Laboratorio Calcolo Numerico

Esercizio 1 Si vuole determinare un polinomio interpolatore della funzione

$$f(x) = \frac{1}{1+x^2}$$

nell'intervallo $I = [-5, 5]$. Considerata la forma di Newton del polinomio interpolante, si costruisca una tabulazione della funzione data costituita da 11 nodi equidistanti nell'intervallo I, e si memorizzino i nodi in una vettore `xn` ed i corrispondenti valori della funzione nel vettore `fxn`.

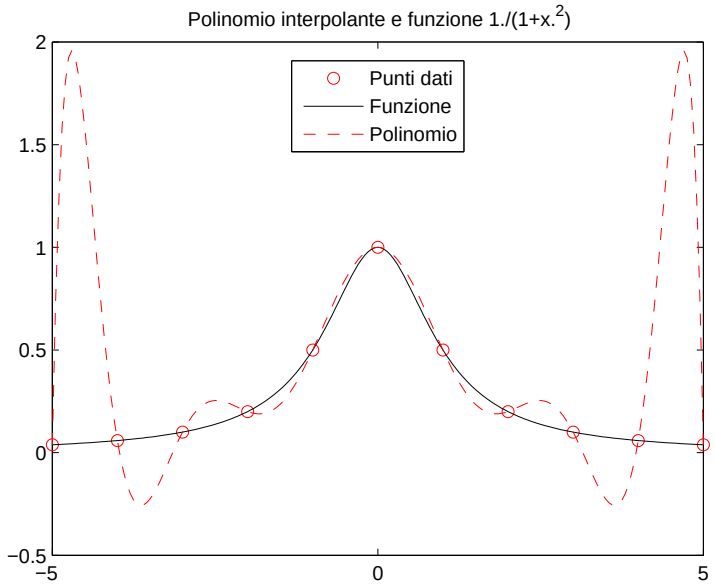
Si scriva una funzione di nome `polnewton` che abbia la seguente intestazione e che calcoli i coefficienti della base di Newton con le differenze divise.

```
function c = polnewton (x,y)
% POLNEWTON Calcola i coefficienti del polinomio interpolatore
%           utilizzando la forma di Newton con le differenze
%           divise
%
% Uso:
%   c = polnewton (x,y)
%
% Dati di ingresso:
%   x    vettore dei nodi
%   y    vettore dei valori della funzione da interpolare nei nodi
%
% Dati di uscita:
%   c    vettore colonna dei coefficienti ordinati per indici
%        crescenti (c_0, c_1, ... )
```

Si definisca poi una function di nome `hornerN` che, dati i coefficienti c_i ed i nodi, permetta di valutare un polinomio interpolatore $P(x)$ in un valore x^*. La function avrà la seguente intestazione

```
function fxstar = hornerN (x,c,xstar)
% HORNERN Calcola il valore del polinomio interpolatore in xˆ*
%         utilizzando la forma di Newton e l'algoritmo di Horner
%
% Uso:
%   fxstar = hornerN (x,c,xstar)
%
% Dati di ingresso:
%   x      vettore dei nodi
%   c      vettore dei coefficienti della forma di Newton
%          ordinati per indici crescenti (c_0, c_1, ... )
%   xstar  valore in cui si vuole valutare il polinomio
%
% Dati di uscita:
%   fxstar valore di P(xˆ*)
```

Si scriva poi uno script che calcoli il valore del polinomio di interpolazione in 201 valori dell'intervallo I, per poterlo rappresentare graficamente (linea tratteggiata rossa).
Sullo stesso grafico si rappresentino anche i punti della tabulazione (con un cerchietto rosso) e la funzione data (linea intera nera).
Il risultato dovrà essere quello della seguente figura:



Traccia per risolvere l'esercizio 1

Per scrivere la funzione `polnewton` si deve prendere come base l'algoritmo di pag. 292. L'algoritmo va attentamente letto anche nella descrizione iniziale per capire il significato delle differenti variabili. Poi tale algoritmo va modificato per rispondere alle richieste del testo dell'esercizio (in uscita si vuole, ad esempio, **solamente** la prima riga dello schema e non tutta la tabella) ed in ingresso non viene dato il numero dei nodi (componenti dei vettori) ed anche per adattarsi alla sintassi Matlab che utilizza solo indici di riga e di colonna che iniziano da 1 (e non da zero).

Si noti che nell'algoritmo originale si usa, per lo schema triangolare, una matrice indicata con la variabile C ed all'interno con la notazione $c_{i,j}$ si rappresenta l'elemento di riga i e colonna j di tale matrice. Della matrice, essendo uno schema triangolare, si usa solo una parte triangolare.

Per non confondere la matrice C con il vettore `c` richiesto nelle specifiche della function `polnewton`, chiamiamo nell'algoritmo tale matrice *ctab*. Inoltre la variabile locale interna $n1$ contiene il numero di nodi $n + 1$. Ovviamente anche i valori assunti dalle variabili nei cicli ad indice fisso dovranno essere modificati.

A questo punto, gli studenti, dovrebbero scrivere prima l'algoritmo **modificato** come segue, e poi tradurlo in una function Matlab di nome `polnewton`

$$[\mathbf{c}] = \textbf{Diff\_divise}\ (\mathbf{x}, \mathbf{y})$$

$n1 =$ lunghezza vettore **x**
if $n1 \neq$ lunghezza vettore **y**
    Errore. La tabulazione non contiene lo stesso numero di componenti. Stop.
end if
for $i = 1, \ldots, n1$

$ctab(i,1) = y(i)$
end for i
for $j = 2, \ldots, n1$
for $i = 1, \ldots, n1 - j + 1$
$ctab(i,j) = (ctab(i+1, j-1) - ctab(i, j-1))/(x(i+j-1) - x(i))$
end for i
end for j
c = vettore colonna contenente la prima riga di *ctab*

Per tradurre in Matlab la prima istruzione si veda il comando Matlab `length` e per l'ultima, si usi il comando `c = ctab(1,:)` che estrae la riga 1 di una matrice.

Vediamo ora come impostare il secondo algoritmo per poterlo tradurre nella function `hornerN` le cui specifiche sono

L'algoritmo si trova a pag. 271 del libro di testo. Come in precedenza si deve prima modificare il seguente algoritmo originale per adattarlo alle esigenze del Matlab per quanto riguarda gli indici delle componenti dei vettori.

Algoritmo originale:

$$[u] = \textbf{hornerN}\ (x, c, x^*)$$

$u = c_n$
for $j = n-1, \ldots, 0$ **step** -1
$u = u \cdot (x^* - x_j) + c_j$
end for j

dove

$d_j = (x^* - x_j)$ ed $u = P_n(x^*) = c_0 + c_1 d_0 + c_2 d_0 d_1 + \cdots + c_n d_0 d_1 \cdots d_{n-1} = (\cdots(((c_n)d_{n-1} + c_{n-1})d_{n-2} + c_{n-2})d_{n-3} + \cdots + c_1)d_0 + c_0$.

Ecco l'algoritmo modificato:

$$[u] = \textbf{hornerN}\ (\mathbf{x}, \mathbf{c}, xstar)$$

$n1 =$ lunghezza vettore **x**
if $n1 \neq$ lunghezza vettore **c**
Errore. Inconsistente numero di componenti. Stop.
end if
$u = \mathbf{c}(n1)$
for $j = n1 - 1, \ldots, 1$ **step** -1
$u = u * (xstar - \mathbf{x}(j)) + \mathbf{c}(j)$
end for j

A questo punto basta tradurlo in Matlab, rispettando i nomi indicati per i parametri di ingresso e di uscita.

Vediamo uno schema per lo script (ovviamente mancano le istruzioni Matlab che dovranno essere inserite dagli studenti!

script `interp`

```
%====================================================================
% Script per interpolazione con la forma di Newton
% Necessita delle function polnewton e hornerN
%====================================================================

% Definisce la funzione
% ..............................
% Definisce l'intervallo [a,b]
% ..............................
% Definisce il numero di nodi
% ..............................
% Definisce la tabulazione dei punti (ascisse equispaziate ed ordinate
% come valori della funzione nei nodi definiti)
% ..............................
% Calcola i coefficienti del polinomio con la base di Newton usando
% la function polnewton
% ..............................

% Grafici
% Definisce le ascisse per i grafici (201 punti equispaziati)
% ..............................
% Valuta la funzione nelle 201 ascisse
% ..............................
% Valuta il polinomio nelle 201 ascisse (ciclo for che usa
% la function hornerN)
% ..............................

% Creazione della figura che contiene
%   i punti della tabulazione
%   la rappresentazione della funzione originale
%   la rappresentazione del polinomio interpolante
% ..............................
```

Esercizio 2

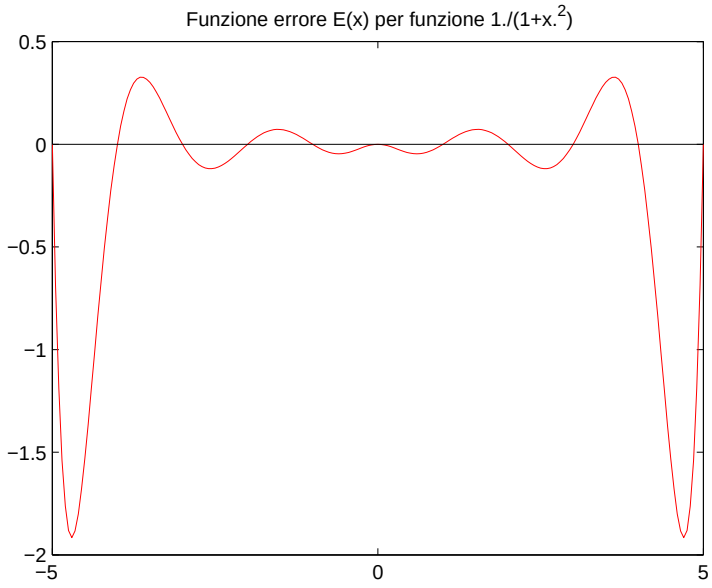
Si vari (aumentando e diminuendo) il numero di punti di interpolazione (e quindi il grado del polinomio interpolante) e si cerchi di capire cosa accade.

Esercizio 3

Si produca anche una figura che rappresenti nello stesso intervallo l'asse x e la funzione errore definita come

$$E(x) = f(x) - P(x).$$

ottenendo la seguente figura



Esercizio 4 (facoltativo)

Risultati analoghi a quelli degli esercizi precedenti si possono ottenere con due funzioni Matlab:

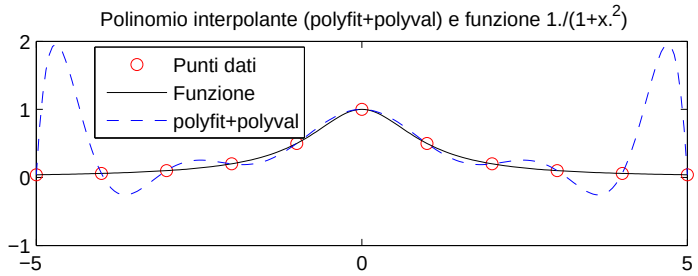
```
POLYFIT Fit polynomial to data.
   P = POLYFIT(X,Y,N) finds the coefficients of a polynomial P(X) of
   degree N that fits the data Y best in a least-squares sense.
   P is a row vector of length N+1 containing the polynomial coefficients in
   descending powers, P(1)*X^N + P(2)*X^(N-1) +...+ P(N)*X + P(N+1).
```

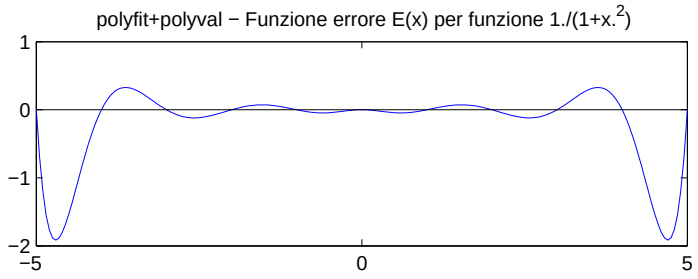
che permette di determinare un vettore che contiene i coefficienti del polinomio interpolante rispetto alla base canonica;

```
POLYVAL Evaluate polynomial.
   Y = POLYVAL(P,X) returns the value of a polynomial P evaluated at X. P
   is a vector of length N+1 whose elements are the coefficients of the
   polynomial in descending powers.
       Y = P(1)*X^N + P(2)*X^(N-1) + ... + P(N)*X + P(N+1)
```

che valuta un polinomio di cui si conoscono i coefficienti rispetto alla base canonica su tutti i valori precedentemente memorizzati su di un vettore.

Si producano le stesse figure precedentemente richieste (che risulteranno praticamente uguali) e poi si cerchi di strutturarle con il comando `subplot` come segue


Polinomio interpolante (polyfit+polyval) e funzione 1./(1+x.^2)
Punti dati
Funzione
polyfit+polyval
2
1
0
−1
−5
0
5


polyfit+polyval − Funzione errore E(x) per funzione 1./(1+x.^2)
1
0
−1
−2
−5
0
5