

Metodi Numerici per il Calcolo

Esercitazione 5: Interpolazione Polinomiale

A.A.2023/24

Scaricare dalla pagina web del corso l'archivio `matlab_mnc2324_5.zip` e scompattarlo nella propria home directory. Verrà creata una cartella con lo stesso nome contenente alcuni semplici script e function Matlab/Octave. Si svolga la seguente esercitazione che ha come obiettivo quello di sperimentare l'interpolazione polinomiale di dati e funzioni, quindi di punti e curve 2D.

A. Interpolazione polinomiale

1. Interpolazione polinomiale di dati

Si completi lo script `spolint_newt_dati.m`, per l'interpolazione polinomiale nella base di Newton, del set di dati `dataset1.txt`. (Sugg. si utilizzino le function `newton.m`, `lsolve.m` e `newtval.m` presenti nella cartella). Si ripeta l'esercizio completando gli script `spolint_lagr_dati.m` e `spolint_bern_dati.m`, per l'interpolazione polinomiale nelle basi di Lagrange e Bernstein. Si utilizzino le function `lagrval2.m` presente nella cartella, `bernst.m` e `decast_val.m` del toolbox `anmglib_4.0`.

2. Interpolazione polinomiale di funzione

Si completi lo script `spolint_lagr_fun.m` per implementare l'interpolazione polinomiale di grado n di una funzione $f(x)$, $x \in [a, b]$ a partire da $(x_i, f(x_i))_{i=0, \dots, n}$ utilizzando la base di Lagrange. Prevedere due differenti set di punti x_i di interpolazione (equispaziati e di Chebyshev; si utilizzi la function `chebyshev2.m` presente nella cartella). Eseguire più volte per sperimentare l'interpolazione di funzione all'aumentare del grado n e al variare della distribuzione dei punti (equispaziati e punti di Chebyshev):

Si consideri la funzione test di Runge (function `runge.m`):

$$f(x) = 1/(1 + x^2) \quad x \in [-5, 5].$$

3. Interpolazione polinomiale di funzioni nella base di Bernstein

Si completi lo script `spolint_bern_fun.m` simile a `spolint_lagr_fun.m`, ma che utilizzi la forma di Bernstein.

(Sugg. per definire la matrice del sistema lineare si usi la function `bernst.m` e per valutare il polinomio interpolante si utilizzi la function `decast_val.m` del toolbox `anmglib_4.0`).

4. Sulla convergenza dell'interpolante polinomiale

Si modifichi lo script dell'esercizio A.2 (lo si chiami `spolint_lagr_fun_test.m`) per sperimentare l'interpolazione delle seguenti funzioni test all'aumentare del grado n e al variare della distribuzione di punti (equispaziati e punti di Chebyshev).

$$\begin{aligned}\text{fun1.m} \quad & f(x) = \sin(x) - \sin(2x) \quad x \in [-\pi, \pi] \\ \text{fun2.m} \quad & f(x) = \begin{cases} 0.5 & \text{se } x \geq 0 \\ -0.5 & \text{se } x < 0 \end{cases} \quad x \in [-2, 2] \\ \text{fun3.m} \quad & f(x) = e^x \quad x \in [-2, 1].\end{aligned}$$

Per ogni funzione test eseguire il codice più volte con differenti valori del grado e tipo di distribuzione di punti; fare delle considerazioni sulla convergenza dell'interpolante alla funzione.

B. Curve di Bézier e di Bézier a tratti di interpolazione

1. Dato un set di punti 2D lo si vuole interpolare con una curva 2D di Bézier. Si completi lo script `sbezinterp_p2d.m` per realizzare quanto richiesto; si utilizzi la function `curv2.bezier_interp` del toolbox `anmglib.4.0`.
2. Data una curva 2D in forma parametrica la si vuole interpolare con una curva di Bézier. Si completi lo script `sbezinterp_curv2d.m` utilizzando come parametri di interpolazione sia punti equispaziati che punti di Chebyshev. Si determini e analizzi l'errore di interpolazione all'aumentare del numero di punti con cui si campiona la curva analitica.
3. Dato un set di punti 2D (vedi file `paperino.txt`) lo si vuole interpolare con una curva di Bézier cubica a tratti (interpolazione di Hermite). Si consideri la function `curv2.ppbezierCC1_interp` del toolbox `anmglib.4.0` e la si utilizzi per ottenere quanto richiesto. (Sugg. si completi lo script `sppbezierinterp_p2d.m`)
4. Data una curva 2D in forma parametrica la si vuole interpolare con una curva di Bézier cubica a tratti C^1 (interpolazione di Hermite), campionando valori e valori di derivata prima.
Lo script `sppbezierinterp_curve2d_circle.m` vuol far questo per la curva circle (vedi function `cp2_circle.m`), ma la function del toolbox `anmglib.4.0`, `curv2.ppbezierCC1_interp`, non prevede in input i valori di derivata; la vogliamo modificare in:

$$\text{ppP} = \text{curv2_ppbezierCC1_interp_der}(\mathbf{Q}, \mathbf{Q1}, \text{tpar})$$

Come argomenti preveda un array di punti 2D (\mathbf{Q}), un array di vettori tangenti nei punti ($\mathbf{Q1}$) e l'array dei valori parametrici (tpar); l'output sia la struttura della curva di Bézier a tratti. Si richiede di valutare l'errore fra la curva analitica e l'interpolante e di sperimentare lo script per interpolare anche altre curve date in forma analitica.