

MuPAD

Álgebra Lineal

Prof. José Neville Díaz Caraballo

1. ¿Qué es MuPAD?

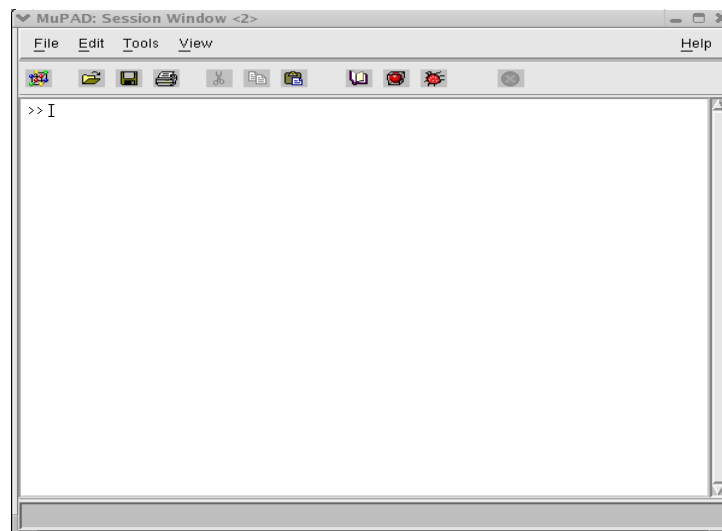
MuPAD es un programado que resuelve problemas simbólicos, numéricos, además de crear gráficas. Existen versiones para Windows, Unix y Linux. Este documento estará presentado en Linux Fedora Core 4.

2. ¿Cómo obtener MuPAD?

La version 2.5 de **MuPAD** es gratis para uso académico. Lamentablemente desde la salida de su versión 3 no existen versiones gratis. Para obtener su copia debe hacer una búsqueda con google (MuPAD light 2.5).

3. Pantalla Inicial

El Programa luce de esta manera:



Los comandos deben ser escritos en la línea de comandos >> . Para ejecutar debes oprimir "enter" o "shift enter". Esto depende de cómo lo hayas configurado en el menú de view. MuPAD es sumamente sensitivo con los espacios, así que debes tener cuidado con lo que escribes.

4. Operaciones y funciones.

MuPAD utiliza prácticamente los mismos símbolos y operaciones comunes de otros paquetes algebraicos simbólicos o gráficos.

<i>Símbolo</i>	<i>Significado</i>	<i>Ejemplo</i>	<i>Significado</i>
+	suma	$x+1$	$x + 1$
-	resta	$x-1$	$x - 1$
*	multiplicación	$3*x$	$3x$
/	división	$x/3$	$x \div 3$
^	exponente	x^2	x^2
-	opuesto	-3	-3
:=	asignación	$x:=3$	$x=3$
sqrt(expr)	raíz cuadrada	sqrt(x)	\sqrt{x}
exp(expr)	exponencial	exp(x)	e^x
sin(x)	seno	sin(x-3)	$\sin(x-3)$
cos(x)	coseno	cos(2*x)	$\cos(2x)$
tan(x)	tangente	tan(x-1)	$\tan(x)$
ln(x)	logaritmo natural	ln(x+4)	$\ln(x+4)$
abs(x)	valor absoluto	abs(2*x-1)	$ 2x-1 $
int(f,x)	integral	int(x^2,x)	$\int x^2 dx$
limit(f, x = x ₀)	límite	limit(2*x+1,x=infinity)	$\lim_{x \rightarrow \infty} 2x + 1$
diff(f,x)	derivada	diff(x^2,x)	$\frac{d}{dx} x^2$
solve(f = 0, x)	resolver, raíces	solve (x ² + 3x + 1 = 0, x)	$x^2 + 3x + 1 = 0$
subs(f = 0, x	intercepto en y	subs (x ² + 3x + 1 = 0, x)	$0^2 + 3(0) + 1 = 1$
plotfunc3d((f,x=-a..b, y=c..d)	gráfica en 3 dimensiones	plotfunc3d((x+y),x=-5..5, y=-5..5)	$f(x,y)=x+y$
plotfunc2d(f,x=a..b)	gráfica en 2 dimensiones	plotfunc2d(x^2,x=-5..5)	$f(x)=x^2$

5. ¿Cómo obtener ayuda en MuPAD?

Para solicitar ayuda en MuPAD, desde la línea de comandos >> escriba ?nombre, por ejemplo >>?int abrirá una pantalla con la ayuda que tiene sobre int (integral).

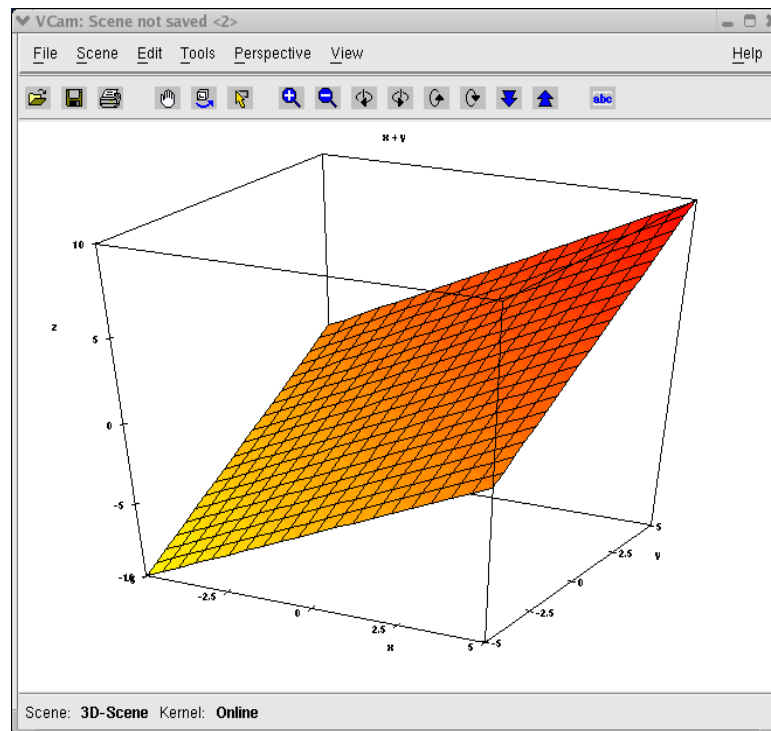
6. Ejemplos en MuPAD

Los comandos serán presentados en la letra Nimbus Mono L, por ejemplo plotfunc2d.

- Gráfica de $f(x, y) = x + y$

```
>> plotfunc3d((x+y), x=-5..5, y=-5..5)
```

$x=-5..5$ representa el intervalo que se presentará en la gráfica. Similarmente para $y=-5..5$.



- Calcular la integral $\int \sin(x) dx$

```
>> int(sin(x), x)
```

$-\cos(x)$

- *Cálculos de fracciones*

>> 1/5 + 2/3 -5

-62/15

- *Cálculo de $\sqrt{12}$*

>> sqrt(12)

1/2
2 3

- *Simplificar $\sin^2(x)+\cos^2(x)$*

>> simplify((sin(x))^2 +(cos(x))^2)

1

- *Definiendo la función $f(x)=\frac{(x^2-5*x+6)}{(x-1)}$*

>> f:=(x^2-5*x+6)/(x-1)

$$\frac{x^2 - 5x + 6}{x - 1}$$

- *Encontrando las raíces de $f(x)=\frac{(x^2-5*x+6)}{(x-1)}$*

>> solve(f=0,x)

{2, 3}

- *Encontrando el intercepto en y de $f(x)=\frac{(x^2-5*x+6)}{(x-1)}$*

>> subs(f,x=0)

-6

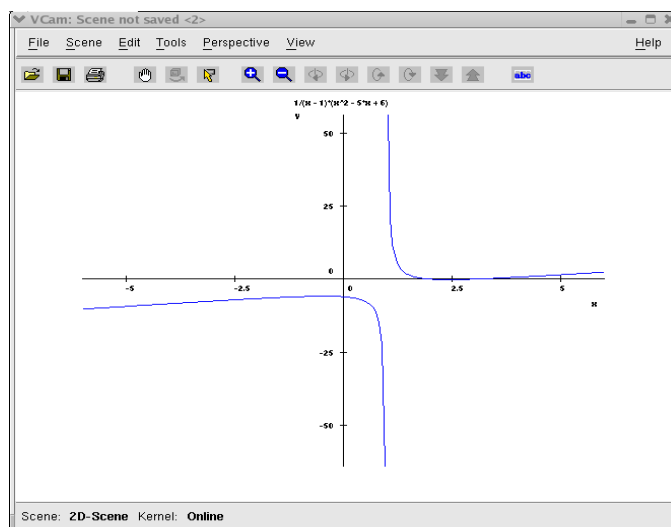
- *Calculando la derivada*

>> f2:=diff(f,x)

$$\frac{2x - 5}{x - 1} - \frac{x^2 - 5x + 6}{(x - 1)^2}$$

- *Haciendo la gráfica de* $f(x) = \frac{(x^2 - 5x + 6)}{(x - 1)}$

>> plotfunc2d(f,x=-6..6)



- *Calculando el límite* $\lim_{x \rightarrow \infty} \frac{10 - x}{x^2 + 1}$

>> limit(((10-x)/(x^2+1)),x=infinity)

0

- *Calculando la* $f(x) = \int x \cos x^2 dx$

>> F:=int(cos(x^2)*x,x)

$$\frac{\sin(x^2)}{2}$$

- Calculando la integral $\frac{1}{2} \int_{-\pi/6}^{\pi/6} (3\cos(3z))^2 dz$

```
>> F:=.5*int((3*cos(3*z))^2,z=-PI/6..PI/6)
```

0.75PI

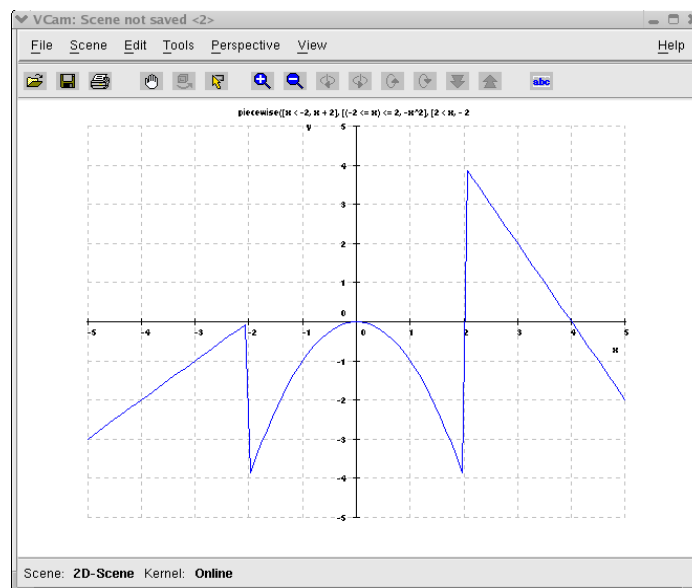
- Función a trozos $f(x) = \begin{cases} x+2 & x < -2 \\ -x^2 & -2 \leq x \leq 2 \\ -2x+8 & x > 2 \end{cases}$

```
f:=piecewise([x<-2,x+2],[-2<=x<=2,-x^2],[x>2,-2*x+8])
```

```
piecewise(x + 2 if x < -2, - x^2 if (-2 <= x) <= 2, - 2 x + 8 if 2 < x)
```

- Haciendo la gráfica $f(x)$

```
>> f:=piecewise([x<-2,x+2],[-2<=x<=2,-x^2],[x>2,-2*x+8]):
plotfunc2d(BackGround = RGB::White, ForeGround=
RGB::Black, GridLines=Automatic,Ticks=[Steps =1, Steps
=1],f(x),x=-5..5,y=-5..5)
```



- *Evaluando una función a trozos*

```
>> eval(subs(f(x), x=1))
```

-1

- *Calculando la doble integral* $\int \int_R (f(x,y)) dA$ donde $f(x,y) = 1 - 6yx^2$ y

$R: 0 \leq x \leq 2, -1 \leq y \leq 1$

$$\int \int_R (f(x,y)) dA = \int_{-1}^1 \int_0^2 (1 - 6yx^2) dx dy$$

```
>> int(int(1-6*y*x^2, x=0..2), y=-1..1)
```

- *Calculemos la descomposición por fracciones parciales de* $f = \frac{1}{x^2 + 5x + 6}$

```
>> f:=1/(x^2-5*x+6)
```

$$\frac{1}{x^2 - 5x + 6}$$

```
>> partfrac(f)
```

$$\frac{1}{x - 3} - \frac{1}{x - 2}$$

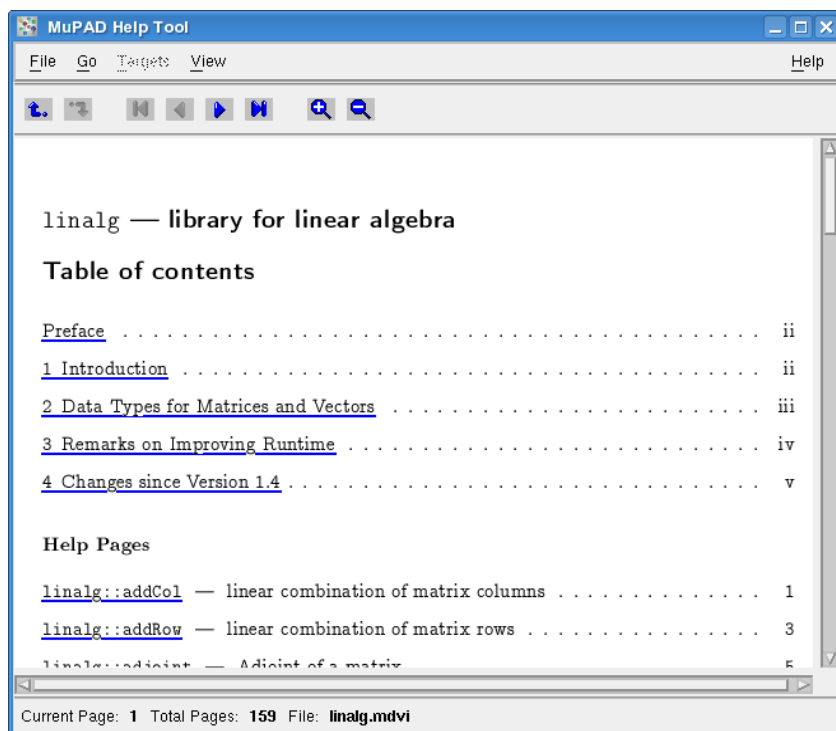
7. Álgebra Lineal

El paquete que lleva a cabo las rutinas de álgebra lineal es `linalg`. Este paquete contiene una buena cantidad de subrutinas. La manera más simple de familiarizarse con él es utilizando la ayuda.

```
>> ?linalg
```

El signo de pregunta nos envía al archivo de ayuda del comando `linalg`.

Como podemos observar contiene un gran número de subrutinas. Una vez identificada la que deseamos usar solo oprima en el nombre y recibirá una explicación, así como unos ejemplos. Observemos la pantalla de ayuda para el paquete `linalg`.



- Definir una matriz con dimensión 3x4. La entrada de datos se hace por filas.
`>> A:=matrix(3,4,[[1,-1,1,8],[2,3,-1,-2],[3,-2,-9,9]])`

```

+-          +-
|  1, -1,  1,  8  |
|                |
|  2,  3, -1, -2  |
|                |
|  3, -2, -9,  9  |
+-          +-

```

- Eliminación Gaussiana (mucho cuidado con los signos y mayúsculas)

```
>> B:=linalg::gaussElim(A)
```

```

+-          +-
|  1, -1,  1,  8  |
|                |
|  0,  5, -3, -18  |
|                |
|  0,  0, -57/5, -57/5  |
+-          +-

```


Universidad de Puerto Rico en Aguadilla

Note que solo convierte la matriz en una triangular superior. Vamos a utilizar otras funciones para lograr la matriz identidad.

Éste comando nos va a permitir ahorrar un poco en la escritura.

```
>> export(linalg,addRow,swapRow,multRow)
```

```
>> multRow(B,3,-5/57)    Mutliplica la fila 3 por -5/57.
```

```
+ - - +
| 1, -1, 1, 8 |
|
| 0, 5, -3, -18 |
|
| 0, 0, 1, 1 |
+ - - +
```

```
>> addRow(% ,3,2,3)    La fila 3 la multiplicas por 3 y se la sumas a la fila 2.
```

```
+ - - +
| 1, -1, 1, 8 |
|
| 0, 5, 0, -15 |
|
| 0, 0, 1, 1 |
+ - - +
```

```
>> addRow(% ,3,1,-1)    La fila 3 la multiplicas por -1 y se la sumas a la fila 1.
```

```
+ - - +
| 1, -1, 0, 7 |
|
| 0, 5, 0, -15 |
|
| 0, 0, 1, 1 |
+ - - +
```

Universidad de Puerto Rico en Aguadilla

>> multRow(% , 2, 1/5) Multiplicas la fila 2 por 1/5.

$$\begin{array}{cccc|c} +- & & & & -+ \\ | & 1, & -1, & 0, & 7 & | \\ | & & & & & | \\ | & 0, & 1, & 0, & -3 & | \\ | & & & & & | \\ | & 0, & 0, & 1, & 1 & | \\ +- & & & & -+ \end{array}$$

>> addRow(% , 2, 1, 1) La fila 2 la multiplicas por 1 y se la sumas a la fila 1.

$$\begin{array}{cccc|c} +- & & & & -+ \\ | & 1, & 0, & 0, & 4 & | \\ | & & & & & | \\ | & 0, & 1, & 0, & -3 & | \\ | & & & & & | \\ | & 0, & 0, & 1, & 1 & | \\ +- & & & & -+ \end{array}$$

La solución del sistema es $x=4$, $y=-3$, $z=1$

El proceso que acabamos de lograr se le conoce como eliminación Gauss Jordan.

MuPAD tiene esta rutina bajo el nombre de `linalg::gaussJordan`.

>> `linalg::gaussJordan(A)`

$$\begin{array}{cccc|c} +- & & & & -+ \\ | & 1, & 0, & 0, & 4 & | \\ | & & & & & | \\ | & 0, & 1, & 0, & -3 & | \\ | & & & & & | \\ | & 0, & 0, & 1, & 1 & | \\ +- & & & & -+ \end{array}$$

- Esta rutina te presenta solo las soluciones.

```
>> linalg::matlinsolve(A)
```

$$\begin{array}{cc} +- & -+ \\ | & 4 & | \\ | & & | \\ | & -3 & | \\ | & & | \\ | & 1 & | \\ +- & -+ \end{array}$$

- Calcular el determinante de una matriz.

```
>> B:=linalg::delCol(A,4)    Estamos eliminandole la columna 4, esto nos
permite tener una matriz cuadrada.
```

$$\begin{array}{ccc} +- & & -+ \\ | & 1, & -1, & 1 & | \\ | & & & & | \\ | & 2, & 3, & -1 & | \\ | & & & & | \\ | & 3, & -2, & -9 & | \\ +- & & & & -+ \end{array}$$

```
>> linalg::det(B)    Calculando el determinante de la matriz B.
```

-57

- Resolver el siguiente sistema de ecuaciones:

```
>> solve({x-y+z=8, 2*x+3*y-z=-2, 3*x-2*y-9*z=9}, {x,y,z})
```

{[x = 4, y = -3, z = 1]}

```
>> linsolve({x-y+z=8, 2*x+3*y-z=-2, 3*x-2*y-9*z=9}, {x,y,z})
```

[x = 4, y = -3, z = 1]

Universidad de Puerto Rico en Aguadilla

>> C:=linalg::delCol(A,4) Este comando le elimina la columna 4 a la matriz A.

$$\begin{array}{ccc|c} +- & & & +- \\ | & 1, & -1, & 1 & | \\ | & & & & | \\ | & 2, & 3, & -1 & | \\ | & & & & | \\ | & 3, & -2, & -9 & | \\ +- & & & +- & \end{array}$$

>> F:=1/C Calculando la inversa.

$$\begin{array}{ccc|c} +- & & & +- \\ | & 29/57, & 11/57, & 2/57 & | \\ | & & & & | \\ | & -5/19, & 4/19, & -1/19 & | \\ | & & & & | \\ | & 13/57, & 1/57, & -5/57 & | \\ +- & & & +- & \end{array}$$

>> G:=linalg::delCol(A,1) Eliminando las columnas de manera recursiva. Le eliminamos a la matriz A la columna 1.

$$\begin{array}{ccc|c} +- & & & +- \\ | & -1, & 1, & 8 & | \\ | & & & & | \\ | & 3, & -1, & -2 & | \\ | & & & & | \\ | & -2, & -9, & 9 & | \\ +- & & & +- & \end{array}$$

>> G:=linalg::delCol(G,1)

$$\begin{array}{cc|c} +- & & +- \\ | & 1, & 8 & | \\ | & & & | \\ | & -1, & -2 & | \\ | & & & | \\ | & -9, & 9 & | \\ +- & & +- & \end{array}$$

Universidad de Puerto Rico en Aguadilla

```
>> G:=linalg::delCol(G,1)
```

```
+--      +-  
|      8  |  
|      |  |  
|     -2  |  
|      |  |  
|      9  |  
+--      +-
```

```
>> F*G      Calculando las soluciones
```

```
+--      +-  
|      4  |  
|      |  |  
|     -3  |  
|      |  |  
|      1  |  
+--      +-
```

```
>> A:=matrix(3,2,[[1,1],[0,1],[1,1]])  Definiendo una matriz 3x2.
```

```
+--      +-  
|  1, 1  |  
|      |  |  
|  0, 1  |  
|      |  |  
|  1, 1  |  
+--      +-
```

Universidad de Puerto Rico en Aguadilla

>> QR:=linalg::factorQR(B) Calculando la factorización QR.

```

-- +-
| |      1/2      1/2      1/2      1/2      1/2      |
| |      14      14      195      65      1083      |
| |      -----, - -----, ----- |
| |      14      210      285      |
| |      |
| |      1/2      1/2      1/2      1/2      1/2      |
| |      14      22 14      195      65      1083      |
| |      -----, -----, ----- |
| |      7      1365      3705      |
| |      |
| |      1/2      1/2      1/2      1/2      1/2      |
| |      3 14      5 14      195      65      1083      |
| |      -----, - -----, - ----- |
| |      14      546      741      |
-- +-

```

```

+-
| |      1/2      | |
| |      1/2      14      1/2      | |
| |      14      , - -----, - 2 14      | |
| |      14      | |
| |      |
| |      1/2      1/2      1/2      1/2      | |
| |      14      195      4 14      195      | |
| |      0, -----, ----- | |
| |      14      65      | |
| |      |
| |      1/2      1/2      | |
| |      65      1083      | |
| |      0,      0, ----- | |
| |      65      | |
+-

```

>> QR[1]*QR[2] Verificando nuestro resultado.

```

+-          +-
|  1, -1,  1 |
|              |
|  2,  3, -1 |
|              |
|  3, -2, -9 |
+-          +-

```

>> [L,U,pivlist]:=linalg::factorLU(A) Calculando la factorización LU. Note como podemos guardar los argumentos.

```

-- +-          +-          +-          --
|  |  1, 0, 0 |  |  1, 1 |  |              | |
|  |          |  |  |   |  |              |
|  |  0, 1, 0 |, |  0, 1 |, [1, 2, 3] |
|  |          |  |  |   |  |              |
|  |  1, 0, 1 |  |  0, 0 |  |              |
-- +-          +-          +-          --

```

>> L*U Verificando nuestro resultado.

```

+-          +-
|  1, 1 |
|        |
|  0, 1 |
|        |
|  1, 1 |
+-          +-

```

>> B:=matrix(2,3,[[1,1,100],[.9,1,100]]) Creando una matriz 2x3. Este es un sistema de ecuaciones donde la inversa es difícil de calcular para la computadora.

```

+-          +-
|  1,  1, 100 |
|              |
|  0.9, 1, 100 |
+-          +-

```

Universidad de Puerto Rico en Aguadilla

>> B:=linalg::delCol(B,3) Eliminando la columna 3.

```
+ -      - +  
|  1,  1 |  
|      |  
| 0.9, 1 |  
+ -      - +
```

>> inverse:=1/B

```
+ -      - +  
| 10.0, .0.0 |  
|      |  
| -9.0, 10.0 |  
+ -      - +
```

>> inverse*B

```
+ -      - +  
| 1.0, 0.0 |  
|      |  
| 0.0, 1.0 |  
+ -      - +
```

>> C Vamos a calcular la transpuesta de la matriz C.

```
+ -      - +  
| 7, 1, 7 |  
|      |  
| 0, 8, 4 |  
+ -      - +
```

>> Ct:=linalg::transpose(C) Calcular la transpuesta.

```
+ -      - +  
| 7, 0 |  
|      |  
| 1, 8 |  
|      |  
| 7, 4 |  
+ -      - +
```


8. Referencias

MuPAD- A practical Guide, Kai Gehrs & Frank Postel.

Calculus Seventh Edition, Larson, Hostetler & Edwards. Houghton Mifflin.

<http://faculty.prairiestate.edu/skifowit/htdocs/classes/mupad.htm>

<http://faculty.prairiestate.edu/skifowit/htdocs/classes/mupad2.htm>

<http://faculty.prairiestate.edu/skifowit/htdocs/classes/mufails.htm>

<http://faculty.prairiestate.edu/skifowit/htdocs/classes/m172.htm>

<http://www.sciface.com/STATIC/DOC30/eng/tutorium.html>