PRACTICA 1. INTEGRALES MÚLTIPLES. CAMPOS ESCALARES Y VECTORIALES.

1 Integrales múltiples

Con wxMAXIMA es posible realizar integrales múltiples que se resuelvan aplicando el teorema de Fubini:no hay más que escribir el comando "integrate" (igual que para una variable). Veamos el siguiente ejemplo:

```
--> integrate(integrate(3*x*y,x,0,1),y,2,6);
```

o, cambiando el orden de integración

```
--> integrate(integrate(3*x*y,y,2,6),x,0,1);
```

En el ejemplo anterior el recinto de integración era rectangular, por lo que es sencillo expresar los intervalos donde varían x e y. Cuando el recinto de integración no es tan sencillo nos toca a nosotros definir los límites concretos de la integral; para eso, pueden ayudarnos los dibujos que hace MAXIMA.

Veamos el siguiente ejemplo: queremos calcular la integral de la función f(x,y)=e^{x/y} en el recinto del primer cuadrante (x e y positivas) encerrado por las curvas x=y^3 y x=y^2. Buscamos en qué puntos se cortan las curvas:

```
--> solve(y^3=y^2,y);
```

Pintamos la gráfica. Para ello previamente llamamos al paquete "draw" (también podemos hacerlo usando plot2d, como hemos visto en la práctica 0):

```
--> load(draw)$
    draw2d(color=blue,implicit(y^3=x,x,0,1,y,0,1),color=red,
    implicit(y^2=x,x,0,1,y,0,1));
```

Observamos que hemos dibujado en azul la curva x=y^3, mientras que en rojo aparece la curva x=y^2. Esta gráfica nos ayuda a ver los límites de integración: Observamos que si "x" varía en [0,1], entonces "y" varía en [x^(1/3),x^(1/2)]; o más facilmente, si "y" varía en [0,1], entonces "x" varía en [y^3,y^2]. Si tomamos esta última opción, podemos integrar y obtenemos el resultado:

```
--> integrate(integrate(exp(x/y),x,y^3,y^2),y,0,1);
```

Veamos un ejemplo más. Queremos integrar la función x^2+y^2 en el recinto descrito de la siguiente manera
{(x,y)\ (x^2+y^2)^2 <= 4(x^2-y^2), x>=0}.
Si intentamos dibujar la gráfica en implícitas, vemos que nos va a salir complicado, e incluso wxMaxima tarda demasiado tiempo en realizar la representación:

```
-->  load(draw)$
     draw2d(color=blue,implicit((x^2+y^2)^2=4*(x^2-y^2),x,0,2,y,0,2);
```

Por ello, interrumpimos el programa, ya que observamos que un cambio a polares nos facilitaría el trabajo, pero wxMaxima no lo realiza automáticamente; es por ello por lo que debemos hacerlo nosotros.
El dominio, en polares, se puede describir como:
{r<= 2*sqrt(cos(2t))con t en el intervalo [-pi/4,pi/4]}

Si dibujamos la curva r= 2*sqrt(cos(2t)), se observa que t varia en el intervalo [-pi/4,pi/4]: (Hacemos la gráfica como hemos visto en el apartado3.2 de la Práctica 0)

```
-->  plot2d([2*(cos(2*ph)^(1/2))], [ph,-%pi/4,%pi/4],
     [plot_format, gnuplot],
     [gnuplot_preamble, "set polar;set size ratio 1; set zeroaxis;"],
     [x,0,3])$
```

Entonces la integral quedaría:
(la función a integrar es, x^2+y^2, que en polares es r^2; por tanto integraremos r^3, ya que tenemos que multiplicar por el jacobiano r):

```
-->  integrate(integrate(r^3,r,0,2*sqrt(cos(2*t))),t,-%pi/4,%pi/4);
```

Nota:puede daros un error o preguntaros si el cos(2*t) es positivo, negativo o cero. Contestad que positivo; el error se debe a que no puede manejar números imaginarios como límites de integración. Una solución sencilla es cambiar la anterior por la instrucción siguiente, donde el coseno va en valor absoluto:

```
-->  integrate(integrate(r^3,r,0,2*sqrt(abs(cos(2*t)))),
     t,-%pi/4,%pi/4);
```

2 Campos Escalares y Vectoriales

Veamos ahora algunas instrucciones sencillas para trabajar con cálculo vectorial. Comenzamos cargando el paquete adecuado:

```
-->  load(vect)$
```

Vamos ahora a definir un campo escalar, al que llamamos f,
y que es una función de 3 variables x, y, z:

```
--> f(x,y,z):= y^2+z^2-x;
```

Le pedimos a Maxima su gradiente y después su gradiente en (2,3,-1):

```
--> gradf:[diff(f(x,y,z),x),diff(f(x,y,z),y),diff(f(x,y,z),z)];
```

```
--> at(gradf,[x=2,y=3,z=-1]);
```

Notemos que hemos calculado el gradiente definiéndolo como un vector cuyas componentes son las derivadas parciales de la función escalar f.

El gradiente puede calcularse de manera más rápida usando grad(f):

```
--> express(grad(f(x,y,z)));
```

aunque para que lo muestre como estamos habituados a verlo haremos:

```
--> ev(%, diff);
```

Otra opción para calcular el gradiente es usar el paquete linearalgebra que contiene el comando jacobian (así de paso aprendemos como se puede calcular la matriz jacobiana):

```
--> load(linearalgebra)$
    jacobian([f(x,y,z)],[x,y,z]);
    at(%,[x=2,y=3,z=-1]);
```

También podemos calcular la matriz hessiana de f:

```
--> hessian(f(x,y,z),[x,y,z]);
```

o el Laplaciano (d^2/dx^2 + d^2/dy^2 + d^2/dz^2) de un campo escalar:

```
--> express(laplacian(f(x,y,z)));
    ev(%,diff);
```

Definamos ahora un campo vectorial

```
--> F: [x*y,x^2*z,exp(x+y)];
```

Vamos a calcular su divergencia:

```
--> div(F);
```

```
--> express(%);
    ev(%, diff);
```

Exactamente igual, se calcula el rotacional, usando el comando "curl" en lugar de "div".

```
--> express(curl(F));
    ev(%,diff);
```

3 Ejercicios

PROBLEMA 1. Calcula las siguientes integrales dobles:
a. f(x,y) = sqrt(4x^2-y^2) entre x=1, y=0, y=x
b. f(x,y) = x*e^{-x^2/2}, entre x=0, y=1, y=x^2

PROBLEMA 2. Usando integración doble o triple, calcular el volumen de las siguientes regiones, representándolas previamente:
a. 0<=z<=x^2+y^2 (paraboloide), x+y<=1 (plano), en el primer octante.
b. x^2+y^2<=z^2 (cono), x^2+y^2+z^2<= 4 (esfera).

PROBLEMA 3. Considera el campo vectorial F: [x*y,x^2*z,exp(x+y)]. Súmalo al gradiente del campo escalar f:x^2*z*cosy y calcular la divergencia y el rotacional del campo resultante.

PROBLEMA 4. Considera el campo vectorial F anterior y el campo A (gradiente de x*y*z*tan(y)). Calcula la divergencia de (F+A) y la suma de la divergencia de F y la divergencia de A. ¿coinciden ambos valores?

PROBLEMA 5.
a. Considera un campo escalar cualquiera f (por ejemplo, el del Problema 3 anterior). Probar que rot(grad(f))es el vector nulo.
b. Considera un campo vectorial cualquiera F (por ejemplo, el del Problema 3). Probar que se verifica div(rot(F))=0

INSTRUCCIONES PARA LA ENTREGA DE LAS PRÁCTICAS:
- Las prácticas hay que hacerlas en fichero Maxima (extensión .wxm) y remitirlas por email.
- Aconsejable realizar un fichero para cada práctica (llamandolo, por ejemplo, Practica 1-nombre alumno.wxm)
- En los ficheros ir explicando lo que se va a realizar, incluyendo los comentarios que sean necesarios, como en el ejemplo que va a continuación.
- Las erratas que se cometan, se pueden eliminar, por lo que no es preciso que las entradas y salidas del programa sean consecutivas.

```
--> /*Esto es un ejemplo de como hay que realizar las prácticas*/
    /*En este caso se trata de calcular la integral de la función
    3*x*y en el intervalo [0,1]x[2,6]*/

    integrate(integrate(3*x*y,x,0,1),y,2,6);
```

```
--> /*Y si necesito hacer referencia al resultado anterior vuelvo
    a escribir un comentario debajo del mismo*/;
```

```
--> /*aunque me den errores
    a la hora de introducir los comentarios, no pasa nada*/;
```