`Laboratorio Didattico di Matematica Computazionale`

`a.a. 2018-2019`

ESERCITAZIONE 5

Grafici in 3D

Attenzione: in questa esercitazione, gli esercizi da consegnare sono solo quelli della sezione 5.

1. Superfici e curve implicite

Impariamo innanzi tutto l'uso del comando `meshgrid`. Se `x` e `y` sono due vettori, il comando

```
[X,Y]=meshgrid(x,y)
```

crea una matrice `X` le cui righe sono tutte uguali a `x`, e una matrice `Y` le cui colonne sono tutte uguali a `y`.

Provate a dare i comandi seguenti e osservate come sono fatte `X` e `Y`:

```
x=1:5;
y=6:11;
[X,Y]=meshgrid(x,y)
```

Ora, supponiamo di avere una funzione reale $f(x, y)$ definita su un rettangolo $[a, b] \times [c, d] \subset \mathbb{R}^2$. Vogliamo scegliere una griglia sul rettangolo e valutare f sui punti della griglia, per darne poi una rappresentazione grafica. Definiamo quindi il vettore `x` come una discretizzazione dell'intervallo $[a, b]$ e il vettore `y` come una discretizzazione dell'intervallo $[c, d]$, quindi diamo l'istruzione `[X,Y]=meshgrid(x,y)`. Il generico punto di indici i, j sulla griglia definita dalle due discretizzazioni avrà quindi coordinate `X(i,j), Y(i,j)`. Ora possiamo valutare f sulla griglia senza usare cicli.

Per esempio, consideriamo la funzione $f(x, y) = x^2+y^2$ definita sul rettangolo $[-0.5, 0.5]\times[-0.5, 0.5]$. Per valutarla nei punti di una griglia definita sul rettangolo possiamo scrivere

```
x=-0.5:0.02:0.5;
y=x;
[X,Y]=meshgrid(x,y)
A=X.^2+Y.^2;
```

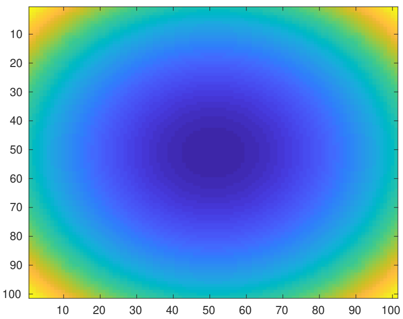
La matrice `A` contiene i valori cercati di $f(x, y)$. Possiamo visualizzarli in due dimensioni con il comando

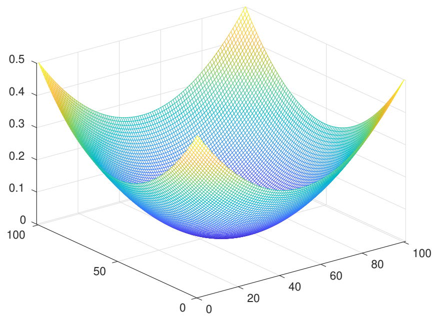
```
imagesc(A)
```

oppure in $\mathbb{R}^3$ con il comando

```
mesh(A)
```

Il risultato è riportato qui sotto.





Nel caso 3D potete anche ruotare la figura per vedere meglio com'è fatta la superficie.

Provate a usare il comando `mesh(x,y,A)` oppure `mesh(X,Y,A)` invece di `mesh(A)`. Che cosa cambia?

Che cosa succede se si usa `surf(A)` al posto di `mesh(A)`?

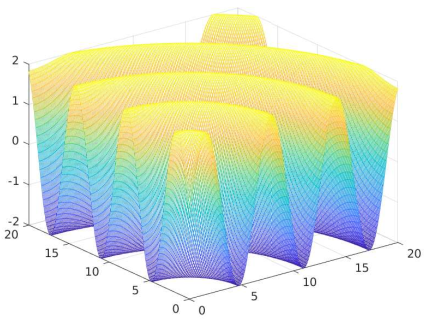
Esercizio 1 *Usando le funzioni* `meshgrid` *e* `mesh`*, disegnare il grafico delle funzioni seguenti:*

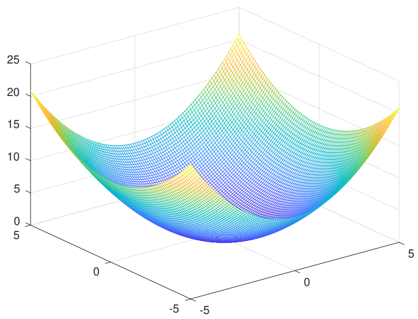
1. $f(x,y) = 2\sin((x^2+y^2)^{1/2})$ *con* $(x,y) \in [0,20] \times [0,20]$*,*
2. $f(x,y) = x^2/2 + y^2/3)$ *con* $(x,y) \in [-a,a] \times [-b,b]$ *e* $a,b > 0$ *scelti a piacere,*
3. $f(x,y) = e^{-x^2-y^2}$ *con* $(x,y) \in [-a,a] \times [-b,b]$ *e* $a,b > 0$ *scelti a piacere,*
4. $f(x,y) = xe^{-(x-y^2)^2-y^2}$ *con* $(x,y) \in [-a,a] \times [-b,b]$ *e* $a,b > 0$ *scelti a piacere.*

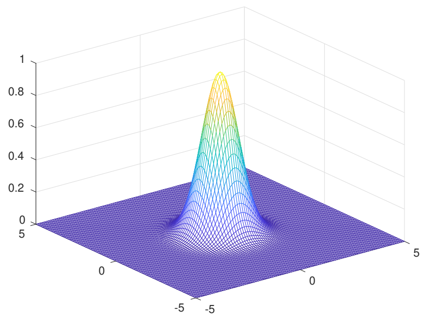
Suggerimento: potete definire la funzione $f(x,y)$ *in modo anonimo e poi applicarla alle matrici prodotte da* `meshgrid`*. Per esempio, per la prima funzione si può scrivere*

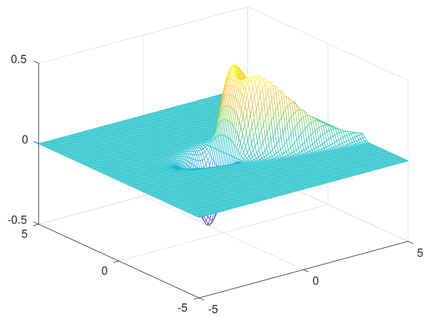
```
f=@(x,y) 2*sin((x.^2+y.^2).^(1/2));
x=0:0.1:20;
y=x;
[X,Y]=meshgrid(x,y);
meshgrid(x,y,f(X,Y))
```

Dovreste ottenere delle figure simili alle seguenti:





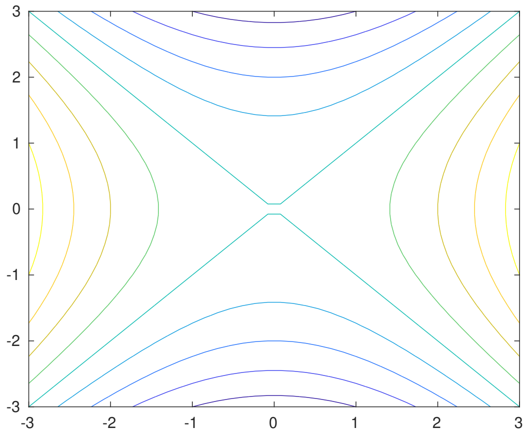




Il comando `contour` permette di disegnare le curve di livello. Per esempio, consideriamo la funzione $f(x,y) = x^2 - y^2$ definita su $[-3,3] \times [-3,3]$ e diamo i comandi

```
f=@(x,y) x.^2-y.^2;
x=linspace(-3,3:40);
y=x;
contour(X,Y,f(X,Y))
```

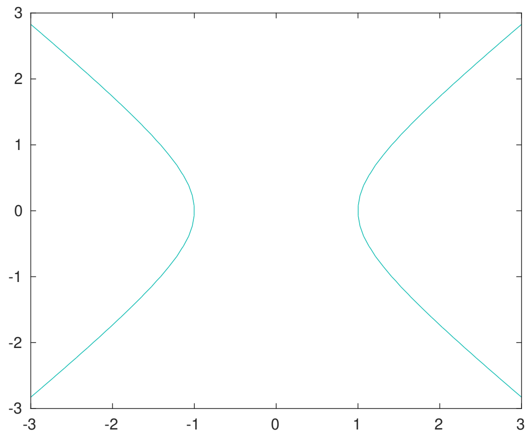
Otteniamo la figura seguente, nella quale i livelli sono assegnati automaticamente:



È anche possibile scegliere i livelli delle curve da disegnare. In particolare, possiamo usare `contour` per disegnare curve di cui è data l'equazione implicita. I comandi

```
c=1; % livello della curva
v=[c;c];
contour(X,Y,f(X,Y),v)
```

permettono di disegnare la curva di equazione $x^2 - y^2 = 1$, cioè la curva di livello 1 della $f(x,y)$ definita sopra:



Esercizio 2 *Usando il comando* `contour`*, disegnare la curva di equazione* $x^2 + 2xy + y^2 - 2x = 3$ *con* $(x, y) \in [-3, 5] \times [-5, 3]$.

2. Curve parametriche

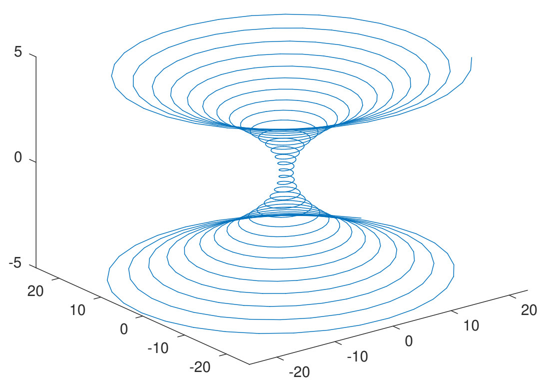
Una curva parametrica in $\mathbb{R}^3$ è definita da tre equazioni che descrivono la dipendenza di ciascuna coordinata x, y, z dal parametro t. Consideriamo la curva di equazioni

$$x(t) = (1 + t^2)\cos(20t),$$
$$y(t) = (1 + t^2)\sin(20t),$$
$$z(t) = t,$$

con $t \in [-5, 5]$. Per disegnarla possiamo usare il comando `plot3` come segue:

```
t=-5:0.01:5;
x=(1+t.^2).*cos(20*t);
y=(1+t.^2).*sin(20*t);
z=t;
plot3(x,y,z)
```

e otteniamo il grafico



Esercizio 3 *Disegnare un'elica con raggio e passo costante.*

Esercizio 4 *Disegnare le curve definite dalle equazioni parametriche seguenti:*

(a)

$$
\begin{aligned}
x(t) &= (2+\cos(1.5t))\cos(t),\\
y(t) &= (2+\cos(1.5t))\sin(t),\\
z(t) &= 2\sin(1.5t), \qquad t \in [0, 4\pi]
\end{aligned}
$$

(b)

$$
\begin{aligned}
x(t) &= (4+\sin(20t))\cos(t),\\
y(t) &= (4+\sin(20t))\sin(t),\\
z(t) &= cos(20t), \qquad t \in [0, 2\pi]
\end{aligned}
$$

(c)

$$
\begin{aligned}
x(t) &= t,\\
y(t) &= t^2,\\
z(t) &= t^3, \qquad t \in [-2, 2].
\end{aligned}
$$

3. Superfici parametriche

Una superficie parametrica in $\mathbb{R}^3$ è definita da tre equazioni che descrivono la dipendenza di ciascuna coordinata x, y, z da due parametri r, t.

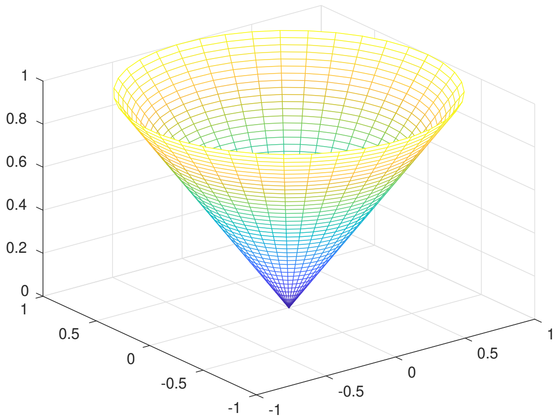
Vogliamo disegnare la superficie in $\mathbb{R}^3$ definita dalle equazioni parametriche

$$
\begin{aligned}
x(r,t) &= r\cos(t),\\
y(r,t) &= r\sin(t),\\
z(r,t) &= r,
\end{aligned}
$$

con $r \in [0,1]$ e $t \in [0, 2\pi]$. Per farlo usiamo i comandi:

```
r=linspace(0,1,40);
t=linspace(0,2*pi,40);
[R,T]=meshgrid(r,t);
x=R.*cos(T);
y=R.*sin(T);
z=R;
mesh(x,y,z)
```

Come potete osservare, otteniamo un cono (perché?).



Provate a usare il comando `surf` invece di `mesh`, seguito da `shading interp` e `axis off`.

Esercizio 5 *Disegnare una sfera. Sarà utile dare il comando* `axis equal` *per avere le stesse proporzioni sui tre assi coordinati.*

Esercizio 6 *Disegnare la superficie definita dalle equazioni parametriche:*

$$\begin{aligned} x(u,v) &= 2(1-e^{\frac{u}{6\pi}})\cos(u)\cos^2(v/2), \\ y(u,v) &= 2(-1+e^{\frac{u}{6\pi}})\sin(u)\cos^2(v/2), \\ z(u,v) &= 1-e^{\frac{u}{3\pi}}-\sin(v)+e^{\frac{u}{6\pi}}\sin(v), \end{aligned}$$

dove $u \in [0, 6\pi]$*,* $v \in [0, 2\pi]$*. Ottenete una conchiglia?*

Esercizio 7 *Ecco una lista di superfici famose. Cercate sul web le loro equazioni parametriche e provate a disegnarle.*

- *bottiglia di Klein,*
- *superficie di Enneper,*
- *ombrello di Whitney,*
- *superficie di Steiner.*

Disegnate anche degli esempi di superfici di Lissajous*, che hanno equazioni*

$$\begin{aligned} x(u,v) &= sin(u), \\ y(u,v) &= sin(v), \\ z(u,v) &= sin(d-au-bv)/c, \end{aligned}$$

dove $u \in [-\pi, \pi]$*,* $v \in [-\pi, \pi]$ *e* a*,* b*,* c*,* d *sono parametri reali da scegliere a piacere (provate all'inizio con* $a = b = c = 1$*,* $d = 0$ *e poi sperimentate con altri valori).*

4. Animazioni e filmati (facoltativo)

In Matlab è possibile creare delle animazioni a partire da grafici in 2D o 3D. Il codice che segue è una modifica di un esempio presente nell'help di Matlab.

```
% setup
figure
Z = peaks;
surf(Z)
axis tight manual
ax = gca;
ax.NextPlot = 'replaceChildren';
% creo l'animazione
loops = 120;
F(loops) = struct('cdata',[],'colormap',[]);
for j = 1:loops
     X = sin(j*pi/30)*Z;
     surf(X,Z)
     drawnow
     F(j) = getframe;
end
% visualizzo l'animazione tre volte
movie(F,3)
```

Provate ad eseguirlo e cercate di capire come funziona. Ricordiamo che `peaks` è un comando che calcola una superficie predefinita dotata di "picchi" e "avvallamenti". L'animazione viene creata nel ciclo `for`: ad ogni passaggio si disegna un grafico con il comando `surf` e lo si acquisisce usando `getframe`. A questo punto i dati per l'animazione, cioè le varie *frame*, sono immagazzinati in `F`. Il comando `movie` esegue l'animazione un certo numero di volte.

Alcune idee per modificare il codice:

- Modificate la colormap: potete sceglierne una da usare per tutta l'animazione, o anche cambiarla ad ogni *frame*. Potete scegliere una colormap predefinita oppure crearne una voi (controllate l'help).
- Rallentate o accelerate l'animazione.
- Sostituite `peaks` con un'altra superficie o curva a vostro piacimento, che farete evolvere nel tempo.

5. Esercizi

Esercizio 8 *Disegnare il grafico della funzione*

$$f(x,y) = \left| \frac{1}{1-(x+iy)^{10}} \right|,$$

dove i è l'unità immaginaria e $(x,y) \in [-2,2] \times [-2,2]$. Si presume naturalmente che le singolarità della funzione non si trovino sulla griglia di discretizzazione del dominio.

Esercizio 9 *Disegnare la curva nota come* folium di Cartesio*, che ha equazione*

$$x^3 + y^3 = 3xy,$$

con $(x,y) \in [-3,3] \times [-3,3]$. Attenzione: la curva ha un punto di autointersezione. Se nel vostro disegno la curva non si autointerseca ma ha due componenti connesse, raffinate la discretizzazione.

Esercizio 10 *Disegnare la curva in* $\mathbb{R}^3$ *di equazioni parametriche*

$$\begin{aligned} x(t) &= \sin(t), \\ y(t) &= \sin(2t), \\ z(t) &= \sin(3t), \end{aligned}$$

con $t \in [0, 2\pi]$.

Facoltativo: creare un'animazione in cui un punto si muove lungo la curva.

Esercizio 11 *Disegnare la striscia di Möbius, che è descritta dalle equazioni parametriche seguenti:*

$$\begin{aligned} x(u,v) &= \cos(u) + v\cos(u/2)\cos(u), \\ y(u,v) &= \sin(u) + v\cos(u/2)\sin(u), \\ z(u,v) &= v\sin(u/2), \end{aligned}$$

con $(u, v) \in [0, 2\pi] \times [-0.4, 0.4]$.