



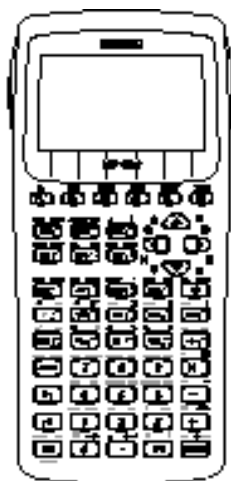
HP 49G

Guía del Usuario FETI V. 1.0

Por: Aradenătorix Veckhôm Vacelaevus

aradnix@yahoo.com

<http://www.adictoshp.org>



introducción:

FETI es una biblioteca diseñada para optimizar algunas funciones del CAS de la 49 relacionadas con el cálculo diferencial, integral y vectorial, tales como DIV, IBP, DERIV, CURL, HESS, LAP, iLAP, etc. Por lo cual solo es útil si se trabaja simbólicamente.

Asimismo quiero señalar que esta biblioteca contiene la traducción al castellano de los mensajes y una pequeña modificación del código fuente para que sea útil en la ROM 1.19-6 del programa “*Pretty Laplace*” de Jorge Aranda y una versión corregida de “*Hessiana & Jacobiana*” de Pablo Antolín. Agradezco a ambos por su cooperación y aprobación para la publicación de esta suite.

FETI está hecha en su totalidad en User-RPL, por lo cual tal vez la ejecución sea algo lenta, pero segura. Espero que les sea útil y se diviertan con ella. **NOTA:** *Debe actualizarse primero la ROM a la versión 1.19-6¹*

instalación:

Para instalar: envía FETI a tu calculadora, posteriormente almacénala en el puerto cero presionando **O** **K**. Si deseas hacer una copia de respaldo, desde **j** copia FETI y pégala en el puerto 2.

Repite el proceso anterior para JACOBI, y procura dejarlo en HOME, pues te será indispensable para la correcta ejecución de JACOBIANO.

Para borrar: asegúrate de estar en HOME o en el directorio donde hayas guardado FETI, abre una lista **÷** presionando **!** **+** y presiona la tecla de función que corresponda a FETI (**A**, **B**, **C**, **D**, **E** ó **F**), una vez hecho esto presiona **`**. Ahora presiona **!** **N** (**°**) para abrir el menú de programación y una vez dentro presiona **B** **E** **F** y listo. Si no funcionó entonces te sugiero que escribas en la pila: **CF (- 1 1 7)** **`** y vuélvelo a intentar. Te recuerdo que todo lo dicho aquí está escrito para ser usado en RPN, así que te sugiero que uses este modo (al menos para la instalación).

ejecución:

Presiona la tecla **J** para ver las variables almacenadas en HOME, y usa **!** **J** (**\$**) y **L** las veces que sean necesarias hasta ver el directorio FETI en la barra de menús. La ejecución variará de acuerdo al comando que se elija. En general, para manejar adecuadamente FETI sugiero mantener activado el “Soft menu”, es decir, el marcador -117 (para mayor información consulte la *Guía de HP49G Pocket*).

¹ He optado por la ROM 1.19-6 por ser la más reciente y estable de todas, además de que en la parte de transformadas, con ella se puede trabajar mejor con funciones como la delta de Dirac o la función de Heaviside. Y en cuanto a la evaluación de funciones o variables expresadas en términos de otra variable, tales como $f(x)$ o $y(t)$, no es lo que el usuario podría esperar, pero al menos los resultados son más entendibles en la ROM 1.19-6.

descripci on de l os programas que i ncl uye #FETI#:

En sentido estricto, FETI es un directorio que contiene 12 programas útiles para cálculo, estos son los que explicaremos en esta sección:

DIVERG

Tipo:	Programa
Descripción:	Es una optimización del comando DIV, el cual proporciona la divergencia de una función.
Acceso:	J \$FETI\$
Entrada:	Nivel n+2/ Argumento 1: El elemento primero del vector de dimensión n+2 donde n es el nivel 1. Nivel 2/ Argumento 2: El último elemento del vector. Nivel 1/ Argumento 3: La dimensión del vector (generalmente 3).
Salida:	La divergencia de la función con respecto a las variables empleadas, expresada en una forma reducida.
Marcadores:	Debe configurarse al modo exacto (marcador -105 desactivado). No debe configurarse al modo numérico (marcador -03 desactivado).
Ejemplo:	Encuentra la divergencia de la siguiente función vectorial: $\vec{v} = x^2 y \hat{i} + x^2 y \hat{j} + y^2 z \hat{k}$
Comando:	En RPN <div style="display: flex; justify-content: space-between;"> <div> 4: X^2*Y 3: X^2*Y 2: Y^2*Z 1: 3 </div> <div> DI VERG(X^2*Y, X^2*Y, Y^2*Z, 3) </div> </div> En modo algebraico:
Resultado:	(X+Y) ^2
Consulta también:	CURL, DERIV, HESS
Código fuente:	« fVn LNAME DI V Crea un vector de la dimensión especificada, proporciona un vector con los nombres de las variables contenidas en una expresión simbólica, en orden alfabético y proporciona la divergencia de una función vectorial. Expande y simplifica una expresión algebraica., luego agrupa los términos de una expresión polinómica en los reales. EXPAND COLLECT »

GRADIENTE

Tipo:	Programa	
Descripción:	Es una aplicación del comando DERIV, para obtener el vector de las derivadas parciales (gradiente) de una función vectorial.	
Acceso:	J \$FETI\$	
Entrada:	Una función.	
Salida:	El vector gradiente de la función con respecto a las variables especificadas.	
Marcadores:	Debe configurarse al modo exacto (marcador -105 desactivado). No debe configurarse al modo numérico (marcador -03 desactivado).	
Ejemplo:	Encuentra el gradiente de la siguiente función, respecto a las variables x, y e z .: $2x^2y + 3y^2z + xz$	
Comando:	En RPN $2 * X ^ 2 * Y + 3 * Y ^ 2 * Z + X * Z$	En modo algebraico: GRADI ENT E ($2 * X ^ 2 * Y + 3 * Y ^ 2 * Z + X * Z$)
Resultado:	[' 4 * Y * X + Z ' ' 2 * X ^ 2 + 6 * Z * Y ' ' X + 3 * Y ^ 2 ']	
Consulta también:	DERVX, RISCH	
Código fuente:	« LNAME DERI V EXPAND Proporciona un vector con los nombres de las variables contenidas en una expresión simbólica, en orden alfabético. Proporciona las derivadas parciales de una función. Expande y simplifica la expresión. »	

HESSIANA

Tipo:	Programa	
Descripción:	Es una forma simplificada del comando HESS que proporciona la matriz hessiana y el gradiente de una expresión.	
Acceso:	J \$FETI\$	
Entrada:	Una función.	
Salida:	Nivel 2/ Elemento 1: La matriz Hessiana con respecto a las variables empleadas. Nivel 1/ Elemento 2: El vector gradiente de la función con respecto a las variables especificadas.	
Marcadores:	Debe configurarse al modo exacto (marcador -105 desactivado). No debe configurarse al modo numérico (marcador -03 desactivado).	
Ejemplo:	Encuentra la matriz hessiana y el gradiente de la siguiente función, respecto a las variables x, y e z .: $2x^2y + 3y^2z + xz$	
Comando:	En RPN $2 * X ^ 2 * Y + 3 * Y ^ 2 * Z + X * Z$	En modo algebraico: HESSI ANA ($2 * X ^ 2 * Y + 3 * Y ^ 2 * Z + X * Z$)
Resultado:	2: [[' 4 * Y ' ' 4 * X ' 1] [' 4 * X ' ' 6 * Z ' ' 6 * Y ']	

**Consulta
también:
Código
fuente:**

1: [[1 ' 6*Y' 0]]
 1: [' 4*Y*X+Z' ' 2*X^2+6*Z*Y' ' X+3*Y^2']
 CURL, DERIV, DIV

« LNAME HESS DROP

Proporciona un vector con los nombres de las variables contenidas en una expresión simbólica, en orden alfabético; luego proporciona la matriz hessiana y el gradiente respecto a las variables especificadas y un vector de variables (LNAME). Elimina el objeto del nivel 1 de la pila, en nuestro caso, a LNAME.

EXPAND SWAP EXPAND
 SWAP

Expande y simplifica una expresión algebraica e intercambia la posición de dos objetos en los niveles 1 y 2 de la pila.

Ibid.

Ibidem.

»

IPP

Tipo: Programa

Descripción: Proporciona un algoritmo más simple y fácil de utilizar para el comando IBP, el cual efectúa la integración por partes de una función que pueda expresarse como el producto de dos funciones. **Dado que el comando IBP fue concebido para usarse exclusivamente en RPN, este programa también tendrá que ser ejecutado en RPN** pues incluye a dicho comando.

Quiero añadir además, que esta es la propuesta inicial del autor, aunque al final, como anexo agrego una versión alterna del mismo programa que requiere directamente de $u(x)$ y de $v'(x)$ como argumentos, para que el usuario que así lo desee, pueda modificar el código de éste programa que al final se explica.

Acceso: J \$FETI\$

Entrada: Nivel 2/ Argumento 1: la primera función $u(x)$.
 Nivel 1/ Argumento 2: la segunda función $v(x)$.

NOTA: Dada la disposición de los argumentos en el programa, después de elegir a $u(x)$, será necesario efectuar primero la integral de $v'(x)$ mediante INTVX o RISCH para poder aplicar IPP.

Salida: Nivel 1/ Elemento 1: El resultado final de $u(x) \cdot v(x) - \int v(x) \cdot u'(x) dx$

NOTA: Cabe señalar que la calculadora es capaz de dar el resultado final directamente tanto de una integral por partes simple como de una que requiera de integrar por partes más de una vez, siempre y cuando la integral resultante en alguna iteración no sea muy compleja.

Marcadores: Debe configurarse al modo RPN (marcador -95 desactivado).
 Debe configurarse al modo exacto (marcador -105 desactivado).
 No debe configurarse al modo numérico (marcador -03 desactivado).
 Asegúrate de que el valor actual de VX sea el mismo valor respecto al cual está trabajando.

Ejemplo: Utiliza la integración por partes para obtener la integral de:

$$\int x \cos(x) dx$$

JACOBIANO

Consulte el apéndice para ver la explicación acerca de Jacobi.

NOTA: Además, agregará dentro del directorio FETI, la variable JAC que contiene a la matriz jacobiana resultante.

$$x = e^{u-v} \quad y = e^{u+v} \quad z = e^{u+v+w}$$

5

Resultado:

```

2:      EXP( U+V+W ) W ' , 3)
1:      3
2: [ EXP( U+V ) EXP( U+V ) : JACOBI ANO( ' EXP( U- V ) ' ,
    EXP( U+V+W ) ] ' EXP( U+V ) ' , ' EXP( U+V+
1: - ( 2* EXP( U- V ) * EXP( U W ' )
    +V ) * EXP( U+( V+W ) ) ) { [ EXP( U+V ) , EXP( U+V ) , EXP
$JAC$$$N$$$I VERGRADI HESSI $I PP$ ( U+V+W ) ] ,
    - ( 2* EXP( U- V ) * EXP( U +V ) * EXP( U+( V+W
    ) ) ) }
$JAC$$$N$$$I VERGRADI HESSI $I PP$
    
```

Además, al presionar A y B , podemos conocer los valores de JAC y N respectivamente:

```

Jac:
[ [ ' EXP( U- V ) ' ' - EXP ( U- V ) ' O ]
  [ ' EXP( U+( V+W ) ) ' ' EXP( U+( V+W ) ) ' ]
  [ ' EXP( U+V ) ' ' EXP( U+V ) ' O ] ]
    
```

Consulta también:
Código fuente:

N: 3
 DERIV, INTVX, PREVAL, RISCH

```

« fVn DUP OBJ f OBJ f DROP
' N' STO JACOBI EXPAND
DUP ' JAC' STO DET
    
```

Ensambla un vector de la dimensión especificada, duplica el vector, separa las componentes del vector e indica el número de elementos en él dos veces; elimina el segundo número.
 Agrega N y guarda el valor generado por OBJ→ como N, ejecuta a JACOBI y expande el resultado obtenido que es la matriz jacobiana.
 Duplica el resultado anterior, agrega JAC y guarda dicho resultado como JAC, finalmente obtiene el determinante de la matriz JAC.

»

iLAPLACE

Tipo: Programa
Descripción: Proporciona la transformada inversa de Laplace de una expresión, y permite hacerlo en términos de s y t en vez de x . Dado que realiza la misma acción que ILAP, la expresión debe evaluarse como una fracción racional.
Acceso: J \$FETI\$
Entrada: Una función racional.

Salida: La transformada inversa de Laplace, en términos de t .
Marcadores: Debe configurarse al modo RPN (marcador -95 desactivado).
 Debe configurarse al modo real (marcador -103 desactivado).
 Debe configurarse al modo exacto (marcador -105 desactivado).
 No debe configurarse al modo numérico (marcador -03 desactivado).

Ejemplo: Encuentre la transformada inversa de Laplace de:

$$\frac{1}{(x-5)^2}$$

Comando: En RPN:

En modo algebraico:

2:
 1:
 ' 1 / (X - 5) ^ 2 '
 t * EXP (5 * t)
 LAP

i LAPLACE (1 / (X - 5) ^ 2)

Resultado:

Consulta

también:

Código

fuelle:

```
«
  I F DEPTH 0 ==
    THEN 1392 . 075 BEEP
    " ERROR:
    Pila vacía"
    MSGBOX KILL
  ELSE PATH CASDIR VX
    ' s ' VX' STO ROT { X
    s } | ILAP EXPAND { s
    t } | SWAP' VX' STO
    SWAP
    END EVAL
»
```

Inicia la estructura condicional, pide la cantidad de objetos en la pila antes de ejecutar DEPTH y prueba si el valor devuelto por DEPTH es igual con cero. Comienza la cláusula verdadera en la estructura condicional pone la frecuencia y duración del pitido y hace el sonido. Comienza la cadena del mensaje de error.

Crea un cuadro de mensaje y cancela el programa.

Comienza una cláusula alterna dentro de la estructura condicional, devuelve una lista con la ruta del directorio actual y abre el directorio CASDIR e invoca el valor actual de VX.

Agrega la variable s , agrega la variable VX y guarda el valor de s como VX, luego sustituye x por s en la expresión; ejecuta ILAP y EXPANDe la expresión. Sustituye la variable s por t e intercambia los niveles 1 y 2 y almacena como VX al objeto contenido en el nivel 1.

Intercambia nuevamente los niveles 1 y 2. Finaliza la estructura condicional y evalúa.

LAPLACE

Tipo:	Programa	
Descripción:	Efectúa la transformada de Laplace de una expresión con respecto a s y t en vez de x .	
Acceso:	J \$FETI\$L	
Entrada:	Una expresión.	
Salida:	La transformada de Laplace, en términos de s .	
Marcadores:	Debe configurarse al modo RPN (marcador -95 desactivado). Debe configurarse al modo real (marcador -103 desactivado). Debe configurarse al modo exacto (marcador -105 desactivado). No debe configurarse al modo numérico (marcador -03 desactivado).	
Ejemplo:	Encuentra la transformada de Laplace de:	
	e^x	
Comando:	En RPN:	En modo algebraico:
	2:	LAPLACE (EXP(X))
	1:	
	' EXP(X) '	
Resultado:	1 / (s - 1)	
Consulta también:	ILAP	
Código fuente:	<pre> « I F DEPTH 0 == THEN 1392 . 075 BEEP " ERROR: Pila vacía" MSGBOX KILL ELSE PATHCASDIR VX ' t ' ' VX' STO ROT { X t } LAP { t s } SWAP' VX' STO SWAP END EVAL » </pre>	
		<p>Inicia la estructura condicional, pide la cantidad de objetos en la pila antes de ejecutar DEPTH y prueba si el valor devuelto por DEPTH es igual con cero. Comienza la cláusula verdadera en la estructura condicional pone la frecuencia y duración del pitido y hace el sonido. Comienza la cadena del mensaje de error.</p> <p>Crea un cuadro de mensaje y cancela el programa.</p> <p>Comienza una cláusula alterna dentro de la estructura condicional, devuelve una lista con la ruta del directorio actual y abre el directorio CASDIR e invoca el valor actual de VX.</p> <p>Agrega la variable t, agrega la variable VX y guarda el valor de t como VX, luego sustituye x por t en la expresión; ejecuta LAP y sustituye la variable t por s.</p> <p>Intercambia los niveles 1 y 2 y guarda al objeto en el nivel 1 como VX e intercambia de nueva cuenta los niveles 1 y 2.</p> <p>Finaliza la estructura condicional y evalúa.</p>

PARTDERIV

Tipo:	Programa	
Descripción:	Proporciona las derivadas parciales de una función, con respecto a las variables empleadas.	
Acceso:	J \$FETI\$L	
Entrada:	Una función.	
Salida:	Nivel n+1/ Argumento 1: La derivada parcial de la función con respecto a la primera variable en orden alfabético, donde n es el nivel 1. Nivel 1/ Argumento 2: La última derivada parcial respecto a la última variable de la función.	
Marcadores:	Debe configurarse al modo exacto (marcador -105 desactivado). No debe configurarse al modo numérico (marcador -03 desactivado).	
Ejemplo:	Encuentra las derivadas parciales de la siguiente función, respecto a las variables x, y e z .: $2x^2y + 3y^2z + xz$	
Comando:	En RPN:	En modo algebraico:
Resultado:	$2 * X^2 * Y + 3 * Y^2 * Z + X * Z$ En RPN:	$PARTDERIV(2 * X^2 * Y + 3 * Y^2 * Z + X * Z)$ En modo algebraico:
Consulta también:	3: $4 * X * Y + Z$ 2: $2 * X^2 + 6 * Z * Y$ 1: $X + 3 * Y^2$ DERIV, DERVX	
Código fuente:	« LNAME DERI V EXPAND OBJ f DROP »	Proporciona un vector con los nombres de las variables contenidas en una expresión simbólica, en orden alfabético. Proporciona las derivadas parciales de una función. Expande la expresión algebraica, separa las componentes del vector e indica el número de elementos en él, finalmente elimina dicho número.

ROTACIONAL

Tipo:	Programa	
Descripción:	Proporciona el rotacional de una función vectorial, sin necesidad de armar el vector.	
Acceso:	J \$FETI\$L	
Entrada:	Nivel n+2/ Argumento 1: El elemento primero del vector de dimensión n+2 donde n es el nivel 1. Nivel 2/ Argumento 2: El último elemento del vector. Nivel 1/ Argumento 3: La dimensión del vector (generalmente 3).	
Salida:	El rotacional de la función vectorial con respecto a las variables empleadas.	

Marcadores: Debe configurarse al modo exacto (marcador -105 desactivado).
 No debe configurarse al modo numérico (marcador -03 desactivado).

Ejemplo: Encuentra el rotacional de la función vectorial siguiente:

$$\vec{v} = x^2 y \hat{i} + x^2 y \hat{j} + y^2 z \hat{k}$$

Comando: En RPN:

En modo algebraico:

4: X^2*Y
 3: X^2*Y
 2: Y^2*Z
 1: 3

ROTACIONAL ([X^2*Y X^2*Y Y^2*Z])

Resultado: [' 2*Z*Y' 0 ' - (X^2- 2*Y*X) ']

Consulta también: DERIV, DERVX, DIV

Código fuente: « fVn LNAME CURL

Crea un vector de la dimensión especificada, Proporciona un vector con los nombres de las variables contenidas en una expresión simbólica, en orden alfabético y obtiene el rotacional del vector.

EXPAND OBJ f OBJ f DROP

Expande la expresión algebraica, separa las componentes del vector e indica el número de elementos en él dos veces; elimina el segundo número.

' n' STO ROT EVAL

Lo guarda con el nombre n y pasa el elemento ubicado en el nivel 3 al primero y evalúa dicho elemento.

UNROT SWAP EVAL SWAP

Pasa el elemento ubicado en el nivel 1 al nivel 3, intercambia la posición de los objetos en los niveles 1 y 2, se evalúa el nivel 1 y se vuelven a intercambiar los niveles 1 y 2.

EVAL n fVn

Evalúa nuevamente el nivel 1, invoca a n y crea un vector de dimensión n .

»

→Vn

Tipo: Subrutina

Descripción: Arma un vector numérico o simbólico de la dimensión especificada. se puede decir que es una generalización de los comandos →V2 y →V3 del menú VECTOR

Acceso: J \$FETI\$L

Entrada: Nivel n+2/ Argumento 1: El elemento primero del vector de dimensión n+2 donde n es el nivel 1.

Nivel 2/ Argumento 2: El último elemento del vector.

Nivel 1/ Argumento 3: Un entero² que represente la dimensión del vector.

Salida: El vector de la dimensión especificada

Marcadores: Debe configurarse al modo exacto (marcador -105 desactivado).
 No debe configurarse al modo numérico (marcador -03 desactivado) si es que se desea armar un vector simbólico.

Ejemplo: Ensamble el vector siguiente:
 $\vec{u} = x - 3^2 \hat{i} + x^2 y \hat{j} + z + 4 \hat{k}$

Comando: En RPN: En modo algebraico:

4: (X- 3) ^2 fVn(' (X- 3) ^2' , ' X^2*Y' , ' Z+4' , 3)
 3: X^2*Y
 2: Z+4
 1: 3

Resultado: [' (X- 3) ^2' ' X^2*Y' ' Z+4']

Consulta también: →V2, →V3

Código fuente: « fLI ST AXL Crea una lista de la dimensión especificada, y luego convierte la lista en un vector.
 »

DELEND

Tipo: Programa

Descripción: Simplemente borra los objetos y variables creados al ejecutar algunos programas de FETI, o bien al ocurrir un error en la ejecución, tales como las variables JAC y n para evitar así, que te estropeen cálculos posteriores. El nombre lo tome de la celebre frase de Marco Porcio Catón “el Censor” (234-149 a.C.) con la que siempre concluía sus discursos: DELEND EST CARTAGO.

Acceso: J \$FETI\$L

Entrada: Ninguna.

Salida: Ninguna, si es que lo hiciste correctamente.

Marcadores: Ninguno en especial que yo sepa.

Ejemplo: Elimine las variables JAC y n:

Comando: En RPN: En modo algebraico:

RAD XYZ DEC R= 'X'
 ^T CALCULO FETI}

5:
 4:
 3:
 2:
 1:

DELEND

JAC n OIVER GRADI HESSI IPP

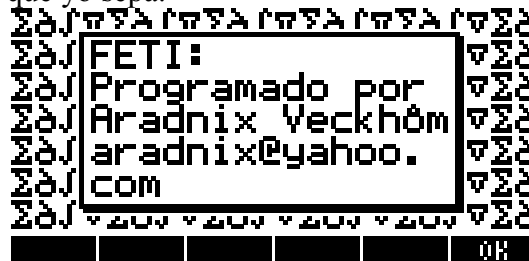
Resultado: En RPN: En modo algebraico:

² Hago énfasis de que sea un entero, por sentido común, pero si se quisiera dar como argumento un numero racional o irracional, la calculadora inmediatamente hará algo parecido a ORND para redondearlo al entero más próximo sin importar los argumentos presentes, si no me creen, hagan la prueba con π o con e .

	RAD XYZ DEC R= 'X' \$T CALCULO FETI\$	RAD XYZ DEC R= 'X' \$T CALCULO FETI\$	ALG
	5: 4: 3: 2: 1:		
	DIVER GRADI HESSI IPP JACOB ILAPL		NOVAL
Consulta también:	PURGE.		
Código fuente:	« { n N JAC U V } PURGE »	:DELEND A Una lista que contiene los nombres de los objetos a purgar. Purga los objetos incluidos en la lista.	

ACERCA

Tipo:	Programa
Descripción:	Simple, tranquilo e inofensivo programita que indica el nombre de la biblioteca, el nombre del autor y mi dirección de correo electrónico para cualquier queja, sugerencia, comentario, mentada, etc, con un pequeño fondo muy <i>ad hoc</i> con el tema de los programas aquí descritos. Mil disculpas a todos aquellos a los que el nombre de este programa les interfiera con alguna otra biblioteca.
Acceso:	J \$FETI\$L
Entrada:	Ninguna
Salida:	Un fondo con un mensaje
Marcadores:	Ninguno en especial que yo sepa.
Ejemplo:	



Código fuente:	Esta vez tendrán que averiguarlo por ustedes mismos.
----------------	--

apendice:

Una vez cubierta la explicación de cada uno de los programas, vale la pena aclarar ciertos puntos que creo, no han quedado del todo claros hasta ahora.

En algunos programas como GRADIENTE, PARTDERIV, ROTACIONAL o DIVERG, el uso de $\rightarrow V_n$ permite armar vectores de la dimensión que sea, desde 1 hasta ∞ , claro que,

generalmente trabajaremos con dimensiones 2 y 3. Asimismo, LNAME nos asegura que los objetos sobre los que trabaje serán en estricto orden alfabético lo cual es muy útil al momento de interpretar el resultado final. Pero ten cuidado, porque si ingresas una función de 3 variables y en la función solo especificas 2 (por ejemplo) estos programas lo tomarán como una función de dos variables.

En el caso de iLAPLACE y LAPLACE, que son una versión mejorada de la versión de Jorge Aranda, tenemos la ventaja de que podemos usar indistintamente x , s ó t , según sea el caso de las dos últimas variables. Claro que por la disposición inmediata de X en el teclado, lo más cómodo es usar X , y dada la estructura de ambos programas, podemos hacer uso del resultado en términos de s y t con estos programas sin mayor problema. Por ejemplo, si yo quiero obtener la transformada inversa de Laplace de $\frac{1}{x-5^2}$, aplico iLAPLACE y obtengo:

<pre> RAD XYZ DEC R= 'X' ~T CALCULO FETI~ ----- 3: 2: 1: </pre>	$\frac{1}{(X-5)^2}$	<pre> RAD XYZ DEC R= 'X' ~T CALCULO FETI~ ----- 4: 3: 2: 1: </pre>	$t \cdot e^{(5 \cdot t)}$
<pre> DIVER GRADI HESSI IPP JACOBI LAPL i LAPL </pre>			

Ahora si yo quiero recuperar la expresión original, aplico la transformada de la place mediante LAPLACE y entonces:

<pre> RAD XYZ DEC R= 'X' ~T CALCULO FETI~ ----- 4: 3: 2: 1: </pre>	$t \cdot e^{(5 \cdot t)}$	<pre> RAD XYZ DEC R= 'X' ~T CALCULO FETI~ ----- 3: 2: 1: </pre>	$\frac{1}{(s-5)^2}$
<pre> DIVER GRADI HESSI IPP JACOBI LAPL LAPLACOLCT DIVER GRADI HESSI IPP JACOBI LAPL </pre>			

Como vemos, el resultado que obtuve no está expresado en términos de x como pudiera pensarse, sino en términos de s , y VX sigue valiendo X , por lo que puedo seguir utilizando X sin problemas. Por esta razón, creo que se puede aplicar parte de la estructura de LAPLACE o iLAPLACE, para crear aplicaciones similares que les permitan trabajar en términos de una variable diferente a X sin afectar sustancialmente el valor de VX . Cabe mencionar que para aquellos que necesiten usar transformadas y transformadas inversas de Laplace para cuestiones de electrónica o en variable compleja, pueden activar y desactivar el modo complejo mediante el siguiente atajo: „ I , el cual les permitirá fácil y rápidamente pasar del modo real al complejo y viceversa de ser necesario.

Ahora bien, anteriormente me comprometí a dar en esta sección una forma alterna de IPP, así como de describir la subrutina JACOBIANO. Pues bien, agarrense que ahí les van:

IPP (Versión alterna)

Tipo:	Programa	
Descripción:	Proporciona un algoritmo más simple y fácil de utilizar para el comando IBP, el cual efectúa la integración por partes de una función que pueda expresarse como el producto de dos funciones. Dado que el comando IBP fue concebido para usarse exclusivamente en RPN, este programa también tendrá que ser ejecutado en RPN pues incluye a dicho comando.	
Acceso:	J \$FETI\$	
Entrada:	Nivel 2/ Argumento 1: la primera función $u(x)$. Nivel 1/ Argumento 2: la segunda función $v(x)$.	
Salida:	Nivel 1/ Elemento 1: El resultado final de $u(x) \cdot v(x) - \int v(x) \cdot u'(x) dx$ NOTA: Cabe señalar que la calculadora es capaz de dar el resultado final directamente tanto de una integral por partes simple como de una que requiera de integrar por partes más de una vez, siempre y cuando la integral resultante en alguna iteración no sea muy compleja.	
Marcadores:	Debe configurarse al modo RPN (marcador -95 desactivado). Debe configurarse al modo exacto (marcador -105 desactivado). No debe configurarse al modo numérico (marcador -03 desactivado). Asegúrate de que el valor actual de VX sea el mismo valor respecto al cual está trabajando.	
Ejemplo:	Utiliza la integración por partes para obtener la integral de: $\int x \cos(x) dx$	
Comando:	2: ' X' 1: COS (X)	
Resultado:	SI N(X) * X + COS (X)	
Consulta también:	INT, INTVX, PREVAL, RISCH	
Código fuente:	« f U V « U V * V I NTVX I BP I NTVX + » »	Declara los valores de las variables internas U y V. Invoca a U y V y luego las multiplica, llama a V aplica integración por partes a los dos argumentos anteriores. Integra al argumento en el nivel 1 respecto a VX y suma los niveles 2 y 1 para dar el resultado final.

Como puede verse, el programa se simplifica de esta manera, y puede resultar más fácil de usar para algunas personas, sin embargo, a veces no resulta tan sencillo el integrar a $v(x)$ lo cual podría conllevar a algún error al usar esta versión alterna; por lo cual, personalmente prefiero efectuar dicha operación de antemano antes de usar IPP.

Pero tal vez el usuario final de estos programas encuentre más cómoda y poco peligrosa esta versión que aquí propongo. Así que tal vez en alguna versión posterior, agregue definitivamente

esta versión, pero para ello, vosotros tendréis primero que mandarme dicha petición por e-mail ¿vale?

Pasando a algo más constructivo, he aquí la subrutina JACOBI:

JACOBI

Tipo: Subrutina

Descripción: A partir de la descomposición del vector de transformación T y de la variable N, obtiene la matriz jacobiana.

NOTA: Para la correcta ejecución de JACOBI se requiere ejecutar desde la pila todos los comandos contenidos en el programa JACOBIANO hasta antes de JACOBI³, o bien, ingresar los argumentos que se enuncian más adelante.⁴

Acceso: J \$FETI\$

Entrada: Nivel n+1/ Argumento 1: El elemento primero del vector de dimensión n+1 donde n es el nivel 1.
 Nivel 1/ Argumento 2: El último elemento del vector.

NOTA: Además, se deberá agregar dentro del directorio FETI, la variable N con la dimensión del vector de transformación T.

Salida: Nivel 1/ Elemento 1: La matriz jacobiano.

Marcadores: Debe configurarse al modo RPN (marcador -95 desactivado).
 Debe configurarse al modo exacto (marcador -105 desactivado).
 No debe configurarse al modo numérico (marcador -03 desactivado).

Ejemplo: Encuentra el jacobiano de la transformación:

$$x = e^{u-v} \quad y = e^{u+v} \quad z = e^{u+v+w}$$

Comando: En RPN: En modo algebraico:

Resultado:

```

3:          EXP( U- V )  JACOBI ANO( ' EXP( U- V ) ' ,
2:          EXP( U+V )   ' EXP( U+V ) ' , ' EXP( U+V+
1:          EXP( U+V+W ) W ) ' )
[ [ ' EXP( U- V ) ' ' - EXP( U- V ) ' O ]
  [ ' EXP( U+( V+W ) ) ' ' EXP( U+( V+W ) ) ' ]
  [ ' EXP( U+V ) ' ' EXP( U+V ) ' O ] ]
    
```

Consulta también: DERIV, INTVX, PREVAL, RISCH

Código fuente:

```

«
  CASE N 2 ==
    THEN
      « DROP DROP
  LNAME ' V' STO DUP
    
```

Inicia el ciclo y pide la primera condición: si N es igual a 2. Entonces, si la condición es cierta, ejecuta la subrutina: Elimina el objeto en el nivel 1 de la pila 2 veces. Ensambla el vector de la dimensión

³ Para mayores detalles consulte el código fuente de JACOBIANO.

⁴ Otra opción es insertar el comando KILL antes de la instrucción JACOBI en el código fuente de JACOBIANO para obtener el mismo resultado sin necesidad de efectuar las operaciones manualmente.


```

OBJ f DROP SWAP V

DERI V SWAP V DERI V 2

ROW f TRAN ' V' PURGE
    » EVAL
END N 3 ==

THEN
    « DROP DROP

DROP LNAME ' V' STO

DUP OBJ f DROP V DERI V

UNROT V DERI V UNROT V

DERI V UNROT 3 ROW f

' V' PURGE
    » EVAL
END
    
```

especificada, agrega la variable V la guarda en la memoria duplica el vector.

Separa las componentes del vector e indica el número de elementos contenidos en él; elimina dicho valor, intercambia los argumentos en los niveles 1 y 2, y llama a V.

Deriva a V respecto a la primera variable en orden alfabético, intercambia los niveles 1 y 2, llama de nuevo a V y lo deriva respecto a la segunda variable.

Forma una matriz de 2 renglones, la transpone y purga la variable V.

Evalúa el resultado.

Termina la instrucción. Si la primera condición no fuese cierta, se ejecuta la segunda condición: si N es igual a 3.

Entonces, si la condición es cierta, ejecuta la subrutina:

Elimina el objeto en el nivel 1 de la pila 2 veces.

Elimina una vez más el objeto en el nivel 1 de la pila, 2 veces, ensambla el vector de la dimensión

especificada, agrega la variable V la guarda en la memoria.

Duplica el vector, separa las componentes del vector e indica el número de elementos contenidos en él; elimina dicho valor, llama a V y la deriva respecto a la primera variable.

Desplaza el objeto contenido en el nivel 1 al nivel 3, llama a V y la deriva respecto a la segunda variable, de nuevo desplaza el objeto del nivel 1 al 3 y llama a V.

Deriva a V respecto a la última variable, mueve al elemento del nivel 1 al 3 y forma una matriz de 3 renglones.

Elimina a V.

Evalúa el resultado final

Finaliza la instrucción.

END
»

Finaliza el programa.

Bien, este es el corazón de JACOBIANO, pues dicho programita, consta de una estructura iterativa o ciclo, en el cual, según las condiciones de los argumentos, se evalúa una de las tres condiciones que contiene. En otras palabras, lo que hace JACOBI, es utilizar la dimensión del vector T de transformaciones, para generar la matriz jacobiana, y a partir de ella regresar al programa jacobiano, para obtener el determinante de dicha matriz.

Como se puede observar, la estructura de JACOBI esta diseñada para obtener un resultado a partir de 2 condiciones, pues la condición $N=1$ puede omitirse, pues dado que es imposible obtener el determinante de una matriz de 1×1 , veo como innecesaria dicha condición.

Además como una propuesta se podría intentar obtener una generalización para un caso donde $N > 3$.

De hecho, si sois observadores, veréis que dada la complejidad del caso, este programa se ajusta perfectamente para una división modular según el axioma “Divide y vencerás”.

Ya por último señalo que iré revisando y depurando los programas para versiones futuras, de igual forma espero agregar nuevos programas a FETI dentro de la misma temática, para complementarla.

l i g a s a s i t i o s d e i n t e r e s :

Ya para concluir, se me ocurre presentar una pequeña lista de ligas a páginas relacionadas con cuestiones de las calculadoras HP:

- <http://www.hpcalc.org>: la meca de las calculadoras hp. En esta página pueden encontrar casi todo lo que deambula por la red en cuanto a estas fabulosas calculadoras, ya que es raro que se les escape algo. Desde aquí pueden bajar programas, documentos, etc., así como subir los propios (seguramente aquí fue donde encontraron FETI, ¿no es cierto?).
- <http://www.adictoshp.org>: Modestia aparte, es una de las mejores páginas para encontrar información acerca de las hp. Contiene foros donde se pueden exponer dudas e interactuar con usuarios de toda Latinoamérica y España (principalmente) , así como una sección de descargas con programas, artículos y demás; por cierto, desde aquí pueden también actualizar la ROM de su HP para lo cual encontrarán una guía donde se explica bastante bien, como actualizar la ROM.

Por otra parte, creo que para algunos usuarios, sobre todo si estudian ingeniería en electrónica o telecomunicaciones, esta página les será de utilidad:

- <http://pagina.de/hp49>: Es la página de Jorge Aranda Moro, coautor de FETI. Aquí encontrarán herramientas útiles para aquellos que estudian las carreras arriba mencionadas, además de que contiene ligas a sitios de interés más general.

Además, si de programar se trata, pueden bajar los siguientes manuales, que listo en el orden que considero más pedagógico, aunque los dos primeros son sospechosamente similares:

- En <http://hpuser.8m.com> pueden encontrar el primer manual, escrito por el señor Jorge Broncazo Reyes, el cual aunque es breve, los introduce al ambiente de programación en User RPL.
- Por otra parte, en <http://www.angelfire.com/ok/hp48g> podrán encontrar el manual de Luis M. Gallardo D. el cual tampoco es muy extenso pero contiene varios ejemplos realmente ilustrativos.
- Por último, en <http://www.angelfire.com/co/48gx/49g.html> pueden encontrar, el más completo de los tres manuales, escrito por el Ing. Jorge Ayala Niño de Guzmán, que fuera de algunas faltas de ortografía, es un muy buen manual (mi manual de cabecera) para aprender User RPL. La verdad, dada la antigüedad del texto, no me fío mucho del vínculo del manual, así que si la página a la que hago referencia ya no existe, les recomiendo buscar en <http://www.sethiansweb.com/hpcivil/>, desde ahí podrán llegar a la última versión disponible.

A fin de cuentas, lo que hace al maestro es la práctica.

Ya para concluir esta sección, si lo que desean es conseguir un buen manual para su HP 49G, les sugiero, bajar la guía de la nueva (aun en la fecha en que escribí esta guía) calculadora hp 49g+, la cual se encuentra en PDF, y pesa algo así como 10 MB, o aproximadamente 7 MB si la encuentran comprimida en formato ZIP.

Hasta la fecha de publicación de FETI V 1.0, la guía está disponible en la página de Hewlett Packard, en la siguiente liga:

http://h20015.www2.hp.com/en/manualCategory.jhtml?reg=&cc=us&pagetype=manual&prodId=hp49ggraph351775&lc=en&docparent=manual&sw_lang=es&cat=manual&plc=null

epi l o g o:

Agradezco la enorme paciencia de todos los usuarios que hayan llegado hasta esta sección. Les pido me disculpen por la exhaustiva explicación que he plasmado en este mamotreto ¡de casi 20 páginas! (en verdad no me imaginé que fuese taaaan largo). Espero asimismo que el esfuerzo de leer esta guía haya sido útil para entender cabalmente la función y estructura de cada uno de los programas que se incluyen en esta suite. Ojalá no se hayan aburrido demasiado con ella, y lo principal, que les sea útil en sus cursos de análisis matemático y cálculo.

Les reitero que estoy abierto a críticas, sugerencias y propuestas. Les recuerdo además que todo el que quiera los puede modificar para mejorarlos...eso si...envíenme una copia a mi, no sean ingratos. Por cierto si hay algún ocioso que tenga el interés de compilar FETI en Sys RPL, sea bienvenido. Para ello pueden enviar cualquier sugerencia, propuesta o duda a cualquiera de las siguientes direcciones:

- aradnix@yahoo.com
- aradenatorix@yahoo.de
- revolt_veckhom_jones_cuevas@yahoo.com.au
- aradnix@list.ru

Si han de hacer uso de ellas, por favor envíen su mensaje sólo a una de ellas, pues reviso todas mis cuentas casi a diario, por lo que solo en caso de que les reboten el mensaje, hagan favor de reenviéndolo a otra dirección, pues seguramente la que es cogieron está saturada de momento.

Hemos llegado al fin de esta guía, espero que la hayan disfrutado y los haya dejado con menos dudas.

Ciudad de México, Diciembre de 2003.
Aradenătorix Veckhôm Vacelaevus.

Este texto es propiedad intelectual de Víctor Daniel Vela Cuevas (o que, ¿creían que Aradnix era mi nombre verdadero?) © (2003) y no puede ser utilizado ni modificado sin su previa y expresa autorización.