Actividad 8. Bitácora de Maxima

Rosa Luz Zamora Peinado

Marzo de 2016

Introducción

En la presente actividad se presentan ejercicios realizados siguiendo un manual de Cálculo multivariable utilizando Maxima: Multivariable Calculus with Maxima de G. Jay Kerns [**5**].

Maxima es un sistema de álgebra computacional. Se basa en una versión 1982 de Macsyma, que fue desarrollado en el MIT con fondos del Departamento de Energía de los Estados Unidos y otras agencias gubernamentales. Una versión de Macsyma se mantuvo por Bill Schelter desde 1982 hasta su muerte en 2001. En 1998, Schelter obtuvo el permiso del Departamento de Energía para liberar su versión bajo la GPL. Esa versión, que ahora se llama Maxima, es mantenida por un grupo independiente de usuarios y desarrolladores.

Esta nueva herramienta nos permitirá solucionar problemas que de hacerse .ª mano-esultan complicaciones que podemos evitar utilizando funciones como las que se presentan más adelante. En específico, a continuación se resuelven problemas que se abordan en los cursos de Cálculo III y IV similares a los del libro de Cálculo Vectorial de Marsden y Tromba. Van desde gráficas en 2 y 3 dimensiones de funciones vectoriales hasta integrales dobles y triples, multiplicadores de lagrange, integrales de línea, cambio de variables y/o coordenadas y campos vectoriales. En un futuro, podremos evitarnos la parte operacional de los problemas utilizando esta herramienta.

2.1. Vectores y álgebra lineal

A continuación una muestra de cómo operar vectores:

```
/* [wxMaxima: input   start ] */
A: [5,2,-4];
B: [3,-7,1];
C: [1,0,-2];
C . (A ~ B);
express(C . (A ~ B));
/* [wxMaxima: input   end   ] */
```

```
(A)[5,2,-4]
(B)[3,-7,1]
(C)[1,0,-2]
(%o27) -[1,0,-2] . [3,-7,1]~[5,2,-4]
(%o28) 56
```

2.2. Líneas, planos y superficies cuadráticas

```
load(draw);
hyperboloid:  2*x^2 + 4*y^2 - z^2 = 1;
draw3d(palette = [4,6,9], enhanced3d = true, implicit(hyperboloid,
x,-2,2, y,-2,2, z,-1.5,1.5));
```

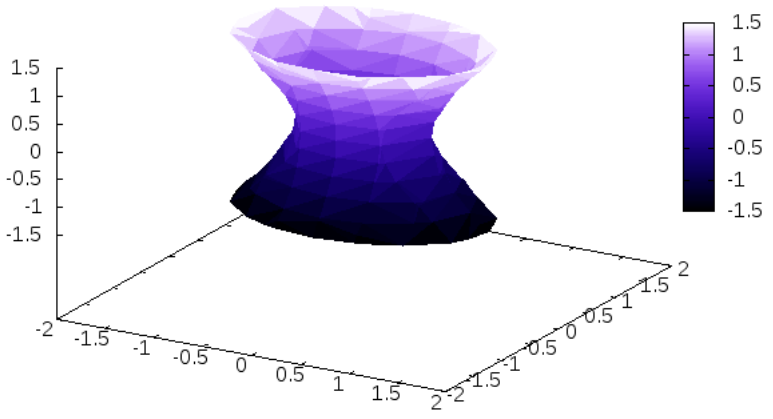


Figura 1: Hiperboloide

2.3. Funciones escalares evaluadas

- graficar una función vectorial

```
r(t) := [10*cos(t),2*sin(t),t];
r(4);
float(r(4));
load(draw);
draw3d(color=magenta,parametric(-cos(t), sin(t),cos(t), t,-4, 4));

(%o32) r(t):=[10*cos(t),2*sin(t),t]
(%o33) [10*cos(4),2*sin(4),4]
(%o34) [-6.53643620863612,-1.513604990615856,4.0]
```

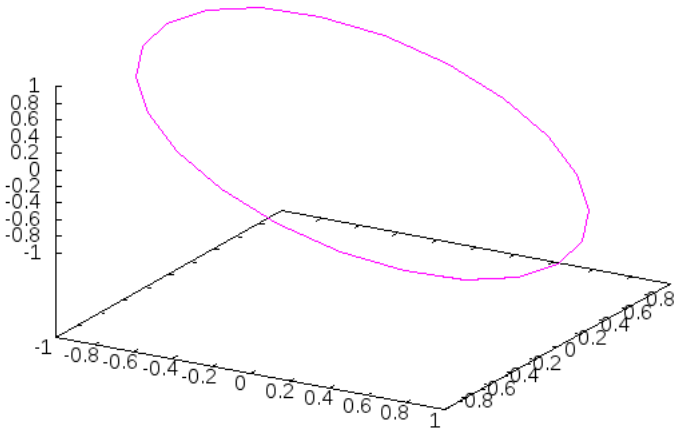


Figura 2: Grafica de $r(t = 4) = (10\sin(t), 2\cos(t), t)$

2.4. Longitud de arco y curvatura

Con un ejemplo muy similar al del manual de Maxima a continuación se calcula la curvatura de una función $r(t) = [\cos(t), \sin(t), t]$ obteniendo una curvatura $\kappa = \frac{1}{\sqrt{2}}$.

```
r(t) := [cos(t),sin(t),t];

define(rp(t), diff(r(t), t));
float(rp(1));
load(eigen);
uvect(rp(t));
trigsimp(rp(t));

define(T(t), trigsimp(rp(t)));
define(Tp(t), diff(T(t), t));
uvect(Tp(t));
trigsimp(uvect(Tp(t)));
define(N(t), trigsimp(uvect(Tp(t))));
load(vect);
express(T(t) ~ N(t));
trigsimp(T(t) ~ N(t));
define(B(t), trigsimp(T(t) ~ N(t)));
float(B(4));

sqrt(Tp(t) . Tp(t))/sqrt(rp(t) . rp(t));
trigsimp(sqrt(Tp(t) . Tp(t))/sqrt(rp(t) . rp(t)));
define(kappa(t), trigsimp(sqrt(Tp(t) . Tp(t))/sqrt(rp(t) . rp(t))));
```

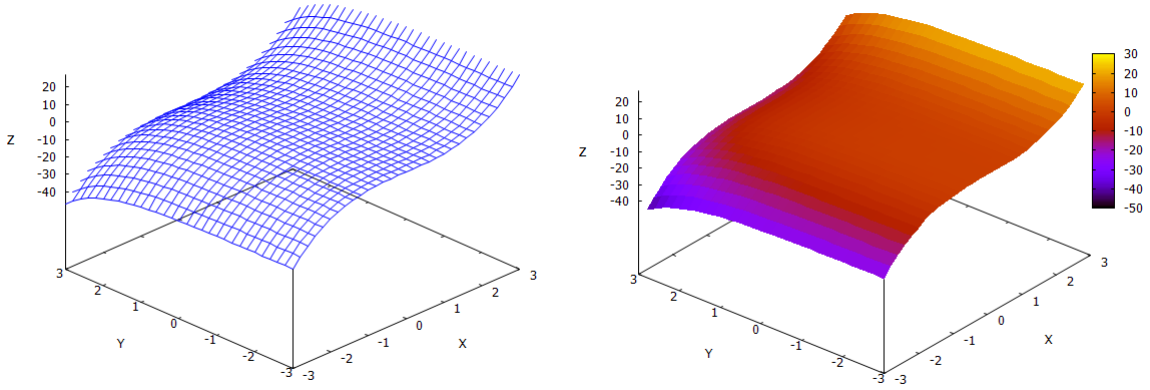
```
integrate(r(t), t);

(%o54) kappa(t):=1/sqrt(2)
```

y enseguida, la longitud de arco para la misma función, obteniendo una longitud de arco de $2^{3/2}\pi$.

```
r(t) := [cos(t), sin(t),t];
define(rp(t), diff(r(t), t));
integrate(trigsimp(sqrt(rp(t) . rp(t))), t, 0, 2*%pi);
(%o33) r(t):=[cos(t),sin(t),t]
(%o34) rp(t):=[-sin(t),cos(t),1]
(%o35) 2^(3/2)*%pi
```

3. Funciones de varias variables

Solo para introducir este tipo de funciones, dejaré aquí abajo el plot de una función de dos variables: $f(x, y) = (x^3 - e^y)$



3.1. Derivadas Parciales

Podemos tener una función de varias variables y de grado mayor a 2 en cada variable, pero Maxima calculará sus derivadas parciales de cualquier grado. Veamos la cuarta derivada parcial respecto a x y la segunda respecto a y de la función $G = f(x, y) = 2x^8y^2 + 3x^4y - 5xy^9$

```
G: (2*(x^8)*(y^2))+(3*(x^4)*y)-(5*x*(y^9));
diff(G,x,1,y,2,x,3);
/* [wxMaxima: input   end   ] */
(%o81) 6720*x^4
```

Se obtuvo que: $\frac{\partial G}{\partial^4 x \partial^2 y} = 6720x^4$

3.2. Aproximación lineal y diferenciales

Una forma de hacer aproximación lineal en Maxima es utilizando una serie de Taylor. A continuación un ejemplo para $f(x, y) = 3x^2\cos(2y)$ evaluada en [0,1] arrojando su aproximación $= 27\cos(10) + (18\cos(10)(x - 3) - 54\sin(10)(y - 5)) + ...$

```
f(x,y) := 3*x^2 * cos(2*y);
taylor(f(x,y), [x,y], [3,5], 1);
(%o93) f(x,y):=3*x^2*cos(2*y)
(%o94)/T/ 27*cos(10)+(18*cos(10)*(x-3)-54*sin(10)*(y-5))+...
```

Y de la función anterior, podemos encontrar el diferenciar completo con solo teclear "diff"sin especificar nada. Se obtiene: $6x\cos(2y)dx - 6x^2\sin(2y)dy$

```
f(x,y) := 3*x^2 * cos(2*y);
diff(f(x,y));
(%o95) f(x,y):=3*x^2*cos(2*y)
(%o96) 6*x*cos(2*y)*del(x)-6*x^2*sin(2*y)*del(y)
```

3.3. Regla de la cadena y diferenciación implícita

Ahora trabajemos con una función de dos variables $f(x, y)$ que a su vez dependen de otras dos variables $X(u, v), y(u, v)$. Maxima utiliza la regla de la cadena para obtener las derivdas de f respecto a s o a t. A contnuación, la función $f(X, y) = 3x^2\cos(2y)$ con $x = st$ y $y = 3t+1$ es derivada respecto a s y respecto a t, obteniendo $\frac{\partial f}{\partial s} = 6st^2\cos(2(3s^3(t+1))) - 54s^4t^3\sin(2(3s^3(t+1)))$ y $\frac{\partial f}{\partial t} = 6s^2t\cos(2(3s^3(t+1))) - 18s^5t^2\sin(2(3s^3(t+1)))$

```
f(x,y) := 3*x^2 * cos(2*y);
[x,y] : [s * t, s^3 * 3*t+1];
diff(f(x,y), s);
diff(f(x,y), t);
(%o103) f(x,y):=3*x^2*cos(2*y)
(%o104) [s*t,3*s^3*t+1]
(%o105) 6*s*t^2*cos(2*(3*s^3*t+1))-54*s^4*t^3*sin(2*(3*s^3*t+1))
(%o106) 6*s^2*t*cos(2*(3*s^3*t+1))-18*s^5*t^2*sin(2*(3*s^3*t+1))
```

Por como están definidas x y y, tenemos que derivar respecto a otras variables, como u, v o bien, "matarlas"para poder derivar a f respecto a ellas.

3.4. Derivadas direccionales y gradiente

En el siguiente código se muestra como calcular el gradiente con el comando grad de la función $f(x, y) = 3x^2\cos(2y)$, donde se obtiene que el gradiente ∇ de $f(x, y)$ es $\nabla f(x, y) = [6x\cos(2y), -6x^2\sin(2y)]$.

```
f(x,y) := 3*x*x*cos(2*y);
load(vect);
scalefactors([x,y]);
gdf: grad(f(x,y));
ev(express(gdf), diff);
define(gdf(x,y), %);
(%o1) f(x,y):=3*x*x*cos(2*y)
(%o2) "C:\maxima-5.38.0\share\maxima\5.38.0_dirty\share\vector\vect.mac"
(%o3) done
(gdf)3*grad(x^2*cos(2*y))
(%o5) [6*x*cos(2*y),-6*x^2*sin(2*y)]
```

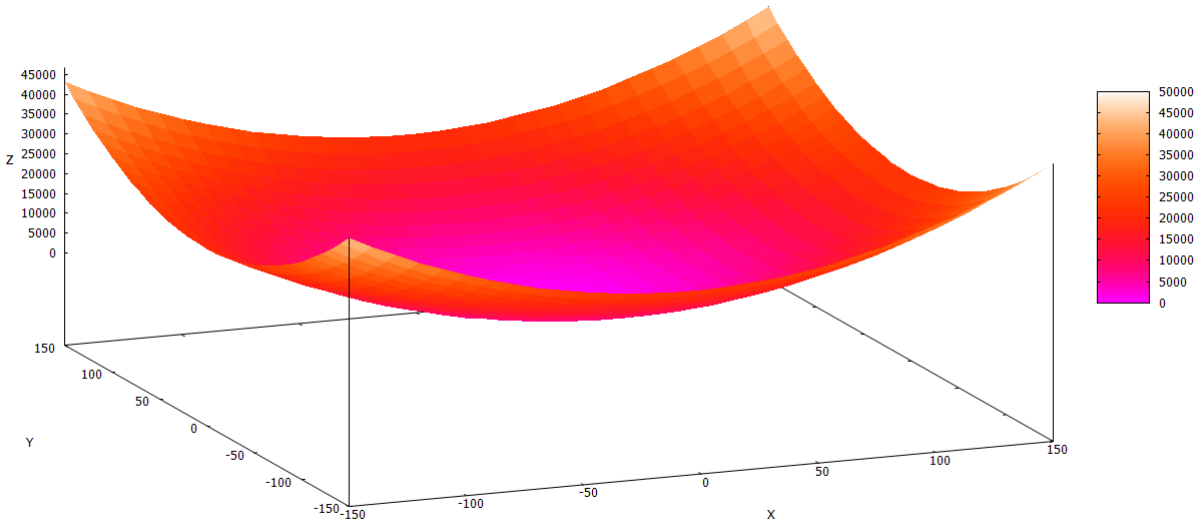
```
(%o6) gdf(x,y):=[6*x*cos(2*y),-6*x^2*sin(2*y)]
```

y para calcular la derivada direccional de la función anterior no hay una herramienta en específico, sino que se sigue el procedimiento aprendido en la teoría. Se obtiene una derivada direccional en el punto $[1, 0]$ con la dirección de $\overrightarrow{v} = [2, 3]$ de $\frac{12}{\sqrt{13}}$

```
f(x,y) := 3*x*x*cos(2*y);
load(vect);
scalefactors([x,y]);
gdf: grad(f(x,y));
ev(express(gdf), diff);
define(gdf(x,y), %);
v: [2,3];
(gdf(1,0) . v)/sqrt(v . v);
ev(%, diff);
float(%);
(%o1) f(x,y):=3*x*x*cos(2*y)
(%o2) "C:\maxima-5.38.0\share\maxima\5.38.0_dirty\share\vector\vect.mac"
(%o3) done
(gdf)3*grad(x^2*cos(2*y))
(%o5) [6*x*cos(2*y),-6*x^2*sin(2*y)]
(%o6) gdf(x,y):=[6*x*cos(2*y),-6*x^2*sin(2*y)]
(v)[2,3]
(%o8) 12/sqrt(13)
(%o9) 12/sqrt(13)
(%o10) 3.328201177351375
```

3.5. Optimización y extremos locales

Del paraboloide $z = f(x, y) = x^2 + 8x + y^2 - 4y + 20$, hacemos primero un plot para darnos una idea de su ubicación.



En el siguiente código se muestra cómo verificar si un punto es máximo o mínimo con el método que usualmente se ve en clase. Calculando las primeras derivadas, solucionando para x y y y luego calculando la segunda derivada para comprobar. En el punto $[-4, 2]$ existe un mínimo, ya que la segunda derivada evaluada en ese punto fue positiva.

```
f(x,y) :=  x^2 + 8*x +y^2 - 4*y +20;
load(draw);
draw3d(enhanced3d = true, palette=[2,4,12], explicit(f(x,y), x, -150, 150, y, -15
fx : diff(f(x,y), x);
fy : diff(f(x,y), y);
solve([fx,fy], [x,y]);
H: hessian(f(x,y), [x,y]);
determinant(H);
subst([x = -4, y = 2], diff(fx, x));
subst([x = -4, y = 2], determinant(H));
f(-4,2);
(%o78) f(x,y):=x^2+8*x+y^2+(-4)*y+20
0 errors, 0 warnings
(%o79) "C:\maxima-5.38.0\share\maxima\5.38.0_dirty\share\draw\draw.lisp"
(%o80) [gr3d(explicit)]
(fx)2*x+8
(fy)2*y-4
(%o83) [[x=-4,y=2]]
(H)matrix([2,0],[0,2])
(%o85) 4
(%o86) 2
(%o87) 4
(%o88) 0
```

3.6. Multiplicadores de Lagrange

Maxima utiliza el mismo método que se enseña en clase para calcular los valores extremos de una función sobre otra con los multiplicadores. A continuación se muestra el ejemplo del manual, en el que se encuentran los valores extremos de $f(x, y) = 2x^2 + y^2$ sobre la circunferencia $x^2 + y^2 = 1$.

```
f(x,y) := 2 * x^2 + y^2;
g: x^2 + y^2;
eq1: diff(f(x,y), x) = h * diff(g, x);
eq2: diff(f(x,y), y) = h * diff(g, y);
eq3: g = 1;
solve([eq1, eq2, eq3], [x, y, h]);
[f(1,0), f(-1,0), f(0,-1), f(0,1)];
(%o109) f(x,y):=2*x^2+y^2
(g)y^2+x^2
(eq1)4*x=2*h*x
(eq2)2*y=2*h*y
```

```
(eq3)y^2+x^2=1
(%o114) [[x=1,y=0,h=2],[x=-1,y=0,h=2],[x=0,y=-1,h=1],[x=0,y=1,h=1]]
(%o115) [2,2,1,1]
```

Integrales múltiples

4.1. Integrales dobles

Otra de las cosas que se pueden hacer en Maxima son las integrales dobles. Hagamos un ejemplo simple en el que se integra la función $f(x,y) = x^2+6x+y^2+4y-8xy+10$ acotada de $y = x^2$ a $y = 2x$ y de $x = 0$ a $x = 2$, resultando $\frac{-136}{105}$.

```
f(x,y) := x^2 - 6*x +y^2+4*y-8*x*y+10;
integrate(integrate(f(x,y), y, x^2, 2*x), x, 0, 2);
(%o141) f(x,y):=x^2-6*x+y^2+4*y+(-8)*x*y+10
(%o142) -136/105
```

4.2. Integración en coordenadas polares

Como también se vio en clase, a veces es necesario hacer un cambio de coordenadas para que nuestras integrales se vuelvan más sencillas. A continuación se hace la integral de $f(x,y) = 3x^2 + 4y^2$ haciendo un cambio a coordenadas polares de $x = r\cos\theta$ y $y = r\sin\theta$. Resultando $\frac{19\pi}{4}$.

```
f(x,y) := 3*x^2 + 4*y^2;
[x,y]: [r * cos(theta), r * sin(theta)];
integrate(integrate(f(x,y) * r, r, 0, 2*cos(theta)), theta, -%pi/2,
%pi/2);
(%o145) f(x,y):=3*x^2+4*y^2
(%o146) [r*cos(theta),r*sin(theta)]
(%o147) (19*%pi)/4
```

4.3. Integrales triples

Y ahora, con una sola línea de código, haremos la integral triple de una función de tres variables: $f(x,y,z) = 3x^2y^2z + 2xyz + x + y + z$, acotada por $0 < z < xy$, $0 < y < x$ y $-1 < x < 1$. El resultado es: $\frac{1}{3}$.

```
integrate(integrate(integrate(3*x^2*y^2*z+2*x*y*z+x+y+z,z,0,x*y),y,0,x),x,-1,1);
(%o3) 1/3
```

4.4. Integrales en coordenadas cilíndricas y esféricas

Otro de los cambios de coordenadas que suelen hacerse son las cilíndricas y las esferas, donde las variables x, y, z pasan a ser $r\cos\theta, r\,\mathrm{sen}\,\theta, z$ en las cilíndricas y $\rho\sin\phi\cos\theta, \rho\sin\phi\sin\theta, \rho\cos\theta$. En el siguiente código se hace una integral con cambio a coordenadas cilíndricas: $f(x,y,z) = x^2yz$ acotada por $-3 < z < 2$, $-2 < r < 2$ y $0 < \theta < \pi$, resultando $\frac{-64}{3}$.

```
f(x,y,z) := x*x*y*z;
[x,y,z] : [r*cos(theta), r*sin(theta), z];
integrate(integrate(integrate(f(x,y,z)*r, z,-3,2), r,-2,2), theta,0,%pi);
(%o36) f(x,y,z):=x*x*y*z
(%o37) [r*cos(theta),r*sin(theta),z]
(%o38) -64/3
```

y ahora un ejemplo de integral con cambio a coordenadas esféricas de la función anterior, ahora acotada por $-3 < \rho < 5$, $0 < \theta < \frac{\pi}{6}$ y $0 < \phi < \frac{\pi}{2}$, resultando $\frac{30117\pi}{128}$.

```
f(x,y,z) := x*x*y*z;
[x,y,z] : [rho*sin(phi)*cos(theta), rho*sin(phi)*sin(theta), rho*cos(theta)];
integrate(integrate(integrate(f(x,y,z)*rho^2*sin(phi),rho,-3,5),theta,0,%pi/6),
phi,0,%pi/2);
(%o51) f(x,y,z):=x*x*y*z
(%o52) [sin(phi)*rho*cos(theta),sin(phi)*rho*sin(theta),rho*cos(theta)]
(%o53) (30117*%pi)/128
```

4.5. Cambio de variables

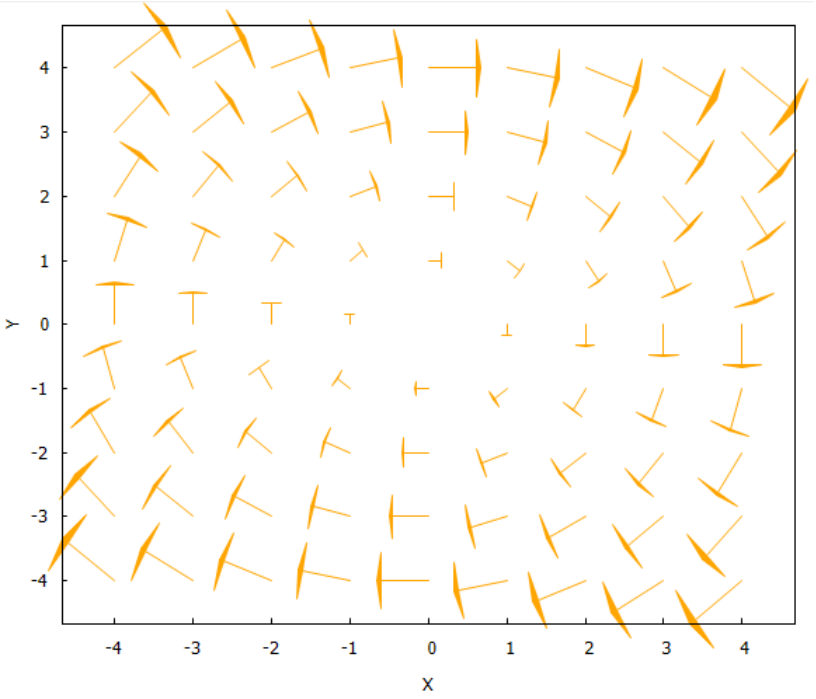
Un método más generalizado para calcular integrales es el cambio de variables en el que se hacen transformaciones lineales tales que ahora las variables dependen de otras: $x \Rightarrow x(u,v)$ y $y \Rightarrow y(u,v)$. Al igual que los casos anteriores, en Maxima es posible realizar las operaciones necesarias para llevar a cabo lo aprendido en clase. A continuación se hace una integral de la función: $f(x,y) = x^2 + y^2 + 3xy$ donde $x = u^2 - v$ y $y = v + u^2$, acotadas por $0 < u < 1$ y $-2 < v < 5$. Resultado: $\frac{-196}{3}$.

```
f(x,y) := x^2 + y^2 +3*x*y;
[x,y]: [u^2 - v, v + u^2];
J: jacobian([x,y], [u,v]);
J: determinant(J);
integrate(integrate(f(x,y) * J, u,0,1), v,-2,5);
(%o21) f(x,y):=x^2+y^2+3*x*y
(%o22) [u^2-v,v+u^2]
(J)matrix([2*u,-1],[2*u,1])
(J)4*u
(%o25) -196/3
```

5.2. Campos vectoriales

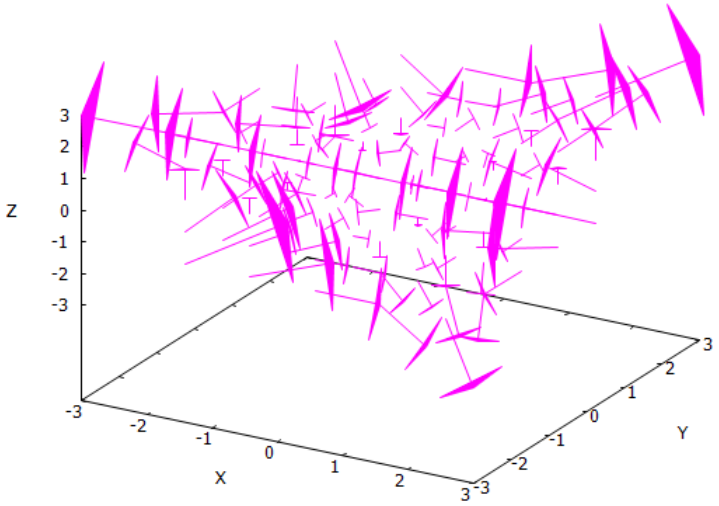
Para mostrar que Maxima también grafica campos vectoriales, a continuación se muestra un ejemplo muy común $F(x,y) = (y, -x)$, el cual son círculos concéntricos al origen.

```
load ( draw );
coord : setify ( makelist (k ,k , -4 ,4));
points2d : listify ( cartesian_product ( coord , coord ));
vf2d (x , y ):= vector ([ x , y ] ,[ y , -x ]/6);
vect2 : makelist ( vf2d ( k [1] , k [2]) , k , points2d );
apply ( draw2d , append ([ color = orange ] , vect2 ));
```



y en seguida se muestra un plot en 3d de el campo vectorial $F(x, y, z) = (zy, xz, xy)$:

```
coord : setify ( makelist (k ,k , -2 ,2));
points3d : listify ( cartesian_product ( coord , coord , coord ));
vf3d (x ,y , z ):= vector ([ x ,y , z ] ,[z*y ,x*z , y*x ]/4);
vect3 : makelist ( vf3d ( k [1] , k [2] , k [3]) , k , points3d );
apply ( draw3d , append ([ color = magenta ] , vect3 ));
```



5.2. Integrales de línea

Una integral de línea o curvilínea es aquella integral cuya función es evaluada sobre una curva. Y es que en los cursos de Cálculo aprendemos a resolverlas, pero con Maxima es posible evitar hacer los cálculos detallados para encontrar dichas integrales.

Esta herramienta suele representar la longitud de una curva en el espacio o bien, el trabajo que se realiza para mover algún objeto a lo largo de una trayectoria teniendo en cuenta campos de fuerzas (descritos por campos vectoriales) que actúen sobre el mismo [**4**]. Si tenemos una curva suave C definida por las ecuaciónes paramétricas: $x = x(t)$, $y = y(t)$, $z(t)$, para un intervalo $a \leq t \leq b$, y una función f definida en C. Solemos trabajar con una función vetorial$\mathbf{r}$ en lugar de las ecuaciones paramétricas: $\mathbf{r}(t) = [x(t), y(t), z(t)]$ para todo $a \leq t \leq b$.

- Longitud de arco
  Estas integrales son de la forma

  $$\int_C f(x, y, z)\, ds$$

  la cual puede ser reparametrizada como

  $$\int_a^b f(x, y, z)\sqrt{(\frac{dx}{dt})^2 + (\frac{dy}{dt})^2 + (\frac{dz}{dt})^2}\, dt$$

  o más compactamente, como:

  $$\int_a^b f(r(t))|r'(t)|\, dt$$

  . A continuación se integra $f(x, y, z) = x^2y + y^2z + z^2x$ a lo largo de la curva C parametrizada por $(4\cos t, 9\sin t, t)$, dando como resultado 2894,458139487642.

  ```
  f(x,y,z) := y*x^2 + x*y^2 + x*z^2;
  [x,y,z]: [4*cos(t), 9*sin(2*t), t];
  rp: diff([x,y,z], t);
  romberg(f(x,y,y)*sqrt(rp . rp), t, 0, 1);
  f(x,y,z):=y*x^2+x*y^2+x*z^2
  [4*cos(t),9*sin(2*t),t]
  (rp)[-4*sin(t),18*cos(2*t),1]

  2894.458139487642
  ```

- Trabajo realizado por un campo
  Aquí, tratamos de programar integrales de la forma

  $$\int_C \mathbf{F} \cdot\, dr = \int_C \mathbf{F}(r(t)) \cdot r'(t)\, dt$$

  también escrita como

  $$\int_C P\, dx + Q\, dy + R\, dz$$

  donde $\mathbf{F} = (P, Q, R)$. A continuación trabajamos con $\mathbf{F}(x, y, z) = (x^2yz, xy^2z, xyz^2)$. Integraremos a lo largo de la curva C parametrizada por $(t, t^2 + t, t^3 + t^2)$, para $-3 \leq t \leq 4$. El resultado es:1,072427226111111 $\times 10^7$

```
F(x,y,z) := [x*x*y*z, x*y*y*z, x*y*z*z];
[x,y,z]: [t, t^2+t, t^3+t^2];
romberg(F(x,y,z) . diff([x,y,z], t), t, -3, 4);
F(x,y,z):=[x*x*y*z,x*y*y*z,x*y*z*z]
[t,t^2+t,t^3+t^2]

1.072427226111111*10^7
```

5.3. Campos vectoriales conservativos y búsqueda de potenciales escalares
Sabemos por la teoría vista en los cursos de cálculo que un campo vecotorial $\mathbf{F}$ as conservativo si existe una función f tal que $\mathbf{F} = \nabla f$.

Podemos revisar si un campo es conservatico con su rotacional. Por ejemplo, a continuación, revisamos el campo $\mathbf{F}(x, y) = (3x^2y+3xy^2, y^2)$, nos resulta que su rotacional es: $\nabla \times \mathbf{F} \neq 0$, por lo que se trata de un campo no conservativo.

```
F(x,y) := [3*x^2-y, 5*x^2*y^3];
load(vect);
scalefactors([x,y]);
curl(F(x,y));
express(%);
ev(%, diff);
F(x,y):=[3*x^2-y,5*x^2*y^3]
(%o2) "C:\maxima-5.38.0\share\maxima\5.38.0_dirty\share\vector\vect.mac"
done
curl([3*x^2-y,5*x^2*y^3])
'diff((5*x^2*y^3),x,1)-'diff((3*x^2-y),y,1)
 10*x*y^3+1
```

Bibliografía

[1] Física Computacional(2016-2) *Actividad 7*. Recuperado en marzo de 2016 de http://computacional1.pbworks.com/w/page/105676740/Actividad%207%20(2016-1)

[2] Wikipedia *Espacio Fásico*. Recuperado en marzo de 2016 de https://es.wikipedia.org/wiki/Espacio_f%C3%A1sico

[3] Scipy-Cookbook .Recuperado en marzo de 2016 de http://scipy-cookbook.readthedocs.org/items/LoktaVolterraTutorial.html

[4] Wikipedia *Integral de línea*. Recuperado en marzo de 2016 de https://es.wikipedia.org/wiki/Integral_de_l%C3%ADnea

[5] GKerns *Multivariable Calculus with Maxima*. Recuperado en marzo de 2016 de http://gkerns.people.ysu.edu/maxima/maximaintro/maximaintro.pdf