ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

VIGILADA MINEDUCACIÓN

Departamento de Matemáticas

Inducción 2020 - 1

Elementos básicos de `Mathematica`

En estos breves apuntes presentamos algunas de las funciones y comandos básicos que se utilizan en el programa `Wolfram Mathematica`. También se presentan algunos ejemplos los cuales pueden ser *ejecutados* por el usuario para comprobar la instrucción.

Contenido

1. Solicitud del programa `Mathematica`

Se debe enviar un correo a `mesadeayuda@escuelaing.edu.co`, con los siguientes datos:

- Nombre completo.
- Correo institucional.
- Materia que lo solicita.
- Carrera que cursa.

A vuelta de correo recibirá dos correos uno de Wolfram con el Key y los pasos a descargar, así:

Confirmación clave instalación programa Wolfram

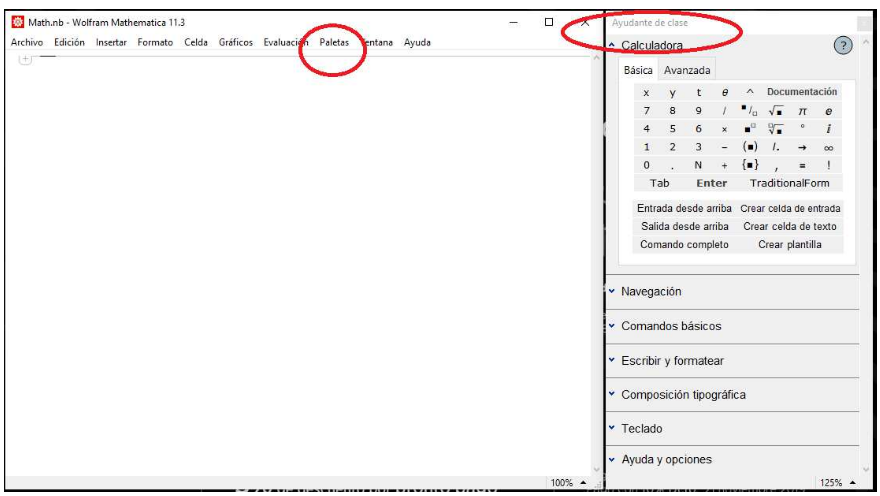
Wolfram Customer Support `info@wolfram.com`

Dear,
You have been assigned the following activation key for Mathematica for Students for Sites:

Activation Key: `3478***********`
Product: Mathematica for Students for Sites `12.0.0.0`
Platform: All
To download your product from the Wolfram User Portal, go to:
`https://user.wolfram.com/portal/requestAK/88bc83fea41d710addd24f217ebe6d968c900cb1?`
`activationKey=34780171A3X3GL&u=luis.paipa&40escuelaing.edu.co\\`
`&v=6a8b29ea54b1c1a3c014bfa76adfe800ee0c4e18`

Y el segundo correo lo recibirá de la oficina OSIRIS donde se direcciona a los instaladores para una mejor descarga.

Una vez iniciado el programa, el usuario puede obtener una galería con símbolos y funciones propias del *quehacer* Matemático. El Ayudante de clase, de escritura o de Matemáticas básicas, se visualiza activandolo en la opción *paletas*.



2. Elementos básicos en `Mathematica`

Las funciones en `Mathematica` inician con letra mayúscula, en algunos casos llevan letras intermedias también en mayúscula. Los argumentos de las funciones van entre corchetes (`[...]`), los parentesis (`(...)`) se usan para agrupar y las llaves (`{...}`) para declarar listas.

Toda instrucción se debe *ejecutar*, esto se consigue con **Shift**+**Enter** o **Enter** con el teclado numérico.

2.1. `Wolfram Mathematica` básico

Algunos de los operadores y funciones matemáticas más usadas son[1]

[1]Tomado de *http://www3.uji.es/ planelle/APUNTS/IAQ/manual_math.pdf*

Comandos elementales

`x^y` Potencia	`x y z` o `x*y*z` Producto
`-x` Inverso aditivo	`x+y` Suma
`x/y` División	

Algunas funciones básicas en Wolfram

`Sqrt[x]`	Raíz cuadrada de x
`Exp[x]`	Función exponencial (e^x)
`Log[x]`	Logaritmo natural ($log_e x$)
`Log[b,x]`	Logaritmo en base b ($log_b x$)
`Sin[x], Cos[x], Tan[x], Cot[x]`	Funciones trigonométricas (con argumentos en radianes)
`Sec[x], Csc[x]`	
`ArcSin[x], ArcCos[x], ArcTan[x]`	Funciones trigonométricas inversas
`ArcCot[x], ArcSec[x], ArcCsc[x]`	
`n!`	Factorial de n (producto de los enteros $1, 2, \cdots, n$)
`Abs[x]`	Valor absoluto
`Round[x]`	Redondeo entero de x
`Mod[n,m]`	n módulo m (residuo de dividir n entre m)
`Random[ ]`	Número seudo aleatorio entre 0 y 1
`Max[x, y, ...],Min[x, y, ...]`	Máximo, mínimo de $x, y, ...$
`FactorInteger[n]`	Factores primos de n

A continuación presentamos algunos de las simbolos de constantes (con la sintaxis requerida) que se utilizan en el programa `Wolfram Mathematica`.

Algunas constantes básicas en `Wolfram Mathematica`

`Pi`	$\pi \approx 3{,}14159$
`E`	$e \approx 2{,}71828$ (normalmente aparece como `e`)
`Degree`	$\pi/180$ factor de conversión de grados a radianes (normalmente aparece como $^\circ$)
`I`	$i^2 = -1$ (normalmente aparece como `I`, unidad imaginaria)
`Infinity`	∞

2.2. Aritmética básica

Las operaciones básicas utilizan los caracteres comunes. Por ejemplo,

```
2+3        N[Pi,50]     Pi//N       N[20/3,50]    2^100       9!
```

Se puede definir varables y luego usarlas. El ";" evita que al ejecutar una línea se muestre el producto.

```
a=2;            b=5;            c=a+b;          2*a+3*b
a+b             a-b             a^b             (a+c)/(b-a)
```

2.3. Algunas funciones elementales

Para factorizar, simplificar, expandir, descomponer en fracciones parciales, operar fracciones o borrar una `EXPRESION`, se usan respectivamente, las siguientes funciones:

Algunas funciones básicas en `Mathematica`

```
Factor[EXPRESION]      Expand[EXPRESION]     Together[FRACCIONES]
Simplify[EXPRESION]    Apart[FRACCION]       Clear[VARIABLES]
```

Por ejemplo, el usario puede ejecutar las siguientes instrucciones

```
Factor[a^16-b^16]      Together[x/y+y/x]     Apart[4*y^3/(y^4-16)]
Expand[(x+2)^9]        Clear[a,b]            Simplify[(x^9-y^9)/(x^3-y^3)]
```

Si una `EXPRESION` es una combinación de funciones trigonométricas, la instrucción `TrigExpand[EXPRESION]` la reescribe en términos de las funciones $\sin(\cdot)$ y $\cos(\cdot)$. La sentencia `TrigFactor[EXPRESION]` factoriza usando funciones trigonométricas y `TrigReduce[EXPRESION]` convierte producto de funciones trigonométricas en suma. `TrigFactorList[EXPRESION]` produce la lista de factores en que se descompone una expresión trigonométricas. Por ejemplo,

```
TrigExpand[Sin[5*x]-Cos[6*x]]        TrigFactor[Sin[5*x]-Cos[5*x]]
TrigReduce[Sin[2*x]*Cos[3*x]]        TrigFactor[Sin[x-y]-Cos[x+y]]
```

Vectores

El vector $\mathbf{u} = \langle a, b, c\rangle$ se define en `Mathematica` como una lista `u={a,b,c}`.

El producto punto, el producto cruz y la norma de los vectores $\mathbf{u}$ y $\mathbf{v}$ se determina como sigue

```
Dot[u,v]        Cross[u,v]        Norm[u]
```

Se muestra la sintaxis necesaria para hallar el producto punto, el producto cruz y la norma de los siguientes vectores

- `u={3,5,-8}`
  `v={-1,2,2}`
- `u={t,Exp[-t],Sin[t]}`
  `v={-t,Exp[t],2t}`

El símbolo % recupera el cálculo inmediatamente anterior, por ejemplo, con alguno de los casos anteriores se puede ejecutar éstas instrucciones:

```
Norm[u+v]        Simplify[%]        Simplify[%,Assumptions->t>0].
```

Funciones vectoriales

Una función vectorial $\mathbf{r}(t) = \langle x(t), y(t), z(t)\rangle$ se define como un vector `r={x[t],y[t],z[t]}`.

Para extraer cada una de las tres componentes, se debe usar, según el caso, las siguientes instrucciones

```
r[[1]]        r[[2]]        r[[3]]
```

La instrucción `p={Cos[t],Sin[t],Cos[2*t]}` define una función vectorial, declarada como p.

Resolución de ecuaciones

Para resolver la ecuación $f(x) = 0$, en donde x es la incógnita, se usa la instrucción:
`Solve[f(x)==0,x]`, **no olvidar** "==". También se puede usar `NSolve` para resolución numérica.

- `Solve[x^4==1,x]`
- `Solve[x^4y^2==1,y,Reals]`
- `NSolve[x^4+4==0,x]`

3. Matrices y sistemas de ecuaciones lineales

Para declarar una matriz de debe hacer una lista con sus filas, que a su vez son una lista de las entradas de esa fila. Por ejemplo

```
A={{3,2,1},{-7,-6,-3},{7,7,5}};
```

El determinante, la traspuesta, la inversa y el rango de A se hallan como sigue

```
Det[A]          Transpose[A]          Inverse[A]          MatrixRank[A]
```

El sistema de Ecuaciones

$$\left.\begin{aligned} 2x + 3y &= 8 \\ 3x + 2y &= 7 \end{aligned}\right\}$$

Se resuelve mediante la siguiente instrucción

```
Solve[{2x+3y==8,3x+2y==7},{x,y}]
```

Los valores y vectores propios de A se hallan usando las siguientes instrucciones

```
Eigenvalues[A]                          Eigenvectors[A]
```

Las siguientes dos ecuaciones corresponden a dos planos

$$\left.\begin{aligned} 2x - 3y + 4z &= -1 \\ 3x + 2y - 4z &= 0 \end{aligned}\right\}$$

cuya intersección es la recta con ecuaciones paramétricas $x = t$, $y = 5t + 1$, $z = 13t/4 + 1/2$.
Las siguientes tres instrucciones, permiten dibujar los dos planos y la recta de intersección en un solo gráfico, en ellas se usan las funciones `ContourPlot3D`, `ParametricPlot3D` y `Show`, cuya descripción se presenta más adelante.

```
a=ContourPlot3D[{2x-3y+4z==-1,3x+2y-4z==0},{x,-3,3},{y,-3,3},{z,-4,4}];

b=ParametricPlot3D[{t,5*t+1,13*t/4+1/2},{t,-2,2},
PlotStyle->{Red,Thickness->0.015}];

Show[a,b]
```

La instrucción `RowReduce[A]` lleva la matriz A a la forma escalonada reducida.

Además, al definir

```
M={{1,1,1},{1,1,2},{1,2,3}};
b={1,-1,1};
```

La instrucción

```
MatrixForm[a=Transpose[Join[Transpose[M],{b}]]]
```

junta las matrices M y b, en una matriz rectangular, la lleva a la forma escalonada reducida y la muestra en forma matricial; además se declara como `a`, para cálculos posteriores.

4. Límites, Derivas e Integrales

La sentencia `Limit[f[x],x->a]` determina $\lim_{x \to a} f(x)$.

Límites

```
Limit[Sin[2*t]/t,t->0]                 Limit[(1+3/t)^t,t->Infinity]
```

Derivadas e integrales en `Mathematica`

- Para derivar la función $f(x)$ con respecto a x, se ejecuta la sentencia: `D[f(x),x]`
- Para evaluar la integral indefinida de la función $f(t)$ con respecto a t, se ejecuta la instrucción: `Integrate[f(t),t]`
- Si la integral es definida en la forma $\int_a^b f(t)dt$, la instrucción a usar es la siguiente: `Integrate[f(t),{t,a,b}]`

Ejemplo de lo anterior son las siguientes instrucciones

- `D[ArcSin[x], x]`
- `D[Sqrt[1/(1+r)],r]`
- `D[Sqrt[x^3+x-2],x]`
- `Integrate[1/(1+r),r]`
- `Integrate[x*Exp[-x],{x,0,1}]`
- `Integrate[t*Exp[t^2],{t,0,1}]`

Derivas parciales

La instrucción `D[f[x,y],x,x]` permite hallar $\frac{\partial^2 f}{\partial x^2}$.
De manera análoga para obtener $\frac{\partial^2 f}{\partial x \partial y}$ se utiliza `D[f[x,y],y,x]`.

Por ejemplo

```
D[x*Exp[x*y],x,x]                      D[x^y,x,x]
D[x*Exp[x*y],y,x]                      D[x^y,y,x]
```

5. Sumas

La expresión `Sum[a(k),{k,a,b}]`, permite obtener $\sum_{k=a}^{b} a(k)$. Enseguida se presentan algunos ejemplos

Ejemplos de sumas y series

```
Sum[k^2,{k,1,n}]                Sum[1/(2k+1)^2,{k,1,Infinity}]
Sum[1/k^2,{k,1,Infinity}]       Sum[(-1)^k/k^2,{k,1,Infinity}]
```

6. Animación de Objetos

Para *animar* un `OBJETO`, con variable k, para $k = a, \cdots, b$ e incrementos de r unidades, se utiliza

Las Funciones `Manipulate` y `Animate`

```
Manipulate[OBJETO,{k,a,b,r}]        Animate[OBJETO,{k,a,b}]
```

La r se puede omitir en el caso de la función `Animate`. El lector puede ejecutar las siguientes instrucciones para evidenciar un poco mejor, el uso de las funciones mencionadas

- `Manipulate[Expand[(a+b)^k],{k,1,20,1}]`
- `Animate[Expand[(a+b)^k],{k,1,20,1}]`
- `Manipulate[Integrate[x^k,x],{k,-5,5,1/2}]`
- `Manipulate[Integrate[(Sin[x])^k,x],{k,-2,10,1}]`

7. Gráficas

7.1. Gráficas en el plano

Presentamos algunas de las posibilidades que ofrece `Mathematica` para dibujar funciones en el plano.

Funciones para graficar en el plano

- `Plot[f(x),{x,a,b}]`
  Dibuja la función $y = f(x)$ con $x \in [a, b]$.
- `PolarPlot[r(t),{t,a,b}]`
  Grafica la función $r(t)$ escrita en coordenadas polares, con $t \in [a, b]$.
- `ContourPlot[f(x,y)==0,{x,a,b},{y,c,d}]`
  Dibuja la ecuación de dos variables $f(x, y) = 0$, con $x \in [a, b]$ y $y \in [c, d]$, se debe incluir el símbolo de igualdad y no olvidar el "==".
- `ParametricPlot[{x[t],y[t]},{t,a,b}]`
  Dibuja la curva parametrizada $\overrightarrow{r}(t) = \langle x(t), y(t) \rangle$, $t \in [a, b]$
- `DiscretePlot[f(x),{x,a,b,r},ExtentSize->Right]`
  Permite hacer el grafico discreto de $f(x)$ en $[a, b]$, con subintervalos de longitud r. La opción `ExtentSize -> Left/Right/None/Full/All/Scaled[Δ]`
  determina la disposición (o altura) de cada rectangulo.

Si se desean incluir varias expresiones en las funciones anteriores, se puede hacer una lista con estas y encerrarlas en llaves (`{...}`).
Las siguientes intrucciones son ejemplos de las anteriores funciones

- `Plot[Exp[-10*t]*(2+20t)+1,{t,0,1}]`
- `PolarPlot[1-Cos[t],{t,0,2*Pi}]`
- `PolarPlot[3/(6+Cos[10*u]),{u,0,2*Pi}]`
- `ContourPlot[{y^2+(1-x^2)^2==4,y^2+x^2==4}, {x,-2,2},{y,-2,2}]`
- `ParametricPlot[{Cos[t],Sin[2*t]},{t,0,2Pi}]`.
- `Manipulate[PolarPlot[t*Sin[k*t], {t,0,2Pi}],{k,1,5,0.5}]`
- `ParametricPlot[{{2*Cos[t],2*Sin[t]},{Cos[t],Sin[t]},`
  `{Cos[t]*(1-Cos[t]),Sin[t]*(1-Cos[t])}},{t,0,2*Pi}]`

Es posible definir una función y luego usarla para cálculos posteriores, la sintaxis básica es:

Definición de funciones

```
f[x_,y_] := Expresion
```

Con las siguientes sentencias, se define la función vectorial $\mathbf{r}(t,u)$ y luego se evalua en $t = \pi/3$ y $u = 1/2$

```
r[t_,u_]:=(1-u)*{Cos[2*t],Sin[2*t],t}+u*{Sin[3*t],Cos[3*t],0}
r[Pi/3,1/2]
```

Se pueden definir funciones a trozos, con k condiciones y $x \in [a,b]$,

Definición de funciones a trozos

```
Piecewise[{{Expresion 1,condición 1},  ...,{Expresion k,condición k}},{x,a,b}]
```

Ahora se muestran algunos ejemplos

- `f[x_,y_]:=x^2-3y^2*x^2-y^4`
  Luego de ejecutarla se puede evaluar en la forma $f[2,3]$ cuyo resultado es la evaluación respectiva.
- La función
  $$f(x,y) = \begin{cases} \frac{xy}{x^2-y^2} & \text{si } x > y \\ \sin(x+y) & \text{En caso contrario} \end{cases}$$
  se define como: `f[x_,y_]:=Piecewise[{{x*y/(x^2-y^2),x>y}},Sin[x+y]]`
- La función
  $$f(x,y) = \begin{cases} 4-x^2 & \text{si } x \leq 1 \\ (2x-3)^2 & \text{si } x \in (1,2) \\ 2\sin(x) & \text{si } x \geq 2 \end{cases}$$
  se puede definir como

  ```
  f[x_]:=Piecewise[{{4-x^2,x<=1},{(2*x-3)^2,x<2},{2*Sin[x],x>=2}}]
  ```

7.2. Gráficos en el espacio

De manera análoga a las funciones descritas anteriormente para conseguir gráficos en el plano, se puede agregar la extensión `3D` y unas opciones adicionales con el fin de obtener gráficos en el espacio. Presentamos unas de las funciones más utilizadas en `Mathematica`.

Funciones para graficar en el espacio

- `Plot3D[f(x,y),{x,a,b},{y,c,d}]`

  Dibuja la función $z = f(x,y)$, para $(x,y) \in [a,b] \times [c,d]$.
  Si son varias funciones, se puede hacer una lista de ellas, es decir
  `Plot3D[{f(x,y),g(x,y),...,h(x,y)},{x,a,b},{y,c,d}]`

- `ParametricPlot3D[{x(t),y(t),z(t)},{t,a,b}]`
  Dibuja la curva parametrizada $\overrightarrow{r}(t) = \langle x(t), y(t), z(t)\rangle$, $t \in [a,b]$.

- `ParametricPlot3D[{x(u,v),y(u,v),z(u,v)},{u,a,b},{v,c,d}]`

  se utiliza para graficar superficies paramatrizadas en la forma
  $\overrightarrow{r}(u,v) = \langle x(u,v), y(u,v), z(u,v)\rangle$, $u \in [a,b]$, $v \in [c,d]$.

- Para graficar conjuntos de nivel (contornos) puede utilizarse
  `ContourPlot3D[f(x,y,z)==0,{x,a,b},{y,c,d},{z,p,q}]`
  Este comando presenta el gráfico de una ecuación de tres variables, con $x \in [a,b]$, $y \in [c,d]$ y $z \in [p,q]$.

Para dibujar en el punto (a,b,c) el vector $\langle p,q,r\rangle$ se utiliza la siguiente instrucción

Vectores en el espacio

```
Graphics3D[Arrow[{{a,b,c},{p,q,r}}]]
```

Si el grosor es u, la instrucción es `Graphics3D[Arrow[{{a,b,c},{p,q,r}},u]]`. Los gráficos se pueden personalizar en sus atributos con el uso de algunas opciones, presentamos algunas de las más utilizadas:

Opciones de los gráficos

`PlotStyle`	Opciones del gráfico, color, líneas, etc.
`PlotRange`	Rango de valores en el gráfico
`PlotPoints`	Número de puntos sobre el dibujo
`Thickness`	Grosor de la línea
`Mesh`	Activa la Malla, `True/False`
`Tube[a]`	Configura un tubo de radio a.
`Thick`	Crea una línea más gruesa
`MeshStyle`	Estilo de la malla (color)
`Opacity[a]`	Da opacidad, se usa en plotStyle, con $0 \leq a \leq 1$
`ContourStyle`	Opciones en `ContourPlot3D`
`Boxed`	Muestra gráfico en caja, `True/False`
`Mesh->{n,m}`	Crea la malla, con n y m líneas
`AxesOrigin->{a,b,c}`	Muestra los ejes con centro en (a,b,c).
`AxesLabel->{u,v,w}`	Etiqueta los ejes como u, v y w.
`Ticks`	Indica las marcas deseadas en la caja.
`ColorFunction`	Determina el color de la función.
`Blend[{color1,...}]`	Permite combinar varios colores.
`BoxRatios->{a,b,c}`	Establece proporciones de la caja en un gráfico.

Ilustraremos lo anterior con algunos ejemplos.

- `Plot3D[{5-y^2,x^2+y^2},{x,-4,4},{y,-4,4}, PlotPoints->60,PlotStyle->Opacity[0.6]]`
- `Graphics3D[{Red,Arrowheads[0.1], Arrow[Tube[{{1,1,-1},{2,2,0}},0.01]]}]`
- `DiscretePlot3D[2-x^2-y^2,{x,-1,1,0.1},{y,0,1,0.2}, ExtentSize->Full]`
- `ContourPlot3D[{z==6-(x-1)^2-y^2,4*x^2+y^2==36, x^2+4*y^2==4},{x,-6,10},{y,-7,7},{z,-35,6}, PlotPoints->60,ContourStyle->{Red,Yellow,Green}]`
- `ParametricPlot3D[{Cos[u]*Cosh[v],Sin[u]*Cosh[v], Sinh[v]},{u,0,2*Pi},{v,-2,2},PlotStyle->Orange,MeshStyle->Gray]`
- `ContourPlot3D[z==3*x-x^3-2*y^2+y^4,{x,-3,3}, {y,-2,3},{z,-3,3},MeshStyle->Gray,PlotPoints->60]`
- `ContourPlot3D[z==3*x-x^3-2*y^2+y^4,{x,-3,3},{y,-2,3}, {z,-3,3},MeshStyle->Gray,PlotPoints->60, ColorFunction->Function[{x,y,z},ColorData["Rainbow"][z]]]`

La composición de varias funciones puede hacer complicado el tratamiento, la función `Show` permite ver en un solo gráfico varios objetos obtenidos con diferentes funciones, por ejemplo:

Uso de la función `Show`

```
Show[ParametricPlot3D[{Cos[t],Sin[t],1},{t,0,2\[Pi]},PlotStyle->Red],
Graphics3D[{Blue,Arrowheads[0.05],Arrow[Tube[{{1,0,1},{1,1,1}},0.02]]}],
AxesOrigin->{0,0,0},PlotRange->{{-2,2},{-2,2},{-1,2}}]
```

Otra posibilidad, es ejecutar las instrucciones de manera separada asignándoles un nombre a cada una; posteriormente se muestran en un solo gráfico utilizando la función `Show`. En el siguiente ejemplo, las primeras dos instrucciones al ejecutarlas, presentan de manera independiente las gráficas y le asignan como nombres `A` y `b`, y luego con la función `Show` se muestran ambas en una sola.

Ejecución de gráficos por separado

```
A=ContourPlot3D[{z==4-y-4x+2x^2,(1-x)^2+(2-y)^2==1},
{x,-3,3},{y,-1,4},{z,-2,3}]

b=ParametricPlot3D[{1-Cos[t],2-Sin[t],Sin[t]+2(Cos[t])^2},
{t,0,2Pi},PlotStyle->{Red,Thickness->0.02}]

Show[A,b]
```

7.3. Regiones en el espacio

La siguiente función será una de las más utilizadas para representar un sólido.

Regiones en el Espacio

```
RegionPlot3D[DESIGUALADES,{x,a,b},{y,c,d},{z,u,v}]
```

Este comando dibuja el sólido determinado por unas `DESIGUALADES` que se concatenan con los símbolos `&&`, para $x \in [a,b]$, $y \in [c,d]$, $z \in [u,v]$.

Presentamos enseguida algunas situaciones que ilustran el uso de esta función.

1. La región del espacio acotada por las gráficas de las expresiones $z = 0, z = 1 - y,\ \ x^2 = y,\ \ y = 1$; se obtiene con la siguiente instrucción:

```
RegionPlot3D[0<z<1-y&&x^2<y<1,{x,-1,1},{y,0,1},{z,0,1},
PlotPoints->90,PlotStyle->Orange,Mesh->False,Ticks->{{-1,0,1},{-1,1},{0,1}}]
```

2. Las ecuaciones $(x-1)^2 + y^2 = 1,\ \ (x - 3/2)^2 + y^2 = 4,\ \ z = 0,\ \ 2z + y = 4$. Determinan un sólido. Para obtenerlo se pueden ejecutar las siguientes instrucciones:

```
RegionPlot3D[z>0&&z<4-y/2&&(x-1)^2+y^2>1&&(x-1.5)^2+y^2<4,
{x,-1,4},{y,-2,2},{z,0,5},Mesh->False,PlotPoints->90]
```

3. Las expresiones $(x^2+y^2)^2 = z+1,\ \ z = 4(1-x^2-y^2)$, definen un sólido, las siguientes instrucciones permiten obtener su gráfico.

```
RegionPlot3D[(x^2+y^2)^2-1<z<4*(1-x^2-y^2)^2&&-Sqrt[1-x^2]<y<Sqrt[1-x^2],
{x,-1,1},{y,-1,1},{z,-1,4},PlotPoints->90,Mesh->False]
```

4. Las expresiones $x^2+y^2+z^2 = 4$, $x^2+y^2 = 2+z^2$, $x^2+y^2 = \frac{2}{5}$, representan respectivamente una esfera, un hiperboloide de dos mantos y un cilindro. La región del espacio interior a la esfera y al hiperboloide, y fuera del cilindro se obtiene con las siguientes instrucciones

```
RegionPlot3D[x^2+y^2+z^2<4&&x^2+y^2<2+z^2&&x^2+y^2>0.4,
{x,-2,2},{y,-2,2},{z,-2,2},PlotPoints->90]
```

5. En coordenadas cilíndricas, las siguientes desigualdades $1 \le r \le 2,\ \ r \le z \le 2$, determinan la región acotada por un cilindro, un semicono y el plano $z = 2$.

   Para obtener este gráfico se definen r y θ, luego se escriben las desigualdades y se agregan otras opciones

```
r=Sqrt[x^2+y^2];
\[Theta]=ArcTan[y/Abs[x]];
RegionPlot3D[1<=r<=2&&r<=z<=2,{x,-2,2},{y,-2,2},{z,1,2},PlotPoints->90,
Mesh->True,AxesLabel->Automatic,PlotRange->{{-2,2},{-2,2},{1,2.2}}]
```

7.4. Graficos en coordenadas esféricas

En coordenadas esféricas (ρ,θ,ϕ), la región del espacio determinada por las siguientes expresiones

$$1 \le \rho \le 2,\ \ 0 \le \theta \le \frac{3\pi}{4},\ \ \frac{\pi}{6} \le \phi \le \frac{3\pi}{4},$$

se puede visualizar usando `Mathematica`. Para conseguirlo, es posible realizarlo de dos formas, definiendo las variables $\rho,\ \theta,\ \phi$ y luego establecer la región que se ilustra a continuación

Regiones en el Espacio

```
\[Rho]=Sqrt[x^2+y^2+z^2];
\[Theta]=ArcTan[y/Abs[x]];
\[Phi]=ArcCos[z/\[Rho]];

RegionPlot3D[1<\[Rho]<2&&0<\[Theta]<3Pi/4&&
Pi/6<\[Phi]<3*Pi/4,{x,-2,2},{y,0,2},{z,-2,2},
PlotPoints->90,Mesh->False]
```

o también se puede utilizar la función `SphericalPlot3D`, para graficar una función de la forma $\rho = f(\theta, \phi)$, con $\theta \in [a, b]$ y $\phi \in [c, d]$. La sintaxis genérica es

Graficos en Coordenadas Esféricas

```
SphericalPlot3D[f(\theta, \phi),{\[Theta],a,b]},{\[Phi],c,d]}].
```

- `SphericalPlot3D[1,{\[Theta],Pi/6,Pi}, {\[Phi],Pi/6,5*Pi/3}]`
- `SphericalPlot3D[Cos[\[Phi]],{\[Theta],0,Pi}, {\[Phi],0,Pi}]`
- `SphericalPlot3D[\[Theta],{\[Theta],0,Pi}, {\[Phi],0,2*Pi}]`

7.5. Superficie de revolución

El gráfico de la región que se obtiene de girar la función $y = f(x)$ alrededor del eje y, con $x \in [a, b]$, se obtiene con ayuda de la siguiente instrucción

Sólidos de revolución

```
RevolutionPlot3D[f(x),{x,a,b}]
```

Si la curva está parametrizada en la forma $\langle x(t), y(t), z(t)\rangle$ y esta gira alrededor del vector $\langle p, q, r\rangle$, con $t \in [a, b]$, la sintaxis que se usa es la siguiente

Sólido de revolución con una curva parametrizada

```
RevolutionPlot3D[{x(t),y(t),z(t)},{t,a,b},RevolutionAxis->{p,q,r}]
```

Por ejemplo, la función $\mathbf{r}(t) = \langle 1 - \cos(t), \sin(t), t - t^2\rangle$, con $t \in [0, 1]$, gira aldededor del vector $\langle 1, 1, 0\rangle$, para ver el región que se genera, se puede ejecutar la instrucción

```
RevolutionPlot3D[{1-Cos[t],Sin[t],t-t^2},{t,0,1},
RevolutionAxis->{1,1,0},Ticks->{{0,0.5},{0,0.5},{-0.2,0.2}}]
```

8. Integración Múltiple

Para evaluar integrales dobles o triples en `Wolfram Mathematica`, se puede proceder como sigue.

Para una integral doble de la forma $\int_a^b \int_c^d f(x, y)dydx$, la instrucción es

Integrales dobles

```
Integrate[f(x,y),{x,a,b},{y,c,d}]
```

Si se desea ver la integral sin evaluar se puede agregar la sentencia `//HoldForm`, y si se desea evaluar numéricamente se debe agregar `//N`. Por ejemplo

1. La integral $\int_0^2 \int_{-x}^{x^2} xy^2 dxdy$, se evalua con la siguiente orden

```
Integrate[x*y^2,{x,0,2},{y,-x,x^2}]
```

2. Para evaluar numéricamente la integral doble $\int_0^2 \int_{\sqrt{y}}^{2-y} 30x^2 dxdy$, se puede usar la siguiente instrucción:

```
Integrate[30x^2,{y,0,1},{x,Sqrt[y],2-y}]//N
```

3. La integral impropia $\int_0^\infty \int_0^\infty e^{-x^2-y^2} dydx = \frac{\pi}{4}$, se evalua como sigue

```
\begin{tcolorbox}[breakable,enhanced]
Integrate[Exp[-x^2-y^2],{x,0,Infinity},{y,0,Infinity}]
```

La integral triple $\int_a^b \int_c^d \int_e^s f(x,y,z)\, dzdydx$, se puede evaluar mediante la instrucción

Integrales triples

```
Integrate[f(x,y,z),{z,r,s},{y,c,d},{x,a,b}]
```

De manera alternativa, se puede utilizar la siguiente notación:

`Integrate[Integrate[Integrate[f(x,y,z),{x,a,b}],{y,c,d}],{z,r,s}]`

Nótese en este caso, el orden en la escritura. Ahora ilustraremos el uso de esta función con unas integrales triples

1. La integral, $\int_{-3}^3 \int_0^2 \int_0^1 x^2y^3z^4 dzdydx$, se evalua como sigue

```
Integrate[Integrate[Integrate[x^2y^3z^4,{x,-3,3}],
{y,0,2}],{z,0,1}]
```

2. La evaluación de la integral $\int_0^1 \int_{-\sqrt[2]{x}}^{\sqrt[2]{x}} \int_0^x x^2 dzdydx$, se hace con la orden

```
Integrate[x^2,{x,0,1},{y,-Sqrt[x],Sqrt[x]},{z,0,x}]
```

3. Para ver la forma sin evaluar de la integral $\int_0^{\pi}\int_0^{\pi/6}\int_0^2 \rho^3\cos(\theta)\sin^2(\phi)\,d\rho d\theta d\phi$, se ejecuta la siguiente instrucción

```
Integrate[Integrate[Integrate[\[Rho]^3*Cos[\[Theta]]*
Sin[\[Phi]],{\[Rho],0,2}],{\[Theta],0,Pi/6}], {\[Phi],0,Pi}]//HoldForm
```

4. La integral doble impropia, $\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^2 e^{-x^2-y^2}dydx = \frac{\pi}{2}$, se evalua mediante la ejecución de la siguiente instrucción

```
Integrate[x^2*Exp[-x^2-y^2],{x,-Infinity,Infinity},{y,-Infinity,Infinity}]
```

5. La integral triple impropia

$$\frac{1}{\sqrt{\pi}}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-x^2-y^2-z^2}dzdydx = \pi$$

se evalúa usando `Mathematica` mediante la ejecución de la siguiente instrucción

```
1/Sqrt[Pi]*Integrate[Exp[-x^2-y^2-z^2],{x,-Infinity,Infinity},
{y,-Infinity,Infinity},{z,-Infinity,Infinity}]
```

9. Ecuaciones diferenciales

La instrucción

```
DSolve[y'[x]==f[x,y],y[x],x]
```

Permite resolver una ecuación diferencial de la forma $y' = f(x,y)$ siendo x la variable independiente y y la variable dependiente. si se agregan condiciones iniciales usar llaves ({...}). Presentamos ahora algunos ejemplos que ilustran estas posibilidades.

- `DSolve[A'[t]+4*A[t]/(100-t)==6/5,A[t],t]`
- `DSolve[{y''[x]-y'[x]==0,y[0]==0,y'[0]==1},y[x],x]`
- `DSolve[{y'[x]+y[x]==Sin[x],y[0]==0},y[x],x]`
- `DSolve[{x^3*y'''[x]+6*x^2*y''[x]+7*x*y'[x]+y[x]==x^3},y[x],x]`

  Se pueden resolver Sistemas de Ecuaciones diferenciales, en este caso se debe hacer una lista de ecuaciones diferenciales y de las funciones incognitas, además de usar llaves ({...}).

- `Simplify[DSolve[{x''[t]==-4*x[t]+Sin[t],y''[t]==4*x[t]-8*y[t]},`
  `{x[t],y[t]},t]]`
- `DSolve[{x''[t]==y[t],y''[t]==x[t],x[0]==0,y[0]==1,x'[0]==0,`
  `y'[0]==-1},{x[t],y[t]},t]`