PRÁCTICA 3. CÁLCULO DIFERENCIAL EN VARIAS VARIABLES. INTEGRALES MÚLTIPLES

1 INTRODUCCIÓN.

El alumno debe ir realizando todos los EJEMPLOS que aparecen en esta práctica e ir anotando los resultados en la hoja que se le ha entregado. Además de estos ejemplos, hay planteados unos EJERCICIOS, que, a diferencia de lo que ocurre con los EJEMPLOS, no vienen desarrollados en el lenguaje Maxima, sino que tendrá que ser el alumno el que tenga que introducirlos para obtener el resultado.

2 Cálculo diferencial en varias variables.

2.1 Derivadas parciales de funciones de varias variables

Vamos a ver como calcular las derivadas parciales de funciones de varias variables, tanto de primer orden como de orden superior.

```
--> /*EJEMPLO 1: Para calcular la primera derivada parcial respecto
    de x, de f(x,y,z)=x+sin(x*y)+log(x*y*z), y la llamamos fx*/

    f(x,y,z):=x+sin(x*y)+log(x*y*z)$
    fx:diff(f(x,y,z),x);
```

```
--> /*EJEMPLO 2: Y lo mismo para las derivadas parciales respecto
    de z e y*/

    fz:diff(f(x,y,z),z);
    fy:diff(f(x,y,z),y);
```

```
--> /*EJEMPLO 3: Podemos evaluar fx en un punto
    (%pi/2,y=1/2,z=1)*/

    fx,x=%pi/2,y=1/2,z=1;
```

```
--> /*EJEMPLO 4: Podemos calcular derivadas sucesivas, como por
    ejemplo, fxx, fxy o fxzy*/

    fxx:diff(f(x,y,z),x,2);
    fxy:diff(f(x,y,z),x,1,y,1);
    fxzy:diff(f(x,y,z),x,1,z,1,y,1);
```

```
--> /*EJEMPLO 5: Comprobamos que la función z(x,t)=sen(x-c*t),
    con c constante, satisface la ecuación de ondas
    ftt=c^2 fxx*/

    z(x,t):=sin(x-c*t)$
    is(ratsimp(diff(z(x,t),t,2)=c^2*diff(z(x,t),x,2)));
```

EJERCICIO 1.
a) Sea f(x,y)=5x^2-2xy+3y^3 x^2. Calcular las siguientes derivadas:
fxy, fyx, fxx(3,4).

b) Probar que la función f(x,y)=e^x*sin(y) es armónica, es decir,
verifica la ecuación fxx+fyy=0.

c) Calcular la pendiente de la recta tangente a la gráfica de
f(x,y)=x*y^3+x^3*y en P(1,-1,-2) en la dirección del eje x y
en la del eje y.

2.2 Extremos relativos de funciones de dos variables

Vamos a ver como podemos calcular los puntos críticos de una función
con dos variables, calcular su hessiano, y evaluar éste en los puntos
críticos.

```
--> /*EJEMPLO 6: Vamos a calcular los puntos críticos de la función
    f(x,y)=x*y*exp(-x^2-y^2)*/

    f(x,y):=x*y*%e^(-x^2-y^2)$
    puntoscriticos:solve([diff(f(x,y),x),diff(f(x,y),y)],[x,y]);
```

Ahora definimos el hessiano de una función f(x,y) con Maxima
y lo llamamos Hessiano:

```
--> /*EJEMPLO 7: Definicion general de Hessiano*/

    Hessiano(x,y):=(diff(f(x,y),x,2)*diff(f(x,y),y,2))-
    (diff(f(x,y),x,1,y,1)*diff(f(x,y),x,1,y,1));
```

```
--> /*EJEMPLO 8: Vamos a obtener el Hessiano para la función anterior
    (al que llamaremos hessianof) y vamos a evaluar éste en los puntos
    críticos obtenidos anteriormente*/

    Hessiano(x,y)$
    hessianof(x,y):=ev(Hessiano(x,y),diff)$
    hessianof(x,y);
```

```
--> /*EJEMPLO 9 (continuación)*: Vamos a ver que en los puntos
    críticos (3) y (4) de la lista puntoscriticos obtenida
    anteriormente la función presenta mínimos relativos*/

    is(at(hessianof(x,y),puntoscriticos[3])>0);
    is(at(hessianof(x,y),puntoscriticos[4])>0);
```

```
--> /*EJEMPLO 9 (continuación): Ahora evaluamos fxx en estos puntos
    críticos*/

    fxx(x,y):=diff(f(x,y),x,2)$
    fxx(x,y);
    is(at(fxx(x,y),puntoscriticos[3])>0);
    is(at(fxx(x,y),puntoscriticos[4])>0);
```

Por tanto la función tiene dos mínimos en los puntos críticos (3) y (4) obtenidos anteriormente.

```
--> /*EJEMPLO 9 (continuación): Ahora representaremos gráficamente
    la superficie y los puntos donde se alcanzan estos mínimos*/

    load(draw)$
    draw3d(color=green,explicit(f(x,y),x,-2,2,y,-2,2),color=black,
    point_size=2,point_type=filled_circle,
    points([[1/sqrt(2),-1/sqrt(2),
    f(1/sqrt(2),-1/sqrt(2))],[-1/sqrt(2),1/sqrt(2),
    f(-1/sqrt(2),1/sqrt(2))]]));
```

```
--> /*EJEMPLO 10: Ahora hacemos lo mismo con los puntos críticos
    (2) y (5) de la lista puntoscriticos anterior*/

    is(at(hessianof(x,y),puntoscriticos[2])>0);
    is(at(hessianof(x,y),puntoscriticos[5])>0);
    is(at(fxx(x,y),puntoscriticos[2])<0);
    is(at(fxx(x,y),puntoscriticos[5])<0);
```

```
--> /*EJEMPLO 10 (Continuación): Ahora representamos la superficie
    y los puntos máximos*/

    load("draw")$
    draw3d(color=green,explicit(f(x,y),x,-2,2,y,-2,2),color=black,
    point_size=2,point_type=filled_circle,
    points([[1/sqrt(2),1/sqrt(2),f(1/sqrt(2),1/sqrt(2))],
    [-1/sqrt(2),-1/sqrt(2),f(-1/sqrt(2),-1/sqrt(2))]]));
```

Por lo que f tendrán en dichos puntos máximos relativos.

Veamos por último que ocurre con el punto crítico (0,0):

```
--> /*EJEMPLO 11: Podemos ver que en (0,0) hay un punto de silla
    por el criterio de ser el hessiano negativo*/

    is(at(hessianof(x,y),puntoscriticos[1])<0);
```

2.3 Derivada de la función compuesta

Veremos ahora como aplicar la regla de la cadena para calcular derivadas de funciones compuestas de varias variables. Con la orden depends podemos declarar dependencias de funciones.
Veremos que en el caso de tener funciones compuestas, la orden diff aplica la regla de la cadena. Además calcularemos derivadas de funciones compuestas cuyas funciones componentes vienen definidas de forma explícita.

```
-->  /*EJEMPLO 12: Sea la función de dos variables f(x,y), donde
     x e y son funciones de una variable independiente t: x=x(t)
     e y=y(t). Para calcular la derivada df/dt, declaramos las
     dependencias con la orden depends y derivamos con diff:*/

     depends(f,[x,y],[x,y],t);
     diff(f,t);
     remove(all,dependency)$
```

Notemos que en este último ejemplo, la orden diff, aplica la regla
de la cadena para calcular la derivada:
df/dt = @f/@x dx/dt + @f/@y dy/dt
La orden remove(all,dependency) se usa para cancelar las dependencias
de las variables.

```
-->  /*EJEMPLO 13: Consideramos el caso de una función de dos
     variables x e y, f(x,y), que a su vez dependen de otras dos
     variables independientes s y t, x(s,t), y(s,t). Queremos calcular
     las derivadas parciales @f/@s y @f/@t:*/

     depends(f,[x,y],[x,y],[s,t]);
     diff(f,s);
     diff(f,t);
     remove(all,dependency)$
```

```
-->  /*EJEMPLO 14: Sea f(x) tal que x=x(t). Calculamos la derivada
     segunda de f respecto de t:*/

     depends(f,x,x,t)$
     diff(f,t,2);
     remove(f,dependency)$
```

Suele ser habitual que la relación de dependencia nos venga dada en
forma explícita, como veremos en los siguientes ejemplos:

```
-->  /*EJEMPLO 15: Sea f(x,y)=xy+y^2, tal que x(t)=sen(t), y(t)=e^t.
     Calcularemos df/dt:*/

     f(x,y):=x*y+y^2$
     diff(f(sin(t),%e^t),t);
     kill(f)$
```

```
-->  /*EJEMPLO 16: Sea f(x,y)=xy+x^2+y^2, donde x(t)=rcos(t),
     y(t)=rsen(t). Calculamos @f/@r y @f/@t:*/

     f(x,y):=x*y+x^2+y^2$
     diff(f(r*cos(t),r*sin(t)),r);
     diff(f(r*cos(t),r*sin(t)),t);
```

```
-->  /*EJEMPLO 16 (cont.): Y si queremos evaluar @f/@t en el punto
     (pi,0):*/

     diff(f(r*cos(t),r*sin(t)),t)$
     %,r=%pi,t=0;
     kill(f)$
```

2.4 Derivación implícita

Para usar la derivación implícita, es más aconsejable derivar directamente en la ecuación que nos den, aunque previamente hemos de definir la dependencia entre las variables, como vemos en los ejemplos siguientes:

```
-->  /*EJEMPLO 17: Sea la ecuación 2sin(x)cos(y)=1. Calculamos
     y'(x) por derivación implícita:*/

     depends(y,x)$
     ec:2*sin(x)*cos(y)=1;
     diff(ec,x);
     der:solve(%,'diff(y,x));
```

Notemos que hemos declarado la dependencia con depends, hemos definido la ecuación y la hemos llamado ec, y hemos derivado y despejado la derivada. La salida es una lista de un solo elemento que guardamos en una variable que llamamos der. Este valor es, del primer (único) elemento de esa lista, la parte que está a la derecha del igual. Podemos usar para extraer esa parte la orden second(exp) y lo guardamos en una variable que llamamos dydx.

```
-->  /*EJEMPLO 17 (cont.):*/

     dydx:second(der[1]);
```

```
-->  /*EJEMPLO 17 (cont.): Y si ahora calculamos la derivada de la
     función en el punto x=pi/4, y=pi/4, y'(pi/4)*/

     dydx,x=%pi/4,y=%pi/4;
     remove(all,dependency)$
     kill(all)$
```

```
-->  /*EJEMPLO 18: Calculamos las derivadas parciales primeras de
     z(x,y)  tal que z está definida implícitamente por la ecuación
     x^2+y^2+z^2=25*/
     /*En primer lugar definimos la dependencia de z y a la ecuación
     la llamamos ec*/

     depends(z,[x,y])$
     ec:x^2+y^2+z^2=25$
```

```
-->  /*EJEMPLO 18 (cont.): Para calcular dz/dx derivamos toda la
     ecuación con respecto a x y despejamos. Guardamos la derivada
     dz/dx en una variable llamada dzdx (para ello usamos la orden
     second como anteriormente)*/

     diff(ec,x);
     solve(%,diff(z,x));
     dzdx:second(%[1]);
```

```
-->  /*EJEMPLO 18 (cont.): Algo similar hemos de hacer para calcular
     dz/dy. En este caso guardamos la derivada parcial en la
     variable dzdy:*/

     diff(ec,y)$
     solve(%,diff(z,y))$
     dzdy:second(%[1]);
```

```
--> /*EJEMPLO 18 (cont.): Para calcular la segunda derivada de z
    respecto a dos veces x, derivamos la ecuación dos veces con
    respecto a x*/

    diff(ec,x,2);
```

```
--> /*EJEMPLO 18 (cont.): Observamos que en la ecuación derivada
    tenemos tanto dz/dxx como dz/dx. Antes de despejar, tendremos
    que sustituir la derivada dz/dx que hemos calculado antes y que
    hemos guardado en dzdx. Para hacer esto, usamos la orden
    subst(exp1,exp2,exp), que sustituye en la expresión exp,
    la exp2 por exp1. Por último, despejamos dz/dxx:*/

    diff(ec,x,2);
    subst(dzdx,'diff(z,x),%);
    solve(%,diff(z,x,2));
```

```
--> /*EJEMPLO 18 (cont.): Para calcular dz/dyy:*/

    diff(ec,y,2);
    subst(dzdy,'diff(z,y),%);
    solve(%,diff(z,y,2));
```

```
--> /*EJEMPLO 18 (cont.): Calculamos dz/dxdy derivando la ecuación
    primero con respecto a y y luego con respecto a x:*/

    diff(ec,x,1,y,1);
    subst(dzdy,'diff(z,y),%);
    subst(dzdx,'diff(z,x),%);
    solve(%,diff(z,x,1,y,1));
```

EJERCICIO 2.
a) Calcular la rectan tangente a la curva x^2(x^2+y^2)=y^2 en el punto (1/(2^1/2),1/(2^1/2)).

b) Calcular las derivadas y', y'', y''', donde x^2+xy+y^2=3.

c) Calcula dx/dt y dy/dt si x+y+t=0, x^2+y^2+t^2 = 1.

3 *INTEGRALES MÚLTIPLES.*

3.1 Integrales dobles

En una práctica anterior hemos visto cómo podemos utilizar Maxima para resolver integrales de funciones de una variable. ¿Qué necesitamos para calcular integrales de funciones de varias variables? Básicamente, el teorema de Fubini, que nos permite calcular una integral doble como dos integrales simples (idem para integrales triples).

```
--> /*EJEMPLO 19: La integral de la función f(x,y)=3xy en [0,1]x[2,6]*/
    integrate(integrate(3*x*y,x,0,1),y,2,6);
```

```
--> /*EJEMPLO 20: Cambiando el orden de integración*/
    integrate(integrate(3*x*y,y,2,6),x,0,1);
```

```
--> /*EJEMPLO 21: La integral de la función f(x,y)=x^2+y^2 en
    [0,2]x[1,5]*/
    integrate(integrate(x^2+y^2,x,0,2),y,1,5);
```

Podemos calcular estas integrales en recintos más complicados:
EJEMPLO 4: Si queremos calcular la integral de f(x,y)=x en el primer cuardrante del círculo x^2+y^2<=1, es aconsejable que en primer lugar dibujemos el conjunto:

```
--> /*EJEMPLO 22:*/
    load(draw)$
    draw2d(implicit(x^2+y^2=1,x,0,1,y,0,1));
```

Así podemos ver la variabilidad de cada variable (en este caso, si x varia en [0,1], se tiene que y varia entre 0 y (1-x^2)^(1/2), por lo que podemos calcular la integral sin más que

```
--> /*EJEMPLO 22 (cont.):*/
    integrate(integrate(x,y,0,sqrt(1-x^2)),x,0,1);
```

Como nos da un error (ya que inicialmente no hemos indicado que la variable x toma un valor entre 0 y 1, pondremos

```
--> /*EJEMPLO 22 (cont.):*/
    integrate(integrate(x,y,0,sqrt(abs(1-x^2))),x,0,1);
```

Observemos que que en este ejemplo no ha sido necesario cambiar a coordenadas polares. Para integrales en varias variables, interesa cambiar de variable cuando el dominio o la función son complicados. En el primer caso, si no podemos escribir el dominio de una de las formas habituales (con x variando entre constantes e y variando entre funciones de x; o al revés) tendremos que cambiar de variable nosotros (Maxima no lo hace automáticamente). El caso de funciones complicadas no nos debe preocupar, pues Maxima se encarga de resolver la integral en la inmensa mayoría de los casos, y si no se puede resolver directamente, siempre queda la posibilidad de aproximar el valor de integral (lo que haremos usando quad_quags(f(x),x,a,b) o romberg(f(x),x,a,b); ambas sentencias nos dan el valor aproximado de la integral de f(x) en [a,b]).

```
--> /*EJEMPLO 23: Calcular la integral de la función f(x,y)=exp(x/y) ,
    en el conjunto de puntos (x,y) tal que 0<=y^3<=x<=y^2*/
    /*Primero calculamos los puntos de corte y dibujamos la gráfica
    de la región*/
    solve(y^3=y^2,y);
    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));
```

```
--> /*EJEMPLO 23 (cont.): De lo hecho se deduce que la integral que
    queremos calcular es:*/
    integrate(integrate(exp(x/y),x,y^3,y^2),y,0,1);
```

Veamos un ejemplo con función a integrar relativamente sencilla pero con dominio no tanto. Aquí observaremos que no hay manera fácil de escribir la integral en términos de x e y. Parece mejor cambiar a coordenadas polares. Aunque antes de hacerlo podemos echar un vistazo al dominio. Recordamos que el rango de valores de las variables x e y es algo que debemos decidir nosotros:

```
--> /*EJEMPLO 24: Calcular la integral de la función f(x,y)=x^2+y^2 ,
    en el conjunto de puntos (x,y) tal que (x^2+y^2)^2<=4(x^2-y^2),
    con x>=0:*/
    load(draw)$
    draw2d(color=blue,implicit((x^2+y^2)^2=4*(x^2-y^2),
    x,0,2,y,-1,1));
```

Operando aparte y viendo la gráfica, llegamos a la conclusión que el argumento t varía en [-pi/4,pi/4], mientras que el módulo r varía entre 0 y 2*(cos(2*t))^(1/2). Además, la función f(x,y) se transforma en f(r,t)=r^2. Por tanto, y sin que se os olvide multiplicar f por el jacobiano (que es r), tendremos

```
--> /*EJEMPLO 24 (cont.): La integral a calcular en polares es:*/
    integrate(integrate(r^3,r,0,2*sqrt(cos(2*t))),t,-%pi/4,%pi/4);
```

Como de nuevo nos da un error (por la raíz cuadrada), pondremos:

```
--> /*EJEMPLO 24 (cont.): La integral a calcular en polares es:*/
    integrate(integrate(r^3,r,0,2*sqrt(abs(cos(2*t)))),t,-%pi/4,%pi/4);
```

EJERCICIO 3: Calcular la integral de las siguientes funciones en los recintos determinados por las siguientes ecuaciones:
a) f(x,y)=(4x^2-y^2)^1/2, x=1, y=0, y=x.
b) f(x,y)=x*exp(-x^2/y), x=0, Y=1, y=x^2.
c) f(x,y)=y, y>=0, x^2+y^2=a^2, y^2=2ax, x=0.

EJERCICIO 4: Calcular el volumen limitado superiormente por el cono 4x^2+4y^2-z^2=0, inferiormente por el plano z=0 y lateralmente por el cilindro x^2+(y-2)^2=4.

3.2 Integrales triples

Análogamente a las integrales dobles, podemos calcular integrales triples. Veamos en el ejemplo siguiente como se puede obtener el volumen de un elipsoide (parecido a un balón de rugby) de semiejes a, b y c. Este ejercicio lo vamos a realizar mediante integrales triples (recordamos que el volumen, mediante integrales triples, se obtiene integrando dxdydz en el volumen que se quiere calcular:

```
--> /*EJEMPLO Nº 25: Calculamos el volumen de un elipsoide
    con semiejes a=4,b=2 y c=2*/
    /*Lo hacemos en el primer octante por comodidad de los
    límites de integración*/
    zmax:2*sqrt(1-x^2/16-y^2/4)$
    ymax:2*sqrt(1-x^2/16)$
    integrate(integrate(integrate(1,z,0,zmax),y,0,ymax),x,0,4);
```

Notemos como nos da un aviso sobre que el límite de integración ha de ser real. Por ello, pondremos

```
--> assume(1-x^2/16-y^2/4>0)$
    assume(1-x^2/16>0)$
    zmax:2*sqrt(1-x^2/16-y^2/4)$
    ymax:2*sqrt(1-x^2/16)$
    integrate(integrate(integrate(1,z,0,zmax),y,0,ymax),x,0,4);
```

Y como esto es solo el volumen en el 1er octante, habremos de multiplicar por 8 para obtener el volumen total:

```
--> %*8;
```

```
--> /*EJEMPLO Nº 26: Usando integración triple, calcular el volumen
    comprendido entre  x^2+y^2<=z^2 (cono) y x^2+y^2+z^2<= 4 (esfera)
    representando previamente la región*/
    /*Realizamos la representación gráfica del recinto*/
    remvalue(all)$
    load(draw)$
    draw3d([color=red,implicit(x^2+y^2+z^2=4,x,-5,5,y,-5,5,z,-5,5),
    color=blue,implicit(x^2+y^2=z^2,x,-5,5,y,-5,5,z,-5,5)]);
```

Para conocer el recinto de integración, debemos hallar los puntos donde se cortan la esfera y el cono. En este caso es trivial, ya que si sustituimos una ecuación en la otra, obtendremos como intersección la circunferencia x^2+y^2=2. Ésta será por tanto la proyección de dicha intersección sobre el plano XY. Por tanto, el volumen pedido será el doble (ya que la figura tiene intersección en los 8 octantes y nosotros vamos a calcular solo el de la parte superior) de la integral triple de dxdydz, donde x e y varian en la circunferencia anterior, mientras que z varia entre el cono y la esfera

```
--> 2*integrate(integrate(integrate(1,z,(x^2+y^2)^(1/2),
    (4-x^2+y^2)^(1/2)),
    y,-(2-x^2)^(1/2),(2-x^2)^(1/2)),x,-sqrt(2),sqrt(2));
```

Si ejecutamos la sentencia anterior, notamos que Maxima no nos da el resultado. Por tanto, haremos un cambio a coordenadas cilíndricas, siendo la integral a resolver la dada por (la función será r, que es el jacobiano, mientras que z varia entre el cono (que tiene por ecuación en cilíndricas: r) y la esfera (con ecuación en cilíndricas: (4-r^2)^(1/2)); r varía entre 0 y sqrt(2), mientras que theta varía entre 0 y 2pi):

```
--> 2*integrate(integrate(integrate(r,z,r,(4-r^2)^(1/2)),
    r,0,sqrt(2)),t,0,2*%pi);
```

Como también tenemos problemas, realizaremos nosotros aparte la primera de las integrales (resultará r*((4-r^2)^(1/2)-r) y ahora obtendremos la integral doble restante

```
--> 2*(integrate(r*((4-r^2)^(1/2)-r),r,0,sqrt(2)),t,0,2*%pi);
```