Laboratorio Calcolo Numerico

Esercizio 1 Si desidera valutare un polinomio

$$P(x) = 3 + x + 4x^2 + 5x^3 + 7x^4 + 2x^5$$

in una discretizzazione dell'intervallo $I = [-3, 1]$, con l'algoritmo di Horner.

Sia dato un polinomio P_n di grado al più n ($P_n \in \mathcal{P}_n$) scritto nella forma

$$P_n(x) = \alpha_n x^n + \alpha_{n-1}x^{n-1} + \cdots + \alpha_1 x + \alpha_0,$$

ed un punto $\bar{x}$. Si vuole valutare $P_n(\bar{x})$ ovvero il valore del polinomio nel punto $\bar{x}$.
Possiamo applicare l'espressione precedente ovvero calcolare

$$P_n(\bar{x}) = \alpha_0 + \sum_{i=1}^{n} \alpha_i \bar{x}^i$$

effettuando $j - 1$ moltiplicazioni per ogni potenza x^j ($j = 1, \ldots, n$), n moltiplicazioni per i prodotti relativi ai coefficienti ed infine n addizioni, ovvero, in totale, $n^2 + n/2$ moltiplicazioni ed n addizioni.
In alternativa possiamo utilizzare il seguente schema iterativo (detto algoritmo di Horner)

Poniamo $u = P_n(\bar{x})$. Si ha

$$u = \alpha_0 + \alpha_1\bar{x} + \alpha_2\bar{x}\bar{x} + \cdots + \alpha_n\bar{x}\bar{x}\cdots\bar{x},$$

che può anche essere scritto nella forma

$$u = (\cdots((\alpha_n\bar{x} + \alpha_{n-1})\bar{x} + \alpha_{n-2})\bar{x} + \cdots + \alpha_1)\bar{x} + \alpha_0.$$

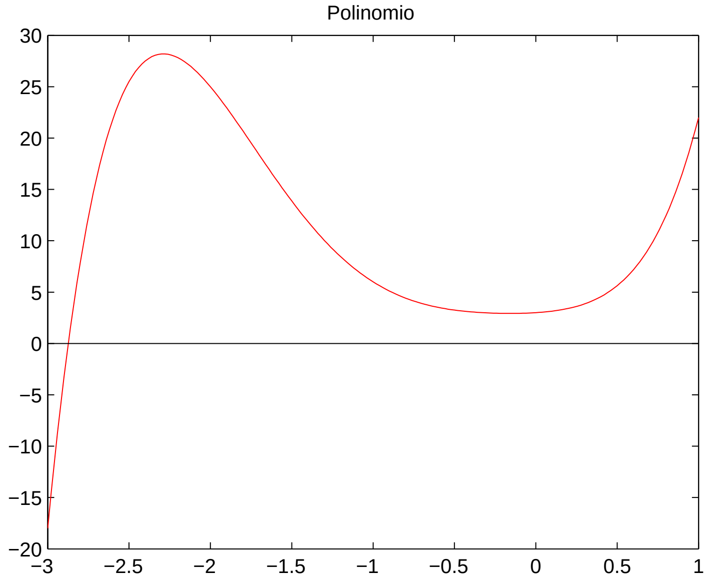
Pertanto potremo utilizzare il seguente algoritmo che richiede $2n$ addizioni ed n moltiplicazioni.

$u = \alpha_n$
for $j = n-1, \ldots, 0$ **step** -1
  $u = u \cdot \bar{x} + \alpha_j$
end for j

Si definisca una function di nome `horner` che, dati i coefficienti α_i (memorizzati per indici crescenti in un vettore) ed un valore, permetta di valutare un polinomio $P_n(x)$ in tale valore $\bar{x}$. La function avrà la seguente intestazione

```
function fxbar = horner (alpha,xbar)
% HORNER Calcola il valore di un polinomio in xbar
%        utilizzando l'algoritmo di Horner
%
% Uso:
%   fxbar = horner (alpha,xbar)
%
% Dati di ingresso:
%   alpha  vettore dei coefficienti ordinati per indici
%          crescenti (alpha_0, alpha_1, ... )
%   xbar   valore in cui si vuole valutare il polinomio
%
% Dati di uscita:
%   fxbar  valore di P(xbar)
```

Si scriva poi uno script che calcoli il valore del polinomio in 201 valori dell'intervallo I, per poterlo rappresentare graficamente e si produca una figura con il polinomio dato.
Il risultato dovrà essere quello della seguente figura:



Suggerimento: Si deve prima modificare l'algoritmo originale per adattarlo alle esigenze del Matlab per quanto riguarda gli indici delle componenti dei vettori e le specifiche della funzione assegnate.

Ecco l'algoritmo modificato:

$$[fxbar] = \textbf{Horner } (\alpha, xbar)$$

$n1$ = lunghezza vettore α
$fxbar = \alpha(n1)$
for $j = n1 - 1, \ldots, 1$ **step** -1
    $fxbar = fxbar * xbar + \alpha(j)$
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 per la rappresentazione di un polinomio nell'intervallo [a,b]
% valutandolo in 201 punti con lo schema di Horner
% Necessita della function horner
%======================================================================

% Definisce il polinomio (coefficienti con indici CRESCENTI)
% .................................
% Definisce l'intervallo [a,b]
% .................................
% Definisce le ascisse per il grafico (201 punti equispaziati)
% .................................
% Valuta il polinomio nelle 201 ascisse (ciclo for che usa
% la function horner)
% .................................

% Creazione della figura che contiene
%   la rappresentazione del polinomio
% .................................
```

Esercizio 2

Risultati analoghi a quelli dell'esercizio precedente si possono ottenere utilizzando la funzione Matlab:

```
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 ponga attenzione al fatto che in tale funzione Matlab i coefficienti devono essere inseriti nel vettore con gli indici descrescenti (e non crescenti).

Si produca la stessa figura precedentemente richiesta.