

Analisi Numerica

A.A. 2020/21 - CdS Matematica LM

Esercitazione sulle Funzioni B-spline

In tutte le esercitazioni di LAB da questa in poi useremo i toolbox **an_spline_lib**, **nurbs** e **GeoPDEs**. Un toolbox (o libreria software) è un insieme di funzioni predefinite e predisposte per essere utilizzate da un programma software attraverso opportune chiamate. Lo scopo dei toolbox è fornire una collezione di funzioni base pronte all'uso evitando al programmatore di dover riscrivere le stesse funzioni e facilitando così le operazioni di sviluppo e mantenimento. Si vedano di seguito le informazioni per scaricarli e installarli; in fondo seguono alcune informazioni per sapere cosa fanno.

Installazione di **an_spline_lib**, **nurbs** e **GeoPDEs** toolbox per MATLAB, impostazione dei path e prime informazioni per utilizzarle

GeoPDEs è un toolbox per MATLAB per l'analisi isogeomtrica. Contiene molte funzioni predefinite che implementano il metodo degli elementi finiti per la soluzione di equazioni differenziali, l'assemblaggio di matrici e/o l'imposizione di condizioni al contorno. **an_spline** è un toolbox per la gestione di funzioni e approssimazione spline. Entrambi questi toolbox si appoggiano sul toolbox **nurbs**, che oltre a mettere a disposizione molte funzionalità per creare e gestire geometrie spline/NURBS mette a disposizione gli algoritmi di base per il calcolo delle funzioni spline.

a) Scaricare il pacchetto GeoPDEs da

<http://rafavzqz.github.io/geopdes/download/>

e decomprimerlo. Verrà creata la cartella **GeoPDEs-full** contenente un manuale e altri file zippati.

b) Copiare la cartella **GeoPDEs-full**, se lo si desidera, in una propria cartella di lavoro e decomprimere tutti i file zippati.

c) Scaricare dalla pagina web del corso l'archivio **an_spline_lib.tgz**, salvarlo/spostarlo nella cartella **GeoPDEs-full**, quindi scompattarlo.

d) Aprire MATLAB, riconoscere la cartella **GeoPDEs-full** come corrente. Quindi creare un nuovo script MATLAB chiamato **GeoPDEs_setpath.m**, che contenga i seguenti comandi:

```
my_path = '/home/user/GeoPDEs-full';  
addpath (genpath (fullfile (my_path)));
```

dove la variabile `my_path` deve essere sostituita con il path del vostro computer. Ogni volta che si avvia MATLAB, si esegua questo script in modo che MATLAB possa accedere alla cartella specificata da `my_path`, ovvero a tutte le funzioni predefinite dei toolbox GeoPDEs, `an_spline` e `nurbs`.

Dalla pagine WEB del corso scaricare l'archivio `anmat_LAB3.zip` e scompattarlo; la cartella omonima contiene alcuni semplici script e funzioni utili per questa esercitazione.

In ambiente Matlab, prima si esegua lo script `GeoPDEs_setpath.m`, quindi si renda corrente quest'ultima cartella; questo permetterà di poter utilizzare le funzioni dei toolbox.

1. La funzione del toolbox `an_bspl_sym` implementa la formula ricorrente sulle funzioni base B-spline Normalizzate in forma simbolica e ne determina l'espressione. Realizzare uno script che la richiami per determinare l'espressione analitica delle funzioni base B-spline dello spazio $S(\mathcal{P}_m, M, \Delta)$, dove $m = 3$, $[a, b] = [0, 4]$, $\Delta = \{1, 2, 3\}$, $M = (1, 1, 1)$ e Δ^* a nodi aggiuntivi coincidenti. Si modifichi poi lo script per verificare in alcuni nodi le condizioni di continuità delle funzioni B-spline di base così determinate.

2. La funzione del toolbox `an_bspl` valuta numericamente le funzioni B-spline Normalizzate in un vettore di punti e restituisce la matrice delle funzioni base nei punti. Lo script `mainBspl_plot` la richiama e rappresenta graficamente le funzioni B-spline Normalizzate di uno spazio $S(\mathcal{P}_m, M, \Delta)$ definito esplicitamente al suo interno.

-si analizzi lo script `mainBspl_plot` e le funzioni che richiama al fine di ritrovare l'algoritmo che viene utilizzato;

-nello script `mainBspl_plot` viene generata una mesh uniforme dell'intervallo di definizione al fine della rappresentazione grafica. Analizzare la function `an_mesh` del toolbox ed utilizzarla per generare una differente mesh di rappresentazione grafica; che caratteristica hanno i punti della mesh così fatta?

-utilizzare lo script `mainBspl_plot` per rappresentare le funzioni base dei seguenti spazi:

- $m = 2, \Delta^* = \{0, 0, 0.25, 0.5, 1, 1\}$;
- $m = 3, \Delta^* = \{0, 0, 0, 0.25, 0.5, 1, 1, 1\}$;
- $m = 3, \Delta^* = \{-0.5, -0.25, 0, 0.25, 0.5, 1, 1.25, 1.5\}$;
- $m = 4, \Delta^* = \{0, 0, 0, 0, 0.25, 0.5, 1, 1, 1, 1\}$;
- $m = 4, \Delta^* = \{0, 0, 0, 0, 0.25, 0.5, 0.5, 1, 1, 1, 1\}$;

- $m = 4$, $\Delta^* = \{0, 0, 0, 0, 0.25, 0.5, 0.5, 0.5, 1, 1, 1, 1\}$;

-utilizzare la funzione `an_bspl` per capire quali valori non nulli restituisce nel caso che il punto di valutazione sia interno ad un intervallo nodale piuttosto che coincidente con un nodo; in quest'ultimo caso considerare casi in cui il nodo sia semplice o multiplo.

3. Nello script `mainBspl_plot` scommentare le istruzioni per la rappresentazione grafica delle B-spline derivata prima e seconda. A tal fine vengono richiamate le funzioni del toolbox `an_bpsspl` che restituisce i valori e valori della derivata prima delle B-spline e `an_bpsspl` che restituisce i valori e valori delle derivate prima e seconda delle B-spline.

Modificare lo script `mainBspl_plot` affinché disegni una singola funzione B-spline insieme alla sua derivata prima e seconda e quindi si verifichino dai grafici le condizioni di continuità nei nodi.

4. Analizzare la funzione del toolbox `an_knot_mult` per capire quali informazione calcola e quindi restituisce se le viene fornita in input una partizione nodale estesa.

5. Si rappresenti il seguente polinomio

$$p(x) = 3 - 4x + 2x^2 \quad x \in [0, 3]$$

nella base delle B-spline normalizzate dello spazio $S(\mathcal{P}_m, M, \Delta)$ con $m = 3$, $[a, b] = [0, 3]$, $\Delta = \{1, 1.5, 2.5\}$ e Δ^* a nodi aggiuntivi coincidenti. E se lo spazio spline fosse di ordine $m = 4$?

6. Si consideri la funzione `mainVD` che, definita una funzione continua f in $[a, b]$ e definito su questo intervallo lo spazio spline $S(\mathcal{P}_m, M, \Delta)$ con Δ^* a nodi aggiuntivi coincidenti, la approssima con la funzione spline

$$s(x) = \sum_{i=1}^{m+K} f(\xi_i) N_{i,m}(x)$$

dove $\xi_i = \frac{1}{m-1} \sum_{j=i+1}^{i+m-1} t_j$. Verificare sperimentalmente che all'aumentare della dimensione dello spazio spline (cioè infittendo la partizione nodale in $[a, b]$ e lasciando l'ordine/grado invariato) la funzione spline $s(x)$ converge alla funzione f su $[a, b]$. Calcolare e stampare il max dell'errore in valore assoluto

$$\max_{x \in [a, b]} |f(x) - s(x)|.$$

(**Sugg.** si infittisca la partizione nodale dividendo a metà ogni intervallo nodale e si consideri come diminuisce l'errore rispetto ad $h = \max_{i=1, \dots, k} (x_i - x_{i-1})$).

Cosa succede invece se si chiama la funzione `mainVD` passando due valori per `mpK` e `g` tali `mpK` sia uguale a `g+1`?

Provare infine ad approssimare altre funzioni.

Introduzione al toolbox `an_spline_lib`

Le funzioni del toolbox `an_spline_lib` permettono un semplice approccio alle funzioni spline; sfruttano le funzioni di calcolo del toolbox `nurbs` e possono essere suddivise in base a cosa fanno: ci sono funzioni di valutazione delle funzioni base B-spline normalizzate e delle loro derivate, valutazione di funzioni spline nel caso univariato e relative utilities, funzioni che implementano vari metodi di interpolazione spline nel caso univariato. Segue la lista delle funzioni disponibili con una breve descrizione, ma per il loro utilizzo si faccia riferimento all'help on-line delle singole funzioni una volta in ambiente Matlab.

B-spline evaluation

`an_bernst`: evaluate the functions of the Bernstein basis

`an_bspl_sym`: determine the functions of the B-spline basis symbolically

`an_bspl`: evaluate basic B-spline functions (call `findspan` and `basisfun`)

`an_bpspl`: evaluate basic B-spline function and their first derivatives (call `basisfunder`)

`an_bpsspl`: evaluate basic B-spline function and their first and second derivatives (call `basisfunder`)

`an_mesh`: define a mesh

`an_knot_mult`: knot partition info

`anSpl1val`: evaluate a B-spline function (ALG-1)

`anSpl2val`: evaluate a B-spline function (ALG-2 due to de Boor)

Univariate spline interpolation methods

`an_interp_SH3`: Hermite C^1 cubic B-spline interpolation

`an_interp_der`: cubic B-spline interpolation with derivative conditions

`an_interp_nak`: cubic B-spline interpolation with not a knot conditions

`an_not_a_knot`: define not a knot partition

`an_interp_nat`: cubic B-spline interpolation with natural conditions

`an_interp_per`: cubic B-spline interpolation with periodic conditions

B-spline geometric tools

`an_knotrem`: knot removal

`an_knotremMQ`: least square knot removal

`anSpl2val_sddiv`: B-spline function subdivision

`an_href`: define an h-refined knot partition

`an_pref`: define a p-refined knot partition

Introduzione al toolbox `nurbs`

Per introdursi al toolbox nurbs si guardi la sezione D del documento `GeoPDEs-report.pdf` presente nella cartella `GeoPDEs-full`, la lista che segue delle funzioni, ma soprattutto l'help on-line delle singole funzioni una volta in ambiente Matlab.

toolbox nurbs:

collection of routines for the creation, and manipulation of Non-Uniform Rational B-Splines (NURBS), based on the NURBS toolbox by Mark Spink.

Basic operations for NURBS curves, surfaces and volumes

nrbmak: construct the NURBS structure given the control points and the knots.

nrbkntins: insert a single or multiple knots into a NURBS curve, surface or volume.

nrbdegelev: elevate the degree of the NURBS curve, surface or volume.

nrbderiv: construct the first and second derivative representation of a NURBS curve, surface or volume.

nrbdeval: evaluation of the derivative and second derivatives of NURBS curve, surface or volume.

nrbeval: evaluate a NURBS at parametric points.

nrbeval_der_w: compute the derivatives of a NURBS object at the point u with respect to the weight of the i -th control point.

nrbeval_der_p: compute the derivative of a NURBS object at a given point with respect to the coordinates of the i -th control point.

nrbinverse: compute parametric point starting from physical point by inverting the NURBS map with a Newton scheme

crvkntremove: remove one knot from the knot-vector of a NURBS curve.

Operations for constructing NURBS curves and surfaces

nrbtform: apply transformation matrix to the NURBS.

nrbreverse: reverse the evaluation directions of a NURBS geometry.

nrbtransp: transpose a NURBS surface, by swapping U and V directions.

nrbpermute: rearrange the directions of a NURBS volume or surface.

nrbline: construct a straight line.

nrbcirc: construct a circular arc.

nrbrect: construct NURBS representation of a rectangular curve.

nrbsquare: create the NURBS surface for a square.

nrb4surf: constructs a NURBS bilinear surface.

nrbcylind: construct a cylinder or cylindrical patch.

nrbextract: construct NURBS curves by extracting the boundaries of a NURBS surface, or NURBS surfaces by extracting the boundary of a NURBS volume.

nrbextrude: construct a NURBS surface by extruding a NURBS curve, or construct a NURBS volume by extruding a NURBS surface.

nrbrevolve: construct a NURBS surface by revolving a NURBS curve, or construct a NURBS volume by revolving a NURBS surface.

nrb ruled: construct a ruled surface between two NURBS curves.
nrb coons: construction of a Coons patch.
nrb test crv: constructs a simple test curve.
nrb test surf: constructs a simple test surface.
nrb clamp: compute the knot vector and control points of the clamped curve/surface.
nrb unclamp: compute the knot vector and control points of the unclamped curve or surface.
nrb multipatch: construct the information for gluing conforming NURBS patches, using the same format as in GeoPDEs.
nrb modp: modify the coordinates of specific control points of any NURBS map.
nrb modw: modify the weights of specific control points of any NURBS map.

Plot and export

nrb plot: plot a NURBS curve or surface, or the boundary of a NURBS volume.
nrb ctrl plot: plot a NURBS entity along with its control points.
nrb knot plot: plot a NURBS entity with the knots subdivision.
nrb export: export NURBS geometries to a format compatible with the one used in GeoPDEs.
nrb 2iges: write a NURBS curve or surface to an IGES file.

B-Spline functions

bsp eval: evaluate B-Spline at parametric points.
bsp deriv: B-Spline derivative.
bsp knot ins: insert knots into a B-Spline.
bsp degree lev: degree elevate a univariate B-Spline.
bsp interp crv: B-Spline interpolation of a 3d curve.
bsp interp surf: B-Spline surface interpolation.
basis fun: basis function for B-Spline.
basis fund: B-Spline Basis function derivatives.
basis knot ins: compute the coefficient matrix for non-uniform B-splines subdivision.
find span: find the span of a B-Spline knot vector at a parametric point.
num basis fun: list non-zero Basis functions for B-Spline in a given knot-span.
t basis fun: compute a B- or T-Spline basis function, and its derivatives, from its local knot vector.

B-splines geometric entities

curv deriv cpts: compute control points of n-th derivatives of a B-spline curve.
curv deriv eval: compute the derivatives of a B-spline curve.
surf deriv cpts: compute control points of n-th derivatives of a NURBS surface.
surf deriv eval: compute the derivatives of a B-spline surface.

NURBS geometric entities and functions

nrbbasisfun: basis functions for NURBS.
nrbmeasure: compute the distance between two given points along a NURBS curve.
nrbbasisfunder: NURBS basis functions derivatives.
nrbnumbasisfun: numbering of basis functions for NURBS.
nrbcrvderieveval: evaluate n-th order derivatives of a NURBS curve.
nrbsurfderievevalL: evaluate n-th order derivatives of a NURBS surface.

Knots construction and refinement

aveknt: compute the knot averages (Greville points) of a knot vector.
kntuniform: generate uniform open knot vectors in the reference domain.
kntrefine: refine a given knot vector by dividing each interval uniformly, maintaining the continuity in previously existing knots.
kntbrkdegreg: construct an open knot vector by giving the sequence of knots, the degree and the regularity.
kntbrkdegmult: construct an open knot vector by giving the sequence of knots, the degree and the multiplicity.

Vector and Transformation Utilities

vecnorm: normalise the vectors.
vecmag: magnitude of the vectors.
vecmag2: squared magnitude of a set of vectors.
vecangle: an alternative to atan, returning an arctangent in the range 0 to 2π .
vecdot: the dot product of two vectors.
veccross: the cross product of two vectors.
vecrot: transformation matrix for a rotation around the axis given by a vector.
vecrotx: transformation matrix for a rotation around the x axis.
vecroty: transformation matrix for a rotation around the y axis.
vecrotz: transformation matrix for a rotation around the z axis.
vecscale: transformation matrix for a scaling.
vectrans: transformation matrix for a translation.