Corso di Approssimazioni Numeriche, AA 2007/08

ESERCITAZIONE DI LABORATORIO SU:

Nodi di Chebyshev, di Chebyshev-Lobatto, di Fekete e di Leja

Prof. S. De Marchi - 16 gennaio 2008

1 Definizioni

Anzitutto vediamo le definizioni dei punti, relativamente all'intervallo $[-1, 1]$.

C : i nodi, o punti, di **Chebyshev** (insieme che indicheremo con la lettera **C**) sono gli zeri del polinomio ortogonale di primo tipo di Chebyshev, $T_n(x)$, di grado n :

$$x_k^{(c)} = \cos\left(\frac{(2k-1)\pi}{2n}\right), \quad k = 1, \dots, n. \quad (1)$$

E : i nodi, o punti, di **Chebyshev-Lobatto** (insieme che indicheremo con la lettera **E**) sono come i punti dell'insieme **C**, ma *estesi* in modo da includere anche gli estremi dell'intervallo

$$x_k^{(e)} = \cos\left(\frac{k\pi}{n-1}\right), \quad k = 0, \dots, n-1. \quad (2)$$

F : i nodi, o punti, di **Fekete** (insieme che indicheremo con la lettera **F**) sono i punti che hanno la seguente proprietà: *massimizzano il modulo del determinante di Vandermonde*. Pertanto, essi si determinano risolvendo il seguente problema di massimo

$$\max_{X_n \in [-1, 1]} |VDM(X_n)|, \quad (3)$$

dove X_n è un qualsiasi insieme di n punti di $[-1, 1]$. Si dimostra che in $[-1, 1]$ i punti di Fekete sono gli zeri del polinomio $(1-x^2)L'_n(x)$, dove L_n indica il polinomio di Legendre di grado n .

L : i nodi, o sequenze, di **Leja** (insieme che indicheremo con la lettera **L**) si costruiscono con il seguente *algoritmo*. Preso $x_1 \in I = [-1, 1]$, x_2 è tale che $|x_2 - x_1| = \max_{x \in I} |x - x_1|$. Quindi x_3 sarà tale che $|(x_3 - x_2)(x_3 - x_1)| = \max_{x \in I} |x - x_1||x - x_2|$ e in generale

$$|(x_{n+1} - x_n) \cdots (x_{n+1} - x_1)| = \max_{x \in I} \prod_{k=1}^n |x - x_k|. \quad (4)$$

2 Esercitazione proposta

Fissato un grado polinomiale n , l'esercitazione consiste di implementare dei pezzi di codice in Octave/Matlab per rispondere alle seguenti domande.

1. Costruirsi delle funzioni in Octave (o in Matlab) che calcolano i punti degli insiemi **C**, **E**, **F**, **L** (per semplicità ci poniamo in $[-1, 1]$). Fare anche un plot comparativo in cui si vede come sono distribuiti i punti degli insiemi **C**, **E**, **F**, **L**. Tutti questi punti hanno cosiddetta *distribuzione dell'arcocoseno*, tipica dei punti di Chebyshev.

Sugg. 1 : per calcolare il massimo del modulo del determinante di Vandermonde usare la funzione Octave/Matlab (dal toolbox `optim`): `fmins` oppure `fminunc`.

Sugg. 2 : per calcolare i punti di Fekete basta applicare le funzioni `r_jacobi.m` e `lobatto.m` del package di quadratura **OPQ** di W. Gautschi al link

<http://www.cs.purdue.edu/archives/2002/wxg/codes/OPQ.html>.

Come verifica, per calcolare il polinomio di Legendre e la sua derivata prima forniamo in allegato la funzione `PolLegendre.m` che calcola i punti di Fekete e plotta la derivata prima del polinomio di Legendre

2. Verificare che i punti di Fekete effettivamente massimizzano il modulo del determinante di Vandermonde (usare la funzione Octave/Matlab `vander`).
3. Ricordando la definizione della *costante di Lebesgue*

$$\Lambda_n := \max_{-1 \leq x \leq 1} \sum_{k=1}^n |l_k(x)|,$$

fare un plot comparativo dell'andamento delle costanti di Lebesgue degli insiemi **C**, **E**, **F**, **L**.

Per comodità accludiamo due utili funzioni allo scopo di "accelerare" le implementazioni. Ovvero le funzioni `PolLegendre` e `PolLagrange` descritte al loro interno.

```
function [L x L1] = PolLegendre(N,a,b)
%
% restituisce le matrici L, L1 con
%
% l_ij = P_{i-1}(x(j))
% l1_ij = P'_{i-1}(x(j))
%
% x e' il vettore riga dei nodi di Gauss--Legendre--Lobatto
% nell'intervallo [a,b], con P_i(x) polinomio di Legendre di
% grado i valutato in x, P'_i(x) derivata prima del polinomio
% di Legendre di grado i valutata in x
%-----
ab = r_jacobi(N);
```

```

x = lobatto(N-2,ab,-1,1); % i punti di Fekete in x(1,:)
x = x(:,1)*(b-a)/2+(a+b)/2

if (N == 1)
    return
end

% Plot di verifica

M=1000;
xx=linspace(a,b,M)';
L = ones(N,length(xx));
L(2,:)= xx;

L1 =zeros(N,length(xx));
L1(2,:)=ones(1,M);

for i = 2:N-1
    L(i+1,:) = ((2*(i-1)+1).*xx'.*L(i,)-(i-1).*L(i-1,:))./i;
    L1(i+1,:) = ((2*(i-1)+1)*(L(i,)+xx'.*L1(i,))-(i-1)*L1(i-1,:))./i;
end

plot(x,zeros(1,N),'o',xx,L1(N,:))

%-----

function l=PolLagrange(x,z,s)
%-----
% Calcola l's-esimo pol. elementare di Lagrange
% x = punto su cui valutare
% z = nodi di interpolazione
% s = indice del polinomio
%-----
n=length(z);
l=1;
for j=1:n,
    if (j ~= s),
        l=l.*(x-z(j))/(z(s)-z(j));
    end;
end;
return

```

Tempo assegnato: 2 ore.

3 Approfondimenti

Indichiamo alcuni importanti riferimenti per poter approfondire e ampliare la conoscenza degli argomenti trattati nell'esercitazione. Lo scopo è di stimolare l'interesse dello studente e che potranno diventare anche un possibile argomento per gli approfondimenti richiesti nel corso.

3.1 Punti di Chebyshev, Chebyshev estesi e di Fekete

Consiglio la lettura del lavoro di L. Brutman *Lebesgue function for polynomial interpolation - a survey*, Ann. Numer. Math. 4 (1997), 111-127. Si tratta di un lavoro in cui sono raccolte e discusse le informazioni storiche nonché le proprietà degli insiemi di punti di Chebyshev, Chebyshev estesi e di Fekete. Il lavoro ha anche un'ampia e completa bibliografia.

Proprio in questi giorni sono stati anche scoperti i *punti di Fekete approssimati* che si ottengono (in maniera efficiente) usando il semplice codice Matlab

```
n=input('Numero punti = ');
x=linspace(-1,1,1000); %prendiamo 1000 punti equispaziati
A=gallery('chebvand',n,x);
b=rand(n,1);
y=A\b;
p=find(y~=0);
pts=x(p); % i punti approssimati di Fekete
```

Far vedere, sperimentalmente, che si tratta di approssimazioni dei punti di Fekete. L'interesse di questo algoritmo, mai prima d'ora usato, è che si tratta di punti che si ottengono risolvendo un sistema lineare con il metodo QR (la cui complessità è dell'ordine di n^3).

3.2 Sequenze di Leja

Per quanto riguardano le sequenze di Leja, consiglio il seguente lavoro su: S. De Marchi, *Leja sequences: some results and applications* Appl. Math. Comput. 152(3) (2004), 621-647.

Qui vale la pena ricordare che i punti di Leja $\{x_i\}$ possono essere *estratti* da una discretizzazione di un generico intervallo $[a, b]$ anche in modo rapido originando i cosiddetti "*fast Leja points*" di Baglama & al., ETNA 7,(1998), 124-140;

Inoltre, i punti di Leja hanno un'importante proprietà: sono una sequenza stabile per l'interpolante in forma di Newton (cf. L. Reichel, *Newton Interpolation at Leja Points*, BIT Numerical Analysis 30 (1990), 332-346).