

 UNIVERSIDAD DE OVIEDO DEPARTAMENTO DE MATEMÁTICAS	Asignatura	Métodos Numéricos	Página 1 de 5
	Tema	Interpolación y ajuste Multidimensional Superficies y volúmenes	
	Práctica	4.4	
	Autor	César Menéndez Fernández	

1 Superficies y volúmenes

En esta práctica se van a analizar diferentes tipos de interpolación en dimensión mayor que uno. Se limitará a la generalización de conceptos vistos en clase o similares, sin entrar en métodos más complejos.

1.1 Interpolación sobre una rejilla bidimensional

Para la interpolación de superficies, MATLAB ofrece la función **interp2**, generalización de la función **interp1** vista en la práctica 4.1.

interp2(x,y,z,xi,yi,'método')	Obtiene los valores correspondientes a los puntos (xi,yi) con el método seleccionado a partir de los datos (x,y,z). Los datos x, y, z deben ser matrices bidimensionales, los dos primeros corresponden a la malla generada en el plano XY y z son los valores de la función en dichos puntos. En todos los casos los métodos vienen definidos mediante funciones segmentarias. Los métodos posibles son: 'nearest': punto más cercano (grado 0, aproximación por planos constantes) 'linear': interpolación bilineal (grado 1 en x y en y; los valores se interpolan mediante la combinación de los cuatro puntos más cercanos) 'cubic': interpolación bicúbica (grado 3 en x y en y; mantiene la continuidad de la función y su derivada, suavizando la curva)
surf(x,y,z)	Crea una superficie tridimensional con los valores de x,y,z. También surfc y surfz
mesh(x,y,z)	Crea una malla tridimensional con los valores de x,y,z. También meshc y meshz

Todos los métodos exigen que X e Y sean monótonas, crecientes o decrecientes. Habitualmente el mallado en el plano XY se realiza mediante la instrucción **meshgrid**. Todas las opciones aplican previamente el método seleccionado para obtener datos igualmente espaciados en ambos ejes. Si ya lo están, se puede acelerar el proceso añadiendo un asterisco al nombre del método (p.e. '*linear').

A continuación realizamos un ejemplo para comparar los métodos bidimensionales sobre un mallado de 7x7 utilizando la función **peaks**.

```
% Generación de la malla no equiespaciada con los datos
t=[-3,-2,-0.5,0,0.75,1.75,3];
[x,y]=meshgrid(t); z=peaks(x,y);
% Generación de la malla fina para la representación
[xi,yi]=meshgrid(-3:0.25:3);
% Interpolación y comparación de resultados
zi1=interp2(x,y,z,xi,yi,'nearest');figure;surfc(xi,yi,zi1);title('cercano')
hold on;plot3(x,y,z,'b.','markersize',20);hold off
zi2=interp2(x,y,z,xi,yi,'linear');figure;surfc(xi,yi,zi2);title('bilineal');
hold on;plot3(x,y,z,'b.','markersize',20);hold off
zi3=interp2(x,y,z,xi,yi,'cubic');figure;surfc(xi,yi,zi3);title('bicubica')
hold on;plot3(x,y,z,'b.','markersize',20);hold off
figure(4);surfc(xi,yi,peaks(xi,yi));title('funcion')
hold on;plot3(x,y,z,'b.','markersize',20);hold off
```

La limitación de este tipo de interpolación es que exige conocer el valor de la función sobre una rejilla o cuadrícula, caso que no siempre es posible. Además no devuelve las funciones de interpolación, sino sólo valores en los puntos indicados.

Ejercicio recomendado:

1. Calcular la superficie de interpolación por los tres métodos anteriores tomando los datos generados por la función $f(x, y) = e^{-\left(\frac{x^2+y^2}{4}\right)} \cos(x^2 + y^2)$ sobre una malla de 9 puntos de lado, equiespaciados en $[-\pi, \pi]$. Comparar visualmente el resultado obtenido con la función original.
2. La tabla siguiente muestra valores de permeabilidad medidos sobre una cuadrícula. Calcular la superficie de interpolación por los tres métodos anteriores y obtener, de forma aproximada, el valor de la máxima permeabilidad.

y \ x	-50	-25	0	25	50
-50	2.2256	11.2256	14.6700	12.5589	4.8922
-25	5.8256	14.8256	18.2700	16.1589	8.4922
0	7.4256	16.4256	19.8700	17.7589	10.0922
25	7.0256	16.0256	19.4700	17.3589	9.6922
50	4.6256	13.6256	17.0700	14.9589	7.2922

1.2 Interpolación con puntos dispersos mediante triangulación

MATLAB dispone de rutinas de ayuda en la interpolación de puntos que no conforman una malla. Estas instrucciones permiten generar una superficie facetada a partir de datos dispersos. A continuación describimos las más importantes:

<code>convhull(x,y)</code>	Contorno convexo (área que contiene a todos los puntos definidos por sus coordenadas {x,y} y cuyo contorno es tal que la conexión mediante una recta de dos cualesquiera mantiene a la recta dentro del área).
<code>delaunay(x,y)</code>	Triangulación de Delaunay de los puntos definidos por sus coordenadas {x,y}
<code>delaunay3(x,y,z)</code>	“Triangulación” 3-D (tetraedros) de Delaunay de los puntos definidos por sus coordenadas {x,y,z}
<code>voronoi(x,y)</code>	Diagrama de Voronoi de los puntos definidos por sus coordenadas {x,y}
<code>dsearch(x,y,tri,xi,yi)</code>	Devuelve el índice de los puntos {x,y} más cercanos a {xi,yi}. Necesita utilizar la matriz tri devuelta por la triangulación de Delaunay
<code>rectint(A,B)</code>	Calcula el área de la intersección de los rectángulos definidos por los vectores A y B, que deben ser de la forma [x,y,dx,dy], indicando su inicio y su anchura y altura.
<code>inpolygon(x,y,xv,yv)</code>	Devuelve un vector del tamaño de x e y indicando si el punto es interior al definido por los vértices xv e yv
<code>polyarea(x,y)</code>	Calcula el área del polígono determinado por vértices cuyas coordenadas son x e y
<code>tsearch(x,y,tri,xi,yi)</code>	Devuelve un vector de índices del tamaño de xi e yi indicando el triángulo que lo contiene en la triangulación de Delaunay tri
<code>trimesh(tri,x,y,z)</code>	Utiliza la triangulación de Delaunay para representar los triángulos cuyos vértices vienen definidos por las coordenadas x,y,z
<code>tetramesh(tri,x)</code>	Utiliza la triangulación 3-D de Delaunay para representar los tetraedros definidos x
<code>trisurf(tri,x,y,z)</code>	Idem con una superficie
<code>trisurf(tri,x,y,z)</code>	Idem con una superficie
<code>griddata</code>	patch, tetramesh, trimesh, triplot,

trimesh Utiliza la triangulación de Delaunay para representar los triángulos cuyos vértices vienen definidos por las coordenadas x,y,z

griddata

Apliquémoslas sobre un ejemplo (ver la ayuda de MATLAB: Help → Matlab Help → Matlab → Mathematics → Polynomials and Interpolation → Interpolation → Triangulation and Interpolation of Scattered Data).

Se comienza generando n valores aleatorios en el intervalo [-3,3] para usarlos como soporte de la función utilizada para generar los datos del ejercicio 2 anterior y trazamos el contorno convexo:

La función de la que se extrajeron los datos fue $f(x, y) = 20 - \left(\frac{x-3}{15}\right)^2 - \left(\frac{y-7.5}{25}\right)^2$

```
% Generación de n puntos aleatorios en [-3,3]
N=20;rand('state', 0);T=rand(2,N)*6-3;
x=T(1,:);y=T(2,:);
plot(x,y,'.','markersize',10)
k = convhull(x,y);
hold on, plot(x(k),y(k),'-r'), hold off
grid on
```

Generamos la triangularización de Delaunay y la usamos como base de la superficie poliédrica que se genera mediante las instrucciones *trimesh* y *trisurf*:

```
plot(x,y,'.','markersize',10)
tri = delaunay(x,y);
hold on, triplot(tri,x,y), hold off

figure
hidden on
f=inline('20-(x/15-0.2).^2-(y/25-0.3).^2');
z=f(x,y);
trimesh(tri,x,y,z)
grid on
zlabel('Altura')
figure
hidden on
z=f(x,y);
trisurf(tri,x,y,z)
grid on
zlabel('Altura')
```

Generamos un mallado más fino y utilizamos *griddata* para interpolar valores intermedios utilizando alguno de los métodos vistos anteriormente.

```
figure
[xi,yi] = meshgrid(linspace(-3,3,51));
zi = griddata(x,y,z,xi,yi,'cubic');
[c,h] = contour(xi,yi,zi,'b-');
clabel(c,h)
xlabel('Longitude'), ylabel('Latitude')
```

También se puede obtener las zonas de proximidad a un determinado dato utilizando la instrucción *voronoi*.

```
voronoi(x,y)
grid on
xlabel('Longitude'), ylabel('Latitude')
```

1.3 Ajuste multidimensional

Si no se necesita que la función pase exactamente por esos puntos, se puede acudir a las técnicas de ajuste vistas en la práctica 4.2. Al igual que en el caso unidimensional, es necesario definir cuál va a ser la base de funciones. Consideremos que $B=\{1, x, y, x^2, xy, y^2\}$, las instrucciones de MATLAB para obtener la función de ajuste $z(x,y)=c_{00}1+c_{10}x+c_{01}y+c_{20}x^2+c_{11}xy+c_{02}y^2$, utilizando los valores x e y del ejemplo y representarla son

```
X=ones(length(x),6);
X(:,2)=x';X(:,3)=y';X(:,4)=x'.^2; X(:,5)=x'.y';X(:,6)=y'.^2;
C=X\z';
[xi,yi] = meshgrid(linspace(-3,3,51));
zi = ones(length(xi),6);
zi=C(1)+C(2)*xi+C(3)*yi+C(4)*xi.^2+C(5)*xi.*yi+C(6)*yi.^2;
[c,h] = contour(xi,yi,zi,'b-');
clabel(c,h)
xlabel('Longitude'), ylabel('Latitude')
```

1.4 Interpolación con puntos dispersos mediante normales de Gauss

Cuando el número de datos es muy escaso, se utilizan técnicas de kriging. Una de las más conocidas es la combinación de distribuciones normales en cada punto (campanas de Gauss). Consideremos que se tienen los siguientes datos extraídos del ejercicio 2; su número es tan escaso que incluso la triangulación mediante Dalaunay no nos puede dar una función continua razonable.

Y	-50	-25	0	25
Y	25	-50	0	50
Z	7.0256	11.2256	19.8700	14.9589

En este caso se plantea

$$P(x,y) = \alpha_1 e^{-\frac{(x+50)^2+(y-25)^2}{\sigma_1^2}} + \alpha_2 e^{-\frac{(x+25)^2+(y-50)^2}{\sigma_2^2}} + \alpha_3 e^{-\frac{(x-0)^2+(y-0)^2}{\sigma_3^2}} + \alpha_4 e^{-\frac{(x-25)^2+(y-50)^2}{\sigma_4^2}}$$

$$P(x_k, y_k) = f(x_k, y_k) \quad k=1,2,3,4$$

Suponiendo conocidas σ_k , queda un sistema de 4 ecuaciones con 4 incógnitas cuya resolución nos permite obtener la función de interpolación. La elección de los valores de σ_k se realiza bien por conocimientos previos, o bien para que la superficie resultante de una continuidad razonable. La evaluación e interpolación se puede hacer mediante la función *itp2dgaus* adjunta

```
function [zi,wi]=itp2dgaus(x,y,z,xi,yi,sigma)
%
% Interpola en dos dimensiones utilizando campanas de gauss
% function [zi,wi]=itp2dgaus(x,y,z,xi,yi,sigma)
%
% ENTRADA:
% x : Coordenadas x de los datos
% y : Coordenadas y de los datos
% z : Coordenadas z de los datos
% xi : Coordenadas x de los puntos a interpolar (usando meshgrid)
% yi : Coordenadas y de los puntos a interpolar (usando meshgrid)
% sigma: Desviación típica de las campanas de Gauss (tantas como datos)
%
% SALIDA
% zi : Coordenadas z de los puntos a interpolar
% wi : Pesos de las campanas generadas
```

