Tutoraggio #8

Federico Pichi

10 giugno 2019

Esercizio 1

Si considerino le equazioni di Lotka-Volterra per le due popolazioni batteriche y_1 e y_2

$$\frac{dy_1}{dt} = C_1 y_1 (1 - b_1 y_1 - d_2 y_2) \ ,$$
$$\frac{dy_2}{dt} = -C_2 y_2 (1 - b_2 y_2 - d_1 y_1) \ ,$$

dove C_1 e C_2 sono i fattori di crescita (positivi) delle due popolazioni, i coefficienti d_1 e d_2 governano il tipo di interazione, mentre b_1 e b_2 sono legati alla disponibilità dei nutrienti. Si risolva il sistema, utilizzando *ode45*, con $C_1 = C_2 = 1$, $b_1 = b_2 = 0$, $d_1 = d_2 = 1$, $t_0 = 0$, $t_f = 10$, $y_1^0 = 2$, $y_2^0 = 2$, $h = 5 \cdot 10^{-4}$ e $n_{max} = 20000$. Si ripeta l'esercizio ponendo $y_1^0 = 1.5$, $y_2^0 = 1.5$ e commentare i risultati.

Esercizio 2

Si consideri una matrice $A \in \mathbb{R}^{2m \times 2m}$ corrispondente ad un'immagine in due dimensioni tale che valga k, $1 \leq k \leq m$ su ogni elemento della riga k-esima e $(2m$-k+$1)$-esima. Utilizzando il comando *rand* si modifichino i suoi valori nell'intervallo (-1, 1) e la si visualizzi utilizzando il comando *imagesc*. Si rielabori la matrice ottenuta tramite una media dei valori in una data finestra centrata nel generico pixel.

Esercizio 3

Implementare in Matlab un biliardo ellittico, verificando che partendo da uno dei due fuochi, la pallina va sempre a ricadere nel fuoco opposto.

Soluzione 1

Vogliamo risolvere il sistema di Lotka-Volterra sull'intervallo temporale $[0, 10]$ con un passo di integrazione $h = 5 \cdot 10^{-4}$. Il primo plot rappresenta l'evoluzione nel tempo delle due componenti della soluzione. Come si vede, esse mostrano un andamento periodico. Il secondo grafico mostra invece la traiettoria uscente dal dato iniziale nel cosiddetto piano delle fasi, cioè in un piano cartesiano che ha come coordinate y_1 e y_2. Appare evidente che la traiettoria si mantiene in una regione limitata di questo piano. Se provassimo a partire dal dato iniziale $[1.2, 1.2]$ troveremmo una traiettoria ancor più confinata che sembra mantenersi chiusa attorno al punto $[1, 1]$. Ciò è dovuto al fatto che il sistema ammette 2 punti di equilibrio (ovvero punti nei quali $y_1' = 0$ e $y_2' = 0$) e uno di essi è proprio $[1, 1]$ (mentre l'altro è $[0, 0]$).

Codice 1: Soluzione 1

```
t0f=[0 10];
h=5.e-4;
u0=[2;2];
u0n=[1.5;1.5];
fun= @(t,y) [y(1).*(1-y(2));-y(2).*(1-y(1))];
[t,u]=ode45(fun,t0f,u0);
[tn,un]=ode45(fun,t0f,u0n);

plot(t,u(:,1),tn,un(:,1),tn,un(:,2),t,u(:,2))
legend('y1 con dato iniziale [2;2]','y2 con dato iniziale [2;2]', 'y1 con dato iniziale [1.5;1.5]','y2 con dato iniziale [1.5;1.5]','Location','NorthEastOutside')
title('Lotka-Volterra con ode45')

figure
plot(u(:,1),u(:,2),un(:,1),un(:,2))
legend('Dato iniziale [2;2]','Dato iniziale [1.5;1.5]','Location','NorthEastOutside')
title('Piano delle fasi')
```

Soluzione 2

In questo codice riduciamo la perturbazione di B rispetto ad A. Un modo valido se la matrice presenta una certa regolarita', cioe' per cui i valori in elementi vicini variano 'poco' (rispetto alla grandezza della perturbazione) tra loro e' quelo di operare una media spaziale locale. Dunque potremmo costruire una versione C meno distorta di B tale che : C(i,j)=1/5*(B(i,j)+B(i+1,j)+B(i-1,j)+B(i,j+1)+B(i,j-1)). Si capisce subito pero' che abbiamo una difficolta' quando ci troviamo ai bordi. Procediamo quindi differenziandoli in tre gruppi: quelli in posizione (1,1), (1,2m), (2m,2m), (2m,1); quelli lungo il bordo che non appartengono al primo gruppo, e quelli interni. Nel primo gruppo ogni elemento avra' solo altri due elementi 'vicini', nel secondo tre e nel terzo quattro. Per calcolare i valori di C non dobbiamo far altro che sommare un elemento con tutti quelli adiacenti e dividere per il numero di termini della somma.

Codice 2: Soluzione 2

```

m=16;
A = zeros(2*m);

for k = 1:m
    A(k, :) = k;
    A(m+k, :) = m-k+1;
end


imagesc(A)
axis equal
axis off
colormap gray
print -dpdf plot1.pdf

B=A+2*(rand(2*m,2*m)-.5);
figure
imagesc(B)
axis equal
axis off
colormap gray
print -dpdf plot2.pdf


C(1,1)=(B(1,1)+B(1,2)+B(2,1))/3;
C(1,2*m)=(B(1,2*m-1)+B(1,2*m)+B(2,2*m))/3;
C(2*m,2*m)=(B(2*m,2*m)+B(2*m,2*m-1)+B(2*m-1,2*m))/3;
C(2*m,1)=(B(2*m,1)+B(2*m,2)+B(2*m-1,1))/3;

for s=2:2*m-1
    C(1,s)=(B(1,s)+B(1,s-1)+B(1,s+1)+B(2,s))/4;

    C(2*m,s)=(B(2*m,s)+B(2*m,s-1)+B(2*m,s+1)+B(2*m-1,s))/4;

    C(s,1)=(B(s,1)+B(s-1,1)+B(s+1,1)+B(s,2))/4;

    C(s,2*m)=(B(s,2*m)+B(s-1,2*m)+B(s+1,2*m)+B(s,2*m-1))/4;
end

for s=2:2*m-1
    for t=2:2*m-1
        C(s,t)=(B(s,t)+B(s,t+1)+B(s,t-1)+B(s+1,t)+B(s-1,t))/5;
    end
end

figure
imagesc(C)
axis equal
axis off
colormap gray
print -dpdf plot3.pdf
```

```
sqrt(sum(sum((B-A).*(B-A))))
sqrt(sum(sum((C-A).*(C-A))))
```

Soluzione 3

Codice 3: Soluzione 3

```
a=5;
b=4;

x=linspace(-a,a,100);
y1=b*sqrt(1-x.^2/(a)^2);
y2=-b*sqrt(1-x.^2/(a)^2);
plot(x,y1,'k')
hold on
plot(x,y2,'k')

c=sqrt(a^2-b^2);
F1=[-c,0];
F2=[c,0];
plot([-c c],[0 0],'*')

teta=pi*(3/4);
dt=0.1;
xold=[c,0];

while 2>1

xa=xold+[cos(teta)*dt, sin(teta)*dt];
if (xa(1).^2)/a^2+(xa(2).^2)/b^2 < 1
    xnew=xa;
    plot([xold(1),xnew(1)],[xold(2),xnew(2)],'r')
    xold=xnew;
else
    if xold(2)>=0
        segno=1;
    else
        segno=-1;
    end

    for i=1:10
        xn=(xa+xold)/2;
        if (xn(1).^2)/a^2+(xn(2).^2)/b^2 < 1
            xold=xn;
        else
            xa=xn;
        end
    end
    xnew=xn;
    plot([xold(1),xnew(1)],[xold(2),xnew(2)],'r')
    xold=xnew;

    tan1=-segno*((b/a^2)*xnew(1))/sqrt(1-((xnew(1)^2)/a^2));
    tnor=-1/tan1;
    teta=2*atan(tnor)-teta;
    dt=-dt;
end
pause
end
```