

Corso Matlab: Test Finale 30.09.2010

TRACCIA SOLUZIONE

1. Creare una matrice NxM di numeri casuali compresi tra 1 e 10:

$(10 - 1) * \text{rand}(N, M) + 1$

**2. scrivere una funzione matlab che prende come ingresso:
valore minimo v e massimo V, il numero di righe N ed il numero di colonne
M.**

**Ritorna una matrice di numeri compresi tra v e V, quadrata NxN se N<M,
MxM altrimenti.**

```
function y = randNM(min, max, N, M)

if (N<M)
    y = (max - min)*rand(N, N) + min;
else
    y = (max - min)*rand(M, M) + min;
end
```

3. Risolvere il sistema lineare

Ax=b

dove

**A = randNM(parametri a piacere);
b = rand(parametri a piacere);**

```
x = inv(A)*b;
```

4. Risolvere il sistema non lineare

$x^2 + y^2 = 18$
 $x^2 + 3xy = 36$
 $x^3 - 2y = 21$

condizioni iniziali: $x(0) = y(0) = z(0) = 0$

```
function F = myfun(x)
F = [x(1)^2 + x(2)^2 - 18;
     x(1)^2 + 3*x(1)*x(2) - 36;
     x(1)^3 - 2*x(2) - 21];

x0 = [1; 1];           % Make a starting guess at the solution
options=optimset('Display','iter'); % Option to display output
[x,fval] = fsolve(@myfun,x0,options) % Call optimizer

x = [3 3]
```

5. Simulare il seguente sistema dinamico

$x1(k+1) = x1(k) + x2(k+1) \pmod{1}$
 $x2(k+1) = x2(k) + 3/(2\pi) * \sin(2\pi x1(k)) \pmod{1}$

cond. iniziali: random tra 0 e 1

Disegnare 1000 punti successivi raggiunti durante l'evoluzione del sistema.

```
x = rand(2,1);

figure
hold on
plot(x(1), x(2), 'r.')
```

```

for k = 1 : 1 : 1000
    x(2) = mod(x(2) + 3 / (2*pi) * sin(2*pi*x(1)), 1);
    x(1) = mod(x(1) + x(2), 1);
    plot(x(1), x(2), 'r.')
end

```

6. Trovare il minimo della funzione pippo utilizzando un qualsiasi metodo di ottimizzazione:

```

function y = pippo(x)
y = 1./((x-.3).^2 + .01) + 1./((x-.9).^2 + .04) - 6;

```

Valutare questa funzione per un set di valori nell'intervallo seguente: $x = 0:.002:1$;

```
y = pippo(x);
```

Tracciare il grafico della funzione

```
plot(x,y)
```

Il grafico mostra che la funzione ha un minimo locale vicino a $x = 0.6$.

La funzione `fminsearch` trova il minimo valore di x . Il primo argomento di `fminsearch` è il nome della funzione di cui deve essere trovato il minimo, il secondo argomento è una supposizione grezza dell'ubicazione del minimo.

```
minimo = fminsearch(@pippo, 0.5)
```

```
minimo = 0.6370
```

Valutare la funzione in corrispondenza del minimo:

```
pippo(p)
ans =
11.2528
```

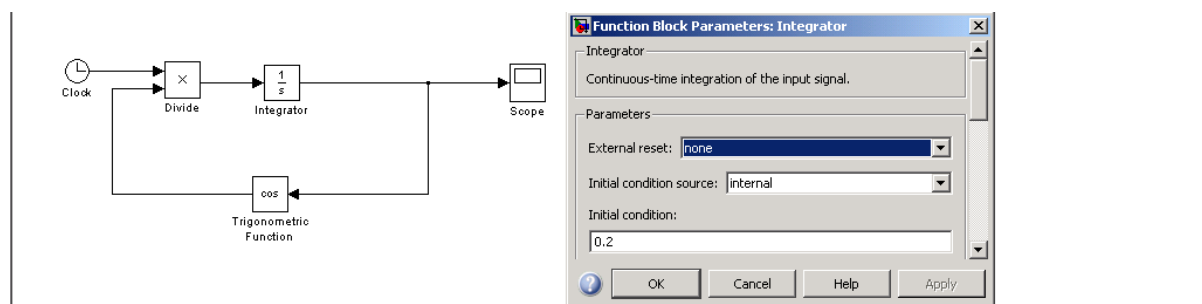
Plottare sul grafico della funzione pippo il valore del minimo trovato.

```
plot(y, pippo(y), 'r-', fminsearch(@(x) pippo(x), 0.5), pippo(fminsearch(@(x)
pippo(x), 0.5)),'bo')
```

7. Analizzare in Simulink il seguente sistema tempo variante

$$\mathbf{x'(t) = t \cdot \cos(x(t)); x(0) = 0.2}$$

Suggerimento: utilizzare il blocco "Clock"



Impostare le condizioni iniziali sull'integratore (doppio click sul blocchetto integratore)

8. Esportare l'uscita del sistema al punto 7 sul workspace di matlab e disegnarla

Usare il blocco "To Workspace" e collegarlo al segnale che arriva allo scope nel diagramma precedente.

9. Analizzare in Simulink l'evoluzione del sistema descritto dalla seguente equazione

$$y'' + (1/t)*y' + (2/(t^2))*y = u$$

per $t \geq 1$

condizioni iniziali: $y(1) = 1$

$y'(0) = 1$

ingresso: $u = \cos(t)$

Suggerimento: utilizzare il blocco "Clock"

Procedere come nel caso dell'esercizio svolto al punto 7.

10. Importare l'ingresso $\cos(t)$ dal workspace di matlab e disegnare l'andamento della variabile y'

Suggerimento: Usare il blocco "From Workspace"