

Calcolo Numerico

Esercizi sulle splines

SI RACCOMANDA AGLI STUDENTI DI **commentare adeguatamente** SCRIPT E FUNCTION MATLAB.

1. Si definisca una function

`errore_spline_lineare`

avente la seguente intestazione

```
function maxerr=errore_spline_lineare(f,a,b,N)

% Oggetto:
% Errore della spline lineare "s1" interpolante "f" nei
% punti  $x_k=a+(k-1)h$ ,  $k=1,\dots,N+1$ ,  $h=(b-a)/N$ 
%
% Input:
% f: funzione da interpolare
% a,b: dominio di "f"
% N: numero di subintervalli equispaziati in cui e'
%    suddiviso [a,b]
%
% Output:
% maxerr: stima del massimo errore di interpolazione,
%         ovvero  $\max(\text{abs}(f(x)-s1(x)))$ 
```

Tale function

- definisca il vettore \mathbf{x} in cui $x_k = a + (k - 1) \cdot h$, $h = (b - a)/N$, $k = 1, \dots, N + 1$ (punti equispaziati, non serve utilizzare il ciclo for!);
- definisca il vettore \mathbf{y} in cui $y_k = f(x_k)$;
- definisca il vettore \mathbf{t} in cui $t_k = a + (k - 1) \cdot h$, $h = (b - a)/M$, $k = 1, \dots, M + 1$ in cui $M = 100000$ (punti equispaziati!);
- valuti la spline lineare s interpolante le coppie $(x_k, f(x_k))$, $k = 1, \dots, N + 1$ nei punti test t_k , $k = 1, \dots, M + 1$ e ponga il risultato nel vettore \mathbf{z} (si utilizzi il comando `interp1`);
- valuti la funzione f nei punti test t_k , $k = 1, \dots, M + 1$ e ponga il risultato nel vettore `ftx`;
- calcoli il massimo errore `maxerr`, ovvero $\max_{k=1,\dots,M+1} |f(t_k) - s(t_k)|$.

2. Si definisca una funzione `demo_runge_spline_lineare` che

- non abbia variabili di input, né di output;
- definisca la funzione di Runge f ove $f(x) = 1/(1 + x^2)$, in forma vettoriale;
- ponga `a=-5` e `b=5`;
- utilizzando la chiamata

```
maxerr=errore_spline_lineare(f,a,b,N)
```

calcoli per $N = 2, 4, 8, 16, 32, 64, 128, 256, 512 = 2^9$ il valore assunto da `maxerr` e immagazzini il valore ottenuto per $N = 2^k$ nella k -sima componente del vettore `eev`; in altri termini si calcoli per ogni $N(k)=2^k$, $k = 1, \dots, 9$, la quantità `maxerr` relativa a tale scelta di $N(k)$ e la si salvi in `eev(k)` (attenzione, la funzione `errore_spline_lineare` non é vettoriale, quindi necessita un ciclo `for`);

- esegua in una figura i grafici in scala semilogaritmica delle coppie $(2^k, \text{eev}(k))$,
- utilizzi quale titolo della figura la stringa

```
'Errori di interpolazione spline lineare: nodi equispaziati'
```

ed il plot abbia la preferenza `'LineWidth',2`;

- salvi su un file

`errori_interpolazione_spline_lineare.txt`

- i valori del tipo 2^k utilizzati, gli errori `eev`, cosicché la tabella risultante abbia alla k -sima riga,
- la quantità 2^k con 5 cifre prima della virgola e nessuna dopo la virgola, in notazione decimale,
 - l'errore `eev(k)`, ovvero la k -sima componente del vettore `eev`, con 1 cifra prima della virgola, una dopo la virgola, in notazione esponenziale.

3. Si definisca una function

`errore_spline_cubica`

avente la seguente intestazione

```
function maxerr=errore_spline_cubica(f,a,b,N)

% Oggetto:
% Errore della spline lineare "s3" interpolante "f" nei
% punti  $x_k=a+(k-1)h$ ,  $k=1,\dots,N+1$ ,  $h=(b-a)/N$ 
%
% Input:
% f: funzione da interpolare
% a,b: dominio di "f"
% N: numero di subintervalli equispaziati in cui e'
%    suddiviso [a,b]
%
% Output:
% maxerr: stima del massimo errore di interpolazione,
%         ovvero  $\max(\text{abs}(f(x)-s_3(x)))$ 
```

Tale function

- definisca il vettore `x` in cui $x_k = a + (k - 1) \cdot h$, $h = (b - a)/N$, $k = 1, \dots, N + 1$ (punti equispaziati, non serve utilizzare il ciclo `for!`);
- definisca il vettore `y` in cui $y_k = f(x_k)$;
- definisca il vettore `t` in cui $t_k = a + (k - 1) \cdot h$, $h = (b - a)/M$, $k = 1, \dots, M + 1$ in cui $M = 100000$ (punti equispaziati, non serve utilizzare il ciclo `for!`);
- valuti la spline cubica s_3 con condizioni *not-a-knot* interpolante le coppie $(x_k, f(x_k))$, $k = 1, \dots, N + 1$ nei punti test t_k , $k = 1, \dots, M + 1$ e ponga il risultato nel vettore `z` (si utilizzi il comando `spline`);
- valuti la funzione f nei punti test t_k , $k = 1, \dots, M + 1$ e ponga il risultato nel vettore `ftx`;
- calcoli il massimo errore `maxerr`, ovvero $\max_{k=1,\dots,M+1} |f(t_k) - s_3(t_k)|$.

4. Si definisca una funzione `demo_runge_spline_cubica` che

- non abbia variabili di input, né di output;
- definisca la funzione di Runge `f` ove $f(x) = 1/(1 + x^2)$, in forma vettoriale;
- ponga `a=-5` e `b=5`;
- utilizzando la chiamata

```
maxerr=errore_spline_cubica(f,a,b,N)
```

calcoli per $N = 2, 4, 8, 16, 32, 64, 128, 256, 512 = 2^9$ il valore assunto da `maxerr` e immagazzini il valore ottenuto per $N = 2^k$ nella k -sima componente del vettore `eev`;

- esegua in una figura i grafici in scala semilogaritmica delle coppie $(2^k, \text{eev}(k))$,
- utilizzi quale titolo della figura la stringa

```
'Errori di interpolazione spline cubica: nodi equispaziati'
```

ed il plot abbia la preferenza `'LineWidth',2`;

- salvi su un file

`errori_interpolazione_spline_cubica.txt`

i valori del tipo 2^k utilizzati, gli errori `eev`, cosicché la tabella risultante abbia alla k -sima riga,

- la quantità 2^k con 5 cifre prima della virgola e nessuna dopo la virgola, in notazione decimale,
- l'errore `eev(k)`, ovvero la k -sima componente del vettore `eev`, con 1 cifra prima della virgola, una dopo la virgola, in notazione esponenziale.