Subject: Calculating sample distances in trenches.

You have collected samples at various intervals along a surface trench and have the x (easting), y (northing), and z (elevation) information digitized for each point. You would like to use *COMPOSIT* on these trenches, but you lack the appropriate survey and distance data. How can TECHBASE be used to determine the trench length and the exact distance between each of your sampling stations?

Procedure:

The solution to this dilemma can be found using *TBCALC* and a few simple equations. These equations can be created in an editor and stored in format files for future use.

Before beginning your calculations, you must decide what type of distance data you need. Do you want just A) true distance from point-to-point, or B) to generate an equivalent of from,to distances within the trench? For both cases, the solution is to use the Pythagorean Theorem to determine the distance between each pair of sampling points. This **Technote** will discuss how to create both types of distance information.

In TECHBASE, use the program *DEFINE* to create a FLAT table named TRENCHES. This table will hold the sample information and should contain the following fields:

```
tr_x, tr_y= map coordinates of each sample
tr_z  = elevation of each sample
tr_id = trench name
tr_Au, etc...= sample values
```

Normally, this type of data also has a “station” (sequence) number within each trench:

tr_station= sample number within trench

A) To determine the distance from the previous sample:

Add a tr_dist field (REAL, ACTUAL) to the TRENCHES table.
In *TBCALC*, run the following (with a *sOrt* on: **tr_id**, **tr_station**):

```
tr_id   tr_id[-1]  $COMPARE  99  SKIP # makes sure that both are in the same trench
tr_x   tr_x[-1]   -   2   ^       # good 'ol Pythagorean Theorem
tr_y   tr_y[-1]   -   2   ^   +
tr_z   tr_z[-1]   -   2   ^   +
sqrt   =   tr_dist            # store distance to previous sample in tr_dist
```

B) To determine an equivalent of from, to, we can use similar logic:

1) Add **tr_from** and **tr_to** fields (REAL,ACTUAL) to the TRENCHES table.

2) In *TBCALC*, run the following (with a *sOrt* on: **tr_id**, **tr_station**):


```
temp01   =   tr_from# default at the start of the trench (temp01
     # starts the count at 0)
tr_id   tr_id[-1]  $COMPARE   29   SKIP# makes sure that both are in the same trench
tr_x   tr_x[-1]   -   2   ^# good 'ol Pythagorean Theorem
tr_y   tr_y[-1]   -   2   ^   +
tr_z   tr_z[-1]   -   2   ^   +
sqrt   =   temp02# distance to previous station stored in temp02
temp01   temp02   2   /   +&   =   tr_from# store FROM (1/2 distance to previous
     # station) in tr_from
temp01   temp02   +&   =   temp01# new temp01 is now the total distance from
     # start

     # skip to here

tr_id   tr_id[+1]  $COMPARE   28   SKIP# makes sure that both are in the same trench
tr_x   tr_x[+1]   -   2   ^# good 'ol Pythagorean Theorem
tr_y   tr_y[+1]   -   2   ^   +
tr_z   tr_z[+1]   -   2   ^   +
sqrt   =   temp02# distance to next station stored in temp02
temp01   temp02   2   /   +&   =   tr_to# store TO (1/2 distance to next station) in
     # tr_to
1   99   skip

     # skip to here (last sample in trench)

temp01   =   tr_to# store TO at end-of-trench
0   =   temp01# start next trench, and reset temp01 to 0
```

For an example of this method, try using the data below to duplicate the results.

					|--A--|	|---------B---------|
tr_id	tr_stationtr_x	tr_y	tr_z	tr_dist	tr_fromtr_to
T1	1 1000	1000	150		0.0010.31
T1	2 1020	1000	155	20.62	10.3126.29
T1	3 1015	1010	157	11.36	26.2942.05
T1	4 1030	1020	148	20.15	42.0552.12
T2	1 1060	1010	160		0.007.23
T2	2 1070	1020	163	14.46	7.2320.58
T2	3 1080	1025	168	12.25	20.5826.70

For more information on *TBCALC* and the proper notation required for equations, refer to **"tbCalc - Database calculator" on page tb-55**

