Metodi Numerici con Laboratorio di Informatica - A.A. 2015-2016
Esercizi Laboratorio n° 2

Esercizio 1.
Definire le seguenti funzioni

- $f(x) = x\sin(x) + \left(\frac{1}{2}\right)^{\sqrt{x}}$
- $g(x) = x^4 + \log(x^3 + 1)$

e valutarle sul vettore `x=[0 1 2 3]`.

Esercizio 2.
Disegnare il grafico della funzione

$$f(x) = 2 + (x - 3)\sin(5(x - 3))$$

per $0 \leq x \leq 6$. Sovrapporre a questo grafico quelli delle due rette che limitano l'andamento di tale funzione, disegnate con linea tratteggiata.
Suggerimento Le due rette in questione sono $y = -x + 5$ e $y = x - 1$.

Esercizio 3. **Errori di cancellazione**

Tracciare il grafico della funzione $f(x) = (1 - x)^6$ e il grafico di

$$g(x) = x^6 - 6\,x^5 + 15\,x^4 - 20\,x^3 + 15\,x^2 - 6\,x + 1$$

dapprima sull'intervallo $[0, 2]$ e poi sull'intervallo $[0.995,\ 1.005]$. Sebbene $f = g$ i loro grafici mostrano comportamenti differenti, come si può commentare questo fatto?

Esercizio 4. **Errori di *roundoff* e *overflow***
Si consideri il seguente codice MATLAB

```
format long
f(1)=1;
f(2)=1;
i=2;
while f(i)< ???
    f(i+1) = f(i) + f(i-1);
    i=i+1;
```

```
end
n=i-1;
g(n)=f(n);
g(n-1)=f(n-1);
for i=(n-1):-1:2
    g(i-1)=g(i+1) - g(i);
end
```

Al variare della condizione nel ciclo `while` il programma calcola i primi n numeri di Fibonacci e successivamente calcola la stessa sequenza in ordine inverso.

1. Come bisogna completare il criterio d'arresto del ciclo `while` affinché n sia il più grande indice per cui tutti i numeri di Fibonacci calcolati siano rappresentabili *esattamente* come numeri a doppia precisione di MATLAB, dunque senza essere affetti da errore di *roundoff*?
2. Qual è l'indice del più grande numero di Fibonacci che può essere *approssimato* come numero a doppia precisione MATLAB senza *overflow*?
3. Per valori di n compresi tra i due valori trovati, `f(i)` e `g(i)` sono molto diversi. Si dia una spiegazione del perché ciò accada.

Si ricorda che MATLAB usa lo standard IEEE a doppia precisione.

Polinomi in MATLAB e fenomeno di Runge

In MATLAB un generico polinomio di grado N

$$p(x) = a_0 + a_1\, x + a_2\, x^2 + ... + a_N\, x^N$$

è definito tramite un vettore contenente gli $N+1$ coefficienti a_i, $i = 0, ...N+1$, ordinati a partire dal coefficiente relativo al grado maggiore fino al termine di grado zero, esplicitando eventuali coefficienti nulli.
Ad esempio al polinomio $p(x) = 3 - 7\,x + 4\,x^3$ si associa il vettore `p=[4 0 -7 3]`.
`polyval`
La funzione `polyval` valuta il valore di un polinomio p in una griglia di punti. La sua sintassi è

`y=polyval(cp,x),`

dove `cp` è un vettore contenente i coefficienti del polinomio p, supponiamo di grado N, dunque contenente $N+1$ elementi;

`x` è un vettore contenente i nodi su cui vogliamo valutare p;

`y` è il vettore contenente gli elementi $y_i = p(x_i)$, con $i = 1, ...,$`length(x)`.

Esempio `polyval`. Valutare il polinomio $p(x) = 3 - 7\,x + 4\,x^3$ su una griglia di 11 nodi equispaziati da 0 a 1.

```
x=linspace(0,1,11); %in alternativa x=[0:0.1:1];
cp=[4 0 -7 3];
y=polyval(cp,x);
```

`polyfit`

Dati $N+1$ punti (x_i, y_i) con $i = 1...N+1$, la funzione `polyfit` restituisce il vettore dei coefficienti del polinomio di grado N che interpola i punti assegnati.

`p=polyfit(x,y,N),`

dove `x` è un vettore contenente $N+1$ nodi;

`y` è un vettore contenente i valori che la funzione assume negli $N+1$ nodi; `N` è il grado del polinomio interpolatore; `p` è il vettore contenente i coefficienti del polinomio interpolatore.

Esempio `polyfit`. Trovare il polinomio interpolatore p di grado 3 che interpola la funzione $f(x) = \sin(3x)$ nei nodi $x = \{0, \pi/6, \pi/3, \pi/2\}$ e tracciare il grafico sia di f che di p.

```
f=@(x)sin(3*x);
x=[0 pi/6 pi/3 pi/2];
y=f(x);
p=polyfit(x,y,3); %trovo coefficienti del polinomio interpolatore
%% Per disegnare i grafici
xx=linspace(-0.5,2, 100);
yy=polyval(p,xx);
plot(x,y,'*');
hold on
plot(xx,f(xx))
```

```
plot(xx, yy)
legend('punti di interpolazione', 'f(x)', 'p(x)')
```

Esercizio 5. **Fenomeno di Runge**

Il seguente codice calcola il polinomio p_N di grado N, scelto dall'utente, che interpola la funzione di Runge $f(x) = \dfrac{1}{1 + 25\,x^2}$ in $N + 1$ nodi equispaziati sull'intervallo $[-1, 1]$ e traccia il grafico di f, p_N e dei punti di interpolazione.

```
f=@(x)1./(1+25*x.^2);
N=input('Grado polinomio interpolatore: ');
nodi=linspace(-1,1,N+1); %nodi devono essere N+1
y_n=f(nodi);
p=polyfit(nodi,y_n, N); %p= vettore contenente i coeff. del polinomio interp di grado N
% per disegnare raffino griglia
xx=linspace(-1,1,100);
yy=f(xx);
yp=polyval(p,xx);
figure(1)
plot(xx,yy)
hold on
plot(xx,yp, '--')
plot(nodi,y_n, '*')
```

1. Confrontare il grafico di ciascun polinomio interpolatore con quello della funzione data facendo variare il grado N da 2 a 12.

2. Il fenomeno di Runge può essere evitato utilizzando opportune distribuzioni di nodi. Si modifichi considerando i nodi di Chebyshev

$$x_i = \cos(\frac{(i-1)*\pi}{N}) \qquad i = 1, ..., N+1,$$

e si confronti il grafico di ciascun polinomio interpolatore con quello della funzione data facendo variare il grado N da 2 a 12.