# Esame di Analisi Numerica

*Laurea Magistrale in Statistica ed Informatica*

*Prof. S. De Marchi*

`Padova, 15 dicembre 2009`

Il candidato dovrà scrivere su **ogni** foglio il cognome, nome, numero di matricola. Consegnare fogli leggibili!. Inviare poi tutti i files a `demarchi@math.unipd.it` **NOTA: non allegate immagini in formato Matlab .fig ma .jpg o .eps.**

1. Si consideri la funzione $f(x) = \frac{1}{2} - \frac{1}{x^2+1}$ di cui si vogliamo trovare gli zeri.

   - Individuare le due radici reali $\alpha_1$ e $\alpha_2$ di $f$ ed i corrispondenti intervalli separatori (che denoteremo con $I_{\alpha_1}$ e $I_{\alpha_2}$). Far vedere che $\alpha_1 + \alpha_2 = 0$?
   - Scelto $I_{\alpha_1}$, si determini a priori quante iterazioni sono necessarie per determinare $\alpha_1$ con il metodo di bisezione con $tol = 1.e-8$.
   - Calcolare numericamente $\alpha_1$ e $\alpha_2$.

2. Data la matrice

   `A=diag(10*ones(7,1))+diag(3*ones(6,1),1)+diag(3*ones(6,1),-1)` e termine noto $b$ cosicché la soluzione sia `x=[1,zeros(1,4),1,1]'`.

   - Dire perché convergono i metodi iterativi di Jacobi e Gauss-Seidel.
   - Fissata la tolleranza $\epsilon = 1.e{-}9$, sia $P$ la matrice d'iterazione con $\|P\| < 1$. Risolvendo

     $$\frac{\|P\|^k}{1 - \|P\|} \|x^1 - x^0\| < \epsilon$$

     possiamo determinare *a priori* il numero di iterazioni $k$ necesarie ad ottenere una soluzione a meno di $\epsilon$. Partendo dalla soluzione iniziale **x=0** e usando la norma infinito $\| \cdot \|_\infty$, determinare il numero $k$ sia per il metodo di Jacobi che per quello di Gauss-Seidel.
   - Infine, si calcoli numericamente la soluzione del sistema $Ax = b$, con i metodi di Jacobi e Gauss-Seidel.

Tempo: **2 ore**.

## SOLUZIONI

### Primo esercizio

```
% --------------
% Esercizio 1
%---------------
disp('Primo esercizio')
% 1) La funzione  pari e corrisponde all'opposto della
%    funzione di Runge  traslata
%    rispetto all'asse y di 1/2. Ci aspettiamo quindi di
%    avere 2 zeri visto che la traslazione di 1/2  sufficiente
%    alla funzione per tagliare le ascisse.
%    In ogni caso, la cosa si verifica analiticamente e banalmente.
%    Infatti,
%              f(x)=(x^2-1)/(2*(x^2+1))=0 se e solo se alfa1=-1, alfa2=1.

  x=linspace(-2,2,1000);
  y=1/2-1./(x.^2+1);
  plot(x,y); grid;
  pause(3)

% 2) Deriva della parita' di f
% 3) Sia ora I_alfa1=[-2,-1/2]. Per sapere a priori con la bisezione
% quante iterazioni fare, basta risolvere la disuguaglianza
%     |b-a|/2^(k+1) < epsilon ovvero   k > log_2(|b-a|/espilon)-1
   disp(' Le iterazioni da fare con il metodo di bisezione sono ')
   tol=1.e-8;
   k=floor(log2(abs(b-a)/tol))-1

 % 4) Troviamo la radice con il metodo di bisezione per verificare anche
 % il numero d'iterazioni a priori. La funzione bisezione1.m e' una piccola
 % variante di bisezione.m presentata in classe.

 fun=inline('1/2 - 1/(x^2+1)');
 b=-2; a=-1/2;
 [sol, k]=bisezione1(fun,a,b,tol)
```

Eseguendo il programma, si trova che a priori servirebbero 26 iterazioni, mentre in effetti se ne fanno 24 perchè una sovrastima.

**Secondo esercizio**

```
clear;
format long
%---------------
% Esercizio 2
%---------------
disp('Secondo esercizio')

% Parte (i)
% La risposta e' ovvia: la matrice e' tridiagonale,
% diagonalmente dominante quindi Jacobi e Gauss-Seidel
% convergono.
% Come verifica ulteriore calcoliamo la norma infinito
% di entrambe le matrici J, di Jacobi e GS, di Gauss- Seidel.

% matrice
A = diag(ones(7,1)*10)+diag(ones(6,1)*3,+1)+diag(ones(6,1)*3,-1);
% termine noto
b = A*[1, zeros(1,4) 1 1]';

% Matrice di Jacobi e sua norma infinito
 D=diag(diag(A)); L=tril(A)-D; U=triu(A)-D;
 J = -(inv(D))*(L+U);
 normJ=norm(J,inf)

% Matrice di Gauss_Seidel e sua norma infinito

D=diag(diag(A)); L=tril(A)-D; U=triu(A)-D; GS=-inv((D+L))*U;
normGS=norm(GS,inf)
```

Facendo eseguire il codice appena scritto otteremo i seguenti valori per `normJ` e `normGS`

```
normJ =

   0.60000000000000

normGS =

   0.42825900000000
```

Per rispondere alla seconda parte, posto $\epsilon = 1.e - 9$, ricaviamo $k$ dalla (**??**) risolvendo la disequazione

$$k \log(\|P\|) \leq \log\left(\frac{\epsilon(1-\|P\|)}{\|x^1 - x^0\|}\right) \ .$$

Ricordando che se $\|P\| \leq 1$ (come accade nel caso di convergenza!) allora $\log(\|P\|) < 0$, si ottiene

$$k \geq \frac{\log\left(\frac{\epsilon(1-\|P\|)}{\|x^1 - x^0\|}\right)}{\log(\|P\|)} \ . \tag{1}$$

Ora sostituendo in (1) la matrice $P$ con la matrice $J$ e $GS$ e $x^1$ con la prima iterata del metodo di Jacobi e Gauss-Seidel, si ottiene

$$\begin{array}{ll} k_j \geq 42.9 & con\ Jacobi \\ \\ k_{GS} \geq 25.3 & con\ Gauss-Seidel \end{array} \ .$$

Inoltre, essendo $x_0 = 0$ $\|x^1\|_\infty = 1.3$ con Jacobi e $\|x^1\|_\infty = 1.21$ con Gauss-Seidel.

Infine per la parte (c) basta applicare le funzioni Matlab implementate in classe per il calcolo della soluzione con i metodi di Jacobi e Gauss-Seidel ottendo

```
%
% Jacobi
%
xn =

    1.0000
    0.0000
    0.0000
    0.0000
    0.0000
    1.0000
    1.0000


iter =

    37
%
% Gauss-Seidel
%
 xn =

    1.0000
```

```
    0.0000
   -0.0000
    0.0000
   -0.0000
    1.0000
    1.0000


iter =

    17
```

Come si nota, il numero di iterazioni ottenute sperimentalmente, è inferiore ai valori stimati. Ma questo è corretto poiché (1) è una sovrastima determinata usando l'iterazione $x^1$.