

## Esercitazione 7

Nella prima parte di questa esercitazione vedremo come realizzare una procedura che *determina il valore del polinomio che interpola dati assegnati in un insieme di punti*. Nella seconda parte utilizzeremo la realizzazione per disegnare il grafico degli elementi della base di Lagrange e per studiare, in alcuni esempi, il problema del campionamento e ricostruzione.

### Prima parte

Assegnate una colonna  $x$  di  $k+1$  numeri reali *distinti*, una colonna  $y$  di  $k+1$  numeri reali e una colonna  $xp$  di  $m$  numeri reali, la definizione seguente è la realizzazione di una procedura che restituisce la colonna dei valori in  $xp_1, \dots, xp_m$  del polinomio  $p$  che interpola i dati  $(x_1, y_1), \dots, (x_{k+1}, y_{k+1})$ .

```
function yp = PolyInterpNewt(x,y,xp)
//
// Calcola la colonna yp dei valori in xp(1),...,xp(m) del polinomio p
// che interpola i dati (x(1),y(1)),..., (x(k+1),y(k+1))
//
k = length(x) - 1;
// Calcola la colonna c dei coefficienti della forma di Newton di p
N = zeros(k+1,k+1);
N(:,1) = ones(k+1,1);
for j = 2:k+1,
    N(j:k+1,j) = N(j:k+1,j-1) .* (x(j:k+1) - x(j-1));
end;
c = SA(N,y);
// Calcola la colonna yp dei valori di p
yp = c(k+1);
for j = k:-1:1,
    yp = c(j) + (xp - x(j)) .* yp;
end;
endfunction
```

*Infatti:* se il linguaggio mette a disposizione il tipo *numero reale* si constata che (a) le componenti della colonna  $c$  sono i coefficienti di  $p$  rispetto alla base di Newton:

$$1, \quad x - x_1, \quad (x - x_1)(x - x_2), \quad \dots, \quad (x - x_1) \cdots (x - x_k)$$

e (b) le componenti della colonna  $yp$  sono i valori di  $p$  in  $xp_1, \dots, xp_m$  calcolati a partire dai coefficienti  $c_1, \dots, c_{k+1}$  con il *metodo di Horner*.

Per comprendere quest'ultimo punto, si consideri il caso  $k = 3$ . In tal caso la forma di Newton del polinomio interpolante è:

$$p(x) = c_1 + c_2(x - x_1) + c_3(x - x_1)(x - x_2) + c_4(x - x_1)(x - x_2)(x - x_3)$$

che si riscrive, secondo il metodo di Horner:<sup>1</sup>

$$p(x) = c_1 + (x - x_1) \cdot [c_2 + (x - x_2) \cdot (c_3 + (x - x_3) \cdot c_4)]$$

La sequenza dei valori di  $yp$  calcolati dalla procedura è:

```
* yp = c4
* yp = c3 + (xp - x3) * c4
* yp = c2 + (xp - x2) .* (c3 + (xp - x3) * c4)
* yp = c1 + (xp - x1) .* [c2 + (xp - x2) .* (c3 + (xp - x3) * c4)]
```

---

<sup>1</sup>Vedere, ad esempio: [https://en.wikipedia.org/wiki/Horner's\\_method#Description\\_of\\_the\\_algorithm](https://en.wikipedia.org/wiki/Horner's_method#Description_of_the_algorithm).

Gli elementi della sequenza, dal secondo, sono *colonne* di  $m$  numeri reali e l'ultimo è la colonna dei valori di  $p$  in  $x_{p_1}, \dots, x_{p_m}$ . Si osservi che, nota la colonna  $c$ , il *costo* del calcolo di  $yp$  è:  $3mk$ .

## Seconda parte

Come primo esempio di utilizzazione di `PolyInterpNewt` si disegnano i grafici degli elementi della *base di Lagrange* generata dai cinque punti che dividono l'intervallo  $[0, 1]$  in quattro sottointervalli di uguale lunghezza:

```
-->k = 5;

-->n = 50; // punti aggiuntivi per il disegno

-->x = linspace(0,1,k+1)';

-->y = eye(k+1,k+1);

-->xp = linspace(0,1,k+1 + k*n)';

-->for i = 1:k+1, yp(:,i) = PolyInterpNewt(x, y(:,i), xp); end;

-->plot2d(xp,yp);

-->xtitle('Elementi della base di Lagrange');

-->xgrid(); xlabel('x'); legend('l_1','l_2','l_3','l_4','l_5',-1);

-->a = gca();

-->a.title.font_size = 2; a.x_label.font_size = 2;

-->a.children(1).font_size = 2;
```

Le istruzioni generano il disegno riportato in Figura 1. L'approssimazione del grafico dell' $i$ -esimo

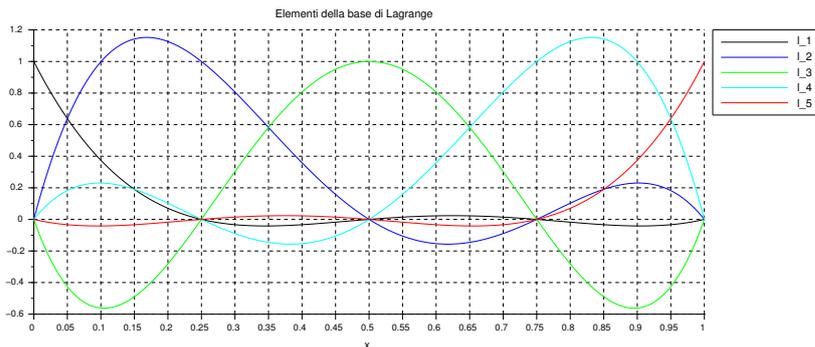


Figura 1: Base di Lagrange relativa alle ascisse: 0, 0.25, 0.5, 0.75, 1.

elemento della base di Lagrange  $\ell_i(x)$  è generata, come noto, assegnando in ingresso all'istruzione `plot2d` i vertici della spezzata da disegnare:  $xp$  e  $yp(:, i)$ . La procedura `PolyInterpNewt` è utilizzata per generare le componenti di  $yp(:, i)$ , ovvero i valori del polinomio che interpola i dati definiti dalle componenti delle colonne  $x$  ed  $y(:, i)$  —  $i$ -esima colonna di `eye(k+1, k+1)`, matrice identica di ordine  $k+1$ . Le componenti di  $xp$ , abbastanza numerose da dar luogo ad un'approssimazione dei grafici *visivamente gradevole*, sono scelte in modo da *includere* le ascisse utilizzate per definire la base di Lagrange.

Come secondo esempio si disegnano, sull'intervallo  $[0, 1]$ , approssimazioni dei grafici della funzione  $f(x) = 2 + x - 5x^2 + 4x^3$  e del polinomio  $p(x)$  che interpola i campioni di  $f$  agli istanti  $0, 0.25, 0.5, 0.75, 1$ . Si ottiene il disegno riportato in Figura 2. Si osserva che le curve blu e rossa

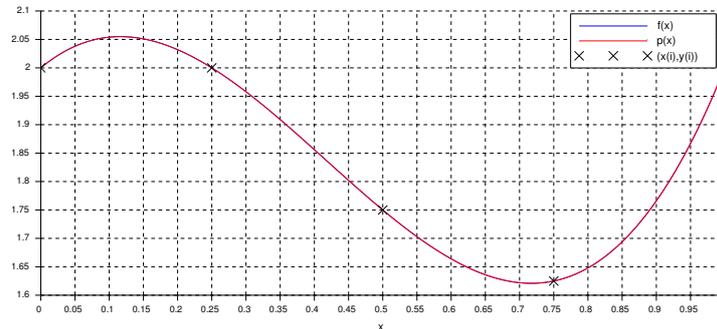


Figura 2: Campionamento e ricostruzione di  $f(x) = 2 + x - 5x^2 + 4x^3$ .

che approssimano, rispettivamente, i grafici di  $f(x)$  e  $p(x)$ , sono *visivamente sovrapposte*. Questo suggerisce, correttamente, che sia  $f = p$ . Infatti:  $p$  è il polinomio di grado al più *quattro* che interpola i campioni di  $f$ , che è un polinomio di grado *tre*. Quindi, in base al Teorema di esistenza ed unicità per il problema dell'interpolazione polinomiale:  $p = f$ . Un valore approssimato dell'errore di ricostruzione di  $f$  si ottiene con:

```
-->x = linspace(0,1,5)';
-->y = f(x);
-->xp = linspace(0,1,5 + 4*100)';
-->yp = PolyInterpNewt(x, y, xp);
-->max(abs(yp - f(xp)))
ans =
    6.661D-16
```

Il valore ottenuto è coerente con l'uguaglianza  $p = f$ , infatti  $yp$  ed  $f(xp)$  sono valori *dello stesso polinomio* calcolati con *algoritmi differenti*.

Come esempio finale si disegnano approssimazioni dei grafici, sull'intervallo  $[0, 1]$ , della funzione  $f(x) = e^{-x} \sin 3x$  e del polinomio  $p(x)$  che interpola i campioni di  $f$  agli istanti  $0, 0.25, 0.5, 0.75, 1$ . Si ottiene il disegno riportato in Figura 3. In questo caso le curve blu e rossa che approssimano,

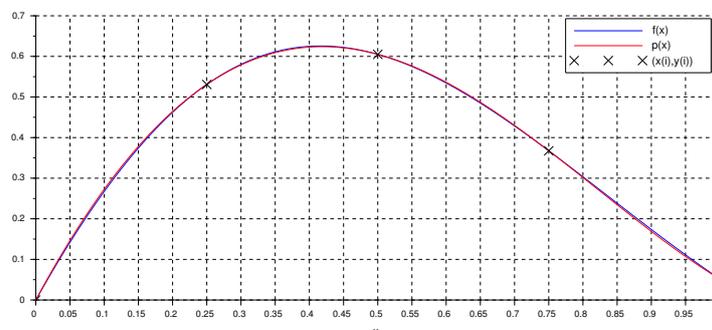


Figura 3: Campionamento e ricostruzione di  $f(x) = e^{-x} \sin 3x$ .

rispettivamente, i grafici di  $f(x)$  e  $p(x)$ , sono *visivamente distinte*. Questo suggerisce, correttamente, che sia  $f \neq p$ . Infatti,  $p$  è un polinomio mentre  $f$  non lo è. Un valore approssimato dell'errore di ricostruzione si ottiene come nel caso precedente:  $e(f) \approx 5.4 \cdot 10^{-3}$ .

Se *aumentiamo il numero  $k + 1$  degli istanti di campionamento suddividendo l'intervallo  $[0, 1]$  in  $k$  intervalli uguali*, si osserva che l'approssimazione dell'errore di ricostruzione *diminuisce*: per  $k = 9$  si ottiene  $e(f) \approx 2.0 \cdot 10^{-7}$ , per  $k = 19$  si ha  $e(f) \approx 6.2 \cdot 10^{-14}$ .

---

## Esercizi

---

1. Determinare il costo del calcolo di `PolyInterpNewt(x,y, xp)` in funzione di  $k$  (numero delle componenti di  $\mathbf{x}$  ed  $\mathbf{y}$ ) ed  $m$  (numero delle componenti di  $\mathbf{x}_p$ ).

2. Spiegare i comandi:

```
-->a = gca();
```

```
-->a.title.font_size = 2; a.x_label.font_size = 2;
```

```
-->a.children(1).font_size = 2;
```

utilizzati per la Figura 1.

3. Siano  $k$  un intero positivo e  $\mathbf{x} = \text{linspace}(0, 1, k)'$ . Detti  $\ell_1(x), \dots, \ell_k(x)$  gli elementi della base di Lagrange relativa alle ascisse  $\mathbf{x}_1, \dots, \mathbf{x}_k$ , utilizzare *Scilab* per ottenere, con  $k = 5, 10, 15$ :

(1) Un grafico (approssimato) della funzione  $\lambda(x) = |\ell_1(x)| + \dots + |\ell_k(x)|$ ;

(2) Un'approssimazione di  $\lambda(\mathbf{x}_1, \dots, \mathbf{x}_k) = \max_{x \in [0, 1]} \lambda(x)$ .

4. Siano  $k$  un numero intero positivo,  $\mathbf{x} = \text{linspace}(0, 1, k)'$  e  $f(x) = e^{-x} \sin 3x$ . Detto  $p$  il polinomio che interpola i campioni di  $f$  agli istanti di campionamento  $\mathbf{x}_1, \dots, \mathbf{x}_k$ , utilizzare *Scilab* per ottenere, con  $k = 5, 10, 20$ , una curva che approssima il grafico della funzione  $|f(x) - p(x)|$  sull'intervallo  $[0, 1]$ .