Pequeña ayuda como usar los programas del paquete.

Por Manuel Dario Fajardo Hernández
Ing. Eléctrica USAC.

La carpeta comprimida tiene varios archivos:

La librería o archivo binario llamada AnalisisNumericoMDFH, el cual se puede instalar a la calculadora por medios tradicionales.

Una carpeta llamada Análisis Numérico para funciones, el cual contiene los siguientes códigos de los programas hechos por el autor "ESE SOY YO", menos un programa:

BISECCION
PUNTO FIJO
NEWTON
SECANTE
FALSAPOSICION
STEFFESEN
MULLER
LAGRANGE
NEVILLE
GAUSSEIDEL
JACOBI
PUNTOFIJOVARIASVARIABLES
NEWTONVARIASVARIABLES
SERIEDEFOURIER "Otro autor: David Enrique Torres. Ver tutorial de dicha persona"

Una carpeta llamada Análisis Numérico para Cálculo diferencial, el cual contiene los siguientes códigos de los programas hechos por el autor.

RICHARDSON
DIFERENCIACIONNUMERICA
TRAPECIOS
SIMSONP
ROMBERG
CUADRATURAGAUSSIANA
EULER
METODOTAYLORIORDENSUPERIOR

Los programas están hechos en y para el modo algebraico usando el lenguaje de usuario de la calculadora: User RPL modo Algebraico. La calculadora sin embargo permite usarlos en modo RPN, solo que se agrega los datos entre apostrofes ''.

Todos los programas son automáticos, una vez que los datos se ingresan y se da ENTER, la respuesta se visualiza hasta que el programa termina

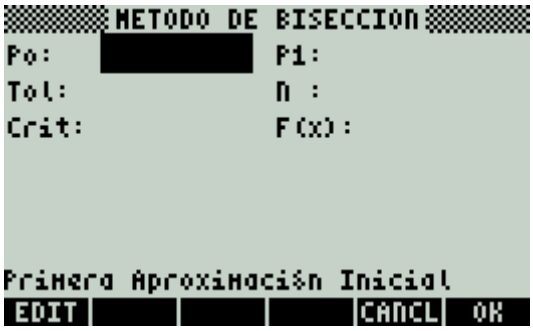
DE BISECCION A MULLER

Para los programas de Bisección, Punto Fijo, Newton, Secante, Steffesen, Muller, Se ingresa los valores iniciales, Po, P1, A, B, etc. Luego el error o tolerancia de la aproximación Tol, la cantidad de iteraciones n, y Luego la función o ecuación que se esta trabajando, luego ENTER.

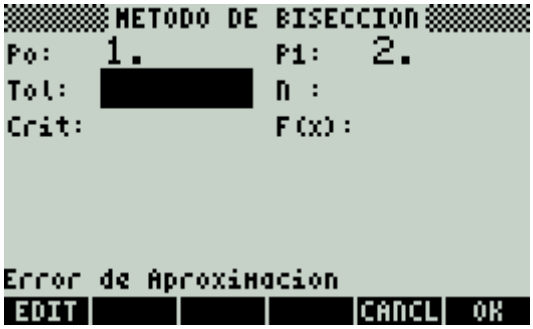
La salida casi siempre es una matriz, cada fila de la matriz contiene, el número de iteración correspondiente, el valor de P aproximación y el valor del Error. Esta matriz se guarda en la variable M en HOME junto con otras variables que se crean durante la ejecución del programa.

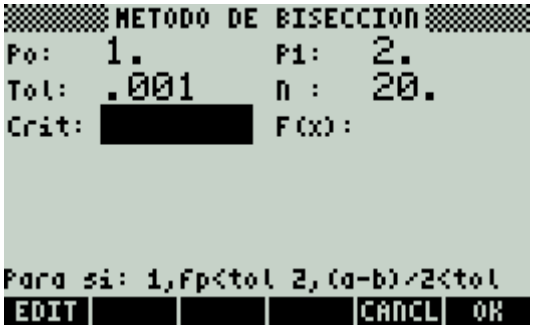
A continuación, un ejemplo con el método de Bisección:
Se quiere una valor aproximado de $\sqrt{2}$, con una precisión 10^{-3}, para esto se puede usar la función x^2-2, bastara unas 20 iteraciones.

La pantalla de entrada siempre es un menú INFORM:



Sabemos que la raíz de la función esta entre 1 y 2. Se introduce el error o tolerancia minima y la cantidad de iteraciones

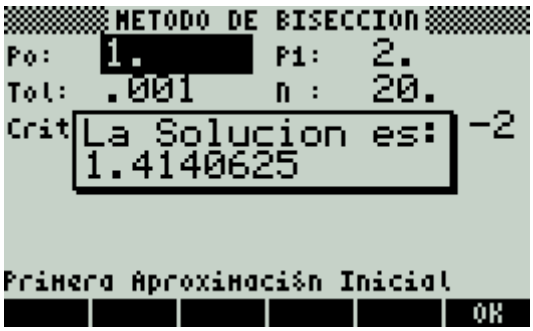


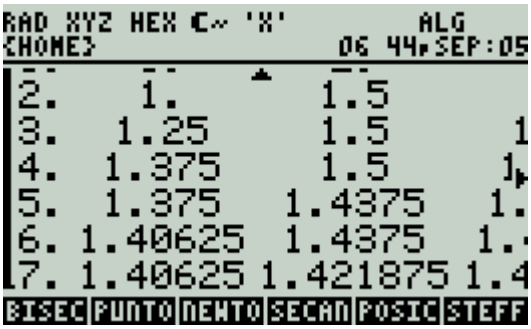


En la casilla Crit, se utiliza para el criterio de paro, en bisección se puede calcular el error con (B-A)/2 o con El valor de P valuada en la función, la cual debe ser aproximadamente cero. Así que si se quiere utilizar criterio (B-A)/2 < Tolerancia, introduces 2 y si se quiere utilizar criterio f (P) < Tolerancia, introduces el 1.

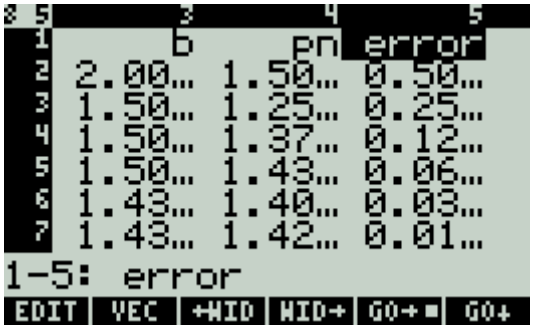
Solo en este programa se da la opción de dos criterios de paro para el algoritmo. Para el resto de los programas se utiliza el criterio de paro ABS (P_n-P_{n-1}) < Tolerancia. Para el ejemplo se utiliza el criterio 2, y luego introducimos la ecuación, se puede usar también el editor de ecuaciones EQW para editar la ecuación.

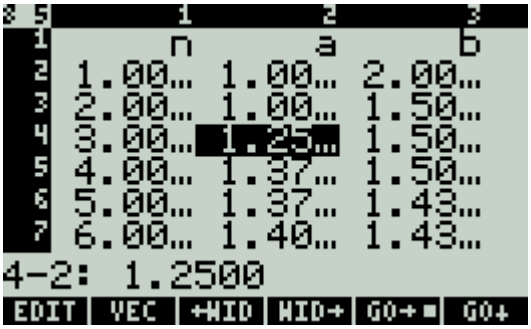
Luego se da OK, y se espera un momento, El programa tardara unos 10 segundos, Algunos programas pueden tardar más tiempo y tardar varios minutos, el que más se tarda es el de cuadratura gaussiana debido a la cantidad de operaciones a hacer y lo complicado del método en si.
Al terminar el programa aparece un mensaje con el valor de P. Luego al dar OK u ENTER otra vez, aparecerá la matriz M en la pantalla principal. Para ver mejor todas las iteraciones hechas por el programa es aconsejable usar el editor de matrices, para poder desplazarse por todos los elementos de la matriz M que aparece luego en la pantalla principal.



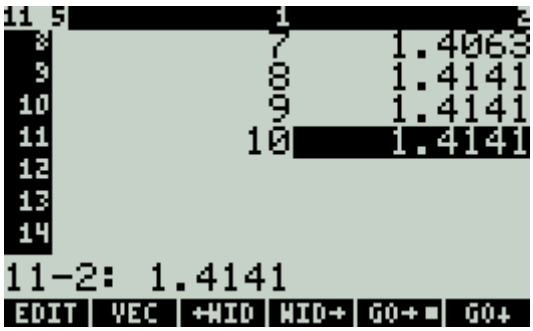


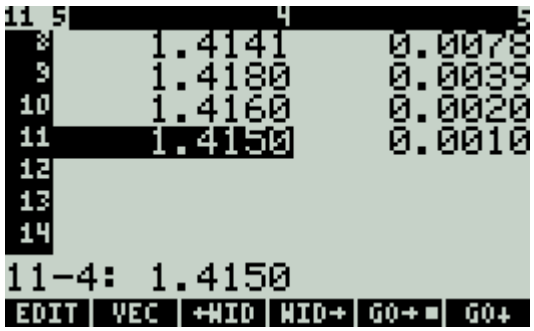
Con 4 cifras significativas y en el editor de matrices MTRW se ve algo así:





El programa termina en la décima iteración, en la fila 11 y columna 4 esta el valor de P con cuatro cifras significativas.





Nota: El número de fila se suma uno al de N, pues la fila del encabezado [a, b, pn, error], se suma a la matriz, por eso la iteración 10 aparece en la fila 11.

Aquí se muestran algunas de las variables que generó el programa, incluido la matriz *M*

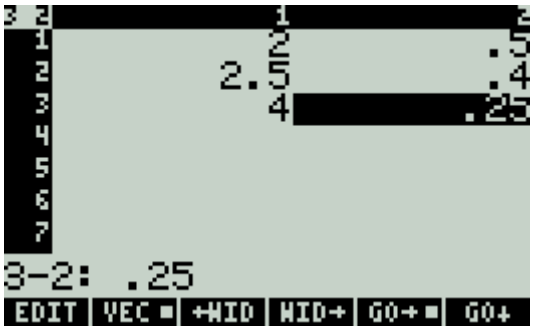
INTERPOLACION DE LAGRANGE Y DE NEVILLE

Para el Programa Lagrange, solo se requiere una única entrada que es la matriz con los datos a interpolar. La calculadora también ofrece un comando que calcula el polinomio interpolante de Lagrange, la diferencia es que en mi versión los datos son ingresados en forma vertical.

Ejemplo: Buscando un polinomio de Lagrange para la siguiente tabla de datos

x	F(x)
2	0.5
2.5	0.4
4	0.25

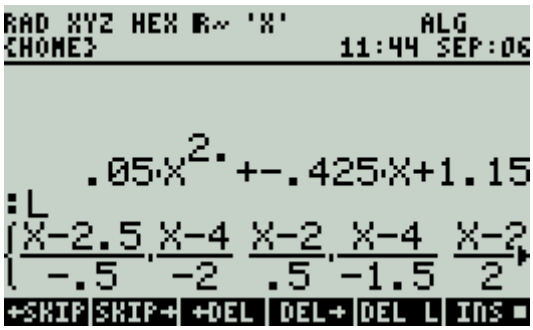
Primero se ingresa la matriz con los datos de la tabla, se puede usar el editor de matrices si se desea y luego OK.

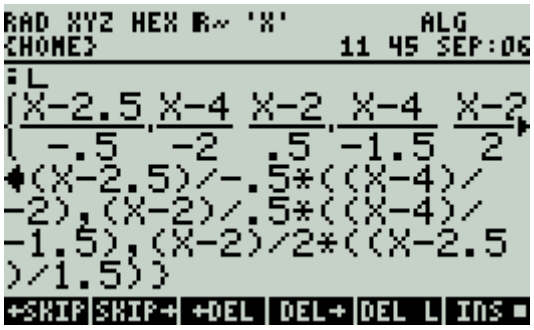


La salida seria la misma que proporciona el comando LAGRANGE de la calculadora, Seria este polinomio interpolante.

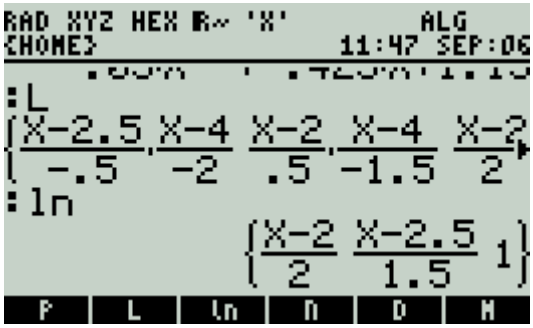
Sin embargo el programa provee información extra que se puede hallar en HOME.

La variable P que contiene al polinomio resultante. La variable L tiene la función de nodos $L_k(x)$, que genera el algoritmo.





La función $L_n(x)$ del polinomio de Lagrange



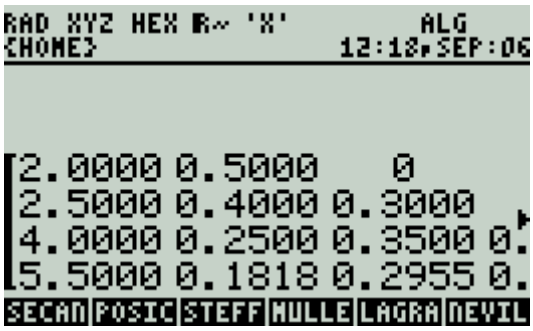
El Programa del algoritmo de Neville, tiene la misma entrada de datos, solo que hay una casilla llamada Xo, donde se introduce el punto donde se desea interpolar.

Pruebo el programa con el siguiente ejemplo:
Con 4 cifras significativas, interpolar el valor de la función en P = 1.5, con los siguientes datos.

x	F(x)
2	0.5
2.5	0.4
4	0.25
5.5	0.181818

Interpolando den x = 3

Luego OK, La salida es la matriz M con las interpolaciones.



El dato de la esquina es la interpolación en x = 3

GAUSSEIDEL Y JACOBI

También esta el programa de Gausseidel y Jacobi, que sirven para aproximar la solución de sistemas de ecuaciones.

La entrada para este programa siempre será la matriz del sistema de ecuaciones, de forma diagonalmente dominante. No hay que indicar la cantidad de ecuaciones ni indicar el nombre de las variables, como dije antes los programas son lo mas automáticos que pude hacerlos. Solo se debe ingresar la matriz, el código del programa identifica el tamaño de la matriz y de ahí genera la variables necesarias.

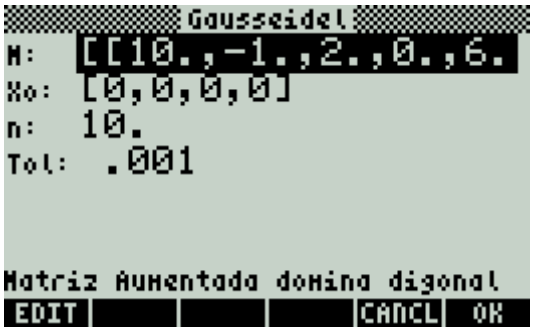
Ejemplo: Resolver la siguiente sistema de ecuaciones con el método de Gausseidel con una tolerancia de 10^{-3}, y 10 iteraciones.

$$10X_1 - X_2 + 2X_3 = 6$$
$$-X_1 + 11X_2 - X_3 + 3X_4 = 25$$
$$2X_1 - 1X_2 + 10X_3 - X_4 = -11$$
$$3X_2 - X_3 + 8X_4 = 15$$

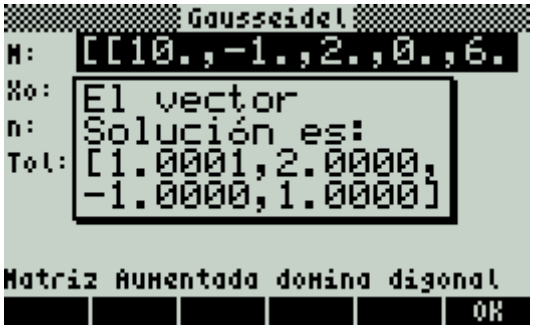
La matriz ya esta en su forma diagonalmente dominante.

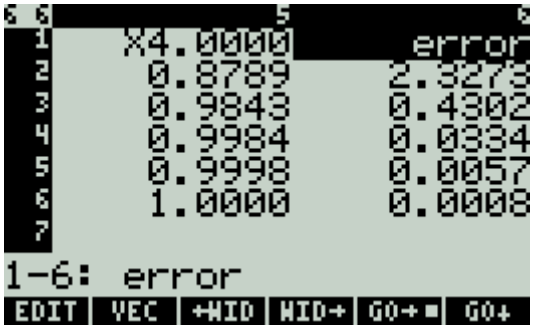
Esta es la presentación:

Se ingresa la matriz usando el editor de matrices MTRW, el vector inicial, la cantidad de iteraciones y la tolerancia.



El resultado del programa es este:





Las variables importantes generadas son:

La variable P1 que contiene el vector solución, la matriz M de la tabla y Mo de la matriz del sistema de ecuaciones, y la variable H con la lista de la variables {x1, x2,…, xn}

El programa de Jacobi tienen las mismas entradas.

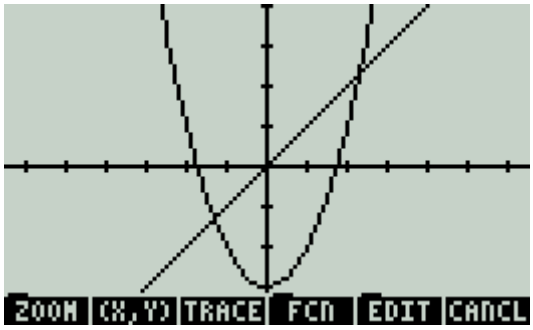
PUNTO FIJO Y NEWTON PARA SISTEMAS DE ECUACIONES NO LINEALES DE MULTIPLE VARIABLES.

Para el programa de punto fijo de varias variables, se ingresa el sistema de ecuaciones en un vector fila, en cada elemento o componente del vector se ingresa una ecuación, se deben introducir las variables como X!, X2, hasta Xn, no hay necesidad de indicar la cantidad de variables, el programa reconoce por si solo la cantidad de variables a utilizar y las genera automáticamente, el programa puede con cualquier tamaño de sistema de ecuaciones, hasta donde la calculadora sea capaz de aguantar, sistemas de 2 x 2, 3 x 3,…, hasta n x n.

Ejemplo: Se trabajara con el siguiente sistema de ecuaciones no lineales

$y=x^2 - 3$
$y=x$

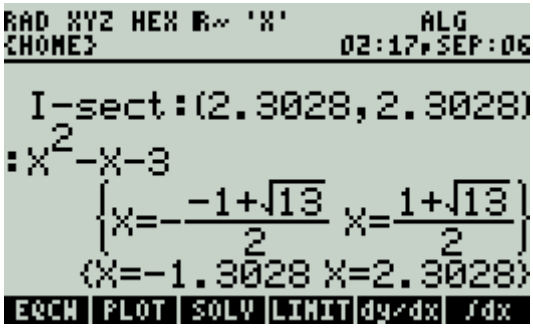
Si graficamos las dos ecuaciones se ve esto.



De forma convencional para hallar las intersecciones de ambas curvas se haría esto:

$X^2 - x - 3 = 0$ y se resolvería la ecuación por otros medios.

La solución seria:



Pero ese no es el punto, en este ejemplo se usa el sistema igualado a cero, y se debe obtener una solución X_1 y X_2.

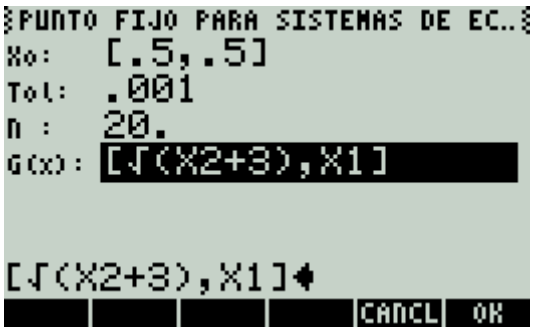
Las ecuaciones a trabajar serían:

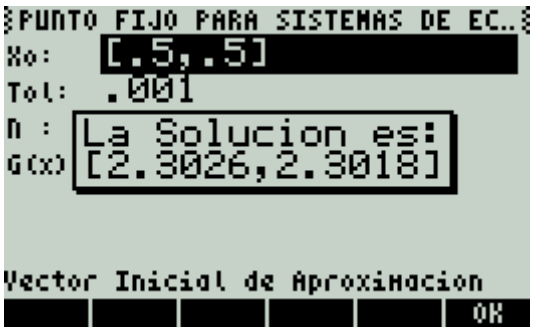
$X_1^2 - X_2 - 3 = 0$
$X_1 - X_2 = 0$

Luego se despeja X_1 y X_2, según lo exige el método de punto fijo x = f (x). Las siguientes ecuaciones formaran los elementos del vector de entrada de ecuaciones G(x).

$X_1 = \sqrt{X_2 + 3}$
$X_2 = X_1$

Los datos de entrada del programa son, el vector inicial Xo = [0.5, 0.5], con una tolerancia de 10^{-3}, y 20 iteraciones. La salida es:

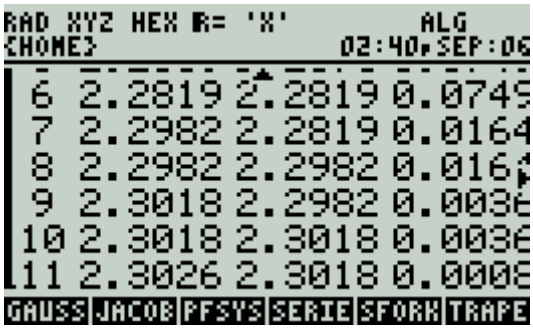


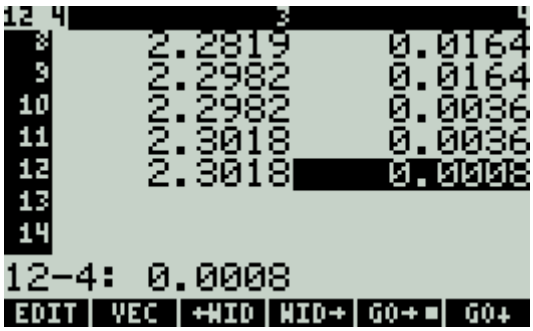


Como se ve en la gráfica, la intersección de la curvas esta en el punto en X = 2.302 Y = 2.302, Como se ve es el mismo resultado obtenido por métodos analíticos de una sola variable.

En este caso el valor de X_1 coincide con el de X_2, pero no quiere decir, que en otros casos las variables sean las mismas.

La matriz de salida del programa es la siguiente.





Las variables que se generan son:

La variable P tiene el vector Solución, La variable d, es la matriz con Jacobiano del sistema de ecuaciones, esta la utiliza el programa para verificar si hay punto fijo en el punto especificado. D (f(x)) < 1.

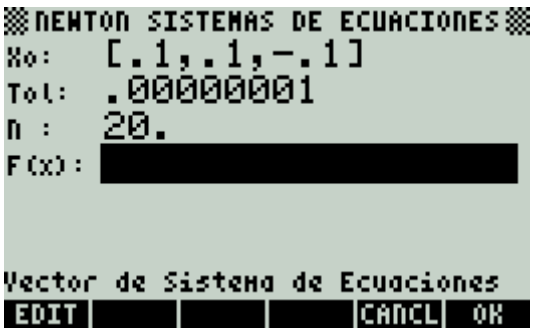
Por motivos de aprendizaje, en punto fijo de varias variables, las variables se deben especificar como X_1 hasta Xn, Pues el código del programa genera las variables de este modo. En el método de Newton para sistemas de ecuaciones no lineales, se pueden usar diferentes nombres de variables, pues el comando LNAME extrae las variables del sistema.

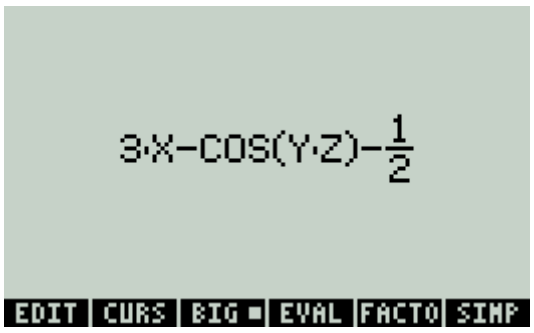
A continuación presenta un ejemplo con un sistema de ecuaciones no lineales para resolver con el método de newton multivariable.

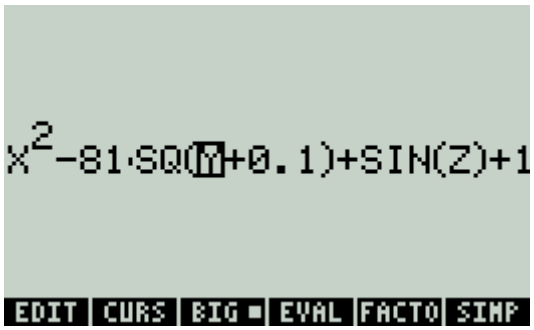
Para este ejemplo se utiliza el siguiente sistema:

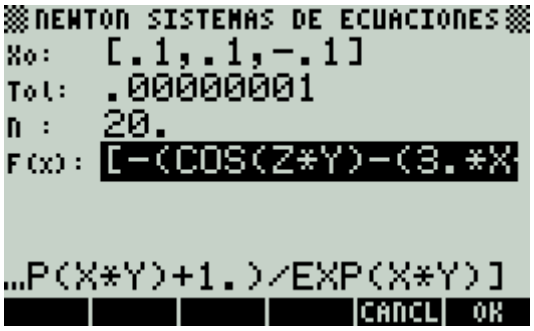
$$3x - \cos(yz) - \frac{1}{2} = 0$$
$$x^2 - 81(y + 0.1)^2 + \sin(z) + 1.06 = 0$$
$$e^{-xy} + 20z + \frac{10\pi - 3}{3} = 0$$

Con una precisión de 10^{-8}, y 20 iteraciones.



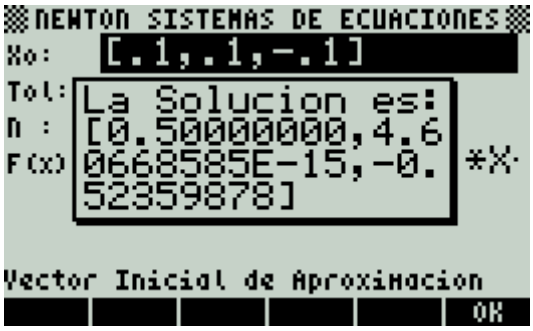


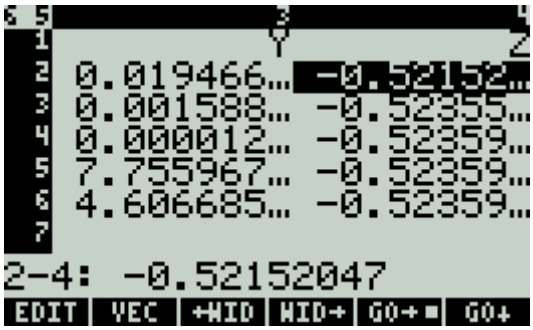


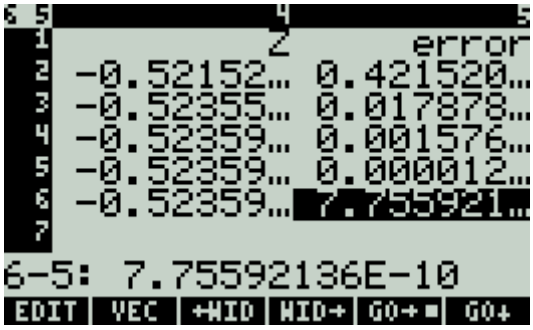


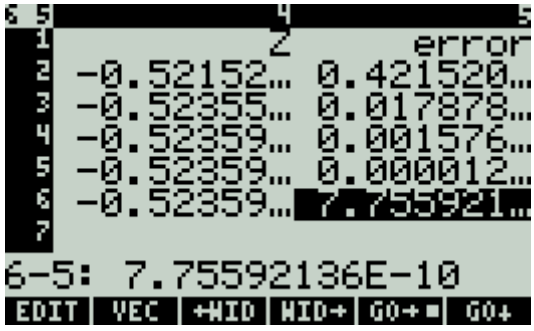
Luego se da OK u ENTER, El programa de punto fijo y newton para sistemas de ecuaciones, son algo tardados pero, si los datos están bien introducidos siempre darán una respuesta.

La salida seria la siguiente.

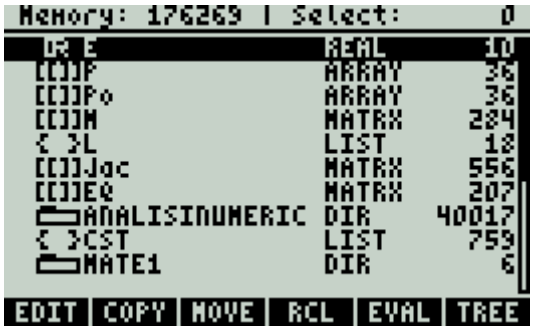








Las variables generadas son estas:

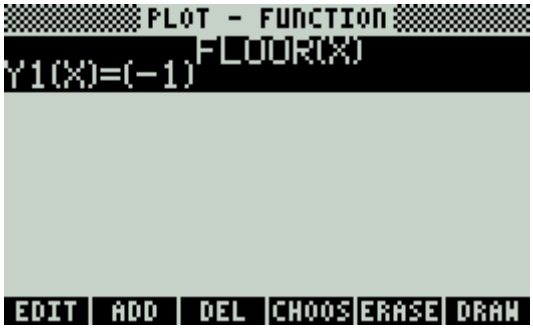


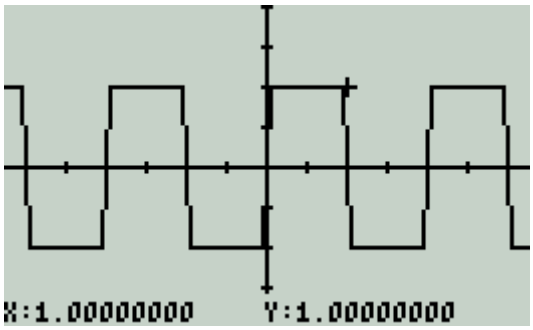
SERIE TRIGONOMETRICA DE FOURIER

El programa de serie de Fourier, calcula la serie trigonométrica de Fourier, la serie de Fourier es muy útil, para cualquier fenómeno ondulatorio, y muy útil en análisis de fenómenos eléctricos variantes en el tiempo.

La calculadora tiene también su propio comando, pero calcula la serie compleja en términos de e ^ (jωt).

Para ejemplificar, utilizo la función de onda cuadrada. Para graficarla en la HP utilizo la siguiente línea:

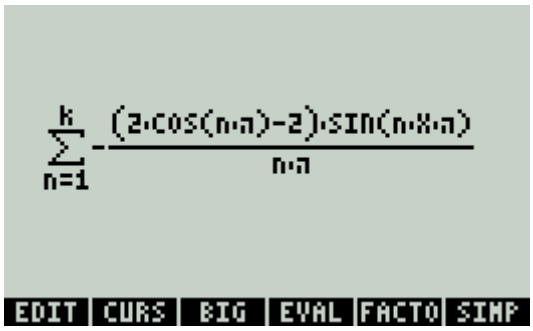




El comando FLOOR (X) extra la parte entera de X.
Luego usamos el programa de Serie de Fourier.

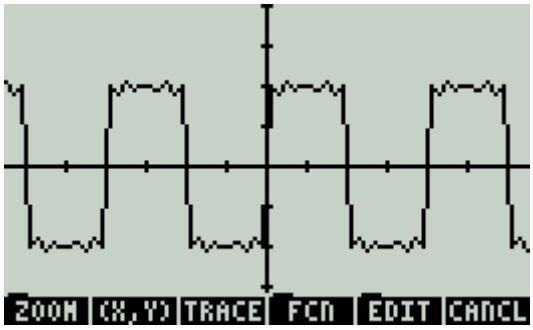
La casilla tramos, se usa cuando la función es por partes, por cada tramo se ingresa una función en el caso de la onda cuadrada son dos, si solo fuera una función, se ingresa 1 en esta casilla. La casilla términos, es para indicar cuantos términos tendrá la sumatoria para mostrar.

Luego se da OK y a esperar. La salida es la siguiente.

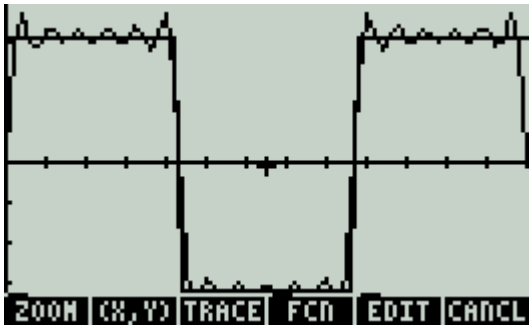


Las variables generadas son:

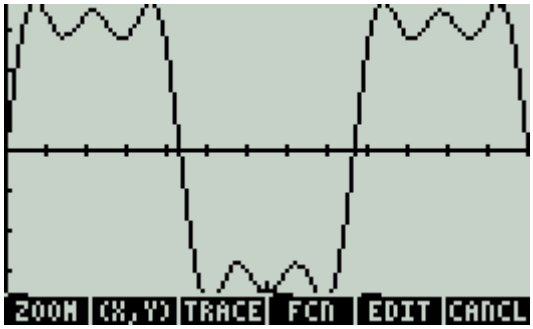
SFourier, tiene la expresión de la Serie, y EQ la serie para graficar, en este caso con 11 términos. Las constantes ao, an y bn y la variable k. Para una gráfica con más o menos términos, solo se cambia el valor de k



La grafica anterior tiene 11 términos. La siguiente gráfica es para Comparar con la gráfica de la onda cuadrada.



Al cambiar el valor de k por k = 5, se ve la diferencia con la gráfica anterior que tenia 11 términos, pues la serie fue graficada con una cantidad menor de armónicos o términos. Pero entre mayor sea el valor de armónicos k, más tiempo le llevara a la calculadora graficar la serie.



CALCULO NUMERICO DIFERENCIAL.

RICHARDSON

Uno de los primeros programas, es el del método de Extrapolación de Richardson, no es un método numérico para el cálculo diferencial en si, pero se usan en otros métodos como método de refinamiento, como lo es el caso del método de Romberg.

Al contrario de un método de interpolación, que usa varios puntos para aproximar, un valor de una función en un punto cualquiera dentro del dominio de un rango de valores, la extrapolación, usa un conjunto de valores, resultado de la utilización de un mismo método con varios tamaños de paso, para encontrar una solución en punto, por lo que la extrapolación hace un refinamiento de un resultado a partir de resultados menos precisos.

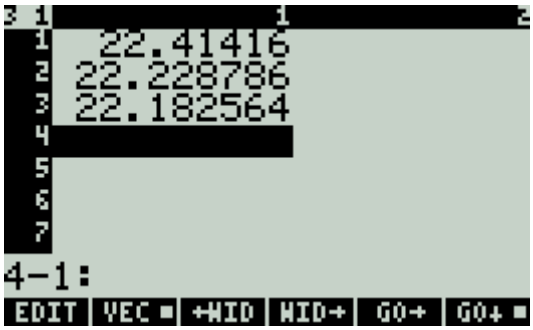
Los datos se ingresan de forma parecida en el método de Lagrange y Neville, pero la matriz, es ahora para extrapolar.

En el ejemplo siguiente se hará una aproximación más precisa de la función f (x) = X*e ^x + e ^x en X = 2, a partir de la siguiente tabla de datos, con valores menos precisos que resultaron de otros métodos.
Se usara la formula de base 2, que es más, general, pues la de base 4, elimina términos de potencias pares de h^2.

El valor exacto es f (2) = 22.16716829

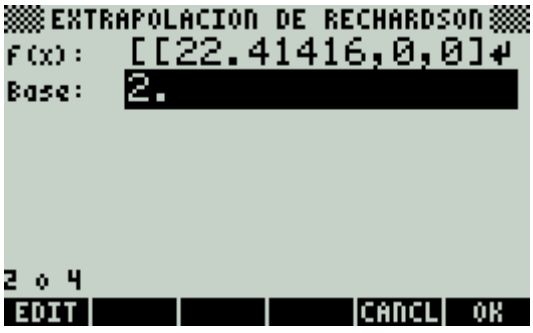
$N_1(h)$
22.41416
22.228786
22.182564

La presentación es la siguiente:

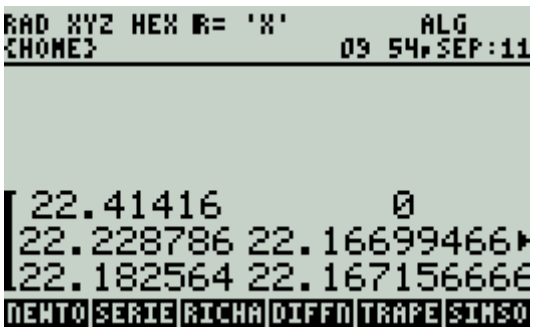


Luego se escoge la base H que puede se 2 o 4, Pues la formula general de iteración de Richardson es:

$$N_j(h) = N_{j-1}(h/2) + \frac{N_{j-1}(h/2) - N_{j-1}(h)}{H^{j-1} - 1}$$



Al dar ENTER,



El último valor de la esquina es una la aproximación más precisa de f(x). El error es de la extrapolación es de 10^{-5}

Se generan dos variables, M y Mo, M es la matriz de la extrapolación Y Mo de los datos de entrada.

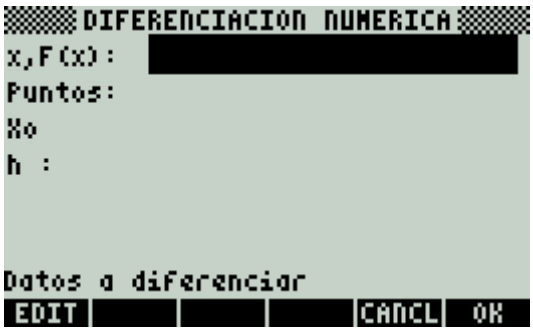
DERIVADA NUMERICA

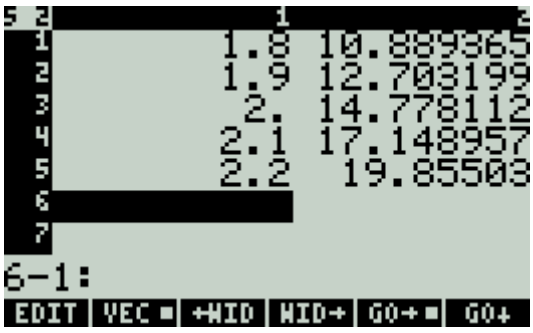
Para aproximar la derivada de un punto a partir de un conjunto de datos, se usa comúnmente las formulas de n puntos, que se deducen a través de series de Taylor de grado 2, 3, 4, 5.

Para el siguiente ejemplo se usaran valores de la tabla generada con la función F (x) = x*e ^x. Aproximaremos la derivada de F (2) con formulas de 3 y 5 puntos.

x	F(x)
1.8	10.889365
1.9	12.703199
2	14.778112
2.1	17.148957
2.2	19.855030

Se introduce la matriz con los datos.

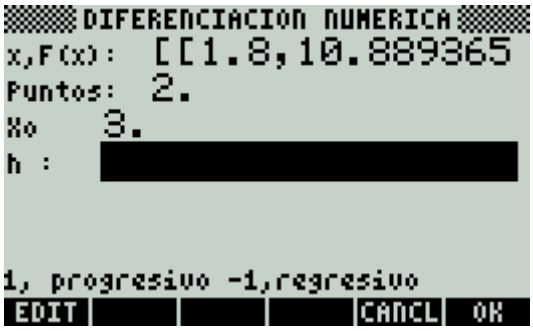




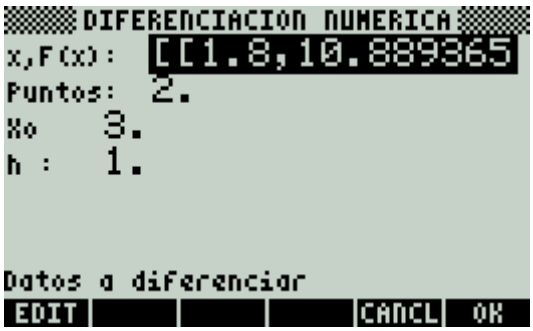
La casilla puntos, es para escoger la formula. Un 2 para la formula de 2 puntos, conocida como formula de tres puntos centrada en Xo. Un 3 para la formula de tres puntos progresivos. Un 4 para formula de cuatro puntos o conocida también como formula de cinco puntos centrado en Xo, y un 5 Para la formula de cinco puntos progresivos.

Como el valor de X = 2 esta en medio de la tabla, es conveniente usar formulas de 3 y 5 puntos centradas.

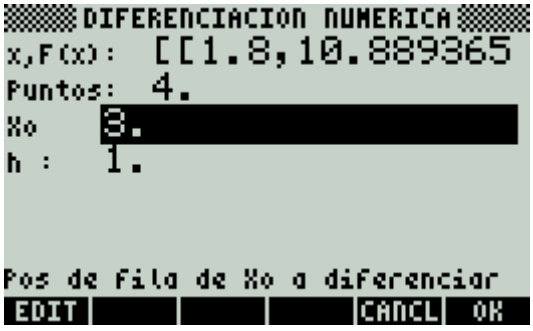
Se ingresa el valor de Xo = 3, Pues el algoritmo del programa se basa en la posición de fila del valor que se esta aproximando dentro de la matriz. Como x=2 y f (2) están en la fila 3, en Xo se ingresa el número 3.



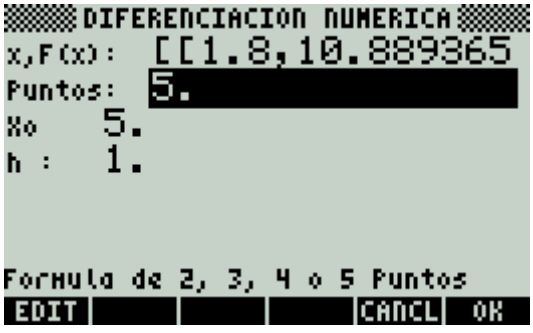
El valor h, debe ser 1, pero para las formulas progresivas h puede ser -1, para volver las formulas progresivas a formulas de puntos regresivos.



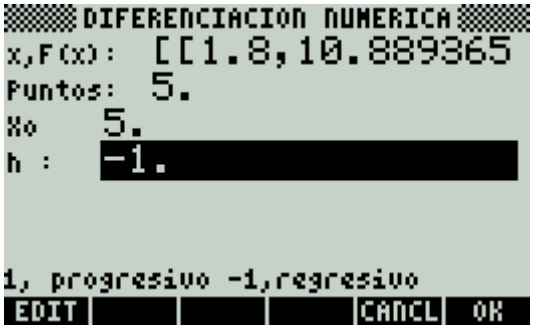
Luego con la formula de 4 puntos o formula de 5 puntos centrada.



Ahora la derivada en X = 1.8 usando formula de 5 puntos progresivos.



Ahora la derivada en X = 2.2 usando formula de 5 puntos regresivos.

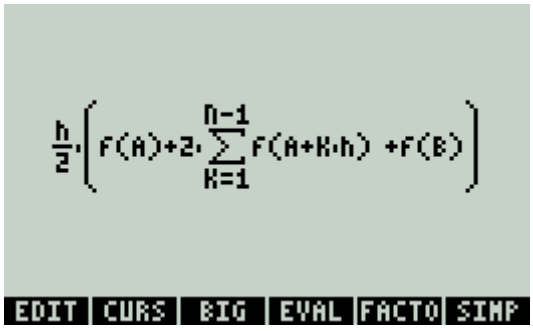


Para no hacer la matriz de datos a cada rato, solo se ingresa la variable M en la primera casilla, el valor de cada derivada se guarda en la variable d.

TRAPECIOS Y SIMPSON

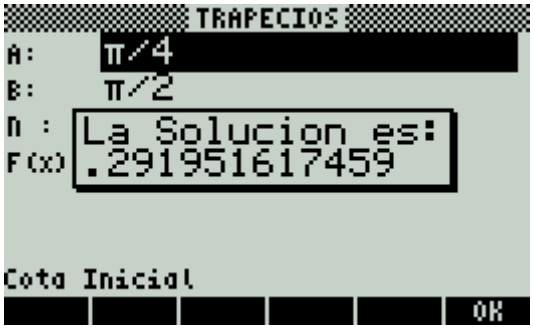
Los métodos de trapecios, Simson, Romberg y Cuadratura Gaussiana se ingresan los mismos datos, los limites de integración A y B, la cantidad de iteraciones N y la función a integrar. Para los ejemplos de trapecios y Simson se usara la función: f(x) = cos(x) de π/4 a π/4 con n = 4 y para Romberg y Cuadratura Gaussiana se usara f (x) = EXP (-x^2) de 0 a 1.

El programa de Trapecios usa el algoritmo general de la serie del método de trapecios, donde h = (A – B)/2 y f la función.



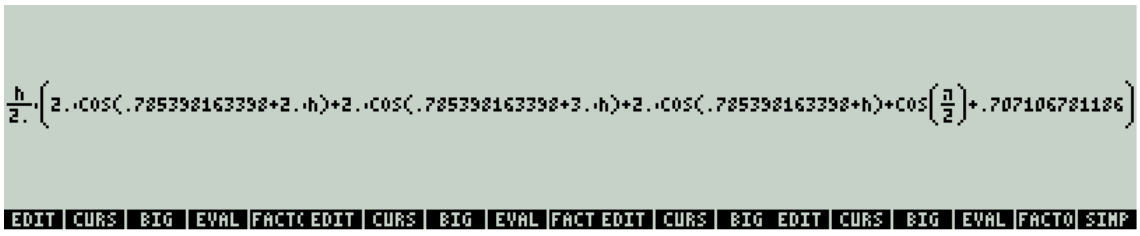
La entrada del programa es:

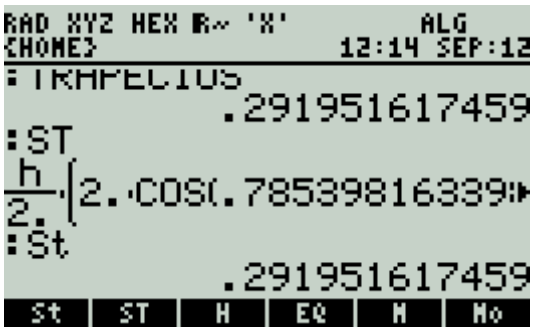
La salida es:



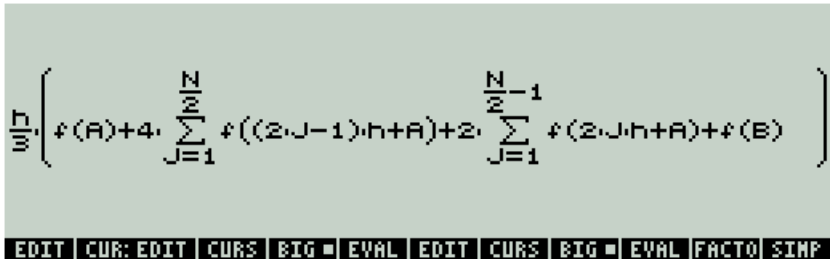
Las variables generadas son:

La variable ST y St tienen la información, sin embargo los términos de ST no están es un orden deseado, pero son los mismo que se obtienen del método de manera manual con h = (B-A)/N.

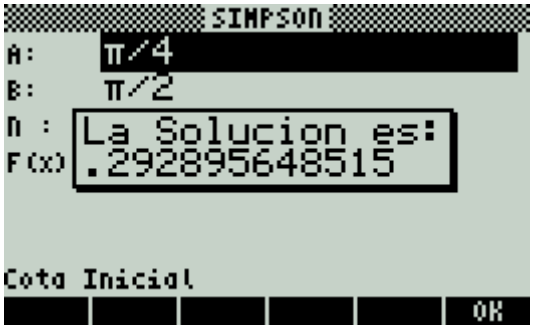




Para Simson, el algoritmo general del método es la siguiente serie.

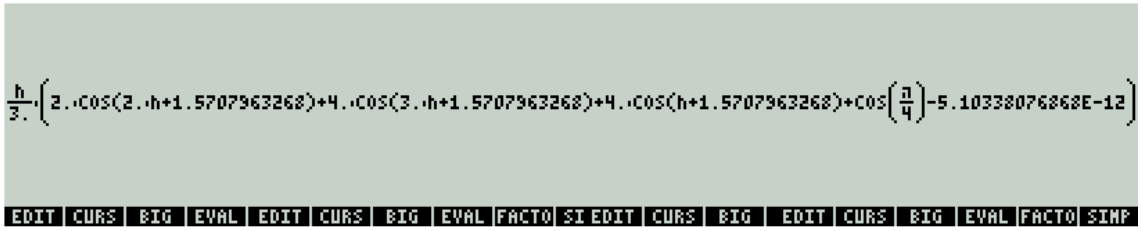


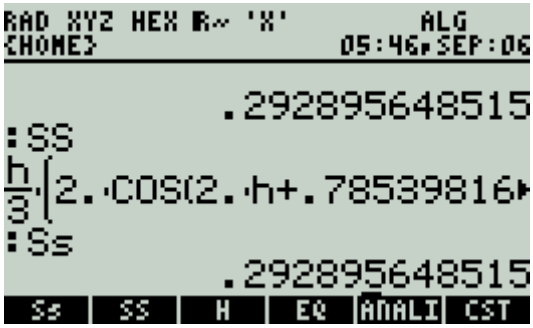
El programa ejecutándose:



Las variables que se crearon.

El contenido de las variables SS y Ss.

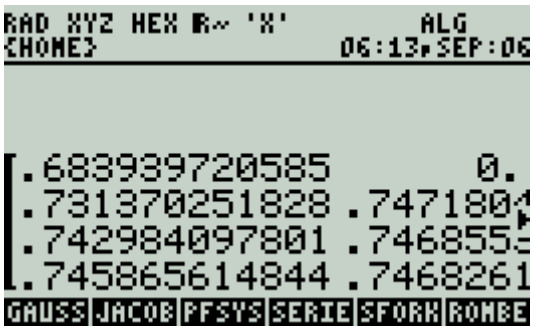




ROMBERG Y CUADRATURA GAUSSIANA

Para Romberg usamos la segunda función de ejemplo

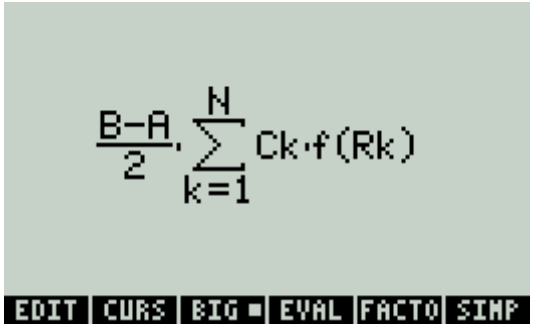
La salida del programa, son los valores de trapecios con los valores de la extrapolación de Richardson.



El ultimo valor esta en la variable Sr, que es el valor de interés.

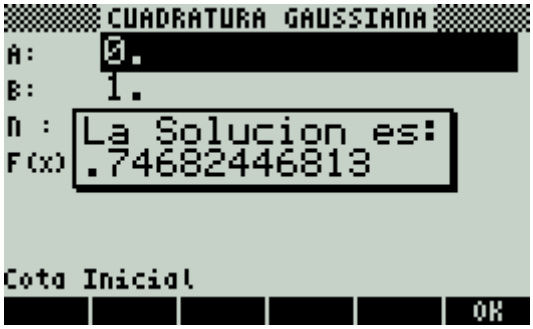
Luego viene el método de integración más complicado de la librería, y el que mas tiempo toma, el programa trabaja con cualquier valor de N hasta donde la calculadora aguante, pero mejor si se usa un N pequeño menor a 5, pues un N mayor, le tomara un poco más de cinco minutos.

El método a continuación es el de Cuadratura Gaussiana, y el algoritmo de la serie general de programa es:



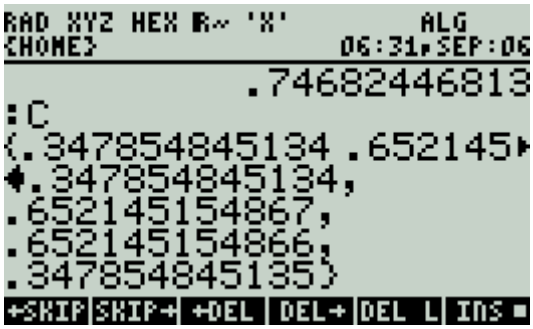
$$C_k = \int_{-1}^{1} \prod_{\substack{j=1 \\ j \neq k}}^{n} \frac{x - x_j}{x_k - x_j} dx$$

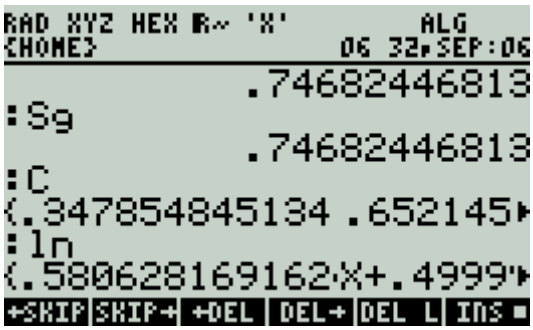
Para el ejemplo, se utiliza los mismos valores de entrada usados en el ejemplo de Romberg.

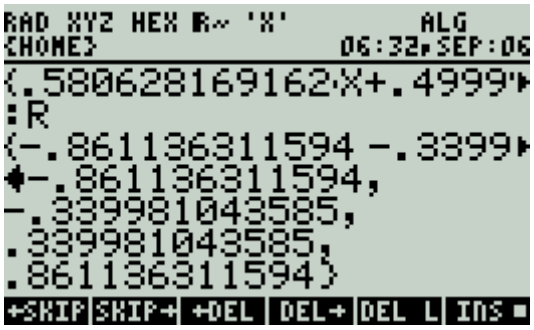


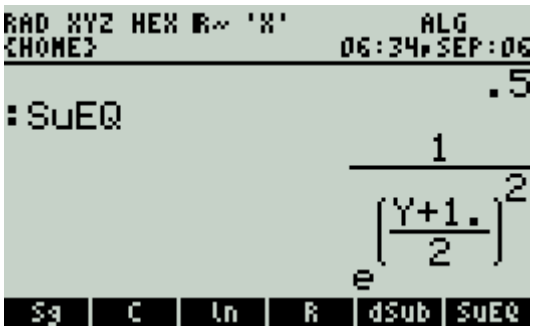
Las variables de salida son:

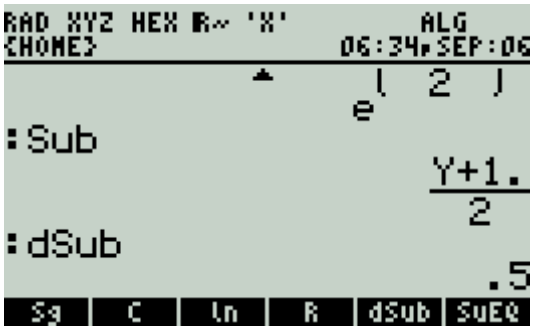
Sg tiene el valor de la integración, C son los valores de las constantes Ck obtenidos de la integración de la interpolación de Lagrange y de las raíces de los polinomios de Legendre, el Ln es exactamente el mismo que genera el programa de Lagrange, la variable R son las raíces Rk del enésimo polinomio de Legendre, Sub es la ecuación de recta en términos de Y que se sustituye en X, SuEQ es la función con la sustitución en términos de Y, y dSUB que es igual a (B-A)/2, es el diferencial extra que surgen del cambio de variable de X a Y











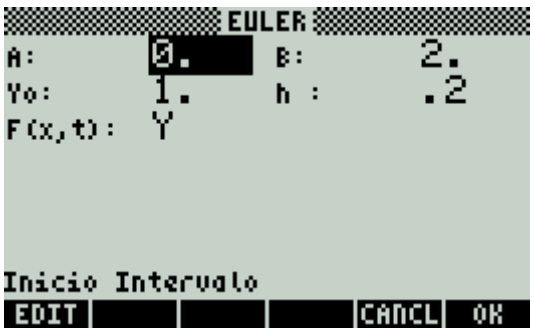
ECUACIONES DIFERENCIALES
EULER, RUNGE KUTTA 4to. GRADO, TAYLOR DE ORDEN SUPERIOR

Para ecuaciones diferenciales de valor inicial, doy dos opciones, El método simple de Euler y el método de Taylor de Orden Superior. La entrada de datos es la misma para los dos. Para el ejemplo se resuelve la ecuación diferencial
dy/dx = y o y' = y con la condición inicial y (0) = 1en un intervalo que va de 0 < x <2 h = 0.2 y n = 4.

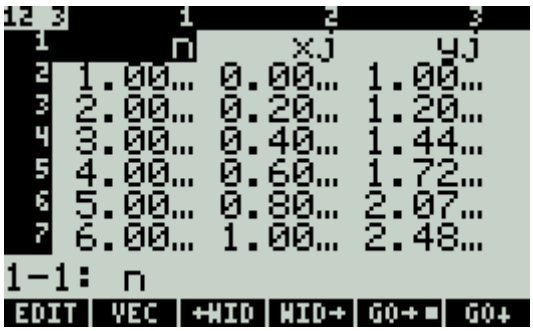
La ecuación tiene solución exacta que es y = exp. (x), pero lo usare de todos modo el ejemplo. La ecuación se introduce de la forma:

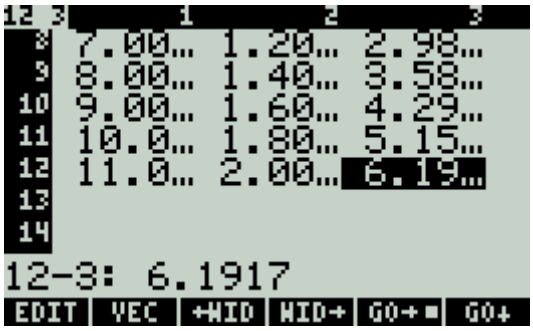
dy/dx = F(x, y)

Para resolver el problema, primero se pone en uso, el método de Euler:



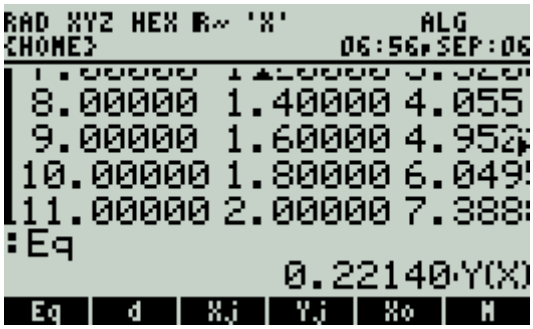
El resultado es la tabla con las iteraciones del algoritmo de Euler,



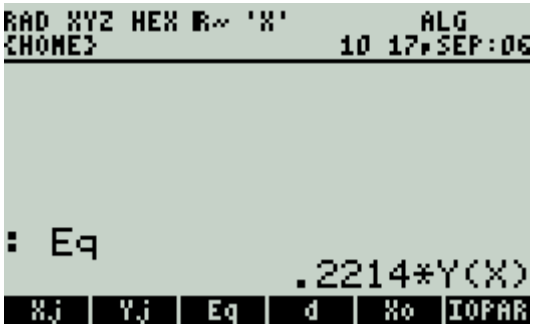


En el método de Taylor se usa el mismo ejemplo, con un orden n = 4.

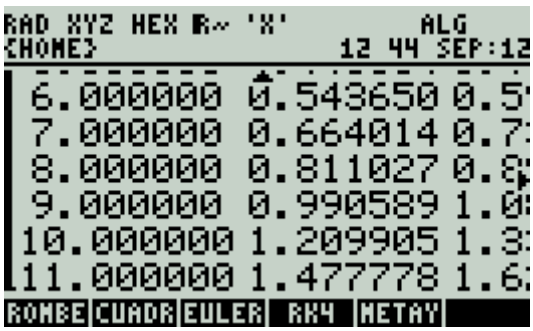
Y la salida es:

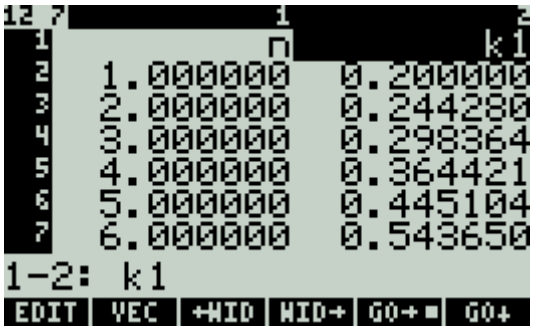


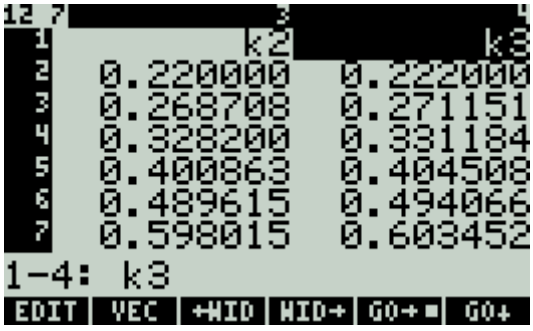
Como ve, el método es más preciso. La variable Eq, contiene la expresión de Taylor resultante.

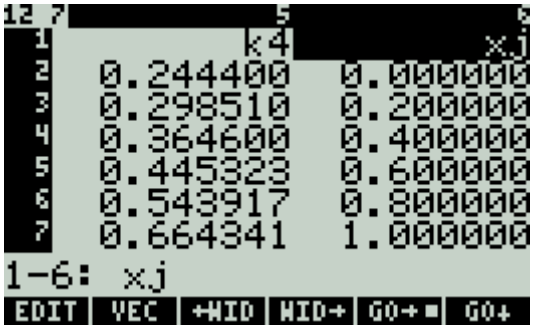


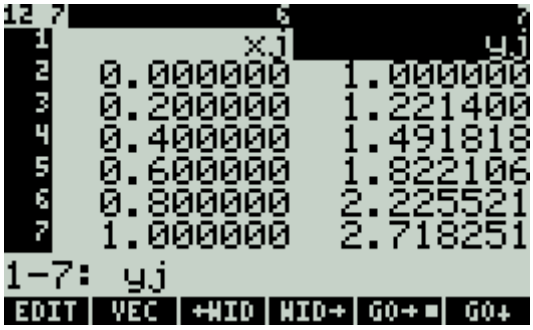
El método RK4, tiene la siguiente entrada y salida de datos:

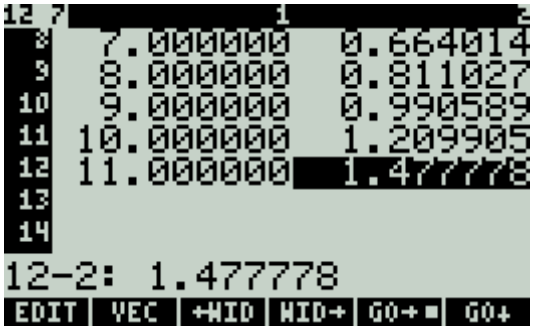


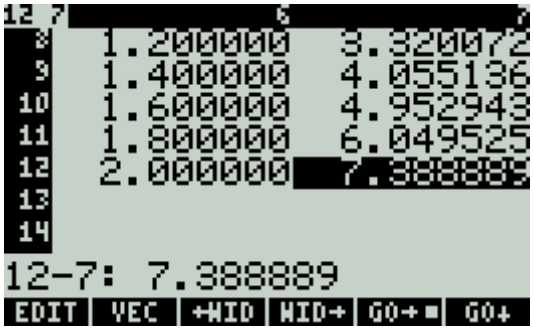












La tabla también muestra las constantes K_{jn} de cada iteración.

Comentarios y conclusiones:
Existe infinidad de versiones para algunos de los programas de la librería, entre los más comunes, están el de bisección, newton, punto fijo, secante, trapecios y Simson, entre otros, algunos talvez no tan fáciles de conseguir para estas calculadoras hp, como el de Neville, Jacobi, Gausseidel, Punto Fijo y Newton para sistemas de ecuaciones, Romberg, Cuadratura Gaussiana y método de Taylor.

En el archivo hay dos carpetas con los códigos de los programas, los cuales copiándolos, se pueden ejecutar en la calculadora, se pueden modificar, si en caso los programas no son de gusto tal como están, los programas se pueden modificar sin compromiso, pero las versiones originales en este archivo son propias de los autores. El único pago que puedes ofrecer a mi persona es un comentario a mi correo de Google manueldariofajardo@gmail.com. No acepto insultos, si los quisiera iría a un bar de mala muerte. Acepto consejos, guías o tutoriales, otros programas, comentarios, de lo que sirvió y no sirvió, recomendaciones y sugerencias, anécdotas, y Alguno que otro chiste

Puedo resolver algunas dudas, tampoco soy un Master.

Los programas están diseñados para la mayoría de calculadora hp, como dije en el comentario de adictos hp, no incluye a la vieja hp48g, pues esta no se traga el modo algebraico.

Las versiones de CAS en las que se probó la librería son:

4.2001	Revisión #2.09	2006
4.20060602	Revisión #2.09	2006
4.20031005	Revisión #1.23	2003
4.20031005	Revisión #1.24	2003

En caso de no funcionar algún programa, mándame los datos de la versión del ROM de tu calcu usando los comandos VER y VERSION que despliegan la información y la lista de variables que se generan los programas que fallen, para poder hacer correcciones.

El porque uso el modo algebraico, razones de comodidad, intente usar RPN, tiene sus ventajas al agilizar las operaciones de la calcu, pero nunca me acostumbre, porque me sacaba de onda, cuando introducía grandes cálculos, "eso de meter un valor y datos de entrada primero y luego la operación tiene ventajas, pero para la expresiones algebraicas largas, un dolor de cabeza para mí y no me gusta estar usando apostrofes a cada rato".

Espero que les sea útil, y Éxitos en sus proyectos, carreras y vidas.

Bibliografía utilizada para la programación de algoritmos y códigos.

Análisis Numérico de Richard L. Burden y J. Douglas Faires
Séptima Edición. Editorial Thomson
México 2002

Guía Avanzada de la calculadora HP49G+
Hewlett-Packard 2004.

Tutorial de Programación HP USER RPL modo Algebraico v1.2
De David Enrique Torres.

Algunos ejemplos que recibí en clase y otros que me invente en el aire.