Calcolo Numerico
Informatica, a.a. 2008/09
Prof.ssa L. D'Amore

Costruzione e valutazione di una spline cubica naturale, interpolante un insieme di nodi assegnati

1. Scrivere un elemento di software matematico

   spline(in: n,x,y,z; out: s,iflag)

   per la costruzione e valutazione di una spline *cubica naturale interpolante* un insieme di n nodi, che implementi il seguente schema:

   (a) Controllo sui dati di input segnalando all'utente, mediante un indicatore di errore $iflag$, se $n < 2$ o la presenza di nodi coincidenti;

   (b) chiamata delle routine:

      i. *ordinamento(n,x,y)*: ordina i nodi x_i (ed i valori corrispondenti y_i) in ordine crescente;

      ii. *costruzione(n,x,y, b, dinf, d, dsup)*: costruisce il vettore dei termini noti ($b = 3B\Delta f$) e la matrice dei coefficienti, A, del sistema la cui soluzione fornisce i valori che la derivata prima assume nei nodi;

      iii. *risoluzione(n,b, dinf, d, dsup)*: risolve il sistema $A\lambda = 3B\Delta f$ mediante la **fattorizzazione LU specifica per matrici tridiagonali**, *a diagonale dominante*, eventualmente utilizzando 3 array per la memorizzazione delle diagonali di A e di B;

      iv. *valutazione(n,x,y,z,s)*: esegue, dapprima, la ricerca binaria dell'intervallo di appartenenza del punto di valutazione, z, assegnato in input; costruisce, poi, il polinomio che rappresenta la spline in tale intervallo. Se il punto di valutazione appartiene ad un intervallo del tipo $[x_i, x_{i+1}]$, $i = 1, \ldots, n-1$, ovvero se cade tra due nodi di costruzione consecutivi, la routine calcola i coefficienti della formula di Newton per il polinomio interpolante di Hermite:

$$\begin{aligned} p_i(x) &= a_0 + a_1(x - x_i) + a_2(x - x_i)^2 + a_3(x - x_i)^2(x - x_{i+1}) \\ a_0 &= y_i \\ a_1 &= \lambda_i = y[x_i, x_i] \\ a_2 &= y[x_i, x_i, x_{i+1}] \\ a_3 &= y[x_i, x_i, x_{i+1}, x_{i+1}] \end{aligned}$$

      e valuta p_i in z, utilizzando l'algoritmo di Horner. Invece, se:

$$z < x_1 \;\Rightarrow\; \begin{array}{l}\text{costruisce e valuta} \\ \text{la retta di coefficiente angolare } \lambda_1\end{array}$$

$$\begin{array}{rcl} s & = & y(1) + (z - x(1)) * \lambda_1 \\ z > x_n & \Rightarrow & \text{costruisce e valuta} \\ & & \text{la retta di coefficiente angolare } \lambda_n \\ s & = & y(n) + (z - x(n)) * \lambda_n \end{array}$$

dove λ_1 e λ_n rappresentano i valori che la derivata prima della spline assume, rispettivamente, nel primo e nell'ultimo nodo.

2. **Esercizio**: Siano assegnati i nodi ed i valori corrispondenti:

$$\{(x_i, y_i)\}_{i=1,\ldots,5} = \{(-2, 3), \quad (-0.5, 2) \quad (1, 4.5), \quad (3, 0.3), \quad (6, 4.8)\}$$

   (a) Costruire, mediante un elemento di software matematico la spline *cubica naturale interpolante* i nodi assegnati calcolandone i coefficienti a_0, a_1, a_2, a_3 intervallo per intervallo.

   (b) Tracciare il grafico della spline con `matlab`; ad esempio nel primo intervallo la spline si può rappresentare secondo il procedimento seguente:

>>	%definizione dell'intervallo in cui rappresentare
>>	%il polinomio di Hermite
>>	$i1$ = [-2:0.01:-0.5]
>>	%valutazione del polinomio di Hermite interpolante
>>	%(-2,3) e (-0.5, 2) nei punti dell'array $i1$
>>	$y1 = a_0 + a_1 * (i1 + 2) + a_2 * (i1 + 2).\hat{}2 +$
>>	$a_3 * ((i1 + 2).\hat{}2). * (i1 + 0.5)$
>>	%grafico del polinomio di Hermite nell'intervallo [-2,-0.5]
>>	$plot(i1, y1,' b-')$

   Analogamente ripetere la procedura per la rappresentazione della spline negli altri intervalli, ponendo particolare attenzione alla rappresentazione della spline in $(-\infty, -2]$ e $[6, \infty)$.

   (c) Mettendo insieme i risultati ottenuti al punto precedente, rappresentare la spline cubica naturale interpolante nell'intervallo $(-\infty, +\infty)$, tracciando sullo stesso grafico, con colori e stili diversi, le curve che rappresentano i tratti polinomiali della spline, attraverso le istruzioni[1]

$$>> \qquad plot(i1, y1,' b-')$$

[1] Aiutarsi con

$$>> \qquad help\ plot$$

per informazioni sulla funzione `plot`, in particolare sulla scelta del colore e dello stile della curve tracciate.

$$
\begin{array}{ll}
>> & \textit{hold on} \\
>> & plot(i2, y2,' r - .') \\
>> & \textit{hold on} \\
>> & plot(i3, y3,' g - -') \\
>> & \textit{hold on} \\
>> & \ldots
\end{array}
$$

Infine, riportare sullo stesso grafico i punti $(x_i, y_i)_{i=1,\ldots,5}$ assegnati:

```
>> hold on
>> plot(x,y,'r*')
```

dove $x = [x_1, \ldots, x_5]$ e $y = [y_1, \ldots, y_5]$.

(d) Confrontare il grafico ottenuto, attraverso i risultati forniti dal software implementato, sovrapponendo ad esso quello che si ottiene mediante la funzione matlab `csape` che, con l'opzione `'variational'`, costruisce la spline cubica naturale interpolante i nodi in $\underline{\mathtt{x}}$:

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

dove xx è un vettore di punti di valutazione in $[-2, 6]$, ad esempio $[-2 : 0.1 : 6]$.

3. **Esercizio** 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. il polinomio di grado cinque, interpolante i nodi; 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)?