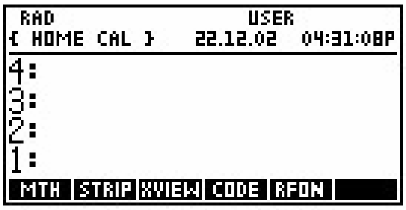
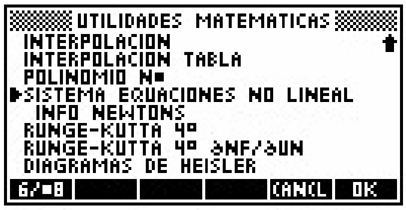
MTH/HEISLER

Esta librería contiene diversas aplicaciones matemáticas entre otras, esta estructurada de la siguiente forma primero el comando **MTH** que permite acceder a una ventana donde se puede seleccionar la aplicación deseada

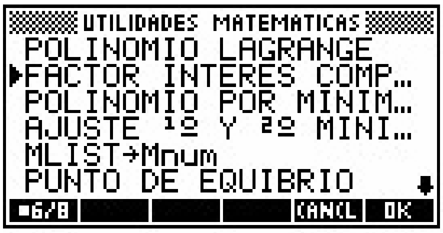


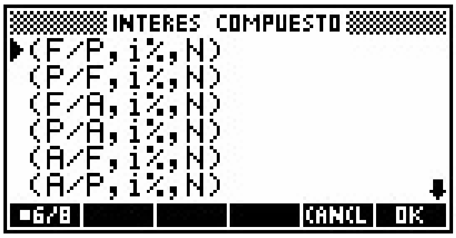


Las aplicaciones matemáticas listadas en la ventana son las siguientes:

- ✓ **POLINOMIO LAGRANGE**
  Dada una matriz [[x…xn] [y…yn]] regresa la variable **PL** en el menú **VAR** que es la función polinomio de lagrange

- ✓ **FACTOR INTERÉS COMPUESTO**
  Primero muestra ventana con las opciones de cálculo





luego permite escoger la opción calculo y resolver a través del solver o descargar la formula en la pila

- **POLINOMIO POR MINIMOS2**
  Dada una matriz [[x…xn] [y…yn]] y el grado del polinomio devuelve en el menú **VAR** dos variables, la primera **PN** que es la función poli nómica y **C** que es el vector columna con los coeficientes del polinomio **PN**

- **AJUSTE 1º Y 2º MINIMOS2**
  Dada una matriz [[x…xn] [y…yn]] y luego seleccionando el tipo de ajuste lineal o parábola devuelve en el menú **VAR** las siguientes variables: **R** (correlación de los datos), **Y** (función de aproximación), **W** (data con los cálculos necesarios para hallar el polinomio y la correlación), **A** y **B** (matriz y vector necesarios para hallar coeficientes del polinomio **A·C=B**)

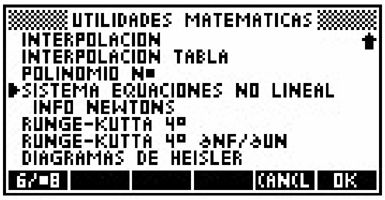
- **Mlist→Mnum**
  Dada una matriz {{ x…xn } { y…yn }} en la pila es convertida en una matriz [[x…xn] [y…yn]]

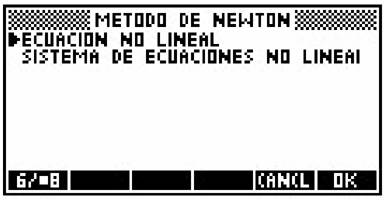
- **INTERPOLACIÓN TABLA**
  Dada una matriz [[x1,1 x2,1…xN,M]] con N,M:= 1….N,M y el valor a interpolar con respecto a la primera columna, devuelve en la pila el vector fila correspondiente a este valor

- **POLINOMIO Nº**
  Dada una matriz [[x…xn] [y…yn]], devuelve el polinomio **PN** que pasa por los puntos [xi yi] con i:=0…n

- **SISTEMA DE ECUACIONES NO LINEAL**
  Esta aplicación consta de dos partes la primera nos permite resolver una ecuación no lineal por el método de **NEWTON-RAPHSON** y la segunda por este mismo método generalizado para sistemas de ecuaciones no lineales, la sintaxis del programa es la siguiente:

$$\text{Sistema de ecuaciones}\begin{cases} X\cdot Y=3 \\ X-Y=9 \end{cases} \quad \text{en el programa} \rightarrow \{\ 'X\cdot Y\text{-}3'\ \ 'X\text{-}Y\text{-}9\ '\ \}$$



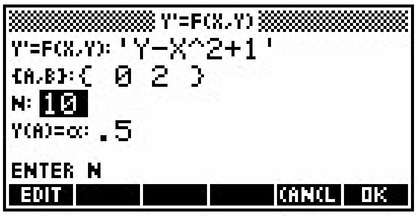


Como resultado el programa devuelve un menú temporal donde se encuentran los valores que satisfacen a dicho sistema de ecuaciones, estas últimas igualadas a cero y evaluadas para los valores hallados

- ✓ **RUNGE-KUTTA 4º**

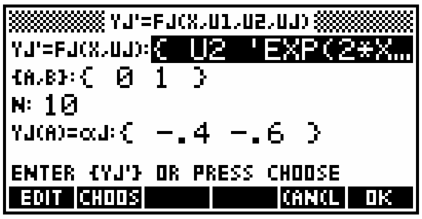
Resuelve problemas de valor inicial

$$\frac{d}{dx}y = f(x,y) \quad a \le x \le b \quad y(a) = \alpha$$



- ✓ **RUNGE-KUTTA 4º δNF/δUN**

Resuelve ecuaciones de orden superior y sistemas de ecuaciones diferenciales, ver el siguiente ejemplo:



$$\frac{d^2}{dx^2}y - 2\frac{d}{dx}y + 2\cdot y = e^{2\cdot x}\cdot\sin(x) \qquad 0 \le x \le 1 \qquad y(0) = -0.4$$

$$\frac{d}{dx}y(0) = -0.6$$

$$y = U1 \qquad \frac{d}{dx}U1 = \frac{d}{dx}y = U2 \qquad \frac{d^2}{dx^2}U1 = \frac{d}{dx}U2 = \frac{d^2}{dx^2}y$$

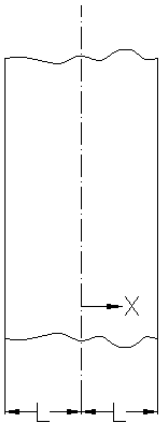
$$\boxed{\begin{array}{ll} \frac{d}{dx}U1 = U2 & 0 \le x \le 1 \\ & U1(0) = -0.4 \\ \frac{d}{dx}U2 = e^{2\cdot x}\cdot\sin(x) + 2\cdot(U2 - U1) & U2(0) = -0.6 \end{array}}$$

En el programa → **:{ 'U2' 'EXP(2*X)*SIN(X)+2*(U2-U1)' }**
:{ 0 1 }
: 10
:{ -0.4 -0.6 }

✓ **DIAGRAMAS DE HEISLER**

<u>SÓLIDOS UNIDIMENSIONALES</u>

Placa Infinita

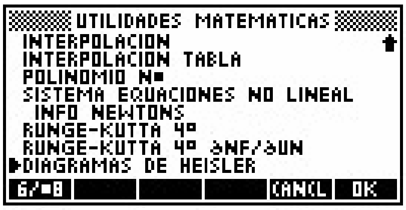


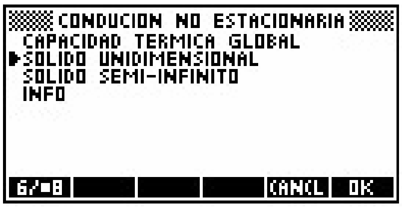
Para estas condiciones iniciales y propiedades termo físicas se pide hallar la temperatura a una distancia de 1.25 cm después de haber transcurrido 60 s

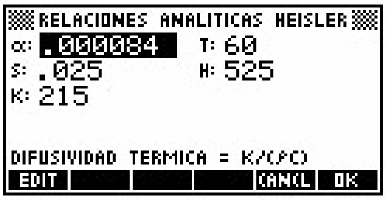
$$T\infty = 70C \qquad \alpha = 8.4\cdot 10^{-5}\,\frac{m^2}{s} \qquad H = 525\,\frac{W}{m^2} \qquad K = 215\,\frac{W}{m}$$

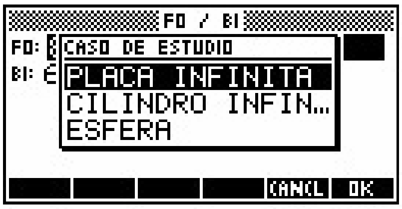
$$Ti = 200C$$

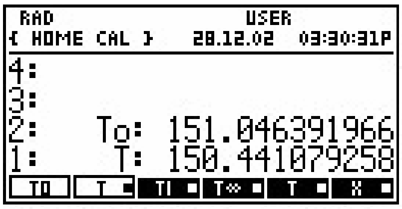
$$L = 2.5\cdot 10^{-2} m$$











con la ayuda del programa encontramos la solución al problema, tenemos que para una distancia de 1.25 cm trascurrido los 60 s la temperatura es de 150.44 C y adicionalmente para una distancia 0 cm es de 151.04 C, *el programa es un poco lento pero a mi parecer es mas cómodo y definitivamente mas exacto que el método grafico.*