

TP 3 MATLAB - Derivacion

MASTER T.E.C.I.

1 Metodos de Taylor

1.1 Derivacion en dim 1 :

Nota : Se tiene que crear en un script a parte la funcion :

```
function J=func1(x)
J=sin(cos(x)*2*pi);
end
```

Nota : Se tiene que crear en un script a parte la funcion :

```
function J=func1d(x)
J=cos(cos(x)*2*pi)*2*pi*(-sin(x));
end
```

Nota : Se tiene que crear en un script a parte la funcion :

```
function J=func1ds(x)
J=2*pi*(sin(cos(x)*2*pi)*sin(x)*2*pi*(-sin(x))-(cos(x))*cos(cos(x)*2*pi))
end
```

Nota : Se tiene que crear en un script a parte :

```
clear all
close all
clc
x=0 :.1 :2*pi;
y=func1(x);

for i=1 :length(x)
yd(i)=func1d(x(i));
end

figure(1)
clf
```

```

hold on
plot(x,y,'r')
plot(x,yd,'b')
axis tight

disp('Calculo de la derivada primera mediante una aproximacion de taylor de un solo
punto')
derivt1=[];
eps=1e-6;

for i=1 :length(x)
yr=func1(x(i));
ym=func1(x(i)+eps);
derivt1(i)=(ym-yr)/eps;
end

figure(1)
plot(x,derivt1,':g')
errt1=norm(derivt1-yd,2);
disp(['Error Taylor 1 : ', num2str(errt1)])

disp('Calculo de la derivada primera mediante una aproximacion de taylor centrada')
derivt2=[];
eps=1e-6;

for i=1 :length(x)
yp=func1(x(i)+eps);
ym=func1(x(i)-eps);
derivt2(i)=(yp-ym)/(2*eps);
end

errt2=norm(derivt2-yd,2);
disp(['Error Taylor 2 : ', num2str(errt2)])

for i=1 :length(x)
yds(i)=func1ds(x(i));
end

figure(2)
clf
hold on
plot(x,y,'r')
plot(x,yds,'b')
axis tight

disp('Calculo de la derivada seguna mediante una aproximacion de taylor')
derivts=[];

```

```

eps=1e-6;

for i=1 :length(x)
yr=func1(x(i));
yp=func1(x(i)+eps);
ym=func1(x(i)-eps);
derivts(i)=(yp-2*yr+ym)/(eps)^2;
end

figure(2)
plot(x,derivts,'g')
errts=norm(derivts-yds,2);
disp(['Error Taylor 1 : ', num2str(errts)])

```

1.2 Derivacion en dim 2 :

Nota : Se tiene que crear en un script a parte la funcion :

```

function J=func2(x)
J=x(1)^2+sin(x(2));
end

```

Nota : Se tiene que crear en un script a parte la funcion :

```

function grad=func2g(x)
grad(1)=2*x(1);
grad(2)=cos(x(2));
end

```

Nota : Se tiene que crear en un script a parte la funcion :

```

function hess=func2h(x)
hess(1)=2;
hess(2)=0;
hess(3)=0;
hess(4)=-sin(x(2));
end

```

Nota : Se tiene que crear en un script a parte :

```

clear all
close all
clc
x=-2 :.5 :2;
y=0 :.5 :2*pi;
[xgrid,ygrid]=meshgrid(x,y);

```

```

for i=1 :length(x)
for j=1 :length(y)
z(j,i)=func2([xgrid(j,i),ygrid(j,i)]);
grex=func2g([xgrid(j,i),ygrid(j,i)]);
hsex=func2h([xgrid(j,i),ygrid(j,i)]);
gradx(j,i)=grex(1);
grady(j,i)=grex(2);
hessxx(j,i)=hsex(1);
hessxy(j,i)=hsex(2);
hessyx(j,i)=hsex(3);
hessyy(j,i)=hsex(4);
end
end

```

```

figure(1)
clf
hold on
surface(xgrid,ygrid,z)
contour(xgrid,ygrid,z)
view([45,45])
quiver(x,y,gradx,grady)
title('Gradiente exacto')

```

```

disp('Calculo del gradiente mediante una aproximacion de taylor centrada')
epsi=1e-6

```

```

for i=1 :length(x)
for j=1 :length(y)
zpx=func2([xgrid(j,i)+epsi,ygrid(j,i)]);
zmx=func2([xgrid(j,i)-epsi,ygrid(j,i)]);
gradtx(j,i)=(zpx-zmx)/(2*epsi);
zpy=func2([xgrid(j,i),ygrid(j,i)+epsi]);
zmy=func2([xgrid(j,i),ygrid(j,i)-epsi]);
gradty(j,i)=(zpy-zmy)/(2*epsi);
end
end

```

```

figure(2)
clf
hold on
surface(xgrid,ygrid,z)
contour(xgrid,ygrid,z)
view([45,45])
quiver(x,y,gradtx,gradty)
title('Gradiente aprox.')

```

```

disp(['Errores en el gradiente : '])

```

```

norm(gradx-gradtx,2)
norm(grady-gradty,2)

disp('Calculo de la hessiana mediante una aproximacion de taylor centrada')

for i=1 :length(x)
for j=1 :length(y)
zpr=func2([xgrid(j,i),ygrid(j,i)]);
zpx=func2([xgrid(j,i)+epsi,ygrid(j,i)]);
zmx=func2([xgrid(j,i)-epsi,ygrid(j,i)]);
hstxx(j,i)=(zpx+zmx-2*zpr)/(epsi)^2;
zpr=func2([xgrid(j,i),ygrid(j,i)]);
zpx=func2([xgrid(j,i),ygrid(j,i)+epsi]);
zmx=func2([xgrid(j,i),ygrid(j,i)-epsi]);
hstyy(j,i)=(zpx+zmx-2*zpr)/(epsi)^2;
zpx1=func2([xgrid(j,i)+epsi,ygrid(j,i)+epsi]);
zmx1=func2([xgrid(j,i)-epsi,ygrid(j,i)+epsi]);
zpx2=func2([xgrid(j,i)+epsi,ygrid(j,i)-epsi]);
zmx2=func2([xgrid(j,i)-epsi,ygrid(j,i)-epsi]);
hstyx(j,i)=(((zpx1-zmx1)/(2*epsi))-((zpx2-zmx2)/(2*epsi)))/(2*epsi);
hstxy(j,i)=(((zpx1-zpx2)/(2*epsi))-((zmx1-zmx2)/(2*epsi)))/(2*epsi);
end
end

disp(['Errores en la Hessiana : '])
norm(hessxx-hstxx,2)
norm(hessyy-hstyy,2)
norm(hessxy-hstxy,2)
norm(hessyx-hstyx,2)

```

2 Ejercicios

- Crear una funcion $mifunc(x, y, z) = \exp(-x^2) + \cos(y/10) - \sin(z)$.
- Crear una funcion $migradex(x, y, z)$ que devuelve el gradiente exacto de $mifunc$ en el punto (x, y, z) .
- Crear una funcion $migradtl(x, y, z)$ script que permite calcular el gradiente aproximado de $mifunc$ en el punto (x, y, z) usando una proximacion de Taylor centrada con una perturbacion de $\epsilon = 10^{-6}$.
- Crear un script que calcule el gradiente de $mifunc$ en los puntos $(0,0,0)$ y $(-2,1,3)$ usando $migradex$ y $migradtl$.