Corso di Laurea Triennale in Ingegneria e Scienze Informatiche

---

Metodi Numerici **A.A. 2020-2021**

Esercitazione N. 5
Approssimazione ai minimi quadrati ed Interpolazione di dati e funzioni

---

Obiettivo
Sperimentazione numerica relativa all'approssimazione ai minimi quadrati ed all'interpolazione di dati e funzioni.

Sperimentazione numerica per l'approssimazione ai minimi quadrati

1. Si implementi e si utilizzi la function `metodoQR` per l'approssimazione ai minimi quadrati (metodo QRLS) delle seguenti configurazioni di dati:

$$
\begin{aligned}
&x_1 = [-3.5,\ -3,\ -2,\ -1.5,\ -0.5,\ 0.5,\ 1.7,\ 2.5,\ 3];\\
&y_1 = [-3.9,\ -4.8,\ -3.3,\ -2.5,\ 0.3,\ 1.8,\ 4,\ 6.9,\ 7.1];\\
&x_2 = [-3.14,\ -2.4,\ -1.57,\ -0.7,\ -0.3,\ 0,\ 0.4,\ 0.7,\ 1.57];\\
&y_2 = [0.02,\ -1,\ -0.9,\ -0.72,\ -0.2,\ -0.04,\ 0.65,\ 0.67,\ 1.1];\\
&x_3 = numpy.linspace(0, 3, 12);\\
&y_3 = exp(x_3) * cos(4 * x_3) + numpy.random.randn(12,);\\
&x_4 = [1.001,\ 1.0012,\ 1.0013,\ 1.0014,\ 1.0015,\ 1.0016];\\
&y_4 = [-1.2,\ -0.95,\ -0.9,\ -1.15,\ -1.1,\ -1];
\end{aligned}
$$

   al variare del grado n tra 1 e 3.

2. Per i dati (x_i, y_i) riportati nei seguenti array

$$
\begin{aligned}
&x = [0.0004,\ 0.2507,\ 0.5008,\ 2.0007,\ 8.0013];\\
&y = [0.0007,\ 0.0162,\ 0.0288,\ 0.0309,\ 0.0310];
\end{aligned}
$$

   - costruire la retta di regressione;
   - costruire la parabola approssimante i dati nel senso dei minimi quadrati;
   - costruire la cubica approssimante i dati nel senso dei minimi quadrati;

   Quale tra le tre approssimazioni risulta la migliore? (Confrontare i grafici e la norma euclidea al quadrato del vettore dei residui).

3. Scrivere uno script Python per approssimare la seguente configurazione di punti

$$
\begin{aligned}
&x = numpy.arange(10, 10.6, 0.1);\\
&y = numpy.array([11.0320, 11.1263, 11.1339, 11.1339, 11.1993, 11.1844]);
\end{aligned}
$$

mediante un polinomio ai minimi quadrati di grado 4 costruito con il metodo QRLS. Perturbare poi il secondo punto nel seguente modo

$$x(2) = x(2) + 0.013; \quad y(2) = y(2) - 0.001;$$

e calcolare il polinomio ai minimi quadrati relativo alla configurazione perturbata. Commentare e motivare i risultati ottenuti.

Codici per l'interpolazione polinomiale nella forma di Lagrange
Scrivere la function `interpL` che calcoli il polinomio interpolante in forma di Lagrange
Tale function deve assumere come dati in input:

- x vettore dei nodi di interpolazione,
- y vettore dei valori della funzione nei nodi di interpolazione,
- xx vettore dei punti in cui si vuole valutare il polinomio interpolante.

In output deve essere restituito yy vettore contenente i valori assunti dal polinomio interpolante.

Funzioni Python utili:

- `numpy.poly()` restituisce i coefficienti di un polinomio di zeri assegnati,
- `numpy.polyval(p, x)` valuta un polinomio in un punto assegnato.

Sperimentazione numerica

(a) Si disegnino i grafici dei polinomi di Lagrange associati ai nodi $\{0, 1/4, 1/2, 3/4, 1\}$ e ai nodi $\{-1, -0.7, 0.5, 2\}$.

(b) Realizzare uno script che calcoli nella forma di Lagrange i polinomi che interpolano le funzioni test $\sin(x)$ e $\cos(x)$ nei punti $x_k = k\pi/2$, con $k = 0, 1, 2, 3, 4$. Visualizzare graficamente i polinomi ottenuti insieme alle funzioni assegnate.

(c) La temperatura T in prossimità del suolo subisce una variazione dipendente dalla latitudine L secondo la seguente tabella:

L	-55	-45	-35	-25	-15	-5	5	15	25	35	45	55	65
T	3.7	3.7	3.52	3.27	3.2	3.15	3.15	3.25	3.47	3.52	3.65	3.67	3.52

Si vuole costruire un modello che descriva la legge $T = T(L)$ anche per latitudini non misurate. A tal fine si scriva uno script che fornisca la variazione di temperatura alle latitudini $L = \pm 42$ utilizzando il polinomio interpolante. Visualizzare in un grafico i dati assegnati, il polinomio interpolante e le stime di T ottenute per $L = \pm 42$.

(d) Scrivere uno script che calcoli il polinomio interpolante un insieme di punti $P_i = (x_i, y_i)$, $i = 0, ..., n$, nella forma di Lagrange con x_i scelti dall'utente come:

- punti equidistanti in un intervallo $[a, b]$,

- punti definiti dai nodi di Chebyshev nell'intervallo $[a, b]$, ossia

$$x_i = \frac{(a+b)}{2} + \frac{(b-a)}{2} \cos\left(\frac{(2i+1)\pi}{2(n+1)}\right), \quad i = 0, ..., n$$

e $y_i = f(x_i)$ ottenuti dalla valutazione nei punti x_i di una funzione test $f : [a, b] \to \mathbb{R}$. Testare lo script sulle funzioni

- $f(x) = \sin(x) - 2\sin(2x), \quad x \in [-\pi, \pi]$,
- $f(x) = \sinh(x), \quad x \in [-2, 2]$,
- $f(x) = |x|, \quad x \in [-1, 1]$,
- $f(x) = 1/(1 + x^2), \quad x \in [-5, 5]$ (funzione di Runge).

Calcolare l'errore di interpolazione $r(x) = f(x) - p(x)$, tra la funzione test $f(x)$ e il polinomio di interpolazione $p(x)$. Visualizzare il grafico di $f(x)$ e $p(x)$, ed il grafico di $|r(x)|$. Cosa si osserva? Cosa accade all'aumentare del grado n di $p(x)$? (Si costruisca una tabella che riporti i valori di $\|r(x)\|_\infty$ al variare di n).

(e) Per $n = 5, 10, 15, 20$ fornire un'approssimazione della costante di Lebesgue scegliendo $x_1, x_2, ..., x_{n+1}$ equispaziati in $[-1, 1]$ oppure coincidenti con i nodi di Chebyshev $x_i = \cos\left(\frac{(2i+1)\pi}{2(n+1)}\right)$, $i = 0, ..., n$.

(f) Si interpolino mediante il polinomio $p_{21}(x)$ i 22 punti (x_i, y_i) con x_i equispaziati in $[-1, 1]$ e $y_i = \sin(2\pi x_i)$. Si considerino poi le ordinate $\tilde{y}_i = y_i + \varepsilon_i$, dove ε_i denota l'i-esima componente del vettore $0.0002 * numpy.random.randn(22,)$, e si calcoli il corrispondente polinomio interpolante $\tilde{p}_{21}(x)$. Si visualizzino e si commentino i risultati ottenuti, calcolando anche l'errore relativo sul polinomio interpolante e sui dati.