ARTÍCULO 4

La revolución del cálculo simbólico matemático en aplicaciones computacionales

PORFIRIO SUÑAGUA SALGADO[1]

Resúmen

La revolución del cálculo simbólico en el ambiente computacional ha marcado un cambio trascendental en la aplicación de nuevas tecnologías educativas en el proceso de enseñanza y aprendizaje de las matemáticas con recursos de movimiento a colores que se pueden lograr con un poco de conocimiento en programación orientada a objetos. Algunas aplicaciones por excelencia que hacen este tipo de cálculos son: REDUCE, MAPLE, SMP, MATHEMATICA, muMATH, DERIVE, AXIOM, entre otros. En este trabajo presentaremos las bondades de MATHEMATICA con todas sus potencialidades importantes.

4.1 Antecedentes

4.1.1. Orígenes de la Computación Matemática

MUCHO antes de la aparición de las computadoras, célebres matemáticos desarrollaron algoritmos para resolver problemas de procesos relativamente complejos con soluciones aproximadas. En general los métodos consistían en una mezcla de cálculos numéricos y algebraicos, como por ejemplo se puede destacar los complicados cálculos matemáticos realizados por los astrónomos en el siglo XIX para el estudio de las órbitas de los planetas. El caso conocido es el del astrónomo francés Delaunay que estuvo 20 años trabajando en la obtención de una fórmula para determinar la posición de la Luna, los primeros 10 años fueron dedicados a obtener dicha fórmula y los otros 10 para comprobarla y estudiar las perturbaciones producidas por errores de medida. Curiosamente esta fórmula fue aceptada y usada satisfactoriamente durante décadas, sin que nadie notara que había tres términos erróneos. Este descubrimiento fue gracias al cálculo simbólico y unas veinte horas de cálculo computacional.

Con la aparición de las primeras computadoras, las técnicas del cálculo numérico sufrieron un gran impulso. Los primeros ordenadores estuvieron dedicados principalmente a procesos de simulación, optimización, análisis numérico, etc. como una gran ayuda solo en el cálculo enteramente numérico con gran velocidad. Fue entonces cuando muchos algoritmos, ya conocidos en los periodos anteriores, comenzaron a poder usarse de forma efectiva. Los trabajos sobre derivación analítica mediante el computador de Kahrimanian y Nolan en el año 1953 son considerados como el primer intento del uso del ordenador para la manipulación algebraica formal y los orígenes de una nueva disciplina conocida por varios nombres: *cálculo simbólico, cálculo formal* o *álgebra computacional*. Los primeros sistemas fueron MACAULAY (especializado en anillos), LIE (álgebra lineal), CoCoA (Ideales de Polinomios), GAP (grupos), SHEEP (relatividad).

[1] psunagua@umsa.bo, Universidad Mayor de San Andrés

4.1.2. Principales Características de los Sistemas de Cálculo Simbólico

LA mayor dificultad que encontró la computación simbólica con respecto a la numérica, es que el hardware no estaba diseñado para la manipulación de expresiones algebraicas, siendo por tanto mas laboriosa la programación de tales sistemas por que en este caso depende de la longitud de la palabra del ordenador.

Otra característica de estos sistemas es la naturaleza de los datos que manejan; son necesarios estructuras especiales para la representación de datos no numéricos. En el aspecto del almacenamiento cabe destacar el uso de listas, estructuras el árbol y en general estructuras con administración dinámica de memoria.

Desde el punto de vista computacional, otra propiedad importante es el uso, y en muchos casos abuso, de la técnica de la recursividad. La recursividad permite al programador escribir un programa de manera sencilla pero a costa de una mayor lentitud y uso de demasiada memoria.

4.1.3. Sistemas Generales del Cálculo Simbólico

A continuación describimos algunos sistemas conocidos y usados actualmente:

REDUCE Es quizás el mas ampliamente conocido y usado de estos sistemas, sus orígenes se remontan al año 1963, inicialmente fue desarrollado para resolver problemas de física de alta energía. Su creador es Anthony C. Hearn. Tras su evolución se convirtió en el primer sistema de cálculo simbólico de propósito general, estando disponible a partir del año 1970, el programa fue escrito en LISP (unas 50000 líneas de código). Alguna de sus principales características es pa portabilidad y la modularidad, lo cual permite un fácil mantenimiento y extensión.

MACSYMA Fue desarrollado en el MIT (*Massachusetts Institute of Technology*) entre los años 1969 y 1982, pero no fue comercializado hasta los mediados de los 70. Al igual de REDUCE está escrito en LISP y contiene gran número de funciones predefinidas lo cual le hace un sistema muy poderoso.

MAPLE Desarrollado en la Universidad de Waterloo de Canadá a partir del año 1980, no fue hasta 1983 cuando estuvo disponible. Su núcleo de MAPLE está escrito en lenguaje C, gracias a lo cual resulta muy rápido. Posee además un gran número de librerías escritas en un lenguaje propio similar al Pascal.

SMP Escrito por Stephen Wolfram a finales de los setenta y principios de los ochenta, constituye el germen del sistema MATHEMATICA, por esta razón se abandonó su desarrollo. Inicialmente estaba dirigido a los físicos e ingenieros.

MATHEMATICA Surgió como una ampliación de SMP, también escrito y desarrollado por Stephen Wolfram, su lanzamiento se produjo el año 1988. Destaca del resto de los sistemas sobre todo en sus posibilidades gráficas, tanto bidimensional como tridimensional con recursos de animación de imágenes. Su implementación está hecha en C.

muMATH Su desarrollo alcanzó a finales de los setenta y fue concebido para la enseñanza.
Constituyó el primer sistema de cálculo simbólico creado específicamente para micro ordenadores, está escrito en LISP. Sus autores son David Stoutemyer y Albert Rich.

DERIVE Surgió como una ampliación de muMATH en 1988, por lo que está escrito también en LISP.

SCRATCHPAFD Es uno de los proyectos más ambiciosos en el ámbito del Álgebra Computacional. Fue desarrollado por IBM desde los mediados de los setenta. Permite manejar objetos mucho mas abstractos y resulta muy útil para las matemáticas puras, por lo que la complejidad de su uso es mucho mayor. Ha sido un producto experimental hasta la aparición comercial con el nombre de AXIOM.

AXIOM Constituye la versión comercial de SCRATCHPAFD. Su presentación en Europa se produjo en 1982. Se puede considerar como un prototipo de la siguiente generación de sistemas de Algebra Computacional. Su implementación está en LISP y requiere de máquinas IBM.

4.2 Introducción

COMO se ha visto anteriormente, hay una gran variedad de sistemas computacionales que fueron desarrollados con el objetivo de realizar operaciones de manera simbólica, una de ellas es MATHEMATICA que me ha llamado la atención por su simplicidad de uso de sus objetos (funciones) y la opción de programación modular y una variedad de comandos de necesidad en el quehacer matemático que permite construir nuevas funciones de manera sencilla. Además las gráficas que se producen son extraordinariamente flexibles en cuanto a especificación mediante sus parámetros. Por lo tanto en lo que sigue describiremos este sistema, mostrando más adelante muchos ejemplos de aplicación que pueden ser usadas como recursos didácticos en el Proceso de Enseñanza y Aprendizaje de las matemáticas especialmente en las Universidades.

4.2.1. Diferentes Concepciones de MATHEMATICA

Una calculadora simbólica: Mathematica permite trabajar con expresiones simbólicas. Es decir, uno puede definir una función de una cierta variable x, quedando esta almacenada tal como es y no en forma de algoritmo, de modo que la función pueda ser evaluada no solo en valores numéricas sino en cualquier otro símbolo de la naturaleza de su dominio.

Una herramienta de cálculo simbólico: Mathematica permite derivar e integrar funciones y resolver ecuaciones algebraicas y diferenciales en forma simbólica. también puede calcular límites, manipular series de potencias, utilizar transformadas clásicas como de Laplace, Fourier, etc.

Un paquete gráfico: Permite dibujar en dos y tres dimensiones, elegir las perspectivas, los puntos de vista, el sistema de representación, el sistemas de coordenadas y además soporta animaciones.

Un lenguaje de programación de alto nivel: Mathematica permite la programación en tres niveles como sigue:

1. Programación de tipo procedural con uso de bloques, iteraciones, ciclos, recursiones, etc.
2. Programación funcional, con la posibilidad de definir funciones puras, operadores funcionales, etc.
3. Programación basada en reglas, suministrando reglas que indican cómo operar o transformar las expresiones simbólicas, las funciones, etc.

Un sistema para crear documentos interactivos: Incluyen texto, gráficos, sonidos, animaciones, etc.

Un sistema de apoyo a otros programas: otros programas pueden comunicarse con Mathematica, enviarle tareas y recibir resultados de las tareas realizadas.

4.2.2. Diferentes Interfaces de Mathematica

MATHEMATICA es capaz de ser utilizado en diferentes ordenadores por existir muchas versiones para cada plataforma estructuralmente diferente. La parte principal se llama *Kernel* o *núcleo* es idéntica en todas las versiones, sobre este núcleo se constituyen diferentes interfaces (*front, end*), que ya son específicos de cada ordenador de modo que puedan aprovecharse el ordenador de manera de tener un rendimiento óptimo.

Muchos de los interfaces soportan unos documentos llamados *libros de notas* o (*notebooks*). Estos documentos permiten recibir los comandos y la edición de programas en orden jerárquico, texto, gráficos y grupos de ellos (*animaciones*). Uno de sus usos es el de crear materias pedagógico en los que se combinan textos explicativos, gráficos y animaciones. Además los gráficos generados por Mathematica pueden traducirse fácilmente en formatos Postscript, EMF, WMF, lo cual permite el intercambio con otros programas. Similarmente las salidas pueden ser guardadas como en otros formatos como TEX, Fortran, C, RichText. Por otra parte la comunicación con otros programas es posible gracias a que soporta el estándar de comunicaciones *MathLink*.

4.3 Sintaxis

EN el programa Mathematica se pueden distinguir dos grandes partes. Una de ellas es el *núcleo*, que es la encargada de ejecutar todos los comandos y realizar todos los cálculos que ello conlleve. La otra consiste en la interface de usuario (*front, end*) Existe un tipo especial que permite generar documentos interactivos en los que se mezclan gráficos y textos y en el que se incluirán todos los comandos de Mathematica a evaluar por el núcleo. A este tipo de documentos se les denomina apuntes o anotaciones (*notebooks*) y en este caso, la interface se dice que es de tipo *apuntes*.

Como Mathematica es un programa interactivo, los comandos de ingreso o las ordenes se marca con `In[n]` y los resultados correspondientes se muestra en `Out[n]` el espacio donde se ingresa las ordenes, así como el de los resultados se conoce con el nombre de *celdas*. Además en número `n` se reinicia cada vez que se inicia el *kernel*.

1. Todas las ordenes o comandos en una celda ejecutamos con `<Shift><Enter>` Cada línea de comando en una misma celda debe estar separado por "`;`". El comando de ingreso una vez ejecutado el mismo se pondrá en `In[n]` y su resultado estará en `Out[n]`

2. El símbolo de multiplicación para dos números reales o complejos es el asterisco "`*`" o simplemente un espacio " ", Ej : `5*7 = 5 7`. Los demás operaciones son tradicionales, como por ejemplo 3^5=`3^5`

3. El símbolo porcentaje "`%`" hace referencia al último resultado computado, también el doble porcentaje "`%%`" hace referencia al penúltimo resultado obtenido, etc.

4. La variables así como las constantes pueden ser una cadena de caracteres alfanuméricas, de modo que el espaciador para la multiplicación es importante especialmente entre variables, es decir "`ax`" no es lo mismo que "`a x`", pero si no hay problema con constantes conocidas como "`2x`" si lo es como "`2*x`", siempre que la constante conocida esté por delante como coeficiente.

5. Las letras MAYÚSCULAS y las minúsculas no son lo mismo, de hecho todas las funciones, opciones, variables y constantes importantes en el programa comienzan con mayúsculas, salvo las definidas por el usuario que será como lo definan.
   Ejemplo:

   `Pi =` π (el número irracional "pi")

   `I=` i (la unidad imaginaria de los números complejos)

   `E=` e (el número irracional $e = \exp(1)$)

   `Infinity=` ∞ (el infinito)

Exp (la función exponencial)

Sin (la función seno trigonométrico)

Log (la función logaritmo natural o neperiano)

Sqrt (la función raíz cuadrada $\sqrt{\cdot}$)

N[x,n] (Comando que expresa en número x con n cifras significativas)

6. Las *llaves*, *corchetes* y *parentesis*:
   - Los paréntesis se utilizan normalmente para agrupar e indicar prioridad, sin embargo el texto comprendido entre "`(*`" y "`*)`" no se evalúa considerándose, por lo tanto, como comentario
   - Los corchetes son exclusivos de las funciones y son los que delimitan los argumentos de las mismas. Ejemplo: `Sin[x]` que no es lo mismo que `Sin(x)`
   - Las llaves, se utilizan para definir las *listas* de elementos separados por comas. Por ejemplo {`2,a,b`}. Una matriz no es mas que una lista de listas en la que el elemento (i,j) corresponde al j-ésimo de la lista que ocupa el lugar i-ésimo. Ejemplo:

$$\{\{1,2,3\},\{4,5,6\},\{7,8,9\}\} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

7. Para obtener una información de las funciones o de sus opciones el programa dispone de un sistema de AYUDA (Help). Si se teclea "?" seguido de una secuencia de caracteres, el resultado será un listado de todos los comandos que contienen la cadena dada (donde el * es el comodin que representa cualquier caracter).
   Ejemplo: `In[2]:=?Graphics* <Shift><Enter>`
   `Graphics GraphicsArray GraphicsData`
   `GraphicsGrouping GraphicsSpacing Graphics3D`

   También se puede explorar la AYUDA desde `Help Browser` del Kernel donde está agrupada por categorías y también se puede buscar un cierto tema o comando. Las páginas de ayuda contienen una descripción del comando, su sintaxis y algunos ejemplos.

   Para ver la sintaxis de los comandos se puede solicitar con
   `Information[`*comando*`]`,
   donde además le mostrará todas las opciones que acepta la función.

8. Algunos comandos especializados no funcionan directamente, porque son parte de un paquete especial, por lo que antes de ejecutar el comando se debe cargar el paquete correspondiente con la sintaxis
   ``<<Contexto`NombreDelPaquete` ``,
   como por ejemplo
   ``<<Graphics`ParametricPlot3D` ``

9. En una ventana adicional aparece una variedad de símbolos con sus sintaxis correspondientes las mismas haciendo un `Clic` con el *mouse* se puede insertar.
   Además desde el *Menú Principal* del Kernel `File` seguido de `Palettes` se puede conseguir para otras categorías

10. Las operaciones con matrices son en "+" para la adición y **"."** para la multiplicación. Para la multiplicación por un escalar a una matriz se utiliza el "$*$". El mismo operador aplicado a vectores o matrices de la misma dimensión se multiplica componente a componente como la multiplicación de Hadamard.

11. Las funciones se pueden evaluar en una *lista de argumentos*, donde la función se aplica a cada elemento de la lista.
    Ejemplo: `Cos[{a,x+1,2}]={Cos[a],Cos[x+1],Cos[2]}`

12. Las *reglas* son asignaciones de variables con otras variables o constantes. Ejemplo: `x->Pi`

13. Las animaciones se pueden producir marcando en la celda que las agrupa todas las gráficas y hacer CRTL-Y con el teclado, la animación se mostrará sobre el espacio mas visible.

14. Para hacer que suenen los comandos sonoros, puede hacer “click” con el ratón en la esquina superior de la celda que contiene el sonido como gráfico.

4.4 Aplicación Mathematica

ALGUNOS de los comandos de Mathematica están ilustradas en el Anexo 4.4.1, las mismas pueden ser probadas en el sistema para repetir las salidas y luego probarlas con situaciones análogas. En el resumen se muestra una variedad de comandos que corresponden a las áreas de aritmética básica, resolución de ecuaciones y desigualdades, límites, derivación, integración, gráficas de revolución, animación de gráficos, resolución de ecuaciones diferenciales y algunos programas que cumplen algunas tareas específicas y finalmente se incluye una pequeña parte de aplicaciones estadísticas.
Una ventana tradicional de Mathematica en Windows y en Linux se muestra en la Figura 4.1.

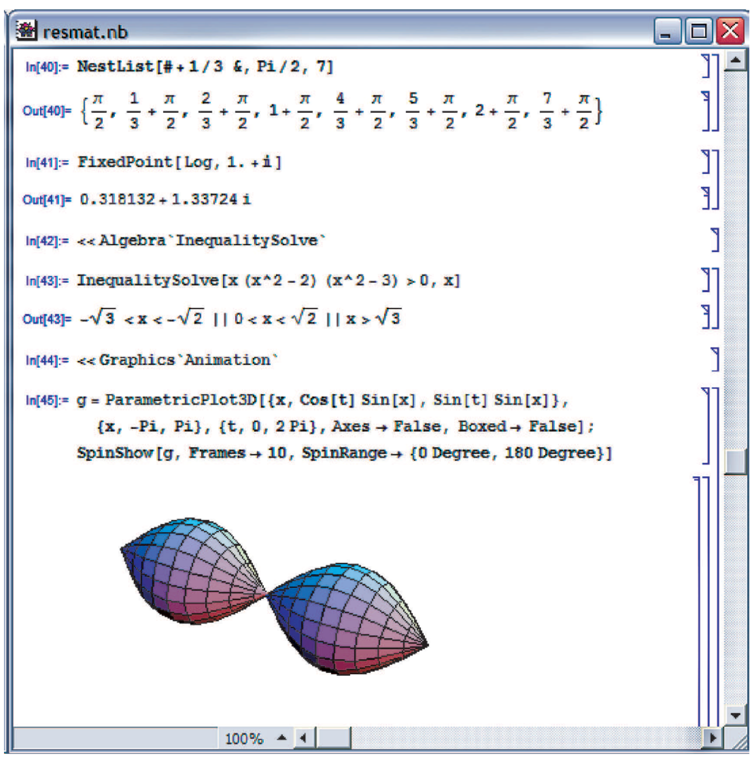


Figura 4.1: Ventana clásica de *Notebook*

4.4.1. Comandos de Mathematica

```
5 3^ 2 + (6 - 8)*3

39
```

```
N[Sqrt[2], 20]

1.4142135623730950488
```

```
Prime[Range[15]]

{2,3,5,7,11,13,17,19,23,29,31,37,41,43,47}
```

```
Cos[x + y + Pi] - Sin[x - y - Pi]//
{Cos[a_+b_]->Cos[a]*Cos[b]-Sin[a]*Sin[b],Sin[a_ + b_]
->Sin[a]*Cos[b]+Cos[a]*Sin[b]}

-Cos[x]Cos[y]+ Cos[y]Sin[x]- Cos[x]Sin[y] + Sin[x]Sin[y]
```

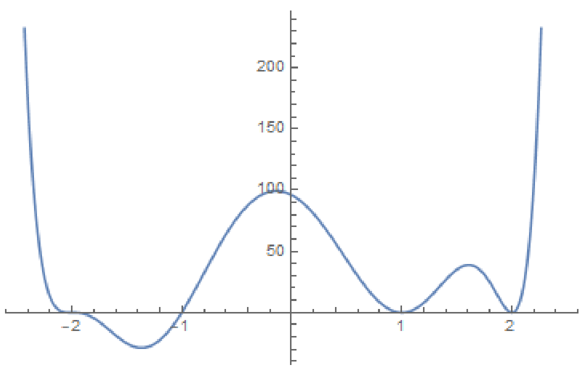
```
TrigReduce[%]

-Cos[x + y] + Sin[x - y]
```

```
q[x_] = (x - 1)^2(x + 2)^3(x + 1)(x - 2)^ 2(x^2 + 3);

-Cos[x + y] + Sin[x - y]
```

```
Plot[q[x], x, -2.5, 2.5]
```




```
-Graphics-
```

```
Expand[q[x]]
```

$$96 - 48x - 160x^2 + 56x^3 + 62x^4 - 3x^5 + 9x^6 - 6x^7 - 8x^8 + x^9 + x^{10}$$

```
M := {{1, 2, 0},{2, 0, -1}};
V := {a, b, c};
MatrixForm[M]MatrixForm[V] == MatrixForm[M.V]
```

$$\begin{pmatrix} 1 & 2 & 0 \\ 2 & 0 & -1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} == \begin{pmatrix} a+2b \\ 2a-c \end{pmatrix}$$

```
a={{0,-1}, {1,0}}
MatrixForm[MatrixExp[a*t] ]
```

$$\begin{pmatrix} \mathrm{Cos}[t] & -\mathrm{Sin}[t] \\ \mathrm{Sin}[t] & \mathrm{Cos}[t] \end{pmatrix}$$

```
matriz ={{p,q,0},{1,0,-1},{0,q,p}}
Eigenvalues[matriz][%]

{0, p, p}
```

```
Eigenvectors[matriz]
```

$$\{\{1, -\frac{p}{q}, 1\}, \{1, 0, 1\}, \{0,0, 0\}\}$$

```
A = {{1, 2, -1},{2, 3, 1},{0, -2, 1}}; b = {1, 1, -2};
{x, y, z} = LinearSolve[A, b]
```

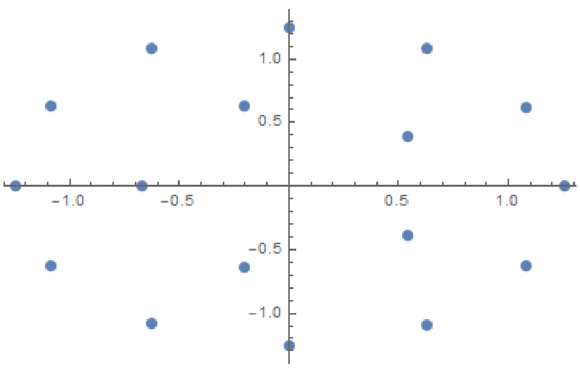
```
{-1, 1, 0}
```

```
Clear[a, b, x, y, z]; Solve[a x^2 + b x + c == 0, x]
```

$$\{\{x\to \frac{-b-\sqrt{b^2-4ac}}{2a},\},\{x \to \frac{-b+\sqrt{b^2-4ac}}{2a},\}\}$$

```
DibujaRaices[polinomio_, var_] :=
Module[{puntos = {}, lista = {}, raiz =0},(*Prog. modular*)
lista = var /. NSolve[polinomio == 0, var]; (*Almacena raices*)
While[lista ≠ {}, (*MIENTRAS lista no sea vacia*)
      raiz = N[First[lista]]; (*Extrae la primera raíz*)
      lista = Delete[lista, 1]; (*Borra el primer elemento*)
      AppendTo[puntos, {Re[raiz], Im[raiz]}] (*Agrega coordenadas*)
]; (*Fin del bucle MIENTRAS*)
ListPlot[puntos, PlotStyle → PointSize[0.02]] (*Dibuja los puntos*)
]
```

```
DibujaRaices[x^17 - 15 x^5 - 2, x]
```




```
-Graphics-
```

```
Limit[Tan[x]/x, x → 0]
```

```
1
```

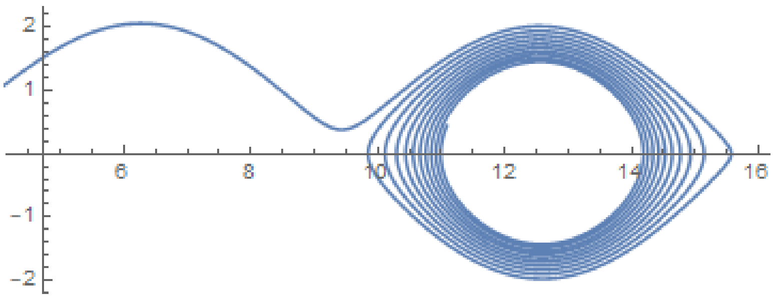
```
D[Sin[x], x], D[x^3 y^5, {y, 2}], D[x^6 y^(-2), x, y]}
```

$$\{Cos[x],\ 20x^3y^3\ ,-\frac{12x^5}{y^3}\}$$

```
Integrate[E^(-x^2), {x, 0, Infinity}]
```

$$\frac{\sqrt{\pi}}{2}$$

```
solucion = NDSolve[{x'[t] == y[t], y'[t] == -.01 y[t] - Sin[x[t]], x[0] == 0, y[0] ==
2.1},
{x, y},{t, 0, 100}]; ParametricPlot[Evaluate[{x[t], y[t]} /. solucion], {t, 0, 100}]
```

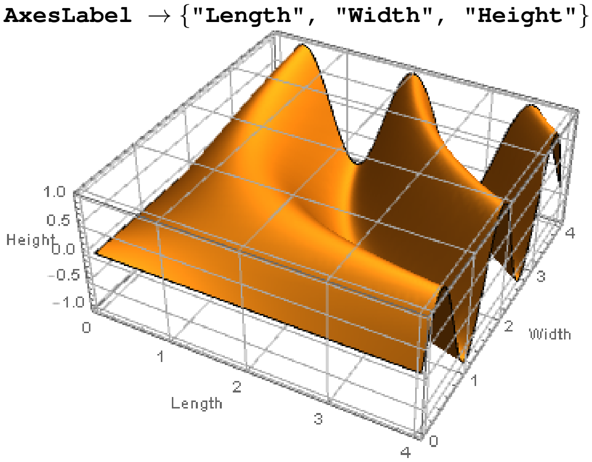



```
-Graphics-
```

```
ConstrainedMin[x + 3 y + 7 z, {x - 3 y < 7, 2 x + 3 z ≥ 5, x + y + z < 10},{x, y, z}]
```

$\{\frac{5}{2},\{x \to \frac{5}{2},\ y \to 0,\ z \to 0\}\}$

```
Plot3D[Sin[x y], {x, 0, 4},{y, 0, 4}, PlotPoints → 40, Mesh → False, FaceGrids →
All,
AxesLabel →{"Length", "Width", "Height"}]
```




```
-SurfaceGraphics-
```

4.5 Conclusiones

COMO se puede observar en las demostraciones de una variedad de comandos de Mathematica que han permitido ilustrar básicamente todo el recorrido del Cálculo Diferencial e Integral, Algebra Lineal, Optimización, Desigualdades, Resolución de Ecuaciones Algebraicas y Diferenciales, Ecuaciones en Diferencias Finitas, Transformadas, Gráficos, Animación de gráficos, Programación, etc. Por lo tanto, el Programa Mathematica al soportar principalmente el cálculo simbólico puede constituir una herramienta aliada del docente de las materias de Matemáticas tanto en las Universidades como en Educación Secundaria, puesto que visualmente se pueden mostrar los conceptos esenciales como los el *límite*, la *derivada*, la *integral*, la *convergencia* mucho mas con el recurso de la animación y vista de gráficos por un punto específico. De modo que la MATHEMATICA es una herramienta muy útil para la enseñanza de las matemáticas en todos los niveles de modo que el aprendizaje sea significativo.

Referencias Bibliográficas

[1] Enrique Castillo, et. al. (1996), *MATHEMATICA*, Editorial Paraninfo, Tercera Edición.

[2] Manual de Mathematica, Versión 2 y 4.