**THE NEWCASTLE GEOTHERMAL SYSTEM
IRON COUNTY, UTAH**

GEOLOGY, HYDROLOGY, AND CONCEPTUAL MODEL

VOLUME 2: APPENDICES

Prepared for:

**The U.S. Department of Energy
Geothermal Technology Division**

**State Cooperative Geothermal Research
Grant Number: DE-FG07-88ID12756**

Prepared by:

**Robert E. Blackett^{1/}, Michael A. Shubat^{1/}, David S. Chapman^{2/},
Craig B. Forster^{2/}, Charles M. Schlinger^{2/}, Charles E. Bishop^{1/}**

March 1990

**^{1/} Utah Department of Natural Resources
Geological and Mineral Survey**

**^{2/} University of Utah
Department of Geology and Geophysics**

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

APPENDIX A

Fault Slip Data

This appendix contains raw data used in the fault slip analysis. Data was collected from four sites, sites A through D. Minor fault slip measurements are listed for each site, and each row of data is one measurement. The index number is an arbitrary sequential number. Strike is the strike of the fault plane, measured in the northern hemisphere. Dip is the dip of the fault plane, which has two letters attached to the end showing the quadrant of the dip direction. Rake is the rake of the slickenside in the plane of the fault, which has two letters attached to the end showing the quadrant of the plunge direction of the rake. Sense is the sense of slip of the fault: N = normal (rake > 45°), R = reverse (rake > 45°), D = dextral (rake < 45°), S = sinistral (rake < 45°).

MASTER

INDEX #	STRIKE	SITE A DIP	RAKE	SENSE
1	N66E	32NW	30SW	S
2	N88W	67NE	19NW	S
3	N26W	56SW	05SE	D
4	N27W	62SW	06SE	D
5	N46W	65SW	08SE	D
6	N33W	86SW	05NW	D
7	N35W	77SW	05SE	D
8	N88E	63SE	13NE	S
9	N54W	27NE	58SE	N
10	N44W	88SW	05SE	D
11	N45W	31NE	67SE	N
12	N78E	81SE	02SW	S
13	N77E	74SE	03SW	S
14	N43W	88NE	11NW	D
15	N32W	57SW	72SE	N
16	N65E	68SE	08NE	S
17	N33W	89SW	16NW	D
18	N24E	82SE	52SW	N
19	N22W	47NE	21SE	D
20	N48W	69NE	04NW	D
21	N32W	78SW	14NW	D
22	N44W	42NE	04SE	D
23	N33W	78SW	24NW	D
24	N56E	34SE	52NE	N
25	N22E	70SE	25NE	D
26	N21E	11SE	82SW	N
27	N22W	20NE	35SE	D
28	N21W	68SW	19NW	D
29	N60E	49NW	54SW	N
30	N35W	88SW	07NW	D
31	N43W	60SW	22NW	D
32	N11E	86SE	17NE	S
33	N25W	68SW	22NW	D
34	N69W	73SW	13NW	D
35	N32W	62SW	22NW	D
36	N82E	24NW	71NE	R
37	N43E	31NW	67NE	N
38	N75W	35NE	37NW	S
39	N43E	24NW	89SW	R
40	N24E	34NW	61NE	R
41	N06E	78NW	02SW	D
42	N74W	56SW	04SE	D
43	N64W	69SW	08NW	D
44	N34E	84SE	04NE	S
45	N85W	32SW	17SE	D
46	N81W	56SW	06NW	D
47	N26W	44SW	47NW	N
48	N81E	34NW	79NE	R
49	N46W	47SW	08SE	D
50	N20E	42SE	32NE	S
51	N39E	74SE	15NE	S

INDEX #	STRIKE	DIP	RAKE	SENSE
52	N08E	79NW	56SW	N
53	N57W	84SW	21SE	D
54	N48E	80SE	10SW	S
55	N11W	12NE	42SE	S
56	N42E	20NW	46SW	N
57	N47E	14NW	57SW	N
58	N47W	39SW	16NW	D
59	N34W	75SW	24NW	D
60	N08W	29NE	16SE	D
61	N73E	28NW	68SW	N
62	N47W	74NE	13NW	D
63	N32W	86SW	18NW	D
64	N61W	64SW	03SE	D
65	N13W	34NE	12NW	S
66	N47E	18SE	78NE	N
67	N35W	74NE	19NW	D
68	N48E	85SE	03SW	S
69	N44W	71NE	19NW	D
70	N38W	81NE	09NW	D
71	N27E	73NW	03SW	S
72	N08W	84NE	38NW	S
73	N24E	89SE	36NE	S
74	N04W	15NE	66SE	N
75	N44E	19SE	56SW	N
76	N22W	67NE	86SE	N
77	N40W	89SW	21NW	D
78	N10E	88SE	27NE	S
79	N43W	79NE	14SE	D
80	N60W	51SW	06NW	D
81	N06E	86NW	35NE	S
82	N14W	32NE	69SE	N
83	N54W	78SW	13NW	D
84	N20E	08NW	48NE	N
85	N68W	85NE	15NW	D
86	N02W	84NE	15NW	S
87	N65W	80NE	14NW	D
88	N22E	80SE	06NE	S
89	N76W	86NE	36NW	D
90	N81W	79SW	02NW	D
91	N60W	86SW	25NW	D
92	N12W	21NE	14SE	D
93	N41W	76SW	16NW	D
94	N36E	54NW	55NE	N
95	N12E	19SE	77SW	R
96	N15E	88SE	36NE	S
97	N48W	30NE	18NW	D
98	N15W	82NE	28NW	S
99	N78W	65SW	12NW	D
100	N82W	51SW	11SE	D
101	N76W	81SW	08SE	D
102	N42W	66SW	28NW	D
103	N51W	72SW	16NW	D
104	N05W	84SW	18NW	S

INDEX #	STRIKE	DIP	RAKE	SENSE
105	N48W	79SW	33NW	D
106	N35W	33SW	46NW	N
107	N24E	59NW	50NE	R
108	N60W	77SW	34NW	D

INDEX #	STRIKE	SITE B DIP	RAKE	SENSE
1	N15W	46NE	03NW	S
2	N51W	78NE	37NW	D
3	N61W	89NE	32NW	D
4	N64W	88SW	44NW	D
5	N82W	60NE	46NW	R
6	N10W	29NE	13SE	S
7	N75W	85SW	34NW	D
8	N15W	42SW	13SE	S
9	N15E	54SE	15SW	S
10	N51W	32SW	10SE	D
11	N54W	71SW	14NW	D
12	N16E	63NW	64NE	N
13	N06W	84SW	16NW	S
14	N55W	67SW	25NW	D
15	N55W	63SW	52NW	N
16	N81W	83SW	29NW	D
17	N34E	58NW	87NE	N
18	N89W	88SW	26NW	D
19	N56W	73SW	14SE	D
20	N70E	84NW	44SW	S
21	N58E	88NW	43SW	S
22	N70W	79NE	43NW	D
23	N61E	82SE	49SW	R
24	N84E	59NW	43SW	D
25	N71W	86SW	33NW	D
26	N22W	37SW	76SE	N
27	N59E	87NW	13NE	S
28	N60E	34NW	84NE	N
29	N76E	85SE	10SW	S
30	N39E	32NW	64NE	R
31	N63E	87SE	08SW	S
32	N26W	77NE	22NW	D
33	N22W	68SW	10NW	D
34	N34E	84SE	24SW	S
35	N32E	88NW	39SW	S
36	N46E	24NW	74NE	N
37	N72E	29NW	76SW	N
38	N31E	36SE	36SW	S
39	N30E	85SE	18SW	S
40	N45E	77NW	02NE	S
41	N52W	35SW	29SE	D
42	N34W	79SW	06NW	D
43	N37E	65SE	15SW	S
44	N62W	37SW	36SE	D

INDEX	#	STRIKE	DIP	RAKE	SENSE
	45	N16E	62NW	05NE	S
	46	N31E	66NW	11SW	S
	47	N28E	80NW	12NE	S
	48	N63E	79SE	14SW	S
	49	N42W	86NE	02NW	D
	50	N60E	72SE	21SW	S
	51	N32E	73SE	17SW	S
	52	N14E	56SE	02NE	S
	53	N33W	84NE	13NW	D
	54	N87E	43SE	42NE	D
	55	N59E	88NW	02NE	S
	56	N53E	69NW	14SW	S
	57	N13W	44SW	77SE	N
	58	N03E	78NW	80NE	N
	59	N72E	46SE	11SW	S
	60	N46E	71SE	23SW	S
	61	N19E	88NW	03NE	S
	62	N38W	87SW	10NW	D
	63	N32E	64SE	13NE	S
	64	N11W	78SW	56NW	N
	65	N16E	74NW	57SW	N
	66	N44E	38SE	49NE	N
	67	N38W	85NE	09SE	D
	68	N10W	69NE	38NW	S
	69	N45W	75SW	19NW	D
	70	N11E	77SE	07SW	S
	71	N10E	71NW	04NE	S
	72	N23E	49SE	30SW	S
	73	N47W	77SW	13NW	D
	74	N09E	84NW	04SW	S
	75	N61E	86NW	12NE	S
	76	N36W	74NE	12NW	D
	77	N55W	56NE	17NW	D
	78	N36E	68SE	32NE	S
	79	N25E	34SE	42SW	S
	80	N27W	81SW	67SE	N
	81	N15E	60SE	14NE	S
	82	N49W	89NE	27SE	D
	83	N59W	31SW	18SE	D
	84	N22E	62SE	18NE	S
	85	N48E	77NW	04NE	S
	86	N44E	61SE	05NE	S
	87	N72W	30SW	57SE	R
	88	N11E	70SE	11SW	S
	89	N57E	26SE	89SW	R
	90	N59W	42SW	24SE	D
	91	N28W	61SW	03SE	D
	92	N40W	74SW	03NW	D
	93	N56E	76SE	14SW	S
	94	N22E	57NW	09NE	S
	95	N23E	86NW	12NE	S
	96	N59E	82NW	27NE	S
	97	N35E	69NW	13SW	S

INDEX #	STRIKE	DIP	RAKE	SENSE
98	N05E	79SE	07SW	S
99	N44W	77NE	04NW	D
100	N60W	39SW	18SE	D
101	N41W	72SW	86NW	N
102	N05E	20SE	62NE	R
103	N45W	80SW	79SE	N
104	N11E	89NW	34SW	S
105	N17E	59NW	58SW	N
106	N35W	62NE	26SE	D
107	N24W	39SW	70NW	N
108	N05W	65SW	68SE	N
109	N81E	55NW	73NE	N
110	N06W	68SW	44NW	D
111	N66W	28NE	72NW	N
112	N08W	57SW	47SE	N
113	N09E	75NW	26SW	S
114	N24E	84NW	17SW	S
115	N21E	75NW	14SW	S
116	N72W	81NE	23NW	D
117	N27W	69SW	04NW	D
118	N26E	76NW	08SW	S
119	N42W	82SW	13NW	D
120	N13W	74SW	62NW	N
121	N33E	78SE	04NE	S
122	N05W	86SW	77NW	N
123	N10E	85NW	78NE	N
124	N21E	83NW	70NE	N
125	N05E	89NW	79SW	N
126	N12E	77SE	14SW	S
127	N47E	69SE	15NE	S
128	N25W	67NE	30NW	D
129	N54W	41NE	21SE	D
130	N39E	69SE	43SW	S
131	N21E	78SE	16SW	S
132	N28W	79NE	43NW	D
133	N30E	66SE	13NE	S
134	N44E	70SE	13SW	S

INDEX #	STRIKE	SITE C DIP	RAKE	SENSE
1	N52E	48SE	79SW	N
2	N48W	78SW	78SE	N
3	N46E	68NW	83NE	N
4	N72E	80NW	03NE	D
5	N87E	83NW	05NE	D
6	N21E	88SE	71NE	N
7	N47E	59SE	63NE	N
8	N31W	67SW	32SE	S
9	N55E	67NW	83SW	N
10	N74E	81NW	11NE	D
11	N39E	86SE	83NE	N

INDEX #	STRIKE	DIP	RAKE	SENSE
12	N27E	83NW	44NE	D
13	N71E	59NW	81NE	N
14	N39E	61NW	72NE	N
15	N46E	85NW	61NE	N
16	N88W	71NE	31SE	D
17	N88W	71NE	68NW	N
18	N51E	63SE	62NE	N
19	N62E	54SE	79NE	N
20	N71E	74NW	31NE	D

INDEX #	STRIKE	SITE D DIP	RAKE	SENSE
1	N83E	47SE	63NE	N
2	N58E	51SE	64NE	N
3	N46E	78SE	34NE	S
4	N76E	41NW	02NE	S
5	N74W	75NE	13NW	D
6	N52E	89SE	64NE	N
7	N55E	89NW	46NE	N
8	N64W	22NE	22SE	D
9	N39E	57SE	49NE	N
10	N46W	81NE	20SE	D
11	N73E	74NW	49SW	N
12	N66W	40NE	11SE	D
13	N65W	35NE	74SE	N
14	N52W	40NE	05SE	D
15	N86W	73NE	05NW	D
16	N36E	66NW	25NE	S
17	N55W	65SW	16SE	D
18	N74W	43NE	14SE	D
19	N81W	47NE	21NW	D
20	N86W	70NE	04NW	D
21	N02E	86NW	74NE	R
22	N73W	16NE	70NW	R
23	N38E	60NW	19NE	S
24	N29W	52NE	10NW	D
25	N45E	78SE	70NE	N
26	N34W	60NE	18SE	D
27	N20E	76SE	80NE	N
28	N25E	79SE	68SW	N
29	N06W	62NE	04NW	S
30	N38W	67NE	14SE	D
31	N19W	44NE	19SE	D
32	N60W	43SW	22SE	D
33	N45W	24SW	34SE	D
34	N24W	76SW	26NW	D
35	N15W	36NE	06SE	D
36	N89W	65SW	67NW	R
37	N15W	84NE	08NW	D
38	N39E	22SE	24SW	S
39	N21E	57SE	66NE	N

INDEX #	STRIKE	DIP	RAKE	SENSE
40	N04W	52NE	30SE	S
41	N47W	17NE	12NW	D
42	N29W	84NE	87NW	R
43	N78W	80SW	13NW	D
44	N35E	33SE	65NE	N
45	N55W	86NE	21SE	D
46	N01W	82NE	44NW	S
47	N73W	64SW	12NW	D
48	N78W	87NE	46NW	R
49	N34W	24SW	17NW	D
50	N68E	48NW	11NE	S
51	N89E	72SE	12SW	S
52	N62E	67SE	08SW	S
53	N52E	41SE	28NE	D
54	N89W	52SW	89NW	N
55	N81W	77SW	06SE	D

APPENDIX B

Computer Programs Written for Fault Slip Analysis

The first program, INVERT, was written to verify the methods and software of Reches (1987), and the algorithm used in this program closely follows that of Reches (1987). The reader is referred to his article for additional information. The goal of the program is to fit a reduced stress tensor to a population of fault slip measurements.

- Input:
- 1) A data set of fault slip measurements in the format produced by the REFORMAT program.
 - 2) Cohesion - If familiar with rock mechanics and have some knowledge of this parameter in your study area, then supply this number. Otherwise, enter a small value (0.01).
 - 3) Coefficient of friction - start with small values (0.01) and then on subsequent trials increase this value to see if the solution improves.

- Output:
- 1) The number of faults.
 - 2) The relative eigenvalues of the principal axes of the solution tensor.
 - 3) The trend and plunge of the three principal axes of the solution tensor, sigma 1, sigma 2, and sigma 3.
 - 4) The phi value of the solution tensor where $\phi = (\sigma_2 - \sigma_3) / (\sigma_1 - \sigma_3)$.
 - 5) The average misfit of solution tensor, which is a measure of the "goodness of fit." Average misfit is the average of the angle between the observed slip on a fault and the predicted slip (using the solution tensor) for each fault in the set.

- Notes:
- 1) As written, the program must be run on an IBM compatible PC with a math co-processor chip.
 - 2) The program can be modified to display the chi-square statistic, a measure of probable error, and many error-related parameters (see Press and others, 1986).
 - 3) This and the other programs in this appendix were written using Turbo Pascal version 4.0.
 - 4) If you find any errors or would like an executable copy of this and the other programs, please contact: Michael Shubat, Utah Geological and Mineral Survey, 606 Black Hawk Way, Salt Lake City, Utah 84108.

```

{$N+}
{$R+}
{Based on the algorithm by Z. Reches. The program uses a)
{frictional constraint, and solves for 5 components of}
{the stress tensor. Sigma33 is scaled to 100}
{$M 42000,0,655360}
program INVERT;
uses
  EigenVal;
const
  values = 6;
  columns = 5;
  zero = 0.0000000000000000;
  one = 1.0000000000000000;
  maxnum = 270;
type
  afilename = string[12];
  datapoint = array[1..values] of Float;
  datavector = array[1..maxnum] of datapoint;
  glndata = array[1..maxnum] of Float;
  glmma = array[1..columns] of Float;
  glmpbynp = array[1..maxnum,1..columns] of Float;
  glnpbynp = array[1..columns,1..columns] of Float;
  glnparray = glmma;
  glmparray = glndata;
  TandP = array[1..2] of Float;
var
  data : datavector;
  stddev, answers : glndata;
  unknowns: glmma;
  nonsing: glmpbynp;
  V : glnpbynp;
  W : glnparray;
  extrarows, mma, n, np, mp, i, j : integer;
  mu, coh, stat, phi, misfit, min : Float;
  sig1, sig2, sig3, test : TNvector;
  problem : boolean;
  A, B, C : TandP;
  filename : afilename;
  thisfile : text;

function Dot(v1, v2 : TNvector) : Float;
begin
  Dot := v1[1]*v2[1]+v1[2]*v2[2]+v1[3]*v2[3];
end;

procedure ZerotestR(var z:Float);
begin
  if (z = zero) then
    z := 0.0000000000000001;
end;

procedure D2R(m:integer; var n:Float);
const

```

```

    rad = 0.01745329;
begin
    n := m * rad;
end;

```

```

procedure R2D(m:Float; var n:integer);
const
    deg = 57.29577951;
var
    temp : Float;
begin
    temp := m * deg;
    n := Round(temp);
end;

```

```

procedure DCToTP(Q:TNvector; var result:TandP);
var
    step1, step2, step3, signx, signy : Float;
    quad : integer;
begin
    step1 := Q[3] * Q[3];
    step2 := Sqrt(1 - step1);
    ZerotestR(step2);
    step3 := Q[3] / step2;
    result[2] := ArcTan(step3);
    ZerotestR(Q[1]);
    ZerotestR(Q[2]);
    result[1] := ArcTan(Q[2] / Q[1]);
    signx := Q[1] / Abs(Q[1]);
    signy := Q[2] / Abs(Q[2]);
    if signx > 0 then if signy > 0 then quad := 1
                        else quad := 4
                        else if signy > 0 then quad := 2
                        else quad := 3;

    case quad of
        2 : result[1] := result[1] + 3.14159265;
        3 : result[1] := result[1] + 3.14159265;
        4 : result[1] := result[1] + 6.28318531;
    end;
end;

```

```

procedure TPtoDC(this:TandP; var that:TNvector);
begin
    that[1] := Cos(this[2]) * Cos(this[1]);
    that[2] := Cos(this[2]) * Sin(this[1]);
    that[3] := Sin(this[2]);
end;

```

```

procedure Cross(A, B:TNvector; var C:TNvector);
begin
    C[1] := (A[2] * B[3]) - (A[3] * B[2]);
    C[2] := (A[3] * B[1]) - (A[1] * B[3]);
    C[3] := (A[1] * B[2]) - (A[2] * B[1]);
end;

```

```

procedure GetNormal(plane:TandP; var G:TNvector);
var

```

```

    str          : TandP;
    dipdir, strikdir : TNvector;
begin
    str[1] := 1.57079633 + plane[1];
    str[2] := zero;
    TPtoDC(plane, dipdir);
    TPtoDC(str, strikdir);
    Cross(dipdir, strikdir, G);
end;

```

```

procedure Normalize(var H:TNvector);
var

```

```

    i      : integer;
    k, inv : Float;
begin
    k := zero;
    for i := 1 to 3 do k := k + sqr(H[i]);
    k := Sqrt(k);
    ZerotestR(k);
    inv := one / k;
    for i := 1 to 3 do H[i] := inv * H[i];
end;

```

```

procedure GetData(var numbers : datavector;
                  var n       : integer;
                  var afile   : text;
                  var name    : filename);

```

```

var
    dip, dipdir, pl, tr, i, j : integer;
    S, norm                  : TNvector;
    ss, nn                  : TandP;

```

```

begin
    Write('enter the name of the data file: ');
    Readln(name);
    Assign(afile,name);
    Reset(afile);
    Readln(afile,n);
    for i := 1 to n do
        begin
            Readln(afile, dip, dipdir, pl, tr);
            D2R(tr, ss[1]);
            D2R(pl, ss[2]);
            TPtoDC(ss,S);
            numbers[i][1] := S[1];
            numbers[i][2] := S[2];
            numbers[i][3] := S[3];
            D2R(dipdir,nn[1]);
            D2R(dip,nn[2]);
            GetNormal(nn,norm);
            numbers[i][4] := norm[1];
        end
    end

```

```

    numbers[i][5] := norm[2];
    numbers[i][6] := norm[3];
end;
Close(afile);
for i := n+1 to 2*n do
    for j := 1 to 6 do numbers[i,j] := numbers[i-n,j];
end;

procedure func(      x      : datapoint;
                 var p      : glmma;
                 n, index, mma : integer);

var
    n1, n2, n3, s1, s2, s3 : Float;
begin
    s1 := x[1];
    s2 := x[2];
    s3 := x[3];
    n1 := x[4];
    n2 := x[5];
    n3 := x[6];
    if (index <= n-extrarows) then
    begin
        p[1] := -s2*n1*n3+s3*n2*n1;           {coeff for sigma11}
        p[2] := s1*n2*n3-s3*n2*n1;           {coeff for sigma22}
        p[3] := s1*n1*n3-s2*n2*n3+s3*(n2*n2+n1*n1); {coeff for tau12}
        p[4] := -s1*n1*n2+s2*(n1*n1-n3*n3)+s3*n3*n2; {coeff for tau13}
        p[5] := s1*(n3*n3-n2*n2)+s2*n2*n1-s3*n3*n1; {coeff for tau23}
    end;
    if (index > n-extrarows) then
    begin
        p[1] := n1*s1-mu*n1*n1;
        p[2] := n2*s2-mu*n2*n2;
        p[3] := n1*s2+s1*n2-2.0*mu*n1*n2;
        p[4] := n1*s3+s1*n3-2.0*mu*n1*n3;
        p[5] := n2*s3+s2*n3-2.0*mu*n2*n3;
    end;
end;

procedure svdcmp(var      a:glmpbynp;
                 m, n, mp, np:integer;
                 var      w:glnparray;
                 var      v:glnpbynp);

label 1, 2, 3;
const
    nmax = 100;
var
    nm, l, k, j, its, i, jj : integer;
    x, y, z, scale, s, h, g, f, c, anorm : Float;
    rv1 : array[1..nmax] of Float;

function sign(a, b:Float) : Float;
begin
    if b >= zero then sign := abs(a) else sign := -abs(a)
end;

```



```

function max(a,b:Float) : Float;
begin
  if (a > b) then max:= a else max := b
end;

begin
  g := zero;
  scale := zero;
  anorm := zero;
  for i := 1 to n do
    begin
      l := i + 1;
      rv1[i] := scale * g;
      g := zero;
      s := zero;
      scale := zero;
      if (i <= m) then
        begin
          for k := i to m do
            begin
              scale := scale + abs(a[k,i])
            end;
          if (scale <> zero) then
            begin
              for k := i to m do
                begin
                  a[k,i] := a[k,i] / scale;
                  s := s + a[k,i] * a[k,i]
                end;
              f := a[i,i];
              g := -sign(Sqrt(s), f);
              h := f * g - s;
              a[i,i] := f - g;
              if (i <> n) then
                begin
                  for j := 1 to n do
                    begin
                      s := zero;
                      for k := i to m do
                        begin
                          s := s + a[k,i] * a[k,j]
                        end;
                      f := s / h;
                      for k := i to m do
                        begin
                          a[k,j] := a[k,j] + f * a[k,i]
                        end
                      end
                    end;
                  end;
                for k := i to m do
                  begin
                    a[k,i] := scale * a[k,i]
                  end
                end
              end
            end
          end
        end
      end
    end
  end
end

```

```

    end
end;
w[i] := scale * g;
g := zero;
s := zero;
scale := zero;
if ((i <= m) and (i <> n)) then
begin
    for k := 1 to n do
    begin
        scale := scale + abs(a[i,k])
    end;
    if (scale <> zero) then
    begin
        for k := 1 to n do
        begin
            a[i,k] := a[i,k] / scale;
            s := s + a[i,k] * a[i,k]
        end;
        f := a[i,1];
        g := -sign(sqrt(s), f);
        h := f * g - s;
        a[i,1] := f - g;
        for k := 1 to n do
        begin
            rv1[k] := a[i,k] / h
        end;
        if (i <> m) then
        begin
            for j := 1 to m do
            begin
                s := zero;
                for k := 1 to n do
                begin
                    s := s + a[j,k] * a[i,k]
                end;
                for k := 1 to n do
                begin
                    a[j,k] := a[j,k] + s * rv1[k]
                end
            end
        end;
        for k := 1 to n do
        begin
            a[i,k] := scale * a[i,k]
        end
    end
end;
anorm := max(anorm, (abs(w[i]) + abs(rv1[i])))
end;
for i := n downto 1 do
begin
    if (i < n) then
    begin

```

```

if (g <> zero) then
begin
  for j := 1 to n do
  begin
    v[j,i] := (a[i,j] / a[i,1]) / g
  end;
  for j := 1 to n do
  begin
    s := zero;
    for k := 1 to n do
    begin
      s := s + a[i,k] * v[k,j]
    end;
    for k := 1 to n do
    begin
      v[k,j] := v[k,j] + s * v[k,i]
    end
  end
end;
for j := 1 to n do
begin
  v[i,j] := zero;
  v[j,i] := zero
end
end;
v[i,i] := one;
g := rv1[i];
l := i
end;
for i := n downto 1 do
begin
  l := i + 1;
  g := w[i];
  if (i < n) then
  begin
    for j := 1 to n do
    begin
      a[i,j] := zero
    end
  end;
end;
if (g <> zero) then
begin
  g := one / g;
  if (i <> n) then
  begin
    for j := 1 to n do
    begin
      s := zero;
      for k := 1 to m do
      begin
        s := s + a[k,i] * a[k,j]
      end;
      f := (s / a[i,i]) * g;
      for k := i to m do

```

```

        begin
            a[k,j] := a[k,j] + f * a[k,i]
        end
    end
end;
for j := i to m do
begin
    a[j,i] := a[j,i] * g
end
end
else
begin
    for j := i to m do
    begin
        a[j,i] := zero
    end
end;
a[i,i] := a[i,i] + one
end;
for k := n downto 1 do
begin
    for its := 1 to 30 do
    begin
        for l := k downto 1 do
        begin
            nm := l - 1;
            if ((abs(rv1[l]) + anorm) = anorm) then goto 2;
            if ((abs(w[nm]) + anorm) = anorm) then goto 1
        end;
1:      c := zero;
        s := one;
        for i := 1 to k do
        begin
            f := s * rv1[i];
            if ((abs(f) + anorm) <> anorm) then
            begin
                g := w[i];
                h := Sqrt(f * f + g * g);
                w[i] := h;
                h := one / h;
                c := (g * h);
                s := -(f * h);
                for j := 1 to m do
                begin
                    y := a[j,nm];
                    z := a[j,i];
                    a[j,nm] := (y * c) + (z * s);
                    a[j,i] := -(y * s) + (z * c)
                end
            end
        end;
end;
2:      z := w[k];
        if (l = k) then
        begin

```

```

    if (z < zero) then
    begin
        w[k] := -z;
        for j := 1 to n do
        begin
            v[j,k] := -v[j,k]
        end
    end;
    goto 3
end;
if (its = 30) then
begin
    Writeln('no convergence in 30 svdcmp iterations');
    Readln
end;
x := w[1];
nm := k - 1;
y := w[nm];
g := rv1[nm];
h := rv1[k];
f := ((y - z) * (y + z) + (g - h) * (g + h)) / (2.0*h*y);
g := Sqrt(f * f + one);
f := ((x - z) * (x + z) + h * ((y/(f + sign(g,f)))-h))/x;
c := one;
s := one;
for j := 1 to nm do
begin
    i := j + 1;
    g := rv1[i];
    y := w[i];
    h := s * g;
    g := c * g;
    z := Sqrt(f * f + h * h);
    rv1[j] := z;
    c := f / z;
    s := h / z;
    f := (x * c) + (g * s);
    g := -(x * s) + (g * c);
    h := y * s;
    y := y * c;
    for jj := 1 to n do
    begin
        x := v[jj,j];
        z := v[jj,i];
        v[jj,j] := (x * c) + (z * s);
        v[jj,i] := -(x * s) + (z * c)
    end;
    z := Sqrt(f * f + h * h);
    w[j] := z;
    if (z <> zero) then
    begin
        z := one / z;
        c := f * z;
        s := h * z
    end
end

```

```

end;
f := (c * g) + (s * y);
x := -(s * g) + (c * y);
for jj := 1 to m do
begin
  y := a[jj,j];
  z := a[jj,i];
  a[jj,j] := (y * c) + (z * s);
  a[jj,i] := -(y * s) + (z * c);
end
end;
rvl[1] := zero;
rvl[k] := f;
w[k] := x
end;
3: end
end;

procedure svbksb(
                                u:glmpbynp;
                                w:glnparray;
                                v:glnpbynp;
                                m, n, mp, np:integer;
                                b:glmparray;
                                x:glnparray);
var
  jj, j, i, r, c : integer;
  s : Float;
  tmp : glnparray;
begin
  for j := 1 to n do
  begin
    s := zero;
    if (w[j] <> zero) then
    begin
      for i := 1 to m do s := s + u[i,j] * b[i];
      s := s / w[j];
    end;
    tmp[j] := s;
  end;
  for j := 1 to n do
  begin
    s := zero;
    for jj := 1 to n do s := s + v[j,jj] * tmp[jj];
    x[j] := s;
  end;
end;

procedure svdfit(
                                x:datavector;
                                y,sig:glndata;
                                ndata:integer;
                                var   a:glmma;
                                var   mma:integer;
                                var   u:glmpbynp;

```

```

        var      v:glnpbynp;
        var      w:glnparray;
        mp,np:integer;
        var chisq:Float);

const
    tol = 1.0e-12;
var
    i, j, zz, ww      : integer;
    wmax, tmp, thresh, sum : Float;
    b                  : glndata;
    afunc              : glmma;
begin
    for i := 1 to ndata do
        begin
            func(x[i], afunc, ndata, i, mma);
            tmp := one / sig[i];
            for j := 1 to mma do u[i,j] := afunc[j] * tmp;
            b[i] := y[i] * tmp;
        end;
        svdcmp(u, ndata, mma, mp, np, w, v);
        wmax := zero;
        for j := 1 to mma do if (w[j] > wmax) then wmax := w[j];
        thresh := tol * wmax;
        for j := 1 to mma do if (w[j] < thresh) then w[j] := zero;
        svbksb(u, w, v, ndata, mma, mp, np, b, a);
        chisq := zero;
        for i := 1 to ndata do
            begin
                func(x[i], afunc, ndata, i, mma);
                sum := zero;
                for j := 1 to mma do sum := sum + a[j] * afunc[j];
                chisq := chisq + sqr((y[i]-sum) / sig[i]);
            end
        end;
    end;

procedure GetEigenSys(    invector:glmma;
                        var      phi:Float;
                        var      sigma1:TNvector;
                        var      sigma2:TNvector;
                        var      sigma3:TNvector;
                        var trouble:boolean);

var
    tensor, eigenvectors : TNmatrix;
    eigenvalues           : TNvector;
    iter, maxit, dimen, i, j, max, min, mid    : integer;
    toler, x, y           : Float;
    Error                 : byte;

begin
    tensor[1,1] := invector[1]+100.0;
    tensor[1,2] := invector[3];
    tensor[1,3] := invector[4];
    tensor[2,1] := tensor[1,2];
    tensor[2,2] := invector[2]+100.0;

```

```

tensor[2,3] := invector[5];
tensor[3,1] := tensor[1,3];
tensor[3,2] := tensor[2,3];
tensor[3,3] := 100.0;
dimen := 3;
maxit := 200;
toler := 1.0E-12;
trouble := False;
Jacobi(dimen, tensor, maxit, toler, eigenvalues,
      eigenvectors, iter, error);

{order the eigenvalues}
max := 1;
min := 1;
mid := 1;
if eigenvalues[2] > eigenvalues[1] then
begin
  max := 2;
  if eigenvalues[3] > eigenvalues[2] then max := 3;
end
else
begin
  if (eigenvalues[3] > eigenvalues[1]) then max := 3
end;
if eigenvalues[2] < eigenvalues[1] then
begin
  min := 2;
  if eigenvalues[3] < eigenvalues[2] then min := 3
end
else
begin
  if (eigenvalues[3] < eigenvalues[1]) then min := 3
end;
for i := 1 to 3 do if ((max <> i) and (min <> i))
                  then mid := i;
Writeln('eigenvalue[max]= ', eigenvalues[max]:8:4);
Writeln('eigenvalue[mid]= ', eigenvalues[mid]:8:4);
Writeln('eigenvalue[min]= ', eigenvalues[min]:8:4);
sigma1[1] := eigenvectors[max,1];
sigma1[2] := eigenvectors[max,2];
sigma1[3] := eigenvectors[max,3];
sigma2[1] := eigenvectors[mid,1];
sigma2[2] := eigenvectors[mid,2];
sigma2[3] := eigenvectors[mid,3];
sigma3[1] := eigenvectors[min,1];
sigma3[2] := eigenvectors[min,2];
sigma3[3] := eigenvectors[min,3];
phi := (eigenvalues[mid]-eigenvalues[min])/
      (eigenvalues[max]-eigenvalues[min]);
if (error <> 0) then
begin
  Writeln('eigensystem solution did not converge ');
  trouble := True;
end;
end;

```



```

procedure Transform(      M1      : TNmatrix;
                        var M2      : TNmatrix;
                        L          : TNmatrix);

var
  i, j, p, q : integer;
  step1       : Float;
begin
  for i := 1 to 3 do
    for j := 1 to 3 do M2[i,j] := zero;
  for i := 1 to 3 do
    for j := 1 to 3 do
      for p := 1 to 3 do
        for q := 1 to 3 do
          begin
            {M2[i,j] := L[i,p]*L[j,q]*M1[p,q] + M2[i,j];}
            step1 := L[i,p] * L[j,q];
            M2[i,j] := step1 * M1[p,q] + M2[i,j];
          end;
        end;
      end;
    end;
  end;

end;

procedure GetTrac(      Mat  : TNmatrix;
                      norm  : TNvector;
                      var trac : TNvector);

var
  i,j : integer;
begin
  for i := 1 to 3 do trac[i] := zero;
  for i := 1 to 3 do
    for j := 1 to 3 do trac[i] := Mat[i,j]*norm[j]+trac[i];
  end;
end;

function ArcCos(angle : Float) : Float;
var
  denom : Float;
begin
  denom := Sqrt(one-sqr(angle));
  ZerotestR(denom);
  ArcCos := 1.570796326794897-ArcTan(angle/denom);
end;

procedure AverageMisfit(var afile      : text;
                        name           : afilename;
                        phi             : Float;
                        sig1, sig2, sig3 : TNvector;
                        var avgmisfit   : Float);

var
  T, Tnew, L          : TNmatrix;
  S, norm, P, slip, B : TNvector;
  i, j, n, dip, dipdir, tr, pl : integer;
  ss, fault           : TandP;
  angle               : Float;

```

```

begin                                     (start AverageMisfit procedure)
  (build the principal tensor)
  for i := 1 to 3 do for j := 1 to 3 do T[i,j] := zero;
  T[1,1] := one;
  T[2,2] := phi;
  (build the matrix of direction cosines )
  for i := 1 to 3 do L[i,1] := sig1[i];
  for i := 1 to 3 do L[i,2] := sig2[i];
  for i := 1 to 3 do L[i,3] := sig3[i];
  Transform(T,Tnew,L);
  Assign(afile,name);
  Reset(afile);
  Readln(afile,n);
  avgmisfit := zero;
  for i := 1 to n do
  begin
    Readln(afile, dip, dipdir, pl, tr);
    D2R(tr, ss[1]);
    D2R(pl, ss[2]);
    TPtoDC(ss,slip);
    D2R(dipdir, fault[1]);
    D2R(dip, fault[2]);
    GetNormal(fault,norm);
    GetTrac(Tnew,norm,P);
    Normalize(P);
    Cross(norm,P,B);
    Normalize(B);
    Cross(B,norm,S);
    Normalize(S);
    angle := Dot(slip,S);
    angle := ArcCos(angle);
    R2D(angle,j);
    avgmisfit := avgmisfit+j;
  end;
  Close(afile);
  avgmisfit := avgmisfit/n;
  Writeln('average misfit of fitted tensor is:',avgmisfit:8:2);
end;

begin                                     (start main program)
  Writeln('enter value for cohesion:');
  Writeln('(if unsure, enter 0.01)');
  Readln(coh);
  Writeln('enter value for coefficient of friction:');
  Writeln('(if unsure, enter 0.01)');
  Readln(mu);
  mma := 5;
  GetData(data,n, thisfile, filename);
  Writeln(n);
  extrarows := n;
  for i := 1 to n do answers[i] := zero;
  for i := n+1 to n+extrarows do answers[i] := coh+mu*100;
  for i := 1 to n+extrarows do stddev[i] := one;
  svdfit(data,answers,stddev,n+extrarows,unknowns,mma,nonsing,

```

```

                                V,W,n+extrarows,mma,stat);
GetEigenSys(unknowns, phi, sig1, sig2, sig3, problem);
if problem then Writeln('problem with eigen solution');
if not problem then
begin
  Normalize(sig1);
  Normalize(sig2);
  Normalize(sig3);
  {make sure it's a right-handed system}
  Cross(sig1,sig2,test);
  if (Dot(sig3,test) < -0.5) then
    for i := 1 to 3 do sig3[i] := -sig3[i];
  DCToTP(sig1, A);
  DCToTP(sig2, B);
  DCToTP(sig3, C);
  R2D(A[1],i);
  R2D(A[2],j);
  Writeln('sigma one (trend/plunge):' , i:3, '/', j:3);
  R2D(B[1],i);
  R2D(B[2],j);
  Writeln('sigma two (trend/plunge):' , i:3, '/', j:3);
  R2D(C[1],i);
  R2D(C[2],j);
  Writeln('sigma three (trend/plunge):' , i:3, '/', j:3);
  Writeln('phi = ', phi:4:2);
  AverageMisfit(thisfile, filename, phi, sig1, sig2, sig3,
misfit);
  end;
end.

```

The second program, REFORMAT, is designed to take fault slip measurements recorded in the field and translate them into a trend and plunge format where a reverse component of slip is signified by a negative plunge.

Input: An ASCII file in the format displayed in Appendix A.

Output: A file of transformed data that can be used directly with the INVERT program. File will have the same name as the input file, but will have a ".REC" extension.

- Note: 1) The output format, used by Reches (1987) and the program INVERT, is the motion of the hanging wall with respect to the footwall, with negative plunges signifying reverse components of slip.
- 2) This program does not require a math co-processor.

```
program REFORMAT;
```

```
type
```

```
  measure = record
    index      : integer;
    strike     : integer;
    strikeQ    : char;
    dip        : integer;
    dipQ       : string[2];
    rake       : integer;
    rakeQ      : string[2];
    sense      : char;
    trend      : integer;
    plunge     : integer;
    fdipdir    : integer;
    plungel    : integer;
    trendl     : integer;
    bdip       : integer;
    bdipdir    : integer;
  end;
```

```
var
```

```
  fault : array[1..1000] of measure;
  infile, outfile : text;
  n      : integer;
  c      : char;
  bed    : boolean;
  name   : string[12];
```

```
procedure GetData (var l:integer);
```

```
var
```

```
  dummy      : string[2];
  i, result  : integer;
  c          : char;
```

```
begin
```

```
  Writeln('Enter the name of the file containing input data');
  Writeln('in the format: index number, strike, dip, rake, and
                                     sense. ');
  Writeln('Each entry on a line must be separated by at least one
                                     space');
  Writeln('and one line of data used for each fault measurement. ');
  Writeln('An example line of data is: 23  N35W 25SW 39SE  N');
  Writeln('Note that N=normal, R=reverse, S=sinistral, and
                                     D=dextral. ');
  Writeln('You should use a spreadsheet to make the data file
                                     and');
```

```
  Writeln('export it as an ASCII file (not delimited). ');
```

```
  Writeln;
```

```
  Write('file name:  ');
```

```
  Readln(name);
```

```
  Assign(infile, name);
```

```
  Reset(infile);
```

```
  l := 0;
```

```
  while not Eof(infile) do
```

```
  begin
```

```
    l := l + 1;
```

```

Read(infile, fault[1].index);
repeat
  Read(infile, c);
until c <> ' ';
dummy := '';
if not(c in ['N','n']) then Writeln('strike not in northern
                                hemisphere, please start over');

for i := 1 to 2 do
begin
  Read(infile, c);
  dummy := dummy + c;
end;
Val(dummy, fault[1].strike, result);
if result > 0 then Writeln('problem in reading strike value',
                          result:3);
Read(infile, fault[1].strikeQ);
repeat
  Read(infile, c);
until c <> ' ';
dummy := '';
repeat
  dummy := dummy + c;
  Read(infile, c);
until c > '9';
Val(dummy, fault[1].dip, result);
if result > 0 then Writeln('problem in reading dip value',
                          result:3);

dummy := '';
dummy := dummy + c;
Read(infile, c);
dummy := dummy + c;
fault[1].dipQ := dummy;
dummy := '';
repeat
  Read(infile, c);
until c <> ' ';
repeat
  dummy := dummy + c;
  Read(infile, c);
until c > '9';
Val(dummy, fault[1].rake, result);
if result > 0 then Writeln('problem in reading rake value',
                          result:3);

dummy := '';
dummy := dummy + c;
Read(infile, c);
dummy := dummy + c;
fault[1].rakeQ := dummy;
repeat
  Read(infile, c);
until c <> ' ';
fault[1].sense := c;
Readln(infile);
end;

```

```
Close(infile);  
end;
```

```
procedure MakeCaps(var l:integer);
```

```
var
```

```
  i : integer;
```

```
begin
```

```
  for i := 1 to l do
```

```
  begin
```

```
    case fault[i].strikeQ of
```

```
      'e' : fault[i].strikeQ := 'E';
```

```
      'w' : fault[i].strikeQ := 'W';
```

```
    end;
```

```
    if fault[i].dipQ = 'ne' then fault[i].dipQ := 'NE';
```

```
    if fault[i].dipQ = 'nw' then fault[i].dipQ := 'NW';
```

```
    if fault[i].dipQ = 'se' then fault[i].dipQ := 'SE';
```

```
    if fault[i].dipQ = 'sw' then fault[i].dipQ := 'SW';
```

```
    if fault[i].rakeQ = 'ne' then fault[i].rakeQ := 'NE';
```

```
    if fault[i].rakeQ = 'nw' then fault[i].rakeQ := 'NW';
```

```
    if fault[i].rakeQ = 'se' then fault[i].rakeQ := 'SE';
```

```
    if fault[i].rakeQ = 'sw' then fault[i].rakeQ := 'SW';
```

```
    case fault[i].sense of
```

```
      'n' : fault[i].sense := 'N';
```

```
      'r' : fault[i].sense := 'R';
```

```
      'd' : fault[i].sense := 'D';
```

```
      's' : fault[i].sense := 'S';
```

```
    end;
```

```
  end;
```

```
end;
```

```
procedure DataCheck(var l:integer);
```

```
var
```

```
  code, i      : integer;
```

```
  error, flag : boolean;
```

```
begin
```

```
  error := False;
```

```
  for i := 1 to l do
```

```
  begin
```

```
    flag := True;
```

```
    code := 0;
```

```
    if not(fault[i].strikeQ in ['E','W']) then
```

```
    begin
```

```
      flag := False;
```

```
      code := 1;
```

```
    end;
```

```
    if not(((fault[i].dipQ = 'NW') or (fault[i].dipQ = 'NE') or  
      (fault[i].dipQ = 'SW') or (fault[i].dipQ = 'SE')) and flag)
```

```
    then
```

```
    begin
```

```
      flag := False;
```

```
      code := 2;
```

```
    end;
```

```
    if not(((fault[i].rakeQ = 'NW') or (fault[i].rakeQ = 'NE') or  
      (fault[i].rakeQ = 'SW') or (fault[i].rakeQ = 'SE')) and
```

```

        flag) then
begin
    flag := False;
    code := 3;
end;
if flag then
begin
    case fault[i].strikeQ of
        'E' : begin
            if not((fault[i].dipQ = 'NW') or (fault[i].dipQ =
                'SE')) then code := 4;
            if not((fault[i].rakeQ = 'NE') or (fault[i].rakeQ
                = 'SW')) then code := 5;
            end;
        'W' : begin
            if not((fault[i].dipQ = 'NE') or (fault[i].dipQ =
                'SW')) then code := 4;
            if not((fault[i].rakeQ = 'NW') or (fault[i].rakeQ
                = 'SE')) then code := 5;
            end;
        end;
        if not(fault[i].rake = 45) then if fault[i].rake > 45 then if
            not(fault[i].sense in ['N', 'R']) then code := 6
            else code := code
            else if
                not(fault[i].sense in ['D', 'S']) then code := 6;
    end;
    case code of
        1 : Writeln('typo in strike in line', i:3);
        2 : Writeln('typo in dip in line', i:3);
        3 : Writeln('typo in rake in line', i:3);
        4 : Writeln('error in strike and dip agreement in line',
            i:3);
        5 : Writeln('error in strike and rake agreement in line',
            i:3);
        6 : Writeln('error in rake and sense agreement in line',
            i:3);
    end;
    if not(flag) or (code > 0) then error := true;
end;
if error then Writeln('there was a data entry error, correct it
    now');
end;

procedure lineate(var l:integer);
const
    DegToRad = 0.01745329252;
var
    i, str          : integer;
    dip, rake, az, pl : real;
    s, sq           : char;
    dq, rq          : string[2];
    r               : boolean;

```



```

begin
  for i := 1 to 1 do
    begin
      dip := fault[i].dip * DegToRad;
      rake := fault[i].rake * DegToRad;
      az := ArcTan(Sin(rake) / Cos(rake) * Cos(dip));
      pl := Sin(rake) * Sin(dip);
      pl := ArcTan(pl / Sqrt(-pl * pl + 1));
      az := az / DegToRad;
      pl := pl / DegToRad;
      str := fault[i].strike;
      sq := fault[i].strikeQ;
      dq := fault[i].dipQ;
      rq := fault[i].rakeQ;
      if (sq = 'E') and (dq = 'SE') and (rq = 'NE') then az := str
        + az;
      if (sq = 'E') and (dq = 'SE') and (rq = 'SW') then az := str
        + 180 - az;
      if (sq = 'E') and (dq = 'NW') and (rq = 'NE') then az := str
        - az;
      if (sq = 'E') and (dq = 'NW') and (rq = 'SW') then az := str
        + 180 + az;
      if (sq = 'W') and (dq = 'SW') and (rq = 'NW') then az := 360
        - str - az;
      if (sq = 'W') and (dq = 'SW') and (rq = 'SE') then az := 180
        - str + az;
      if (sq = 'W') and (dq = 'NE') and (rq = 'NW') then az := 360
        - str + az;
      if (sq = 'W') and (dq = 'NE') and (rq = 'SE') then az := 180
        - str - az;

      if az >= 360 THEN az := az - 360;
      if az < 0 THEN az := az + 360;
      fault[i].trend := Round(az);
      fault[i].plunge := Round(pl);
      if ((sq = 'E') and (dq = 'SE')) then fault[i].fdipdir := str
        + 90;
      if ((sq = 'E') and (dq = 'NW')) then fault[i].fdipdir := str
        + 270;
      if ((sq = 'W') and (dq = 'SW')) then fault[i].fdipdir := 270
        - str;
      if ((sq = 'W') and (dq = 'NE')) then fault[i].fdipdir := 90 -
        str;

      if fault[i].sense = 'S' then s := 'S';
      if fault[i].sense = 'D' then s := 'D';
      if sq = 'W' then if dq = 'NE' then if rq = 'NW' then if s = 'S'
        then r := True
        else r := False
        else if s = 'S'
        then r := False
        else r := True
      else if rq = 'NW' then if s = 'S'
        then r := False
        else r := True
        else if s = 'S'

```

```

else if dq = 'NW' then if rq = 'NE' then if s = 'S'
then r := True
else r := False
then r := False
else r := True
else if s = 'S'
then r := True
else r := False
else if rq = 'NE' then if s = 'S'
then r := True
else r := False
else if s = 'S'
then r := False
else r := True;

if fault[i].sense = 'N' then r := True;
if fault[i].sense = 'R' then r := False;
if r then
begin
    fault[i].plunge1 := fault[i].plunge;
    fault[i].trend1 := fault[i].trend;
end
else
begin
    fault[i].plunge1 := 0 - fault[i].plunge;
    fault[i].trend1 := fault[i].trend + 180;
    if fault[i].trend1 > 360 then fault[i].trend1 :=
        fault[i].trend1-360;
end;
end;
end;

procedure GetDipDir(var l:integer);
var
    i, dipdir, dip : integer;
begin
    Write('enter the dip direction of bedding');
    Readln(dipdir);
    Write('enter the dip:');
    Readln(dip);
    for i := 1 to l do
    begin
        fault[i].bdip := dip;
        fault[i].bdipdir := dipdir;
    end;
end;

procedure WriteFile(var l:integer);
var
    i, j : integer;
begin
    Writeln('the output file will have the extension ".rec"');
    Writeln('and can be used directly in Reches fault slip analysis
        program');
    j := Pos('.', name);

```

```

if j > 0 then Delete(name, j, 4);
name := name + '.rec';
Assign(outfile, name);
Rewrite(outfile);
Writeln(outfile, 1);
for i := 1 to 1 do
    Writeln(outfile, fault[i].dip:10, fault[i].fdipdir:10,
        fault[i].plunge:10, fault[i].trend:10);
Close(outfile);
end;

procedure WriteFile1(var l:integer);
var
    i, j : integer;
begin
    Writeln('the output file will have the extension ".bed" and can
        be used');
    Writeln('directly in Reches fault slip analysis program or RETILT
        program');

    j := Pos('.', name);
    if j > 0 then Delete(name, j, 4);
    name := name + '.bed';
    Assign(outfile, name);
    Rewrite(outfile);
    Writeln(outfile, 1);
    for i := 1 to 1 do
        Writeln(outfile, fault[i].dip:10, fault[i].fdipdir:10,
            fault[i].plunge:10, fault[i].trend:10,
            fault[i].bdip:10, fault[i].bdipdir:10);
    Close(outfile);
end;

begin
    GetData(n);
    MakeCaps(n);
    DataCheck(n);
    Lineate(n);
    Writeln('do you know the bedding orientation?(Y,N)');
    Readln(c);
    bed := False;
    if c in ['Y','y'] then
        begin
            GetDipDir(n);
            bed := True;
        end;
    if bed then WriteFile1(n)
        else WriteFile(n);
end.

```

The third program, SLIP, calculates the predicted slip on a fault from the supplied information. The user supplies the ϕ value and trend and plunge of the first and third principal axes of the reduced stress tensor. This information is obtained from the output of the INVERT program. The SLIP program then prompts you for the orientation of a fault plane. It then computes the predicted slip direction of the hanging wall of the fault. Next, you can enter the observed slip direction of the fault (if known) and it will compute the misfit angle. You can calculate the slip for many faults without having to re-enter the tensor data.

```

{$R+}
program SLIP;
{M. Shubat, Utah Geological and Mineral Survey, Salt Lake City, UT,
84108}
{version 1.0, August 1, 1989, written using Turbo Pascal version
4.0}

uses Crt;
const
  zero = 0.0000000000000000;
  one  = 1.0000000000000000;
type
  matrix = array[1..3, 1..3] of real;
  vector = array[1..3] of real;
  TandP  = array[1..2] of real;
var
  T, Tnew, L           : matrix;
  s1, s2, s3, S, N, B, P : vector;
  answer, fault        : TandP;
  i, j, temp1, temp2   : integer;
  k                    : real;
  ch, res              : char;

procedure R2D(m:real; var n:integer);
const
  deg = 57.29577951;
var
  temp : real;
begin
  temp := m * deg;
  n := Round(temp);
end;

procedure D2R(m:integer; var n:real);
const
  rad = 0.01745329;
begin
  n := m * rad;
end;

procedure TPtoDC(this:TandP; var that:vector);
begin
  that[1] := Cos(this[2]) * Cos(this[1]);
  that[2] := Cos(this[2]) * Sin(this[1]);
  that[3] := Sin(this[2]);
end;

procedure ZerotestR(var z:real);
begin
  if (z > -0.0000000000000001) and (z < 0.0000000000000001) then
    z := 0.0000000000000001;
end;

procedure DCtoTP(Q:vector; var result:TandP);
var

```



```

    Readln(i,j);
    D2R(i,sig3[1]);
    D2R(j,sig3[2]);
    TptoDC(sig3,F3);
end;

```

```

procedure GetNormal(plane:TandP; var G:vector);
var
    str          : TandP;
    dipdir, strikedir : vector;
begin
    str[1] := 1.57079633 + plane[1];
    str[2] := zero;
    TptoDC(plane, dipdir);
    TptoDC(str, strikedir);
    Cross(dipdir, strikedir, G);
end;

```

```

procedure Transform(var M1, M2, L:matrix);
var
    i, j, p, q : integer;
    step1      : real;
begin
    for i := 1 to 3 do for j := 1 to 3 do M2[i,j] := zero;
    for i := 1 to 3 do for j := 1 to 3 do for p := 1 to 3 do for q
                                                := 1 to 3 do
        begin
            {M2[i,j] := L[i,p] * L[j,q] * M1[p,q] + M2[i,j];}
            step1 := L[i,p] * L[j,q];
            M2[i,j] := step1 * M1[p,q] + M2[i,j];
        end;
    end;
end;

```

```

procedure Normalize(var H:vector);
var
    i      : integer;
    k, inv : real;
begin
    k := zero;
    for i := 1 to 3 do k := k + H[i] * H[i];
    k := Sqrt(k);
    ZerotestR(k);
    inv := one / k;
    for i := 1 to 3 do H[i] := inv * H[i];
end;

```

```

procedure GetFault(var R:TandP);
var
    i,j : integer;
begin
    Write('enter the DIP DIRECTION and DIP of a fault plane,
                                                separated by a space: ');
    Readln(i,j);
    D2R(i,R[1]);

```

```

    D2R(j,R[2]);
end;

procedure GetTrac(Mat:matrix; norm:vector; var trac:vector);
var
    i,j : integer;
begin
    for i := 1 to 3 do trac[i] := zero;
    for i := 1 to 3 do for j := 1 to 3 do trac[i] := Mat[i,j] *
        norm[j] + trac[i];
    end;

function ArcCos(angle:real) : real;
var
    denom : real;
begin
    denom := Sqrt(one - angle * angle);
    ZerotestR(denom);
    ArcCos := 1.570796326794897 - ArcTan(angle / denom);
end;

procedure FindMisfit(V:vector);
var
    i,j : integer;
    comp : TandP;
    slk : vector;
    angle : real;
begin
    Write('enter the TREND and PLUNGE of the slick, format ala
        Reches:');
    Readln(i,j);
    D2R(i,comp[1]);
    D2R(j,comp[2]);
    TPToDC(comp,slk);
    angle := slk[1] * V[1] + slk[2] * V[2] + slk[3] * V[3];
    angle := ArcCos(angle);
    R2D(angle,i);
    Writeln('the misfit angle is: ',i:3);
end;

begin
    GetT(T,s1,s3);
    Cross(s1,s3,s2);
    {put s1, s2, and s3 into a right-handed framework by changing s2
        direction}
    for i := 1 to 3 do s2[i] := -s2[i];
    {build the matrix of direction cosines between old and new axes}
    for i := 1 to 3 do L[i,1] := s1[i];
    for i := 1 to 3 do L[i,2] := s2[i];
    for i := 1 to 3 do L[i,3] := s3[i];
    Transform(T,Tnew,L);
    repeat
        GetFault(fault);

```



```

GetNormal(fault,N);
GetTrac(Tnew,N,P);
Normalize(P);
Cross(N,P,B);
Normalize(B);
Cross(B,N,S);
Normalize(S);
DctoTP(S,answer);
R2D(answer[1],temp1);
R2D(answer[2],temp2);
Window(1,1,80,25);
ClrScr;
Writeln('the TREND and PLUNGE of the slip axis is:
        ',temp1:3,temp2:4);
Writeln('(hanging wall w.r.t. footwall)');
Writeln('(negative plunge means a reverse component to slip)');
Writeln;
Write('do you know the observed slick for the fault?(Y/N): ');
Readln(res);
if res in ['Y','y'] then FindMisfit(S);
Writeln;
Write('do you wish to try another fault plane?(Y/N): ');
Readln(ch);
until not(ch in ['Y','y']);
end.

```

APPENDIX C: GRAVITY STATION LOCATIONS

(Note: Easting, Northing and Elev. all in ft)

APPENDIX C: GRAVITY STATION LOCATIONS

<u>Station</u>	<u>Easting</u>	<u>Northing</u>	<u>Elev.</u>	<u>Latitude</u>	<u>Longitude</u>
a	1391171.00	379872.25	5187.87	37.691147	113.604218
b	1391453.62	379736.12	5189.18	37.690792	113.603233
c	1391777.62	379598.72	5190.19	37.690434	113.602104
d	1392066.50	379446.75	5190.87	37.690033	113.601097
b1	1411572.12	360950.50	5733.03	37.640430	113.532310
b2	1411806.75	360797.03	5719.02	37.640026	113.531494
b3	1411984.75	360641.25	5704.88	37.639606	113.530861
b4	1412099.12	360346.03	5678.10	37.638805	113.530449
b5	1412242.25	360129.19	5674.62	37.638218	113.529938
b6	1412411.87	359901.56	5676.91	37.637604	113.529335
b7	1412577.62	359695.19	5653.02	37.637047	113.528748
b8	1412696.37	359455.37	5658.61	37.636395	113.528320
b9	1412911.75	359199.00	5648.64	37.635704	113.527557
b10	1413159.62	358990.75	5651.49	37.635147	113.526688
b11	1413400.62	358775.84	5670.12	37.634571	113.525833
b12	1413638.62	358549.03	5718.70	37.633961	113.525002
b13	1413788.62	358382.41	5763.87	37.633514	113.524467
bb1	1411922.37	360284.91	5693.24	37.638626	113.531052
bb2	1412148.75	360144.53	5680.30	37.638252	113.530258
a0	1408279.25	363587.59	6342.72	37.647476	113.543884
a1	1408384.37	363370.06	6282.79	37.646885	113.543503
a2	1408507.62	363203.31	6224.48	37.646435	113.543060
a3	1408655.25	363038.81	6168.75	37.645992	113.542542
a4	1408775.50	362845.09	6134.94	37.645466	113.542114
a5	1408943.00	362649.69	6071.33	37.644939	113.541519
a6	1409146.75	362518.78	6036.52	37.644592	113.540802
a7	1409291.50	362328.94	6031.91	37.644081	113.540291
a8	1409414.25	362102.47	5988.77	37.643467	113.539848
a9	1409645.50	361886.78	5943.91	37.642887	113.539032
a10	1409788.87	361662.37	5939.52	37.642281	113.538521
a11	1409964.62	361440.97	5879.23	37.641682	113.537903
a12	1410173.50	361162.84	5882.64	37.640930	113.537155
a13	1410381.37	361051.28	5863.78	37.640636	113.536430
a14	1410543.00	360915.28	5838.26	37.640274	113.535866
a15	1410747.75	360724.69	5809.56	37.639763	113.535141
a16	1411040.50	360580.22	5781.99	37.639385	113.534119
a17	1411307.75	360597.47	5754.82	37.639446	113.533195
a18	1411529.12	360306.56	5726.68	37.638660	113.532410
d1	1402116.62	369936.44	5249.46	37.664536	113.565643
d2	1402286.75	369781.12	5252.61	37.664120	113.565048
d3	1402456.37	369613.19	5253.34	37.663670	113.564445
d4	1402603.50	369433.00	5254.35	37.663185	113.563927
d5	1402778.12	369267.50	5256.67	37.662739	113.563309
d6	1402962.50	369126.75	5260.08	37.662365	113.562660
d7	1403150.87	368990.12	5264.61	37.661999	113.562004
d8	1403365.62	368798.62	5269.78	37.661488	113.561241
d9	1403698.00	368663.66	5273.44	37.661137	113.560089
d10	1403873.62	368510.41	5276.05	37.660728	113.559471
d11	1404041.00	368356.94	5279.71	37.660316	113.558876
d12	1404239.00	368138.97	5277.22	37.659729	113.558174

<u>Station</u>	<u>Easting</u>	<u>Northing</u>	<u>Elev.</u>	<u>Latitude</u>	<u>Longitude</u>
d13	1404434.37	367976.34	5276.64	37.659294	113.557487
d14	1404589.37	367823.66	5279.60	37.658886	113.556946
d15	1404824.50	367646.50	5285.60	37.658413	113.556122
d16	1405008.75	367505.47	5296.51	37.658035	113.555473
d17	1405172.50	367377.16	5313.99	37.657692	113.554893
d18	1405333.25	367237.87	5338.40	37.657322	113.554329
d1r	1402123.00	369935.44	5248.03	37.664536	113.565620
f1	1404106.37	368266.09	5280.02	37.660069	113.558647
f2	1404239.00	368454.47	5282.76	37.660595	113.558205
f3	1404380.00	368642.91	5283.58	37.661121	113.557732
f4	1404492.87	368810.28	5282.71	37.661587	113.557350
f5	1404592.25	369016.87	5280.97	37.662163	113.557022
f6	1404665.25	369237.75	5280.91	37.662773	113.556786
f7	1404827.25	369407.69	5283.48	37.663250	113.556244
f8	1404931.62	369588.25	5284.06	37.663750	113.555893
f9	1405059.12	369768.47	5284.87	37.664253	113.555466
f10	1405191.87	369948.62	5285.60	37.664757	113.555023
f11	1405305.50	370127.28	5287.10	37.665253	113.554642
f12	1405351.62	370427.97	5285.41	37.666080	113.554512
f13	1405476.37	370647.31	5286.68	37.666691	113.554092
c1	1408197.25	363807.34	6250.80	37.648075	113.544182
c2	1408150.50	363932.00	6180.55	37.648415	113.544350
c3	1408071.87	363999.16	6126.60	37.648594	113.544624
c4	1408159.00	364211.56	6082.38	37.649181	113.544342
c5	1408108.87	364457.53	6025.09	37.649853	113.544533
c6	1407926.37	364624.22	5974.41	37.650299	113.545181
c7	1407745.75	364808.22	5932.90	37.650795	113.545815
c8	1407523.87	364962.00	5919.42	37.651203	113.546593
c9	1407296.00	365097.03	5833.18	37.651562	113.547394
c10	1407160.37	365319.59	5748.76	37.652164	113.547874
c11	1407009.50	365501.72	5698.34	37.652657	113.548409
c12	1406845.87	365689.81	5651.12	37.653160	113.548988
c13	1406606.00	365913.25	5571.31	37.653759	113.549835
c14	1406401.87	366112.34	5535.87	37.654297	113.550552
c15	1406176.37	366352.25	5490.08	37.654942	113.551353
c16	1405993.62	366530.44	5457.68	37.655418	113.551994
c17	1405837.87	366703.47	5425.37	37.655884	113.552544
c18	1405657.62	366849.19	5402.32	37.656273	113.553177
c19	1405493.87	367011.03	5373.61	37.656708	113.553757
e1	1404002.12	368065.53	5277.23	37.659512	113.558990
e2	1403865.37	367888.22	5273.80	37.659019	113.559448
e3	1403727.37	367699.75	5272.10	37.658493	113.559914
e4	1403588.00	367515.66	5270.58	37.657978	113.560379
e5	1403448.62	367317.94	5271.41	37.657429	113.560844
e6	1403306.25	367131.62	5271.59	37.656906	113.561325
e7	1403162.25	366945.72	5271.45	37.656387	113.561806
e8	1403021.37	366761.25	5272.71	37.655872	113.562279
h1	1405671.12	370965.12	5285.69	37.667576	113.553444
h2	1405850.87	371132.03	5287.56	37.668045	113.552841
h3	1406011.00	371397.09	5286.21	37.668781	113.552307
h4	1406167.12	371626.72	5282.73	37.669422	113.551781

<u>Station</u>	<u>Easting</u>	<u>Northing</u>	<u>Elev.</u>	<u>Latitude</u>	<u>Longitude</u>
h5	1406314.75	371738.78	5286.98	37.669739	113.551285
h6	1406502.87	371880.97	5288.31	37.670139	113.550644
h7	1407022.12	372122.53	5289.99	37.670834	113.548866
t1	1401949.00	370104.28	5245.24	37.664986	113.566238
t2	1401747.12	370301.37	5243.76	37.665516	113.566948
t3	1401522.00	370499.22	5241.09	37.666046	113.567741
t4	1401271.75	370727.44	5237.62	37.666656	113.568626
t5	1401040.37	370932.75	5234.59	37.667206	113.569435
t6	1400843.12	371102.22	5235.66	37.667660	113.570129
t7	1400646.87	371320.78	5234.25	37.668247	113.570824
t8	1400453.75	371507.75	5231.86	37.668751	113.571510
t9	1400267.50	371722.41	5231.29	37.669327	113.572166
t10	1400030.12	371927.87	5228.74	37.669876	113.573006
t11	1399809.50	372133.87	5227.53	37.670429	113.573784
t12	1399609.87	372336.19	5226.10	37.670971	113.574486
t13	1399467.75	372465.44	5226.28	37.671318	113.574989
t14	1399256.87	372673.59	5221.62	37.671879	113.575729
t15	1399038.50	372868.44	5220.21	37.672398	113.576500
t16	1398849.87	373086.91	5218.94	37.672989	113.577171
t17	1398657.37	373301.97	5218.65	37.673565	113.577850
t18	1398453.12	373523.56	5217.75	37.674164	113.578575
t19	1398241.50	373733.28	5216.89	37.674725	113.579323
t20	1398031.12	373948.84	5217.21	37.675304	113.580063
t21	1397774.00	374117.94	5217.89	37.675755	113.580963
t22	1397496.50	374299.25	5215.71	37.676235	113.581940
i1	1402882.87	366421.62	5278.50	37.654930	113.562729
i2	1402735.37	366244.44	5280.95	37.654438	113.563225
i3	1402524.87	366075.31	5280.58	37.653957	113.563942
i4	1402355.37	365886.97	5284.17	37.653431	113.564514
i5	1402234.75	365697.97	5288.14	37.652905	113.564911
i6	1402063.62	365461.31	5293.62	37.652245	113.565483
i7	1401925.50	365280.16	5296.93	37.651741	113.565948
i8	1401782.75	365092.97	5300.99	37.651218	113.566429
i9	1401633.00	364901.78	5306.41	37.650684	113.566933
s1	1392214.12	378891.56	5191.83	37.688519	113.600540
s2	1392517.25	378612.91	5190.64	37.687771	113.599472
s3	1392817.62	378382.16	5192.90	37.687157	113.598419
s4	1393077.87	378103.09	5196.78	37.686405	113.597496
s5	1393353.12	377839.31	5191.68	37.685699	113.596527
s6	1393642.62	377583.06	5192.26	37.685013	113.595505
s7	1393934.87	377316.84	5193.02	37.684299	113.594475
s8	1394240.87	377020.03	5194.23	37.683506	113.593391
s9	1394556.00	376733.66	5194.14	37.682739	113.592285
s10	1394864.12	376466.53	5194.36	37.682022	113.591194
s11	1395153.00	376172.72	5198.29	37.681232	113.590179
s12	1395480.50	375896.31	5197.51	37.680496	113.589027
s13	1395787.37	375650.03	5202.34	37.679836	113.587944
s14	1396142.00	375397.94	5204.94	37.679169	113.586700
s15	1396487.75	375128.34	5211.40	37.678448	113.585487
s16	1396782.00	374899.91	5210.60	37.677837	113.584450
s17	1397121.00	374635.22	5211.98	37.677132	113.583260

<u>Station</u>	<u>Easting</u>	<u>Northing</u>	<u>Elev.</u>	<u>Latitude</u>	<u>Longitude</u>
a1	1407478.87	367696.00	5428.11	37.658707	113.546951
a2	1407324.00	367527.75	5445.92	37.658237	113.547478
a3	1407178.50	367364.97	5456.84	37.657780	113.547966
a4	1407061.62	367186.09	5462.32	37.657284	113.548355
a5	1406983.50	366993.50	5474.15	37.656750	113.548615
a6	1406855.25	366788.22	5478.92	37.656178	113.549042
b1	1406653.75	368494.12	5307.52	37.660851	113.549866
b10	1404989.12	367027.37	5321.36	37.656723	113.555504
b11	1404801.62	366860.22	5315.12	37.656254	113.556137
b12	1404614.00	366713.91	5312.19	37.655838	113.556778
b13	1404436.75	366552.75	5306.39	37.655384	113.557373
b14	1404287.75	366402.91	5306.28	37.654964	113.557877
b15	1404097.25	366228.72	5310.00	37.654476	113.558525
b16	1403932.12	366066.47	5315.37	37.654018	113.559082
b17	1403738.37	365908.25	5318.54	37.653576	113.559738
b18	1403577.62	365735.62	5323.45	37.653091	113.560280
b19	1403394.25	365572.44	5325.65	37.652630	113.560898
b2	1406455.12	368339.66	5302.25	37.660416	113.550537
b20	1403220.62	365408.22	5331.66	37.652168	113.561485
b21	1403044.25	365278.84	5331.77	37.651802	113.562088
b22	1402891.75	365124.81	5331.65	37.651371	113.562599
b23	1402728.37	364992.09	5331.20	37.650997	113.563156
b24	1402528.50	364830.87	5330.45	37.650543	113.563835
b25	1402352.37	364690.34	5332.67	37.650146	113.564430
b26	1402205.75	364551.59	5335.95	37.649757	113.564926
b27	1402056.37	364399.37	5340.78	37.649330	113.565430
b28	1401889.87	364231.44	5341.62	37.648857	113.565994
b3	1406289.62	368159.78	5305.53	37.659908	113.551094
b4	1406112.37	367993.53	5312.67	37.659443	113.551697
b5	1405921.37	367822.06	5318.38	37.658962	113.552345
b6	1405723.87	367677.84	5321.82	37.658554	113.553017
b7	1405544.62	367522.25	5321.15	37.658115	113.553619
b8	1405363.37	367365.37	5323.73	37.657673	113.554237
b9	1405176.25	367197.03	5323.45	37.657200	113.554871
c1	1406429.00	366291.53	5513.26	37.654789	113.550476
c10	1405257.75	364830.75	5536.52	37.650707	113.554405
c11	1405238.87	364644.00	5561.44	37.650192	113.554459
c12	1405178.62	364505.62	5579.33	37.649811	113.554657
c13	1405040.87	364315.41	5571.18	37.649281	113.555115
c14	1404873.62	363995.44	5588.43	37.648392	113.555672
c15	1404713.62	363813.78	5600.52	37.647881	113.556213
c16	1404589.37	363700.94	5560.55	37.647564	113.556633
c17	1404468.25	363568.12	5535.01	37.647194	113.557037
c18	1404314.50	363395.50	5546.66	37.646709	113.557556
c19	1404158.12	363226.06	5555.10	37.646236	113.558083
c2	1406283.62	366136.00	5515.00	37.654354	113.550964
c20	1404025.75	363042.06	5564.42	37.645721	113.558525
c21	1403862.75	362793.78	5562.41	37.645031	113.559067
c22	1403603.37	362505.59	5544.50	37.644222	113.559944
c23	1403463.00	362263.00	5550.51	37.643547	113.560410
c24	1403244.50	361947.87	5551.44	37.642670	113.561142
c3	1406156.00	365937.69	5514.87	37.653801	113.551392

<u>Station</u>	<u>Easting</u>	<u>Northing</u>	<u>Elev.</u>	<u>Latitude</u>	<u>Longitude</u>
c4	1406019.12	365868.28	5496.59	37.653603	113.551857
c5	1405954.62	365676.19	5498.09	37.653072	113.552063
c6	1405867.62	365519.19	5518.96	37.652634	113.552353
c7	1405745.75	365310.19	5519.33	37.652054	113.552757
c8	1405596.75	365163.69	5533.20	37.651642	113.553261
c9	1405434.37	364988.03	5528.86	37.651150	113.553810
d1	1402933.75	362743.09	5474.20	37.644836	113.562271
d10	1404478.25	364374.09	5467.77	37.649406	113.557068
d11	1404646.62	364542.12	5470.44	37.649879	113.556496
d12	1404846.37	364723.25	5455.43	37.650387	113.555824
d13	1404953.12	364929.03	5468.49	37.650959	113.555466
d14	1405127.75	365122.94	5471.52	37.651501	113.554878
d15	1405377.37	365386.09	5465.87	37.652241	113.554039
d16	1405604.12	365630.06	5468.28	37.652924	113.553276
d17	1405773.75	365799.94	5469.00	37.653400	113.552696
d18	1405989.37	365988.72	5488.91	37.653931	113.551971
d19	1406162.00	366171.16	5499.09	37.654442	113.551384
d2	1403163.62	362998.16	5483.89	37.645550	113.561501
d20	1406366.12	366328.69	5502.01	37.654888	113.550697
d21	1406595.62	366509.41	5491.09	37.655396	113.549919
d22	1406831.87	366621.66	5480.01	37.655720	113.549110
d23	1407089.00	366820.66	5525.17	37.656281	113.548233
d24	1407259.00	366951.72	5542.85	37.656651	113.547661
d25	1407436.50	367055.19	5561.08	37.656944	113.547050
d26	1407544.25	367201.03	5620.44	37.657352	113.546692
d3	1403313.00	363184.84	5489.92	37.646072	113.560997
d4	1403502.87	363369.72	5498.64	37.646591	113.560356
d5	1403683.12	363528.06	5496.56	37.647034	113.559746
d6	1403855.75	363716.56	5491.98	37.647564	113.559166
d7	1404003.00	363897.28	5485.36	37.648067	113.558670
d8	1404154.00	364037.75	5480.80	37.648464	113.558159
d9	1404314.00	364233.91	5462.03	37.649010	113.557625
e1	1402610.00	363573.87	5407.44	37.647095	113.563454
e10	1404024.50	364951.69	5395.94	37.650967	113.558678
e11	1404183.12	365070.87	5388.94	37.651302	113.558136
e12	1404332.87	365200.59	5380.39	37.651669	113.557632
e13	1404495.37	365340.47	5379.48	37.652061	113.557083
e14	1404635.25	365476.81	5378.93	37.652443	113.556610
e15	1404793.25	365611.53	5383.85	37.652824	113.556068
e16	1404967.25	365756.81	5385.50	37.653233	113.555481
e17	1405111.87	365882.84	5392.56	37.653587	113.554993
e18	1405302.87	365986.75	5362.50	37.653885	113.554337
e19	1405468.62	366067.62	5417.36	37.654118	113.553772
e2	1402767.25	363723.62	5409.62	37.647518	113.562927
e20	1405632.87	366164.31	5432.15	37.654392	113.553215
e21	1405801.00	366324.81	5448.80	37.654842	113.552643
e22	1405964.75	366461.44	5455.31	37.655228	113.552094
e23	1406123.50	366576.91	5453.60	37.655556	113.551552
e24	1406288.37	366713.31	5447.46	37.655937	113.550995
e25	1406418.50	366842.56	5439.71	37.656300	113.550552
e26	1406568.62	366942.47	5426.00	37.656586	113.550041
e27	1406697.62	367131.84	5422.55	37.657112	113.549614

<u>Station</u>	<u>Easting</u>	<u>Northing</u>	<u>Elev.</u>	<u>Latitude</u>	<u>Longitude</u>
e28	1406837.75	367280.19	5411.09	37.657528	113.549141
e3	1402966.75	363909.56	5409.49	37.648041	113.562248
e4	1403116.00	364075.56	5407.27	37.648506	113.561745
e5	1403271.75	364214.28	5412.76	37.648895	113.561218
e6	1403421.75	364374.06	5412.55	37.649342	113.560715
e7	1403612.12	364532.00	5410.48	37.649788	113.560066
e8	1403742.50	364680.31	5406.23	37.650204	113.559631
e9	1403889.62	364811.91	5399.82	37.650574	113.559128
f1	1401928.00	368643.53	5247.88	37.660976	113.566200
f10	1403129.00	366812.94	5272.11	37.656021	113.561913
f11	1403310.25	366654.16	5278.13	37.655598	113.561272
f12	1403489.37	366494.69	5286.43	37.655170	113.560638
f13	1403670.87	366351.53	5295.20	37.654785	113.560005
f14	1403859.25	366204.28	5306.23	37.654392	113.559341
f15	1404017.12	366035.31	5319.73	37.653938	113.558784
f16	1404166.62	365869.41	5332.93	37.653492	113.558258
f17	1404305.25	365689.47	5345.63	37.653008	113.557762
f18	1404420.25	365504.94	5362.29	37.652508	113.557350
f19	1404529.50	365327.03	5383.69	37.652027	113.556961
f2	1402010.87	368409.06	5252.23	37.660336	113.565895
f20	1404653.75	365170.66	5404.21	37.651604	113.556519
f21	1404735.62	364997.84	5428.76	37.651134	113.556221
f22	1404804.87	364831.94	5446.67	37.650684	113.555969
f3	1402135.50	368186.97	5255.58	37.659733	113.565445
f4	1402269.87	367993.72	5257.59	37.659210	113.564964
f5	1402403.50	367791.69	5262.62	37.658665	113.564491
f6	1402401.62	367540.16	5261.53	37.657974	113.564476
f7	1402583.87	367306.87	5260.86	37.657345	113.563828
f8	1402754.75	367154.34	5262.51	37.656937	113.563232
f9	1402944.50	366980.69	5266.17	37.656471	113.562561
g1	1401453.75	368372.62	5247.12	37.660202	113.567818
g10	1402353.37	366960.50	5259.06	37.656380	113.564598
g11	1402447.37	366808.56	5262.00	37.655968	113.564262
g12	1402553.87	366683.41	5264.41	37.655632	113.563889
g13	1402649.62	366536.19	5268.12	37.655231	113.563545
g14	1402738.75	366392.62	5273.43	37.654842	113.563225
g15	1402839.50	366239.03	5280.63	37.654427	113.562866
g16	1402931.75	366080.97	5288.18	37.653999	113.562538
g17	1403031.25	365917.47	5296.56	37.653557	113.562180
g18	1403136.12	365765.41	5306.77	37.653145	113.561806
g19	1403231.37	365613.28	5318.09	37.652733	113.561462
g2	1401567.87	368207.53	5250.01	37.659756	113.567406
g20	1403340.62	365465.97	5332.25	37.652336	113.561073
g21	1403429.75	365321.66	5342.83	37.651943	113.560760
g22	1403545.25	365169.19	5359.19	37.651531	113.560349
g23	1403648.25	365014.34	5374.63	37.651115	113.559982
g24	1403763.25	364863.09	5392.38	37.650707	113.559570
g25	1403872.12	364731.62	5407.56	37.650352	113.559189
g26	1403968.00	364596.94	5422.48	37.649986	113.558846
g27	1404065.87	364454.22	5439.50	37.649601	113.558495
g28	1404190.12	364296.25	5454.88	37.649174	113.558052
g29	1404285.75	364143.25	5468.78	37.648762	113.557709

<u>Station</u>	<u>Easting</u>	<u>Northing</u>	<u>Elev.</u>	<u>Latitude</u>	<u>Longitude</u>
g3	1401667.75	368041.69	5251.39	37.659306	113.567047
g30	1404411.37	364012.87	5485.34	37.648411	113.557266
g31	1404506.12	363906.19	5501.61	37.648125	113.556931
g4	1401761.75	367875.75	5252.57	37.658855	113.566711
g5	1401855.87	367715.47	5254.56	37.658421	113.566376
g6	1401963.87	367540.34	5256.87	37.657948	113.565987
g7	1402057.87	367398.28	5258.85	37.657562	113.565651
g8	1402155.62	367247.31	5256.65	37.657154	113.565308
g9	1402246.00	367096.00	5257.12	37.656746	113.564980
h1	1401680.50	364093.75	5339.80	37.648468	113.566704
h10	1402812.37	365561.75	5305.25	37.652565	113.562912
h11	1402940.50	365719.25	5303.32	37.653008	113.562477
h12	1403052.00	365867.91	5299.94	37.653423	113.562103
h13	1403163.50	366031.12	5296.05	37.653877	113.561729
h14	1403295.87	366188.28	5293.53	37.654316	113.561287
h15	1403419.50	366327.59	5290.49	37.654705	113.560867
h16	1403546.75	366441.41	5289.61	37.655025	113.560440
h17	1403703.12	366582.37	5287.37	37.655422	113.559914
h18	1403874.00	366719.03	5287.12	37.655807	113.559334
h19	1404010.00	366840.41	5284.58	37.656151	113.558868
h2	1401794.75	364248.25	5315.78	37.648899	113.566322
h3	1401885.75	364391.56	5333.37	37.649296	113.566017
h4	1401996.37	364540.75	5329.96	37.649715	113.565651
h5	1402106.25	364711.03	5322.80	37.650188	113.565285
h6	1402268.62	364945.41	5317.02	37.650841	113.564735
h7	1402381.87	365086.22	5314.16	37.651234	113.564362
h8	1402562.12	365301.47	5307.42	37.651836	113.563751
h9	1402702.25	365441.69	5306.07	37.652229	113.563278
i1	1405889.37	368738.59	5295.38	37.661476	113.552521
i10	1404408.50	367205.25	5282.73	37.657177	113.557518
i11	1404248.25	367058.87	5282.85	37.656765	113.558067
i12	1404100.62	366916.50	5284.20	37.656364	113.558563
i2	1405706.62	368577.50	5290.98	37.661022	113.553139
i3	1405525.25	368377.56	5291.04	37.660461	113.553757
i4	1405364.37	368305.19	5281.03	37.660255	113.554306
i5	1405194.87	368079.47	5284.85	37.659622	113.554871
i6	1405044.37	367887.66	5283.24	37.659088	113.555374
i7	1404927.50	367685.25	5284.71	37.658524	113.555763
i8	1404796.37	367502.91	5285.52	37.658016	113.556206
i9	1404594.37	367369.66	5282.43	37.657639	113.556892
j1	1406046.00	369068.41	5298.35	37.662392	113.552010
j10	1407555.12	370965.31	5308.47	37.667690	113.546936
j11	1407716.12	371145.94	5307.48	37.668194	113.546394
j12	1407906.12	371406.56	5306.88	37.668922	113.545761
j13	1407981.37	371652.09	5304.71	37.669601	113.545517
j14	1408119.00	371807.09	5303.94	37.670036	113.545052
j15	1408264.62	371969.03	5302.56	37.670486	113.544563
j16	1408511.12	372173.37	5304.58	37.671062	113.543732
j17	1408561.87	372640.34	5301.43	37.672348	113.543587
j2	1406195.75	369247.31	5299.85	37.662891	113.551506
j3	1406324.00	369390.47	5301.18	37.663292	113.551071
j4	1406468.25	369552.28	5302.51	37.663746	113.550583

<u>Station</u>	<u>Easting</u>	<u>Northing</u>	<u>Elev.</u>	<u>Latitude</u>	<u>Longitude</u>
j5	1406571.12	369691.00	5303.70	37.664131	113.550240
j6	1406754.87	370148.09	5304.34	37.665398	113.549644
j7	1407013.62	370195.09	5307.86	37.665543	113.548752
j8	1407128.62	370422.37	5309.54	37.666172	113.548370
j9	1407340.00	370776.81	5310.35	37.667160	113.547668
k1	1404426.75	368902.75	5278.50	37.661839	113.557587
k10	1406163.25	370086.47	5297.90	37.665192	113.551682
k11	1406391.12	370168.50	5299.95	37.665432	113.550896
k12	1406634.25	370280.19	5301.06	37.665752	113.550064
k13	1406708.37	370507.03	5300.75	37.666382	113.549828
k14	1406760.62	370773.50	5299.46	37.667114	113.549667
k15	1406877.00	371053.41	5299.22	37.667892	113.549286
k16	1406882.25	371331.75	5297.10	37.668655	113.549294
k17	1407145.12	371838.97	5292.49	37.670063	113.548424
k18	1407308.00	372045.69	5293.37	37.670643	113.547874
k19	1407445.62	372283.81	5290.93	37.671303	113.547417
k2	1404597.37	369050.56	5278.37	37.662254	113.557007
k20	1407385.87	372451.72	5290.70	37.671761	113.547638
k21	1407424.37	372637.50	5291.00	37.672272	113.547516
k22	1407500.25	372848.66	5290.19	37.672859	113.547272
k23	1407575.25	373068.16	5291.46	37.673466	113.547028
k3	1404750.25	369286.47	5281.02	37.662910	113.556496
k4	1404971.75	369402.00	5282.77	37.663242	113.555740
k5	1405153.50	369515.03	5285.01	37.663563	113.555122
k6	1405329.62	369629.25	5286.98	37.663887	113.554527
k7	1405504.75	369736.37	5288.97	37.664192	113.553925
k8	1405699.50	369873.06	5290.72	37.664577	113.553268
k9	1405924.87	369968.62	5294.38	37.664856	113.552498
11	1408996.62	371543.78	5313.83	37.669365	113.542000
12	1408922.87	371318.59	5323.05	37.668743	113.542244
13	1408857.37	371018.59	5332.09	37.667915	113.542442
14	1408784.50	370725.81	5327.16	37.667107	113.542671
15	1408720.87	370531.34	5340.22	37.666569	113.542877

APPENDIX D

1987 Gravity Observations

The following data are in a format required by the earth tide correction program con77. The format is as follows: number of stations in group (e.g., 13), date (e.g., Nov. 11 1987: 11/11/87), number 7 (used to get the time into Greenwich time; this will be 8 for data acquired when daylight savings time is in effect), meter number (e.g., g461), 1.000000, comments (e.g., ent(erprise) loop,etc), followed by station name(e.g., gbs), local military standard time (e.g., 10:55), meter reading with decimal place implied between the 4th and 5th number (e.g., 3001.414). The next 23 lines make up an input data file for con77.

13111187 7 g461 1.000000 ent loop,etc

gbs	10553001414
ent	11353002382
gbs	12123001414
ent	13013002391
gbs	13533001440
gbs	15193001442
a	16143002415
b	16203002146
c	16293001907
d	16373001671
gbs	17063001451

08111187 7 g264 1.000000 ent loop,etc

gbs	10443001235
ent	11283002157
gbs	12073001161
gbs	12183001188
gbs	12213001205
ent	13003002147
gbs	13473001192
gbs	15133001211

18111287 7 g264 1.000000 profile B

gbs	08043001155
b1	10042985669
b2	10472986371
b3	11092987051
b4	11322988514
b5	11512988210
gbs	12143001133
b6	13242988529
b7	13372989841
b8	13532989399
b9	14042989889
b10	14172989463
b11	14292988120
b12	14492984872
b13	15032982162

bb1 15572987693
bb2 16062988419
gbs 16393001199
22111287 7 g461 1.000000 profile A

gbs 07543001362
a0 09582946514
a1 10282951227
a2 10422955521
a3 11022959508
a4 11152961942
a5 11462965832
gbs 12223001469
a6 13192967795
a7 13312968298
a8 13412971096
a9 13562973864
a10 14132974220
a11 14302977896
a12 14432977852
a13 14562978852
a14 15112980336
a15 15302982014
a16 15442983516
a17 15572984980
a18 16082986458
gbs 16463001435

31111387 7 g461 1.000000 profile C,E

gbs 08083001382
c1 09092952452
c2 09262957105
c3 09382960595
c4 09552963352
c5 10072967058
c6 10212970185
c7 10362972879
c8 10492973321
c9 11052978458
c10 11192983658
c11 11332986551
c12 11502989085
c13 12062993059
gbs 12333001481
c14 13222994052
c15 13382995905
c16 14012997181
c17 14082998579
c18 14172999445
c19 14263000608
e1 15023002331
e2 15103002618
e3 15213002849
e4 15313003039

e5	15393003229
e6	15473003359
e7	15553003546
e8	16133003748
e9	16233003869
gbs	16353001459

34111387 7 g264 1.000000 profile d,f

gbs	07463001119
d1	09333000089
d2	10123000441
d3	10263000645
d4	10353000796
d5	10483000895
d6	10583000910
d7	11123000917
d8	11213001069
d9	11323001245
d10	11393001449
d11	11483001519
d12	11513002622
d13	12133002622
d14	12233002922
d15	12343003063
d16	12443003171
d17	12563002706
d18	13063001786
gbs	13213001221
d1	14233000501
f1	14473001711
f2	14563001511
f3	15013001414
f4	15103001358
f5	15193001309
f6	15243001206
f7	15333001109
f8	15393001051
f9	15513000879
f10	16023000832
f11	16143000709
f12	16433000386
f13	16583000341
gbs	17133001163

34111587 7 g264 1.000000 profile H

gbs	09063001109
h1	10193000381
h2	10243000425
h3	10353000441
h4	10443000492
h5	10523000516
h6	10583000509
h7	11103000720
h8	11333000489

h9	11333000489
gbs	12023001207
t1	13003000499
t2	13083000348
t3	13173000241
t4	13073000306
t5	13283000197
t6	13322999885
t7	13352999820
t8	13422999761
t9	13492999611
t10	13542999518
t11	14032999410
t12	14102999425
t13	14142999410
t14	14192999361
t15	14512999467
t16	14532999301
t17	15002999222
t18	15492999092
t19	15542999001
t20	15592998302
t21	16062998741
t22	16112998779
gbs	16523001215

11111587 7 g461 1.000000 profile I

gbs	09153001368
i1	10273003910
i2	10403003952
i3	10523004151
i4	11033004188
i5	11143004101
i6	11243004010
i7	11323003983
i8	11443003852
i9	11573003713
gbs	12243001429

20111587 7 g461 1.000000 profile S

gbs	12243001429
s1	13063001438
s2	13203001259
s3	13333000926
s4	13433000639
s5	13553000773
s6	14063000632
s7	14153000481
s8	14323000253
s9	14473000249
s10	14543000061
s11	15062999800
gbs	15293001505
s12	15522999853

s13	16092999453
s14	16112999255
s15	16142999049
s16	16202999213
s17	16242999236
gbs	16553001479

APPENDIX D (CONT.)

1988 Gravity Observations

The following data are in a format required by the earth tide correction program con77. The format is as follows: number of stations in group (e.g., 3), date (e.g., Oct. 14 1988: 10/14/88), number 8 (this will get the time into Greenwich time, use 7 if daylight savings is not in effect), meter number (e.g., g264), 1.000000, comments (e.g., loop [to enterprise]), followed by station name(e.g., gbs), local military time (e.g., 09:12), meter reading with decimal place implied between the 4th and 5th number (e.g., 3002.181). The next 29 lines make up an input data file for con77.

```
03101488 8 g264 1.000000 loop
gbs 09123002181
ent 10013003178
gbs 10493002231
```

```
24101488 8 g264 1.000000 loop
gbs 10493002231
a1 12243001110
a2 12342999924
a3 12412999296
a4 12482999209
a5 12562998523
a6 13052998112
gbs 13473002198
c1 14162995637
c2 14212995472
c3 14262995526
c4 14372996520
c5 14422996639
c6 14512995560
c7 14562995683
c8 15022994950
c9 15092998481
c10 15152995214
c11 15262994162
c12 15472993909
c13 15592994269
c14 16122993518
c15 16232992880
gbs 16533002239
```

```
26101588 8 g264 1.000000 loop
gbs 08473002112
c16 09362995352
c17 09412996599
c18 10012995730
c19 10062994891
c20 10112994332
c21 10172994794
c22 10232996277
```


c23	10292996168
c24	10352996040
d1	10522998905
d2	10572998301
d3	11002997713
d4	11042997139
d5	11072997098
d6	11132997218
d7	11222997218
d8	11292998028
gbs	11593002240
d9	13152999089
d10	13222998706
d11	13392998506
d12	13442999460
d13	13532998371
d14	13582998057
gbs	14563002271

14101588 8 g264 1.000000 loop

gbs	14563002271
d15	15542998121
d16	16002998055
d17	16052997809
d18	16162996779
d19	16202996150
d20	16362996112
d21	16442997052
d22	16562998283
d23	17012996042
d24	17052995106
d25	17082994122
d26	17142990110
gbs	17433002250

53101688 8 g264 1.000000 loop

gbs	08523002160
g1	09033002418
g2	09213002374
g3	09263002535
g4	09323002759
g5	09383002889
g6	09433003002
g7	09493003162
g8	10003003572
g9	10073003908
g10	10193004090
g11	10263004360
g12	10303004509
g13	10373004709
g14	10433004790
g15	10473004820
g16	10513004844
g17	10593004760
g18	11083004572

g19	11143004251
g20	11233003739
g21	11293003454
g22	11333002823
g23	11373002288
g24	11433001559
g25	11483000921
g26	11523000304
g27	11582999687
g28	12022999242
g29	12092998922
g30	12152998469
g31	12122997980
gbs	13073002243
h1	14073003630
h2	14133003645
h3	14233003770
h4	14303003861
h5	14363004159
h6	14423004320
h7	14463004460
h8	14513004720
h9	14553004725
h10	15003004690
h11	15063004679
h12	15113004728
h13	15173004769
h14	15223004685
h15	15263004685
h16	15313004635
h17	15373004590
h18	15433004520
h19	15493004587
gbs	16153002264

26101788 8 g264 1.000000 loop

gbs	09183002175
k1	09463002192
k2	10023002273
k3	10143002221
k4	10173002209
k5	10283002131
k6	10373002099
k7	10433002049
k8	10503002081
k9	10593002010
k10	11033001899
k11	11073002125
k12	11173002277
k13	11393002020
k14	11443002009
k15	11543001883
k16	12043001861
gbs	13143002279

k17	13443001950
k18	13503001921
k19	13543001969
k20	13583001823
k21	14023001674
k22	14073001509
k23	14143001369
gbs	15273002310

03101488 8 g461 1.000000 loop

gbs	09023002827
ent	09553003816
gbs	10443002844

22101488 8 g461 1.000000 loop

gbs	13423002792
b1	14243005079
b2	14373005162
b3	14473005059
b4	14583004639
b5	15073004291
b6	15133004091
b7	15243004089
b8	15443003948
b9	15523003971
b10	16083004096
b11	16163004594
b12	16223004797
b13	16413005100
b14	16503005240
b15	17013005062
b16	17093004872
b17	17163004795
b18	17243004652
b19	17303004623
b20	17393004325
gbs	17543002778

28101588 8 g461 1.000000 loop

gbs	08433002739
b21	09513004364
b22	09593004446
b23	10053004483
b24	10143004539
b25	10213004412
b26	10273004223
b27	10343004019
b28	10403004063
e1	10583001945
e2	11053001658
e3	11163001516
e4	11213001213
e5	11273001213
e6	11343001205
gbs	12043002806

e7 13243001318
e8 13343001543
e9 13413001948
e10 13473002105
e11 13533002482
e12 13593002937
e13 14083002903
e14 14153002930
e15 14243002512
e16 14343002480
e17 14403002038
gbs 15043002850
13101588 8 g461 1.000000 loop
gbs 15043002850
e18 16003001345
e19 16083000673
e20 16222999959
e21 16312998934
e22 16372998490
e23 16472998735
e24 16532999280
e25 16592999825
e26 17053000761
e27 17113001063
e28 17203001939
gbs 17483002825

37101688 8 g461 1.000000 loop
gbs 08463002798
f1 09063002886
f2 09223002980
f3 09343003005
f4 09493003238
f5 10003003308
f6 10103003591
f7 10273004189
f8 10353004458
f9 10403004819
f10 10453004992
f11 10553005179
f12 11053005152
f13 11163005104
f14 11263005037
f15 11323004802
f16 11433004428
f17 11483004142
f18 11563003678
f19 12093002770
f20 12163002016
f21 12223000945
f22 12303000326
gbs 13033002890
i1 14363003975
i2 14493004159

i3	14573004191
i4	15053004331
i5	15093004501
i6	15183004601
i7	15263004814
i8	15323004879
i9	15383005125
i10	15473005110
i11	15533005079
i12	15593005141
gbs	16173002910

25101788 8 g461 1.000000 loop

gbs	09233002734
j1	09573003602
j2	10073003514
j3	10133003424
j4	10213003411
j5	10283003395
j6	10453003072
j7	10533003227
j8	11023002930
j9	11113002787
j10	11193002978
j11	11283003077
j12	11443003052
j13	11513003000
j14	11563003034
j15	12053003010
gbs	13113002896
j16	13413002981
j17	13523002865
l1	14073003855
l2	14163003512
l3	14233003389
l4	14303004125
l5	14403003525
gbs	15253002925

Figure D1. Station locations (a), Bouguer gravity (b), and station elevations (c) along line B of the detailed gravity survey at Newcastle. The horizontal scale is easting, Utah state-plane coordinates.

Figure D2. Station locations (a), Bouguer gravity (b), and station elevations (c) along line C of the detailed gravity survey at Newcastle. The horizontal scale is easting, Utah state-plane coordinates.

Figure D3. Station locations (a), Bouguer gravity (b), and station elevations (c) along line D of the detailed gravity survey Newcastle. The horizontal scale is easting, Utah state-plane coordinates.

Figure D4. Station locations (a), Bouguer gravity (b), and station elevations (c) along line E of the detailed gravity survey Newcastle. The horizontal scale is easting, Utah state-plane coordinates.

Figure D5. Station locations (a), Bouguer gravity (b), and station elevations (c) along line F of the detailed gravity survey at Newcastle. The horizontal scale is easting, Utah state-plane coordinates.

Figure D6. Station locations (a), Bouguer gravity (b), and station elevations (c) along line G of the detailed gravity survey Newcastle. The horizontal scale is easting, Utah state-plane coordinates.

Figure D7. Station locations (a), Bouguer gravity (b), and station elevations (c) along line HIJ (combination of lines H, I, and J) of the detailed gravity survey at Newcastle Utah. The horizontal scale is easting, Utah state-plane coordinates.

Figure D8. Station locations (a), Bouguer gravity (b), and station elevations (c) along line K of the detailed gravity survey at Newcastle. The horizontal scale is easting, Utah state-plane coordinates.

APPENDIX E

**Tabulated temperature-depth, thermal conductivity, and lithology data
for thermal monitoring wells NC-14 through NC-27**

**Temperature vs. Depth Newcastle Well 14
Seasonal Variation**

September, 1988		November, 1988		March, 1989			
Depth (m)	Temp. (C)	Depth (m)	Temp. (C)	Depth (m)	Temp. (C)	Depth Interval (feet)	Thermal Conductivity (W/m/K)
5	27.73	64.19	96.58			No Data Available!	
10	44.25	63.15	95.85				
15	94.70	62.12	95.57				
20	94.72	61.11	94.94				
25	94.74	60.08	94.63				
30	94.75	59.07	94.59				
35	94.76	58.06	94.52				
40	94.77	57.06	94.52				
45	94.77	56.05	94.44				
50	94.75	55.05	94.40				
55	94.86	54.04	94.40				
60	94.94	49.04	94.29				
65	95.95	44.04	94.29				
		39.04	94.25				
		34.05	94.25				
		29.05	94.29				
		24.05	94.29				
		19.05	94.25				
		14.05	94.25				
		9.05	94.17				
		8.05	82.51				
		7.04	62.95				
		6.04	58.60				
		5.03	59.81				
		4.02	30.54				
		3.02	28.58				
		1.98	25.98				
		0.98	22.27				
Gradient St. Dev. (K/m)						Estimates:	
0.7292285 0.2677538						Mean = 1.76	
Heat Flow St. Error (mW/m/m)						St. Err. = 0.176	
1283.4421 128.34421							

NOTE: the temperature gradient shown was calculated using September, 1988 data between 5 meters and 65 meters.

Temperature vs. Depth Newcastle Well 16
Seasonal Variation

September, 1988 ■

March, 1989 □

September, 1988

November, 1988

March, 1989

Depth (m)	Temp. (C)	Depth (m)	Temp. (C)	Depth (m)	Temp. (C)	Depth Interval (feet)	Assumed 30% Por. Thermal Cond. (W/m/K)	Matrix Cond. (W/m/K)
1	22.17			1	10.42			
3	20.98			2	11.70	0 to 10	1.77	2.81
5	20.49			3	13.28	0 to 10	1.80	2.87
7	21.07			4	15.37	0 to 10	1.78	2.84
9	23.36			5	17.83	10 to 20	1.81	2.90
11	25.43			6	19.77	20 to 30	1.81	2.90
13	26.76			7	21.71	30 to 40	1.85	3.00
15	29.41			8	23.06	40 to 50	1.76	2.79
17	31.12			9	24.31	50 to 60	1.79	2.85
17.74	31.70			10	25.22	# meas. =	6	6
Gradient	St. Dev.			11	26.28	Mean =	1.80	2.88
(K/m)				12	27.12	St. Dev. =	0.03	0.07
0.93	0.04			13	27.97	St. Err. =	0.01	0.03
Heat Flow	St. Error			14	29.21			
(mW/m/m)				15	30.14			
1673.20	167.32			16	30.97			
				17	31.73			
				17.50	32.00			

0 to 10	Fine sand (quartz sand); light brown (source = tuffs?)
10 to 20	" "
20 to 30	Medium quartz sand; light brown
30 to 40	Fine sand & tuff pebbles (w/some calcite cement)
40 to 50	" "
50 to 60	" "

NOTE: the temperature gradient shown was calculated using September, 1988 data between the depths of 5 meters and 18 meters.

Temperature vs. Depth Newcastle Well 17
Seasonal Variation

September, 1988 ■

March, 1989 □

September, 1988		November, 1988		March, 1989				
Depth (m)	Temp. (C)	Depth (m)	Temp. (C)	Depth (m)	Temp. (C)	Depth Interval (feet)	30% Por. Assumed Cond. (W/m/K)	Matrix Cond. (W/m/K)
1	23.53			1	9.77			
3	21.04			2	10.95	0 to 10	1.63	2.51
5	20.13			3	13.26	10 to 20	1.80	2.88
7	20.99			4	15.52	20 to 30	1.84	2.98
9	22.29			5	17.69	30 to 40	1.71	2.68
11	24.57			6	19.61	40 to 50	1.58	2.40
13	26.60			7	21.12	50 to 60	1.51	2.24
15	28.29			8	22.19	#meas.	6	6
17	29.89			9	22.88	Mean =	1.67	2.59
18.17	30.51			10	23.69	St. Dev. =	0.13	0.28
Gradient	St. Dev.			11	24.71	St. Err. =	0.06	0.13
(K/m)				12	25.78			
0.85	0.03			13	26.70			
Heat Flow	St. Error			14	27.52			
(mW/m/m)				15	28.22			
1414.73	141.47			16	28.99			
				17	29.82			
				18	30.38			

0 to 10	Fine to medium sand; light brown (source = tuffs?)
10 to 20	" "
20 to 30	" "
30 to 40	" "
40 to 50	Medium sand w/ gravel (large pebbles of tuff)
50 to 60	" "

NOTE: the temperature gradient shown was calculated using September, 1988 data between the depths of 5 meters and 18 meters.

Temperature vs. Depth Newcastle Well 18
Seasonal Variation

September, 1988 ▲

November, 1988 ◆

March, 1989 ✕

September, 1988

November, 1988

March, 1989

Depth (m)	Temp. (C)	Depth (m)	Temp. (C)	Depth (m)	Temp. (C)			
1	23.53	12.04	82.67	1	19.67	30% por.		
3	21.04	11.03	71.30	2	22.48	Depth	Assumed	Matrix
5	20.13	10.03	67.82	3	26.80	Interval	Cond.	Cond.
7	20.99	9.02	61.97	4	34.43	(feet)	(W/m/K)	(W/m/K)
9	22.29	8.02	56.18	5	41.44	0 to 10	1.73	2.73
11	24.57	7.01	50.95	6	45.91	0 to 10	1.72	2.70
13	26.60	6.00	47.89	7	53.17	0 to 10	1.77	2.81
15	28.29	5.00	44.26	8	59.26	10 to 20	1.81	2.91
17	29.89	3.99	40.27	9	65.99	10 to 20	1.80	2.88
18.17	30.51	2.96	36.06	10	73.87	20 to 30	1.79	2.86
		1.95	28.86	11	76.93	20 to 30	1.72	2.71
		1.01	20.15	12	79.95	0 to 10	1.74	2.74
				13	82.25	10 to 20	1.80	2.89
						20 to 30	1.76	2.78
						30 to 40	1.94	3.21
						40 to 50	1.88	3.06
						50 to 60	1.87	3.05
						60 to 70	1.87	3.04
				Gradient	St. Dev.	# meas. =	7	7
				(K/m)		Mean =	1.83	2.96
				5.74	0.18	St. Dev. =	0.07	0.17
				Heat Flow	St. Error	St. Err. =	0.03	0.07
				(mW/m/m)				
				10517.91	1051.79			

0 to 10	Reddish-brown fine sand w/ gravel (source = tuffs?)
10 to 20	" "
20 to 30	Fine/medium sand; brown color
30 to 40	Large pebbles (1/2") of tuff w/ med. sand
40 to 50	Medium to coarse sand w/ large pebbles (tuff)
50 to 60	Medium to coarse sand
60 to 70	Medium to coarse sand w/ large pebbles (tuff)

NOTE: the temperature gradient shown was calculated using March, 1989 data between the depths of 0 meters and 13 meters.

Temperature vs. Depth Newcastle Well 19
Seasonal Variation

September, 1988 ▲
November, 1988 □
March, 1989 *

September, 1988		November, 1988		March, 1989		30% Por.		Matrix Cond. (W/m/K)
Depth (m)	Temp. (C)	Depth (m)	Temp. (C)	Depth (m)	Temp. (C)	Depth Interval (feet)	Assumed Cond. (W/m/K)	
1	23.53	12.04	82.67	1	20.08			
3	21.04	11.03	71.30	2	27.06	0 to 10	2.49	4.59
5	20.13	10.03	67.82	3	33.01	0 to 10	1.41	2.03
7	20.99	9.02	61.97	4	35.41	0 to 10	1.95	3.31
9	22.29	8.02	56.18	5	42.24	10 to 20	1.85	3.01
11	24.57	7.01	50.95	6	45.19	20 to 30	1.85	2.99
13	26.60	6.00	47.89	7	53.52	30 to 40	1.80	2.87
15	28.29	5.00	44.26	8	57.19	40 to 50	1.78	2.85
17	29.89	3.99	40.27	9	61.00	50 to 60	1.77	2.82
18.17	30.51	2.96	36.06	10	63.47	60 to 70	1.79	2.86
		1.95	28.86	11	67.93	# meas. =	7	7
		1.01	20.15	12	73.64	Mean =	1.83	2.95
				13	77.60	St. Dev. =	0.06	0.17
				13.84	80.31	St. Err. =	0.03	0.07
				Gradient (K/m)	St. Dev.			
				4.61	0.11			
				Heat Flow (mW/m/m)	St. Error			
				8427.11	842.71			

NOTE: the temperature gradient shown was calculated using March, 1989 data between 0 meters and 14 meters.

Temperature vs. Depth Newcastle Well 20
Seasonal Variation

September, 1988 ✕

November, 1988 □

March, 1989 ▲

September, 1988		November, 1988		March, 1989		30% Por.		Matrix Cond. (W/m/K)
Depth (m)	Temp. (C)	Depth (m)	Temp. (C)	Depth (m)	Temp. (C)	Depth Interval (feet)	Assumed Cond. (W/m/K)	
1	25.98	20.76	46.78	1	13.60			
2	26.18	19.78	46.06	2	14.99	0 to 10	1.73	2.72
3	25.68	18.78	44.74	3	17.28	10 to 20	1.73	2.72
4	25.60	17.77	43.29	4	20.05	20 to 30	1.68	2.61
5	25.87	16.76	41.82	5	22.89	30 to 40	1.72	2.71
6	26.42	15.76	40.38	6	25.28	40 to 50	1.71	2.68
7	27.24	14.75	38.94	7	27.39	50 to 60	1.61	2.47
8	28.76	13.75	37.56	8	29.28	60 to 70	1.71	2.69
9	30.18	12.77	36.06	9	30.89	# meas. =	7	7
10	31.65	11.77	34.54	10	32.37	Mean =	1.70	2.65
11	33.17	10.76	33.06	11	33.88	St. Dev. =	0.04	0.09
12	34.80	9.75	31.61	12	35.36	St. Err. =	0.02	0.04
13	36.22	8.75	30.21	13	36.83			
14	37.63	7.77	29.00	14	38.28			
15	39.10	6.77	27.71	15	39.59			
16	40.46	5.76	26.65	16	40.99			
17	41.89	4.75	25.45	17	42.37			
18	43.35	3.75	23.64	18	44.13			
19	44.70	2.77	21.18	19	45.23			
20	45.96	1.77	17.85	20	46.46			
Gradient (K/m)	St. Dev.	1.01	14.87	20.62	47.00			
1.41	0.02	0.76	14.36					
Heat Flow (mW/m/m)	St. Error	0 to 10	Gravel w/ little sand, 1/8"-1/2" dia. (source = tuffs)					
2393.81	239.38	10 to 20	"	"				
		20 to 30	"	"				
		30 to 40	"	"				
		40 to 50	"	"				
		50 to 60	"	"				

NOTE: the temperature gradient shown was calculated using September, 1988 data between 5 meters and 20 meters.

Temperature vs. Depth Newcastle Well 21
Seasonal Variation

September, 1988 $\triangle$

November, 1988 $\square$

March, 1989 $\times$

September, 1988		November, 1988		March, 1989		30% Por.		Matrix Cond. (W/m/K)
Depth (m)	Temp. (C)	Depth (m)	Temp. (C)	Depth (m)	Temp. (C)	Depth Interval (feet)	Assumed Cond. (W/m/K)	
1	25.96	17.01	49.38	1	13.28	0 to 10	1.60	2.43
3	25.44	16.00	48.52	2	14.86	10 to 20	1.67	2.60
5	26.33	15.00	46.98	3	17.32	20 to 30	1.57	2.36
7	28.83	13.99	45.37	4	20.72	30 to 40	1.69	2.62
9	33.02	12.98	43.64	5	24.06	40 to 50	1.66	2.57
11	37.86	11.98	41.87	6	27.21	50 to 60	1.69	2.63
13	42.25	10.97	39.81	7	30.15	# meas. =	6	6
15	45.93	9.97	37.60	8	32.42	Mean =	1.64	2.53
17	48.90	8.96	35.21	9	34.83	St. Dev. =	0.05	0.11
17.5	49.41	7.96	32.92	10	37.49	St. Err. =	0.02	0.05
		6.95	30.93	11	40.01			
		5.94	29.17	12	42.03			
		4.94	27.27	13	43.76			
		3.93	25.24	14	45.18			
		2.93	23.06	15	46.93			
		1.92	18.92	16	48.54			
		1.01	15.45	17	49.55			
				17.07	49.60			
Gradient (K/m)	St. Dev.							
1.94	0.06	0 to 10	Lt. brn. silty to clay-size sand w/ pebbles up to 0.25" d.					
Heat Flow (mW/m/m)	St. Error	10 to 20	Lt. brn. silty to clay-size sand w/ pebbles up to 0.5" d.					
3192.91	319.29	20 to 30	Lt. brn. silty to clay-size sand w/ pebbles up to 0.25" d.					
		30 to 40	Lt. brn. silty to clay-size sand w/ pebbles up to 1/8" d.					
		40 to 50	Lt. brn. silty to clay-size sand w/ pebbles up to 1" d.					
		50 to 60	Lt. brn. silty to clay-size sand w/ pebbles up to 1" d.					

NOTE: the temperature gradient shown was calculated using September, 1988 data between 5 meters and 17.5 meters.

Temperature vs. Depth Newcastle Well 22
Seasonal Variation

September, 1988 ▲

March, 1989 □

September, 1988

November, 1988

March, 1989

Depth (m)	Temp. (C)	Depth (m)	Temp. (C)	Depth (m)	Temp. (C)	Depth Interval (feet)	30% Por. Assumed Cond. (W/m/K)	Matrix Cond. (W/m/K)
1	22.56			1	11.58			
2	22.22			2	13.28	0 to 10	1.63	2.49
3	21.61			3	15.07	10 to 20	1.55	2.32
4	21.62			4	17.62	20 to 30	1.73	2.73
5	21.97			5	20.11	30 to 40	1.29	1.79
6	22.60			6	21.94	40 to 50	1.66	2.57
7	23.79			7	24.12	50 to 60	1.68	2.62
8	25.14			8	25.49	# meas. =	6	6
9	26.52			9	26.99	Mean =	1.57	2.37
10	27.90			10	28.36	St. Dev. =	0.16	0.33
11	29.29			11	29.61	St. Err. =	0.07	0.15
12	30.73			12	30.70			
13	32.03			13	32.19			
14	33.35			14	33.48			
15	34.47			15	34.49			
16	35.57			16	35.53			
17	36.73			17	36.63			
18	37.50			18	37.64			
Gradient (K/m)	St. Dev. 0.02							
Heat Flow (mW/m/m)	St. Error 1995.51	0 to 10	Silty sand, no pebbles; medium brown					
		10 to 20	Very fine silt, no pebbles; medium brown					
		20 to 30	" "					
		30 to 40	" "					
		40 to 50	(Washed) pebbles of tuff 1/8" to 1/2" dia.					
		50 to 60	" "					

NOTE: the temperature gradient shown was calculated using September, 1988 data between 5 meters and 18 meters.

Temperature vs. Depth Newcastle Well 23
Seasonal Variation

September, 1988 ✕

November, 1988 ▲

March, 1989 ◆

September, 1988		November, 1988		March, 1989		30% Por.		
Depth (m)	Temp. (C)	Depth (m)	Temp. (C)	Depth (m)	Temp. (C)	Depth Interval (feet)	Assumed Cond. (W/m/K)	Matrix Cond. (W/m/K)
1	23.92	18.04	25.31	1	11.15	0 to 10	1.80	3.14
2	22.72	17.04	25.06	2	12.18	10 to 20	2.06	3.51
3	20.82	16.00	24.48	3	13.49	20 to 30	1.70	2.65
4	19.34	15.03	23.88	4	15.40	30 to 40	1.83	2.96
5	18.67	14.02	23.27	5	16.77	40 to 50	1.78	2.84
6	18.62	13.01	22.61	6	17.84	50 to 60	1.65	2.56
7	18.91	12.01	21.97	7	18.86	# meas. =	6	6
8	19.38	11.03	21.40	8	19.72	Mean =	1.80	2.91
9	19.89	10.03	20.83	9	20.63	St. Dev. =	0.14	0.35
10	20.61	9.02	20.26	10	21.22	St. Err. =	0.06	0.16
11	21.34	8.02	19.77	11	21.74			
12	22.10	7.01	19.41	12	22.27			
13	22.64	6.04	19.19	13	22.77			
14	23.29	5.03	18.95	14	23.40			
15	23.89	4.02	18.52	15	24.00			
16	24.54	3.02	17.55	16	24.56			
17	25.03	2.01	15.55	17	25.08			
18	25.45	1.01	13.44	18	25.38			
Gradient St. Dev. (K/m)								
0.58 0.02		0 to 10	Fine sand w/ some small pebbles (tuff); reddish color					
Heat Flow St. Error (mW/m/m)		10 to 20	Fine sand w/ many pebbles (tuff) up to 0.5" d.; reddish					
1044.29 104.43		20 to 30	"	"	Sand			
		30 to 40	"	"	Content			
		40 to 50	"	"	Decrease			
		50 to 60	"	"	V			

NOTE: the temperature gradient shown was calculated using September, 1988 data between 5 meters and 18 meters.

Temperature vs. Depth Newcastle Well 24
Seasonal Variation

September, 1988 ✕

November, 1988 □

March, 1989 △

September, 1988		November, 1988		March, 1989		30% Por.		Matrix Cond. (W/m/K)
Depth (m)	Temp. (C)	Depth (m)	Temp. (C)	Depth (m)	Temp. (C)	Depth Interval (feet)	Assumed Cond. (W/m/K)	
1	25.87	17.19	48.91	1	14.92	20 to 30	1.63	2.50
2	26.23	16.18	48.29	2	15.68	20 to 30	1.43	2.07
3	26.32	15.18	46.65	3	18.55	30 to 40	1.59	2.41
4	26.54	14.17	44.77	4	22.13	30 to 40	1.45	2.11
5	27.08	13.17	43.08	5	25.45	50 to 60	1.64	2.52
6	28.31	12.16	41.25	6	28.81	50 to 60	1.63	2.51
7	29.66	11.16	39.40	7	31.30	50 to 60	1.50	2.23
8	31.63	10.15	37.48	8	33.61	0 to 10	1.56	2.34
9	33.69	9.14	35.50	9	35.79	10 to 20	1.57	2.38
10	35.97	8.14	33.40	10	37.77	20 to 30	1.53	2.29
11	38.30	7.13	31.69	11	39.61	30 to 40	1.52	2.26
12	40.36	6.16	30.20	12	41.36	50 to 60	1.59	2.42
13	42.19	5.15	28.26	13	43.08	# meas. =	5	5
14	43.85	4.15	26.34	14	44.88	Mean =	1.55	2.34
15	45.53	3.14	24.01	15	46.60	St. Dev. =	0.03	0.07
16	47.47	2.14	21.08	16	48.38	St. Err. =	0.02	0.03
17	48.92	1.01	17.01	17	49.32			
18	25.45			18	25.38			
Gradient St. Dev. (K/m)								
1.91 0.03								
Heat Flow St. Error (mW/m/m)								
2971.91 297.19								
		Interval (feet)				Lithologic Description		
		0 to 10				Fine/med. lt. brn. sand w/ some pebbles up to 0.5" d.		
		10 to 20				" "		
		20 to 30				" "		
		30 to 40				Fine sand plus pebbles (tuff)		
		40 to 50				No Sample.		
		50 to 60				Fine sand plus pebbles (tuff)		

NOTE: the temperature gradient shown was calculated using September, 1988 data between 5 meters and 17 meters.

Temperature vs. Depth Newcastle Well 25
September, 1988

September, 1988		November, 1988		March, 1989		30% Por.		Matrix Cond. (W/m/K)
Depth (m)	Temp. (C)	Depth (m)	Temp. (C)	Depth (m)	Temp. (C)	Depth Interval (feet)	Assumed Cond. (W/m/K)	
1	23.91					0 to 10	1.90	3.12
3	21.63					10 to 20	1.79	2.87
5	21.69					20 to 30	1.63	2.50
7	22.91					30 to 40	1.66	2.56
9	24.77					40 to 50	1.68	2.61
11	26.86					50 to 60	1.66	2.57
13	29.10					# meas. =	6	6
15	31.35					Mean =	1.72	2.69
17	33.27					St. Dev. =	0.11	0.24
18.29	33.98					St. Err. =	0.05	0.11
Gradient St. Dev. (K/m)								
0.98 0.03		Interval		Lithologic				
Heat Flow St. Error		(feet)		Description				
1683.63 168.36		0 to 10		Very fine, brown silt; no pebbles/gravel				
		10 to 20		" "				
		20 to 30		Gradation to fine sand begins in this interval				
		30 to 40		Fine sand plus pebbles (tuff)				
		40 to 50		Medium/fine sand plus pebbles (tuff)				
		50 to 60		" "				

NOTE: the temperature gradient shown was calculated using September, 1988 data between 5 meters and 18.3 meters.

Temperature vs. Depth Newcastle Well 26
Seasonal Variation

September, 1988 $\triangle$

March, 1989 $\blacksquare$

September, 1988		November, 1988		March, 1989		30% Por.		Matrix Cond. (W/m/K)
Depth (m)	Temp. (C)	Depth (m)	Temp. (C)	Depth (m)	Temp. (C)	Depth Interval (feet)	Assumed Cond. (W/m/K)	
1	23.43			1	9.33			
3	19.96			2	10.15	0 to 10	0.78	0.87
5	17.66			3	12.04	10 to 20	1.62	2.48
7	17.86			4	13.92	30 to 40	1.50	2.22
9	18.97			5	16.71	40 to 50	1.57	2.38
11	20.12			6	17.76	50 to 60	1.49	2.20
13	21.47			7	18.52	# meas. =	4	4
15	22.42			8	19.14	Mean =	1.54	2.31
17	23.20			9	19.77	St. Dev. =	0.06	0.13
18.11	23.48			10	20.66	St. Err. =	0.04	0.08
				11	21.17			
				12	21.62			
				13	22.10			
				14	22.55			
				15	22.99			
				16	23.25			
				16.57	23.41			
Gradient (K/m)	St. Dev.							
0.52	0.02							
Heat Flow (mW/m/m)	St. Error	Interval (feet)	Lithologic Description					
803.70	80.37	0 to 10	Very fine, brown silt; no pebbles/gravel					
		10 to 20	"					
		20 to 30	Gradation to fine sand begins in this interval					
		30 to 40	Fine sand plus pebbles (tuff)					
		40 to 50	Medium/fine sand plus pebbles (tuff)					
		50 to 60	"					

NOTE: the temperature gradient shown was calculated using September, 1988 data between 7 meters and 18.1 meters.

Temperature vs. Depth Newcastle Well 27
Seasonal Variation

September, 1988 ▲

March, 1989 □

September, 1988		November, 1988		March, 1989		30% Por.		Matrix Cond.
Depth (m)	Temp. (C)	Depth (m)	Temp. (C)	Depth (m)	Temp. (C)	Depth Interval (feet)	Assumed Cond. (W/m/K)	
1	19.93			1	6.49			
3	17.97			2	7.74	0 to 10	1.58	2.39
5	15.12			3	9.30	10 to 20	1.74	2.74
7	15.30			4	11.04	20 to 30	1.68	2.62
9	16.10			5	12.66	30 to 40	1.49	2.20
11	16.98			6	14.29	40 to 50	1.61	2.47
13	17.94			7	14.97	50 to 60	1.55	2.33
15	18.74			8	15.87	# meas. =	6	6
17	19.52			9	16.10	Mean =	1.60	2.44
18.23	19.79			10	16.49	St. Dev. =	0.09	0.20
				11	17.02	St. Err. =	0.04	0.09
				12	17.59			
				13	17.87			
				14	18.50			
				15	18.80			
				16	19.18			
				17	19.46			
				17.47	19.58			
Gradient (K/m)	St. Dev. (K/m)							
0.41	0.01							
Heat Flow (mW/m/m)	St. Error	Interval (feet)	Lithologic Description					
661.48	66.15	0 to 10	little sand; mostly gravel/pebbles of tuff					
		10 to 20	"					
		20 to 30	"					
		30 to 40	"					
		40 to 50	"					
		50 to 60	"					

NOTE: the temperature gradient shown was calculated using September, 1988 data between 5 meters and 18.2 meters.

APPENDIX F: Hydrochemistry Sampling Methods and Analytical Data

Common Ion Analyses

Six water samples were collected from wells in and around the project area for elemental analyses. Samples were taken from two geothermal wells, one warm irrigation well, and three cold irrigation wells (Figure 23 in main text). Fluid was gathered from the wells during pumping, either at the wellhead or nearest accessible point. Water was pumped through a 0.45 μ m filter using a peristaltic pump. The filtered sample was then portioned into one 500 ml polyethylene bottle, and into one 30 ml polyethylene bottle acidified with 0.1 M HNO₃ (nitric acid).

Samples were delivered to the University of Utah Research Institute, Earth Science Laboratory for analyses. Analyses consisted of determining concentrations of 36 elements using an inductively coupled plasma spectrometer, and determining concentration of HCO₃⁻, CO₃²⁻, Cl⁻, F⁻, SO₄²⁻, and PO₄³⁻ by other methods. Table A1 presents the results of the water analyses.

Sample No:	WS-1	WS-2	WS-3	WS-4	WS-5	WS-6
Location:	C36/15 18acc	C36/15 17cdd	C36/15 20bbb	C36/15 17bba	C36/15 19bba	C36/15 9cad
Species	[concentration in ppm]					
Na	24.58	273.28	290.24	249.61	63.98	104.80
K	2.76	15.24	16.97	11.74	2.64	5.05
Ca	54.05	64.57	78.68	64.36	63.82	62.34
Mg	25.57	0.75	0.69	5.51	25.59	13.32
Fe	N.D.	N.D.	N.D.	N.D.	N.D.	N.D.
Al	N.D.	N.D.	N.D.	N.D.	N.D.	N.D.
SiO ₂	42.12	79.19	69.37	63.24	31.89	46.80
B	N.D.	0.34	N.D.	0.62	0.15	0.19
Li	0.50	0.52	0.60	0.33	N.D.	0.09
Sr	N.D.	1.3	1.56	0.36	0.55	0.35
Total Alkalinity						
HCO ₃ ⁻	181.00	58.00	44.00	104.00	182.00	228.00
CO ₃ ²⁻	N.D.	N.D.	N.D.	N.D.	N.D.	N.D.
Cl ⁻	48.00	69.00	104.00	76.00	44.00	82.00
F ⁻	0.30	7.30	6.30	4.70	0.48	1.03
SO ₄ ²⁻	82.00	569.00	637.00	478.00	208.00	115.00
Total Dissolved Solids						
meas	364.00	1154.00	1236.00	1016.00	530.00	545.00
calc	368.88	1109.01	1227.03	1005.60	530.39	543.00
100*meas/calc	98.68	104.06	100.73	101.03	99.89	100.35
pH(lab)	7.74	8.03	7.98	7.89	7.48	7.60

Table F1: Summary of water analyses from wells in the Newcastle study area.

Isotopic Analyses

Water samples were collected at four cold springs, one cold well, and two geothermal wells from the region outlying the project area (Figure A1) for determination of oxygen and hydrogen isotopes. Samples were collected in 250 ml, tinted glass bottles with "poly-seal" caps. Dr. John Bowman of the University of Utah, Department of Geology and Geophysics performed the analyses.

The results, shown on Table A2, are expressed in the standard per mil notation relative to the V-SMOW standard for both oxygen and hydrogen. Realistic analytical error is ± 0.15 per mil for oxygen and ± 2 per mil for hydrogen. Figure A2 presents a plot of the analytical data with reference to the global meteoric water line (MWL). All cold water samples, as expected, plot either on or near the MWL. The two samples from geothermal wells show slight enrichment of ^{18}O compared to values for cold water.

Sample	^{18}O	D
-----	-----	-----
Cove Spr.	-13.0	-95
Pinto Spr.	-13.9, -14.0	-104, -103
Hildebrand	-14.2	-107
Hygro Well	-14.0, -14.3	-108
Joel Spr.	-14.0	-103
Esc. Mine	-13.0	-94
Hamblin Spr.	-12.6, -12.4	-91, -92

Table F2: Isotopic analyses from springs and wells around the Newcastle study area.

END
5/31/90