

Teoria dell'Approssimazione e Applicazioni, A.A. 2014/15

ESERCITAZIONE DI LABORATORIO DEL 12 MARZO 2015

Prof. Stefano De Marchi

1. Costruire il polinomio d'approssimazione di Bernstein

$$B_n(f)(x) = \sum_{k=0}^n f\left(\frac{k}{n}\right) \binom{n}{k} x^k (1-x)^{n-k}$$

per la funzione $f(x) = x(1-x)$, $x \in [0, 1]$. Far vedere graficamente che $\lim_{n \rightarrow \infty} B_n(f) = f$.

2. Calcolare la funzione e la costante di Lebesgue dei punti di Chebyshev e di Fekete di $[-1, 1]$, per $n = 3, \dots, 50$. Per costruire i polinomi di Lagrange, la corrispondente matrice, $\lambda_n(x)$ e Λ_n usare le funzioni del Cap. 2.
3. Usando l'algoritmo di chebfun <http://www.chebfun.org/examples/approx/BestApprox.html> calcolare il BPA di $f(x) = x^{k+1}$ su $[-1, 1]$ per $k = 2, 4, 6$. Come BPA usare $n \leq k$. Ripetere la stessa cosa con $f(x) = \sin((2k+1)x)$.

```
%-----
function [p,err] = remez(f,n); % compute deg n BA to chebfun f
iter = 1; delta = 1; deltamin = delta;
[a,b] = domain(f);
xk = chebpts(n+2); xo = xk; % initial reference
sigma = (-1).^[0:n+1]; % alternating signs
normf = norm(f);
while (delta/normf > 1e-14) & iter <= 20
    fk = feval(f,xk); % function values
    w = bary_weights(xk); % compute barycentric weights
    h = (w*fk)/(w*sigma); % levelled reference error
    if h==0, h = 1e-19; end % perturb error if necessary
    pk = fk - h*sigma; % polynomial vals in the
    p=chebfun(@(x)bary(x,pk,xk,w),n+1); % reference chebfun of trial
    e = f - p; % polynomial chebfun of the
    [xk,err] = exchange(xk,e,h,2); % error replace reference
    if err/normf > 1e5 % if overshoot, recompute with
    [xk,err] = exchange(xo,e,h,1); % one-point exchange
end
xo = xk;
delta = err - abs(h); % stopping value
if delta < deltamin, % store poly with minimal norm
deltamin = delta;
pmin = p; errmin = err;
end
iter = iter + 1;
end
p = pmin; err = errmin;
%-----
```