has its own format, specified by a control card: all cards in that data set must conform to that format.

Routine use of data obtained from an information system is more efficient if the contour program can accept data tape or disk format to avoid punching new cards for data already in machine-readable form. One way to utilize large data files is to generate a data tape processable by the contour program.

Efficient program execution is improved by geographic sorting of data. Although programs sort data internally for calculation purposes, the points are plotted as they appear in the input; hence pre-sorting will improve program efficiency if data points are posted.

Gridded Data

Although geologic observations are generally irregularly distributed, it is nonetheless important to be able to introduce regularly spaced information in a contour program. This option is used to contour data collected on a grid, to modify or update portions of the grid, and to combine or transform previously gridded data. Ideally, the program should accept either gridded or irregular data, depending on the user's needs. The program should also allow the user to override certain generated grid values, so that he can modify the initial map, storing the result for future reference.

Grid Generation

It is important to understand in a general way the manner in which grid values are generated and the relative merits of a few ways that can be used to generate such a grid. Because of the importance of this step, some programs even permit user option in selecting a grid-generating technique. The merits of any specific gridding technique depend on the use(s) intended for the program; a general purpose program should, however, attempt to satisfy as many users as possible, either by providing user options or by adopting a gridding approach which satisfies most users.

Alternative Approaches to Contour Generation

Automatic contouring is dependent on creation of a grid; that process should develop a set of lines which are similar to contours drawn by a specialist. The formal steps leading to such grid generation could be designed to imitate as exactly as possible the human approach to contouring, or they could take a completely different approach. Most specialists start drawing contours in areas of dense control, then join those separate areas through a process of adjustment in the sparsely controlled map regions. The process usually involves erasures of intervening areas to fit one set of lines to another. Analysis of the overall pattern commonly results in further erasures and adjustments to emphasize trends and other features not originally detected. Some researchers begin the process by making a series of straight-line interpolations between data points, so as to create an "objective" interpretation which they can subsequently adjust according to their own judgment.

To imitate the preceding processes by machine, one should design a program that creates grid points in areas of control, then moves outward to uncontrolled grid values until the entire map surface has been calculated. Imitation of the "erasure" approach to fitting such areas is awkward by machine, requiring many repetitive instructions that can be anticipated and compressed. Creation of straight-line interpolations, although easy to calculate, results in irregular straight-line contour segments which are dependent on the original distribution of data points. Moreover, there is no way to generalize the scientific judgments involved in correcting such lines.

An alternative approach to generating contours is to create a grid without attempting to imitate the human approach to the problem. Examples of programs that tend toward each of these philosophies are illustrated below.

Examples of Contour Logic

Numerical approximation.—In this method used by one available program,[3] the data points (Fig. 8) are projected onto a grid (Fig. 9). The four grid points surrounding each data point are selected for first pass analysis. In order to obtain values for these grid points, a plane is passed through each data point. The plane is determined by searching for the nearest data point in each octant around the data point (Fig. 10). The plane is derived as a least squares function of these values, weighted

[3] ACI Contouring Package available through E. Assiter, President, Applications Consultants Inc., West Holcombe Boulevard, Houston, Texas 77[illegible]

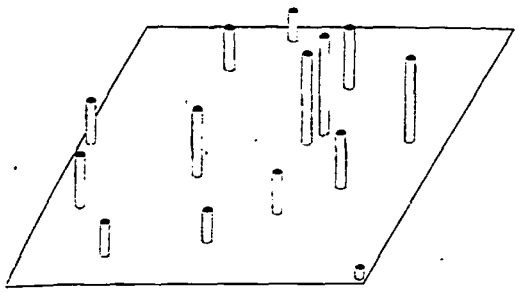


FIG. 8.—Data points to be used in generating contours. (Reproduced by permission of E. Assiter.)

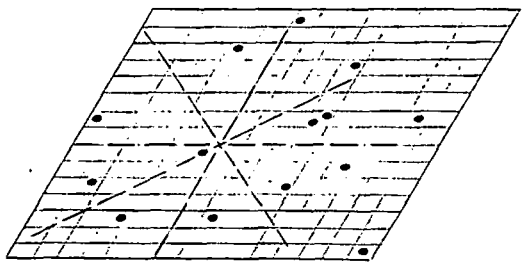


FIG. 10.—Search for nearest well in each octant around each grid intersection adjoining a data point. (Reproduced by permission of E. Assiter.)

cording to their distance from the data point. The plane must pass through the central data point (Fig. 11), and it must agree as closely as possible with the selected data points surrounding it. A weighting function for distance is introduced so that closer points will exert greater influence in the fitting process.

The intersection of this plane with the grid point determines its value and slope (Fig. 12). Similar calculations are made around each data point resulting in the determination of several grid values. The slopes of the planes at calculated grid points are then used to determine the grid values one grid spacing away from the original data points (Fig. 13). The original data points are not used in these or subsequent calculations. After the second set of grid values is made, a third group and a fourth group are calculated by use of slopes from the outermost calculated points until all the values have been determined.

The logic used in this method closely approximates manual methods in certain important aspects, such as working away from control points and using established dips to extrapolate new ones in areas of poor control. This approach represents one of the most successful techniques in contour programs. It does, however, have two potential limitations: (1) the searching required to generate the values is potentially cumbersome, requiring an octant search at each data point, keeping track of calculated and blank grid values, and adjusting for data clusters in single cells; and (2) because of the extrapolation technique, large gaps in con-

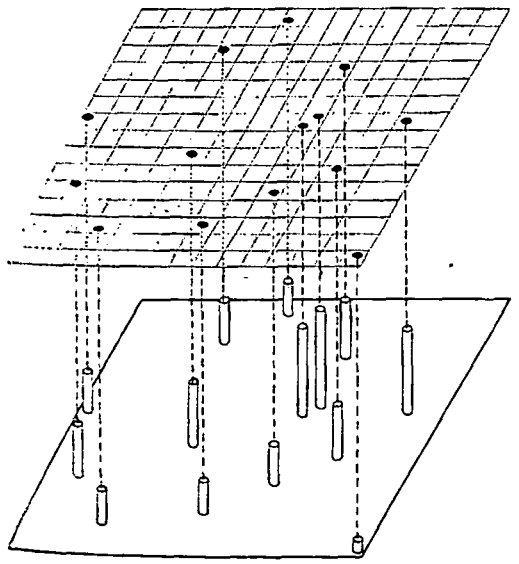


FIG. 9.—Data points projected onto orthogonal grid. (Reproduced by permission of E. Assiter.)

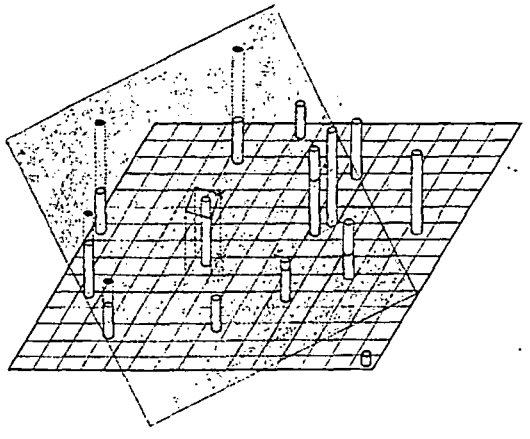


FIG. 11.—Creation of plane passing through nearest data point and representing weighted least squares fit of selected adjacent points. (Reproduced by permission of E. Assiter.)

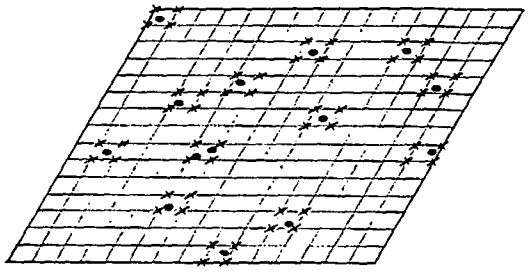


FIG. 12.—Determination of grid values adjacent to each data point. (Reproduced by permission of E. Assiter.)

trol sometimes result in gradients increasing sharply away from control. Because the program moves sequentially away from data points, the tendency to develop steep gradients is not strictly controlled.

A technique used in another program[4] begins by creating a plane through each data point (Fig. 11), using much the same approach as described above. One major difference is the location of points used: instead of requiring that the nearest control value in each octant be located, the program selects the 8 nearest points (the user can modify this number from 5 to 20) and weights them by distance, but does not check their distribution around the data point. This technique is safe in most circumstances; where data points are clustered or rectilinear, however, unnatural gradients may occur.

After a plane is passed through each data point by use of the weighting techniques described, the program then begins at one corner of the area to be mapped and proceeds sequentially to evaluate each grid point. This evaluation is accomplished (Fig. 14) by determining

[4] General Purpose Contour Program (GPCP), copyright Calcomp Inc.

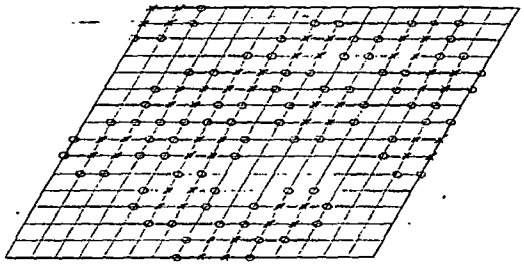


FIG. 13.—Extrapolation of surface to adjoining grid intersections. (Reproduced by permission of E. Assiter.)

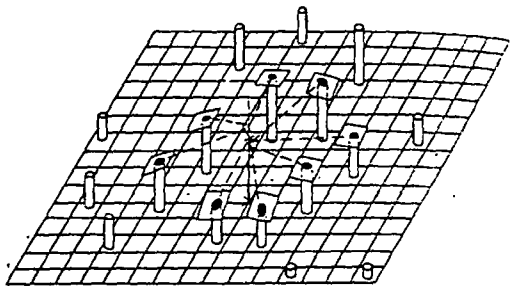


FIG. 14.—Grid generation function used in Californ... Computer Products, Inc., GPCP program. All grid values calculated from intersection of planes calculated for 8 nearest data points.

the intersection at that grid point of each plane of the 8 nearest data points (again, 5 to 20 points can be chosen as desired); the weights of the plane intersections are again a function of their distance, and a weighted average value is determined.

This technique is unlike manual approaches, the map is essentially evaluated from bottom to top, in that planes are calculated independently for each data point and then used to generate all grid values. However, this approach usually gives a satisfactory solution to the grid generated, particularly in areas of no control, because the extrapolation technique places a tighter restriction on areas of poor control by referring back to original data points and, in effect, extending a linear extrapolation rather than one whose slope might increase with distance.

In a slightly different grid generation technique utilized by another program (McIntyre *et al.*, 1968) the data point distribution is first analyzed to determine the average spacing: the radius of a circle which would enclose 10 points as an overall average is determined. The program then proceeds to evaluate grid points sequentially. Each grid value is taken to be the center of a circle whose radius fits the description given above. If enough (10 or more) points are available, a quadratic surface is fitted through these points, and its intersection with the grid point is used as the value for that point. If less than 10 points are present, the radius is doubled, and, if 10 points are then found, the same procedure is used. If fewer than 10 points lie in this larger circle, the grid value is flagged and no contours are drawn in its immediate surroundings.

A further check is made on the reasonable-

ness of the calculated values: the data points utilized are evaluated for their range of values. If the calculated grid intersection lies more than 20 percent above or below this range, it is also ignored. This provision eliminates boundary condition problems which can arise in a quadratic fit of data points.

This program can be modified at its inception to adjust such parameters as the weighting for distance, the number of points required for a grid evaluation (this number must be more than 6), or the range limitation on automatic blanking.

One factor important to all these techniques is the distance weighting function. This function could be designed to produce an extremely smooth surface, if little difference is attached to increasing distance, or it could create a more complex configuration if heavy weighting is laid on the nearest points. The selection of a range of weighting functions is not offered in available contour programs but would greatly enhance their generality. Certain inherently "noisy" surfaces require emphasis on the local variations. In such cases a strong weighting for local control points should be used. In other situations, also involving "noisy" data, the analyst may wish instead to smooth out the local points and view regional changes, so that the weighting function should be minimized. Although such an option requires considerable user sophistication, it should be made available.

The preceding grid generation discussions, although somewhat detailed, contribute importantly to the understanding of contour programs. As a user becomes more familiar with machine contouring, he will wish to increase his knowledge of the way in which he can exercise his judgment to improve the machine grid interpolation.

Operations on Grid Values

Once a grid has been generated, the user may wish to intervene so as to modify or store this matrix of points. Several reasons for such user intervention may be noted.

Specific grid value alteration.—The user should be allowed to adjust certain areas overriding calculated grid values with points of his own choosing. This choice could be made by specifying, by row and column, the value or values which he wishes to change, and the new values to be inserted.

Storage and retrieval of entire grids.—The user may wish to store the grid which was generated for future reference, in either the card or tape format. Grids frequently contain several thousand points, hence punch card storage of a group of such values becomes unwieldy and a tape format is essential. The program should also have a retrieval option whereby a complete grid can be recalled. It is likely that the retrieval operation will become more commonplace as machine contouring gains acceptance. Hence, the storage and retrieval process should be facilitated by including on the tape additional parameters that should aid the user by providing shortcuts to control card preparation.

Combinations of arrays.—A major use of machine contouring lies in further utilization of the grid values generated. Two surfaces can be added or subtracted (Fig. 1), the volume between them calculated (*see* Pruner, 1966), or they could be combined in other ways. Grid manipulation is just beginning to be explored by users and by contour program designers. In the future, the provision to add and subtract arrays will probably be included in many programs.

Grid combination requires that the grids have the same dimensions and spacing, a restriction that facilitates a simple point by point processing. One program includes a volume calculation routine which takes into account an upper and lower surface, as well as other discontinuities on the boundaries. This routine is especially useful in petroleum reservoir calculations, where faults, tilted water tables, and unconformities often truncate the reservoir in complex geometric relationships.

Transformation of data.—Simple data transforms should be provided within, or allied to, the program. More complex, special purpose routines should be developed independently. Residual calculations (Fig. 2) are fairly commonplace, but computer adjustment of reservoir volume by a variable porosity factor probably would have to be specially designed.

Contour Generation

The preceding discussions have described steps preparatory to drawing contour lines. The program has read-in data points, calculated a grid, laid out the map, and perhaps posted data points. It is ready to start drawing the contours. The user now faces a new set of decisions.

Blanking

First, he will want to determine whether he really needs contours throughout the rectangle to be mapped, or whether certain areas should

be blanked because of lack of control or the need for local special treatment.

Specification of the blanking area can be accomplished in two ways. In some programs (*e.g.*, McIntyre *et al.*, 1968) blanking is automatic a given distance away from the original control points.

Other programs allow the user to specify each area to be contoured by giving its boundaries. In Figure 1, for example, the combination of two fault segments on one map was accomplished by specifying boundaries for each segment. The addition of dotted lines in Figure 1a was likewise accomplished by first blanking the southeast corner of the map and then drawing only that area with the use of dashed instead of solid lines. Figure 15 shows a similar situation in which data from four fault blocks, where contoured together, give a distorted picture (Fig. 15a). Contoured independently, however, as shown in Figures 15b and 15c, the individual configuration of each fault block is revealed. This blanking method is somewhat more complex, but it is also considerably more flexible than automatic blanking techniques.

Spacing

Contour spacing is an essential user specification. A constant spacing factor between lines is usually sufficient; however, some maps may require other types of spacing for contour values. In data that show a lognormal distribution, for example, it is desirable to change the spacing so as to emphasize this characteristic. Alternatively, the user may wish to highlight certain data ranges and generalize others. Certain programs skip lines when they come too close together. Hence, a general purpose program should have a certain amount of flexibility in contour spacing. Incorporation of such options makes automatic contouring adaptable to a wide variety of uses.

Smoothness

Contouring involves an attempt to present isolines which are accurate and also pleasing to the eye. Angular lines are not accepted in manual interpretations, nor are they necessary in automatic routines. Analog plotters may be used to provide smooth contours (McIntyre *et al.*, 1968, p. 1) and digital interpolation techniques are now standard in most packages. The degree of smoothing should be a factor to consider, because it is related to the time required to generate the lines and because critical detail may be missed by smoothing lines.

Digital smoothing of contours can be created in two ways: first, by adjusting the grid spacing (denser for smoother lines) and, second, by the number of intermediate points used in interpolation between the grid points. Figure 16, for example, illustrates a smoothed version of Figure 3. In general, when the grid spacing is larger, the lines will be angular unless more intermediate interpolating points are used.

Efficiency

The question of efficiency in drawing the contours is important from the machine-time standpoint. If a plotter is online to a batch process main computer, the time becomes of extreme importance. Even in offline plotting, the time used to draw lines is important when many maps are drawn.

Efficiency in drawing lines is mainly a function of the type of plotter used. In pen and ink plotters, the speed at which the pen moves is independent of whether the pen is on the paper or in the air; hence, a minimum of "pen-up" time is desirable. Other devices, which paint lines with a moving light source or by an electrostatic method, are not restricted to pen movements. Some of these devices, however, move only once across a band of paper, so that all information for that area must be transferred at one time.

Pen-plotting efficiency derives from minimizing pen-up time. This can be accomplished by covering a square of the paper completely, then moving on to the next. This technique is efficient if the square is moderate in size and if the lines are drawn alternately from left to right and then vice versa so that the pen has the shortest distance to move to a new line. Other techniques follow one contour over the entire page before moving to the next contour. This approach is efficient when few local closed anomalies exist, but becomes decreasingly efficient when numerous small closed contours must be drawn.

The use of a band pass approach to the data is unquestionably one of the least efficient methods for pen and ink plotters, unless steps are taken to alternate direction of drawing and ensure a wide band at each pass. This approach is used in certain cathode ray tube plotters and results in the rapid depiction of such information once the basic data have been stored. In such CRT devices, the speed of the contour

generation part of the program is independent of the density of lines to be drawn, which is certainly not the case with pen and ink methods.

Certain programs offer both band pass and continuous line generating techniques. This selection would not change until the plotter is replaced by one with dissimilar plotting characteristics.

Special Types of Contour Lines

Certain programs allow the user to choose which contour lines are to be labeled, which lines should be heavy, and whether tick marks are desired in closed minimum contours. These features, which add to the visual acceptance of the presentation, are usually part of the standard specifications.

Annotation

Special map annotations are useful in preparing finished illustrations (*e.g.*, Figs. 1, 6). The user should be able to draw a line across the graph by giving the end point coordinates. Character information can likewise be specified, giving the starting location of the character string, the angle, size of lettering and characters which make up the string. Most plotters' instructions are designed to facilitate user control of special annotation. Although this feature is not essential in some uses, it is quite useful in producing a finished product.

Multiple Job Runs

The most common first approach to contouring by machine involves a single set of data. Sometimes, however, one may wish to draw two sets of contours on a single map (Fig. 15), whether the contours represent two separate areas or a second superimposed set. From a program standpoint, each of these operations requires passing through the data point input, grid generation, blanking, and contouring phases, but the lines will be drawn on the same map. Other applications require having two or more separate maps in one computer run, cycling through the entire program from map specification through contour generation and annotation.

Two types of job combinations, therefore, should be available: it should be possible to proceed to a new, separate map, initiating the program and starting afresh; one should also be able to modify a map by adding new contours. Many contour programs allow the first option; at least one program permits stacking maps on one surface. If the latter capability is not present, the sequential maps should be plotted on clear film so that they can be overlaid.

Summary

This report on the common applications for machine contouring considers the design of a general purpose contour program from the viewpoint of a user unfamiliar with the programming aspects of complex general purpose systems. Emphasis is placed on developing user familiarity with those factors which primarily affect the results he is likely to obtain.

There is no substitute for experience in the use of automated programs. A trained user can reproduce most of the judgmental factors in his final interpretation with a minimum of experimentation. In order to gain some insight into this process, however, a few guidelines might be useful.

Many people using machine contour programs expect a perfect result the first time the program is run on a new set of data. Failure on the part of the program logic to produce such a result is often taken as an indication that "machine contouring is no good." This notion can be compared with instructing a specialist to contour all his information with a felt pen, allowing no erasures. Brisk electric eraser sales in technical research situations provide mute testimony to the unlikelihood of such a requirement.

Similarly, the machine may be regarded as fallible, but capable of being instructed to produce a better map. Contouring by machine is rarely a one-pass effort if the map is to be stored or used in subsequent operations. The user will often wish to adjust his input and control cards, override certain calculated array values, shift contour spacing in certain areas, *etc.*, before storing the results for future reference. This is a natural process which can be achieved if the researcher has ready access to the machine.

Often it is necessary to consider the reasons for the failure of a map to agree with preconceived notions: data may be too irregularly distributed, clusters of data may need to be thinned, or a real anomaly may exist which had not been suspected. This type of interaction allows the user to apply his expert judgment and the machine to do the drudgery of producing the result based on that cooperative effort. The experience of interaction can be exciting when performed in this way, and the benefits truly

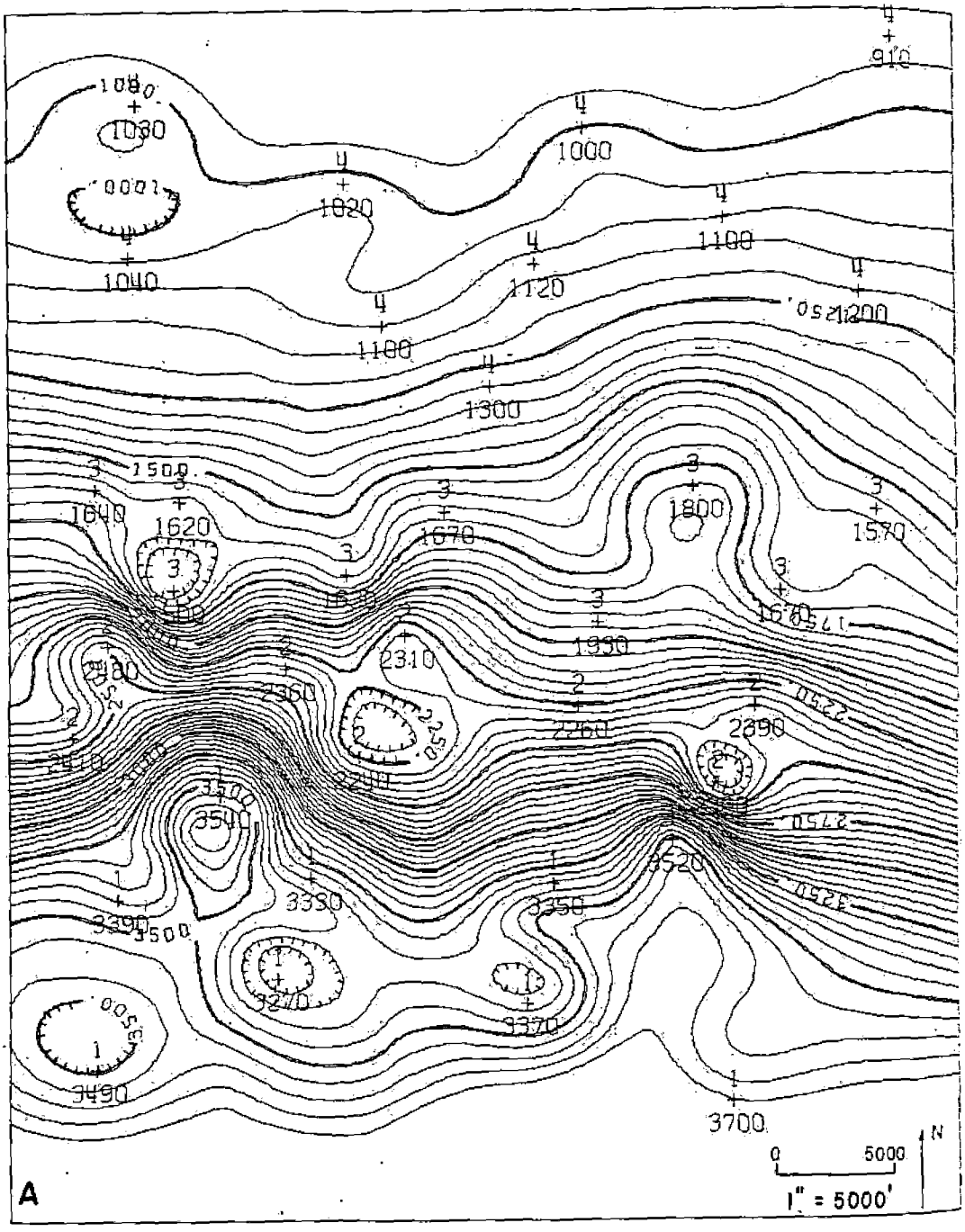
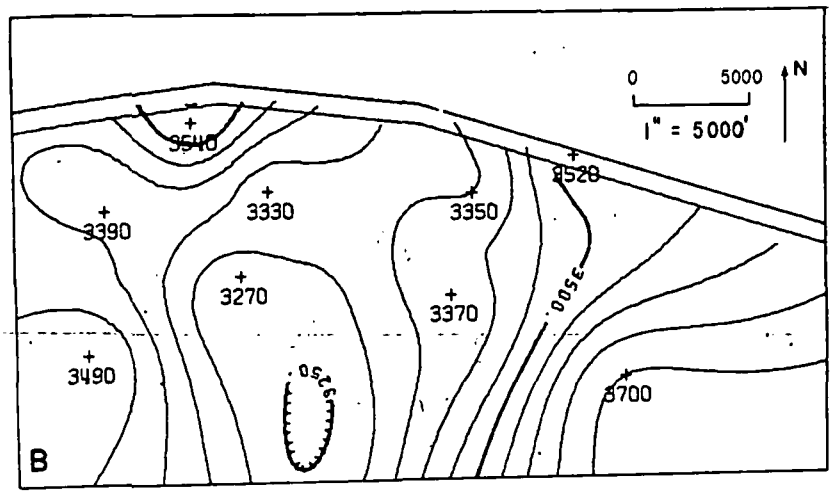
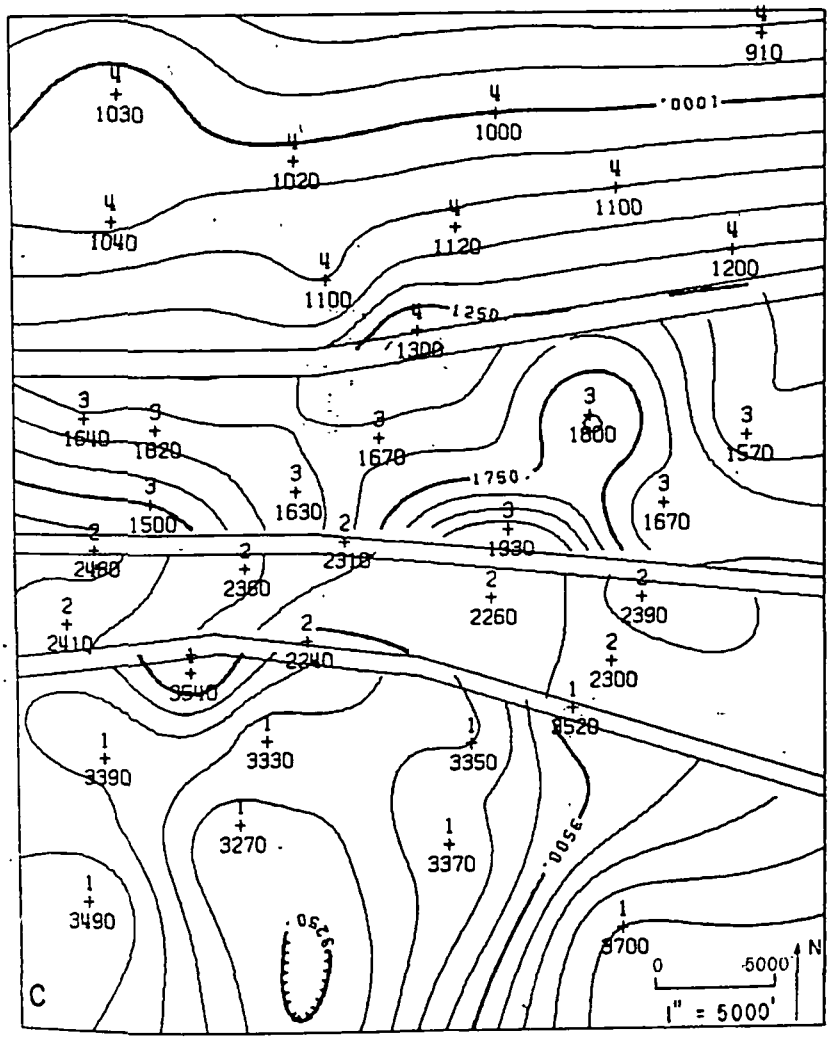


FIG. 15.—Use of blanking provision to create isopach map of several fault blocks. Hypothetical example typical of Gulf Coast Tertiary stratigraphy. **A.** Isopach uncorrected. Original data points from four fault blocks contoured as single map. **B.** Contours of lowermost fault block separately. Data points for this fault block are read in, a grid generated for entire map, then area above fault is blanked out, using special blanking provision provided in GPCP program. **C.** Combination isopach map of four fault blocks. Same technique is used for each fault block; contours are drawn separately but on same map to produce finished result.


0 5000 N
1" = 5000'
3540
3528
3390
3330
3350
3270
3370
3500
3490
3250
3700
B


910
1030
1000
0001
1020
1100
1040
1120
1200
1100
1250
1300
1640
1820
1800
1570
1678
1750
1630
1670
1500
1930
2488
2318
2360
2260
2390
2410
2340
2300
3540
3528
3390
3330
3350
3270
3370
3500
3490
3250
3700
0 5000 N
1" = 5000'
C

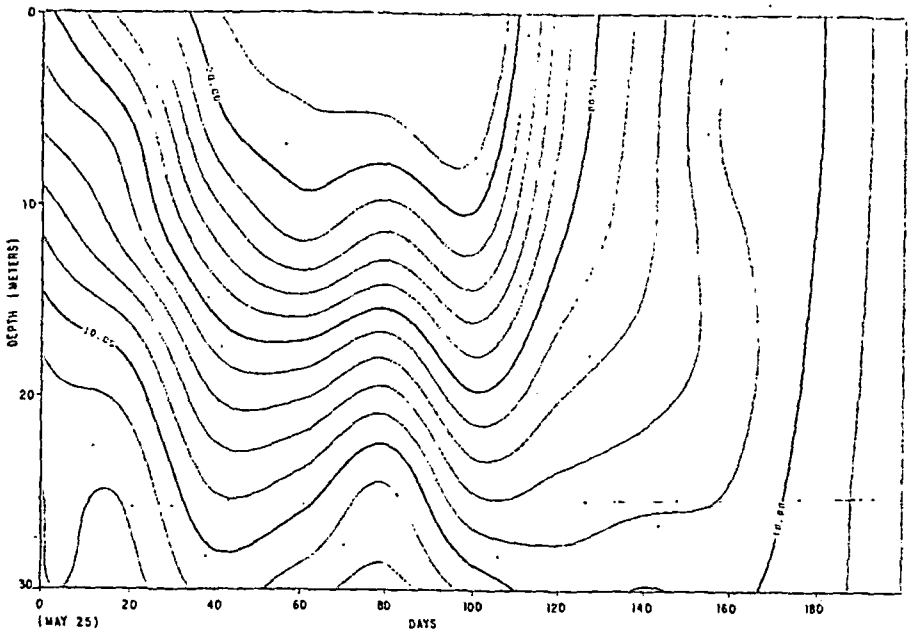


FIG. 16.—Same map as Figure 3, using same grid spacing (11 × 7) but utilizing a smoothing technique for drawing lines between data points.

unexpected in terms of generating ideas and maintaining a consistent suite of graphs for any particular project.

SELECTED REFERENCES

Assiter, E., 1967, Surface approximations and contour mapping: Unpub. ms. available from Edward J. Assiter, President, Applications Consultants Inc., 3431 West Holcombe Boulevard, Houston, Texas 77025.

Batcha, J. P., and J. R. Reese, 1964, Surface determination and automatic contouring for mineral exploration, extraction and processing: Colorado School Mines Quart., v. 59, no. 4, p. 1–14.

Cain, J. C., and J. R. Neilon, 1963, Automatic mapping of the geomagnetic field: Jour. Geophysical Research, v. 68, no. 16, p. 4690.

Calcomp, Inc., 1968, GPCP: General purpose contouring program: Preliminary manual, Anaheim, California.

Goldman, C. R., M. Gerletti, U. Javornicky, and E. de Amezaga, 1968, Correlations and relationships between primary productivity and numerous ecological factors in Lago Maggiore: Lago Maggiore, Italy.

Harbaugh, J. W., and D. F. Merriam, 1968, Computer applications in stratigraphic analysis: New York, J. Wiley & Sons, Inc.

IBM, 1965, Numerical techniques and contour map plotting: IBM Data Processing Application, White Plains, New York.

McIntyre, D. B., D. D. Pollard, and R. Smith, 1968, Computer programs for automatic contouring: Kansas Geol. Survey Computer Contr. 23, 75 p.

Ojakangas, D. R., and W. L. Basham, 1964, Simplified computer contouring of exploration data: Stanford Univ. Pubs. Geol. Sci., v. 9, no. 2, p. 757–770.

Pruner, Harold, 1966, Computer cuts unit mapping and calculation time for Shell: Oil and Gas Jour., v. 64, no. 28, p. 80–81.

Rens, F. J., 1965, A FORTRAN program for coordinate mapping using IBM 7090 computer: U.S. Office Naval Research Geography Branch Tech. Rept. 10, ONR Task No. 389–135, 94 p.

Tobler, W. R., 1965, Automation in the preparation of thematic maps: Cartographic Jour., v. 2, no. 1, p. 32–38.

Structural A
Plains of So

Abstract Elevati
100,000 mi² (250,0
and central Alberta
maps of four stra
carbonate rocks, s
Fish Scales Sandst
Shale. Each map,
as undulations, mo
of sinusoid surfac
phases, and orient
resented mainly by
smaller undulatior
filtering permits th
show only those un
that fall within a
process requires o
mining the spatial
the desired undul
the frequency dom
those in the desi
filter into the dista
the digitized input
In this study on
showing undulatior
The dominant tren
southeast, parallel
basement. Relief, u
unconformity and l
of the undulations
bounded by steepl
a northwest-trendi
placement of 10 m
dent with known p

INTRODUCTION

Digital comp
large amounts o
the geologist of

[1] Manuscript rec
May 5, 1969.

[2] Union Oil Con
[3] University of A

The writers grat
A. J. Broscoe, R.
G. W. Smith, C. I
University of Alb
E. K. Shaw of Im
of Stanford Univ
and advice during
this paper. Finan
National Research
Survey of Canad
Ltd., and the Un
supplied the infor
ture-contour map

SPSS - STATISTICAL PACKAGE FOR THE SOCIAL SCIENCES

SUBJ
COMP
CIP

The SPSS program consists of a collection of statistical programs which can be called into operation to perform any of a number of predefined statistical tasks. Before any of the statistical programs are called, it is necessary that the user's data be defined.

Two basic units of data are "cases" and "variables". The term "case" refers to the information corresponding to an individual person, place or thing (a rock sample, for example). A variable name (8A) of the user's choice is assigned to each information item to be processed. Both data definition and task assignment are specified by means of SPSS control statements, prepared in the following form

control field (columns 1-15)	specification field (columns 16-80)

The control field contains a unique control word or word set which specify a type of declaration or directive to be utilized by the SPSS system. The specification field provides the specifics pertaining to the named declaration or directive.

The SPSS control statements are in general arranged in the following sequence:

RUN NAME STATEMENT
DATA DEFINITION STATEMENTS
TASK DEFINITION STATEMENTS (first task)
DATA RECORDS
ADDITIONAL TASK DEFINITION STATEMENTS

DATA DEFINITION STATEMENTS

Data definition statements inform SPSS of the variable names, their input format, and the number of cases. They are also used to put labels on variables and values for the user's convenience.

examples:

```
VARIABLE LIST   POROUS, DENSE, TYPE, COLOR
INPUT FORMAT    FIXED (1X,F4.0,1X,F3.0,1X,F1.0,1X,F1.0)
N OF CASES      180
INPUT MEDIUM    CARDS
VAR LABELS      POROUS, POROSITY/DENSE, ROCK DENSITY/TYPE,
                ROCK TYPE/COLOR, ROCK COLOR
VALUE LABELS    TYPE (1) SANDSTONE (2) MARBLE (3) GRANITE
```

TASK DEFINITION STATEMENTS

The task definition statements activate, define, and control the calculations performed on the data. More than one task can be defined in one program run.

For most tasks, 3 statements are required; a procedures statement, an options statement, and a statistics statement. The procedures statement selects the SPSS subprogram. Columns 1-15 contain a specific control word; columns 16-80 contain the variables to be entered into the calculations as well as parameters and arguments required for the specified computation.

examples:

```
CROSSTABS       TABLES = ROCK BY DENSITY/POTASH BY SILICA
PEARSON CORR    SODA, LIME, TITAN, COPPER, URANM
REGRESSION      VARIABLES = SODA TO ZINC
FACTOR          VARIABLES = A,B,C,D/TYPE = RAO/
                ROTATE = VARIMAX
```

The options statements specify various options available for each subprogram, usually dealing with output format, treatment missing data, etc.
The statistics statements specify which of available statistics the user wants calculated.

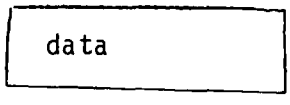
examples:

```
OPTIONS         1,3,7
STATISTICS      1,2,4
```

SEVEN BASIC SPSS STATEMENTS

There are seven basic statements which must occur in any SPSS program. These are shown in the examples below

```
RUN NAME        YOUR RUN NAME HERE
VARIABLE LIST   VAR1, VAR2, VAR3
INPUT FORMAT    FIXED (T4,F1.0,T8,F1.0,2x,F5.3)
N OF CASES      81
INPUT MEDIUM    CARDS

      --procedures card--

READ INPUT DATA
```




```
FINISH
```

DATA

Data may be entered into SPSS either in a fixed or free format. For small cases, the data is simply included in the program file. Provisions are available, however, for calling data from tapes, disc, other files etc. In general, format specifications are consistent with FORTRAN usage.

TASK DEFINITION SUMMARY

The following table presents a summary of the control fields for all task definition statements. The first two provide the procedures and information on where raw output data should be written and what cases should be processed together. Each of the next 16 activate one of the SPSS routines.

RAW OUTPUT UNIT	
RUN SUBFILES	
CONDESCRIPTIVE	(statistics of continuous data)
FREQUENCIES	(statistics of discrete data)
AGGREGATE	(statistics of aggregated characteristics)
CROSSTABS	(crosstabulation)
BREAKDOWN	(statistics of continuous variable by discrete groups)
T-TEST	(T-test of difference of means of subgroups)
PEARSON CORR	(Pearson correlation)
NONPAR CORR	(nonparametric correlation)
SCATTERGRAM	(scattergram)
PARTIAL CORR	(partial correlation coefficients)
REGRESSION	(multiple regression analysis)
ANOVA	(analysis of variance)
ONEWAY	(ANOVA with one independent variable)
DISCRIMINANT	(discriminant analysis)
FACTOR	(factor analysis)
CANCORR	(canonical correlation analysis)
GUTTMAN SCALE	(Guttman scales)
OPTIONS	
STATISTICS	
READ INPUT DATA	

OTHER POWERS

MISSING VALUES - there are options supplied for most programs to deal with missing data values by ignoring them, assigning them etc.

SUBFILES - if data can be divided into subfiles (such as thermal conductivity measurements from 3 geographic areas) the subfiles can be run alone, or in groups, or together, or may for some programs be run against each other.

RECODING - data may be changed easily by simple recode statements-for example, all "graywackes" could be lumped together with "shale" by one statement.

COMPUTE - compute statements may be used to modify data or compute new variables, for example.

```
COMPUTE    POTLOG = LGIO (POTASH)
```

will log transform the variable POTASH to create a new variable POTLOG which may be used in subsequent operations (for example, such a transform is necessary in order to calculate Pearson correlations from log normal data distributions).

Operators include all standard FORTRAN operations: +, -, /, *, **, SQRT, LN, ABS, SIN, etc. IF statements can also be used for variable transformation. In general, these look like FORTRAN without the periods.

examples:

```
IF          (POTASH  EQ  SODA)     INDEX = 7

IF          (SILICA/60.0 GT 5 AND SODA+POTASH LE LIME) CLASS=1
```

ERRORS

Fortunately, SPSS was created for DUMMIES and is very forgiving. The diagnostics are quite good, and default options cover many omissions (and usually inform the user politely of the coverage).

EXPERTS

UUCC has personnel who have worked extensively with SPSS and can answer many questions related to the program. Short courses on SPSS are taught quarterly, and documentation is available from the consultants. Carol Withrow will be implementing SPSS on PRIME and will be knowledgeable about JCL. Carol, Bev Miller, and Odin Christensen have taken the SPSS course and used the program. Ultimately, refer to

Nie, N.H., Hall, C.H., Jenkins, J.G., Steinbrenner, K., and Bent, D.H. SPSS-Statistical Package for the Social Sciences, McGraw-Hill.

available in one of the ESL computer people's offices or at the UU bookstore.

SPSS "CODEBOOK" WORKSHEET for Data File _____ Cards per Case = 1 Page 1 of 1

Item No.	Item Description	Variable Name / Variable Label (max = 8 char / max = 40 ")	Val	Value Label (max = 16 char)	Card 1 Columns	Format	Var. Type
1	Sex	SEX	(1)	MALE	2	F1	Nom.
		GENDER	(2)	FEMALE			
2	Class Standing	CLASS	(1)	FRESHMAN	5	F1	Ord.
		CLASS STANDING	(2)	SOPHOMORE			
			(3)	JUNIOR			
			(4)	SENIOR			
			(5)	GRADUATE			
3	Grade-point average	GPA	range:	0.00 - 4.00	8-10	F3.2	Ratio
		GRADE-POINT AVE					
		RAGE					

A-1

Data

col. 2	col. 5	cols. 8-10
1	2	236
2	2	203
1	3	314
1		340
2	1	324

case 3: male junior with GPA = 3.14

Rules for forming variable names:

- Must begin with alphabetic character.
- May use up to 8 alpha or num chars.
- No blanks or special characters allowed.

This form is a worksheet for the following SPSS control cards: VARIABLE LIST / VAR LABELS / VALUE LABELS / INPUT FORMAT

FILE NONAME (CREATION DATE = 78/10/19.)
* M U L T I P L E R E G R E S S I O N *
DEPENDENT VARIABLE.. PERFORM JOB PERFORMANCE INDEX

S U M M A R Y T A B L E

STEP	VARIABLE ENTERED	VARIABLE REMOVED	F TO ENTER OR REMOVE	SIGNIFICANCE	MULTIPLE R	R SQUARE	R SQUARE CHANGE	SIMPLE R	OVERALL F	SIGNIFICANCE
1	TEST2		15.58907	.001	.68126	.46411	.46411	.68126	15.58907	.001
2	TEST1		16.00402	.001	.85086	.72397	.25986	.13560	22.29372	.000

FILE NONAME (CREATION DATE = 78/10/19.)
* M U L T I P L E R E G R E S S I O N *

OBSERVATION	Y VALUE	Y ESTIMATE	RESIDUAL	-2SD 0.0 +2SD
1.	130.0000	109.1615	20.83849	I .
2.	168.0000	159.9743	8.025713	I .
3.	158.0000	143.9905	14.00948	I .
4.	37.00000	17.68794	19.31206	I .
5.	189.0000	148.7424	40.25757	I .
6.	160.0000	225.4201	-65.42008	I
7.	144.0000	141.2903	2.709654	I .
8.	176.0000	155.2762	20.72376	I .
9.	170.0000	132.4891	37.51090	I .
10.	184.0000	162.3502	21.64976	I .
11.	107.0000	165.4818	-58.48183	I
12.	57.00000	86.96842	-29.96842	I
13.	111.0000	104.5714	6.428557	I .
14.	11.00000	77.46461	-66.46461	I
15.	217.0000	232.3322	-15.33224	I
16.	80.00000	101.4396	-21.43959	I
17.	223.0000	238.3800	-15.37997	I
18.	299.0000	231.6299	67.37007	I .
19.	124.0000	103.1679	20.83211	I .
20.	109.0000	116.1814	-7.181382	. I

NOTE - (*) INDICATES ESTIMATE CALCULATED WITH MEANS SUBSTITUTED
R INDICATES POINT OUT OF RANGE OF PLOT

NUMBER OF CASES PLOTTED 20.
NUMBER OF 2 S.D. OUTLIERS 0 OR 0 PERCENT OF THE TOTAL

VON NEUMANN RATIO 1.91528 DURBIN-WATSON TEST 1.81951

NUMBER OF POSITIVE RESIDUALS 12.
NUMBER OF NEGATIVE RESIDUALS 8.
NUMBER OF RUNS OF SIGNS 8.

NORMAL APPROXIMATION TO SIGN DISTRIBUTION IMPOSSIBLE.
USE A TABLE FOR EXPECTED VALUES.

REGRESSION
VARIABLES=PERFORM TO TEST2/
REGRESSION=PERFORM WITH TEST1 TO TEST2 (1)/RESIDUALS
STATISTICS ALL

Required to obtain a plot of the residuals.

CODING SHEET FOR SPSS CONTROL CARDS

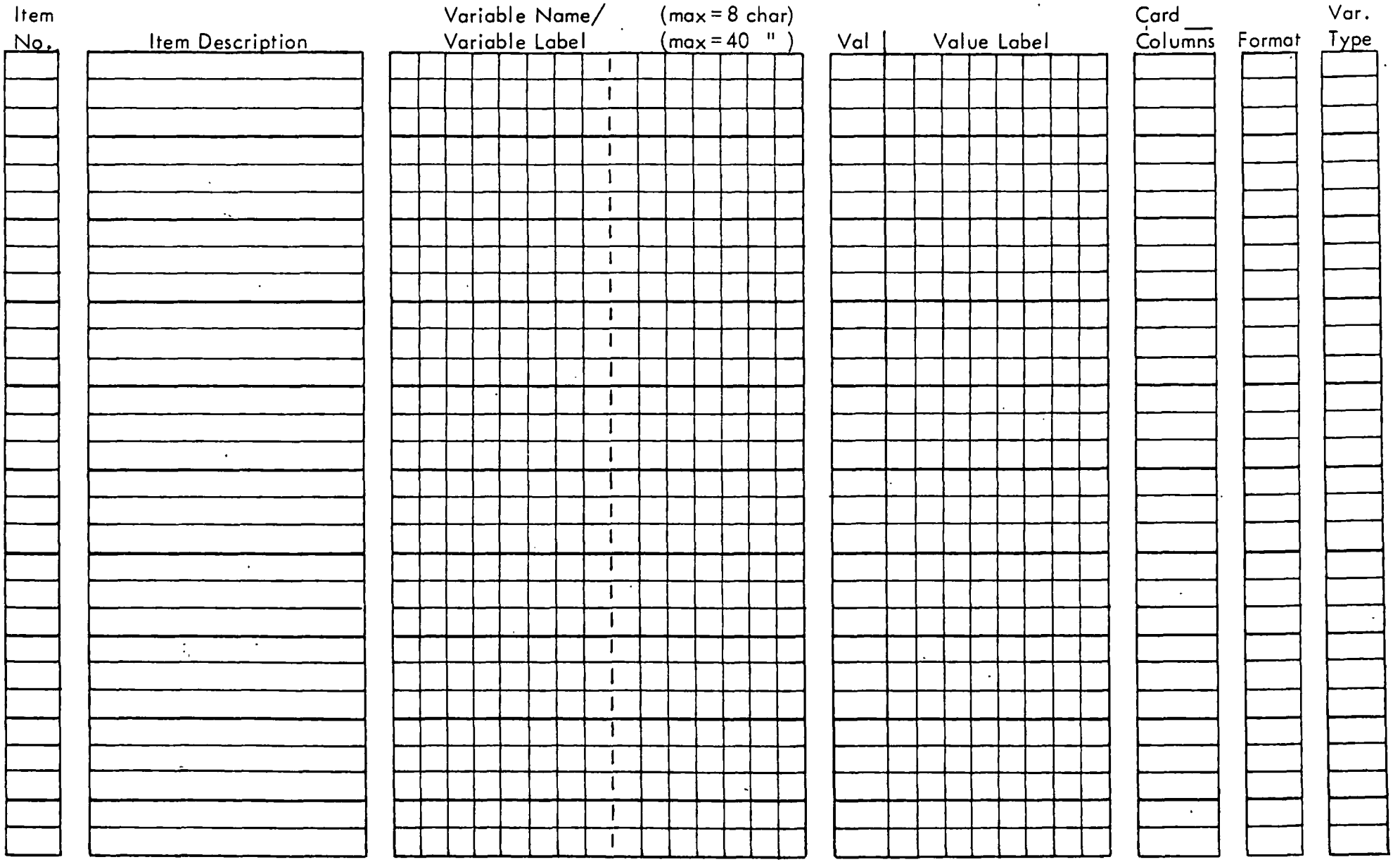
( ) Denotes basic SPSS control cards

1) EDIT	(7) INPUT MEDIUM	13) COMPUTE	19) STATISTICS	)		)	
(2) RUN NAME	8) VAR LABELS	14) ASSIGN MISSING	(20) READ INPUT DATA	)		)	
3) FILE NAME	9) VALUE LABELS	15) IF	21) SAVE FILE	)		)	
(4) VARIABLE LIST	10) MISSING VALUES	16) SELECT IF	(22) FINISH	)		)	
(5) INPUT FORMAT	11) PRINT FORMATS	17) FREQUENCIES	23) COMMENT	)		)	
(6) N OF CASES	12) RECODE	18) OPTIONS	)	)		)	

```
control card #  1    2    2    3    3    4    4    5    5    6    6    7    7    8
                6    0    5    0    5    0    5    0    5    0    5    0    5    0
2               SPSS  ILLUSTRATIVE RUN 1
23              **********************************************************
                *          EXAMPLE OF SIMPLE SPSS RUN                    *
                *  (USES THE 7 BASIC SPSS CONTROL CARDS)                 *
                **********************************************************
4               SEX, CLASS
5               (T2, F1,  T5, F1)
6               5
7               CARD
17              GENERAL=SEX, CLASS
20
                Data cards go here.
22
```

Basic SPSS Control Cards	Misc. SPSS Control Card	SPSS Procedure Card	Data Cards
RUN NAME			
	COMMENT		
VARIABLE LIST			
INPUT FORMAT			
N OF CASES			
INPUT MEDIUM			
		FREQUENCIES	
READ INPUT DATA			1 2 236
			2 2 203
			1 3 314
			1 340
			2 1 324
FINISH			

SPSS "CODEBOOK" WORKSHEET for Data File _____ Cards per Case = ___ Page ___ of ___



This form is a worksheet for the following SPSS control cards: VARIABLE LIST/VAR LABELS/VALUE LABELS/INPUT FORMAT

CODING SHEET FOR SPSS CONTROL CARDS

( ) Denotes basic SPSS control cards

Page ___ of ___

1) EDIT	(7) INPUT MEDIUM	13) COMPUTE	19) STATISTICS	) _____	) _____
(2) RUN NAME	8) VAR LABELS	14) ASSIGN MISSING	(20) READ INPUT DATA	) _____	) _____
3) FILE NAME	9) VALUE LABELS	15) IF	21) SAVE FILE	) _____	) _____
(4) VARIABLE LIST	10) MISSING VALUES	16) SELECT IF	(22) FINISH	) _____	) _____
(5) INPUT FORMAT	11) PRINT FORMATS	17) _____	) _____	) _____	) _____
(6) N OF CASES	12) RECODE	18) OPTIONS	) _____	) _____	) _____

control card no.

1 2 2 3 3 4 4 5 5 6 6 7 7 8
6 0 5 0 5 0 5 0 5 0 5 0 5 0

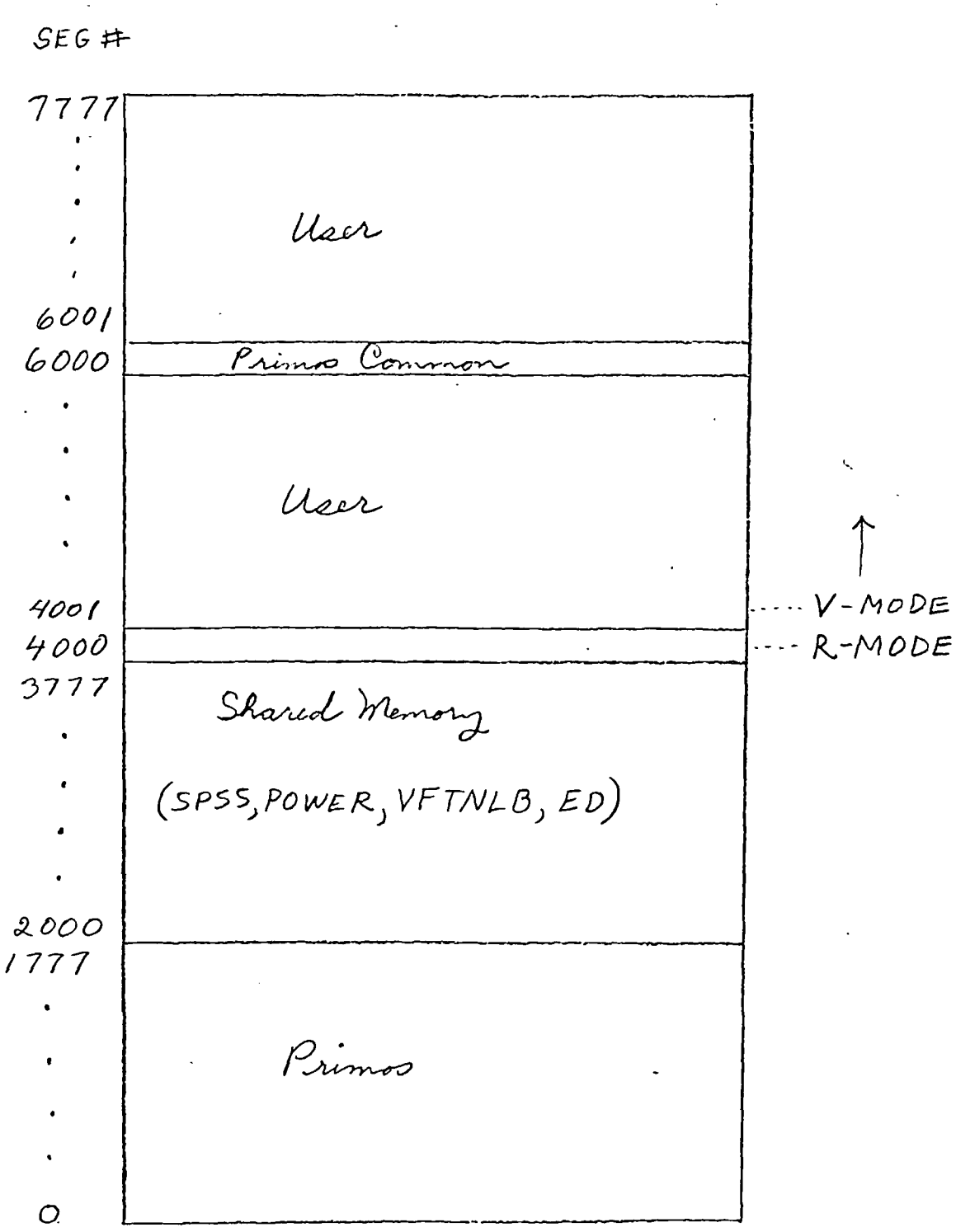
Command Files

1. COMOUTPUT or COMO files are used for saving terminal output. See notes from previous lecture.

2. COMINPUT files cause input command stream to be read from the file rather than the terminal.

3. PHANTOM files initiate a long-running program that need not be associated with a terminal.

Primos Virtual Memory:



Example of SEG Loader Use

```
OK, FOR PROGRAM1
0000 ERRORS [<.MAIN.>FTN-REV17.2]

OK, FOR SUB
0000 ERRORS [<SUB   >FTN-REV17.2]

OK, SEG
[SEG rev 17.1]
# LOAD #PROGRAM1
$ LO B_PROGRAM1
$ LO B_SUB
$ LI ESL
$ LI VAPPLB
$ LI
LOAD COMPLETE
$ SAVE
$ QUIT
```

SEG Loader MAP Command

```
OK, SEG
[SEG rev 17.1]
# LOAD #PROGRAM1
$ LO B_PROGRAM1
$ LO B_SUB
$ LI VAPPLB
$ LI
$ MA 3
**TYPA       4002  000052


$ LI ESL
LOAD COMPLETE
$ SAVE
$ MA 1
*START  004002  000000  *STACK  004001  001122  *SYM    000025

SEG. #    TYPE        LOW       HIGH       TOP
004001    PROC       001000    001121    001121
004002    DATA       000000    000103    000103

$ QUIT
```

```
(        1) * CREATE #KCOBW
(        2) * COMPILE AND LOAD
(        3) * USE LOADER DEFAULTS
(        4) *
(        5) ED
(        6)
(        7) LOAD KCOBW
(        8) LOAD PASSWD
(        9) FILE SOURCE
(       10) FOR SOURCE
(       11) C 7
(       12) FILMEM ALL
(       13) SEG
(       14) LOAD #KCOBW
(       15) LO B_SOURCE
(       16) LI ESL
(       17) LI VAPPLB
(       18) LI
(       19) SAVE
(       20) MA 3
(       21) MA M_KCOBW
(       22) QUIT
(       23) DELETE SOURCE
(       24) DELETE B_SOURCE
(       25) CO -END
```

Listing of C_KCOBW

This is the terminal output that results from the execution of C_KCOBW.

```
OK, CO C_KCOBW
OK, * CREATE #KCOBW
OK, * COMPILE AND LOAD
OK, * USE LOADER DEFAULTS
OK, *
OK, ED
INPUT

EDIT
LOAD KCOBW
EDIT
LOAD PASSWD
EDIT
FILE SOURCE

OK, FOR SOURCE
0000 ERRORS [<.MAIN.>FTN-REV17.2]
0000 ERRORS [<DATAS >FTN-REV17.2]
0000 ERRORS [<DATAM >FTN-REV17.2]
0000 ERRORS [<GRAPHS>FTN-REV17.2]
0000 ERRORS [<PASSWD>FTN-REV17.2]

OK, C 7
OK, FILMEM ALL

OK, SEG
[SEG rev 17.1]
# LOAD #KCOBW
$ LO B_SOURCE
$ LI ESL
$ LI VAPPLB
$ LI
LOAD COMPLETE
$ SAVE
$ MA 3


$ MA M_KCOBW
$ QUIT

OK, DELETE SOURCE
OK, DELETE B_SOURCE
OK, CO -END
OK,
```

```
*START  004002  000010  *STACK  004001  013644  *SYM    000057

SEG. #    TYPE        LOW       HIGH      TOP
004001    PROC##    000100    013642    013642
004002    DATA      000000    011175    011175

*BASE   004001  000100  000111  000777  000777

ROUTINE        ECB          PROCEDURE      ST. SIZE  LINK FR.
####       4002  000010    4001  001000    000042  001004  4002  177400
DATAS      4002  002220    4001  007054    000040  000152  4002  001610
DATAM      4002  002370    4001  010170    000040  000150  4002  001762
GRAPHS     4002  002534    4001  011276    000062  006226  4002  002132
PASSWD     4002  010762    4001  012536    000024  000046  4002  010360
OPNWRT     4002  011036    4001  012672    000036  000042  4002  010426
CLOSEF     4002  011076    4001  013030    000032  000036  4002  010470
TYPA       4002  011126    4001  013126    000034  000022  4002  010526
E$27       4001  013142    4001  013162    000042  000000  4002  010550
ABS        4002  011150    4001  013276    000016  000020  4002  010550
F$ERX      4001  013330    4001  013304    000020  000006  4002  010570

DIRECT ENTRY LINKS
  ERRPR$    4001  013352    EXIT      4001  013356    SRCH$$    4001  013362
  TNOU      4001  013366    TNOUA     4001  013372    TSRC$$    4001  013376
  F$WA      4001  013402    F$RA      4001  013406    F$A1      4001  013412
  F$A2      4001  013416    F$A7      4001  013422    F$CB      4001  013426

COMMON BLOCKS
            4002  001004  001204

OTHER SYMBOLS
  AL10$X    4001  013432    ALOG$X    4001  013434
```

LOAD STATE

– BASE AREA

OTHER SYMBOLS

```
 1) * CREATE #KCOBW
 2) * COMPILE AND LOAD
 3) * USE LOADER DEFAULTS
 4) *
 5) ED
 6)
 7) LOAD KCOBW
 8) LOAD PASSWD
 9) FILE SOURCE
10) FOR SOURCE
11) C 7
12) FILMEM ALL
13) SEG
14) LOAD #KCOBW
15) LO B_SOURCE
16) LI VAPPLB
17) LI
18) SAVE
19) MA 3
20) MA M_KCOBW
21) QUIT
22) DELETE SOURCE
23) DELETE B_SOURCE
24) CO -END
```

Listing of C_KCOBW (Same as before, but omitting the loading of library ESL.)

```
OK, CO C_KCOBW
OK, * CREATE #KCOBW
OK, * COMPILE AND LOAD
OK, * USE LOADER DEFAULTS
OK, *
OK, ED
INPUT

EDIT
LOAD KCOBW
EDIT
LOAD PASSWD
EDIT
FILE SOURCE

OK, FOR SOURCE
0000 ERRORS [<.MAIN.>FTN-REV17.2]
0000 ERRORS [<DATAS >FTN-REV17.2]
0000 ERRORS [<DATAM >FTN-REV17.2]
0000 ERRORS [<GRAPHS>FTN-REV17.2]
0000 ERRORS [<PASSWD>FTN-REV17.2]

OK, C 7
OK, FILMEM ALL

OK, SEG
[SEG rev 17.1]
# LOAD #KCOBW
$ LO B_SOURCE
$ LI VAPPLB
$ LI
$ SAVE
$ MA 3
**OPNWRT    4002  002424  **CLOSEF    4002  002504  **TYPA      4002  0025

$ MA M_KCOBW
$ QUIT

OK, DELETE SOURCE
OK, DELETE B_SOURCE
OK, CO -END
```

This is the terminal output that results from the execution of this version of C_KCOBW. Notice there are three unsatisfied references.

```
*START  004002  000010  *STACK  004001  013360  *SYM    000047

SEG. #    TYPE        LOW       HIGH      TOP
004001    PROC##    000100    013356    013356
004002    DATA      000000    011053    011053

*BASE   004001  000100  000111  000777  000777

ROUTINE        ECB          PROCEDURE      ST. SIZE  LINK FR.
####       4002  000010    4001  001000    000042  001004  4002  177400
DATAS      4002  002220    4001  007054    000040  000152  4002  001610
DATAM      4002  002370    4001  010170    000040  000150  4002  001762
GRAPHS     4002  002534    4001  011276    000062  006226  4002  002132
PASSWD     4002  010762    4001  012536    000024  000046  4002  010360
E$27       4001  012672    4001  012712    000042  000000  4002  010426
ABS        4002  011026    4001  013026    000016  000020  4002  010426
F$ERX      4001  013060    4001  013034    000020  000006  4002  010446

DIRECT ENTRY LINKS
  EXIT      4001  013102    TNOU      4001  013106    TNOUA     4001  013112
  F$WA      4001  013116    F$RA      4001  013122    F$A1      4001  013126
  F$A2      4001  013132    F$A7      4001  013136    F$CB      4001  013142

COMMON BLOCKS
            4002  001004  001204

OTHER SYMBOLS
  AL10$X    4001  013146    AL0G$X    4001  013150  **OPNWRT    4002  002424
**CLOSEF    4002  002504  **TYPA      4002  002530
```

M_KCOBW after the error
load on the preceding page

```
OK, SEG #KCOBW
Please enter password.
HOCUS
Password is correct.  Program continuing.

Error: condition "POINTER_FAULT$" raised at 6(0)/40116.
Entry to inner ring was from call at 4001(3)/1002.
ER! PM
SA,EA,P,A,B,X,K=
100 54146 1002 0 0 0 14000

PB,SB,LB,XB:
4(4001/1002) 64001/13644 4002/177400 6/30342
OK, SLIST M_KCOBW
*START  004002  000010  *STACK  004001  013632  *SYM    000057

SEG. #    TYPE        LOW       HIGH      TOP
004001    PROC##    000100    013630    013630
004002    DATA      000000    011201    011201

*BASE   004001  000100  000111  000777  000777

ROUTINE        ECB          PROCEDURE     ST. SIZE  LINK FR.
####      4002  000010    (4001  001000)   000042  001010  4002  177400
DATAS     4002  002224     4001  007042    000040  000152  4002  001614
DATAM     4002  002374     4001  010156    000040  000150  4002  001766
GRAPHS    4002  002540     4001  011264    000062  006226  4002  002136
PASSWD    4002  010766     4001  012524    000024  000046  4002  010364
OPNWRT    4002  011042     4001  012660    000036  000042  4002  010432
CLOSEF    4002  011102     4001  013016    000032  000036  4002  010474
TYPA      4002  011132     4001  013114    000034  000022  4002  010532
E$27      4001  013130     4001  013150    000042  000000  4002  010554
ABS       4002  011154     4001  013264    000016  000020  4002  010554
F$ERX     4001  013316     4001  013272    000020  000006  4002  010574

DIRECT ENTRY LINKS
  ERRPR$    4001  013340    EXIT      4001  013344    SRCH$$    4001  013350
  TNOU      4001  013354    TNOUA     4001  013360    TSRC$$    4001  013364
  F$WA      4001  013370    F$RA      4001  013374    F$A1      4001  013400
  F$A2      4001  013404    F$A7      4001  013410    F$CB      4001  013414

COMMON BLOCKS
            4002  001010  001204

OTHER SYMBOLS
  AL10$X    4001  013420    ALOG$X    4001  013422
```

Examples of Reloading

1. Purpose: To obtain a load map

```
SEG
LOAD * #PROGRAM1
MA M_PROGRAM1
QUIT
```

2. Purpose: To reload a portion of a runfile

```
SEG
LOAD * #PROGRAM1
RL B_SUB
SAVE
QUIT
```

UNIT NUMBERS

Fortran Unit Number	Primos Unit Number	Reserved Use
1		User terminal
2		
3		
4		Line printer
5	1	Input
6	2	Listing
7	3	Binary
8	4	
9	5	
10	6	COMINPUT
11	7	
12	8	
13	9	
14	10	
15	11	SEG's Load Map
16	12	
17	13	
18	14	Statos Printer/Plotter
19	15	ESL Contour Routine
20	16	ESL Plot
21	17	COMO Files, Mag Tape

Advanced COMINPUT file usage:

Invocation of one COMINPUT file from another. The file C_PATH invokes C_COMPILE and C_LOAD.

The following is a listing of C_PATH. Lines beginning with * and a space are comments.

```
* C_PATH COMPILES AND LOADS #PATH
* PRIMOS UNIT 7 IS USED BY
* THE NESTED COMPINPUT FILES.
*
CO C_COMPILE 7
CLOSE 7
CO C_LOAD 7
CLOSE 7
DELETE B_SOURCE
CO -END
```

This is the end of the listing of C_PATH. Listings of C_COMPILE and C_LOAD follow. The editor is used to concatenate all routines, so they may be compiled and loaded with single statements.

```
* C_COMPILE FOR PATH
*

FILMEM
ED

LOAD MAIN
LOAD PASSWD
LOAD BEGIN
LOAD DIFSUB
LOAD LEAVE
LOAD ACTIV
LOAD HIT
LOAD SCALE
LOAD FASMOL
LOAD HTVOL
LOAD LIST
LOAD ACTIV
LOAD ASKFIL
LOAD DERIV
LOAD MATPRI
LOAD INVLV
LOAD REMOVE
LOAD SCALE
LOAD MININ
LOAD DECMP
LOAD REFINE
LOAD SETUP
LOAD ADD
LOAD BYE
FILE SOURCE
FOR SOURCE
DELETE SOURCE
CO -CONTINUE
```

The file C_COMPILE has created a binary file, B_SOURCE, which is input to C_LOAD.

```
* C_LOAD FOR PATH
* CREATE #PATH

FILMEM
SEG
LOAD #PATH
LO B_SOURCE
LI ESL
LI VAPPLB
LI
SAVE
MA 3
MA 1
MA M_PATH
QUIT
CO -CONTINUE
```

SOURCE LEVEL DEBUGGER

Typically the way to debug a FORTRAN program is to input print statements and see what values are printed during execution. This usually necessitates repetitive editing, compiling, loading and executing. The source level debugger (DBG) puts an end to that. It allows interactive examining of variables and program flow without changing the FORTRAN source.

To use the debugger,

- compile source code with the -DEBUG option ←
- load normally; (must be a V-mode program; e.g. compiled with FOR and loaded with SEG)
- obtain listings of the source code with the -L option on the SPOOL command (for line numbers of the source code statements)
- create a COMO file (optional)
- invoke the debugger by

  DBG filename

  where "filename" (or pathname) is the name of the executable program being debugged; the debugger responds with a prompt character ">"
- issue commands to the debugger
- terminate the debugging session with

  QUIT

  (control will return to PRIMOS)
- end the COMO file (if used) with

  COMO -END

OK, is PRIMOS control indicator

> is Debug control indicator

OK, COMO SAVE
OK, DBG #TEST
dbg. ~~~~ rev. ~~~~
> BRK SUB1\#
> RST

After having entered DBG, the user must give some commands directing the debugger to monitor some part of the program being debugged. Typically, it will be

- set breakpoints; an area of the program where the user wants to halt execution and examine some variable or path
- cause some variable(s) to be monitored whenever it changes value
- step thru the program instruction by instruction
- trace values, entry/exit to subroutines, or statement number

Once the initial instructions to the debugger are given, the program to be debugged is started executing by

>RESTART or >RST

This command can be given to restart the program at the beginning after execution has begun. (Variables initialized by a DATA statement will not be reinitialized if RST is given after execution has begun.)

When the program has halted after a breakpoint or step, the user can resume execution to the next breakpoint (or end of the program) by typing

>CONTINUE or >C

The user can stop executing and exit from the debugger at any time by typing

>QUIT or >Q

(Commands to the debugger can be given only when the prompt character, ">", appears.)

SOME DBG COMMANDS

Setting Breakpoints

A breakpoint is set in the program where the user wishes to stop execution temporarily to examine something. The command is

>BRK break-id or >BREAKPOINT break-id

where

break-id = the breakpoint identifier for that breakpoint. It consistes of procedure name ($MAIN for main); ;\; line #

e.g.

>BRK $MAIN\9

The program halts at source code line #9. The line # refers to the source code line #'s obtained from the SPOOL -L listing. The line number on the break-id must be an executable line (no Comment lines or FORMAT lines) The halt occurs before that line is executed. In the above example, line #8 would have executed, but not line #9 when the BRK occurs.

A BRK is usually set immediately after entering DBG. The breakpoint is set at a point in the program where things are suspected of going wrong.

BRK's can be set any time one is under control of the debugger (whenever the prompt is given).

After a BRK is set, every time the program executes, the code at the BRK will halt and the user will enter some DBG command or CONTINUE.

BRK's can be cleared (removed) so that a halt no longer occurs there. This is useful if a BRK has been set in a loop and the user no longer wants to halt there. The command to clear BRK's is

>CLEAR break-id or >CLR break-id

where break-id is the same as that in the BRK statement

e.g.

>CLEAR $MAIN\9

To list all the current BRK's type

>LISTALL or >LSTA

Evaluate

Once a BRK has occured, the user usually wants to examine some variable. That is done with the evaluate command, :.

For example,

>: J will evaluate the current value of variable J (the space after : is necessary)

To evaluate a variable as an ASCII string, type

>:A variable

e.g.

>:A TITLE TITLE will be printed as a string of characters rather than as a floating point number.

Expressions can also be evaluated. For example,

>: I+J the sum of integers I and J will be output

When evaluating an array, the range of the array can be specified by typing the beginning and ending value of the index

separated by "...".

For example

given DIMENSION B(1000,1000)

>: B	outputs 1000*1000 numbers
>: B(1,3)	outputs element B(1,3)
>: B(I,J)	outputs element B(i,j)
>: B(1...100,20)	outputs 100 numbers starting with element B(1,20),B(2,20), B(3,20)...B(99,20),B(100,20)

Stepping

The user can step through a section of code instruction by instruction starting from the beginning or from a previously set BRK. This is useful for tracing program flow. The step command is given by

>STEP or >S

and executes one statement every time it is given.

A variation of this command is

>STEP n or >S n

which causes the execution of n consecutive statements before the next halt.

e.g.

>STEP 4

causes the next 4 statements to execute before returing to the DBG command level.

EXAMPLE

```
OK, FOR DRIVER -DEBUG -L
0000 ERRORS [<.MAIN.>FTN-REV17.2]

OK, FOR TEST -DEBUG -L
0000 ERRORS [<TEST  >FTN-REV17.2]

OK, SLIST L_DRIVER
(0001)          DIMENSION C(10)
(0002)          DO 10 I=1,10
```

```
(0003)          C(I)=FLOAT(I)
(0004)       10 CONTINUE
(0005)          DO 30 I=1,10
(0006)          CALL TEST(I,C(I),3)
(0007)          WRITE(1,20) I,C(I)
(0008)       30 CONTINUE
(0009)       20 FORMAT(I5,F10.3)
(0010)          CALL EXIT
(0011)          END
PROGRAM SIZE:    PROCEDURE - 000136    LINKAGE - 000064    STACK
- 000026
0000 ERRORS [<.MAIN.>FTN-REV17.2]

OK, SLIST L_TEST
(0001)          SUBROUTINE TEST(J,V,N)
(0002)          DIMENSION V(100)
(0003)          SUM=0.
(0004)          JN1=J+N-1
(0005)          DO 10 I=J,JN1
(0006)          SUM=SUM+V(I)
(0007)       10 CONTINUE
(0008)          V(J)=SUM
(0009)          RETURN
(0010)          END
PROGRAM SIZE:    PROCEDURE - 000076    LINKAGE - 000026    STACK
- 000036
0000 ERRORS [<TEST  >FTN-REV17.2]

OK, SEG
[SEG rev 17.1]
# LOAD #TEST
$ LO B_DRIVER
$ LO B_TEST
$ LIB VAPPLB
$ LIB
LOAD COMPLETE
$ SAVE
$ QUIT

OK, COMO NUTTER
OK, DBG #TEST

**Dbg**    revision 17.3  (28-January-1980)

> BRK DRIVER\6
No such procedure.
> BRK $MAIN\6
> BRK $MAIN\7
> RST

**** breakpointed at $MAIN\6 ($10+2)
> : I
I = 1
> : C(I)
```

```
C = 1.0
> : C
C(1) = 1.0
C(2) = 2.0
C(3) = 3.0
C(4) = 4.0
C(5) = 5.0
C(6) = 6.0
C(7) = 7.0
C(8) = 8.0
C(9) = 9.0
C(10) = 10.0
> C

**** breakpointed at $MAIN\7 ($10+3)
> : C(1...4)
C(1) = 6.0
C(2) = 2.0
C(3) = 3.0
C(4) = 4.0
> C
   1      6.000

**** breakpointed at $MAIN\6 ($10+2)
> : C(1...5)
C(1) = 6.0
C(2) = 2.0
C(3) = 3.0
C(4) = 4.0
C(5) = 5.0
> : C(1)
C = 2.0
> C

**** breakpointed at $MAIN\7 ($10+3)
> : C(1...5)
C(1) = 6.0
C(2) = 2.0
C(3) = 12.0
C(4) = 4.0
C(5) = 5.0
> RST?BRK TEST\5
> BRK TEST\6
> BRK TEST\8
> LSTA
Type  Location
brk   $MAIN\6 ($10+2), count = 2
brk   $MAIN\7 ($10+3), count = 2
brk   TEST\5, count = 0
brk   TEST\6, count = 0
brk   TEST\8 ($10+1), count = 0
> RST

**** breakpointed at $MAIN\6 ($10+2)
```

```
> : I
I = 1
> : C(1...5)
C(1) = 1.0
C(2) = 2.0
C(3) = 3.0
C(4) = 4.0
C(5) = 5.0
> C

**** breakpointed at TEST\5
> : J
J = 1
> : JN1
JN1 = 3
> C

**** breakpointed at TEST\6
> : V(1)
V = 1.0
> C

**** breakpointed at TEST\6
> : V(I)
V = 2.0
> : SUM
SUM = 1.0
> C

**** breakpointed at TEST\6
> : V(I)
V = 3.0
> : SUM
SUM = 3.0
> C

**** breakpointed at TEST\8 ($10+1)
> : SUM
SUM = 6.0
> C

**** breakpointed at $MAIN\7 ($10+3)
> : C(1...5)
C(1) = 6.0
C(2) = 2.0
C(3) = 3.0
C(4) = 4.0
C(5) = 5.0
> C
     1      6.000

**** breakpointed at $MAIN\6 ($10+2)
> : I
I = 2
```

```
> : C(1...5)
C(1) = 6.0
C(2) = 2.0
C(3) = 3.0
C(4) = 4.0
C(5) = 5.0
> : C(I)
C = 2.0
> C

**** breakpointed at TEST\5
> : J
J = 2
> : JN1
JN1 = 4
> : V(1)
V = 2.0
> C

**** breakpointed at TEST\6
> : V(I)
V = 3.0
> : I
I = 2
> C

**** breakpointed at TEST\6
> : V(1)
V = 4.0
> : SUM
SUM = 3.0
> C

**** breakpointed at TEST\6
> : V(I)
V = 5.0
> C

**** breakpointed at TEST\8 ($10+1)
> : SUM
SUM = 12.0
> QUIT

OK, COMO -END
```

REFERENCES

1. The Source Level Debugger Guide; IDR 4033 (Chapters 1,2,5 and Appendix A,B).

ASCII FORTRAN IV Compiler

Some Definitions

source code	An text file consisting of FORTRAN IV statements; usually created with the EDitor.
binary file	Translation of a source code file generated by a compiler and in a format required by the loader
short integer	integer using 16 bits (1 word or 2 bytes); with a range of -32768 to +32768
long integer	integer using 32 bits (2 words or 4 bytes); with a range of -2147483648 to +2147483648
R-mode	executes in physical real memory space; program size limited by size of one segment (64K words)
V-mode	executes in virtural memory space; program size can be larger than amount of physical memory

---

There are two versions of the same compiler on the PRIME.

FOR
- (system standard; recommended)
- uses long integer and V-mode as default options
- all application libraries are compiled using this version

FTN
- uses short integer and R-mode as default options

A source code file is compiled by typing.

FOR filename

The compiler generates a binary file called "B_filename" that will reside in the current UFD (or sub-UFD). Error messages from the compilation will be printed at the user terminal.

The compiler has a number of options which are used to override the system defaults or produce additional information. Some of the most frequently used options are listed below.

-L or -LISTING	creates a listing file called "L_filename", which contains source code and error messages (-L SPOOL directs the listing file to the printer immediately and deletes it from the UFD)
-BIG	used whenever the source code has arrays larger than 64K octal words (64K octal = 64*1024 decimal=65,536) Long integers and real numbers are 2 words long, so arrays larger than 32,768 real #'s or integers require the use of this option.
-DEBUG	generates appropriate code for using the source level debugger (DBG)

The format for using options on the compiler is

FOR filename option1 option2 . . .

e.g.

FOR filename -BIG -DEBUG

Note that the "-" is part of the option name.

EXAMPLE

The following source code file is named TEST.

```
      DIMENSION A(10)
      DO 10 I=1,10
      A(I) = 10.0*SIN(FLOAT(I))
   10 CONTINUE
      WRITE(1,400) (A(J),J=1,10)
      CALL EXIT
      END
```

The results of compilation are

```
OK, FOR TEST -DEBUG -L
(0007)          END
**** LINE 0005  [ END ]  $400 - UNDEFINED STMT NO.
0001 ERRORS [<.MAIN.>FTN-REV17.2]
```

The message refers to the forgotten FORMAT statement referenced in line 5.

The format of the compiler error messages is

```
xxxx ERRORS [<YYYYYY>FTN-REVzz.z]
```

where

xxxx = # of errors; 0000 indicates no errors

YYYYYY = .MAIN. for a main routine
subroutine name for a subroutine (or function)

zz.z = PRIMOS rev number

REFERENCES

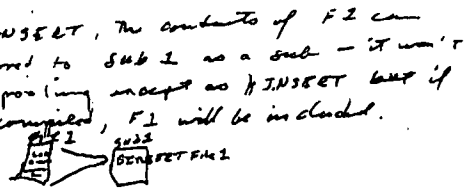
1. The FORTRAN Programmers Guide; Rev.16; PRIME FDR 3057-101.

2. The FORTRAN Programmer's Companion; Rev16; PRIME FDR 3338.

PRIME Extensions to FORTRAN IV

Some extensions to ASCII FORTRAN IV on the PRIME are listed below.

$INSERT



The form of the command is

```
$INSERT filename
```

and starts in column 1. It inserts into the source code, at compilation time, the file whose name is "filename" (or pathname). The $INSERT command should not be nested. It is commonly used for COMMON block specification or blocks of PARAMETERS.

For example, filename COM contains

```
      COMMON / BLOCK1/ A(10,20),B(10,20),C(10,20)
      COMMON / BLOCK2/ E(50),AST(2)
```

and filename DRIVER has the following code,

```
C-----DRIVER PROGRAM
      DIMENSION W(100),T(100)
$INSERT COM
```

results in the following code after compilation

```
C-----DRIVER PROGRAM
      DIMENSION W(100),T(100)
      COMMON / BLOCK1/ A(10,20),B(10,20),C(10,20)
      COMMON / BLOCK2/ E(50),AST(2)
```

PARAMETER statement

The PARAMETER statement allows a constant to be assigned to a variable name. The variable can be used any place a constant can (except in FORMAT statements). This is handy for use in DIMENSION statements. If changing the size of arrays, only the PARAMETER value needs to be changed

For example,

```
      DIMENSION A(100),B(100),C(100),D(100)
      DIMENSION E(100,100), F(100,100)
```

can be replaced with

```
      PARAMETER N=100
      DIMENSION A(N),B(N),C(N),D(N)
      DIMENSION E(N,N),F(N,N)
```

FREE FORMAT I/O (called List-directed READ in manuals)

Free format frees the programmer from including format statements for READs from free format input devices such as the user terminal. Values in a free format input are separated by a comma or a carriage return. A free format READ will read any number of records until all items in the I/O list have been satisfied or a slash is encountered. The format statement # is replaced by a "*" in the READ statement.

For example,

```
source line:   READ(1,*) A,B,C
input data:    151.,0.,2.0E2
results in:    A=151.
               B=0.
               C=2.0E2
```

you can insert text if it is enclosed with single quotes 'text'

Generalized Subscripts

The compiler allows any integer-valued expression as an array subscript. It may contain constatns, variables, function references or other array references.

Some Additional Comments

-PRIME FORTRAN has no COMPLEX DOUBLE PRECISION type (UNIVAC has)

-READ/WRITE unit numbers for terminals is "1" (standard FORTRAN uses 4 or 5)

-use the CALL EXIT statement instead of STOP

REFERENCES

1. The FORTRAN Programmers Guide; Rev.16; PRIME FDR 3057-101.

2. The FORTRAN Programmer's Companion; Rev16; PRIME FDR 3338.

FORTRAN PROGRAMMING MANUALS

1. A FORTRAN IV Primmer, E. I. Organick, Addison-Wesley.

2. A Guide to FORTRAN IV Programming, D. McCracken.

Wiley.

3. A Short Course in FORTRAN IV Programming, R.M. Lee, McGraw-Hill.

COMO files

There is a PRIMOS command called

COMOUTPUT pathname options

affectionately known as

COMO pathname options

that sends the output stream to file "pathname" (also to the terminal unless told not to). If the output file is a filename instead of a pathnmae, it will be placed in the current UFD.

This action is halted by typing

COMO -END

Usually this command is used when a hardcopy (or listing) of the terminal session is desired for later reference.

This can be a very useful command when using the Source Level Debugger, especially when debugging programs with many large arrays whose values need to be examined.

For example,

```
OK, COMO filename
OK, DBG
>
```

stick to one name for como filename because it usually is looked at only once + deleted, it prevents overwriting another file that has the same name

```
>QUIT
OK, COMO -END
```

<- matlisa. CLOSE-ALL doesn't work here

At this point, the file can be SPOOLed with

SPOOL filename

or examined with the editor

```
ED filename
P 23
```

To get rid of a COMO file

OK, DELETE filename

```
QUIT
```

SPOOLing the file does not delete it from the UFD.

COMO files are usually used to obtain hard copy of the terminal session (or part of it) and are not generally saved. After SPOOLing it, it should be deleted.

REFERENCES

1. PRIMOS Commands Reference Guide; FDR 3108.

2. The Programmer's Companion -- PRIMOS Commands; FDR 3250-000

UURI/ESL TAPE HANDLING

Policies:

All tape handling must be done through one of the computer staff. Exceptions are those who have been trained and given special permission by Terry Killpack. Without this permission no one is to go into the machine room. If you need a tape mounted, one will be mounted for you by John Atwood or one of the other computer staff. Data backups are done entirly by computer staff. This can be accomplished by submitting an ACTION REQUEST FORM to Terry Killpack with the proper information.

Procedure:

After a tape has been mounted for you, the tape drive must be assigned to the terminal at which you are logged in. This is done be entering the command:

AS MTO

The computer will then respond with a message stating that the unit has been assigned, or that it is in use.

Tape Writing:

Tape writing is accomplished by one of two methods depending on the tape format. If the tape is to be written as a backup tape or if it is to be used to transfer data to another Prime Computer, MAGSAV should be used. If the tape is to be used on anything other than a Prime, then MAGNET must be used with the WRITE option.

Tape Reading:

Tape reading is accomplished by one of two methods depending on the tape format. If the tape has been written using MAGSAV, then MAGRST must be used. If the tape has been written by anything other than Prime's MAGSAV utility, then MAGNET must be used with a READ option.

Finishing:

When finished with the tape, the user should either log out, or if there is to be more work done in the session, the user should release the tape drive with the command:

UN MTO

MAGRST / MAGSAV

labled, unformatted tape read / write utilities.

▶MAGRST [-7TRK]

Restores files, directory-trees or partitions from a magnetic tape created with MAGSAV. The tape unit must be ASSIGNED. The MAGRST dialog is as follows:

Query	Possible Responses
TAPE UNIT:	(0-7) Physical unit number
ENTER LOGICAL TAPE NUMBER:	1 = 1st logical tape, 2 = 2nd etc. Rewinds and positions tape. 0 = tape already positioned
READY TO RESTORE:	YES (Restore entire tape) NO (Request new tape drive and logical tape numbers) PARTIAL (Restore part of tape) $I [filename] n cause subsequent partial or entire restore to print n-level index at terminal or into optional filename. NW [filename]n print n-level index into file or at terminal but do not restore anything to disk. $A ufd [passwd] ldisk Change home UFD. Pathnames not allowed; ldisk number required.
TREE NAME:	Enter treename for partial restore.

▶MAGSAV [options]

Writes a PRIMOS disk file, directory-tree or partition to 7 or 9-track magnetic tape. The tape unit must be ASSIGNed.

Options:	
-LONG	1024-word records (default=512)
-INC	Incremental dump
-UPDT	Set DUMPED bit
-7TRK	7-track tape format (default=9 track)

The MAGSAV dialog is as follows:

Query	Possible responses
TAPE UNIT:	(0-7) physical tape unit
ENTER LOGICAL TAPE NUMBER:	1 = 1st logical, 2 = 2nd, etc. Rewinds and repositions tape. 0 = already positioned.
TAPE NAME:	Any name up to 6 characters.
DATE:	Valid date in format mm dd yy. (default = current date from PRIMOS)
REV. NO:	Any number.
NAME:	filename to be saved or one of the following: $Q Terminate logical tape and return to PRIMOS. $R Do $Q and rewind tape. $I [filename]n Index to n levels. MFD Save entire disk. (Must be attached to MFD) * Save contents of current directory. $A ufd [passwd] ldisk Change home UFD. Pathnames not allowed; ldisk number required.

MAGNET:
unlabled, formatted
tape read/write
utility.

▶MAGNET

MAGNET is an interactive magnetic tape utility that transfers non-Prime-format magnetic tapes to and from PRIMOS disk files. MAGNET prints a release number and a date and then requests the user to enter an option. The user may then issue one of the option commands. Each option conducts a dialog. The queries and responses are described below.

POSITION
Positions the tape to a specified file and record number. Absolute position rewinds tape before spacing. Relative position allows tape to be moved forward or backward from the current position.

READ
Transfers a file from magnetic tape to disk, optionally providing unblocking and EBCDIC or BCD translation.

WRITE
Similar to READ but transfers file from disk to tape. WRITE also provides facilities for blocking and character translation.

COPY
Copies a file (or files) from one magnetic tape to another. No character translation is provided for this operation.

QUIT
Returns to PRIMOS command level.

MAGNET DIALOG QUERIES

Prompt	Response
LIST OPTIONS? OR WHAT?	YES to see list of options, or NO.
MTU #	Physical unit number followed by /7 (seven-track) or /9 (nine-track - default).
RELATIVE OR ABSOLUTE?	ABSOLUTE: file and record numbers are relative to beginning of tape. Must be positive.
	RELATIVE: file and record number are relative to current position. May be positive (forward) or negative (backward). 0 = current position.
FILE # OR MT FILE	Absolute file number or relative number of records to forward or backspace.
LOGICAL RECORD SIZE	Number of bytes in a disk record. Must be 10,000 or less.
BLOCKING FACTOR	Number of logical records (line images) in one tape record.
ASCII, BCD, BINARY OR EBCDIC?	Type of translation desired between tape and disk. ASCII = none, BCD = BCD to ASCII, etc.
FULL OR PARTIAL?	FULL = translate entire line. PARTIAL = translate specified fields.
ENTER PAIRS OF ... etc.	Enter starting/ending column numbers, one pair per line. Terminate with CR.
OUTPUT FILENAME	Disk file to be created during READ.
INPUT FILENAME	Disk file used as source of WRITE.
'FROM' tape	MTU # of tape to be copied.
'TO' tape	MTU # of tape to receive copy.

UURI
ESL
COMPUTER

ACTION REQUEST FORM

REQUESTOR: _____ DATE: _____

UURI ACCOUNT: _____

URGENCY: _____

GENERAL DESCRIPTION:

ASSIGNMENT: _____

COMPLETED: _____

UURI/ESL Computing Facility Guide

by

Terry J. Killpack

March 1981

Table of Contents

COMPUTING FACILITIES

There are three rooms located in the building at 420 Chipeta Way that contain the computing facilities for UURI/ESL. The rooms are called the machine room, the peripheral room and the user room. The machine room contains the Prime 400 CPU, memory, disks, tape drive, and system terminal. Users are not generally allowed access to the machine room. The peripheral room contains various peripherals such as the line printer, the Statos electrostatic plotter, the Zeta pen plotter, a DOE Tektronix 4014 terminal and a DOE Tektronix 4081 system. Users are allowed access to this room during working hours and some ESL users working on DOE contracts are given access to the peripheral room during nonworking hours. The user room contains a CRT terminal, a Tektronix 4014 graphics terminal with a tablet and a DECwriter terminal. Users are given access to the user room 24 hours a day. The combination to the user room door is II and IV pushed together and then III.

After hours (after 5 pm) the last user to exit the user room must pull the door closed, securing the room with the combination lock. The normal lock should not be used. The power on the Tektronix 4014 graphics terminal should also be turned off before securing the user room.

No food or drink should be taken into any of the three computer facility rooms at any time. Smoking should be avoided in the machine room.

There are three telephone lines into the computer that may be accessed 24 hours a day from remote sites. There are two 300 BAUD lines (581-5402 & 6711) and one 1200 BAUD line (581-6645).

Tektronix Graphics Terminals

The UURI/ESL computer facility has several Tektronix storage tube terminals. These terminals require some special care in their use. The screens on these storage tube terminals can become permantly damaged (burned) if certain steps are not taken to protect them. After loging out of a job on a Tektronix terminal the user should erase all writing from the screen before leaving the terminal. This is done by pressing the page reset key on the keyboard of the terminal. The screen should not be overwritten. The programmer or user should not allow vectors or characters to be drawn in the same place on the screen over and over again. This can burn the screen. The power on the Tektronix terminals should be turned off if the room that they are in is being secured for the night.

PLOTTING SYSTEMS

Introduction

The UURI/ESL computing facility has several plotters that are available for use in application programming or production jobs. The UURI/ESL Prime system has a device-independent plotting system that can be used to direct plots to any of three devices. The plotting system is interfaced to the Tektronix 40XX graphics terminals, the Statos electrostatic plotter, and the Zeta pen plotter. The UUPLT manual (ESL-46) contains information on how to direct plots to different devices. The basic subroutine that is used to open plots and direct them to a specific device is called OPNPLT. There are other routines that can be used to open plots and they are discussed in the plot manual.

Tektronix plots are generated directly and the user must be logged into a Tektronix terminal to perform Tektronix plotting. The Statos and Zeta plots are generated indirectly by sending plot files to plot queues. When a Statos plot is generated the system sends a message to the user terminal that a statos plot has been generated and that it contains a certain number of vectors. This plot will not actually be plotted until the user powers on the Statos plotter and then sends the plot to the Statos queue with the SPLOT command. When a Zeta plot is generated the system sends a message to the user terminal that a zeta plot has been generated and that it contains a certain number of pen movement inches. This plot is actually produced by using the ZPLOT command and then waiting for the system operator to process the Zeta plot queue. The Zeta plot queue will be processed once a day.

Statos Plotting

The Statos electrostatic plotter is controled by a system phantom job that continually processes the Statos queue much like the system spooler processes print files. Plots are sent to the queue with the SPLOT command. Plot files are generated by user or production programs using the UUPLT library subroutines.

SPLOT Command

Plots are sent to the Statos plotter with the SPLOT command much like print files are sent to the spool queue with the SPOOL command. However, the SPLOT command does not require a file name like the SPOOL command.

When the user types the SPLOT command at the user terminal the system looks for any Statos files that have been generated under the current login name and if any are found, the files are placed on the Statos plot queue where the Statos phantom job will process them on a first come first serve basis.

The user may obtain a summary of the plots that are on the queue by using the LIST token, (SPLOT -LIST). This command will produce a list of the plots that are on the Statos queue (plots that have been "SPLOTed") at the user terminal.

If the phantom job has not started processing the user's plot, it is possible to cancel the plot from the queue with the cancel token, (SPLOT -CANCEL PLSXXX). PLSXXX is the queue name of the plot that was given to the user at the time the SPLOT command was used and is the name that is displayed with the LIST token.

If the user generates a plot that has some known problem, it is possible to delete the the plot file from the system before it is sent to the queue by using the SPLOT command with the DELETE token, (SPLOT -DELETE). The system will respond with a verification request and the user answers yes or no.

The default "pens" on the Statos plotter are vector thicknesses of 1, 3, 5, and 7. The PENSEL or THICK subroutines in the UUPLT library may be used to seclect vector thicknesses.

Zeta Plotting

The Zeta pen plotter is controlled by operator software that processes the Zeta plot queue. Once a day the system operator will used the operator software to actually produce all the zeta plots that have been sent to the queue with the ZPLOT command. Plot files are generated by user programs or production jobs that use the UUPLT subroutines in the plot library.

ZPLOT Command

Plots are sent to the Zeta queue with the ZPLOT command. No file name is required.

When the user types the ZPLOT command at the user terminal the system looks for any Zeta files that have been generated under the current login name and if any are found, the files are placed on the Zeta queue where the operator software will process them.

The user may obtain a summary of the plots that are on the queue by using the LIST token. Zeta plots may be canceled with the CANCEL token and they may be deleted with the DELETE token.

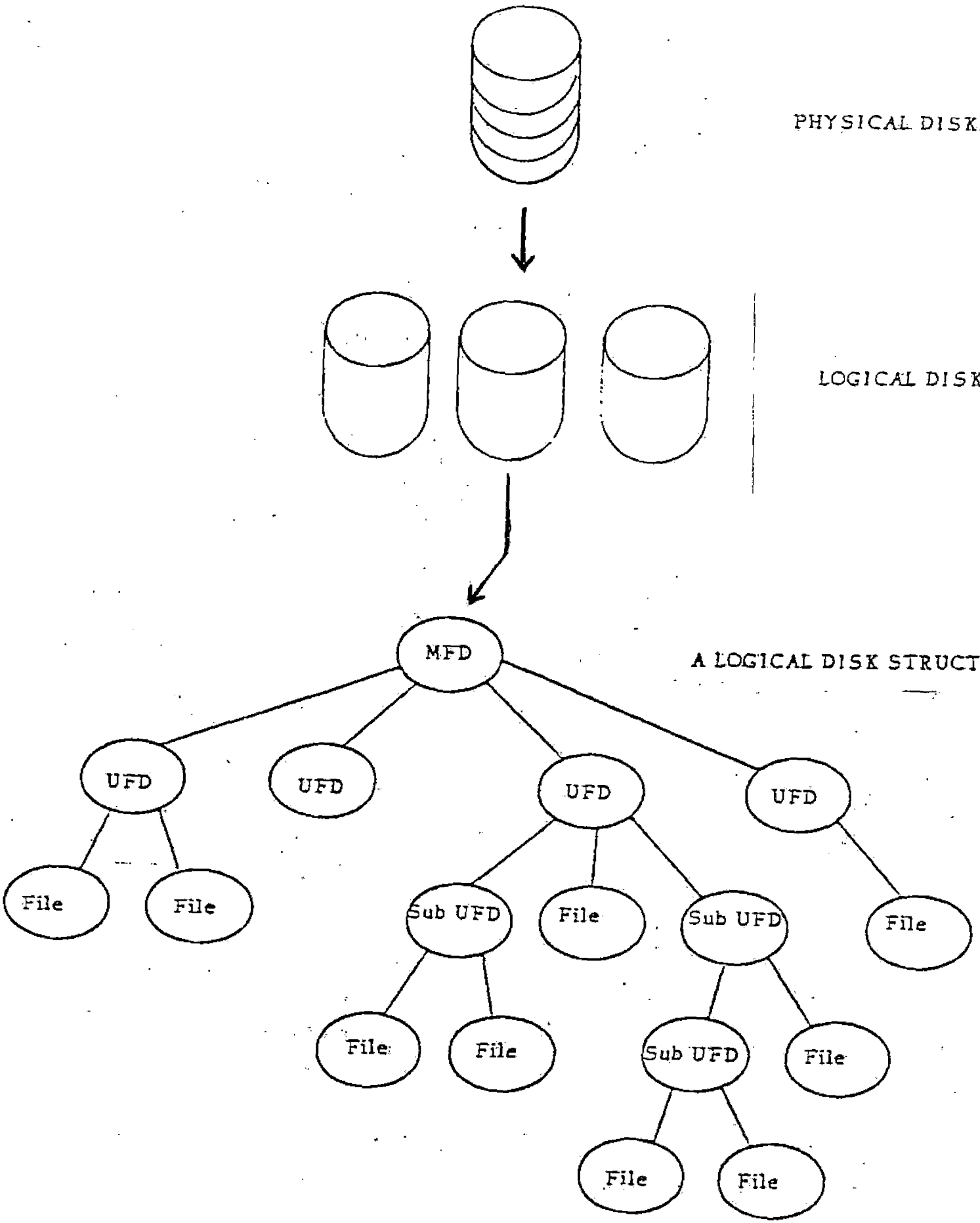
The default pens on the Zeta plotter are ball point. Turret number 1 contains black, 2 contains red, 3 contains blue, and 4 contains green. The pens may be changed by sending an operator request using the CALOPR subroutine in the UUPLT library. There are extra charges for operator requests.

FILE STRUCTURE

There are two types of files on the PRIME. One type is a directory; a file that contains information about and pointers to other files. The other type is a file that contains information (usually a data file or a program file).

Directory files:	MFD	(Master file directories)
	UFD	(User file directories)
	SUFD	(Sub User file directories)
	Seg. Dir.	(Segment directories; SEGSAM files)
Information files:	SAM	(Sequential Access)
	DAM	(Direct Access)

Directory files will be referred to as "directories" and information files will be referred to simply as "files".


PHYSICAL DISKS
LOGICAL DISKS
A LOGICAL DISK STRUCTURE
MFD
UFD
UFD
UFD
UFD
File
File
Sub UFD
File
Sub UFD
File
File
File
Sub UFD
File
File
File

SAM files

SAM files have records that must be accessed sequentially. Some examples of SAM files are

- source code files (created by ED)
- binary files (created by FOR)
- map files (created by SEG MAP)
- listing files (created by FOR)
- CO files (COMINPUT)
- data files (created by ED)

DAM files

DAM files have records that can be accessed randomly or directly without having accessed the previous records. Some examples of SAM files are

- COMO files (COMOUTPUT)
- direct access files (from FORTRAN DEFILE subroutine)

SEGSAM files

SEGSAM files are the segment directories created by the SEG loader. These segment directories contain information about where the run image is stored.

ESL FILE NAMING CONVENTIONS

---

source files	filename
binary files	B_filename
listing files (from FOR)	L_filename
map files (from SEG)	M_filename
Runfiles (from SEG)	#filename
COMINPUT files	C_filename
COMOUTPUT files	filename
Phantom command files	PH_filename
RUNOFF files	R_filename

PRIMOS FILE COMMANDS

The following is a list of PRIMOS commands dealing with files (data/program and directories).

Command	Usage Example	Purpose
ATTACH	A UFD.NAME A *>SUB A SUB 1 2	Change current directory
CLOSE	C ALL pathname funit	Close all files Close file called "pathname" Close file open on PRIMOS unit "funit"
CMPF	CMPF FILE1 FILE2	Compare two files and show differences
CNAME	CNAME OLDFILE NEWFILE	Change filename
COMINPUT	CO C_STUFF	Execute a command file
COMOUTPUT	COMO LOOKY	Create a copy of the output stream
CREATE	CREATE NEW.SUBUFD	Create a new SubUFD
DELETE	DELETE FILENAME	Delete a file
ED	ED FILE1	Edit a file
FOR	FOR FILE1	Compile source file
FUTIL	FUTIL	Invoke the File UTILity processor
LISTF	LISTF	List current directory (gives name of current directory)
LUFD	LUFD	List current directory
OPEN	OPEN DATA1 8 1	Open file on PRIMOS file unit
PH	PH PH_JOB	Execute a phantom

		job
RUNOFF	RUNOFF	Invoke the RUNOFF processor
SEG	SEG #PROG SEG	Execute a runfile Invoke the SEG loader
SIZE	SIZE PROGRAM.NAME	Display size of file
SLIST	SLIST SOURCE	Display a source file
SPOOL	SPOOL FILE1	Produce line printer listing of a file

FUTIL PROCESSOR

FUTIL is a file utility whose subcommands permit the user to copy and delete files and directories on a per-file or directory basis. FUTIL has an ATTACH subcommand that allows attaching to subdirectories by giving a pathname.

INVOKING FUTIL FROM PRIMOS

To invoke FUTIL, input the command

FUTIL

When loaded, FUTIL prints the prompt character ">", and awaits a command from the user terminal.

FUTIL COMMANDS

ATTACH directory-pathname

ATTACH moves the home UFD to the directory defined by pathname. The first directory in the pathname may be *.

TO pathname

TO defines the TO directory for subsequent copying commands. The pathname in the command is where things will be copied to. The default TO directory is the home UFD. The TO command does not change the home UFD.

FROM pathname

FROM defines the FROM directory for subsequent copy and delete commands (e.g., COPY, DELETE, CLEAN, TRECPY, TREDEL, UFDCPY, and UFDDEL). The pathname in the command is where things are copied or deleted from. The default FROM directory is the home UFD. Use of FROM never changes the home UFD.

The FROM (and TO) directories remain in effect until changed. Be careful when deleteing files or directories after having set the FROM directory to something other than the home UFD.

e.g.,

```
OK, FUTIL
>FROM YOURS
>TO MINE
>COPY YOURFILE
>DELETE FILEA
```

Results in copying file YOURS>YOURFILE to MINE>YOURFILE and deleting YOURS>FILEA if it exists. The probable intention was to delete MINE>FILEA, but the FROM directory was not updated.

COPY from-name [to-name]

Copies the named SAM or DAM file from the FROM directory to the TO directory. If the "to-name" is ommitted, the copies have the same name. If the "to-name" already exits in the TO directory, it is deleted and then "from-name" is copied to "to-name". The type of the two files will be the same.

e.g. >COPY TEST TEST-A

TRECPY directory-a [directory-b]

Copies directory tree in FROM directory into the TO directory. A directory tree consists of all files and sub-directories that have their root in that directory. "directory-b" must not exist prior to the TRECPY command. If "directory-b" is omitted, the two directories have the same name. Use this command for copying SEGSAM (runfiles) files.

UFDCPY

This command copies all the files and directory trees from the FROM directory to the TO directory. Files already existing in the TO directory with names identical to those in the FROM directory will be overwritten. SEGSAM files already exiting in the TO directory with names identical to those in the FROM directory are not allowed and will not be copied.

CLEAN prefix

Deletes all files in the FROM directory beginning with "prefix". This command is usually used to clean all binary files from a UFD by

CLEAN B_

DELETE filea [fileb] ...

This command deletes "filea" and optionally, "fileb", from the FROM directory. "filea" and "fileb" cannot be directories.

e.g., DELETE FILEA FILEB

or

DELETE FILEA
DELETE FILEB

TREDEL directory-a

This command deletes the directory tree specified by "directory-a" from the FROM directory. This command is used for deleting SEGSAM runfiles. SEGSAM runfiles are preferably deleted by using the DELETE command in the SEG processor.

UFDDEL

This command deletes the entire FROM directory.

QUIT

This command terminates the FUTIL processor.

Passwords, owner, read and write access rights are required for some of the FUTIL commands to execute.

OPENING AND CLOSING FILES

Data files used by FORTRAN IV programs must be explicitly opened and closed by the program or by the user.

UNDER PRIMOS CONTROL

OPEN pathname funit key

where

pathname - file to be opened

funit - PRIMOS file unit on which pathname is opened (note: PRIMOS file unit is not the same as the FORTRAN logical unit)

key - logical OR of Newfile key, Action key, and Reference key

Newfile keys:	0000	- for SAM files
	2000	- for DAM files
Action keys:	1	- open for reading
	2	- open for writing
	3	- open for reading and writing
	4	- close
	7	- rewind
Reference keys:	0	- file is in current UFD

(incomplete list - complete list is in PRIMOS Commands manual under OPEN command)

For example,

```
OPEN DATA1 8 1
```

means open, on PRIMOS unit 8 (FORTRAN logical unit 12), a file named DATA1 as a "read only" SAM file

(OPEN can be abbreviated by "O")

```
CLOSE   pathname
        funit
        ALL
```

where

pathname - name of a previously opened file

funit - PRIMOS file unit that is currently opened

ALL - all files currently opend

For example, to close the file opened in the example above, any of the following forms would do

```
CLOSE ALL

CLOSE 8

CLOSE DATA1
```

(CLOSE can be abbreviated by "C")

FROM WITHIN A FORTRAN PROGRAM

There are several subroutines in the library ESL (UURI) that will open and close files from within a FORTRAN IV program. To access these subroutines, the UURI library must be loaded by the SEG loader, so the statement

```
LIB UURI
```

must be included when loading the program.

OPNNEW

OPNNEW opens or closes a sequential write file. This subroutine actually opens the file for reading and writing.

OPNNEW checks for the existence of the file and if the file exists it is closed and then deleted before a new file is opened (created).

Usage

```
CALL OPNNEW(LU,NAME,LN)
```

Description of parameters

LU- (int.) FORTRAN logical unit
NAME- alphanumeric name of file
LN- (int.) # of characters in NAME

OPNWRT

OPNWRT opens a sequential write file.
The file is actually a read and write file.

Usage

CALL OPNWRT(LU,NAME,LN)

Description of parameters

LU- (int.) FORTRAN logical unit
NAME- alphanumeric name of file
LN- (int.) # of characters in NAME

OPNRED

OPNRED opens a sequential read file.
OPNRED checks for the existence of the file.

Usage

CALL OPNRED(LU,NAME,LN)

Description of parameters

LU- (int.) FORTRAN logical unit
NAME- alphanumeric name of file
LN- (int.) # of characters in NAME

CLOSEF

CLOSEF closes a FORTRAN logical unit number

Usage

CALL CLOSEF(IU)

Description of parameters

IU- (int.) FORTRAN logical unit

SOME ADDITIONAL FILE UTILITY ROUTINES FROM
THE UURI (ESL) LIBRARY FOR USE IN FORTRAN PROGRAMS

The following are some useful subroutines for file handling from within a FORTRAN program.

DEFILE (DEFINE FILE)

Create or access DAM file in user UFD

Usage

```
CALL DEFILE(IUNIT,FNAME,NC,LR4,ISTAT)
```

Description of parameters

IUNIT	FORTRAN logical unit
FNAME	file name
NC	# of characters in FNAME
LR4	fixed length of records (in 4 BYTE words)
ISTAT	status returned 0 = existing file opened for RD/WT 1 = new file opened for RD/WT 2 = unit already open 3 = unsuccessful open 4 = file in use

This subroutine is used to create the work and merge files associated with some of the interactive programs residing on the PRIME, such as IP2D, MT2D, etc.
These files cannot be examined with the EDITOR or be printed with SPOOL.

DELE

DELE deletes a file given the file name and the number of characters in the name.

Usage

```
CALL DELE(NAME,LN)
```

Description of parameters

NAME	(alphanumeric) the name of the file that is to be deleted.
LN	(int.) # of characters in NAME

SPOUT

To SPOOL a file from a FORTRAN program

Usage

```
CALL SPOUT(NAME,LNAME,FTN)

LIB VSPOO$
```

Routine called

SPOOL$ (VSPOO$)

Description of parameters

NAME	=	filename for spooling
LNAME	=	# of characters in NAME
FTN	=	LOGICAL, TRUE if FORTRAN FORMS CONTROL

BASIC EDITOR COMMANDS

The following commands are the simplest and most used of the EDITOR commands:

ED file-name
: Where file-name represents the name of the file to be edited. Command is given while in PRIMOS to start the editing session. If no file name is given, a new file is created.

<CR>
: Carriage Return is used to set the mode from INPUT to EDIT, or from EDIT to INPUT. Either mode is set to the other mode by pressing the RETURN key after a null line (entering two <CR>s in a row), or by pressing the RETURN key immediatly following a semi-colon. While in the INPUT mode each line is accepted as data until a change mode sequence is encountered While in the EDIT mode, each line is expected to begin with one of the valid EDITOR commands.

INSERT string
: Where string represents the characters in the line to be inserted following the current line. The INSERT command may be abreviated to I.

DELETE
: The DELETE command is used to delete the current line. The DELETE command may be abreviated to D.

  Aternate uses of the DELETE Command are:

  D n
  : Where n is an integer number designating the number of lines including the current line, to be deleted.

  D TO string
  : Where string represents a string identifying the first line following the current line not to be deleted.

LOCATE string
: Where string represents the text to be located. Checking is performed below the current line only. The LOCATE command may be abreviated to L.

FIND(n) string
: Where n represents the column in which string is required to begin, and string represents the text to be searched for. The search is performed below the current line only. If (n) is left off of the command, the column is assumed to be column one. The parenthesies are required when specifying a column number. The FIND command may be abreviated to F.

NFIND(n) string

Where n represents the column in which string is required to begin, and string represents the text to be searched on. NFIND searches for the first line below the current line which does not contain the specified string in the specified column. If (n) is left off of the command, the column is assumed to be column one. The parenthesies are required when specifying a column number. The NFIND command may be abreviated to NF.

PRINT n

Where n represents the number of lines including the current line, to be printed. The PRINT command may be abreviated to P.

WHERE

WHERE is used to query the editor for the current line number. The WHERE command may be abreviated to W.

TOP

TOP is used to position the current line pointer at the top of the file. The TOP command may be abreviated to T.

BOTTOM

BOTTOM is used to position the current line pointer at the bottom of the file. The BOTTOM command may be abreviated to B.

NEXT n

Where n is the number of lines to move the current line pointer up or down. If the number is negative, the pointer is moved up. If the number is positive, the pointer is moved down. If no number is specified, the pointer is moved down one line. The NEXT command may be abreviated to N.

POINT n

Where n represents the line number to position the current line pointer. n must be specified, and must be greater than 0. The POINT command may be abreviated to PO.

CHANGE /string-1/string-2/ G n

Where / can be any character not included within either string, and string-1 represents the present string, and string-2 represents the string to which string-1 is to be changed. If the optional parameter G is specified, all occurances of string-1 will be changed, otherwise only the first occurance will be changed. n represents an optional integer specifying the number of lines including the current line, to be changed. The CHANGE command may be abreviated to C.

APPEND string

Where string represents the string of characters to be appended to the end of the current line. The APPEND command may be abreviated to A.

RETYPE string

Where string represents the text to replace the current line. The RETYPE command may be abreviated to R.

UNLOAD file-name

Where file-name represents the file to which the current line is to be copied. The UNLOAD command may be abreviated to U.

Two variations of the UNLOAD command are also available. These are:

U file-name n

and

U file-name to string

The explanations are as with DELETE above.

LOAD file-name

Where file-name is the name of a file to be loaded after the current line. The LOAD command may be abreviated to LO.

FILE file-name

Where file-name is the name of the file to place the contents of the editing session. If file-name is omitted, the file is placed over the file from which it was read. If no file-name was specified upon entry to the EDITOR, one must be specified in the FILE command. The FILE command may be abreviated to FIL.

QUIT

QUIT is used to exit from the EDITOR without saving the contents of the EDITOR session.

PAUSE

PAUSE is used to temporarily leave the EDITOR without saving the EDITOR work space. This allows you to check your UFD for duplication of names before filing. To return to the EDITOR, type in START. The command PAUSE may be abreviated to PA, and the START command may be abreviated to S.

!!!!! WARNING !!!!! You will destroy the work space if you type ED again, or if you log out, or if you type in any command other than an internal PRIMOS command. To check for duplicate file names use LISTF. Do Not Use LUFD. !!!!!

The following symbols have special meaning to the editor:

" is the erase character symbol. To input a " into the text you must precede the " with the character ^.

? is the kill line symbol. To input a ? into the text you must precede the character ? with the character ^.

; is used by the EDITOR to signal the end of a command line. This allows the user to input multiple commands on the same line. To input a semi-colon into the text you must input some other character and then use the CHANGE command to change it to the semi-colon. If the semi-colon is encountered while in the INPUT mode, the text following the semi-colon will be input on a new line. If a semi-colon is encoutered while in the EDIT mode in any command other than the CHANGE command, the EDITOR will expect another command to follow.

For more information about these and other commands, see the PRIME manual entitled: The New User's Guide to EDITOR and RUNOFF. This manual is located in the manual rack in the user's room.

****

****

The following is a sample session with the editor. I have added comments to the right of commands where needed. Lines preceded with a ">" are user input lines; all others are computer response.

****

```
   OK, ED              [PRIMOS prompt and command to enter EDITOR.]
   INPUT
> A
> B
> C
> D
> E
> F
> G
> G
> I
>                      [Input a RETURN to change to EDIT mode.]
   EDIT
> T
> P 23
   .NULL.
   A
   B
   C
   D
   E
   F
   G
   G
   I
   BOTTOM
> W
   LINE  10
> PO 1
   A
> PO 3
   C
> L A
   BOTTOM              [The "A" is above the current line.]
> T
> L A
   A
> L G
   G
> N
   G
> C /G/H/
   H                   [The computer responds with the change.]
> T
> P 23
   .NULL.
```

```
   A
   B
   C
   D
   E
   F
   G
   H
   I
   BOTTOM
> PO 6
   F
> A IND
   FIND
> T
> F(2) I
   FIND
> D
> T
> P 23
   . NULL.
   A
   B
   C
   D
   E
   G
   H
   I
   BOTTOM
> N-4
   E
> I F
> T
> P 23
   . NULL.
   A
   B
   C
   D
   E
   F
   G
   H
   I
   BOTTOM
> FILE TEST          [File the session in a file named TEST.]

   OK,
```

****

NATURAL ENVIRONMENT RESEARCH COUNCIL

INSTITUTE OF GEOLOGICAL SCIENCES

Report No. 73/6

Computer mapping of drift lithology from borehole records

D.W. Rhind, BSc, PhD

London: Her Majesty's Stationery Office 1973

The Institute of Geological Sciences was formed by the incorporation of the Geological Survey of Great Britain and the Museum of Practical Geology with Overseas Geological Surveys and is a constituent body of the Natural Environment Research Council.

It is recommended that reference to this report be made in the following form:

RHIND, D. W. 1973. Computer mapping of drift lithology from borehole records. *Rep. Inst. Geol. Sci.* No. 73/6. 12 pp.

SBN 11 880609 2

CONTENTS

ILLUSTRATIONS

Computer mapping of drift lithology from borehole records

D. W. RHIND,[1] BSc, PhD

Introduction

The Institute of Geological Sciences produces one medium-scale series of geological maps for most areas of the United Kingdom, the one-inch series at a scale of 1:63 360 until 1971, and subsequently at a scale of 1:50 000. While the advent of some 1:25 000 scale special sheets, the publication of 1:10 560 scale sheets in coalfield areas and the production of a few hydrogeological, geophysical and geochemical maps are additional to this basic coverage, they extend, in the main, over extremely limited areas. The primary reasons for this limitation of output would appear to be the considerable expense and staff requirements in producing maps by conventional cartographic and printing techniques, as there seems little doubt that a wide-ranging demand exists for graphic displays of many geological variables. Thus the published maps must attempt to have a multiplicity of functions.

Three criticisms of substance can be levelled at most of the Institute's existing maps of drift geology in glaciated parts of northern Britain; these criticisms result partly from cost constraints and the limitations in the methods of output.

1. The drift classifications chosen may be hybrids of genetic, chronological, lithological and morphological factors. Thus the maps are of limited value for researchers who require either homogeneity in the nature of the classification criteria or a different combination of factors to those used in the published map.

2. Little attempt is made on the maps to illustrate anything other than those deposits at the ground surface, except by implication, by exaggerating the extent of lower materials outcropping in valleys, and by the use of horizontal sections. In one area near Edinburgh, for example, 'boulder clay' is shown on the Institute's 1:25 000 scale map where extensive exposures show this to vary between only 1 and 2 m in thickness and to overlie up to 16 m of sand with some gravel. However, it may be impossible to represent the often gross variations in drift lithology (where these are known) on any one map, particularly a small scale one.

Because of such gross variations, drift maps often have a generally lower predictive power than solid geology equivalents. This suggests that it might be sensible to include more details (where these are known) of the drift than of solid rocks of comparable importance.

3. The published maps are generally out of date; new sections are borehole evidence are always becoming available and ideally should be immediately incorporated into maps, particularly where they change the pre-existing concepts of the drift character and thickness.

It would be facile to pretend that easy and cheap solutions exist to these and other problems of drift representation and classification, particularly if the printed, multi-colour, 'definitive' map must continue to be produced. The obvious solution would be to store all the field data in a form in which the individual user could access and manipulate it rapidly (and probably repeatedly) to suit his own needs. The case for data files of certain borehole and exposure records has been argued elsewhere (for example, Rhind and Sissons, 1971; Gover and others, 1971) and there is a good basis for believing that computer manipulation of such data can eventually meet at least some, and probably many, of the individual needs of professional geologists and other users of geological information.

There is also good reason to believe that the use of automated cartographic techniques can produce maps or other displays quickly and in a form usable by the geologist (for example, Rhind, 1971a). This paper describes one technique for automated production of special-purpose maps and draws upon part of an existing computer file (Rhind and Sissons, 1971) for the 40 x 29 km Edinburgh area as its input data. Sections of the data have already been mapped in other ways (Rhind, 1971b; Rhind, 1972) but here the aim has been to produce 'quick-look' and cheap maps of drift lithology at any desired depth below the ground from borehole and section records. Thus, if successful, it has direct relevance to criticism 2 noted above; on the understanding that a well-maintained data file

[1]NERC Experimental Cartography Unit, 6A Cromwell Place, London SW7 2DE

would be updated whenever necessary, it also has relevance to 3. The storage of 'raw', rather than 'heavily-processed', data in the file permits some flexibility in the classification scheme utilised by the individual user (see criticism 1).

Procedures and Conventions

IN OUTLINE

Maps were produced by computer from the National Grid 10-metre coordinates of boreholes or sections in the 5 km square NT 27 SW and their logs. The computer read down each log until it reached a specified depth and it then extracted the lithology at that level. This lithology was given a numeric code and the three (X, Y, Z) coordinates were stored on a magnetic tape. Arbitrarily chosen points within those areas in which the Institute has mapped rock as exposed at the surface (Zero Drift data, see Rhind and Sissons, 1971) were given the code for rock and were also stored. A second computer program then produced maps from the these coordinates; areas were shaded with specified symbols in relation to the Z value (that is, the lithology) of the nearest data point. The resultant maps (Figs. 2a to 2e) are discussed in the results section.

IN DETAIL

Borehole records were extracted from a file which was produced by merging the FF and PFF files described in Rhind and Sissons (1971); this new file was thus constructed of a fixed length summary record for each bore or section, followed by a detailed record (where this exists) of variable length, the length being specified in the summary record. Selection of points from this and from the Zero Drift File was carried out using a point-within-polygon routine to permit any irregular area of interest to be specified.

Relatively little transformation of the data was attempted before mapping. While the use of multi-variate classification techniques for facies definition (for example, Parks, 1969) is an attractive one, the limited and unsystematic recording of drift characteristics in each bore made this approach impossible. Similarly, the absence of accurate altitude values for many bores meant that the simple computation required to produce maps of lithology at a specified altitude, rather than at a specified depth below the ground, could not be carried out.

Mapping of the data was carried out using the proximal option in the SYMAP program (Fisher, 1968). This produces maps shaded with symbols indicating the value of the 'nearest neighbour' data point; thus the geometrical figures defined by the boundaries of the area around each data point are close approximations to Thiessen polygons (Haggett, 1965, p. 247). This method was chosen because isopleth mapping of the data was impossible (the Z values do not lie on an interval or ratio scale and the distribution cannot be considered continuous). Choropleth mapping was also impossible as the positions of drift boundaries were unknown, their definition being a prime objective of the study.

In using the proximal option, assignment of Z values on a nominal scale to equate with the lithology encountered at the specified depth would have been sufficient in the vast majority of cases. In SYMAP, the map is produced from characters which are each 2.5 x 3.1 mm in size; in a number of cases (the frequency is related to the data-point distribution and the map scale) two or more data points lie within the area of one character and, while these are indicated in the printed output, a mean Z value is used for extrapolation. In the area mapped, 'autocorrelation' of Z data seems high, the vast majority of superimposed data points revealing the same lithology at the same level. The only exceptions to this were the overlap of some bores showing clay and silt at one level mapped. However, this could only be established after producing the maps and it was necessary to allow for circumstances in which variations in lithology were rapid. This was done by coding the lithologies on a mixed ordinal/interval basis (Table 1), the procedure for this being as follows.

Those elements of the lithology which are conventionally defined by their position on a particle size range were placed on an arbitrarily chosen scale commencing at 1.0 for clay and ending at 5.0 for boulders. It was considered important to invest some element of 'reality' here but the need to keep this part of the scale compressed (see below) and the crudity of estimation of particle sizes in the original records (mostly compiled by a boring supervisor and unchecked by geologists) were such that it seemed unnecessary to use any of the conventional classification scheme thresholds for the class limits. Subsequent experiments showed that, for this data set, the numerical values of the limits chosen were not critical as the 'autocorrelation' in the data was high (see above). All clays (for example, lacustrine clays, glacial tills) were lumped into the same category at this stage, despite the well-known differences in their physical constitutions and strengths.

Undifferentiated drift was given the value 0.0.

Peat and other organic deposits were given the code - 10000.0, and the map class intervals chosen such that even if coincidence of one bore containing organic deposits with many others

Table 1. Numerical codes and symbolism for drift lithology

'Lithology'	Code	Mapped class interval	Symbol	Data point symbol
Peat and other organic deposits	- 10000.0		.	1
Undifferentiated drift	0.0	-0.5	,	2
Clay	1.0	0.5	—	3
Silt	2.0	1.5	=	4
Sand	3.0	2.5	+	5
Sand and gravel	3.5	3.25	I	6
Gravel	4.0	3.75	X	7
Boulders	5.0	4.5	⊖	8
Madeground	1000.0	5.5	⊠	9
Rock	10000.0	1005.5	■	*

containing clay, silt, sand, etc., was encountered, the symbol for organic materials would be mapped. This convention was utilised both because of the high practical and academic importance of buried peat layers and because of their scarcity.

Rock was given the code 10000.0. Thus, taken with suitable map class intervals, coincidence of anything other than rock and organic deposits (a low probability combination) would result in the extreme value member being represented at that position.

Made ground was given the code 1000.0. This, while a genetic rather than a strictly lithological term, had to be included by virtue of its frequency (many bores contained it, without defining its physical constitution) and because of its practical importance in construction works.

Many other classifications could be mapped by varying the groupings and class intervals within the data which is stored on tape.

Results and Conclusions

Fig. 1 illustrates the distribution of control points in the 5 km British National Grid square NT 27 SW. Distributional characteristics of the total data set for this area have already been published (Rhind, 1971a). At most, 841 of the bores and 'zero drift' locations in this area were utilised. As one of the categories of records scanned was that in which bores did not reach rock but attained depths below that specified, the numbers of control points diminished with increase in the specified depth. To this extent, reliability of the results diminished with increasing depth, although the probability of encountering bedrock obviously increases with depth.

Figs. 2a to 2e indicate the supposed drift distribution at depths, respectively, of 0.1, 1.0, 3.1 (10 feet), 5.0 and 10.0 m below the ground. The areas shaded with a symbol 'N' in Figs. 2a, 2b, 2d and 2e indicate those areas more than 0.5 km from the nearest borehole. Numerics on these maps indicate the position of a borehole and the class into which its value falls. The tenth level is indicated by an asterisk. Where more than one bore lies within a print character location, the symbol 's' is inserted. Fig. 2c is a coloured equivalent of these maps and was produced from black, red, blue and yellow colour separations. Multiple printouts from the same stored map were used to produce the separations, the symbolism being suppressed when the lithology was not to be printed in that colour. The white areas are those more than 0.5 km from a data point. It is possible to get acceptable registration of superimposed colours when the maps are photo-reduced to approximately one-half of their original size. Fig. 3 illustrates the field-mapped portrayal of the drift types in this area, taken from the appropriate published 1:10 560 scale geological map.

Comparisons of the computer output with the published drift geological map for this area are complicated by the following factors.

1. The surveys were made at different times. The published map is based on a survey completed in 1962, while some bores used in the 'computer map' were drilled as late as 1967.

2. A superior map-wide coverage of data was

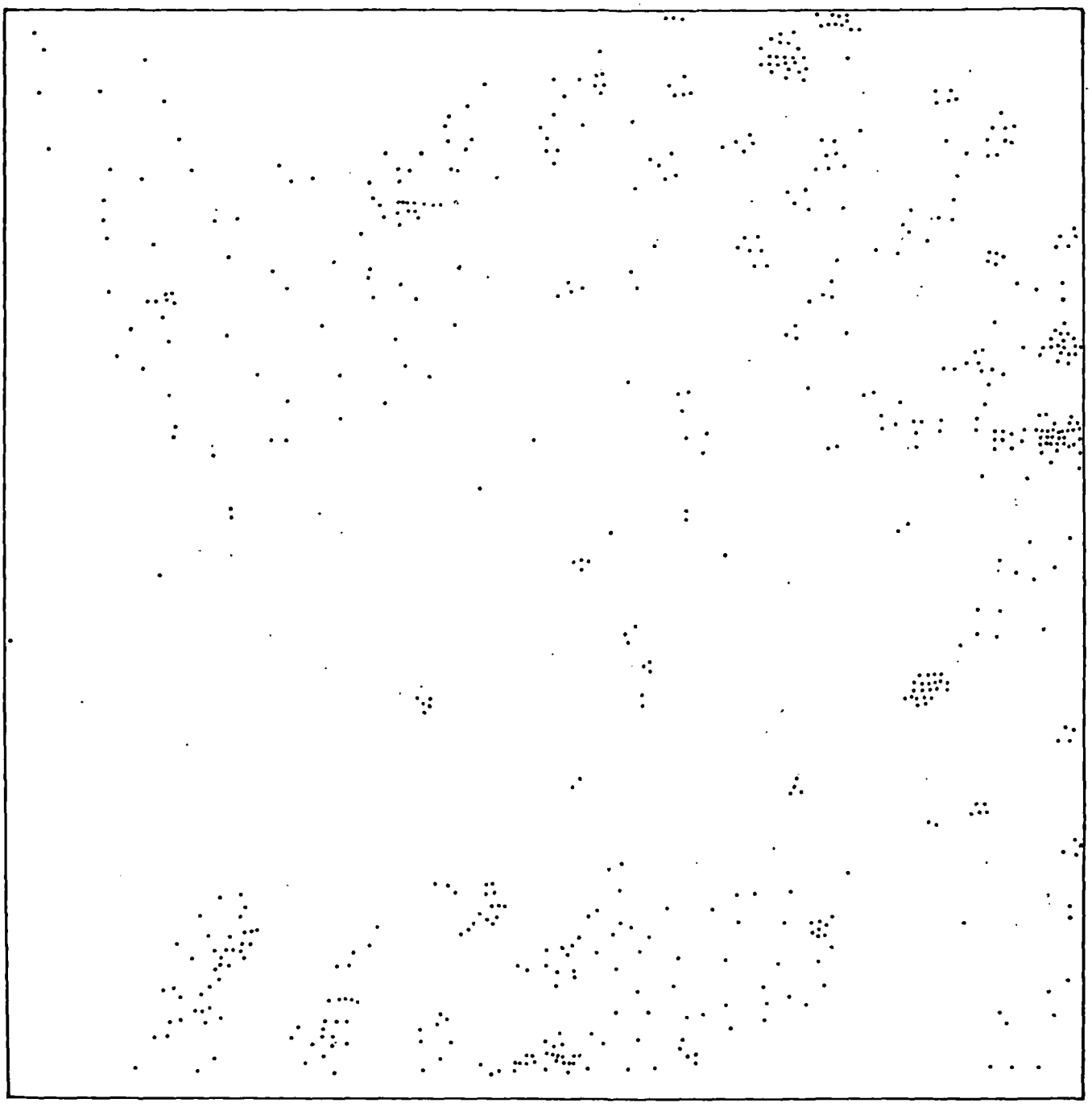


Fig. 1. Distribution of ground surface data points (that is, borehole and 'Zero Drift' locations) within BNG square NT 27 SW

available to the geologist by virtue of his ability to shallow auger wherever he felt more information to be necessary on the drift distribution.

3. The variability in depth to which the published map refers (frequently the areal extent of a thin drift deposit will also determine whether it is mapped). As most of the area is covered by drift disturbed by man at very shallow depths (see the distribution of made ground and undifferentiated drift at the 0.1 m depth in Fig. 2a) and as areas with a drift cover of up to 1 m are known to have been mapped as drift-free, the most appropriate of the computer maps for the comparison seems to be Fig.2b. Given that the area was mapped by a number of geologists, however, it is unlikely that any real consistency exists in the 'depth mapped'.

4. The classifications used in the published map are significantly different from those used in the computer-generated map. This was unavoidable with the available data.

5. The drillers' lithological interpretations are not always the same as that of the field geologist.

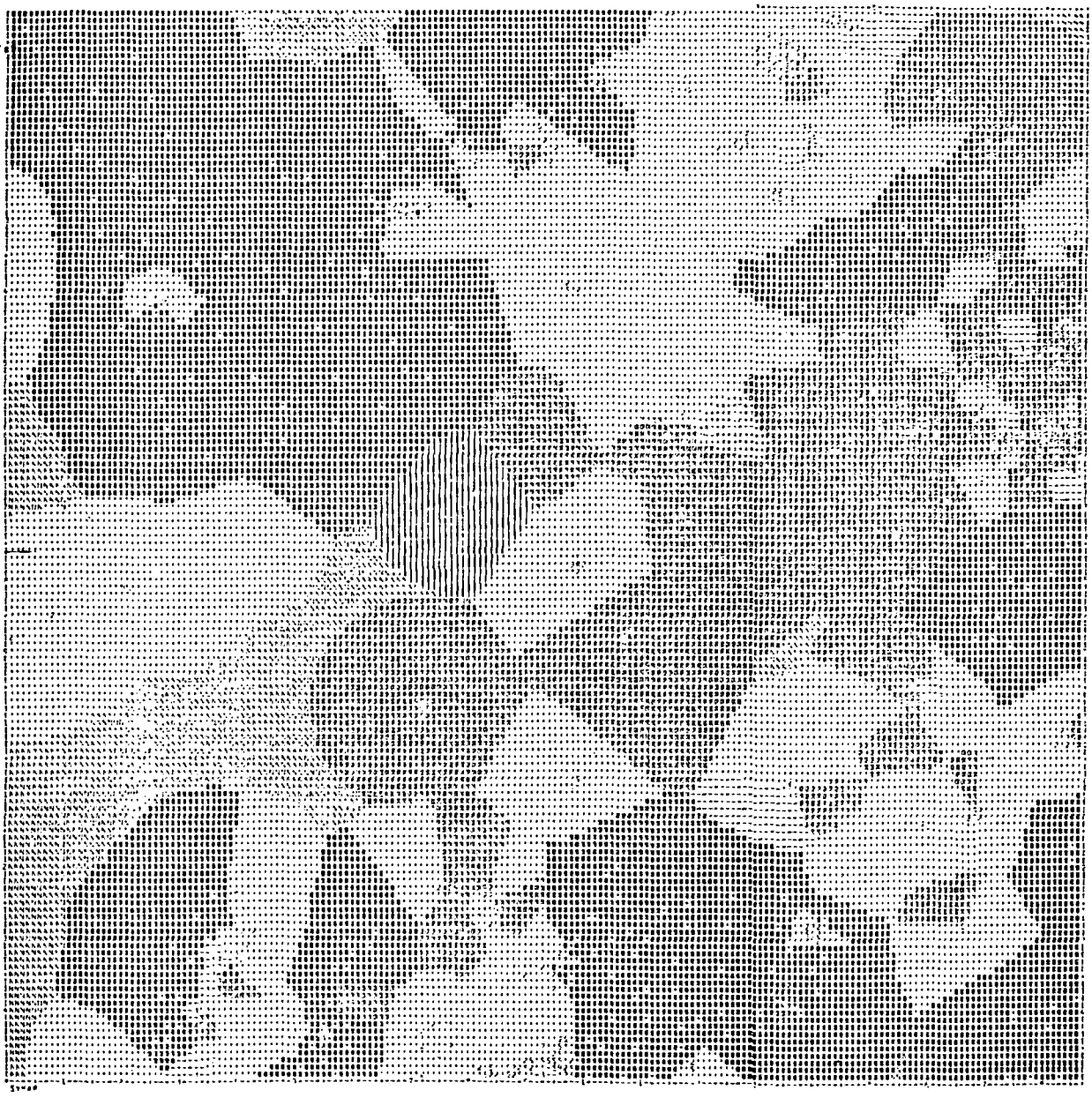


Fig. 2a. Distribution of drift lithologies within BNG square NT 27 SW at a depth of 0.1 m

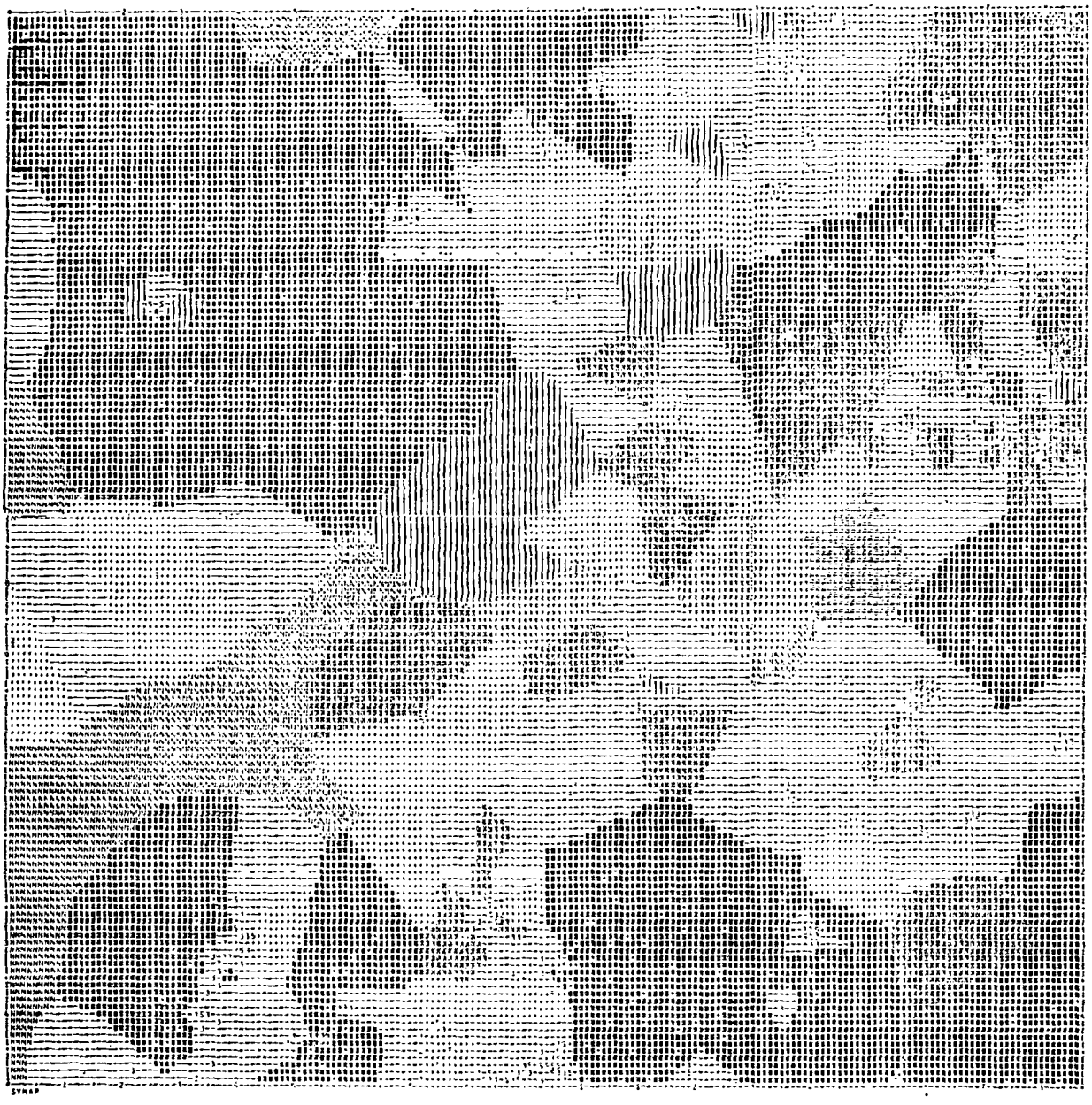
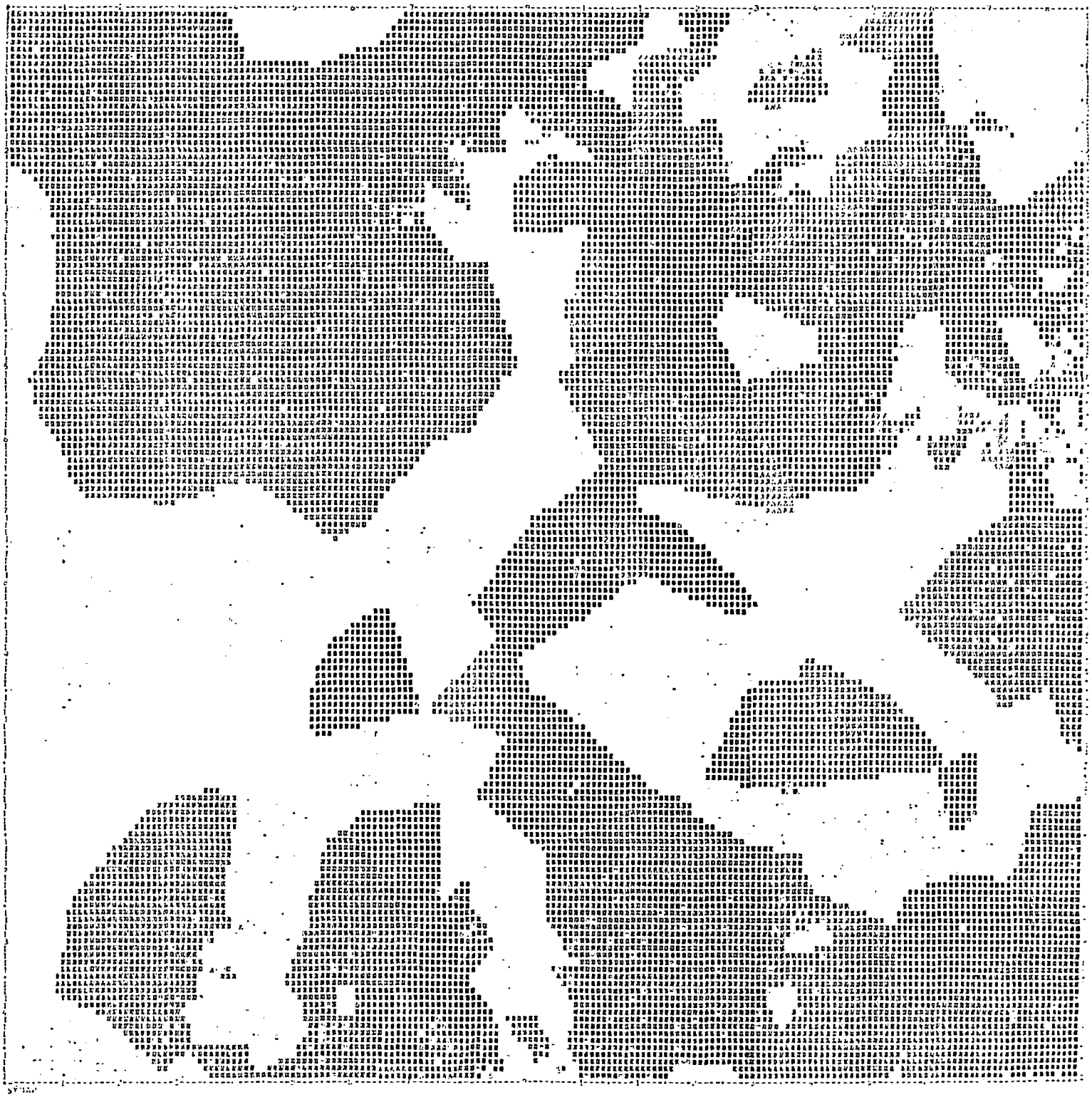


Fig. 2b. Distribution of drift lithologies within BNG square NT 27 SW at a depth of 1.0 m



IGS Report No. 73/6

Fig. 2c. Distribution of drift lithologies within BNG square NT 27 SW at a depth of 3.1 m (10 ft)

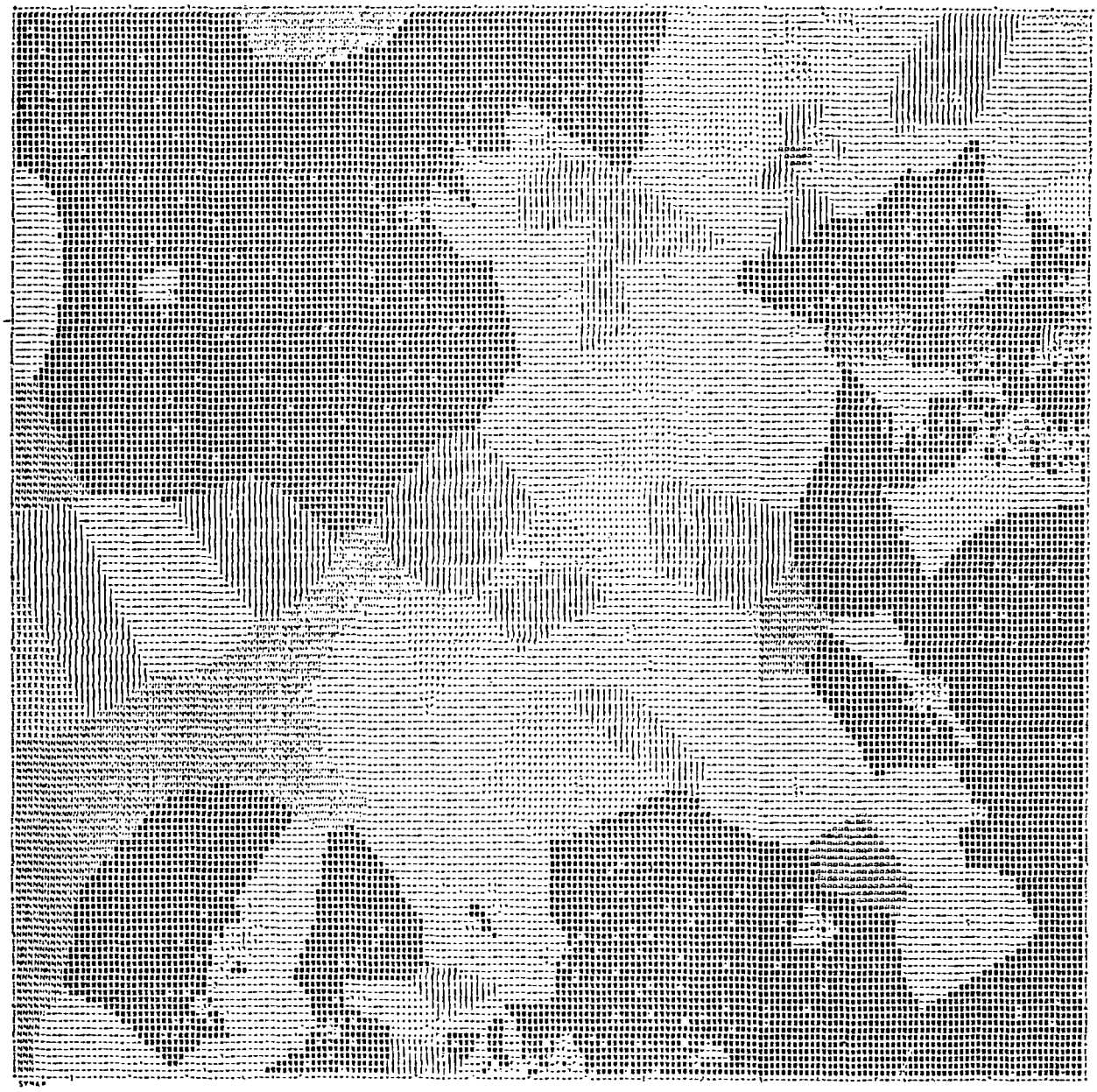


Fig. 2d. Distribution of drift lithologies within BNG square NT 27 SW at a depth of 5.0 m

Fig. 2e. Distribution of drift lithologies within BNG square NT 27 SW at a depth of 10.0 m

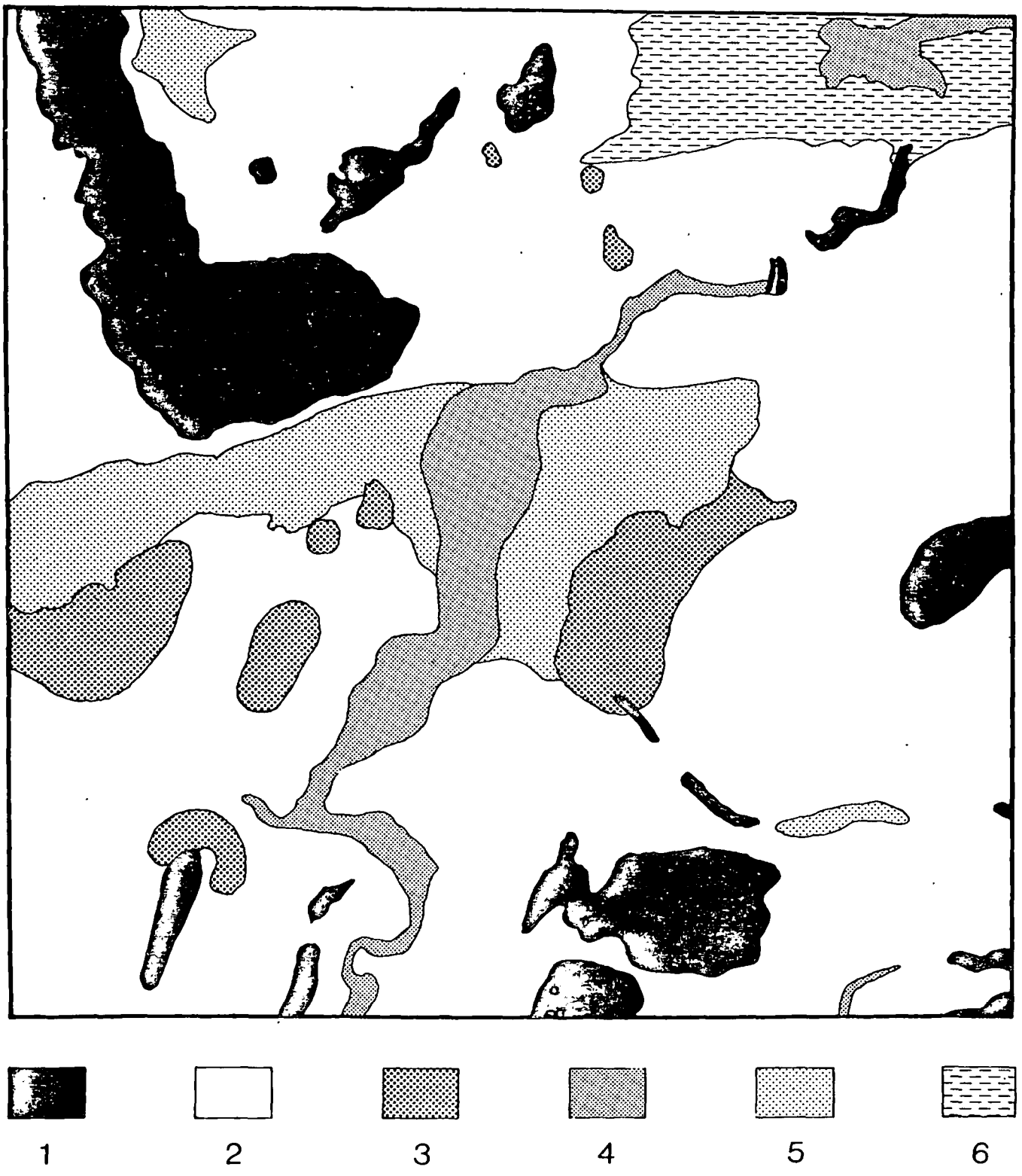


Fig. 3. Ground surface geology within BNG square NT 27 SW (reduced from the 1/10 560 scale geological map)

Key
1 Rock
2 Boulder clay
3 'Glacial' sand and gravel
4 'Flood plain' alluvium
5 'Lake' alluvium
6 High raised beach (late Glacial)

Despite these complications, the correlation between Figs. 2b and 3 is not trifling. The main south-west orientated band of non-clay deposits and a number of other common features are obvious. Apart from the angularities of the Thiessen polygons, perhaps the most disconcerting aspect of the maps is the 'cliff effect'. In some areas the boundaries of rock are the same at 3.1, 5.0 and 10.0 m. Such features are artifacts, caused by the nearest neighbour criterion; unless lithological changes occur with increasing depth such that both members of a pair of bores indicate a lithology of type X, or unless one bore becomes unusable through not penetrating to that depth in drift, no extension of the limits of distribution of that lithology from those at a higher level is possible. An examination of a fairly common sequence of strata, illustrated in Fig. 4, emphasises the crudity of the 'nearest neighbour' approach when unqualified by either the geometry or the age relationships of the beds. Fig. 4a illustrates a hypothetical cross-section, while Fig. 4b consists of 'maps' of the plotted and actual boundaries at three levels below the ground surface. The distribution of rock at 5.0 and 10.0 m down should correspond to the isarithms of these thicknesses already published in Rhind (1971a) or Rhind (1972). Many similarities do exist; the differences must be attributed to the quite different computational method used to obtain results and the slightly different data sets used.

Given that (in this country) field mapping of the uppermost drift is extensive and is likely to produce more accurate maps than any scheme relying entirely on a borehole data base, it is highly desirable to incorporate data on known ground surface distributions of formations which, in association with that derived from boreholes and recorded sections, are used to produce subsurface maps. In the simplest (and least realistic) sense this involves a projection downwards of these formations from the surface and some assessment of the more probable drift type, based on quantified descriptions of the horizontal and vertical components of rate of change of drift lithology. For example, is the drift at a point X metres below sand at the ground surface around the borehole and Y metres from a borehole containing clay at the depth X more likely to be sand or clay? Programming of this decision-making process would not be complex, nor would execution time be excessive providing the amount of data used to describe the formations was not substantial and was held in a suitable ('raster') fashion. The assessment of reasonable values of rate of change across the map would, however, be difficult, the degree of difficulty being related to the complexity of the area of study and its size. More realistically, and particularly for solid-rock data (to which the technique is generally applicable), it is desirable to include not merely a projection of the surface distributions but data on the geometry and age relationships of the beds. These data may be

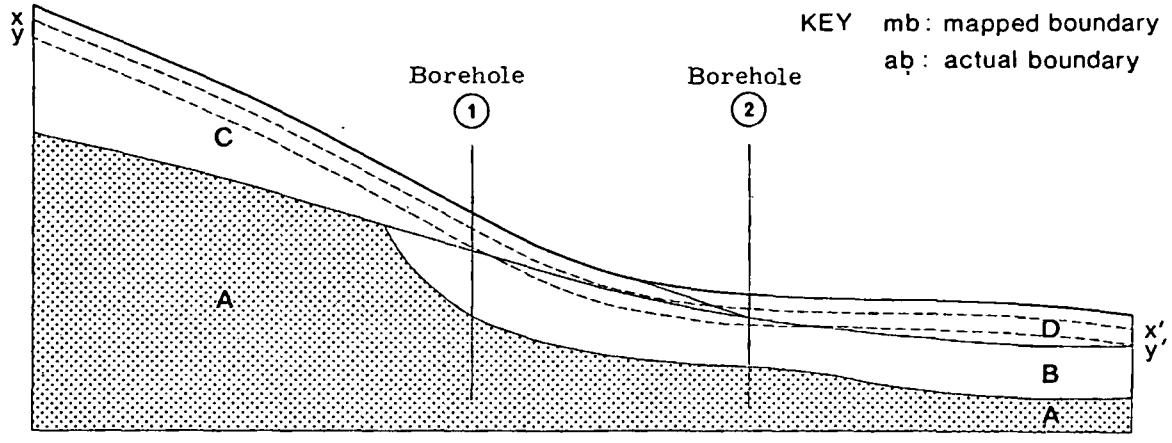
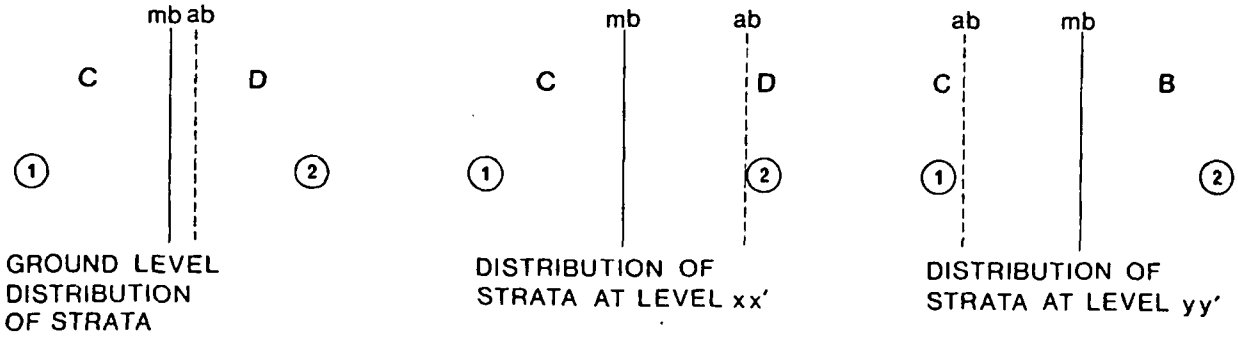


Fig. 4a (above). Hypothetical cross-section of geological strata to be reconstructed from borehole data alone. Fig. 4b (below). Maps produced of distribution of strata in section at three selected levels, using techniques described in the text



obtained either in bores or at the surface. Outline plans to do this have been drawn up but the procedures involved are highly complex, particularly if discontinuities in strata are permitted.

In the present study, one type of 'vertical projection' and the most simple stratigraphic relationship detectable at the ground surface (that between drift and rock) are already utilised in the adoption of the solid rock code with termination of searching at rockhead (even if the specified depth is considerably below that point) and by virtue of the use of 'pseudo-bores' from the Zero Drift File. Other stratigraphic information could not be incorporated in this study as non-lithological attributes were not available for this data set.

Indeed, the highly tenuous nature of virtually all the drift stratigraphies yet worked out for northern Britain suggests that, except as an aid to hypothesis testing, additional sophistication would be unusable.

Another improvement in the capabilities of the program which at first sight appears trivial is the substitution of a fixed datum (for example, Ordnance Datum) for that of the ground surface; in certain circumstances it might be useful to produce large scale maps at, say, 50 m above sea level (where this altitude might be the base of building foundations). Nonsensical results might be produced by the current program interpolating between two bores which commenced at ground surface altitudes of 60 and 20 m - rock would probably be shown to exist far out into the air at the 50 m level. The simple palliative for such circumstances would be to create a mask from the intersections of the 50 m plane with a digital model (or figure-field) of the ground topography, and use this mask to delete all nonsensical parts of the distribution of lithologies.

A further improvement of the technique could also be effected by recognising that the exposed morphology of geological strata is highly elongate in a broadly east-west orientation in the Edinburgh area (Sissons, 1967; Burke, 1969; Sissons, 1971) and this probably holds true for much of the buried morphology. Thus the probability of encountering a different lithology in travelling a unit distance is not isotropic and a relaxation of the simple 'shortest distance' rule to encourage the formation of boundaries with this orientation should prove worth investigating. The 'anisotropic' nature of the drift also provides difficulties in the specification of a map-wide distance constraint for interpolation or extrapolation from data points. Given the same control, it is clearly more reasonable to extrapolate or interpolate in a raised estuarine mudflat area (like the Carse Lands) than it is in a complex area of fluvio-glacial deposits. Similarly, it should be statistically possible to extrapolate or interpolate (with a given confidence) over greater distances east-west than north-south, because of the landscape moulding.

Since the initial project was completed in 1969, other data have become available for the area but have not been encoded: thus the maps illustrated cannot be considered definitive. In addition, it must be stressed that these maps are only examples of possible output. Many other maps could also be created; these include those from selections of bores for plotting on the basis of such criteria as age (for example, only those bored since 1955), method of boring or contractor (for example, exclude all those by contractor Y). A modification to the program would permit the production of crude 'quick-look' versions of profile-type maps. These, widely produced and used by the Dutch and West German Geological Surveys, represent not one but a fixed succession of lithological or stratigraphic units by one colour or tone.

References

Burke, M.J. 1969. The Forth Valley: An ice-moulded lowland. Trans.Inst.Br.Geogr., Vol.48, pp.51-59.

Fisher, H.T. 1968. Reference manual for synagraphic computer mapping: SYMAP V. Laboratory for Computer Graphics, Graduate School of Design, Harvard University.

Gover, T.N., Read, W.A. and Rowson, A.G. 1971. A pilot project on the storage and retrieval by computer of geological information from cored boreholes in central Scotland. Rep.No. 70/13, Inst.geol.Sci. 30 pp.

Haggett, P. 1965. Locational Analysis in Human Geography. (London: Arnold).

Parks, J.M. 1969. Multivariate facies maps. In Symposium on Computer Applications in Petroleum Exploration. Ed. D.F. Merriam, Kansas Geological Survey Computer Contribution 40.

Rhind D.W. 1971a. The production of a multi-colour geological map by automated means. Nachr.Kart.-u.Vermess Wes. Reihe 1, Heft 52, pp.47-51.

———— 1971b. Automated contouring: empirical comparisons of some differing techniques. Cart.Jnl, Vol.8, pp.145-158.

———— 1972. One-sided constraints in

automated contouring of drift thickness. Bull. geol. Soc. Am., Vol. 83, pp. 2525-2532.

———— and Sissons, J.B. 1971. Data banking of drift borehole information for the Edinburgh Area. Rep. No. 71/15, Inst. geol. Sci., 19 pp.

Sissons, J.B. 1967. The Evolution of Scotland's Scenery. (Edinburgh: Oliver and Boyd).

———— 1971. The geomorphology of central Edinburgh. Scott. geogr. Mag., Vol. 87, pp. 185-196.

Dd. 505906 K 8

Printed in England for Her Majesty's Stationery Office
by Unwin Brothers Limited, The Gresham Press, Old Woking, Surrey

1979 In Chemical Modeling in Aqueous Systems (ed E.A. Jenne) Am. Chem. Soc Symp. Ser. 93, Am. Chem. Soc, Wash D.C., p

COMP - GEOCHEM

Chemical Modeling of Trace Metal Equilibria in Contaminated Soil Solutions Using the Computer Program GEOCHEM

SHAS V. MATTIGOD and GARRISON SPOSITO

Department of Soil and Environmental Sciences, University of California, Riverside, CA 92521

The study of trace metal chemistry in soils has assumed added importance recently because of the possible deleterious effects on soil organisms, plants, and human beings that could result from the accumulation of these metals to toxic levels in soils contaminated, e.g., by the land disposal of sewage sludges or by the seepage of geothermal brines lost accidently from storage lagoons. Although the soil solution is an open, dynamic, natural water system whose composition reflects many reactions occurring simultaneously among its aqueous constituents and between those constituents and the assembly of mineral and organic solid phases present, a knowledge of the general features of soil trace metal equilibria is expected to be a useful guide to predicting what will occur in nature if contamination takes place. These general features cannot be assessed conveniently by laboratory experiments because of the complexity of soil solutions. A more viable alternative is assessment by a computer model that is based on thermodynamic association and solubility product constants.

The computer program GEOCHEM is adapted and being developed for soil solutions from the REDEQL2 program created originally by Morel and Morgan and their coworkers at Caltech (1). The principal ways in which GEOCHEM differs from REDEQL2 are: 1) it includes thermodynamic data for a few hundred additional soluble complexes and solids that are particularly relevant to trace metal equilibria in soil, 2) it contains a subroutine for cation exchange on constant charge surfaces that is based on a thermodynamic model, and 3) it contains a subroutine for the estimation of single-ion activity coefficients at ionic strengths up to 3 M.

A number of important theoretical problems have arisen in connection with the development of GEOCHEM. These problems may be classified broadly into four categories: (a) stability constants for trace metal complexes with inorganic ligands; (b) stability constants for trace metal complexes with organic ligands; (c) solubility product constants for soil clay minerals, and (d) thermodynamic cation exchange constants and exchanger

0-8412-0479-9/79/47-093-837$05.00/0

phase activity coefficients. Any reasonably accurate computation of trace metal equilibria in a soil solution must be preceded by a successful attack on these four problem areas.

Stability Constants for Soluble Inorganic Complexes of Trace Metals

Studies involving trace metal speciation in soil solutions require the values of stability or association constants of complexes of the trace metals with a number of inorganic and organic ligands which exist in these environments. Much of these data can be obtained readily from published compilations (2-7). However, experimental data on association constants for complexes of the trace metals with ligands such as CO_3^{2-}, HCO_3^-, PO_4^{3-}, HPO_4^{2-}, $H_2PO_4^-$ are not often available. In view of this lack of experimental data, modeling studies of chemical equilibrium in natural water systems have typically followed two approaches. One of the approaches, represented by computer programs such as SOLMNEQ and WATEQ (8, 9), is to include only those complexes for which the values of experimental association constants are available. The second approach consists of including in the equilibrium calculations estimates of the values of the association constants of complexes that are believed to form, but for which experimental data do not yet exist, along with experimentally derived values of association constants (10, 11). The latter approach seems preferable because it tends to include in the calculations all known interactions of metals and ligands through complex formation and thus provides results which are expected to be qualitatively more reliable than the results obtained through the former approach.

The methods available for calculating ion association constants consist of theoretical methods based on electrostatics (12, 13) and empirical methods based on correlations of association constants with properties of ions such as electronegativity, charge, radii, coordination number, etc. (14, 15, 16, 17, 18). Recently, Mattigod and Sposito (19) estimated the association constants for complexes of a number of cations of the first transition metal series with many of the inorganic ligands considered to be of importance in the trace metal chemistry of soil solutions. The association constants calculated by Mattigod and Sposito (19) with the method of Kester and Pytkowicz (13), for carbonate and phosphate complexes appear to be systematically different from those estimated recently with the correlation method of Nieboer and McBryde (17). The latter method has a firmer empirical basis and, therefore, has been preferred. Values of the revised estimated and measured association constants for some trace metal-inorganic ligand complexes are listed in Table I. These and other estimated values have been incorporated into GEOCHEM. They will be replaced when reliable experimental values become available.

There is some experimental evidence which indicates that mixed-ligand complexes of trace metals may be very stable and dominant in soils (20). In modeling the speciation of elements in sea water, Dyrssen and Wedborg (10) found that mixed-ligand complexes of certain trace metals were significant. At present, experimental data on mixed-ligand complexes are rather sparse. However, stability constants can be estimated by a statistical method suggested by Dyrssen, Jagner and Wengelin (21). Currently, GEOCHEM includes estimated stability constants of hydroxy-chloro complexes of a few trace metals. Lack of data on many mixed-ligand complexes is clearly one of the significant shortcomings in modeling studies at present.

Table I.

Estimated and measured common logarithms of association constants for some trace metal-inorganic ligand ion pairs at 25°C, 1 atm

Ligand	Mn^{2+}	Fe^{2+}	Co^{2+}	Ni^{2+}	Cu^{2+}	Zn^{2+}
CO_3^{2-}	4.52	5.31	5.53	5.78	6.73[1]	4.76[2]
HCO_3^-	1.95[3]	2.72	2.89	3.08	4.29	2.79
PO_4^{3-}	7.19	7.93	8.13	8.37	9.85	8.02
HPO_4^{2-}	3.58[4]	3.60[5]	3.04	2.93[6]	3.20[6]	3.30[7]
$H_2PO_4^-$	1.35	2.70[5]	1.49	1.53	1.76	1.60[7]

[1]Schindler et al (22), [2]Bilinski, Huston and Stumm (23)
[3]Morgan (24), [4]Smith and Alberty (25), [5]Nriagu (26),
[6]Sigel et al (27), [7]Nriagu (28)

Stability Constants for Soluble Organic Complexes of Trace Metals

The soluble, metal-complexing, organic fraction of sewage sludge is a heterogeneous assembly of molecules that is characterized by wide ranges of chemical composition, molecular

weight, and functional group acidity (29). This complexity poses a difficult problem for the chemical modeling of soil solutions contaminated by the incorporation of trace metal-bearing sludges. A succesful model will require, in particular, information about the many possible reactions of trace metals with the fulvic acid fraction of sludges.

The dissociation of protons from fulvic acid (FA) extracted from anaerobically-digested sewage sludge has been investigated by potentiometric titration (29). In that study, a number of important facts was brought to light which suggest that the behavior of fulvic material in a soil solution will indeed be complex. According to Sposito and Holtzclaw (29), there appear to be four separate classes of dissociable functional groups that range in acidity from very strong (ionized at pH < 2) to very weak (ionized at pH > 10). (These classes are just four out of a continuum of classes of functional group acidity in sludge-derived FA, since no acid or alkaline final end points appear in the titration curve between pH 1 and 11.) Based on this evidence, Sposito et al. (30), suggested that the acidic functional group classes in sludge-derived FA be designated I, II, III and IV for those groups that titrate approximately in the pH ranges < 3, 3-5, 5-8, > 8, respectively.

The reactions between sludge-derived fulvic material and trace metals are poorly understood and, at present, no reliable thermodynamic stability constants are available. A provisional approach to resolving this difficulty, which has been employed often in modeling studies, is to identify carefully certain classes of known organic acids whose proton dissociation constants fall into the ranges observed for the soil solution organics and which can be expected to be present or to simulate closely the organic acids present (whether "natural" or from contamination) in a soil solution. Table II lists a set of aromatic, aliphatic, and amino acids that are frequently observed in soil leachates, together with the relative concentrations of each in a mixture of model water-soluble soil organics that provides the total of 2.2 meq of H^+ found per gram of sludge-derived FA (30). It has been observed that the functional groups in these acids are similar to the functional groups present in sludge fulvic material. The measured stability constants for trace metal complexes with these model organic acids are assumed to be good approximations to the unknown stability constants for the assembly of soil solution organics, in the sense that a mixture of the model organic acids whose proton titration curve semiquantitatively simulates that of sludge-derived FA (Figure 1) will also produce a comparable distribution of various trace metals among organic and inorganic species in a soil solution. The several obvious limitations and dangers in this provisional approach, presently used in GEOCHEM, are not considered to be as serious as completely neglecting the organic speciation of trace metals in a soil

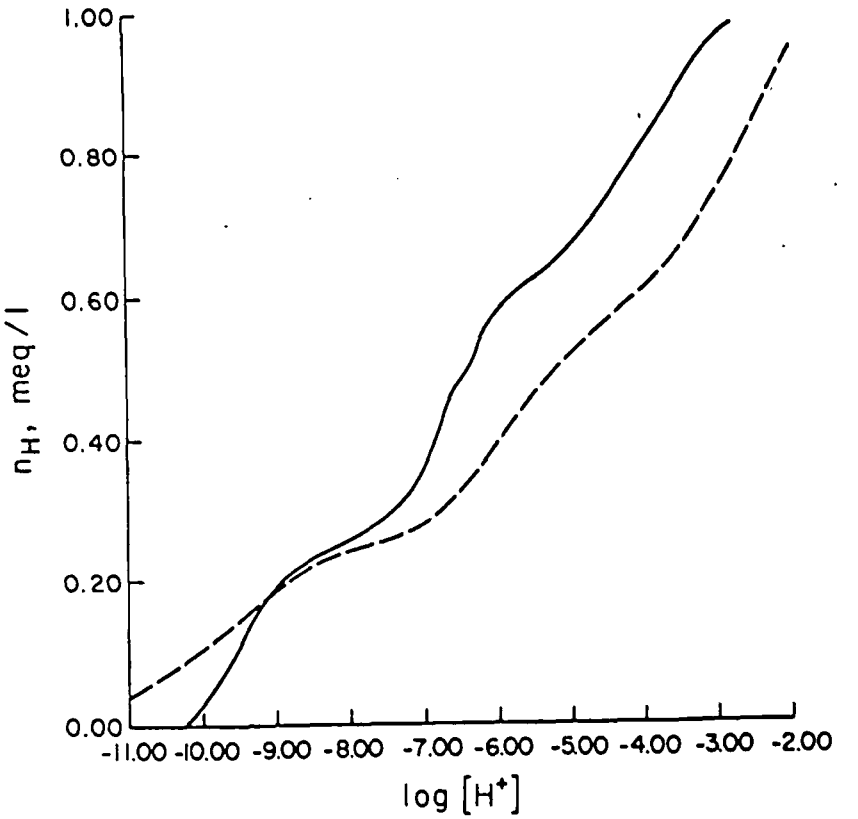


Figure 1. Comparison between the potentiometric titration curve for a sewage-sludge-derived fulvic acid (——) and the simulated titration curve for a model fulvic acid (– – –)

solution whose organic constituents may approach 10^{-3} M in total concentration (e.g., a soil solution affected by sewage sludge application).

Table II.

Organic acids used as a model for sludge-derived fulvic acid

Acid	Concentration (pM)
Benzenesulfonic	4.27
Salicylic	4.27
Phthalic	3.97
Citric	4.14
Maleic	3.97
Ornithine	4.36
Lysine	4.36
Valine	4.36
Arginine	4.49

Solubility Product Constants for Soil Clay Minerals

Solid phases comprise the dominant fraction of the soil volume and, as such, the dissolution, precipitation and ion exchange properties of the soil minerals have a profound influence on the composition of soil solutions. Therefore, it is essential to include the reactions these solids may undergo in soils in simulation studies of chemical equilibrium. Reliable thermodynamic data on the stability of many of the important oxides, hydroxides, carbonates, phosphates, and silicates are available and can be incorporated into a computer program such as GEOCHEM. In many soils, phyllosilicates dominate the clay-size fraction. Smectites as a group of the phyllosilicates are ubiquitous in soils and sediments. Because of their large surface areas and high cation exchange capacities, smectites, when present, play a significant part in influencing the composition of soil solutions. Unlike other clay minerals, smectites also exhibit a broad range of chemical compositions and, therefore, it is unlikely that the standard free energy of formation of every naturally-occurring smectite will be determined experimentally. This difficulty has prompted a

number of attempts to estimate $\Delta G^{\circ}_{f,298.15}$ for silicates in general (31, 32, 33) and smectites in particular (34, 35). These methods are based on simple physical models which are related primarily to classical electrostatics. The success of these methods is based on the observation that the short-range coordination environment of a cation that is found in silicate minerals does not tend to vary much from one mineral to another. Since covalency plays its most important role in nearest neighbor interactions, it follows that its contribution to the structural energy of a mineral will be about the same among silicates and, therefore, that the relative stabilities of the minerals largely will be determined by longer-ranged ionic interactions.

Nriagu (34), in comparing his method with the method of Tardy and Garrels (35), concluded that the accuracy of estimation of $\Delta G^{\circ}_{f,298.15}$ values of phyllosilicates in either method was about the same. But, each of these methods contains a critical, ad hoc assumption that is difficult to justify on geochemical grounds. Tardy and Garrels (35) had to assume that the hydroxyl ions in Mg-bearing layer silicates are associated only with magnesium ions, insofar as thermochemical calculations are concerned. Nriagu (34) was forced to decrease the values of $\Delta G^{\circ}_{f,298.15}$ for solid KOH and NaOH by about 15 percent from the measured values in order to obtain agreement between the results of his method and experimental data for Na- and K-containing clay minerals. These problems with the methods of Tardy and Garrels (35) and Nriagu (34) prompted the development of a method by Mattigod and Sposito (36) for estimating the $\Delta G^{\circ}_{f,298.15}$ values of smectites which not only shows a slight improvement in prediction, but also is free of the arbitrary assumptions inherent in some of the other methods. This method employs the concept of ionic bonding as applied to clay minerals, mentioned earlier, along with a statistical equation to relate the charge on a smectite due to isomorphous substitution and the valence and radius of the interlayer cation to the free energy changes brought about by changes in coordination environment. These changes in free energy occur in transferring a cation from relatively high-potential sites in a hydroxide to a relatively low-potential interlayer site in a smectite. The relationship developed by Mattigod and Sposito (36) is:

$$\Delta G^{\circ}_{f} = \Sigma n_i \Delta G^{\circ}_{f}(n_i) - (\Sigma n_i Z_i - 12)\, \Delta G^{\circ}_{f} H_2O_{(l)} - |\delta| \qquad (1)$$

and $$\ln |\delta| = 1.9283\ C + 0.3501\ R - 0.2819\ Z + 3.5427$$

where, ΔG°_{f} = standard free energy of formation of a smectite
n_i = reaction coefficient of the i^{th} hydroxide
Z_i = charge on the i^{th} cation (including Si)

ΔG_f^o = standard free energy of formation of the i^{th} hydroxide component

C = cation exchange capacity (due to isomorphous substitution) of the smectite per formula unit, in eq/fw

R = Pauling radius (Å) of the exchangeable (interlayer) cation

Z = valence of the exchangeable cation

Equation 1 provided estimates of $\Delta G^o_{f,298.15}$ for smectites which showed better agreement with the experimentally derived values than the other two methods (34, 35). A comparison among the methods is given in Table III.

Cation Exchange Phenomena

The task of calculating the composition of a soil exchanger phase in equilibrium with a soil solution has two distinct parts: the thermodynamic exchange equilibrium constants must be determined and the activity coefficients of the components of the exchanger phase must be estimated. The first part of the problem, that of obtaining exchange equilibrium constants, is not particularly difficult since a number of measurements of ΔG_{ex} for the common clay minerals (e.g., smectites, vermiculites, and kaolinites) has been published. Alternatively, a semi-empirical approach such as the James-Healy model included in REDEQL2 (1) may be employed to estimate a standard free energy of adsorption for a cationic species.

Smectites are one of the most important soil clay minerals as regards cation exchange. The reversible exchange reactions of these minerals with metal cations may be pictured as kinetically-favored (i.e., rapid) precipitation-dissolution reactions (44). From this point of view, it is meaningful to write the "exchange half-reaction":

$$MX_m(s,aq) = M^{+m}(aq) + mX^{-1}(s,aq) \qquad (2)$$

where X refers to one equivalent of the anionic portion of a smectite exchanger and M^{+m} is an exchangeable cation. The typical exchange reaction then is pictured as a set of pairs of reactions such as Eq. 2, with each reaction pair involving two different metal cations. The analogy between Eq. 2 and the dissolution reaction for a solid is evident.

However, there are some important differences between the equilibrium constant for Eq. 2 and the usual K_{so}. First, the compound on the left-hand side of Eq. 2 is not a dry solid at standard temperature and pressure, but instead is a homoionic smectite in contact with an aqueous solution. The standard state for $MX_m(s,aq)$ accordingly is the homoionic clay mineral at standard temperature and pressure in equilibrium with an infin-

Table III. Comparison between estimated and experimental $\Delta G^o_{f,298.15}$ values (kJ/mol) of some smectites.

Mineral	Composition	Estimated values: TARDY and CARRELS method (35)	Estimated values: MRIACU method (34)	Eq. 1, This paper	Experimental values	(Ref.)
1. Aberdeen	$M_{0.2075}(Al_{0.18}Si_{3.82})(Al_{1.29}Fe^{3+}_{0.335}M_{0.445})O_{10}(OH)_2$	-5220.6	-5173.0	-5211.1	-5218.7	(37)
2. Aberdeen	$Al_{0.1383}(Al_{0.18}Si_{3.82})(Al_{1.29}Fe^{3+}_{0.335}M_{0.445})O_{10}(OH)_2$	-5193.3	-5139.7	-5182.8	-5200.0	(38)
3. Belle Fourche	$M_{0.1325}(Al_{0.065}Si_{3.935})(Al_{1.515}Fe^{3+}_{0.225}M_{0.29})O_{10}(OH)_2$	-5223.6	-5184.6	-5209.4	-5222.8	(37)
4. Belle Fourche	$Al_{0.0883}(Al_{0.065}Si_{3.935})(Al_{1.515}Fe^{3+}_{0.225}M_{0.29})O_{10}(OH)_2$	-5206.2	-5175.3	-5188.5	-5213.6	(38)
5. Smectite from Houston Black Clay	$M_{0.225}(Al_{0.30}Si_{3.70})(Al_{1.345}Fe^{3+}_{0.405}M_{0.27})O_{10}(OH)_2$	-5209.4	-5158.8	-5208.5	-5214.7	(39)
6. Beidellite (Mg-sat)	$M_{0.135}Ca_{0.01}Na_{0.07}K_{0.095}(Al_{0.45}Si_{3.55})$ $(Al_{1.41}Fe^{3+}_{0.415}Fe^{2+}_{0.055}M_{0.205})O_{10}(OH)_2$	-5220.2	-5180.5	-5185.0	-5200.1	(40)
7. Beidellite (K-sat)	$K_{0.37}Ca_{0.01}Na_{0.07}(Al_{0.45}Si_{3.55})$ $(Al_{1.41}Fe^{3+}_{0.415}Fe^{2+}_{0.055}M_{0.205})O_{10}(OH)_2$	-5238.4	-5211.8	-5205.7	-5215.4	(40)
8. Colony	$M_{0.195}(Al_{0.19}Si_{3.81})(Al_{1.52}Fe^{3+}_{0.22}M_{0.29})O_{10}(OH)_2$	-5258.3	-5217.2	-5253.3	-5262.6	(41)
9. Colony I	$M_{0.185}(Al_{0.19}Si_{3.80})(Al_{1.58}Fe^{3+}_{0.19}M_{0.26})O_{10}(OH)_2$	-5261.4	-5219.0	-5253.1	-5267.6	(42)
10. Colony II	$M_{0.21}(Al_{0.19}Si_{3.81})(Al_{1.52}Fe^{3+}_{0.21}M_{0.29})O_{10}(OH)_2$	-5264.6	-5222.7	-5261.9	-5262.1	(42)
11. Castle Rock	$M_{0.21}(Al_{0.32}Si_{3.68})(Al_{1.52}Fe^{3+}_{0.14}M_{0.46})O_{10}(OH)_2$	-5329.8	-5293.8	-5333.0	-5336.8	(42)
12. Upton	$M_{0.17}(Al_{0.07}Si_{3.93})(Al_{1.55}Fe^{3+}_{0.20}M_{0.24})O_{10}(OH)_2$	-5236.9	-5195.6	-5226.8	-5218.0	(42)
13. Clay Spur (Na-sat)	$Na_{0.27}Ca_{0.1}K_{0.02}(Al_{0.06}Si_{3.94})(Al_{1.52}Fe^{3+}_{0.19}M_{0.22})O_{10}(OH)_2$	-5268.4	-5227.4	-5242.1	-5226.4	(43)
14. Cheto (Ca-sat)	$Ca_{0.185}Na_{0.02}K_{0.02}(Al_{0.07}Si_{3.93})(Al_{1.52}Fe^{3+}_{0.14}M_{0.33})O_{10}(OH)_2$	-5287.5	-5227.2	-5262.7	-5245.3	(43)

itely dilute solution containing the cation M^{+m} (45). In addition, the species $X^{-1}(s,aq)$ is the anionic part of the smectite exchanger, considered as a (charged) solid at standard temperature and pressure that is in contact with an infinitely dilute aqueous solution. Thus X^{-1} is not a dissolved species, as it would be if Eq. 2 referred to an ordinary dissolution reaction. Lastly, it should be noted that the equilibrium constant for Eq. 2 cannot be determined easily because of the difficulty of measuring the activity of $X^{-1}(s,aq)$. This fact need not undermine the utility of Eq. 2, however, because one may write

$$\mu^{\circ}(X^{-1}(s,aq)) = \frac{1}{m}[\mu^{\circ}(MX_m(s,aq)) - \mu^{\circ}(M^{+m}(aq))] \quad (3)$$

for any M and X for which the equilibrium constant in Eq. 2 may be taken to be unity. With the free energy of formation of $\mu^{\circ}(X^{-1}(s,aq))$ so determined, the values of the equilibrium constants of all other exchange half-reactions can be found, given the available values of the free energies of formation of one mole of the aqueous metal cations and one equivalent of the homoionic clay minerals.

There is evidence (46, 47) to suggest that the half-reaction

$$HX(s,aq) = H^{+}(aq) + X^{-1}(s,aq) \quad (4)$$

may be assigned an equilibrium constant near unity, as may be the exchange reaction between Na^{+} and H^{+} on a smectite. The lack of selectivity between H^{+} and Na^{+} is the more firmly established hypothesis of the two (46). If Eq. 4 truly had an equilibrium constant of unity, then H-smectites would behave as very strong electrolytes in aqueous solutions (e.g., a 1% suspension of H-montmorillonite with a CEC equal to 1 meq/g would exhibit a pH value of 2). The available published data on H-smectite suspensions are not conclusive, but they suggest strong electrolyte behavior. Given this hypothesis, it follows that

$$\mu^{\circ}(X^{-1}(s,aq)) = \mu^{\circ}(HX(s,aq)) \quad (5)$$

for any smectite.

The direct measurement of $\mu^{\circ}(HX(s,aq))$ is difficult because pure H-smectites are not easy to prepare. But this free energy of formation can be estimated using the accurate correlation technique discussed previously (36) and the hypothesis of no selectivity in H^{+}-Na^{+} exchange. Thus

$$\mu^{\circ}(HX(s,aq)) = \mu^{\circ}(NaX(s,aq)) + \mu^{\circ}(Na\ (aq)) \quad (6)$$

where $\mu^{\circ}(NaX(s,aq)) = \Delta G^{\circ}_{f,298.15}$ for one equivalent of Na-smectite, estimated according to Eq. 1. It then follows that the equilibrium constant for Eq. 2 is given by the expression

$$\log K_{Mex} = \frac{1}{5.707}[\mu^{\circ}(MX_m(s,aq)) - \mu^{\circ}(M^{+m}(aq)) - m\mu^{\circ}(HX(s,aq))] \quad (7)$$

at 298.15 K, where $\mu^{\circ}(MX_m(s,aq)) = \Delta G^{\circ}_{f,298.15}$ for m equivalents of M-smectite. The values of K_{Mex}/K_{Naex} for two montmorillonites and several exchangeable cations of interest in soils are listed in Table IV. These equilibrium constants may be treated in much the same way as are ordinary solubility product constants in the GEOCHEM program. Aside from understanding that the K_{Mex} values are relative, there is a need to modify the equilibrium calculation for a soil solution so as not to permit $X^{-1}(s,aq)$ to contribute to the ionic strength of a solution in contact with the exchanger phase.

Table IV.

$\log K_{Mex}/K_{Naex}$ values for the dissociation of various cations from two smectite surfaces

Cation	Smectite	
	Camp Berteau	Chambers
Mg^{2+}	- 1.81	- 3.51
Ca^{2+}	- 2.12	- 3.37
Na^{+}	0	0
K^{+}	- 1.41	- 1.04
Ni^{2+}	- 6.52	- 8.18
Cu^{2+}	-11.61	-13.26
Zn^{2+}	- 8.27	- 9.85
Cd^{2+}	-10.67	-11.95
Pb^{2+}	-21.09	-22.01

The problem of estimating the activity coefficients of the species MX_m which make up the exchanger phase is quite formidable. Fundamentally, what is needed is an equation for the activity coefficient that is comparable in simplicity and generality with the Davies equation. Unfortunately, the measurements of exchanger activity coefficients that are available apply only to two-component systems and exhibit no parameters of the same generality as is the ionic strength in the case of aqueous solutions. On the other hand, there is evidence that two-component smectite exchangers containing cations such as Na^+, K^+, Ca^{2+}, Mg^{2+}, and the trace metals are nearly ideal solutions (48, 49). This fact and the absence of information concerning multicomponent exchangers suggest that a good, working first approximation is to set all activity coefficients in a smectite exchanger equal to unity. This approach assumes that exchange selectivity will in most cases of interest be dominated by the interaction between the exchangeable cations and the exchanger instead of by interactions among the exchangeable cations themselves. Clearly, much work remains to be done on this important problem. This ion exchange model is being tested presently for its reliability in predicting the exchangeable ion composition on smectites.

Ionic Strength Corrections

In the computer program REDEQL2 (and GEOCHEM), the activity coefficients for aqueous species are calculated from the Davies equation. Generally, this equation is used for the computation of activity coefficients of ions in aqueous solutions up to an ionic strength of 0.5 M, even though Dyrssen et al. (21) indicate that this equation can be used for ionic strength up to 1.0 M. In saline soils and soils contaminated with geothermal brines, the ionic strengths of the soil solution may exceed 0.5 M. This fact poses the necessity of using equations which have been developed to describe the activity coefficients of ions in concentrated, multicomponent electrolyte solutions. As part of a study on the chemistry of ore-forming fluids, Helgeson (50) has proposed that the true individual ion activity coefficients for ions present in small concentrations in multicomponent electrolyte solution having sodium chloride as the dominant component be approximated by a modified form of the Stokes-Robinson equation. The equation proposed is:

$$\log \gamma_i = -\frac{AZ_i^2 I^{\frac{1}{2}}}{1+a_i B I^{\frac{1}{2}}} + B^{\circ} I \qquad (8)$$

where, γ_i = individual activity coefficient of the i^{th} trace ion
Z_i = charge on the i^{th} trace ion
I = true ionic strength

A and B = the Debye-Hückel Parameters
a_i = "Distance of closest approach" of the i^{th} ion (cm)
B° = Deviation function (0.041 at 25°C)

According to Helgeson (50), equation 8 can be used to estimate the individual ion activity coefficient for ions present in small concentrations in sodium chloride solutions of true ionic strength up to 3.0 M. Since saline soils and geothermal brines are often dominated by sodium chloride, it will be appropriate to use the equation proposed by Helgeson (50). Therefore, in GEOCHEM, ionic activity coefficient calculations for such systems are performed by equation 8.

The "distance of closest approach" parameters for many of the ions and complexes included in GEOCHEM are not available. This lack of data has been circumvented by assigning arbitrarily 4, 5, and 6 A for mono-, di- and tri- (and higher) valent ions respectively. A similar assumption has been made previously by Truesdell and Jones (9).

Applications

At present, there is a serious effort being made to assess the impact of geothermal power development on agriculture in the Imperial Valley of California. One aspect of the problem involves the possible contamination of agricultural soils from accidental spills of geothermal brine. A good semi-quantitative guide to trace metal solubility and mobility in contaminated soils can be compiled from results obtained from GEOCHEM.

As an example, one of the possible means of contamination of soils is from irrigating them with water accidentally contaminated with brines. GEOCHEM can be used to simulate the change in trace metal speciation when varying quantities of brine mix with irrigation water. One example of a computed metal speciation in a mixture of brine and irrigation water is shown in Table V. The computations predict that the alkali metals will be present mainly as free ions. Among the alkaline earth metals, Ca, Mg, and Sr are present principally as free ions; a minor part of these metals is predicted to be present as the $CaSO_4^{\circ}$, $MgSO_4^{\circ}$, and $SrSO_4^{\circ}$ species. Also, a minor portion of Mg is predicted to be in the form of a solid borate. However, the bulk of Ba is predicted to precipitate as barite ($BaSO_4$), with some as solid $BaAsO_4$. The metals present in trace quantities such as Cu, Ni, Zn, and Pb, are predicted to be present principally as $MHCO_3^+$ and MCO_3° complexes, the exception being the metal Cd, which is present mainly as Cd^{2+} and $CdCl^+$. Regarding boron and arsenic, the calculations predict that the solution concentrations of these elements would be controlled by the solubility of solid phases such as $Mg_2B_2O_5 \cdot H_2O$ and $BaAsO_4$. (In simulating the trace metal speciation when soils

Table V.
Calculated metal and ligand speciation in a mixture of brine and irrigation water

Component	Total Analytical Conc. (-log M)	Dominant Species*
Ca	2.63	Ca^{2+}, $CaSO_4^o$
Mg	3.08	Mg^{2+}, $Mg_2B_2O_5 \cdot H_2O$ (Solid), $MgSO_4^o$
Sr	5.18	Sr^{2+}, $SrSO_4^o$
Ba	6.08	$BaSO_4$(Solid), $BaAsO_4$(Solid)
K	3.28	K^+
Na	1.82	Na^+
Cs	6.59	Cs^+
Li	4.18	Li^+
Fe	6.19	$Fe(CO_3)_3^{3-}$
Mn	6.59	Mn^{2+}, $MnHCO_3^+$, $MnSO_4^o$
Cu	6.76	$CuHCO_3^+$, $CuCO_3^o$
Cd	7.92	Cd^{2+}, $CdSO_4^o$, $CdCl^+$
Zn	6.82	Zn^{2+}, $ZnHCO_3^+$, $ZnSO_4^+$
Ni	6.64	Ni^{2+}, $NiCO_3^o$, $NiHCO_3^+$
Pb	6.63	Pb^{2+}, $PbCO_3^o$, $PbHCO_3^+$
Al	7.00	$Al_2Si_2O_5(OH)_4$(Solid), $Al(OH)_3^o$
CO_3	2.49	HCO_3^-, $H_2CO_3^o$
SO_4	2.48	SO_4^-, $CaSO_4^o$, $NaSO_4^-$
Cl	1.97	Cl^-
F	4.66	F^-
PO_4	6.72	$CaHPO_4^o$, $MgHPO_4^o$, $NaHPO_4^-$, HPO_4^{2-}, $H_2PO_4^-$
Si	3.59	$Si(OH)_4^o$
B	4.05	$Mg_2B_2O_5 \cdot H_2O$ (Solid)
As	6.80	$BaAsO_4$(Solid)
Se	7.10	$CaSeO_3^o$, $MgSeO_3^o$, $HSeO_3^-$
NO_3	4.61	NO_3^-

* only those species which constitute greater than 5% of the respective total analytical concentrations are listed.

are irrigated with these contaminated waters, additional reactions such as ion exchange and adsorption would have to be included, of course.) It is envisioned that these types of computations of the speciation of metals and ligands would be helpful in predicting the effects of brine spillage on the trace metal solubility and mobility in the soils of the Imperial Valley.

There is also an urgent need to know the possible deleterious effects of sewage sludge application on agricultural land. One of the potential problems arising out of the disposal of sewage sludge is the likelihood of accumulation of trace metals to toxic levels in soils. This concern is reflected in a recent report by Page (51), which examines the multiplicity of fates for trace elements introduced into agricultural soils by the land application of municipal wastes. One possible means of assessing the effects of sewage sludge application would be to simulate the probable speciation of the various elements with a computer program such as GEOCHEM. Table VI lists a part of the result of a simulation, using GEOCHEM, regarding the probable trace metal distribution in soil solutions extracted from soils

Table VI.
Distribution of metals and ligands (% of total) in saturation extracts of some sewage sludge amended soils

	-log Total conc.	Free metal	CO_3	SO_4	Cl	"Fulvate"
			San Miguel*			
Cu	5.69	19	14	5	<1	62
Cd	3.73	57	-	11	29	2
Zn	4.67	78	2	19	<1	<1
			Holtville†			
Cu	5.36	-	-	-	-	100
Cd	7.01	56	<1	8	27	8
Zn	4.67	70	9	13	-	5

Note: Other components included in the simulations were: Ca, K, Na, Fe, Mn, PO_4, NO_3, OH. These computations are based on "equilibrium" soil solutions and therefore, ion exchange and adsorption modeling are not included in the simulation.

*pH = 5.1 † pH = 7.6

amended with sewage sludge. These results subsequently were used in interpreting the observed patterns of Cd uptake by plants which were growing in these soils. It is seen from Table VI that Cu is present predominantly as a soluble organic complex in both soils, but roughly a fifth of total soluble copper in the San Miguel soil (pH=5.1) is present as Cu^{2+} ion. The computations indicate that, in the same soil, about half the soluble Cd would be free, whereas a third would be present as complexes with Cl. Even though the total soluble Cd in the Holtville Soil is very much smaller than the total soluble Cd concentration found in the San Miguel soil, the predicted relative proportion of Cd^{2+}, Cd bound to SO_4 and Cd bound to Cl are similar for both soils. In either soils, it is predicted that soluble Cd-organic complexes would not be significant. Similarly, soluble Zn-organic complexes are insignificant in either soil. It is also predicted that approximately 78% of the soluble Zn in the San Miguel soil would be in the form of Zn^{2+} ion and the rest would be present as $ZnSO_4^o$. But, in the Holtville soil, about 70% of the soluble Zn is in the free ionic form and about 9% and 13% of soluble Zn is predicted to be in the form of $ZnHCO_3^+$ and $ZnSO_4^o$, respectively.

An interesting aspect of these simulations is that the speciation of Cu, Cd and Zn with respect to inorganic and organic ("fulvate") ligands were qualitatively similar to what was observed in gel filtration studies (52). In these studies it was noticed that Cu was predominately bound to organics whereas Zn and Cd were observed to be predominately bound to inorganics.

The two examples of simulation of metal-ligand speciation in soil systems suggests that such studies will be a useful adjunct to the examination of trace metal chemistry in soils.

Acknowledgements

This research was supported in part by a grant from Kearney Foundation of Soil Science, in part by a contract from The Lawrence Livermore Laboratory as a contribution to its Imperial Valley Environmental Project, and in part from the Environmental Protection Agency under grant number R804516010. The contents of this paper do not necessarily reflect the views and policies of EPA. GEOCHEM will be available from the authors, on request, after January 1979.

Abstract

The computer program GEOCHEM is adapted and being developed for soil solutions from the REDEQL2 program originally created by Morel and Morgan at Caltech. GEOCHEM differs from REDEQL2 in its inclusion of 1) data for a few hundred additional soluble complexes and solids that are relevant to trace metal studies, 2) a subroutine for cation exchange that is based in thermodynamics, and 3) a subroutine for the estimation of single-ion activity coefficients at ionic strengths up to 3 M. Four categories of important theoretical problems were confronted in connection with the development of GEOCHEM. These problems were the lack of data regarding (a) stability constants of trace metal complexes with many important inorganic and mixed ligands and (b) stability constants of trace metal complexes with naturally occurring organic ligands; (c) solubility product constants for soil clay minerals, and (d) thermodynamic exchange constants and exchanger phase activity coefficients. The resolution of these problems has been discussed. Two representative applications of GEOCHEM in its current form to the calculation of trace metal equilibria in a mixture of irrigation water and a geothermal brine and in the aqueous phase of a sewage sludge amended soil are presented and discussed.

Literature Cited

1. McDuff, R.E., and Morel, F.M.M. Description and use of the chemical equilibrium program REDEQL2. Tech. Rpt. EQ-73-02. W.M. Keck Lab, Caltech 75 p. (1974).
2. Baes, C.F., Jr., and Mesmer, R.E. "The Hydrolysis of Cations." 489 p. John Wiley and Sons, New York, 1976.
3. Bjerrum, J., Schwarzenbach, G., and Sillen, L.G. "Stability Constants of Metal-Ion Complexes, with Solubility Products of Inorganic Substances." 131 p. The Chem. Soc., No. 17, London, 1958.
4. Martell, A.E., and Smith, R.N. "Critical Stability Constants," 257 p. Plenum Press, New York 1976.
5. Sillen, L.G., and Martell, A.E. "Stability Constants of Metal-Ion Complexes." 754 p., The Chem. Soc. London, 1964.
6. Sillen, L.G., and Martell, A.E. "Stability Constants of Metal-Ion Complexes." 865 p. The Chem. Soc. London, 1971.
7. Yatsimirskii, K.B., and Vasilev, V.P. "Instability Constants of Complex Compounds. 214 p. Consultants Bureau, New York, 1960.
8. Kharaka, Y.K., and Barnes, I. SOLMNEQ: Solution-mineral equilibrium computations, NTIS #PB215899, 75 p. 1973.
9. Truesdell, A.H., and Jones, B.F. WATEQ, A computer program for calculating chemical equilibria of natural waters, NTIS #PB-220464, 73 p. 1973.

10. Dyrssen, D., and Wedborg, M. Equilibrium calculations of the speciation of elements in sea water, p. 181-195, in Goldberg, E.D., ed., "The Sea," John Wiley and Sons, New York, 1974.
11. Zirino, A., and Yamamoto, S. A pH-dependent model for the chemical speciation of copper, zinc, cadmium, and lead in sea water. Limnol. Oceangr. 17, 661-671 (1972).
12. Fuoss, R.M. Ionic association III. The equilibrium between ion-pairs and free ions. J. Amer Chem. Soc. 80, 5059-5061 (1958).
13. Kester, D.R., and Pytkowicz, R.M. Theoretical model for the formation of ion-pairs in sea water. Mar. Chem. 3, 365-374 (1975).
14. Garrels, R.M., and Christ, C.L. "Solution, Minerals, and Equilibria." 450 p., Harper and Row, New York, 1965.
15. Nieboer, E., and McBryde, W.A.E. Free energy relationships in coordination chemistry I. Linear relationships among equilibrium constants. Can. J. Chem. 48, 2549-2564 (1970).
16. Nieboer, E., and McBryde, W.A.E. Free energy relationships in coordination chemistry II. Requirements for linear relationships. Can. J. Chem. 48, 2565-2573 (1970).
17. Nieboer, E., and McBryde, W.A.E. Free energy relationships in coordination chemistry III. A comprehensive index to complex stability. Can. J. Chem. 51, 2512-2524 (1973).
18. Van Panthaleon Van Eck, C.L. On the mutual stability relationship of MeA complexes consisting of a metal ion Me and a ligand A in aqueous solution, for consequetive values of n. Recl. Trav. Chim. Pays-Bas. 72, 529-537 (1953).
19. Mattigod, S.V., and Sposito, G. Estimated association constants for some complexes of trace metals with inorganic ligands. Soil Sci. Soc. Am. J. 41, 1092-1097 (1977).
20. Ramamoorthy, S., and Manning, P.G. Equilibrium studies of metal-ion complexes of interest to natural waters - VIII. J. Inorg. Nucl. Chem. 36, 695-698 (1974).
21. Dyrssen, D., Jagner, D., and Wengelin, F. "Computer Calculation of Ionic Equilibria and Titration Procedures." 250 p. Almquist and Wiksell, Stockholm, 1968.
22. Schindler, P., Reinert, M., and Gamsjager, H. "Loslichkeitkonstanten und Breie Bildungenthalpie von $Cu_2(OH)_2CO_3$ (Malachit) and $Cu_3(OH)_2(CO_3)_2$ (Azurit) bei 25°C. Helv. Chim. Acta. 51, 1845-1856 (1968).
23. Bilinski, H., Huston, R., and Stumm, W. Determination of the stability constants of some hydroxo and carbonate complexes of Pb (II), Cu (II), Cd (II) and Zn (II) in dilute solutions by anodic stripping voltammetry and differential pulse polarography. Anal. Chim. Acta. 84, 157-164 (1976).
24. Morgan, J.J. Chemical equilibria and kinetic properties of manganese in natural waters. p. 561-624, in Faust, S.D., Hunter, J.V., eds. "Principles and Application of Water Chemistry." John Wiley and Sons, New York, N.Y., 1965.

25. Smith, R.M., and Alberty, R.A. The apparent stability constants of ionic complexes of various adenosine phosphates with divalent cations. J. Amer. Chem. Soc. 78, 2376-2380 (1956).
26. Nriagu, J.O. Stability of vivianite and ion-pair formation in the system $Fe_3(PO_4)_2$-H_3PO_4-H_2O. Geochim. Cosmochim. Acta. 36, 459-470 (1972).
27. Sigel, H., Becker, K., and McCormick, D.B. Ternary complexes in solution. Biochem. Biophys. Acta. 148, 655-664 (1967).
28. Nriagu, J.O. Solubility equilibrium constant of α-Hopeite Geochim. Cosmochim. Acta. 37, 2357-2361 (1973).
29. Sposito, G., and Holtzclaw, K.M. Titration studies on the polynuclear, polyacidic nature of fulvic acid extracted from sewage sludge-soil mixtures. Soil Sci. Soc. Amer. J. 41, 330-336 (1977).
30. Sposito, G., Holtzclaw, K.M., and Keech, D.A. Proton binding in fulvic acid extracted from sewage sludge-soil mixtures. Soil Sci. Soc. Amer. J. 41, 1119-1125 (1977).
31. Chen, Chao-Hsia. A method for estimation of standard free energies of formation of silicate minerals at 298.15 K. Amer. J. Sci. 275, 801-807 (1975).
32. Karpov, I.K., and Kashik, S.A. Computer calculation of standard isoparic-isothermal potentials of silicates by multiple regression from a crystallochemical classification. Geochem. Int. 5, 706-713 (1968).
33. Tardy, Y., and Garrels, R.M. Prediction of Gibbs energies of formation of compounds from the elements. II. Monovalent and divalent metal silicates. Geochim. Cosmochim. Acta. 41, 87-92 (1977).
34. Nriagu, J.O. Thermochemical approximations for clay minerals. Amer. Mineral. 60, 834 (1975).
35. Tardy, Y., and Garrels, R.M. A method for estimating the Gibbs energies of formation of layer silicates. Geochim. Cosmochim. Acta. 38, 1101-1116 (1974).
36. Mattigod, S.V., and Sposito, G. Improved method for estimating the standard free energies of formation of smectites. Geochim. Cosmochim. Acta. (In press).
37. Kittrick, J.A. Montmorillonite equilibria and the weathering environment. Soil Sci. Soc. Amer. Proc. 35, 815-820 (1971).
38. Kittrick, J.A. Stability of monmorillonites: II. Aberdeen montmorillonite. Soil Sci. Soc. Amer. Proc. 35, 820-823 (1971).
39. Carson, C.D., Kittrick, J.A., Dixon, J.B., and McKee, T.R. Stability of soil smectite from a Houston black clay. Clays and Clay Min. 24, 151-155 (1976).
40. Misra, V.K., and Upchurch, W.J. Free energy of formation of beidellite from apparent solubility measurements. Clays and Clay Min. 24, 327-331 (1976).
41. Weaver, R.M., Jackson, M.L., and Syers, J.K. Magnesium and silicon activities in matrix solutions of montmorillonite containing soils in relation to clay mineral stability. Soil Sci. Soc. Amer. Proc. 35, 823-830 (1971).

42. Weaver, R.M., Jackson, M.L., and Syers, J.K. Clay mineral stability as related to activities of aluminum, silicon and magnesium in matrix solution of montmorillonite-containing soils. Clay and Clay Min. 24, 246-252 (1976).
43. Huang, W.H., and Keller, W.D. Gibbs free energies of formation calculated from dissolution data using specific mineral analysis. III. Clay minerals. Amer. Mineral. 58, 1023-1028 (1973).
44. Sposito, G., and Mattigod, S.V. On the chemical foundation of the sodium adsorption ratio. Soil Sci. Soc. Amer. J. 41, 323-329 (1977).
45. Gaines, G.L., Jr., and Thomas, H.C. Adsorption studies on clay minerals. II. A formulation of the thermodynamics of exchange adsorption. J. Chem. Phy. 21, 714-718 (1953).
46. Foscolos, A.E., and Barshad, I. Equilibrium constants between both freshly prepared and aged H montmorillonites and chloride salt solutions. Soil Sci. Soc. Amer. Proc. 33, 242-247 (1969).
47. Peigneur, P., Maes, A., and Cremers, A. Heterogeneity of charge density distribution in montmorillonite as inferred from cobalt adsorption. Clays and Clay Min. 23, 71-75 (1975).
48. Gast, R.G. Alkali metal cation exchange on Chambers montmorillonite. Soil Sci. Soc. Amer. Proc. 36, 14-19 (1972).
49. Sposito, G., and Mattigod, S.V. Ideal behavior in Na - trace metal cation exchange on Camp Berteau montmorillonite. Clays and Clay Min. (In press).
50. Helgeson, H.C. Thermodynamics of hydrothermal systems at elevated temperatures and pressures. Amer. J. Sci. 267, 729-804 (1969).
51. Page, A.L. Fate and effects of trace elements in sewage sludge when applied to agricultural lands. A literature review study. Rept. Office Res. Dev. U.S. Environ. Protection Agency, Cincinnati, Ohio, 97 p., 1974.
52. Sposito, G., Holtzclaw, K.M., and Baham, J. Analytical properties of the soluble, metal-complexing fractions in sludge-soil mixtures: II. Comparative structural chemistry of fulvic acid. Soil Sci. Soc. Amer. J. 40, 691-697 (1976).

Received November 16, 1978.

38

A Comparison of Computerized Chemical Models for Equilibrium Calculations in Aqueous Systems[1]

D. K. NORDSTROM', L. N. PLUMMER', T. M. L. WIGLEY', T. J. WOLERY', J. W. BALL', E. A. JENNE', R. L. BASSETT', D. A. CRERAR', T. M. FLORENCE', B. FRITZ", M. HOFFMAN", G. R. HOLDREN, JR.", G. M. LAFON", S. V. MATTIGOD", R. E. McDUFF", F. MOREL", M. M. REDDY", G. SPOSITO", and J. THRAILKILL"

The interpretation of chemical processes in aqueous systems requires the use of modern electronic computers, particularly in the calculation of multicomponent, multiphase equilibria. Commonly, the first concern of solution chemists and aqueous geochemists is to calculate the distribution and activities of species on the assumption that equilibrium exists throughout the aqueous phase. Species distribution can then be used in several areas of analytical and applied chemistry, e.g. to examine the availability of free and reactive ions, to test solubility hypotheses, and to determine the potential bioavailability of nutrients or toxic substances. Species distribution also forms the basis for more complex computations involving solutions which change composition by reaction with other solutions and with gases and solids. Equilibrium calculations of this type are particularly helpful in solving interpretive problems encountered in such fields as chemical and environmental engineering, geochemistry, biochemistry and aquatic ecology.

This symposium demonstrates quite clearly that we depend heavily on chemical models, especially computerized models, to interpret aqueous chemical processes. Several computer programs which solve problems of simultaneous chemical equilibria are being used by a rapidly increasing number of investigators and it is necessary to review the inherent assumptions and limitations of these aqueous models. There is a temptation to use these models as ready-made interpretations

[1] Current addresses for authors can be found on page 892.

0-8412-0479-9/79/47-093-857$09.00/0

SUBJ
COMP
DOES

DEVELOPMENT OF EXPERT SYSTEM APPLICATIONS USING PERSONAL CONSULTANT™

by

William M. Turpin
Manager, AI Applications
Texas Instruments Data Systems Group
Austin, Texas

During the past year and a half since the introduction of Personal Consultant in 1984, we at Texas Instruments have had an opportunity to work with and observe what many customers have done with the product. Additional customer understanding has been gained through our training classes and knowledge engineering services. These observations, plus our own involvement in expert system creation, have helped us to better understand the product features and management techniques which contribute to successful projects.

This paper presents four case histories of how actual expert system applications were created using Personal Consultant. They were selected from actual customer applications as being representative of various issues involved in expert system creation. These case histories show how varied expert system applications are in terms of their problem domain and intended user community. They also show different strategies for learning about and managing expert system technology.

CASE HISTORY #1 - CAMPBELL SOUP COOKER DIAGNOSIS

The first case history involves a traditional application of expert system technology. It involves a diagnostic domain which is well suited to rule-based backward chaining systems. It was also produced using a knowledge engineer working in conjunction with a single domain expert.

This application was done for manufacturing plants at the Campbell Soup Company. These plants use giant product sterilizers, commonly called cookers, in the production of soup products. Each cooker processes several hundred cans of soup at a time. So when one malfunctions, it is very important to get the problem diagnosed and fixed quickly. If a fix for the cooker is not found soon enough, then you "lose the soup".

Many routine cooker problems can be handled by plant operators on a day-to-day basis. Occasionally, however, difficulties arise that require the advice of an expert--someone who understands the design, installation, and operation of the equipment. Unfortunately, there are not enough human experts to go around. This can result in long delays while an expert is located and flown in. To make matters worse, Campbell's best expert was nearing retirement.

- more -

Campbell's management decided to use this problem situation as a test of the viability of expert systems. They contracted with Texas Instruments to provide knowledge engineering in the creation of this expert system. Campbell supplied the domain expert, while TI provided the knowledge engineer. The two would meet three to four days a month to design and review the evolving system.

After several months, a prototype system of about 30 rules was developed. It did not have much depth of knowledge, but it demonstrated that the expert system approach was feasible. It also served as a catalyst for adding additional knowledge to the system. After about 150 rules had been entered, the system was ready for production use. At the time of this writing, the system is being used at eight of Campbell's manufacturing facilities.

The Campbell Soup expert system was created using the Personal Consultant Expert System Tools to run on IBM PC-XT computers. The Personal Consultant provides several features which were important in this application:

- An easy-to-use interface so that plant maintenance personnel can quickly learn to use the system

- English explanation facilities help facilitate working with the domain expert and provide ongoing training for the maintenance personnel

- Multiple runtime versions have been created to distribute the system to several plants while maintaining central control of the knowledge base

Another important aspect of this application was Campbell's decision to get help in the creation of their first expert system application. By using a low-cost, PC-based expert system shell and the consulting services of a knowledge engineer, they were able to significantly increase their chances for success. They were also able to get involved with expert system technology in a very timely manner. This has allowed them to prove the feasibility of an expert system approach and stay competitive in their product market.

CASE HISTORY #2 - TEXAS INSTRUMENTS EPI REACTOR DIAGNOSIS

The second case history involves the use of an expert system for a similar diagnostic task. However, it is quite different in the manner that it was produced. It was created using multiple domain experts and a self-taught knowledge engineer.

This system is an equipment diagnosis situation at a semiconductor manufacturing facility within Texas Instruments. The manufacturing process for semiconductors uses many different machines to produce

silicon "wafers" with intricate patterns etched in layers to produce complex electronic circuits. Each machine is very expensive and operates with very small tolerance levels.

At the TI manufacturing facility in Sherman, Texas, one such machine, called an epitaxial reactor, is an essential step in the production of semiconductors. These reactors use very high temperatures and controlled mixtures of gases to grow a very thin silicon film (the epitaxial layer) on silicon wafers. Since the reactors process several wafers at a time, an equipment failure can produce a large amount of material scrap and hold up the rest of the manufacturing process.

In September 1984, this machine was selected as the most serious bottleneck in the manufacturing process, and an expert system approach was selected to help reduce downtime. It was felt that there were a large number of problems that the machine operators and local maintenance staff could remedy with appropriate help from an expert system. If successful, such a system could save thousands of dollars.

The programmer/analyst assigned to this project had no previous AI or LISP programming experience but was willing and anxious to learn. After selecting the Personal Consultant system as her programming tool, the new knowledge engineer attended a one-week course on the system. This gave her the initial training necessary to go back to the work environment and concentrate on her application.

To obtain the necessary expertise about the epi reactors, two committees were formed. The first was a technical committee charged with defining the scope of the system, the problems to attempt to diagnose, and the diagnostic procedures. It was composed of the knowledge engineer, a manufacturing engineer, several service technicians, and a service representative from the manufacturer of the epi reactor. They decided to initially focus on common problems which a reactor could help identify through its operator panel.

The second committee was formed of the planned users of the system. It included the knowledge engineer, service technicians, and machine operators. They helped determine how the system would fit into the actual work environment and how users would be trained. They would go in on weekends to test the operation of the prototype system in the production environment.

After six months, the development team had created a 400 rule system called the Intelligent Machine Prognosticator (IMP) which could suggest solutions to 25 common problems. By the time of this writing, the system had grown to about 1000 rules (segmented into seven knowledge bases) covering about 300 reactor problems. This system has been installed on a portable TI Professional Computer which is

rolled to the epi reactors as needed on a moveable cart. The results so far have been an outstanding forty-four percent increase in average time between reactor failures. Work is also in progress on creating expert systems for other selected machines at this facility.

This application was facilitated by several Personal Consultant features. The easy-to-use interface and ability to make runtime versions for delivery were both essential. The product training also helped the new knowledge engineer gain experience and confidence more quickly. The English explanation of the system's operation also helps train the maintenance personnel and instill user confidence in the system.

The use of two committees was also a key element in the success of the IMP system. The pooling of several experts' knowledge helped make the system more intelligent. The inclusion of the system's eventual users during the design phase was also instrumental in getting the system accepted in the work environment. Many people overlook the fact that an expert system is useless unless the people it is intended to help come to it for advice.

PERSONAL CONSULTANT APPLICATION SUMMARY

CASE HISTORY	1	2	3	4
WHO	Cambell Soup	Texas Instruments	Westinghouse	Purdue/USDA
DOMAIN	Soup Manufacturing	Semiconductor Manufacturing	Powerplant Design	Grain Marketing
TASK	Equipment Diagnosis	Equipment Diagnosis	Material Selection	Marketing Advise
PROBLEM	Expert Retiring	Expert Not Available	Knowledge Transfer	Knowledge Transfer
END USER	Technician	Technician	Engineer	Farmer
KNOWLEDGE ENGINEER	Hired Consultant	Programmer/Analyst	Domain Expert	Graduate Students
SIZE	150 Rules	1000 Rules	300 Rules	150 Rules
FEATURES	Runtime	Runtime	LISP functions Frames Graphics	LISP functions Data Access Graphics
STATUS	Production Usage	Production Usage	Refinement	Refinement

CASE HISTORY #3 - WESTINGHOUSE CORROSION EXPERT

The next case history involves a domain in which the expert system gives design advise. It was produced by the domain expert himself who

learned the necessary computer programming skills. The resulting system is intended to help engineers design better, longer lasting products.

This expert system was created in the Research and Development labs at Westinghouse. The subject of the system is the selection of metal alloys for the construction of steam generators in nuclear power plants. A researcher there had discovered how to combine dissimilar metal alloys to produce a component with specific corrosive properties--which were sometimes different from either of the metals alone. He was faced with the choice of how to disseminate this information to the appropriate engineers at Westinghouse. His solution was in the form of an expert system that could help further his research while providing a good medium to transfer the knowledge.

This metalurgical expert had an interest in computers but did not have any formal computer education. Therefore, after selecting Personal Consultant as his expert system tool, his first step was to attend training classes on the product. With this training, he was able to go back to Westinghouse and create several small expert systems--one of which helped salespeople configure electrical panel boards. During this period, he also started teaching himself LISP so that he could add custom features into his expert systems.

At the time of this writing, this system is still being expanded and refined. In particular, the researchers are studying the discrepancies between theoretical relationships and empirical evidence--both of which have been encoded into production rules. When these two systems of logic produce conflicting advice, it helps suggest an area needing further research.

This project has used several features of Personal Consultant in order to achieve its results:

- Reliance on certainty factor calculations to incorporate and report imprecise knowledge
- Extensive use of graphics has been employed to enhance the display of conclusions
- The problem domain was structured into modular components (frames) to make it more manageable
- User defined LISP functions have been written to extend the functionality of the system
- The symbolic form of the rules (not compiled) has been useful in finding and reporting conflicting knowledge

Other features, such as the easy-to-use interface and explanation facilities, will help the transfer of knowledge to other engineers at Westinghouse.

This application also demonstrates that it is sometimes possible to train the domain expert on expert system creation. The non-procedural style of rule-based expert systems can simplify the knowledge encoding process much like computer spreadsheets simplify the manipulation of tabular, numeric data. If the domain expert is willing and able to learn about expert systems, he can be very effective at creating applications in his domain.

CASE HISTORY #4 - PURDUE GRAIN MARKET ADVISOR

The last case history involves an application in the domain of marketing. It was produced in a university setting by a team of knowledge engineers and domain experts. It adds some new features to an expert system in order to make the system more capable.

In 1985, Purdue University received a grant from the United States Department of Agriculture (USDA) to study improvements that could be made in the US farm industry. As a result of this study, Purdue has created several expert systems which can help individual farmers become more productive. One of these systems, called the Grain Market Advisor, helps farm operators determine the best method of marketing the grain they produce.

The grain marketing process is a confusing problem to many farmers because of price uncertainties, commodity markets, and joint marketing alternatives. Yet, the marketing strategy employed by the farmer can have a tremendous impact on the selling price of a crop. A good marketing strategy can also help reduce some of the risk involved in farming. In order to give good, consistent advise on grain marketing, Purdue decided to create an expert system.

During a consultation with GMA, grain price history information is retrieved from a commercial data base via a dial-up telephone line. This price information is used to help predict future price trends. Graphics are produced from this data to help convey these price trends to the user. Given these trends, the system helps decide which risk-sharing strategies, such as selling grain futures, are relevant. These strategies are then tested to see if they can be used in the current situation.

Before starting work on GMA, Purdue sent a graduate student from the project to live and work with the Personal Consultant development team. During the Spring of 1985, the student was able to learn about the product and develop user interface routines which would be helpful to the project. This cooperative effort proved to be beneficial to both parties involved. Back at Purdue, this student became the lead knowledge engineer for the project. The resulting expert systems, including the GMA, were presented to the USDA last December.

Custom LISP functions were used extensively in GMA to perform the data base access and present graphical information. These additional

features allow the system to be more capable and easier to understand. When combined with the standard explanation facilities, the system becomes very transparent and capable of explaining its reasoning. This will be very important to the user of the system who will be pricing an entire year's crop production based on advise from the system.

Certainty factors were employed to represent imprecise knowledge, determine default values, and rank competing grain marketing alternatives. The easy-to-use interface will be essential when dealing with users who do not work with computers on a day-to-day basis. The price point and availability of personal computers coupled with the ability to make runtime versions of the application make this system feasible for mass distribution within the farming industry.

SUMMARY

In addition to being successful, these four case histories share several common themes relating to PC-based expert system development. These projects, along with others, can be used to generalize the following observations:

- Many organizations are getting involved with and creating expert system applications
- Useful, cost-effective expert systems are being produced for use on personal computers
- Current expert system technology can be applied in many different problem domains
- Many successful projects start with a small prototype which is extended and refined over time
- Companies without prior AI experience are being successful by getting assistance

These case histories also demonstrate that the creation of expert systems is no longer limited to high-tech, early adopter companies. The availability of powerful PC-based tools, such as Personal Consultant, have significantly reduced the risk and training involved in creating useful applications in many problem domains. The development of expert systems is quickly moving from AI research labs into business practice at many companies. Soon, they will be a standard method of improving quality and managing flow of knowledge within an organization.

##

COMPUTER SEARCHABLE BIBLIOGRAPHIC DATABASES IN THE PHYSICAL AND LIFE SCIENCES AND ENGINEERING

AVAILABLE

AT THE

MARRIOTT LIBRARY

UNIVERSITY OF UTAH

Compiled by

Ann Marie Breznay

September 1979

FEDERAL REGISTER, March 1977-present, weekly updates.
SDC

Corresponding to *Federal Register Abstracts*, this file covers rules, regulations, notices, meetings, and hearings of the Federal Government on a range of subjects, including defense, environment, technology and transportation. See also PTS FEDERAL INDEX.

FOUNDATION DIRECTORY, current file, semiannual revisions.
LRS (26)

This file provides descriptions of nonprofit, nongovernmental foundations, which make grants of $100,000 or more annually or which have assets of $1 million or more. Grants are given primarily in education, health, welfare, sciences, international activities and religion, in that order. Corresponds to the printed *Foundation Directory*.

FOUNDATION GRANTS, 1972-present, bimonthly updates.
LRS (27)

Also published by the Foundation Center, this file contains information on grants awarded by major American philanthropic foundations in many fields, including health and the sciences. Grants of less than $5,000 and grants to individuals are not included.

GPO MONTHLY CATALOG, 1973-present, monthly updates.
LRS (66)

This file corresponds to the printed *Monthly Catalog of U.S. Government Publications* and is an index of public documents generated by the U.S. Federal Government. It covers Congressional documents, Senate and House hearings on private and public bills and laws, as well as documents from other government agencies and departments on every major issue.

GRANTS, 1977-present, monthly updates and revisions.
SDC

In addition to grants awarded by private foundations, this file includes information on grants offered by federal, state and local governments, commercial organizations and associations in over 88 academic disciplines. Its print counterparts are *Grant Information System (GIS)* and *Faculty Alert Bulletin*.

LIBCON, 1968-present (MARC), 1969-present (MET), weekly updates.
SDC

This file includes Library of Congress English language records from MARC tapes and MET (main entry/title) records from LC depository card sets. Covers all subject areas.

MAGAZINE INDEX, 1977-present, monthly updates.
LRS (47)

This file provides coverage of over 370 popular magazines, including the 173 indexed in the *Reader's Guide to Periodical Literature*, on virtually every subject from current events, editorials, and fiction to consumer products, nutrition and technology.

NTIS, 1964-present, biweekly updates.
BRS, LRS (6), SDC

The National Technical Information Service database consists of technical reports on both hard and soft sciences which result from research sponsored or funded by any branch or agency of the U.S. Government. Includes analyses prepared by federal agencies, their contractors or grantees, and nonclassified DOD reports. Corresponds to the printed *Weekly Government Abstracts* and *Government Reports Announcements*.

NATIONAL FOUNDATIONS, 1975-present, annual updates and revisions.
LRS (78)

Records of all 20,000 U.S. foundations which award grants are provided in this file. Approximately 17,000 small foundations which are excluded from coverage in the FOUNDATION DIRECTORY because their assets total less than $1 million are listed in NATIONAL FOUNDATIONS, including those foundations which restrict their donations to local or regional recipients. This file overlaps and supplements the FOUNDATION DIRECTORY file (LRS 26), where more complete data on the larger foundations are available.

SCISEARCH, 1974-present, biweekly updates.
LRS (34:1978-present; 94:1974-1977)

This file, a multidisciplinary index to the literature of science, technology and medicine, contains all the records published in the *Science Citation Index*. Over 2,600 major journals are indexed, which results in the inclusion of 90 percent of the world's significant scientific and technical literature.

SSIE, fiscal year 1974-present, monthly updates.
LRS (65: last two fiscal years only), SDC

The Smithsonian Science Information Exchange file covers research in progress and recently completed research sponsored primarily by agencies of the federal government. Provides summaries of the research projects including basic and applied research in all areas of the life, physical, social, behavioral and engineering sciences.

U.S. CONTRACT AWARDS, June 1978-present, monthly updates.
SDC

This file corresponds to the printed title and provides information about contracts awarded by the U.S. Government in various fields.

WORLD PATENT INDEX, 1977-present, monthly updates.
SDC

Central Patents Index and *World Patents Index* are the printed correspondents of this file, which contains data related to patent specifications issued by the Patent Offices of the major industrial nations. Subjects covered include pharmaceuticals, agricultural chemicals, plastics, chemistry, mechanical and electrical.

BIOLOGICAL AND LIFE SCIENCES

AGRICOLA, 1970-present, monthly updates.
BRS, LRS (10), SDC

Produced by the National Agricultural Library of the U.S. Dept. of Agriculture, this database provides comprehensive worldwide coverage in the field of agriculture and such related fields as rural sociology, agricultural economics, veterinary medicine, plant science, agricultural chemistry, entomology, soils, water management, food and nutrition. Corresponds to the printed *Bibliography of Agriculture*.

AQUACULTURE, 1970-present, irregular updates.
LRS (112)

This file includes information on growing water organisms.

AQUATIC SCIENCES AND FISHERIES ABSTRACTS (ASFA), 1975-present, monthly updates.
LRS (44)

ASFA is a comprehensive database on life sciences of the seas and inland waters, as well as related legal, political and social topics, such as biological and chemical oceanography, law of the sea, water pollution, geological and physical oceanography and limnology.

BIOCODES, 1969-present, irregular updates.
SDC

Dictionary file of the biosystematic (taxonomic) codes and cross (concept) codes used in *Biological Abstracts* and BIOSIS PREVIEWS.

BIOSIS PREVIEWS, 1969-present, monthly updates.
BRS, LRS (5:1972-present; 55:1969-1971), SDC

This file, which corresponds to the printed *Biological Abstracts* and *Bioresearch Index*, covers all areas of the life sciences, including bioengineering, biophysics, genetics, behavioral sciences, pharmacology, public health, veterinary science and virology.

CAB ABSTRACTS, 1973-present, monthly updates.
LRS (50)

This comprehensive file of agricultural information contains all records in the journals published by the Commonwealth Agricultural Bureaux. Significant papers are abstracted and less important works are reported with bibliographic details only in every branch of agricultural sciences, including education, genetics, pest control and taxonomy.

CA CONDENSATES/CASIA, 1967-present, biweekly updates.
BRS, LRS (2,3,4), SDC

Corresponding to the printed *Chemical Abstracts* minus the abstracts, this file provides comprehensive coverage of the literature of chemistry and its applications. Patents, books, reference proceedings and government research reports are included; subject divisions include applied chemistry and chemical engineering, biochemistry, macromolecular chemistry, organic chemistry, physical and analytical chemistry.

EXCERPTA MEDICA, 1975-present, weekly updates.
LRS (72,73)

This database of worldwide research literature in the medical sciences includes all records in the printed *Excerpta Medica* plus an additional 100,000 records annually that do not appear in the printed journals. Covers all fields of medicine, as well as literature in pharmaceutics, forensic science, health economics, hospital management and public health.

FOOD SCIENCE AND TECHNOLOGY ABSTRACTS (FSTA), 1969-present, monthly updates.
LRS (51), SDC

This file, which corresponds to the printed *Food Science and Technology Abstracts*, covers literature related to all human food commodities and aspects of food processing except the production of raw foods. Allied disciplines such as agriculture, chemistry, biochemistry, physics, home economics and engineering are also covered.

MEDLARS, 1966-present, monthly updates.
BRS

Corresponds to the print *Index Medicus, Index to Dental Literature* and the *International Nursing Index* and covers all types of medical sciences literature.

NAL SERIALS, 1978-present, monthly updates.
BRS

This file provides a listing of all serial records of the National Agricultural Library.

USDA/CRIS, July 1974-present, quarterly updates.
LRS (60)

The Current Research Information System of the U.S. Department of Agriculture database provides information about current research in agriculture and related sciences, sponsored or conducted by USDA research agencies, State agricultural experiment stations, State forestry schools, and other cooperating State institutions.

CHEMICAL SCIENCES

APILIT, 1964-present, monthly updates.
SDC

This file, produced by the American Petroleum Institute, covers literature relating to the petroleum refining and petrochemical industry, including scientific and technical developments and engineering work.

APIPAT, 1964-present, monthly updates.
SDC

Also produced by the American Petroleum Institute, this database covers patents related to the petroleum refining and petrochemical industry that have been issued by the United States, Belgium, Canada, France, Germany, Great Britain, Holland, Italy, Japan and South Africa.

CA CONDENSATES/CASIA, 1967-present, biweekly updates.
BRS, LRS (2,3,4), SDC

This file corresponds to the printed *Chemical Abstracts* and provides comprehensive coverage of the literature of chemistry. Patents, books, reference proceedings and government research reports are included. Subject divisions are: applied chemistry and chemical engineering, biochemistry, macromolecular chemistry, organic chemistry, physical and analytical chemistry. From 1972 onward, the CASIA files are integrated into the LRS and SDC files, providing additional index entries and CAS Registry Numbers.

CA PATENT CONCORDANCE, 1972-present, semiannual updates.
LRS (43)

This file correlates patents issued by different countries for the same basic invention. The first patent is published in *Chemical Abstracts;* subsequent patents for the same invention are entered in CA PATENT CONCORDANCE.

CHEMDEX, 1972-present, quarterly updates.
SDC

This dictionary file covers all compounds cited in the literature. It can be searched by chemical name, molecular formula, or by group or row within the periodic chart.

CHEMNAME, 1972-present, irregular updates.
LRS (31)

This file contains a listing of chemical substances in a dictionary-type, nonbibliographic file. For each substance listed, the CAS Registry Number, molecular formula, CA Substance Index Name, available synonyms, ring data and other chemical substance data are listed.

CHEMICAL INDUSTRY NOTES (CIN), 1974-present, weekly (SDC) and biweekly (LRS) updates.
LRS (19), SDC

Corresponding to the printed *Chemical Industry Notes*, this database contains articles which cover the following chemical industry areas: production, pricing, sales facilities, products and processes, corporate activities, government activities, and people in the chemical industry.

CLAIMS/CHEM, 1950-1970, closed file.
CLAIMS/U.S. PATENTS, 1971-present, quarterly updates.
LRS (23,24)

These databases contain chemical and chemically related patents from the *Official Gazette* of the U.S. Patent Office. Foreign equivalents from Belgium, France, Great Britain, West Germany and the Netherlands are included for approximately 20% of the U.S. Patents in the file.

CLAIMS/CLASS, 1977-present, irregular updates.
LRS (25)

Intended for use in conjunction with the CLAIMS/CHEM and CLAIMS/U.S. PATENTS files, this database is a classification code and title dictionary for all classes and selected subclasses of the U.S. Patent Classification System.

COMPENDEX, 1970-present, monthly updates.
LRS (8), SDC

This machine-readable version of the *Engineering Index* provides information from the world's significant literature in engineering and related subject areas, including chemical engineering, materials' properties and testing. It covers about 3,500 journals, publications of engineering societies and organizations, papers from the proceedings of conferences, and selected government reports and books.

PAPERCHEM, 1969-present, monthly updates.
SDC

This database, which corresponds to the printed publication *Abstract Bulletin of the Institute of Paper Chemistry*, provides comprehensive worldwide coverage of the scientific and technical literature that deals with the theoretical principles, technologies, raw materials, products, and practices of the pulp-, paper-, and board-manufacturing and -utilizing industries.

ENGINEERING

BHRA FLUID ENGINEERING, 1974-present, quarterly updates.
LRS (96)

This file provides indexing and abstracting of world-wide information on all aspects of fluid engineering, including theoretical research as well as technology and applications. Fields covered include civil engineering, hydraulics, fluid flow, fluidics feedback and tribology. No print counterpart.

CLAIMS/U.S. PATENTS, 1971-present, quarterly updates.
LRS (23,24)

This database contains all patents listed in the general, chemical, electrical and mechanical sections of the *Official Gazette* of the U.S. Patent Office.

COMPENDEX, 1970-present, monthly updates.
LRS (8), SDC

Engineering Index is the printed version of this file, which covers the world's significant engineering and technological literature. It includes coverage of approximately 3,500 journals, publications of engineering societies and organizations, papers, conference proceedings, government reports and books.

INTERNATIONAL AEROSPACE ABSTRACTS, 1968-present.
NASA

This file is the same as the printed version and covers all literature related in any way to the aerospace industry.

INSPEC, 1969-present, monthly updates.
BRS, LRS (12,13), SDC

The online INSPEC file corresponds to the printed *Physics Abstracts*, *Electrical and Electronics Abstracts*, and *Computer and Control Abstracts* and provides coverage of the world's literature in physics, electrical and electronics engineering.

ISMEC, 1973-present, monthly updates.
LRS (14), SDC

The ISMEC BULLETIN is the print counterpart of this database, which covers leading international journals and conferences on all aspects of mechanical engineering, production engineering, and engineering management.

METADEX (METALS ABSTRACTS/ALLOYS INDEX), 1966-present, monthly updates.
LRS (32)

This file provides comprehensive coverage of international literature on the science and practice of metallurgy. Six basic categories of metallurgy are covered: Materials, processes, properties, products, forms, and influencing factors. The print versions are: *Review of Metal Literature* (1966-67), *Metals Abstracts* (1968-present), and *Alloys Index* (1974-present).

MRIS ABSTRACTS, 1970-present, monthly updates.
LRS (63)

The Maritime Research Information Service Abstracts database provides information on technical reports and journal articles in the maritime field, including marine engineering, corrosion, naval engineering and strength of materials and structural analysis.

PIRA, 1975-present, monthly updates.
LRS (48)

PIRA is a machine-readable version of *Paper and Board Abstracts, Printing Abstracts, Packing Abstracts,* and *Management and Marketing Abstracts.* It includes periodicals, books, standards, specifications, legislation, conference papers, research reports and other information in the fields of paper and board manufacturing.

RAPRA ABSTRACTS, 1972-present, monthly updates.
LRS (95)

RAPRA (Rubber and Plastics Research Association) is a comprehensive database covering the commercial, technical and research aspects of the rubber and plastics industries. Subjects covered include: synthesis and polymerization, processing technology, properties and testing.

SAE, 1965-present, quarterly updates.
SDC

This database, produced by the Society of Automotive Engineers, provides access to a select number of technical papers on the technology of the automotive and automotive-related industries.

SAFETY, June 1975-present, bimonthly updates.
SDC

Safety Science Abstracts Journal is the printed publication corresponding to this database, which covers periodicals, government reports, conference proceedings, books, dissertations and patents dealing with general safety, industrial and occupational safety, transportation safety, aviation and aerospace safety, environmental and ecological safety and medical safety.

STAR, 1962-present.
NASA

The *Scientific and Technical Aerospace Reports* file provides access to unclassified scientific and technical reports published by NASA on any aspect of the aerospace industry. STAR reports are available on microfiche in the Documents division.

TITUS, 1970-present, monthly updates.
SDC

This file, which has no printed counterpart, provides comprehensive worldwide coverage of topics relating to the textile industry. It includes information about processes and machines, analysis and testing of products and textile engineering.

WELDASEARCH, 1967-present, monthly updates.
LRS (99)

The WELDASEARCH database provides coverage of the international literature on all aspects of the joining of metals and plastics and related areas such as metals spraying and thermal cutting. Information is extracted from several thousand journals, research reports, books, standards, patents, theses and special publications.

WORLD TEXTILES, 1970-present, monthly updates.
LRS (67)

This is the machine-readable version of *World Textile Abstracts* and indexes world literature on the science and technology of textile and related materials, including the synthesis, physics and chemistry of polymers for fibers. Its coverage is roughly the same as that of the TITUS file.

ENVIRONMENT

APTIC, 1966-September 1978, closed file.
LRS (45)

The Air Pollution Technical Information Center database covers all aspects of air pollution, its effects, prevention and control, including the social, political, legal and administrative aspects of the field. Includes all entries from *Air Pollution Abstracts* (no longer published).

ENVIROLINE, 1971-present, monthly updates.
LRS (40), SDC

Environment Abstracts is the printed title corresponding to this file, which covers the world's environmental information. Over 5,000 source publications are covered in such fields as management, law, economics, geology, biology, and chemistry as they relate to the environment.

ENVIRONMENTAL IMPACT STATEMENTS, 1977-present, monthly updates.
BRS

This file indexes all environmental impact statements required by the Environmental Protection Agency. It includes information on the agency or organization involved, positive or negative impacts of the action and the alternatives considered.

EPB, 1973-present, bimonthly updates.
LRS (68)

The *Environmental Periodicals Bibliography* is the printed version of this file, which indexes over 250 periodicals to provide information about general human ecology, atmospheric studies, energy, land resources, water resources and nutrition and health.

OCEANIC ABSTRACTS, 1964-present, bimonthly updates.
LRS (28), SDC

This database, the online version of *Oceanic Abstracts*, covers worldwide technical literature on oceans, seas, and marine-related subjects, including geology, optics, mining, diving, ships, and laws and regulations.

POLLUTION ABSTRACTS, 1970-present, bimonthly updates.
BRS, LRS (41), SDC

Pollution Abstracts is the printed publication corresponding to this file, which contains references to environmentally related literature on pollution, its sources and its control.

ENERGY

APILIT, 1964-present, monthly updates.
SDC

This file, produced by the American Petroleum Institute, covers literature relating to the petroleum refining and petrochemical industry, including scientific and technical developments and engineering work.

APIPAT, 1964-present, monthly updates.
SDC

Included in this file, which is also produced by the American Petroleum Institute, are patents related to the petroleum refining and petrochemical industry.

CA CONDENSATES/CASIA, 1967-present, biweekly updates.
BRS, LRS (2,3,4), SDC

This file corresponds to the printed *Chemical Abstracts* and provides comprehensive coverage of the literature of chemistry, including energy chemistry literature and patents.

COMPENDEX, 1970-present, monthly updates.
LRS (8), SDC

This machine-readable version of the *Engineering Index* provides information from the world's significant literature in engineering and related subject areas, including transportation, mining engineering, petroleum engineering, nuclear technology, heat and thermodynamics.

ENERGYLINE, 1971-present, bimonthly updates.
LRS (69), SDC

Energy Information Abstracts is the print version of this file, which also includes energy-related records from *The Energy Index*. It provides information on scientific, technical, socioeconomic, governmental policy and planning and current affairs aspects of energy.

INSPEC, 1969-present, monthly updates.
BRS, LRS (12,13), SDC

The online INSPEC file corresponds to the printed *Physics Abstracts*, *Electrical and Electronics Abstracts* and *Computer and Control Abstracts* and provides coverage of the applied energy literature in physics, electrical and electronics engineering. Journal articles, government reports, patents, technical reports, books, conference proceedings and theses are included.

ISMEC, 1973-present, monthly updates.
LRS (14), SDC

ISMEC (Information Service in Mechanical Engineering) covers leading international journals and conferences on mechanical engineering subjects, including mechanical, nuclear, electrical, electronic, civil, optical, medical and industrial process engineering; mechanics; production processes, energy and power. Its print version is the *ISMEC Bulletin*.

P/E NEWS, 1975-present, weekly updates.
SDC

This database, which has no printed correspondent, covers energy and petroleum business news from five major sources: *Middle East Economic Survey*, *The Oil Daily*, *The Petroleum Economist*, *Petroleum Intelligence Weekly* and *Platts Oilgram News Service*.

TULSA, 1965-present, quarterly updates.
SDC

This database corresponds to the printed *Petroleum Abstracts* and provides coverage of literature and patents related to exploration, development and production of petroleum and other fossil fuels, including natural gas and coal.

GEOSCIENCES

CA CONDENSATES/CASIA, 1970-present, biweekly updates.
BRS, LRS (2,3,4), SDC

This file corresponds to the printed *Chemical Abstracts* and provides access to the technical literature for chemistry and chemical engineering, geochemistry and related fields.

GEOARCHIVE, 1969-present, monthly updates.
LRS (58)

Geophysics, geochemistry, geology, paleontology, and mathematical geology are covered in this database which indexes books, conferences, serials, dissertations, maps and technical reports. It has no printed equivalent, but among the publications printed from the information contained in the database are *Geotitles Weekly*, *Geocom Bulletin*, *Geoscience Documentation*, and the *Bibliography of Vertebrate Paleontology*.

GEOREF, 1961-present, monthly updates.
SDC

This geological reference database, produced by the American Geological Institute, provides coverage of the literature in about 30 different geoscience areas, including extraterrestrial geology, petrology, marine geology and geophysics. The file corresponds to several printed publications: *Bibliography and Index of North American Geology* (1961-70); *Bibliography and Index of Geology Exclusive of North America* (1967-68); *Geophysical Abstracts* (1966-71); *Bibliography of Theses in Geology* (1965-66); and *Bibliography and Index of Geology* (1969-present).

METEOROLOGICAL AND GEOASTROPHYSICAL ABSTRACTS, 1972-present, irregular updates.
LRS (29)

The MGA data file provides current citations in English for the most important meteorological and geoastrophysical research published in both foreign and domestic literature. Journals, monographs, proceedings and reviews are covered. Subjects include meteorology, astrophysics, hydrology, glaciology and related areas. Corresponds to the printed title.

OCEANIC ABSTRACTS, 1964-present, bimonthly updates.
LRS (28), SDC

This database, as does its print counterpart, covers the literature of oceanography and marine related subjects, including geology, pollution, meteorology, coastal resources, mining and laws and regulations.

HEALTH SCIENCES

ALCOHOL USE AND ABUSE, 1968-present, monthly updates.
BRS

This database provides an index to literature covering alcoholism and the effects of alcohol on the body. It deals primarily with the evaluation of treatment, the chemically dependent female, family therapy, and the MMPI, with minor emphasis on the elderly and the adolescent. No print counterpart.

DRUG INFO, 1968-present, monthly updates.
BRS

The literature on drugs, their chemistry and effects on the body is covered in this database, which focuses on the educational, sociological and psychological aspects of alcohol and drug use/abuse. Monographs, journals, conference papers, instructional guides and films are included.

EXCERPTA MEDICA, 1974-present, weekly updates.
LRS (72,73)

This file covers, as does the printed title, the biomedical literature. Nursing, dentistry, psychology, and the paramedical professions, such as podiatry and optometry are excluded, but coverage of articles on drugs and potential drugs is extensive and health-related sciences such as pollution control, forensic science and public health are included.

INTERNATIONAL PHARMACEUTICAL ABSTRACTS (IPA), 1970-present, bimonthly updates.
LRS (74)

The IPA file, which corresponds to the printed title, covers information on the pharmaceutical profession and the development and use of drugs. Coverage ranges from clinical, practical and theoretical to the economic and scientific aspects of the literature.

MEDLARS, 1966-present, monthly updates.
BRS

Corresponds to the printed *Index Medicus*, *Index to Dental Literature* and *International Nursing Index* and covers all types of medical literature, including psychiatry and pharmacology.

PHARMACEUTICAL NEWS INDEX (PNI), 1974-present, monthly updates.
BRS, LRS (42), SDC

This database covers the following industry newsletters to provide information about the pharmaceuticals industry, cosmetics, medical devices and related health fields: *FDC Reports (The Pink Sheet); Drug Research Reports (The Blue Sheet); Medical Devices, Diagnostics and Instrumentation Reports (The Gray Sheet); Weekly Pharmacy Reports (The Green Sheet); Quality Control Reports (The Gold Sheet); PMA Newsletter;* and *Washington Drug and Device Letter*.

POPULATION BIBLIOGRAPHY, 1966-present, bimonthly updates.
LRS (104)

Although more coverage is given to socioeconomic aspects, biomedical aspects of population, abortion, demography, family planning, fertility studies and population research are also covered. Journals, monographs, technical reports, government documents, conference proceedings, dissertations and unpublished reports are included; emphasis is on developing countries and the United States.

DATABASE	VENDOR	SUBJECT
APIPAT	SDC	Petroleum refining, petrochemical industry.
BHRA FLUID ENGINEERING	LRS (96)	Fluid engineering.
CA CONDENSATES/CASIA	BRS LRS (2,3,4) SDC	Chemistry and related areas.
CA PATENT CONCORDANCE	LRS (43)	Chemistry (cross-indexes chemical patent numbers.)
CLAIMS/CHEM	LRS (23)	Chemical and chemically related patents issued from 1950-1970.
CLAIMS/CLASS	LRS (25)	Classification code and title dictionary for all classes and selected subclasses of the U.S. Patent Classification System.
CLAIMS/U.S. PATENTS	LRS (24)	Chemical and chemically related patents issued from 1971-present.
ENVIROLINE	LRS (40) SDC	Environment and related fields.
FOODS ADLIBRA	LRS (79)	Food technology and packaging.
FSTA	LRS (51) SDC	Food science and technology.
INSPEC	BRS LRS (12,13) SDC	Physics, electrotechnology, computers and control
PAPERCHEM	SDC	Paper, pulp and board manufacturing and utilizing.
SAFETY	SDC	Safety topics in all fields.
TITUS	SDC	Textile and allied industries.
TULSA	SDC	Exploration, production and development of oil, coal and natural gas.
WELDASEARCH	LRS (99)	All aspects of the joining of metals and plastics.
WORLD ALUMINUM	LRS (33)	Aluminum, from ore processing through end uses.
WORLD TEXTILES	LRS (67)	All aspects of textile science and technology.
WPI	SDC	All patents issued by the world's major industrial nations.

PHYSICAL SCIENCES

CA CONDENSATES/CASIA, 1970-present, biweekly updates.
BRS, LRS (2,3,4), SDC

This file corresponds to the printed *Chemical Abstracts* and provides access to the technical literature for chemistry, chemical engineering, macromolecular chemistry, and physical and analytical chemistry.

COMPENDEX, 1970-present, monthly updates.
LRS (8), SDC

The computerized engineering index is the machine-readable version of the *Engineering Index* which covers the significant engineering literature of the world. Journals, publications of societies, proceedings and government reports are included; subjects covered include materials' properties and testing, fluid flow, heat and thermodynamics, light and optical technology, sound and acoustical technology, and engineering physics.

INSPEC, 1969-present, monthly updates.
BRS, LRS (12,13), SDC

This database, which covers all areas of pure and applied physics, is the machine-readable counterpart of three abstracting journals: *Physics Abstracts*, *Electrical and Electronics Abstracts* and *Computer and Control Abstracts*. Journal articles and, to a lesser extent, government reports, patents and monographs are included to provide information in such areas as atomic and molecular physics, and elementary particle physics.

METEOROLOGICAL AND GEOASTROPHYSICAL ABSTRACTS, 1972-present, irregular updates.
LRS (29)

The MGA file, which corresponds to the printed title, provides current citations in English for the most important meteorological and geoastrophysical research published in both foreign and domestic literature. Subjects covered include astrophysics, physical oceanography and glaciology.

SPIN, 1975-present, monthly updates.
LRS (62)

SPIN (Searchable Physics Information Notices) covers all major areas of physics, emphasizing American and Russian physics research. Subjects covered include elementary particles, atomic and molecular physics, fluids, plasmas, materials science, physical chemistry, and such related areas as geophysics, astrophysics, biophysics and biomedical engineering.

TECHNOLOGY

FOODS ADLIBRA, 1974-present, monthly updates.
LRS (79)

This database covers new developments in food technology and packaging, including information on the food industry, nutrition, research in processing methods and packaging, government guidelines and regulations, marketing and statistics. Some U.S. and British patents are also included.

FSTA, 1969-present, monthly updates.
LRS (51)

FSTA (Food Science and Technology Abstracts) covers research and new development literature in areas related to food science and technology, including agriculture, biochemistry, engineering, and home economics. It indexes journals, patents and books, as does the printed version.

FROST AND SULLIVAN DM^2, 1975-present, quarterly updates.
LRS (59)

This database provides information about U.S. Government contract awards, request-for-proposals, planning estimates, R & D sources sought and advanced planning procurement information for the engineered systems and services market. It covers fields such as aircraft, communications, data processing, missiles and space and basic research.

PAPERCHEM, 1969-present, monthly updates.
SDC

Abstract Bulletin of the Institute of Paper Chemistry is the printed counterpart of this file, which covers the scientific and technical literature that deals with the theoretical principles, technologies, raw materials, products and practices of the pulp, paper and board manufacturing and utilizing industries.

PIRA, 1975-present, monthly updates.
LRS (48)

The PIRA file is the online counterpart of four abstracting journals: *Paper and Board Abstracts, Printing Abstracts, Packaging Abstracts* and *Management and Marketing Abstracts*. It is useful for information in the field of paper and board mills and manufacturing, printing, photography, bookbinding, packaging and shipping materials, advertising, retailing and industrial relations.

RAPRA ABSTRACTS, 1972-present, monthly updates.
LRS (95)

Rubber and Plastics Research Association Abstracts covers the commercial, technical and research aspects of the rubber and plastics industries. It includes information on materials (including synthesis and polymerization), processing technology, applications of polymers, toxicity reports, economic and commercial information.

TITUS, 1967-present, monthly updates.
SDC

This database, which has no printed counterpart, covers the textile and allied industries including: processes and machines for fabric production, fibers, yarns, textile finishing including bleaching and dyeing, information about industrial fabrics including tire cord and carpets, analysis and testing of textile products and equipment, and textile engineering.

WORLD TEXTILES, 1970-present, monthly updates.
LRS (67)

This database is the machine-readable version of *World Textile Abstracts* and covers the world literature on the science and technology of textile and related materials; technical economics, production, and management of the textile industry; and on the consumption of and international trade in textile materials and products.

SUBJ COMP
EMCP

** ELECTRICAL-METHODS COMPUTER PROGRAM INVENTORY **
** SUMMARY REPORT by REF NO. **
** 01/24/77 **

REF NO.	PROGRAM NAME	PROGRAM TITLE(or PURPOSE)	NUMERICAL METHOD(s)	DOCUMENTATION AVAILABLE	SOURCE CODE	MACHINE USED	OPER. SYS.	AUTHOR or CONTRIBUTOR ADDRESS and PHONE NO.
001	EMFIN	EM fields about a finite electric wire source. (computes all E,H fields for layered models in freq. and time domains)	Num. integration by convolution or Gaussian quadrature	NTIS Report PB-238-199 (1974)	Fortran	DEC-10	TOPS-10	Walter L. Anderson U.S.Geological Survey Mail Stop 964 Box 25046, Den.Fed.Ctr. Denver, Colorado 80225 (303) 234-3139
002		Improved digital filters for evaluating Fourier and Hankel transform integrals.(subr.package)	Num. integration by convolution and lagged-convolution methods	NTIS Report PB-242-800 (1975)	Fortran	DEC-10	TOPS-10	Walter L. Anderson (address in REF 001)
003		An optimal method for evaluating a class of convolution integrals with related kernels.	Num. integr. subr. by convolution for integrals with related kernels	NTIS Report PB-251-156 (1976)	Fortran	DEC-10	TOPS-10	Walter L. Anderson (address in REF 001)
004	MARQHZ	Marquardt inversion of vertical magnetic field measurements from a grounded wire source. (inverts freq and/or dist Hz soundings from ele. dipole or bipole)	Modified Marquardt nonlinear least squares using num. integration by convolution and adaptive quadrature over finite interval	NTIS Report (in press)	Fortran	CYBER-74-28 (CDC-6600)	NOS-1.1	Walter L. Anderson (address in REF 001)
005	SCATPW	EM scattering by multiple conductors in earth due to a plane wave source. (2D rectangular models, constant conductivity contrast w/r half-space)	Integral eq. approx. and num. integration by convolution	NTIS Report PB-261-183 (1976)	Fortran	DEC-10	TOPS-10	Walter L. Anderson (address in REF 001)
006	SCATLN	EM scattering by multiple conductors in earth due to a line-source. (2D rectangular models, constant conductivity contrast w/r half-space)	Integral eq. approx. and num. integration by convolution	USGS internal documentation	Fortran	DEC-10	TOPS-10	Walter L. Anderson (address in REF 001)
007	EMTRAN1	Transient tangental ele. and vert.mag.fields about vert.mag.dipole for m-layer stratified earth.	Num. integration by Gaussian quadrature	NTIS Report PB-221-240 (1973)	Fortran	IBM-Sys/360 model 65	OS360 with HASP2	Walter L. Anderson (address in REF 001)

REF NO.	PROGRAM NAME	PROGRAM TITLE(or PURPOSE)	NUMERICAL METHOD(s)	DOCUMENTATION AVAILABLE	SOURCE CODE	MACHINE USED	OPER. SYS.	AUTHOR or CONTRIBUTOR ADDRESS and PHONE NO.
008	EMTRAN2	(same title as REF 007)	Num. integration by convolution and spline interpolation	NTIS Report (see REF 007)	Fortran	IBM-Sys/360 model 65	OS360 with HASP2	Walter L. Anderson (address in REF 001)
009		Calc. of Schlumberger sounding curves by convolution.	Num. integration by convolution	NTIS Report PB-232-056 (1974)	Fortran	IBM-Sys/360 model 65	OS360 with HASP2	Adel Zohdy U.S. Geological Survey Mail Stop 964 Box 25046, Den.Fed.Ctr. Denver,Colorado 80225 (303) 234-5465
010		Calc. of bipole-dipole apparent resistivity maps over horiz.strat. media.		NTIS Report PB-232-727 (1974)	Fortran	IBM-Sys/360 model 65	OS360 with HASP2	Adel Zohdy,W.D.Stanley (address in REF 009)
011		Automatic interpretation of Schlumberger sounding curves over horizontally stratified media.	Modified Dar Zarrouk and num. integration by convolution	NTIS Report PB-232-703 (1973)	Fortran	IBM-Sys/360 model 65	OS360 with HASP2	Adel Zohdy (address in REF 009)
012		Forward calc. & automatic inversion of Wenner sounding curves.	Modified Dar Zarrouk and num. integration by convolution	NTIS Report PB-247-265 (1975)	Fortran	DEC-10	TOPS-10	Adel Zohdy,R.J.Bisdorf (address in REF 009)
013	TOTAL	Finite element modeling of IP-resistivity 2-D earth.	Finite elements	"none yet"	Fortran	UNIVAC 1108	EXEC8	Luiz Rijo Dept.of Geol.& Geophysics University of Utah 709 Mineral Science Bldg. Salt Lake City,Utah 84112 (801) 581-5026
014	SCHLUM-BERGER	Finite element modeling of 2-D Schlumberger soundings.	Finite elements	"none yet"	Fortran	UNIVAC 1108	EXEC8	Luiz Rijo (address in REF 013)
015	LINE-SOURCE	Finite element modeling of 2-D earth in the presence of a line-source	Finite elements	"none yet"	Fortran	UNIVAC 1108	EXEC8	Luiz Rijo (address in REF 013)
016	TEPW	Finite element modeling of 2-D magneto-telluric problems.	Finite elements	"none yet"	Fortran	UNIVAC 1108	EXEC8	Luiz Rijo (address in REF 013)
017	RDINV	Inverts dipole-dipole resistivity data.	Linear leaast square inversion with Marquardt stabiliza-tion	"none yet"	Fortran	UNIVAC 1108	EXEC8	Bill Petrick Dept.of Geol.& Geophysics University of Utah 709 Mineral Science Bldg. Salt Lake City,Utah 84112 (801) 581-8713

REF NO.	PROGRAM NAME	PROGRAM TITLE(or PURPOSE)	NUMERICAL METHOD(s)	DOCUMENTATION AVAILABLE	SOURCE CODE	MACHINE USED	OPER. SYS.	AUTHOR or CONTRIBUTOR ADDRESS and PHONE NO.
018	RSINV	Inverts Schlumberger resistivity data.	Least squares inversion with Marquardt stabilization	"none yet"	Fortran	UNIVAC 1108	EXEC8	Bill Petrick (address in REF 017)
019	RSEMV	Simultaneously inverts finite source EM and Schlumberger resistivity data.	Least squares inversion with Marquardt stabilization	"none yet"	Fortran	UNIVAC 1108	EXEC8	Bill Petrick (address in REF 017)
020	RSEMV	Simultaneously inverts finite source EM and dipole-dipole resistivity data.	Least squares inversion with Marquardt stabilization	"none yet"	Fortran	UNIVAC 1108	EXEC8	Bill Petrick (address in REF 017)
021	MTINV	Inverts MT data.	Least squares with Marquardt method	"none yet"	Fortran	UNIVAC 1108	EXEC8	Bill Petrick (address in REF 017)
022	MTRES	Simultaneously inverts MT and Schlumberger resistivity data.	Least squares with Marquardt method	"none yet"	Fortran	UNIVAC 1108	EXEC8	Bill Petrick (address in REF 017)
023	ROCK3	Inversion for either resistivity or dielectric measurements.		"none yet"	Fortran	UNIVAC 1108	EXEC8	William Pelton Dept.of Geol.& Geophysics University of Utah 709 Mineral Science Bldg. Salt Lake City,Utah 84112 (801) 581-5026
024	HILBRT	Performs Hilbert transformation.		"none yet"	Fortran	UNIVAC 1108	EXEC8	Ralph Shuey Dept.of Geol.& Geophysics University of Utah 709 Mineral Science Bldg. Salt Lake City,Utah 84112 (801) 581-7750
025		Analysis of electrical measurements on rocks.		"none yet"	Fortran	UNIVAC 1108	EXEC8	Ralph Shuey,Wes Wilson (address in REF 024)
026		Fields about a point electrode on a layered earth.		"none yet"	Fortran	UNIVAC 1108	EXEC8	Don Pridmore Dept.of Geol.& Geophysics University of Utah 709 Mineral Science Bldg. Salt Lake City,Utah 84112 (801) 581-5026,582-0859

REF NO.	PROGRAM NAME	PROGRAM TITLE(or PURPOSE)	NUMERICAL METHOD(s)	DOCUMENTATION AVAILABLE	SOURCE CODE	MACHINE USED	OPER. SYS.	AUTHOR or CONTRIBUTOR ADDRESS and PHONE NO.
027		Magnetic fields from a point dipole in or on a 2-layered earth.	Num. integration by convolution	"fair"	Fortran	UNIVAC 1108	EXEC8	Don Pridmore (address in REF 026)
028		3-D resistivity algorithm	Finite elements	"fair"	Fortran	UNIVAC 1108	EXEC8 Ver31	Don Pridmore (address in REF 026)
029	BOSTICK INVERSE	Continuous magneto-telluric inverse. (conversion of MT appar. resistivity curve to resistivity versus depth)	(See Bostick, University of Texas)		Other	TI SR-52		M.G. Bloomquist c/o Mobil R & D P.O. Box 900 Dallas, Texas 75221 (214) 331-6531
030	MTMODEL	Magneto-telluric modeling (MT layered model appar. resistivity & continuous inverse)			Other	HP-9810 plus plotter		G.L. Hoehn c/o Mobil R & D P.O. Box 900 Dallas, Texas 75221 (214) 331-6531
031	WF-LLL2B	Sommerfeld package for antenna/ground interaction. (fields by Hertzian and finite-sized antennas in presence of ground)	Num. integration and approximate methods	NTIS Report UCRL-51821 (1975)	Fortran	CDC-7600 or CDC-6600		R.J. Lytle, D.L. Lager Lawrence Livermore Lab. P.O. Box 808 Livermore, Calif. 94550

Computers & Geosciences, Vol. 1, pp. 231–240. Pergamon Press, 1976. Printed in Great Britain

A FORTRAN IV PROGRAM FOR INTERPOLATING IRREGULARLY SPACED DATA USING THE DIFFERENCE EQUATIONS FOR MINIMUM CURVATURE

C. J. SWAIN

Department of Geology, The University, Leicester, U.K.

(*Received* 28 *February* 1975; *revised* 11 *July* 1975)

Abstract—The interpolation of potential field data is a common problem in geophysics. It is accomplished each time a contour map or profile is drawn. These operations can be difficult and subjective if the data points are irregularly spaced. The method described here produces a "numerical surface", or grid of values, that approaches the smoothest surface passing through the control points. Such a grid can be contoured automatically by standard routines, producing acceptable results. Alternatively, sections can be drawn through the surface on any azimuth—a more satisfactory procedure for constructing profiles than projecting the data points onto straight lines.

Key Words: FORTRAN, Interpolation, Contour maps, Geophysics.

INTRODUCTION

This program is based on a paper by Briggs (1974) using the difference equations derived from the requirement of minimum total curvature for the surface at the grid points. The equations are solved iteratively. The method uses the equations for data points both on and off the grid. They are described adequately by Briggs and therefore are not given here. The iterative method of solution requires that initial values be assigned to the grid points. Whereas in theory any values (e.g. zero) could be used, in practice the starting values should be as close as possible to the assymptotic values (i.e. the solution) in order to minimize the number of iterations required.

DEVELOPMENT

From the beginning, the method was determined to be highly successful for data that were not too irregularly spaced. In quantitative terms, if the ratio of the closest data spacing to the largest gaps is not more than *ca.* 5 or 10 to 1 then all the required detail could be shown in the solution in less than about 100 iterations, starting with values found by a weighted means method (Crain and Bhattacharyya, 1967). Starting with an overall mean may require many more iterations. Slightly faster convergence of the solution may be obtained by initially fitting weighted paraboloids at each grid point, but this can give extreme values in large gaps and at the edges where all the data are on one side of a grid point. In fact this method alone can be used successfully in interpolating fairly evenly spaced data (Falconer, 1971; La Porte, 1962). However, the problem to be solved was the interpolation of gravity data from Kenya for which the spacing is uneven, the ratio of largest to smallest distance between data points being greater than 100:1. Clearly a development of the basic methods was needed as the requirement of a close grid (of the order of the closest data spacing) for showing the detail conflicts with that of a large grid to enable a solution to be obtained in a reasonable number of iterations. The development was remarkably simple both in concept and application: it involves a progressive reduction of the grid spacing by factors of two, an iterative solution was obtained for each spacing. The number of points changes by a factor of four at each stage so only the final stage with the full number of grid points is significant as regards computer time.

The weighted means formula used here is the same as that used by Falconer (1971, p. 5). For a grid point at (A, B) only those data points within S of (A, B) are considered. A quadratic weighting function is used:

$$\text{WEIGHT}(I) = \left[\frac{S - D(I)}{D(I)}\right]^2 \quad \text{for } S > D(I),$$

where D is the distance of a data point from (A, B).

The value assigned to the grid point is:

$$\text{HEIGHT}(A, B) = \frac{\sum \text{WEIGHT}(I) \times H(X(I), Y(I))}{\sum \text{WEIGHT}(I)},$$

where $(X(I), Y(I))$ are the coordinates of the Ith data point and H is its value. The summations are over all points (I) for which $S > D(I)$.

Initially every Lth element of the $N \times M$ array $U(I, J)$ is given a value using weighted means ($L = 1, 2, 4, 8 \ldots$). Thus, $M - 1$ and $N - 1$ must be multiples of L. An iterative solution of the difference equations for this grid spacing is determined, then L is reduced to $L/2$ and the value $U(I, J)$ is assigned to the points $(I + L, J)$ $(I, J + L)$ and $(I + L, J + L)$. A new iterative solution for these grid points is determined with the values at (I, J) fixed by setting the appropriate elements of the reference array to -1. As one in four grid points is fixed, convergence proceeds rapidly at this stage. A new iterative solution based only on the control points is determined. This method guarantees that at every stage reasonable values will be assigned to each grid point and that therefore convergence will occur as rapidly as possible in the solution of the difference equations. The procedure is repeated until $L = 1$. Convergence is tested by retaining the largest change in $U(I, J)$ that occurs in one iteration. It

is halted when this goes below a certain value (Δ). In trial data sets from a gravity survey of Kenya, with a grid spacing of 0.02° (2.2 km) and a starting L of 4 or 8 convergence to a Δ of 0.05 mgal could be obtained usually within 100 iterations (Fig. 1).

PROGRAM DESCRIPTION

The program consists of a short main routine and 5 subroutines (Appendix). The main program first reads the data, calculates N, M and various control parameters, then calls SORT followed by the main subroutine MINC. SORT rearranges the station order to give faster access to the data by HEIGHT, which is the function used to give starting values to the points on the preliminary grid. SORT and HEIGHT are both direct conversions of ALGOL procedures given by Falconer (1971), the latter modified to enable weighted mean interpolation to be accomplished. Together they constitute a rapid means of obtaining initial interpolated values.

The next section of MINC determines the nearest grid point (subscripts I, J) to each data point (subscript K). The value K is assigned to $IU(I, J)$. If there is more than one data point within a grid square centered on a particular grid point, the closest is selected; if none, $IU(I, J) = 0$. In earlier versions of the program, the weights $B1$–$B5$ (used in the "offgrid-point" equations) were calculated and stored in five-arrays, but due to lack of core storage this was modified and the weights are recalculated at each iteration. This is at the expense of time but not seriously as only perhaps one in ten grid points is close to a data point. The other information (besides K) used in the difference equations is stored in IU, namely an integer 1–4 (×10,000) determining how the weights $B1$–$B5$ are to be calculated. If XI and ETA (the X and Y distances that a data point lies off the grid) are less than SMALL (an arbitrary fraction of the grid spacing, e.g. 1/10) the data point is taken to coincide with the grid point, which has the value of the data point assigned to it and this is not recalculated at each iteration. This applies also if the data point lies in the outer two rows or columns as these data points cannot be used with the offgrid-point equations. In these situations $IU(I, J) = -K$: again the closest data point is selected.

In subroutine ITERAT a new value $U(I, J)$ is calculated for each grid point in turn, starting with (1, 1), from the surrounding values. There are seven different equations, depending on where the grid point lies and whether it is close to a data point (see Briggs, 1974). The values are calculated row by row. After the first two iterations, DUIJ, the largest change in U in one complete pass, is printed

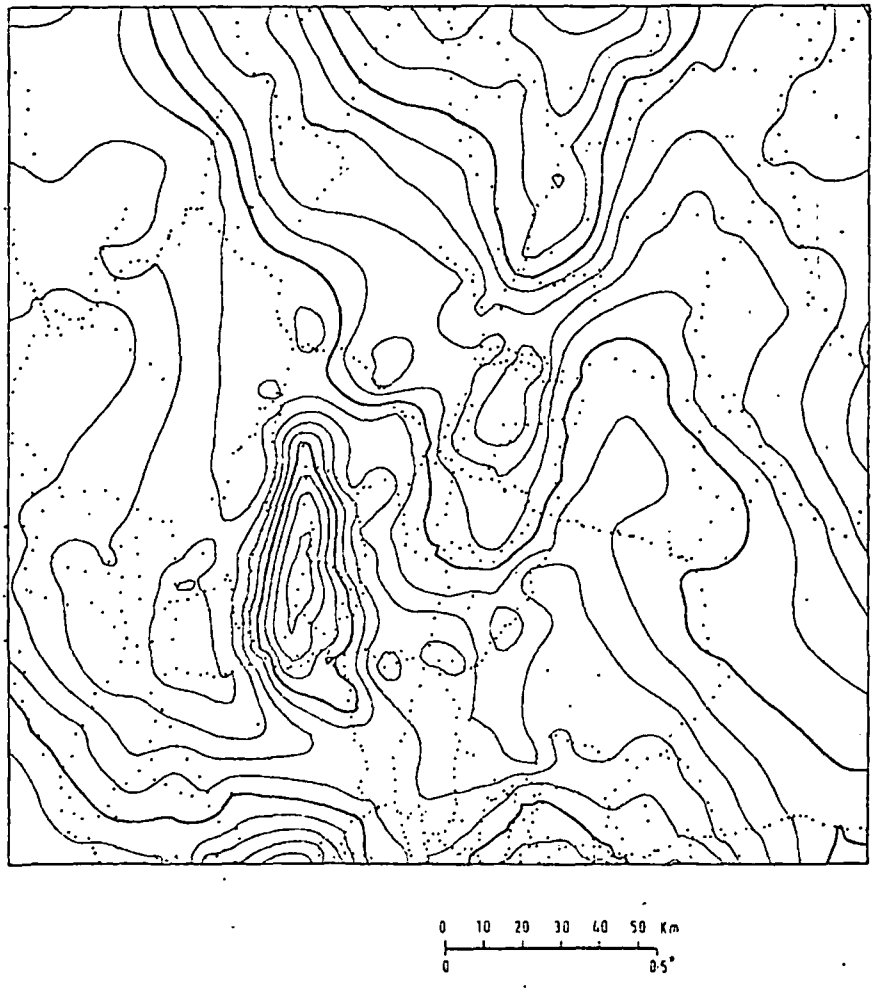


Figure 1. Gravity data from Rift Valley (Kenya) contoured on 10 mgal interval. Grid spacing 2.2 km [0.02°]. Result after 40 iterations [Δ = 0.05 mgal].

after every iteration, together with the slope B of the graph of Log_{10}(DUIJ) against Log_{10}(ITER) for the previous two iterations, and also the required number of iterations required for DUIJ to become less than DELTA, predicted from the slope B. It seems that the largest change (DUIJ) tends to occur in one particular area (or anomaly) within the array for a number of iterations during which B usually decreases implying that the values are converging to a solution. B may increase sharply after a certain number of iterations, if DUIJ moves to another area, but then should start to decrease again. This sequence may be repeated several times before DUIJ becomes less than DELTA. Values for DELTA have been selected empirically; it is not critical except in regard to computer time.

At the end of iteration, control is returned to MINC, L is replaced by $L/2$, the values at $(I+L, J)$, $(I, J+L)$ and $(I+L, J+L)$ are assigned before returning to the point at which the data points are selected so that the whole process is repeated. If $L = 1$ at the end of iteration, the array U is printed by PRINTX in integer form (to save space) and written to the local file TAPE 3, so that (if the file is saved) it can be contoured later.

RESULTS

The map shown in Figure 1 took 40 iterations (at $L = 1$) for a DELTA of 0.05 mgal. With DELTA reduced to 0.02 mgal., 115 iterations were required. The only significant differences between the contour maps produced by these two runs occur at edges and in areas of poor control, namely the northwest and southeast corners.

Contouring

Figure 1 was produced by a simple standard contouring routine (part of the GHOST package (Prior, 1973) which draws straight-line segments and employs linear interpolation. In this example a closely spaced grid was required to show all the detail and the results are acceptable. A more sophisticated approach would be to use a contour following routine (e.g. procedure "followcontour" of Falconer, 1971) using the offgrid-point equations to calculate the value at any point. It is intended to try this approach in the future. The method should overcome the problem, mentioned by Briggs, of contour lines not precisely honoring control points.

Language

The program is written in ANSI FORTRAN IV with the following nonstandard features:

(1) The PROGRAM statement (in main program).
(2) Mixed arithmetic is used extensively.
(3) In Format statements " , " is omitted after " / ".
(4) Nonstandard subscripts (e.g. $U(I-L1, 1)$) are used on lines 40, 61, 88, 116, 143 and 164 of subroutine ITERAT.

These features are supported by most FORTRAN compilers.

Time

The array U which is contoured in Figure 1 has dimensions 101 × 101. 1029 control points were used and the time taken by the interpolation program was 25.7 min on an ICL 4130. This is a rather old machine, said to be about one hundredth the speed of modern computers.

CONCLUSIONS

This is the most satisfactory method that the author has investigated in interpolating the type of data commonly produced by reconnaissance surveys—namely detailed, widely spaced traverses. Results are considered superior to those of standard packages such as Calcomp's GPCP because the surface produced has the minimum curvature property.

Acknowledgment—This work has been supported by the Natural Environment Research Council grant No. GR3/1486. I should like to thank I. C. Briggs for helpful correspondence.

REFERENCES

Briggs, I. C., 1974, Machine contouring using minimum curvature: Geophysics, v. 39, no. 1, p. 39–48.

Crain, I. K., and Bhattacharyya, B. K., 1967, Treatment of non-equispaced two-dimensional data with digital computer: Geoexploration, v. 5, no. 4, p. 173–194.

Falconer, K. J., 1971, A general purpose algorithm for contouring over scattered data points: National Physical Lab., Div. Numerical Analysis and Computing Rept NAC6 (September 1971), 44 p.

La Porte, M., 1962, Elaboration rapide de cartes gravimetriques deduites de l'Anomalie de Bouguer à l'aide d'une calculatrice electronique: Geophysical Prospecting, v. 10, no. 3, p. 238–259.

Prior, W. A. J., 1973, The GHOST graphical output system users manual: UKAEA Research Group (Culham), CLM-PDN 8/71, 104 p.

APPENDIX

LISTING OF MINCURV, SUBROUTINES PRINTX, SORT, MINC, ITERAT AND FUNCTION HEIGHT

```
      PROGRAM MINCURV(INPUT,OUTPUT,TAPE7=INPUT,TAPE2=OUTPUT)
C
C-------INTERPOLATION BY THE METHOD OF BRIGGS (GEOPHYSICS,APRIL,1974).
C
C-------PROGRAMMED BY C J SWAIN ,LEICESTER UNIVERSITY.
C
      DIMENSION U(49,89),IU(49,89),MAP(49)
      DIMENSION X(962),Y(962),H(962),DATA(3,962)
      DIMENSION MARK(9,9)
      COMMON /BLK1/XMIN,YMIN,PN,PM
      COMMON /BLK2/XMAX,YMAX,NLX,NLY,LX,LY,IMAX,SOS,S,SS,EPS
      COMMON /BLK3/L
      EQUIVALENCE (U(1,1),DATA(1,1))
      INTEGER SOS
      REAL LX,LY
      IERROR=0
C
C-------S=SEARCH RADIUS
```

```
C--------DELTA=LIMIT FOR DUIJ  AT WHICH ITERATION STOPS
C--------L*XINC,L*YINC = STARTING GRID SIZE
C--------MARK HAS DIMENSIONS NLX,NLY
C--------XMAX,YMAX ARE ADJUSTED DOWN SO THAT N-1,M-1 ARE MULTIPLES OF L
C--------NU,MU = DIMENSIONS OF U,IU
C--------NC=DIMENSION OF X,Y,H,DATA
C
      READ(7,1001)S,L,DELTA
      WRITE(2,2001)S,L,DELTA
      READ (7,1004)XMIN,XMAX,YMIN,YMAX,XINC,YINC
      READ(7,1002)NC,NU,MU
      L1=L
      EPS=(XMAX-XMIN)*0.000001
      SQS=4
      NLX=INT((XMAX-XMIN)*SQS/(2.*S)+.0001)
      NLY=INT((YMAX-YMIN)*SQS/(2.*S)+.0001)
      IF(NLX.EQ.0)NLX=1
      IF(NLY.EQ.0)NLY=1
      LX=(XMAX-XMIN)/NLX
      LY=(YMAX-YMIN)/NLY
      SS=S*S
      N=INT((XMAX-XMIN)/XINC+.001)+1
      N=((N-1)/L)*L+1
      XMAX=XMIN+(N-1)*XINC
      M=INT((YMAX-YMIN)/YINC+.001)+1
      M=((M-1)/L)*L+1
      YMAX=YMIN+(M-1)*YINC
      WRITE(2,2004)XMIN,XINC,XMAX,N,YMIN,YINC,YMAX,M
C
      READ(5)IMAX
      DO94J=1,IMAX
   94 READ(5)(DATA(I,J),I=1,3)
      WRITE(2,95)IMAX
   95 FORMAT(1X/19HNO. OF DATA POINTS=,I4/)
C
      CALL SORT(MARK,NLX,NLY,X,Y,H,DATA,NC)
      CALL MINC(U,IU,NU,MU,N,M,X,Y,H,NC,MARK,NLX,NLY,MAP,XINC,YINC,
     1   DELTA,IERROR)
      IF(IERROR.EQ.1)STOP
      CALL PRINTX(U,NU,MU,N,M,0,MAP)
      WRITE(3)N,M
      WRITE(3)((U(I,J),I=1,N),J=1,M)
      STOP
 1001 FORMAT(F5.0,I5,F5.0)
 1002 FORMAT(5I5)
 1004 FORMAT(9F5.2)
 2001 FORMAT(1X/5X,7HRADIUS=,F5.2,7HDEGREES,5X,11HSTARTING L=,I2,5X,
     1   6HDELTA=,F5.3/)
 2004 FORMAT(5X,10HLONGITUDE=,F5.2,3H BY, F5.2,3H TO, F6.2,8H DEGREES,
     1   1H(,I3,7HPOINTS)/
     2         5X,9HLATITUDE=,F5.2,3H BY, F5.2,3H TO, F6.2,8H DEGREES,
     1   1H(,I3,7HPOINTS)/)
      END
      SUBROUTINE PRINTX(A,NU,MU,N,M,NSPACE,MAP)
C
C--------PRINTS ARRAY A IN INTEGER FORM
C--------NSPACE=0 FOR CLOSE SPACING
C--------NSPACE=1 FOR SQUARE GRID WITH HALF INCH SPACING FOR HAND CONTOUR
C
      DIMENSION A(NU,MU),MAP(NU)
      COMMON /BLK3/L
      IN=25*L
      IF(NSPACE)3,3,4
    3 IN=30*L
    4 DO100IB=1,N,IN
      IE=IB+IN-1
      IF(IE-N)2,2,1
    1 IE=N
    2 IF(NSPACE)5,5,6
    5 WRITE(2,2000)(I,I=IB,IE,L)
      GOTO7
    6 WRITE(2,2010)(I,I=IB,IE,L)
    7 DO101LL=1,M,L
      J=M-LL+1
      DO102I=IB,IE,L
  102 MAP(I)=INT(A(I,J)+.5)
      IF(NSPACE)8,8,9
    8 WRITE(2,2001)J,(MAP(I),I=IB,IE,L)
      GOTO101
    9 WRITE(2,2011)J,(MAP(I),I=IB,IE,L)
  101 CONTINUE
  100 CONTINUE
      RETURN
 2000 FORMAT(1X////5X,30I4/)
 2001 FORMAT(1X,I3,1X,30I4)
 2010 FORMAT(1H1,4X,25I5/)
 2011 FORMAT(1H //I3,2X,25I5)
      END
      SUBROUTINE SORT(MARK,NA,NB,X,Y,H,DATA,NC)
C-------SORTS DATA INTO RECTANGLES AND STORES IN X,Y,H, TO ALLOW FASTER
C       ACCESS TO DATA BY FUNCTION HEIGHT.  THE FIRST POINT IN RECTANGLE
C       (I,J) HAS SUBSCRIPT MARK(I,J).
```

```
      DIMENSION MARK(NA,NB),X(NC),Y(NC),H(NC),DATA(3,NC)
      COMMON /BLK1/XMIN,YMIN,PN,PM
      COMMON /BLK2/XMAX,YMAX,NLX,NLY,LX,LY,N,SQS,S,SS,EPS
      REAL LX,LY
      INTEGER SQS
      INTEGER R
      R=1
      M=1
      DO6I=1,NLX
      DO2K=1,N
      IF(INT((DATA(1,K)-XMIN)/LX).EQ.I-1)GOTO1
      IF(I.EQ.1.AND.DATA(1,K).LE.XMIN)GOTO1
      IF(I.EQ.NLX.AND.DATA(1,K).GE.XMAX)GOTO1
      GOTO2
    1 X(M)=DATA(1,K)
      Y(M)=DATA(2,K)
      H(M)=DATA(3,K)
      M=M+1
    2 CONTINUE
      DO5J=1,NLY
      MARK(I,J)=R
      M1=M-1
      DO4L=R,M1
      IF(INT((Y(L)-YMIN)/LY).EQ.J-1)GOTO3
      IF(J.EQ.1.AND.Y(L).LE.YMIN)GOTO3
      IF(J.EQ.NLY.AND.Y(L).GE.YMAX)GOTO3
      GOTO4
    3 TEMP=X(L)
      X(L)=X(R)
      X(R)=TEMP
      TEMP=Y(L)
      Y(L)=Y(R)
      Y(R)=TEMP
      TEMP=H(L)
      H(L)=H(R)
      H(R)=TEMP
      R=R+1
    4 CONTINUE
    5 CONTINUE
    6 CONTINUE
      RETURN
      END
      FUNCTION HEIGHT(A,B,MARK,NA,NB,X,Y,H,NC,IERROR)
C
C-------COMPUTES VALUE AT A,B AS WEIGHTED MEAN OF DATA INSIDE RADIUS S
C
      DIMENSION MARK(NA,NB),X(NC),Y(NC),H(NC)
      COMMON /BLK1/XMIN,YMIN,PN,PM
      COMMON /BLK2/XMAX,YMAX,NLX,NLY,LX,LY,N,SQS,S,SS,EPS
      REAL LX,LY
      INTEGER SQS
      SH=0.
      SW=0.
      NPTS=0
      IR=INT((A-S-XMIN)/LX)+1
      JR=INT((B-S-YMIN)/LY)+1
      IS=IR+SQS
      JS=JR+SQS
      IF(IR.LT.1)IR=1
      IF(IR.GT.NLX)IR=NLX
      IF(JR.LT.1)JR=1
      IF(JR.GT.NLY)JR=NLY
      IF(IS.LT.1)IS=1
      IF(IS.GT.NLX)IS=NLX
       IF(JS.LT.1)JS=1
      IF(JS.GT.NLY)JS=NLY
      DO4I=IR,IS
      LAST=MARK(I,JS+1)-1
      IF(JS.GE.NLY)LAST=MARK(I+1,1)-1
      IF(JS.GE.NLY.AND.I.GE.NLX)LAST=N
      KM=MARK(I,JR)
      DO3K=KM,LAST
      U=X(K)-A
      V=Y(K)-B
      DD=U*U+V*V
      IF(DD.GE.SS)GOTO3
      IF(DD.GE.EPS)GOTO2
      HEIGHT=H(K)
      RETURN
```

```
      2 D=SQRT(DD)
        WEIGHT=(S-D)*(S-D)/DD
        SH=SH+WEIGHT*H(K)
        SW=SW+WEIGHT
        NPTS=NPTS+1
      3 CONTINUE
      4 CONTINUE
        IF(NPTS)5,5,12
     12 HEIGHT=SH/SW
        RETURN
      5 WRITE(2,6)
      6 FORMAT(1X,23HSEARCH RADIUS TOO SMALL)
        HEIGHT=0.
        IERROR=1
        RETURN
        END
        SUBROUTINE MINC(U,IU,NU,MU,N,M,X,Y,H,NC,MARK,NA,NB,MAP,XINC,YINC,
       1   DELTA,IERROR)
        DIMENSION U(NU,MU),IU(NU,MU),MAP(NU)
        DIMENSION X(NC),Y(NC),H(NC)
        DIMENSION MARK(NA,NB)
        COMMON /BLK1/XMIN,YMIN,PN,PM
        COMMON /BLK2/XMAX,YMAX,NLX,NLY,LX,LY,IMAX,SQS,S,SS,EPS
        COMMON /BLK3/L
        REAL LX,LY
        INTEGER SQS
        SMALL=0.05
C
C       SET INITIAL U VALUES WITH GRID SIZE XINC*L BY YINC*L
C
        B=YMIN
        DO81J=1,M,L
        A=XMIN
        DO80I=1,N,L
        U(I,J)=HEIGHT(A,B,MARK,NA,NB,X,Y,H,NC,IERROR)
        IF(IERROR.EQ.1)GOTO65
     80 A=A+XINC*L
     81 B=B+YINC*L
        CALL PRINTX(U,NU,MU,N,M,0,MAP)
C
C-------MODE=0 WHILE FITTING SURFACE TO DATA POINTS
C-------MODE=1 WHILE FITTING SURFACE TO ALTERNATE GRID POINTS AFTER
C           HALVING GRID SPACING
C
     64 MODE=0
        DO106J=1,M,L
        DO106I=1,N,L
    106 IU(I,J)=0
        PN=1./(XINC*FLOAT(L))
        PM=1./(YINC*FLOAT(L))
C
C  SELECT NEAREST GRID SQUARE,CENTRED ON (I,J),CONTAINING EACH DATA PT
C
        DO77K=1,IMAX
        XI=PN*(X(K)-XMIN)+1.
        ETA=PM*(Y(K)-YMIN)+1.
        I=INT(XI+.5)
        J=INT(ETA+.5)
        XI=XI-I
        ETA=ETA-J
        I=L*(I-1)+1
        J=L*(J-1)+1
        IF(I.LT.1.OR.I.GT.N.OR.J.LT.1.OR.J.GT.M)GOTO77
        IF(I.LE.L+1.OR.I.GE.N-L)GOTO75
        IF(J.LE.L+1.OR.J.GE.M-L)GOTO75
        IF(ABS(XI).LT.SMALL.AND.ABS(ETA).LT.SMALL)GOTO76
        IF(IU(I,J))77,102,101
C
C       IF MORE THAN ONE DATA POINT WITHIN GRID SQUARE THEN SELECT CLOSEST
C
    101 KK=IU(I,J)
        KUK=KK/10000
        KK=KK-KUK*10000
        DIST2=(PN*(X(KK)-XMIN)+1.-I)**2+(PM*(Y(KK)-YMIN)+1.-J)**2
        DIST1=XI*XI+ETA*ETA
        IF(DIST1-DIST2)102,77,77
    102 CONTINUE
```

```
C
C     STATION NUMBER  INTO REFERENCE ARRAY PACKED WITH INTEGER 1 TO 4
C       SPECIFYING WHICH QUADRANT POINT IS IN
C
      IU(I,J)=K
      IF(XI.GT.0..AND.ETA.GT.0.)IU(I,J)=10000+IU(I,J)
      IF(XI.LT.0..AND.ETA.GT.0.)IU(I,J)=20000+IU(I,J)
      IF(XI.LT.0..AND.ETA.LT.0.)IU(I,J)=30000+IU(I,J)
      IF(XI.GT.0..AND.ETA.LT.0.)IU(I,J)=40000+IU(I,J)
      GOTO77
   75 KK=-IU(I,J)
      IF(KK)77,76,74
   74 DIST2=(PN*(X(KK)-XMIN)+1.-I)**2+(PM*(Y(KK)-YMIN)+1.-J)**2
      DIST1=XI*XI+ETA*ETA
      IF(DIST1-DIST2)76,77,77
   76 IU(I,J)=-K
      U(I,J)=H(K)
   77 CONTINUE
   78 CALL ITERAT(U,IU,NU,MU,N,M,DELTA,X,Y,H,NC)
      IF(MODE.EQ.1)GOTO64
      IF(L.EQ.1)GOTO65
      N1=N-L
      M1=M-L
      L1=L
C
C     HALVE GRID SPACING AND ASSIGN NEW U VALUES
C
      L=L/2
      DO111J=1,M1,L1
      JL=J+L
      DO110I=1,N1,L1
      IL=I+L
      UIJ=U(I,J)
      U(IL,J)=UIJ
      U(I,JL)=UIJ
      U(IL,JL)=UIJ
      IU(I,J)=-1
      IU(IL,J)=0
      IU(I,JL)=0
      IU(IL,JL)=0
  110 CONTINUE
      U(N,JL)=U(N,J)
      IU(N,J)=-1
      IU(N,JL)=0
  111 CONTINUE
      DO112I=1,N1,L1
      IL=I+L
      U(IL,M)=U(I,M)
      IU(I,M)=-1
      IU(IL,M)=0
  112 CONTINUE
      IU(N,M)=-1
      MODE=1
      GOTO78
   65 RETURN
      END
      SUBROUTINE ITERAT(U,IU,NU,MU,N,M,DELTA,X,Y,H,NB)
C
C---APPLIES FINITE DIFFERENCE EQUATIONS FOR MINIMUM CURVATURE TO ARRAY U
C      CONTROLLED BY STATIONS VALUES X(K),Y(K),H(K). STATION NUMBERS K
C      ARE CONTAINED IN REFERENCE ARRAY IU TOGETHER WITH AN INTEGER 1 TO
C      4 WHICH DETERMINES HOW THE OFF-GRID POINT EQUATIONS ARE APPLIED
C
      DIMENSION U(NU,MU),IU(NU,MU),X(NB),Y(NB),H(NB)
      COMMON /BLK1/XMIN,YMIN,PN,PM
      COMMON /BLK3/L
      L1=2*L
      L2=L+1
      L3=2*L+1
      L4=3*L+1
      N1=N-L
      N2=N-2*L
      N3=N-3*L
      M1=M-L
      M2=M-2*L
      M3=M-3*L
      ITER=0
```

```
      TER=0.0
      GOTO99
   98 WRITE(2,2005)
      DUIJ2=DUIJ
   99 CONTINUE
      DUIJ=0.
C
C              FIRST ROW         J=1
C
      IF(IU(1,1))2,1,2
    1 U(1,1)=(-.5*(U(1,L3)+U(L3,1))+U(1,L2)+U(L2,1))
    2 IF(IU(L2,1))4,3,4
    3 U(L2,1)=(-U(L2,L3)-U(L3,L2)-U(1,L2)-U(L4,1)+2.*U(1,1)+4.*(U(L3,1)+
     1  U(L2,L2)))/6.
    4 DO6I=L3,N2,L
      I1=I+L
      I2=I-L
      IF(IU(I,1))6,5,6
    5 U(I,1)=(-U(I-L1,1)-U(I+L1,1)-U(I,L3)-U(I1,L2)-U(I2,L2)+4.*(U(I2,1)
     1  +U(I,L2)+U(I1,1)))/7.
    6 CONTINUE
      IF(IU(N1,1))8,7,8
    7 U(N1,1)=(-U(N1,L3)-U(N2,L2)-U(N,L2)-U(N3,1)+2.*U(N,1)+4.*(U(N2,1)+
     1  U(N1,L2)))/6.
    8 IF(IU(N,1))10,9,10
    9 U(N,1)=(-.5*(U(N,L3)+U(N2,1))+U(N,L2)+U(N1,1))
C
C        SECOND ROW           J=2
C
   10 IF(IU(1,L2))12,11,12
   11 U(1,L2)=(-U(L3,L2)-U(L2,L3)-U(L2,1)-U(1,L4)+2.*U(1,1)+4.*(U(1,L3)+
     1  U(L2,L2)))/6.
   12 IF(IU(L2,L2))14,13,14
   13 U(L2,L2)=(-U(L2,L4)-U(L4,L2)-U(1,L3)-U(L3,1)-2.*U(L3,L3)+8.*(U(L2,
     1    L3)+U(L3,L2))+4.*(U(L2,1)+U(1,L2)))/18.
   14 DO16I=L3,N2,L
      I1=I+L
      I2=I-L
      IF(IU(I,L2))16,15,16
   15 U(I,L2)=(-U(I-L1,L2)-U(I+L1,L2)-U(I,L4)-2.*(U(I2,L3)+U(I1,L3))-
     1  U(I2,1)-U(I1,1)+8.*(U(I2,L2)+U(I,L3)+U(I1,L2))+4.*U(I,1))/19.
   16 CONTINUE
      IF(IU(N1,L2))18,17,18
   17 U(N1,L2)=(-U(N1,L4)-U(N3,L2)-U(N,L3)-U(N2,1)-2.*U(N2,L3)+8.*(U(N1,
     1    L3)+U(N2,L2))+4.*(U(N1,1)+U(N,L2)))/18.
   18 IF(IU(N,L2))20,19,20
   19 U(N,L2)=(-U(N2,L2)-U(N1,L3)-U(N1,1)-U(N,L4)+2.*U(N,1)+4.*(U(N,L3)+
     1  U(N1,L2)))/6.
C
C        THIRD TO (M-2)TH ROW            J=3 TO J=M-2
C
   20 DO38J=L3,M2,L
      J1=J+L
      J2=J-L
      J3=J+L1
      J4=J-L1
      IF(IU(1,J))22,21,22
   21 U(1,J)=(-U(1,J4)-U(1,J3)-U(L3,J)-U(L2,J2)-U(L2,J1)+4.*(U(1,J2)+
     1  U(L2,J)+U(1,J1)))/7.
   22 IF(IU(L2,J))24,23,24
   23 U(L2,J)=(-U(L2,J4)-U(L2,J3)-U(L4,J)-2.*(U(L3,J2)+U(L3,J1))
     1  -U(1,J2)-U(1,J1)+8.*(U(L2,J2)+U(L3,J)+U(L2,J1))+4.*U(1,J))/19.
   24 DO34I=L3,N2,L
      I1=I+L
      I2=I-L
      IF(IU(I,J))34,25,26
   25 UIJ=.05*(-U(I+L1,J)-U(I,J3)-U(I-L1,J)-U(I,J4)-2.*(U(I1,J1)+
     1 U(I2,J1)+U(I1,J2)+U(I2,J2))+8.*(U(I1,J)+U(I2,J)+U(I,J1)+U(I,J2)))
      GOTO33
C
C     IF DATA POINT IS OFF GRID CALCULATE B COEFFICIENTS
C
   26 K=IU(I,J)
      KUK=K/10000
      K=K-KUK*10000
      XI=PN*(X(K)-XMIN)+1.
      ETA=PM*(Y(K)-YMIN)+1.
      XI=ABS(XI-AINT(XI+.5))
```

```
      ETA=ABS(ETA-AINT(ETA+.5))
      XY1=2./(XI+ETA+1.)
      B5=(2./(XI+ETA))*XY1
      B1=1.-XI*(1.+XI)*B5*.5
      B4=1.-ETA*(1.+ETA)*B5*.5
      B2=(1.+XI-ETA)*XY1
      B3=(1.+ETA-XI)*XY1
      IF(KUK-2)28,29,27
   27 IF(KUK-4)30,31,999
   28 BSUM=B1*U(I1,J2)+B2*U(I,J2)+B3*U(I2,J)+B4*U(I2,J1)
      GOTO32
   29 BSUM=B1*U(I2,J2)+B2*U(I,J2)+B3*U(I1,J)+B4*U(I1,J1)
      GOTO32
   30 BSUM=B1*U(I2,J1)+B2*U(I,J1)+B3*U(I1,J)+B4*U(I1,J2)
      GOTO32
   31 BSUM=B1*U(I1,J1)+B2*U(I,J1)+B3*U(I2,J)+B4*U(I2,J2)
   32 UIJ=(-U(I+L1,J)-U(I,J3)-U(I-L1,J)-U(I,J4)-2.*(U(I1,J1)+U(I2,J1)+
     1 U(I1,J2)+U(I2,J2))+4.*(U(I1,J)+U(I2,J)+U(I,J1)+U(I,J2))+
     2 4.*(BSUM+B5*H(K)))/(4.*(1.+B1+B2+B3+B4+B5))
   33 DUIJ1=ABS(UIJ-U(I,J))
      IF(DUIJ1.GT.DUIJ)DUIJ=DUIJ1
      U(I,J)=UIJ
   34 CONTINUE
      IF(IU(N1,J))36,35,36
   35 U(N1,J)=(-U(N1,J4)-U(N1,J3)-U(N3,J)-2.*(U(N2,J2)+U(N2,J1))
     1-U(N,J2)-U(N,J1)+8.*(U(N1,J2)+U(N2,J)+U(N1,J1))+4.*U(N,J))/19.
   36 IF(IU(N,J))38,37,38
   37 U(N,J)=(-U(N,J4)-U(N,J3)-U(N2,J)-U(N1,J2)-U(N1,J1)+4.*(U(N,J2)+
     1  U(N1,J)+U(N,J1)))/7.
   38 CONTINUE
C
C            (M-1)TH ROW                    J=M-1
C
      IF(IU(1,M1))40,39,40
   39 U(1,M1)=(-U(L3,M1)-U(L2,M2)-U(L2,M)-U(1,M3)+2.*U(1,M)+4.*(U(1,M2)+
     1  U(L2,M1)))/6.
   40 IF(IU(L2,M1))42,41,42
   41 U(L2,M1)=(-U(L2,M3)-U(L4,M1)-U(1,M2)-U(L3,M)-2.*U(L3,M2)+
     1  8.*(U(L2,M2)+U(L3,M1))+4.*(U(L2,M)+U(1,M1)))/18.
   42 DO44I=L3,N2,L
      I1=I+L
      I2=I-L
      IF(IU(I,M1))44,43,44
   43 U(I,M1)=(-U(I-L1,M1)-U(I+L1,M1)-U(I,M3)-2.*(U(I2,M2)+U(I1,M2))-
     1  U(I2,M)-U(I1,M)+8.*(U(I2,M1)+U(I,M2)+U(I1,M1))+4.*U(I,M))/19.
   44 CONTINUE
      IF(IU(N1,M1))46,45,46
   45 U(N1,M1)=(-U(N1,M3)-U(N3,M1)-U(N,M2)-U(N2,M)-2.*U(N2,M2)+
     1  8.*(U(N1,M2)+U(N2,M1))+4.*(U(N1,M)+U(N,M1)))/18.
   46 IF(IU(N,M1))48,47,48
   47 U(N,M1)=(-U(N2,M1)-U(N1,M2)-U(N1,M)-U(N,M3)+2.*U(N,M)+4.*(U(N,M2)+
     1  U(N1,M1)))/6.
C
C              MTH ROW                  J=M
C
   48 IF(IU(1,M))50,49,50
   49 U(1,M)=(U(1,M1)+U(L2,M)-.5*(U(1,M2)+U(L3,M)))
   50 IF(IU(L2,M))52,51,52
   51 U(L2,M)=(-U(L2,M2)-U(L3,M1)-U(1,M1)-U(L4,M)+2.*U(1,M)+4.*(U(L3,M)+
     1  U(L2,M1)))/6.
   52 DO54I=L3,N2,L
      I1=I+L
      I2=I-L
      IF(IU(I,M))54,53,54
   53 U(I,M)=(-U(I-L1,M)-U(I+L1,M)-U(I,M2)-U(I2,M1)-U(I1,M1)+
     1  4.*(U(I2,M)+U(I,M1)+U(I1,M)))/7.
   54 CONTINUE
      IF(IU(N1,M))56,55,56
   55 U(N1,M)=(-U(N1,M2)-U(N2,M1)-U(N,M1)-U(N3,M)+2.*U(N,M)+4.*(U(N2,M)+
     1  U(N1,M1)))/6.
   56 IF(IU(N,M))58,57,58
   57 U(N,M)=(U(N,M1)+U(N1,M)-.5*(U(N,M2)+U(N2,M)))
   58 ITER=ITER+1
C
C              ONE COMPLETE ITERATION
C
      TER=TER+1.0
      IF(ITER.EQ.1)GOTO98
```

```
      SLOPE=ALOG(OUIJ2/OUIJ)/ALOG((ITER-1.)/ITER)
      B=ALOG(OUIJ)-SLOPE*ALOG(ITER)
      TEST=(ALOG(DELTA)-B)/SLOPE
      IF(TEST.LT.15.5)GOTO59
      WRITE(2,2002)ITER,OUIJ,SLOPE
      GOTO60
   59 ITEST=INT(EXP(TEST))
      WRITE(2,2001)ITER,OUIJ,SLOPE,ITEST
   60 OUIJ2=OUIJ
      IF(OUIJ.LT.DELTA.OR.ITER.GE.500)RETURN
      GOTO99
  999 WRITE(2,2012)
      RETURN
 2001 FORMAT(3X,I4,4X,2(F10.4,4X),4X,I8)
 2002 FORMAT(3X,I4,4X,2(F10.4,4X),5X,7HTOO BIG)
 2005 FORMAT(3X,6HNO. OF,5X,7HLARGEST,4X,14HRATE OF CHANGE,4X,13HPROJECT
     1ED NO./10HITERATIONS,4X,6HCHANGE,6X,9H(LOG-LOG),7X,13HOF ITERATION
     2S)
 2012 FORMAT(1X/5HERROR,11H KUK G.T. 4/)
      END
```

Computers & Geosciences, Vol. 5, pp. 289–300
Pergamon Press Ltd., 1979. Printed in Great Britain

A FORTRAN PROGRAM FOR PLOTTING MINERAL STABILITIES IN THE Fe–Cu–S–O SYSTEM IN TERMS OF LOG ($\Sigma SO_4/\Sigma H_2S$) OR LOG fO_2 vs pH OR T

EDWARD M. RIPLEY
Geology Department, Indiana University, Bloomington, IN 47401, U.S.A.

and

HIROSHI OHMOTO
Department of Geosciences, The Pennsylvania State University, University Park, PA 16802, U.S.A.

(*Received* 11 *November* 1978)

Abstract—A computer program for the generation of mineral-stability diagrams in terms of log ($\Sigma SO_4/\Sigma H_2S$) vs pH or T is presented. Simple modifications of the program to produce log fO_2 vs pH or T diagrams also are documented. Such diagrams are useful particularly in the geochemical interpretation of hydrothermal sulfide ore deposits. Plotting of other geochemical parameters such as log fS_2, mole fractions of aqueous sulfur species, sulfur isotopic compositions, and metal complex solubilities also is possible utilizing the mineral-stability diagram as a base. Both line-printer and digital-plotter methods are explained.

Key Words: FORTRAN, mineral-stability diagrams, Fe–Cu–S–O system, log ($\Sigma SO_4/\Sigma H_2S$), pH, T.

INTRODUCTION

Log fO_2 (or aO_2) vs pH diagrams, first computed by Barnes and Kullerud (1961) following methods outlined by Garrels and Naeser (1958) are valuable aids in the geochemical interpretation of many hydrothermal sulfide ore deposits. More recently, Ripley and Ohmotto (1977) have introduced log ($\Sigma SO_4/\Sigma H_2S$) (ratio of total oxidized to total reduced sulfur in an ore fluid) vs pH or T diagrams. Such diagrams are useful particularly where changes in the $\Sigma SO_4/\Sigma H_2S$ ratio of the fluid have influenced geochemical parameters such as sulfur isotopic composition. For both log fO_2 and log ($\Sigma SO_4/\Sigma H_2S$) vs pH or T diagrams, mineral stability fields are dependent on the molality of total aqueous sulfur ($m_{\Sigma S}$), cation molality (Ca, Mg, Na, K), ionic strength (I), and T or pH (depending on which variable is plotted as the abscissa). Because changes in these variables can influence strongly diagram topology, it maybe necessary to construct several diagrams in order to arrive at possible interpretations of collected data. The purpose of this paper is to present a FORTRAN program which generates mineral-stability fields in terms of log ($\Sigma SO_4/\Sigma H_2S$) or log fO_2 vs pH or T. In addition to mineral stability fields, parameters such as mole fractions of aqueous sulfur species, log fS_2, $\delta^{34}S$ values, and solubility data may be plotted on the diagrams.

METHOD OF DIAGRAM CONSTRUCTION

For both types of diagrams the basics of construction essentially are those described in Ohmoto (1972). Relevant equations for construction of stability fields of aqueous sulfur species and minerals in the Fe–Cu–S–O system are given in Table 1. The equilibrium constants for these reactions have been fit as functions of absolute temperature by the method of least squares (see Ripley and Ohmoto, 1977). Activity coefficients of ionic species were calculated initially between ionic strengths of 0.25 and 3.0 using Helgeson's (1967) Delta approximation (Ohmoto, 1978, unpublished data), and then fit as functions of T and ionic strength by a multiple regression program (PSU Computation Center, 1968). The concentrations of Ca, Mg, K, Na, and Ba are read into the program from a data card. The molality of total sulfur is specified in the main program.

Table 1. Reactions used in constructing log ($\Sigma SO_4/\Sigma H_2S$) or log fO_2 vs pH or T diagrams

(1)	$H_2S(aq) = H^+ + HS^-$
(2)	$HS^- = H^+ + S^{2-}$
(3)	$2H^+ + SO_4^{2-} = H_2S(aq) + 2O_2$
(4)	$HSO_4^- = H^+ + SO_4^{2-}$
(5)	$KSO_4^- = K^+ + SO_4^{2-}$
(6)	$NaSO_4^- = Na^+ + SO_4^{2-}$
(7)	$CaSO_4^0 = Ca^{2+} + SO_4^{2-}$
(8)	$MgSO_4^0 = Mg^{2+} + SO_4^{2-}$
(9)	$H_2S(aq) + \frac{1}{2}O_2 = H_2O(l) + \frac{1}{2}S_2$
(10)	$FeS_2 = FeS + \frac{1}{2}S_2$
(11)	$3FeS_2 + 2O_2 = Fe_3O_4 + 3S_2$
(12)	$2FeS_2 + \frac{3}{2}O_2 = Fe_2O_3 + 2S_2$
(13)	$3FeS + 2O_2 = Fe_3O_4 + \frac{3}{2}S_2$
(14)	$6Fe_2O_3 = 4Fe_3O_4 + O_2$
(15)	$3FeS_2 + Fe_3O_4 = 6FeS + 2O_2$
(16)	$4FeS_2 + Cu_xFeS_4 = 5CuFeS_2 + S_2$
(17)	$CaSO_4 = Ca^{2+} + SO_4^{2-}$
(18)	$BaSO_4 = Ba^{2+} + SO_4^{2-}$

The first step in the construction of the diagrams is to determine the stability fields of the aqueous sulfur species considered (here H_2S, HS^-, SO_4^{2-}, HSO_4^-, $CaSO_4^0$, $MgSO_4^0$, KSO_4^-, $NaSO_4^-$). Mole fractions of each aqueous sulfur species relative to the total sulfur content of the ore fluid may be computed from equations given in Ohmoto (1972), and presented as statements following

the fO_2 calculation in the main program (Appendix). It should be noted that mole fractions cannot be computed without numerical values of $\log fO_2$. Although this presents no problem for $\log fO_2$ vs pH or T diagrams, $\log fO_2$ must first be computed for $\log(\Sigma SO_4/\Sigma H_2S)$ diagrams. The mole ratio of total sulfate to total sulfide in an ore-forming fluid may be defined as:

$$R = \frac{\Sigma SO_4}{\Sigma H_2S} = \frac{m_{SO_4^{2-}} + m_{HSO_4^-} + m_{NaSO_4^-} + m_{KSO_4^-} + m_{CaSO_4^0} + m_{MgSO_4^0}}{m_{H_2S} + m_{HS^-}}$$

R may be specified in terms of pH and cation concentration by utilizing activity coefficients (γ) and equilibrium constants (K) for reactions 1 and 4 through 7 in Table 1:

$$R = \frac{m_{SO_4^{2-}} \cdot B}{m_{H_2S} \cdot A}$$

where

$$A = 1 + \frac{(K_1)(\gamma_{H_2S})}{(\gamma_{HS^-})(a_{H^+})}$$

and

$$B = 1 + \gamma_{SO_4^{2-}}\left(\frac{a_{H^+}}{(K_4)(\gamma_{HSO_4^-})} + \frac{(m_{K^+})(\gamma_{K^+})}{(K_5)(\gamma_{KSO_4^-})} + \frac{(m_{Na^+})(\gamma_{Na^+})}{(K_6)(\gamma_{NaSO_4^-})} + \frac{(m_{Ca^{2+}})(\gamma_{Ca^{2+}})}{(K_7)(\gamma_{CaSO_4^0})} + \frac{(m_{Mg^{2+}})(\gamma_{Mg^{2+}})}{(K_8)(\gamma_{MgSO_4^0})}\right).$$

The $m_{SO_4^{2-}}$ may be expressed as a function of m_{H_2S} and fO_2 by use of reaction 4, Table 1:

$$m_{SO_4^{2-}} = \frac{(\gamma_{H_2S})(m_{H_2S})(fO_2)^2}{(K_3)(a_{H^+})^2(\gamma_{SO_4^{2-}})}.$$

R may now be related to fO_2, pH, and cation concentration by the equation:

$$R = \frac{(fO_2)^2(\gamma_{H_2S})(B)}{(K_3)(a_{H^+})^2(\gamma_{SO_4^{2-}})(A)}.$$

Given R, pH, γ, and cation molality values, $\log fO_2$ may be calculated and used in computing mole fractions of aqueous sulfur species. For $\log fO_2$ or $\log R$ vs T diagrams pH is fixed, or varied with temperature according to an appropriate buffer reaction. In the partial program listed in the Appendix pH is varied according to neutrality.

With all appropriate variables defined, $\log fS_2$ values may be computed from:

$$\log_{10} fS_2 = 2[\log K_9 + \log X_{H_2S} + \log m_{\Sigma S} + \log \gamma_{H_2S} - \log a_{H_2O} + \log fO_2]$$

[see Ohmoto (1972) for a similar expression].

Mineral-stability fields now may be constructed as functions of $\log fO_2$ (see Table 1), $\log fO_2$ (magnetite–hematite boundary), or $X_{SO_4^{2-}}$ (barite–anhydrite fields).

PROGRAM DETAILS

Mole fractions, $\log fS_2$, and $\log fO_2$ (for $\log R$ diagrams) values are calculated at each pH or T and $\log fO_2$ or $\log R$ intersection, and thus define two-dimensional arrays. Mineral-stability fields are determined initially by searching these arrays, and locating the boundary between two given mineral species. Consider, the pyrite–pyrrhotite boundary. The equilibrium constant for equation (10) (Table 1) defines the fS_2 value at the boundary. At higher fS_2 values pyrite is stable, and at lower values pyrrhotite is stable. At each point of higher than boundary fS_2 values, a "1" is assigned to that location in the array, and indicates pyrite stability. A "0" is assigned for pyrrhotite stability. At this stage the pyrite–pyrrhotite array may be printed on a line printer (Fig. 1), and a curve or line drawn between the "0's" and "1's". Although the pyrite–pyrrhotite boundary is determined, it is possible that when other Fe–S–O phases (magnetite and hematite) are considered, some or all of this boundary may be metastable. It is possible to print "0–1" arrays for all mineral pairs (pyrite (py)–magnetite (mt), pyrite–hematite (hm), pyrrhotite (po)–magnetite, and hematite–magnetite), and by utilizing traced overlays determine specific mineral-stability fields. If the line printer is to be utilized it is perhaps more efficient to print only the boundary curve (defined by a particular symbol) for each mineral pair on one output sheet. The boundary between the "0's" and "1's" can be determined by a routine similar to that discussed later. The approach described in this paper is utilization of a digital plotter (Calcomp or Versatec at Indiana University) to produce a composite diagram with metastable boundaries eliminated.

Once the "0–1" arrays have been computed for all mineral pairs, the stability fields of each mineral may be determined by summing arrays. For example, if pyrite stable is labeled "1" in the array plots of py–po, py–mt and py–hm, then the stability field of pyrite will be where the sum of the arrays equals 3. The summation method accomplishes the same end as traced overlays. If a particular mineral is labeled "1" in one array and "0" in another, the subroutine CHANGE reverses the values for one array, and allows the normal summation to proceed. For ease in later stability boundary plotting, all array values are reconverted to "0's" and "1's". The pyrite field denoted by "3's" is switched back to "1's", and all other numbers in the array given the value "0" (Fig. 2).

It should be noted that only one array denotes the stability boundaries of anhydrite, barite, and pyrite + bornite going to chalcopyrite. This is because no other Ba, Ca, or Cu-bearing minerals are involved in the calculations.

The final step in the diagram construction is plotting via the digital plotter (Figs. 3 and 4). The subroutine MINPLOT searches for changes in each array, and plots a given symbol where a change from "0" to "1" or "1" to "0" occurs. The search begins at the top of each column and continues downward until a change is determined. When a change is detected the symbol is plotted one-half distance between the "0" and "1". The search then

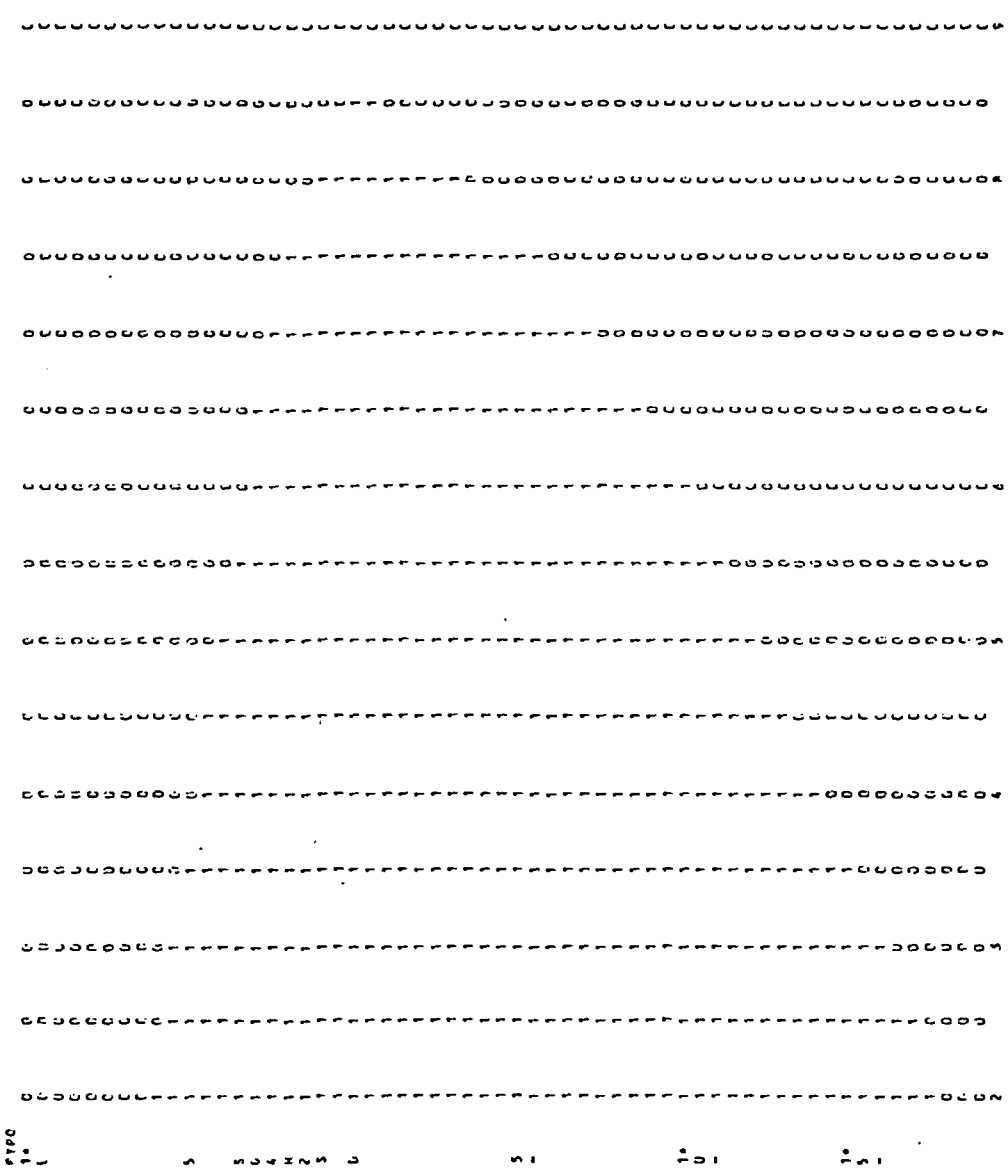


Figure 1. Stability field of pyrite (1) vs pyrrhotite in terms of $\log(\Sigma SO_4/\Sigma H_2S)$ vs pH. Line printer.

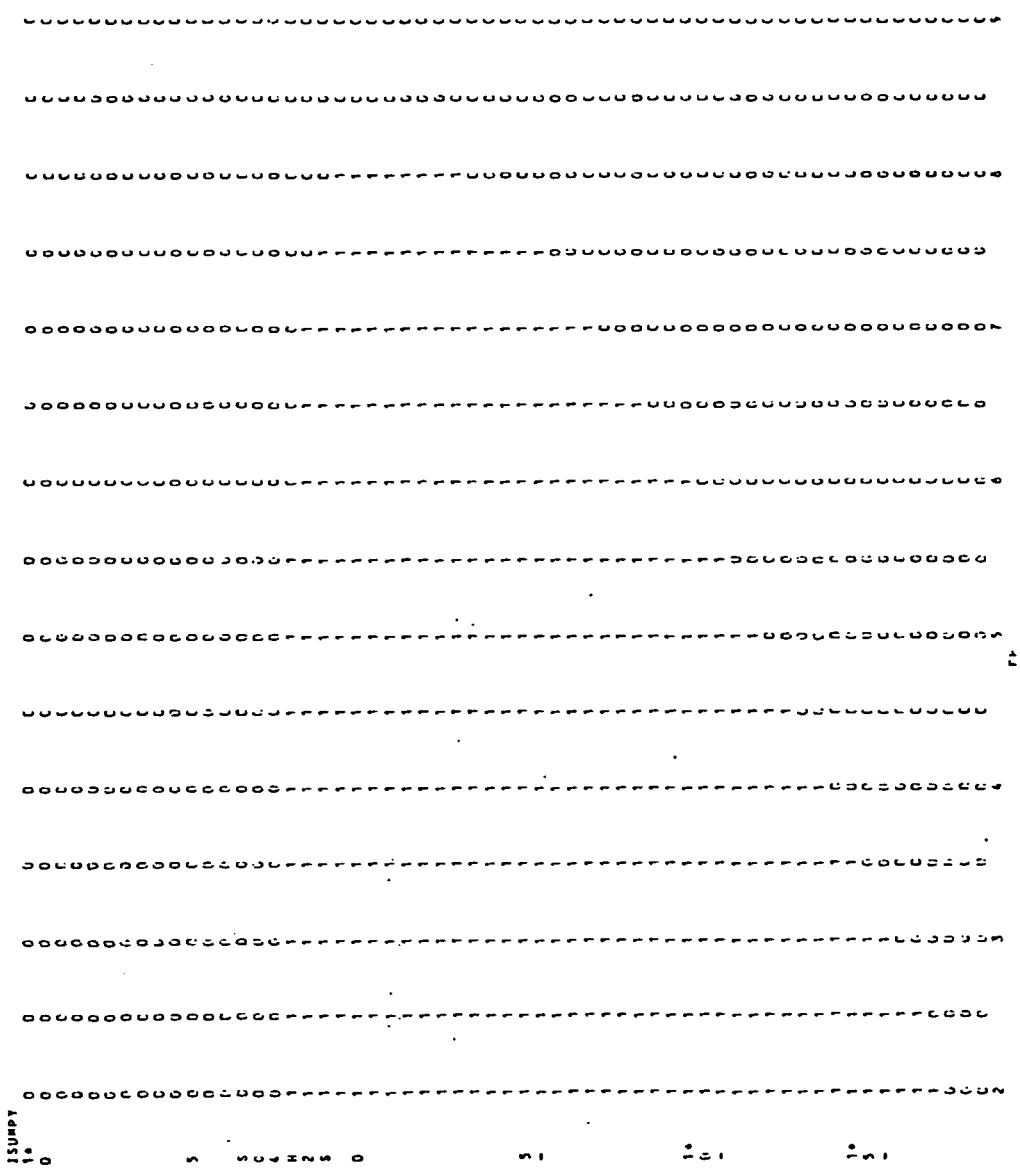


Figure 2. Stability field of pyrite (1) relative to pyrrhotite, magnetite, and hematite in terms of $\log(\Sigma SO_4/\Sigma H_2S)$ vs pH. Line printer.

begins from the bottom of the same column until a change occurs. This method outlines fields where at one given pH, or T, the stability boundary crosses two $\log fO_2$ or $\log R$ values (e.g. pyrite field). If there is only one $\log fO_2$ or $\log R$ values for a given pH or T, the assigned symbol is simply overprinted (e.g. hematite field).

Diagram axes and individual mineral-stability fields are labeled in accordance with Calcomp or other digital plotter specifications [subroutines SYMBOL, NUMBER, AXIS, etc.—see California Computer Products, Inc. (1976)]. It is of importance to note that individual sym-

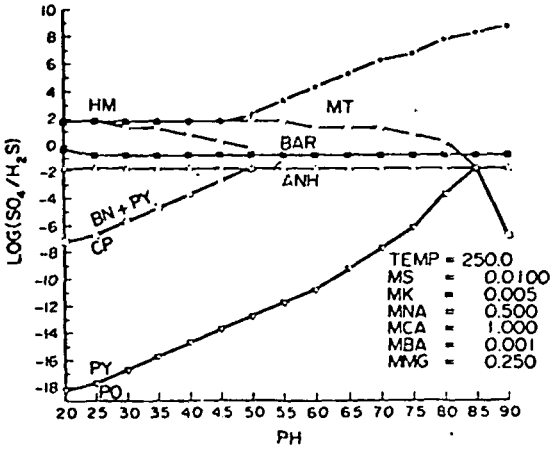


Figure 3. Log ($\Sigma SO_4/\Sigma H_2S$) vs pH diagram, drawn using Calcomp plotter.

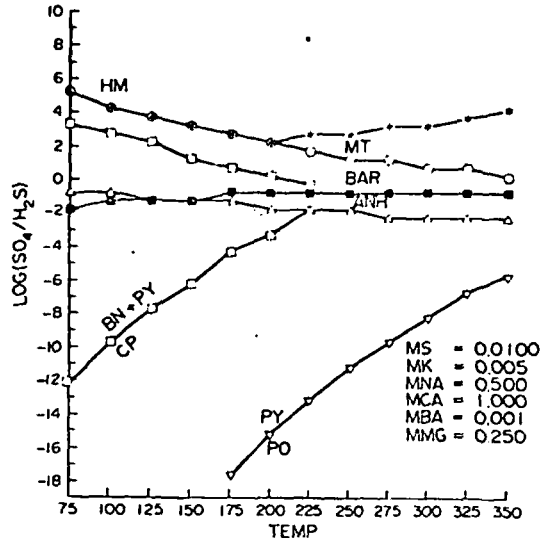


Figure 4. Log ($\Sigma SO_4/\Sigma H_2S$) vs T diagram, drawn using Calcomp plotter.

pH. Line printer.

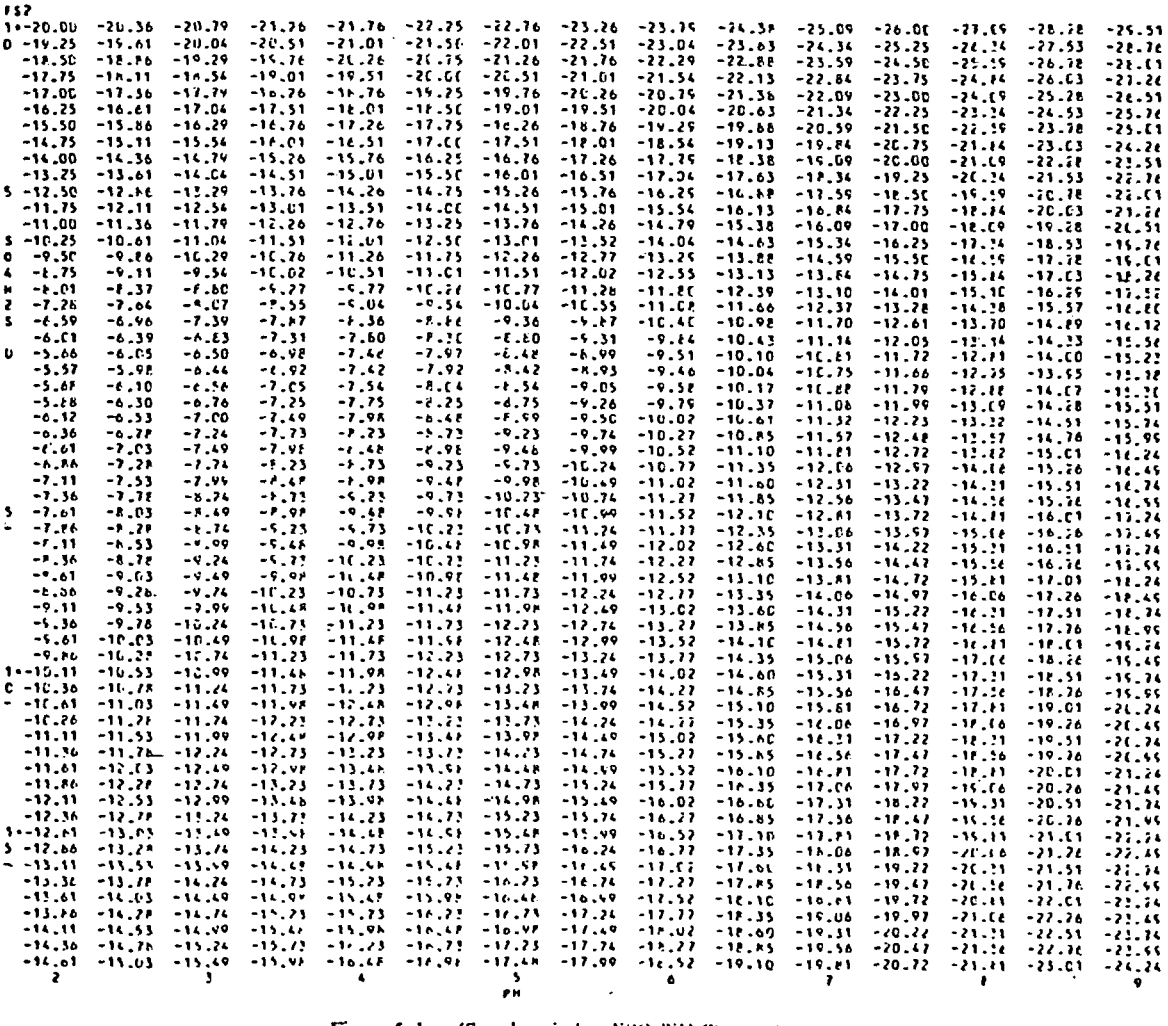


Figure 5. Log fS_2 values in log ($\Sigma SO_4/\Sigma H_2S$) vs pH space.

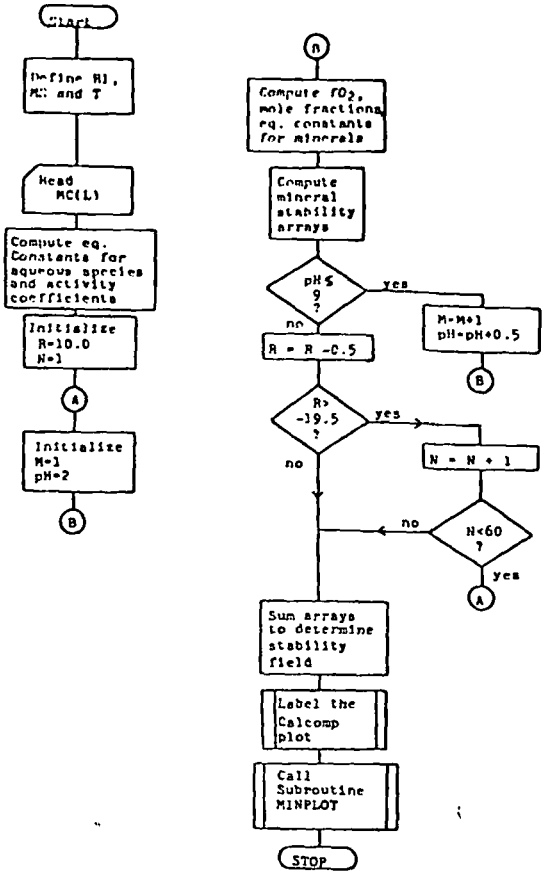


Figure 6. Generalized flow diagram for main SO4H2S-PH program.

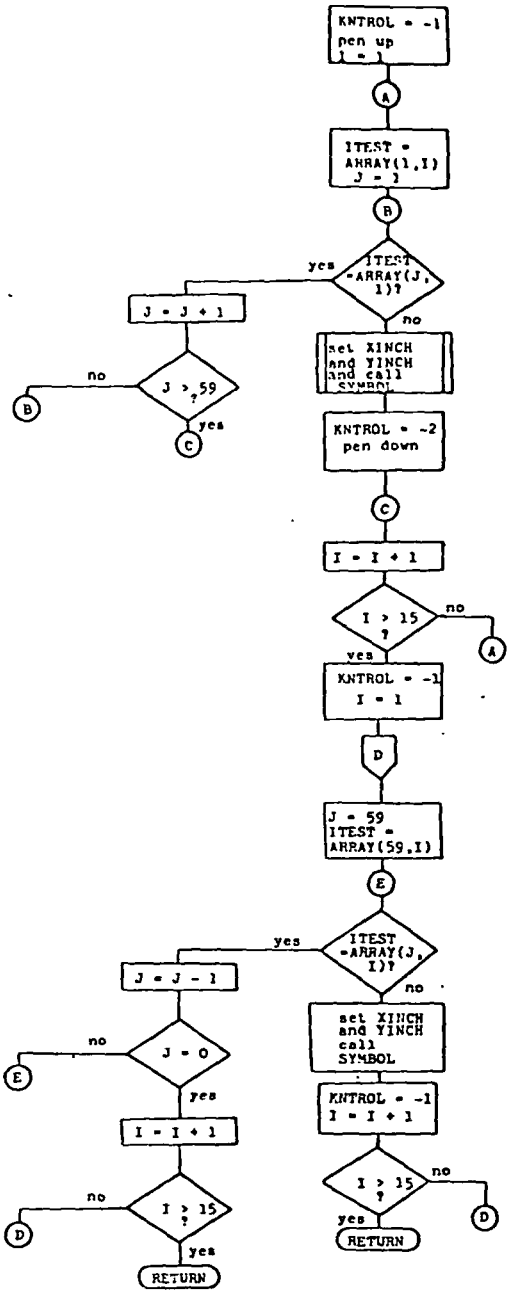


Figure 7. Generalized flow diagram for subroutine MINPLOT.

bols drawn by the plotter are connected by straight lines. In some instances it is necessary to smooth these lines. Where both upper and lower boundary lines exist it is necessary to manually draw a smooth, rounded curve between the last plotted symbols in order to complete the nose of the stability field (normally necessary only for the bornite + pyrite field, see Figure 3). The need for line smoothing diminishes as smaller increments of pH or T are employed, thereby increasing the number of values in the array.

Programs (Appendix) are written for log R vs pH or T. Generalized flow diagrams are presented in the Appendix. In order to generate log fO_2 vs pH or T diagrams it is necessary to initialize and change log fO_2 rather than log R. This step eliminates the calculation of log fO_2 as a function of R. All other calculations remain the same.

ADDITIONAL DATA OUTPUT

Values of mole fractions of each aqueous sulfur species, log fS_2 (Fig. 5), and log fO_2 at each given pH or T and log R or log fO_2 intersection can be output using the line printer. The values may be either computer or hand contoured. Only the input for mineral stability determinations is necessary. The Calcomp plot of mineral stabilities is drawn at the same scale as the line printouts for easy overlaying and tracing. Given appropriate equations for the solubility of each mineral, solution ppm values also may be plotted [e.g. ppm $FeCl^+$ in the pyrite field—see Ripley and Ohmoto (1977) for further details]. Combining aqueous equilibria and sulfur isotopic fractionation factors allows for the calculation and contouring of $\delta^{34}S$ values for various aqueous and solid sulfur species [see Ohmoto (1972) and Ripley and Ohmoto (1977), for methods].

Acknowledgments—The authors wish to thank Dr. Robert F. Blakely, Indiana University, for his helpful suggestions during the development of the plotter routines, and a critical review of the manuscript. Financial support for use of an IBM-360 computer (PSU) and CDC-6600 computer (IU) was provided by The Pennsylvania State University, Indiana University, and NSF grant EAR 76-03724 to H. Ohmoto.

REFERENCES

Barnes, H. L., and Kullerud, G., 1961, Equilibrium in sulfur-containing aqueous solutions in the system Fe-S-O, and their correlation during ore deposition: Econ. Geology, v. 56, no. 4, p. 648–688.

California Computer Products, Inc., 1976, Programming Calcomp electromechanical plotters: Calcomp, Anaheim, California, 32 p.

Garrels, R. M., and Naeser, C. R., 1958, Equilibrium distribution of dissolved surfur species in water at 25°C and 1 atm total pressure: Geochim. et Cosmochim. Acta, v. 15, no. 1/2, p. 113–131.

Helgeson, H. C., 1967, Solution chemistry and metamorphism, *in* Abelson, P. H., ed., Researches in geochemistry (v. 2): John Wiley & Sons, New York, p. 362–404.

Ohmoto, H., 1972, Systematics of sulfur and carbon isotopes in hydrothermal ore deposits: Econ. Geology, v. 67, no. 5, p. 551–578.

Pennsylvania State University Computation Center, 1968, Step-up multiple linear regression, FORTRAN/360, 6 p.

Ripley, E. M., and Ohmoto, H., 1977, Mineralogic, sulfur isotope, and fluid inclusion studies of the stratabound copper deposits at the Raul mine, Peru: Econ. Geology, v. 72, no. 6, p. 1017–1041.

Listing of Program SO4H2S - PH

```
      PROGRAM SO4H2S(INPUT,OUTPUT,PLOT,TAPE3=PLOT)
C
C                  *****DEFINITION OF VARIABLES*****
C     RI=IONIC STRENGTH
C     MC(L)=MOLALITY OF CATIONS,1=K,2=NA,3=CA,4=BA,5=MG
C     MS=MOLALITY OF TOTAL SULFUR
C     T=TEMPERATURE(DEGREES C)
C     TK=TEMPERATURE(DEGREES K)
C     K1-K15,K26,AND K30 = LOG OF EQUILIBRIUM CONSTANTS - COMPLETE
C     REFERENCES GIVEN IN RIPLEY AND OHMOTO(1977)
C     K1=H2S-HS-,NAUMOV,ET AL. 1974
C     K2=HS--S2-, NAUMOV, ET AL. 1974
C     K3=SO42--H2S,HELGESON,1969
C     K4=HSO4--SO32-,HELGESON,1969
C     K5=KSO4--SO42-,HELGESON,1969
C     K6=NASO4--SO42-,HELGESON,1969
C     K7=CASO40-SO42-,HELGESON,1969
C     K8=H2S-S2,HELGESON,1969
C     K9=PY-PO,HELGESON,1969
C     K10=PY-MT,HASS AND ROBIE,1973-ROBIE AND WALDBAUM,1968
C     K11=PY-HM,HASS AND ROBIE,1973-ROBIE AND WALDBAUM,1968
C     K13=BN+PY-CP,SCHNEEBERG,1973
C     K14=BARITE,HELGESON,1969
C     K15=ANHYDRITE,HELGESON,1969
C     K26=MT-HM,HASS AND ROBIE,1973
C     K30=MGSO40-SO42-,HELGESON,1969
C     G1-G9=LOG OF ACTIVITY COEFFICIENTS
C     G1=K+,CL-,H2CO3(AP)
C     G2=HS-,OH-
C     G3=NA+,HSO4-,KSO4-,NASO4-
C     G4=SO42-
C     G5=CO32-
C     G6=S2-,BA2+
C     G7=CA2+,FE2+,MG2+
C     G8=H+
C     G9=H2S
C     G10= ACTIVITY OF WATER
C     R=LOG(TOTAL SULFATE/TOTAL SULFIDE)
C     AA,BB,C,CC,D,E,F,W,Y =EQUATIONS FROM OHMOTO(1972) USED IN
C     CALCULATING MOLE FRACTIONS
C     XH2S,XSO4,ETC.= MOLE FRACTIONS OF AQUEOUS SULFUR SPECIES
C     PYPO THROUGH HMMT = STABILITY FIELD ARRAYS
C     PYPO=PYRITE-PYRRHOTITE
C     PYMT=PYRITE-MAGNETITE
C     PYHM=PYRITE-HEMATITE
C     POMT=PYRRHOTITE-MAGNETITE
C     BPCP=BORNITE-PYRITE-CHALCOPRITE
C     BAR=BARITE
C     ANH=ANHYDRITE
C     HMMT=HEMATITE-MAGNETITE
C     ISUMPY,ISUMPO,ISUMMT,ISUMHM=SUMMED ARRAYS FOR PY-PO-MT-HM TO
C     DETERMINE STABILITY BOUNDARIES WITH METASTABLE EXTENSIONS ELIMINATED
C     FACT,AXSLEN,AND IOPTS REFER TO AXES DRAWING IN THE CALCOMP ROUTINE
C
      DIMENSION MC(5),FS2(59,15),XH2S(59,15),XSO4(59,15),PYMT(59,15),PYP
     10(59,15),PYHM(59,15),POMT(59,15),HMMT(59,15),FO2(59,15)
      DIMENSION IOPTS(4)
      COMMON ISUMPO(59,15),ISUMPY(59,15),ISUMMT(59,15),ISUMHM(59,15),BPC
     1P(59,15),ANH(59,15),BAR(59,15)
      REAL MC,MS ,K1,K2,K3,K4,K5,K6,K7,K8,K9,K10,K11,K12,K13,K14,K15,K30
      INTEGER PYPO,PYMT,PYHM,POMT,BPCP,BAR,ANH,HMMT
C     SET IONIC STRENGTH
      RI=3.0
C     READ MOLALITY OF CATIONS
      READ 33,(MC(L),L=1,5)
   33 FORMAT(5F5.3)
C     SET MOLALITY OF TOTAL SULFUR
```

```
      MS=.01
C     SET TEMPERATURE
      T=250.00
      TK=T+273.15
C     CALCULATE ED. CONSTANTS AND ACTIVITY COEFFICIENTS
  200 K1=-21.83590+1.144752*(10**4)*(1/TK)-.02119629*(((1/TK)*10**4)*
     1*2)
      K2=-21.402171+.8979304*(10**4)*(1/TK)-.01949693*(((1/TK)*10**4)*
     2*2)
      K3=51.097150-6.2207626*(10**4)*(1/TK)+.02950445*(((1/TK)*10**4)**
     32)
      K4=-19.04772+1.0026124*(10**4)*(1/TK)-.01494899Z*(((1/TK)*10**4)*
     4*2)
      K5=-14.707674+.94694370*(10**4)*(1/TK)-.01640023Z*(((1/TK)*10**4)*
     5*2)
      K6=-14.707674+.94694370*(10**4)*(1/TK)-.01640023Z*(((1/TK)*10**4)*
     6*2)
      K7=-16.2092794.9106194Z*(10**4)*(1/TK)-.01504604Z*(((1/TK)*10**4)*
     7*2)
      K8=-3.997260l+1.0278291*(10**4)*(1/TK)-.0006058626Q*(((1/TK)*10**4
     8)**2)
       K30=-1.4339642D 01+6.7485454D-01*(10**4)*(1/TK)-9.3966842D-03*(((
     11/TK)*10**4)**2)
      G1=-.116275D-01*TK+.254149D-04*(TK**2)-.212773D-07*(TK**3)-.370492
     2D 00*RI-.560977D-03*(RI*TK)+.408358D-05*(RI*TK**2)+.562063D 00*(RI
     1**2)-.116101D-02*(RI**2*TK)-.971375D-06*(RI**2*TK**2)-.133414D 00*
     2(RI**3)+.368906D-03*(RI**3*TK)+.169091D 01
      G2=-.981695D-02*TK+.225898D-04*(TK**2)-.190552D-07*(TK**3)+.190198
     1D 00*RI-.244054D-02*(RI*TK)+.438897D-05*(RI*TK**2)+.228678D 00*(RI
     2**2)-.734320D-04*(RI**2*TK)-.984814D-06*(RI**2*TK**2)-.727801D-01*
     3(RI**3)+.171191D-03*(RI**3*TK)+0.134309D 01
      G3=-.788100D-02*TK+.181313D-04*(TK**2)-.156664D-07*(TK**3)+.222665
     1D 00*RI-.259538D-02*(RI*TK)+.468092D-05*(RI*TK**2)+.231635D 00*(RI
     2**2)-.773887D-04*(RI**2*TK)-.102592D-05*(RI**2*TK**2)-.744076D-01*
     3(RI**3)+.176906D-03*(RI**3*TK)+0.107458D 01
      G4=-.673274D-01*TK+.157173D-03*(TK**2)-.124776D-06*(TK**3)-.525777
     1D 00*RI+.318549D-03*(RI*TK)-.663783D-06*(RI*TK**2)+.267917D 00*(RI
     2**2)-.107228D-03*(RI**2*TK)-.129425D-06*(RI**2*TK**2)-.523191D-01*
     3(RI**3)+.611407D-04*(RI**3*TK)+0.904088D 01
      G5=-.604050D-01*TK+.141201D-03*(TK**2)-.112610D-06*(TK**3)-.422920
     1D 00*RI-.103522D-03*(RI*TK)+.186092D-06*(RI*TK**2)+.278724D 00*(RI
     2**2)-.163622D-03*(RI**2*TK)-.217548D-06*(RI**2*TK**2)-.568750D-01*
     3(RI**3)+.817596D-04*(RI**3*TK)+0.807875D 01
      G6=-.551895D-01*TK+.129257D-03*(TK**2)-.103499D-06*(TK**3)-.261603
     1D 00*RI-.793606D-03*(RI*TK)+.121835D-05*(RI*TK**2)+.252522D 00*(RI
     2**2)-.539835D-04*(RI**2*TK)-.433122D-06*(RI**2*TK**2)-.589110D-01*
     3(RI**3)+.926649D-04*(RI**3*TK)+0.735019D 01
      G7=-.462285D-01*TK+.108475D-03*(TK**2)-.874989D-07*(TK**3)-.984740
     1D-01*RI-.144465D-02*(RI*TK)+.240432D-05*(RI*TK**2)+.244967D 00*(RI
     2**2)-.438033D-04*(RI**2*TK)-.630625D-06*(RI**2*TK**2)-.638577D-01*
     3(RI**3)+.117713D-03*(RI**3*TK)+0.611608D 01
      G8=-.124454D-02*TK+.912956D-06*(TK**2)-.127757D-08*(TK**3)-.188045
     1D 00*RI-.967654D-03*(RI*TK)+.380400D-05*(RI*TK**2)+.462723D 00*(RI
     2**2)-.888323D-03*(RI**2*TK)-.663675D-06*(RI**2*TK**2)-.105777D 00*
     3(RI**3)+.266715D-03*(RI**3*TK)+0.303285D 00
      G9=+.125184D-01*TK-.291534D-04*(TK**2)+.219998D-07*(TK**3)+.602681
     1D 00*RI-.289303D-02*(RI*TK)+.399704D-05*(RI*TK**2)-.384144D-01*(RI
     2**2)+.356240D-03*(RI**2*TK)-.654073D-06*(RI**2*TK**2)-.801604D-02*
     3(RI**3)+.181898D-04*(RI**3*TK)-0.175251D 01
      G10=-0.183456D-03*TK+0.431672D-06*(TK**2)-0.333294D-09*(TK**3)-0.3
     102036D-01*RI-0.333711D-04*(RI*TK)+0.800007D-07*(RI*TK**2)+0.867395
     2D-02*(RI**2)-0.371796D-04*(RI**2*TK)+0.285709D-07*(RI**2*TK**2)-0.
     3177471D-02*(RI**3)+0.466669D-05*(RI**3*TK)+0.102555D 01
C     SET INITIAL LOG(SO4/H2S),R VALUE
      R=10.00
C     THERE ARE N=59 ROWS AND M= 15 COLUMNS FOR R VS PH OR T DIAGRAMS
      N=1
   15 M=1
C     SET INITIAL PH
      PH=2.0
C     CALCULATE TOTAL SULFIDE TERM A AND TOTAL SULFATE TERM B
   17 A=1+((10**K1)*(10**G9))/((10**G2)*(10**(-PH))))
      B=1+((10**G4)*(((10**(-PH))/((10**K4)*(10**G3)))+((MC(1)*(10**(G1)
     1))/((10**(K5))*(10**(G3))))+((MC(2)*(10**(G3)))/((10**(K6))*(10**(
     2G3))))+((MC(3)*(10**(G7)))/((10**(K7))*(10**(G3))))+((MC(5)*(10**(
     1G7)))A(10**(K30)))))
C     CALCULATE FO2 AS FUNCTION OF R,PH,A, AND B
      FO2(N,M)=.5*(R+K3-(2*PH)+G4+ALOG10(A)-ALOG10(B)-G9)
C     CALCULATE MOLE FRACTIONS OF SULFUR SPECIES - EQUATIONS GIVEN IN
C     OHMOTO(1972)
      AA=K1+G9+PH-G2
      AA=10.**AA
      DD=K2+G2+PH-G3
      BB=10.**BB
      C=((2*FO2(N,M)+G9+(2*PH))-(K3))
      C=10.**C
      CC=-G4
      CC=10.**CC
      D=r(PH+K4+G3)
      D=10.**D
```

```
      E=(G)+ALOG10(MC(1)))-(K5+G3)
      E=10.**E
      F=(G3+ALOG10(MC(2)))-(K6+G3)
      F=10.**F
      W=(G7+ALOG10(MC(3)))-K7
      W=10.**W
      Y=(G7+ALOG10(MC(5)))-K30
      Y=10.**Y
      SUM=1+AA*(1+BB)+C*(CC+D+E+F+W+Y)
      XH2S(N,M)=1./SUM
      XSO4(N,M)=(C*CC)/SUM
C     CALCULATE EQ. CONSTANTS FOR MINERALS
      K9=7.1628600-.76173052*(10**4)*(1/TK)+.001034461B*(((1/TK)*10**4)*
     9*2)
      K10=14.840143+1.0443311*(10**4)*(1/TK)+.0017655910*(((1/TK)*10**4)
     1**2)
      K11=5.504288B+1.3466488*(10**4)*(1/TK)-.003767323*(((1/TK)*10**4)*
     2*2)
      K12=-7.675240743.3988816*(10**4)*(1/TK)-.0019288453*(((1/TK)*10**4
     3)**2)
      K13=12.560-1.1067*(10**4)*(1/TK)
      K14=-25.9175760+1.2591227*(10**4)*(1/TK)-.023535797*(((1/TK)*10**4
     4)**2)
      K15=-25.292210+1.2831973*(10**4)*(1/TK)-.020329233*(((1/TK)*10**4)
     6**2)
      K26=14.894874-2.6110181*(10**4)*(1/TK)+.0017536020*(((1/TK)*10**4)
     4**2)
C     CALCULATE LOG FS2
      FS2(N,M)=((K8+ALOG10(MS*XH2S(N,M))+G9+(.5*FO2(N,M)))-(ALOG10(G10))
     1)*2
C     TEST FOR STABILITIES OF PY,PO,MT,HM,BN,CP,BAR,ANH
      IF(FS2(N,M)-(2*K9))27,18,18
   27 PYPO(N,M)=0
      GO TO 19
   18 PYPO(N,M)=1
   19 IF(FS2(N,M)-((K10+(2*FO2(N,M)))/3))20,21,21
   20 PYMT(N,M)=0
      GO TO 22
   21 PYMT(N,M)=1
   22 IF(FS2(N,M)-((K11+(1.5*FO2(N,M)))/2))23,24,24
   23 PYHM(N,M)=0
      GO TO 25
   24 PYHM(N,M)=1
   25 IF(FS2(N,M)-((K12+(2*FO2(N,M)))/1.5))26,37,37
   26 POMT(N,M)=0
      GO TO 38
   37 POMT(N,M)=1
   38 IF(FS2(N,M)-K13)29,30,30
   29 BPCP(N,M)=0
      GO TO 71
   30 BPCP(N,M)=1
   71 IF(XSO4(N,M)-(10**K14/(MC(4)*(10**G6)*(10**G4)*MS)))72,73,73
   72 BAR(N,M)=0
      GO TO 34
   73 BAR(N,M)=1
   34 IF(XSO4(N,M)-(10**K15/(MC(3)*(10**G7)*(10**G4)*MS)))35,36,36
   35 ANH(N,M)=0
      GO TO 3
   36 ANH(N,M)=1
    3 IF(FO2(N,M)-K26)1,2,2
    1 HMMT(N,M)=0
      GO TO 39
    2 HMMT(N,M)=1
   39 IF(PH-9.0)7,16,16
C     INCREASE PH BY 0.5
    7 PH=PH+.5
      M=M+1
      GO TO 17
C     DECREASE R BY 0.5
   16 R=R-.50
      IF(R+19.50)99,99,28
   28 N=N+1
      IF(N-60)15,99,99
C     DETERMINE STABILITY FIELDS OF PY,MT,HM,ETC.
C     PY STABLE = 3
   99 DO 211 I=1,59
      DO 210 J=1,15
      ISUMPY(I,J)=PYPO(I,J)+PYMT(I,J)+PYHM(I,J)
  210 CONTINUE
  211 CONTINUE
C     SUBROUTINE CHANGE CONVERTS 0 TO 1 AND 1 TO 0-SIMPLIFIES STABILITY
C     FIELD DETERMINATIONS
C     PO STABLE =1
C     HM STABLE =1
      CALL CHANGE(PYPO)
      DO 221 I=1,59
      DO 220 J=1,15
      ISUMPO(I,J)=PYPO(I,J)+POMT(I,J)
  220 CONTINUE
  221 CONTINUE
```

```
      CALL CHANGE(PYHM)
      DO 231 I=1,59
      DO 230 J=1,15
      ISUMHM(I,J)=HMMT(I,J)+PYHM(I,J)
  230 CONTINUE
  231 CONTINUE
C     CHANGE 3 OF PYRITE TO 1,AND 0,1,OR2 ALL TO 0
C     CHANGE 2 OF PO TO 1 , AND 1 TO 0
C     CHANGE 2 OF HM TO 1 , AND 1 TO 0
      DO 43 I=1,59
      DO 42 J=1,15
      IF(ISUMPY(I,J).NE.3) GO TO 45
      ISUMPY(I,J)=1
      GO TO 46
   45 ISUMPY(I,J)=0
   46 IF(ISUMPO(I,J).NE.2) GO TO 47
      ISUMPO(I,J)=1
      GO TO 50
   47 ISUMPO(I,J)=0
   50 IF(ISUMHM(I,J).NE.2) GO TO 51
      ISUMHM(I,J)=1
      GO TO 42
   51 ISUMHM(I,J)=0
   42 CONTINUE
   43 CONTINUE
      DO 233 I=1,59
      DO 232 J=1,15
      ISUMMT(I,J)=ISUMPY(I,J)+ISUMPO(I,J)+ISUMHM(I,J)
  232 CONTINUE
  233 CONTINUE
C     AT THIS POINT MAY PLOT FIELDS ( INCLUDING METASTABLE PROJECTIONS )
C     OF MINERALS
C     ARRAYS SUCH AS PYPO OR PYMT MAY BE PRINTED , OR ONLY STABLITY
C     FIELDS(IE ISUMPO,ETC.)
C     THE ARRAYS MAY ALSO BE SEARCHED AND ONLY AN ASSIGNED CHARACTER
C     PRINTED AT THE CHANGE FROM 0 TO 1 OR VICE VERSA
C     THE CALCOMP PLOTTER ROUTINE MAY ALSO BE CALLED , AS IS DONE BELOW
C     LABEL PLOT
      CALL IDENT(3)
      CALL SYMBOL(8.0,4.00,0.15,5HTEMP=,0.0,5)
      CALL NUMBER(9.10,4.00,0.15,T,0.00,0)
      CALL SYMBOL(8.0,3.75,0.15,3HMS=,0.0,3)
      CALL NUMBER(9.10,3.75,0.15,MS,0.00,4)
      CALL SYMBOL(8.0,3.50,0.15,3HMK=,0.0,3)
      CALL NUMBER(9.10,3.50,0.15,MC(1),0.00,3)
      CALL SYMBOL(8.0,3.25,0.15,4HMNA=,0.0,4)
      CALL NUMBER(9.10,3.25,0.15,MC(2),0.00,3)
      CALL SYMBOL(8.0,3.00,0.15,4HMCA=,0.0,4)
      CALL NUMBER(9.10,3.00,0.15,MC(3),0.00,3)
      CALL SYMBOL(8.0,2.75,0.15,4HMBA=,0.0,4)
      CALL NUMBER(9.10,2.75,0.15,MC(4),0.00,3)
      CALL SYMBOL(8.0,2.50,0.15,4HMMG=,0.0,4)
      CALL NUMBER(9.10,2.50,0.15,MC(5),0.00,3)
      FACT=7.0/(11.0+7.0/32.0)
      AXSLEN=11.0+7.0/32.0
      IOPTS(1)=-1.0
      IOPTS(2)=1
      IOPTS(3)=1
      IOPTS(4)=0
      CALL IUAXIS(0.0,2.0,FACT,0.5,2HPH,-2,1.0,1.0,AXSLEN,0,+2,+0.10,IOP
     1TS(1))
      FACT=29.0/(9.0+21.0/32.0)
      AXSLEN=9.0+21.0/32.0
      CALL IUAXIS(90.0,-19.0,FACT,1.00,12HLOG(SO4/H2S),+12,1.0,1.0,AXSLE
     1N,0,+2,+0.10,IOPTS(1))
C     SUBROUTINE MINPLOT SEARCHES ARRAYS AND PLOTS STABILITY BOUNDARIES
      CALL MINPLOT(ISUMPY,1,1)
      CALL MINPLOT(ISUMPO,2,2)
      IF(MS.LE.0.001) GO TO 94
      CALL MINPLOT(ISUMMT,3,3)
   94 CALL MINPLOT(ISUMHM,5,5)
      CALL MINPLOT(BPCP,6,6)
      CALL MINPLOT(BAR,7,7)
      CALL MINPLOT(ANH,8,8)
      CALL CLOSEPF
  199 STOP
      END
      SUBROUTINE CHANGE(ARRAY)
      DIMENSION ARRAY(59,15)
      TYPE INTEGER ARRAY
      DO 101 I=1,59
      DO 100 J=1,15
      IF(ARRAY(I,J).EQ.1) GO TO 1
      ARRAY(I,J)=1
      GO TO 100
    1 ARRAY(I,J)=0
  100 CONTINUE
  101 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MINPLOT(ARRAY,ISYMBOL,IFLAG)
C     PASS AN ARRAY , A SYMBOL TO BE PLOTTED , AND AN IDENTIFICATION
      DIMENSION ARRAY(59,15)
      COMMON ISUMPO(59,15),ISUMPY(59,15),ISUMMT(59,15),ISUMHM(59,15),BPC
     1P(59,15),ANH(59,15),BAR(59,15)
      TYPE INTEGER ARRAY
      TYPE INTEGER BAR,ANH,BPCP
      N=0
      KNTROL=-1
C     TEST FOR CHANGES IN ARRAY. IF 0>1 THEN DRAW A SYMBOL HALF WAY
C     BETWEEN THE 0 AND 1 AND GO TO THE NEXT COLUMN
C     START THE SEARCH AGAIN FROM THE BOTTOM. THIS IS DONE IN ORDER TO
C     DRAW LINES BETWEEN SYMBOLS. IF ONLY ONE BOUNDARY LINE OCCURS AT
C     ONE PH THEN OVERPRINTING SIMPLY OCCURS
C     ALSO TEST FOR CHANGES FROM ONE COLUMN TO THE NEXT. THIS IS
C     IMPORTANT FOR BARITE AND ANHYDRITE AS THEIR BOUNDARIES MAY BE
C     NEARLY VERTICAL. TEST BEGINS WHEN I.GE.2
      DO 100 I=1,15
      ITEST=ARRAY(1,I)
      DO 110 J=1,59
      IF(I.GE.2) GO TO 600
      GO TO 900
  600 IF(J.EQ.1) GO TO 700
      GO TO 900
  700 K=I-1
      IF(ARRAY(1,I).NE.ARRAY(1,K)) GO TO 800
      GO TO 900
C     SYMBOL IS STANDARD CALCOMP PROCEDURE
C     THE FOLLOWING STATEMENTS ALLOW FOR LABELLING OF STABILITY FIELDS
C     XINCH AND YINCH GIVE X AND Y COORDINATES FOR PLOTTING RELATIVE
C     TO 0-0(LOWER LEFT CORNER) OF PLOT
  800 YINCH=1.0+(59.0-J-0.5)*(9.0+21.0/32.0)/58.0
      XINCH=1.0+(I-1.5)*(11.0+7.0/32.0)/14.0
      CALL SYMBOL(XINCH,YINCH,0.1,ISYMBOL,0.0,KNTROL)
      IF(IFLAG.EQ.3) GO TO 899
      KNTROL=-2
  900 IF(ITEST.NE.ARRAY(J,I)) GO TO 120
  110 CONTINUE
  100 CONTINUE
  899 KNTROL=-1
      DO 130 I=1,15
      J=59
      ITEST=ARRAY(59,I)
  150 IF(ITEST.NE.ARRAY(J,I)) GO TO 140
      J=J-1
      IF(J.EQ.0) GO TO 130
      GO TO 150
  130 CONTINUE
      RETURN
  140 N=N+1
      XINCH=1.0+(I-1)*(11.0+7.0/32.0)/14.0
      YINCH=1.0+(59.0-(J+1)-0.5)*(9.0+21.0/32.0)/58.0
      CALL SYMBOL(XINCH,YINCH,0.1,ISYMBOL,0.0,KNTROL)
      KNTROL=-2
C     LABEL STABILITY FIELDS
C     IFLAG IS IDENTIFICATION NUMBER FOR EACH ARRAY
C     LABELLING IS REFERENCED FROM SEARCH BEGINNING AT BOTTOM
      IF(IFLAG.EQ.7.OR.IFLAG.EQ.8) GO TO 10
      IF(N.EQ.2) GO TO 75
      GO TO 130
   10 IF(N.EQ.8) GO TO 11
      GO TO 130
   11 KNTROL=-1
      YINCH=YINCH+0.10
      IF(IFLAG.EQ.7) GO TO 60
      IF(IFLAG.EQ.8) GO TO 61
   75 KNTROL=-1
      YINCH=YINCH+0.25
      IF(IFLAG.EQ.6) GO TO 62
      IF(IFLAG.EQ.1) GO TO 63
      IF(IFLAG.EQ.2) GO TO 65
      IF(IFLAG.EQ.5) GO TO 66
      IF(IFLAG.EQ.3) GO TO 67
   60 CALL SYMBOL(XINCH,YINCH,0.2,3HBAR,0.0,3)
      YINCH=YINCH-0.10
      GO TO 69
   61 CALL SYMBOL(XINCH,YINCH,0.2,3HANH,0.0,3)
      YINCH=YINCH-0.10
      GO TO 69
   62 CALL SYMBOL(XINCH,YINCH,0.2,5HBPY+PY,20.0,5)
      YINCH=YINCH-0.50
      CALL SYMBOL(XINCH,YINCH,0.2,2HCP,20.0,2)
      YINCH=YINCH+0.50
      GO TO 67
   63 CALL SYMBOL(XINCH,YINCH,0.2,2HPY,0.0,2)
      GO TO 67
   65 YINCH=YINCH-0.50
```

```
      CALL SYMBOL(XINCH,YINCH,0.2,2HFO,0.0,2)
      YINCH=YINCH+0.50
      GO TO 67
   66 CALL SYMBOL(XINCH,YINCH,0.2,2HHM,0.0,2)
      XINCH=XINCH+6.0
      CALL SYMBOL(XINCH,YINCH,0.2,2HMT,0.0,2)
      XINCH=XINCH-6.0
   67 YINCH=YINCH-0.25
   69 CALL SYMBOL(XINCH,YINCH,0.1,ISYMBOL,0.0,KNTROL)
      KNTROL=-2
      GO TO 130
  120 XINCH=1.0+(I-1)*(11.0+7.0/32.0)/14.0
      YINCH=1.0+(59.0-J-0.5)*(9.0+21.0/32.0)/58.0
      CALL SYMBOL(XINCH,YINCH,0.1,ISYMBOL,0.0,KNTROL)
      KNTROL=-2
      GO TO 100
      END
```

LEAST SQUARES POLYNOMIAL FITTING TO GRAVITATIONAL DATA AND DENSITY PLOTTING BY DIGITAL COMPUTERS*

STEPHEN M. SIMPSON, JR.†

ABSTRACT

The fitting of a nth order polynomial in x and y to gravity data by least squares is discussed. A consideration of the normal equations for the general case shows certain simplifications resulting from rectangularity in data distribution. Some sample residual maps are constructed. Density plotting, made possible by the digital computer, is described and illustrated. It is shown that this process can serve as a substitute for contouring when only a qualitative picture is desired.

INTRODUCTION

There is some question whether or not the removal of regional effects from gravity may be effectively accomplished by the method of least squares. One of the basic assumptions in its application is that the regional be a relatively low order effect. By this is meant that, over the area analyzed, the true regional could be well approximated by a low order polynomial while other anomalies could not. In gravity work this is almost by definition a justified assumption. Agocs (1951) sets up an artificial example in which this is the case. He uses a plane surface, or first order polynomial, for the fitting, and demonstrates how the residual anomaly is better determined by least squares than by the use of the "arithmetic mean regional" procedure. Perhaps the strongest argument in favor of least squares residuals as opposed to methods in which the regional is estimated locally (by integration or summation of Bouguer gravity in a neighborhood of the point considered) is that the user has a tight control on the order of the approximating function. High local anomalies or blunders in observation must give locally unrealistic estimates of the regional from local summation methods. It may be argued that these very anomalies are weighted by their squares in the least squares method, thereby causing undue effects in the overall picture. However, if a reasonably large number of data points is considered, the actual effect is a slight shifting or warping of the polynomial surface, without materially affecting the relative residual contours. In any event, for the purposes of this paper it is assumed that the method deserves further study.

For practical application it is desirable to use polynomials of various orders. Computation and data handling become a serious problem as the polynomial order increases, which probably tends to hinder experimentation with the least squares method. It is the principal aim of this paper to discuss the use of poly-

* Manuscript received by the Editor December 29, 1953.

† Instructor, Department of Geology and Geophysics, Massachusetts Institute of Technology, Cambridge, Mass.

nomials of general order and the possibilities of reducing the computational problem. In addition the author wishes to present a possible alternative to contour mapping as a means of examining residuals. This new method, made possible by the high speed digital computer, has very general application to contouring problems. Special applications of the technique to other geophysical data reduction problems are being investigated at MIT.

THE NORMAL EQUATIONS

Gravity values $g(xy)$ are given at discrete points over a pattern P in the xy plane, and it is desired to approximate these values by a polynomial such as

$$G(xy) = c_{00} + c_{10}x + c_{01}y,$$

with c_{00}, c_{10}, and c_{01} taking those values which will make

$$I(c_{00}, c_{10}, c_{01}) = \sum_P [g(xy) - G(xy)]^2 = \sum_P R^2(xy)$$

assume a minimum value. $R(xy)$ may then be considered as the gravity value corrected, in this case, for a linear regional trend.

For a polynomial of order n,

$$G(xy) = \sum_{i=0}^{n} \sum_{j=0}^{n-i} c_{ij}x^i y^j, \tag{1}$$

$I(c_{ij}\text{'s})$ may be expressed as

$$I(c_{ij}\text{'s}) = \sum_P \left[\sum_{i=0}^{n} \sum_{j=0}^{n-i} \sum_{k=0}^{n} \sum_{l=0}^{n-k} c_{ij}c_{kl}x^{k+i}y^{l+j} \right] - 2\sum_P g(xy) \sum_{i=0}^{n} \sum_{j=0}^{n-i} c_{ij}x^i y^j + \sum_P g^2(xy).$$

To minimize I we differentiate partially with respect to c_{ij} and equate with zero.

$$\frac{\partial I}{\partial c_{ij}} = 2\sum_{k=0}^{n} \sum_{l=0}^{n-k} c_{kl} \sum_P x^{k+i}y^{l+j} - 2\sum_P g x^i y^j = 0$$

or

$$\sum_{k=0}^{n} \sum_{l=0}^{n-k} c_{kl} \sum_P x^{k+i}y^{l+j} = \sum_P g x^i y^j$$

where

$$\begin{aligned} j &= 0, 1, \cdots, (n-i) \\ i &= 0, 1, \cdots, n. \end{aligned} \tag{2}$$

There are the standard normal equations. It is useful to use matrix notation to discuss the properties of this system of equations. In this notation the system becomes:

$$\left[\sum_P x^{k+i}y^{l+j}\right][c_{kl}] = \left[\sum_P gx^iy^j\right]. \tag{3}$$

The formal solution is

$$[c_{kl}] = \left[\sum_P x^{k+i}y^{l+j}\right]^{-1}\left[\sum_P gx^iy^j\right]. \tag{4}$$

DISCUSSION OF THE NORMAL EQUATIONS

The first feature to notice about equation (3) is that the matrix $[\sum x^{k+i}y^{l+j}]$ is a function only of n and P. The inverse matrix therefore does not depend on gravity values and may be found for any polynomial order once the pattern P is given. If the same pattern P is used in various geologic areas the same inverse applies and the solution is obtained by finding the column vector and premultiplying by a known inverse matrix. This is significantly simpler than inverting matrices for each set of data.

It is particularly fortunate that the patterns used in taking gravity data not only are repeated but in addition are frequently rectangular (excluding subsurface studies). Fr. A. Willers (1948, p. 322) demonstrates features resulting

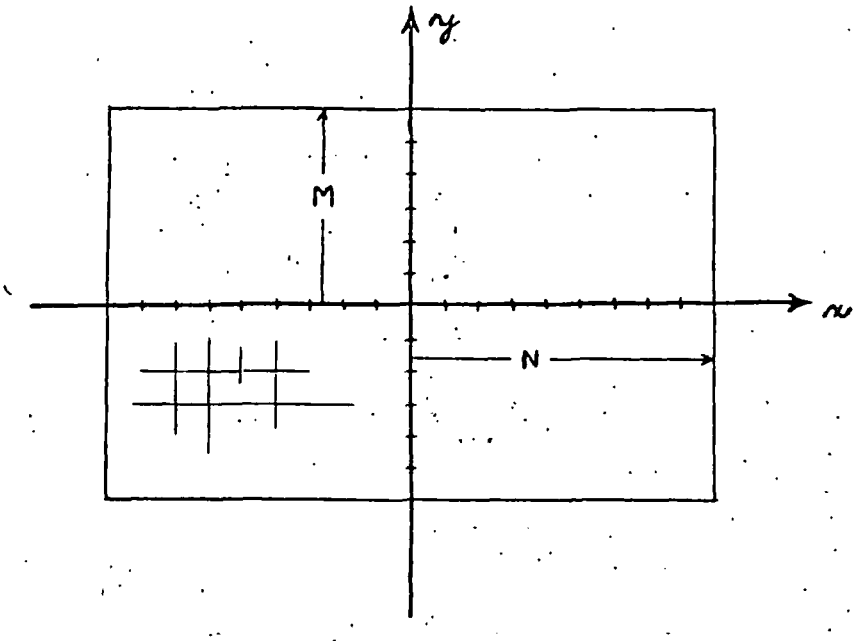


FIG. 1. The rectangular pattern of observations.

from the (normalized) rectangular pattern in the continuous case. In the discrete case axes may be chosen so that the rectangle is symmetrical about them, and ordinates and abscissae taken to be integers as in Figure 1. For this pattern an element of the matrix $[\sum x^{k+i}y^{l+j}]$ becomes

$$\sum_P x^{k+i}y^{l+j} = \sum_{\alpha=-N}^{N} \sum_{\beta=-M}^{M} \alpha^{k+i}\beta^{l+j} = \left[\sum_{-N}^{N} \alpha^{k+i}\right]\left[\sum_{-M}^{M} \beta^{l+j}\right]. \qquad (5)$$

When either $k+i$ or $l+j$ is odd the corresponding summation vanishes and the element is zero. Only when $k+i$ and $l+j$ are both even is an element non-zero. Now the element $\sum x^{k+i}y^{l+j}$ is the coefficient of c_{kl} in the equation specified by i and j (equation (2)). Divide the c_{kl}'s and the equations into four groups each as follows:

I k odd, l odd	IA i odd, j odd
II k odd, l even	IIA i odd, j even
III k even, l odd	IIIA i even, j odd
IV k even, l even	IVA i even, j even.

All equations in group IA have non-zero coefficients only for c_{kl} in class I since for groups II, III, and IV either $k+i$ or $l+j$ is odd. Similarly group IIA involves only c_{kl} in class II, etc. Furthermore, since $\{kl\}$ and $\{ij\}$ assume the same range of values, there are as many members of a particular equation group as there are members of the corresponding c_{kl} group. In other words, the original system of normal equations has broken down into four independent systems, each sufficient to determine all the c_{kl} in one of the four groups of these unknowns. As might be expected by symmetry, equation groups IIA and IIIA have identical coefficient matrices in the case of a square grid.

The above statements apply to a polynomial of any order. Between polynomials of consequetive order there exists a relationship which may be stated as follows: if the order n of the polynomial is odd (even), all c_{kl} for which $k+l$ is odd (even) will be the same as for the polynomial of order $n+1$. This follows from the fact that, if n is odd the extra c_{kl}'s dealt with in the $(n+1)$th polynomial must all have even $k+l$ so that only equation groups IIA and IIIA are affected. A similar statement holds if n is even.

Summary of Properties of Normal Equations When a Rectangular Grid is Used

1. The normal equations break down into four independent systems corresponding to four classes of coefficients. (Two of these systems have identical coefficient matrices for a square grid.)
2. Specifying the rectangular grid specifies all four inverse matrices regardless of gravity values on the grid.
3. Two of the four systems for an nth order polynomial must be equivalent to two of the four systems for the $(n+1)$th order polynomial.

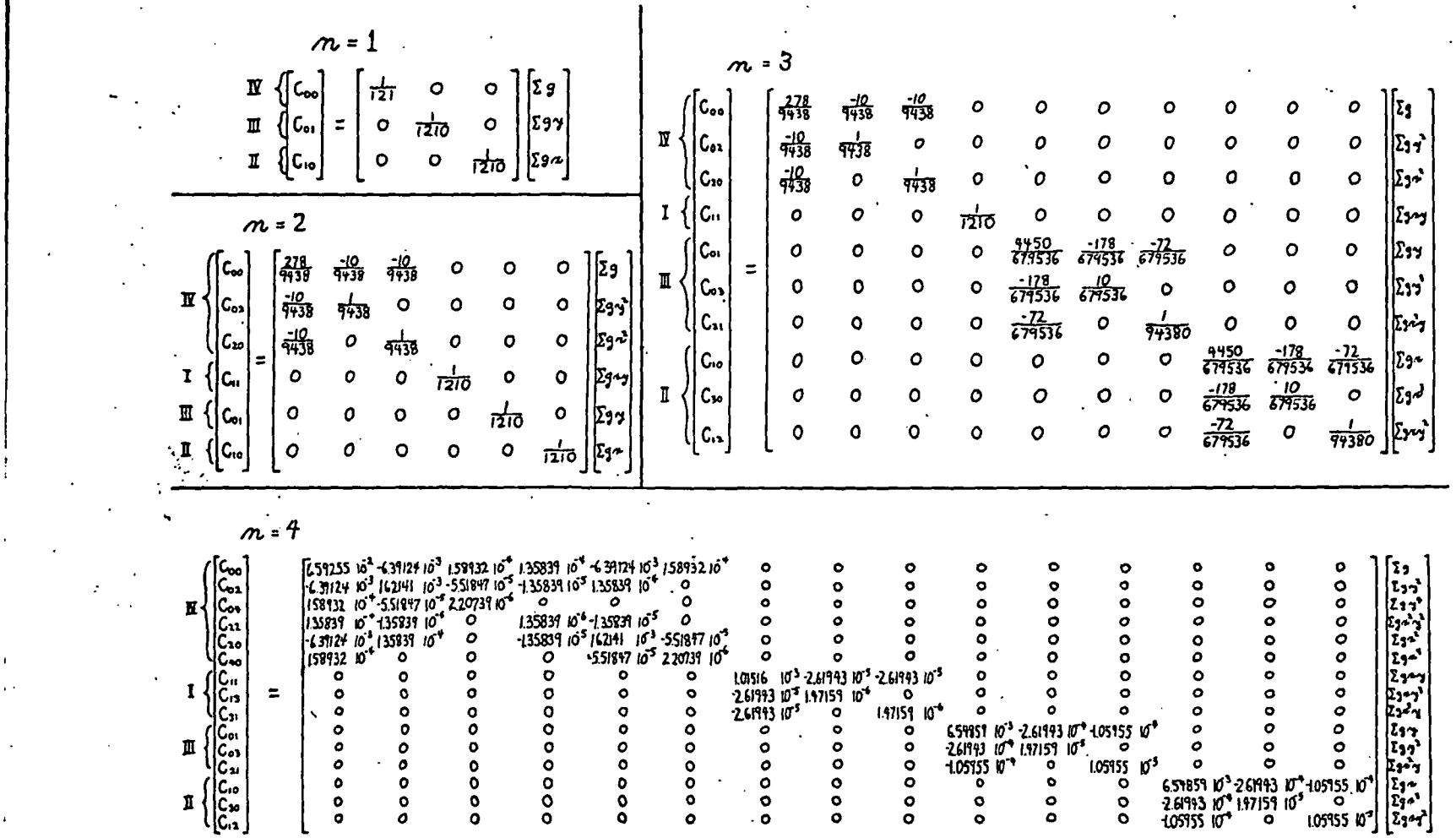


FIG. 2. The solutions in matrix form of the normal equations for various polynomial orders n. Solutions are for a square grid of 121 points.

4. All independent systems have coefficient matrices which are positive definite and symmetric, so that non-singularity is guaranteed.[1]

FORMATION OF THE NORMAL EQUATIONS

The symmetry properties of rectangular grids which reduce normal equations to four separate systems, further simplify the computation of the elements of the coefficient matrix. Limiting the polynomial to order four or less, the relevant elements are

$$\begin{array}{cccccccccc} \sum x^0y^0 & \sum x^2 & \sum x^4 & \sum x^6 & \sum x^8 & \sum y^2 & \sum y^4 & \sum y^6 & \sum y^8 \\ \sum x^2y^2 & \sum x^2y^4 & \sum x^2y^6 & \sum x^4y^2 & \sum x^4y^4 & \sum x^6y^2 \end{array} \tag{6}$$

where the summations are taken over the grid considered.

Referring to Figure 1 it is evident that $\sum x^k$ over the grid equals $\sum x^k$ on a single horizontal line times the number of lines. Thus

$$\sum x^k = (2M + 1) \sum_{\alpha=-N}^{N} \alpha^k,$$

but since k is even

$$\sum x^k = 2(2M + 1) \sum_{\alpha=1}^{N} \alpha^k. \tag{7}$$

Likewise

$$\sum y^l = 2(2N + 1) \sum_{\alpha=1}^{M} \alpha^l. \tag{8}$$

From equation (5).

$$\sum x^k y^l = \left[\sum_{-N}^{N} \alpha^k \right] \left[\sum_{-M}^{M} \beta^l \right] = 4 \left[\sum_{\alpha=1}^{N} \alpha^k \right] \left[\sum_{\beta=1}^{M} \beta^l \right]. \tag{9}$$

Hence the sums (6) are easily derivable from relations (7), (8), and (9) if the quantities $\sum_{\alpha=1}^{N} \alpha^k$ are tabulated. Using this method the elements (6) were computed for six representative grids measuring 10 by 10, 10 by 20, 20 by 20, 30 by 30, 40 by 40, and 50 by 50. They are shown in Table I.

EXAMPLES

From Table I the normal equations were set up for a square grid of 121 points ($N = 5$, $M = 5$). The solutions, in matrix form, to these equations are shown in Figure 2 for polynomials of order 1, 2, 3, and 4. Figure 2 illustrates graphically

[1] The matrix $[\sum xy]$ may be expressed as X^TX, X real, which guarantees symmetry and positive definiteness. The sub-systems may be shown to correspond to principal minors of $[\sum xy]$, for which these properties also hold (Frazer, Duncan, and Collar, 1946, p. 33; Cramer, 1946, p. 115).

TABLE I

$\Sigma x^k y^l$ OVER VARIOUS GRIDS MEASURING $2N \times 2M$
for $k=0, 2, 4, 6, 8$ $l=0, 2 \cdots (8-k)$

	$N=5,\ M=5$	$N=5,\ M=10$	$N=10,\ M=10$	$N=15,\ M=15$	$N=20,\ M=20$	$N=25,\ M=25$
$\Sigma x^0y^0=$	121	231	441	961	1,681	2,601
$\Sigma x^2 =$	1,210	2,310	16,170	76,880	235,340	563,550
$\Sigma x^4 =$	21,538	41,118	1,063,986	11,055,344	59,258,612	219,671,790
$\Sigma x^6 =$	451,330	861,630	83,093,010	1,889,941,040	17,749,376,420	101,885,895,150
$\Sigma x^8 =$	10,185,538	19,445,118	7,044,715,986	351,321,903,344	5,784,339,914,612	51,429,792,671,790
$\Sigma y^2 =$	1,210	8,470	16,170	76,880	235,340	563,550
$\Sigma y^4 =$	21,538	557,326	1,063,986	11,055,344	59,258,612	219,671,790
$\Sigma y^6 =$	451,330	43,524,910	83,093,010	1,889,941,040	17,749,376,420	101,885,895,150
$\Sigma y^8 =$	10,185,538	3,690,089,326	7,044,715,986	351,321,903,344	5,784,339,914,612	51,429,792,671,790
$\Sigma x^2y^2=$	12,100	84,700	592,900	6,150,400	32,947,600	122,102,500
$\Sigma x^2y^4=$	215,380	5,573,260	39,012,820	884,427,520	8,296,205,680	47,595,554,500
$\Sigma x^2y^6=$	4,513,300	435,249,100	3,046,743,700	151,195,283,200	2,484,912,698,800	22,075,277,282,500
$\Sigma x^4y^2=$	215,380	1,507,660	39,012,820	884,427,520	8,296,205,680	47,595,554,500
$\Sigma x^6y^2=$	4,513,300	31,593,100	3,046,743,700	151,195,283,200	2,484,912,698,800	22,075,277,282,500
$\Sigma x^4y^4=$	3,833,764	99,204,028	2,567,043,556	127,180,677,376	2,088,984,590,224	18,552,747,144,100

the various simplifying properties associated with rectangular distributions. The elements of the solution vectors are arranged according to the groups I, II, III, and IV to show the independence of these groups.

Figure 2 is used in the following way. Suppose it is desired to fit gravity data over such a grid by a plane surface

$$G(xy) = c_{00} + c_{10}x + c_{01}y.$$

It is necessary to compute the quantities $\sum g$, $\sum gx$, and $\sum gy$. $G(xy)$ may then be written immediately as

$$G(xy) = \frac{\sum g}{121} + \frac{\sum gx}{1,210}x + \frac{\sum gy}{1,210}y.$$

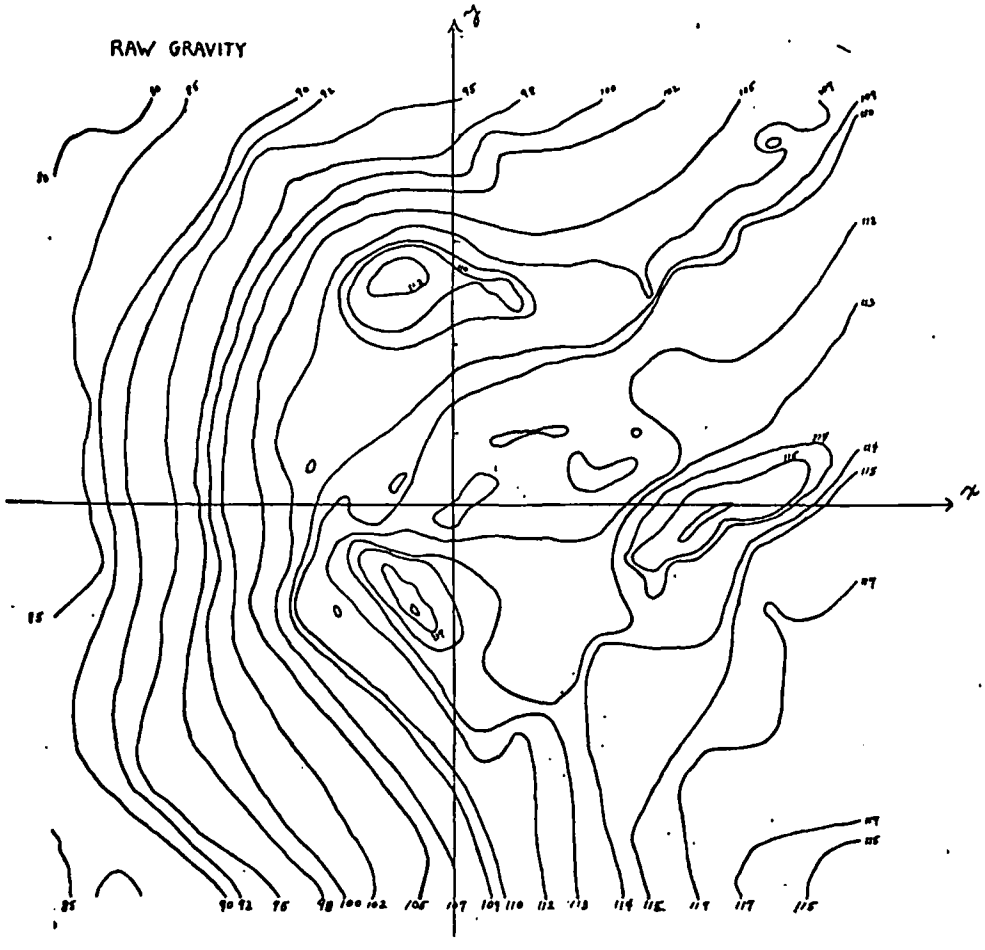


FIG. 3. Raw gravity to be analyzed.

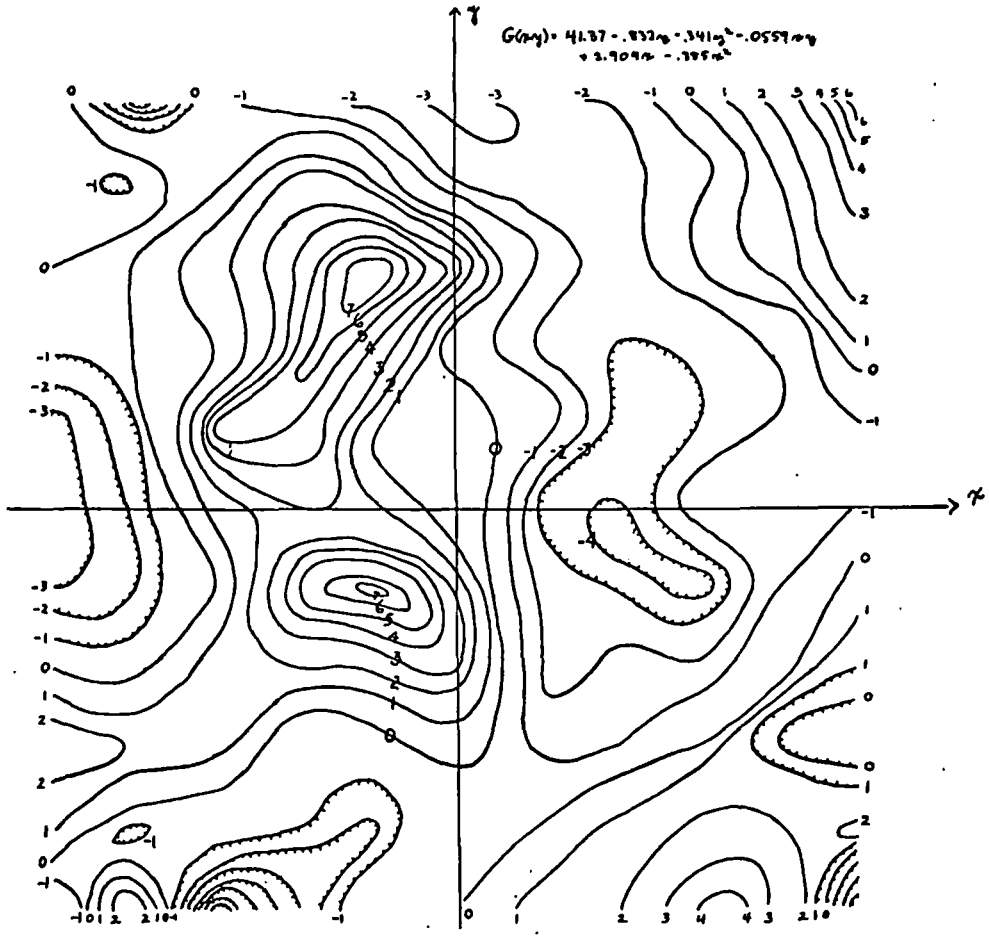


FIG. 4. Residual gravity with second order polynomial.

Following this, the equation must be solved at each point on the grid to determine the residuals. For higher order polynomials the determination of the c_{kl}'s is also quite rapid.

Figure 3 is a contour of gravity readings taken over a grid of this type. For these data, topographic corrections were known to be low order effects (approximately second order) and were not removed. Because of this, a first order polynomial was not used. Instead the data were fitted successively by polynomials of order 2, 3, and 4 as an experiment in removing the regional and topographic effects simultaneously. The resulting residual contours are shown in Figures 4, 5, and 6.

Without evaluating the contours individually, it seems that there is a closer similarity between them than might be expected. This similarity is better illustrated in Figure 7, which represents contours of the three polynomials themselves.

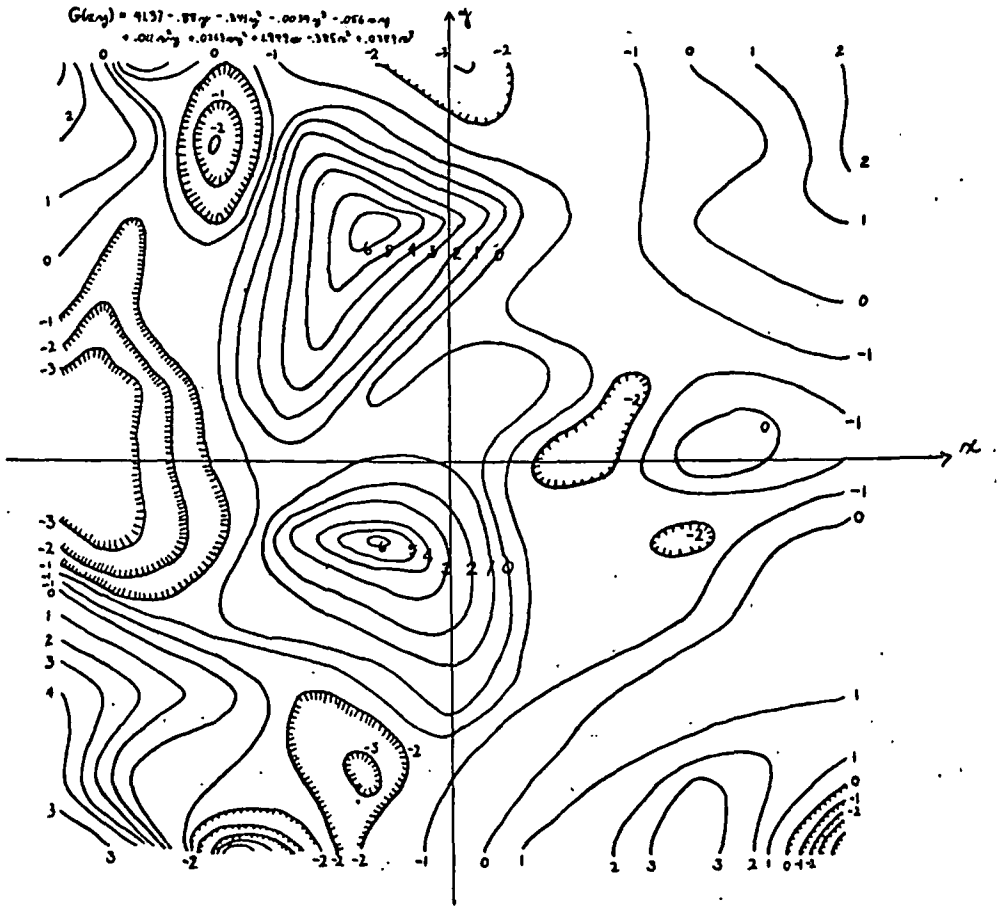


FIG. 5. Residual gravity with third order polynomial.

The figure points out that the regional (plus topographic) anomaly approximations were not rapidly affected by the addition of higher order terms. The indication here is that the true regional anomaly is lower than fourth order and that other anomalies are considerably higher order effects. In this situation the second or third order contours are probably to be preferred.

MACHINE COMPUTATIONS AND THE DENSITY PLOT

The preceding discussion has been concerned with reducing the problem of finding the solutions of normal equations to a minimum. There still remain the formation of the vector $[\sum g x^i y^j]$, the computation of the residuals, and the contouring of the results. These are all time consuming tasks. The author has been privileged to have the use of the Whirlwind I computer in his work at MIT and

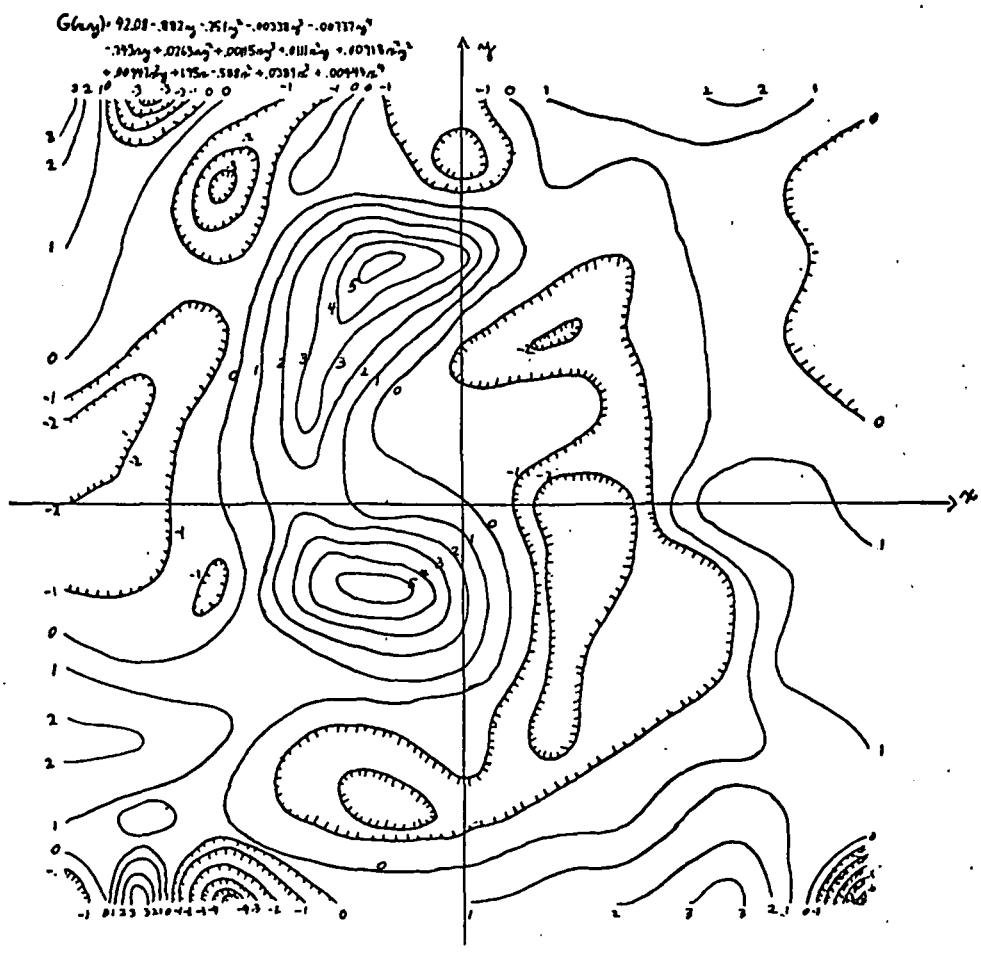


FIG. 6. Residual gravity with fourth order polynomial.

would like to illustrate ways in which machines of this nature can handle such problems.

Whirlwind I is a high speed digital computer of high reliability. It is equipped with a "delayed printer" form of output, by which computational results may be transferred rapidly to magnetic tape. Then, while the machine is computing on another problem, a Flexowriter types out the results from the tape. Results may also be removed on an oscilloscope, photographs being taken of the displays. These are especially useful features in problems where there is a high ratio of output to computation, such as the problem under discussion.

Figure 8 is a reproduction of the output form of a "program" which computes residuals over rectangular grids. The residuals are automatically typed out over the same rectangular pattern to allow contouring without transcribing. The

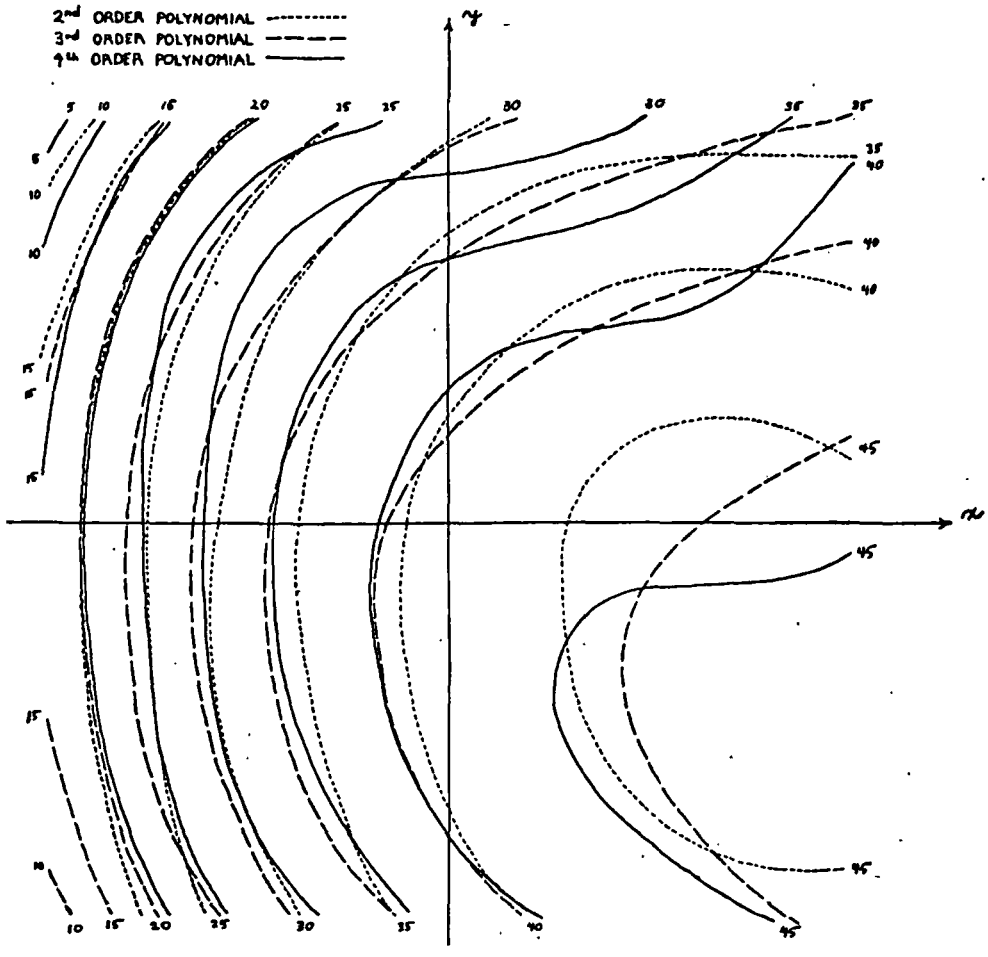


FIG. 7. Contours of the polynomials for Figures 4, 5, and 6.

machine time used in this case (these are the residuals of Figure 6) including input, computation, and recording the residuals on magnetic tape is less than one minute.

Using the oscilloscope it is possible to do a form of contouring directly. However, it is also possible to present the results in what might be called a density plot display. Within each little square surrounding a point on the scope (representing a point on the xy plane) is displayed a number of spots corresponding to the residual value at the point. Actually the number of spots displayed is logarithmically proportional to the residual because of the non-linearity of eye response to light intensity. The result is a shaded diagram in which the residual peaks stand out as white areas. Using a "program" which does this, a photograph of the residuals for the second order polynomial used elsewhere in this paper was

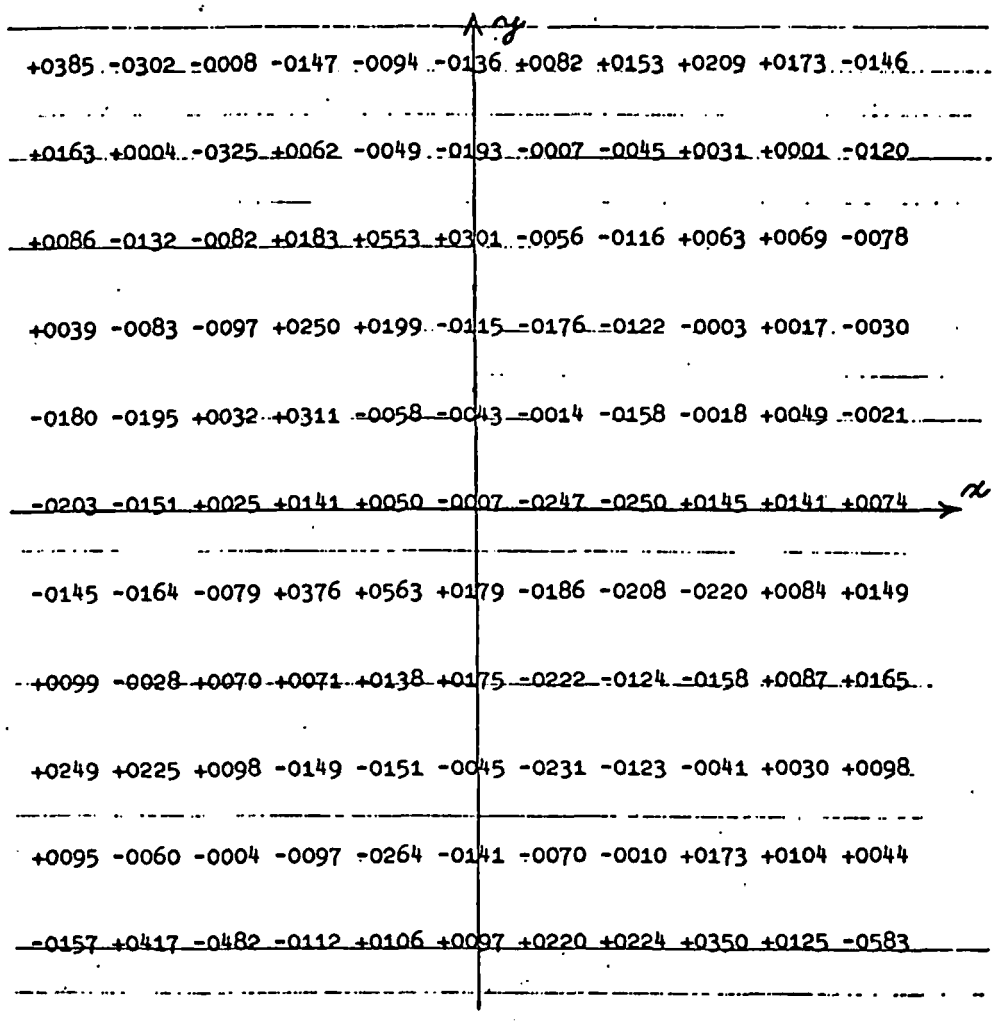


FIG. 8. Sample output of a program which computes residuals.

taken. It is shown in Figure 9 alongside of the corresponding contour map for comparison. The display takes about 25 seconds of machine time.

Residuals presented as density plots as in Figure 9 have immediate visual appeal and do not require the close examination that contours do. Furthermore the plot is unique and wandering contours in flat areas are avoided. It is apparent that the technique is not confined to gravity residuals but applies equally well to magnetic data or to any other problem involving two dimensional contours.

Further descriptions of Whirlwind are found in various issues of *Digital Computer News Letter*. Other features and coding details are described in Report R-196 of Electronic Computer Division, Servomechanisms Laboratory, MIT.

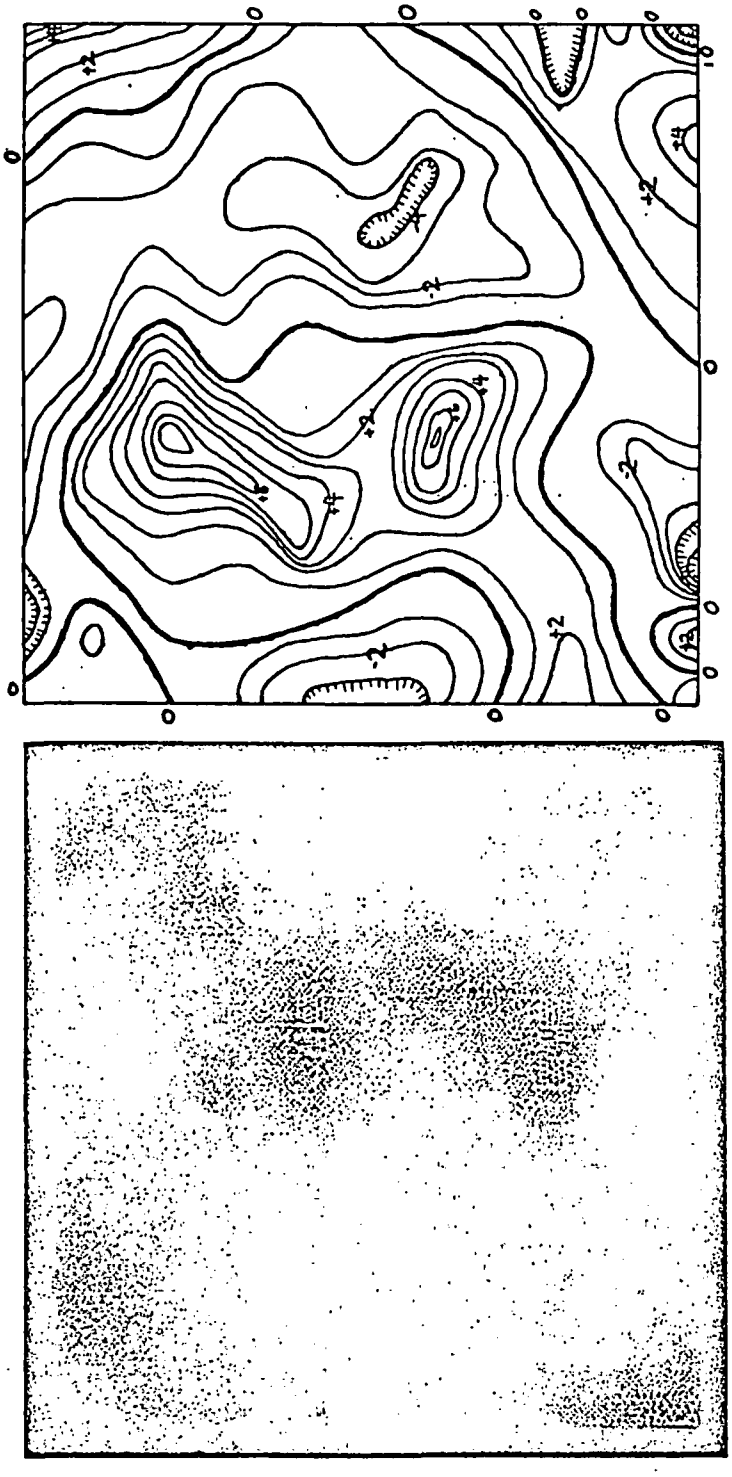


FIG. 9 The density plot display as compared with the more usual contour map.

CONCLUSIONS

The foregoing review of least squares polynomial fitting suggests a method of bringing polynomials of order greater than one within practical range. This is accomplished by a standardization of the pattern on which data are taken. For any pattern which is rectangular the solutions may be represented in the form of Figure 2.

As for density plotting, the author is convinced that digital computers have great versatility in treating geophysical problems. He presents this technique as an application of the digital computer which may not have been previously considered.

ACKNOWLEDGMENTS

The author is grateful to T. R. Madden for suggesting the least squares study. He wishes to acknowledge the assistance of M. K. Smith in developing the idea of density plotting. Thanks are due also to the personnel of the MIT Digital Computer Laboratory for their cooperation.

REFERENCES

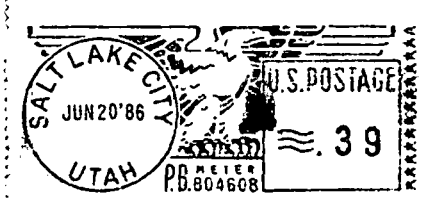
Agocs, W. B., 1951, Least squares residual anomaly determination: Geophysics, v. 16, p. 686–690.

Cramer, Harold, 1946, Mathematical methods of statistics: Princeton, Princeton Univ. Press.

Frazer, R. A., W. J. Duncan, and A. R. Collar, 1946, Elementary matrices: New York, The MacMillan Company.

Willers, Fr. A., 1948, Practical analysis: New York, Dover Publications, Incorporated.

Earth Science Laboratory
391 Chipeta Way
Salt Lake City, Utah 84108



circulate

THE UTAH COMPUTER LAW AND BUSINESS REPORT

a publication of

THE UTAH COMPUTER LAW AND BUSINESS FORUM

(a nonprofit corporation)

Spring 1986 Issue

CONTENTS

The Utah Computer Law and Business Report is published on a quarterly basis. Information and news relevant to or of interest to computer-related Utah businesses or Utah computer professionals (technical, management, marketing, public relations, sales, financial, etc.) is welcome. Communications should be should be directed to: Jon C. Christiansen, Van Cott, Bagley, Cornwall & McCarthy, P.O. Box 45340 Salt Lake City, Utah 84145 or (801)532-3333.

DISCLAIMER

The contents of this Report are for general informational purposes only and the Report and its authors disclaim any liability for any use thereof. This Report is not a substitute for your legal, accounting, financial and business advisors.

FEATURE ARTICLES

CERICOR - ANOTHER SUCCESS STORY

We are pleased to report an additional inside view of the sale of assets of Cericor, Inc. to Hewlett-Packard Company ("HP"). Cericor was a local high tech company competing in the computer aided engineering marketplace. Public announcements have indicated that HP proposes to maintain the Cericor operation in Salt Lake City. HP paid $30,800,000 for Cericor. HP is a six billion dollar company with worldwide operations and 85,000 employees. Cericor is 2-1/2 years old and has 64 employees.

For a better picture of what happened and what is expected, we arranged for this interview with James U. Jensen, who was Cericor's Chief Financial Officer at the time of the acquisition. Jim is a founding partner in Jensen & Swinton, a local law firm, emphasizing start-up and high growth representation. Jim is the founder of the Mountain West Venture Group, a local venture capital club. He previously served as Vice President and General Counsel of Dictaphone Corporation in Rye, New York and has advanced degrees in law and business from Columbia University. He lectures widely on high-tech and start up matters.

UCL&BR: Jim, how did you first become involved with Cericor?

Jensen: I began working with Cericor as outside counsel in the fall of 1983. The company had previously been started and through my affiliation with Van Cott, Bagley, I was privileged to work on Cericor matters.

UCL&BR: Could you tell from the beginning that the company had the potential growth shown by the sale to HP?

Jensen: Perhaps the single most important ingredient in success of a high-tech start up is the people. This company was lead by a very capable CEO, David L. Bailey. With him were several other key players who formed the founding team. Other talented people were added regularly. As the team grew, it seemed to continue to work well together. The company had two other ingredients which I found attractive. First was a commitment to have all employees share in the equity of the company. Second was a sense of urgency in completing the product and getting it to market.

UCL&BR: Speaking of product, what is the product and why didn't you mention that as a key ingredient?

Jensen: The company's other founders, Stan Coleby, Mike Forster and Ed Allred had worked together at Evans & Sutherland and had developed an understanding of object-oriented data-bases and a system for handling complexity which Cericor applied to computer aided engineering. Many people have recognized this field as providing considerable growth opportunities. Certainly, the product is important in any start-up. The product in this instance was a reflection of the founders' vision that newly developed technology could be applied to a growing market need. In the hands of a less talented or committed team, the "product" may never have received the recognition of its value shown by the HP deal.

UCL&BR: Could Cericor have achieved its goal without HP?

Jensen: At this point we will never know. It is clear that HP had become an important Cericor customer and that HP wanted to purchase the Cericor assets before considerable sales had occurred. Any high-tech start-up must realize that one important avenue for the technology to make its way to the market is through combining the company's product with products of one or more larger established companies.

In the final analysis, HP decided that it wanted to apply the technology more directly to HP's product line and wanted all of the attention of the founders; the HP offer was sufficiently attractive that the stockholders of Cericor decided to accept.

UCL&BR: Speaking of stockholders, did the company have venture capital backing and did it have many Utah stockholders?

Jensen: Perhaps it is a comment on the availability of risk-oriented Utah investors that only a handful of original investors were from Utah. The company had raised over 7 million dollars from several rounds of financing prior to selling to HP, but the presence of a so called Venture Capital fund came relatively late in the company's history. Nonetheless, four outside investors had seats on the seven-member board of directors; directors from HP and Data General joined the company in midyear 1985. The two other outside directors joined the company previously. Each outside director brought to the company considerable experience, knowledge, contacts and judgment -- leadership depth which is often seen lacking in other startups.

UCL&BR: It was previously announced that Cericor had sold stock to HP and to Data General in equal amounts totaling about 22% of the company. How did it come about that these two large computer companies bought into a start-up at the same time?

Jensen: As you know, the high-tech field has its ups and downs just like any other industry. Many large companies see smaller start-ups as a valuable window on new technology and a way to stay ahead of the technical advances of competitors.

UCL&BR: Was it a handicap to Cericor that it was located here in Utah?

Jensen: Yes and no. This is where the founders lived, so it was natural to start the company here. Utah has a very supportive environment for entrepreneurs, and the government is becoming more supportive. It is a long plane ride, however, to many markets, including the capital markets. Competent employees can be attracted here (although not everyone would choose to move from Boston or Palo Alto to Salt Lake City). Competent outside professionals and experts are also available; and our outside directors were regularly available in person or by phone.

UCL&BR: How did you move from outside counsel to chief financial officer and what is in store for you now?

Jensen: Throughout my career I have worked on a variety of finance and commercial matters. I had worked with Cericor on a variety of matters and on its several financings, so I had a pretty good idea what Cericor looked like before I joined the company. HP is a very fine company but probably doesn't need a chief financial officer or general counsel in

its Salt Lake City Division. I was still having fun when the stockholders decided to accept HP's offer. After we closed the deal and I decided to look at other opportunities. I have found that Utah has a growing number of high quality start-ups and that its universities are very cooperative in licensing new technology into growth oriented companies. I am now back working in other venture situations in various capacities and regularly look at a variety of other exciting opportunities here in our community.

UCL&BR: In a world of specialization, is a move such as your exceptional?

Jensen: Perhaps, but I have been fortunate over the years to enjoy a broad range of experiences applicable to the growth of companies. I enjoy new challenges and the excitement of helping to bring together all the various areas of expertise required to help a product make its way to the marketplace.

UCL&BR: Were you disappointed to see the company sold so soon after you joined it?

Jensen: Of course. I had looked forward to a longer relationship with Cericor, but I believe the HP deal was in the best interest of the stockholders. I will miss the close association with the people of Cericor (now HP), but life moves on. I am confident our community will see more successes like this in the future.

UCL&BR: Thank you.

FASB ISSUES FINAL REGULATIONS ON ACCOUNTING FOR INTERNALLY DEVELOPED COMPUTER SOFTWARE

By Michael B. Malan, C.P.A.
Senior Consultant
Deloitte, Haskins and Sells
Salt Lake City, Utah

In August, 1985, the Financial Accounting Standards Board (FASB) published new guidelines that will hopefully end the controversy about accounting for software started in 1969 when IBM first began separately marketing its hardware and software products.

Consistent with prior accounting pronouncements, the statement requires that all research and development costs for developing new software products be expensed as incurred. However, this statement provides definitive project milestones that, when reached, will allow further production costs to be capitalized on the Company's balance sheet as an asset.

Statement of Financial Accounting Standards No. 86: Accounting for the Costs of Computer Software to be Sold, Leased, or Otherwise Marketed states that, "costs incurred internally in creating a computer software product shall be charged to expense when incurred as research and development costs until technological feasibility has been established for the product."

After the technological feasibility of the product has been established, all costs of producing a

product master, including coding and testing costs, should be carried as an asset on the Company's balance sheet and written off over the product's useful life.

"The technological feasibility of a computer software product is established when all planning, designing, coding and testing activities that are necessary to establish that the product can be produced" is completed. At a minimum, the Company must be able to demonstrate that it has completed and tested a detail program design or it has completed a working model of the product. The Statement defines a detail program design as "the detail design of a computer software product that takes product function, feature, and technical requirements to their most detailed, logical form and is ready for coding." A working model is defined as "an operative version of the computer software product that is completed in the same software language as the product to be ultimately marketed, performs all the major functions planned for the product, and is ready for initial customer testing."

The statement becomes effective for all fiscal years beginning after December 15, 1985 and earlier application is permitted.

The new pronouncement may be helpful to computer software companies pursuing additional debt or equity financing. By carrying some software development costs of an internally developed software product as an asset, a Company will have a stronger balance sheet and income statement. In turn, this will make the Company much more attractive to prospective lenders and investors.

UTAH NEWS AND BUSINESS DEVELOPMENTS

EVANS AND SUTHERLAND FINANCIAL UPDATE

Evans and Sutherland Computer Corporation of Salt Lake City, Utah has announced financial results for 1985 and the first quarter of 1986.

Sales for 1985 were $79,588,316 as compared to 1984 sales of $70,116,171. New earnings for 1985 were $6,190,576 ($0.72 per share) as compared to 1984 net earnings of $7,309,118 ($0.83 per share). It should be noted that research and development expenditures for 1985 were $17,480,555, an increase of more than $7,000,000 over 1984 R&D expenses.

Sales during the first quarter (ending March 28) of 1986 were $23,305,000 as compared to $15,591,000 for 1985 first quarter sales. Net earnings for the 1986 first quarter were $1,228,000 as compared to $349,000 for the previous year. R&D expenditures for the first quarter of 1986 were $4,817,000.

Evans & Sutherland designs, builds and sells special-purpose digital computers which are used to create visual images of data bases stored in computer memory. Such systems are used for computer-aided engineering and design (CAE/CAD/CAM), training simulation, engineering simulation, scientific research, education, graphic arts, entertainment, etc. The Company also produces, sells, and services software for such computing systems.

Dr. David C. Evans, President, stated: "The order rate for Simulation products has continued to be very good, with approximately $23 million booked during this first quarter. Our CT (Continuous Tone) line, especially the new CT6, has been very strong, and we are pleased with its success in competitive situations. Another new product, the SPX, has been introduced and is being bid on selected proposals. In addition, NOVOVIEW bookings from civil aviation and military customers continue at a very good rate. In Interactive Systems, sales of the PS 300 products are currently running ahead of last year, although we are facing increasingly strong competition in this very mature product line. Our new product offerings, Romulus-D and our new graphics hardware technology, are well along in development, and initial orders should be seen before the end of the year. This advanced graphics hardware technology will also be marketed by other firms, such as Digital Equipment Corporation, with whom we signed a joint technology agreement in March of this year."

EVANS AND SUTHERLAND ANNOUNCES NEW CONTRACTS WITH GENERAL DYNAMICS AND GOODYEAR AEROSPACE

Evans and Sutherland announced 1985 contracts with the Fort Worth Division of General Dynamics Corporation and Goodyear Aerospace Corporation of Akron, Ohio. The General Dynamics contract is for CT5A and CT6 Computer Image Generation (CIG) Systems and is valued at $12,600,000. The Goodyear contract is for a CT6 Computer Image System and is valued in excess of $2,600,000.

General Dynamics will use the CT5A and CT6 Systems to produce out-the-window imagery and sensory simulation in engineering flight simulators for the development of advanced F16 aircraft as well as next generation tactical fighter aircraft. General Dynamics manufactures the F16, the tactical aircraft used by the U.S. Air Force and NATO. The CT6 from Evans & Sutherland is the latest model in the CT series of computer image generation systems. The CT6 offers new effects and patterns, including a feature called Advanced Surface Texturing. This technology significantly enhances image details to provide a more "real world" appearance.

Goodyear will use the CT6 system to produce electro-optical and infrared sensor imagery in an engineering flight simulator for the development of tactical fighter aircraft simulation systems. The CT6 system is Evans & Sutherland's next generation system, following in the footsteps of its successful predecessor, the CT5/5A.

BRIGHAM YOUNG UNIVERSITY TEACHES ENTIRE COURSE VIA COMPUTER

Brigham Young University has become the first major university to teach an entire college course via computer. Drs. Gordon Stokes and Larry Christiansen, of BYU's Computer Science Department, have been using WICAT Systems hardware and software to teach Computer Science 103, a Pascal programming course, since January 1985. "We had so many students signing up for our introductory computer courses," said Dr. Christiansen, "that we decided to create a computer program to ease the load on teachers. About three years ago, our enrollment for such courses just skyrocketed, jumping from 100 students to 500, then 700 and then 1,000 students. Dr. Stokes explained that the original intent in developing the course was to "use the computer course as a fill-in or part-time substitute for a teacher. But now that we have used it several times, we believe that it is better than a teacher for this type of class, so we use it to teach the entire course."

Although other universities are using computer-assisted instruction at the college level, none of them have used a computer to teach an entire course. "As far as we know, BYU is the only major university that has totally replaced the lecture with a computer as the teacher," Dr. Stockes said. "This course was designed as a curriculum, a two-hour semester course that would replace the normal classroom situation. Other courses, especially languages, use computers as a training tool but not as a teacher."

At the beginning of the course, students are given a ten-minute overview explaining the 25 modules in the course, showing how to begin on their computer terminals, and detailing computer-lab procedures. After that, students get all their instruction from "Elrond" (the name given to the program), though they can get help from a lab assistant, who is always on hand during computer-lab hours. According to the chief lab assistant Gary Stokes, a graduate student in computer science, "The 36 student terminals in the computer lab are connected to two System 220s, each of which runs the Elrond program. During the course, students are required to take four regularly scheduled exams at their terminals in order to keep them from

procrastinating until the last two or three weeks of the course. Elrond automatically grades the exams and records the grades."

Brigham Young University and WICAT Systems plan to jointly market the Pascal course to other universities and colleges in the near future.

DAYNA NOW SHIPPING MacCHARLIE IBM COMPATIBILITY FOR APPLE'S MACINTOSH

Dayna Communications of Salt Lake City is now shipping MacCharlie, a hardware and software product which enables the popular Apple Macintosh personal computer to use software written for the IBM PC. Until now, the Apple Macintosh was limited to software written specifically for the Macintosh since it was not originally designed to be IBM PC compatible.

MacCharlie permits Macintosh users to access the large library of IBM PC-compatible software, to connect to IBM serial networks, to exchange PC and Macintosh data files, and to use IBM peripheral devices such as letter quality printers. It also serves as a printer buffer to the Macintosh. "With MacCharlie, you can have the best of both worlds; the ease of use, portability and innovative features of the Macintosh as well as the extensive software library and expandability of the IBM PC, all in a single, compact system," said Dayna Communications' Chairman, William K. Sadleir. "MacCharlie provides a sudden and complete bridge over the gap between the Macintosh and the IBM PC."

MacCharlie functions as a co-processing device. It consists of two primary components: a keyboard extension that has 10 function keys on the left side and 18 keys on the right side and an expansion unit that physically connects to the right side of the Macintosh, which contains the electronics and the disk drives. In the Macintosh mode, the 18 keys function as a normal Macintosh numeric keypad. In the PC mode, they function as a numeric entry pad and perform cursor control and editing functions as do the corresponding keys on the IBM PC keyboard. The user simply connects MacCharlie to the Macintosh with the provided cables and it is ready to use. Virtually anything a monochrome IBM PC can do, MacCharlie can do.

The unit comes standard with two 5.25-inch, double-sided, double density disk drives with 360 kilobytes (K) of storage and with 640K of random access memory (RAM). MacCharlie causes the Macintosh to act like an IBM PC or standard Macintosh, depending on the mode the user selects.

MacCharlie sells at a suggested retail price of $1,795. The system comes complete with MS DOS 3.1 and GWBASIC. The company is marketing MacCharlie through computer retail stores. "The interest is absolutely incredible, particularly among our large corporate customers," says Thomas A. Guttschow, owner of TZ Computers in Bloomington, Illinois. "MacCharlie is a shoe-in for 1985's Product of the Year. It's certainly a product that will find ample shelf space in our store." Dayna began shipping products in August 1985.

In creating MacCharlie, Dayna's primary motivation was not to cause the Macintosh to behave like the IBM PC, but rather to make the Apple Macintosh more appealing to IBM-biased computer users. "We recognized when Macintosh was introduced in early 1984 that it was technologically superior to the IBM PC. However, potential users, particulary business users, would be reticent to make the purchase decision if it meant abandoning their IBM PC-compatible software and hardware investment. The decision should new be easy. Buy a Macintosh and MacCharlie and use the software that best suits your particular application," Sadleir said.

Computer industry experts seem to agree. Since the computer market is software driven, MacCharlie could well be a primary vehicle for making the Macintosh more accepted in the business community. Businesses are intrigued by the enhanced features of the Apple Macintosh but have been reluctant to buy. Many have invested so much money in IBM PC-compatible software and hardware that, in spite of Macintosh's advantages, they just cannot justify the change. MacCharlie, however, should solve that problem since the IBM PC-compatible software and hardware can be used by Macintosh as long as it is connected to MacCharlie.

Existing Macintosh users should also benefit from MacCharlie. "With MacCharlie, Dayna has virtually announced the availiability of almost 10,000 new software programs for the Macintosh," said Jerry Brereton, former General Manager of Businessland in Salt Lake City, and now Senior Vice President of Marketing and Sales at Dayna. "Even IBM PC-compatible software is enhanced by the combination of Macintosh and MacCharlie since many of the features which exist on Macintosh, such as the desktop utilities and the clipboard, will still work with MacCharlie while in the IBM PC mode," said Brereton.

Dayna Communications is a privately held corporation headquarted in Salt Lake City, Utah. Dayna, through its affiliates, has been in the software business nearly ten years and, among other products, has developed sophisticated management information systems for large financial institutions.

IOMEGA ANNOUNCES PLANS TO BUILD PLANT IN SALT LAKE AREA

Iomega Corporation President, Gabriel P. Fusco, announced plans at a press conference also attended by Gov. Norman Bangerter to build a Salt Lake area plant to manufacture Iomega's Bernoulli Box disk drive subsystem and disk drive cartridges. The plant will be approximately 140,000 square feet in size and will increase the company's engineering and production space by about 90%. The plant is expected to employ 300 to 500 persons by the end of 1986. Iomega currently employes over 1,000 persons at its plants in Roy, Utah.

The Boulder, Colorado area had been under serious consideration for the expansion. In the end, however, the efforts of Gov. Bangerter, the Utah Business & Economic Development Division and others were successful in attracting Iomega to Salt Lake. Iomega sees Utah as a high-tech center of research and development, production and support services.

The Bernoulli Box is a high storage (20 megabytes) disk drive subsystem compatible with the IBM PC, XT, AT, the TI Professional and the Apple Macintosh. The Bernoulli Box is a unique product in that its operation is based upon Bernoulli's Law of fluid dynamics. According to Mr. Fusco, this technology is protected by patents which in turn protect Iomega against competitive inroads which have plaqued most other U.S. disk drive manufacturers.

IOMEGA DISCLOSES 1985 FINANCIAL RESULTS

Iomega Corporation of Roy, Utah achieved 1985 sales of $115,515,000 which is more than double the sales performance of $51,117,000 for 1984. Net income reported for 1985 was $14,894,000 ($1.01 per share) as compared to $2,521,000 ($0.19 per share) for 1984. Research and developement expenditures for 1985 were $6,189,000, an increase of $2,441,000 over 1984.

Gabriel P. Fusco, President, has projected sales in excess of $200,000,000 for 1986.

WICAT SYSTEMS, INC. REPORTS PROFITS FOR ITS THIRD FISCAL QUARTER

WICAT Systems (NASDAQ:WCAT) of Orem, Utah reported results for its third fiscal quarter, ended December 29, 1985. Revenues were $8,849,000 up 26% over the $7,009,000 reported for the same period one year ago. Profit for the quarter was $54,000 or less than one cent per share, compared with the third quarter loss a year earlier of $2,184,000 or eleven cents per share. Revenues for the nine months ended December 29 were $29,681,000 up 48% from $20,067,000 for the nine months a year earlier. The profit for the nine months of $377,000 or two cents per share, compares to a loss of $5,807,000 or twenty-eight cents per share for the nine months last year. The Company's cash reserves increased during the quarter by nearly $2,000,000 to more than $32,000,000. WICAT Systems, Inc. designs, produces, markets and services educational computer systems and curicula for schools and homes, computer-based training systems for industry and government, and general purpose multi-user supermicro computer systems.

WICAT ANNOUNCES PROMOTIONS

The following personnel changes have been announced by WICAT Systems during the first quarter of 1986:

1. Derin Ford to Manager of Sales Administration.
2. Joe Staples to Director of Quality.
3. Mark Nelson to Director of Sales Support.
4. Jerry Geiger to Vice President of Manufacturing.
5. Terry Young to National Depot Manager.
6. Don Stephens to Production Manager, Manufacturing Division.
7. Dan Hostetter to Production and Inventory Control Manager, Manufacturing Division.

8. Dr. Peter G. Fairweather to Corporate Product Manager for Education and Training.

9. Bob Larsen to Manager of WISE Runtime Software Development Group.

SPIELBERG, HARVARD AND WICAT SYSTEMS JOIN IN SEARCH FOR E.T.

It sounds like another episode of "Amazing Stories," but this one is true. Hollywood director Steven Spielberg ("Close Encounters" and "E.T."), Harvard University and WICAT Systems are partners in a search for the real E.T.

The search began in March of 1983, when Paul Horowitz, an astronomer at Harvard University, hooked up a WICAT Systems computer to a giant radio telescope to listen for alien signals. Grants from Carl Sagan and the Planetary Society kept the project going, but a recent donation of $100,000 from Spielberg has turned it into the world's biggest search for extraterrestrial life. "Its's the most comprehensive search that I know of," Horowitz said. "By the fall of 1986 we will have finished a complete scan of the sky."

Called Project META (Megachannel Extra-Terrestrial Assay), the search uses an 84-foot diameter radio telescope to scan the sky for distant radio signals and evidence of intelligent life in outer space. Spielberg's grant allowed Horowitz, the project's developer, to expand the search to 8.4 million radio frequencies. Previously, the telescope could scan only 128,000 channels.

The upgraded antenna can now reach the outer edges of the Milky Way Galaxy, some 80,000 light years away from Harvard's Oak Ridge Observatory, where the search continues around the clock. The stationary dish works as the Earth's rotation sweeps it around the sky. The antenna automatically adjusts itself one-half degree every 24 hours so that it sweeps in a 100-degree-wide band around the sky every 200 days.

The radio telescope, located about 30 miles northwest of Boston, methodically "listens" to millions of radio frequencies while a computerized spectral analyzer, developed by Horowitz, examines the channels for anything unusual. Spielberg's donation helped Horowitz build the analyzer, which uses 144 processors, each with a 68000 CPU running at 10 MHz. Each processor is dedicated to a specific range of frequencies. "The 144 independent processors are all essentially slaves to a WICAT System 150," Horowitz said. "It's the same old machine, the unmapped single-user 150 we've been using since March of 1983." The Harvard astronomer is using two System 150s with Project META. One is used to control the real-time acquisition and storage of data and to adjust the frequency of the receivers. The other is used to look at the data from the on-line machine and develop software for the project. Horowitz praised the performance of the WICAT Systems computer. "It's worked 100 percent of the time, 24 hours a day," he said. "It has worked remarkably reliably."

The upgraded Project META was initiated September 29, 1985 at a ceremony attended by Spielberg, Sagan and Horowitz at the observatory. It costs

about $25,000 a year to run the operation, and funding comes from outside sources such as the Planetary Society and private donations. Although the search hasn't picked up any extraterrestrial signals yet, Horowitz said there is a high probability that intelligent life exists elsewhere in our galaxy, and that it is within "hearing distance" of Project META. "Within 50 light years of Earth, there are 150 stars similar to our sun. Within 1,000 light years of Earth, there are a million sunlike stars," he said. But that doesn't mean he'll be hearing from E.T. anytime soon. "It could take years, even decades, for a message to reach us. That's why the search is automatically monitored," he said.

$10,000,000 CONTRACT BETWEEN WICAT AND SYSCON CORPORATION FOR U.S. NAVY

WICAT Systems, Inc. has announced that it will be providing a large quantity of its computer systems to SYSCON Corporation as part of a production contract recently won by SYSCON to provide NAVTAG Training Systems to the U.S. Navy. In addition to the systems, WICAT will provide support and maintenance service. The total agreement has an estimated value in excess of ten million dollars, according to company officials.

The Naval Training Equipment Center commissioned SYSCON to develop computer-based strategic "war games" for training tactical officers on-board naval vessels. The resulting Naval Technical Game (NAVTAG) Training System enables officers to give sensor, maneuvering, or weapon employment commands in hypothetical, life-like situations.

The U.S. Navy chose the WICAT System 150WS as the microcomputer delivery system for NAVTAG. The 150's mass disk storage will support the prompted, interactive program where the "game" director chooses a scenario already stored in the system or creates one of his own. He then monitors play from the 150's integrated CRT while the players interact though two connected video terminals. The results of their tactical decisions are reflected in a graphic screen display over one minute intervals of real time. The result is a sophisticated training system that closely resembles actual combat engagements.

NOVELL RELEASES 1985 FINANCIAL DATA AND ANNOUNCES SUCCESS OF $20,000,000 STOCK OFFERING

Financial statements released by Novell, Inc. of Orem, Utah for fiscal year 1985 (ending October 26) show sales of $33,618,000 and income of $4,159,000 ($0.41 per share). These figures are up impressively from the previous year in which Novell reported sales of $10,862,000 and income of $585,000 ($0.06 per share). Novell designs, manufactures and markets a family of local area network (LAN) and communication products which allow personal computers to share files, data and resources. Novell's operating results for the first quarter ended January 25, 1986 have also been announced. Sales in the first

quarter of fiscal 1985 were $14,472,000 compared with $4,821,000 in the same period in fiscal 1985, an increase of 200%. Novell's income before extraordinary credit in the 1986 first quarter increased 329% to $2,071,000, compared with $483,000 one year ago. Earnings per share in the current quarter were $0.20, versus $0.05 for the first quarter of fiscal 1985.

Raymond H. Noorda, President of Novell, stated, "Sales of both LAN systems and LAN software were strong in the quarter. Novell's sales increased substantially through all major channels of distribution. Novell distributed product through five channels: 1. Distributors. 2. Original equipment manufacturers. 3. Computer retail chains. 4. Value added resellers. 5. Directly to end-users."

In February 1986, Novell completed the sale of 1,192,000 shares of its Common Stock to the public. Net proceeds of the offering, approximately $20 million, were used to repay all of Novell's debt and to increase working capital.

NOVELL, 3COM ANNOUNCE AGREEMENT TO IMPROVE ETHERNET PERFORMANCE

Novell, Inc., a personal computer networking and data communications company, has announced a marketing agreement with 3Com Corporation, under which Novell will sell the EtherLink Plus network adapter board designed by 3COM with Novell's own Advanced NewWare/286 operating system. Novell's reason for offering Advanced NetWare/286 with the EtherLink Plus board is to provide a more efficient, higher performance network for Ethernet users. "With Advanced NetWare/286 in combination with the EtherLink Plus board," explained Craig Burton, vice president of marketing for Novell, "we believe that we are providing the best possible package for the Ethernet niche. As the premier operating system company, we feel that it is an important step to team up the most popular Ethernet hardware with the most sophisticated, widely used network operating system available."

Burton emphasized the significance of the agreement. "Novell and 3Com have worked together to provide this product combination. The EtherLink Plus board is much more efficient and provides better network performance than the original EtherLink board. End users who want the strongest performance will benefit by using the new board in the file server, and even more by using the Novell Server 286A rather than an IBM PC AT." The EtherLink Plus board is designed for use with an IBM PC AT or compatible server, added Burton.

Novell will begin shipping NetWare/E+ in all its configurations during the first quarter of 1986. The board and operating system together will sell for $2,495.

SATELLITE SOFTWARE INTERNATIONAL REPORTS GROWTH IN 1985

Satellite Software International of Orem, Utah, well known for its Word Perfect software product, reported sales of $23,000,000 for 1985, up from $9,000,000 in 1984. The company also reports a 20% increase in earnings. Strong sales of Word Perfect 4.1 and a site licensing plan have been credited as the basis for this strong growth.

NEW PRODUCT ANNOUNCEMENTS

LOW COST TERMINAL NOW AVAILABLE FROM WICAT

WICAT Systems is now offering its customers a cost-competitive computer terminal, the T7100. The T7100 terminal replaces the T7000 terminal. The new terminal can be used with the following WICAT Systems supermicro computers: System 1250, System 1255, System 1260 and System 2220. Most WICAT Systems software applications, with a minimum number of modifications, will run on the T7100 terminal. Graphics applications will run on the MG8000 and WIT terminals, which the company will continue to produce.

The T7100 is an ANSI-compatible, desktop video display terminal that is capable of interfacing with a variety of computer systems and peripheral devices. The terminal consists of a keyboard and display unit. The keyboard attaches to the display unit via a coiled cable extending to a maximum of six feet. It has a step-sculptured profile with an adjustable slope of 7, 11 and 15 degrees. Sixteen programmable function keys are available for sending user-developed or default codes to the screen or the host computer. Forms or command sequences are stored in a 256-byte nonvolatile memory. It also has a separate numeric keypad, separate editing keys, and four PF (general purpose function) keys.

The display unit features a 14-in. viewing screen that is seated on a pedestal which allows the unit to tilt and swivel. The screen can display 24 horizontal lines of 80 or 132 characters. In addition, a programmable user line on the last row of the screen and a status line on the first row of the screen may be selected for display by the user.

A unique feature of the T7100 is the CRT saver. After about 10 minutes, if no information is sent to the terminal by either the user or host computer, all displayed information (except for the user line) disappears from the viewing screen. It reappears as soon as a key is pressed or a prompt is received from the host computer. This feature prolongs the operating life of the video display.

Other features of the display unit are extended page mode, selectable scrolling rates, spit screen, auto-paging, and double-high/double-wide character display.

IBM ANNOUNCES NEW EDUCATIONAL SOFTWARE PRODUCTS, INCLUDING PROGRAMS DEVELOPED BY WICAT

International Business Machines Corporation (IBM) has announced thirty-five educational software products for use on the IBM PC family, twenty-five of which were developed by WICAT Systems of Orem, Utah. The WICAT Systems products offer individualized instruction in the basic skill areas for grades kindergarten through eight. Courses in reading, language arts and math have been designed to run on the entire family of IBM Personal Computers. "WICAT Systems is delighted to have IBM marketing our PC courseware products," said Dr. Dustin H. Heuston, Chariman of WICAT Systems. "We believe the announced curricula will not

only be used by schools but in the future may be a significant product for use in the home." The announced software products were developed over the past three years as part of a $15 million Limited R&D Partnership for which WICAT Systems serves as general partner.

WICAT AND TWA DEVELOP TRAINING SYSTEM FOR PILOTS

WICAT Systems and Trans World Airlines (TWA) have completed joint development of a computer-simulated training system for new DC-9 pilots. The course covers operating procedures for a Flight Guidance System (autopilot) and includes demonstrations and practice exercises. Expenses of developing the training package were shared by WICAT Systems and TWA, according to Andy Gibbons, WICAT Systems' development consultant for the project.

WICAT Systems will market the training package to DC-9 users worldwide. TWA is already using the course and expects it "to pay for itself within less than a year," according to Glen Hersch, TWA's Director of Operations Training, who coordinated the development project. McDonnell-Douglas Aircraft plans to use the system for training pilots of Scandanavian Airways Systems (SAS).

According to Hersch, trainees use the WICAT Systems computer-based course after they receive two hours of classroom instruction. Following the course, trainees must fly a full-motion platform flight simulator, or simulated cockpit. The experience gained through the computer-based course reduces the time the trainees must spend in the simulated cockpit. This reduces training costs because operating the simulated cockpit it expensive.

"Although similar pilot-training systems are available from other vendors," said Gibbons, "ours is the only one created by nonprogrammers. The WISE (WICAT's Interactive System for Education) authoring program allows us to created the program for the training package without programmers--at about half the cost of employing programmers to do so. Furthermore, it took us less than half the time it would have taken a team of programmers--in all, approximately eight months. This makes our package very cost competitive."

The hardware for the training package includes a System 155; a WIT (WICAT Interactive Terminal) with touchscreen capability, color, and graphics; and a videodisc player.

IOMEGA'S HALF-HEIGHT NETWORK SERVER ADDS HIGH PERFORMANCE DATA-STORAGE CAPABILITIES TO THE MACINTOSH OFFICE

Answering the data-storage limitations of the AppleTalk Network™, IOMEGA Corporation has announced the availability of its 20-megabyte half-height AppleTalk Network Server, an add-on device specially designed to supplement the Macintosh Office™ from Apple Computer, Inc.

The AppleTalk Network Server can back up data at a rate of 1-megabyte-per-minute onto rugged 10-megabyte cartridges. The cartridge can be removed from the drive and locked away for security or transported between different

locations, even by mail. In addition, the network server's storage capacity is virtually unlimited. By adding cartridges in 10-megabyte increments, users can affordably expand their data-storage system to meet their growing needs.

The system can accommodate as many as 20 Macintosh computers at one time, allowing each user 1-megabyte of on-line, password-protected data-storage capacity.

"To be effective, a local area network requires a specialized data-storage system," explained Scott D. McVay, vice president of marketing for Iomega. "By applying our proven Bernoulli technology, we can now bring the power of data independence to the Macintosh Office."

Based on the principle of fluid dynamics, first articulated by 18th century mathematician Daniel Bernoulli, the Bernoulli Box takes the best of both floppy and hard disk technology, combining Winchester drive performance with cost-effective removable-media features.

The 20-megabyte AppleTalk Network Server includes two 10-megabyte cartridges, AppleTalk support software and Iomega utilities, such as whole disk copying, partitioning, password protection and formatting. The suggested retail price is $3,750.

ADVANCED NETWARE 2.0: ENHANCED OPERATING SYSTEM SOFTWARE FROM NOVELL

Novell, Inc., announces the release of Advanced NetWare 2.0. This enhanced operating system software offers several powerful features: DOS 3.1 compatibility, enhanced user utilities, a NetWare tutorial, an Auto-Spool feature, internal and external bridging capabilities, generic operating system software, network diagnostics software, NETBIOS emulation, and an internetwork communication protocol.

According to Craig Burton, Novell's vice president of marketing, Novell's release of Advanced NetWare 2.0 "marks several major breakthroughs in networking software operating system technology. This new product marks the sixth iteration of major file server release software from Novell. No other company has the experience and technology base that these releases represent."

The feature of the new operating system are designed to simplify networking for the end user.

RECENT LEGAL DEVELOPMENTS OF IMPORTANCE AND INTEREST

SOFTWARE PIRACY SUIT SETTLED

The Association of Data Processing Service Organizations, Inc. (ADAPSO) and MicroPro International reached a settlement last year with American Brands, Inc., and its subsidiary, Wilson Jones Company, in a software copyright infringement suit filed by ADAPSO and MicroPro last January. The suit alleged unauthorized copying of three MicroPro programs marketed under the trademarks MailMerge, SpellStar and WordStar. In a joint statement issued by the parties, American Brands and Wilson Jones agreed that the MicroPro programs had been "improperly duplicated by some Wilson Jones personnel." The defendants further agreed to institute "appropriate procedures. . .to ensure against such improper duplication in the future".

Editor's Comments: Settlement of the MicroPro suit was a very practical and reasonable resolution to this dispute, but a disappointment to those computer lawyers, including myself, who were looking forward to a judicial resolution of the copyright and contract issues involved in the case. Such issues include the scope of fair use, backup and adaption rights of the owner of a copy of a computer program and the enforceability of "shrink-wrap" license contracts. The lesson to be learned from the MicroPro suit is that companies who fail to take measures to prevent unauthorized copying by employees are vulnerable to copyright infringement suits. The problem, without doubt, is wide spread, as many employees do not hesitate to make unauthorized copies of licensed software. The best way for a company to protect itself against a costly and embarrassing suit is to require all of its employees having access to computer software to read and sign a policy statement on software use. The company must also enforce the policy statement. A copy of a sample policy statement suggested by ADAPSO can be obtained from the Utah Computer Law and Business Report by contacting me at the address or telephone number given on page 22. Software developers and vendors should encourage their customers to adopt and use such policy statements.

OWNER OF INFRINGED SOFTWARE COPYRIGHT DENIED RECOVERY OF ATTORNEYS FEES

In the case of Whelan Associates, Inc. v. Jaslow Dental Lab, a federal district court found that Jaslow had willfully and intentionally infringed a computer software copyright of Whelen. The court originally ordered Jaslow to pay Whelan its attorneys' fees. Upon reconsideration, however, the court held that it could not award attorneys' fees to Whelan because the infringement of the copyright began prior to the effective date of the registration of the copyright. Section 412 of the Copyright Act prohibits the award of attorneys fees under such circumstances.

Editor's Comments: This case illustrates one reason why any company which develops or markets software should promptly register its copyrights with the United States Copyright Office. If a copyright is not registered before the infringement begins, the copyright owner will not be allowed to recover its attorneys' fees from the infringer. The legal fees incurred in a copyright infringement suit can be substantial. It is also noted that the Copyright Act does not allow a

court to award "statutory damages" in lieu of actual damages and infringer's "profits" if the copyright was not registered before the infringement began. Actual damages and infringer's profits are difficult to prove. This difficulty often means that "statutory damages" (which require no proof and are awarded at the discretion of the court) represent the copyright owners only means for recovery of a monetary award. Prompt registration of all software copyrights is an essential cornerstone to an effective and successful plan for the protection of software.

SOFTWARE PIRATE FOUND GUILTY OF CRIMINAL VIOLATIONS OF COPYRIGHT ACT

A San Francisco jury has found Thomas Taylor guilty in a federal criminal action brought against Taylor for copying and distributing computer software and manuals belonging to Stenograph Corporation. Taylor faces up to four years in prison and a $50,000 fine.

Editor's Comments: When faced with a blatant case of software piracy, the software owner should always give consideration to filing criminal claims with federal prosecutors.

DEC DENIED PRELIMINARY INJUNCTION IN VIDEO DISPLAY TERMINAL CASE

Digital Equipment Corporation (DEC) was denied preliminary injuctive relief in its suit against C. Itoh and Co., Ltd. DEC sought a court order prohibiting C. Itoh from duplicating many of the design features of DEC's VT220 terminal. Such duplicated features include keyboard design (placement, arrangement and spacing of keys, shape, dimension and color scheme) and wedge-shaped monitor. Even though the terminal design was original with DEC and obviously copied by C. Itoh, the court held that the design features were unprotectable. Many of the features were deemed to be functional and therefore unprotectable under the trademark and unfair competition laws. These functional features were not patented by DEC. The nonfunctional features had not attained secondary meaning and therefore were also unprotected under the trademark and unfair competition laws. Secondary meaning is attained after extensive and substantially exclusive use sufficient to establish in the minds of the relevant purchasing public an association between the feature and a single source for the goods (i.e., the feature must distinguish the goods from goods sold by all others).

Editor's Comments: An important factor in the cout's decision that many of the features were functional and therefore unprotectable was DEC's own advertisements which placed great emphasis on the functionability of the features. A company's advertising strategies should be in harmony with its trademark protection policies. This is simply a matter of communication between advertising/marketing personnel and the company's trademark attorney. It should also be noted that for many products with unique design features an aggressive policy to procure design patents may prove to be an effective means of protecting those features against copying.

ADAPSO PROPOSAL FOR SOFTWARE PROTECTION PASSES MUSTER WITH JUSTICE DEPARTMENT

A proposal for a software protection system made by the Association of Data Processing Service Organizations (ADAPSO) has been reviewed by the U.S. department of Justice. A business review letter issued by the Department concluded that the system proposed by ADAPSO would not be in violation of U.S. antitrust laws because the system would be an unreasonable restraint of trade. The Department recognized that the software protection system has the potential of fostering, rather than restricting, competition.

ADAPSO is an organization comprised of software developers. The software protection system proposed by ADAPSO consists of a key, a key ring and a lock. The "lock" is made a part of the software. The "key ring" plugs into the computer. The three components work together to protect the software against unauthorized use. Software vendors would receive hardware and software serial numbers through a clearinghouse.

COMPUTER PROGRAM COPYRIGHT CASES PRODUCE DIFFERENT RESULTS

Two recently decided cases illustrate the uncertainty, imprecision and inconsistency which shroud the resolution of computer program copyright infringement claims. The cases were brought by different owners of different copyrighted computer programs. In each case the copyright owner sought a preliminary injunction against the alleged copyright infringer. In one case, brought before a Minnesota federal district court, a preliminary injunction was granted. In the other case, brought before a New York federal district court, a preliminary injunction was denied. A preliminary injunction is a court order issued prior to the trial. A preliminary injunction will be issued only if the party seeking the injunction can establish a likelihood of success on the merits and irreparable injury if relief is delayed until after trial. In computer program infringement cases, the injunction sought is typically an order prohibiting the infringer from further production, sale or distribution of the infringing program.

In the Minnesota case, the E.F. Johnson Co. convinced the court that Uniden Corporation of America had infringed certain copyrighted computer programs embodied in an Intel 8049 microprocessor chip which was the heart of Johnson's "clear channel LTR" communications systems product. Uniden engineers produced an LTR-compatible product, the Uniden FTS-250T, by disassemblying Johnson's computer program and preparing flow charts which were used to create a computer program for the Hitachi microprocessor employed in the Uniden FTS-250T. "Substantial similarity" is an essential element of copyright infringement. The court used an "interative" approach to resolve the issue. Under this approach there is infringement if the alleged infringer used the copyrighted program in preparing the infringing program and if the alleged infringer's program was produced by duplication of substantial portions of the copyrighted program. The analysis is accomplished by looking for specific quantitative and qualitative similarities between the two programs.

Guided by expert testimony and other evidence the court found many similarities between the programs. Johnson's LTR "sample error table" was duplicated in the Uniden program. Both programs employed a 56 bit sampling technique even though Uniden's Hitachi microprocessor could have sampled bits at a much high rate. Superfluous code in Johnson's LTR program was included in the Uniden program. Many LTR subroutines were duplicated in the Uniden program. Uniden's manuals also included portions that were copied verbatim form Johnson's manuals. This was enough to convince the court that the programs were "substantially similar" even though a line-by-line comparison of the Johnson LTR code (translated into Hitachi language) to the Uniden code showed that the programs were not identical.

In the New York case, Q-Co. Industries, Inc. brought suit against two former employees, Sidney Hoffman and Dilip Som, who had developed for Q-Co. a prompter computer program (the "VSP-500") for the Atari 800-XL microcomputer. A prompter is a device which scrolls text on a television screen for the benefit of a speaker or performer. Suit was filed after Hoffman and Som left Q-Co. and formed a new company to develop a similar prompter computer program (the "CPC-1000") for the IBM PC microcomputer. Notwithstanding many structural and design similarities between the VSP-500 program and the CPC-1000 program, the court held that there could be no copyright infringement because VSP-500 was written in Basic and Atari assembler whereas CPC-1000 was written in Pascal and IBM assembler. The court viewed the similarities as similar "ideas" rather than similar "expression." The distinction is important because ideas are not protectivible under copyright law. The court did not appear to use the approach or analysis used in the Johnson case. Under the Johnson case, the structural and design similarities between VSP-500 and CPC-1000 might have supported the copyright infringement claim.

Even though the Q-Co. court was not receptive to the copyright infringement claim, it did find merit in the allegation that trade secrets had been misappropriated by Hoffman and Som. This is an example of a case where trade secret protection is broader than copyright protection. Companies which develop software (as well as other intellectual property) should be certain that they have in place a trade secret program (including company policies and employee agreements) which maximizes protection under the trade secret laws. The court probably would have granted a preliminary injunction in view of the trade secret claim had it not been for the fact that the CPC-1000 program was not yet ready for commerical distribution and, therefore, Q-Co. was not yet facing immediate irreparable harm.

THE COMPUTER LAW CORNER

"WHY MANY COMPANIES MAY NOT OWN 'THEIR' SOFTWARE"

by

Jon C. Christiansen
B.S. Engineering, J.D. Law

XYZ, Inc. undertakes the development of a new 32 bit supermicrocomputer. XYZ determines that it is desirable from both a technical and marketing perspective to develop a new operating system for its computer. Lacking experience and expertise in the development of operating systems, XYZ engages the services of Messrs. A, B and C as independent contractors to develop and code the operating system. After an eighteen month development period and after the payment of a substantial sum of money to A, B and C, version 1.0 of the operating system is completed. TUV, Inc., a competitor of XYZ, subsequently undertakes to develop a clone of the XYZ computer (i.e. an XYZ-compatible computer). TUV engages the services of Messrs. A, B and C to provide TUV with an operating system. Within ten days an operating system virtually indentical to the one developed for XYZ is delivered to TUV. Does XYZ have a valid claim against A, B and C or TUV?

The answer is probably no! Because the operating system consists of computer programs, it is a copyrightable work. The copyright to this or any other work is owned by the author. XYZ would be deemed the author only if the operating system qualified under Section 101 of the Copyright Act as a "work made for hire." There are two categories of works made for hire: (1) works made by employees and (2) works specially ordered or commissioned. The first category requires that the work be made by an employee within the scope of his employment. If A, B and C were employees of XYZ and if the operating system were developed within the scope of their employment, the operating system would be a work made for hire and XYZ would be the author of the operating system (and therefore the owner of the copyright). A, B and C, however, were not employees of XYZ and therefore the first category is inapplicable. A specially ordered or commissioned work is a work made for hire under the second category only if the parties agree in writing that the work will be considered a work made for hire and even then only if the work is to be used "as a collective work, as part of a motion picture or other audiovisual work, as a translation, as a supplementary work, as a compilation, as an instructional text, as a test, as answer material for a test, or as an atlas." There was no written agreement by A, B and C that the operating system was a work made for hire. Furthermore, even if there were such a written agreement the operating system would still not be a work made for hire because it does not fit within any of the types of works identified above.

Because the operating system was not a work made for hire, A, B and C jointly own the copyright. Under the circumstances, XYZ would almost certainly be deemed to hold an implied nonexclusive license to reproduce and distribute the operating system in connection with the new XYZ computer. XYZ would own the copyright only if A, B and C were to assign in writing the copyright to XYZ.

Being neither the owner of the copyright nor an exclusive licensee under the copyright, XYZ has no claim against A, B and C or TUV for copyright infringement. XYZ may try to improve its position by asserting that there was an implied exclusive license from A, B and C or that there was an implied contractual obligation on the part of A, B and C to assign the copyright to XYZ. XYZ chances of success on such issues would be highly uncertain at best and would depend upon the specific facts of the case.

The above discussion is directed only to copyright issues. The operating system programs almost certainly embody trade secrets or were developed with the use of trade secrets. Trade secrets are intangible property which are owned by the individuals who create the trade secrets. Ownership may be transferred by assignment. A, B and C did not expressly assign or agree to assign these trade secrets to XYZ. To prevail on this issue, XYZ would have to establish that, under the circumstances, there was an implied contractual obligation on the part of A, B and C to assign their trade secrets. At best, substantial uncertainty clouds the resolution of this issue. A, B and C may have brought many trade secrets with them to the XYZ project. For example, A, B and C may have used previously developed subroutines or algorithms in the operating system code. Such pre-existing trade secrets almost certainly remain the property of A, B and C in the absence of a clear contractual obligation of assignment to XYZ. XYZ would have a stronger (but still uncertain) claim with respect to trade secrets (e.g. subroutines and code) developed during the XYZ project. Very possibly, XYZ would be left with no more than an implied, nonexclusive license to use the trade secrets as embodied in the operating system.

It is possible, but not probable, that the operating system is an embodiment of one or more patentable inventions. Similar to trade secrets, patent rights are held by the inventors unless and until assigned by them. A, B and C would retain ownership of any patent rights because XYZ failed to contractually protect its interests.

The bottom line is that whenever a company contracts for the development of computer programs it must be certain that copyright, trade secret and patent ownership and license issues are satisfactorily resolved in a written agreement. The party performing the development should also ensure that the written agreement resolves these issues in a satisfactory manner. Leaving these issues unresolved is to invite uncertainty, dispute and litigation. The foregoing also applies to joint projects for the development, enhancement or integration of computer programs. In my experience, I have repeatedly found that the absence of a written agreement (or worse, the use of an inadequate written agreement) to properly resolve these and many other related issues results in serious problems which are resolved later only with great difficulty or at great cost or not at all.

5716C

UTAH COMPUTER LAW AND BUSINESS REPORT

MAILING LIST

To be placed on the mailing list for the Utah Computer Law and Business Report, complete the form below (or a photocopy thereof) and mail it to Jennifer Jensen, c/o Van Cott, Bagley, Cornwall & McCarthy, P.O. Box 45340, Salt Lake City, Utah 84145. There is no charge or fee.

- -

Mailing List Application Form (UCL&BR)
(please type or print clearly)

NAME: _____
ADDRESS: _____
_____
TELEPHONE: _____
PROFESSION: _____
COMPANY: _____
POSITION WITH COMPANY: _____

PERSONAL COMPUTER EXPERT SYSTEM FOR ELASTOMER SELECTION

E. H. Schmauch, W. H. Thomason, and T. L. Pugh
Conoco Inc.
P. O. Box 1267
Ponca City, OK 74603

ABSTRACT

Proper selection of elastomers for application in oilfield environments can be very difficult because of the great variation in physical and chemical conditions that are encountered. In an effort to better utilize existing elastomer expertise, a personal computer expert system to aid in elastomer selection is under development at Conoco. Expert systems are a form of artificial intelligence which imitate the decision-making processes of experts. This elastomer expert system will enable an expert to share his knowledge and skill with nonexperts via a personal computer. This paper discusses the expert system development and its main features, and provides examples of its use.

ELASTOMER SELECTION FOR OIL FIELD APPLICATIONS IS DIFFICULT

The premature failure of elastomer seals in production equipment can be expensive and hazardous. In order to assist Conoco's operating divisions in making the optimum initial selection of an elastomer seal, we have conducted during the past few years a myriad of tests to determine the chemical compatibility of many commercial elastomer materials with various oil field environments and also acquired similar test data from the literature and contract research. Because of the great variation in physical and chemical conditions that are encountered in oil field environments, the efficient utilization of this data to identify an elastomer that will provide good performance in a specific environment can be very difficult.

Effective elastomer seals in oil field service must perform over large temperature and pressure fluctuations and resist degradation by solvents, acids, gases, corrosion inhibitors, biocides, scale inhibitors, and other chemicals. In many applications, an elastomer may be exposed to several treating chemicals in addition to the produced fluids. Further, variations in temperature and/or chemical concentrations may change the environment from one where a given elastomer performs well to an environment in which the elastomer will fail prematurely.

In an effort to better utilize our elastomer data base for the selection of an optimum elastomer product for a specific oil field environment, we are developing a computer expert system that will aid in elastomer selection. A portion of our present data was converted into a "rule base" for use by the expert system. This initial effort which is described in this paper has produced very favorable results, and we plan to incorporate our total and future elastomer data base into this system.

INTRODUCTION TO EXPERT SYSTEMS

We are all aware of the value of a human expert. The ability to permanently capture his expertise in the form of computer software is potentially of equal value. The goal of an expert system is to enable a user to obtain a solution to a problem through an interactive session with a computer. The computer expert system first asks the user questions about his problem, similar to the questions a human expert would ask during a consultation. An example of a portion of the queries along with the user's responses for the elastomers expert system is shown in Figure 1. When the expert system has received sufficient input to draw conclusions, the conclusions are displayed on the computer monitor, as is shown in Figure 2.

In general, the greater the similarity between a computer session with an expert system and a person-to-person session with a human expert, the more useful an expert system becomes. Many expert systems allow the user to inquire why a certain question is asked. The expert system will respond by explaining its current line of reasoning. Another feature which makes the computer session more friendly is outputting deductions and/or partial conclusions at various points through the session. This is equivalent to a human expert saying, for example, "With what you have told me so far, we can narrow it down to these three elastomers: Viton GF, Viton A, and Aflas GT-787." The expert system then continues until it reaches its final conclusions. Both the why function and the outputting of deductions and partial conclusions help make the computer session with the expert system more closely resemble a conversation with a human expert.

Expert systems have been utilized in a number of disciplines,[1,2] including oil well drilling, mineral exploration, diesel locomotive repairs, medical diagnosis and treatment, chemical analysis, organic chemical synthesis, and computer system configuration. Development of specialized, limited-in-scope expert systems has been the most successful, while development of more generalized, all-encompassing expert systems has been more difficult.[3] Therefore, although personal computers may be limited to smaller, specialized expert systems, it is these expert systems that are often the most useful.

An expert system is distinguished from a traditional computer program by its separation of the control logic from the problem logic.[4] Control logic refers to the programming which controls the sequence of program execution. The problem logic contains the data and information which is specific to a given problem. It is possible to write a traditional computer program which would act as an expert in a certain field, but the intermingling of the control logic and problem logic would make the program very difficult to develop and maintain. In an expert system, the control logic is referred to as the inference engine and the problem logic as the rule base. The rule base contains the expertise. The inference engine makes logical deductions based on the expertise in the rule base. The same inference engine is usable in any number of expert systems, while the rule base must be developed for each different expert system application.

The main components of an operational expert system are the rule base, inference engine, and user interface. The required interactions between these components are illustrated in Figure 3. Also illustrated is the rule compiler which is required for the development of an expert system.

The rule compiler converts rules to a usable rule base. The human expert (or experts) must express his expertise as a series of IF-THEN rules. An example of an IF-THEN rule from an expert system for corrosion failure analysis is shown in Figure 4. An IF-THEN rule states that, if a certain statement or group of statements is true, then a second statement or group of statements is also true. The statements in the IF portion of a rule are referred to as antecedents and those in the THEN portion as consequents.[5] A typical expert system contains hundreds of rules. The rule compiler translates the rules from a form that humans understand to a form that the inference engine will understand. The compiled rules make up the rule base. The role of a rule compiler in expert systems is analogous to the role of a FORTRAN compiler in traditional programming, translating source code that is understandable by humans to executable code that is understandable by the computer hardware.

The inference engine is the heart of an operational expert system. It communicates with the user through the user interface and makes logical deductions based on the rule base. Utilizing the expertise expressed in the rule base, the inference engine determines which questions to ask the user. Combining the user's responses with the expertise of the rule base, the inference engine makes logical deductions and eventually draws conclusions. To extend the analogy, the inference engine's role in an expert system is the role of the machine hardware in traditional programming. In the future, hardware implementations of inference engines are likely to be developed which will greatly enhance the power of expert systems. Presently, inference engines are implemented in software. It is important to note that the inference engine does not operate on the rule base in a sequential manner but calls upon pieces of expertise as they are needed. The inference engine can store and recall facts as it searches for the problem's solution. This keeps it from asking the same question or performing the same logical deductions more than once. The facts are analogous to variables in traditional programming. The inference engine queries the user through the user interface. An interaction with the user interface should be as similar to a conversation with a human expert as practically possible.

The goal of an inference engine is to prove or disprove hypotheses in the expert system's domain of expertise. For example, in our elastomer expert system, the hypotheses are that each elastomer in the rule base can withstand the environment of interest. Any hypothesis that the expert system proves to be true corresponds to an elastomer that can withstand the given environment. A hypothesis is expressed as a consequent in one or more of the rules in the rule base. Not all consequents are hypotheses; some are only intermediate facts used in proving or disproving one or more hypotheses. Any consequent which is not a hypothesis will be an antecedent in another rule. An inference engine proves and disproves hypotheses by one of two alternative control strategies.[5] The first is forward chaining or antecedent reasoning. The inference engine starts with the antecedents in the rule base and, by asking questions and making deductions, proves or disproves the various hypotheses. In backward chaining or consequent reasoning, the inference engine starts with the hypotheses. It scans the rule base for

rules with the first hypothesis as a consequent. Each antecedent in a rule with the hypothesis as its consequent is considered a subgoal. The inference engine continues to recursively[6] set subgoals until it has an antecedent as a subgoal that is not a consequent for any rule. Then the user is queried about this antecedent. The inference engine continues in this manner until the hypothesis is proven or disproven. Then it proceeds to the next hypothesis and continues until all have been considered. The goal of some inference engines is to find a single true hypothesis and, in this case, execution stops after one hypothesis is proven. Some inference engines combine antecedent and consequent reasoning into a sophisticated mixed control strategy.

EXPERT-3

In our laboratory, we have written EXPERT-3, an expert system development tool for the IBM PC and compatibles. EXPERT-3 is an enhancement of EXPERT-2 by Jack Park[7] and is written in the computer language FORTH.[8] The inference engine uses the backward chaining control strategy. Table 1 summarizes the valid user responses for EXPERT-3. Expert system development systems for the PC are commercially available, but we chose to develop our own so we could modify the rule compiler, inference engine, and user interface to fine-tune them for the types of applications we are interested in--namely, materials selection.

Each EXPERT-3 rule contains one or more antecedents and one or more consequents. There are three types of antecedents. The first is the standard IF clause common to most expert systems, as is shown in Figure 4. If the user must be queried about this antecedent, possible responses are "yes," "no," or "I don't know." The second type of antecedents allows comparisons of a variable--for example, temperature--as is shown in Figure 5. Supported comparisons are equal (EQ), not equal (NE), less than (LT), and greater than (GT). The user will be queried about the variable, and the response can be either an integer or "I don't know," as is demonstrated in Figure 6. The third type of antecedent allows facts to be categorized. In our applications, the elastomer might be exposed to one or more members of a category of downhole chemicals--for example, alcohols. Each category must be defined, as is shown in Figure 7, before it is used in a rule. Categories are used in antecedents with the operators ONLY, EXACTLY, or INCLUDE. The antecedent is true if the members of the category present are a subset of (ONLY), exactly equal to (EXACTLY), or a superset of (INCLUDE) the list of members given in the antecedent. A rule using the category defined in Figure 7 is shown in Figure 8. Figure 9 illustrates the user queries with responses for the same category.

EXPERT-3 has two types of consequents. The first is the basic THEN clause common to all expert systems and as was shown in all figures thus far. The second is THENHYP, which specifies that this consequent is a hypothesis, a goal for the inference engine. This is demonstrated in Figure 10. Any EXPERT-3 operator (IF, THEN, AND, ONLY, etc.) can be suffixed with NOT (IFNOT, THENNOT, ANDNOT, ONLYNOT, etc.) to denote negative logic.

EXPERT-3 allows the user to respond with "I don't know" to queries he is uncertain about by inputting a question mark. The inference engine will attempt to prove or disprove each hypothesis without this piece of information. When the conclusions are output, the inference engine indicates any hypotheses that could not be proven or disproven because of one or more "I don't know" responses. An example of this is shown in Figure 11. The "I don't know" response is a very

important feature of EXPERT-3. Without it, the user would be forced to guess when he was uncertain and guessing can obviously lead to incorrect conclusions.

A consultation with an expert system developed by EXPERT-3 is divided into stages. Each stage tests rules pertaining to a certain subtopic in the expert system's domain of expertise. For example, the elastomers expert system's domain of expertise is the ability of different elastomers to withstand varying oil patch environments. It is divided into three stages: temperature and pressure, produced fluids, and completion and injection fluids. Conclusions are output following each stage as is shown in Figures 2 and 11. The user can also request the inference engine to output partial conclusions at any point during a stage by inputting a "C". Hypotheses that are disproven in one stage are not considered in the next stage. After the conclusions are output following each stage, the user can continue to the next stage, repeat the current stage if he wishes to change some of his answers, or restart the consultation from the beginning. The final conclusions are those output after the last stage.

Unlike some expert systems, EXPERT-3 does not stop after the first hypothesis it proves to be true but tests every hypothesis in the rule base. This is necessary in an expert system for materials selection since, for example, we do not just want to know one elastomer that can withstand a given oil patch environment, we want to know all that can. EXPERT-3 attempts to minimize the number of questions asked the user during a consultation. When considering a hypothesis, the inference engine first checks if it can be proven or disproven with the facts it already knows. The user will be queried only if the hypothesis cannot be proven or disproven with the existing facts.

We attempted to make the user interface as friendly as possible. The user's responses are just one character (Y for yes, N for no, ? for I don't know, etc.), except for numerical values, which are entered as integers. The valid user responses are summarized in Table 1. After each question, the valid responses are displayed in parentheses. If the user depresses a key that is inappropriate for the current query, the user interface will sound a "beep" and ignore the character. Therefore, illegal input to the inference engine is impossible. The user can ask the system why it is following a certain line of questioning by inputting a "W". The expert system will indicate its current line of reasoning by outputting the hypothesis it is currently considering and the rule it is testing, as is shown in Figure 12. The user can also request information about an item in the system's domain of expertise by inputting an "I". For example, in the elastomer expert system, the user can request information about a certain elastomer, its properties, and what testing it has undergone, as is shown in Figure 13.

ELASTOMERS EXPERT SYSTEM

The initial goal of the elastomers expert system was to select which elastomer or elastomers would withstand a certain oil patch environment. As the system was developed, it became obvious that this goal was too ambitious for our elastomers data. Not every elastomer was tested in every type of environment. So the goal was modified to selecting candidate elastomers for a given environment. The candidates would be elastomers that were not known to fail because of any of the conditions of the given environment. This is not the same as being known not to fail in the given environment. We had to follow this candidate selection approach because of the incompleteness of our elastomer data. After the expert system selects candidates for a given

environment, the user can then select one of the candidates for his application or perform further testing on the candidates in conditions corresponding to his environment. If he chooses the latter, the results of the tests can be added to the rule base to improve the expert system.

The elastomers expert system contains three stages: temperature and pressure, produced fluids, and completion and injection fluids. The stages were ordered so the data in which we had the most confidence, temperature and pressure, was first and the data in which we had the least confidence, completion and injection fluids, was last. All three types of antecedents (basic IFs, variables, and categories) were utilized in the elastomers expert system.

CONCLUSIONS

A major conclusion is that it is feasible to develop and operate a useful expert system on a personal computer. Personal computers obviously have much less processing power than main frames, but for specialized expert systems, such as materials selection, PCs are more than adequate. An expert system with a broader domain of expertise--for example, a consultant in all areas of corrosion--might require more processing power than a PC offers. In the development of the elastomers expert system, we were limited by our expertise about elastomers, not by the PC hardware or the EXPERT-3 software.

From the development of the elastomers expert system, we learned that the coding of a human expert's expertise into rules can be difficult. It is not always obvious how to express an expert's intuition in the required IF-THEN syntax. The development of an expert system forces the expert to examine his expertise in a systematic manner which, in many cases, such as the elastomers expert system, provides a clearer picture of the extent of the expertise and where holes exist in the data. The coupling of expert system development with experimental design could lead not only to better expert systems, but also to better expertise.

REFERENCES

1. Offshore Computers, June 1985, p. 10.

2. R. E. Dessy, Anal. Chem., Vol. 56, No. 11, p. 1200A (1984).

3. R. O. Duda and E. H. Shortliffe, Science, Vol. 220, No. 4594, p. 261 (1983).

4. S. P. Ennis, Oil and Gas Journal, Vol. 81, No. 30, p. 184 (1983).

5. R. O. Duda and J. G. Gasching, Byte, Vol. 6, No. 9, p. 238 (1981).

6. A. Ralston, Ed., Encyclopedia of Computer Science, Van Nostrand Reinhold Company, New York, New York, p. 1209 (1976).

7. J. Park, MVP-FORTH Expert System Toolkit, Mountain View Press, Mountain View, California (1983).

8. J. S. James, Byte, Vol. 5, No. 8, p. 100 (1980).

TABLE 1--EXPERT-3 User Responses

Y	- Yes
N	- No
C	- Draw conclusions at this point
I	- Give me information
W	- Why are you asking this question?
?	- I don't know

Variables are input as integer values.

```
ENTER VALUES --- Must be integers
Enter ? if you don't know value
Enter C for conclusions at this point

Carbon dioxide (psi)   25

Hydrogen sulfide (ppm)   5

Is there acid (pH < 2) present? ( Y,N,C,I,W,? ) N

Is there caustic (pH > 13) present? ( Y,N,C,I,W,? ) Y

Is there a sulfite oxygen scavenger present? ( Y,N,C,I,W,? ) Y

Are there aromatic hydrocarbons present? ( Y,N,C,I,W,? ) N
```

FIGURE 1--Example of expert system queries.

```
After the final stage, the following elastomer(s) are candidates:
Aflas 100H 75              Aflas 100H 95              Buna N (Parker)
Buna N MPI                 Buna N (National C90)

After the final stage, the following elastomer(s) are not suggested:
Viton A (National V23)     Viton GF (National V34)    Aflas GT-787
Aflas GT-797               Aflas GT-799               Aflas GT-791
Aflas 100H 85
```

FIGURE 2--Example of expert system conclusions.

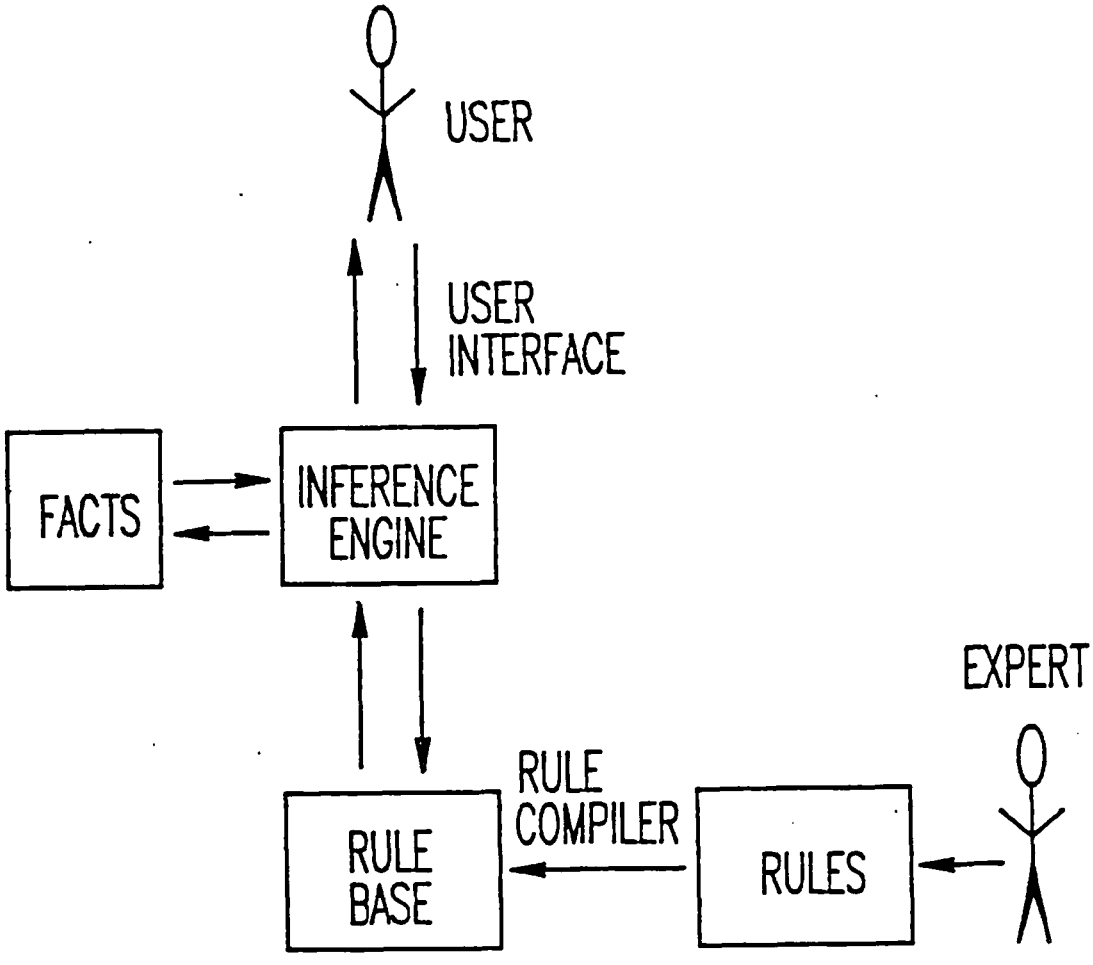


FIGURE 3--Diagram of an expert system.

```
IF   Is there a scale
AND  Does corrosion scale react with HCl/As solution
AND  Does reaction with HCl/As solution produce yellow residue
THEN Sulfur is involved
```

FIGURE 4--Example of a rule.

```
IFNOT   Maximum temperature (F) GT 350
ANDNOT  Minimum temperature (F) LT -30
THEN    Temperature ok for Viton
```

FIGURE 5--EXPERT-3 rule using variables.

```
ENTER VALUES --- Must be integers
Enter ? if you don't know value
Enter C for conclusions at this point

Maximum temperature (F)    200

Minimum temperature (F)    0

Maximum pressure (psig)    ?
```

FIGURE 6--EXPERT-3 variable queries.

```
CATEGORY ALCOHOLS    Are there any alcohols present
                     Please indicate which alcohols:
                       Methanol
                       Ethylene glycol
                       Any other alcohol
END
```

FIGURE 7--EXPERT-3 definition of category.

```
alcohols ONLY
           Ethylene glycol
         END
ANDNOT Maximum temperature (F) GT 250
THEN   Viton A can handle the alcohols
```

FIGURE 8--EXPERT-3 rule using category.

```
Are there any alcohols present? ( Y,N,C,I,? ) Y

Please indicate which alcohols:

Methanol? ( Y,N,C,I,? ) Y

Ethylene glycol? ( Y,N,C,I,? ) ?

Any other alcohol? ( Y,N,C,I,? ) N
```

FIGURE 9--EXPERT-3 category queries.

```
IF Is there a scale
AND Does corrosion scale react with HCl/As solution
AND Does corrosion scale fizz when HCl/As solution is added
ANDNOT Sulfur is involved
AND Are there pits
AND Do the pits have sharp edges
THENHYP Carbon dioxide corrosion
```

FIGURE 10--EXPERT-3 rule using THENHYP.

```
After the final stage, the following elastomer(s) are candidates:
Aflas 100H 75              Aflas 100H 95

After the final stage, I need more data for the following elastomer(s):
Viton A (National V23)     Viton GF (National V34)    Aflas GT-787
Aflas GT-797               Aflas GT-799               Aflas GT-791
Aflas 100H 85              Buna N (Parker)            Buna N MPI
Buna N (National C90)
```

FIGURE 11--EXPERT-3 conclusions with insufficient data.

```
Will the production be mostly liquid? ( Y,N,C,I,W,? ) W

I am trying to prove:  Viton A (National V23)
I am testing rule #3
IFNOT Maximum pressure (psig) GT 15000
IF Will the production be mostly liquid
THEN Pressure ok for Viton

Will the production be mostly liquid? ( Y,N,C,I,W,? )
```

FIGURE 12--EXPERT-3 why feature.

```
Will the production be mostly liquid? ( Y,N,C,I,W,? ) I

Information is available on:
A)Viton A (National V23)  B)Viton GF (National V34) C)Aflas GT-787
D)Aflas GT-797            E)Aflas GT-799            F)Aflas GT-791
G)Aflas 100H 75           H)Aflas 100H 85           I)Aflas 100H 95
J)Buna N (Parker)         K)Buna N (National C90)   L)Buna N MPI

Enter letter: A

                    Viton A (National V23)
    Temperature (F) range:  -30 to 350
    Maximum pressure (psig);  dynamic service in gas:      10,000
                              dynamic service in liguid:   15,000
                              static  service in gas:      15,000
                              static  service in liguid:   20,000
    Fails in CO2 > 250 psig with H2S > 5% above 300 degrees.
    Fails in bromine salts above 250 degrees.
    Fails in caustic.
    Fails in acid without lead oxide filler.
    Fails in aromatic hydrocarbons above 250 degrees.
    Biocides:   compatible with XC-102.
    Inhibitors: compatible with Milchem Brine Pak.
    Alcohols:   compatible with ethylene glycol.

Will the production be mostly liquid? ( Y,N,C,I,W,? )
```

FIGURE 13--EXPERT-3 information feature.

COMP: MINEQL

MINEQL

A COMPUTER PROGRAM FOR THE CALCULATION OF CHEMICAL EQUILIBRIUM COMPOSITION OF AQUEOUS SYSTEMS

John C. Westall
Joseph L. Zachary
Francois M.M. Morel

Water Quality Laboratory
Ralph M. Parsons Laboratory
for Water Resources and Environmental Engineering
Department of Civil Engineering
Massachusetts Institute of Technology

Technical Note No. 18 1976

Sponsored by
EPA Grant No. R-803738

Acknowledgement

The authors wish to acknowledge Dr. James M. Westall, Jr. and the Clemson University Computation Center for consultation and technical support; Mr. Patrick Wong for programming assistance; and Mrs. Jean Coleman for preparation of the manuscript.

CONTENTS

List of Tables

List of Figures

Introduction

This report describes the computer program for the computation of chemical equilibria in aqueous systems, MINEQL. The purpose of the report is to present the program in sufficient detail that the reader not only can use the program exactly as it is written, but also can modify it to suit whatever his particular needs may be.

The evolution of the program, examples of its use, and general approach to the chemical equilibrium problem are described in Part I of the report. Part II presents a "MINEQL-1 User's Manual".

A detailed mathematical description of the solution to the chemical equilibrium problem used in MINEQL appears in Appendix 1. It was felt that the solution could best be presented by an abstract description of the mathematical problem (without any reference to its chemical analog), followed by a very simple example of a chemical problem (which is given in Appendix 2). This avoids the intermingling of chemical and mathematical concepts which is sometimes confusing. Appendix 3 gives a description of the program itself.

The three appendices are related in that they present the same concepts - the actual theory and its application in MINEQL - from three different viewpoints: mathematical, chemical, and computational. They should be read in conjunction with one another. First the concepts involved in the basic chemical equilibrium problem, (i.e., where there are no solid phases present) should be mastered, and then the extension to the consideration of solid phases, which is somewhat more complicated, should be undertaken.

This report contains a complete description of all of the concepts employed in the program MINEQL. However, a working knowledge of the program is obtainable without attention to all of the details.

It is hoped that this roadmap through MINEQL will enable the reader to maximize his use of the program without becoming bogged down in the details of the report.

A. Evolution of MINEQL

The program MINEQL has evolved from ideas and experience gained from the use of the series of programs REDEQL (7,8). The original aim in the creation of MINEQL was a compact program, necessarily of limited flexibility and problem size, but suitable for execution on small (16 K - 64 K) computers. However, it soon became apparent that the programming approach used in these early versions of MINEQL could profitably be applied to the larger chemical equilibrium problems as well. In the expansion of MINEQL, as in the development of any large program, trade-offs had to be made between compactness and efficiency of execution on the one hand, and clarity of programming and ease of use on the other. The foremost of these considerations observed in the design of MINEQL are clarity of programming and reasonable ease of use. In some cases, the flexibility of the program through input of control cards has been sacrificed for the sake of maintaining compactness, but it is intended that the clarity of programming should enable even a minimally competent programmer to modify the program to suit his needs.

Experience gained in the use of the program REDEQL has given guidance in the selection of input and output features. Studies for which REDEQL has been used and for which the I/O of MINEQL is particularly suited include both laboratory and natural systems. For example, REDEQL has been used to study the speciation of metals in algal culture media, the stability constants for metal chelates in complex media, the degradation of NTA in natural waters, the speciation of trace metals in sewage oxidized and diluted by seawater, and the pH stability of

the oceans.

The version of MINEQL presented here is designed for the large scale problems mentioned above and requires a full size computer (128K) for execution. It can be conveniently modified for smaller computers if necessary.

Recent publications which include material relevant to the computation of chemical equilibria in aqueous solution begin with the program HALTAFALL by Sillen and co-workers (5). Then there follows a collection of papers by Morel and co-workers (8-10) which discuss many facets of the chemical equilibrium problem.

Zirino and Yamamoto (11) have considered the speciation of trace metals in seawater and give a discussion of activity coefficient corrections for that medium. Truesdell and Jones (12) in the description of their program WATEQ give another good discussion of activity corrections and present a well documented table of thermodynamic data including ΔH for temperature correction.

B. Methodology for Chemical Equilibrium Problems

The problem of finding the equilibrium composition/speciation of an aqueous system is a programming problem which in its most general form can be expressed as follows: minimize the Gibbs free energy of the system subject to the constraints of mass balance. All of the many methods for computation of chemical equilibria are but variations on this theme, but as Zeleznik and Gordon have suggested (1), it is just these variations that convert the theory into a program useful for a particular application.

Two basic approaches exist to the chemical equilibrium problem. One which has been developed extensively in a series of reports from the Rand Corporation (2 - 4), involves the following procedure: from an initial guess of the solution, a positive feasible solution to the mass balance equations is found without consideration of the Gibbs Free Energy of the system; then subject to the constraint of the mass balance equations the Gibbs free energy is minimized. Another approach, the so-called equilibrium constant approach, begins with an initial guess for a set of components from which the minimum Gibbs free energy composition is readily calculated from equilibrium constants, then the mass balance equations are solved by iteration. This is the approach used in the programs of Sillen (5), Perrin (6), and Morel (7), and others.

Thus in the first case, mass balance equations are the constraints as the Gibbs free energy function is minimized, and in the second case, the Gibbs free energy function is directly minimized and the mass

balance equations are solved. Zeleznik and Gordon present a complete discussion of this topic (1).

While the former "direct minimization of the Gibbs free energy function" method seems more elegant in a thermodynamic sense, the latter "equilibrium constant" method appears to be more an extension of chemists' intuition, and thereby more useful and more tractable in many ways. The equilibrium constant method is used in MINEQL.

One may view the equilibrium constant approach in the following way: at every step in the iteration towards the final solution a chemical equilibrium problem is solved (i.e., the computed composition represents the minimum Gibbs free energy for a set of mass balance constraints); the task is to iterate to the correct chemical equilibrium problem (i.e., the chemical equilibrium problem with the desired set of mass balance constraints).

Similarly, one may consider a computation involving solid phases as an iteration toward the correct problem, but at a higher level of hierarchy: for a given set of solid phases in equilibrium with the aqueous phase, a chemical equilibrium problem can be solved, as described above. But then it is necessary to find the correct set of solid phases in equilibrium with the aqueous phase, (i.e., that set for which no solubility products are exceeded and no mass balance conditions are less than zero). This is achieved by successively redefining the set of solid phases considered to be in equilibrium with the aqueous phase and solving the resulting equilibrium problem until the conditions specified above for solid phases are met.

At this point it is recommended that the reader branch to Appendix 1, "Basic Problem" for a description of the mathematics of MINEQL, before proceeding to the User's Manual.

PART II MINEQL-1 USER'S MANUAL

A. Definitions

B. Description of the Program

C. Input

D. Output

E. Redox

F. Gases at Fixed Partial Pressure

G. Special Usage

H. Error Messages

I. Storage

3. Soluble species - include every species in the aqueous phase. These can be simple ions: Ca^{2+}, H^{+}, OH^{-}; ion pairs: $CaCO_3$ (aq); chelates Ca EDTA; etc. Soluble species can be both the soluble components and complexes (products of reactions involving components). The concentration (activity) of every soluble species is variable, being a function of the concentration of the components of which the species is composed.

4. Solids - are species of a fixed activity (for normal solid phases such as $CaCO_3$ (s), $Ca(OH)_2$(s), this fixed activity is equal to one). However, gases at a fixed partial pressure are also solids according to this definition. Solids may be in two states, dissolved or precipitated.

5. Dissolved solids play no direct role in a chemical equilibrium computation. The fact that a dissolved solid exists indicates simply that the solubility data for the solid is present, and after an equilibrium computation has been made, the solubility of this solid phase may be checked. If the solubility product is exceeded, the solid may be precipitated and become directly involved in the computation.

6. Precipitated solids are those solid phases which are present at a fixed activity. This implies a certain fixed relationship among the components with the result that the chemical equilibrium problem loses one degree of freedom. This is equivalent to transforming the set of components to include the solid phase, which is then set at fixed activity. If, after a chemical equilibrium computation has been made with a given solid phase, the amount of that solid phase is calculated

II. MINEQL-1 User's Manual

A. Definitions

There are a number of terms used throughout this report which have a particular significance outside of their normal English language, chemical, or mathematical use. These terms will be defined.

1. Components - in a mathematical sense this is the independent basis set from which every species can be defined and upon which the mass balance equations are based. In a chemical sense, the components are a set of chemical entities such that every species can be represented as the product of a reaction involving only these components, and no component can be represented as the product of a reaction involving only the other components. The particular set of components for a given problem is certainly not unique, but once a set of components has been chosen, every species has a unique representation in terms of this set. There has been a tendency to consider only neutral entities for the set of components; however, there is no reason for the exclusion of ionic entities from this set, (nor are there any such restrictions in Gibbs original definition of components (15)) and in this discussion ionic entities will be included as components.

2. Species - mathematically, a species represents a log linear combination of the components; chemically, a species is the product of a chemical reaction involving the components as reactants. The species include every chemical entity to be considered in a chemical equilibrium problem. Species are of two kinds: soluble species and solids.

to be negative, the solid may be dissolved, i.e., removed from the computation and the fixed relationship among the constituents disestablished.

The six definitions given above are applicable to chemical equilibrium problems in general. For the purpose of presenting the program MINEQL, these definitions will further be used to define six types of species which have significance within the program.

Type I Species - are the soluble species which correspond to the components.

Type II Species - are the complexes (all soluble species which are not components).

Type III Species - are precipitated solids which are not allowed to dissolve, even if the amount of these solids becomes negative. Examples are gases at a fixed partial pressure, or simply a solid phase which is specified to be present. In addition, if the concentration (activity) of any soluble species is to be fixed, i.e., set to a certain value (for example, fixed pH), this fixed soluble species is included as a Type III species.

Type IV Species - are precipitated solids which are subject to dissolution if the amount present becomes less than zero.

Type V Species - are dissolved solids which are subject to precipitation if the solubility product is exceeded.

Type VI Species - are species which are not to be considered at all (e.g. dissolved solids which are not subject to precipitation, or the electron, which does not exist in solution).

B. Description of the Program

MINEQL, as outlined in the following description, is relatively straightforward. Execution is controlled by the main program which calls a series of subroutines (Figure 1). A description of the principle variables is found in Table I.

MAIN begins by initializing the program variables ITMAX, EPS, THRSH, ITER & XMU. Then INPUT is called which combines the user supplied data for the components to be included and their analytical concentrations with the thermodynamic data for all of the species formed from these components. INION reads the ionic charge data for ionic strength correction and IONCOR performs the ionic strength correction. The problem is then totally defined and OUTPUT (entries IONCMP and IONSPC) prints these data for verification. Subsequently SOLID is called to modify the problem for the existence of the solid phases, then SOLVE solves the resulting problem for the soluble species. SOLIDX solves for the amounts of the solid species, performs precipitation and dissolution, and "un-modifies", or restores, the matrices to the state in which they were before SOLID was called. If no precipitation or dissolution occurred, OUTPUT prints the results (entries OUTCMP, OUTSPC, OUTPC); if precipitation or dissolution did occur, the solution sequence is restarted with the new set of solids by a return to subroutine SOLID.

An important principle of MINEQL worthy of further comment is the various types of species. Each species is handled computationally according to its type. The program keeps track of the type of a species by its storage location in the A and K matrices and the index array NN:

the species are stored contiguously by types (Type I, Type II, etc.), and NN(L) specifies how many species of Type L there are. The initial assignment of storage locations and indexing is done in subroutine INPUT. Further exchange of storage locations and indexing is done in subroutine SOLIDX as solids are precipitated or dissolved by changing the type of a species.

C. Input

The input data which defines a problem for MINEQL is read in several groups. The first group of data contains the information for the components to be included in the computation: one card for each component, followed by a blank card to delimit the group. After all of the components have been read in, the input subroutine retrieves from the thermodynamic data set (see the end of this section for description) data for all of the species which are formed from the given set of components. The default type specification for these species are:

Components:	Type I	soluble species
Complexes:	Type II	soluble species
Solids:	Type V	dissolved solids subject to precipitation

Then follows several groups of data in which the type specification of a species may be modified: the first card of one of these groups defines the species type being considered; it is followed by one card for each species to be assigned to that type, and finally a blank card to delimit that group of data. The input is completely general: there may be any number of cards in any group and there may be any number of groups to define species types. All input is terminated by a final blank card.

A diagram of the exact format of the input data is given in Table IV. Input in all cases is grouped in format fields of seven characters to simplify data preparation. The identification numbers of the components are given in Table II. There are 91 chemical entities listed: cations have identification numbers 1-50, the electron is number

99, and anion and neutral ligands have identification numbers 101-199. Ligands are generally given in their least protonated form. The identification numbers of the species are given in Table III.

The input for species type allows for great flexibility in modifying formation constants for species or for adding new species: 1) if only IDYT is specified (the rest of the card left blank), species IDYT is simply assigned to the indicated type; 2) if only IDYT and GKT are specified (the rest of the card left blank), species IDYT is assigned to the indicated type and the formation constant is set equal to GKT; 3) if IDYT is an identification number not previously included in the computation, and GKT and (IDXT(J), IAT(J), J = 1,4) are included on the card, a new species is generated with the data and assigned to the indicated type. (If less than four components are all that are necessary to specify the new species, only the necessary number need be specified and the rest of the card left blank). A flow chart of the process is given in Figure 2.

One further comment is necessary concerning specification of Type III species, (components at a fixed activity): Since computationally these species are equivalent to solids (for which $[X_1]^{a_1} \cdot [X_2]^{a_2} \ldots K=1$), to specify a component present at a concentration X, the log formation constant must be logK = -log[X]. For example, the following input group fixes pH = 8:

00003		Type III species
00050	8.00	I.D. # of H; -log[H]
blank card		termination of group

In addition to the input data described above (which is normally supplied through the card reader), there are two data sets which are normally stored on disc or tape. These data sets, the files to which they are directed, and their contents are:

THRM. DATA	FT10F001	Thermodynamic data
IONC. DATA	FT11F001	Ionic charge data for ionic strength correction

Appropriate job control statements must be included to allocate these data sets to the files (Table VI).

D. Output

The output for MINEQL is broken down into three areas, each of which is accessed by ENTRY statements in subroutine OUTPUT.

1. Input data for verification:

   ENTRY OINCMP* — Data for components

   ENTRY OINSP — Thermodynamic data for species

2. Output data for a solved problem

   ENTRY OUTCMP — Data for components

   ENTRY OUTSP — Species

3. Percentage Distribution of components

   ENTRY OUTPC*

* Produces a new page and page heading

OINCMP prints for each component:

IDX = component identification number

NAME = name of component

X = guess of concentration of component

GX = log of X

T = total (analytical) concentration of component

OINSPC prints for each species:

IDY = species identification number

GK = log of stability constant of species

NAME(J), A(I,J) = name and stoichiometric coefficient of each component j in species

OUTCMP prints for each component:

IDX = component identification number

X = equilibrium concentration of component

T = total (analytical) concentration of component

Y = remainder of component (amount not accounted for in mass balance equation)

NAME = name of component

OUTSPC prints for each species:

IDY = species identification number

C = equilibrium concentration of species

GC = log of C

GK = log of stability constant for species

NAME(J),A(I,J) = name and stoichiometric coefficient of each component j in species

For the solid phases the values C and GC assume a slightly different meaning:

1) For solids which are dissolved (Type V and Type VI), C and GC are the values of the solubility expression ($C = K \cdot X_1^{a_1} X_2^{a_2} \ldots X_n^{a_n}$) and its log.

2) For solids which are precipitated (Type IV), C and GC represent the amount of solid per liter of solution and the log of that number.

3) For fixed solids and fixed soluble species (Type III):

a) if C is positive, C and GC represent the amount of that species per liter of solution that must be removed to achieve the desired "fixed" condition.

b) if C is negative, C and GC represent the amount of that species per liter of solution that must be added to achieve the desired "fixed" condition

The percentage distribution (OUTPC) gives, for each component, its distribution among the species (subject to a threshold; if the fraction of a component in a species is less than the threshold, this species is not listed in the distribution)

PC = percentage of the component in the species

IDY = identification number of the species

NAME(J),A(I,J) = name and stoichiometric coefficient of each component j in the species

There is an anomaly in this output which should be mentioned: for the component H^+, some species have negative as well as positive stoichiometric coefficients. Thus the percentage distribution for H^+ is not always a meaningful concept.

E. Redox Reactions

To include redox reactions in a computation it is necessary to include the electron, component identification number 99. Normally the pε ($p\varepsilon = -\log_{10} \varepsilon$; $E_H = 0.59 \log_{10} \varepsilon$ @ 25°C) will be fixed at a given value for a computation: thus species 99 should be included as a Type III (fixed component) species. If the pε is to be calculated, species 99 must be included as a Type VI (species not included) species since the free electron does not exist in solution.

A list of the Redox reactions is given in Table VII. The default type specification for redox complexes is Type II; for redox solids it is Type V (dissolved solids subject to precipitation), and for redox reactions which relate two components, e.g. Fe^{3+}/Fe^{2+} or Cu^{2+}/Cu^{+}, Type III.

F. Gaseous Phases

To include a gas phase at fixed partial pressure, it is necessary to include the gas as a Type III species (fixed solid) and to specify its partial pressure. An example using $CO_2(g)$ will be given.

The formation expression for $CO_2(g)$ is, under ideal conditions,

$$[H^+]^2\ [CO_3^{2-}]\ K = [CO_2(g)] \doteq P_{CO_2} \qquad \log K = 10^{18.0}$$

Since computationally every solid phase has an activity of one, the expression is rearranged to

$$[H^+]\ [CO_3^{2-}]\ \left[\frac{K}{P_{CO_2}}\right] = 1$$

and the effective constant for the phase at a fixed partial pressure P_{CO_2} becomes

$$\log K' = \log K - \log P_{CO_2}$$

Then to define a problem where an aqueous phase is in equilibrium with the atmosphere ($P_{CO_2} = 10^{-3.5}$)

$$\log K' = 18.0 - (-3.5) = 21.5$$

and the following input group makes that specification:

00003	Type III species
25000 21.5	ID# of $CO_2(g)$; log K'
blank card	termination of group

The default species type for gases is Type VI, species not considered. Thus if the gas is not explicitly specified, it will not be considered.

G. Special Usage

MINEQL has been written in such a way that the user has great flexibility in specification of his problem by simply making a few modifications in the MAIN; that is, the subroutines are available from which one may construct his own program. (The basic version of MAIN is presented in Figure 1).

There are numerous ways in which the MAIN of MINEQL can be modified to solve particular problems. Two cases will be given as examples which can be used as patterns for other modifications.

1) Simulate a titration with strong acid with increments from 1 to 20mM acid.

MAIN

```
      :
      :
      CALL INPUT
      CALL INION
      CALL IONCOR(XMU)
      CALL OINCMP
      CALL OINSPC
      DO 100 J = 1,20                          ←
      ITER = 0                                 ←
      T(IADX(50)) = T(IADX(50)) + 1.0E-3       ←
10    CONTINUE
      CALL SOLID
      CALL SOLVE
      CALL SOLIDX(K)
      IF (K.NE.0) GO TO 10
      CALL OUTCMP
      CALL OUTSPC
      CALL OUTPC(THRSH)
100   CONTINUE                                 ←
      STOP
      END
```

2) Run a series of computations with pH fixed from pH = 2 to pH = 10.

MAIN

```
      :
      :
      CALL INPUT
      CALL INION
      CALL IONCOR(XMU)
      CALL OINCMP
      CALL OINSPC
      DO 100 J = 1,20                    ←
      ITER = 0                           ←
      GK(LADY(50)) = J                   ←
10    CONTINUE
      CALL SOLID
      CALL SOLVE
      CALL SOLIDX(K)
      IF (K.NE.0) GO TO 10
      CALL OUTCMP
      CALL OUTSPC
      CALL OUTPC(THRSH)
100   CONTINUE                           ←
      STOP
      END
```

Note that species number 50 must have previously been specified as Type III (fixed soluble species) through input data.

H. Error Messages

There are several errors which cause an error message to be printed and execution to be terminated. The error messages, the subroutine from which they are called, the probable cause and corrective response are given below.

1. "COMPONENTS > NXDIM", SUBROUTINE INPUT:

   The number of components in the problem is greater than the dimension of the arrays allocated for their storage. Response: Either remove non-essential components from the computation or increase the dimension of the arrays (See Section I, "Storage").

2. "SPECIES > NYDIM", SUBROUTINE INPUT:

   The number of species in the problem is greater than the dimension of the arrays allocated for their storage. Response: Either remove non-essential components from the computation or increase the dimension for the arrays (See Section I, "Storage").

3. "ID NOT FOUND: INPUT", SUBROUTINE INPUT:

   The species identification number used to re-specify an existing species type was not found, and the stoichiometry was not valid for generating a new species. Response: Check (a) input format; (b) whether the indicated species was included in the computation

4. "ID NOT FOUND: IADY" FUNCTION IADY

   A species identification number was not found. Response: Verify that the identification number is valid, and that the species is included in the computation

5. "ID NOT FOUND: IADX" FUNCTION IADY

A component identification number was not found. Response: Verify that the identification number is valid, and that the component is included in the computation.

6. "PHASE RULE VIOLATION" SUBROUTINE SOLID

More solid phases have been specified than there are allowed by the Phase Rule. Response: Restart computation with an alternate set of Type III and/or Type IV solids.

7. "ITERATIONS > ITMAX" SUBROUTINE SOLVE

The number of iterations allowed in subroutine SOLVE has been exceeded. Response: Check a) the value of ITMAX; b) whether ITER has been reinitialized if solving several problems within the same computation; c) whether initial guesses have been properly entered; d) if Type III and Type IV solids have been chosen wisely.

8. "SINGULAR Z MATRIX" SUBROUTINE SIMQ

The Jacobian has been found to be singular. Response: Check for an input error; the Jacobian is very rarely singular if the problem has been correctly specified.

I. Storage

The approximate storage requirements for MINEQL are given below. The data are based on 8 bit bytes, 4 byte words, 1 word variables for FORTRAN IV G-1 on an IBM 370/168. To change the dimensions of the arrays only two major changes are necessary:

1) Change NXDIM and NYDIM in MAIN
2) Change all of the dimensions in all of the COMMON/VAR/ statements. This can be done conveniently with a TSO editor.

PASSIVE STORAGE		
1. Thermodynamic Data Deck	1500 cards	120K
2. Source Program Deck	600 cards	48K
ACTIVE STORAGE		
1. Object Program minus Storage for arrays		18K
2. Storage for Arrays		
a) 10 components 100 species		7K
b) 25 components 300 species		38K
c) 30 components 2000 species		59K
d) 35 components 500 species		84K
3. Auxilliary Subprograms Called by linkage editor		24K
4. I/O Buffers		24K

TOTAL CORE FOR EXECUTION:

COMPONENTS	SPECIES	CORE
10	100	73K
25	300	104K
30	400	125K
35	500	150K

Table I

Description of Principle Variables

A(I,J)	- the stoichiometric coefficient of component J in species I
C(I)	- the concentration of species (I)
EPS	- the tolerance in the convergence test (EPS = 10^{-4} is a convenient value)
GC(I)	- the $\log_{10}$ of C(I)
GK(I)	- the $\log_{10}$ of the stability constant for species I
GX(J)	- the $\log_{10}$ of X(J)
IDX(J)	- the identification number of component J
IDY(I)	- the identification number of species I
ITMAX	- the maximum number of iterations allowed for a problem (for a single case ITMAX = 40 is usually sufficient to achieve convergence, if the problem is to converge at all)
ITER	- the counter for number of iterations
NN(L)	- the number of species of type L
NXDIM	- the X (component) dimension of the arrays
NYDIM	- the Y (species) dimension of the arrays
T(J)	- analytical (total) concentration of component J
THRSH	- threshold for percentage distribution output (THRSH = 0.01, 1% is a convenient value)
X(J)	- concentration of component J
XMU	- ionic strength
Y(J)	- the remainder in the mass balance equation for component J
Z(J,K)	- the element of the Jacobian representing $\partial Y_J/\partial X_K$

TABLE II

List of Components* and

Identification Numbers

1	Ca^{+2}	18	Ag^{+}
2	Mg^{+2}	19	Cr^{+3}
3	Sr^{+2}	20	Al^{+3}
4	K^{+}	21	Cs^{+}
5	Na^{+}	22	Li^{+}
6	Fe^{+3}	23	Be^{+2}
7	Fe^{+2}	24	Sc^{+3}
8	Mn^{+2}	25	TiO^{+2}
9	Cu^{+2}	26	Sn^{+2}
10	Ba^{+2}	27	Sn^{+4}
11	Cd^{+2}	28	La^{+3}
12	Zn^{+2}	29	Ce^{+3}
13	Ni^{+2}	30	Au^{+}
14	Hg^{+2}	31	Th^{+4}
15	Pb^{+2}	32	UO_2^{+2}
16	Co^{+2}	33	Cu^{+}
17	Co^{+3}	50	H^{+}
		99	e^{-}

101	CO_3^{-2}	120	$TART^{-2}$	139	ALA^-
102	SO_4^{-2}	121	EN°	140	TYR^{2-}
103	Cl^-	122	DIP°	141	MET^-
104	F^-	123	$SUSAL^{-3}$	142	VAL^-
105	Br^-	124	GLY^{-1}	143	THR^-
106	I^-	125	GLU^{-2}	144	PHE^-
107	NH_3	126	PIC^{-1}	145	ISO^-
108	S^{2-}	127	NTA^{-3}	146	LEU^-
109	PO_4^{3-}	128	$EDTA^{-4}$	147	PRO^-
110	$P_2O_7^{4-}$	129	$DCTA^{-4}$	148	$B(OH)_4^-$
111	$P_3O_{10}^{5-}$	130	$CYST^{-2}$	149	SO_3^{2-}
112	$SiO_2(OH)_2^{2-}$	131	NOC^{-3}	150	SCN^-
113	$S_2O_3^{2-}$	132	$PHTH^{2-}$	151	NH_2OH
114	CN^-	133	ARG^-	152	MoO_4^{2-}
115	AC^-	134	ORN^-	153	WO_4^{2-}
116	$ACAC^-$	135	LYS^-	154	AsO_4^{3-}
117	CIT^{-3}	136	HIS^-	155	HVO_4^{2-}
118	OX^{-1}	137	ASP^-	156	SeO_3^{2-}
119	SAL^{-2}	138	SER^-	157	NO_3^-
				158	TRIS

*

AC = acetate

ACAC = acetylacetone

CIT = citrate

OX = oxalate

SAL = salicylate

TART = tartrate

EN = enthylenediamine

DIP = dipyridyl

SUSAL = sulfosalicylate

GLY = glycine

GLUT = glutamate

PIC = picolinate

NTA = nitrilotriacetate

EDTA = ethylenediamine-tetraacetate

DCTA = 1,2-diaminocyclohexane tetraacetate

CYST = cysteine

NOC = nocardamine (desferri-ferrioxamine)

PHTH = phthalate

ARG = arginine

ORN = ornithine

LYS = lysine

HIS = histidine

ASP = aspartate

SER = serine

ALA = alanine

TYR = tyrosine

MET = methionine

VAL = valine

THR = threonine

PHE = phenylalanine

ISO = isoleucine

LEU = leucine

PRO = proline

TRIS = tris(hydroxymethyl)-aminomethane

Table III

Because of its length, Table III is included as computer output.

TABLE IV. ORGANIZATION OF INPUT

Description of Data	Format
one group for components	
IDXT, GXT, TT	(I5,2X,F7.2,E7.2)
.	
.	
.	
blank card	
one group for each type of species	
LTYPE	(I5)
IDYT,*GKT, *(IDX(J),IAT(J),J=1,4)	(I5,2X,F7.2,4(I4,I3))
.	
.	
.	
blank card	
one card to terminate all input	
blank card	

+ See following page for description. Sample input data is given in Table V.
* Optional input. See text

+ Description of Symbols

For each component a card where:

IDXT = component identification number

GXT = guess for the $\log_{10}$ of the concentration (molar) of the free component

TT = analytical (total) concentration (molar) of the component

For each species type to which species are to be assigned a card where:

LTYPE = type of species

followed by a card for each species to be assigned to that type where:

IDYT = species identification

*GKT = $\log_{10}$ of formation constant for the species

*(IDXT(J), IAT(J), J = 1,4) = stoichiometry of the species:

IDXT = component identification number, and IAT = stoichiometric coefficient of that component

* Optional Input. See text

Table V. Sample Input Data

```
 01  -3.0   1.00E-3
 02  -3.0   1.00E-3
 05  -4.0   1.02E-4
 50  -7.0   1.02E-4
 06  -20.0  1.00E-6
 07  -20.0  0.00E-6
101  -6.0   1.00E-4
102  -3.0   1.00E-3
103  -3.0   2.00E-3
128  -10.0  1.00E-6
099  -12.0  0.00E-6

 03
25000  21.5
00099  12.0

 04
20310

 06
20000
```

For a solution of the following composition

$MgSO_4$	10^{-3}M
$CaCl_2$	10^{-3}M
$NaHCO_3$	10^{-4}M
$FeCl_3$	10^{-6}M
Na_2EDTA	10^{-6}M

under the following conditions:

1. The solution is in equilibrium with atmospheric CO_2
2. pε fixed at 12; the redox reaction Fe^{III}/Fe^{II} is to be considered
3. The solid phase $Fe(OH)_3$ is presumed to be present
4. The solid phase $CaCO_3$ is not to be considered (because of unfavorable kinetics, for instance).

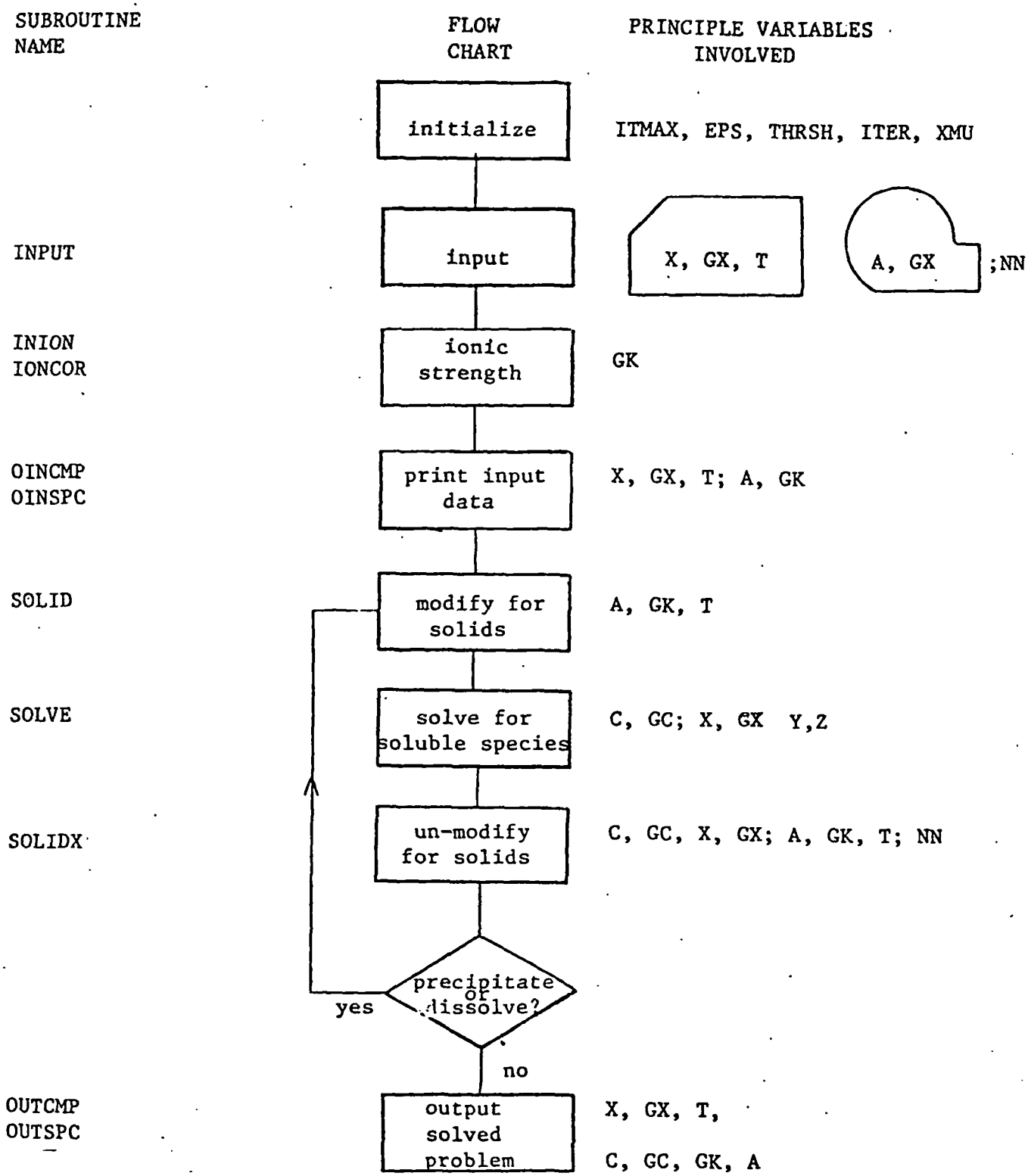
Table VI. Job Control Language
for MINEQL at M.I.T.

```
//    'USERID'
/*MITID
/*MAIN    LINES=4
//    EXEC    FORCLG
//C.SYSIN     DD  *,DCB=BLKSIZE=2000


INSERT MAIN HERE


/*
//L.SYSIN     DD  DSN=U.M10924.P11428.MINI.OBJ,DISP=SHR,LABEL=(,,,IN)
//G.FT10F001  DD  DSN=U.M10924.P11428.THRM.DATA,DISP=SHR,LABEL=(,,,IN)
//G.FT11F001  DD  DSN=U.M10924.P11428.IONC.DATA,DISP=SHR,LABEL=(,,,IN)
//G.SYSIN     DD  *,DCB=BLKSIZE=2000


INSERT PROBLEM DATA HERE


/*
//
```

Table VII. Redox Reactions

Redox reactions:	log K	I.D.
$Fe^{3+} + e^- = Fe^{2+}$	13.2	15000
$Co^{3+} + e^- = Co^{2+}$	31.6	15010
$Sn^{4+} + 2e^- = Sn^{2+}$	-5.1	15020
$Cu^{2+} + e^- = Cu^{+}$	2.7	15030
$SO_4^{=} + 8e^- + 8H^+ = S^{=} + 4H_2O$	20.0	15040
$Cu^{2+} + 2e^- = Cu^{o}(s)$	11.4	21510
$Hg^{2+} + 2e^- = Hg^{o}(s)$	28.7	21520
$S^{=} - 2e^- = S^{o}(s)$	16.2	21530
$Fe^{2+} + 2S^{=} - 2e^- = FeS_2(s)$	45.3	21450
$3Fe^{3+} + e^- + 4H_2O = Fe_3O_4(s) + 8H^+$	6.4	21460
$Mn^{2+} + 2H_2O - 2e^- = MnO_2(s) + 4H^+$	-42.0	21470
$3Mn^{2+} + 4H_2O - 2e^- = Mn_3O_4(s) + 8H^+$	-61.7	21480
$Mn^{2+} + 2H_2O - e^- = MnOOH(s) + 3H^+$	-25.7	21490
$Pb^{2+} + 2H_2O - 2e^- = PbO_2(s) + 4H^+$	-49.2	21500
$2Cr^{3+} + 7H_2O - 6e^- = Cr_2O_7^{2-} + 14H^+$	-135.2	13600
$Cr^{3+} + 4H_2O - 3e^- = HCrO_4^- + 7H^+$	-68.4	13610
$Cr^{3+} + 4H_2O - 3e^- = CrO_4^- + 8H^+$	-74.9	13620
$2Hg^{2+} + 2e^- = Hg_2^{2+}$	30.7	13630

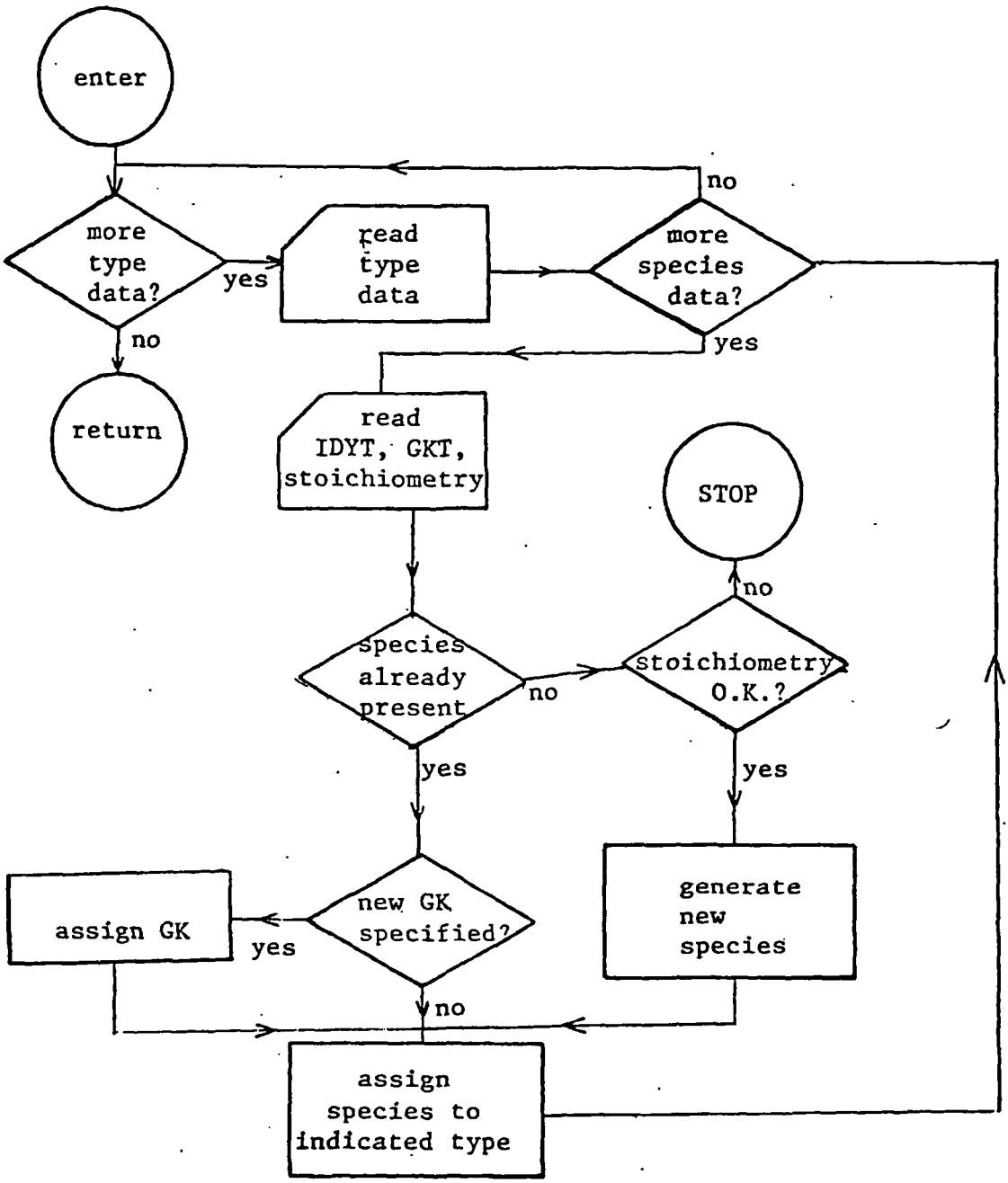
Figure 1. Flow Chart of MAIN *



* Listing of MAIN

```
00010 C   MAIN
00020        COMMON/PARM/NXDIM,NYDIM,ITMAX,ITER,EPS,NN(6),NNN
00030        COMMON/VAR/GX(30),X(30),T(30),Y(30),Z(30,30),
00040       1C(400),GC(400),GK(400),A(400,30),IDX(30),IDY(400)
00050 C
00060        NYDIM=400
00070        NXDIM=30
00080        ITMAX=30
00090        EPS=1.E-4
00100        ITER=0
00110        THRSH=.01
00120        XMU=0.5
00130 C
00140        CALL INPUT
00150        CALL INION
00160        CALL IONCOR(XMU)
00180        CALL OINCMP
00190        CALL OINSPC
00200 10     CONTINUE
00210        CALL SOLID
00220        CALL SOLVE
00230        CALL SOLIDX(K)
00240        F (K.NE.0) GO TO 10
00250        CALL OUTCMP
00260        CALL OUTSPC
00270        CALL OUTPC(THRSH)
00280        STOP
00290        END
```

Figure 2. Flow Chart of Species Type Specification



Appendix 1

Mathematical Description

1. Basic Problem* (aqueous phase only, no solid phases)

   1.1 General Description of Problem

   1.2 Computation of Complexes

   1.3 Computation of Mass Balance Equations

   1.4 Computation of Jacobian

   1.5 Iteration by the Newton Raphson Method

      1.5.1 Application to Matrix Equations

      1.5.2 Avoidance of Convergence to Negative Solutions

   1.6 Convergence Test

   1.7 Summary

2. Transformation of Basis

   2.1 Transformation of C

   2.2 Transformation of Y

   2.3 Computation of Mass Balance for new constituent

   2.4 Formulation of Transformation

3. Computation of Solids**

   3.1. One Solid Phase

      3.1.1 Modification of C

      3.1.2 Modification of Y

      3.1.3 Solution for Amount of Solid

3.2 Multiple Solid Phases

3.2.1 Review

3.2.2 One Solid Phase

3.2.3 Condensed Notation

3.2.4 Condensed Notation: One Solid Phase

3.2.5 Condensed Notation: Two Solid Phases

3.2.6 Many Solid Phases

* Corresponds to Subroutine SOLVE

** Corresponds to Subroutine SOLID and SOLIDX

1.1 General Description of the Problem

The problem of chemical equilibrium in an aqueous system can be described in mathematical terms as the set of equations

$$C_i = K_i \prod_{j=1}^{n} X_j^{a(i,j)}$$

for i=1,m

$$Y_j = \sum_{i=1}^{m} a(i,j)\ C_i - T_j$$

for j = 1,n

for which the solution is the set $\{X_j:\ j=1,n\}$ such that $\{Y_j:\ j=1,n\} = \{0\}$ given $\{K_i,\ a(i,j),\ T_j:\ j=1,n,\ i=1,m\}$. The significance of X,Y,K,a & T will be discussed below.

The solution to the problem is found by making an initial guess for the $\{X_j\}$ and iterating by the multidimentional Newton Raphson method to the root of the equation. The method converges over a wide range of initial guesses.

1.2 Computation of Complexes

$$C_i = K_i \prod_{j=1}^{n} X_j^{a(i,j)}$$

C_i is the concentration of the complex of stoichiometry

$$X_1{}_{a(i,1)} \ldots X_n{}_{a(i,n)}$$

K_i is the stability constant of the complex C_i

X_j is the concentration of component C_j

$a(i,j)$ is the stoichiometric coefficient of X_j in C_i

1.3 Computation of Mass Balance Equation

$$Y_j = \sum_{i=1}^{m} a(i,j)\ (C_i) - T_j$$

T_j is the total analytical concentration of component j

Y_j is the difference between the imposed total analytical concentration of component j (T_j), and the amount of component j in all of the complexes as computed from the X_j's. The solution to the problem is, of course, the set of X_j's such that all Y_j's = 0.

1.4 Computation of the Jacobian

$$Z(j,k) = \sum_{i=1}^{m} a(i,j)\ a(i,k)\ C_i/X_k$$

Z(j,k) is the element of the Jacobian representing $(\partial Y_j/\partial X_k)$.

It is derived as follows:

$$\frac{\partial Y_j}{\partial X_k} = \sum_{i=1}^{m} \left(\frac{\partial Y_j}{\partial C_i}\right)\left(\frac{\partial C_i}{\partial X_k}\right)$$

(1) $$\left(\frac{\partial Y_j}{\partial C_i}\right) = a(i,j)$$

(2) $$\left(\frac{\partial C_i}{\partial X_k}\right) = (a(i,k)\ X_k^{a(i,k)-1})\ K_i \prod_{\substack{\ell=1 \\ \ell\neq k}}^{n} X_\ell^{a(i,\ell)}$$

$$= \frac{a(i,k)}{X_k}\ (K_i\ X_k^{a(i,k)} \prod_{\substack{\ell=1 \\ \ell\neq k}}^{n} X_\ell^{a(i,\ell)}$$

$$= a(i,k)\,\frac{C_i}{X_k}$$

$$\left(\frac{\partial Y_j}{\partial X_k}\right) = \sum_{i=1}^{m} a(i,j)\; a(i,k)\; C_i/X_k$$

1.5 Iteration with the Newton Raphson Method

The Newton Raphson method for the determination of the root of the equation

$$y(x) = 0$$

yields the following iterative formula for the N + 1 iteration for x, given the value of x and $\frac{dy}{dx}$ at the N^{th} iteration

$$x^{N+1} = x^N - y^N \left[\left(\frac{dy}{dx}\right)\Big|_{x^N}\right]^{-1}$$

1.5.1 Application to matrix equations

In order to avoid matrix inversion the equation above is modified to:

$$y^N = \left[\left(\frac{dy}{dx}\right)\Big|_{x^N}\right](x^N - x^{N+1})$$

and the matrix equation

$$\bar{Y}^N = \bar{\bar{Z}}^N(\bar{X}^N - \bar{X}^{N+1})$$

(where $\bar{\bar{Z}}$ is the Jacobian of $\bar{Y}$ with respect to $\bar{X}$) is solved for $(\bar{X}^N - \bar{X}^{N+1})$ by Gaussian elimination. Then

$$\bar{X}^{N+1} = \bar{X}^N - (\bar{X}^N - \bar{X}^{N+1})$$

1.5.2 Avoidance of Negative Convergence

If for some X_j, $X_j^{N+1} = X_j^N - (X_j^N - X_j^{N+1}) < 0$

(i.e. the equation begins to converge to a physically meaningless negative solution), X_j^{N+1} is set to $X_j^N/10$.

1.6 Test for Convergence

In theory, the system of equations

$$C_i = K_i \prod_{j=1}^{n} X_j^{a(i,j)}$$

$$i = 1,m$$

$$Y_j = \sum_{i=1}^{m} a(i,j)\, C_i - T_j$$

$$j = 1,n$$

is "solved" when the set $\{X_j:\ j=1,n\}$ is found such that $\{Y_j:\ j=1,n\} \equiv 0$. In practice some criterion must be established such that when, by iteration, each Y_j becomes sufficiently close to zero, the system of equations is said to be "solved". Since each Y_j is the sum of terms which vary widely in order of magnitude, the convergence criterion is chosen to reflect the magnitude of Y_j relative to the maximum of the terms of which Y_j is the sum. Thus, the criterion for convergence is

$$\frac{|Y_j|}{\max(Y_j)} < \varepsilon \qquad \text{for } j=1,n$$

where $\max(Y_j)$ is the maximum of the absolute values of the set of terms $\{(a(i,j)\, C_i),\ i=1,m;\ T_j\}$ of which Y_j is the sum.

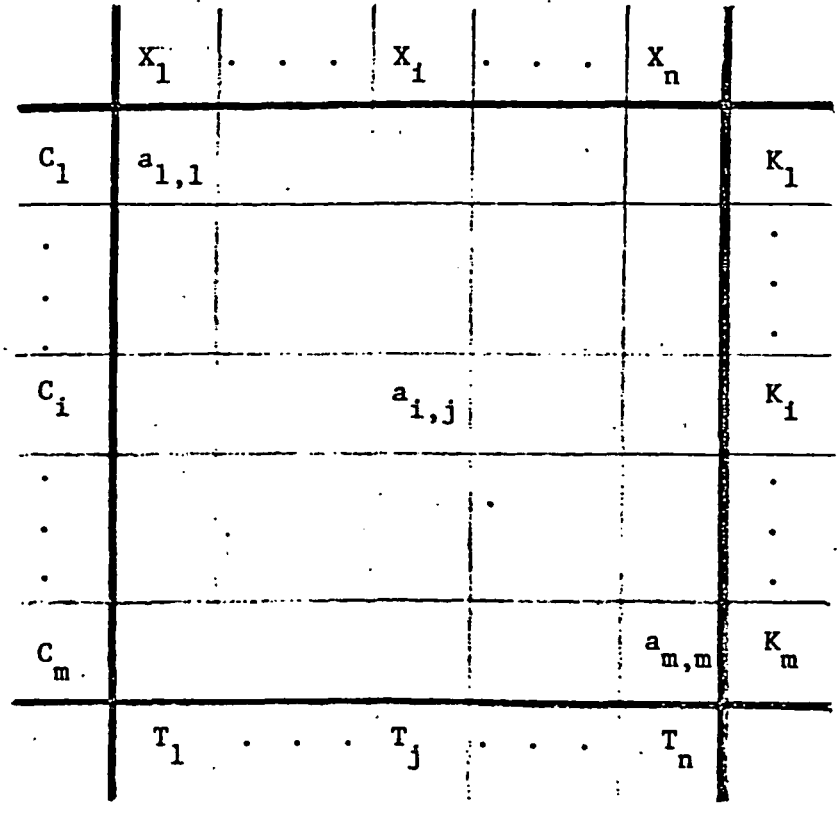
In the case where $a(i,j) \geq 0$ for all i,j, $\max(Y_j)$ is clearly equal to T_j, and the convergence criterion

$$\frac{Y_j}{\max(Y_j)} < \varepsilon$$

has the physical interpretation that all of element j within $\varepsilon \times 100\%$ can be accounted for; in cases where some $a(i,j) < 0$ (as is often encountered when solid phases are present, see Appendix 1) this physical interpretation is no longer valid, but the criterion for convergence certainly is.

1.7 Summary

The problem can be visualized in the following way



1. The C_i's are a function of the X_j's

$$C_i = K_i \prod_{j=1}^{n} X_j^{a(i,j)}: \quad I=1,m \text{ or } \bar{C} = \bar{f}(X_1,\ldots,X_n)$$

2. The Y_j's are a function of the C_i's

$$Y_j = \sum_{i=1}^{m} a(i,j)\, C_i - T_j: \quad j=1,n \text{ or } \bar{Y} = \bar{g}(C_1,\ldots,C_m)$$

3. Then $\bar{Y}$ is a composite function of $\bar{X}$.

$$\bar{Y} = \bar{g}(\bar{f}(x))$$

or

$$\begin{bmatrix} Y_1 = Y_1(X_1,\ldots X_N) \\ \\ Y_N = Y_N(X_1,\ldots X_N) \end{bmatrix}$$

4. Then the Jacobian of $\bar{Y}$ with respect to $\bar{X}$ is

$$\bar{\bar{Z}} = \begin{bmatrix} \left(\frac{\partial Y_1}{\partial X_1}\right) & \cdots & \\ \vdots & \ddots & \\ \left(\frac{\partial Y_N}{\partial X_1}\right) & & \left(\frac{\partial Y_N}{\partial X_N}\right) \end{bmatrix}$$

5. And the value of $\bar{X}^{N+1}$ (at the N+1 iteration) is found from the solution (by Gaussian elimination) to the matrix equation

$$\begin{bmatrix} Y_1^N \\ \vdots \\ Y_n^N \end{bmatrix}^t = \begin{bmatrix} \left.\frac{\partial Y_1}{\partial X_1}\right|_{\bar{X}^N} & & \\ & \ddots & \\ & & \left.\frac{\partial Y_N}{\partial X_N}\right|_{\bar{X}^N} \end{bmatrix} \begin{bmatrix} X_1^N - X_1^{N+1} \\ \vdots \\ X_n^N - X_n^{N+1} \end{bmatrix}$$

2. Transformation of Basis

In order to improve the efficiency of computation, the basis $\{X_1,\ldots,X_n\}$ can be transformed such that the resulting Z matrix appraoches a diagonal matrix. This transformation is carried out by substitution of one of the C_i, say C_{ii}, for one of the X_j, say X_{jj}, in the basis. C_{ii} then becomes basis element X_{n+1}.

2.1 Transformation of C_i's

In the basic problem

$$C_i = K_i \prod_{j=1}^{n} X_j^{a(i,j)}$$

Now solve for X_{jj} in terms of C_{ii}, and substitute this expression for X_{jj} in the C_i's.

a) Separate terms in C_{ii}

$$C_{ii} = K_{ii} \prod_{j=1}^{n} X_j^{a(ii,j)} = K_{ii} \prod_{\substack{j=1 \\ j\neq jj}}^{n} X_j^{a(ii,j)} X_{jj}^{a(ii,jj)}$$

b) Solve for X_{jj}

$$X_{jj} = C_{ii}^{\frac{1}{a(ii,jj)}} \cdot \left[K_{ii} \prod_{\substack{j=1 \\ j\neq jj}}^{n} X_j^{a(ii,j)} \right]^{\frac{-1}{a(ii,jj)}}$$

c) Substitute for X_{jj} in the expressions for the C_i's.

$$C_i = K_i \prod_{\substack{j=1 \\ j\neq jj}}^{n} X_j^{a(i,j)} \cdot \left[C_{ii}^{\frac{1}{a(ii,jj)}} \left[K_{ii} \prod_{\substack{j=1 \\ j\neq jj}}^{n} X_j^{a(ii,j)} \right]^{\frac{-1}{a(ii,jj)}} \right]^{a(i,jj)}$$

2.2 Transformation of the Y_j's

In the basic problem

$$Y_j = \sum_{i=1}^{m} a(i,j)\, C_i - T_j$$

Now solve the equation $Y_{jj} = 0$ for C_{ii} and substitute the expression for C_{ii} into the other Y_j's.

a) Separate terms in $Y_{jj} = 0$

$$Y_{jj} = \sum_{\substack{i=1 \\ i\neq ii}}^{m} a(i,jj)\, C_i + a(ii,jj)C_{ii} - T_{jj} = 0$$

b) Solve for C_{ii}

$$C_{ii} = -\Big(\sum_{\substack{i=1 \\ i\neq ii}}^{m} a(i,jj)C_i - T_{jj}\Big)/a(ii,jj)$$

c) Substitute for C_{ii} in the expressions for the Y_j's

$$Y_{\substack{j \\ j\neq jj}} = \sum_{\substack{i \\ i\neq ii}}^{m} a(i,j)C_i - T_j - \Big[\frac{a(ii,j)}{a(ii,jj)}\Big(\sum_{\substack{i=1 \\ i\neq ii}}^{m} a(i,jj)C_i - T_{jj}\Big)\Big]$$

2.3 Computation of Mass Balance for $C_{ii} \equiv X_{n+1}$

In the place of Y_{jj} (which is now trivial), a new Y_{n+1} which is based on the new basis element, $C_{ii} \equiv X_{n+1}$, must be computed:

$$Y_{n+1} = \frac{Y_{jj}}{a(ii,jj)} = \sum_{\substack{i=1 \\ i\neq ii}}^{m} \frac{a(i,jj)}{a(ii,jj)}\, C_i + C_{ii} - \frac{T_{jj}}{a(ii,jj)}$$

2.4 Formulation of Transformation

When the expressions of 2.1(c), 2.2(c) and 2.3 are rearranged, a similarity is seen which suggests a simple formula for the modification of the a , T, and K matrices.

eqn. 2.1(c)

$$C_i = K_i \cdot K_{ii}^{-\frac{a(ii,jj)}{a(ii,jj)}} \cdot \prod_{\substack{j=1 \\ j\neq jj}}^{n} X_j^{\left(a(i,j) - \frac{a(ii,j)\ a(i,jj)}{a(ii,jj)}\right)} \cdot C_{ii}^{\frac{a(i,jj)}{a(ii,jj)}}$$

eqn. 2.2(c)

$$Y_{\substack{j \\ j\neq jj}} = \sum_{\substack{i=1 \\ i\neq ii}}^{m} \left(a(i,j) - \frac{a(ii,j)\ a(i,jj)}{a(ii,jj)}\right) C_i - \left(T_j - \frac{a(ii,j)}{a(ii,jj)}\ T_{jj}\right)$$

eqn. 2.3

$$Y_{n+1} = \sum_{i=1}^{m} \frac{a(i,jj)}{a(ii,jj)}\ C_i - \frac{T_{jj}}{a(ii,jj)}$$

Now consider the following transformation

New	Old
$a'(i,j)$	$a(i,j) - \frac{a(ii,j)\ a(i,jj)}{a(ii,jj)}$
T_j'	$T_j - \frac{a(ii,j)}{a(ii,jj)} T_{jj}$
K_i'	$K_i \cdot K_{ii}^{-\frac{a(i,jj)}{a(ii,jj)}}$

and for the new basis element $X_{n+1} \equiv C_{ii}$

$a'(i,n+1)$	$\frac{a(i,jj)}{a(ii,jj)}$
T'_{n+1}	$\frac{T_{jj}}{a(ii,jj)}$

Then equations 2.1(c), 2.2(c) and 2.3 now read, in terms of the transformed variables,

$$C_{i \atop i=1,m} = K_i' \prod_{\substack{j=1 \\ j\neq jj}}^{n+1} X_j^{a'(i,j)}$$

$$Y_{j \atop j\neq jj \atop j=1,n+1} = \sum_{i=1}^{m} a'(i,j)\ C_i - T'_j$$

The above two equations are of course similar in form to the equations for the basic problem discussed in section I. Then it can be shown by induction that these transformations can be carried out successively any number of times.

3. Computation of Solids

The method for computing chemical equilibrium in the presence of solid phases will be shown as follows: Computation where one solid phase is present, introduction of a generalized notation, computation for two solid phases, and extension to an arbitrary number of solid phases.

Again we will denote the solid phase as C_{ii} (where C_{ii} = the activity of the solid phase $\equiv 1$), the amount of solid present as S_{ii}, and will eliminate basis element X_{jj} as the constraint $C_{ii} = 1$ is put on the system.

3.1 One Solid Phase

3.1.1 Eliminate X_{jj} via solid C_{ii}

In the basic problem

$$C_i = K_i \prod_{j=1}^{n} X_j^{a(i,j)}$$

Now since solid C_{ii} exists

$$1 = C_{ii} = K_{ii} \prod X_j^{a(ii,j)} = K_{ii} \prod_{\substack{j=1 \\ j \neq jj}}^{n} X_j^{a(i,j)} \cdot X_{jj}^{a(ii,jj)}$$

Solve for X_{jj}

$$X_{jj} = \left[K_{ii} \prod_{\substack{j=1 \\ j \neq jj}}^{n} X_j^{a(ii,j)} \right]^{-\frac{1}{a(ii,jj)}}$$

and substitute into the C_i's

$$C_i = K_i \prod_{\substack{j=1 \\ j\neq jj}}^{n} X_j^{a(i,j)} \cdot \left[K_{ii} \prod_{\substack{j=1 \\ j\neq jj}}^{n} X_j^{a(ii,j)} \right]^{-\frac{a(i,jj)}{a(ii,jj)}}$$

or rearranging terms (compare Section 2.4)

$$C_i = K_i \cdot K_{ii}^{\frac{-a(i,jj)}{a(ii,jj)}} \left[\prod_{\substack{j=1 \\ j\neq jj}}^{n} X_j^{a(i,j)} \cdot \frac{a(ii,j)\ a(i,jj)}{a(ii,jj)} \right]$$

3.1.2 Eliminate Y_{jj} via C_{ii}

In the basic problem

$$Y_j = \sum_{i=1}^{m} a(i,j)\ C_i - T_j$$

Now, since there exists an amount of solid S_{ii} then

$$Y_j = \sum_{\substack{i=1 \\ i\neq ii}}^{m} a(i,j)C_i + a(ii,j)S_{ii} - T_j$$

and in specific for Y_{jj},

$$Y_{jj} = \sum_{\substack{i=1 \\ i\neq ii}}^{m} a(i,jj)C_i + a(ii,jj)S_{ii} - T_{jj}$$

Set $Y_{jj} = 0$ and solve for S_{ii}

$$S_{ii} = -\left(\sum_{\substack{i=1 \\ i\neq ii}}^{m} a(i,jj)C_i - T_{jj} \right)/a(ii,jj)$$

Substitute S_{ii} into the Y_j's

$$Y_j = \sum_{\substack{i=1 \\ i\neq ii}}^{m} a(i,j)C_i - T_j$$

$$- a(ii,j) \{ \sum_{\substack{i=1 \\ i\neq ii}}^{m} a(i,jj)\, C_i - T_{jj} \}/a(ii,jj)$$

And rearranging terms (compare Section 2.4)

$$Y_j = \sum_{\substack{i=1 \\ i\neq ii}}^{m} \left(a(i,j) - \frac{a(ii,j)\, a(i,jj)}{a(ii,jj)} \right) C_i - \left(T_j - \frac{a(ii,j)}{a(ii,jj)}\, T_{jj}\right)$$

Now all of the Y_j's $\underset{j\neq jj}{}$ are independent of S_{ii}.

The problem has been reduced to an

$(n-1 \times n-1)$ problem in X and Y and a

$(\ 1 \times 1\ )$ problem in S and Y.

3.1.3 Solution for S_{ii}

After the $(n-1 \times n-1)$ problem has been solved for $\bar{X}$, $\bar{C}$ can be computed and S_{ii} is readily found from

$$S_{ii} = -\left(\sum_{\substack{i=1 \\ i\neq ii}}^{m} a(i,jj)C_i - T_{jj} \right)/a(ii,jj)$$

3.2 Solution for Numerous Solid Phases

The solution for numerous solid phases is simply an extension of that for one solid phase. With the computation of each solid phase the matrices are transformed successively, each time as before.

3.2.1 Review

To review, in the basic problem (no solids) a solution to the following set of equations was found:

$$Y_j = \sum_{i=1}^{m} a(i,j)\, C_i - T_j$$

where

$$C_i = K_i \prod_{j=1}^{n} X^{a(i,j)}$$

3.2.2 One Solid Phase

In the case of one solid phase the set of equations below (n-1 × n-1, 1 × 1) was solved:

$$\left[\begin{array}{l} Y_j = \sum_{\substack{i=1 \\ i\neq ii}}^{m} \left(a(i,j) - \frac{a(ii,j)\, a(i,jj)}{a(ii,jj)}\right) C_i - \left(T_j - \frac{a(ii,j)}{a(ii,jj)}\, T_{jj}\right) \\ j\neq jj \\ \\ \text{where} \\ \\ C_i = K_i \cdot K_i^{-\frac{a(i,jj)}{a(ii,jj)}} \prod_{\substack{j=1 \\ j\neq jj}}^{n} X_j^{a(i,j) - \frac{a(ii,j)\, a(i,jj)}{a(ii,jj)}} \\ i\neq ii \end{array} \right]$$

and

$$\left[S_{ii} = -\left(\sum_{\substack{i=1 \\ i\neq ii}}^{m} a(i,jj)\, C_i - T_{jj}\right)/a(ii,jj) \right]$$

3.2.3 Notation

For the case of many solids, it is necessary to introduce a new condensed notation:

for no solids:

new old

$$a^{\circ}(i,j) = a(i,j)$$

$$T^{\circ}_{j} = T_{j}$$

$$K_{i}^{\circ} = K_{i}$$

for one solid

$$a'(i,j) = a^{\circ}(i,j) - \frac{a^{\circ}(ii',j)\, a^{\circ}(i,jj')}{a^{\circ}(ii',jj')}$$

$$T_{j}' = T_{j}^{\circ} - \frac{a^{\circ}(ii',j)}{a^{\circ}(ii',jj')}\, T_{jj'}^{\circ}$$

$$K_{i}' = K_{i}^{\circ} - \frac{a^{\circ}(i,jj')}{a^{\circ}(ii',jj')}\, K_{ii'}^{\circ}$$

and in general

$$a^{\ell+1}(i,j) = a^{\ell}(i,j) - \frac{a^{\ell}(ii^{\ell+1},j)\, a^{\ell}(i,jj^{\ell+1})}{a^{\ell}(ii^{\ell+1}, jj^{\ell+1})}$$

$$T_{j}^{\ell+1} = T_{j}^{\ell} - \frac{a^{\ell}(ii^{\ell+1},j)}{a^{\ell}(ii^{\ell+1}, jj^{\ell+1})}\, T_{jj^{\ell+1}}^{\ell}$$

$$K_{i}^{\ell+1} = K_{i}^{\ell} - \frac{a^{\ell}(i,jj^{\ell+1})}{a^{\ell}(ii^{\ell+1}, jj^{\ell+1})}\, K_{ii^{\ell+1}}^{\ell}$$

where in every case (1,2,...ℓ+1...) solid $S_{ii^{\ell+1}}$, $C_{ii^{\ell+1}}$ and basis element $Y_{jj^{\ell+1}}$ are considered in defining $a^{\ell+1}$, $K^{\ell+1}$, $T^{\ell+1}$ in terms of a^{ℓ}, K^{ℓ}, and T^{ℓ}.

3.2.4 Condensed Notation: One Solid

Then the set of equations in part 3.2.2 (one solid) can be written in the new notation:

$$\left[\begin{array}{l} Y_{\substack{j \\ j\neq jj}} = \sum_{\substack{i=1 \\ i\neq ii'}}^{m} a'(i,j)\, C_i - T_j' \\ \\ \text{where} \\ \\ C_{\substack{i \\ i\neq ii}} = K_i' \prod_{\substack{j=1 \\ j\neq jj'}}^{n} X_j^{a'(i,j)} \end{array}\right] \qquad (n-1 \times n-1)$$

and

$$S_{ii'} = -\left(\sum_{\substack{i=1 \\ i\neq ii}}^{m} a^{\circ}(i,jj')\, C_i - T^{\circ}_{jj'} \right) / a^{\circ}(ii',jj')$$

3.2.5 Condensed Notation: Two Solids

The argument used in Section 3.1 can be used to consider two solids

$$\left[\begin{array}{l} Y_{\substack{j \\ y\neq jj^1,jj^2}} = \sum_{\substack{i=1 \\ i\neq ii^1,ii^2}}^{m} a^2(i,j)\, C_i - T_j^2 \\ \\ \text{where} \\ \\ C_{\substack{i \\ i\neq ii^1,ii^2}} = K_i^2 \prod_{\substack{j=1 \\ j\neq jj^1,jj^2}}^{n} X_j^{a^2(i,j)} \end{array}\right] \qquad (n-2 \times n-2)$$

and

$$\left[\begin{array}{l} S_{ii^2} = -\left(\sum_{\substack{i=1 \\ i \neq ii^1, ii^2}}^{m} a^1(i,jj^2)\, C_i - T'_{jj^2} \right) / a^1(ii^2, jj^2) \\ \\ S_{ii^1} = \left(\sum_{\substack{i=1 \\ i \neq ii^1}}^{m} a^\circ(i,jj^1)\, C_i - T^\circ_{jj^1} \right) / a^\circ(ii^1, jj^1) \end{array}\right] \quad (2 \times 2)$$

Note that in the expression for S_{ii^1}, the summation over i does not exclude ii^2; thus C_{ii^2} in the summation is equivalent to S_{ii^2}, or,

$$S_{ii^1} = -\left(\sum_{\substack{i=1 \\ i \neq ii^1, ii^2}}^{m} a^\circ(i,jj^1)\, C_i + a^\circ(ii^2, jj^2) S_{ii^2} - T^\circ_{jj^1} \right) / a^\circ(ii^1, jj^1)$$

3.3.6 General Expression for Solids

It is now possible to write the general expression for an arbitrary number of solids ns.

$$\left[\begin{array}{l} Y_{j \neq jj(1,\ldots ns)} = \sum_{\substack{i=1 \\ i \neq ii(1,\ldots ns)}}^{m} a^{ns}(i,j) C_i - T_j^{ns} \\ \\ \text{where} \\ \\ C_{i \neq ii(,\ldots ns)} = K_i^{ns} \prod_{\substack{j=1 \\ j \neq jj(1,\ldots ns)}}^{n} X_j^{a^{ns}(i,j)} \end{array}\right]$$

$$(n\text{-}ns \times n\text{-}ns)$$

and

$$\left[S_{ii^{\ell}}{}_{\ell=ns,\ldots 1} = -\left(\sum_{\substack{i=1 \\ i \neq ii^{(\ell \ldots 1)}}}^{n} a^{\ell-1}(i,jj^{\ell}) C_i - T_{jj^{\ell}}^{\ell-1} \right) / a^{\ell-1}(ii^{\ell}, jj^{\ell}) \right]$$

(ns × ns)

Appendix 2

Example of a Chemical Equilibrium Problem

A very simple problem will be solved in order to demonstrate the principles on which MINEQL functions. The chemical approach to the problem is emphasized.

The equilibrium composition of a solution to which 10^{-3} moles/liter of $CaCO_3$ has been added is to be determined. The aqueous phase and two possible solid phases will be considered, but the gas phase will not. The species to be considered in this computation and their respective stability constants are given in Table I. A schematic presentation of these data, similar to their representation in MINEQL, is given in Figure 1. The entities Ca^{2+}, H^+, and CO_3^{2-} have been chosen as components; every species can be written as a linear combination of these components. It is assumed initially that no solid phases are present. The solubility of each of the possible solids will be tested after the equilibrium speciation for the soluble species has been computed.

An initial uneducated guess for $[Ca^{2+}]$, $[H]$, and $[CO_3^{2-}]$ is made:

$$[Ca^{2+}]^\circ = 10^{-3}$$
$$[H^+]^\circ = 10^{-10}$$
$$[CO_3^{2-}]^\circ = 10^{-4}$$

and the concentration of each of the complexes is calculated:

$$[Ca^{2+}] = [Ca^{2+}] \cdot 10^{0.0} = 10^{-3.0}$$
$$[H^+] = [H^+] \cdot 10^{0.0} = 10^{-10.0}$$
$$[CO_3^{2-}] = [CO_3^{2-}] \cdot 10^{0.0} = 10^{-4.0}$$

$$[CaCO_3] = [Ca^{2+}]\,[CO_3^{2-}2]\cdot 10^{3.0} = 10^{-4.0}$$

$$[CaHCO_3^+] = [Ca^{2+}]\,[H^+]\,[CO_3^{2-}]\cdot 10^{11.6} = 10^{-5.4}$$

$$[CaOH^+] = [Ca^{2+}]\,[H^+]^{-1}\cdot 10^{-12.2} = 10^{-5.2}$$

$$[HCO_3^-] = [H^+]\,[CO_3^{2-}]\cdot 10^{10.2} = 10^{-3.8}$$

$$[H_2CO_3^*] = [H^+]^2\,[CO_3^{2-}]\cdot 10^{16.5} = 10^{-6.5}$$

$$[OH^-] = [H^+]^{-1}\cdot 10^{-14.0} = 10^{-4.0}$$

The "calculated analytical concentrations" of Ca^{2+}, H^+, and CO_3^{2-} as determined from the concentration of complexes is then computed:

$$[Ca^{2+}]_T^{calc} = [Ca^{2+}] + [CaCO_3] + [CaHCO_3] + [CaOH^+] = 1.11 \times 10^{-3}$$

$$[H^+]_T^{calc} = [H^+] + [CaHCO_3^+] - [CaOH^+] + [HCO_3^-] + 2[H_2CO_3^*] - [OH^-]$$
$$= 5.62 \times 10^{5}$$

$$[CO_3^{2-}]_T^{calc} = [CO_3^{2-}] + [CaCO_3] + [CaHCO_3^+] + [HCO_3^-] + [H_2CO_3^*]$$
$$= 3.62 \times 10^{-4}$$

Then the difference function between the imposed analytical concentration $[x]_T$ and the calculated $[x]_T^{calc}$ is evaluated:

$$Y_{Ca^{2+}} = [Ca^{2+}]_T^{calc} - [Ca^{2+}]_T = 1.10 \times 10^{-4}$$

$$Y_{H^+} = [H^+]_T^{calc} - [H^+]_T = 5.62 \times 10^{-5}$$

$$Y_{CO_3^{2-}} = [CO_3^{2-}]_T^{calc} - [CO_3]_T = 6.37 \times 10^{-4}$$

According to the Newton-Raphson Method, the difference functions (Y's) and their derivatives may be used to find improved values for $[Ca^{2+}]$, $[H^+]$, and $[CO_3^{2-}]$, such that the calculated analytical concentrations, $[x]_T^{calc}$, more closely approach the imposed analytical concentrations, $[x]_T$. If the initial guesses for Ca^{2+}, H^+, $CO_3 2-$ are denoted by $[x]^\circ$ and the improved values by $[x]^{calc}$, the Newton-Raphson Method for this case yields the three simultaneous equations:

$$1.\quad \frac{\partial Y_{Ca^{2+}}}{\partial [Ca^{2+}]} ([Ca^{2+}]^\circ - [Ca^{2+}]^{calc}) + \frac{\partial Y_{Ca^{2+}}}{\partial [H^+]} ([H^+]^\circ - [H^+]^{calc})$$

$$+ \frac{\partial Y_{Ca^{2+}}}{\partial [CO_3^{2-}]} ([CO_3^{2-}]^\circ - [CO_3^{2-}]^{calc}) = Y_{Ca^{2+}}$$

$$2.\quad \frac{\partial Y_{H^+}}{\partial [Ca^{2+}]} ([Ca^{2+}]^\circ - [Ca^{2+}]^{calc}) + \frac{\partial Y_{H^+}}{\partial [H^+]} ([H^+]^\circ - [H^+]^{calc})$$

$$+ \frac{\partial Y_{H^+}}{\partial [CO_3^{2-}]} ([CO_3^{2-}]^\circ - [CO_3^{2-}]^{calc}) = Y_{H^+}$$

$$3.\quad \frac{\partial Y_{CO_3^{2-}}}{\partial [Ca^{2+}]} ([Ca^{2+}]^\circ - [Ca^{2+}]^{calc}) + \frac{\partial Y_{CO_3^{2-}}}{\partial [H^+]} ([H^+]^\circ - [H^+]^{calc})$$

$$+ \frac{\partial Y_{CO_3^{2-}}}{\partial [CO_3^{2-}]} ([CO_3^{2-}]^\circ - [CO_3^{2-}]^{calc}) = Y_{CO_3^{2-}}$$

The derivatives are readily calculated:

$$\frac{\partial Y_{Ca^{2+}}}{\partial [Ca^{2+}]} = 1 + \frac{[CaCO_3]}{[Ca^{2+}]} + \frac{[CaHCO_3^+]}{[Ca^{2+}]} + \frac{[CaOH^+]}{[Ca^{2+}]} = 1.11$$

$$\frac{\partial Y_{Ca^{2+}}}{\partial [H^+]} = \frac{[CaHCO_3]}{[H^+]} - \frac{[CaOH^+]}{[H^+]} = -2.33 \times 10^4$$

$$\frac{\partial Y_{Ca^{2+}}}{\partial [CO_3^{2-}]} = \frac{[CaCO_3]}{[CO_3^{2-}]} + \frac{[CaHCO_3^+]}{[CO_3^{2-}]} = 1.04$$

$$\frac{\partial Y_{H^+}}{\partial [Ca^{2+}]} = \frac{[CaHCO_3^+]}{[Ca^+]} - \frac{[CaOH^+]}{[Ca^+]} = -2.33 \times 10^{-3}$$

$$\frac{\partial Y_{H^+}}{\partial [H^+]} = 1 + \frac{[CaHCO_3^+]}{[H^+]} + \frac{[CaOH^+]}{[H^+]} + \frac{[HCO_3^-]}{[H^+]} + \frac{4[H_2CO_3^*]}{[H^+]} + \frac{[OH^-]}{[H^+]}$$

$$= 2.67 \times 10^6$$

$$\frac{\partial Y_{H^+}}{\partial [CO_3^{2-}]} = \frac{[CaHCO_3^+]}{[CO_3^{2-}]} + \frac{[HCO_3^-]}{[CO_3^{2-}]} + \frac{2[H_2CO_3^*]}{[CO_3^{2-}]} = 1.63$$

$$\frac{\partial Y_{CO_3^{2-}}}{\partial [Ca^{2+}]} = \frac{[CaCO_2]}{[Ca^{2+}]} + \frac{[CaHCO_3^+]}{[Ca^{2+}]} = 0.104$$

$$\frac{\partial Y_{CO_3^{2-}}}{\partial [H^+]} = \frac{[CaHCO_3^+]}{[H^+]} + \frac{[HCO_3^-]}{[H^+]} + \frac{2[H_2CO_3^*]}{[H^+]} = 1.63 \times 10^6$$

$$\frac{\partial Y_{CO_3^{2-}}}{\partial [CO_3^{2-}]} = 1 + \frac{[CaCO_3]}{[CO_3^{2-}]} + \frac{[CaHCO_3^+]}{[CO_3^{2-}]} + \frac{[HCO_3^-]}{[CO_3^{2-}]} + \frac{[H_2CO_3^*]}{[CO_3^{2-}]} = 3.62$$

Then the system of simultaneous equations

$$1.11 \quad x_1 - 2.33 \times 10^4 \quad x_2 + 10.4 \quad x_3 = 1.10 \times 10^{-4}$$

$$-2.33 \times 10^3 \quad x_1 + 2.67 \times 10^6 \quad x_2 + 1.62 \quad x_3 = 5.62 \times 10^{-5}$$

$$0.104 \quad x_1 + 1.63 \times 10^6 \quad x_2 + 3.62 \quad x_3 = 6.37 \times 10^{-4}$$

is solved for x_1, x_2, and x_3 where

$$x_1 \rightarrow [Ca^{2+}]^\circ - [Ca^{2+}]^{calc} = 3.54 \times 10^{-4}$$

$$x_2 \rightarrow [H^+]^\circ - [H^+]^{calc} = 1.83 \times 10^{-10}$$

$$x_3 \rightarrow [CO_3^{2-}]^\circ - [CO_3^{2-}]^{calc} = 2.68 \times 10^{-4}$$

Then the known values for $[Ca^{2+}]^\circ$, $[H^+]^\circ$, and $[CO_3^{2-}]^\circ$ are substituted in the equations above and

$$[Ca^{2+}]^{calc} = 6.45 \times 10^{-4}$$

$$[H^+]^{calc} = -8.33 \times 10^{-4}$$

$$[CO_3^{2-}]^{calc} = 3.68 \times 10^{-4}$$

It is seen that the new value for $[H^+]^{calc} < 0$; this is physically meaningless. Therefore, by use of an empirical algorithm, $[H^+]^{calc}$ is set to $[H^+]^{calc} = [H^+]^\circ/10$, and the iterative process is resumed with the values $[Ca^{2+}] = 6.45 \times 10^{-4}$

$$[H^+] = 10^{-11}$$

$$[CO_3^{2-}] = 3.68 \times 10^{-4}$$

This iterative process is carried out six times until the value of each difference function is small compared to the terms in the difference function. (The actual convergence criterion is that Y divided by the maximum of individual terms in Y is less than ε; here $\varepsilon = 10^{-4}$).

$$Y_{Ca^{2+}} = [Ca^{2+}]_T^{calc} - [Ca^{2+}]_T$$

$$= [Ca^{2+}] + [CaCO_3] + [CaHCO_3^+] + [CaOH^+] - [Ca^{2+}]_T$$

$$= 6.87 \times 10^{-4} + 2.97 \times 10^{-4} + 4.58 \times 10^{-6} + 1.12 \times 10^{-5} - 10^{-3}$$

$$= -2.29 \times 10^{-10}$$

$$Y_{H^+} = [H^+]_T^{calc} - [H^+]_T$$

$$= [H^+] + [CaHCO_3^+] - [CaOH^+] + [HCO_3^-] + 2[H_2CO_3^*] - [OH^-] - [H^+]_T$$

$$= 3.87 \times 10^{-4} + 4.58 \times 10^{-6} - 1.12 \times 10^{-5} + 2.65 \times 10^{-4} + 4.09 \times 10^{-8}$$
$$- 2.59 \times 10^{-4} - 0$$

$$= 4.07 \times 10^{-10}$$

$$Y_{CO_3^{2-}} = [CO_3^{2-}]_T^{calc} - [CO_3^{2-}]_T$$

$$= [CO_3^{2-}] + [CaCO_3] + [CaHCO_3^+] + [HCO_3^-] + [H_2CO_3^*] - [CO_3^{2-}]_T$$

$$= 4.33 \times 10^{-4} + 2.97 \times 10^{-4} + 4.58 \times 10^{-6} + 2.65 \times 10^{-4}$$
$$+ 2.05 \times 10^{-8} - 10^{-3}$$

$$= -2.02 \times 10^{-10}$$

The full solution to this problem is given in the first column of Table II.

Now it is necessary to compute the solubility of the solid phases:

$$Ca(OH)_2 \quad (s) \quad [Ca^{2+}]\,[H]^{-2} \cdot 10^{-21.9} = 10^{-4.24}$$

$$CaCO_3 \quad (s) \quad [Ca^{2+}]\,[CO_3] \cdot 10^{8.3} = 10^{+1.78}$$

It is seen that the solubility product of $CaCO_3$ (s) is exceeded, i.e. $[Ca^{2+}]\ [CO_3^{2-}]\ K > 1$; the problem must be modified according to the description given in Appendix I to include the solid phase $CaCO_3$ (s). The schematic representation of the "modified" problem is given in Figure 2. Note that by modification the basis element CO_3^{2-} has effectively been eliminated from the computation; now the computation includes only Ca^{2+} and H^+.

The previously calculated values for $[Ca^{2+}]$ and $[H^+]$ are used as initial guesses, and the modified problem is solved as before for the soluble species. The computation converges after five iterations, and the results are given in Table II. The amount of $CaCO_3$ (s) is calculated from

$$[CO_3^{2-}]_T = [CO_3^{2-}] + [CaCO_3] + [CaHCO_3^+] + [HCO_3^-] + [H_2CO_3^*] + CaCO_3\ (s).$$

$CaCO_3$ (s) is found to be 8.73×10^{-4}; this is checked to insure that it is greater than 0.

The solubility of $Ca(OH)_2$ (s) is then checked:

$$[Ca^{2+}]\ [H^+]^{-1} \cdot 10^{-6.00} < 1$$

and found to be soluble.

The problem is solved.

Figure 1. Schematic Representation of Data

Calcium Carbonate Problem

		Ca^{2+}	H^+	CO_3^{2-}	log K
1.	Ca^{2+}	1	0	0	0.0
2.	H^+	0	1	0	0.0
3.	CO_3^{2-}	0	0	1	0.0
4.	$CaCO_3(aq)$	1	0	1	3.0
5.	$CaHCO_3^+$	1	1	1	11.6
6.	$CaOH^+$	1	-1	0	-12.2
7.	HCO_3^-	0	1	1	10.2
8.	H_2CO_3	0	2	1	16.5
9.	OH^-	0	-1	0	-14.0
10.	$Ca(OH)_2(s)$	1	-2	0	-21.9
11.	$CaCO_3(s)$	1	0	1	8.3
		$[Ca^{2+}]_T$	$[H^+]_T$	$[CO_3^{2-}]_T$	
		10^{-3}	0	10^{-3}	

The analytical concentrations according to this representation are

$$[Ca^{2+}]_T = 10^{-3},\ [H^+]_T = 0,\ [CO_3^{2-}]_T = 10^{-3}$$

Figure 2 Schematic Representation of Data for Calcium Carbonate Problem: Modified for $CaCO_3(s)$

		Ca^{2+}	H^+	log K	CO_3^{2-}
1.	Ca^{2+}	1	0	0.0	0
2.	H^+	0	1	0.0	0
3.	CO_3^{2-}	-1	0	-8.3	1
4.	$CaCO_3$	0	0	-5.3	1
5.	$CaHCO_3^+$	0	1	3.3	1
6.	$CaOH^+$	1	-1	-12.2	0
7.	HCO_3^-	-1	1	1.9	1
8.	H_2CO_3	-1	2	8.2	1
9.	OH^-	0	-1	-14.0	0
10.	$Ca(OH)_2(s)$	1	-2	-21.9	0
11.	$CaCO_3(s)$	1	0	8.3	1
		$[Ca^{2+}]_T$	$[H^+]_T$		
		0.0	0.0		

TABLE I Species Considered in Carbonate Problem

1.	Ca^{2+}		
2.	H^{+}		
3.	CO_3^{2-}		
4.	$CaCO_3$	$[Ca^{2+}]\ [CO_3^{2-}]K = [CaCO_3]$	log K = 3.0
5.	$(CaHCO_3)^{+}$	$[Ca^{2+}]\ [H^{+}]\ [CO_3^{2-}]K = [CaHCO_3^{+}]$	log K = 11.6
6.	$(CaOH)^{+}$	$[Ca^{2+}]\ [H^{+}]^{-1}K = [CaOH^{+}]$	log K = -12.2
7.	HCO_3^{-}	$[H^{+}]\ [CO_3^{2-}]K = [HCO_3^{-}]$	log K = 10.2
8.	H_2CO_3	$[H^{+}]^2\ [CO_3^{2+}]K = [H_2CO_3^{*}]$	log K = 16.5
9.	OH^{-}	$[H^{+}]^{-1}\ K = [OH^{-}]$	log K = -14.0
10.	$Ca(OH)_2(s)$	$[Ca^{2+}]\ [H^{+}]^{-2}K = [Ca(OH)_2(s)]$	log K = -21.9
11.	$CaCO_3(s)$	$[Ca^{2+}]\ [CO_3^{2-}]K = [CaCO_3(s)]$	log K = 8.3

The following conditions are assumed: temperature = 25°C, ionic strength $\mu = 0$, concentration ≡ activity

TABLE II Solution of the Calcium Carbonate Problem

Species	Solution 1 no solid phases -log[x]	Solution 2 $CaCO_3(s)$ present -log[x]
Ca^{2+}	3.16	3.91
H^+	10.41	9.91
CO_3^{2-}	3.36	4.38
$CaCO_3$	3.53	5.30
$CaHCO_3^+$	5.34	6.61
$CaOH^+$	4.95	6.21
HCO_3^-	3.58	4.09
$H_2CO_3^*$	7.69	7.70
OH^-	3.59	4.09
$CaCO_3(s)$	-	3.13
$Ca(OH)_2(s)$	-	-

Appendix 3

Description of Subroutines*

1. SUBROUTINE ERROR
2. SUBROUTINE EXCOL

   ENTRY EXROW
3. FUNCTION IADY

   ENTRY IADX
4. SUBROUTINE INION

   ENTRY IONCOR
5. SUBROUTINE INPUT

   ENTRY INTYPE
6. SUBROUTINE OUTPUT

   ENTRY OINCMP

   ENTRY OINSPC

   ENTRY OUTCMP

   ENTRY OUTSPC

   ENTRY OUTPC
7. SUBROUTINE SIMQ
8. SUBROUTINE SOLID
9. SUBROUTINE SOLIDX
10. SUBROUTINE SOLVE

* This section of the report is useful only when a listing of the program is at hand. Because of its length, the program has not been included as part of the text, but it should be available as computer output.

1. SUBROUTINE ERROR(I)

This subroutine prints error message I, prints output data (OUTCMP, OUTSPC) and terminates execution. Description of the error messages is found in the MINEQL User's Manual.

2. SUBROUTINE EXCOL (JO,JJ), EXROW (IO,II)

EXCOL exchanges columns in the IDX, A, X, GX and T arrays; EXROW exchanges rows in the IDY, A, and K arrays.

3. FUNCTION IADY(IDY), ENTRY IADX(IDX)

IADY finds the address (storage location) of species IDY in the K array; if species IDY cannot be found, an error message is issued and execution is terminated. IADX finds the address of component IDX in the X, GX, and T arrays; if component IDX cannot be found, an error message is issued and execution is terminated.

4. SUBROUTINE INION, ENTRY IONCOR

IONCOR performs ionic strength correction according to the Davies (14) approximation. INION reads the ionic charge data and initializes the variables for IONCOR.

Mathematical Treatment: The Davies approximation for the activity coefficient f of an ion in a medium of ionic strength μ is

$$\log f = Z^2 \varepsilon T \left(\frac{\mu^{1/2}}{1 + \mu^{1/2}} - 0.2\mu\right)$$

where:

Z = ionic charge

ε = dielectric constant of solution

T = temperature

μ = ionic strength, is defined by:

$$\mu = \frac{1}{2} \sum_i Z_i^2 C_i \qquad i: \text{ all species in solution}$$

where:

Z_i = ionic charge of species i

C_i = concentration of species i

The thermodynamic equilibrium constant K for a reaction

$$a_1 X_1 + \ldots + a_n X_n = C$$

where X_j = reactant

a_j = stoichiometric coefficient of X_j

is defined by

$$\{X_1\}^{a1} \ldots \{X_n\}^{a_n} \cdot K = \{C\}$$

where { } represents activity. This can also be written

$$[X_1]^{a1} f_1^{a1} \ldots [X_n]^{a_n} f_n^{a_n} \cdot K = [C] f_c$$

where [ ] represents concentration and f the activity coefficient.

Upon rearrangement,

$$[X_1]^{a1} \ldots [X_n]^{a_n} \cdot \frac{f_1^{a1} \ldots f^{a_n}}{f_c} K = [C]$$

Then we can define a constant "corrected for ionic strength"

$$K' = \frac{f_1^{a1} \ldots f_n^{a_n}}{f_c} K$$

Using the fact that $Z_c = \sum_j a_j Z_j$ and the expression for log f given above, we can express the corrected constant

$$\log K' = [\sum_j a_j Z_j^2 - (\sum_j a_j Z_j)^2] \cdot \epsilon T \left(\frac{\mu^{1/2}}{1 + \mu^{1/2}} - 0.2\mu\right) + \log K$$

Description of INION and IONCOR:

INION initializes GF0 = 0, XMU = 0, and reads the data for ionic charge IONZ(J). See below for a description of program variables in terms of the previously defined mathematical variables.

IONCOR sets ET = 0.5 (valid for water at 25°C) and computes GF for the new ionic strength. The difference between the ionic strength functions for the new and the old ionic strengths is computed, DGF = GF - GFØ, and this difference is used to compute the GK's for the new ionic strength relative to the GK's for the old ionic strength. IONCOR prints the value of the ionic strength XMU, and the log f for an ion with Z = 1.

To make ionic strength corrections in MINEQL, INION is called first and only one time; IONCOR (XMU) can then be called any number of times with any desired ionic strength XMU.

Description of Variables

CF,GFØ: $\varepsilon T \left(\frac{\mu^{1/2}}{1 + \mu^{1/2}} - 0.2\mu \right) = \log f/Z^2$

XMU: μ

IONZ(J): Z_j

ET: εT

5. SUBROUTINE INPUT, ENTRY INTYPE

INPUT performs the following task: 1)reads the data to specify the components to be included in the computation, nn initial guess for their free concentrations, and their analytical (total) concentrations; 2) retrieves the relevant thermodynamic data for that set of components;

3) reads the data to specify species types.

The loop on 20 initializes the IADXT array. IADXT(IDX) contains the array storage location of the component identified by IDX. Thus as each component, the guess for its concentration, and its analytical concentration are read in statement 10, IADXT(IDX) is set equal to the index J and the data are stored in IDX(J), GX(J), X(J) and T(J). All other elements in IADX remain equal to zero.

The loop on 200 then generates the Type I species in the A and K matrices. The loop on 400 reads each of the species in the thermodynamic data base. If all of the components of that species are included in the problem (as indicated by a non-zero entry in the IADXT array), the species is entered by species type into the A and K arrays.

The manipulation of species type is described in the MINEQL User's Manual. The ENTRY INTYPE allows for the manipulation of species type from MAIN after the initial specification has been made. If the type or stoichiometry data is entered incorrectly an error message is issued and execution is terminated.

6. SUBROUTINE OUTPUT

OUTPUT has five separate entry points, each of which produces a particular output, as described in the MINEQL User's Manual. Each of the output functions are called by their ENTRY names; a call to OUTPUT is meaningless. The programming is straightforward and will not be described further here.

7. SUBROUTINE SIMQ(Z,Y,N,NXDIM)

This is a subroutine for the solution of n linear equations in n unknowns by Gaussian elimination and each substitution. The methodology is discussed in many numerical analysis texts, e.g. (13). If the system is singular, an error message is issued and execution is terminated. For the solution of linear equations the arguments are:

Z - two dimensional array containing the matrix of coefficients in rows and columns 1 through N. (Z is written over during computation).

Y - one dimensional array containing the values of the non-homogeneous equations in rows 1 through N: the root of the equation is returned in Y.

N - the order of the matrix stored in Z

NXDIM - the actual dimension of Z

8. SUBROUTINE SOLID

SOLID modifies the A, K, and T matrices for the set of precipitated solids (Figure 1). SOLID issues an immediate RETURN if there are no precipitated solids (Type III and Type IV species). If there are precipitated solids, modification begins with the last Type IV species as the pivotal species, and works upward to the first Type III species.

To modify for a given solid species, the row in the A matrix which represents that species is searched for a non-zero A(I,J): the component responsible for the non-zero A(I,J) is to be used as the pivotal component. (If no non-zero coefficient can be found, the situation corresponds to a violation of the Gibbs Phase Rule. An error message is issued and

execution is terminated). When the pivotal component has been found, the column of the pivotal component and the rightmost column of the A matrix (or, more precisely, the rightmost column of that part of the A matrix which does not already contain pivotal components) are interchanged then the pivotal row and pivotal column are on the outside of the A matrix and modificaiton can conveniently be performed on the inside elements.

The mathematics of the modification operation are described in Appendix 1, "Computation of Solids".

9. SUBROUTINE SOLIDX(KK)

The function of SOLIDX is to solve the chemical equilibrium problem for the amount of solids, "unmodify" the A, K, T, GX and X matrices (i.e., restore the stoichiometry to reflect the original set of components) and to dissolve or to precipitate solids as necessary (Figure 2).

The loop on 440 computes the mass balance equations for those constituents which were eliminated from the computation as the arrays were modified for the solids. The loop on 460 solves for the amount of solid, and performs the inverse of the modification procedure used in subroutine SOLID. The loop on 220 computes the solubility expression for all of the dissolved solids.

Then the precipitated solids subject to dissolution (Type IV species) are tested for a negative amount. If the amount of a solid is found to be negative, or if more than one solid is present in a negative amount, the solid with the most negative amount is dissolved and KK, the return code of the subroutine, is set to -1 to indicate dissoluation.

Dissolution is actually accomplished by moving the solid to be dissolved from the range of the Type IV species to the range of the Type V species. Execution returns to MAIN and the chemical equilibrium problem with this new set of solids is restarted at subroutine SOLID.

A similar procedure is used to test dissolved solids for precipitation: the solid which most exceeds its solubility product is precipitated and return code KK = 1 is used to indicate precipitation.

By testing the solids first for dissolution, and dissolving solids one at a time, then testing for precipitation and precipitating the solids one at a time, the program minimizes the chance of creating a system which violates the Gibbs Phase Rule.

If solids are neither precipitated nor dissolved by SOLIDX, return code KK = 0 is given and the equilibrium problem is solved.

10. SUBROUTINE SOLVE

Subroutine SOLVE solves the chemical equilibrium problem for soluble species, performing all of the operations described in Appendix 1, "Basic Problem". SOLVE is indifferent to whether solid phases have modified the chemical equilibrium problem or not - it simply solves the problem for the given constraints.

The loop on 2 computes the concentration of the soluble species; the loop on 201 computes the mass balance equations; the loop on 400 computes the Jacobian; and the loop on 800 checks for convergence.

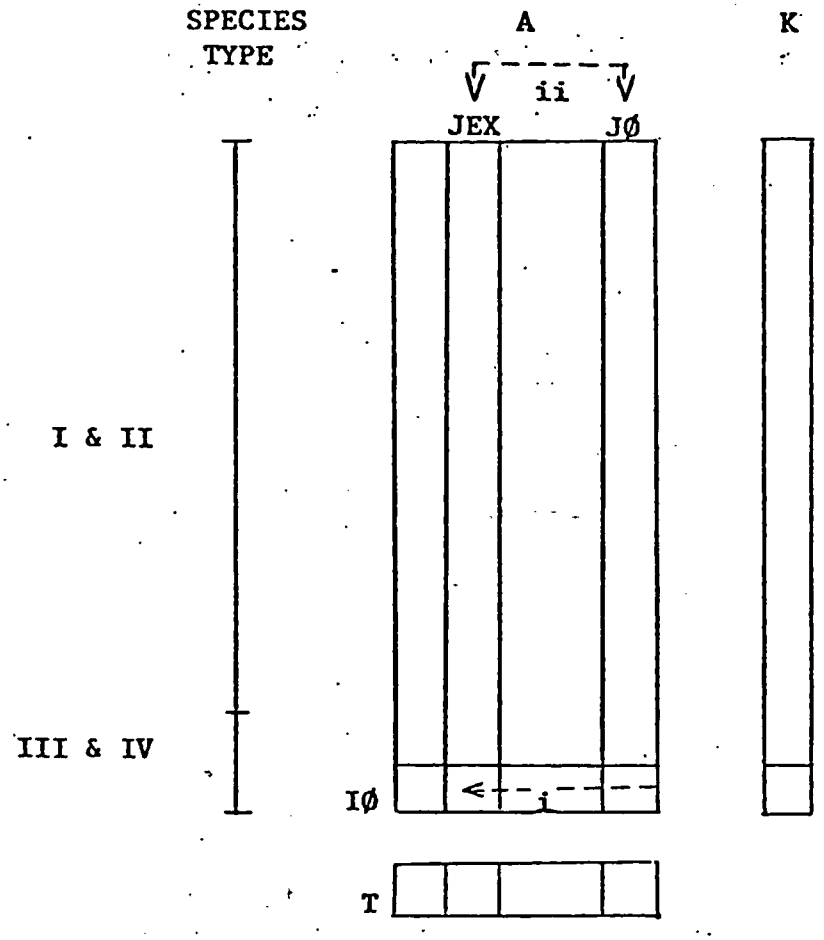
If the number of iterations is greater than the maximum allowable, an error message is issued and execution is terminated. Otherwise the system of linear equations given by the Jacobian, Z, and the mass

balance equations, Y, is solved by SIMQ, and the solution is returned in Y. The "improved" values for the components are computed (X(J) = X(J) - Y(J)) and checked whether they are greater than zero. If the "improved" value is found to be less than zero, the "improved" value is set equal to the old value divided by ten (X(J) = (X(J) + Y(J))/10) by an empirical algorithm.

Execution then reverts to statement 1000 for another iteration.

Figure 1 Modification of A,K & T for Solids*

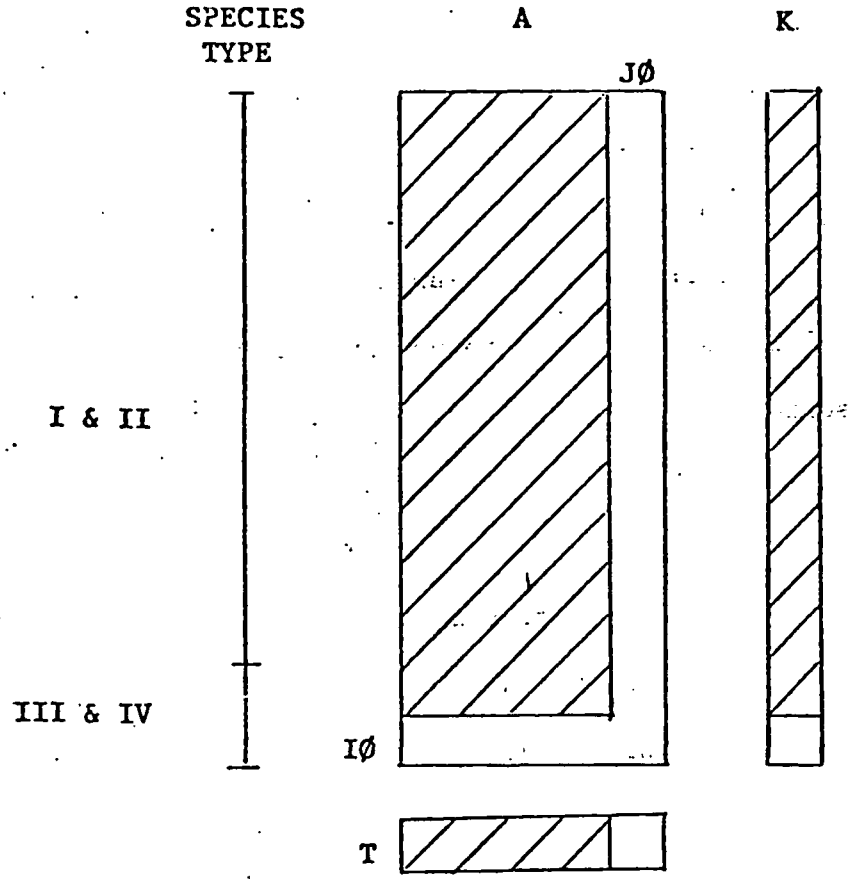
(Subroutine SOLID)



1.a. Exchange columns

i. Find non-zero element in row IØ of A array: call its column JEX

ii. Interchange columns JEX and JØ



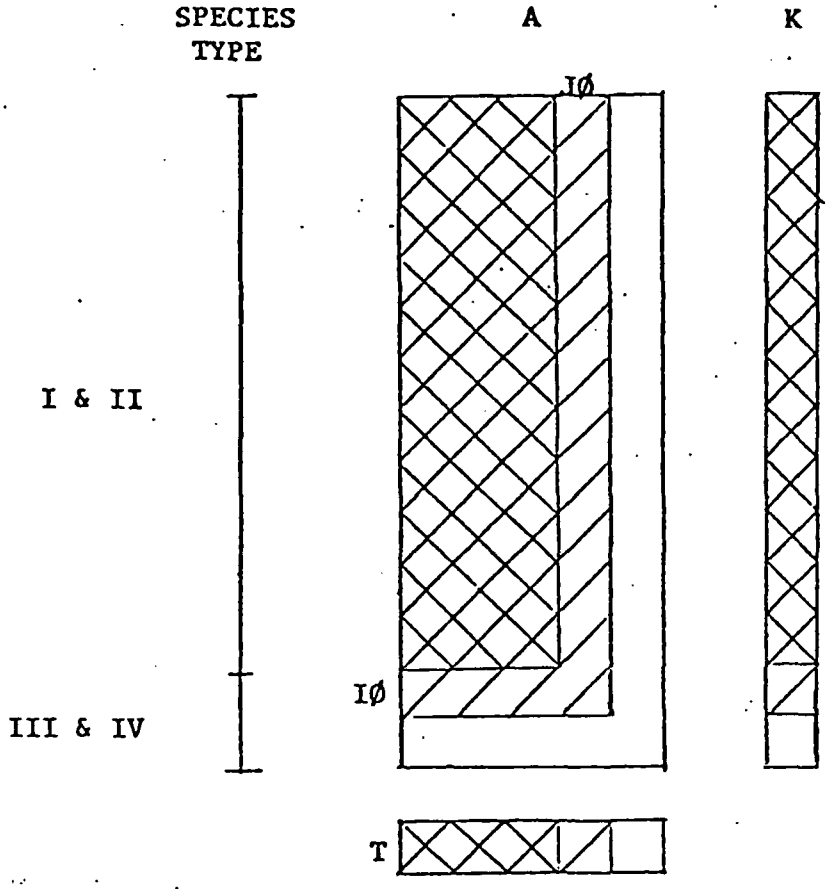
1.b. Modify interior elements of A, K, & T arrays (see Appendix 1).

The hatch pattern denotes those array elements which have been modified according to Appendix 1:

$$A(I,J) = A(I,J) - \frac{A(I\emptyset,J)\ A(I,J\emptyset)}{A(I\emptyset),J\emptyset)}$$

$$T(J) = T(J) - \frac{A(I\emptyset,J)}{A(I\emptyset,J\emptyset)}\ T(J\emptyset)$$

$$K(J) = K(J) - \frac{A(I,J\emptyset)}{A(I\emptyset,J\emptyset)}\ K(I\emptyset)$$



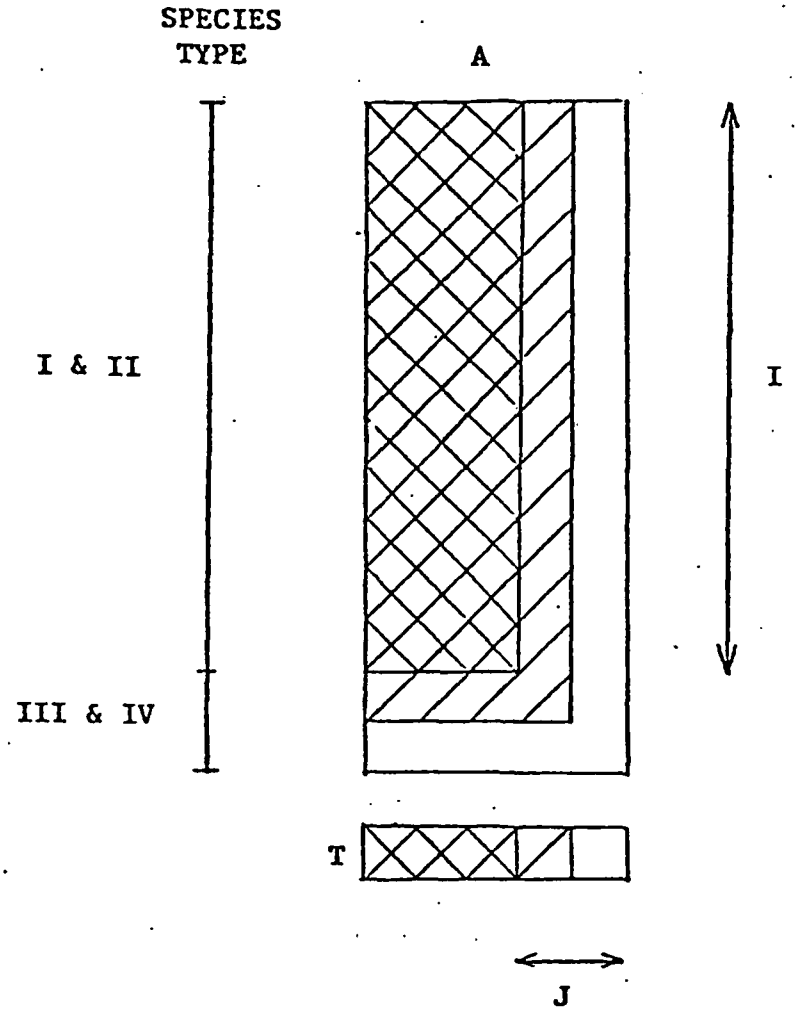
1.c Increment IØ and JØ, and repeat steps a & b.

array elements modified for first solid phase ▨ (preceding step)

array elements modified for second solid phase ▧ (this step)

Figure 2 Un-modification of A,K, & T*

(Subroutine SOLIDX)



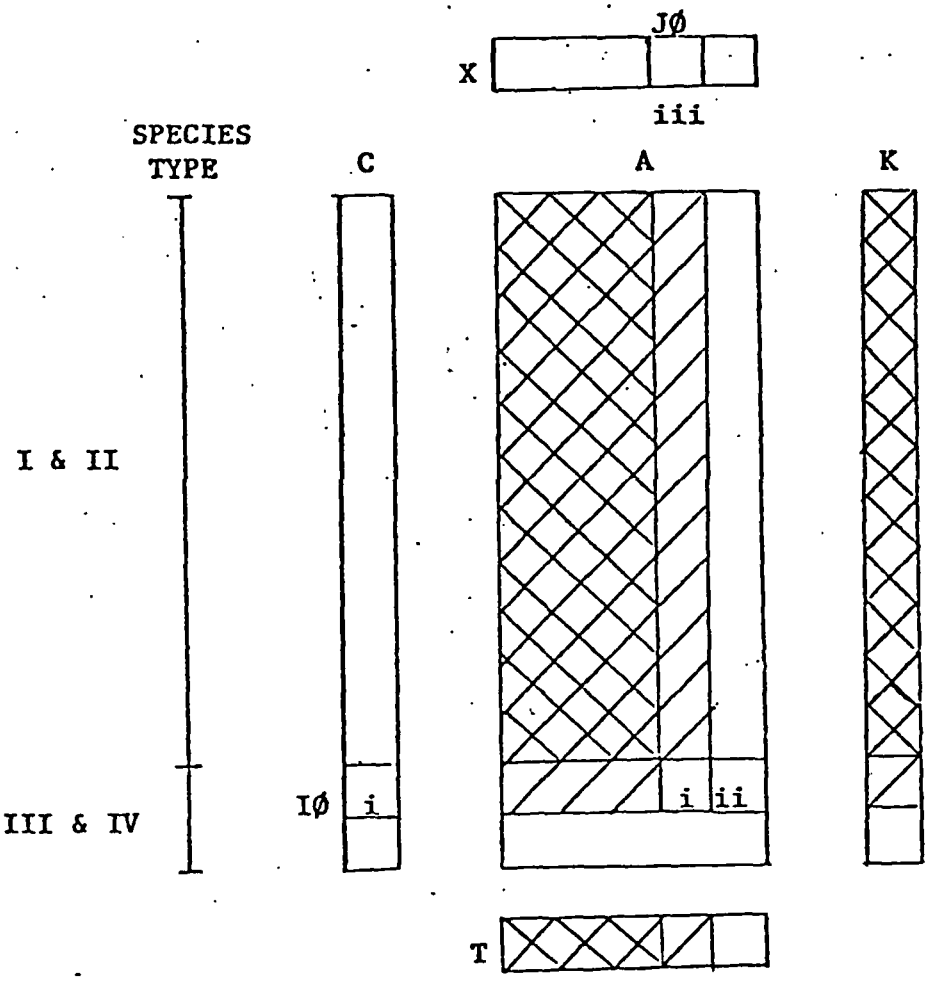
2.a Compute mass balance (y) for constituents which were eliminated from computation during modification for solids:

$$Y(J) = \sum_{I=\text{Type I}}^{\text{Type II}} A(I,J)\, C(I) - T(J)$$

The range of the indicies I and J are given by the double headed arrows.

Array elements modified for first solid phase ⊘

Array elements modified for second solid phase ⊘



2.b Solve for constituent in column JØ

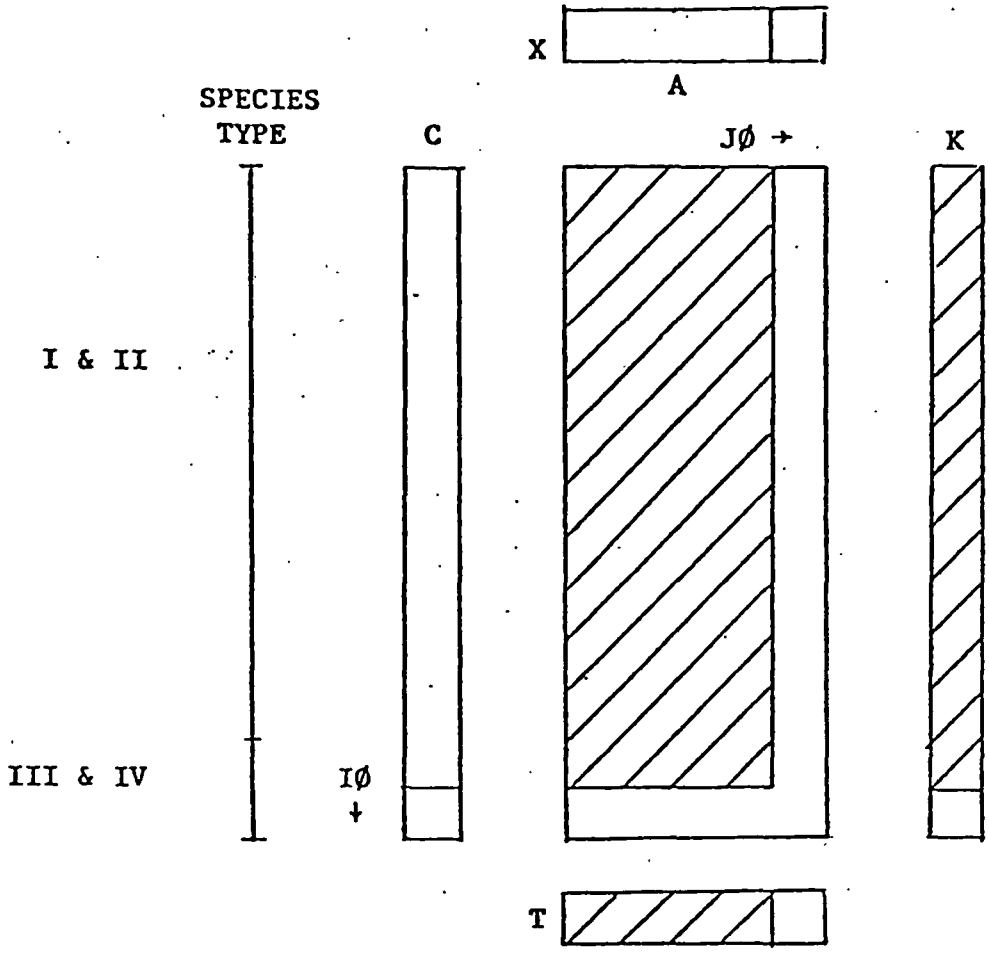
i. Solve for amount of solid in row IØ from mass balance condition in column JØ: $C(I\,\emptyset) = Y(J\emptyset)/A(I\emptyset,J\emptyset)$

ii. Add amount of solid in row IØ to mass balance equations in columns to the right of current JØ: $Y(J) = Y(J)+A(I\emptyset,J)\ C(I\emptyset)$

iii. Solve for concentration (activity) of constituent in column JØ

Array elements modified for first solid phase ⧄

Array elements modified for second solid phase ⧅



2.c Complete un-modification

i. Un-modify interior elements of A, K, & T according to Appendix 1:

$$A(I,J) = A(I,J) + \frac{A(I\emptyset,J)\ A(I,J\emptyset)}{A(I\emptyset,J\emptyset)}$$

$$T(J) = T(J) + \frac{A(I\emptyset,J)}{A(I\emptyset,J\emptyset)}\ T(J\emptyset)$$

$$K(I) = K(I) + \frac{A(I,J\emptyset)}{A(I\emptyset)J\emptyset)}\ K(I\emptyset)$$

ii. Increment IØ and JØ and repeat a & b

Array elements modified for first solid phase ⊠

* The specific examples shown in the figures are only representative of the manipulations performed: the exact number of components and species of each type will of course vary from case to case.

References

1. Frank J. Zeleznik and Sanford Gordon, Industrial and Engineering Chemistry, 60, 27, (1968).

2. N.Z. Shapiro and L.S. Shapley, "Mass Action Laws and the Gibbs Free Energy Function", RM-3935-PR, The Rand Corporation, Dec., 1960.

3. R.J. Clason, "The Numerical Solution of the Chemical Equilibrium Problem", RM-4345-PR, The Rand Corporation, January, 1965.

4. Marian Shapley and Leola Cutter, "Rand's Chemical Composition Program: A Manual", R-495-PR, The Rand Corporation, June, 1970.

5. N. Ingri, W. Kakolowicz, L.G. Sillén, B. Warnquist, Talanta, 14, 1261, (1967).

6. D.D. Perrin and I.G. Sayce, Talanta, 14, 833, (1967).

7. F.M.M. Morel and J.J. Morgan, Environmental Science and Technology, 6, 58 (1972).

8. R.E. McDuff and F.M.M. Morel, Technical Report EQ73-02, Keck Laboratory, Caltech, 1973.

9. F.Morel, R.E. McDuff, and J.J. Morgan, in Trace Metals and Metal-Organic Interactions in Natural Waters, Ann Arbor Science Publishers, 1973.

10. F. Morel, R. McDuff, and J.J. Morgan, Marine Chemistry, 4, 1 (1976).

11. Alberto Zirinio and S. Yamamoto, Limnology and Oceanography, 17, 661, (1972).

12. A.H. Truesdell and Blair F. Jones, Jour. Res. U.S.G.S., 2, 233, (1974).

13. Joan R. Westlake, A Handbook of Numerical Matrix Inversion and Solution of Linear Equations, Wiley, New York, 1968.

14. C.W. Davies, Ion Association, Butterworths, Washington, 1962.

15. J.W. Gibbs, The Collected Works of J.W. Gibbs, Yale University Press, 1957.

The University of Michigan

DEPARTMENT OF GEOLOGICAL SCIENCES

1006 C. C. LITTLE BUILDING
ANN ARBOR, MICHIGAN 48109

(313) 764-1435

Multiple Data Set Analysis

Contract DE-FC07-79ID12064

ERIM -- University of Michigan

Ann Arbor, Michigan

Investigator: P. L. Jackson

Digital modeling and analysis of different types of geophysical data are being developed using computer graphics. The development is focused toward simultaneously analyzing several different types of geophysical measurements. The approach is to construct one digital model to which parameters (seismic velocities, densities, resistivities, etc.) are assigned. Simulated data measurements are then computed and compared with actual data measurements.

The rationale for simultaneous modeling of different data sets is to obtain greater accuracy and confidence in the interpretation of the subsurface. The solution of a single data set is not unique: a range of subsurface configurations can produce identical data for each kind of geophysical measurement. With multiple data sets the intersection of the configuration ranges can be found. This intersection must conform to each type of measurement, and must be as small or smaller than the smallest range for one of the data sets. Not only is the analysis made more refined, but it is confirmed by the different data sets.

The ultimate goal (not completely unrealistic) is to develop simultaneous inverse modeling in three dimensions, in which models with variances of different confidence levels are produced from experimental

data. This goal will not be achieved during this one-year contract, but a foundation for steps toward that goal will be. It is anticipated that at least one type of inverse geophysical measurement (gravity) will be included in the simultaneous modeling, as will a statistical evaluation procedure.

Currently seismics, gravity, resistivity, and magnetic data are simultaneously simulated and presented graphically. Density, P-wave velocity, resistivity, and magnetic anomalies are configured in a single cross-section, and displayed on a cathode ray tube. Using the various geophysical parameters, the computer displays a gravity anomaly profile, P- and S-wave travel time curves, resistivity curves or pseudo-sections, and a magnetic intensity profile. Seismic rays (including headwaves) for both local and teleseismic sources are drawn on the cross-section. Experimental data is displayed along with the simulated data.

The program was designed with three goals in mind:

1. Simplicity in using and altering the models.

2. The ready acceptance of new geophysical data types (AMT, EM sounding, heat flow) and also of other algorithms to compute the data types now used.

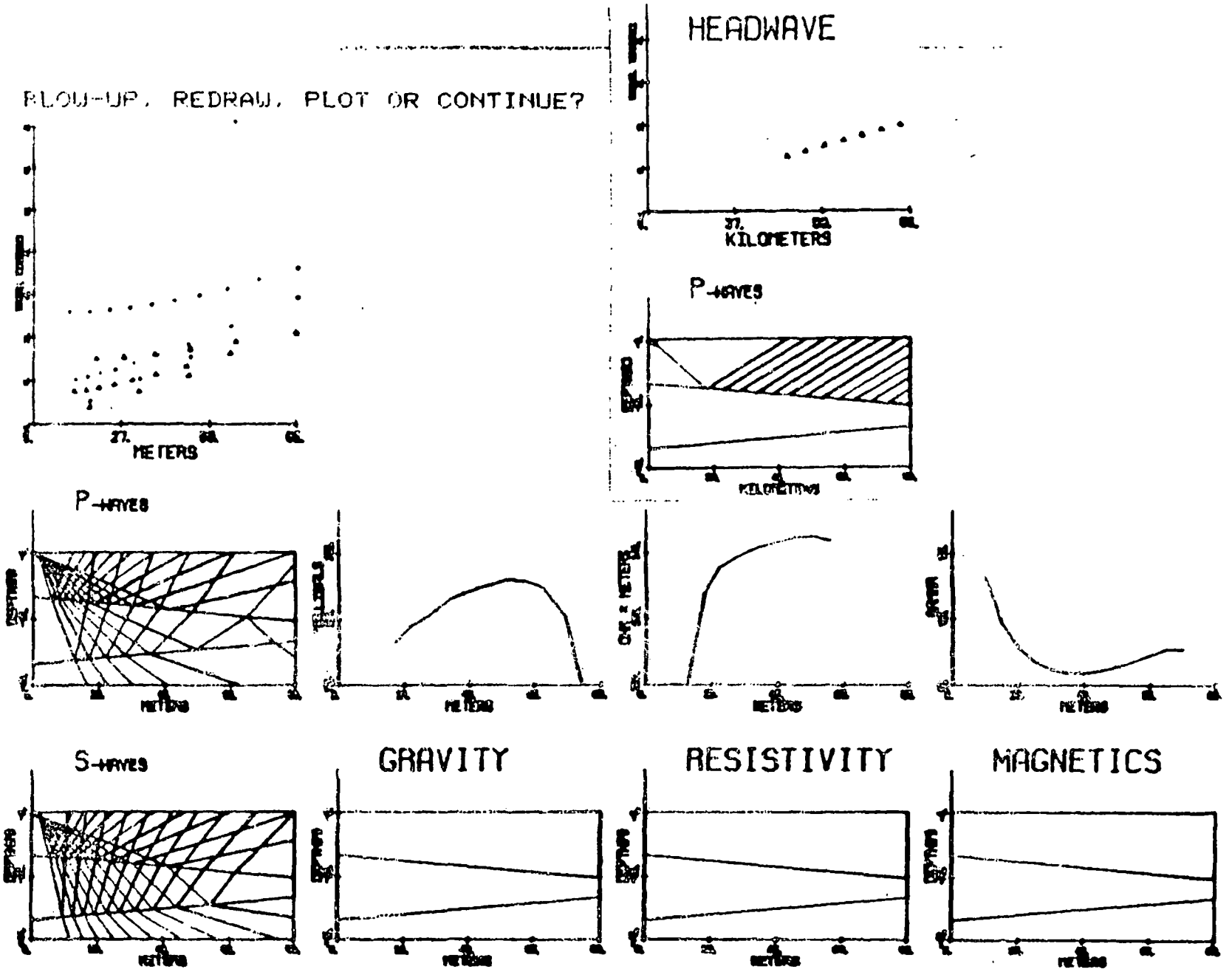
3. Extension to inverse computation and decision making to aid the interpreter.

The algorithms are programed to run on the University of Michigan Amdahl 4701 computer, using a basic interactive graphics package. It is designed to present the model and outputs clearly, and to enable quick alteration of the model.

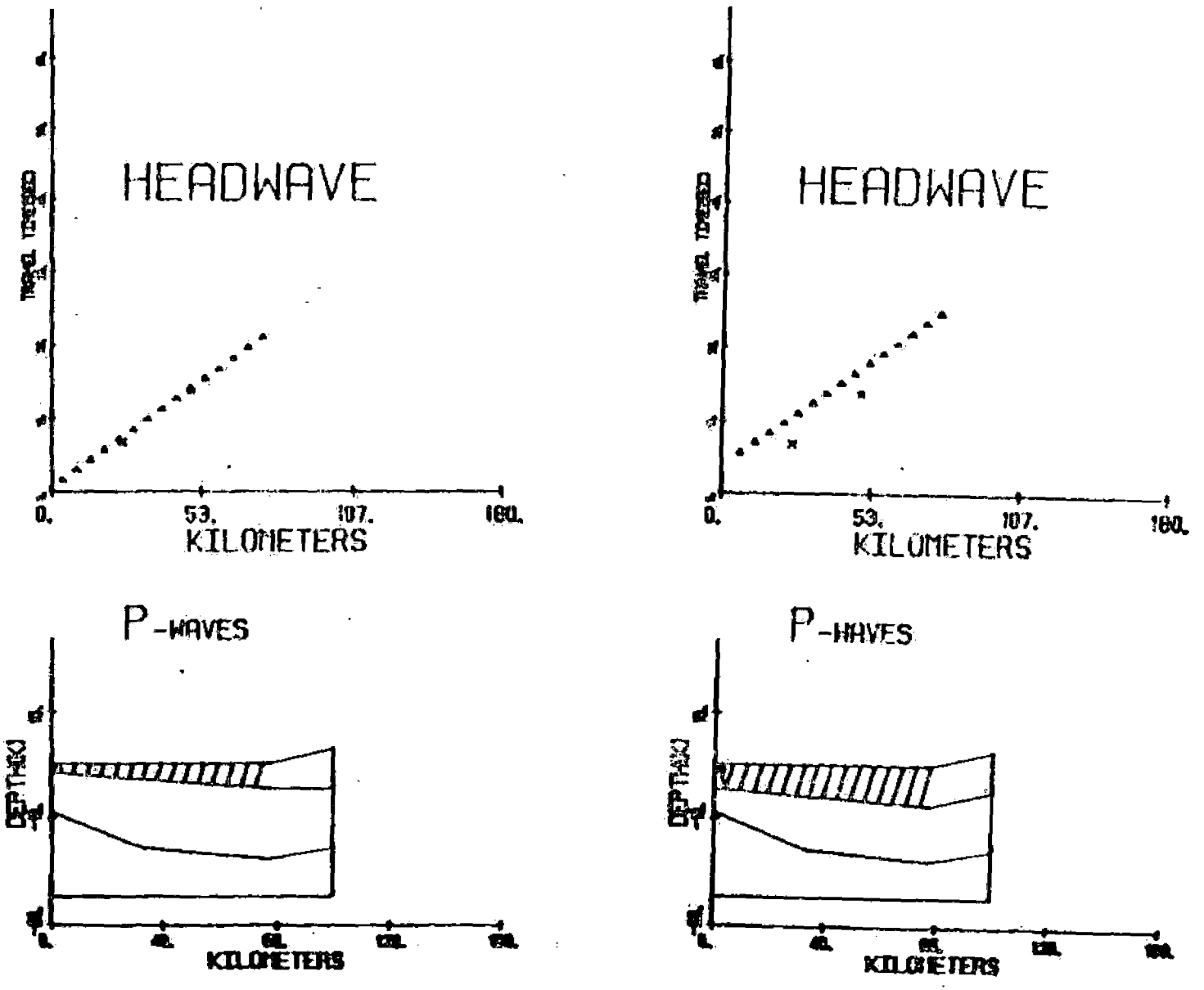
The following pages are duplicates of hard copies of the menus and drawings which are the graphics output.

	ENTER COMMAND STRING
GENERAL	
COMMAND STRING TERMINATOR	(X)
MODELLING PARAMETERS	(M)
ZONE CONSTANTS	(C)
PLOTTING PARAMETERS	(P)
ZONE MODEL	(Z)
SEISMIC MODELLING	
WITH MODEL CHANGES	(S)
WITHOUT MODEL CHANGES	(T)
GRAVITY MODELLING	
WITH MODEL CHANGES	(G)
WITHOUT MODEL CHANGES	(H)
RESISTIVITY MODELLING	
FINITE DIFFERENCE WITH MODEL CHANGES	(R)
FINITE DIFFERENCE WITHOUT MODEL CHANGES	(W)
FINITE ELEMENT WITH MODEL CHANGES	(A)
FINITE ELEMENT WITHOUT MODEL CHANGES	(B)
MAGNETIC MODELLING	
WITH MODEL CHANGES	(D)
WITHOUT MODEL CHANGES	(E)
INPUT THE CURRENT DATABASE	(I)
FINISHED WITH ALL MODELLING	(F)

]. MENU FOR COMMANDS TO CALL AND CHANGE FOUR TYPES OF GEOPHYSICAL MODELING.



2. EXAMPLE OF SIMULTANEOUS MODELING ON SIMPLE MODEL.



4. TWO ALTERATIONS OF MODEL BASED ON WEST END OF GEOTHERMAL AREA REFRACTION LINE. X'S INDICATE EXPERIMENTAL DATA.

GEOMETRY PARAMETERS

	DENSITY	COMP	RIGID	P-VEL	S-VEL	CONDUC	SUS
DEFAULT	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ZONE 1	5.0	41.6	28.8	4.0	2.4	0.0125	0.0020
ZONE 2	4.0	78.7	49.0	6.0	3.5	0.0071	0.0020
ZONE 3	6.0	82.7	50.5	5.0	2.9	0.0040	0.0010

N=NONE
C=CHANGES
H=HELP
COMMAND?

MODELING PARAMETERS

FLAGS	IPLOT=T	IGRAV=T	ISEIS=T	IRESIS=F
	IPLANE=F	METRIC=T	S WAVES=F	P WAVES=T
	VELIN=T	HWTEST=T		
CONSTANTS	DELHW= 5.0	NREFL= 1	SEGLEN= 0.1	FRMIN= 0.1
	HLIM=80.0	VLIM=-40.0		

N=NONE
C=CHANGES
H=HELP
COMMAND?

3. MENU TO ALTER PARAMETERS.

SUBJ
Comp
MTDR2

5/2
23 JAN '78

7-1

GEOTRONICS MTDR2

Data Tape Format Specifications and Header / Trailer Format

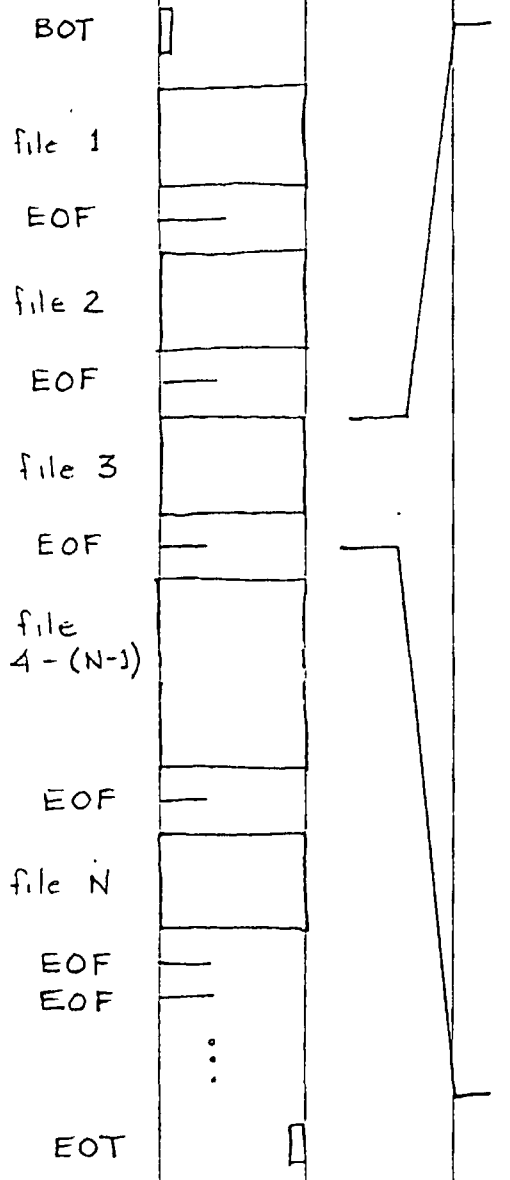
JW

23 JAN '78

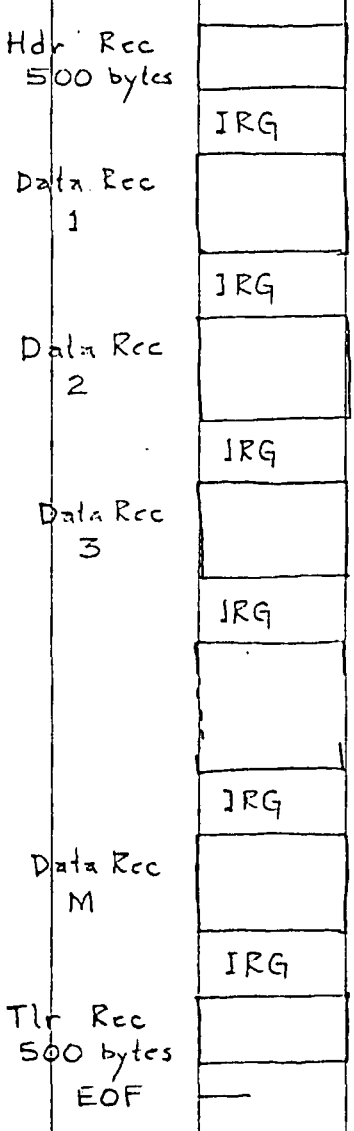
MTDR2 Data Tape Format

Tape: ½ inch, 9 track, NRZI, 800 characters/inch
IBM Standard Parity, Gap and EOF Format
Two 8 bit Characters (Bytes) per 16 bit word
LSB Byte precedes MSB Byte of word on tape

Tape Format



File Format



Definitions:

BOT: beginning of tape marker

EOT: end of tape marker

IRG: inter-record gap

Hdr: Header

Tlr: Trailer repeat of Hdr except for updated info and error flags.

M: Data Rec / file

where: M is next larger integer than NSF/NSR

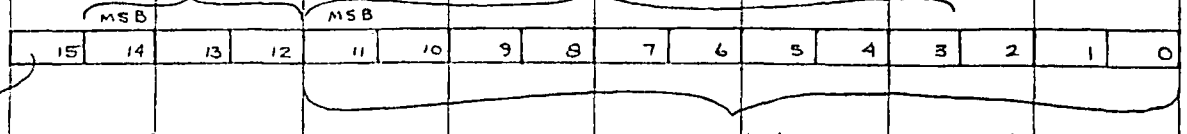
NSF: number of scans per file

NSR: number of scans per record

MTDR2 Data Word, Data Scan Formats

Data Word Structure

GS: Gain Setting — 8 bit data displayed on D/A



End of Scan Flag:

1 ~ last channel of scan
0 ~ all other channels of scan

D: 12 bit data, 2's complement

Scaling: Data → $V = D \times 2^{-GS}$.

where GS: 3 bit unsign binary number (bit 14 MSB)
D: 12 bit 2's complement binary number (bit 11 MSB)

V	GS / D	
+10.235	000 / 011 111 111 111	+ Full Scale
−10.240	000 / 100 000 000 000	− Full Scale

Data Scan Format

A Data Scan is composed of (LCH+1) Data Words, where LCH is the address of the last channel scanned. LCH is set by the MTDR2 field operator. Each Scan sequences from Channel Ø thru Channel (LCH). LCH is constant for a given Data Run (Magtape file) and for standard field operation is set to four. LCH is word 93 in the Hdr/Tlr Record.

The data channels are sampled with a time skew of SK μ seconds. This skew time is different for automatic gain ranging or fixed gain mode. The correct skew time SK is written into word 11ᴮ in the Hdr/Tlr Record by the MTDR2 software.

MTDR2 Magtape Data Record Format

Data Record

The Data Record is a physical magtape record as depicted in the File Format on page 1. Each data record is a Time Word followed by an integral number of data scans. The number of scans per record, NSR is set by the MTDR2 software and is dependent upon the LCH setting.

As indicated in the data word structure, the MSB of the last sample of every scan (End of Scan Flag) is set.

Where the number of data records per file is greater than but not equal to:

NSF / NSR (see definitions page 1),

the last record is zero fill beyond the last data word of the run.

Time Word

The Time Word is the first word (two bytes) of each data record. The second word is the first data word of the scan (Channel Ø). The time word contains the time this data sample was taken if either of two conditions exist:

α) The record is the first data record of the file (follows the Hdr record)

β) The record is the first data record after a time gap has occurred.

In all other cases, the data in the present record has a fixed time period from the data in the previous data record defined by the scan rate, that is the data sampling is continuous. This continuity of sampling between data records is indicated by the Time Word with all bits set.

Time Word Format: $A_3 A_2 A_1 A_0$

for α) and β): A_n is 4 bit BCD integer (n=0, least significant digit)

$A_3 A_2$: time minutes
$A_1 A_0$: time seconds

otherwise: A_n is 4 bit Hexadecimal F for n = 0, 1, 2, 3

23 JAN 1978

MTDR2 Tape Header/Trailer Format — Standard MUX 2
Geotronics Corp.

Word No.	Contents	Units	Structure	Remarks
1	Run Number		$A_3A_2A_1A_0$	
2	Year		$\phi\phi\ A_1A_0$	$19A_1A_0$ years A.D.
3	Day of Year			
4	Run Start Time (Hour)	hours	$\phi\phi\ A_1A_0$	
5	Run Start Time (Min:sec)	min:sec	$A_3A_2A_1A_0$	A_3A_2 min : A_1A_0 sec
6	Scan Rate Index		$\phi\phi\ A_1A_0$	
7	Scan Rate	Scans/Sec	$A_3A_2A_1B_0$	$A_3A_2A_1.\phi \times 10^{B_0}$ where B_0 is 2's complement signed exponent.
8	Log_2(No. Scans/Run)		$\phi\phi\phi\ B_0$	
9	No. Data Channels / Scan		$\phi\phi\phi\ B_0$	
10	No. Scans / Data Record		$B_3B_2B_1B_0$	
11	Channel-to-Channel Skew	μsec	$A_3A_2A_1A_0$	$A_3A_2A_1.A_0$ μsec
12	No. Scans Completed in Run		$B_3B_2B_1B_0$	Valid only in Trailer. Zero in header.
13	Azimuth for Site No. = ϕ	degrees	$A_3A_2A_1A_0$	$A_3A_2A_1.A_0$ Degrees
14	Azimuth for Site No. = 1	degrees	$A_3A_2A_1A_0$	$A_3A_2A_1.A_0$ Degrees
15	Azimuth for Site No. = 2	degrees	$A_3A_2A_1A_0$	$A_3A_2A_1.A_0$ Degrees
16	E_x Line Length for Site No. = ϕ	feet	$A_3A_2A_1A_0$	Note: Azimuth as declination of magnetic North from true North and E_x and E_y line lengths as electrode spacings are loaded as a group indexed by the front panel Site No. thumbwheel switch (or Station No.). Normal convention uses Site No. = ϕ (Base Station) only.
17	E_y Line Length for Site No. = ϕ	feet	$A_3A_2A_1A_0$	
18	E_x Line Length for Site No. = 1	feet	$A_3A_2A_1A_0$	
19	E_y Line Length for Site No. = 1	feet	$A_3A_2A_1A_0$	
20	E_x Line Length for Site No. = 2	feet	$A_3A_2A_1A_0$	
21	E_y Line Length for Site No. = 2	feet	$A_3A_2A_1A_0$	
22	Analog System ID No.		$\phi\phi\phi D_2D_1D_0$	Normally = ϕ. On Geotronics-owned equipment, analog components are coded to yield the following identification: D_2 = MTE Code No. D_1 = MTH Code No. D_0 = MTF Code No.

Word No.	Contents	Units	Structure	Remarks
23	Digital System ID No.		$A_3A_2A_1A_0$	MTDR2 Serial No.
24	Software System ID No.		$A_3A_2A_1A_0$	A_3A_2: A_1A_0 = month: year
25	Trailer Update Flag		$\phi\phi\phi B_0$	Flag normally zero. Flag is set nonzero in Trailer if Trailer is "updated" i.e, site parameters in Trailer are modified from those in Header.

Word No.	Contents	Units	Structure 16 bit word		Remarks
26 : 88	General Data Storage Block — Storage Block - Storage for 63_{10} words (accessible from MTDR2 panel)		$B_3 B_2 B_1 B_0$: "		(binary integer) : "
89 : 124	General Storage Block — Software Block -37_{10} words (accessible from MTDR2 software only)				
89	Header/Trailer Identification		$E_1 E_0$	ASCII	E_1E_0 - "H" "E" (header) E_1E_0 - "T" "A" (trailer)
90	" " "		$E_1 E_0$	ASCII	E_1E_0 - "A" "D" (header) E_1E_0 - "I" "L" (trailer)
91	Record Length		$B_3 B_2 B_1 B_0$		Data record length in # of 8-bit bytes
92	No. Scans/File		$B_3 B_2 B_1 B_0$		
93	Last Channel Scanned		$\emptyset\ \emptyset\ \emptyset\ B_0$		Defines scan length (Chan 0 is first channel).
94	Storage - Channel-to-channel Skew for Auto Gain Ranging Mode of Operation	μsec	$A_3 A_2 A_1 A_0$		$A_3A_2A_1A_0$ μsec (see word 11)
95	Storage - Channel to channel Skew for Fixed Gain mode of operation	μsec	$A_3 A_2 A_1 A_0$		$A_3A_2A_1A_0$ μsec (see word 11)
96	Permanent Tape Error count		$B_3 B_2 B_1 B_0$		Valid in trailer only. If non-zero, contains no. of permanent tape errors, i.e those errors that could not be corrected within specified no. of attempts (usually 3).
97	Fatal Tape Error Message		$B_3\ \emptyset\ \emptyset\ A_0$		Valid in trailer only. If nonzero, contains BCD Error Code for

Word No.	Contents	Units	Structure 16 bit word	Remarks
				fatal error that terminated run in form FQQX (per operator's manual).
98	Sample Error Count		$B_3B_2B_1B_0$	Sample error count as determined by A/D controller "data ready" and not retrieved in time by computer. Preset to -3 in header. Actual error count in trailer. Record aborts on 3rd error. For 1 or 2 errors, zero is substituted for the data word missed.
99	Last Permanent Tape Error Code		$B_3\emptyset A_1A_0$	If non-zero, contains BCD error code for last permanent tape error in form EOXX (per operator's manual). Valid in trailer only.
100 : 108	A/D Amplifier Check Data Array		$1D_3B_2B_1B_0$	The A/D Gain Range Amplifier is checked at the beginning of each run by digitizing a reference voltage at each level of gain. The data samples are stored in this array.
100	Sample for fixed gain = 1 (2^0)		$1D_3B_2B_1B_0$	Value = $B_2B_1B_0 \times 2^{-D_3} \times .005$ V
101	" " " " = 2 (2^1)		"	where $B_2B_1B_0$ is
102	" " " " = 4 (2^2)		"	12 bit, signed, 2's complement data with least
103	" " " " = 8 (2^3)		"	significant bit value
104	" " " " = 16 (2^4)		"	equal to 5 mV.
105	" " " " = 32 (2^5)		"	D_3 = 3 bit binary integer.
106	" " " " = 64 (2^6)		"	
107	" " " " = 128 (2^7)		"	
108	" for auto gain ranging (max gain = 128)		"	

Word No.	Contents	Units	Structure 16 bit word		Remarks
109 : 124	A/D Amplifier Check Error Limits table		$B_3B_2B_1\emptyset$		This table contains the upper and lower limits which are used in checking the data samples in check array for accuracy. Each value is a 12-bit signed, 2's complement number left-justified in word.
				Octal Value	Equiv. Voltage
109	Lower Limit for Gain = 1	"		000140	0.030 V
110	Upper " " " = 1	"		000240	0.050
111	Lower " " " = 2	"		000300	0.060
112	Upper " " " = 2	"		000500	0.100
113	Lower " " " = 4	"		000640	0.120
114	Upper " " " = 4	"		001140	0.200
115	Lower " " " = 8	"		001500	0.260
116	Upper " " " = 8	"		002300	0.380
117	Lower " " " = 16	"		003200	0.520
118	Upper " " " = 16	"		004500	0.740
119	Lower " " " = 32	"		006400	1.040
120	Upper " " " = 32	"		011200	1.480
121	Lower " " " = 64	"		015000	2.080
122	Upper " " " = 64	"		022400	2.960
123	Lower " " " = 128	"		032000	4.160
124	Upper " " " = 128	"		045000	5.920

Word No.	Contents	Units	Structure	Remarks
125	Not used		$\phi\phi\phi\phi$	
126	Analog Switch Setting		$C_3D_2C_1B_0$	Each item in this array contains the mux address used to access the given switch and the resultant switch setting as the data item. $B_0 = \phi$.
157				
	PreAmp Settings			
126	MTE Channel 1 (E_X) Gain		$C_3D_2C_1B_0$	$C_3,C_1 = \phi 1_8$, $B_0 = \phi$
127	MTE Chan. 1 (E_X) Filter Control		$C_3D_2C_1B_0$	$C_3,C_1 = \phi 2_8$, $B_0 = \phi$
128	MTE Channel 2 (E_Y) Gain		$C_3D_2C_1B_0$	$C_3,C_1 = \phi 3_8$, $B_0 = \phi$
129	MTE Chan. 2 (E_Y) Filter Control		$C_3D_2C_1B_0$	$C_3,C_1 = \phi 4_8$, $B_0 = \phi$
130	MTE Channel 3 Gain		$C_3D_2C_1B_0$	$C_3,C_1 = \phi 5_8$, $B_0 = \phi$
131	MTE Channel 3 Filter Control		$C_3D_2C_1B_0$	$C_3,C_1 = \phi 6_8$, $B_0 = \phi$
132	MTH Channel 1 (H_X) Gain		$C_3D_2C_1B_0$	$C_3,C_1 = 11_8$, $B_0 = \phi$
133	MTH Chan 1 (H_X) Filter Control		$C_3D_2C_1B_0$	$C_3,C_1 = 12_8$, $B_0 = \phi$
134	MTH Channel 2 (H_Y) Gain		$C_3D_2C_1B_0$	$C_3,C_1 = 13_8$, $B_0 = \phi$
135	MTH Chan. 2 (H_Y) Filter Control		$C_3D_2C_1B_0$	$C_3,C_1 = 14_8$, $B_0 = \phi$
136	MTH Channel 3 (H_Z) Gain		$C_3D_2C_1B_0$	$C_3,C_1 = 15_8$, $B_0 = \phi$
137	MTH Chan. 3 (H_Z) Filter Control		$C_3D_2C_1B_0$	$C_3,C_1 = 16_8$, $B_0 = \phi$
	Post-Amp (Filter) Settings			
140	MTF Channel 1 (E_X) Gain		$C_3D_2C_1B_0$	$C_3,C_1 = \phi 1_8$, $B_0 = \phi$
141	MTF Channel 1 (E_X) Lo-Cut		$C_3D_2C_1B_0$	$C_3,C_1 = \phi 2_8$, $B_0 = \phi$
142	MTF Channel 1 (E_X) Hi-Cut		$C_3D_2C_1B_0$	$C_3C_1 = \phi 3_8$, $B_0 = \phi$
143	MTF Channel 2 (E_Y) Gain		$C_3D_2C_1B_0$	$C_3,C_1 = \phi 4_8$, $B_0 = \phi$
144	MTF Channel 2 (E_Y) Lo-Cut		$C_3D_2C_1B_0$	$C_3,C_1 = \phi 5_8$, $B_0 = \phi$
145	MTF Channel 2 (E_Y) Hi-Cut		$C_3D_2C_1B_0$	$C_3,C_1 = \phi 6_8$, $B_0 = \phi$
146	MTF Channel 3 (H_X) Gain		$C_3D_2C_1B_0$	$C_3,C_1 = \phi 7_8$, $B_0 = \phi$
147	MTF Channel 3 (H_X) Lo-Cut		$C_3D_2C_1B_0$	$C_3,C_1 = 10_8$, $B_0 = \phi$
148	MTF Channel 3 (H_X) Hi-Cut		$C_3D_2C_1B_0$	$C_3,C_1 = 11_8$, $B_0 = \phi$

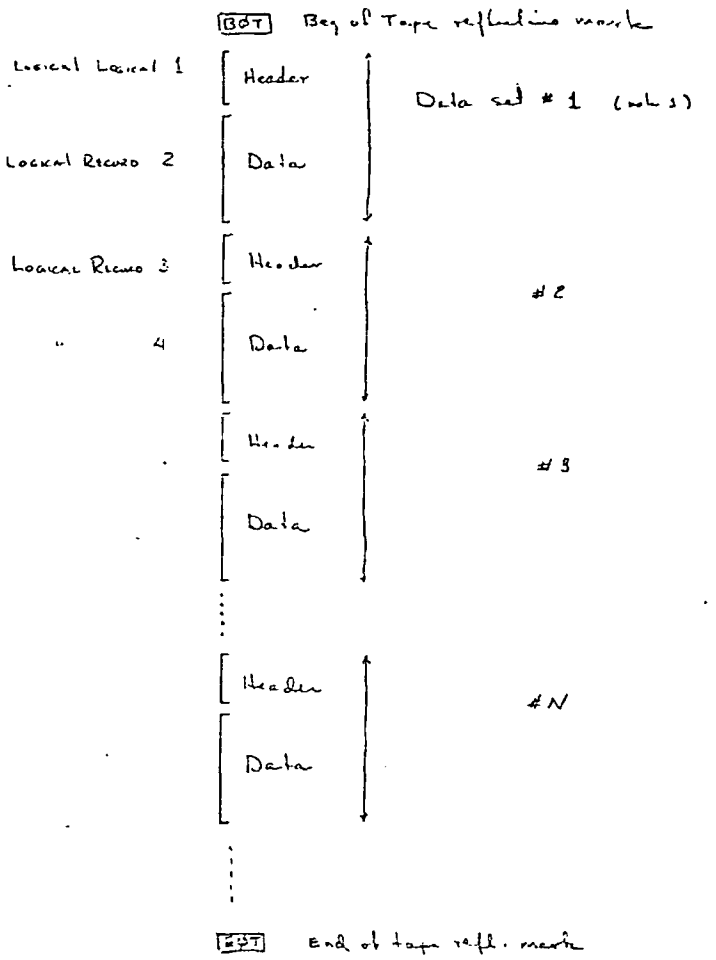
Word No.	Contents	Units	Structure	Remarks
149	MTF Channel 4 (H_y) Gain		$C_3D_2B_1B_0$	$C_3,C_1 = 12_8$, $B_0 = \phi$
150	MTF Channel 4 (H_y) Lo-Cut		$C_3D_2B_1B_0$	$C_3,C_1 = 13_8$, $B_0 = \phi$
151	MTF Channel 4 (H_y) Hi-Cut		$C_3D_2B_1B_0$	$C_3,C_1 = 14_8$, $B_0 = \phi$
152	MTF Channel 5 (H_z) Gain		$C_3D_2B_1B_0$	$C_3,C_1 = 15_8$, $B_0 = \phi$
153	MTF Channel 5 (H_z) Lo-Cut		$C_3D_2B_1B_0$	$C_3,C_1 = 16_8$, $B_0 = \phi$
154	MTF Channel 5 (H_z) Hi-Cut		$C_3D_2B_1B_0$	$C_3,C_1 = 17_8$, $B_0 = \phi$
155			$\phi\phi\phi\phi$	
156	Not used.		$\phi\phi\phi\phi$	
157	Not used		$\phi\phi\phi\phi$	
158	Data Time Sequence Interrupt Flag		$B_3B_2B_1B_0$	Zero value indicates continuous sampling. Nonzero value indicates count of number of time gaps incurred in data.
159	Not used		$\phi\phi\phi\phi$	
160	System Configuration		$A_3A_2A_1A_0$	Normal System Configuration = 2.

Legend:

A_n – 4 bit BCD integer
B_n – 4 bit binary integer
C_n – 5 bit binary integer
D_n – 3 bit binary integer
E_n – 8 bit ASCII character

Subscript n specifies order in word; $n=\phi$ is low order (LSB end)

Save Tape (Merge Tape) Format



Notes:

1) See SAVE TAPE FORMAT manual for detailed format of individual data sets on tape.

2) Each data set corresponds to a recording run, and stores the results for a particular freq. band.

3) The data set recording run no. for each set is contained in the Header information.

4) Each Data set contains 2 Logical Records.

```
      SUBROUTINE OUTTAPE (TITLE,IOS,I1,I2,I3)                          OUTTAPE
C                                                                      OUTTAPE
C     SAVE TAPE FORMAT                                                 OUTTAPE
C                                                                      OUTTAPE
C   ---HEADER RECORD---                                                OUTTAPE
C                                                                      OUTTAPE
C    VARIABLE OR ARRAY                  WORD NUMBER                    OUTTAPE
C         FLAG1                         1                              OUTTAPE
C         NFREQ                         2                              OUTTAPE
C         IOS                           3--82                          OUTTAPE
C         I1                            83                             OUTTAPE
C         I2                            84                             OUTTAPE
C         I3                            85                             OUTTAPE
C         DATE                          86                             OUTTAPE
C         HOUR                          87                             OUTTAPE
C         MIN                           88                             OUTTAPE
C         SEC                           89                             OUTTAPE
C         HEAD2(1-500)                  90--589                        OUTTAPE
C                                                                      OUTTAPE
C                                                                      OUTTAPE
C   ---DATA RECORD---                                                  OUTTAPE
C                                                                      OUTTAPE
C    VARIABLE OR ARRAY                  WORD NUMBER                    OUTTAPE
C         FLAG2                         1                              OUTTAPE
C         NFREQ                         2                              OUTTAPE
C         PASSLVLS                      3--22                          OUTTAPE
C         FR                            23--22+NFREQ                   OUTTAPE
C         NSP                           23+NFREQ--22+2*NFREQ           OUTTAPE
C         PP                            23+2*NFREQ--22+27*NFREQ        OUTTAPE
C         DEPC                          23+27*NFREQ--22+29*NFREQ       OUTTAPE
C         ELIPC                         23+29*NFREQ--22+31*NFREQ       OUTTAPE
C         TANC                          23+31*NFREQ--22+33*NFREQ       OUTTAPE
C         PHOC                          23+33*NFREQ--22+35*NFREQ       OUTTAPE
C         TAC                           23+35*NFREQ--22+37*NFREQ       OUTTAPE
C         COP                           23+37*NFREQ--22+41*N6R58       OUTTAPE
C         PC                            23+41*NFREQ--22+45*NFREQ       OUTTAPE
C         IPC                           23+45*NFREQ--22+49*NFREQ       OUTTAPE
C         COC                           23+49*NFREQ--22+53*NFREQ       OUTTAPE
C         PRC                           23+53*NFREQ--22+58*NFREQ       OUTTAPE
C         ANC                           23+58*NFREQ--22+63*NFREQ       OUTTAPE
C         COHC                          23+63*NFREQ--22+68*NFREQ       OUTTAPE
C         ANGC                          23+68*NFREQ--11+71*NFREQ       OUTTAPE
C         KMMC                          23+71*NFREQ--22+73*NFREQ       OUTTAPE
C         ALPC                          23+73*NFREQ--22+75*NFREQ       OUTTAPE
C         BTAC                          23+75*NFREQ--22+78*NFREQ       OUTTAPE
C         DELC                          23+78*NFREQ--22+80*NFREQ       OUTTAPE
C         KZE                           23+80*NFREQ--22+82*NFREQ       OUTTAPE
C         AKZ                           23+82*NFREQ--22+84*NFREQ       OUTTAPE
C         COK                           23+84*NFREQ--22+85*NFREQ       OUTTAPE
C         ANK                           23+85*NFREQ--22+86*NFREQ       OUTTAPE
C         BTAK                          23+86*NFREQ--22+87*NFREQ       OUTTAPE
C         IXXC                          23+87*NFREQ--22+105*NFREQ      OUTTAPE
C         IXYC                          23+105*NFR50--22+123*NFR58     OUTTAPE
C         IEXXC                         23+123*NFREQ--22+124*NFREQ     OUTTAPE
```

Notes:

- 9 – Header Record flag, FLAG1 = 1
- 10 – ~~NFREQ = 27~~ ~~[illegible]~~
- NFREQ IS DEPENDENT ON NUMBER OF DATA POINTS PROCESSED.
- 19 – TAPE 2 header words in order – see TAPE 2 specs.
- 25 – Data Record flag, FLAG2 = 0
- 27 – not currently used

```
      C       IEXYC                    23*124*NFREQ--22*125*NFREQ        OUTTAPE   56
      C       EPDCOH                   23*125*NFREQ--22*126*NFREQ        OUTTAPE   57
      C       HPDCOH                   23*126*NFREQ--22*127*NFREQ        OUTTAPE   58
    - C                                                                  OUTTAPE   59
      C                                                                  OUTTAPE   60
000010      *COMMON /SPEC/SP(8193),      FR(100),RNSP(100),P(25,140),PP(100,25)  OUTTAPE   61
     1,DEPC(100,2),ELIPC(100,2),RIANC(100,2),RHOC(100,2),RIAC(100,2),          OUTTAPE   62
     2COR(100,2),RC(100,4),PIPC(100,4),COC(100,4),RRC(100,5),ANC(100,5),       OUTTAPE   63
     3 COHC(100,5),ANGC(100,3),RKHMC(100,2),ALPC(100,2),BTAC(100,3),DELC       OUTTAPE   64
     4(100,2),RKZE(100,2),AKZ(100,2),COK(100),ANK(100),BTAK(100),              OUTTAPE   65
     5RIXXC(100,18),RIXYC(100,18),RIEXXC(100),RIEXYC(100),EPDCOH(100),         OUTTAPE   66
     6HPDCOH(100)                                                            OUTTAPE   67
000010       COMMON /HEADER/ HEAD2(500)                                      OUTTAPE   68
000010       COMMON /PASSLVL/ ARRAY(20)                                      OUTTAPE   69
000010       DIMENSION TITLE(8),RIOS(80),IOS(1)                              OUTTAPE   70
000010       INTEGER DATE,CLOCK                                              OUTTAPE   71
```

{E & H predicted coherency - not currently computed

* See subsequent description of variable names in the embedded documentation section for subroutine MACTEL.

```
      SUBROUTINE MAGTEL(P,F,NSP,TITLE,NFREQ,NBIAS)                      MAGTEL     2
C*****************************************************************     MAGTEL     3
C                    ** GEOTRONICS CORP - AUSTIN, TEXAS USA **        MAGTEL     4
C                                                                      MAGTEL     5
C  SUBROUTINE <MAGTEL> - FORTRAN IV              DRW5022X001           MAGTEL     6
C                                                                      MAGTEL     7
C  USED CALL MAGTEL (P,F,NSP,TITLE,NFREQ,NBIAS)                        MAGTEL     8
C                                                                      MAGTEL     9
C       MAGTEL COMPUTES MAGNETOTELLURIC (MT) RESULTS                   MAGTEL    10
C       FROM THE POWER SPECTRA MATRIX <P>. QUANTITIES                  MAGTEL    11
C       COMPUTED ARE DESCRIBED BELOW IN THE NOTATION                   MAGTEL    12
C       GIVEN IN THE <MAGTAN1> HEADER.                                 MAGTEL    13
C       ALL OUTPUT QUANTITIES ARE STORED IN COMMON <SPEC>              MAGTEL    14
C       FOR FURTHER ACCESS BY OUTPUT ROUTINES.                         MAGTEL    15
C                                                                      MAGTEL    16
C  PARAMETERS0                                                         MAGTEL    17
C    <P(J,I)> - AUTO- AND CROSS-POWER SPECTRA MATRIX FOR               MAGTEL    18
C               FIELD COMPONENTS <EX,EY,HX,HY,HZ>.                     MAGTEL    19
C                     I- FREQ INDEX                                    MAGTEL    20
C                     J- COMPONENT INDEX                               MAGTEL    21
C               SPEC COMPONENT LOCATIONS -                             MAGTEL    22
C               J=    1-PEXEX    10-PEYEY 18.19-PHXHY                  MAGTEL    23
C                   2,3-PEXEY 11,12-PEYHX 20,21-PHXHZ                  MAGTEL    24
C                   4,5-PEXHX 13,14-PEYHY    22-PHYHY                  MAGTEL    25
C                   6,7-PEXHY 15,16-PEYHZ 23,24-PHYHZ                  MAGTEL    26
C                   8,9-PEXHZ    17-PHYHX    25-PHZHZ                  MAGTEL    27
C               (CROSS-POWERS ARE STORED WITH REAL AND IMAG             MAGTEL    28
C                PARTS ADJACENT WORDS IN ORDER)                        MAGTEL    29
C              NOTE 1-E-POWER      UNITS - (MV/KM)**2/HZ               MAGTEL    30
C                   H-POWER        UNITS -   GAMMA**2/HZ               MAGTEL    31
C                   E-H-POWER      UNITS - (MV/KM)*GAMMA/HZ            MAGTEL    32
C              NOTE 2-THE COMPONENT ORDER GIVEN IS FOR <P>             MAGTEL    33
C                     UPON INPUT TO <MAGTEL>. THE <P> ORDER            MAGTEL    34
C                     IS MODIFIED IN <MAGTEL> AFTER CALL OF            MAGTEL    35
C                     <ZFIT> AND SOME INFO IS DISCARDED. THE           MAGTEL    36
C                     UNMODIFIED <P(J,I)> INFO IS SAVED IN             MAGTEL    37
C                     <PP(I,J)>. BOTH ARE STORED IN <SPEC>.            MAGTEL    38
C      <F(I)> - FREQ OF ITH WORD IN ALL OUTPUT ARRAYS (HZ).            MAGTEL    39
C    <NSP(I)> - NO. OF INCREMENTAL HARM ASSOC WITH <FR(I)>.            MAGTEL    40
C     <TITLE> - TITLE OF DATA SET.                                     MAGTEL    41
C     <NFREQ> - NO. OF WORDS IN <FR(I)> (I=1,NFREQ).                   MAGTEL    42
C     <NBIAS> - NO. OF COMPONENTS IN <P(J,I)> (J=1,NBIAS)              MAGTEL    43
C                                                                      MAGTEL    44
C  ROUTINES CALLED0 <ZFIT>                                             MAGTEL    45
C                   <IOATAN>                                           MAGTEL    46
C                                                                      MAGTEL    47
C  SPECIAL STORAGE AREAS0                                              MAGTEL    48
C       COMMON BLOCK <SPEC> - 25000 WORDS                              MAGTEL    49
C                                                                      MAGTEL    50
C                                                                      MAGTEL    51
C   MT RESULTS COMPUTED0 (ARRAYS IN COMMON <SPEC>)                     MAGTEL    52
C             NOTE 1-SEE <MAGTAN1> FOR NOTATION.                       MAGTEL    53
C             NOTE 2-I - FREQ INDEX (I=1,NFREQ)                        MAGTEL    54
C                    J - CONTENTS INDEX                                MAGTEL    55
```

Note: A printer code incompatibility causes the following character misprints:

< ⇒ +

: ⇒ 0

```
C                                                                      MAGTEL     56
C       *NFREQ> - NO. OF FREQS.                                        MAGTEL     57
C       *FR(I)> - FREQ - I=1,*NFREQ - (HZ)                             MAGTEL     58
C      *NSP(I)> - NO. INCREMENTAL HARM AVGD FOR *FR(I)>.               MAGTEL     59
C                                                                      MAGTEL     60
C      *P(J,I)> - POWER SPECTRA MATRIX - SEE ABOVE DESCR.              MAGTEL     61
C     *PP(I,J)> -  = *P(J,I)> PRIOR TO ANY MOD OF *P> .                MAGTEL     62
C                                                                      MAGTEL     63
C   *DEPC(I,J)> - J=1,2 - RATIO OF UNPOLARIZED POWER TO TOTAL          MAGTEL     64
C                 POWER OF E AND H FIELDS RESPECTIVELY.                MAGTEL     65
C                                                                      MAGTEL     66
C  *ELIPC(I,J)> - J=1,2 - RATIO OF MINOR TO MAJOR AXIS OF              MAGTEL     67
C                 POLARIZATION ELLIPSE FOR POLARIZED COMPONENTS        MAGTEL     68
C                 OF E AND H(HORIZ) FIELDS RESPECTIVELY.               MAGTEL     69
C                 (+ FOR RT HAND POLARIZ - CLOCKWISE WHEN              MAGTEL     70
C                 LOOKING IN +Z-AXIS DIRECTION)                        MAGTEL     71
C                                                                      MAGTEL     72
C   *IANC(I,J)> - AZIMUTH ANGLE (DEGREES) OF MAJOR AXIS OF             MAGTEL     73
C                 POLARIZ ELLIPSE FOR E AND H(HORIZ) FIELDS.           MAGTEL     74
C                                                                      MAGTEL     75
C 0*RHOC(I,J)> - J=1,2- APPARENT RESISTIVITY (APP RES) FOR             MAGTEL     76
C 0               ZX AND ZY RESPECTIVELY (OHM-METERS).                 MAGTEL     77
C 0 *IAC(I,J)> - J=1,2- PHASE OF ZX AND ZY (DEGREES).                  MAGTEL     78
C 0 *COR(I,J)> - COHERENCY FOR (EX-HY) AND (EY-HX).                    MAGTEL     79
C 0                                                                    MAGTEL     80
C 0 WHERE  ZX = EX/HY AND ZY = EY/HX  (UNROTATED CAGNIARD Z).          MAGTEL     81
C                                                                      MAGTEL     82
C    *RC(I,J)> - J=1,4- APP RES FOR TENSOR *Z> ELEMENTS                MAGTEL     83
C                 ZXX,ZYY,ZXY,ZYX IN ORDER (OHM-METERS).               MAGTEL     84
C   *IPC(I,J)> - J=1,4- PHASE OF ZXX,ZYY,ZXY,ZYX  (DEGREES)           MAGTEL     85
C   *COC(I,J)> - J=1,4- PHASOR COHERENCY FOR ZXX,ZYY,ZXY,ZYX.          MAGTEL     86
C                                                                      MAGTEL     87
C  NOTE--ROTATED *Z> AND *Y> RESULTS --- IN THE FOLLOWING THE          MAGTEL     88
C        XY-AXES ARE ROTATED AT EACH FREQ TO ANGLE *A>=*A(Z)>          MAGTEL     89
C        FOR *Z> AND INVERTED *Y> TENSORS SO THAT                      MAGTEL     90
C        CABS*ZXY(A)+ZYX(A)> IS MAX FOR *A>=*A(Z)>. THE XY-AXES        MAGTEL     91
C        ARE ROTATED FOR *YZ> (EQUATION I-7 OF *MAGTAN1>) TO           MAGTEL     92
C        ANGLE *A>=*A(YZ)> SO THAT CABS*YZY(A)> IS MAX (HZ IS          MAGTEL     93
C        MOST COHERENT WITH EY). THE XY-AXES ARE ROTATED FOR           MAGTEL     94
C        *KZ> (EQUATION I-8 OF *MAGTAN1> TO *A>=*A(KZ)> SO THAT        MAGTEL     95
C        CABS*KZX(A)> IS MAX (HZ IS MOST COHERENT WITH HX).            MAGTEL     96
C        FINALLY THE  IMPEDANCES   *ZTE> (E PARALLEL TO STRIKE)        MAGTEL     97
C        AND *ZTM> (H PARALLEL TO STRIKE) ARE SELECTED FROM            MAGTEL     98
C        *ZXY(A(Z))> AND *ZYX(A(Z))> ON THE BASIS OF THE              MAGTEL     99
C        1ST AN 4TH QUADRANT PRINCIPLE VALUES OF *A(Z)> AND            MAGTEL    100
C        *A(YZ)> -                                                     MAGTEL    101
C           IF(ABS*A(Z)-A(YZ)>.LE.45 DEGR) -->*ZTE>=*ZYX(A(Z))>        MAGTEL    102
C                                             *ZTM>=*ZXY(A(Z))>        MAGTEL    103
C           IF(ABS*A(Z)-A(YZ)>.GT.45 DEGR) -->*ZTE>=*ZXY(A(Z))>        MAGTEL    104
C                                             *ZTM>=*ZYX(A(Z))>        MAGTEL    105
C                                                                      MAGTEL    106
C   *PRC(I,J)> - J=1,2- APP RES - *ZTE>,*ZTM> - *Z> TENSOR             MAGTEL    107
C                3,4- APP RES - *ZTE>,*ZTM> - *Y> TENSOR               MAGTEL    108
C                  5- APP RES - *YZY(A(YZ))>- *Y> TENSOR               MAGTEL    109
C                     (I.E.- APP RES FOR EY/HZ AT *A(YZ)>.)            MAGTEL    110
```

```
C  *ANC(I,J)> - J=1,2- PHASE - *ZTE>,*ZTM>   - *Z> TENSOR      C  MAGTEL  111
C                 3,4- PHASE - *ZTE>,*ZTM>   - *Y> TENSOR      C  MAGTEL  112
C                   5- PHASE - *YZY(A(YZ))>  - *Y> TENSOR      C  MAGTEL  113
C  *COHC(I,J)> - J=1,2- PHASOR COH - *ZTE>,*ZTM> - *Z> TENSOR   C  MAGTEL  114
C                 3,4- PHASOR COH - *ZTE>,*ZTM> - *Y> TENSOR   C  MAGTEL  115
C                   5- PHASOR COH - *YZY(A(YZ))>- *Y> TENSOR   C  MAGTEL  116
C  *ANGC(I,J)> - J=1,2- *A(Z)>-*Z> TENSOR, *A(Z)>-*Y> TENSOR   C  MAGTEL  117
C                   3- *A(YZ)> - *Y> TENSOR                    C  MAGTEL  118
C  *DELC(I,J)> - J=1,2- NORMALIZED DENOMINATOR TERMS ASSOC      C  MAGTEL  119
C                       WITH SOLUTIONS FOR *Z> AND *Y> RESP.   C  MAGTEL  120
C                       (USED TO ASSESS COMPUTATIONAL          C  MAGTEL  121
C                       STABILITY. *Z> OR *Y> ESTIMATE IS      C  MAGTEL  122
C                       ACCEPTED IF *DELC>.GE.*0.1>).          C  MAGTEL  123
C  *ALPC(I,J)> - J=1,2- TENSOR SKEW FOR *Z> AND *Y> RESP.       C  MAGTEL  124
C                       DEF0                                   C  MAGTEL  125
C                        *ALPC>=*ZXX*ZYY>/*ZXY-ZYX>            C  MAGTEL  126
C                        (INDEPENDENT OF *A>).                 C  MAGTEL  127
C  *BTAC(I,J)> - J=1,3- TENSOR ELLIPTICITY FOR *Z>,*Y>,*YZ>,    C  MAGTEL  128
C                       RESP. DEF0                             C  MAGTEL  129
C                        *BTAC>=*YZX(A)>/*YZY(A)>,*A>=*A(YZ)>  C  MAGTEL  130
C  *KMMC(I,J)> - J=1,2- NO.OF INDEPENDENT SOLUTIONS OF          C  MAGTEL  131
C                       *Z> AND *Y> RESP ACCEPTED AND AVGD     C  MAGTEL  132
C                       TOGETHER - USING *DELC> ACCEPTANCE TEST C  MAGTEL  133
C   *KZE(I,J)> - J=1,2- *KZX(A)>,*KZY(A)> RESP FOR *A>=*A(KZ)> C  MAGTEL  134
C                       (EQUATION I-8 OF *MAGTAN1>)            C  MAGTEL  135
C   *AKE(I,J)> - J=1,2- PHASE FOR *KZX(A)>,*KZY(A)>,*A>=*A(KZ)> C  MAGTEL  136
C     *COK(I)> -        (HZ-HX) COHERENCY FOR *A>=*A(KZ)>      C  MAGTEL  137
C     *ANK(I)> -        *A(KZ)> FOR *KZ> TENSOR                C  MAGTEL  138
C     *BTAK(I)> -       *KZ> TENSOR ELLIPTICITY.               C  MAGTEL  139
C                       DEF0 *BTAK> = *KZY(A)>/*KZX(A)>,       C  MAGTEL  140
C                                    *A>=*A(KZ)>               C  MAGTEL  141
C                                                              C  MAGTEL  142
C                                                              C  MAGTEL  143
C  NOTE--THE FOLLOWING ARRAYS PERTAIN TO ROTATION OF *ZXX>     C  MAGTEL  144
C        AND *ZXY> BY 10 DEGREE INCREMENTS FROM *A>=-80 DEG    C  MAGTEL  145
C        TO *A>=+90 DEG FOR EACH FREQ VALUE.                   C  MAGTEL  146
C                                                              C  MAGTEL  147
C  *IXXC(I,J)> - J=1,18- APP RES FOR *ZXX(A)>, -80≠A≠+90 DEGR  C  MAGTEL  148
C                       IN 10 DEGR INCR. (DIVIDED BY 10**IXXC) C  MAGTEL  149
C  *IXYC(I,J)> - J=1,18- APP RES FOR *ZXY(A)>, -80≠A≠+90 DEGR  C  MAGTEL  150
C                       IN 10 DEGR INCR. (DIVIDED BY 10**IXYC) C  MAGTEL  151
C   *IEXXC(I)> - DECIMAL EXPONENT FOR *IXXC>.                  C  MAGTEL  152
C   *IEXYC(I)> - DECIMAL EXPONENT FOR *IXYC>.                  C  MAGTEL  153
C                                                              C  MAGTEL  154
C  NOTE--REFER TO REFERENCE(S) GIVEN IN *MAGTAN1> FOR MORE     C  MAGTEL  155
C        DETAILED DESCRIPTION OF THE MT THEORY AND COMPUTATIONS C  MAGTEL  156
C                                                              C  MAGTEL  157
C                                                              C  MAGTEL  158
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC  MAGTEL  159
```