Calcolo Numerico
a.a. 2010/11
Prof. L.D'Amore
Esercitazione del 30 maggio 2011

Spline cubica *naturale* interpolante e confronti tra modelli di fitting[1]

1. La funzione `matlab csape`, con l'opzione `'variational'`, costruisce, su un insieme di campioni

$$(x_i, y_i)_{i=1,\dots,n}$$

la **spline cubica *naturale* interpolante** i nodi in $\underline{x}$; assegnato un vettore di punti di valutazione, xx, appartenenti ad un intervallo contenente i nodi, la funzione `fnval` consente di valutarla:

```
>> spNat = csape(x,y,'variational');
>> values = fnval(spNat,xx);
>> plot(xx,values,'bd')
```

2. Si costruisca (utilizzando la funzione `csapi` oppure `spline`) la spline cubica (**non naturale**) interpolante i nodi x; la si valuti in corrispondenza delle 121 ascisse xx, equispaziate in $[0, 6]$; si visualizzino, sullo stesso grafico, i dati e le valutazioni:

```
x = [1 1.5 2 4.1 5];
y = [1 -1 1 -1 1];
xx = linspace(0,6,121);
plot(xx,csapi(x,y,xx),'k-',x,y,'ro')
title('Cubic Spline Interpolant to Five Points')
```

A partire dagli stessi dati, costruire l'unico polinomio di Lagrange interpolante i nodi x. Detto yy il vettore delle valutazioni di tale polinomio, sovrapporre la curva che ne rappresenta il grafico nell'intervallo $[0, 6]$ al grafico della spline:

---

[1] In tutti gli esercizi, se svolti con `Scilab` utilizzare la funzione `splin` (con l'opzione '`natural`' per la spline naturale, con l'opzione '`not_a_knot`' per la non naturale), se svolti con `Octave`, si utilizzi la funzione `spline` che costruisce una spline cubica (**non naturale**).

```
hold on
plot(xx,csapi(x,y,xx),'k-',x,y,'ro', xx,yy,'g*')
```

3. **Esercizio 1:** Un esperimento ha prodotto i seguenti dati:

t	0.0	0.5	1.0	6.0	7.0	9.0
y	0.0	1.6	2.0	2.0	1.5	0.0

   (a) Si desidera costruire un modello che realizzi il *fitting* dei dati, che descriva, in modo accurato, la funzione che li ha generati. A tal fine si costruisca:
      i. (in doppia precisione), utilizzando un elemento di software matematico già sviluppato, l'unico polinomio di Lagrange, interpolante i nodi, specificandone il grado; se ne realizzi il grafico nell'intervallo $[0, 9]$.
      ii. la spline cubica naturale interpolante; se ne realizzi il grafico nell'intervallo $[0, 9]$;
      iii. la parabola di migliore approssimazione nel senso dei minimi quadrati; se ne realizzi il grafico nell'intervallo $[0, 9]$.

   (b) Quale modello sembra risultare più accurato?

   (c) Quali osservazioni si possono fare sulla scelta del modello di fitting, in base alle informazioni di cui si dispone sull'insieme dei dati (ovvero sulla natura, sul numero e sull'errore di cui sono affetti)?

4. **Esercizio 2:** Assegnati i campioni:

x	0	1	2	3	4	5	6
y	10	13	16	19	22	25	28

si supponga che le ordinate siano generate da una funzione di cui non sia nota l'espressione analitica e che tali valutazioni siano acquisite mediante uno strumento per cui risultino affette da errore, diventando, così :

```
>> ynoise=y+randn(1,length(x))
```

(a) Assumendo che i dati siano le coppie

$$(\mathtt{x}(\mathtt{i}), \mathtt{ynoise}(\mathtt{i}))_{\mathtt{i}=1,\dots,8}$$

i. calcolare i coefficienti della parabola p, *di migliore approssimazione nel senso dei minimi quadrati*, specificando la funzione `matlab` utilizzata;

ii. calcolare (in doppia precisione), utilizzando un elemento di software matematico già sviluppato, i coefficienti dell'unico polinomio di Lagrange, q, *interpolante i nodi assegnati*, specificandone il grado.

iii. calcolare i coefficienti della spline *s cubica naturale interpolante* i nodi assegnati, specificando la funzione `matlab` utilizzata.

(b) Valutare, in corrispondenza di ascisse a scelta, appartenenti ad un intervallo contenente i nodi, i tre modelli definiti a partire dai dati perturbati, **specificando, in particolare, l'algoritmo di valutazione o le funzioni `matlab` utilizzate**.

(c) Tracciare, nella stessa figura, il grafico dei tre modelli e le coppie $(x, ynoise)$ e (x, y). Confrontare l'andamento dei tre modelli; quale di essi risulta più attendibile per il *fitting* dei dati (x, y), ovvero per l'approssimazione della funzione che li ha generati? Questo risultato vale in generale?