Prove d'esame
Esercizi con Matlab

Andrea Corli*

16 settembre 2015

Sono qui raccolti alcuni esercizi relativi a Matlab assegnati nelle prove d'esame (dal 2011 al 2014) del Corso di Analisi Matematica I (semestrale, 12 crediti), Laurea in Ingegneria Civile e Ambientale, tenuto da me presso l'Università degli Studi di Ferrara. Ho aggiunto anche alcuni esercizi di livello analogo. Ad ogni testo segue una delle possibili soluzioni.

1 Esercizi d'esame

- **10/11/2011 - Prima prova parziale** Usare un ciclo `for` per calcolare e quindi rappresentare graficamente gli elementi della successione definita da $a_1 = 1$, $a_n = \frac{1}{1+a_{n-1}}$ per $n = 2, 3, \ldots, 10$.

```
a(1) = 1;
for n = 2:10
    a(n) = 1/(1+a(n-1));
end
n=1:10
plot(n,a,'o')
```

  Si noti che Matlab avvisa che il vettore `a` cambia di dimensione ad ogni loop e questo può rallentare l'esecuzione (in effetti, questo non accade per calcoli semplici come quello che stiamo eseguendo). Si può ovviare prescrivendo sin dall'inizio la dimensione di `a`:

```
a = ones(1,10);
a(1) = 1;
for n = 2:10
    a(n) = 1/(1+a(n-1));
end
n=1:10;
plot(n,a,'o')
```

- **9/1/2012 - Seconda prova parziale** Usare un ciclo `for` per calcolare simbolicamente gli integrali $\int_n^{+\infty} \frac{1}{x^{3/2}}\, dx$, per $n = 1, 2, 3$, e creare un vettore contenente i tre risultati.

```
syms x
for n = 1:3
    I(n) = int(1/x^(3/2),x,n,Inf);
end
disp(I)
```

  Si noti che Matlab avvisa, in corrispondenza di `I(n)`, una perdita di velocità dovuta al fatto che `I(n)` cambia dimensione in ogni loop. Si può ovviare a questo con

*Dipartimento di Matematica, Università di Ferrara

```
syms x
I=ones(3,1);           % dimensiona I;
I=sym(I);              % trasforma I in simbolico;
for n = 1:3
    I(n) = int(1/x^(3/2),x,n,Inf);
end
disp(I)
```

- **24/1/2012** Si consideri la funzione $f(x) = 3x^4 + x^3 - 6x^2 - 3x + 1$. Scrivere un breve script che ne calcola uno zero e che disegna il grafico di f e della sua derivata numerica in uno stesso grafico.

  Con `roots`:

```
p = [3 1 -6 3 1];
roots(p)
x = linspace(-2,2); %linspace scelto in conseguenza delle radici
f = polyval(p,x);
plot(x,f)
hold on
plot(x(1:99),diff(f)./diff(x),'r')
grid on
```

  Con `fzero`:

```
x = linspace(-10,10);
f = 3*x.^4 + x.^3 - 6*x.^2 - 3*x +1;
fzero(@(x) 3*x.^4+x.^3-6*x.^2-3*x+1,0) %0: non sappiamo dov'e'!
```

  Con una `function` esterna:

```
%polinomio.m
function f = polinomio(x)
p = [3 1 -6 -3 1];
f = polyval(p,x);
```

  poi

```
x = linspace(-10,10);
p = polinomio(x);
fzero('polinomio',0)
```

- **15/11/2012 - Prima prova parziale** Scrivere un breve script per disegnare il grafico della funzione $\operatorname{tg} x$ nell'intervallo $[-\pi, \pi]$, evitando di far apparire righe verticali. Restringere l'asse delle ordinate all'intervallo $[-5, 5]$.

```
x = -pi:2*pi/100:pi;
x(1)                          %il primo
x(101)                        %l'ultimo
x(26) = NaN;
x(76) = NaN;
plot(x,tan(x))
axis([-pi pi -5 5])
```

- **8/1/2013 - Seconda prova parziale** Sia $f(x) = \sin(x)$ in $[0, 2\pi]$. Fissato $n = 10$, scrivere un breve script che calcola e disegna il grafico di f, $f \circ f$, $\dots f \circ f \circ \dots \circ f$, dove la composizione è fatta n volte.

```
x = 0:0.01:2*pi;
f = sin(x);
plot(x,f)
hold on
```

```
for i=1:9
    f = sin(f);
    plot(x,f)
    hold on
end
xlabel('x')
ylabel('f')
axis([0 2*pi -1.5 1.5])
grid on
```

- **22/1/2013** Scrivere un breve script che calcoli le prime tre derivate (simboliche) della funzione $f(x) = e^{-x^2}$ e ne disegni i grafici in tre figure diverse.

```
syms x
f = exp(-x^2);
figure(1)
ezplot(f)
hold on
for i=1:2
   f(i) = diff(f,i);
   figure(i+1)
   ezplot(f(i))
end
```

- **12/2/2013** Scrivere un breve script in Matlab che calcoli i valori di $\int_0^n e^{-x^2}\,dx$ per $n = 1, 2, \ldots, 10$. Usare il comando `num2str` per rappresentare il risultato in maniera facilmente leggibile a command window.

```
for i=1:10
   x=linspace(0,i);
   I = trapz(x,exp(-x.^2));
   disp(num2str(I))
end
```

- **18/06/2013** Rappresentare i grafici delle funzioni f_n, definite nell'intervallo $[0, 2]$ da $f_n(x) = n^x$, per $n = 1, 2, \ldots 10$, in un opportuno grafico a scala logaritmica in cui i grafici risultino rette. Ripetere lo stesso esercizio con $g_n(x) = x^n$ in $[1, 10]$.

```
x=linspace(0,2);
n=1:10;
[X,N]=meshgrid(x,n);
semilogy(x,N.^X)
```

oppure

```
x = linspace(0,2);
for n=1:10
    semilogy(x,n.^x);
    hold on
end
```

Inoltre:

```
x=linspace(1,10);
n=1:10;
[X,N]=meshgrid(x,n);
loglog(x,X.^N)
```

oppure

```
x = linspace(1,10);
for n=1:10
    loglog(x,x.^n);
    hold on
end
```

Si noti il diverso uso del colore.

- **3/9/2013** Si consideri la funzione f definita da $f(x) = x+1$ se $x < 1$ e $f(x) = 4x^2+1$ se $x \geq 1$. Scrivere un breve script in Matlab che disegni il grafico di f nell'intervallo $[-2, 2]$, evitando segmenti verticali nei punti di discontinuità.

```
x = -2:0.01:2;
k = find(x==1);
x(k) = NaN;
f = x.*(x < 1) + 4.*x.^2.*( x >= 1) + 1;
plot(x,f)
grid on
```

Lo stesso esercizio poteva essere risolto col ciclo `for`.

- **19/9/2013** Utilizzando un ciclo `for`, scrivere uno script che disegni i grafici delle funzioni $(x-n)^2$, $n = 1, \ldots, 5$, nell'intervallo $[0, 10]$.

```
x = linspace(0,10);
for i=1:5
    plot(x,(x-i).^2)
    hold on
end
```

Naturalmente grafici analoghi potevano essere prodotti col comando `meshgrid`.

- **7/11/2013 - Prima prova parziale** Sia $\{a_n\}$ la successione definita da $a_1 = \log 10$ e $a_n = \log(1 + a_{n-1})$ per $n \geq 2$. Calcolare e disegnare i primi dieci termini della successione.

```
n=[1:10];
a(1)=log(10);
for i=2:10
    a(i)=log(1+a(i-1));
end
plot(n,a,'o')
```

- **9/1/2014 - Seconda prova parziale** Scrivere uno script che calcola il più piccolo numero N tale che $\sum_{k=1}^{N} \frac{1}{\sqrt{k}} > 10$.

```
s=0;
k=0;
while s < 10
    k = k+1;
    s = s + 1/sqrt(k);
end
disp(num2str(k))
```

- **21/1/2014** Si consideri la successione $a_n = \sin n$ per $n = 1, \ldots, 10$. Si costruisca la successione $\{b_n\}$ tale che $b_n = a_n$ se $a_n > 1/10$ e $b_n = 0$ se $a_n \leq 10$.

```
n = 1:10;
a = sin(n);
for i = 1:length(n)
    if a(i) > 1/10
      b(i) = a(i);
    else
```

```
        b(i) = 0;
    end
end
disp(b)
```

oppure, più semplicemente,

```
n = 1:10;
a = sin(n);
b = a.* (a > 1/10)
```

- **11/2/2014** Scrivere uno script che calcola il primo numero intero n tale che $n! > 10^6$.

```
n=1
while cumprod(n) < 10000
    n=n+1
end
disp(num2str(n))
```

- **9/6/2014** Scrivere un breve script che disegni il grafico di e^{-x} e della sua derivata (numerica).

```
n=100;
x=linspace(0,2,n);
f=exp(-x);
Df=diff(f)./diff(x);
plot(x(1:n-1),f(1:n-1),x(1:n-1),Df(1:n-1))
```

- **14/7/2014** Scrivere un breve script in Matlab che calcoli ascissa e ordinata del punto di intersezione dei grafici delle funzioni e^{-x} e $\sin x$ in $[0, \pi/2]$, tracci i grafici delle due funzioni e metta in evidenza il punto di intersezione.

```
x=linspace(0,pi/2);
z=fzero('exp(-x) - sin(x)', 1);
disp([z,exp(-z)])
plot(x,exp(-x),x,sin(x),z,exp(-z),'or')
```

- **8/9/2014** Definire tramite una `function` la funzione $f(x) = \log(1+x) - \frac{1}{1+x}$; quindi richiamarla in un file per disegnarne il grafico in $[0, 2]$ e calcolarne uno zero.

  Si scrive prima il file `fz.m` contenente la `function`

```
function y = fz(x)
y = log(1+x) - 1./(1+x);
```

che viene richiamata nel file seguente:

```
x=linspace(0,10);
plot(x,fz(x))
fzero('fz',1)
```

2 Altri esercizi

1. Sia $f : [0, 5] \to \mathbb{R}$ la funzione definita da $f(x) = x$ se $x \in [0, 1)$, $f(x) = x^2$ se $x \in [1, 2)$, $f(x) = 4$ se $x \in [2, 5]$. Plottare un grafico di f usando un ciclo `for` e `if`.

```
x = 0:0.01:5;
f = zeros(length(x));          %f ha la stessa lunghezza di x; non serve ma
                               %vedere cosa succede se si commenta questa riga
for n=1:length(x);
    if x(n)<=1;                %sono le componenti di x! x<=1 non ha senso
        f(n)=x(n);
```

```
    elseif 1<x(n) && x(n)<=2;
        f(n)=(x(n)).^2;
    else
        f(n)=4;
    end
end
plot(x,f)
grid on
axis([0 5 0 5])               %per vedere meglio la linea orizzontale a 5
```

Con gli operatori relazionali lo stesso esercizio si risolveva così:

```
x = 0:0.01:5;
f = x.*(x<=1) + x.^2.*(x>1 & x<=2) + 4*(x>2);
plot(x,f)
grid on
axis([0 5 0 5])
```

2. Utilizzando un ciclo `for`, scrivere uno script che disegni i grafici delle funzioni $f_n(x) = \frac{nx}{1+n^2x^2}$, $n = 1, \ldots, 5$, nell'intervallo $[0, 3]$.

```
x = linspace(0,3);
for n=1:5
    plot(x,n*x./(1+n^2*x.^2))
    hold on
end
axis([0 3 0 1])
grid on
```

Con `meshgrid` lo stesso esercizio si risolveva così:

```
x = linspace(0,3);
n = 1:5;
[X,N] = meshgrid(x,n);
plot(x,N.*X ./ (1 + N.^2.*X.^2))
axis([0 3 0 1])
grid on
```

Si noti l'uso diverso del colore di default.

3. Scrivere uno script che disegna i grafici delle funzioni $f_n(x) = \frac{nx}{1+nx^2}$, $n = 1, \ldots, 5$, nell'intervallo $[0, 3]$ e ne calcola i massimi e i punti di massimo.

```
x = linspace(0,3);
n = 1:5;                          %per allocare le dimensioni
f = zeros(size(x));               %"
M = zeros(size(n));               %"
X = zeros(size(n));               %"
for n=1:5
    plot(x,n*x./(1+n*x.^2))
    hold on
    M(n) = max(n*x./(1+n*x.^2));
    X(n) = fminbnd(@(x) -n*x./(1+n*x.^2),0,3);
end
grid on
disp(['valori massimi: ',num2str(M)]);
disp(['punti di massimo: ',num2str(X)]);
```