



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE INGENIERIA

G-603917



APUNTES DE
COMPUTACION APLICADA
A LA INGENIERIA
TOPOGRAFICA
Y GEODESICA

ANTONIO HERNANDEZ NAVARRO
MARIO A. REYES IBARRA

VISION DE INGENIERIA CIVIL, TOPOGRAFICA Y GEODESICA
DEPARTAMENTO DE TOPOGRAFIA

FI/DICTG/85-092

RESUMEN	1
1. INTRODUCCION	2
2. REPASO DE INSTRUCCIONES DEL LENGUAJE BASIC	4
2.1 Proposiciones de entrada	4
2.1.1 INPUT -----	4
2.1.2 READ -----	4
2.1.3 DATA -----	5
2.1.4 RESTORE -----	5
2.2 Proposiciones de salida	6
2.2.1 PRINT -----	6
2.2.2 PRINT USING -----	7
2.2.3 TAB -----	7
2.3 Funciones de librería	7
2.4 Transferencia de control incondicionada	8
2.4.1 GOTO -----	8
2.4.2 ON GOTO -----	8
2.5 Transferencia de control condicionada	9
2.5.1 IF THEN -----	9
2.6 Ciclos	10
2.6.1 FOR NEXT -----	10
2.7 Subrutinas y funciones	11
2.7.1 GOSUB -----	11
2.7.2 RETURN -----	11
2.7.3 DEF FNx -----	12
2.8 Arreglos	12
2.8.1 DIM -----	12
2.8.2 ERASE -----	12
2.9 Postulados de control	13
2.9.1 LET -----	13
2.9.2 REM -----	13
2.9.3 SWAP -----	13

603917

2.10	Fin de programa	13
2.10.1	END, STOP -----	13
2.11	Operaciones con matrices	14
2.11.1	MAT READ -----	14
2.11.2	MAT PRINT -----	14
2.11.3	MAT INV -----	15
2.11.4	MAT CON -----	15
2.11.5	MAT IDN -----	15
2.11.6	MAT ZER -----	15
2.11.7	MAT TRN -----	15
2.11.8	Operaciones aritméticas con matrices	16
3.	REPASO DE INSTRUCCIONES FORTRAN	17
3.1	Proposición de entrada	
3.1.1	READ -----	17
3.2	Proposiciones de salida	17
3.2.1	PRINT -----	17
3.2.2	WRITE -----	18
3.2.3	Proposición FORMAT -----	18
3.3	Funciones de librería	20
3.4	Transferencia de control incondicionada	20
3.4.1	GOTO -----	20
3.4.2	GOTO calculado -----	20
3.5	Transferencia de control condicionada	21
3.5.1	IF aritmético -----	21
3.5.2	IF lógico -----	21
3.6	Ciclos	22
3.6.1	DO CONTINUE -----	22
3.6.2	DO implícito -----	22
3.7	Subrutinas y funciones	23
3.7.1	CALL -----	23
3.7.2	SUBROUTINE -----	23
3.7.3	RETURN -----	23
3.7.4	FUNCTION -----	24
3.8	Arreglos	24
3.8.1	DIMENSION -----	24

3.9	Proposiciones auxiliares	24
3.9.1	C -----	24
3.9.2	DATA -----	25
3.9.3	COMMON -----	25
3.9.4	EQUIVALENCE -----	25
3.10	Fin de programa	26
3.10.1	END, STOP -----	26
4.	PROBLEMAS RESUELTOS	27
4.1	Problemas de Topografía	27
4.1.1	Problema Inverso -----	27
4.1.2	Curva Circular Simple -----	29
4.2	Problemas de Astronomía	34
4.2.1	Orientación Astronómica de una Línea -----	34
4.2.2	Transformación de coordenadas celestes -----	39
4.3	Problemas de Ajustes	41
4.3.1	Media y Desviación Estándar de una Muestra Lineal -----	41
4.3.2	Ajuste de una Red de Nivelación por Mínimos Cuadrados -----	44
4.4	Problemas de Geodesia	46
4.4.1	Reducción de Distancias al Elip- soide -----	46
4.4.2	Longitud de un Arco de Meridiano	49
4.5	Problemas resueltos con calculadoras de bolsillo	53
4.5.1	Cálculo Estadimétrico de Distancias y Desniveles -----	53
4.5.2	Cálculo de Volúmenes por Método de Prismas cuadrangulares -----	55
ANEXO I.	LISTADO DE PROGRAMAS	68
I.1	Programa DISTOP	69
I.2	Programa CHORT	71
I.3	Programa SOL	74

I.4	Programa T COOR	77
I.5	Programa MEDIA	80
I.6	Programa AJUS	82
I.7	Programa RDE	84
I.8	Programa DSM	86
I.9	Programa ESTADIA (TI 58/59)	89
I.10	Programa VOLUMEN (HP 11)	90

BIBLIOGRAFIA

R E S U M E N

En virtud de la rápida evolución que en los últimos años se ha tenido en el campo del procesamiento electrónico de datos y de la necesidad del Ingeniero Topógrafo y Geodesta de atender con prontitud, eficiencia y precisión las demandas de la información que él genera, es necesario preparar al estudiante de esta carrera conforme a las posibilidades actuales de este campo.

Estas son algunas de las más importantes razones por las cuales se considera que, dentro del dinamismo que la informática presenta, se procure mantener actualizado el curso de Computación Aplicada que se imparte en la carrera de Ingeniero Topógrafo y Geodesta de la Facultad de Ingeniería de la UNAM, siendo éste el motivo por el cual se han elaborado estos apuntes.

En ellos se presenta en forma estructurada y lógica la solución de problemas sencillos y típicos de la profesión, deliberadamente simplificados a fin de que éstos puedan ser optimizados o mejorados, para que sirvan como ejemplo de aplicación de diferentes lenguajes de programación, y se da también un breve repaso crítico de instrucciones de dichos lenguajes a nivel de recordatorio, ya que ésta es una asignatura consecuente a la de Computadoras y Programación. Se hace énfasis en que la presentación pretende que exista un alto grado de participación en el curso por parte del estudiante para lograr un máximo de aprovechamiento.

1. INTRODUCCION

Independientemente de cual sea el campo de operación del Ingeniero Topógrafo y Geodesta, siempre estará involucrado con procesos de cálculo numérico que pueden variar desde una simple nivelación diferencial o un promedio angular, hasta el complejo problema de ajustes de redes geodésicas horizontales y verticales.

La Ingeniería Topográfica y Geodésica, siempre se ha caracterizado por ser una disciplina que permanece actualizada en relación con los nuevos recursos de procesamiento de datos, resultantes del desarrollo científico y tecnológico en la materia.

En su época, han sido ampliamente usadas por el calculista herramientas como tablas de logaritmos y calculadoras tanto mecánicas como electromecánicas las que por ser muy conocidas y estar prácticamente en desuso solo se mencionan aquí como referencia histórica.

En la actualidad, debido a la creciente necesidad de aprovechar racionalmente los recursos y a la participación que el Ingeniero de esta especialidad tiene en las diferentes etapas de su localización y explotación, es necesario que se le prepare conforme a un marco adecuado para que pueda proporcionar con oportunidad, eficiencia y precisión la información que de él se requiere, siendo imperante aquí el uso de modernos dispositivos de procesamiento tales como las calculadoras y computadoras electrónicas, que relevan al Ingeniero de tareas repetitivas y por lo tanto rutinarias, dándole así oportunidad de optimizar el uso de sus facultades intelectuales y técnicas, canalizándolas a la solución de problemas de mayor trascendencia.

Son numerosas y diversas las aplicaciones actuales de las computadoras y calculadoras en la Ingeniería Topográfica y Geodésica, por cuya razón es imprescindible que se mantenga actualizado al profesional de esta disciplina en el campo de la computación electrónica para un adecuado aprovechamiento del recurso. Así, encontramos que diferentes dispositivos de este tipo se usan en distintas modalidades como son:

- a) Integradas a equipos receptores de satélites para posicionamiento geodésico.
- b) Integradas a equipos de posicionamiento inercial.
- c) Montadas en teodolitos y distanciómetros electrónicos.
- d) Acopladas a sistemas trazadores de mapas.

e) Usadas por separado como cualquier herramienta de cálculo.

En razón de atender los requerimientos académicos en la materia, la carrera de Ingeniero Topógrafo y Geodesta que se imparte en la Facultad de Ingeniería de la Universidad Nacional Autónoma de México, desde hace tiempo ha incluido dentro de su currícula la asignatura de Computación Aplicada, tomando en cuenta, entre otros, los factores descritos.

Así pues, con las presentes notas se pretende proporcionar a profesores y alumnos de esta asignatura, un elemento didáctico de apoyo en el que se intenta dar un enfoque objetivo y preciso de las posibilidades de esta profesión haciendo uso de las computadoras y calculadoras.

2. REPASO DE INSTRUCCIONES DEL LENGUAJE BASIC

Tomando en consideración que el estudiante de esta asignatura ya conoce las generalidades de los lenguajes de programación de computadoras, en este capítulo y en el siguiente solo se hará un breve repaso de instrucciones en BASIC y FORTRAN, ejemplificando su aplicación y estableciendo una base para su discusión en clase.

La secuencia se da en forma estructurada de acuerdo a la función de cada grupo de instrucciones.

Se hace la aclaración que las instrucciones, en la forma en que se presentan, son aceptadas por la mayoría de los sistemas, aunque pueden existir algunas variantes mínimas.

2.1 Proposiciones de entrada

2.1.1 INPUT

Este postulado se utiliza para introducir datos numéricos o alfanuméricos durante la ejecución del programa, cuando se está trabajando desde una terminal de video. El formato general es:

- INPUT variable,variable,...
- INPUT "mensaje",variable,variable,...

Cuando sea encontrado el postulado INPUT, durante la ejecución del programa, aparecerá un signo de interrogación (?) en la terminal, el cual indica que está en espera de los datos. Ejemplos:

```
10 INPUT A3,J,K
20 INPUT A$
30 INPUT "NOMBRE Y EDAD",N$,E
```

2.1.2 READ

Al igual que la proposición INPUT permite introducir datos al programa, con la diferencia de que éstos están incluidos en el cuerpo del programa. El formato general es:

```
READ variable,variable,...
```

Este postulado está asociado a la proposición DATA.
Ejemplo:

```
10 READ A3,J,K
20 READ A$
30 READ N$,E
```

2.1.3 DATA

El propósito del postulado DATA es asignar los valores constantes a las variables listadas en el postulado READ. El formato general es:

```
DATA constante,constante,...
```

La asignación de valores a las variables se efectúa uno a uno empezando con la primera variable y la primera constante. Ejemplo:

```
900 DATA 3,14,97,100.3
910 DATA MEXICO
920 DATA HERNANDEZ,27
```

2.1.4 RESTORE

Cuando las constantes dadas en una proposición DATA son requeridas nuevamente en el programa, se puede emplear la proposición RESTORE para no repetir la instrucción DATA. El formato general es:

```
RESTORE
```

Ejemplo:

```
10 READ A,B,C
==
115 RESTORE
120 READ N,M,P
```

donde:

```
A=N
B=M
C=P
```

2.2 Proposiciones de salida

2.2.1 PRINT

La proposición PRINT permite extraer a determinada dirección de salida, datos numéricos o alfanuméricos de la computadora. El formato general es:

PRINT

- o PRINT "mensaje"
- o PRINT variable,variable,...
- o PRINT expresión,expresión,...

o cualquier combinación de las anteriores de acuerdo a las siguientes reglas:

- 1) Cada postulado PRINT genera solamente una línea, a menos que la impresión requiera un espacio mayor.
- 2) Un postulado PRINT en blanco saltará un renglón.
- 3) Si un entero a imprimir contiene más de ocho dígitos, se truncará a las seis figuras más significativas y será impreso como un número real con exponente.
- 4) Los mensajes siempre deberán ir encerrados entre comillas.
- 5) Si los datos a imprimir están separados por comas (,) cada línea de salida se dividirá en 5 zonas de igual longitud y un valor se imprimirá en cada zona.
- 6) Si se usa punto y coma (;) en lugar de la coma para separar los datos a imprimir, estos estarán uno a continuación del otro, sin respetar la zonificación.
- 7) Escribir una coma o punto y coma al final de las variables a imprimir, ocasiona que la siguiente impresión ocurra en el mismo renglón que la anterior si el espacio lo permite.

Ejemplos:

250 PRINT

260 PRINT "COMPUTACION APLICADA"

```
270 PRINT J,A$
280 PRINT "J VECES K=";J*K,
290 PRINT "J MENOS K=";J-K
```

2.2.2 PRINT USING

Este postulado permite la impresión de resultados con un formato establecido por medio de signos # encerrados entre comillas. Formato general:

PRINT USING "formato"; lista de variables o expresiones.
Ejemplo:

```
310 PRINT USING "###;#";A;B;
320 PRINT USING "#####";K1
```

2.2.3 TAB

La función TAB asociada al postulado PRINT salta a una posición absoluta del tabulador (n) de la impresora, justificada al margen izquierdo. El formato general es:

PRINT TAB(n);lista de variables o expresiones

Ejemplo:

```
210 PRINT TAB(10);A$;TAB(20);Z*J/2
```

2.3 Funciones de librería

Las funciones de librería, también llamadas funciones elementales o estándar, proporcionan un método sencillo y rápido de evaluar expresiones matemáticas.

Las funciones de librería son funciones preescritas, que están integradas como parte integral del lenguaje BASIC.

FUNCION	EJEMPLO	DESCRIPCION
ABS	Y=ABS (X)	Calcula el valor absoluto de X
ATN	Y=ATN (X)	Calcula el arco tangente de X
COS	Y=COS (X)	Calcula el coseno de X
EXP	Y=EXP (X)	Eleva el número e a la X

INT	Y=INT (X)	Calcula el entero de X
LOG	Y=LOG (X)	Calcula el logaritmo natural de X
SGN	Y=SGN (X)	Determina el signo de X
SIN	Y=SIN (X)	Calcula el seno de X
SQR	Y=SQR (X)	Calcula raíz cuadrada de X
TAN	Y=TAN (X)	Calcula la tangente de X

El argumento de las funciones, X, puede ser una variable o una expresión, y para las funciones trigonométricas estará en radianes.

2.4 Transferencia de control incondicionada

2.4.1 GOTO

Los programas desarrollados en lenguaje BASIC se ejecutan de la línea de menor número a la de mayor, si se requiere un salto en el programa, sin intervenir ninguna lógica en él, se utiliza el postulado GOTO. El formato general es:

GOTO número de línea

Ejemplo:

110 GOTO 235

740 GOTO 10

2.4.2 ON GOTO

En ocasiones se requiere saltar a diferentes partes del programa, dependiendo del valor de una expresión o variable, para ello se integró la proposición ON GOTO, su formato general es:

ON variable o expresión GOTO lista de números de línea

El control se transferirá al primer número de línea, si la variable o expresión es igual 1, al segundo si vale 2 y así para los demás. Si el valor no es entero se truncará la porción decimal.

Ejemplo:

```
95 ON J GOTO 190,230,230
740 ON K*I GOTO 10,130,510,600
```

2.5 Transferencia de control condicionada

2.5.1 IF THEN

Los postulados IF-THEN son la base de la transferencia de control condicionada. El formato general es:

IF expresión lógica THEN número de línea o postulado

La expresión lógica es una relación de comparación entre dos variables o expresiones, de acuerdo con los siguientes operadores:

```
> mayor
>= mayor o igual
= igual
<= menor o igual
< menor
<> diferente
```

AND v

OR o

Si la expresión lógica (condición), se cumple se ejecuta la acción enunciada a continuación del postulado THEN; en caso contrario, el programa continuará la secuencia establecida.

Ejemplo:

```
510 IF A>B THEN 360
900 IF INT (A/2)= A/2 THEN PRINT "A ES PAR"
1015 IF ZI>0 AND T<= C OR T=B THEN STOP
```

2.6 Ciclos

2.6.1 FOR NEXT

Si se conoce cuantas veces se debe repetir un ciclo pre establecido, es posible utilizar los postulados FOR-TO-STEP asociado con NEXT, los cuales controlan el número de veces que se ejecutará el ciclo. Su formato general es:

```
FOR variable=m1 TO m2 STEP m3
NEXT variable
```

donde:

m₁ : valor inicial (variable o constante)
m₂ : valor final (variable o constante)
m₃ : incremento (variable o constante)

Cuando se establece un ciclo se deben respetar las siguientes reglas:

- 1) La variable asociada al control del ciclo puede aparecer dentro de éste, pero no puede ser cambiado o alterado.
- 2) Si los valores inicial y final son iguales y el incremento es diferente de cero, el ciclo se ejecutará una vez.
- 3) El control del ciclo se puede transferir fuera de él pero no hacia dentro.
- 4) El circuito no se ejecutará si
 - a) El valor final es menor que el inicial y el incremento es positivo.
 - b) El valor final es mayor que el inicial y el incremento es negativo.
- 5) Si el incremento es unitario se puede eliminar la parte STEP m₃ quedando

```
FOR variable = m1 TO m2
```

Ejemplo:

```
90 FOR J=1 TO 10
    =
    =
```

```
130 FOR I=N TO M STEP S  
  =  
  =  
150 NEXT I  
  =  
  =  
200 NEXT J
```

2.7 Subrutinas y funciones

Las subrutinas y las funciones son programas dentro de otros, que ejecutan funciones predefinidas. Las subrutinas, al igual que las funciones, pueden ser referenciadas de cualquier parte del programa.

La ventaja de utilizar subprogramas, consiste en la facilidad de escribir y probar pequeñas secciones, así como el ahorro de memoria, especialmente si el subprograma es grande y es llamado varias veces a lo largo del programa.

2.7.1 GOSUB

Se referencia una subrutina por medio del postulado GOSUB seguido por el número de línea del primer postulado del subprograma. El formato general es:

```
GOSUB número de línea
```

Cuando la computadora ejecuta este postulado, el control se transfiere a la línea indicada; pero la computadora "recuerda" en donde fue llamado el subprograma.

2.7.2 RETURN

Para transferir el control de la subrutina al programa principal, al final de ésta se escribe el postulado RETURN. Su formato general es:

```
RETURN
```

Ejemplo:

```
30 GOSUB 80  
  =  
  =  
80 Z = J-K  
  =  
  =  
100 RETURN
```

2.7.3 DEF FNx

A las funciones se les debe asignar un nombre (x) el cual debe ser una de las 26 letras. El formato general es:

```
DEF FNx(Argumento)=expresión
```

El postulado DEF FNx solo define la función, no la evalúa, para ésto el formato que se debe seguir es:

```
Variable=FNx(Argumento)
```

Ejemplo:

```
30 DEF FNA(x)=LOG(X)/X
=
110 M=FNA(J)
```

Mientras que la subrutina puede dar dos o más resultados, sin requerir ningún argumento, las funciones solo dan un valor como resultado y siempre requiere, por lo menos, un argumento.

2.8 Arreglos

2.8.1 DIM

Una tabla o lista de términos se llama arreglo. Por medio del postulado DIM se reserva un número preestablecido de localidades de memoria. El formato general es:

```
DIM lista de arreglos
```

Ejemplo:

```
10 DIM A(N,M),B(3),C(I*J,K,L) *
```

2.8.2 ERASE

Cuando un arreglo va a ser nuevamente utilizado y se requiere borrar los términos almacenados en él, se utiliza el postulado ERASE. Su formato general es:

```
ERASE lista de arreglos
```

Ejemplo:

```
105 ERASE A(10,5)
```

* Algunos sistemas no aceptan variables como dimensión.

2.9 Postulados de control

2.9.1 LET

Este postulado se utiliza para asignar valores constantes a una variable. El formato general es:

LET variable=expresión

En las versiones modernas de BASIC, este postulado es opcional y puede ser eliminado.

Ejemplo:

```
110 LET E=E*P/180
```

o

```
110 E=E*P/180
```

2.9.2 REM

El postulado REM es una proposición no ejecutable, que permite incluir comentarios dentro del cuerpo del programa. Su formato general es:

REM comentario

Ejemplo:

```
10 REM TAREA No. 1
```

2.9.3 SWAP

Con este postulado se intercambian los valores de dos variables, sin tener que hacer las asignaciones. El formato general es:

variable SWAP variable

Ejemplo:

```
35 A SWAP B
```

2.10 Fin de programa

2.10.1 END, STOP

Estos postulados interrumpen la ejecución de los programas; la diferencia entre ellos radica en que END es la última instrucción de un programa y solamente puede ir

al final del mismo, mientras que STOP puede ir en cualquier parte del mismo. Los formatos generales son:

STOP

END

Ejemplos:

100 STOP

=

900 END

2.11 Operaciones con matrices

En BASIC se contemplaron una serie de postulados que permiten la operación de arreglos en forma sencilla.

2.11.1 MAT READ

Esta proposición permite la lectura de un arreglo, su formato general es:

MAT READ arreglo,arreglo,...

Ejemplo:

10 MAT READ K

(donde K es un arreglo previamente dimensionado).

2.11.2 MAT PRINT

Para la impresión de arreglos se emplea este postulado, su formato general es:

MAT PRINT arreglo,arreglo,...

Cuando se pone (,), la línea de impresión se divide en 5 campos de igual longitud; si el renglón de la matriz tiene más de cinco elementos, el sexto será impreso en una nueva línea y así sucesivamente. Con un (;) la impresión de los elementos de la matriz estarán uno a continuación de otro sin respetar la zonificación.

Ejemplo:

2900 PRINT MAT A;

2910 PRINT MAT,C;

2.11.3 MAT INV

Para calcular la inversa de una matriz se emplea el postulado MAT INV, su formato general es:

MAT arreglo=INV arreglo

Ejemplo:

100 MAT J=INV K

2.11.4 MAT CON

Con esta proposición se asigna el valor de uno a cada elemento del arreglo. El formato general es:

MAT arreglo=CON

Ejemplo:

195 MAT M=CON

2.11.5 MAT IDN

Genera la matriz identidad, esto es, cero en todos los elementos de la matriz excepto la diagonal principal que asigna uno. Su formato general es:

MAT arreglo=IDN

Ejemplo:

215 MAT A=IDN

2.11.6 MAT ZER

Este postulado tiene el mismo efecto que la proposición ERASE. El formato general es:

MAT arreglo=ZER

Ejemplo:

3100 MAT F=ZER

2.11.7 MAT TRN

Con esta proposición se ocasiona que se cambien las columnas por los renglones, matriz transpuesta, su formato general es:

MAT arreglo=TRN arreglo

Ejemplo:

95 MAT A=TRN B

2.11.8 Operaciones aritméticas con matrices

Las operaciones aritméticas con las matrices se indican en la misma forma que si fueran escalares, por lo tanto tenemos:

Suma

MAT arreglo=arreglo+arreglo

Resta

MAT arreglo=arreglo-arreglo

Multiplicación

MAT arreglo=arreglo*arreglo

Ejemplo:

90 MAT X=J-K

100 MAT A=B*C

3. REPASO DE INSTRUCCIONES FORTRAN

Esencialmente, es posible dividir los postulados del lenguaje FORTRAN en la misma forma que el lenguaje BASIC, por lo que solo en aquellos casos en que sea conveniente y necesario se dará una explicación adicional. Así pues, todo lo que se ha establecido para el lenguaje BASIC, es válido para FORTRAN.

3.1 Proposición de entrada

3.1.1 READ

Formato general:

READ (i,j,ERR=m₁,END=m₂) variable,variable...

donde:

i: número de unidad de entrada

j: número de formato

m₁: número de etiqueta donde deberá mandar el control si detecta un error

m₂: número de etiqueta donde deberá mandar el control al terminar los datos

Los postulados END y ERR son opcionales, por lo que el formato general puede quedar reducido a:

READ(i,j) variable,variable,...

Ejemplo:

READ(5,100,END=90,ERR=110) A,B,C,

READ(5,1000) J(I),K(L,M)

READ(5,900,END=1000) A,E2

3.2 Proposiciones de salida

3.2.1 PRINT

Formato general:

PRINT J

o

PRINT J,variable,variable,...

donde:

J: número de formato

Ejemplo:

```
PRINT 1100
```

```
PRINT 910,LAT,LON
```

3.2.2 WRITE

Formato general:

```
WRITE(k,j,ERR=m,END=m2) variable,variable,...
```

donde:

k: número de unidad de salida

j: número de formato

m₁: número de etiqueta donde se deberá mandar el control si se detecta un error

m₂: número de etiqueta donde se deberá mandar el control al terminar los datos

Los postulados END y ERR son opcionales quedando reducido el formato general a:

```
WRITE(k,j) variable,variable,...
```

Ejemplo:

```
WRITE(6,120)
```

```
WRITE(6,100) A,B,C
```

```
WRITE(11,900,END=10) J,K,L,A2
```

3.2.3 Proposición FORMAT

La proposición FORMAT se emplea para especificar el número de figuras significativas de las variables tanto de entrada como de salida. El formato general es:

```
J FORMAT (especificaciones)
```

donde:

J: número de formato

y las especificaciones se dan de acuerdo a los siguientes códigos:

- Iw: número de enteros
- Fw.d: números reales
- Ew.d: punto flotante
- Dw.d: doble precisión
- Aw: alfanumérico
- wX: blancos
- WH: caracteres alfanuméricos
- /: salto de renglón

donde:

- w: número de figuras
- d: número de decimales

Cada código debe ir separado uno de otro por una coma (,).

Si dos o más códigos o grupos de códigos se repiten y se quieren agrupar, se puede utilizar paréntesis, antecedido por una constante que indica el número de veces que se va a repetir el código o grupo de códigos.

En el caso del código H, éste puede ser cambiado por apóstrofes al principio y al final de los caracteres alfanuméricos que se van a imprimir, produciendo el mismo efecto. Adicionalmente, este código puede ser antecedido por los siguientes caracteres, produciendo el efecto mencionado:

CARACTER	EFFECTO
blanco	Deja un espacio antes de escribir
0	Deja dos espacios antes de escribir
1	Salta a la siguiente hoja
+	Suprime los espacios

Ejemplo:

```
100 FORMAT (1H0,I5,2(F9.3,5(E12.6,18)))
```

120 FORMAT(11HPOLIGONALES,/,2A4)

3.3 Funciones de librería

FUNCION	EJEMPLO	DESCRIPCION
ABS	Y=ABS(X)	Calcula el valor absoluto de X
IABS	I=IABS(J)	
ATAN	Y=ATAN(X)	Calcula el arco tangente de X
COS	Y=COS(X)	Calcula el coseno de X
EXP	Y=EXP(X)	Eleva el número e a la X
INT	I=INT(X)	Determina el entero de X
ALOG	Y=ALOG(X)	Calcula el logaritmo natural de X
SIN	Y=SIN(X)	Calcula el seno de X
SQRT	Y=SQRT(X)	Calcula la raíz cuadrada de X
TAN	Y=TAN(X)	Calcula la tangente de X

3.4 Transferencias de control incondicionadas

3.4.1 GOTO

Formato general:

GOTO número de etiqueta

ejemplo:

GOTO 10

GOTO 215

3.4.2 GOTO calculado

Es equivalente a ON GOTO de BASIC. El formato general es:

GOTO (lista de etiquetas),variable entera

Ejemplo:

GOTO (4,600,13,9,526),IAC

3.5 Transferencias de control condicionada

En FORTRAN contamos con dos tipos de transferencia condicionada en el postulado IF.

3.5.1 IF aritmético

El formato general es:

IF (expresión aritmética) n_1, n_2, n_3

donde:

n_1, n_2, n_3 : son números de etiqueta y el control se transferirá a n_1, n_2 , ó n_3 dependiendo de si el valor de la expresión aritmética es negativa, cero, o positiva respectivamente.

Ejemplo:

IF (A*B) 10,20,30

IF (Z) 15,50,30

3.5.1 IF lógico

El formato general es:

IF (expresión lógica) S

donde:

S: es cualquier postulado ejecutable excepto un DO u otro IF.

Las expresiones lógicas se forman de acuerdo con los siguientes operadores:

.GT. mayor que

.GE. mayor o igual que

.EQ. igual que

.LE. menor o igual que

.LT. menor que

.NE. diferente que

.AND. Y

.OR. o

Ejemplo:

```
IF (A.GT.B) GOTO 30
IF (A.GE.O.AND.C.LT.B) PRINT 100
```

3.6 Ciclos

3.6.1 DO CONTINUE

Es equivalente a FOR-NEXT en BASIC. El formato general es:

```
DO n variable=m1,m2,m3
```

```
n CONTINUE
```

donde:

n: número de etiqueta

m₁: valor inicial

m₂: valor final

m₃: incremento

Ejemplo:

```
DO 10 I=1,5
```

```
==
```

```
DO 20 H=2,100,2
```

```
==
```

```
20 CONTINUE
```

```
10 CONTINUE
```

3.6.2 DO Implícito

El formato general es:

(Arreglo,variable=m₁,m₂,m₃)

donde m₁,m₂ y m₃ tienen el mismo significado que en DO.

Ejemplo:

(A(I),I=2,10,2)

3.7 Subrutinas y funciones

3.7.1 CALL

Para llamar una subrutina al programa se emplea este postulado. Su formato general es:

- CALL nombre
- o CALL nombre(variable o constante,...)

Ejemplo:

```
CALL G(IG,M,S)
CALL TEM (3.0,X,Y,IT,E)
CALL IMP
```

3.7.2 SUBROUTINE

Postulado empleado para definir una subrutina, esta asociado a CALL y la última tarjeta debe ser RETURN. El formato general es:

SUBROUTINE nombre

o

SUBROUTINE nombre (variable o constante, variable o constante,...)

Ejemplo:

```
SUBROUTINE G (X,Y,Z)
=
RETURN
END

SUBROUTINE IMP
=
RETURN
END
```

3.7.3 RETURN

El formato general es:

RETURN

Ejemplo:

RETURN

3.7.4 FUNCTION

Postulado empleado para definir una función en el programa, a diferencia del BASIC puede tener más de una instrucción y la última al igual que SUBROUTINE debe ser RETURN. El formato general es:

```
t FUNCTION nombre(variable o constante, variable o constante...)
```

donde t es el tipo de función, que puede ser INTEGER, REAL, DOUBLE PRECISION, COMPLEX O LOGICAL.

Ejemplo:

```
FUNCTION UNO(3,A,B)  
=  
RETURN
```

El formato para llamar a la función al programa principal:

variable=nombre de la función (variable o constante, variable o constante,...)

3.8 Arreglos

3.8.1 DIMENSION

Para reservar espacio en la memoria se emplea el postulado DIMENSION. El formato general es:

```
DIMENSION variable(variable o constante entero, ...),...
```

Ejemplo:

```
DIMENSION A(100,100),B(10),C(3,6,9)
```

3.9 Proposiciones auxiliares

3.9.1 C

Para intercalar comentarios en el programa se pone una letra C en la columna 1 y a continuación el texto. El formato general es:

```
C texto
```

Ejemplo:

```
C ESTO ES UN EJEMPLO
```

3.9.2 DATA

Si se desea asignar valores dentro del programa se emplea esta proposición, cuyo formato general es:

```
DATA variable/constante/,...
```

```
DATA variable,.../constante,.../
```

Ejemplo:

```
DATA M1 / 3./,J/5/
```

```
DATA M1,J/3.,5/
```

3.9.3 COMMON

El COMMON permite hacer áreas de memoria comunes a dos o más variables con el consiguiente ahorro de espacio en memoria. El formato general es:

```
COMMON/nombre/variable,variable,...
```

```
COMMON variable,variable,...
```

El primero se conoce como COMMON rotulado.

Ejemplo:

```
COMMON A,B
```

```
COMMON /BLOCK1/FI,AREA,RA
```

3.9.4 EQUIVALENCE

Quando en un programa se quiere hacer que dos variables, con nombre diferente, sean equivalentes se emplea esta proposición. Generalmente esto sucede cuando el programa es la suma de otros programas y se quiere ahorrar el trabajo de cambiar todos los nombres. El formato general es:

```
EQUIVALENCE (variable,variable,...)
```

Ejemplo:

```
EQUIVALENCE (AREA,A)
```

3.10 Fin de programa

3.10.1 END, STOP

El formato general es:

END

STOP

Ejemplo:

IF (A.EQ.B) STOP

==

END



4. PROBLEMAS RESUELTOS

En este capítulo se da solución a 10 problemas típicos de la Ingeniería Topográfica y Geodésica, de los cuales 4 son en Lenguaje BASIC, 4 en FORTRAN y 2 con calculadora de bolsillo.

La organización de la solución se presenta en la siguiente forma estructurada:

1. Objetivo
2. Descripción del problema
3. Algoritmo en diagrama de bloque
4. Datos de entrada
5. Resultados

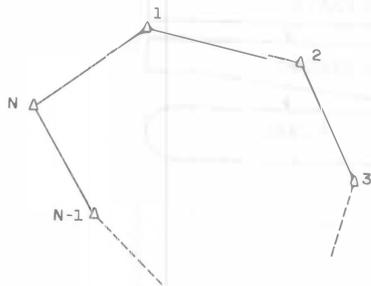
Con el fin de conservar la objetividad en la presentación y desarrollo del problema a resolver, los listados de los programas se presentan en el anexo 1.

4.1 Problemas de Topografía

4.1.1 Problema Inverso

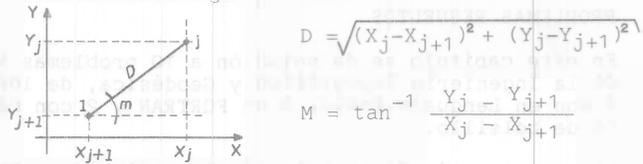
Objetivo: Obtener la distancia y el rumbo de los lados de un polígono a partir de pares de coordenadas conocidas.

Descripción: Este programa está escrito en lenguaje BASIC. Desde el punto de vista de topografía, frecuentemente se requiere dar solución a un problema de este tipo, en cuyo caso se tiene la información siguiente:



VERTICE	X	Y
1	X_1	Y_1
2	X_2	Y_2
3	X_3	Y_3
=	=	=
N-1	X_{N-1}	Y_{N-1}
N	X_N	Y_N

La solución trigonométrica es:



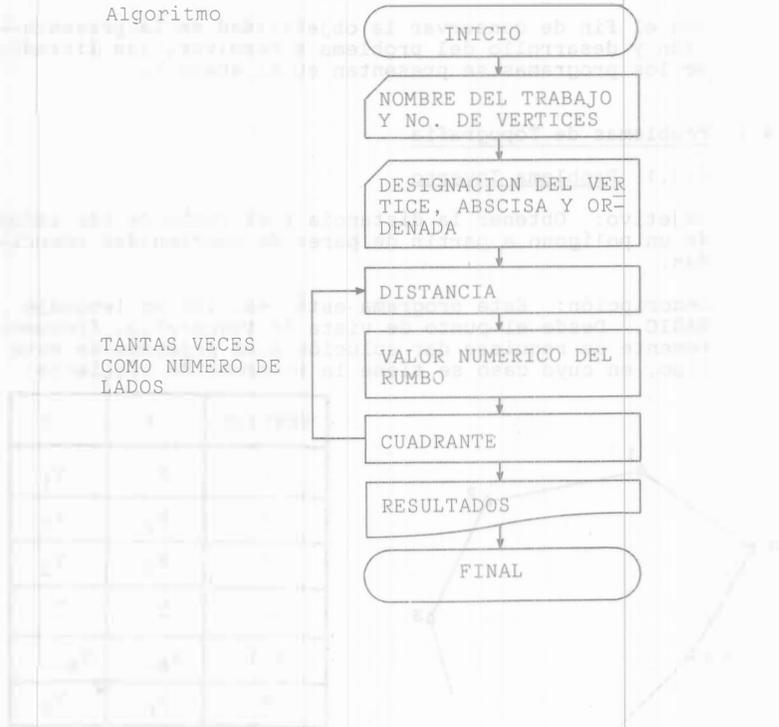
Desde el punto de vista de topografía, dada la definición del rumbo, se tiene:

$$D = \sqrt{(X_j - X_{j+1})^2 + (Y_j - Y_{j+1})^2}$$

$$R = \tan^{-1} \frac{X_j - X_{j+1}}{Y_j - Y_{j+1}}$$

Con la localización del cuadrante respectivo para el rumbo.

Algoritmo



Datos de entrada:

- Tarjeta de comentarios (una por proceso)
- Nombre del trabajo, número de vértices
- Tarjetas de coordenadas (Tantas como vértices de la poligonal), introduciendo nombre del vértice, abscisa y ordenada separados por comas.

DATOS DE ENTRADA



Resultados: Los resultados arrojados por el proceso son:

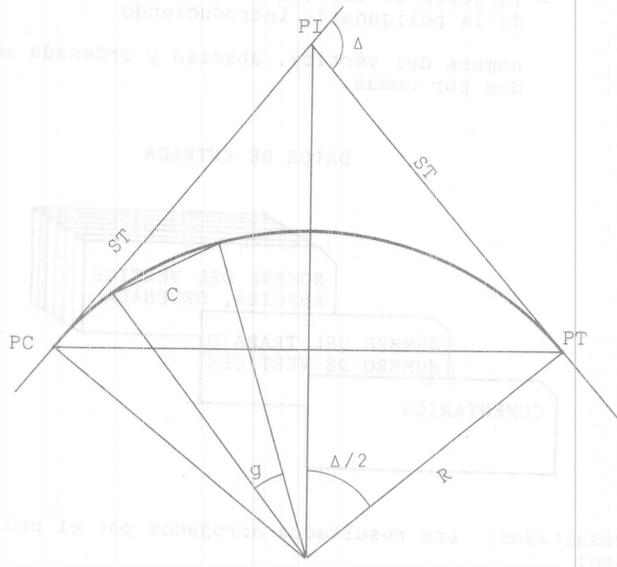
- Denominación del vértice
- Abscisa
- Ordenada
- Par de vértices que forman el lado
- Distancia
- Rumbo

En el listado 4.1.1 se tiene un ejemplo del proceso.

4.1.2 Curva Circular Simple

Objetivo: Calcular los elementos necesarios para el trazo en campo de una curva circular simple.

Descripción: Es un programa escrito en lenguaje FORTRAN. En proyectos de vías de comunicación como son carreteras, calles, vías férreas y canales, es necesario dar gradualmente el cambio de dirección entre dos tangentes, lo cual se logra mediante curvas. Tal es el caso de la curva circular simple.



donde:

- PC: Punto de comienzo
- PI: Punto de intersección
- PT: Punto de término
- LC: Longitud de la curva
- ST: Subtangente
- C: Cuerda
- R: Radio

- Δ : Deflexión total
- g : Grado de curvatura
- δ : Deflexión de trazo

Partiendo de los datos de proyecto, normalmente el cadenamiento del PI, la deflexión total, la longitud de las cuerdas y el grado de curvatura, se determinan los demás elementos mediante las relaciones siguientes:

$$R = \frac{C}{2 \operatorname{sen} g/2}$$

$$ST = R \tan \frac{\Delta}{2}$$

$$PC = PI - ST$$

$$LC = \frac{\Delta}{g} C$$

$$PT = PC + LC$$

Como generalmente las cuerdas de entrada y salida no coinciden con cadenamientos cerrados, es necesario determinar las subcuerdas y sus deflexiones correspondientes, ésto es:

$$CE = \left\lfloor \frac{PC}{C} + 1 \right\rfloor C - PC *$$

$$ge = 2 \operatorname{sen}^{-1} \frac{CE}{2R}$$

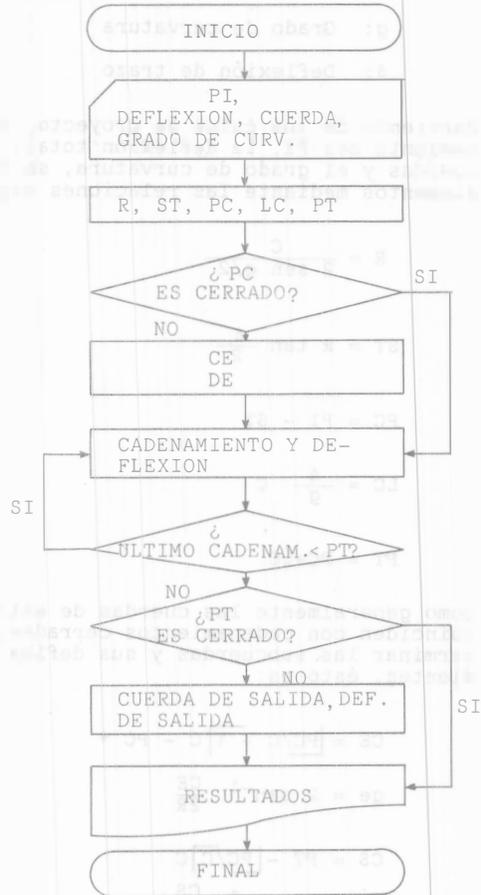
$$CS = PT - \left\lfloor \frac{PC}{C} \right\rfloor C$$

$$gs = 2 \operatorname{sen}^{-1} \frac{CS}{2R}$$

$$\delta = \frac{g}{2}$$

* $\left\lfloor \right\rfloor$: ENTERO

Algoritmo:



Datos de entrada:

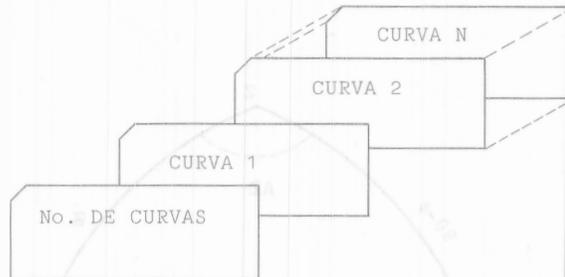
- Número de curvas a procesar

COLUMNAS	FORMATO	DESCRIPCION
1-4	I4	No. de curvas

- Datos de curvas (una tarjeta o renglón por cada curva)

COLUMNAS	FORMATO	DESCRIPCION
1 - 9	F 9.3	PI
10 - 16	F 7.3	Δ
17 - 23	F 7.3	C
24 - 30	F 7.3	g

DATOS DE ENTRADA



Resultados: Los resultados, tal como se ve en el listado 4.1.2 son:

- cadenamiento
- deflexión
- radio
- subtangente
- punto de intersección
- cuerda de entrada
- cuerda de salida
- grado
- deflexión total

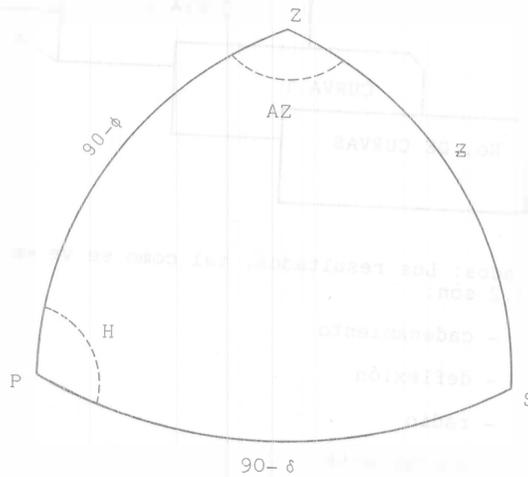
4.2 Problemas de Astronomía

4.2.1 Orientación Astronómica de una Línea

Objetivo: Determinar el azimut de una línea a partir de observaciones a un astro.

Descripción: Este es un programa escrito en lenguaje FORTRAN, capaz de determinar el azimut de una línea a partir de observaciones directas de campo y el auxilio del Anuario que edita el Instituto de Astronomía de la UNAM. El problema se define astronómicamente de la siguiente manera:

DEL TRIANGULO ASTRONOMICO



donde:

- P: Polo celeste
- Z: Zenit
- S: Astro observado
- H: Angulo horario
- Az: Azimut del astro (a determinar)
- ϕ : Latitud del lugar (conocido)
- δ : Declinación del astro en el momento de la observación
- z: Distancia cenital medida

A partir de esta información, el azimut del astro se puede determinar con la siguiente fórmula:

$$\cos Az = \frac{\sin \delta - \sin \phi \cos z}{\cos \phi \sin z}$$

en donde:

- δ : Se obtiene interpolando los datos del anuario según la fecha y la hora de observación.
- ϕ : Se determina por otro medio, por ejemplo de una carta geográfica.
- z: Se mide directamente con el tránsito, tomando además lecturas de presión y temperatura para corregir este elemento por refracción y paralaje con las siguientes fórmulas:

$$R = 60'' \cdot 6 \tan z_0 \quad z = z + RTB - P$$

$$T = \frac{1}{1 + 0.004 t} \quad \text{Donde.}$$

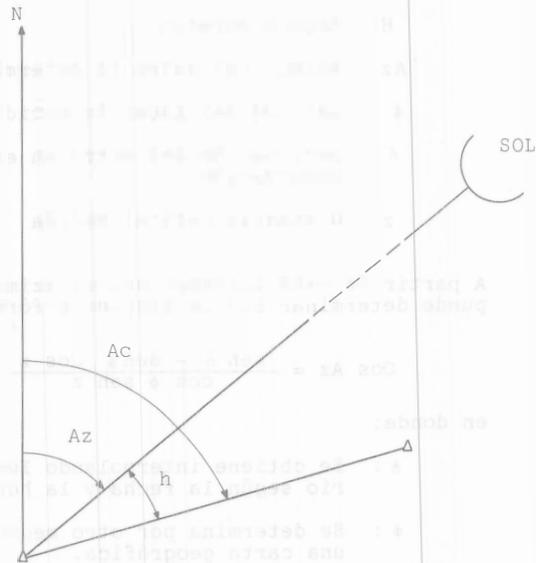
$$B = \frac{Pr}{760} \quad R: \text{corrección por refracción}$$

T: factor termométrico

B: factor barométrico

$$P = 8''8 \sin z_0 \quad P: \text{corrección por paralaje}$$

Una vez que se ha determinado el azimut del astro, el azimut de la línea se obtiene sumando algebraicamente el ángulo promedio señal.- Sol:



donde:

Az: Azimut del sol

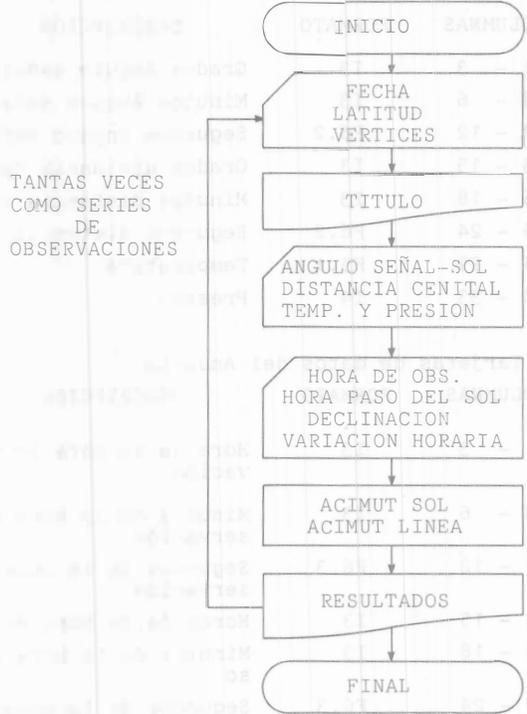
h : Angulo horizontal señal-sol

Ac: Azimut de la línea

$$y \quad Ac = Az + h$$

con lo cual queda resuelto el problema.

Algoritmo



Datos de entrada:

- Tarjeta de fecha, latitud y par de vértices.

COLUMNA	FORMATO	DESCRIPCION
1 - 3	I3	DIA
4 - 6	I3	MES
7 - 11	I5	AÑO
12 - 14	I3	GRADOS DE LATITUD
15 - 17	I3	MINUTOS DE LATITUD
18 - 23	F6.2	SEGUNDOS DE LATITUD
24 - 27	A4	VERTICE ESTION
28 - 31	A4	VERTICE VISADO

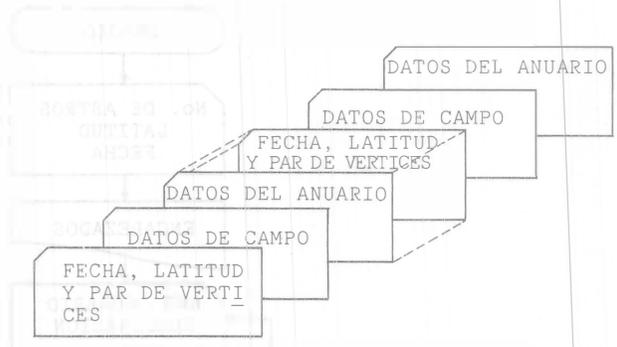
- Tarjetas de datos de campo

COLUMNAS	FORMATO	DESCRIPCION
1 - 3	I3	Grados ángulo señal-sol
4 - 6	I3	Minutos ángulo señal-sol
7 - 12	F6.2	Segundos ángulo señal-sol
13 - 15	I3	Grados distancia cenital
16 - 18	I3	Minutos distancia cenital
19 - 24	F6.2	Segundos distancia cenital
25 - 29	F5.1	Temperatura
30 - 33	I4	Presión

- Tarjetas de datos del Anuario

COLUMNAS	FORMATO	DESCRIPCION
1 - 3	I3	Hora de la hora de observación
4 - 6	I3	Minutos de la hora de observación
7 - 12	F6.3	Segundos de la hora de observación
13 - 15	I3	Horas de la hora del paso
16 - 18	I3	Minutos de la hora del paso
19 - 24	F6.3	Segundos de la hora del paso
25 - 27	I3	Grados de la declinación
28 - 30	I3	Minutos de la declinación
31 - 36	F6.3	Segundos de la declinación
37 - 42	F6.2	Variación horaria en declinación

DATOS DE ENTRADA



Resultados: En el listado 4.2.1, se tiene un ejemplo de resultados, de la siguiente manera:

- Encabezados
- Fecha, latitud, descripción de la línea y su acimut correspondiente.

4.2.2 Transformación de Coordenadas Celestes

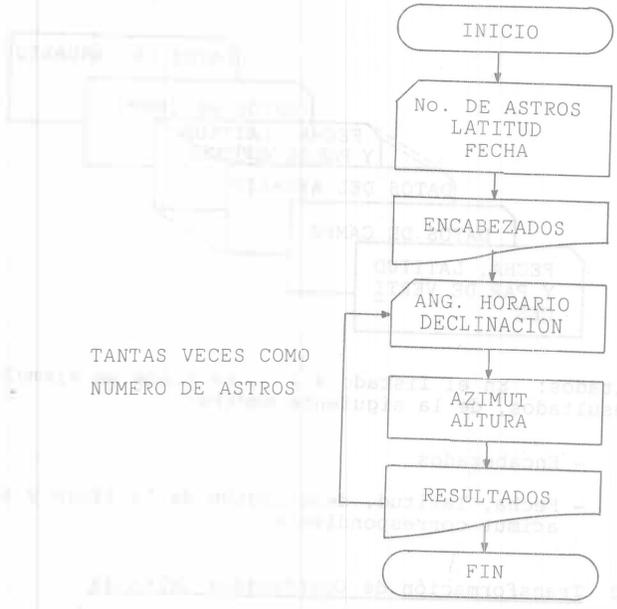
Objetivo: Conociendo el ángulo horario y la declinación de un astro, así como la latitud del punto de observación, determinar el azimut y la altura.

Descripción: Es este un programa escrito en lenguaje BASIC. En observaciones astronómicas hechas para determinar posición geográfica, con frecuencia se requiere efectuar transformaciones entre sistemas de coordenadas para adecuar la información a nivel operacional. A continuación se resuelve el caso de transformar del Sistema Ecuatorial de Angulo Horario al Sistema Horizontal. Esto es, del triángulo astronómico se conocen dos lados (ϕ , δ) y un ángulo (H), con lo cual se puede establecer:

$$\cos z = \sin \phi \sin \delta + \cos \phi \cos \delta \cos H$$
$$\sin Az = - \frac{\cos \delta \sin H}{\sin z}$$

Siendo éstas las fórmulas que permiten efectuar la transformación.

Algoritmo



Datos de entrada:

- Tarjeta de datos generales, donde se debe dar: número de astros, latitud en grados, minutos y segundos, así como la fecha separando cada elemento con una coma.

Tarjeta de coordenadas: En esta se escribe el ángulo horario y la declinación en grados, minutos y segundos, separando igualmente los elementos con comas.

DATOS DE ENTRADA



Resultados: En el listado 4.2.2 se puede ver un ejemplo resuelto, apreciando que los resultados son dados en el siguiente orden:

- Encabezados
- Latitud y fecha
- Angulo horario, declinación, altura y azimut

4.3 Problemas de Ajustes

4.3.1 Media y Desviación Estándar de una Muestra Lineal

Objetivo: Calcular la media y la desviación estándar de los elementos de muestras lineales.

Descripción: Este programa está escrito en lenguaje FORTRAN. Considerando la muestra $E = (E_1, E_2, \dots, E_n)$, el número real M definido como:

$$M = \frac{1}{n} \sum_{i=1}^n E_i \quad (1)$$

se conoce como media o valor más probable de la muestra.

El número real S^2 definido como:

$$S^2 = \frac{1}{n} \sum_{i=1}^n (E_i - M)^2 \quad (2)$$

es llamado varianza (grado de dispersión) de la muestra. La raíz cuadrada de la varianza, S, se conoce como desviación estándar:

$$S = (s^2)^{1/2} \tag{3}$$

La fórmula (2) puede ser modificada para optimizar el funcionamiento del programa.

Desarrollando el binomio de la ecuación (2):

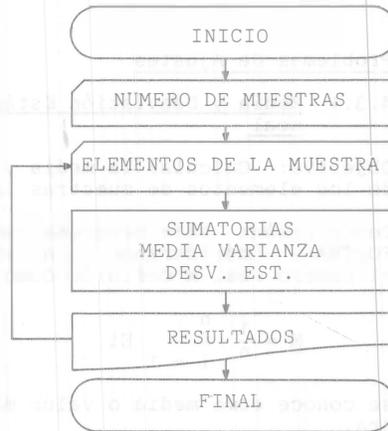
$$\begin{aligned} S^2 &= \frac{1}{n} \sum_{i=1}^n (E_i^2 - 2ME_i + M^2) \\ &= \frac{1}{n} \sum_{i=1}^n E_i^2 - \frac{2M}{n} \sum_{i=1}^n E_i + M^2 \end{aligned}$$

Finalmente:

$$S^2 = \frac{1}{n} \sum_{i=1}^n E_i^2 - M^2$$

Algoritmo:

TANTAS VECES
COMO NUMERO
DE MUESTRAS



Datos de entrada:

- Tarjeta de número de muestras (una por proce-

so)

COLUMNAS	FORMATO	DESCRIPCION
1 - 4	I4	No. de muestras

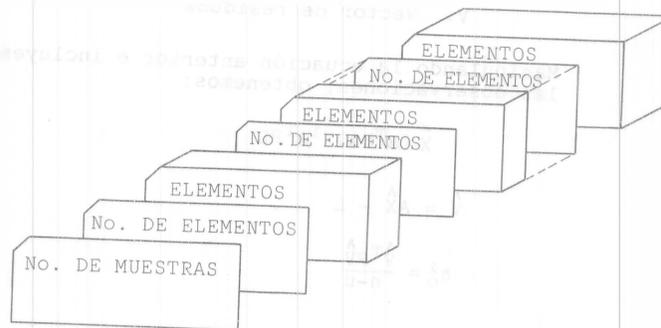
- Tarjeta de número de elementos de la muestra (una por cada muestra)

COLUMNAS	FORMATO	DESCRIPCION
1 - 4	I4	No. de elementos

- Tarjeta de elementos de la muestra (tantas como elementos)

COLUMNAS	FORMATO	DESCRIPCION
1 - 7	F7.3	elementos

DATOS DE ENTRADA



Resultados: Como se puede apreciar en el listado 4.3.1, los resultados son:

- Número de muestras a procesar
- Número de elementos de la muestra
- Elementos
- Media
- Desviación estándar

4.3.2 Ajuste de una Red de Nivelación por Mínimos Cuadrados.

Objetivo: Ajustar los valores observados en el levantamiento de campo de una red de nivelación.

Descripción: Este programa está escrito en lenguaje BASIC. En todo tipo de trabajos topográficos se efectúan mediciones redundantes para garantizar la precisión de los trabajos requeridos; por lo que es necesario determinar el valor más probable y la desviación estándar de los parámetros observados, así como los valores de las incógnitas involucradas. De acuerdo con el método de mínimos cuadrados, tenemos en forma matricial que:

$$AX = L + V$$

Donde:

A: Matriz de diseño

X: Vector de incógnitas

L: Vector de parámetros observados

V: Vector de residuos

Manipulando la ecuación anterior e incluyendo pesos a las observaciones, obtenemos:

$$\hat{X} = (A^T P A)^{-1} A^T P L$$

$$\hat{V} = A \hat{X} - L$$

$$\hat{\sigma}_0^2 = \frac{\hat{V}^T P \hat{V}}{n-u}$$

$$\hat{\Sigma}_{\hat{X}} = \hat{\sigma}_0^2 (A^T P A)^{-1}$$

donde:

\hat{X} : Vector de incógnitas ajustadas

P: Matriz de peso

\hat{V} : Vector de residuos ajustados

$\hat{\sigma}_0^2$: Varianza de peso unitario

N: Número de observaciones

u: Número de incógnitas

$\Sigma \hat{x}$: Matriz de varianzas y covarianzas.

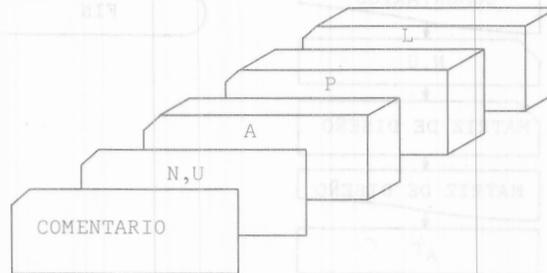
Algoritmo:



Datos de entrada:

- Tarjeta de comentarios
- Número de observaciones e incógnitas
- Matriz de diseño
- Matriz de peso
- Vector de observaciones

DATOS DE ENTRADA



Resultados: Según puede verse en el listado 4.3.2, los resultados del programa son:

- Encabezados
- Matriz de diseño
- Inversa de ecuaciones normales
- Valores ajustados y desviación estándar
- Varianza de peso unitario

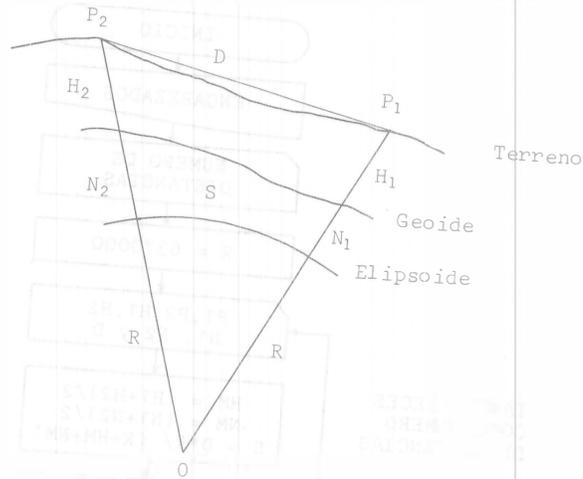
4.4 Problemas de Geodesia

4.4.1 Reducción de Distancias al Elipsoide

Objetivo: Reducir las observaciones realizadas en el

terreno al elipsoide de referencia adoptado.

Descripción: Este programa está escrito en lenguaje BASIC. Dado que la medición de distancias en la superficie terrestre se efectúa a diferentes altitudes, y es necesario tenerlos en un mismo marco de referencia, estas se reducen al elipsoide para realizar los cálculos de las posiciones geodésicas.



En la figura anteriores tenemos que P₁, P₂ puntos sobre el terreno; H₁, H₂ alturas ortométricas de P₁ y P₂; N₁, N₂ alturas geoidales correspondientes a P₁ y P₂; R radio medio terrestre; D, S distancias en el terreno y el elipsoide respectivamente. Así pues:

$$\frac{D}{S} = \frac{R+H_m+N_m}{R}$$

donde:

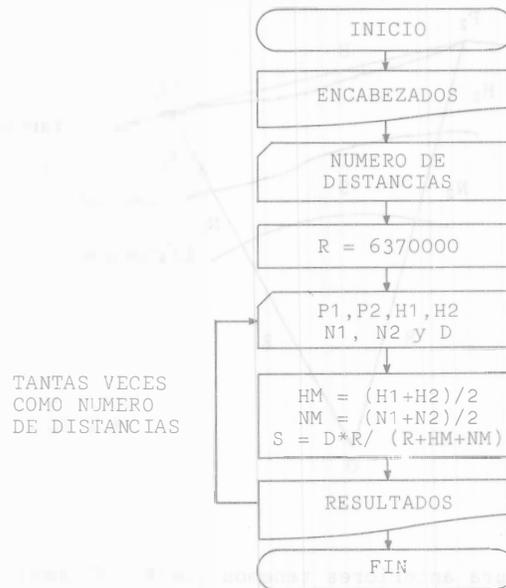
$$H_m = \frac{H_1 + H_2}{2}$$

$$N_m = \frac{N_1 + N_2}{2}$$

De la relación anterior tenemos que:

$$S = \frac{D \cdot R}{R + H_m + N_m}$$

Algoritmo:



Datos de entrada:

- Número de distancias a reducir.
- Punto 1, Punto 2, altura ortométrica 1, altura ortométrica 2, altura geoidal 1, altura geoidal 2, distancia (tantas tarjetas como distancias a reducir).



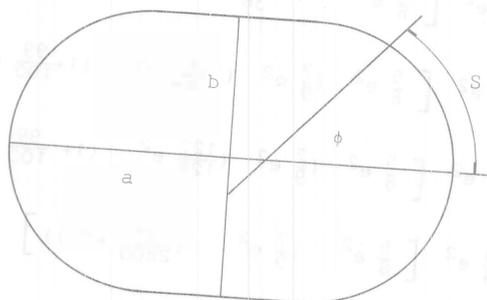
Resultados: Según puede verse en el listado 4.4.1, los resultados son:

- Encabezados
- Valores reducidos

4.4.2 Longitud de un Arco de Meridiano

Objetivo: Calcular la longitud en metros de un arco de meridiano desde el ecuador, hasta un punto de latitud conocida.

Descripción: Este programa está escrito en lenguaje FORTRAN. En trabajos geodésicos, frecuentemente es requerido determinar la distancia entre el ecuador y un punto de latitud conocida, sobre todo cuando se trabaja con proyecciones cartográficas. Este problema se define así:



donde:

- ϕ : Latitud
- a: Semieje mayor del elipsoide
- b: Semieje menor del elipsoide
- s: Arco de meridiano

La solución a este problema se da través de la integral elíptica:

$$s = a (1-e^2) \int_0^\phi (1-e^2 \text{sen}^2\phi)^{-3/2} d\phi$$

donde $e^2 = \frac{a^2 - b^2}{a^2}$ se le conoce como excentricidad

Resolviendo la integral en series se tiene que:

$$s = a (1-e^2) \left[A_0 \phi - \operatorname{sen} \phi \cos \phi (A_1 + A_2 \operatorname{sen}^2 \phi + A_3 \operatorname{sen}^4 \phi + A_4 \operatorname{sen}^6 \phi + A_5 \operatorname{sen}^8 \phi) \right]$$

Siendo:

$$A_0 = 1 + \frac{3}{4} e^2 \left[1 + \frac{15}{16} e^2 \left[1 + \frac{35}{36} e^2 \left[1 + \frac{63}{64} e^2 (1 + \frac{99}{100} e^2) \right] \right] \right]$$

$$A_1 = \frac{3}{4} e^2 \left[1 + \frac{15}{16} e^2 (1 + \frac{35}{36} e^2 (1 + \frac{63}{64} e^2 (1 + \frac{99}{100} e^2))) \right]$$

$$A_2 = \frac{3}{4} e^2 \left[\frac{5}{8} e^2 (1 + \frac{35}{36} e^2 (1 + \frac{63}{64} e^2 (1 + \frac{99}{100} e^2))) \right]$$

$$A_3 = \frac{3}{4} e^2 \left[\frac{5}{8} e^2 (\frac{7}{9} e^2 (1 + \frac{63}{64} e^2 (1 + \frac{99}{100} e^2))) \right]$$

$$A_4 = \frac{3}{4} e^2 \left[\frac{5}{8} e^2 (\frac{7}{9} e^2 (\frac{127}{128} e^2 (1 + \frac{99}{100} e^2))) \right]$$

$$A_5 = \frac{3}{4} e^2 \left[\frac{5}{8} e^2 (\frac{7}{9} e^2 (\frac{1917}{2800} e^4)) \right]$$

Según puede verse en el listado del programa, para facilitar la solución se extraen nueve factores, a los que se les llama "factores geodésicos", éstos son:

$$E (1) = \frac{99}{100} e^2$$

$$E (2) = \frac{63}{64} e^2$$

$$E (3) = \frac{35}{36} e^2$$

$$E (4) = \frac{15}{16} e^2$$

$$E (5) = \frac{3}{4} e^2$$

$$E (6) = \frac{5}{8} e^2$$

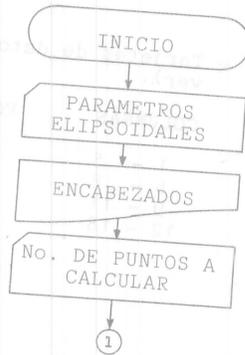
$$E (7) = \frac{7}{9} e^2$$

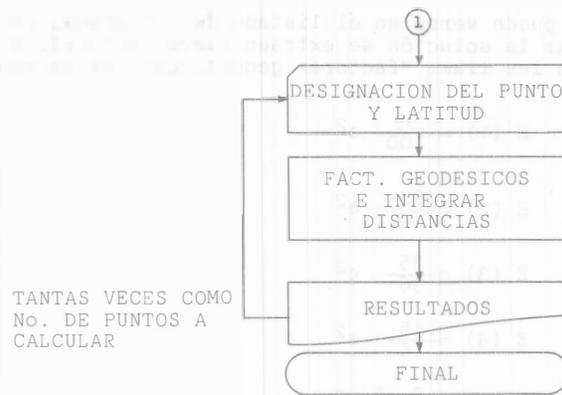
$$E (8) = \frac{27}{128} e^2$$

$$E (9) = \frac{71}{21.875} e^2$$

Los cuales se sustituyen en las fórmulas A₀ a A₅ para posteriormente dar la solución final con la fórmula.

Algoritmo:





Datos de entrada:

- Tarjeta de parámetros elipsoidales

COLUMNAS	FORMATO	DESCRIPCION
1 - 11	F11.3	semieje mayor
12 - 22	F11.9	excentricidad

- Tarjeta de número de casos a resolver.

COLUMNAS	FORMATO	DESCRIPCION
1 - 4	I4	No. de distancias a calcular

- Tarjetas de datos (tantas como casos a resolver).

COLUMNAS	FORMATO	DESCRIPCION
1 - 5	A5	Designación del punto
6 - 8	I3	Grados de latitud
9 - 11	I3	Minutos de latitud
12 - 18	F7.3	Segundos de latitud

TARJETAS DE DATOS



Resultados: Según puede apreciarse en el listado 4.4.2, los resultados arrojados por este programa son:

- Encabezados
- Designación del punto, latitud y distancia del ecuador al punto dado.

4.5 Problemas resueltos con calculadora de bolsillo.

No siempre la mejor solución a un problema numérico de Ingeniería Topográfica está dado por una computadora, ya que en ocasiones resulta más práctico el uso de una calculadora programable de bolsillo, sobre todo cuando se requiere la obtención inmediata de resultados en campo o el volumen de producción de información es relativamente bajo. Tratando de ejemplificar la utilidad de esta herramienta y conservando la técnica de la presentación de este documento, se resuelven a continuación 2 ejemplos sencillos, usando para el primero una calculadora Texas Instrument y para el segundo una Hewlett Packard.

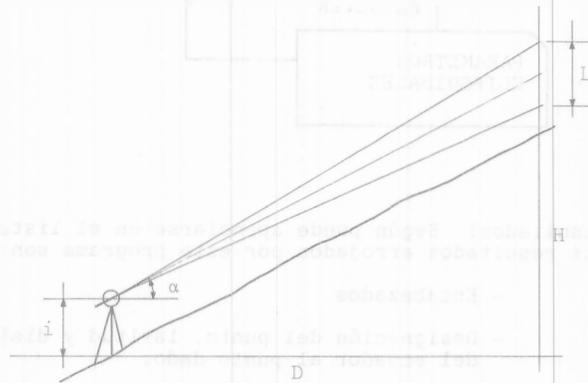
Se recomienda la consulta de los manuales correspondientes a estas calculadoras, previa a la discusión de los siguientes ejemplos.

4.5.1 Cálculo Estadimétrico de Distancias y Desniveles.

Objetivo: Calcular la distancia y el desnivel entre

dos puntos a partir de observaciones estadimétricas.

Descripción: Es un programa realizado en el código específico de la calculadora TI58/59, que suplente las tablas, el ábaco o la regla de estadía ya que una vez programada la calculadora, la solución se da rápidamente con presionar unas cuantas teclas. La estructura del problema es bien conocida:



donde:

L: Lectura estadimétrica

i: Altura de instrumento

D: Distancia horizontal

H: Distancia vertical

α : Angulo vertical

y la solución es:

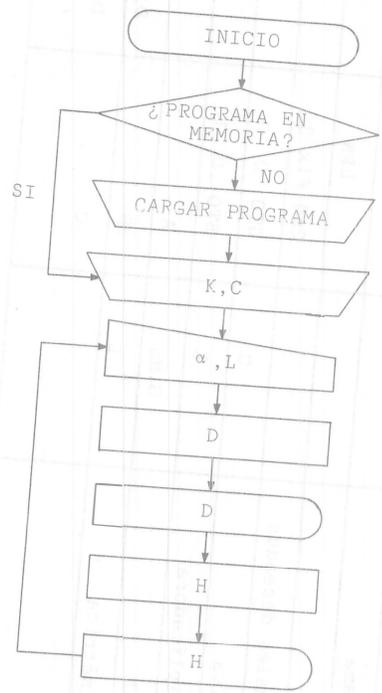
$$D = KL \cos^2 \alpha + c \cos \alpha$$

$$H = \frac{KL}{2} \operatorname{sen} 2 \alpha + c \operatorname{sena}$$

en las que:

- K: constante multiplicadora
- c: constante aditiva

Algoritmo:



Datos de entrada/Salida: En la tabla 4.5.1 se puede observar el manejo del programa y tanto los datos de entrada como los de salida. Se recomienda la prueba del programa como ejercicio.

4.5.2 Cálculo de Volúmenes por Método de Prismas Cuadrangulares.

Objetivo: Calcular el volumen de un prisma en función

PASO	DESCRIPCION	ENTRADA	PRESIONAR	PANTALLA
1			LRN	000 00
2	Cargar el programa			
3			LRN	0
4	Número de decimales deseados		2nd Fix 3	0.000
5	Constante aditiva	C	STO 0 0	C
6	Constante Multiplicadora	K	STO 0 1	K
7	Angulo Vertical	G.MS	A	0
8	Lectura Estadimétrica	L	B	Distancia horizontal
9			C	Desnivel

NOTA: Para más valores repetir pasos 7 al 9.

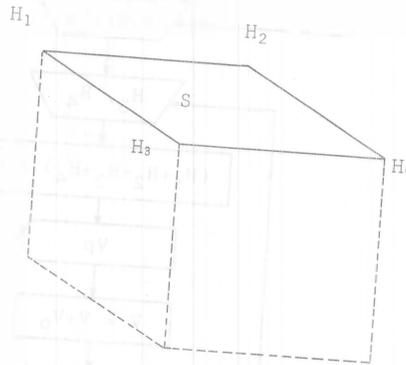
MEMORIAS UTILIZADAS

00 c
01 K
01 Angulo vertical
03 L, KL

TABLA 4.5.1 CALCULO ESTADIMETRICO DE DISTANCIAS Y DESNIVELES

de las alturas de los vértices.

Descripción: Es un programa realizado en el código específico de la calculadora HP11. En el cálculo de volúmenes se tienen que realizar una gran cantidad de operaciones elementales, las cuales se ejecutan con gran facilidad en una calculadora programable. En la siguiente figura se ilustra el problema para un prisma:



donde:

H_1, H_2, H_3, H_4 : altura de los vértices

S: área del cuadrilátero

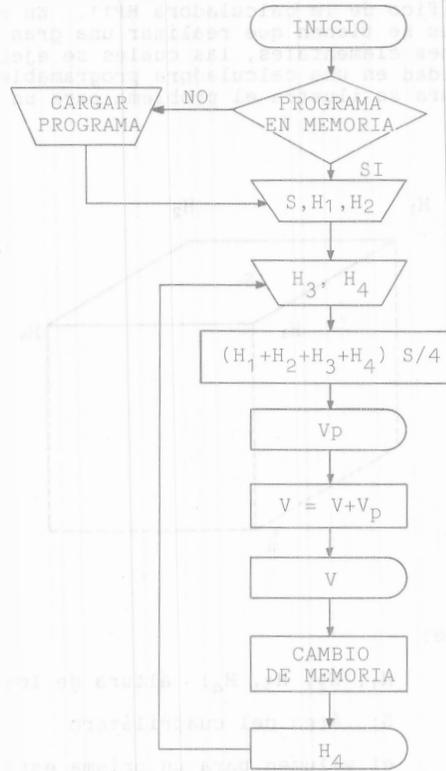
el volumen para un prisma está dado por:

$$V_p = (H_1 + H_2 + H_3 + H_4) S / 4$$

y el volumen para n prismas:

$$V = \sum_{i=1}^n V_{p_i}$$

Algoritmo:



Datos de entrada/salida: En la tabla 4.5.2, se muestran los datos de entrada y salida, así como la operación del programa.

ASO	DESCRIPCION	ENTRADA	PRESIONAR	PANTALLA
2	Cargar el programa		g P/R	
}			g P/R	
4	Superficie del prisma	S	STO 0	S
}	Altura i	hi	STO 1	hi
}	Altura i+1	hi+1	STO 2	hi+1
'	Altura i+2	hi+2	STO 3	hi+2
}	Altura i+3	hi+3	STO 4	hi+3
}	Volumen parcial		f A	Vp
)	Volumen acumulado		R/S	VA
			R/S	hn

NOTA: Para más valores repetir pasos 7 a 11

MEMORIAS UTILIZADAS

00 S
 01 hi
 02 hi+1
 03 hi+2
 04 hi+3
 05 Sumatoria
 06 Volumen parcial
 07 Volumen acumulado

TABLA 4.5.2 CALCULO DE VOLUMENES DE PRISMAS CUADRANGULARES

LISTADO 4.1.1

PROBLEMA INVERSO EN TOPOGRAFIA						
PROGRAMA ELABORADO EN EL DEPARTAMENTO DE TOPOGRAFIA POR:						
ING. MARIO A. PEYES IBARRA						
ING. ANTONIO HERNANDEZ NAVARRO						
DATOS DE PRUEBA						
VERTICE	A	Y	LADO	DISTANCIA	RUMBO	
1	510.000	170.000	1 3	385.000	0 0	0.000 W
3	125.000	170.000	3 9	417.043	37 41	39.266 NE
9	380.000	500.000	9 11	250.000	0 0	0.000 E
11	380.000	750.000	11 21	130.000	0 0	0.000 E
21	510.000	750.000	21 36	259.615	74 21	27.927 SE
36	760.000	080.000	36 75	619.000	0 0	0.000 S
75	760.000	70.000	75 100	262.468	17 44	40.815 NW
100	660.000	320.000	100 1	226.716	46 54	34.796 SW

LISTADO 4.1.2

CALCULO DE LOS ELEMENTOS PARA EL TRAZO DE UNA CURVA HORIZONTAL SIMPLE

PROGRAMA ELABORADO EN EL DEPARTAMENTO DE TOPOGRAFIA POR:
ING. ASTORIO HERNANDEZ NAVARRO
ING. BARTO A. KRYES IBARRA

```

*****
+ EST + P 0 + CADERNALEFOTO + DEFLEXION + 80.125 +
*****
+ P 1 + 10 + 151.198 + 13 59.56 +
*****
+ 5 + 10 + 140.000 + 13 19 11. +
*****
+ 1 + 10 + 179.000 + 10 19 11. +
*****
+ 3 + 10 + 160.000 + 7 19 11. +
*****
+ 2 + 10 + 80.000 + 4 19 11. +
*****
+ 1 + 10 + 60.000 + 1 19 11. +
*****
+ P C + P 1 + 10 + 51.198 + 0 0 0. +
*****

```

```

RADIO: 191.073
FUENTE DE REFLEXION: SANTIAGO: 51.198
CURVA DE ENTRADA: d.802
GRAD DE CURVATURA: 0.600
CORONA DE SALIDA: 11.198
DEFLEXION TOTAL: 30.600

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63

12345678901234567890123456789012345678901234567890123456789012345678901234567890

LISTADO 4.2.1

ACIQUI ASTRONOMICO DE UNA LINEA							
PROGRAMA ELABORADO EN EL DEPARTAMENTO DE TOPOGRAFIA PURI							
ING. ANTONIO REBEAGUEZ HAVARRU							
ING. NARIB A. REYES IBARRA							
10	FECHA:	5 1 1979	LATITUD:	28 37 43.00	ACIQUI DEL PUNTO	1 AL PUNTO	2 142 51 47.98
13	FECHA:	5 1 1979	LATITUD:	28 37 43.00	ACIQUI DEL PUNTO	1 AL PUNTO	2 142 52 4.29
16	FECHA:	5 1 1979	LATITUD:	28 37 43.00	ACIQUI DEL PUNTO	1 AL PUNTO	2 142 52 14.95
19	FECHA:	5 1 1979	LATITUD:	28 37 43.00	ACIQUI DEL PUNTO	1 AL PUNTO	2 142 52 29.07
20							
21							
22							
23							
24							
25							
26							
27							
28							
29							
30							
31							
32							
33							
34							
35							
36							

LISTADO 4.2.2

TRANSFORMACION DE COORDENADAS CELESTES				
PROGRAMA ELABORADO EN EL DEPARTAMENTO DE TOPOGRAFIA POR:				
ING. ANTONIO BERNARDEZ NAVARRO				
ING. KARIO A. REYES IBARRA				
27/11/1949				
ANGULO HORARIO	DECLINACION	ALTURA	AZIMUT	
304 15 42.00	-16 38 39.00	24 26 1.60	299 39	12.14
14 45 12.00	-26 30 0.00	61 55 9.41	17 50	7.71
71 82 15.00	-14 16 11.00	11 48 16.00	70 2	37.27

LISTADO 4.3.1721J

1	CALCULO DE MEDIA Y DESVIACION ESTANDAR DE UNA MUESTRA LINEAL	
2	SERIE ASESORADA EN EL DEPARTAMENTO DE TOPOGRAFIA FOR:	
3	PROGRAMA ELABORADO EN EL DEPARTAMENTO DE TOPOGRAFIA FOR:	
4	ING. MARIO A. PEYES IBARRA	
5	ING. ANTONIO BERRANDEZ NAVARRO	
6		
7	MUESTRA: 1	
8	NUMERO DE ELEMENTOS: 3	
9	ELEMENTOS	
10	1)	3.000
11	2)	4.000
12	3)	5.000
13	MEDIA: 4.000 D.E.: 0.867	
14		
15	MUESTRA: 2	
16	NUMERO DE ELEMENTOS: 10	
17	ELEMENTOS	
18	1)	100.500
19	2)	101.000
20	3)	100.900
21	4)	99.990
22	5)	100.000
23	6)	100.100
24	7)	102.000
25	8)	100.000
26	9)	100.200
27	10)	100.300
28	MEDIA: 100.499 D.E.: 0.372	
29		
30		
31		
32		
33		
34		
35		
36		
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		
50		
51		
52		
53		
54		
55		
56		
57		
58		
59		
60		
61		
62		
63		
64		

LISTADO 4.3.2

ADJUSTE DE UNA RED DE NIVELACION POR TRIANGULOS CUADRADOS
PROGRAMA ELABORADO EN EL DEPARTAMENTO DE TOPOGRAFIA POR:

ING. RAFAEL HERNANDEZ NAVARRE
ING. RAFAEL A. REYES TORRES

PRUEBA
MATRIZ DE DISCRETO
0 1 0
0 0 1
0 1 1
-1 0 0
-1 0 1
-1 1 0
MATRIZ INVERSA DE FUNCIONES DERIVADAS
.5 .25 .25
.25 .5 .25
.25 .25 .5

VALORES AJUSTADOS
Cotas 0.6.7
1.065 0.0707
0.160 0.0707
12.595 0.0707

VARIANZA DE PESO UNITARIO
0.0017

* D.E.: DESVIACION ESTANDAR

234567890123456789012345678901234567890123456789012345678901234567890

LISTADO DE PROGRAMAS

A N E X O

I. LISTADO DE PROGRAMAS

T.1.- PROGRAMA DISTOP

```

1 10 REM PROBLEMA INVERSO EN TOPOGRAFIA
2 20 REM
3 30 REM Lee COMENTARIO
4 40 READ CS
5 50 PRINT TAB(10),"PROBLEMA INVERSO EN TOPOGRAFIA"
6
7 60 PRINT
8 70 PRINT TAB(10),"PROGRAMA ELABORADO EN EL DEPARTAMENTO DE TOPOGRAFIA POR:"
9 80 PRINT
10 90 PRINT TAB(25),"ING. NARIU A. REYES IDARRA"
11 100 PRINT TAB(25),"ING. ANTONIO HERNANDEZ NAVARRO"
12 110 PRINT
13 120 PRINT CS
14 130 PRINT
15 140 PRINT "VERTICE";TAB(12);"X";TAB(25);"Y";TAB(37);"DADO";TAB(45);"DISTANCIA";TAB(56);"RUMBO"
16 150 PRINT
17 160 REM Lee NUMERO DE VERTICES
18 170 READ N
19 180 M=N+1
20 190 DIM P(M),X(N),Y(N)
21 200 REM Lee DESIGNACION Y COORDENADAS
22 210 FOR I=1 TO 3
23 220 READ P(I),X(I),Y(I)
24 230 NEXT I
25 240 P(M)=P(1)
26 250 X(M)=X(1)
27 260 Y(M)=Y(1)
28 270 REM CALCULO DE DISTANCIA Y RUMBO
29 280 FOR I=1 TO N
30 290 DI=SQRT((X(I)-(I+1))^2+(X(I)-X(I+1))^2)
31 300 RB=0
32 310 IF Y(I)=Y(I+1) THEN 360
33 320 RB=ATN((X(I)-X(I+1))/(Y(I)-Y(I+1)))
34 330 RB=RB*57.29578
35 340 RB=ABS(RB)
36 350 REM DETERMINA COORDENATE
37 360 IF Y(I+1)>Y(I) AND X(I+1)>X(I) THEN 470
38 370 IF Y(I+1)>Y(I) AND X(I+1)<X(I) THEN 550
39 380 IF Y(I+1)<Y(I) AND X(I+1)<X(I) THEN 510
40 390 IF Y(I+1)<Y(I) AND X(I+1)>X(I) THEN 490
41 400 IF Y(I+1)=Y(I) AND X(I+1)>X(I) THEN 500
42 410 IF Y(I+1)>Y(I) AND X(I+1)=X(I) THEN 570
43 420 IF Y(I+1)=Y(I) AND X(I+1)<X(I) THEN 450
44 430 QS=" S"
45 440 GOTO 580
46 450 QS=" N"
47 460 GOTO 580
48 470 QS=" NE"
49 480 GOTO 580
50 490 QS=" SE"
51 500 GOTO 580
52 510 QS=" SW"
53 520 GOTO 580
54 530 QS=" NW"
55 540 GOTO 580
56 550 QS=" E"
57 560 GOTO 580
58 570 QS=" W"
59 580 G=INT(RB)
60 590 N=(RB-G)*60
61 600 M=INT(N)
62 610 S=(N-M)*60
63
123456789012345678901234567890123456789012345678901234567890

```

```

1 620 REM IMPRIME RESULTADOS
2 630 PRINT USING"####",P(1);
3 640 PRINT USING"#####.###",X(1);
4 650 PRINT USING"#####.###",Y(1);
5 660 PRINT " ";
6 670 PRINT USING"####",P(I);
7 680 PRINT USING"#####.###",P(I+1);
8 690 PRINT USING"#####.###",DI,
9 700 PRINT USING"####",G;
10 710 PRINT USING"###",M;
11 720 PRINT USING"###.###",S;
12 730 PRINT QS
13 740 PRINT QS
14 750 DATA 1
15 760 DATA DATOS DE PRUEBA
16 770 DATA 8
17 780 DATA 1,510,170
18 790 DATA 3,125,170
19 800 DATA 7,380,300
20 810 DATA 11,560,750
21 820 DATA 21,510,750
22 830 DATA 30,700,000
23 840 DATA 15,700,70
24 850 DATA 100,000,320
25
26
27
28
29
30
31
32
33
34
35
36
37

```



```
1 C
2
3
4 C
5 C
6 C
7 DIMENSION X(60),D(60)
8 RA=57.29576
9
10 C
11 C
12 C
13 C
14 C
15 C
16 C
17 C
18 C
19 C
20 C
21 C
22 C
23 C
24 C
25 C
26 C
27 C
28 C
29 C
30 C
31 C
32 C
33 C
34 C
35 C
36 C
37 C
38 C
39 C
40 C
41 C
42 C
43 C
44 C
45 C
46 C
47 C
48 C
49 C
50 C
51 C
52 C
53 C
54 C
55 C
56 C
57 C
58 C
59 C
60 C
61 C
62 C
63 C
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0
```

PROGRAMA PARA EL CALCULO DE LOS ELEMENTOS PARA EL TRAZO DE UNA CURVA HORIZONTAL SIMPLE

DIMENSION X(60),D(60)
RA=57.29576

LEE NUMERO DE CURVAS
READ(S,100)N

EMPIEZA CICLO PARA CALCULAR CURVAS
DO 10 J=1,N
I=1

LEE ELEMENTOS DE LA CURVA
READ(S,110)FI,DELTA,C,G

RAJO
 $R=C/(\sin(G/2./RA)*Z)$
SUBTANGENTE
 $ST=R*\tan(DELTA/2./RA)$

PUNTO DE COMIENZO
FC=I-51
X(I)=PC
D(I)=0.
I=I+1

LONGITUD DE LA CURVA
LC=DELTA*C/G
PUNTO DE TERMINO
FT=PC+LC

DETERMINA SI EL PC ES CIRCUNFERENTE CERRADO
IF(INT(PC/C).EQ.PC/C) GOTO 20

CALCULA CORDA Y DEFLEXION DE ENTRADA SI SON NECESARIOS
CE=INT(PC/C+1)*C-PC
OE=ASIN(C./Z.*P)
X(I)=X(I-1)+CF
D(I)=D(I-1)+E*FA
I=I+1

CALCULA CADERNIEROS Y DEFLEXIONES
20 X(I)=X(I-1)+C
D(I)=D(I-1)+G/Z.
I=I+1

```
1 IF((X(1-1)+C).LT.PT) GOTO 20
2
3 C
4 DETERMINA SI EL PT ES CADEMIENTO CERRADO
5 C
6 IF (LW*(PT/C).EQ.PT/C) GOTO 30
7
8 C
9 CALCULA CUERDA Y DEFLEXION DE SALIDA SI SON NECESARIOS
10 C
11 CS=PT-INT(PT/C)*C
12 DS=ASIN(CS/(2*R))
13 X(1)=X(I-1)+CS
14 U(I)=U(I-1)+DS*R
15 I=I+1
16 30 CONTINUE
17 I=-1
18 WRITE(G,12U)
19 CALL CAMBIA(X(1),D(1),IG,M,S,IN)
20 WRITE(G,13U)IN,X(1),IG,M,S
21 DO 40 I=-1,2,-1
22 LPU=L-1
23 CALL CAMBIA(X(L),D(L),IG,M,S,IN)
24 WRITE(G,14U)LPU,IN,X(L),IG,M,S
25 CALL CAMBIA(X(1),D(1),IG,M,S,IN)
26 WRITE(G,15U)IN,X(1),IG,M,S
27 WRITE(G,16U)R,ST,P1,CE,CS,G,DELTA
28 CONTINUE
29 C
30 FORMAT
31 C
32 C
33 100 FORMAT(14)
34 110 FORMAT(F9.3,F7.3)
35 120 FORMAT(10X,'CALCULO DE LOS ELEMENTOS PARA EL TRAZO',/,10X,'DE
36 10VA CURVA HORIZONTAL SIMPLE',/,11X,'PROGRAMA ELABORADO EN EL OEP
37 2ARIAMENIO DE TOPOGRAFIA POR:',/,20X,'ING. ANTONIO HERRANDEZ NAVAR
38 3FO',/,20X,'ING. RAJID A. LITES IBARRA')
39 130 FORMAT(//,10X,'CADMIENTO',/,10X,
40 1*' EST * E U * CADMIENTO * DEFLEXION * NOTAS *',/,10X,40(1H*),/
41 2,10X,' * P 1 *',13,' +',F8.3,' *',213,F4.0,' *
42 3,48(1H*))
43 140 FORMAT(10X,' *',14,' *',13,' +',F8.3,' *',213,F4.0,' *
44 1*' ,/,10X,48(1H*))
45 150 FORMAT(10X,' P C * P 1 *',13,' +',F8.3,' *',213,F4.0,' *
46 1,/,10X,48(1H*))
47 160 FORMAT(//,10X,'RADIO:',F8.3,10X,'SURTANGENTE:',F8.3,/,10X,'PUNIO
48 1 DE INTERSECCION:',F9.3,/,10X,'CUERDA DE ENTRADA:',F7.3,10X,'CUE
49 2RDA DE SALIDA:',F7.3,/,10X,'GRADO DE CURVATURA:',F7.3,10X,'DEFLEX
50 3ION TOTAL:',F7.3)
51 END
52 SUBROUTINE CAMBIA(X,G,IG,M,S,I)
53 I=X/1000
54 A=X-1*1000
55 IG=INT(G)
56 TEMP=(G+IG)*60
57 A=INT(TEMP)
58 S=(TEMP-A)*60
59 RETURN
60 END
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
```

I.3.- PROGRAMA SOL

```
C
C
C      PROGRAMA PARA EL CALCULO DEL ACIMUT DE UNA LINEA EN FUNCION DE
C      LAS DISTANCIAS ZENITALES ABSOLUTAS
C
C      RA=57.295779
C
C      LEE FECHA DE OBSERVACION, LATITUD, PUNTO 1 Y PUNTO 2
C
C      WRITE(6,1000)
10  READ(5,100,END=9000) ID,MES,IAN,IA,MDA,SLA,P1,P2
C      FI=SEA(LA,MLA,SLA)/RA
C
C      LEE ANGULO SENAL-SOL, DISTANCIA ZENITAL, TEMPERATURA Y PRESION
C
C      READ(5,110) IASS,MASS,SASS,IZ,EZ,SZ,TEM,IPRE
C      ASS=SEA(IASS,MASS,SASS)
C      Z=SEA(IZ,EZ,SZ)/RA
C      CALL ZC(2,TEM,IPRE)
C
C      LEE HORA DE OBSERVACION, DEL PASO, DECLINACION Y VARIACION HORARIA
C
C      READ(5,120) INO,MRO,SHO,IMP,MPE,SHP,IDE,MEZ,SOE,VH
C      HU=SEA(INO,MRO,SHO)
C      HP=SEA(IMP,MPE,SHP)
C      DE=SEA(IDE,MEZ,SOE)/RA
C      VHEVH(3000,*RA)
C      CALL DRG(HU,HP,DE,VH)
C      U=(SIN(DE)-COS(Z)*SIN(FI))/(SIN(Z)*COS(FI))
1   FORMAT(F10.7)
C      U=ACOS(U)*RA
C      IF (HU.GE.12.) U=360-U
C      IF (U.LT.0.) U=360-U
C      IAZ=INT(U)
C      U=(U-IAZ)*60.
C      *IAZ=INT(U)
C      SAZ=(U-IAZ)*60.
C      WRITE(6,1010)ID,MES,IAN,IA,MDA,SLA,P1,P2,IAZ,MAZ,SAZ
100  FORMAT(2I3,15,2I3,F6.1,2A4)
110  FORMAT(2(2I3,F6.2),F6.1,14)
120  FORMAT(3(2I3,F6.3),F6.2)
1000  FORMAT(15A,'ACIMUT ASTRONOMICO DE UNA LINEA',//,10A,'PROGRAMA ELAB
C      ORADO EN EL DEPARTAMENTO DE TOPOGRAFIA POR:',//,25A,'1 ANO. ANTONIO
C      ZHERRANDEZ WAYARRU',//,25A,'ING. MAFIO A. REYES IDARRA',//)
1010  FORMAT(5X,'FECHA:',2(3,15,5A,'L111UD:',2I4,F6.2,5X,'ACIMUT DEL PU
C      NTO 1',A4,' AL PUNTO 1',A4,14,13,F6.2,//)
C      GO TO 10
9000  END
C      FUNCION SEA(IG,R,S)
C      IS=0
C      IF (IG.LT.0) IS=1
C      IG=ABS(IG)
C      SEA=IG+(R+S/60.)/60.
C      IF (IS.EQ.1) SEA=SEA*-1.
C      RETURN
C      END
C      SUBROUTINE ZC(Z,TEM,IPRE)
C      R=2.437971E-04*IAN(2)
C      T=1./[1.40.004*TEM]
C      P=IPRE/700.
C      F=4.2663E-5*SIN(Z)
```

34567890123456789012345678901234567890123456789012345678901234567890

```
1
2
3      Z=Z+K*IT*U=P
4      RETURN
5      ENDD
6      SUBROUTINE DDU(HO,RP,DE,VH)
7
8      VU=VH*DU
9      DE=DE+VU
10     RETURN
11     ENDD
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
1234567890123456789012345678901234567890123456789012345678901234567890
```

I.4.- PROGRAMA TCOOR

Item	Descripción	Cantidad	Valor Unitario	Valor Total
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```
2 1J RES PROGRAMA PARA LA TRANSFORMACION DE COORDENADAS CELESTES
3 2J RES
4 3J PRINT TAB(15);"TRANSFORMACION DE COORDENADAS CELESTES"
5 4J PRINT
6 5J PRINT TAB(10);"PROGRAMA ELABORADO EN EL DEPARTAMENTO DE TOPOGRAFIA PURA"
7 6J PRINT
8 7J PRINT TAB(25);"ING. ANTONIO BERMUDEZ NAVARRO"
9 8J PRINT TAB(25);"ING. NARIO A. REYES IBARRA"
10 9J FOR I=1 TO 3
11 10J PRINT
12 11J READ J
13 12J RES LEE NUMERO DE PARES DE COORDENADAS, LATITUD Y FECHA
14 13J READ GP,CL,FI,SI,LES
15 14J PRINT TAB(25);GP;SI;FI;TAB(55);PES
16 15J PRINT
17 16J PRINT TAB(4);"ANGULO HORARIO";TAB(25);"DECLINACION";TAB(45);"ALTURA";
18 17J PRINT TAB(63);"AZIMUT"
19 18J PRINT
20 19J READ J,295774
21 20J FI=(GI*(PI+SI/60)/60)/RA
22 21J SI=SIGN(FI)
23 22J CI=COS(FI)
24 23J FOR J=1 TO NP
25 24J READ GH,WH,SI,EG,RE,SE
26 25J H=COS((WH+SI/60)/60)/HA
27 26J IS=0
28 27J IF EG<0 THEN IS=1
29 28J C=COS(EG)+(WB+SE/60)/60)/RA
30 29J IF IS=1 THEN D=-1
31 30J SH=SIGN(H)
32 31J CR=COS(CI)
33 32J SD=SIGN(D)
34 33J CC=COS(CI)
35 34J CZ=SP*SD+CF*CD*Ch
36 35J T=SP*(1-CZ/CC)
37 36J SA=CD*SH/T
38 37J Z=INT(1/CZ)*RA
39 38J A=90+Z
40 39J AZ=ATN(SA/SQR(1-SA*SA))*RA
41 40J GZ=INT(A)
42 41J TI=(A-GZ)*60
43 42J *Z=INT(TI)
44 43J SZ=(TI-FZ)*60
45 44J IS=0
46 45J IF AZ<0 THEN AZ=360+AZ
47 46J GA=INT(AZ)
48 47J IZ=(AZ-GZ)*60
49 48J WA=INT(TZ)
50 49J SA=(TZ-FZ)*60
51 50J PRINT TAB(5);
52 51J PRINT USING"###",GH;
53 52J PRINT USING"###",WH;
54 53J PRINT USING"###.###",SI;
55 54J PRINT TAB(23);
56 55J PRINT USING"###",FI;
57 56J PRINT USING"###",RE;
58 57J PRINT USING"###.###",SE;
59 58J PRINT USING"###.###.###",GZ;
60 59J PRINT USING"###",Z;
61 60J PRINT USING"###.###",SZ;
62 61J PRINT USING"#####",GA;
63
```

12345678901234567890123456789012345678901234567890123456789012345678901234567890

FAC. DE INGENIERIA
BIBLIOTECAS



```

1
2 620 PRINT USING"###",NA;
3 630 PRINT USING"###.###",SA
4 640 PRINT
5 650 NEXT J
6 659 DATA 3,19,24,00,27/11/1999
7 670 DATA 304,15,42,-16,38,30
8 680 DATA 14,45,12,-20,30,00
9 690 DATA 71,42,15,-14,18,11
10 700 END
11
12
13
14
15
16
17
18
19
20
21
22 603917
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8

```

BIBLIOTECA
FAC. DE INGENIERIA



I.5.- PROGRAMA MEDIA

603a17

I.6.- PROGRAMA AJUS

I.7.- PROGRAMA RDE

```

1 10 REM PROGRAMA PARA LA REDUCCION DE DISTANCIAS AL EIPSOIDE
2 20 REM
3 30 DECLARE DOUBLE H,D1,D2,P1,P2,Q,R
4 40 PRINT TAB(10);"REDUCCION DE DISTANCIAS AL EIPSOIDE"
5 50 PRINT
6 60 PRINT TAB(10);"PROGRAMA ELABORADO EN EL DEPARTAMENTO DE TOPOGRAFIA FORI"
7 70 PRINT
8 80 PRINT TAB(25);"ING. ANTONIO HERNANDEZ BAYARRO"
9 90 PRINT TAB(25);"ING. CARLOS H. FLEIS IBARRA"
10 100 FOR I=1 TO 3
11 110 PRINT
12 120 NEXT I
13 130 REM DE NUMERO DE DISTANCIAS Y REDUCCION
14 140 INPUT N
15 150 PRINT " ESTACION DISTANCIA ALIURA ALIURA ALIURA ALIURA DISTANCIA"
16 160 PRINT " DE A TERRENO ORTOMETRICA ORTOMETRICA GEODAL GEODAL EIPSOIDE"
17 170 FOR J=1 TO N
18 180 REM DE P1,P2,P1,P2,A1,A2
19 190 INPUT P1,P2,P1,P2,A1,A2
20 200 REM DE R1,R2
21 210 INPUT R1,R2
22 220 D1=SQR(R1^2+R2^2)
23 230 D2=SQR(R1^2+R2^2)
24 240 REM CALCULO DE DISTANCIA REDUCIDA
25 250 REM D=D1*(1+D1^2/2)
26 260 REM D=D2*(1+D2^2/2)
27 270 PRINT "RESULTADOS"
28 280 PRINT USING "###.###";D1
29 290 PRINT USING "###.###";D2
30 300 PRINT USING "#####.###";D1
31 310 PRINT USING "#####.###";D2
32 320 PRINT USING "#####.###";D1
33 330 PRINT USING "#####.###";D2
34 340 PRINT USING "#####.###";D1
35 350 PRINT USING "#####.###";D2
36 360 PRINT
37 370 NEXT J
38 380 END
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53

```

I.8. PROLOGO - 8.1

I.8.- PROGRAMA DSM

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63

```

PROGRAMA PARA CALCULAR DISTANCIAS DEFIJADAS
 IMPLICITAMENTE (A=1, B=2)
 DIMENSION B(5)

DEF PARÁMETROS DEL EIPSOIDE.
 REAL(5,100) A,B,EZ
 CALL FACTOR(EZ,B)
 CALL INI

DEF NÚMERO DE PUNTOS A CALCULAR
 REAL(5,100) N1

EMPIEZA CICLO PARA CÁLCULO
 DO 10 J=1,N1
 SUMA=0.100

DEF DESIGNACION DEL PUNTO Y DATI100
 REAL(5,120) P,LABANDA,SUA
 P1=(P(1)+1.0)/100.00,LABANDA/5000.00+1.145329250-1
 S1=SIGN(P1)
 C1=DCOS(P1)
 S2=S1*S1
 SP=S2
 DU=0
 DO 20 I=3,6
 SUP=SUM(A+SP*I(I))
 SP=SP*S2
 CUP=C1*SP
 S2A=1.00-EZ)*(P1+S1)-S1*C1+(P(2)+SUMA))
 LABANDA=LABANDA+P
 P1=(P(1)+1.0)/100.00 GO TO 10
 CALL INI
 GO TO 50

CONTINUA

FORMATOS
 100 FORMAT(F11.5,F11.9)
 110 FORMAT(I10)
 120 FORMAT(A5,Z13,F7.3,F13.3,' ' * 7)
 1000 END

SUBROUTINE FACTOR(EZ,B)
 SUBROUTINE PARA EL CÁLCULO DE FACTORES GEODESICOS
 IMPLICITAMENTE (A=1, B=2)
 DIMENSION E(5),S(9)
 E(1)=E2+9.100/100.00
 E(2)=E2+8.100/100.00
 E(3)=E2+35.00/100.00
 E(4)=E2+15.00/100.00
 E(5)=E2+3.00/100.00
 E(6)=E2+5.00/100.00
 E(7)=E2+7.00/100.00

**I.9.- PROGRAMA ESTADIA
(TI 58/59)**



2

LOC.	CODIGO	TECLA
000	76	2nd LbL
	11	A
	88	2nd D.MS
	42	STO
	02	02
	25	CLR
	91	R/S
	76	2nd LbL
	12	B
	65	X
010	43	RCL
	01	01
	95	=
	42	STO
	03	03
	65	X
	43	RCL
	02	02
	39	2nd Cos
	33	X ²
020	85	+
	43	RCL
	02	02
	39	2nd Cos
	65	X
	43	RCL
	00	00
	95	=
	91	R/S
	76	2nd LBL
030	13	C
	43	RCL
	02	02
	65	X
	02	2
	95	
	38	2nd sin
	65	X
	43	RCL
040	55	%
	02	2
	85	+
	43	RCL
	02	02
	38	2nd sin
	65	X
	43	RCL
	00	00
	95	=
050	91	R/S

LOC	CODIGO	TECLA
000	422111	f PRGM
	45 1	f LBL A
	45 2	RCL 1
	45 2	RCL 2
	40	+
	36	ENTER
	45 3	RCL 3
	40	+
	36	ENTER
	45 4	RCL 4
010	40	+
	36	ENTER
	4	÷
	10	÷
	44 5	STO 5
	36	ENTER
	45 0	RCL 0
	20	X
	44 6	STO 6
	31	R/S
020	43 35	g CLX
	45 7	RCL 7
	45 6	RCL 6
	40	+
	44 7	STO 7
	31	R/S
	43 35	g CLX
	45 3	RCL 3
	44 1	STO 1
	43 35	g CLX
030	45 4	RCL 4
	44 2	STO 2
	43 35	g CLX
	45 2	RCL 2
	31	R/S

I.I.O-PROGRAMA VOLUMEN

(111)

B I B L I O G R A F I A

ARECHIGA G., RAFAEL (1979) "Aplicaciones de Computación a problemas de Ingeniería"; Ed. LIMUSA, México

BLACHUT, T.J., CHRZANOWSKI, A., SAASTAMOINEN, J.H. (1979) "Cartografía y Levantamientos Urbanos", I.P.G.H.; México

FORSYTHE, A.I., KEENAN, T.A., ORGANICK, E.I., STENBERG, W. (1979) "Lenguajes de Diagramas de Flujo"; Ed. LIMUSA, México

(1978) "Programación BASIC"; Ed. LIMUSA; México

(1978) "Programación FORTRAN"; Ed. LIMUSA, México

HEWLETT PACKARD "Reference Manual"; USA

HIGASHIDA MIYABARA, SABRO (1979) "Topografía General" México

MACCRACKEN, D.D., (1983) "Programación FORTRAN"; Ed. LIMUSA, México

MEDINA PERALTA, MANUEL (1978) "Elementos de Astronomía de Posición"; Ed. LIMUSA, México

TEXAS INSTRUMENT "Owner Manual"

TRACTON, KEN (1978) "The BASIC Cookbook"; Ed. TAB BOOKS INC; USA

TRAKHTEN BROT, B.A. (1973) "Algoritmos y Computadoras" Ed. LIMUSA; México

VANICEK, PETE (1979) "LEAST SQUARE"; University of New Brunswick; Canadá