TP 2 MATLAB - INTERPOLACION

MASTER T.E.C.I.

1 Comandos Matlab para la interpolacion

1.1 Interpolacion en 1D

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

```
clear all
close all
clc
x=[-1 5 8 10];
y=[0 6 2 5];
k=1;

figure(k)
clf
plot(x,y,'o')
title('puntos de interpolacion')
k=k+1;

figure(k)
clf
hold on
plot(x,y,'o')
plot(x,y,'r')
title('puntos de interpolacion')

xint=-1 :.1 :10
yint = interp1(x,y,xint)
yint2 = interp1(x,y,xint,'lineal')
norm(yint-yint2,2)
k=k+1;

figure(k)
clf
plot(xint,yint,'o')
```

```
title('puntos de interpolados')

k=k+1
figure(k)
clf
hold on
plot(xint,yint,'go')
plot(xint,yint,'r')
plot(x,y,'s')
title('Interpolacion lineal')

yint3 = interp1(x,y,xint,'nearest')
k=k+1;
figure(k)
clf
hold on
plot(xint,yint3,'go')
plot(xint,yint3,'r')
plot(x,y,'s')
title('Interpolacion PWC')

yint3 = interp1(x,y,xint,'cubic')
k=k+1;
figure(k)
clf
hold on
plot(xint,yint3,'go')
plot(xint,yint3,'r')
plot(x,y,'s')
title('Interpolacion cubic (Hermite)')

yint3 = interp1(x,y,xint,'spline')
k=k+1;
figure(k)
clf
hold on
plot(xint,yint3,'go')
plot(xint,yint3,'r')
plot(x,y,'s')
title('Interpolacion cubic (Spline)')

yint3 = spline(x,y,xint)
k=k+1;
figure(k)
clf
hold on
plot(xint,yint3,'go')
```

```
plot(xint,yint3,'r')
plot(x,y,'s')
title('Interpolacion cubic (Spline - funcion spline)')

yint3 = pchip(x,y,xint)
k=k+1;
figure(k)
clf
hold on
plot(xint,yint3,'go')
plot(xint,yint3,'r')
plot(x,y,'s')
title('Interpolacion cubic (Hermit - funcion pchip)')
```

1.2 Interpolacion en 2D

1.2.1 Datos uniformes

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

```
clear all
close all
clc
x=-10 :5 :10;
y=-20 :5 :20;
[xx,yy]=meshgrid(y,x);

for ii=1 :length(x)
for jj=1 :length(y)
zz(ii,jj)=cos(x(ii))+sin(y(jj));
end
end

k=1;
figure(k)
clf
hold on
surface(xx,yy,zz)
title('Original data')
view([45 45])
axis('tight')
plot3(xx,yy,zz,'oy')

xi=-10 :.5 :10;
yi=-20 :.5 :20;
[xxi,yyi]=meshgrid(yi,xi);
```

```
zzi = interp2(xx,yy,zz,xxi,yyi,'nearest');
k=k+1;
figure(k)
clf
hold on
surface(xxi,yyi,zzi)
title('Nearest interpolation')
view([45 45])
axis('tight')
plot3(xx,yy,zz,'oy')

zzi = interp2(xx,yy,zz,xxi,yyi,'linear');
k=k+1;
figure(k)
clf
hold on
surface(xxi,yyi,zzi)
title('Linear interpolation')
view([45 45])
axis('tight')
plot3(xx,yy,zz,'oy')

zzi = interp2(xx,yy,zz,xxi,yyi,'cubic');
k=k+1;
figure(k)
clf
hold on
surface(xxi,yyi,zzi)
title('Hermit interpolation')
view([45 45])
axis('tight')
plot3(xx,yy,zz,'oy')

zzi = interp2(xx,yy,zz,xxi,yyi,'spline');
k=k+1;
figure(k)
clf
hold on
surface(xxi,yyi,zzi)
title('Spline interpolation')
view([45 45])
axis('tight')
plot3(xx,yy,zz,'oy')
```

1.2.2 Datos no uniformes

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

```
clear all
close all
clc
x=[1 5 15 -9 -10 8 7 -8] ;
y=[-5 3 8 -3 10 -10 5 2] ;
z=[1 5 3 7 10 4 8 9] ;
k=1 ;

figure(k)
clf
hold on
plot3(x,y,z,’ok’)
title(’Original data’)
view([45 45])
axis(’tight’)

xi=min(x) :((max(x)-min(x))/99) :max(x) ;
yi=min(y) :((max(y)-min(y))/99) :max(y) ;
[xxi,yyi]=meshgrid(yi,xi) ;

zzi = griddata(x,y,z,xxi,yyi,’nearest’) ;
k=k+1 ;
figure(k)
clf
hold on
surface(xxi,yyi,zzi,’Edgecolor’,’none’)
title(’Linear interpolation’)
view([45 45])
axis(’tight’)
plot3(x,y,z,’ok’)

zzi = griddata(x,y,z,xxi,yyi,’linear’) ;
k=k+1 ;
figure(k)
clf
hold on
surface(xxi,yyi,zzi,’Edgecolor’,’none’)
title(’Linear interpolation’)
view([45 45])
axis(’tight’)
plot3(x,y,z,’ok’)

zzi = griddata(x,y,z,xxi,yyi,’cubic’) ;
```

```
k=k+1;
figure(k)
clf
hold on
surface(xxi,yyi,zzi,'Edgecolor','none')
title('Cubic interpolation')
view([45 45])
axis('tight')
plot3(x,y,z,'ok')
```

1.3 Interpolacion en 3D

1.3.1 Datos uniformes

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

```
clear all
close all
clc
x=-10 :5 :10;
y=-20 :5 :20;
z=-10 :5 :5;
[xg,yg,zg] = meshgrid(y,x,z);
xi=-10 :1 :10;
yi=-20 :1 :20;
zi=-10 :1 :5;
[xgi,ygi,zgi] = meshgrid(yi,xi,zi);

for ii=1 :length(x)
for jj=1 :length(y)
for kk=1 :length(z)
dat(ii,jj,kk)=x(ii)∧2+y(jj)∧2+z(kk)∧2;
end
end
end

dati = interp3(xg,yg,zg,dat,xgi,ygi,zgi,'cubic');
figure(1)
hold on
slice(xg,yg,zg,dat,[-20,0],[0,10],[-10,0]), shading flat
view([45 45])
title('Original data')
figure(2)
hold on
slice(xgi,ygi,zgi,dati,[-20,0],[0,10],[-10,0]), shading flat
view([45 45])
title('Hermit interpolation')
```

1.3.2 Datos no uniformes

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

```
clear all
close all
clc
x= [-9 5 9 0 4 3 6 8];
y= [-1 2 -5 8 9 1 7 -8 ];
z= [1 0 5 9 3 -6 -5 -9];
dat=[0 1 2 5 -5 6 3 4];

xi=-10 :1 :10;
yi=-20 :1 :20;
zi=-10 :1 :5;
[xgi,ygi,zgi] = meshgrid(yi,xi,zi);

dati = griddata(x,y,z,dat,xgi,ygi,zgi,'linear');
figure(1)
hold on
slice(xgi,ygi,zgi,dati,[-20 :3 :20],[-10 :3 :10],[-10 :3 :5]), shading flat
plot3(x,y,z,'o')
view([45 45])
title('Linear interpolation')
```

1.3.3 Interpolacion en dimensiones superiores

1.3.4 Datos uniformes

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

```
clear all
close all
clc
x=-10 :10 :10;
y=-20 :10 :20;
z=-10 :10 :10;
k=-30 :10 :30;
[xg,yg,zg,kg] = ndgrid(x,y,z,k);
xi=-10 :5 :10;
yi=-20 :5 :20;
zi=-10 :5 :10;
ki=-30 :5 :30;
[xgi,ygi,zgi,kgi] = ndgrid(xi,yi,zi,ki);

for ii=1 :length(x)
for jj=1 :length(y)
for kk=1 :length(z)
```

```
for ll=1 :length(k)
dat(ii,jj,kk,ll)=x(ii)∧2+y(jj)∧2+z(kk)∧2+k(kk)∧2 ;
end
end
end
end

dati = interpn(xg,yg,zg,kg,dat,xgi,ygi,zgi,kgi,'cubic') ;
figure(1)
hold on
plot(x,dat( :,1,1,1),'b')
plot(xi,dati( :,1,1,1),'r')
title('Comparison of some data')
legend('Ori.','Interp.')
```

1.3.5 Datos no uniformes

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

```
clear all
close all
clc
x= [-9 5 9 0 4 3 6 8] ;
y= [-1 2 -5 8 9 1 7 -8 ] ;
z= [1 0 5 9 3 -6 -5 -9] ;
k= [9 8 5 -9 -3 7 -1 -10] ;
dat=[0 1 2 5 -5 6 3 4] ;
xi=-10 :1 :10 ;
yi=-20 :1 :20 ;
zi=-10 :1 :5 ;
ki=-10 :1 :10 ;
[xgi,ygi,zgi,kgi] = ndgrid(xi,yi,zi,ki) ;

XO = [x( :),y( :),z( :),k( :)] ;
XI = [xgi( :),ygi( :),zgi( :),kgi( :)] ;
dati = griddatan(XO,dat',XI,'linear') ;
datim=reshape(dati, size(ygi)) ;
```

2 Ejercicios

a) Programar una funcion $sol = minterp(x, y, t, met)$, donde x representa los puntos de mediciones, y los datos y t los puntos de inteprolacion, que devuelve en funcion de la cadena met :
   - Si met='linear' : sol son los datos obtenidos por interpolacion lineal (con la formula vista en clase) de (x, y) en los puntos de interpolacion t.

— Si *met*='lagrange' : *sol* son los datos obtenidos por interpolacion de polinomios de lagrange (con la formula vista en clase) de (x, y) en los puntos de interpolacion t.

b) Aplicar esta *minnterp* considerando los dos metodos ('linear' y 'lagrange') a los datos siguientes

x	-10	-3	0	2	5	10
y	-5	0	1	5	2	5

y dibujar las graficas de los datos interpolados.