Calcolo Numerico
Laboratorio, esercizio sull'interpolazione polinomiale

Si raccomanda agli studenti di **commentare adeguatamente** script e function Matlab.

Definiamo

1. la funzione di Runge

```
function [fx]=runge(x)

% input
% x: vettore di ascisse

% output
% y: vettore la cui "k"-sima componente e' il
%   valore assunto dalla funzione di Runge "f(s)=1/(1+s^2)";
%   nella "k"-sima componente di "x".

fx=1./(x.^2+1);
```

2. la funzione `gcl`

```
function xc=gcl(a,b,n)

% input:
% a,b: estremi dell'intervallo
% n  : numero dei nodi di Chebyshev-Lobatto.
%
% output:
% xc  : vettore di nodi di Chebyshev-Lobatto.

m=1:1:n;
xc=(a+b)/2-((b-a)/2)*cos(pi*(m-1)/(n-1));
```

3. Quindi scriviamo il file `esperimento.m`

```
function esperimento

% oggetto:
% esempio di Runge per grado fissato "n", in nodi equispaziati e di
% Chebyshev-Lobatto.

% grado interpolante.
n=12;
% nodi test
s=-5:10/1000:5;

% ---- interpolazione nodi equispaziati ----
x=-5:10/n:5; y=runge(x);
t=interpol(x,y,s);

% ---- interpolazione nodi GCL ----
xgcl=gcl(-5,5,n+1); ygcl=runge(xgcl);
tt =interpol(xgcl, ygcl,s);

% ---- plot runge vs interpolanti ----

% prima figura
fs=runge(s);
clf;
figure(1); % prima figura (nodi equi.)
plot(s,fs,s,t,'LineWidth',2);
hold on;
title('Errori di interpolazione');
legend('funzione runge','intp. nodi eqsp.');
hold off;
```

```
% seconda figura
figure(2); % seconda figura (nodi GCL)
plot(s,fs,s,tt,'LineWidth',2);
hold on;
legend('funzione runge','intp. nodi GCL');
title('Errori di interpolazione');
hold off;

% valutazione errori assoluti
fs=runge(s); % valutazione funzione di Runge, punti test
ee=max(abs(fs-t)); % nodi equispaziati
ec=max(abs(fs-tt));  % nodi GCL
fprintf('\n \t[Errori interpolazione][E]:%1.2e [GCL]:%1.2e',ee,ec);
fprintf('\n \n');
```

4. Utilizziamo la funzione `interpol.m`

```
function [yy]=interpol(x,y,xx)

%----------------------------------------------
% Interpolazione
%
% In input:
% x: nodi.
% y: valori nei nodi.
% xx nodi su cui calcolare l'interpolante.
% m: grado dell'approssimante ai minimi quadrati
%
% In output:
% yy: valori dell'interpolante o approssimante.
%----------------------------------------------

m=length(x)-1;
coeff=polyfit(x,y,m);
yy=polyval(coeff,xx);
```

1 Esercizio `test_runge`

Prendendo come esempio il file `esperimento`, lo si modifichi in `test_runge` cosicché:

1. abbia la seguente intestazione

```
function [ee,ec]=test_runge(n)

% Oggetto:
% Sia "f" la funzione di Runge e con "p^(E)_n",
% "p^(GCL)_n" le interpolanti polinomiali della
% funzione di Runge "f" rispettivamente in "n+1"
% nodi equispaziati e di Gauss-Chebyshev-Lobatto.
% Si approssimano
% ee=max_{x in [-5,5]} |f(x)-p^(E)_n(x)|
% ec=max_{x in [-5,5]} |f(x)-p^(GCL)_n(x)|
%
% Input:
% n: grado delle interpolanti
%
% Output:
% ee: max_{x in [-5,5]} |f(x)-p^(E)_n(x)|
% ec: max_{x in [-5,5]} |f(x)-p^(GCL)_n(x)|
```

2. abbia come input la variabile n, grado dell'interpolante p_n che non sia necessariamente 12 (si veda l'intestazione!);

3. abbia come output le variabili `ee`, `ec`, ovvero approssimanti $\max_{x\in[-5,5]} |f(x) - p_n^{(E)}(x)|$, $\max_{x\in[-5,5]} |f(x) - p_n^{(GCL)}(x)|$, con $p_n^{(E)}$, $p_n^{(GCL)}$ le interpolanti polinomiali di grado n della funzione di Runge f rispettivamente in $n+1$ nodi equispaziati e di Gauss-Chebyshev-Lobatto;

4. esegua il test dell'interpolazione in

$$s_k = -5 + (k-1)h, \quad k = 1, \ldots, 10001, h = \frac{1}{10000}$$

(in caso di difficoltá si usi opportunamente `linspace`);

5. non contenga grafici;

6. non contenga statistiche fornite all'utente;

2 Esercizio `demo_runge1`

Utilizzando la funzione `test_runge`, si definisca una funzione `demo_runge1` che

1. non abbia variabili di input, né di output;

2. calcoli il valore assunto dalle variabili `ee` e `ec`, definendo i vettori `eev`, `ecv`, aventi lunghezza 13, tali che
   - la n sima componente di `eev` corrisponda al valore `ee` fornito tramite `test_runge` per tale n,
   - la n sima componente di `ecv` corrisponda al valore `ec` fornito tramite `test_runge` per tale n,

3. esegua in una due figure (prima del primo plot si utilizzi il comando `figure(1)` e prima del secondo plot si si utilizzi il comando `figure(2)`) i grafici in scala semilogaritmica sia delle coppie (n,`eev`(n)) che delle coppie (n,`ecv`(n)),

4. utilizzi quale titolo della prima figura la stringa

   `'Errori di interpolazione:  nodi equispaziati'`

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

5. utilizzi quale titolo della seconda figura la stringa

   `'Errori di interpolazione:  nodi GCL'`

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

6. salvi su un file `errori_interpolazione.txt`, i valori di n utilizzati, gli errori `eev`, `ecv`, cosicché la tabella risultante abbia alla k-sima riga,
   - l'indice k con 2 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,
   - l'errore `ecv(k)`, ovvero la k-sima componente del vettore `ecv`, con 1 cifra prima della virgola, una dopo la virgola, in notazione esponenziale.

3 Esercizio `demo_runge2`

Prendendo come esempio il file `esperimento`, lo si modifichi nel file `demo_runge2` cosicché:

- invece di eseguire il grafico della funzione e delle sue interpolanti polinomiali di grado 12, ovvero $p_{12}^{(E)}$, $p_{12}^{(GCL)}$, valuti le funzioni

  $$|f(x) - p_{12}^{(E)}(x)|$$

  $$|f(x) - p_{12}^{(GCL)}(x)|$$

  nei punti

  $$s_k = -5 + (k-1)h, \quad k = 1, \ldots, 1000001, h = \frac{1}{1000000}$$

  e ne disegni i grafici in due figure separate, in scala semilogaritmica.

- la prima figura abbia titolo

  `Errori di interpolazione:  nodi equi.`

- la seconda figura abbia titolo

  `Errori di interpolazione:  nodi GCL`

- non si salvino risultati su testo.