Capitolo 1

Esercizi a.a. 2016-17

Esercizi

Esercizio 1.1 *Dimostrare che il metodo iterativo*

$$x_{k+1} = \Phi(x_k), \qquad k = 0, 1, \ldots,$$

se convergente a x^*, deve verificare la condizione di consistenza

$$x^* = \Phi(x^*).$$

Ovvero, la soluzione cercata deve essere un punto fisso *per la funzione di iterazione che definisce il metodo.*

Esercizio 1.2 *Il codice Fortran*

```
program INTERO
integer*2 numero, i
numero = 32765
do i = 1, 6
   write(*,*) i, numero
   numero = numero +1
end do
write(*,*) '--------'
do i = 1, 6
   write(*,*) i, numero
   numero = numero-1
end do
end
```

produce il seguente output*:*

```
     1  32765
     2  32766
     3  32767
     4 -32768
     5 -32767
     6 -32766
--------
     1 -32765
     2 -32766
     3 -32767
     4 -32768
     5  32767
     6  32766
```

Spiegarne il motivo.

Esercizio 1.3 *Sia data un'aritmetica finita che utilizza la base $b = 8$, con 5 cifre per la mantissa, e che implementa l'arrotondamento. Calcolare la corrispondente precisione di macchina.*

Esercizio 1.4 *Abbiamo visto che, per funzioni sufficientemente regolari,*

$$\Psi_h(x) := \frac{f(x+h) - f(x)}{h} = f'(x) + O(h).$$

Utilizzando Matlab, riempire la seguente tabella, utilizzando il formato `format long e`*, con $f(x) = \mathrm{e}^x$ e fissando $x = 0$. In questo modo dovrebbe aversi $\Psi_h(0) = 1 + O(h)$. Spiegare, quindi, i risultati ottenuti.*

h	$\Psi_h(0)$
10^{-1}	
10^{-2}	
10^{-3}	
10^{-4}	
10^{-5}	
10^{-6}	
10^{-7}	
10^{-8}	
10^{-9}	
10^{-10}	
10^{-11}	
10^{-12}	

Esercizio 1.5 *Dimostrare che, se $f(x)$ è sufficientemente regolare e $h > 0$ è una quantità "piccola", allora:*

$$\frac{f(x_0+h)-f(x_0-h)}{2h} = f'(x_0) + O(h^2),$$

$$\frac{f(x_0+h)-2f(x_0)+f(x_0-h)}{h^2} = f''(x_0) + O(h^2).$$

Esercizio 1.6 *Il metodo iterativo*

$$x_{n+1} = \frac{x_n x_{n-1} + 2}{x_n + x_{n-1}}, \quad n = 1, 2, \dots, \qquad x_0 = 2, \ x_1 = 1.5,$$

definisce una successione di approssimazioni convergente a $\sqrt{2}$. Calcolare a quale valore di n bisogna arrestare l'iterazione, per avere un errore di convergenza $\approx 10^{-12}$.

Esercizio 1.7 *Quante cifre binarie sono utilizzate per rappresentare, mediante arrotondamento, la mantissa di un numero, sapendo che la precisione di macchina è $u \approx 4.66 \cdot 10^{-10}$?*

Esercizio 1.8 *Dimostrare che, detta u la precisione di macchina utilizzata,*

$$-\log_{10} u$$

fornisce, approssimativamente, il numero di cifre decimali correttamente rappresentate nella mantissa.

Esercizio 1.9 *Eseguire le seguenti istruzioni Matlab:*

```
x = 0; delta = 1/10;
while x ~= 1, x = x+delta, end
```

Spiegarne il (non) funzionamento.

Esercizio 1.10 *Individuare l'algoritmo più efficace per calcolare, in aritmetica finita, l'espressione $\sqrt{x^2+y^2}$.*

Esercizio 1.11 *Eseguire l'analisi dell'errore (relativo), dei due seguenti algoritmi per calcolare la somma di tre numeri:*

$$1) \ \ (x \oplus y) \oplus z, \qquad\qquad 2) \ \ x \oplus (y \oplus z).$$

Esercizio 1.12 *Dimostrare che il numero di condizionamento del problema del calcolo di $y = \sqrt{x}$ è $\kappa = \frac{1}{2}$.*

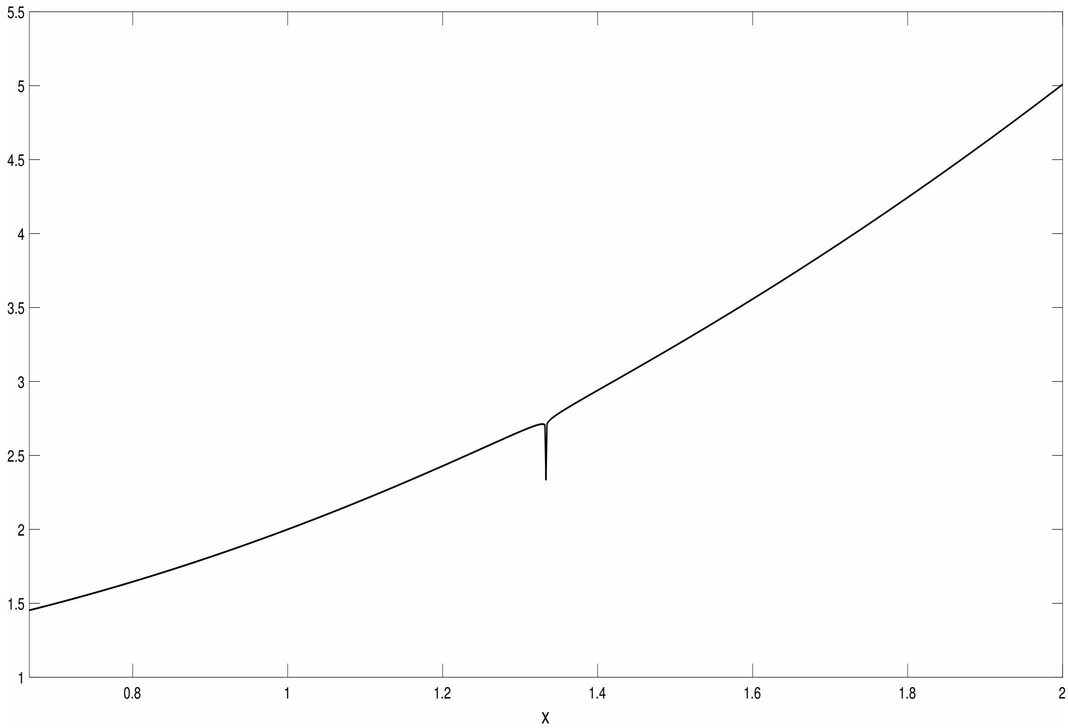

Figura 1.1: grafico di della funzione (1.1) con 1001 punti equispaziati nell'intervallo $[2/3, 2]$, doppia precisione IEEE.

Esercizio 1.13 *Utilizzare la doppia precisione IEEE per graficare la funzione*[1]

$$f(x) = \frac{\ln\left(|3(1-x)+1|\right)}{80} + x^2 + 1, \qquad x \in \left[\frac{2}{3}, 2\right]. \tag{1.1}$$

Utilizzando 1001 punti equispaziati nell'intervallo, $fl(\frac{4}{3})$ *è tra questi. Nondimeno, si ottiene la Figura 1.1, da cui si evince che il minimo della funzione è in* $x = \frac{2}{3}$*, sebbene in* $\frac{4}{3}$ *vi sia un asintoto verticale e, evidentemente,*

$$f(x) \longrightarrow -\infty, \qquad x \to \frac{4}{3}.$$

Più precisamente, calcolando $f(x)$ *in*

$$fl\left(\frac{4}{3}\right) = 1.333333333333333,$$

si ottiene 2.327232110413813*. Spiegare il perché di questo risultato.*

[1] Si ringrazia John L. Gustafson, per questo esempio.

Capitolo 2

Esercizi a.a. 2016-17

Esercizi

Esercizio 2.1 *Definire una procedura iterativa basata sul metodo di Newton per determinare* $\sqrt{\alpha}$*, per un assegnato* $\alpha > 0$*. Costruire una tabella delle approssimazioni relativa al caso* $\alpha = x_0 = 3$*.*

Esercizio 2.2 *Generalizzare il risultato del precedente esercizio, derivando una procedura iterativa basata sul metodo di Newton per determinare* $\sqrt[n]{\alpha}$*, per un assegnato* $\alpha > 0$*. Implementare i casi per* $n = 3, 4, 5$*, per* $\alpha = 3$*.*

Esercizio 2.3 *In analogia con quanto visto nell'Esercizio 2.1, definire una procedura iterativa basata sul metodo delle secanti per determinare* $\sqrt{\alpha}$*. Confrontare con quanto richiesto nell'Esercizio 2.1.*

Esercizio 2.4 *Comparare il metodo di Newton, il metodo di Newton modificato ed il metodo di accelerazione di Aitken, per approssimare gli zeri delle funzioni*

$$f_1(x) = (x - \pi)^{10}, \qquad f_2(x) = (x - \pi)^{10} \mathrm{e}^{2x},$$

per valori decrescenti della tolleranza `tolx`*. Utilizzare, in tutti i casi, il punto iniziale* $x_0 = 5$*.*

Esercizio 2.5 *È possibile, nel caso delle funzioni del precedente esercizio, utilizzare il metodo di bisezione per determinarne lo zero?*

Esercizio 2.6 *Costruire una tabella in cui si comparano, a partire dallo stesso punto iniziale* `x0` $= 0$*, e per valori decrescenti della tolleranza* `tolx`*, il numero di iterazioni richieste per la convergenza dei metodi di Newton, corde e secanti, utilizzati per determinare lo zero della funzione*

$$f(x) = 1 - x - \left(1 + \frac{\cos 10x}{2}\right) \sin x.$$

Calcolare, quindi, il numero di condizionamento di questa radice.

Esercizio 2.7 *Completare i confronti del precedente esercizio inserendo quelli con il metodo di bisezione, con intervallo di confidenza iniziale* $[0, 1]$.

Esercizio 2.8 *Utilizzare il metodo di Newton per determinare la radice della funzione*

$$f(x) = (x - \pi)\mathrm{e}^{10x},$$

partendo dal punto iniziale $x_0 = 0$. *Commentare i risultati ottenuti.*

Capitolo 3

Esercizi a.a. 2016-17

Esercizi

Esercizio 3.1 *Dimostrare che la somma ed il prodotto di matrici triangolari inferiori (superiori), è una matrice triangolare inferiore (superiore).*

Esercizio 3.2 *Dimostrare che il prodotto di due matrici triangolari inferiori (superiori) a diagonale unitaria è a sua volta una matrice triangolare inferiore (superiore) a diagonale unitaria.*

Esercizio 3.3 *Dimostrare che la matrice inversa di una matrice triangolare inferiore (superiore) è a sua volta triangolare inferiore (superiore). Dimostrare inoltre che, se la matrice ha diagonale unitaria, tale è anche la diagonale della sua inversa.*

Esercizio 3.4 *Dimostrare che il numero di* `flop` *richiesti dall'algoritmo di fattorizzazione LU è circa* $2/3n^3$*, se il problema ha dimensione* n *(considerare che* $\sum_{i=1}^{n-1} i^2 = n(n-1)(2n-1)/6$ *e* $\sum_{i=1}^{n-1} i = n(n-1)/2$*).*

Esercizio 3.5 *Scrivere una* function *Matlab che implementi efficientemente l'algoritmo di fattorizzazione LU con pivoving parziale.*

Esercizio 3.6 *Scrivere una* function *Matlab che, avendo in ingresso la matrice* `A` *riscritta dall'algoritmo di fattorizzazione LU con pivoting parziale, ed il vettore* `p` *delle permutazioni, ed un vettore* **b** *contenente i termini noti del sistema lineare, ne calcoli efficientemente la soluzione.*

Esercizio 3.7 *Dimostrare che, se* A *è nonsingolare, le matrici* $A^\top A$ *e* $A\,A^\top$ *sono sdp.*

Esercizio 3.8 *Dimostrare che se* $A \in \mathbb{R}^{m\times n}$, *con* $m \geq n = \mathrm{rank}(A)$, *allora la matrice* $A^\top A$ *è sdp.*

Esercizio 3.9 *Data una matrice* $A \in \mathbb{R}^{n\times n}$, *dimostrare che essa può essere scritta come*

$$A = \frac{1}{2}(A + A^\top) + \frac{1}{2}(A - A^\top) \equiv A_s + A_a,$$

dove $A_s = A_s^\top$ *è detta* parte simmetrica *di* A, *mentre* $A_a = -A_a^\top$ *è detta* parte antisimmetrica *di* A. *Dimostrare inoltre che, dato un generico vettore* $\mathbf{x} \in \mathbb{R}^n$, *risulta*

$$\mathbf{x}^\top A\mathbf{x} = \mathbf{x}^\top A_s\mathbf{x}.$$

Esercizio 3.10 *Dimostrare che il numero di* `flop` *richiesti dall'algoritmo di fattorizzazione* $LDL^\top$ *è circa* $n^3/3$, *se* n *è la dimensione della matrice.*

Esercizio 3.11 *Scrivere una* function *Matlab che implementi efficientemente l'algoritmo di fattorizzazione* $LDL^\top$ *per matrici sdp.*

Esercizio 3.12 *Scrivere una* function *Matlab che, avendo in ingresso la matrice* `A` *prodotta dalla precedente function, contenente la fattorizzazione* $LDL^\top$ *della matrice sdp originaria, ed un vettore di termini noti,* `x`, *calcoli efficientemente la soluzione del corrispondente sistema lineare.*

Esercizio 3.13 *Utilizzare la* function *dell'Esercizio 3.11 per verificare se le matrici*

$$A_1 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 \\ 1 & 2 & 3 & 3 \\ 1 & 2 & 3 & 4 \end{pmatrix}, \qquad A_2 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 \\ 1 & 2 & 3 & 3 \\ 1 & 2 & 3 & 2 \end{pmatrix}$$

sono sdp.

Esercizio 3.14 *Costruire alcuni esempi di applicazione delle* function *degli Esercizi 3.5-3.6 e 3.11-3.12. Verificare che la soluzione ottenuta sia corretta, valutando il corrispondente residuo* $\mathbf{r} = A\mathbf{x} - \mathbf{b}$

Esercizio 3.15 *Si consideri la seguente matrice* bidiagonale inferiore,

$$A = \begin{pmatrix} 1 & & & \\ -100 & 1 & & \\ & \ddots & \ddots & \\ & & -100 & 1 \end{pmatrix}_{10\times 10}.$$

Calcolare $\kappa_\infty(A)$. *Confrontare il risultato con quello fornito dalla* function `cond` *di Matlab. Dimostrare, e verificare, che* $\kappa_\infty(A) = \kappa_1(A)$.

Esercizio 3.16 *Si considerino i seguenti vettori di* $\mathbb{R}^{10}$,

$$\mathbf{b} = \begin{pmatrix} 1 \\ -99 \\ \vdots \\ -99 \end{pmatrix}, \qquad \mathbf{c} = 0.1 \cdot \begin{pmatrix} 1 \\ -99 \\ \vdots \\ -99 \end{pmatrix},$$

ed i seguenti sistemi lineari

$$A\mathbf{x} = \mathbf{b}, \qquad A\mathbf{y} = \mathbf{c},$$

in cui A è la matrice definita nel precedente Esercizio 3.15. Verificare che le soluzioni di questi sistemi lineari sono, rispettivamente, date da:

$$\mathbf{x} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}, \qquad \mathbf{y} = \begin{pmatrix} 0.1 \\ \vdots \\ 0.1 \end{pmatrix}.$$

Confrontare questi vettori con quelli calcolati dalle seguenti due serie di istruzioni Matlab,

```
b=[1 -99*ones(1,9)]';
x(1)=b(1); for i=2:10, x(i)=b(i)+100*x(i-1); end
x=x(:)

c=0.1*[1 -99*ones(1,9)]';
y(1)=c(1); for i=2:10, y(i)=c(i)+100*y(i-1); end
y=y(:)
```

che implementano, rispettivamente, le risoluzioni dei due sistemi lineari. Spiegare i risultati ottenuti.

Esercizio 3.17 *Scrivere una* function *Matlab che implementi efficientemente l'algoritmo di fattorizzazione QR, mediante il metodo di Householder.*

Esercizio 3.18 *Scrivere una* function *Matlab che, avendo in ingresso la matrice* `A` *prodotta dalla* function *del precedente esercizio, contenente la fattorizzazione QR della matrice originaria, e un corrispondente vettore di termini noti* **b**, *calcoli efficientemente la soluzione del corrispondente sistema lineare sovradeterminato.*

Esercizio 3.19 *Utilizzare le* function *degli Esercizi 3.17 e 3.18 per calcolare la soluzione ai minimi quadrati del sistema lineare* $A\mathbf{x} = \mathbf{b}$, *e la norma del corrispondente residuo, nel caso in cui*

$$A = \begin{pmatrix} 3 & 2 & 1 \\ 1 & 2 & 3 \\ 1 & 2 & 1 \\ 2 & 1 & 2 \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} 6 \\ 6 \\ 4 \\ 4 \end{pmatrix}.$$

Esercizio 3.20 *Risolvere il sistema di equazioni nonlineare*

$$x_2 - \cos(x_1) = 0, \qquad x_1 x_2 - \frac{1}{2} = 0,$$

mediante il metodo di Newton, partendo dal punto iniziale $(1, 1)$. *Tabulare i risultati ottenuti, riportando l'approssimazione e la norma dell'incremento a ogni iterata.*

Esercizio 3.21 *Determinare il punto di minimo della funzione* $f(x_1, x_2) = x_1^4 + x_1(x_1 + x_2) + (1 + x_2)^2$, *utilizzando il metodo di Newton per calcolarne il punto stazionario, partendo dal punto iniziale* $(0, 0)$.

ESERCIZI DA SVOLGERE PER L' ELABORATO
CAP 4
CN A.A. 16/17

1. Scrivere una function Matlab che, presi in input due vettori $\mathbf{x}$ e $\mathbf{f}$ di $n+1$ componenti e preso un vettore di tabulazione $xval$ di $nval$ componenti, fornisce in output un vettore $Pval$ di $nval$ componenti tale che $Pval_j$ contiene il valore assunto in $xval_j$ dal polinomio di grado $\leq n$ interpolante le coppie $(x_i, f_i), i = 1, \ldots, n+1$ (usare la forma di Newton, richiamare la function per il calcolo delle differenze divise e usare l'algoritmo di Horner per la valutazione).

2. Testare il funzionamento della precedente function per interpolare la funzione $f(x) = \frac{1}{1+x^2}$ nell'intervallo $[-5\,,\,5]$. Utilizzare, per n crescente (pari) con $n \leq 20$, sia ascisse equispaziate che ascisse generalizzate di Chebyshev e confrontare i risultati (fare una tabella con l'errore per i vari valori di n e confrontare il grafico di $f(x)$ con quello del polinomio interpolante per qualche valore di n). Ripetere il test con la funzione $f(x) = x \sin x$ nell'intervallo $[0\,,\pi]$ (fare uno script per organizzare la sperimentazione).

3. Scrivere una function Matlab che, presi in input due vettori $\mathbf{x}$ e $\mathbf{f}$ di $n+1$ componenti, con $\mathbf{x}$ ad elementi distinti ed f_i che indica il valore assunto in x_i da una funzione $f(x)$, fornisce in output il vettore dei momenti $\mathbf{M}$ di $n+1$ componenti relativo alla spline cubica naturale interpolante nei nodi (usare l'algoritmo di fattorizzazione LU specifico per una tridiagonale a diagonale dominante e espandere $\mathbf{M}$ inserendovi una prima e ultima componente nulle) e i vettori $\mathbf{r}$ e $\mathbf{q}$ di n componenti necessari per costruire in forma polinomiale a tratti la spline cubica s_3 naturale avente $\mathbf{x}$ come nodi e tale che $s_3(x_i) = f_i, i = 1, \ldots, n+1$.

4. Scrivere una function Matlab che, presi in input tre vettori $\mathbf{x}, \mathbf{f}$ e $\mathbf{M}$ di $n+1$ componenti e preso un vettore di tabulazione $xval$ di $nval$ componenti, fornisce in output un vettore $sval$ di $nval$ componenti tale che $sval_j$ contiene il valore assunto in $xval_j$ da una spline cubica s_3 avente $\mathbf{x}$ come vettore dei nodi e tale che $s_3(x_i) = f_i, i = 1, \ldots, n+1$ e $s_3''(x_i) = M_i, i = 1, \ldots, n+1$.

5. Testare le precedenti functions per interpolare con una spline naturale la funzione $f(x) = \frac{1}{1+x^2}$ nell'intervallo $[-5\,,\,5]$. Utilizzare, per n crescente (pari) con $n \leq 20$, ascisse equispaziate e confrontare i risultati con quelli ottenuti con la spline cubica not–a–knot definita mediante il comando `spline` (fare una tabella con l'errore per i vari valori di n e confrontare il grafico della spline naturale con quello di $f(x)$ e con quello della spline not–a–knot per qualche valore di n). Ripetere il test con la funzione $f(x) = x \sin x$ nell'intervallo $[0\,,\pi]$ (fare uno script per organizzare la sperimentazione).

6. FAC/GR3

   Scrivere una function gemella di quella costruita per il punto 3, che fornisce in output i momenti della spline cubica periodica (usare lo specifico algoritmo di fattorizzazione LU).

7. FAC/GR3

   Testare la precedente function per interpolare con una curva spline parametrica chiusa una sequenza di punti assegnati nel piano $\mathbf{P}_i\,,\ i = 1,\ldots,n$, con $\mathbf{P}_n = \mathbf{P}_1$. Utilizzare, a tale scopo dei parametri $0 = t_1 < \ldots < t_n = 1$ tali che $(t_{i+1} - t_i)$ sia proporzionale alla distanza fra $\mathbf{P}_i$ e $\mathbf{P}_{i+1}$.

8. Costruire una function Matlab che calcola il polinomio di migliore approssimazione di grado $m < n$ delle coppie $(x_i, y_i), i = 1,\ldots,n+1$, dopo aver verificato che esistono almeno $m+1$ ascisse distinte (in caso contrario fornisce un messaggio di errore) e il corrispondente valore della funzione obiettivo minimizzata. A tale scopo assemblare la matrice di Vandermonde rettangolare e utilizzare il `backslash`.

9. Sperimentare la function precedente mediante uno script in cui si definiscono i dati per due tests: per entrambi si pone $n = 10$ e si definiscono le ascisse equispaziate in $[-1\,,\,1]$ ma nel primo si pone $m = 1$, e si definiscono le corrispondenti ordinate ponendo $y_i = 5x_i + 2 + \epsilon\gamma_i$; nel secondo invece si pone $m = 2$ e $y_i = 3x_i^2 + 2x_i + 1 + \epsilon\gamma_i$, dove γ_i è un numero random in $[0\,,\,1]$ e ϵ una costante positiva (usare 0.1 e 0.2).

10. Per il primo test dell'esercizio precedente confrontare la retta ottenuta con quella che si ottiene considerando la y come variabile indipendente.

ESERCIZI DA SVOLGERE PER L' ELABORATO
CAP. 5 e 6
CN A.A. 15/16

1. Scrivere due functions Matlab che, presi in input un intervallo $[a\,,\,b]$, una funzione f ivi definita e un intero $n \geq 1$, approssimano $I[f] = \int_a^b f(x)\,dx$ rispettivamente mediante $I_1^n[f]$, e $I_2^n[f]$ (assumere n pari nel secondo caso) ossia mediante le formula composite dei trapezi e di Simpson che utilizzano in tutto $n+1$ valutazioni di funzione su punti equispaziati in $[a\,,\,b]$.

2. Testare e confrontare il funzionamento delle functions costruite al punto precedente per approssimare il seguente integrale:

$$\int_0^{2\pi} xe^{-x}\cos(2x)\,dx = \frac{[3(e^{-2\pi}-1) - 10\pi e^{-2\pi}]}{25}\,.$$

   Riportare il confronto mediante una tabella che, per ogni $n = 2^k, k = 1,\ldots,8$, per ciascuno dei metodi riporta l'errore e, per $n \geq 4$, riporta anche il rapporto R_n fra l'errore corrente e quello precedentemente ottenuto usando $n/2$. Sapresti commentare l'andamento di R_n per i due metodi?

3. Scrivere due functions Matlab che rispettivamente implementano le formulazioni adattative dei metodi dei trapezi e di Simpson. Utiizzare tali functions per approssimare $I[f]$ definito per l'Esercizio 2 utilizzando come tolleranza in input $tol = 10^{-5}$.

4. Scrivere due functions Matlab che implementano i metodi iterativi di Jacobi e Gauss–Seidel. Come criterio di arresto utilizzare quello sul residuo relativo.

5. Testare e confrontare il funzionamento delle due functions costruite al punto precedente per risolvere un sistema lineare avente una matrice dei coefficienti a diagonale dominante.

6. Scrivere una function che implementa in Matlab il metodo delle potenze per il calcolo del google page rank e confrontarne le prestazioni con

quelle dei metodi di Jacobi e di Gauss–Seidel (per la sperimentazione considerare il caso in cui la matrice H iniziale abbia almeno una colonna nulla).