142

UNIVERSIDAD NACIONAL AUTONOMA DE MÉXICO

FACULTAD DE QUÍMICA

ALGUNOS PROGRAMAS ENFOCADOS A INGENIERÍA QUÍMICA PARA CALCULADORAS HEWLETT PACKARD MODELOS 48

T E S I S

QUE PARA OBTENER EL TÍTULO DE

INGENIERO QUÍMICO

P R E S E N T A :

JULIO ALEJANDRO SÁNCHEZ DE LA TORRE

206710

MÉXICO, D.F. *2000*

Jurado Asignado

Presidente:	Jaime Noriega Bernechea
Vocal:	Ramiro Domínguez Danache
Secretario:	Alejandro Allier Montaño
1er Suplente:	Norma Gisela González Mariscal
2° Suplente:	Luis Angel López la Torre

Sitio donde se desarrolló el Tema:

Biblioteca de la Facultad de Química
Biblioteca Central de la UNAM
Instituto Mexicano del Petróleo

Alejandro Allier Montaño
Asesor

Armando Tomás Rosas Molina
Supervisor Técnico

Julio Alejandro Sánchez de la Torre
Sustentante

Agradecimientos:

A mis padres y a mis hermanos por el apoyo ofrecido. A mi tía Estelita por guiarme. A mis cuñados y cuñadas por su ayuda. A mis profesores de la Facultad quienes me apoyaron moral y económicamente. A mis compañeros de generación tanto amigos por incondicionales como enemigos que me enseñaron la diferencia. A todos los integrantes del Hospital San Fernando por restaurar mi salud. Al ing. Armando por apoyarme a conseguir las becas del IMP. A Triny, Vicente y Roberto por su apoyo en lograr la tesis. A Boris y la sra. Silvia por los favores que me han hecho en el IMP. A los ingenieros del IMP por echarme porras. A Gypsy y Marky por acompañarme las noches de desvelo. A Víctor y Gerónimo por resolver mis dudas sobre la tesis. A mis sinodales por revisar mi trabajo para mi titulación. A Madonna por que su música me permitió permanecer despierto en las noches de trabajo. A Mónica y Marisol por compartir tantas cosas. Y Finalmente a Alejandro por dirigir mi tesis. Me disculpo por si olvido a alguien.

Dedicatoria

A Ricardo Gutiérrez Díaz, por quien sin su amistad jamás lo hubiera logrado.

INDICE

Introducción

Si bien es sabido que nuestro entorno evoluciona rápidamente, hay que reconocer también que lo mismo sucede con el mundo de las computadoras. Recordando que las computadoras eran máquinas de gran tamaño con baja capacidad, nos parece mentira contar ahora con computadoras del tamaño de nuestra mano. Estas nuevas máquinas superan por mucho en capacidad y velocidad a aquellos modelos de antaño.

En algún tiempo fueron las reglas de cálculo las que permitían realizar operaciones en las aulas de las instituciones educativas. Para ello se preparaban cursos para aprender a manejarlas y obtener de ellas el mayor provecho posible. Hoy en día los alumnos de la Facultad de Química cuentan con calculadoras avanzadas, y cuyo potencial en la mayoría de los casos, desconocen. Debido a la diversidad de modelos se dificulta tener un solo curso que permita a los estudiantes a aprender de la misma manera que se hacía con la regla de cálculo.

Las calculadoras Hewlett Packard utilizan un procesador Saturn, que mantiene la memoria aun después de apagarse. Esto permite ejecutar programas para operaciones muy repetitivas o muy largas una y otra vez.

Es por ello que esta tesis pretende incursionar en la programación de calculadoras HP 48G y HP 48GX, que son las calculadoras de Hewlett Packard más usadas en la actualidad. Algunos programas se pueden utilizar en el modelo 48SX con algunas modificaciones.

Al momento de realizar esta tesis, se comienza a introducir en el mercado el modelo 49G, que seguramente permitirá en gran medida hacer uso de los programas hechos para los modelos antes mencionados. La existencia de calculadoras 48 en alumnos de segundo semestre asegura la utilidad de esta tesis para por lo menos 3 años. Este tiempo es suficiente, ya que inclusive los programas de compañías como Microsoft cambian sus versiones en este lapso de tiempo debido a los avances tecnológicos. También se considera que las calculadoras 48 seguirán a la venta como una opción más económica. Esto es común en la introducción de las calculadoras Hewlett Packard al mercado, el nuevo modelo siempre tiene adaptaciones del modelo anterior.

Podría pensarse en la programación de PCs para estos problemas, pero el gran tamaño impide cargarlas a un salón de clases, y las Laptops son muy caras aún. Por otro lado una PC o Laptop requiere de Software que es muy costoso y generalmente menos estable que el software de una calculadora.

Todos los equipos de cómputo se componen de Hardware que son las partes físicas de las computadoras, y de software, que son los programas. En el caso de las calculadoras, el software principal ya viene incluido en el hardware. En un lenguaje similar al C, los programas no requieren declaración de variables. El software reconoce el tipo de objeto a manipular, gracias a los llamados delimitadores.

El software principal, se encuentra residente en la memoria ROM (memoria sólo de lectura) de la calculadora e incluye herramientas para el manejo matemático de diferentes objetos como lo son las matrices, los vectores, las ecuaciones, los gráficos y varios más. Sin embargo se cuenta con otra parte de la memoria RAM (*Random Access Memory;* Memoria de acceso aleatorio) que es donde residen los programas que le agregamos a la calculadora; si se pierde el suministro de la batería por un largo tiempo, esta memoria se borra, además de que se cuenta con una cantidad limitada (32 kb para la 48G y 128 kb para la 48GX), ya que una PC/Laptop tiene actualmente alrededor de 64 MB en RAM.

Es por ello que hay que aprovechar al máximo la memoria ROM para así utilizar un mínimo de la memoria RAM. Así pues uno de los objetivos en los programas usados es usar un mínimo de memoria, para que los programas ocupen poco espacio en RAM, que depende de la optimización de los programas.

Por otro lado los programas que se generan en las calculadoras Hewlett Packard, sirven de módulos utilizables en otros programas de la misma forma que se utilizan las subrutinas en otros tipos de lenguaje de programación.

Esto le permite a los usuarios modificar los programas para obtener nuevas aplicaciones, y así personalizar su calculadora.

Aunque las calculadoras Hewlett Packard manejan lenguaje de máquina, éste no se utilizará, ya que requiere de conocimientos relativamente complejos, y que resultan en la generación de programas poco estables. Un error en el

lenguaje de máquina puede resultar en la volatilización (pérdida) de la memoria RAM. Esto es similar a lo que se pretende realizar con el sistema operativo Linux; el usuario puede abrir el código fuente para corregir errores, mientras que esto en Windows está estrictamente prohibido. Otra ventaja que tienen estas calculadoras es la posibilidad de respaldar en disco la información contenida. Esto se logra a través de un cable y un pequeño programa para la PC.

El objetivo a lo largo de esta tesis es ver como algunas aplicaciones sencillas en calculadoras Hewlett Packard pueden resolver problemas relativamente complicados. También se trata de invitar a los estudiantes que posean una HP a que realicen sus propios programas.

CAPITULO 1

Estudio del entorno

Primeramente antes de comenzar a programar, es importante hacer una especie de "estudio de mercado", para encontrar el sentido de hacer una tesis como ésta.

Para ello se levantó una encuesta de calculadoras en grupos representativos de la facultad de química. El formato de la encuesta es el siguiente:

Nombre:_____
Carrera:_____
Marca y modelo de calculadora _____
¿Sabes programarla?S/N__

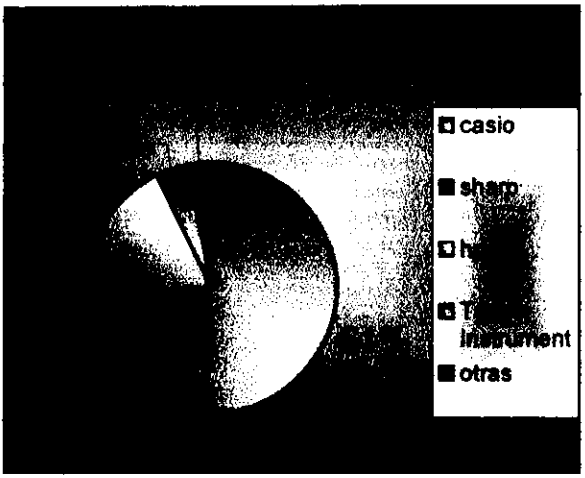
Y el tamaño de la muestra se definió de la siguiente forma:
Confianza 0.05
Probabilidad 0.90
P=0.5
Coeficiente de confianza $1-\alpha=0.90$ $\alpha=0.10$
$\alpha/2= 0.05$ $Z\alpha/2= 1.645$

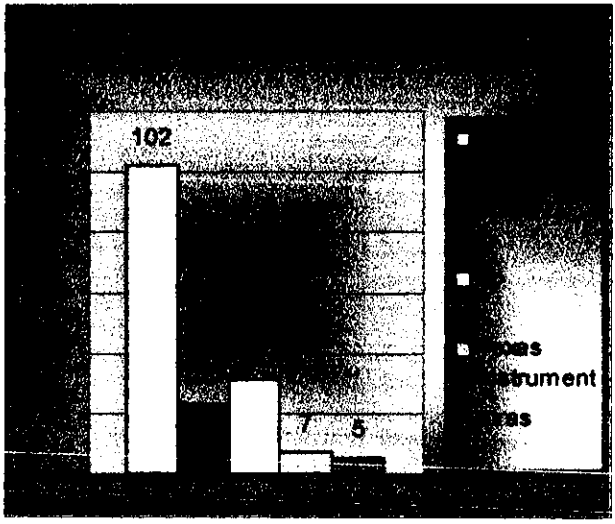
$1.645\hat{}_p=0.4$ $\qquad 1.645\sqrt{\frac{pq}{n}} = 0.4$

$1.645\sqrt{\frac{(.5)(.05)}{n}} = 0.4$ $\qquad$ de donde n = .423

Correspondiendo a 423 elementos para tener un margen de error del 5%

En las encuestas que se realizaron al azar en 423 alumnos de la facultad se reflejan datos importantes como se muestran a continuación:





La marca más común es la Casio. Esta marca es líder en su clase, ya que es económica y sencilla. Además la cantidad de modelos que presenta es muy variado. Algunos modelos son

programables con instrucciones sencillas. Llama la atención que Casio fue la primera marca de calculadoras en incorporar una pantalla de 2 colores, (naranja y verde). Todo esto hace que la Casio sea la calculadora más popular en la Facultad de Química. Sin embargo existen muchos modelos para esta marca, y cada modelo tiene su forma particular de programarse. Las posibilidades de programar en las calculadoras Casio sólo permiten hacer cálculos numéricos por lo que son más limitadas que programar una HP. Hay que agregar el hecho de que la información entre calculadoras no se puede transmitir. En el caso de la información se tendría que teclear en cada calculadora. No se pueden programar los gráficos que son una intención para los intereses de esta tesis. Casio es una buena opción, pero no para la Ingeniería Química.

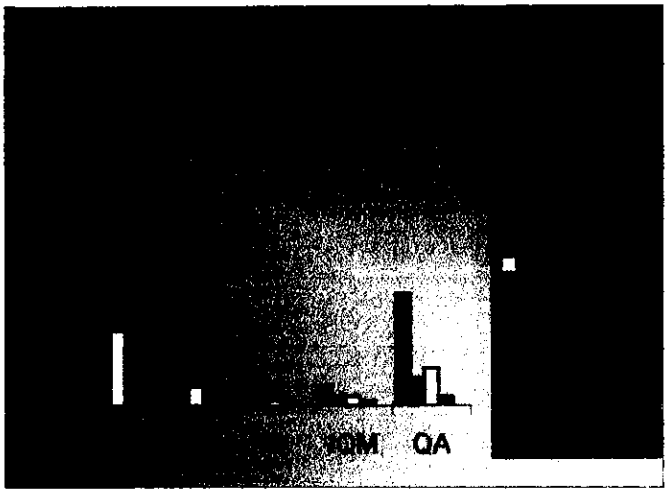
La calculadora marca Sharp es sin duda un producto versátil, pero no tan económico como las calculadoras Casio. Su popularidad es poco menos de la cuarta parte de las Casio. El lenguaje de programación, no es tan simple como el de las calculadoras Casio. Esta marca presenta todas las limitantes de una Casio.

La calculadora Hewlett Packard es una calculadora costosa, complicada, pero al mismo tiempo versátil y poderosa. Todas son programables. Es la que tiene poco menos de la tercera parte de la popularidad que la Casio y es el segundo lugar en preferencia. Con su rayo infrarrojo se pueden compartir datos y programas con gran facilidad para dueños del mismo modelo 48G ó 48GX. Sin embargo su popularidad es casi la tercera parte de la Casio, siendo solamente 2 modelos los utilizados. Una gran diferencia con respecto a Casio, es que

los diseños de HP usan el mismo lenguaje de programación RPN (Reverse Polish Notation). Esta notación polaca inversa permite realizar cálculos más rápidos que la notación algebraica. Para los propósitos de esta tesis es la marca que más conviene, ya que permite programar gráficos, cadenas de textos, integrales y diferenciales en cálculo infinitesimal y obviamente los cálculos numéricos. Otra ventaja que presentan Las HP es el mecanismo de intercambio de información vía infrarrojo. Así se teclea la información en una calculadora y puede pasarse la información de una calculadora a otra.

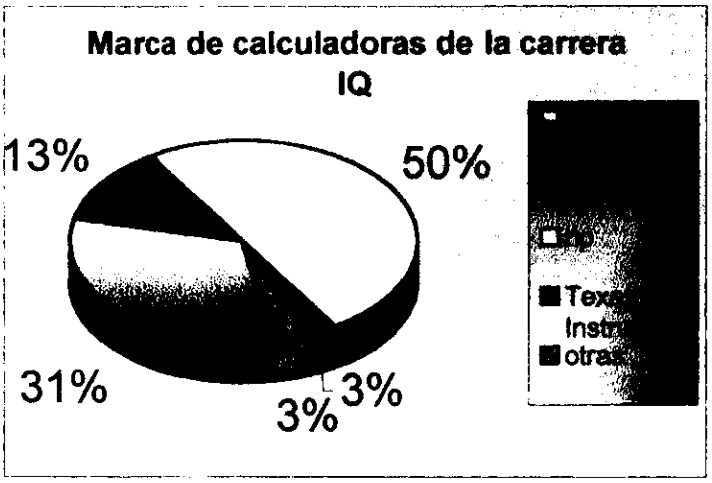
La calculadora Texas Instrument es una calculadora costosa y poderosa. Es tecnológicamente la competencia más fuerte que tiene HP; su costo es menor y la capacidad es la más similar. Todas son programables Esta marca ha presentado un híbrido de calculadora con laptop; el modelo TI-92. Este instrumento contiene aplicaciones gráficas, algebraicas, cálculo y demás. Tiene una pantalla más grande que las HP y de mejor resolución, lo que da una idea panorámica de las operaciones que se realizan. Tiene una sección programable, pero utiliza un lenguaje no estructurado y algo complicado. Se observa a primera vista que se requieren más pasos de programación que en una HP. Los resultados que se obtengan son algo más difíciles de obtener, ya que HP tiene mayor variedad de comandos. Su capacidad en punto flotante es mucho mayor y el rango de números que puede representar es más grande que en una HP. Sin embargo, pese a su excelente calidad, es una calculadora esporádica en la Facultad. Además la marca desapareció del mercado recientemente.

También existen algunos modelos de otras marcas, pero constituyen la minoría de los casos como son las marcas Cedar, Printaform y Olivetti. Estas marcas son tan impopulares que no interesa realizar una tesis para su programación.



Como se puede observar, la carrera de Químico Fármaco Biólogo cuenta con el mayor número de calculadoras y más variado tipo de marcas, donde predominan las calculadoras Casio y Sharp. La carrera que le sigue en número es Química de Alimentos. En tercer lugar está Ingeniería Química.

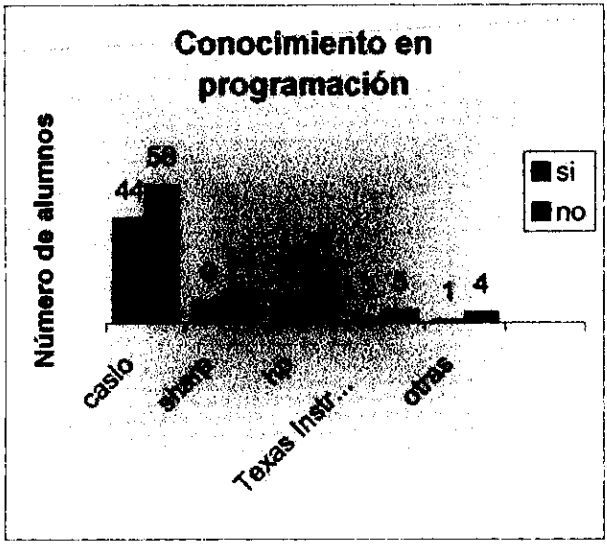
Como esta última es la carrera que más nos interesa para el desarrollo de esta tesis, haremos el análisis en otra gráfica. Se puede observar un comportamiento muy diferente a las



otras carreras, ya que aquí HP domina el mercado. Pero sin duda alguna, es la capacidad la que convence a los estudiantes de Ingeniería Química por encima del precio.

Las calculadoras Casio tienen casi la tercera parte de la población, mientras que las Sharp constituyen el bloque restante junto a la Texas Instrument y las otras marcas. Es importante poner en claro que el dominio de la HP es de la mitad de la población.

Pero también es importante averiguar si los dueños de las calculadoras saben usarlas.



Aquí podemos observar claramente como la mayoría de la gente que maneja cualquier marca de calculadoras afirma no conocer como programarlas. En el caso particular de la HP, apenas la tercera parte de los usuarios es la que conoce como programar. El manual de las Hewlett Packard no es muy didáctico para aprender a programar, tal vez esa es la razón de que muchos estudiantes no se motiven a aprender su programación. La asistencia al curso que ser impartió de "Programación de calculadoras", demostró que sí existe interés en aprender a utilizar estos instrumentos. Debido al curso de programación de computadoras que se imparte en los primeros semestres de la carrera, se sabe que los alumnos saben programar en otros medios. Esto facilita el aprendizaje de la estructuración de los programas.

Si bien la tendencia de las Laptops es penetrar en cualquier lugar, sigue siendo muy costoso, además de ocupar mucho espacio en un pupitre de salón, donde cabe la hoja de examen o el computador, pero no las dos cosas simultáneas.

La visión que ofrece una Laptop es mejor que cualquier calculadora debido al tamaño de su pantalla, su resolución y la tendencia a ser de color. Se trata de llevar el procesador más potente en voga al minitamaño del portafolio. La posibilidad de albergar software de mayor capacidad, es muy atractiva, pero hay que recordar que es más costoso y tiene mayor facilidad de contener errores que pasmen al sistema. La TI-92 es una muy buena competencia a estos costosos computadores.

Conclusión:

Se tiene un mercado amplio de aproximadamente el 34 % de estudiantes que poseen una calculadora HP en la carrera de Ingeniería Química y que no saben como programarla para realizar una tesis que indique la programación de algunas rutinas que permitan resolver problemas relacionados con la carrera. La idea es hacer programas pequeños, que estructurados en programas generales permitan ejecutar acciones complicadas. Es posible hacer rutinas que sirvan de ejemplo a estudiantes para hacer modificaciones que les permitan resolver sus propias necesidades. Pero también podemos extender estas ventajas a las otras carreras que también presentan el uso de estas calculadoras.

Capítulo 2

Filosofía básica de programación en HP.

Las calculadoras Hewlett Packard siempre se han distinguido por utilizar una pila de niveles. En un principio se trató de los niveles *t*, *x*, *y* y *z*, constituyendo un total de cuatro niveles. A partir de los modelos 28, la pila se convirtió en multinivel (más de cuatro). Esto permite almacenar objetos en pila con la única limitante de la cantidad de memoria que puede operar la máquina. Los objetos son las entidades básicas de la calculadora, que se crean para formular problemas y manipular expresiones a fin de hallar resultados.

La pila es una secuencia de niveles numerados, cada uno de los cuales contiene un objeto. Los objetos entran en la pila en el nivel 1, elevando los que ya había en este nivel a niveles superiores. Los objetos salen también de la pila desde el nivel 1, dejando caer los objetos que había en los niveles superiores hasta el inmediato inferior. Todos los objetos son tratados igual en la pila, es decir, sencillamente como objetos.

Existen órdenes para duplicar, borrar y reordenar los objetos de la pila.

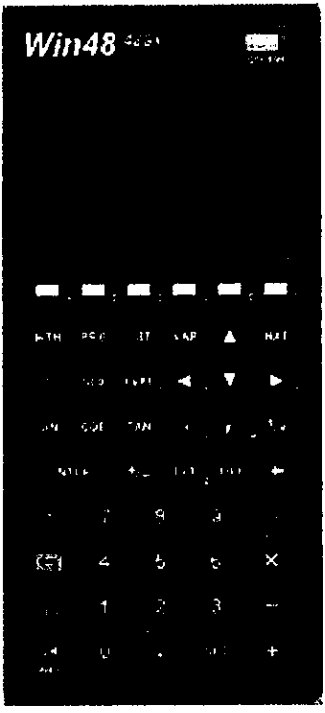
La mayoría de las órdenes toman objetos de la pila (llamados argumentos) como entrada, y devuelven a la pila objetos de salida (llamados resultados). Los argumentos deben estar en la pila antes de que se ejecute la orden. La orden toma los argumentos que necesita de la pila y los sustituye por los

resultados. Por ejemplo, la función SIN toma un valor (un número real o complejo o una expresión algebraica) del nivel 1, calcula el seno y vuelve a poner el resultado en el nivel 1. La función + toma dos valores de la pila que se encuentran en los niveles uno y dos, los suma y pone en la pila el resultado en el nivel uno. Las funciones pueden ser de tipo aritmético (funciones básicas), trascendental que es calculado por series de Taylor (logaritmos y funciones trigonométricas), y algoritmos incorporados (coeficiente de Darcy y distribuciones estadísticas). Los primeros dos tipos pueden ser usados en cálculos algebraicos, mientras que los algoritmos sólo pueden ser usados en la pila y dentro de programas. La jerarquía que siguen las operaciones algebraicas es la misma que se muestra al usar el editor equation.

Este tipo de lógica en la que la orden se sitúa detrás de los argumentos se llama *lógica postfija o RPN* (notación polaca inversa), en memoria del lógico polaco Jan Lukasiewics (1878-1956).

Entrada de objetos

Cuando se pulsa una tecla para empezar a introducir un nuevo objeto, el caracter de la tecla pulsada aparece en la línea de órdenes. Esta línea puede contener cualquier número de objetos representados en forma de texto. Es la línea inferior de la pantalla (inmediatamente encima de las etiquetas de menú). La línea de órdenes aparece también cuando se usa VIEW o EDIT con el fin de visualizar o alterar el contenido de un objeto ya existente.



El contenido de la línea de órdenes se procesa al pulsar ENTER (o cualquier tecla de orden o de función que ejecute automáticamente ENTER como las teclas de menú). El contenido de la línea de órdenes se evalúa como un programa, en cuyo momento desaparece la línea de la pantalla.

La ventaja que tienen las calculadoras Hewlett Packard sobre otras marcas, es el hecho de que maneje objetos. El propósito de los tipos de objetos es ahorrar tiempo de trabajo proporcionando tipos específicos de información. Por ejemplo, hay que imaginar lo que implicaría el uso de números reales para representar vectores y matrices, llevando la cuenta de alguna forma de cada objeto contenido en cada vector o matriz para escribir programas que efectúen cálculos aritméticos. Es más sencillo ingresar los números en

un objeto de tipo matriz–vector el cual se puede manejar como una entidad única y así efectuar cálculos empleando las funciones aritméticas normales.

Además, las calculadoras HP son más eficientes, ya que al basar los tipos múltiples de información, las operaciones simbólicas y la programación en el simple concepto de los tipos de objetos, las calculadoras HP 48G y 48GX reducen al mínimo la cantidad de reglas a recordar. Los objetos se ingresan en la línea de órdenes, se colocan en la pila o se almacenan en variables siempre de la misma forma, sin importar el tipo de objeto del cual se trate.

Estos son los tipos de objetos que se utilizan para programar:

Tipo	Ejemplo	Uso en la tesis
Número real	14.75	Se usa
Número complejo	(8.25,12.1)	Se usa
Cadena	"Hello"	Se usa
Arreglo	[4.8 –1.3 2.1]	Se usa
Unidad	11.5_ft	No se usa
Programa	«DUPN SWAP»	Se usa
Algebraico	'A-B'	Se usa
Lista	{6.8 5 "FIVE"}	Se usa
Comando	FIX	Se usa
Nombre	'Vol'	Se usa

Es posible introducir cualquier número de caracteres en la línea de órdenes o dividir la línea en varias filas pulsando ↵, que intercala un caracter de línea distinta (cambio de línea) en la cadena de la línea de órdenes, en la posición en que se encuentre el cursor. Estos caracteres actúan como

separadores de objetos, pero, por lo demás, se ignoran cuando se evalúa la línea de órdenes.

Los objetos empleados en un programa pueden ser identificados por la calculadora dependiendo del delimitador que tienen. Esto es una ventaja, ya que en los programas no es necesario declarar variables. Para identificar dentro de un programa el tipo de objeto, se usa la instrucción TYPE, la cual regresa un dígito que corresponde al tipo de objeto.

Los objetos también pueden almacenarse en registros mediante un nombre, y pueden visualizarse en el menú VAR. Esto permite llamarlos cuantas veces sea necesario con tan sólo pulsar el botón del menú que le corresponda, o bien invocando su nombre.

La versatilidad de la pila y el buen manejo de los registros permiten realizar mejores programas. La cantidad de registros de memoria que utiliza un programa se ve muy reducida con el manejo de la pila multinivel.

Para manejo de teclas en esta tesis, cuando se trate de pulsar teclas, se enmarcarán entre corchetes. Si es necesario escribir texto, éste se dejará tal cual.

Todo objeto tiene una especie de huella digital, expresada en dos códigos que son únicos para cada objeto. Uno de ellos es un número de base n y el otro es un número normal que representa la cantidad de memoria que ocupa. Cuando se ingresa el nombre de un objeto, el comando BYTES toma del primer nivel el nombre del objeto y regresa el código que es la huella digital, y el tamaño de memoria. Cada programa listado en esta tesis incluirá al inicio los dos códigos. El

lector puede teclear dichos programas en su calculadora y determinar con el comando BYTES los códigos de identificación del programa y compararlos con los reportados en la tesis. **Si los códigos no coinciden** es un indicio de que **los objetos tampoco son iguales** y existe un error que corregir.

Filosofía de programación de las Casio

Para los modelos costosos de Casio se puede decir lo siguiente: las variables en una calculadora Casio son almacenes de letras que admiten números reales que se usan en forma de usuario y en forma de programas. En la forma de usuario se usa para facilitar el acceso a los cálculos. En la forma de programación, es la mejor manera de que la calculadora acceda a los diferentes resultados intermedios y finales. Las calculadoras Casio tienen historial, que es el equivalente a la pila de HP, solo que este no es programable. En el caso de las matrices, estas se encuentran predefinidas, y no se permiten gran cantidad de operaciones. Carecen de listas y cadenas. Su lenguaje de programación es similar al BASIC, y tiene un número de pasos muy limitados. Sin embargo, su capacidad gráfica es muy buena, así como la velocidad a la que corren sus programas. La velocidad de graficación y de ejecución de programas en una Casio es casi el doble que la de una HP, aunque en el caso de los programas en lenguaje de máquina la HP supera este parámetro.

En el caso de las calculadoras sencillas, lo más que realizan los programas son operaciones aritméticas en un modo de

entrada y salida de datos donde no hay ni mensajes ni instrucciones que faciliten el entendimiento del programa.

Filosofía de programación de las Texas Instrument

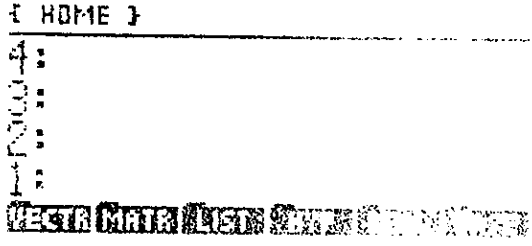
También manejan un historial, hacia lo que apunta será el futuro de todas las calculadoras. Sin embargo al igual que las Casio, éste no es programable. Esta es una seria desventaja comparada con las HP. Los almacenes son letras en las cuales se almacenan números reales, y la ventaja que tienen con respecto a esta semejanza con la Casio, es que el número real que se almacena, puede provenir de una expresión algebraica. Su forma de programar es similar a la que tiene el Fortran, sin embargo el sistema de programación consume gran cantidad de pasos y es poco intuitivo, inclusive algo confuso. Algunas manejan caracteres, lo cual representa una fuerte ventaja que puede competir con las rutinas de la HP. Las capacidades gráficas también son programables, logro anterior de la HP, solo que la HP da opción a mayor número de tipos gráficos. Le hace falta mayor desarrollo en el cálculo diferencial y en los métodos numéricos, ya que no todas tienen estas características. Una desventaja es que no se pueden programar los comandos si no proviene del menú respectivo, ya que lo considera como una operación letra a letra. Esto no sucede ni con Casio ni HP. Otra cuestión es que no se puede compartir la información vía infrarrojo o cable, salvo el caso de la TI-92.

Capítulo 3

Creación de una pequeña base de datos

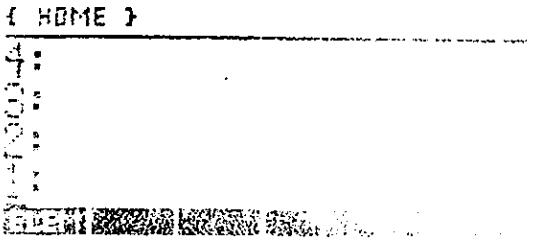
En el presente capítulo se creará una pequeña base de datos para formar una tabla periódica. Después se utilizará la información de esta base para la creación de programas que calculen en forma automática pesos moleculares de diferentes compuestos.

Al encender la calculadora se ve la siguiente imagen si es que aún no se ha ingresado nada en la pila.

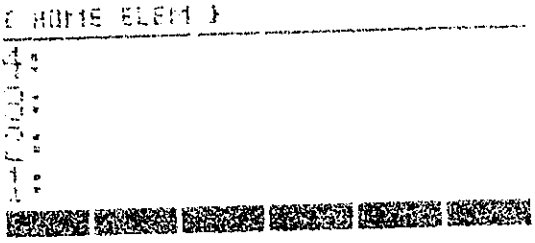


Para estos programas se necesitan los subprogramas EVL y SUS que se encuentran en el apéndice de programas auxiliares.

En primer lugar hay que crear un directorio para tener la base de datos libre de otra información. Para ello teclearemos [α][α] ELEM [ENTER] [α][α] CRDIR.



Aparecerá en el menú un pequeño folder. Oprimimos esa tecla de menú y se entra al directorio y la pantalla tendrá esta imagen:



A continuación hay que teclear los símbolos atómicos en una lista. Esto se consigue abriendo un corchete y dejando un espacio entre símbolo y símbolo. Los símbolos se introducen en el orden de número atómico ascendente.

[# 9377d 567]

{ H He Li Be B C N
O F Ne Na Mg Al Si
P S Cl Ar K Ca Sc
Ti V Cr Mn Fe Co Ni
Cu Zn GBa Ge As Se
Br Kr Rb Sr Y Zr Nb
Mo Tc Ru Rh Pd Ag
Cd In Sn Sb Te I Xe
Cs Ba La Ce Pr Nd
Pm Sm Eu Gd Tb Dy
Ho Er Tm Yb Lu Hf
Ta W Re Os Ir Pt Au
Hg Tl Pb Bi Po At
Rn Fr Ra Ac Th Pa U
Np Pu Am Cm Bk Cf

Es Fm Md No Lw Unq
Unp Unh Uns Uno Une
}

Al finalizar, se tecleará [ENTER] [<¬] [EDIT] y aparecerá como la lista anterior; se teclea nuevamente [ENTER]. Después de ello se teclea el corchete solo ['] seguido de la tecla alfa dos veces [α][α], luego SIMB que es el nombre que le daremos, [ENTER] y [STO]. Para verificar que esté bien tecleado, se escribe el corchete solo ['] seguido de la tecla alfa dos veces [α][α], luego SIMB, [ENTER]. Luego la tecla alfa dos veces [α][α], luego BYTES, [ENTER] y los resultados deberán ser # 9377d y 567, que son los números que están al inicio de las listas. Este procedimiento debe seguirse para el resto de los programas y listas. Si en lugar de aparecer una *d* aparece una *h*, *o* o *b*, hay que seguir la ruta MTH, BASE y DEC para establecer el modo decimal en la calculadora.

Ahora se introduce la lista de pesos atómicos conservando el mismo orden, esto es muy importante, ya que ahorra introducir una lista del orden de lectura de datos

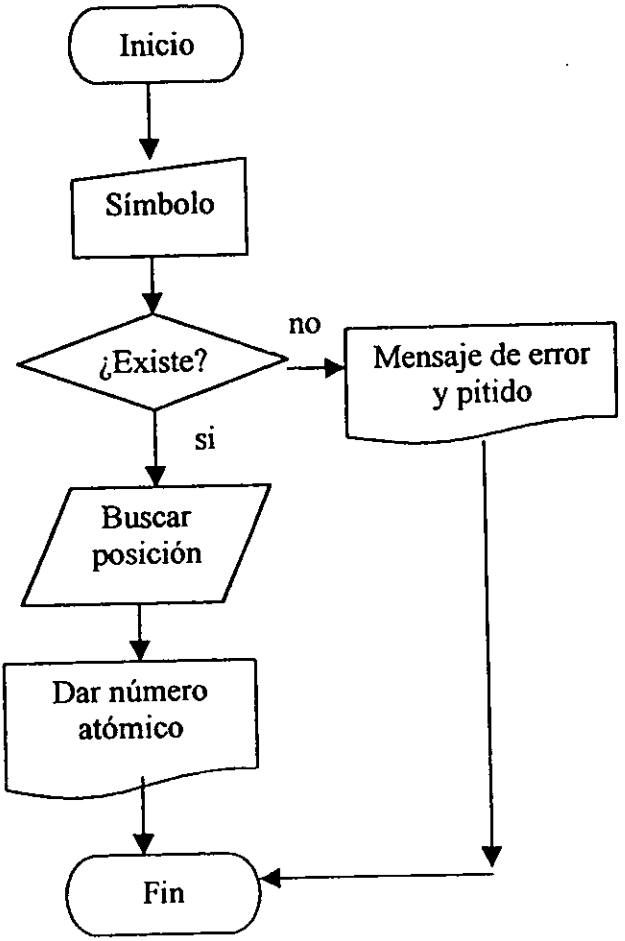
[# 6348d 1093]

{ 1.0079 4.0026
6.939 9.0122 10.811
12.011 14.006
15.999 18.998
20.163 22.989
24.312 26.98 28.086

30.973 32.064
35.453 39.948
39.102 40.08 44.956
47.9 50.942 51.966
54.938 55.847
58.933 58.71 63.54
65.37 69.72 72.59
74.922 78.96 79.909
83.8 85.47 87.62
88.905 91.22 92.906
95.94 98 101.07
102.9 106.4 107.87
112.4 114.82 118.69
121.75 127.6 126.9
131.3 132.9 137.34
138.91 140.12 140.9
144.24 147 150.35
151.96 157.25
158.92 162.5 164.93
167.26 168.93
173.04 174.97
178.49 180.94
183.85 186.2 190.2
192.2 195.09 196.96
200.59 204.37
207.19 208.98 210
210 222 223 226 227
232.03 231 238.03
237 242 243 247 247
249 254 253 256 254
257 }

Al finalizar, se teclea [ENTER] [<¬] [EDIT] y aparece la lista anterior; se teclea nuevamente [ENTER] Después de ello se teclea el corchete solo ['] seguido de la tecla alfa dos veces [α][α], luego PM, [ENTER] y [STO].

A continuación se crea primer programa de esta tesis:
Programa: NA (Número Atómico): Este programa regresa el número atómico del símbolo que se introduzca.



NA: La columna izquierda indica el texto que debe ser tecleado, la columna derecha indica lo que hace cada instrucción y esto corresponde al diagrama de flujo anterior. Al momento de ingresar el símbolo "«" la calculadora automáticamente se pone en modo de edición/codificación de programas.

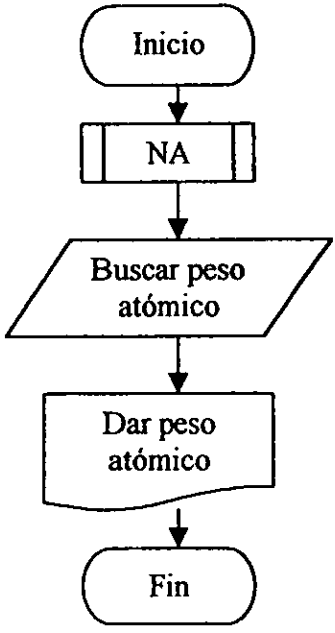
[# 29484d 105.5]

Programa	***Descripción***
« SIMB SWAP POS DUP	Llama la lista de símbolos, ubica la posición y realiza una copia
0 ==	Evalúa si es igual a cero
« 450 2 BEEP	Inicia la subrutina, pone la frecuencia y duración del pitido y hace el sonido
"No existe este elemento"	Texto de error
MSGBOX DROP KILL	Muestra el texto de error en la pantalla, tira el cero y detiene el programa totalmente.
» IFT	Termina la subrutina y ejecuta si la condición es correcta
»	Termina el programa

Al finalizar, se tecleará [ENTER] [<¬] [EDIT] y aparecerá como la lista anterior; se teclea nuevamente [ENTER]. Se almacena el programa en la variable 'NA'; se teclea el

corchete solo [′] seguido de la tecla alfa dos veces [α][α], luego NA, [ENTER] y [STO].

Programa MA (Masa Atómica): Devuelve la masa atómica del símbolo indicado.



[# 46173d 32.5]

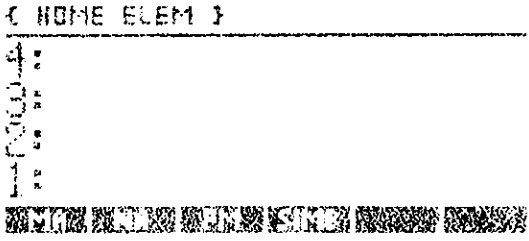
Programa	Descripción
« NA PM SWAP GET	Inicia el programa. Busca el número atómico, llama la lista de pesos atómicos y obtiene la posición específica.
»	Termina el programa

Al finalizar, se teclea [ENTER] [<¬] [EDIT] y aparece como la lista anterior; se teclea nuevamente [ENTER] Se almacena el programa en la variable 'MA'; se teclea el corchete solo ['] seguido de la tecla alfa dos veces [α][α], luego MA, [ENTER] y [STO].

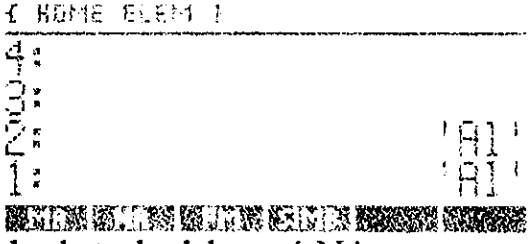
Con esto se ha conseguido crear una base de datos que incluye: símbolo, número atómico y peso atómico.

Ahora se dará un ejemplo del funcionamiento del programa:

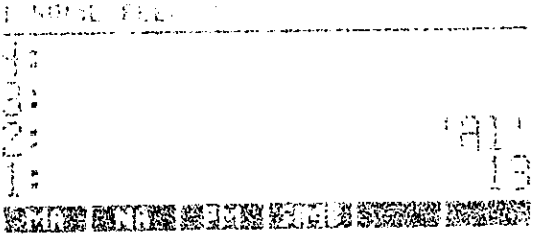
Se tiene la pantalla como sigue;



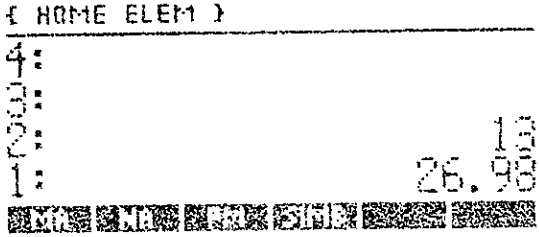
Se teclea algún símbolo químico. Por ejemplo Al. [α][α] A[<¬]L[ENTER] [ENTER] y aparece la siguiente imagen;



Se pulsa la tecla del menú NA y aparece el número atómico.



Se pulsa la tecla [SWAP] MA y aparecerá la siguiente imagen;

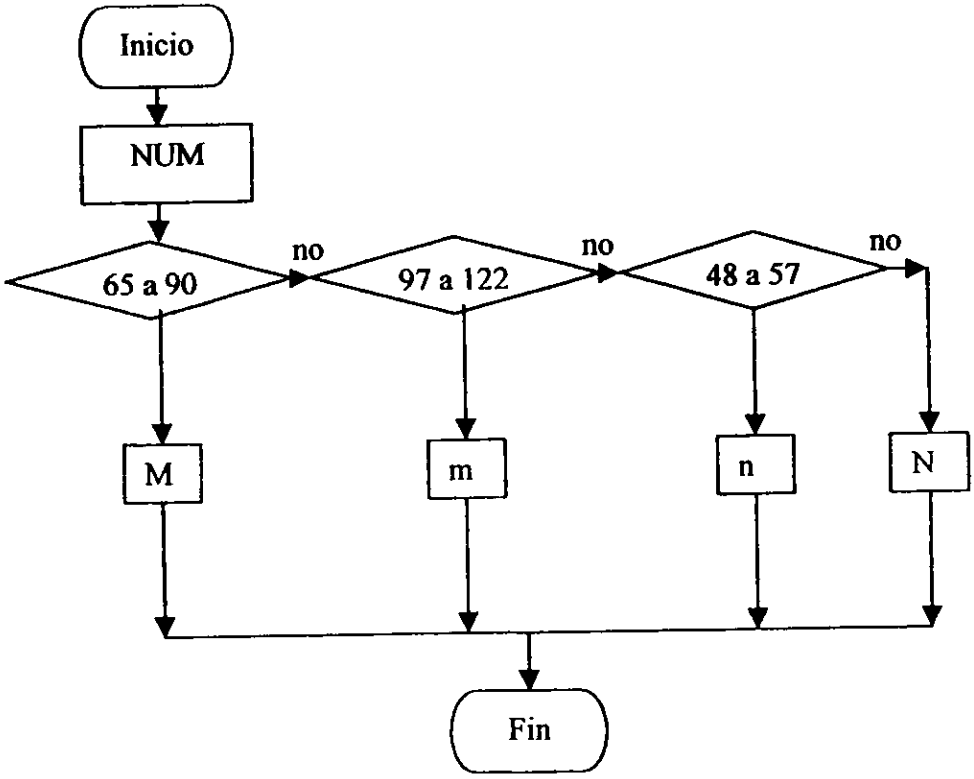


Como se puede ver estos números corresponden al número y a la masa atómica del aluminio.

Si se pide un elemento que no existe, se despliega un mensaje de error y un pitido.

Ahora vamos a listar los programas necesarios para el cálculo de los pesos moleculares. Una vez más se recuerda que se requieren los subprogramas EVL y SUS que se encuentran en los anexos al final de la tesis para poder correr estas rutinas.

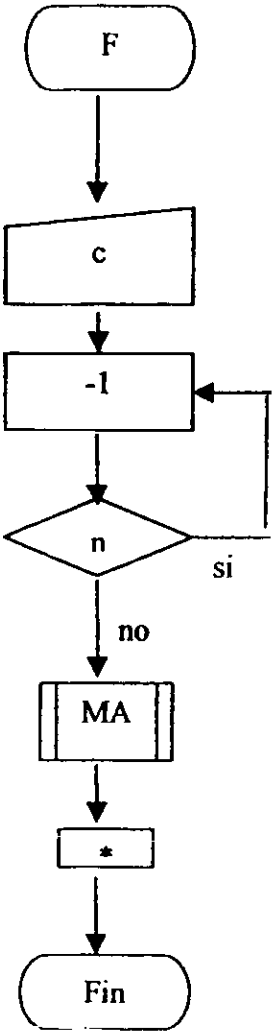
Programa TIPO: Este programa devuelve una M si se encuentra un caracter mayúscula, una m si es minúscula, una n para número y N para los demás caracteres. Por ejemplo: si en el nivel 1 hay un caracter "P" devuelve una 'M' indicando que es una mayúscula. Esto se requiere para que



[# 43215d 196]

Programa	*Descripción*
« NUM → a	Inicia el programa. Da el número de cadena y lo almacena en a
« a 65 ≥ a 90 ≤	Inicia la rutina. Verifica el intervalo 65 a 90
AND M	Unifica criterios, y pone el objeto M
« a 97 ≥ a 122	Inicia la rutina. Verifica el intervalo 97 a 122
≤ AND m	Unifica criterios y pone el objeto m
« a 48 ≥ a 57	Inicia la rutina. Verifica el intervalo 48 a 57
≤ AND n N IFTE	Unifica criterios y pone los objetos n y N ademas de evaluar la condición
» IFTE	Evalúa la condición
» IFTE	Evalúa la condición
»	Termina la rutina
»	Termina el programa

Programa F: Descompone la cadena más sencilla en símbolo y número, obtiene el peso atómico y lo multiplica. Por ejemplo: Si en el primer nivel existe "H2", devuelve un 2.

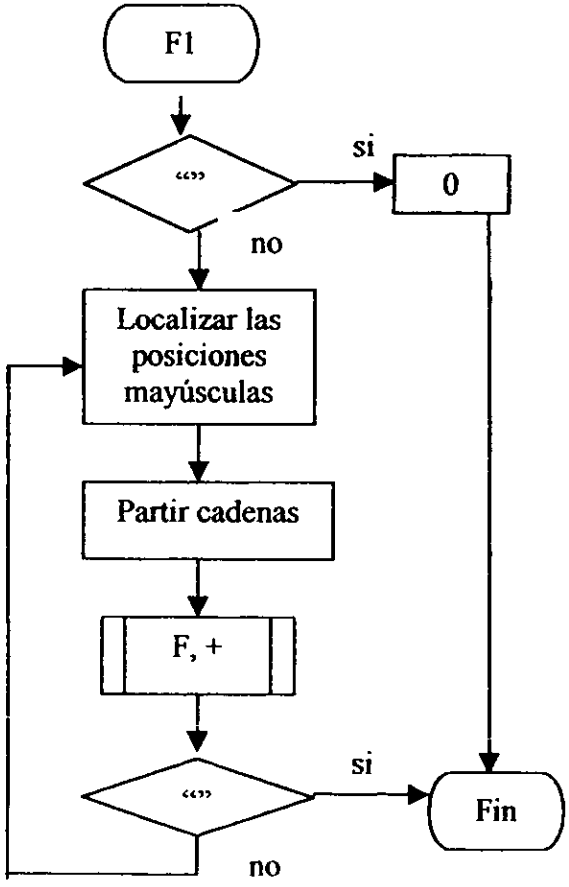


[# 10915 140]

Programa	*Descripción*
« → c	Inicia el programa. Almacena la cadena en c
« c SIZE	Inicia la subrutina y calcula el tamaño de la cadena.
DO 1 -	Empieza el ciclo condicional y resta uno de la longitud de la cadena
UNTIL c OVER	Comienza la evaluación de la condición
DUP SUB TIPO 'n'	Evalúa si el caracter es de tipo numérico
SAME NOT	Da el carácter negativo a la condición anterior.
END DUP c SWAP	Termina el ciclo condicional, duplica, llama a l a cadena y lo voltea.
1 + c SIZE SUB STR→	Toma la parte de la cadena que es numérica y lo transfiere a número
SWAP c 1 ROT SUB	Toma la parte alfabética de la cadena.
STR→ MA *	Convierte la cadena en símbolo, llama al programa MA para calcular el peso molecular, lo multiplica por la parte numérica
»	Termina la subrutina
»	Termina el programa

Una vez tecleado el programa se almacena; [α]F[STO]

Programa F1: obtiene el peso molecular de una cadena sin paréntesis, corchetes ni dobles ligaduras, etc.



[# 13921d 271]

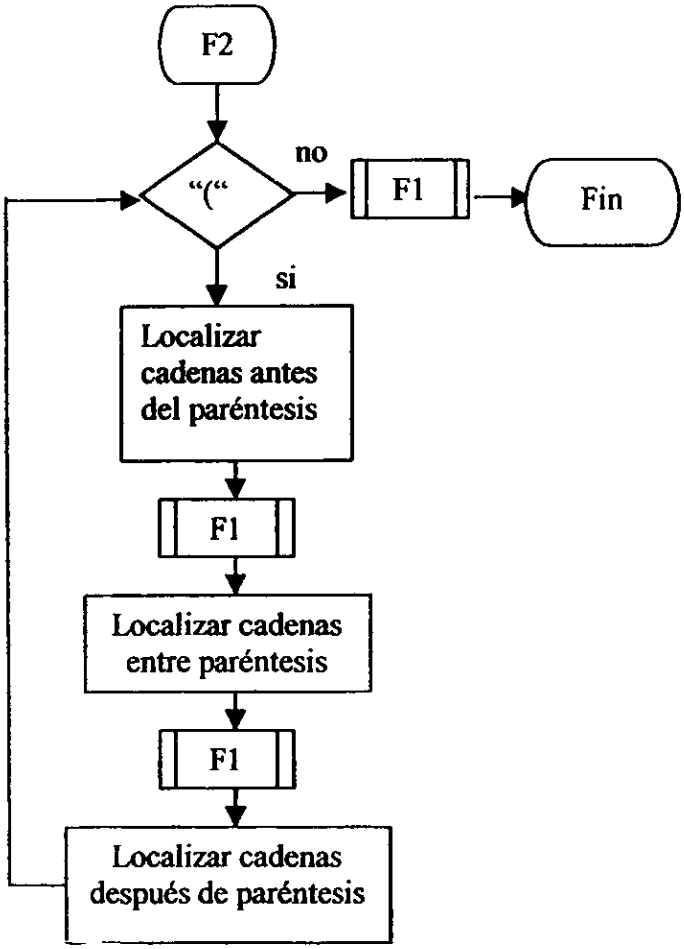
Programa	*Descripción*
« DUP "" SAME	Inicia el programa, hace una réplica y averigua si es una cadena vacía
« DROP 0	Inicia la subrutina tira la cadena vacía y la sustituye por un cero
»	Termina la subrutina
« DUP DUP SIZE	Inicia la subrutina hace dos réplicas y toma el tamaño
DUP SUB TIPO 'n'	Averigua si la posición específica es un número
SAME NOT	Da el carácter negativo a la condición
« "1" +	Inicia una subrutina. Suma un uno a la cadena
» IFT → c	Termina la subrutina. Ejecuta la subrutina en caso de cumplirse la condición de no ser un número el último caracter de la cadena.
« c SIZE	Invoca la cadena y toma su tamaño

DO 1 -	Inicia el ciclo que resta uno
UNTIL c OVER	Termina el ciclo e inicia la condición invocando a la cadena y haciendo una copia del número de posición
DUP SUB TIPO 'M'	Duplica el número para sacar la subcadena e invoca el programa TIPO, también coloca el objeto M
SAME	Averigua si la posición de la cadena es una mayúscula
END DUP c	Termina el ciclo condicional duplica la posición e invoca la cadena
SWAP c SIZE SUB	Voltea la cadena, obtiene el tamaño de la cadena y la corta en una nueva subcadena.
SWAP c 1 ROT 1 –	Voltea la subcadena, invoca a la cadena original, coloca un uno, gira los tres últimos objetos y le resta uno
SUB SWAP F SWAP DUP	Obtiene la subcadena, la voltea, ejecuta el programa F, voltea el resultado y hace una réplica
SIZE	Toma el tamaño de la cadena
« F1 +	Inicia la subrutina condicional , ejecuta el programa actual y suma el resultado
»	Termina la subrutina

	condicional
« DROP	Inicia la subrutina condicional y tira el resultado
» IFTE	Cierra la subrutina condicional y evalúa la subrutina correcta
»	Termina la subrutina
» IFTE	Termina la subrutina condicional y evalúa la subrutina correcta
»	Termina el programa

Una vez tecleado el programa se almacena; [α][α]F1[α][STO]

Programa F2: Obtiene el peso molecular de cadenas con paréntesis.



[# 35782d 315]

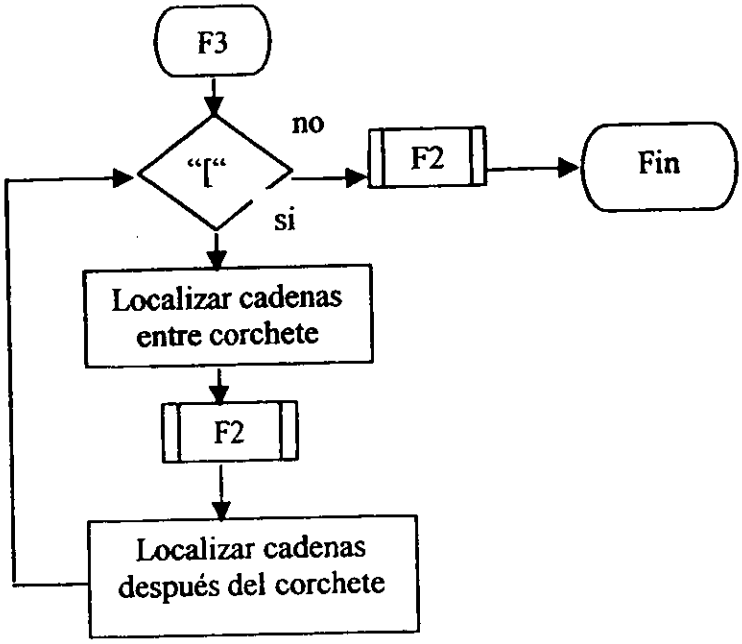
Programa	*Descripción*
« DUP "(" POS	Inicia el programa, hace una réplica, averigua si existe un paréntesis en la cadena
« 0 SWAP	Inicia la subrutina condicional, coloca un cero y lo voltea
DO DUP DUP "("	Inicia el ciclo condicional, hace 2 réplicas y coloca un paréntesis
POS 1 SWAP 1 - SUB	Ubica el paréntesis, coloca un uno, lo voltea, le resta un uno y obtiene la subcadena
F1 ROT + SWAP DUP	Ejecuta el programa F1, gira los tres últimos objetos, suma el resultado, gira el valor y duplica la cadena
"(" POS OVER SIZE	Ubica la posición del paréntesis, hace una copia del segundo nivel y toma el tamaño
SUB DUP DUP ")" POS	Obtiene la subcadena, hace dos réplicas y ubica el cierre del paréntesis
2 SWAP 1 - SUB F1	Coloca un 2, lo gira, resta un uno, obtiene la subcadena y aplica el programa F1

SWAP DUP ")" POS 1	Voltea el resultado, hace una réplica, localiza la posición del cierre del paréntesis y coloca un uno
+	Suma el uno
WHILE DUP2	Inicia el ciclo condicional, hace una réplica de los dos últimos valores
DUP SUB TIPO 'n'	Hace una réplica, obtiene el caracter, aplica el programa TIPO y coloca el objeto n
SAME	Averigua si es del tipo n
REPEAT 1 +	Inicia las operaciones del ciclo condicional y suma uno
END OVER ")"	Termina el ciclo condicional hace una copia del segundo nivel y coloca el cierre del paréntesis
POS 1 + SWAP 3 DUPN	Ubica el cierre del paréntesis, suma uno, voltea el resultado y hace una réplica de los tres últimos niveles
1 - SUB STR→ 5 ROLL	Resta uno, obtiene la subcadena, la convierte a número y gira desde el nivel 5
* 5 ROLL + 4 ROLLD	Multiplica los valores, gira desde el nivel 5, suma los valores y gira hacia el nivel 4 para preservar la contribución

SWAP DROP OVER SIZE	Voltea el objeto, tira el primer nivel, hace una copia del segundo nivel y mide el tamaño de la cadena.
SUB	Obtiene la subcadena
UNTIL DUP "("	Inicia las condiciones del ciclo condicional, hace una réplica y coloca un paréntesis
POS 0 ==	Averigua si no existe un paréntesis en la cadena
END F1 +	Termina el ciclo condicional, aplica el programa F1 y suma el resultado
»	Termina la subrutina condicional
« F1	Inicia la subrutina condicional y aplica el programa F1
» IFTE	Cierra la subrutina condicional y evalúa la subrutina correcta
»	Termina el programa

Una vez tecleado el programa se almacena; [α][α]F2[α][STO]

Programa F3: calcula el peso molecular de una cadena con corchetes.



[# 59809d 240]

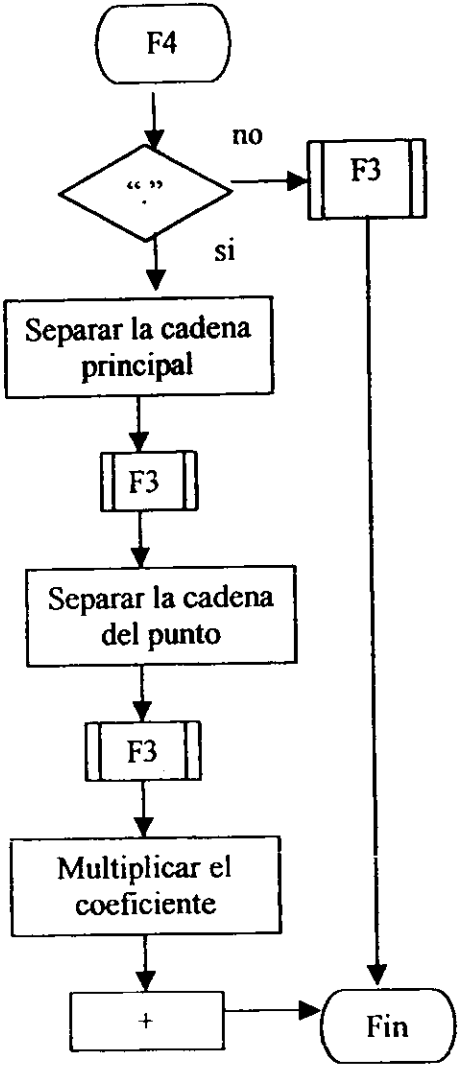
Programa	*Descripción*
« DUP "[" POS	Inicia el programa, hace una réplica y averigua si existe un corchete en la cadena
« DUP 1 OVER "["	Inicia la subrutina condicional, hace una réplica, coloca un uno y vuelve a hacer una réplica de la cadena, coloca un corchete
POS 1 - SUB F2 SWAP	Localiza la posición del corchete, le resta un uno, obtiene la subcadena, aplica el programa F2 y voltea el resultado
DUP DUP "[" POS 1 +	Hace dos réplicas, ubica la posición del corchete y le suma uno
SWAP "]" POS 1 - 3	Voltea el valor, ubica la posición de cierre del corchete, le resta uno y coloca un tres
DUPN SUB F2 4 ROLLD	Hace la copia de los tres últimos niveles, obtiene la subcadena, le aplica el programa F2 y lleva el resultado hacia el nivel 4

2 + SWAP DROP OVER	Suma 2, voltea el valor, tira el último resultado y hace una copia de la cadena
SIZE SUB DUP "["	Obtiene el tamaño de la cadena, saca la subcadena y coloca un corchete
POS	Averigua si existe un corchete en la cadena
« DUP 2 SWAP	Inicia la subrutina condicional, hace una réplica, coloca un 2 y lo voltea
"]" POS 1 - SUB F2	Ubica la posición de cierre del corchete, le resta un uno, obtiene la subcadena y le aplica el programa F2
»	Cierra la subrutina condicional
« F2	Inicia la subrutina condicional y aplica el programa F2
» IFTE + +	Cierra la subrutina condicional, evalúa la subrutina correcta y suma los valores dos veces
»	Cierra la subrutina condicional
« F2	Inicia la subrutina condicional y evalúa el programa F2

» IFTE	Cierra la subrutina y evalúa la subrutina correcta
»	Termina el programa

Una vez tecleado el programa se almacena; [α][α]F3[α][STO]

Programa F4: Calcula el peso molecular de cadenas con puntos. Por ejemplo: en el caso de una molécula pentahidratada de ácido sulfúrico, $H_2SO_4.5H_2O$. Devuelve la contribución del ácido de 98.0758 y del agua 90.074 y los suma.



[# 59445d 238]

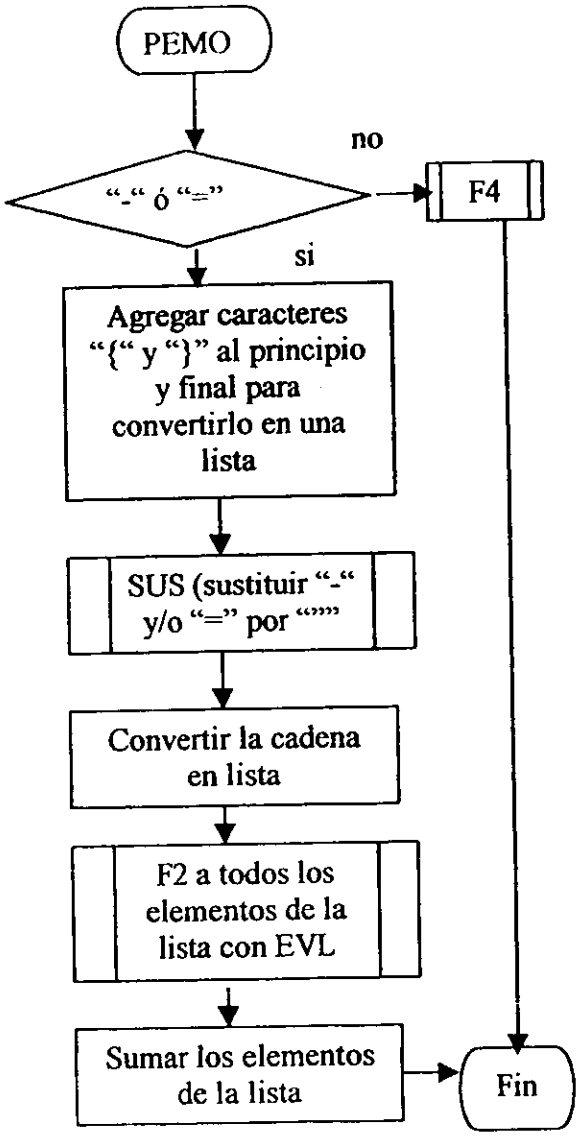
Programa	*Descripción*
« DUP "." POS	Inicia el programa, hace una réplica y averigua si existe un punto en la cadena
« DUP "." POS 1	Inicia la subrutina condicional, hace una réplica, ubica la posición del punto y coloca un uno
SWAP 3 DUPN 1 - SUB	Voltea el uno, hace una réplica de los tres últimos objetos, resta un uno y obtiene la subcadena
F3 4 ROLLD 1 + SWAP	Aplica el programa F3, lleva el resultado hasta el nivel 4 para despejara los niveles y preservar este valor, suma un uno y voltea este objeto
DROP OVER SIZE SUB	Tira el objeto, hace una copia del segundo nivel, obtiene el tamaño y la subcadena
1	Coloca un uno como origen
WHILE DUP2 DUP	Inicia el ciclo condicional, hace una réplica de los dos últimos objetos y vuelve a hacer una réplica

SUB TIPO 'n' SAME	Obtiene el caracter y le aplica el programa TIPO, luego averigua si es tipo n
REPEAT 1 +	Inicia las instrucciones del ciclo condicional, suma un 1
END DUP2 OVER	Termina las instrucciones condicionales, hace una réplica de los dos últimos objetos y hace una réplica del segundo nivel
SIZE SUB F3 3 ROLLD	Mide el tamaño de la cadena, obtiene la subcadena, le aplica el programa F3 y lleva el resultado hacia el nivel 3
1 SWAP 1 - SUB DUP	Coloca un uno, voltea el uno y le resta uno, obtiene la subcadena y hace una réplica
"" ==	Coloca un caracter nulo y averigua si es igual a la réplica
« DROP "1"	Inicia la subrutina condicional tira el caracter nulo y coloca un caracter uno
» IFT STR→ * +	Termina la subrutina condicional, la evalúa en caso de ser cierta, transforma el carácter en número, lo multiplica y lo suma

»	Termina la subrutina condicional
« F3	Inicia la subrutina condicional y evalúa el programa F3
» IFTE	Termina la subrutina condicional y evalúa la subrutina correcta
»	Termina el programa

Una vez tecleado el programa se almacena; [α][α]F4[α][STO]

Programa PEMO: Calcula el peso molecular de cadenas con ligaduras simples, dobles y triples.



[# 17112d 282]

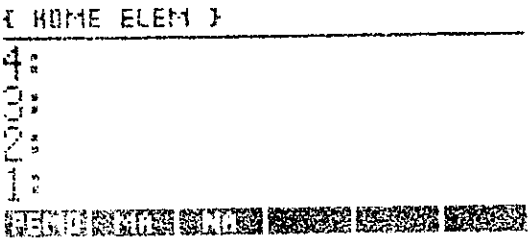
Programa	*Descripción*
« DUP "-" POS OVER	Inicia el programa, hace una réplica, averigua si la cadena tiene un guión, y hace una copia del segundo nivel
"=" POS OR	Averigua si tiene un signo igual y unifica criterios
« "{" 34 CHR +	Inicia la subrutina condicional, coloca un inicio de lista y coloca el carácter de comillas, junta estos caracteres
SWAP 34 CHR "}" + +	Voltea la cadena, agrega otras comillas, y un corchete de cierre de lista, junta todos los caracteres
+	Junta los caracteres
WHILE DUP "-"	Inicia el ciclo condicional, hace una réplica y coloca el guión
POS	Ubica la posición del guión
REPEAT DUP "-"	Inicia las actividades del ciclo condicional, hace una réplica y coloca un guión
POS 34 CHR DUP +	Ubica la posición y le coloca 2 caracteres de comillas
SWAP SUS	Voltea la cadena y lo sustituye con el módulo SUS
END	Termina el ciclo condicional

WHILE DUP "="	Inicia el ciclo condicional, hace una réplica, averigua si existe signo de igual
POS	Localiza el signo de igual
REPEAT DUP "="	Inicia las operaciones para el ciclo condicional, hace una réplica y coloca un signo igual
POS 34 CHR DUP +	Ubica el signo igual, coloca el caracter de comillas, lo duplica y lo junta
SWAP SUS	Voltea la cadena y la sustituye con el módulo SUS
END STR→	Termina el ciclo condicional y transforma la cadena en una lista
« F2	Inicia el programa y evalúa el programa F2
» EVL ΣLIST	Termina el programa, aplica el módulo EVL que pedirá el programa F2 para cada elemento de la lista y suma las contribuciones de la lista
»	Termina la subrutina condicional
« F4	Inicia la subrutina condicional y aplica el programa F4
» IFTE	Termina la subrutina condicional y aplica la subrutina correcta

» Termina el programa

Una vez tecleado el programa se almacena; [α][α]PEMO[α][STO]

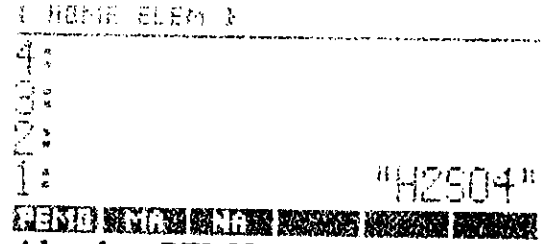
Enseguida se organiza el menú como se desea verlo de costumbre. Para ello se hace una lista. {PEMO MA NA } y luego se utiliza la orden [α][α]CST[α][STO]. Al pulsar [CST] y se obtiene la siguiente pantalla.



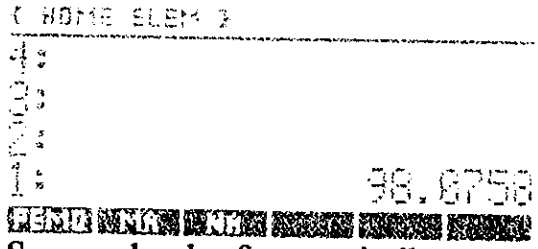
Ahora se organiza el menú. Para ello se teclea una lista con el orden en que se desea que aparezcan las teclas de menú; {PEMO MA NA PM SIMB TIPO F F1 F2 F3 F4}. Al teclear la orden ORDER los programas y subrutinas se ordenan automáticamente.

Práctica

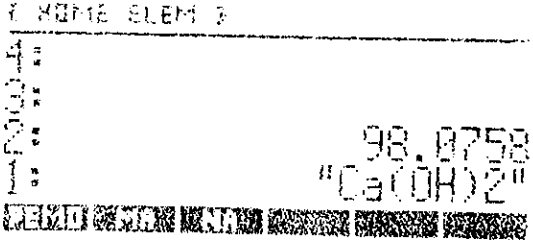
Se obtendrá el peso molecular del ácido sulfúrico. Se teclea la fórmula química del ácido sulfúrico de la siguiente manera: Se abren comillas con la tecla [⇒] [-] [α][α] enseguida la fórmula H2SO4, y se cierra con [ENTER] para verse así.



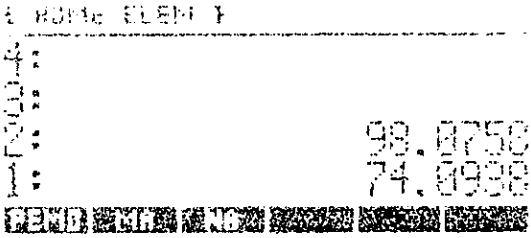
Al pulsar PEMO y se obtiene:



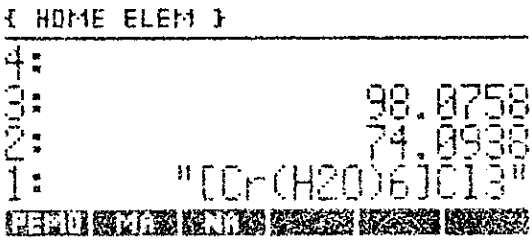
Se puede de forma similar con el hidróxido de calcio; "$Ca(OH)_2$" y se obtiene:



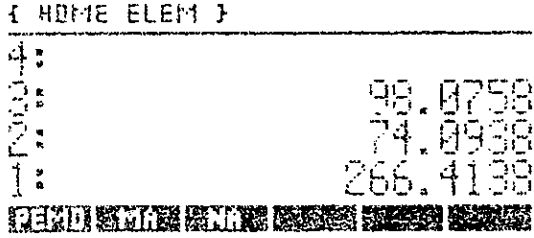
Ejecutando PEMO se obtiene:



Ahora con el siguiente compuesto "$[Cr(H_2O)_6]Cl_3$"



Se obtiene:



Al teclear "$[Cr(H_20)_5Cl]Cl_2.H_2O$"

Capítulo 4

Iteración de ecuaciones simultaneas no lineales

Es común en la carrera encontrar cálculos que requieren iteraciones. Cuando se trata de ecuaciones simultáneas no lineales, el cálculo por tanteo se vuelve una tarea engorrosa. El método numérico que se presenta en este capítulo permite hacer acercamientos más rápidos y de forma organizada.

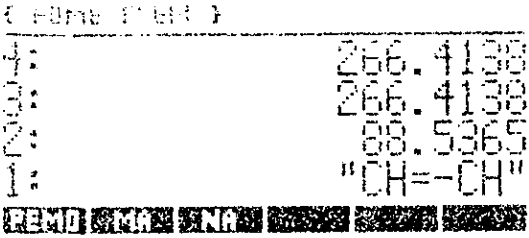
Si se une este método con un programa que realice los cálculos más rápidos y sin errores, se tiene una poderosa herramienta. Esto permite resolver problemas más complicados.

Para hacer un cálculo aproximado de la variación de funciones cuando varían un poco las variables independientes, podemos hacer uso de una matriz jacobiana:

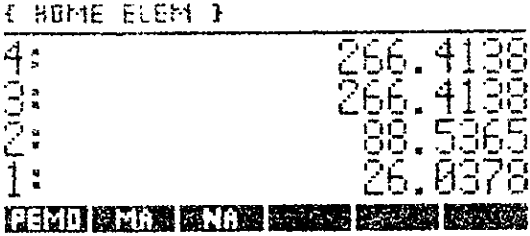
$$\begin{bmatrix} \frac{\partial f1}{\partial a} & \frac{\partial f1}{\partial b} & \cdots & \frac{\partial f1}{\partial n} \\ \frac{\partial f2}{\partial a} & \frac{\partial f2}{\partial b} & \cdots & \frac{\partial f2}{\partial n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial fm}{\partial a} & \frac{\partial fm}{\partial b} & \cdots & \frac{\partial fm}{\partial n} \end{bmatrix} \begin{bmatrix} \Delta a \\ \Delta b \\ \vdots \\ \Delta n \end{bmatrix} = \begin{bmatrix} \Delta f1 \\ \Delta f2 \\ \vdots \\ \Delta fm \end{bmatrix}$$

Si hacemos un despeje la operación queda:

Y finalmente se teclea el etino "CH=–CH"



Obteniendo:



Capítulo 4

Iteración de ecuaciones simultaneas no lineales

Es común en la carrera encontrar cálculos que requieren iteraciones. Cuando se trata de ecuaciones simultáneas no lineales, el cálculo por tanteo se vuelve una tarea engorrosa. El método numérico que se presenta en este capítulo permite hacer acercamientos más rápidos y de forma organizada.

Si se une este método con un programa que realice los cálculos más rápidos y sin errores, se tiene una poderosa herramienta. Esto permite resolver problemas más complicados.

Para hacer un cálculo aproximado de la variación de funciones cuando varían un poco las variables independientes, podemos hacer uso de una matriz jacobiana:

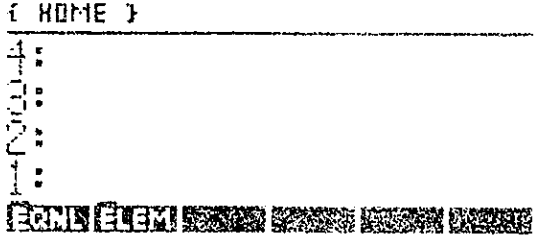
$$\begin{bmatrix} \frac{\partial f1}{\partial a} & \frac{\partial f1}{\partial b} & \cdots & \frac{\partial f1}{\partial n} \\ \frac{\partial f2}{\partial a} & \frac{\partial f2}{\partial b} & \cdots & \frac{\partial f2}{\partial n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial fm}{\partial a} & \frac{\partial fm}{\partial b} & \cdots & \frac{\partial fm}{\partial n} \end{bmatrix} \begin{bmatrix} \Delta a \\ \Delta b \\ \vdots \\ \Delta n \end{bmatrix} = \begin{bmatrix} \Delta f1 \\ \Delta f2 \\ \vdots \\ \Delta fm \end{bmatrix}$$

Si hacemos un despeje la operación queda:

$$\begin{bmatrix} \Delta a \\ \Delta b \\ \vdots \\ \Delta n \end{bmatrix} = \frac{\begin{bmatrix} \Delta f1 \\ \Delta f2 \\ \vdots \\ \Delta fm \end{bmatrix}}{\begin{bmatrix} \frac{\partial f1}{\partial a} & \frac{\partial f1}{\partial b} & \cdots & \frac{\partial f1}{\partial n} \\ \frac{\partial f2}{\partial a} & \frac{\partial f2}{\partial b} & \cdots & \frac{\partial f2}{\partial n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial fm}{\partial a} & \frac{\partial fm}{\partial b} & \cdots & \frac{\partial fm}{\partial n} \end{bmatrix}}$$

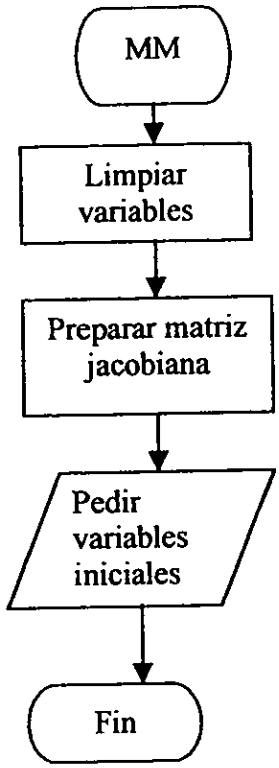
De esta manera, se puede encontrar la variación que se aplica a las variables independientes si se conoce la variación de las ecuaciones igualadas a cero, y la matriz jacobiana del sistema de ecuaciones.

Para empezar hay que organizar el directorio, el cual llamaremos EQNL. Se oprimen las teclas [⇒][’] para estar en HOME. Se teclea [α] [α]EQNL[ENTER],[α][α] CRDIR [ENTER] y se obtiene la siguiente vista:



FOR n "" n CHR +	Inicia el ciclo de conteo a la variable, coloca el delimitador de nombre como cadena, pide el caracter de número n y lo suma a la cadena.
STR→ +	Transforma de cadena a nombre y lo agrega en la lista
NEXT PURGE 65 EQ	Termina el ciclo de conteo, limpia los registros, coloca 65 y llama la lista de ecuaciones
SIZE 64 +	Mide el tamaño de la lista de ecuaciones y le suma 64
FOR n "" n CHR +	Inicia el ciclo de conteo a la variable, coloca el delimitador de nombre como cadena, pide el caracter de número n y lo suma a la cadena.
STR→ → c	Convierte la cadena en nombre y lo almacena en la variable local c
« EQ	Inicia la subrutina llama a la lista de ecuaciones
« c ∂	Inicia un programa en el que llama el nombre local c y ejecuta la derivada
» EVL	Termina el programa y ejecuta el módulo EVL
»	Termina la subrutina

Programa MM: Limpia variables, prepara la matriz jacobiana y solicita los valores de las variables iniciales.



[# 41886d 315.5]

Programa	*Descripción*
« { } 65 EQ SIZE 64	Inicia el programa coloca una lista nula y un 65, llama la lista de ecuaciones , toma su tamaño y coloca 64
+	Suma los 64

FOR n "" n CHR +	Inicia el ciclo de conteo a la variable, coloca el delimitador de nombre como cadena, pide el caracter de número n y lo suma a la cadena.
STR→ +	Transforma de cadena a nombre y lo agrega en la lista
NEXT PURGE 65 EQ	Termina el ciclo de conteo, limpia los registros, coloca 65 y llama la lista de ecuaciones
SIZE 64 +	Mide el tamaño de la lista de ecuaciones y le suma 64
FOR n "" n CHR +	Inicia el ciclo de conteo a la variable, coloca el delimitador de nombre como cadena, pide el caracter de número n y lo suma a la cadena.
STR→ → c	Convierte la cadena en nombre y lo almacena en la variable local c
« EQ	Inicia la subrutina llama a la lista de ecuaciones
« c ∂	Inicia un programa en el que llama el nombre local c y ejecuta la derivada
» EVL	Termina el programa y ejecuta el módulo EVL
»	Termina la subrutina

NEXT EQ SIZE	Termina el ciclo de conteo, llama la lista de ecuaciones y toma su medida
→LIST 'JAC' STO { }	Convierte en lista y almacena en JAC, coloca una lista vacia.
65 EQ SIZE 64 +	Coloca un 65 llama la lista de ecuaciones toma su tamaño y le suma 64
FOR n "" n CHR +	Inicia el ciclo de conteo para la variable n, coloca un delimitador de nombre y le agrega el caracter n
+	Agrega la cadena a la lista
NEXT	Termina el ciclo de conteo
« DUP "" INPUT	Inicia un programa, hace la réplica coloca un caracter nulo y pide la entrada desde el teclado
STR→ SWAP STR→ STO	Convierte la cadena en número, lo voltea convierte la cadena en nombre y almacena el valor en dicho nombre
» EVL	Termina el programa y le aplica el módulo EVL
»	Termina el programa

Una vez tecleado el programa se almacena; [α][α]MM[STO]

DO DUP "{" POS	Inicia un ciclo condicional, donde duplica la cadena y pide la posición de las llaves
"[" SWAP SUS	Coloca un corchete (delimitador de vectores y matrices), lo voltea y lo sustituye con el módulo SUS
UNTIL DUP "{" POS	Cierra el ciclo condicional e inicia las cláusulas de condición, hace una réplica de la cadena y averigua si tiene una llave
0 ==	Averigua si la posición es cero
END	Termina el ciclo condicional
DO DUP "}" POS	Inicia el ciclo condicional, hace una réplica, localiza el cierre de llave
"]" SWAP SUS	Coloca un cierre de corchete lo voltea y lo sustituye con el módulo SUS
UNTIL DUP "}" POS	Termina el ciclo condicional e inicia las cláusulas, hace una réplica y averigua si existen cierres de llave
0 ==	Averigua si la posición es cero

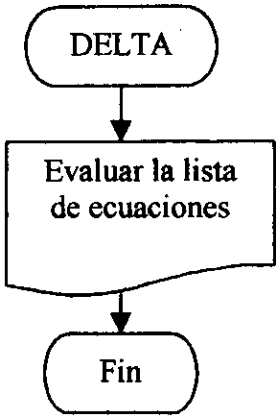
[# 28627d 303]

Programa	*Descripción*
« EQ	Inicia el programa, llama la lista de ecuaciones
« →NUM	Inicia un subprograma y convierte las expresiones algebraicas en números
» EVL LIST→ { } +	Termina el subprograma y lo ejecuta para cada objeto de la lista, convierte el resultado de lista a objetos independientes coloca una lista vacía y encierra el tamaño en la lista
→ARRY JAC	Convierte los objetos independientes en vector e invoca a la matriz jacobiana transpuesta
«	Inicia un subprograma
« →NUM	Inicia un subprograma con la instrucción de convertir las expresiones algebraicas en números
» EVL	Termina el subprograma y da la instrucción de evaluar listas
» EVL →STR	Termina el programa y evalúa la lista de la matriz, la transforma a cadena

DO DUP "{" POS	Inicia un ciclo condicional, donde duplica la cadena y pide la posición de las llaves
"[" SWAP SUS	Coloca un corchete (delimitador de vectores y matrices), lo voltea y lo sustituye con el módulo SUS
UNTIL DUP "{" POS	Cierra el ciclo condicional e inicia las cláusulas de condición, hace una réplica de la cadena y averigua si tiene una llave
0 ==	Averigua si la posición es cero
END	Termina el ciclo condicional
DO DUP "}" POS	Inicia el ciclo condicional, hace una réplica, localiza el cierre de llave
"]" SWAP SUS	Coloca un cierre de corchete lo voltea y lo sustituye con el módulo SUS
UNTIL DUP "}" POS	Termina el ciclo condicional e inicia las cláusulas, hace una réplica y averigua si existen cierres de llave
0 ==	Averigua si la posición es cero

END STR→ TRN /	Termina el ciclo condicional, transforma la cadena a matriz la transpone y da la instrucción de resolver el sistema de ecuaciones mediante el operador /
NEG 65 EQ SIZE 64 +	Cambia de signo a los resultados, coloca un 65, llama la lista de ecuaciones toma el tamaño y le suma 64
FOR h DUP h 64 –	Inicia un ciclo de conteo para la variable h, hace una réplica de los resultados, llama a h y le resta 64
GET "" h CHR +	Obtiene el elemento en la posición adecuada al ciclo, coloca un caracter de inicio de nombre, coloca el caracter adecuado y lo agrega a la cadena
STR→ STO+	Lo convierte de cadena a nombre y almacena el incremento de la variable
NEXT DROP	Cierra el ciclo de conteo y tira la copia de resultados
»	Termina el programa

Programa DELTA: Calcula el error en las ecuaciones

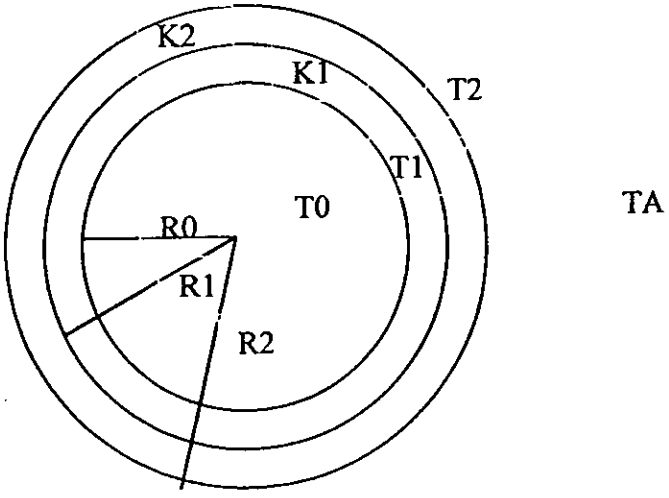


[# 53201d 46.5]

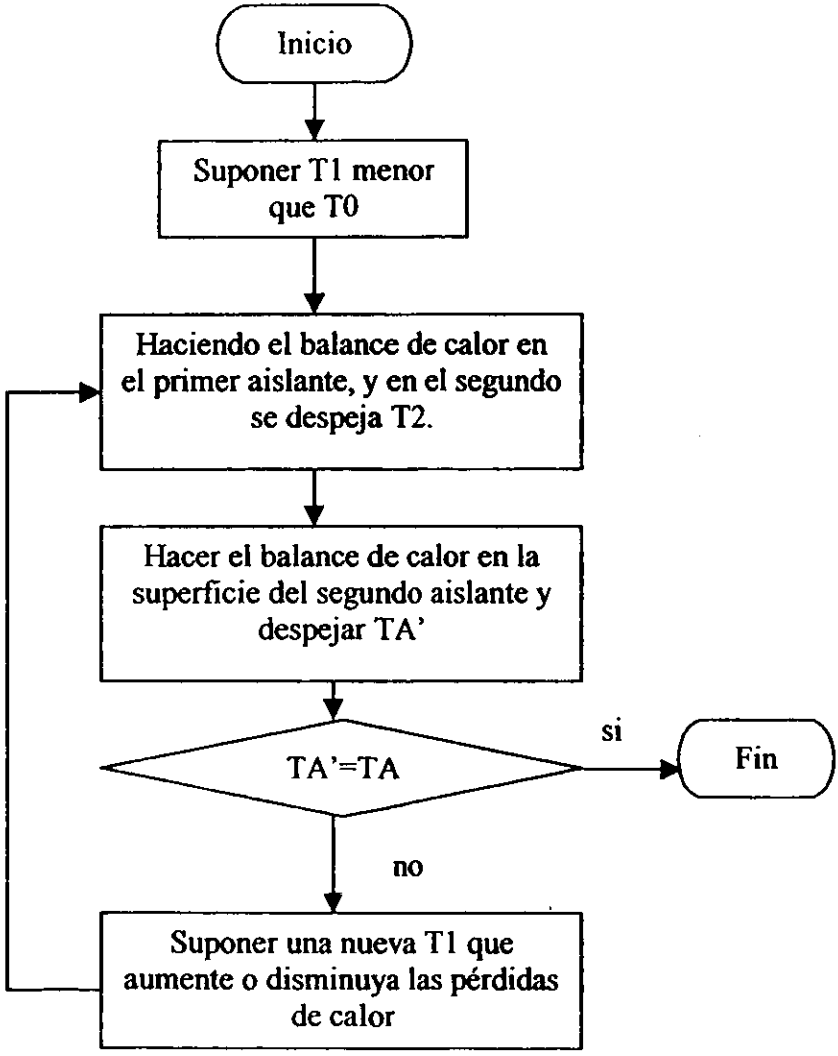
Programa	*Descripción*
« EQ	Llama la lista de ecuaciones
« →NUM	Inicia un subprograma y convierte las expresiones algebraicas en números
» EVL	Termina el subprograma y evalúa la lista.
»	Termina el programa

Práctica 1

En este ejemplo se trata un tubo con dos aislantes, de los que se conoce la temperatura del fluido, la temperatura del medio ambiente y las conductividades que dependen o no de la temperatura. El objetivo será evaluar las pérdidas de calor, a pesar de que se desconocen las temperaturas intermedias de los aislantes. Se desprecian las pérdidas por condensación.



Para resolver el siguiente problema en forma manual se sigue la siguiente secuencia:



Se tienen las siguientes ecuaciones:

$$q = \frac{2\pi K1(T1 - T0)}{Ln\left(\frac{R0}{R1}\right)} \qquad q = \frac{2\pi K2(T2 - T1)}{Ln\left(\frac{R1}{R2}\right)}$$

$$q = ha2\pi R2(T2 - TA)$$

las cuales se tienen que igualar a cero, y entrando al editor de ecuaciones se pueden obtener las siguientes vistas:

La forma de entrar al editor de ecuaciones es usando la tecla morada con el ENTER.

Una vez que se tienen las ecuaciones listas, se cuentan cuantas son, en este caso 3. Se teclea el 3 y enseguida →LIST. Luego se almacena la lista en EQ ('EQ' STO). Ahora se almacenan todos los datos conocidos.

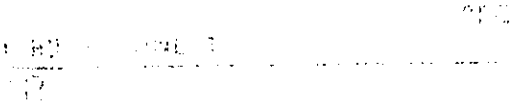
Datos	Forma de introducirlos
T0=300 °C	300 [ENTER][α] T0 [STO]
TA=25 °C	25[ENTER][α]T[α]A [STO]
K1= $680\frac{J}{h°Cm^2}$	680 [ENTER][α]K1 [STO]
K2=$500\frac{J}{h°Cm^2}$+20.4*T	'500+20.4*T'[α]K2[STO]
T=$\frac{T2+T1}{2}$	'(T2+T1)/2'[α]T[STO]
HA=$40000\frac{J}{h°Cm^2}$+170*ΔT	'40000+170*DT'[α]H[α]A [STO]
DT=T2-TA	'T2-TA'[α]D[α] T[STO]
R0=0.025 m	.025[α]R0[STO]
R1=0.030 m	.03 [α] R1[STO]
R2=0.035 m	.035 [α] R2[STO]

Ahora se renombran las variables desconocidas:

Q=A	[α]A[ENTER][α]Q[STO]
T1=B	[α]B[ENTER][α]T1[STO]
T2=C	[α]C[ENTER][α]T2[STO]

Esto es muy importante, ya que el programa interpreta las incógnitas en orden alfabético. De esta manera **el programa es capaz de aceptar tantas variables como letras del alfabeto en inglés existan, (de la A a la Z).**

Todo está listo para ejecutar el programa. En primer lugar se corre el programa MM. Este programa pedirá el primer estimado de los valores desconocidos, es decir, las semillas de las iteraciones.



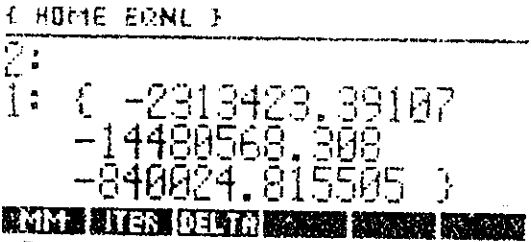
Se teclea un valor de 30000 J

[ENTER]

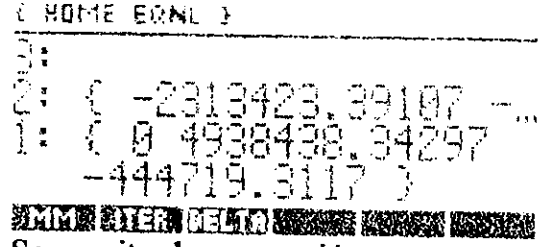
Ahora a T1 se le da un valor de 200 °C

Y finalmente 100°C para T2 que ha sido nombrada como C.

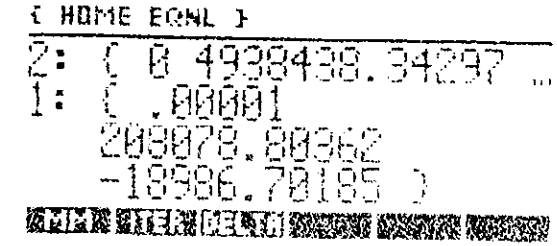
Se evalúa el programa delta para tener una idea inicial de qué tan desviados están los valores de la solución.

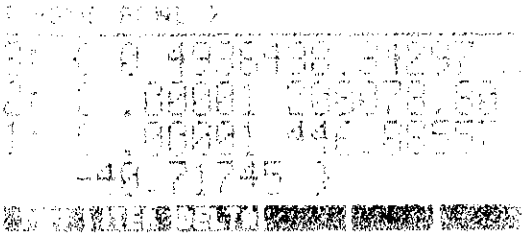


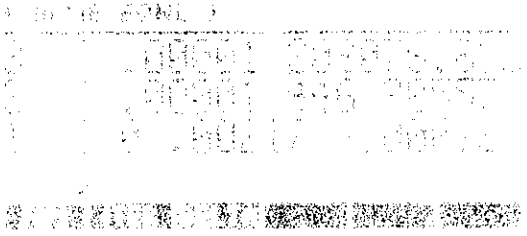
Se corre la primera iteración ITER y enseguida se corre DELTA para tener idea de cuanto se acerca al valor cero.

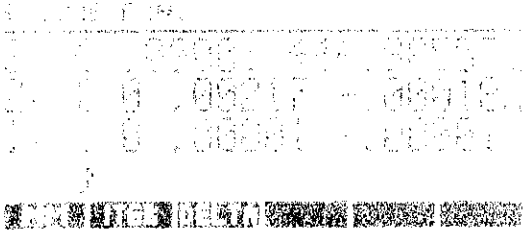


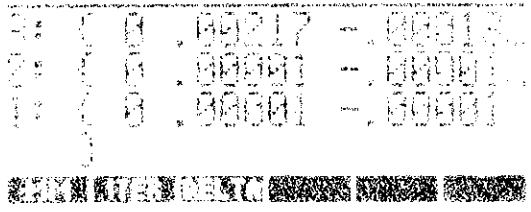
Se repite la operación cuantas veces sea necesario para conseguir la aproximación requerida





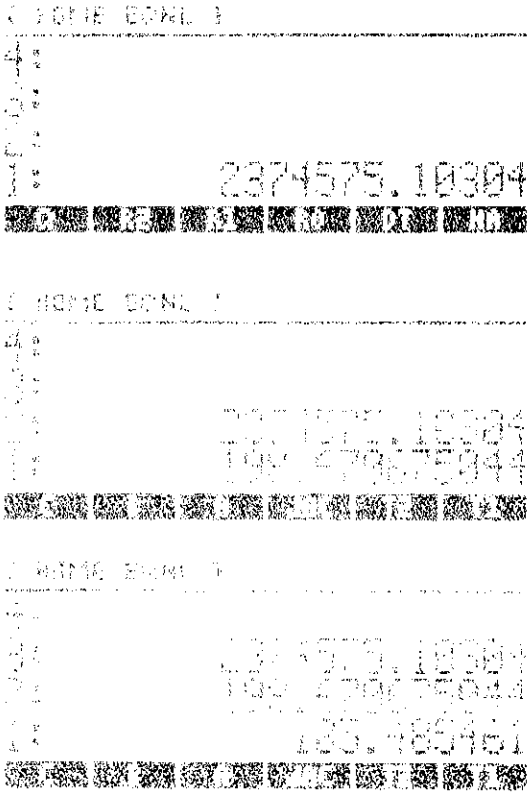






En este caso la aproximación llega a un límite por error de redondeo, en donde las diferencias ya no pueden acercarse más al valor cero.

Después de 6 iteraciones se puede observar el resultado. Se busca el menú de variables y se oprimen aquellos donde se almacenaron las incógnitas. De ser necesario se oprime la tecla NXT hasta encontrar las variables buscadas.



Así se encuentra que el calor perdido es 2.3 millones de Joules por hora por metro lineal de tubería. La temperatura interna de los aislantes es de 199 °C y la temperatura externa del aislante es de 185 °C.

Práctica 2

Problema 18.N_3 del Bird, Steward, Lightfoot, "Fenómenos de transporte".

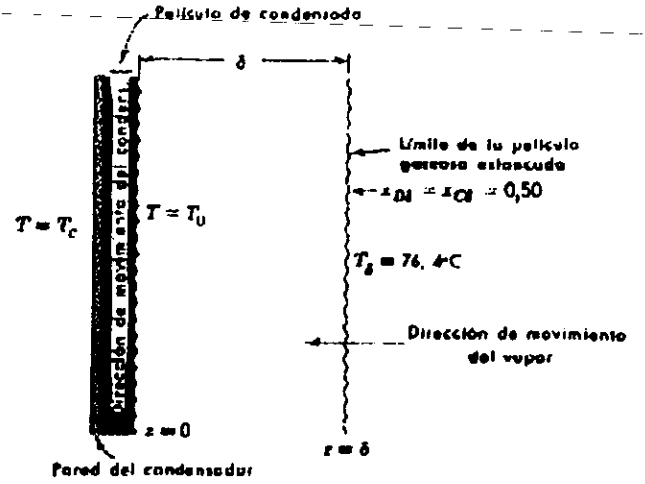
Una mezcla equimolar de vapores de cloroformo (C) y benceno (B), a 1 atm, condensa sobre una superficie fría. La temperatura *Tc* de la superficie fría puede variar. La especie más volátil (C) se acumula en las proximidades de la superficie fría, de forma que el proceso de condensación se retarda debido a la presencia de una «película» de gas, cuya temperatura y composición dependen de la sustancia *z* medida desde la superficie del líquido. La relación de benceno a cloroformo en el condensado puede considerarse igual a la relación de velocidades de condensación. Se tienen los siguientes datos:

El espesor δ de la película es 0.1 mm, y el coeficiente de transmisión de calor *h* de la película de condensado es 975 kcal hr^{-1} m^{-2} $°C^{-1}$.

D_{BC}=0.050 cm^2/s; Cp_C=16.5 kcal/kgmol °C;

Cp_B=22.8 kcal/kgmol °C

El calor molar de vaporización λ_{mezc} y la conductividad calorífica k_{mezc} de las mezclas cloroformo–benceno pueden considerarse constantes e iguales a 7.110 kcal/kgmol y 0.010 kcal/hr m °C, respectivamente. La influencia de la temperatura sobre las propiedades fisicas puede despreciarse.



Se tienen las siguientes ecuaciones:

$$-(N_{BZ} + N_{CZ}) = \frac{cD_{BC}}{\delta} Ln\left\{\frac{X_{B0} - X'_B}{X_{B\delta} - X'_B}\right\} = \frac{cD_{BC}}{\delta} Ln\left\{\frac{X_{C0} - X'_C}{X_{C\delta} - X'_C}\right\}$$

X_A^1 es la fracción molar del componente A en el condensado.

$$-q_z\big|_{Z=0} = (T\delta - T_0)\left(\frac{k_{mezc}}{\delta}\right)\left(\frac{\beta}{1-e^{-\beta}}\right) - \lambda_{mezc}(N_{BZ} + N_{CZ})$$

$$\beta = (N_{BZ} + N_{CZ})Cp_{mezc}\frac{\delta}{k_{mezc}} \qquad Cp_{mezc} = x'_B Cp_B + x'_C Cp_C$$

$$q_Z = h(T_C - T_0)$$

y tomando en cuenta la relación:

$1 = X_B + X_C$ y tomando como flujo N en vez de $N_{Bl} + N_{Cl}$

$$\left\{\frac{X_{C0}-X_C^l}{X_{C\delta}-X_C^l}\right\}=\left\{\frac{(1-X_{B0})-(1-X_B^l)}{(1-X_{B\delta})-(1-X_B^l)}\right\}=\left\{\frac{X_B^l-X_{B0}}{X_B^l-X_{B\delta}}\right\}$$

Así tenemos 4 ecuaciones con 4 incógnitas.

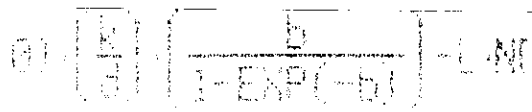
Se igualan a cero las ecuaciones y se introducen a la calculadora, ya sea por el editor de ecuaciones o copiando directamente:

'N+c*DBC/d*LN((XB0-XBl)/(XBd-XBl))' 'EQ1' STO

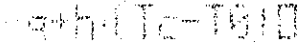
igual con las siguientes ecuaciónes:

'N+c*DBC/d*LN((XBl-XB0)/(XBl-XBd))' 'EQ2' STO

'q+(Td-T0)*(k/d)*(b/(1-EXP(-b)))-L*N' 'EQ3' STO



'-q+h*(Tc-T0)' 'EQ4' STO



y así las otras ecuaciones auxiliares:

'XBl*CPB+XCl*CPC' 'Cp' STO

'1-XBl' 'XCl' STO

'N*Cp*d/k' 'b' STO

y se hace la sustitución de variables desconocidas:

A 'N' STO [mol/s m^2]

B 'XB0' STO

C 'q' STO [J/s m^2]

D 'T0' STO [°C]

Mediantc la ecuación dc gas idcal obtenemos la concentración. (76.4°C=349.55K).

$$\frac{n}{v} = \frac{P}{RT} = \frac{1atm}{0.082\frac{Latm}{molK}(349.55K)} = 3.489*10^{-2}\,\text{mol/L}$$

$C_{Benceno}$=XBδ*C=(0.5)(3.489*10^{-2}mol/L)=0.0174mol/L=

17.44mol/m^3

Así tenemos los siguientes datos:

17.44 'c' STO [mol/m^3]

.000005 'DBC' STO[m^2/s]

.0001 'd' STO [m]

.5 'XBd' STO [ ]

76.4 'Td' STO [°C]

.01163 'k' STO [J/sm°C]

1133.925 'h' STO [J/s m^2°C]

29.768148 'L' STO [J/mol] (corresponde a λmezc)

95.45904 'CPB' STO [J/mol °C]

69.0822 'CPC' STO [J/mol °C]

Ahora suponemos las dos variables que faltan:

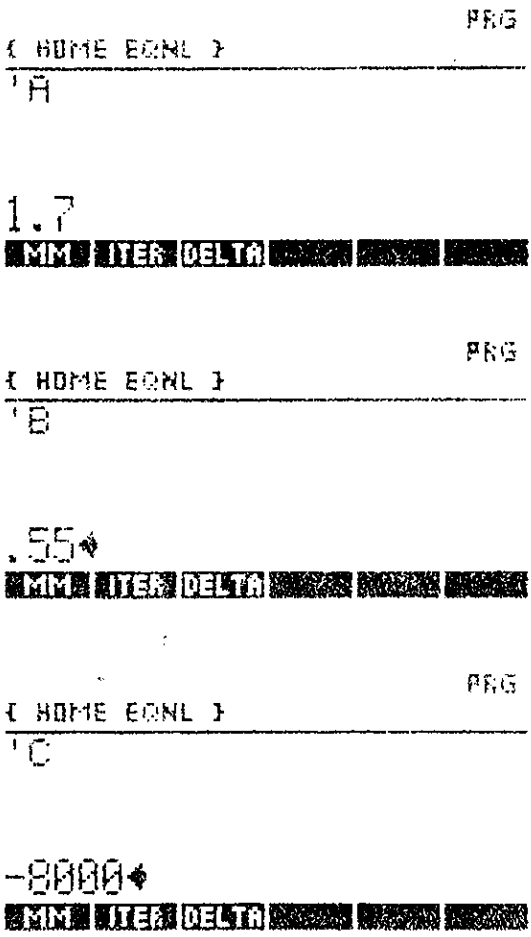
30 'Tc' STO [°C]

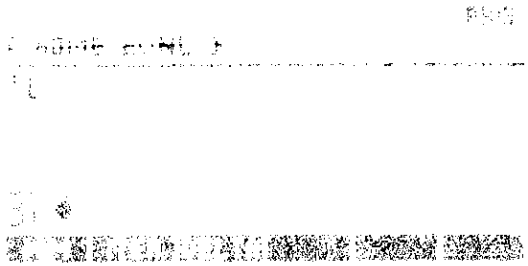
.56 'XBl' STO [ ]

Almacenamos las cuatro ecuaciones:

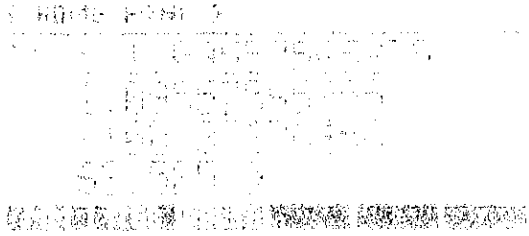
{EQ1 EQ2 EQ3 EQ4} 'EQ' STO

Corremos el programa y nos pregunta los datos.

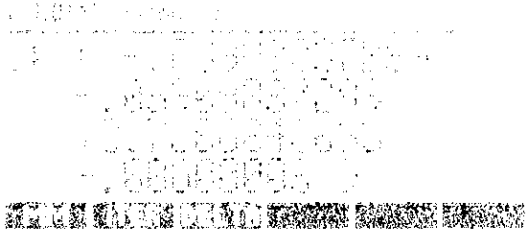




Ahora se pulsa la tecla DELTA del menú [CST]



Se pulsa ITER y nuevamente DELTA, obteniendo la siguiente variación.



Y los resultados son:

A= 1.2745 B=.547 C=-7087

D=36.25

Capítulo 5

Cálculo de volumen para tanques.

Cuando se tienen una serie de caudales a diferentes horas de un cierto periodo de tiempo, se requiere de un tanque que regularice el flujo.

Para ello se busca la tangente superior e inferior de la curva de caudal acumulado.

Si encima de todo hay diferentes concentraciones, es necesario averiguar cuál es el tamaño mayor, si el requerido por flujo volumétrico, o por flujo de soluto.

Se tiene el tiempo H_i con el volumen V_i y el caudal acumulado $Vac_i=(H_i - H_{i-1})*V_i + Vac_{i-1}$ donde i va de 2 a n caudales.

Para caudales con solutos se tiene el tiempo H_i, el volumen V_i, concentración Ci y el caudal acumulado $Vac_i=(H_i - H_{i-1})*V_i*C_i + Vac_{i-1}$ donde i va de 2 a n caudales.

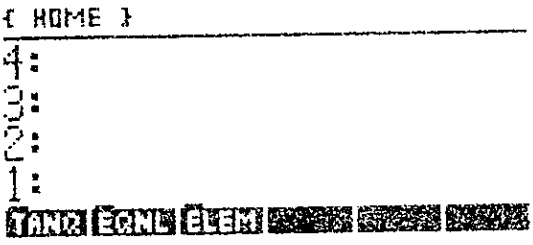
En este capítulo se verá el manejo de algunas listas y la transposición de matrices para resolver esta clase de problemas.

Una vez calculados los caudales acumulados ya sea volumétricos o soluciones contra tiempos, se traza una línea para unir el último caudal acumulado con el primero. Usando la pendiente de esa línea, se trazan arriba y abajo las

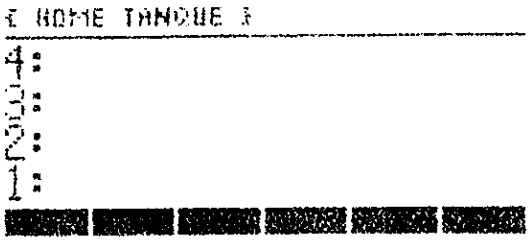
tangentes, y la diferencia en la ordenada al origen, indica el volumen necesario. Si se busca la tangente superior, se prueba punto por punto, la recta que tiene la pendiente mencionada; si los puntos calculados que pasan por la recta son mayores que los puntos de la curva, entonces esta es la tangente, de lo contrario algunos puntos serán mayores a la recta. Lo mismo le sucede a la tangente inferior, pero los puntos tienen que ser mayores que los calculados por la recta.

Estos programas requieren del subprograma DLT que se encuentra en el apéndice, y si se trata de la calculadora 48SX también requerirá del subprograma ΔLIST.

Primero se crea el directorio donde se trabajará este tipo de información: [α] [α] TANQUE [ENTER] [α] [α] CRDIR [ENTER]

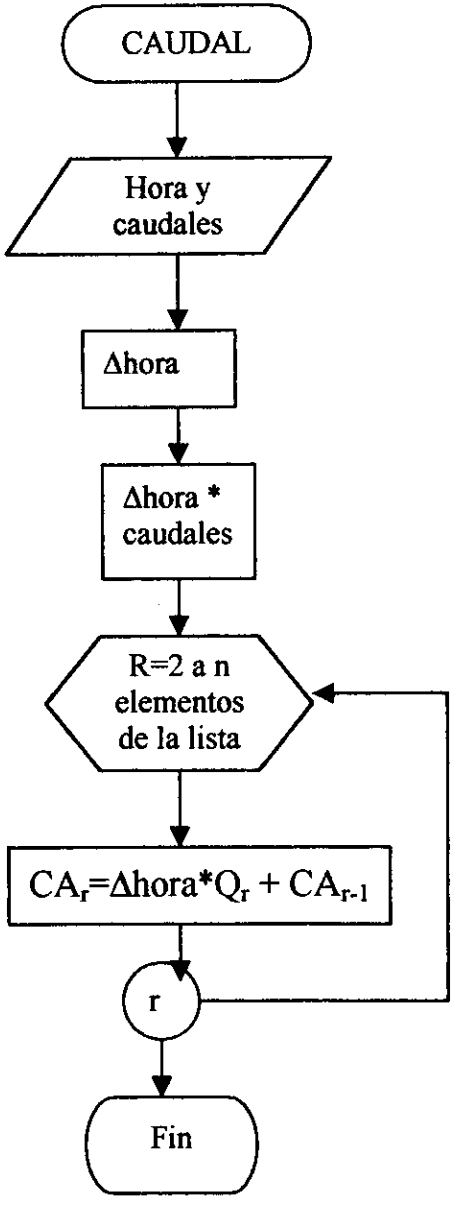


Se oprime la tecla del menú "TANQ"y aparece la siguiente imagen:



Aquí se comienzan a escribir los programas.

CAUDAL :Pide los datos de hora y caudal, para calcular el caudal acumulado.

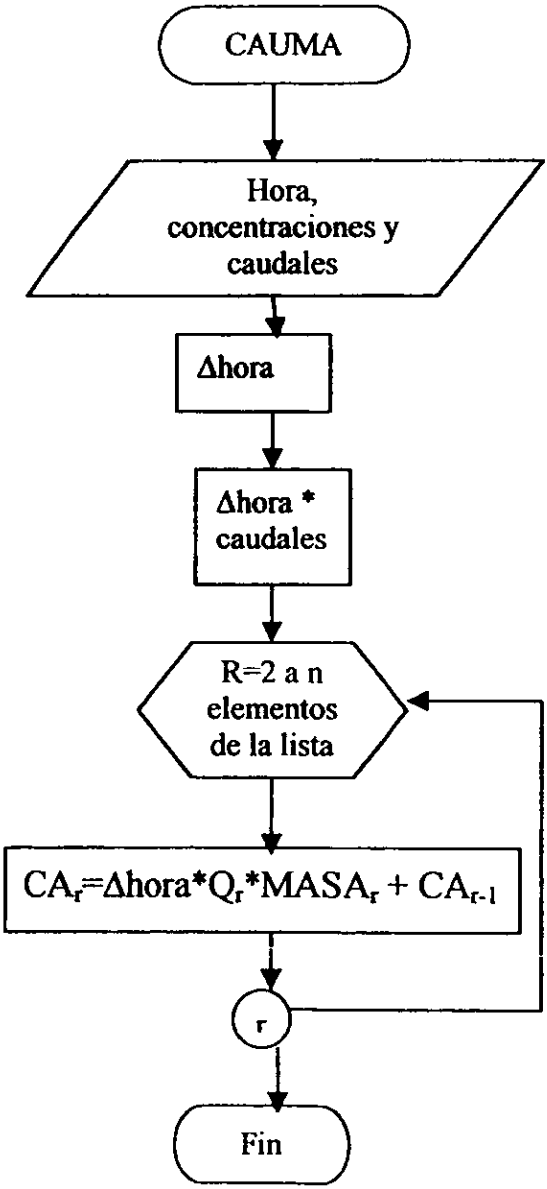


[#50465d 199.5]

Programa	*Descripción*
« "HORA" "{" INPUT	Inicia el programa, pide la lista de horas de los caudales
STR→ 'HORA' STO "Q"	Transforma de cadena a lista y almacena en la variable HORA y coloca la cadena Q
"{" INPUT STR→ 'Q'	Solicita los diferentes caudales coloca la variable Q.
STO { 0 } HORA +	Almacena la lista en la variable, coloca una lista con un cero y le suma la lista de horas
ΔLIST Q	Con el delta de las horas llama a los caudales
« *	Inicia el subprograma que multiplica las dos listas.
» DLT 2 OVER SIZE	Termina el subprograma aplica el programa a las listas, pone un dos, hace una réplica de la lista y toma el tamaño de ésta.
FOR r DUP r 1 -	Inicia un ciclo para la variable r, hace una copia, pone el argumento r y le resta uno.

GET OVER r GET + r	Obtiene el elemento r-1, obtiene una copia de la lista de encima. Luego coloca el ciclo r, obtiene el elemento r y lo suma con el elemento r-1. Coloca el ciclo r.
SWAP PUT	Voltea los últimos niveles y coloca en la lista en el elemento r
NEXT 'CA' STO	Cierra el ciclo, y almacena los caudales acumulados en CA.
»	Termina el programa.

CAUMA: Pide los datos de hora, concentración y caudal, para calcular el caudal acumulado másico.

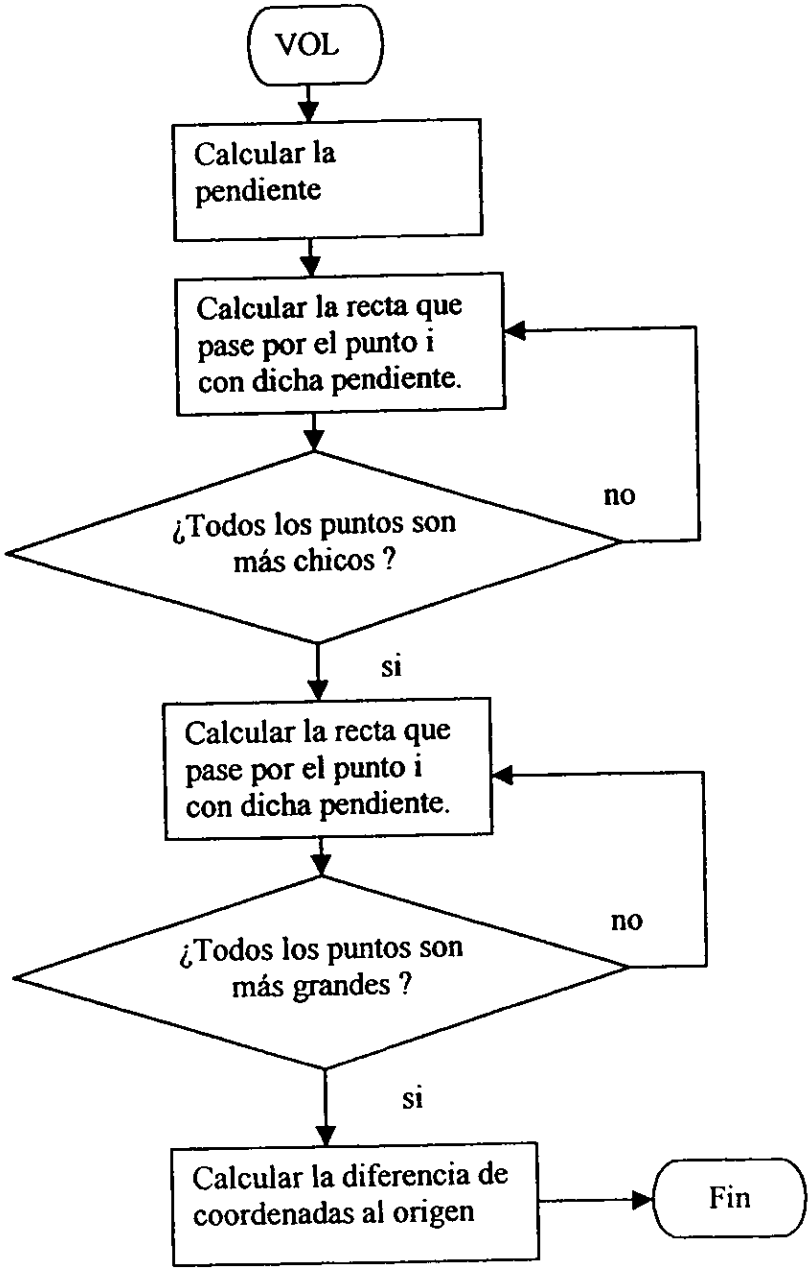


[# 39773d 262.5]

Programa	*Descripción*
« "HORA" "{" INPUT	Inicia el programa, pide la lista de horas de los caudales
STR→ 'HORA' STO "Q"	Transforma de cadena a lista y almacena en la variable HORA y coloca la cadena Q
"{" INPUT STR→ 'Q'	Solicita los diferentes caudales coloca la variable Q.
STO "MASA" "{"	Almacena la lista en la variable. Coloca la cadena MASA y el inicio de la lista.
INPUT STR→ 'MASA'	Solicita las diferentes concentraciones, los convierte de cadena a lista y coloca la variable MASA
STO { 0 } HORA +	Almacena la lista en la variable, coloca una lista con un cero y le suma la lista de horas
ΔLIST Q	Con el delta de las horas llama a los caudales
« *	Inicia el subprograma que multiplica las dos listas.
» DLT MASA	Termina el subprograma aplica el programa a las listas, llama la vairable MASA.
« *	Inicia el subprograma que multiplica las dos listas.

» DLT 2 OVER SIZE	Termina el subprograma aplica el programa a las listas, pone un dos, hace una réplica de la lista y toma el tamaño de ésta.
FOR r DUP r 1 –	Inicia un ciclo para la variable r, hace una copia, pone el argumento r y le resta uno.
GET OVER r GET + r	Obtiene el elemento r-1, obtiene una copia de la lista de encima. Luego coloca el ciclo r, obtiene el elemento r y lo suma con el elemento r-1. Coloca el ciclo r.
SWAP PUT	Voltea los últimos niveles y coloca en la lista en el elemento r
NEXT 'CA' STO	Cierra el ciclo, y almacena los caudales acumulados en CA.
»	Termina el programa.

VOL: Calcula el volumen del tanque, y prepara todo lo que se requiera para la graficación.



[# 57788d 621]

Programa	*Descripción*
« "UNIDAD" "" INPUT	Solicita la unidad
'UNI' STO CA CA	Almacena la unidad en UNI y llama al caudal acumulado dos veces
SIZE GET HORA CA	Toma el tamaño y obtiene el último elemento. Llama la hora y el caudal acumulado
SIZE GET / → m	Toma el tamaño, obtiene el último elemento y divide para obtener la pendiente. Almacena la pendiente en m
« 0 0	Pone dos ceros
DO SWAP 1 +	Comienza el ciclo condicional, voltea los últimos números y le suma uno.
HORA OVER GET m *	Llama la lista de horas, nace una copia del nivel 2, obtiene la hora n, y la multiplica por la pendiente.
NEG OVER CA SWAP	Obtiene el negativo hace una copia del segundo nivel (que es el contador n), llama al caudal acumulado y voltea los dos niveles.
GET + DUP → c	Obtiene el elemento n, lo suma hace una copia y lo almacena como c.

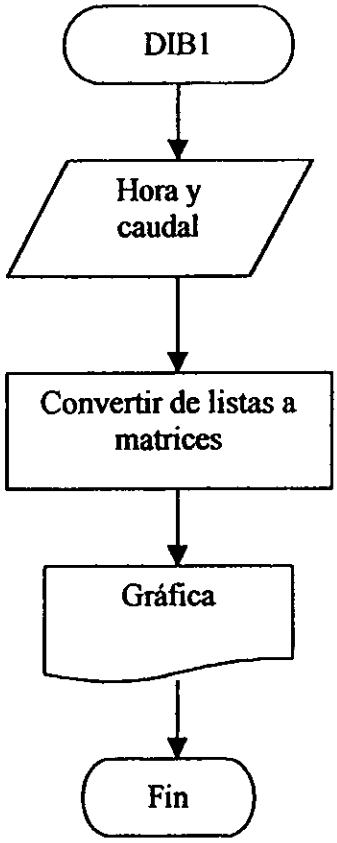
« HORA	Inicia un subprograma y llama las horas
« m * c +	Inicia un subprograma, llama la pendiente, la multiplica y le suma la coordenada al origen.
» EVL	Termina el subprograma y lo evalúa para la lista
» 4 ROLL DROP	Termina el subprograma. Voltea el nivel 4 y tira el valor.
UNTIL CA	Inicia la cláusula del ciclo y llama al caudal acumulado.
« ≤	Inicia un subprograma y coloca el condicional menor igual que.
» DLT ΣLIST	Termina el subprograma y lo aplica a las dos listas, luego suma toda la lista, (si algún punto es más grande, su bandera será falsa o sea cero y la suma será diferente al tamaño de la lista).
CA SIZE ==	Llama la lista de caudal acumulado y verifica que sea igual la suma de la lista que el tamaño.
END SWAP 0	Termina el ciclo condicional voltea los valores y coloca un cero.

DO SWAP 1 +	Comienza el ciclo condicional, voltea los últimos números y le suma uno.
HORA OVER GET m *	Llama la lista de horas, nace una copia del nivel 2, obtiene la hora n, y la multiplica por la pendiente.
NEG OVER CA SWAP	Obtiene el negativo hace una copia del segundo nivel (que es el contador n), llama al caudal acumulado y voltea los dos niveles.
GET + DUP → c	Obtiene el elemento n, lo suma hace una copia y lo almacena como c.
« HORA	Inicia un subprograma y llama las horas
« m * c +	Inicia un subprograma, llama la pendiente, la multiplica y le suma la coordenada al origen.
» EVL	Termina el subprograma y lo evalúa para la lista
» 4 ROLL DROP	Termina el subprograma. Voltea el nivel 4 y tira el valor.
UNTIL CA	Inicia la cláusula del ciclo y llama al caudal acumulado.
« ≥	Inicia un subprograma y coloca el condicional mayor igual que.

» DLT ΣLIST	Termina el subprograma y lo aplica a las dos listas, luego suma toda la lista, (si algún punto es más chico, su bandera será falsa o sea cero y la suma será diferente al tamaño de la lista).
CA SIZE ==	Llama la lista de caudal acumulado y verifica que sea igual la suma de la lista que el tamaño.
END SWAP DROP	Termina el ciclo condicional voltea los valores y tira el valor.
DUP2 m X * + SWAP m	Hace una réplica de los dos últimos valores, multiplica una X por la pendiente, suma los dos valores, voltea los dos últimos valores y llama a la pendiente.
X * + = 'EQ' STO	Multiplica una X por la pendiente, suma la ordenada al origen, iguala las ecuaciones y las almacena en EQ
SWAP - DUP	Voltea el segundo nivel, hace una resta, y luego hace una copia.
"EL VOLUMEN ES "	Coloca una cadena de mensaje.

SWAP →STR + " " +	Voltea el segundo nivel, lo convierte en cadena, agrega los caracteres a la cadena de mensaje y agrega un espacio a la cadena.
UNI + MSGBOX	Llama la unidad, la agrega a la cadena y muestra el resultado en la pantalla.
"PROMEDIO " CA DUP	Coloca la cadena de promedio, llama al caudal acumulado y hace una copia
SIZE GET HORA DUP	Mide el tamaño de la lista, obtiene el último objeto de la lista, invoca la HORA y hace una copia.
SIZE GET / DUP 3	Mide el tamaño de la lista, obtiene el último elemento de la lista, divide los números, hace una réplica y coloca un 3.
ROLLD →STR + UNI +	Voltea el objeto hasta el nivel 3, convierte el número en cadena, lo agrega a la otra cadena, invoca la unidad y la agrega a la cadena.
"/tiempo" + MSGBOX	Coloca la cadena de tiempo, agrega a la cadena anterior y muestra el resultado en pantalla.
»	Termina el subprograma.
»	Termina el programa.

DIB1: Grafica los caudales contra tiempo.

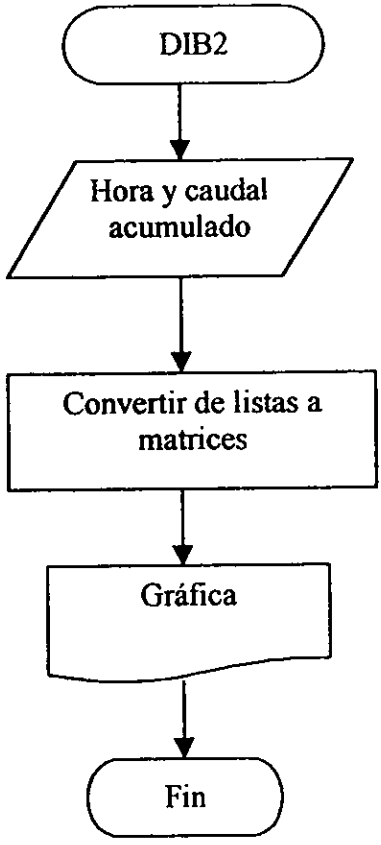


[# 14779d 85.5]

Programa	*Descripción*
« HORA LIST→ { } +	Llama la lista de horas la descompone en sus elementos, encierra el tamaño en una lista

→ARRY CLΣ Σ+ Q	Convierte los elementos en vector, borra la matriz de datos estadísticos, le agrega el vector a la matriz y llama a Q
LIST→ { } + →ARRY	Descompone la lista en sus elementos encierra el tamaño en una lista y convierte los elementos en vector
Σ+ RCLΣ TRN STOΣ	Suma el vector a la matriz estadística, llama a ésta, la transpone y la vuelve a almacenar.
SCATRPLOT	Dibuja la matriz
DO	Empieza el ciclo condicional
UNTIL KEY	Comienza la condición, pide el código tecleado.
END DROP	Termina el ciclo condicional y tira el resultado
»	Termina el programa.

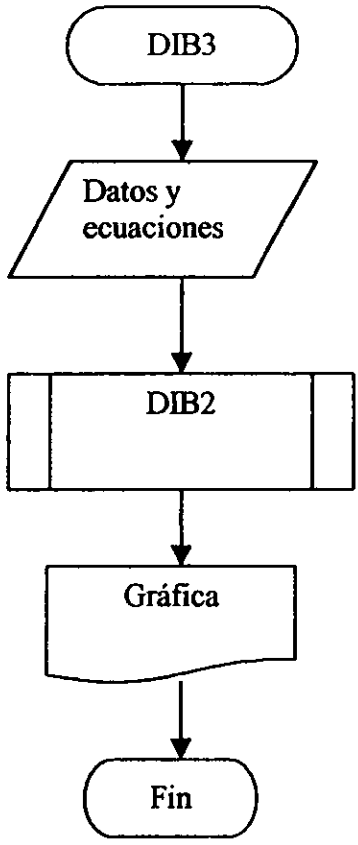
DIB2: Grafica el caudal acumulado contra el tiempo.



[# 422622d 86.5]

Programa	*Descripción*
« HORA LIST→ { } +	Llama la lista de horas la descompone en sus elementos, encierra el tamaño en una lista
→ARRY CLΣ Σ+ CA	Convierte los elementos en vector, borra la matriz de datos estadísticos, le agrega el vector a la matriz y llama al caudal acumulado.
LIST→ { } + →ARRY	Descompone la lista en sus elementos encierra el tamaño en una lista y convierte los elementos en vector
Σ+ RCLΣ TRN STOΣ	Suma el vector a la matriz estadística, llama a ésta, la transpone y la vuelve a almacenar.
SCATRPLOT	Dibuja la matriz
DO	Empieza el ciclo condicional
UNTIL KEY	Comienza la condición, pide el código tecleado.
END DROP	Termina el ciclo condicional y tira el resultado
»	Termina el programa.

DIB3: Grafica las dos tangentes a la curva de caudal acumulado.



[# 8193d 43.5]

Programa	*Descripción*
« DIB2 FUNCTION	Llama al programa DIB2, y cambia la modalidad de gráficas estadísticas a algebraicas.
DRAW	Dibuja las dos tangentes.
DO	Inicia el ciclo condicional
UNTIL KEY	Inicia la condición y pide el código tecleado
END DROP	Termina el ciclo y tira el resultado
»	Termina el programa.

Ahora hay que organizar el archivo. Se teclea [←]{ CAUDAL CAUMA VOL DIB1 DIB2 DIB3 } [ENTER] [α][α]ORDER[ENTER]

Para mantener limpio el archivo se puede hacer un programa que elimine los archivos que se usaron para un cálculo y que ya no se necesiten;
LIMP: Elimina los archivos que no se ocupan.

```
[# 15460d  85.5]
« { ΣDAT Q HORA
MASA EQ PPAR CA
ΣPAR UNI} PURGE
»
```

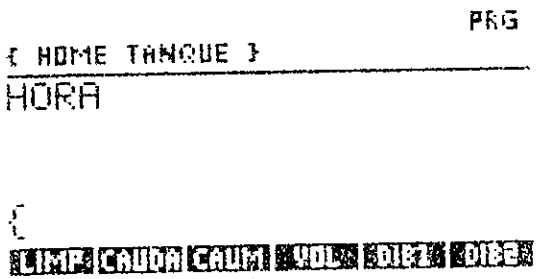
Práctica.

A lo largo de un día se presentó la siguiente variación de caudales, en donde además el efecto de la temperatura, generó una diferente concentración de calcio.

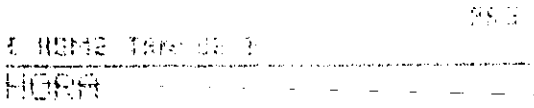
Hora (h)	Q (L/s)	Ca^{++} (mg/L)
2	36	150
4	25	300
6	36	730
8	46.5	1500
10	48.5	8200
12	58	7450
14	66.5	8030
16	79	5200
18	84.5	7900
20	104.5	3500
22	61	2000
24	43	250

¿Qué volumen se requiere para tener un flujo regulado, tanto en volumen como en concentración?

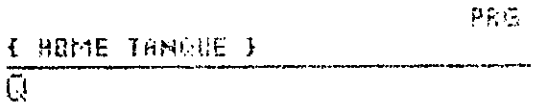
Se pulsa la tecla de menú CAUDA



Y se llenan los datos.

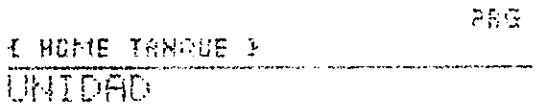


Se cierra con un ENTER. Igual se hace con el flujo Q.

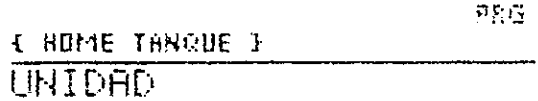


Se aplica ENTER

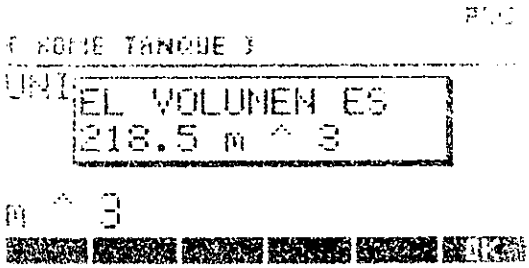
Para calcular el volumen del tanque se oprime la tecla VOL.



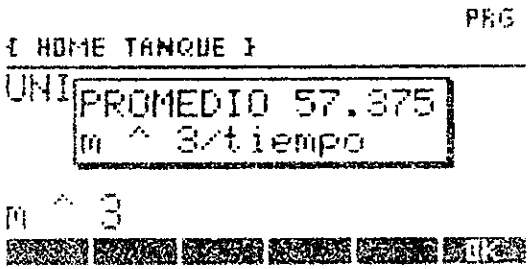
Solicita la unidad que se maneja:



Se pulsa ENTER:

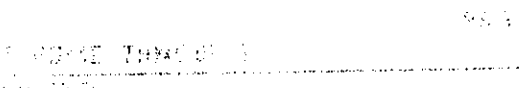


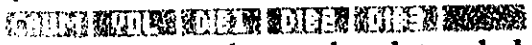
Esto nos indica que por volumen, el tamaño del tanque debe ser mínimo de 218.5 L. Se pulsa la tecla OK:



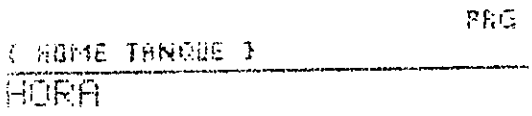
La pantalla indica el promedio que distribuirá el tanque. Pero como la concentración también varía es necesario corroborar.

Para ello se teclea en el menú CAUM

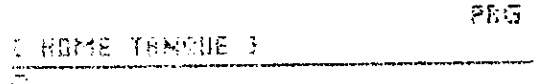




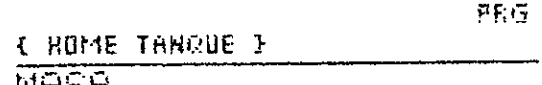
Como ya se teclearon los datos de la hora, se puede acceder a quitar la llave y teclear simplemente HORA



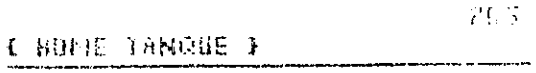
Lo mismo se hace con el flujo Q, solo se teclea Q quitando la llave.



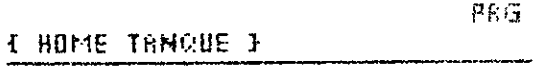
Luego se teclea la masa.



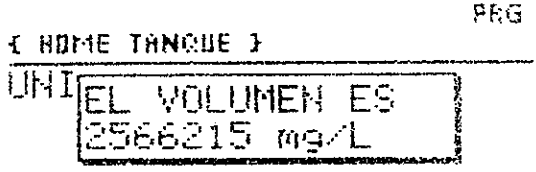
Y así se termina de ingresar los datos Se vuelve a teclear VOL.



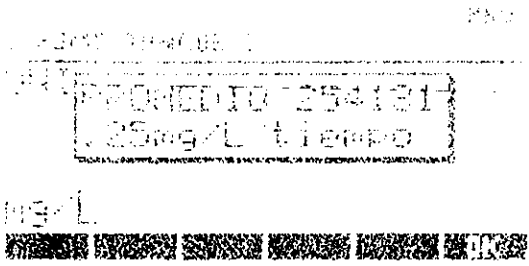
Solicita las unidades:



Se pulsa ENTER y se obtiene el nuevo volumen.



Por masa podemos observar que se requiere más que por volumen. Así se tiene el nuevo volumen del tanque. Se pulsa OK y aparece la concentración promedio.

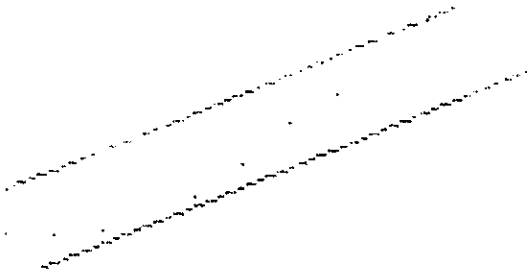


Se tiene también la gráfica del caudal con tan solo pulsar DIB1.

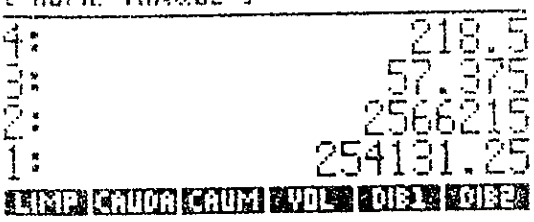
La gráfica del caudal acumulado, con pulsar DIB2.

Y finalmente las tangentes al caudal acumulado, con DIB3.

La variable ΣDAT contiene el resumen del caudal acumulado contra tiempo. Se puede visualizar por medio de su matriz.



Y para limpiar todo de nuevo, se pulsa LIMP.



Capítulo 6

Cálculo de flujos en mallas.

Cuando se realizan cálculos de flujo de agua en tuberías aisladas la ecuación de Bernoulli basta para hacer el balance de pérdidas por fricción. Pero en el caso de las redes, se tienen que hacer iteraciones para saber la cantidad de fluido que pasa por cada tubo. Esto es posible, empleando las siguientes ecuaciones empíricas para el agua:

$$\Delta Ca = \frac{-\Sigma(\frac{\Sigma F}{M})}{\frac{\Sigma F / M}{(\frac{}{Ca})1.85}}$$

donde:

ΔCa es la corrección de flujo en los circuitos

Ca es el caudal que pasa por cada tubería del circuito.

$\frac{\Sigma F}{M}$ es la suma de fricciones por unidad de masa y se definirá enseguida por una correlación empírica.

Esta es la ecuación que define las pérdidas por fricción:

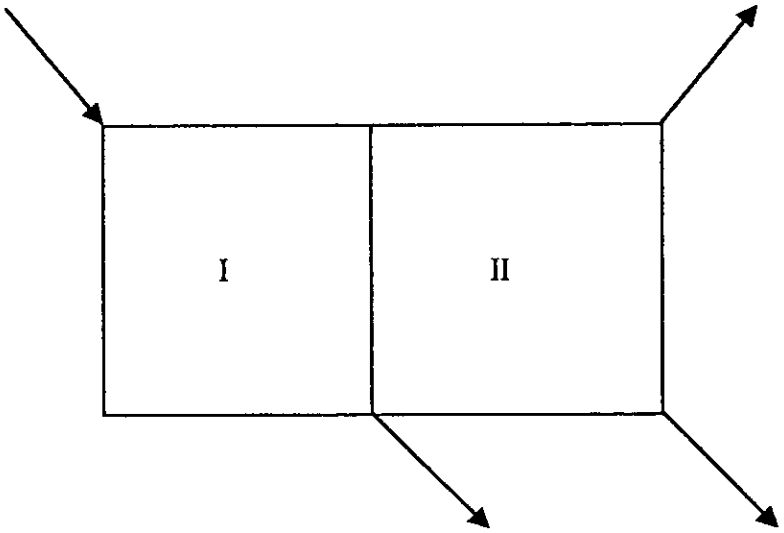
$$\frac{\sum F}{M} = 10.643(\frac{Ca}{C})^{1.852}(\frac{1}{D})^{4.87} L$$

donde :

C es un valor constante que depende de las características de la tubería, y puede consultarse junto con estas ecuaciones en el libro del profesor Antonio Valiente Barderas "Flujo de fluidos". Esta constante toma valores generalmente alrededor de 100 y no se adentrará en su cálculo.

D es el diámetro de la tubería en metros.

L es la longitud de la tubería.



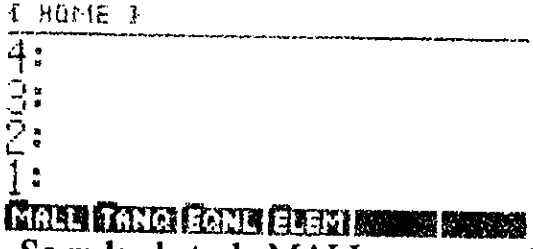
Las correcciones se realizan de la siguiente forma:

$Ca_{(I+1)} = Ca_I + \Delta Ca$

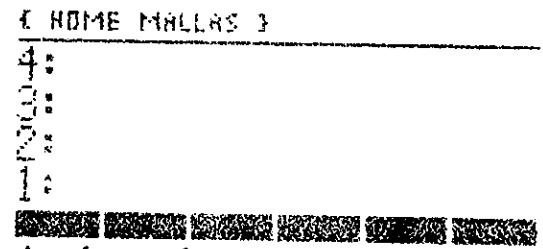
$Ca_{(I+1)} = Ca_I + \Delta Ca_I - \Delta Ca_{II}$

Para comenzar hay que organizar el directorio donde se trabajarán todos los datos.

Desde HOME se teclea [α][α] MALLAS CRDIR [ENTER] y aparece la siguiente imagen:



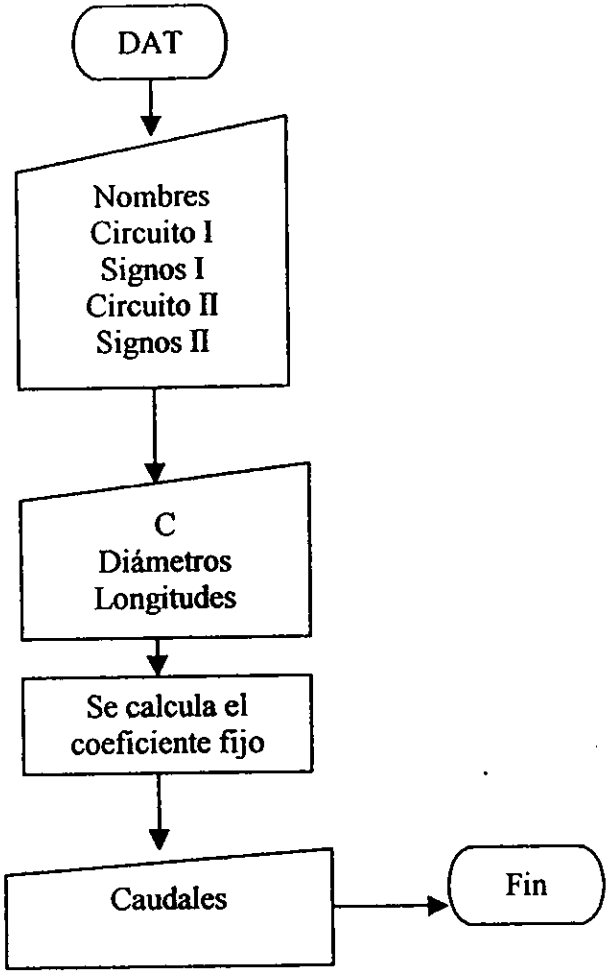
Se pulsa la tecla MALL y aparece la siguiente imagen:



Aquí se teclean todos los programas.

Este es un ejemplo, y si se tuvieran más ciclos, habría que aumentar los datos por cada ciclo. Dada la variedad de arreglos que pueden presentarse en las tuberías, resulta muy difícil hacer un solo programa que los represente a todos, pero con este ejemplo se dan todos los elementos para resolver arreglos más complicados.

DAT: Solicita los datos para comenzar los cálculos.



Circuito I y II son los nombres en lista de los nodos que constituyen a cada circuito.

Signos I y II son el sentido que tiene cada una de las tuberías y se representan con un 1 si van en el sentido del reloj y con un –1 si es contrarreloj. Llevan el mismo orden que circuito I y II.

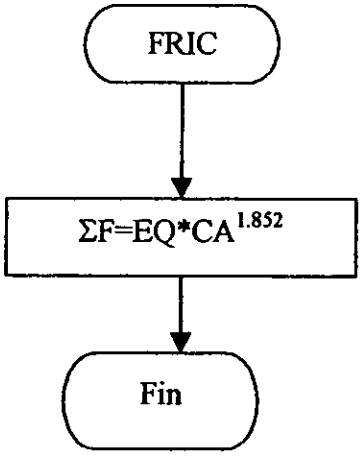
[# 15480d 407.5]

Programa	*Descripción*
« "¿NOMBRES?" "{"	Inicia el programa, muestra la cadena de solicitud de nombres, coloca un inicio de lista.
INPUT STR→ 'NOM'	Solicita la cadena, la convierte de cadena a lista y coloca la variable NOM.
STO "CIRCUITO I"	Almacena la lista en dicha variable y coloca la cadena de entrada de los nombres del circuito I
"{" INPUT STR→	Coloca el inicio de una lista en la cadena, solicita la lista y convierte la cadena a lista.
'CIRI' STO	Coloca la variable CIRI y almacena la lista.
"SIGNOS DE I" "{"	Coloca la cadena de entrada de los signos del circuito I y la cadena de entrada de lista.
INPUT STR→ 'SIGI'	Pide los datos, convierte la cadena en lista y coloca la variable SIGI.
STO "CIRCUITO II"	Almacena en la variable dicha lista y coloca la cadena de entrada de nombres del circuito II

"{" INPUT STR→	Coloca el inicio de la lista en forma de cadena, pide la entrada de datos y convierte la cadena a lista
'CIRII' STO	Coloca la variable CIRII y almacena la lista en dicha variable.
"SIGNOS DE II" "{"	Coloca la cadena que pide los signos del circuito II y coloca la cadena de inicio de lista.
INPUT STR→ 'SIGII'	Solicita los datos convierte la cadena a lista y coloca la variable SIGII
STO "VALOR DE C" ""	Almacena en dicha variable a la lista, coloca la cadena que solicita el valor de C y coloca un caracter nulo
INPUT STR→ INV	Pide la entrada del dato, lo convierte de cadena a número real, y obtiene el inverso.
1.852 ^ 10.643 *	Eleva a la potencia marcada y lo multiplica por el coeficiente marcado.
"¿DIAMETROS?" "{"	Coloca la cadena que interroga los diámetros y coloca la bandera de inicio de lista.

INPUT STR→ INV 4.87	Hace la solicitud de datos, convierte la cadena en lista, hace la inversión y coloca la potencia
^ * "¿LONGITUDES?"	Eleva la potencia mencionada, lo multiplica y coloca la cadena de longitud
"{" INPUT STR→ *	Coloca la cadena de inicio de lista, hace la entrada, convierte la cadena a lista y lo multiplica
'EQ' STO	Coloca la variable EQ y almacena la lista
"¿CAUDALES?" "{"	Coloca las cadenas de entrada de caudales y de inicio de lista
INPUT STR→ 'CA' STO	Hace la entrada de datos, convierte la cadena a lista, coloca la variable CA y almacena los valores en la variable indicada
»	Termina el programa

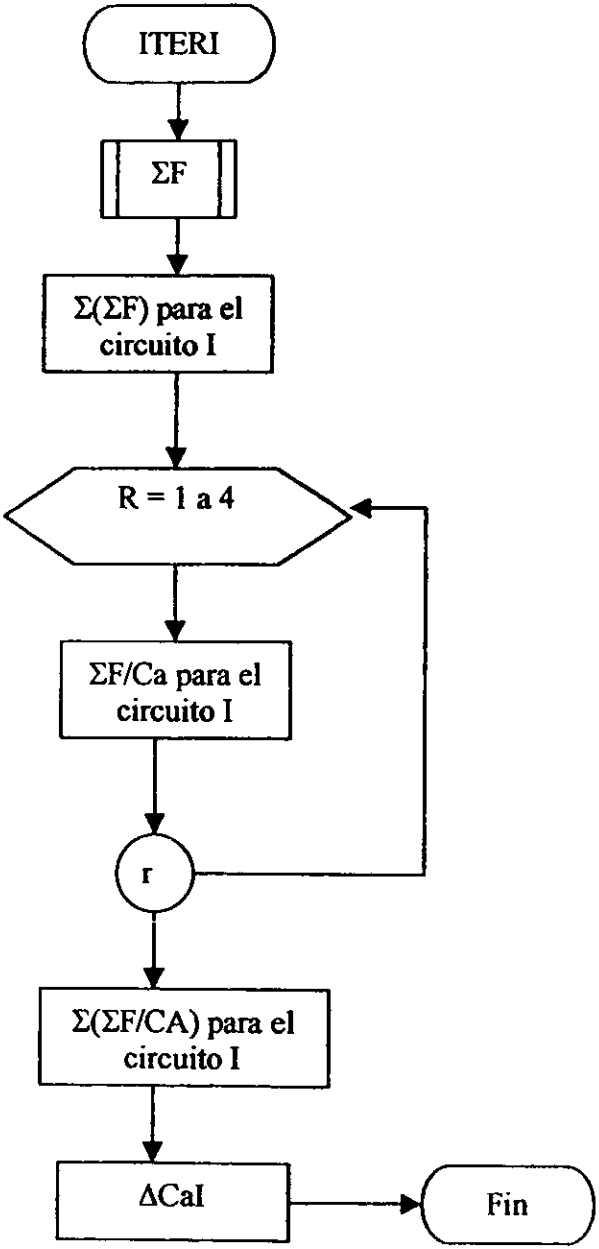
FRIC: A partir del coeficiente fijo calcula las pérdidas por fricción de cada línea dentro de una lista



[# 42554d 58]

Programa	*Descripción*
« CA 1.852 ^ EQ *	Toma los caudales, los eleva a la potencia indicada, invoca los valores almacenados en EQ y los multiplica
'ΣF' STO	Coloca la variable ΣF y almacena los valores.
»	Termina el programa.

ITERI: Calcula la suma de las pérdidas por fricción, las pérdidas por fricción por caudal, su suma y la corrección de caudales para el circuito I.

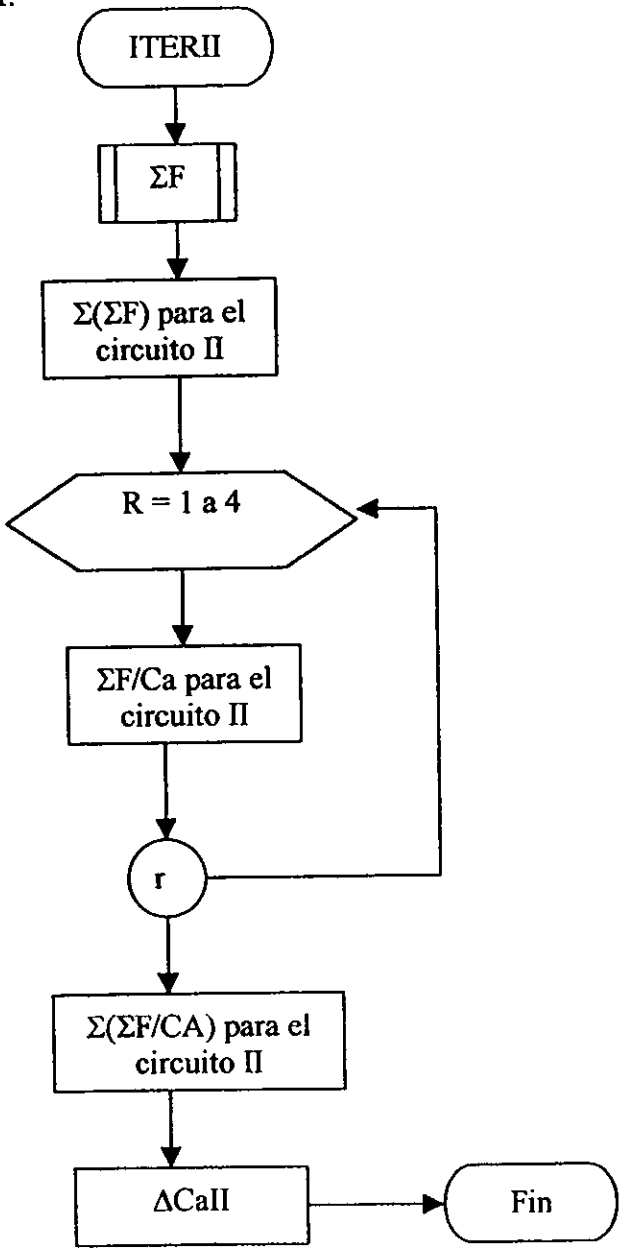


[# 52412d 192]

Programa	*Descripción*
« ΣF NOM	Invoca la variable ΣF y la lista de nombres
« STO 1	Inicia un subprograma, almacena el valor y coloca un uno
» DLT DROP CIRI	Termina el subprograma, lo aplica a toda la lista, tira la lista de resultado e invoca los nombres del circuito I
EVL DUP SIGI ROT *	Evalúa los nombres de la lista de nombres del circuito I, hace una copia, invoca los signos del circuito I, gira el tercer nivel y lo multiplica
ΣLIST NEG SWAP { }	Suma toda la lista, obtiene su inversión de signo, voltea el resultado y coloca una lista nula
1 4	Coloca un uno y un cuatro
FOR r CA NOM CIRI	Inicia un ciclo para la variable local r, llama los caudales y la listas de nombres y circuito I

r GET POS GET +	Invoca el ciclo r, obtiene el nombre del ciclo en el nombre del circuito, obtiene la posición en la lista de nombres, obtiene el caudal y lo suma.
NEXT / DUP 'ΣFCI'	Cierra el ciclo, divide el resultado, hace una copia y coloca la variable ΣFCI
STO ΣLIST / 1.85 /	Almacena en la variable mencionada, suma toda la lista, divide el resultado y lo divide entre el coeficiente marcado.
»	Termina el programa

ITERII: hace lo mismo que ITERI pero con los datos del circuito II.

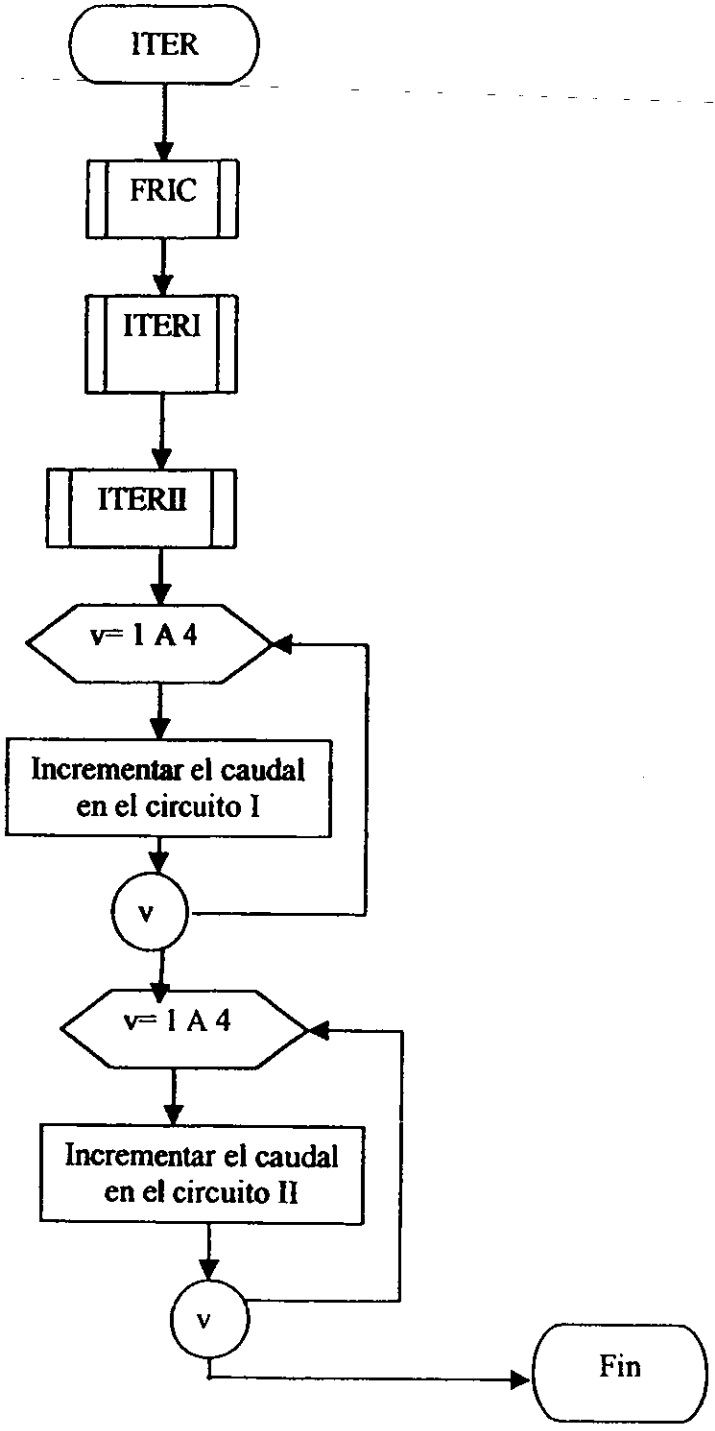


[# 44855d 197]

Programa	*Descripción*
« ΣF NOM	Invoca la variable ΣF y la lista de nombres
« STO 1	Inicia un subprograma, almacena el valor y coloca un uno
» DLT DROP CIRII	Termina el subprograma, lo aplica a toda la lista, tira la lista de resultado e invoca los nombres del circuito II
EVL DUP SIGII ROT *	Evalúa los nombres de la lista de nombres del circuito I, hace una copia, invoca los signos del circuito II, gira el tercer nivel y lo multiplica
ΣLIST NEG SWAP { }	Suma toda la lista, obtiene su inversión de signo, voltea el resultado y coloca una lista nula
1 4	Coloca un uno y un cuatro
FOR r CA NOM	Inicia un ciclo para la variable local r, llama los caudales y la listas de nombres

CIRII r GET POS GET	Llama la lista de nombres del circuito II Invoca el ciclo r, obtiene el nombre del ciclo en el nombre del circuito, obtiene la posición en la lista de nombres, obtiene el caudal
+	Suma el caudal
NEXT / DUP	Cierra el ciclo, divide el resultado y hace una copia
'ΣFCII' STO ΣLIST /	coloca la variable ΣFCII, almacena en la variable mencionada, suma toda la lista y divide el resultado
1.85 /	Divide entre el coeficiente marcado.
»	Termina el programa

ITER: Hace uso de los programas anteriores para corregir los caudales en los circuitos I y II.



[# 5770d 356.5]

Programa	*Descripción*
« FRIC ITERI ITERII	Invoca el programa Fric, IterI e IterII
NOM PURGE 1 4	Invoca los nombres de las corrientes y los borra. Coloca un uno y un cuatro
FOR v NOM CIRI v	Inicia un ciclo para la variable local v, invoca el nombre de las corrientes, luego el nombre de las corrientes del ciclo I y la variable v
GET POS CA OVER GET	Consigue el elemento de la lista en su ciclo, revisa la posición en la lista de nombres, llama la lista de caudales, hace una copia del segundo nivel y obtiene el caudal requerido
SIGI v GET * 4 PICK	Llama los signos del circuito uno, consigue el elemento del ciclo y lo multiplica, copia el objeto del cuarto nivel
+ CA 3 ROLLD 3 DUPN	Suma los objetos, llama al caudal, lo coloca en el nivel 3, hace una copia de los 3 niveles

ABS PUT 'CA' STO	Obtiene el valor absoluto, lo coloca en la lista de caudales y lo almacena nuevamente
SIGN SIGI v ROT PUT	Toma el signo, llama la lista de signos del circuito I, invoca la variable local v, gira desde el tercer nivel y coloca el valor en la lista
'SIGI' STO DROP	Coloca la variable SIGI, almacena en dicha variable y tira el resultado
DROP	Tira el resultado
NEXT SWAP DROP 1	Termina el ciclo, voltea los valores tira el resultado y coloca un uno
4	Coloca un cuatro
FOR v NOM CIRII v	Inicia un ciclo para la variable local v, invoca el nombre de las corrientes, luego el nombre de las corrientes del ciclo II y la variable v
GET POS CA OVER GET	Consigue el elemento de la lista en su ciclo, revisa la posición en la lista de nombres, llama la lista de caudales, hace una copia del segundo nivel y obtiene el caudal requerido
SIGII v GET * 3	Llama los signos del circuito uno, consigue el elemento del ciclo y coloca un tres

PICK + CA 3 ROLLD 3	Copia el nivel tres, suma los dos últimos números, llama el caudal, lo lleva al nivel 3 y coloca un tres
DUPN ABS PUT 'CA'	Hace una copia de los tres últimos niveles, obtiene el valor absoluto, lo coloca en la lista y coloca la variable CA
STO SIGN SIGII v	Almacena la lista en la variable dicha, obtiene el signo, llama la lista de signos del circuito II e invoca la variable v
ROT PUT 'SIGII' STO	Gira los tres últimos objetos, pone el resultado en la lista, y almacena la lista en SIGII
DROP DROP	Tira los 2 últimos resultados
NEXT DROP	Termina el ciclo y tira el resultado
»	Termina el programa

Para mantener limpio el archivo se puede hacer un programa que elimine los archivos que se usaron para un cálculo y que ya no se necesiten;

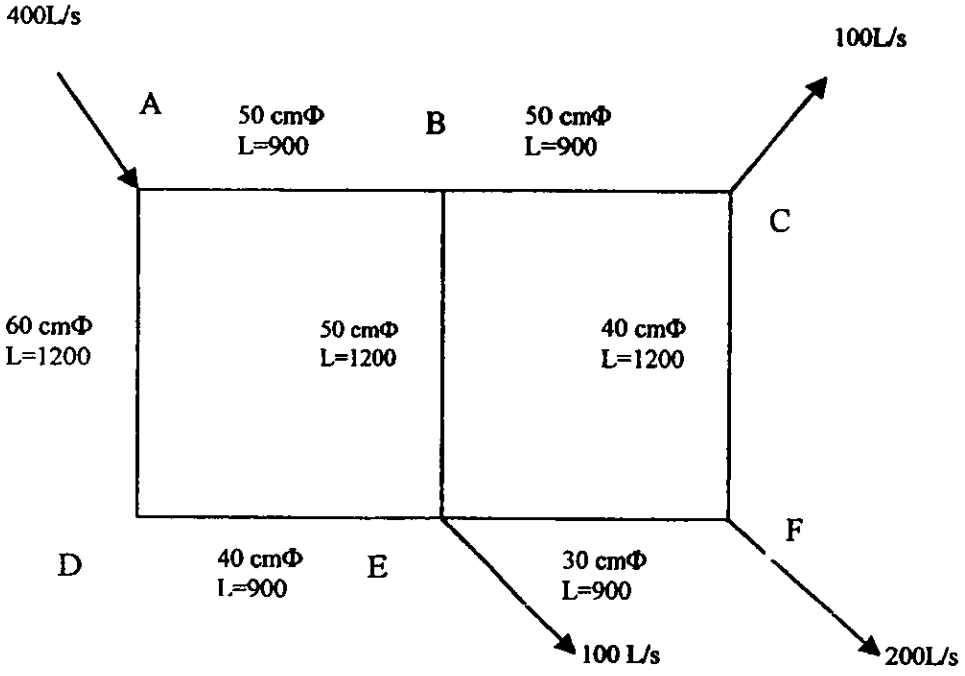
LIMP: Elimina los archivos que no se ocupan.

[# 42117d 97]

« { SIGII SIGI
CIRII CIRI ΣF CA
NOM EQ ΣFCII ΣFCI }
PURGE
»

Práctica

El agua fluye a través del sistema de tuberías mostrado. Determinar los caudales a través de la red. Utilice C=100.



Balance de materia:

Nudo A	$400=C_{AB}+C_{AD}$
Nudo B	$C_{AB}=C_{BC}+C_{BE}$
Nudo C	$C_{BC}=100+C_{CF}$
Nudo D	$C_{AD}=C_{DE}$
Nudo E	$C_{DE}+C_{BE}=100+C_{EF}$
Nudo F	$C_{EF}+C_{CF}=200$

Ahora viene el balance de fricciones: La primera parte es como se teclea el orden en la calculadora, la segunda parte es un cálculo simplificado que realiza la calculadora y almacena en EQ.

$$\left(\frac{\sum F}{M}\right)_{AB} = 10.643\left(\frac{Ca}{C}\right)^{1.852}\left(\frac{1}{D}\right)^{4.87}(L)$$

$$\left(\frac{\sum F}{M}\right)_{BC} = 10.643\left(\frac{Ca}{100}\right)^{1.852}\left(\frac{1}{0.5}\right)^{4.87}(900) = 55.38Ca^{1.852}$$

$$\left(\frac{\sum F}{M}\right)_{AD} = 10.643\left(\frac{Ca}{100}\right)^{1.852}\left(\frac{1}{0.6}\right)^{4.87}(1200) = 30.38Ca^{1.852}$$

$$\left(\frac{\sum F}{M}\right)_{BC} = 10.643\left(\frac{Ca}{100}\right)^{1.852}\left(\frac{1}{0.5}\right)^{4.87}(900) = 55.38Ca^{1.852}$$

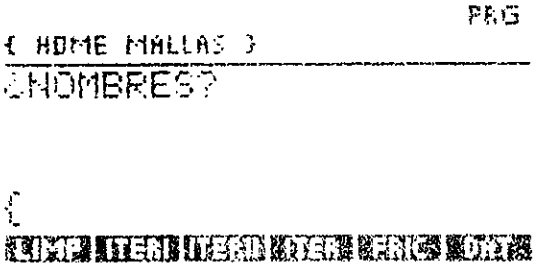
$$\left(\frac{\Sigma F}{M}\right)_{BE} = 10.643\left(\frac{Ca}{100}\right)^{1.852}\left(\frac{1}{0.5}\right)^{4.87}(1200) = 73.83Ca^{1.852}$$

$$\left(\frac{\Sigma F}{M}\right)_{CF} = 10.643\left(\frac{Ca}{100}\right)^{1.852}\left(\frac{1}{0.4}\right)^{4.87}(1200) = 218.88Ca^{1.852}$$

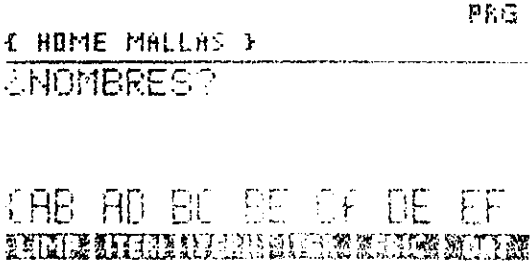
$$\left(\frac{\Sigma F}{M}\right)_{DE} = 10.643\left(\frac{Ca}{100}\right)^{1.852}\left(\frac{1}{0.4}\right)^{4.87}(900) = 164.16Ca^{1.852}$$

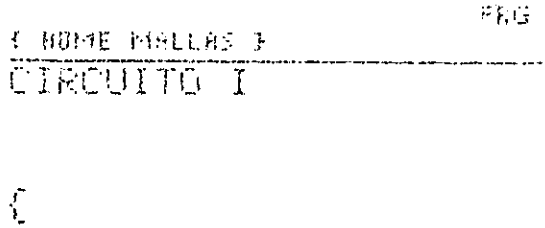
$$\left(\frac{\Sigma F}{M}\right)_{EF} = 10.643\left(\frac{Ca}{100}\right)^{1.852}\left(\frac{1}{0.3}\right)^{4.87}(900) = 666.39Ca^{1.852}$$

Primero se teclea DAT y aparece la imagen:



A continuación se teclean con [α][α] los subíndices en orden del balance de fricciones. **Cf va en minúscula, porque de lo contrario lo interpreta como una instrucción**

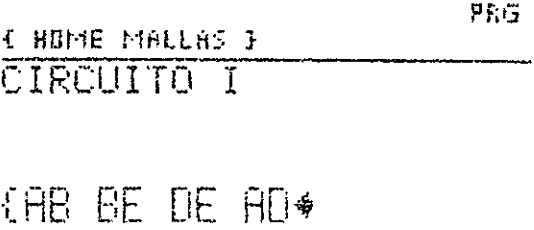




Después de teclear enter, solicita los nombres de las corrientes en el ciclo I.

Ciclo I		Ciclo II	
AB	+	BC	+
BE	+	Cf	+
DE	-	EF	-
AD	-	BE	-

Aquí se toma un orden cualquiera siempre y cuando sea un orden. El signo de las corrientes es positivo si la corriente va en sentido del reloj, y negativo si va a contrarreloj.



Se teclea ENTER

PRG
{ HOME MALLAS }
SIGNOS DE I

{1 1 -1 -1
KCIRC KNOM KLIMP ITER1 ITER2 ITER3
Los signos se teclean como un uno con el signo. En el caso de los negativos, hay que eliminar el espacio entre el signo y el uno con la flecha de corrección.

PRG
{ HOME MALLAS }
CIRCUITO II

{BC CA EF BE
KLIMP ITER1 ITER2 ITER3 KERIC KDAT
Se realiza igual con el ciclo II

PRG
{ HOME MALLAS }
SIGNOS DE II

{1 1 -1 -1
KCIRC KSIGN KCIRC KNOM KLIMP ITER1
Ahora pide el valor de C.

```
                                        PRG
{ HOME MALLAS }
VALOR DE C
```

```
100◆
```

Ahora pide los diámetros. Hay que seguir el orden que se tuvo en los nombres de las tuberías.

```
                                        PRG
{ HOME MALLAS }
¿DIAMETROS?
```

```
{.5 .6 .5 .5 .4 .4 .3◆
```

Lo mismo pasa con las longitudes:

```
                                        PRG
{ HOME MALLAS }
¿LONGITUDES?
```

```
...00 1200 1200 900 900
```

Ahora se estiman los caudales con el balance de materia que ya se hizo.

C_{AB}=200 C_{BE}=50 C_{EF}=150

C_{AD}=200 C_{CF}=50

C_{BC}=150 C_{DE}=200

```
{ NOME MALLAS }
CAUDALES?
```

```
.2 .15 .05 .05 .2 .15
ITERD ITER FERR DAT
```

Así queda lista la calculadora para comenzar a iterar.

En EQ se puede ver la parte fija de la estimación de fricciones que se encuentra en el balance de fricciones.

Ahora se pulsa ITER y comienza el acercamiento.

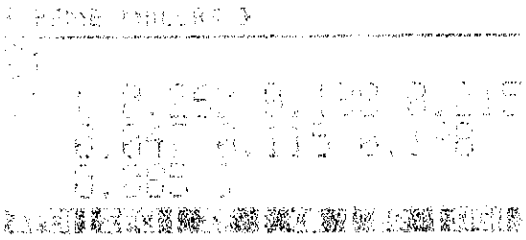
ΣF contiene en orden la suma de fricciones sobre masa.

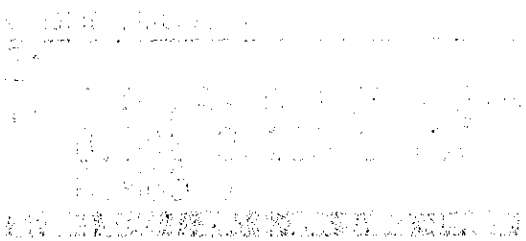
ΣFCI contiene en orden las sumas de fricciones sobre caudal del circuito I.

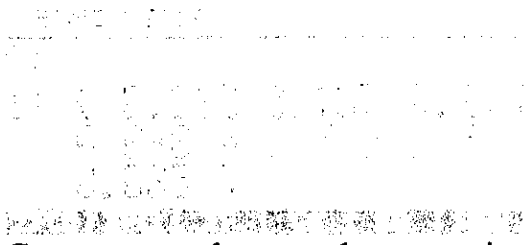
ΣFCII contiene en orden las sumas de fricciones sobre caudal del circuito II

CA tiene los nuevos caudales en el orden que los ingresamos.

```
{ NOME MALLAS }
2:
1:  { 0.253 0.147 0.207
    0.046 0.107 0.147
    0.093 }
```







Como se puede ver a la cuarta iteración ya no se mueve la tercer a posición decimal, con esto se queda ya estable la iteración.

Capítulo 7
Punto pinch

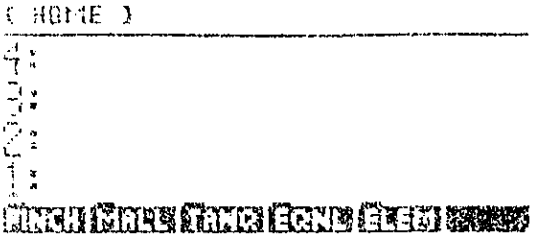
Es común en la industria tener corrientes que se calienten o que se enfríen. El aprovechamiento de las corrientes frías a calentar para enfriar las corrientes calientes, constituye un arte en el arreglo de los intercambiadores de calor. Es muy importante la diferencia de temperaturas mínima que se requiere para evitar que los intercambiadores sean de gran tamaño.

El procedimiento pinch, permite establecer una diferencia de temperaturas mínima, para determinar, cual es la temperatura por la cual se pueden intercambiar calor las corrientes, ya sea por arriba o debajo de esta. Si se cruza esta temperatura, los servicios de enfriamiento y calentamiento, ya no son los mínimos.

Los siguientes programas, dan una muestra de la versatilidad de las listas, para operar datos asociados. Es necesario el subprograma MINIMO que se encuentra en los apéndices.

En primer lugar, se crea el directorio donde residirán los programas y las listas de resultados:

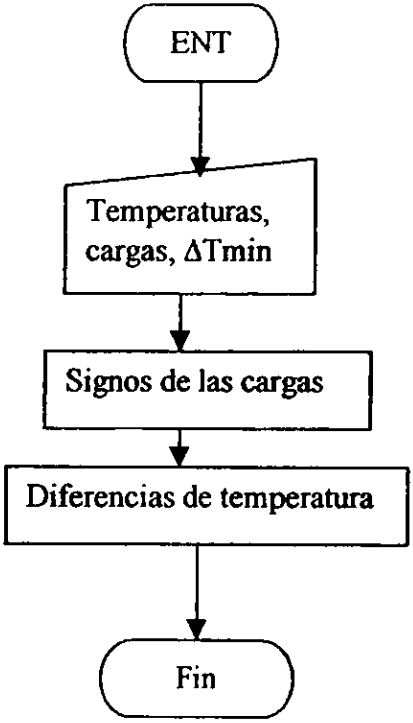
Se teclea: [α][α] PINCH CRDIR [ENTER]



Se pulsa la tecla PINCH y aparece la sig. Imagen:

Ya una vez creado el directorio, se procede a teclear los programas:

ENT: Solicita los datos de entrada y calcula lo referente a las temperaturas.



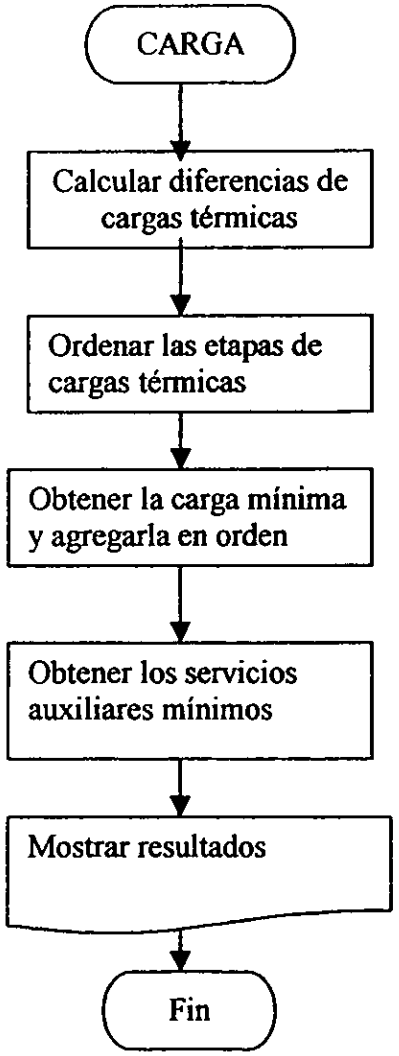
[# 19018d 379.5]

Programa	Descripción
«	Inicia el programa
"Temperatura inicial"	Coloca el mensaje
"{" INPUT STR→ 'L1'	Coloca el inicio del objeto a solicitar, hacc la solicitud, convierte la cadena a objeto y coloca la variable L1
STO	Almacena el dato en la variable

"Temperatura final"	Coloca el texto
"{" INPUT STR→'L2'	Coloca el inicio del objeto a solicitar, hace la solicitud, convierte la cadena a objeto y coloca la variable L2
STO "Carga Térmica"	Almacena el dato en la variable y coloca el texto
"{" INPUT STR→ 'L3'	Coloca el inicio del objeto a solicitar, hace la solicitud, convierte la cadena a objeto y coloca la variable L3
STO L1 L2 - SIGN L3	Almacena en la variable, invoca las listas 1 y 2, las resta, obtiene el signo e invoca la lista L3
* 'L3' STO "ΔT" ""	Multiplica las listas, almacena el resultado en L3 y coloca el texto y comienzo de la solicitud
INPUT STR→ 'ΔT' STO	Hace la solicitud, convierte la cadena a objeto y almacena en ΔT
L1 L2 - SIGN 2 / ΔT	Invoca las listas L1y L2, las resta, solicita el signo, divide entre 2 e invoca ΔT
* 'L5' STO L1 L5 –	Multiplica, almacena en L5, invoca L1 y L5 y los resta.
'L4' STO L2 L5 –	Almacena en L4, invoca L2 y L5 y los resta
'L5' STO L4 L5 +	Almacena en L5 invoca L4 y L5 y los suma

SORT REVLIST 'L6'	Ordena la lista, invierte el orden y coloca la variable L6
STO L6 ΔLIST NEG	Almacena la lista, invoca L6, obtiene la diferencia de los elementos de la lista y les cambia el signo
'L7' STO	Almacena en L7
»	Termina el programa

CARGA: Calcula lo referente a las cargas térmicas, determina los servicios mínimos de calentamiento y enfriamiento, así como la ubicación del punto pinch.



[# 11218d 630]

Programa	Descripción
« 0 'H' STO L7 0 *	Inicia el programa, coloca un cero y lo almacena en H, invoca a L7 y lo multiplica por cero.
'L8' STO 1 L4 SIZE	Almacena la lista en L8, coloca un uno, invoca la lista 4 y toma su tamaño
FOR r L4 r GET L5	Inicia el ciclo para r, invoca L4, obtiene el elemento r de la lista e invoca a L5
r GET MINIMO 'Max'	Obtiene el elemento r de la lista, ordena de menor a mayor los números cono el subprograma MINIMO y coloca la variable Max
STO 'Min' STO L6	Almacena en la variable, almacena en Min e invoca a L6
« Max ≤	Inicia el subprograma, invoca a Max y evalúa si es menor igual que
» EVL L6	Termina el subprograma y evalua para toda la lista, invoca a L6
« Min ≥	Inicia el subprograma, invoca a Min y evalúa si es mayor igual que

» EVL	Termina el subprograma y evalua para toda la lista
« AND	Inicia el subprograma, ejecuta la condición y.
» DLT DUP SIZE	Termina el subprograma, evalúa para ambas listas, hace un duplicado, obtiene el tamaño.
WHILE DUP2 GET	Inicia un ciclo condicional, hace una copia de los dos últimos elementos y obtiene el elemento de la lista
0 ==	Averigua si es igual a cero
REPEAT 1 -	Ejecuta si la condición es verídica, resta un uno.
END DUP2 1 - 1	Termina el ciclo condicional, hace una réplica de los dos últimos objetos, resta un uno y coloca un uno
SWAP SUB 3 ROLLD 1	Voltea, obtiene la sublista, lleva el resultado al nivel 3 y coloca un uno
+ OVER SIZE SUB +	Suma el uno, hace una copia de la lista, obtiene su tamaño, obtiene la sublista y encadena las listas
L3 r GET * L7 * L8	Invoca a L3 y obtiene el elemento r, multiplica la lista, invoca a L7 multiplica e invoca a L8
« +	Inicia el subprograma y suma.

» DLT 'L8' STO	Termina el subprograma y evalúa para las dos listas, almacena en L8
NEXT L8 2 OVER	Cierra el ciclo para r, invoca a L8, coloca un dos y hace una copia de la lista.
SIZE	Obtiene el tamaño.
FOR t DUP t 1 –	Inicia el ciclo para t, hace una copia, coloca t y le resta un uno
GET OVER t GET + t	Obtiene el elemento, hace una copia del nivel 2, invoca a t, obtiene el elemento t de la lista, lo suma , invoca a t
SWAP PUT	Voltea los dos últimos elementos, y coloca el resultado en la lista.
NEXT 'L9' STO L9	Cierra el ciclo para t, almacena la lista en L9, invoca L9
SORT HEAD DUP 0 ≤	Ordena la lista, obtiene el primer elemento de la lista, hace una copia y averigua si es mayor que cero o igual.
« 'H' STO L9	Inicia el subprograma, almacena en H e invoca a L9
« H -	Inicia el subprograma, invoca H y lo resta.
» EVL	Termina el subprograma y evalúa para toda la lista
» IFT 'L10' STO	Termina el subprograma y evalúa si la condición es

	cierta, almacena en L10
"Servicio de calentamiento "	Coloca el texto
H ABS →STR + MSGBOX	Invoca a H, obtiene el absoluto, lo convierte a cadena, lo agrega al texto y da el mensaje en pantalla.
"Servicio de enfriamiento "	Coloca el texto
L10 DUP SIZE GET	Invoca a L10, hace una copia, obtiene el tamaño y consigue el último elemento
→STR + MSGBOX	Lo convierte a cadena, lo agrega al texto y da el mensaje en pantalla
»	Termina el programa

LIMP: Borra los datos que ya no se usarán.

Programa	Descripción
« { L10 L9 H Min Max L8 L7 L6 L4 L5	Coloca la lista de variables
ΔT L3 L2 L1 } PURGE	Borra las variables.
»	

Práctica

Se tienen las siguientes corrientes con las siguientes temperaturas:

	Te (°C)	Ts (°C)	Fc (kW/°C)
1	250	120	1000
2	200	100	4000
3	90	150	3000
4	130	190	6000

Y se quiere usar una ΔT mínima de 10 °C.

Determinar el punto pinch y las cargas térmicas.

Se tiene la pantalla de la calculadora como sigue:

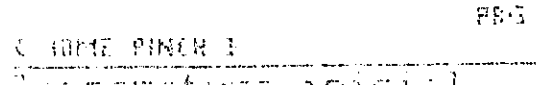
```
{ HOME PINCH }
4:
3:
2:
1:
LINP CARG ENT
```

Se pulsa la tecla ENT:

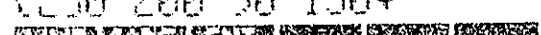
```
                                PRG
{ HOME PINCH }
Temperatura inicial

{
LINP CARG ENT
```

Se teclean las temperaturas de entrada;

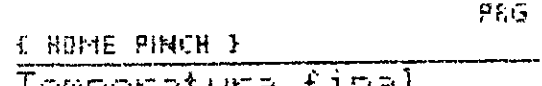


{250 200 90 130◆



Y se teclea ENTER

Aparece la siguiente pantalla:

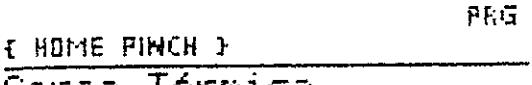


{◆

Se teclean las temperaturas finales:

{120 100 150 190

Se pulsa ENTER y aparece la siguientes pantalla:



{

Se teclean las cargas térmicas en múltiplos de mil, para no manejar tantos ceros:

```
                                    PRG
{ HOME PINCH }
-------------------------------------
Carga Térmica
```

```
{1 4 3 6
LE  L1  LIMP CARG ENT
```

Se teclea ENTER y aparece:

```
                                    PRG
{ HOME PINCH }
-------------------------------------
ΔT
```

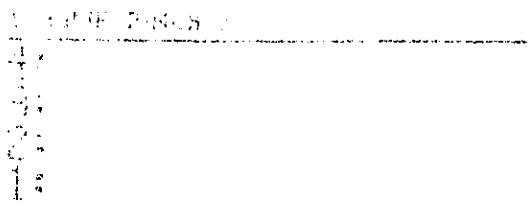
```
4
LE  L2  L1  LIMP CARG ENT
```

Se teclea el dato:

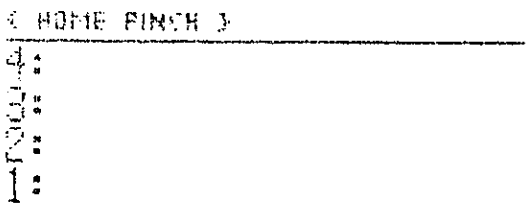
```
                                    PRG
{ HOME PINCH }
-------------------------------------
ΔT
```

```
10
LE  L2  L1  LIMP CARG ENT
```

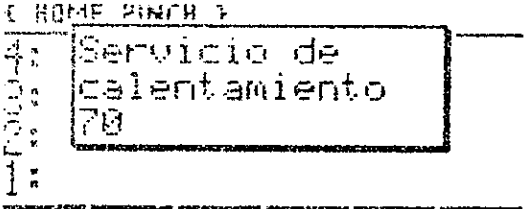
Se pulsa ENTER y aparece la imagen así:



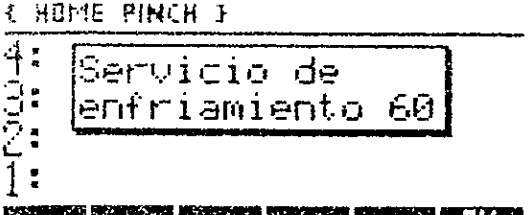
Se pulsa NXT y aparece la tecla CARGA



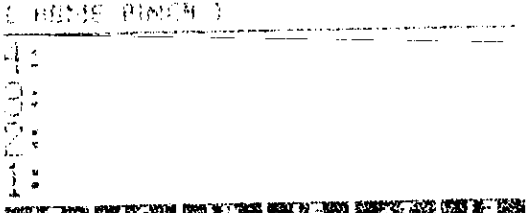
Se pulsa CARG y aparece la siguiente imagen;



Se pulsa OK y la imagen cambia a:



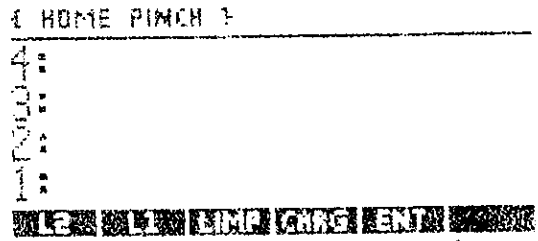
Se pulsa OK y aparece el siguiente menú:



Se van consultado cada una de las listas, empezando por L1 a L5 y se organizan en una tabla:

L1	L2	L3	L4	L5
250	120	1	245	115
200	100	4	195	95
90	150	-3	95	155
130	190	-6	135	195

(Si la lista no aparece en el menú, teclee el botón NXT hasta que aparezca).



Así se tiene que las listas contienen:

L1 Las temperaturas de entrada

L2 La temperatura de salida

L3 Las cargas térmicas con su signo correspondiente.

L4 La temperatura de entrada desplazada por la mitad de ΔT

L5 La temperatura de salida desplazada por la mitad de ΔT

Se pulsa la tecla DEL para borrar la pantalla y se listan el resto de las listas:

L6	L7	L8	L9	L10
245	50	50	50	120
195	0	0	50	120
195	40	-40	10	80
155	20	-80	-70	0
135	20	40	-30	40
115	20	20	-10	60
95	0	0	-10	60
95				

El contenido de estas listas es el siguiente:

L6 Las temperaturas modificadas por el ΔT

L7 Las diferencias de temperaturas.

L8 Las cargas en los diferentes intervalos.

L9 La acumulación de cargas sin servicios.

L10 La acumulación de cargas en los servicios.

Salvo la lista 10, los ceros nos indican que no hubo diferencia de temperatura, con lo cual no hay cargas térmicas.

En la lista 10 el cero nos indica la(s) posición(es) del punto pinch, donde nos divide las temperaturas en bloques para no cruzar las temperaturas para un intercambiador. En este caso es la temperatura 155. En caso que nuestros diseños crucen al punto pinch, los servicios no serán los mínimos.

1	2		3	4	70
		245			120
		195			120
		195			80
		155			0
		135			40
		115			60
		95			60
		95			

Capítulo 8
Propiedades Residuales

El siguiente es un ejemplo de cómo las fórmulas también se pueden programar en la HP 48G/GX.

En ciertos procesos termodinámicos se pueden hacer aproximaciones por medio de las propiedades residuales

$\Delta S=\Delta S_1{}^R+\Delta S^*+\Delta S_2{}^R \qquad \Delta S^*=\int Cp\frac{dT}{T}-R*LN\frac{P2}{P1}$

$$\frac{\Delta S^R}{R}=\frac{(\Delta S^R)^0}{R}+w\frac{(\Delta S^R)^1}{R}$$

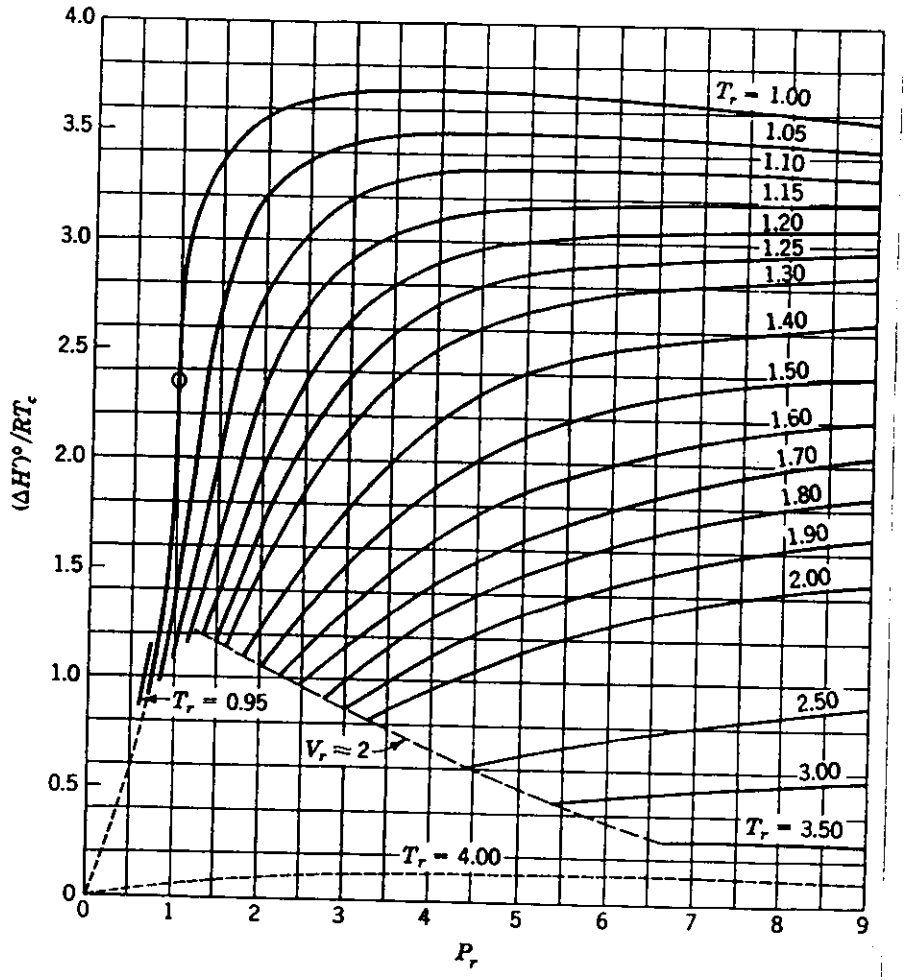
$\Delta H=-\Delta H^R+\Delta H^*$

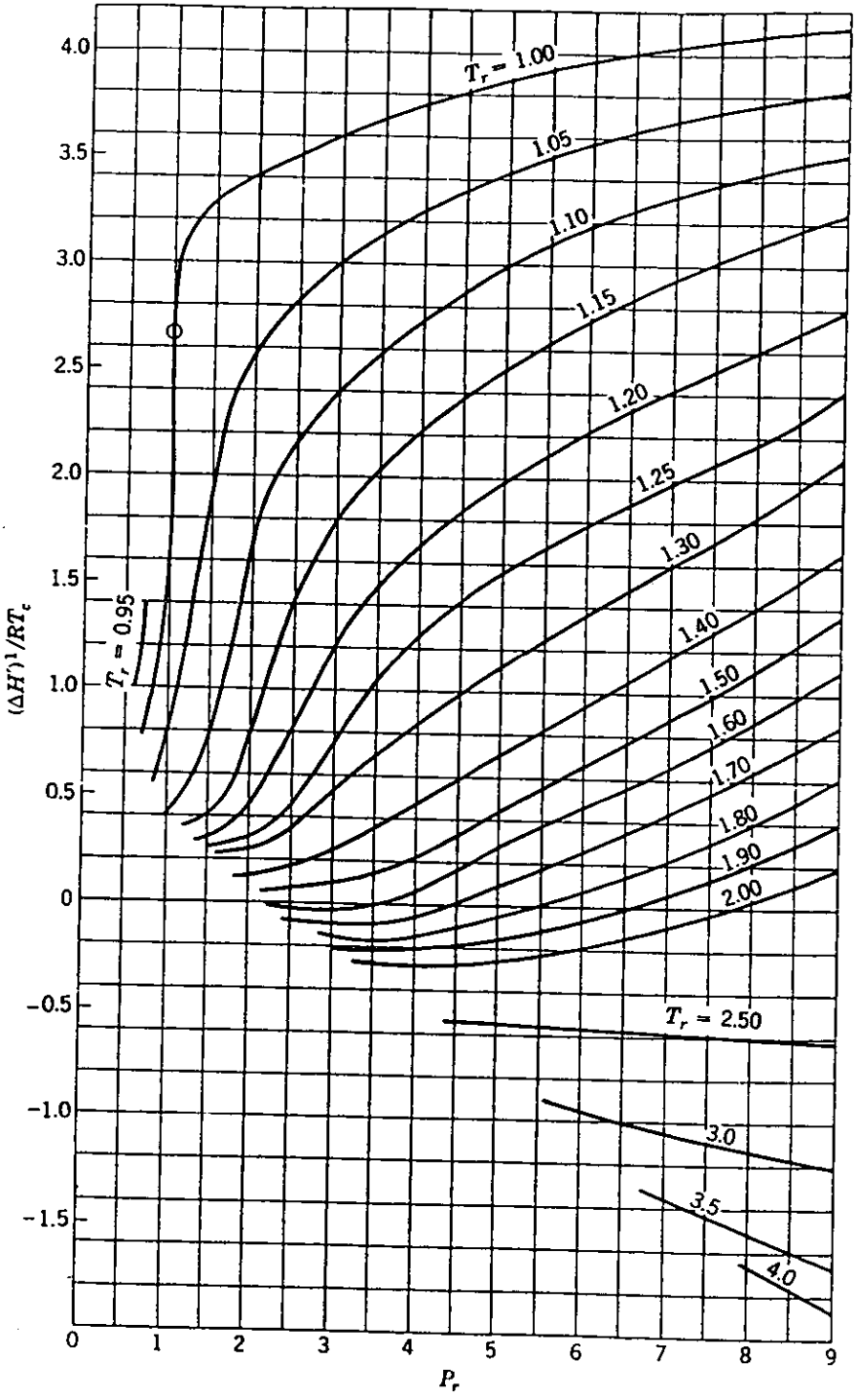
$$\frac{\Delta H^R}{RTc}=\frac{(\Delta H^R)^0}{RTc}+w\frac{(\Delta H^R)^1}{RTc}$$

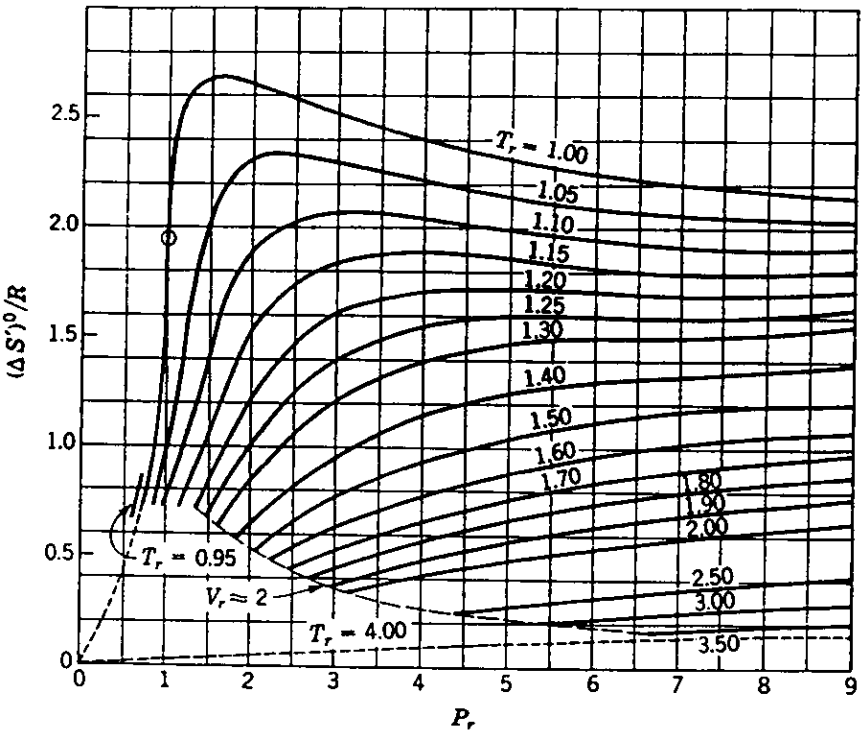
donde w es el factor acéntrico de Pitzer

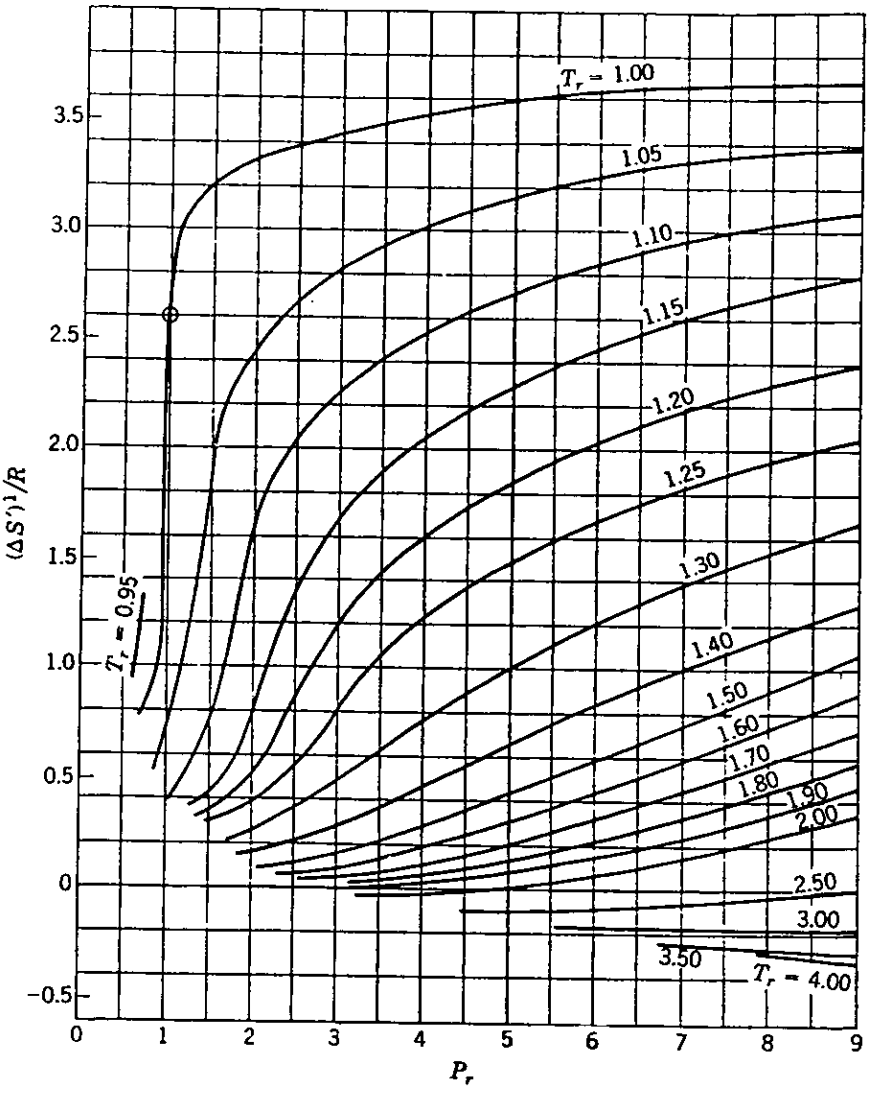
El algoritmo que sigue es un ejercicio donde hay un proceso adiabático y se quiere saber el cambio de entropía.

Estas son las tablas en las que se resumen las propiedades residuales. En un principio se iba a crear una correlación que calculara automáticamente las propiedades residuales, pero las correlaciones no lograron reproducir fielmente los coeficientes. En los apéndices se muestran algunas gráficas de los resultados a este intento:


4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0
0
1
2
3
4
5
6
7
8
9
$(\Delta H^\circ)/RT_c$
P_r
$T_r = 1.00$
1.05
1.10
1.15
1.20
1.25
1.30
1.40
1.50
1.60
1.70
1.80
1.90
2.00
2.50
3.00
$T_r = 3.50$
$T_r = 0.95$
$V_r \approx 2$
$T_r = 4.00$

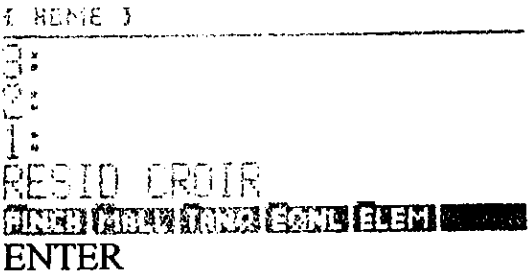

$(\Delta H')^1/RT_c$
P_r
$T_r = 0.95$
$T_r = 1.00$
1.05
1.10
1.15
1.20
1.25
1.30
1.40
1.50
1.60
1.70
1.80
1.90
2.00
$T_r = 2.50$
3.0
3.5
4.0
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0
−0.5
−1.0
−1.5
0
1
2
3
4
5
6
7
8
9

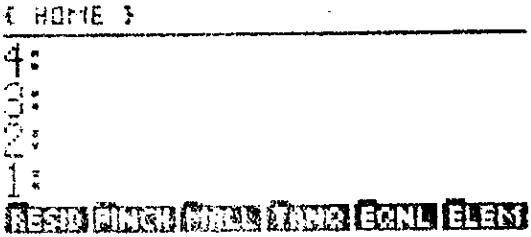

$(\Delta S')^0/R$
P_r
T_r = 1.00
1.05
1.10
1.15
1.20
1.25
1.30
1.40
1.50
1.60
1.70
1.80
1.90
2.00
2.50
3.00
3.50
T_r = 0.95
$V_r \approx 2$
T_r = 4.00
0
0.5
1.0
1.5
2.0
2.5
0
1
2
3
4
5
6
7
8
9


T_r = 1.00
1.05
1.10
1.15
1.20
1.25
1.30
1.40
1.50
1.60
1.70
1.80
1.90
2.00
2.50
3.00
3.50
T_r = 4.00
T_r = 0.95
$(\Delta S')^1/R$
P_r
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0
−0.5
0
1
2
3
4
5
6
7
8
9

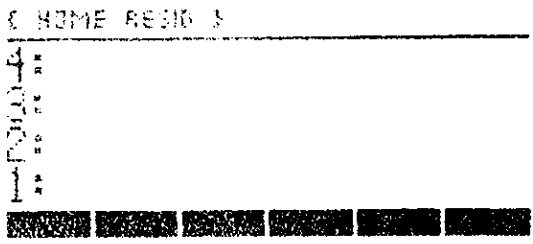
Se prepara el directorio para poner los programas relacionados con las propiedades residuales.

[α][α]RESID CRDIR

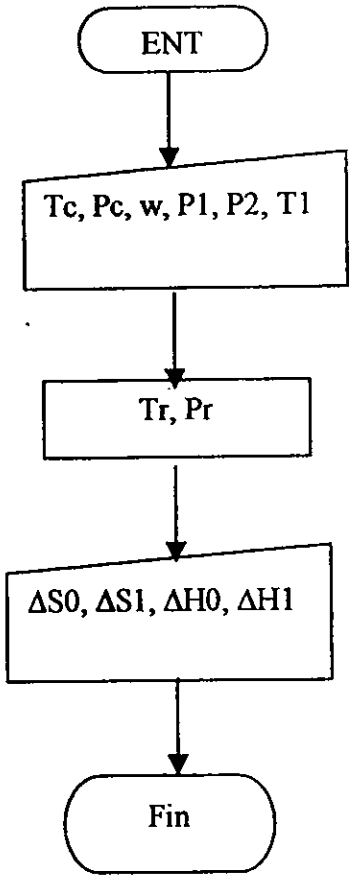




Se pulsa la tecla RESID y ya se está listo para ingresar los programas.



ENT: Solicita los datos necesarios para hacer los cálculos, calcula las propiedades reducidas y las muestra para solicitar los coeficientes viriales de las tablas anteriores.



[#45226d 580]

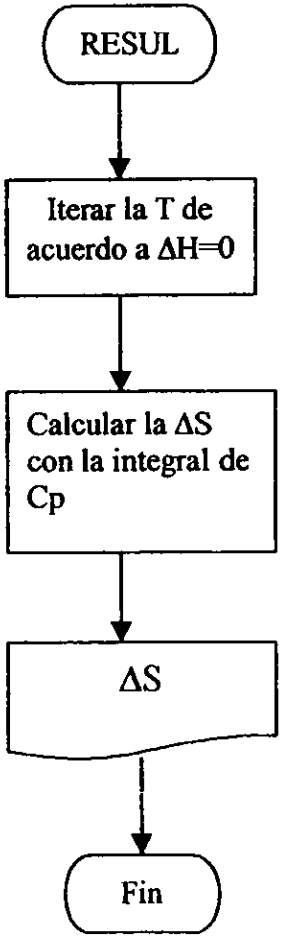
Programa	Descripción
« "Tc" "" INPUT	Inicia el programa, coloca los textos y solicita el dato.
STR→ 'TC' STO "Pc"	Convierte la cadena a número real y la almacena en TC, coloca el texto
"" INPUT STR→ 'PC'	Coloca el inicio de la solicitud, hace la entrada de datos, lo convierte de cadena a número real y coloca la variable PC
STO "w" "" INPUT	Almacena en la variable, coloca los textos de entrada de datos y hace la solicitud de datos
STR→ 'W' STO	Lo convierte de cadena a número real y almacena en W.
"Coeficiente A de Cp"	Coloca el texto
"" INPUT STR→ 'A'	Coloca la cadena de entrada, hace la solicitud de datos, convierte la cadena a número y coloca la variable A.
STO	Almacena el dato en la variable
"Coeficiente B de Cp"	Coloca el texto
"" INPUT STR→ 'B'	Coloca el texto de entrada, hace la solicitud, convierte la cadena a número real y coloca la variable B

STO	Almacena en la variable
"Coeficiente C de Cp"	Coloca el texto
"" INPUT STR→ 'C'	Coloca el texto de entrada, hace la solicitud, convierte la cadena a número real y coloca la variable C
STO "T1" "" INPUT	Almacena en la variable, coloca los textos de entrada y hace la solicitud de datos.
STR→ 'T1' STO "P1"	Convierte la cadena a número real, lo almacena en T1 y coloca el texto.
"" INPUT STR→ 'P1'	Coloca el texto de entrada, hace la solicitud de datos, convierte la cadena a número real y coloca la variable P1
STO "P2" "" INPUT	Almacena en la variable, coloca los textos de entrada y hace la solicitud.
STR→ 'P2' STO T1 TC	Convierte la cadena a número real, almacena el dato en P2, invoca a T1 y TC
/ 2 FIX →STR "Tr "	Divide los números, fija los decimales en 2, convierte el número a texto y coloca el texto Tr
SWAP + "	Voltea las cadenas y las junta, coloca una cadena de cambio de renglón.(Se usa la tecla verde con el punto)
" + P1 PC	Termina el cambio de renglón, lo agrega a la

	cadena, invoca a P1 y PC
/ →STR "Pr " SWAP +	Divide los números, el resultado lo convierte en cadena, coloca el texto Pr, lo voltea y junta las cadenas.
+ "	Junta las cadenas e inicia un salto de renglón.
" + DUP DUP DUP	Termina el salto de renglón, lo agrega a la cadena y hace tres copias.
STD "ΔH°" + ""	Fija el modo estándar de los números, coloca el texto, lo agrega a la cadena y coloca el texto de inicio de la solicitud.
INPUT STR→ SWAP	Hace la entrada, convierte la cadena a número y voltea el resultado
"ΔH'" + "" INPUT	Coloca el texto, lo agrega a la cadena, coloca el texto de entrada y hace la solicitud de datos.
STR→ W * + R * TC *	Convierte la cadena en número, lo multiplica por w, suma el número anterior, multiplica por R, y luego por Tc
'ΔH' STO "ΔS°" + ""	Almacena en ΔH, coloca el texto y lo agrega a la cadena, coloca el texto de entrada
INPUT STR→ SWAP	Hace la solicitud de datos, convierte la cadena a número

	real y voltea el resultado
"ΔS'" + "" INPUT	Coloca el texto, lo agrega a la cadena, coloca la cadena de entrada y hace la solicitud de datos
STR→ W * + R * 'ΔS'	Convierte la cadena a número, lo multiplica por w, se suma el resultado, se multiplica por R y se coloca la variable ΔS
STO	Almacena en la variable
»	Termina el programa

RESUL: calcula la diferencia de entropía, en base a que el cambio de entalpía para un proceso adiabático es cero.



[# 37236d 290.5]

Programa	Descripción
« 'ΔH=R*(A*(T2-T1)+	Inicio del programa, comienzo de la ecuación
B*(T2^2-T1^2)+C*(T2	Continuación de la ecuación
^3-T1^3))' 'T2' T1	Terminación de la ecuación, colocación de T2 como variable a iterar, T1 como semilla a iterar
ROOT →STR "T2 "	Itera el valor de T2 almacenando el resultado, convierte el resultado a cadena, coloca el texto T2
SWAP + MSGBOX 'R*(∫	Voltea las cadenas, las junta, despliega el mensaje y comienza la integral
(T1,T2,A*INV(T)+B+C	Coloca los límites de integración e incluye la ecuación a integrar
*T,T)-LN(P2/P1))'	Termina la ecuación con la variable a integrar y termina la ecuación.
→NUM ΔS - "ΔS "	Convierte la ecuación a número, le resta ΔS y coloca el texto.
OVER →STR + MSGBOX	Hace una copia del segundo nivel, lo convierte a cadena, junta las cadenas y despliega el mensaje
»	Termina el programa

LIMP: borra las variables que no se utilizan al final de hacer el cálculo.

[# 31734d 88]

Programa	Descripción
« { A B C PC TC W	Inicia el programa
T1 T2 P1 P2 ΔS ΔH }	Coloca las variables a borrar
PURGE	Borra las variables
»	Termina el programa

Práctica

Se tiene gas propano a 20 bar y 400K se estrangula en un proceso de flujo en estado estable hasta un bar. Estime el cambio de entropía del propano debido a este proceso y en su estado final, el propano puede considerarse como gas ideal. Para resolver este problema manualmente se siguen exactamente los pasos que están en los diagramas de flujo de los programas ENT y RESUL.

Se tienen los siguientes datos:

Tc=369.8 K Cp=1.213+28.785E-3*T-8.324E-6*T^2

Pc=42.5 bar

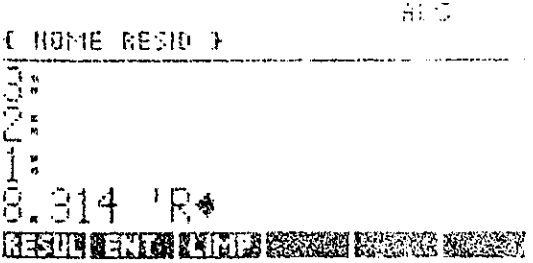
W=0.152

R=8.314 J/mol K

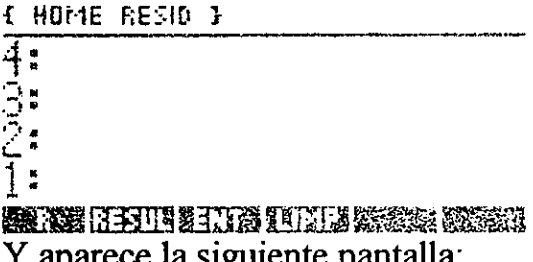
Como primer punto las unidades determinan el valor de R, así que ese dato es el primero en ingresarse a la calculadora y

solo se tecleará la primera vez, o las veces que se utilicen nuevas unidades.

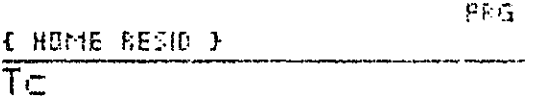
Se teclea 8.314 'R' [STO]



Ya lista la variable R, se introducen los datos pulsando la tecla ENT.



Y aparece la siguiente pantalla:



Se teclea el valor de la temperatura crítica y ENTER:

PRG
{ NOME RESID }
Tc

369.8
Solicita el valor de la presión crítica, se ingresa el valor y enseguida ENTER

PRG
{ NOME RESID }
Pc

42.5
Enseguida solicita el factor acéntrico de Pitzer.

PRG
{ NOME RESID }
ω

.152
Solicita el coeficiente A de Cp

PRG
{ NOME RESID }
Coeficiente A de Cp

1.213
Solicita el coeficiente B de Cp:

PRG
{ HOME RESID }
Coeficiente B de Cp

28.785E-3
[illegible]
Solicita el coeficiente C de Cp:

PRG
{ HOME RESID }
Coeficiente C de Cp

-8.824E-6
[illegible]
Pide la temperatura inicial:

PRG
{ HOME RESID }
T1

400◆
[illegible]
Pide la presión inicial:

PRG
{ HOME RESID }
P1

20◆
[illegible]
Pide la presión final:

PRG

{ HOME RESID }

P2

1◆

Despliega el valor de las propiedades reducidas y solicita el valor de ΔH^0:

PRG

{ HOME RESID }

Tr 1.08
Pr 0.47
ΔH°

.5◆

Solicita el valor de ΔH^1:

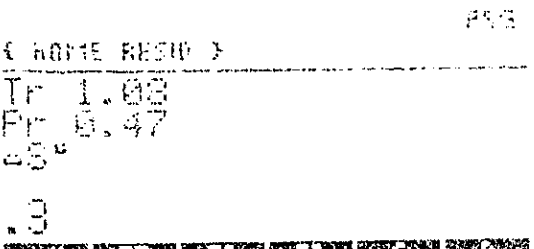
PRG

{ HOME RESID }

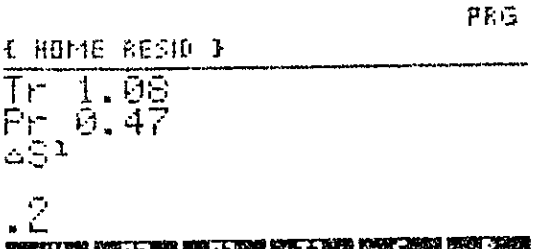
Tr 1.08
Pr 0.47
ΔH¹

.4◆

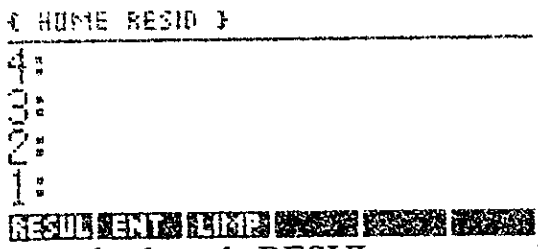
Solicita el valor de ΔS^0:



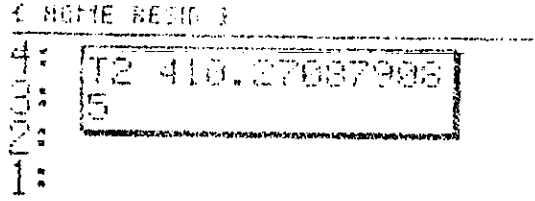
Y por último solicita ΔS^1:



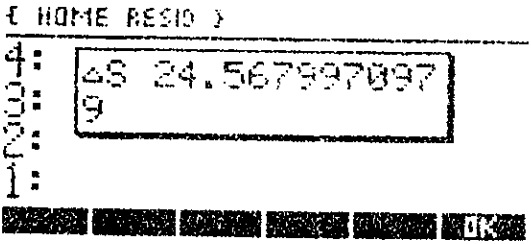
Ahora ya está listo para calcular las propiedades termodinámicas. Se teclea NXT hasta visualizar en el menú a RESUL.



Se pulsa la tecla RESUL y aparece la siguiente pantalla:



Se pulsa OK y aparece:



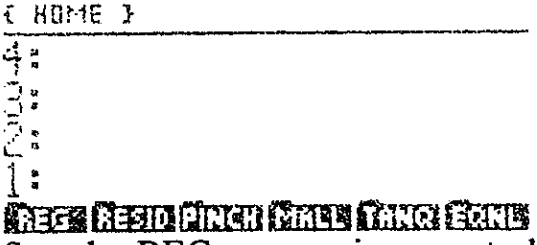
Capítulo 9
Regresiones matemáticas.

Es muy necesario a lo largo de la carrera, ajustar datos experimentales a una ecuación matemática. Hewlett Packard incorporó una potente herramienta en un comando llamado LSQ, el cual nos permite hacer regresiones en una forma sencilla y rápida.

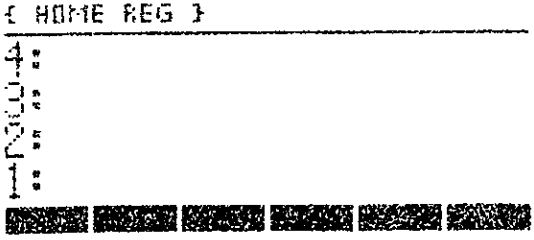
El siguiente algoritmo consiste en la interfase de lenguaje entre el comando mencionado y el usuario.

Se procede a crear un directorio para colocar los programas:

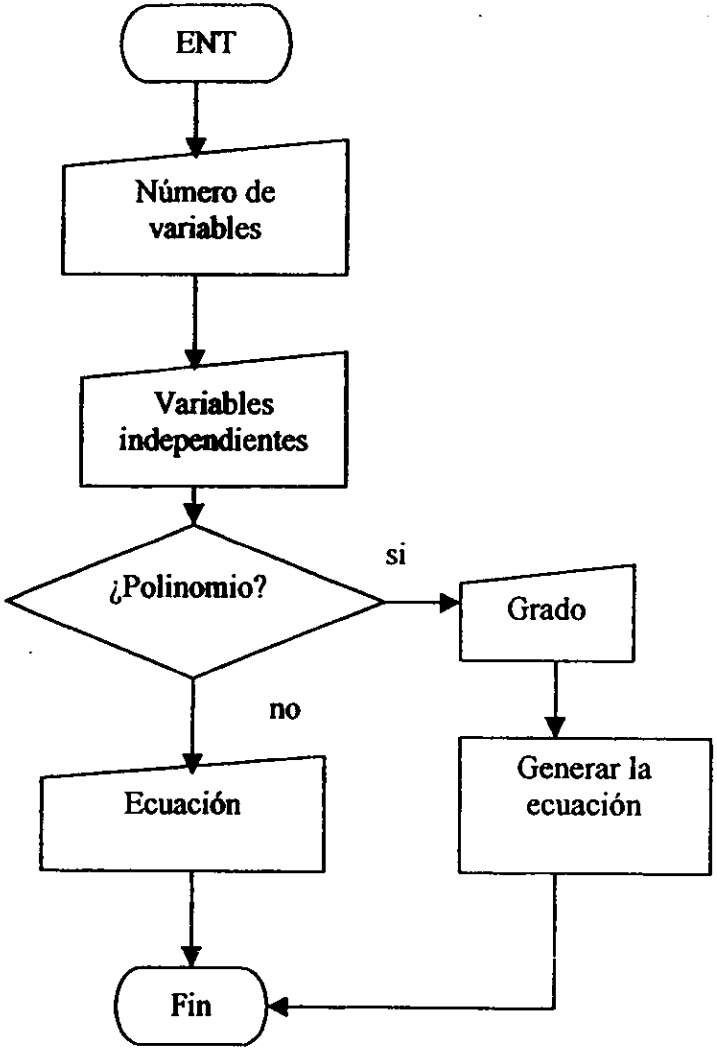
[α][α]REG CRDIR [ENTER]



Se pulsa REG y se comienzan a teclear los programas:



DAT: introduce los datos a ajustar, y solicita el modelo de ecuación.



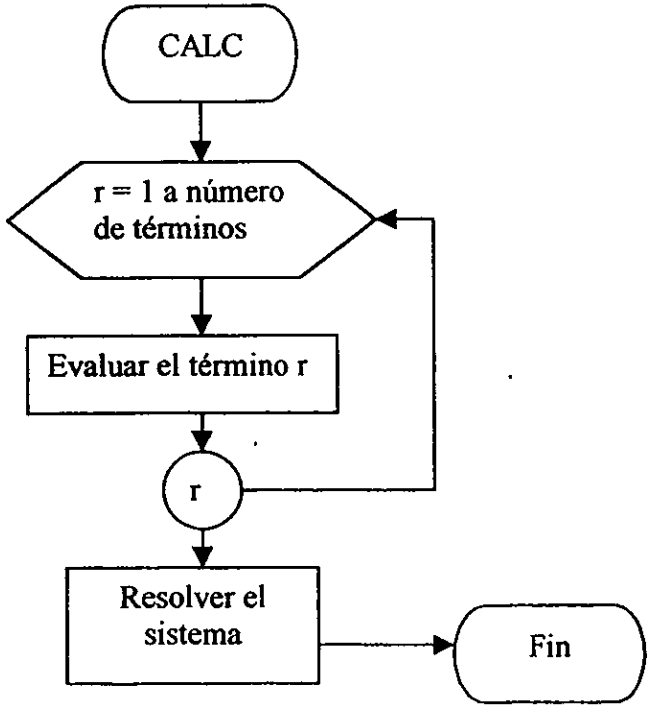
[# 36415d 388.5]

Programa	Descripción
« OPER 1	Inicia el programa, va al subdirectorio OPER y coloca un uno
"¿CUANTAS VARIABLES?"	Coloca el texto
"" INPUT STR→	Coloca el texto de entrada, hace la solicitud de datos y convierte la cadena en número
FOR num num 64 +	Inicia el ciclo para num, invoca a num y le suma 64
CHR "{" INPUT STR→	Pide el caracter como cadena, coloca el corchete de inicio hace la solicitud de datos y convierte la cadena a lista
"'" num 64 + CHR +	Coloca una cadena de comilla, invoca a num, le suma 64, solicita su caracter y lo agrega a la cadena
STR→ STO	Convierte de cadena a variable y almacena los datos
NEXT	Cierra el ciclo para num
"¿RESULTADOS?" "["	Coloca el texto y el texto de entrada
INPUT STR→ 'z' STO	Hace la solicitud de datos, convierte la cadena en vector, lo almacena en z

"ECUACION" {	Coloca el texto ecuación e inicia la lista
"POLINOMIO"	Coloca el texto
"DEFINIDA" } 1	Coloca el texto, finaliza la lista y coloca un uno
CHOOSE DROP	Solicita la opción escogida y tira la bandera
"POLINOMIO" SAME	Coloca el texto y averigua si son iguales
« "¿GRADO?" ""	Inicia el subprograma, coloca el texto y el texto de entrada
INPUT { } SWAP 0	Hace la solicitud, coloca una lista vacía, voltea el resultado y coloca un cero
SWAP STR→	Voltea el resultado y lo convierte de cadena a número
FOR t "«A " t	Inicia el ciclo para t, coloca una cadena con el comienzo de un programa e invoca el valor de t
→STR + " ^" + STR→	Convierte el valor a cadena, lo agrega a la cadena, coloca una cadena de potencia, lo agrega a la cadena y convierte la cadena a programa
+	Anexa el programa a la lista
NEXT	Cierra el ciclo
»	Termina el subprograma

« "¿ECUACION?"	Inicia el subprograma y coloca el texto
"{ « " INPUT STR→	Coloca el texto de entrada, hace la solicitud de la información y convierte la cadena a lista.
» IFTE STEQ	Termina el subprograma, evalúa el subprograma válido, y almacena en EQ
»	Termina el programa

CALC: Regresa la ecuación que más se ajuste al modelo indicado.



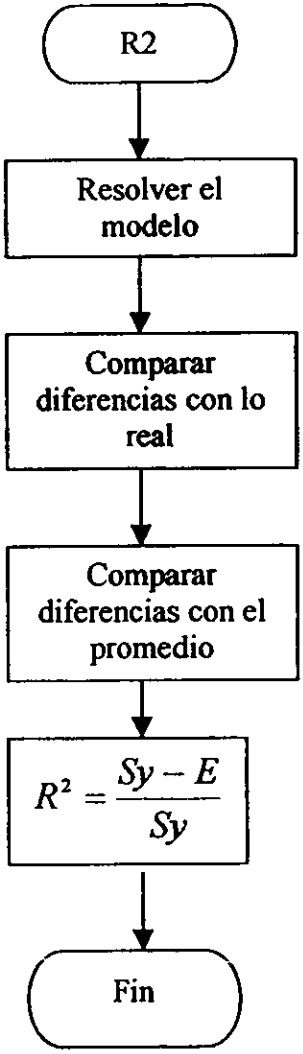
[# 13521d 255.5]

Programa	Descripción
« 'ΣDAT' PURGE 1 EQ	Borra la matriz estadística, coloca un uno e invoca la ecuación
SIZE	Toma el tamaño de la lista
FOR r EQ r GET	Inicia un ciclo para r, invoca la ecuación y obtiene el elemento r de la lista
DUP	Hace una copia
« 1	Inicia un subprograma y coloca un uno
» SAME	Termina el subprograma y averigua si son iguales
« DROP A 0 *	Inicia un subprograma, tira el último elemento, invoca a A y la multiplica por cero
« 1 +	Inicia un subprograma, le suma uno
» EVL	Termina el subprograma y lo evalúa para toda la lista
»	Termina el subprograma
« EVAL	Inicia el subprograma, evalúa el programa actual de la lista
» IFTE LIST→ {	Termina el subprograma, evalúa el subprograma que corresponda a la condición. Deshace la lista y coloca el inicio de una lista

} + →ARRY Σ+	Coloca el fin de una lista vacía, agrega el número a la lista, convierte los elementos en vector y los suma a la matriz estadística
NEXT z RCLΣ TRN	Termina el ciclo, invoca los valores de z, invoca la matriz estadística y la transpone
LSQ ARRY→ 1 GET	Resuelve los mínimos cuadrados, convierte el vector en objetos y obtiene el tamaño del vector
→LIST DUP 'y' STO	Convierte los elementos en lista, hace una copia y la almacena en y
EQ REG	Invoca la ecuación, sube al directorio REG
« EVAL	Inicia un subprograma que evalúa los términos
» EVL * ΣLIST	Termina el subprograma, evalúa para toda la lista, multiplica por la lista de coeficientes, y suma los términos de la lista
OPER	Regresa al subdirectorio OPER
»	Termina el programa

Aquí es necesario crear un subdirectorio para hacer las operaciones sin encimar variables con los programas:
[α][α]OPER CRDIR [ENTER]

R2: Calcula la correlación de la regresión.



[# 10183d 174.5]

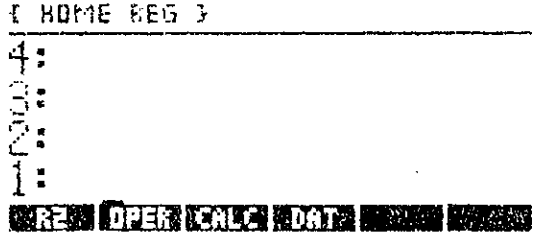
Programa	Descripción
« ΣDAT y SIZE 1	Inicia el programa, invoca la matriz estadística, invoca los resultados y, toma la medida de la lista y coloca un uno
FOR m Σ- y m GET	Inicia el ciclo parar m, quita un vector de variables de la matriz estadística, invoca a y, invoca a m y obtiene el elemento m
* -1	Lo multiplica por el coeficiente, y coloca un menos uno.
STEP 1 y SIZE 1 -	Cierra el ciclo inverso, coloca un uno, invoca a y, mide su tamaño y le resta un uno
START +	Comienza un ciclo, suma a la ecuación el término
NEXT SWAP ΣDAT	Termina el ciclo, voltea el resultado y coloca la variable ΣDAT
STO z - ARRY→ 1 GET	Almacena en la variable, invoca a z, resta los vectores, convierte el vector a sus elementos y obtiene el tamaño del mismo
→LIST SQ ΣLIST z	Convierte los elementos a lista, los eleva al cuadrado, suma la lista e invoca a z

ARRY→ 1 GET →LIST	Convierte el vector en lista
DUP ΣLIST OVER SIZE	Hace una copia, suma la lista, copia el segundo nivel y toma el tamaño
/ - SQ ΣLIST DUP	Divide las cantidades, resta el resultado, obtiene el cuadrado y suma la lista, hace una copia
ROT - SWAP /	Voltea el tercer nivel, lo resta, lo voltea y lo divide.
»	Termina el programa

Práctica

Se quieren ajustar los siguientes datos a una ecuación cúbica:

x	y
-2	4
-1	1
0	0
1	1
2	4



Se pulsa DAT

PRG

{ HOME REG OPER }

¿CUANTAS VARIABLES?

Se trata de una sola variable.

PRG

{ HOME REG OPER }

¿CUANTAS VARIABLES?

1

Se pulsa ENTER y solicita las variables independientes A:

PRG

{ HOME REG OPER }

A

{

Se teclean los valores y ENTER:

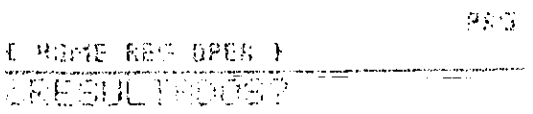
PRG

{ HOME REG OPER }

A

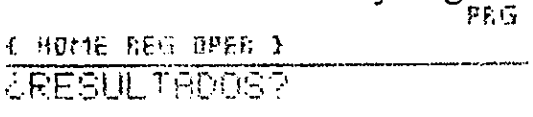
{-2 -1 0 1 2

Ahora solicita los resultados que son f(A):

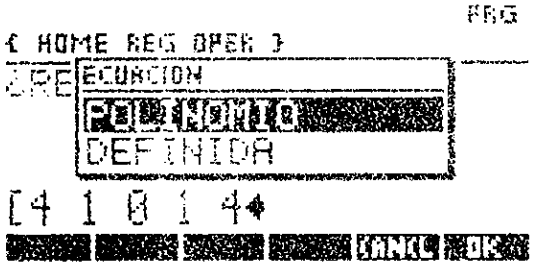


[

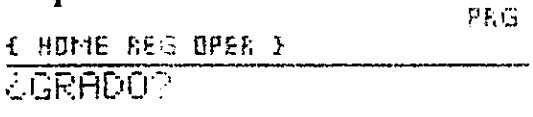
Se teclean los resultados y luego ENTER:



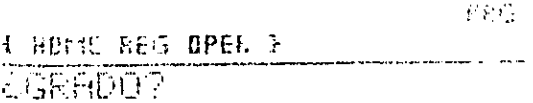
Luego solicita el tipo de modelo a ajustar, como se trata de un polinomio, el programa automáticamente definirá la ecuación.



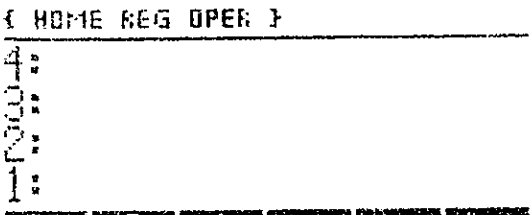
Se pulsa OK



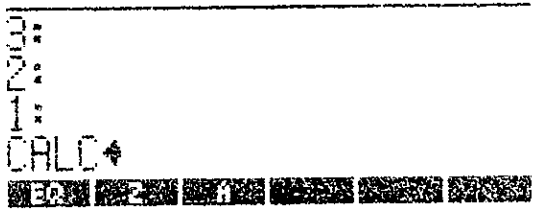
Ahora solicita el grado del polinomio:

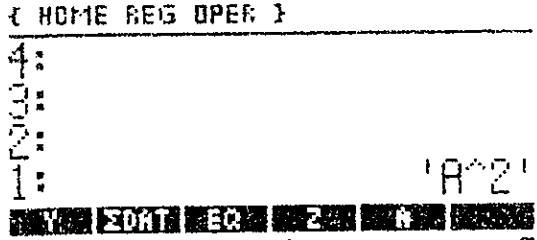


Se pulsa ENTER y aparece la siguiente pantalla:



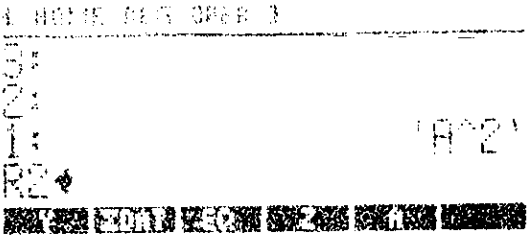
Se teclea CALC y aparece la siguiente pantalla:

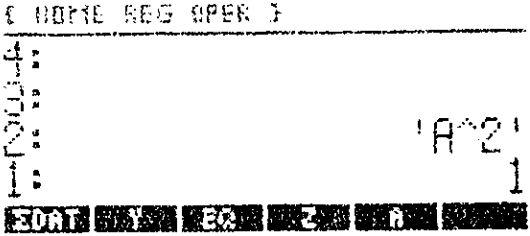




Que ya es la ecuación con sus coeficientes.

Se teclea R2 y aparece el coeficiente de correlación:





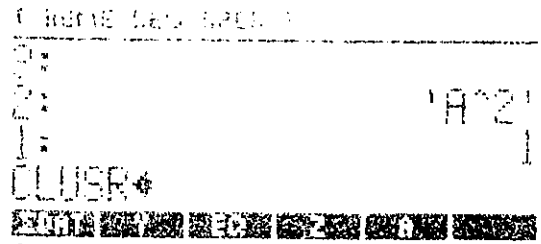
Se tienen los siguientes datos:

A	B	C	Y
2	5	3	6.8
3	6	7	10.6
1	2	2	4.3
2	4	1	6.3
2	1	2	7.3

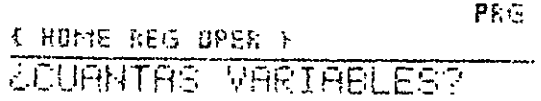
y se desea ajustar un modelo de tipo: $nA + b\frac{C}{A+B}$

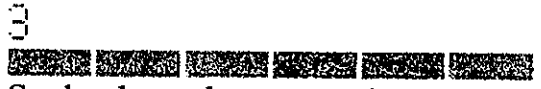
entonces la entrada del modelo a la calculadora se hará en forma de programas por cada término. Como las listas no admiten la suma directa sin concatenar, se usará el comando ADD en lugar del signo más. Así la ecuación termina así: «A» «C A B ADD /»} que es exactamente como se teclearía en la pila. **Es fundamental observar que es un programa por cada término. Si se tuviera una constante independiente, el programa «1» se reserva para tal efecto.**

Se procede a borrar toda la información en el subdirectorio con el comando CLUSR

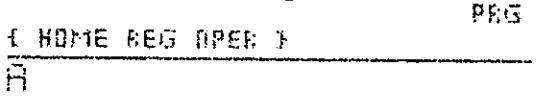


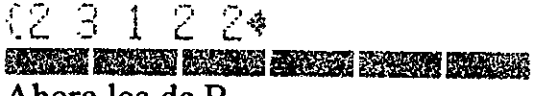
Se teclea DAT y comenzamos por declarar las 3 variables. El programa soporta hasta Z variables.



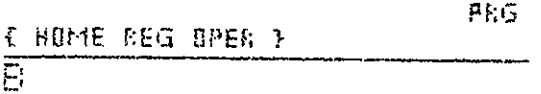


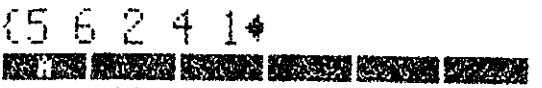
Se dan los valores para A:



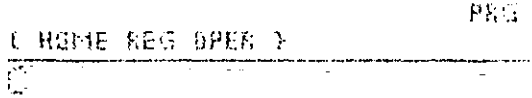


Ahora los de B

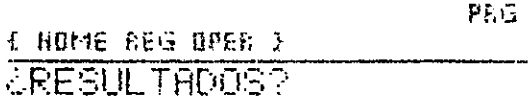




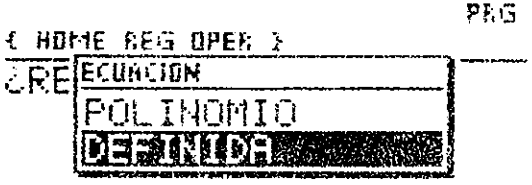
Enseguida los de C:



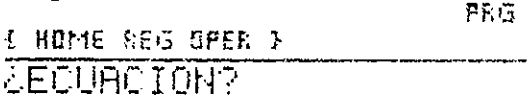
Y finalmente los resultados:



Nos solicita el tipo de ecuación y daremos la opción de definida:



Se pulsa OK



Se teclea la ecuación:

```
{ HOME REG OPER }
«ECUACION?
{ « A»
« C A B ADD ' »
»
```

Y aparece la siguiente pantalla:

```
{ HOME REG OPER }
4:
3:
2:                    'A^2'
1:                        1
```

Se teclea CALC y aparece:

```
{ HOME REG OPER }
2:                        1
1:  '2.995543664*A+
    1.991024372B9*(C-(A
    +B))'
```

Se teclea R2 y aparece:

```
{ HOME REG OPER }
4:                    'A^2'
3:                        1
2:  '2.995543664*A+1.9…
1:        .999637551122
```

Que es el coeficiente de correlación.

Anexo 1

Contenido de la encuesta

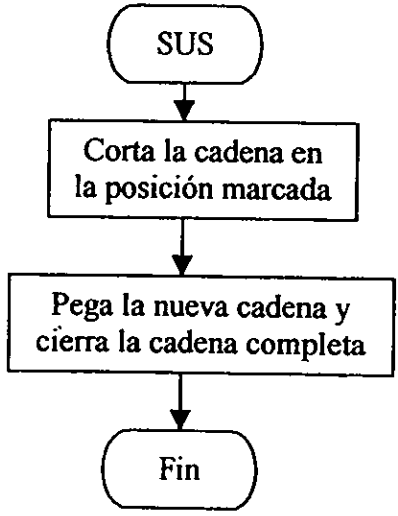
							Saben Programar	
Marca	IQ	QFB	Q	IQM	QA	Total	SI	NO
casio	25	135	25	10	60	255	110	145
sharp	10	23	5	5	15	58	23	35
hp	40	10	3	5	20	78	25	53
Texas Instrument	3	5	3	3	5	19	5	14
otras	3	10				13	3	10
						423		

Anexo 2

Subprogramas:

Los siguientes subprogramas son necesarios para la ejecución de algunos programas antes propuestos. Estos programas se utilizan en programas más grandes, y por su frecuencia en uso, se colocan en el directorio raíz HOME. Se teclean tal cual para los modelos 48G/GX. Inclusive en el modelo 48SX se pueden utilizar tal cual.

SUS: sustituye una cadena en otra cadena.

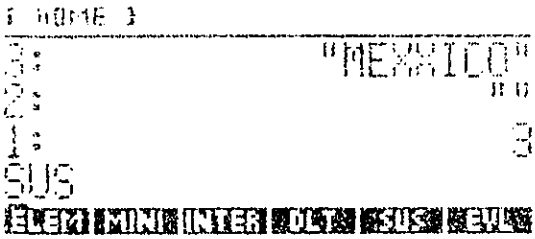


[# 14437d 90.5]

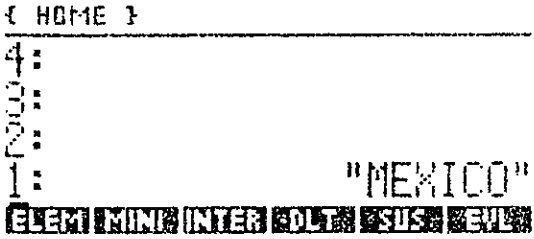
Subprograma	Descripción
« → c p n	Inicia el programa, y almacena en variables locales a c, p y n
« c 1 n 1 - SUB p	Inicia el programa local, invoca a c, coloca un uno, invoca a n, resta un uno, obtiene la subcadena e invoca a p
+ c n 1 + c SIZE	Suma las cadenas, invoca a c, invoca a n, le suma un uno, invoca a c y obtiene su tamaño
SUB +	Obtiene la subcadena y la suma a la cadena principal
»	Termina el programa local
»	Termina el programa.

Ejemplo:

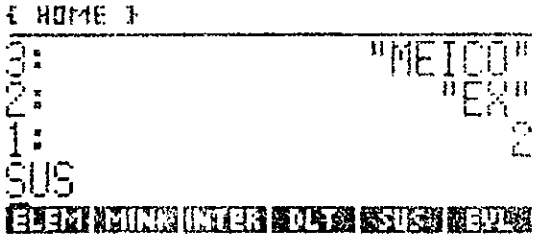
Se quiere borrar un caracter de la cadena MEXXICO



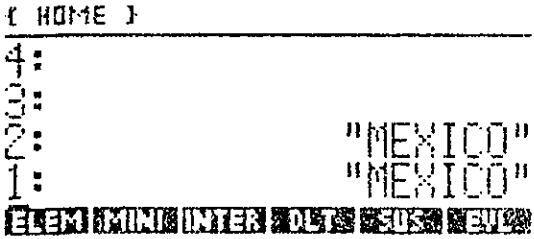
Se coloca la cadena, una cadena nula, la posición de la cadena a sustituir y el comando SUS, luego ENTER.



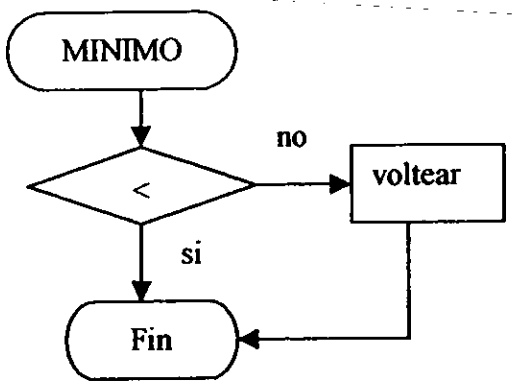
Para sustituir X en MEICO:



Se teclea ENTER



MINIMO: Selecciona 2 números de menor a mayor.

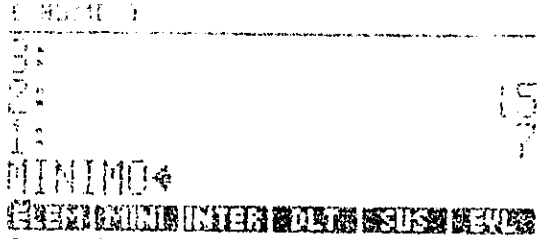


[# 54993d 43]

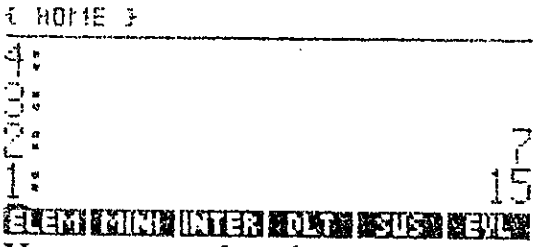
Subprograma	Descripción
« DUP2 >	Inicia el programa, hace una copia de los dos últimos números y averigua si es mayor que
« SWAP	Inicia un subprograma que voltea los dos últimos objetos
» IFT	Termina el programa, aplica el programa si la cláusula es cierta
»	Termina el programa

Práctica

Se tienen dos números a ordenar de menor a mayor que son 15 y 7:

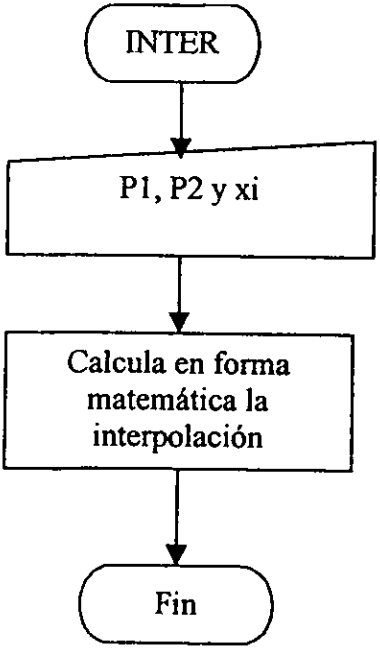


Se teclea ENTER:



Y aparecen ordenados.

INTER: Interpola o extrapola datos a partir de 2 puntos.

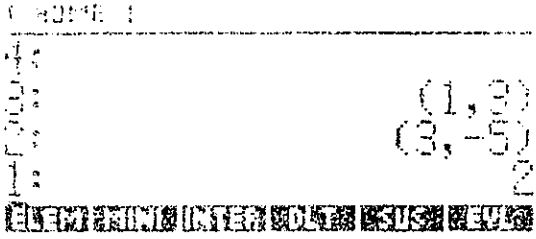


[# 40211d 99.5]

Subprograma	Descripción
« → a b c	Inicia el programa, y almacena en variables locales a a, b y c
« b RE c - a b –	Inicia el programa, invoca el punto b, obtiene la parte real, invoca a c, lo resta, invoca los puntos y los resta
IM * a b - RE / b	Obtiene la parte imaginaria, multiplica los números, invoca los puntos, los resta, obtiene la parte real, divide e invoca el punto b
IM - NEG	Obtiene la parte imaginaria, la resta y le cambia el signo
»	Fin del programa local
»	Fin del programa.

Práctica

Se desea interpolar x=2 cuando se conocen los puntos (1,3) y (3,-5). Se colocan en la pila, primero los puntos conocidos en forma de números complejos, luego el punto a interpolar y se pulsa la tecla INTER.



El resultado es –1.

Los siguientes programas son únicamente para los modelos 48G/GX. Por su sencillez no se incluye el diagrama de flujo.

EVL: toma un programa y lo evalúa para toda una lista. Si no se declara el programa, convierte los elementos según se evalúen.

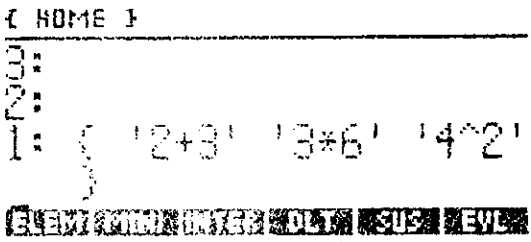
[# 45355d 83]

Subprograma	Descripción
« DUP TYPE 8 ==	Inicia el subprograma, hace una copia y averigua si el tipo de objeto es un programa
« 1 SWAP	Inicia un subprograma, coloca un uno y lo voltea
»	Termina el subprograma

« 1	Inicia el subprograma y coloca un uno
« EVAL	Coloca un subprograma que evalúa los elementos
»	Termina el subprograma
» IFTE DOLIST	Termina el subprograma y evalúa el subprograma adecuado. Ejecuta DOLIST
»	Termina el programa.

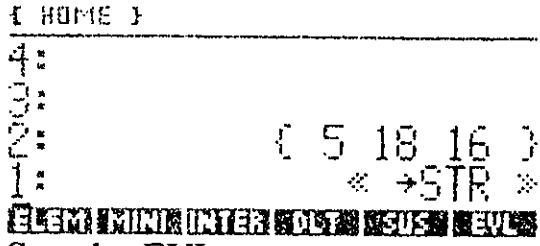
Ejemplo

Se tiene la siguiente lista y se quiere simplificar:

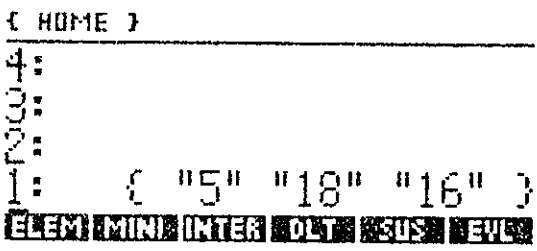


Se pulsa EVL y se obtiene:

Ahora queremos convertir los números en cadenas:



Se pulsa EVL y aparece:



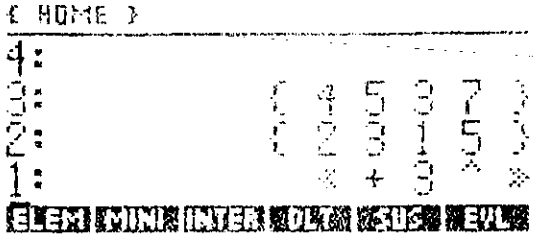
DLT: ejecuta un programa a 2 listas.

[# 15127d 28]

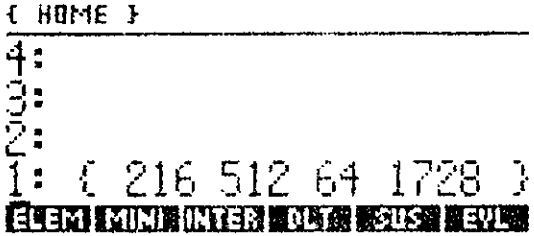
Programa	Descripción
« 2 SWAP DOLIST	Inicia el programa, coloca un 2, lo voltea y ejecuta DOLIST
»	Termina el programa.

Ejemplo

Se quieren sumar las siguientes listas y elevarse al cubo.



Se pulsa DLT y aparece:



Los siguientes programas son para poder utilizar los programas para el modelo 48SX

EVL

[# 25761d 182]

Subprograma	Descripción
« DUP TYPE 8 ==	Inicia el programa, hace una copia, averigua si es de tipo programa
« → l p	Inicia el subprograma, almacena las variables locales l y p
« l 1 l SIZE	Invoca a l, coloca un uno invoca a l y toma su medida
FOR n l n GET	Inicia un ciclo para n invoca a l, invoca a n y obtiene el objeto de la lista

p EVAL n SWAP PUT	Invoca a p, evalúa el programa, invoca a n, voltea el resultado y lo coloca en la lista
NEXT	Cierra el ciclo para n
»	Termina el subprograma
»	Termina el subprograma
« → l	Inicia un subprograma y almacena en la variable local l.
« l 1 l SIZE	Inicia un subprograma, invoca a l, coloca un uno, invoca a l y toma su tamaño
FOR n l n GET	Inicia un ciclo para n, invoca a l, invoca a n y obtiene el objeto
EVAL n SWAP PUT	Evalúa el objeto, invoca a n, voltea el resultado y lo pone en la lista.
NEXT	Cierra el ciclo para n
»	Termina el subprograma
» IFTE	Termina el subprograma y evalúa el subprograma correspondiente
»	Termina el programa

DLT
[# 53757d 95]

Subprograma	Descripción
« → l m o	Inicia el programa, almacena en las variables locales l, m y o
« { } 1 l SIZE	Inicia un programa local, coloca una lista nula, coloca un uno, invoca a l y toma su tamaño
FOR p l p GET m	Inicia un ciclo para p, invoca a l, invoca a p, obtiene el objeto e invoca la lista m.
p GET o EVAL +	Invoca a p, obtiene el objeto, invoca al programa o, evalúa el programa y agrega el resultado a la lista
NEXT	Cierra el ciclo
»	Cierra el programa local
»	Termina el programa

Los siguientes son comandos que no existen en la 48SX:

SORT: Ordena los elementos de una lista de menor a mayor

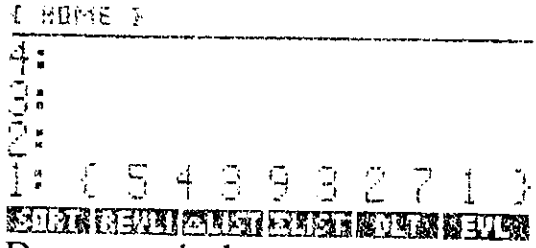
[# 20566d 149]

Subprograma	Descripción
« DUP SIZE 1 - 1	Inicia el subprograma, hace una copia, obtiene el tamaño de la lista, le resta un uno, y coloca un uno

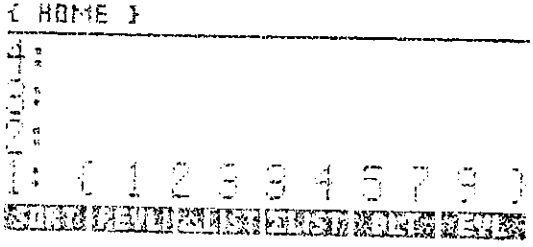
FOR j 1 j	Inicia un ciclo para j, coloca un uno e invoca a j
FOR k k GETI →	Inicia un ciclo para k, invoca a k obtiene el elemento necesario con la siguiente posición en la lista
n1	lo almacena en n1
« GETI → n2	Inicia un subprograma, obtiene el elemento y la posición siguiente, y lo almacena en n2.
« DROP	Inicia un subprograma, tira el elemento
IF n1 RE	Inicia una cláusula condicional, invoca a n1, toma su parte real,
n2 RE >	invoca a n2, toma su parte real y averigua si es mayor que
THEN k n2	Cierra la condición, invoca a k, invoca a n2
PUTI n1 PUT	Coloca el elemento en la posición deseada, conserva la siguiente posición, invoca a n1 y lo coloca en la posición deseada
END	Finaliza la condición
»	Termina el subprograma
»	Termina el subprograma
NEXT -1	Cierra el ciclo y coloca un menos uno

STEP	Cierra el ciclo en forma inversa
»	Termina el programa

Ejemplo:



Da por resultado:

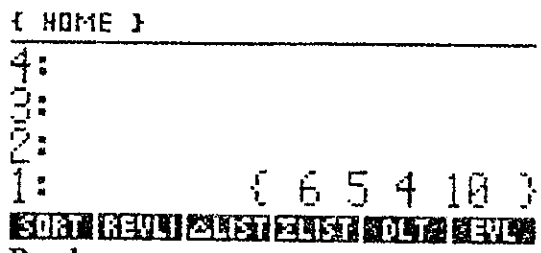


ΣLIST: suma todos los elementos de una lista

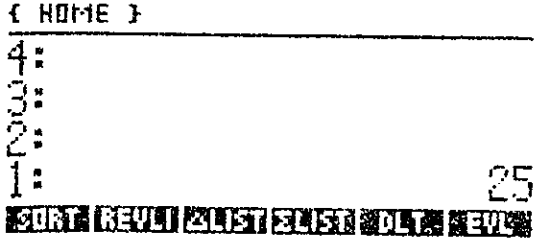
[# 19753d 39.5]

Subprograma	Descripción
« LIST→ 1 - 1 SWAP	Inicia el subprograma, deshace la lista en sus elementos, le resta un uno al tamaño, voltea un uno
START +	Comienza un ciclo, suma el elemento
NEXT	Cierra el ciclo
»	Termina el programa

Ejemplo:



Produce :

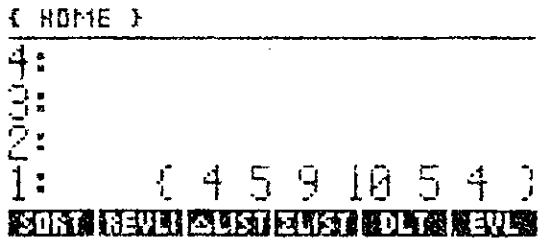


ΔLIST: Calcula la diferencia entre los elementos consecutivos de una lista

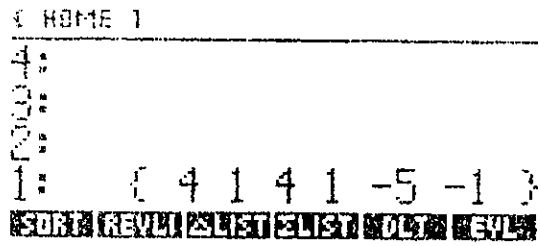
[# 39213d 71]

Subprograma	Descripción
« DUP { 0 } SWAP +	Inicia el programa, hace una copia de la lista, coloca una lista con un cero, voltea las listas y las concatena.
1 OVER SIZE 1 - SUB	Coloca un uno, hace una copia del segundo nivel, obtiene el tamaño, le resta uno y obtiene la sublista
« -	Inicia un subprograma, hace la resta.
» DLT	Termina el subprograma y ejecuta DLT
»	Termina el programa

Ejemplo:



Produce con ΔLIST:

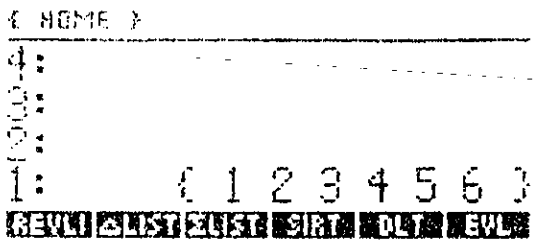


REVLIST: invierte el orden de una lista.

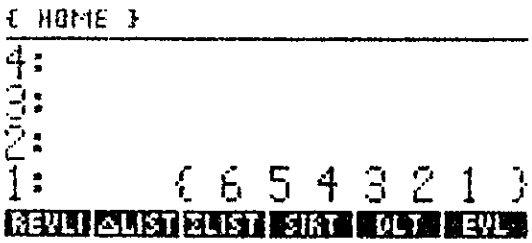
[# 3178d 97.5]

Programa	Descripción
« DUP DUP SIZE → l	Inicia el programa, hace dos copias, mide la lista y almacena localmente en l.
t	Almacena localmente en t
« 1 t	Inicia un programa local, coloca un uno e invoca a t
FOR r t 1 + r –	Inicia un ciclo para r, invoca a t, le suma un uno y le resta r.
l SWAP GET r SWAP	Invoca a l, voltea el resultado, obtiene el elemento de la lista, invoca a r y voltea el valor
PUT	Coloca el valor en la lista
NEXT	Cierra el ciclo
»	Cierra el subprograma
»	Cierra el programa.

Ejemplo:


{ HOME }
4:
3:
2:
1: { 1 2 3 4 5 6 }
REVLI ΔLIST ΣLIST SIRT DLT EVL

Produce;

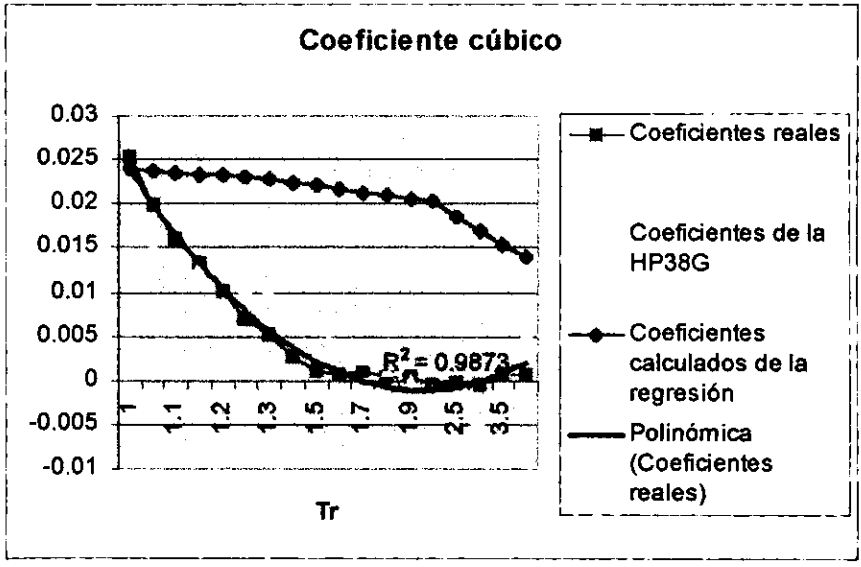

{ HOME }
4:
3:
2:
1: { 6 5 4 3 2 1 }
REVLI ΔLIST ΣLIST SIRT DLT EVL

Anexo 3

El ajuste estadístico de las propiedades residuales.

En los programas de las propiedades residuales, se pretendió hacer ecuaciones que permitieran calcular al programa los coeficientes viriales. Sin embargo el cálculo en el programa EXCEL mostró algunos errores en la correlación de los coeficientes, y el error en los cálculos se encontró superior al 400%. Se recurrió al cálculo de la correlación en una calculadora HP 38G, y el error en los cálculos fue del 200%.

Es por ello que se rechazó el cálculo algebráico y se dejó la lectura tradicional de las gráficas.



Conclusiones

Se generaron 7 directorios con algunos de los programas más representativos que se pueden generar para una calculadora HP 48G/GX.

Se encontró un comportamiento muy estable en los programas generados y una versatilidad en sus aplicaciones. Cada uno de los grupos de programas, permite que el usuario haga sus modificaciones, para mejorar la interfase o hacerla más personalizada.

Exitosamente se lograron generar aplicaciones del interés de la Ingeniería Química. Se hizo uso de las matemáticas incluidas en la calculadora, como lo fue el cálculo diferencial e integral, los mínimos cuadrados, los números complejos, etc. Se demostró que HP 48G/GX es una potente herramienta en la ingeniería.

Se encontró más confiable el software de la calculadora que el software empleado para los ordenadores caseros. Esta es otra ventaja que se encontró en el manejo de calculadoras, ya que una PC requiere de un software que generalmente es muy costoso y que puede incluir errores.

Si existe alguna duda con respecto a los programas aquí presentados, con gusto le atenderé sus dudas al teléfono 56-18-55-85. También estarán a su disposición copias de esta tesis por si alguien está interesado en conseguir una.

Bibliografía:

- Mendenhall, "Estadística matemática con aplicaciones" 2ª Edición.
- Apuntes de la clase de "Química inorgánica" impartida por el profesor Domingo Alarcón, Semestre 95-I
- Kern "Procesos de Transferencia de calor"
- Apuntes de la clase "Métodos numéricos" impartida por el profesor Caritino Semestre 96-I
- Apuntes de la clase "Ingeniería de servicios" impartida por el profesor José Sámano Semestre 98-II
- Apuntes de la clase "Flujo de fluidos" impartida por el profesor Antonio Valiente Semestre 96-I
- Apuntes de la clase "Selección y especificación de equipo" impartida por el profesor Francisco Giral Semestre 98-II
- Apuntes de la clase "Propiedades termodinámicas" impartida por el profesor Antonio Arcos Semestre 96-I
- Smith/Van Ness "Introducción a la termodinámica en ingeniería química" Mc Graw Hill 1ª edición en español.
- Bird/Steward/Lightfoot "Fenómenos de transporte"