ELABORATO DI CALCOLO NUMERICO

Federico Schipani
Tommaso Ceccarini
Giuliano Gambacorta

3 giugno 2015

Indice

L'errore assoluto da un informazione relativa, l'errore relativo da un informazione assoluta!

Sestini, Brugnano, Magherini

CAPITOLO 1

Errori ed aritmetica finita

1.1 Esercizi del libro

Qua ci sono gli esercizi contenuti del libro

1.1.1 Esercizio 1.1

Sia $x = \pi \approx 3.1415 = \tilde{x}$. Calcolare il corrispondente errore relativo ϵ_x. Verificare che il numero di cifre decimali corrette nella rappresentazione approssimata di x mediante $\tilde{x}$ è all'incirca dato da

$$-\log_{10}|\epsilon_x|$$

Soluzione:

$$x = \pi \approx 3.1415, \qquad \tilde{x} = 3.1415$$
$$\Delta_x = x - \tilde{x} = 0,0000926$$
$$\epsilon_x = \frac{\Delta_x}{x} = 0,3 \cdot 10^{-4}$$
$$-log_{10}|\epsilon_x| = -log_{10}|0,3 \cdot 10^{-4}| \approx 4,5$$

Arrotondando $4,5$ si nota che il numero delle cifre decimali corrette nella rappresentazione è 4

1.1.2 Esercizio 1.2

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

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

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

1.1.3 Esercizio 1.3

Dimostrare che il metodo iterativo (1.1)*, convergente a* x^* (*vedi*(1.2))*, 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.

1.1.4 Esercizio 1.4

Il metodo iterativo

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

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* $\approx 10^{-22}$ *(comparare con i risultati in Tabella 1.1)*

1.1.5 Esercizio 1.5

Il codice Fortran

```
      program INTERO
c---variabili intere da 2 byte
      integer*2 numero, i
      numero = 32765
      do i = 1, 10
            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
7. −32765
8. −32764
9. −32763
10. −32762

Spiegarne il motivo

1.1.6 Esercizio 1.6

Dimostrare i teoremi 1.1 e 1.3

1.1.7 Esercizio 1.7

Completare la dimostrazione del teorema 1.4

1.1.8 Esercizio 1.8

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

1.1.9 Esercizio 1.9

Dimostrare che, detta u la precisione di macchina utilizzata,

$$-log_{10}u$$

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

1.1.10 Esercizio 1.10

Con riferimento allo standard IEEE 754 determinare, relativamente alla doppia precisione:

1. il più grande numero di macchina
2. il più piccolo numero di macchina normalizzato positivo
3. il più piccolo numero di macchina denormalizzato positivo
4. la precisione di macchina

Confrontare le risposte ai primi due quesiti col risultato fornito dalle `function` *Matlab* `realmax` e `realmin`

1.1.11 Esercizio 1.11

Eseguire le seguenti istruzioni Matlab:

```
x = 0; delta = 0.1;
while x ~= 1, x = x+delta, end
```

Spiegarne il (non) funzionamento

1.1.12 Esercizio 1.12

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

1.1.13 Esercizio 1.13

Eseguire le seguenti istruzioni Matlab:

```
help eps
((eps/2+1)-1)*(2/eps)
(eps/2+(1-1))*(2/eps)
```

Concludere che la somma algebrica non gode, in aritmetica finita, della proprietà associativa.

1.1.14 Esercizio 1.14

Eseguire e discutere il risultato delle seguenti istruzioni Matlab

$(1e300 - 1e300) * 1e300, \qquad (1e300 * 1e300) - (1e300 * 1e300)$

1.1.15 Esercizio 1.15

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

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

1.1.16 Esercizio 1.17

(Cancellazione numerica) Si supponga di dover calcolare l'espressione

$$y = 0.12345678 - 0.12341234 \equiv 0.0000444,$$

utilizzando una rappresentazione decimale con arrotondamento alla quarta cifra significativa. Comparare il risultato esatto con quello ottenuto in aritmetica finita, e determinare la perdita di cifre significative derivante dalla operazione effettuata. Verificare che questo risultato è in accordo con l'analisi di condizionamento.

1.1.17 Esercizio 1.18

(Cancellazione Numerica) Eseguire le seguenti istruzioni Matlab

```
format long e
a = 0.1
b = 0.099999999999
a-b
```

Valutare l'errore relativo sui dati di ingresso e l'errore relativo sul risultato ottenuto.

1.2 Esercizi integrativi

Questi sono gli esercizi integrativi sul capitolo 1

1.2.1 Esercizio 1

un'approssimazione del secondo ordine di $f'(x_0)$ utilizzando il passo di discretizzazione h e i seguenti tre valori di funzione $f(x_0), f(x_0 + h), f(x_0 + 2h)$ (molecola a tre punti in avanti).

1.2.2 Esercizio 2

Dimostrare che se un numero reale x viene approssimato da $\tilde{x}$ con un certo errore relativo ϵ_x, la quantità $-log_{10}|\epsilon_x|$ fornisce approssimativamente il numero di cifre decimali esatte di $\tilde{x}$.

1.2.3 Esercizio 3

Calcolare il più grande e il più piccolo numero reale di macchina positivo normalizzato che si può rappresentare utilizzando lo standard IEEE 754 nel formato della singola precisione e in quello della doppia precisione.
Soluzione:
Il numero più piccolo è uguale a:

$$R_1 = b^{\nu}$$

Per cui nel caso di questo esercizio:

$$2^{-126}$$

Il numero più grande è uguale a

$$R_2 = (1 - 2^{-24})2^{\varphi}$$

con

$$\varphi = 2^8 - 127$$

quindi

$$R_2 = 6.805646933 \cdot 10^{38}$$

.

1.2.4 Esercizio 4

Siano $x = 2.7352 \cdot 10^2, y = 4.8017 \cdot 10^{-2}$ e $z = 3.6152 \cdot 10^{-2}$. Utilizzando un'aritmetica finita che lavora in base 10 con arrotondamento e che riserva $m = 4$ cifre alla mantissa, confrontare gli errori assoluti $R_1 - R$ e $R_2 - R$, dove $R = x + y + z$ e

$$R_1 = (x \oplus y) \oplus z, \qquad R_2 = x \oplus (y \oplus z).$$

Soluzione:
Per facilitare i calcoli si portano x e y da 10^{-2} a 10^2:

$$y = 4.8017 \cdot 10^{-2} = 0.00048017 \cdot 10^2$$

$$z = 3.6152 \cdot 10^{-2} = 0.00031652 \cdot 10^2$$

Successivamente si calcola R_1, R_2 ed R:

$$R_1 = (x \oplus y) \oplus z) = (2.7352 + 0.00048017) + 0.00031652 = 2,7359 \cdot 10^2$$

$$R_2 = x \oplus (y \oplus z) = 2.7352 + (0.00048017 + 0.00031652) = 2.7360 \cdot 10^2$$

$$R = x + y + z = 2.73604169$$

Attraverso R si calcolano gli errori assoluti su R_1 ed R_2:

$$R_1 - R = -0.000014169 \cdot 10^2$$

$$R_2 - R = 0.000004169 \cdot 10^2$$

1.2.5 Esercizio 5

Un'aritmetica finita utilizza la base 10, l'arrotondamento, $m = 5$ cifre per la mantissa, $s = 2$ cifre per l'esponente e lo shift $\nu = 50$. Per gli interi esso utilizza $N = 7$ cifre decimali. Dire se il numero intero $x = 136726$ è un numero intero di macchina e come viene convertito in reale di macchina. Dire quindi se il numero intero $x = 78345 \cdot 10^{40}$ un reale di macchina e/o se è un intero di macchina.
Soluzione:
Si verifica facilmente che il numero $x = 136726$ è un intero di macchina.
Il numero x viene convertito in reale di macchina in questo modo: $x_r = 1,36726 \cdot 10^5$
Invece il numero $x = 78345 \cdot 10^{40}$ è un reale di macchina in quanto non bastano 5 cifre per rappresentarlo.
Per un maggiore sicurezza si calcola il più grande reale di macchina:

$$R_2 = (1 - 10^{-5})10^{50} = 9.9999 \cdot 10^{49}$$

Essendo $x < R_2$ è confermato che è un reale di macchina.

1.2.6 Esercizio 6

Dimostrare che il numero di condizionamento del problema di calcolo $\sqrt[n]{x}$ è $\frac{1}{n}$.
Soluzione:
Per prima cosa si riscrive la funzione:

$$f(x) = x^{\frac{1}{n}}$$

Il numero di condizionamento è uguale a

$$\kappa = |f'(x)\frac{x}{y}|$$

Si calcola la derivata di $f(x)$ che è uguale a

$$f'(x) = \frac{x^{\frac{1}{n}-1}}{n}$$

Quindi:

$$\kappa = |\frac{x^{\frac{1}{n}-1}}{n}\frac{x}{\sqrt[n]{x}}|$$

Da cui con passaggi algebrici:

$$|\frac{\sqrt[n]{x}\ x^{-1}}{n}\frac{x}{\sqrt[n]{x}}| = |\frac{1}{n}| = \frac{1}{n}$$

.

1.2.7 Esercizio 7

Individuare l'algoritmo più efficace per valutare, in aritmetica finita, la funzione $f(x) = \ln x^4$
Soluzione:
Bisogna semplicemente riscrivere la funzione per evitare problemi di overflow e underflow e poi valutarla. $f(x) = \ln x^4 = 4 \cdot \ln x$
L'algoritmo è questo:
SCRIVERE ALGORITMO

1.2.8 Esercizio 8

Individuare una forma algebrica equivalente ma preferibile in aritmetica finita per il calcolo dell'espressione $(x+2)^3 - x^3$
Soluzione:

$$(x+2)^3-x^3 = (x+2)(x+2)^2-x^3 = (x+2)(x^2+4x+4)-x^3 = x^3+4x^2+4x+2x^2+8x+8-x^3 = 6x^2+12x+8$$

1.2.9 Esercizio 9

Si calcoli l'approssimazione $\tilde{y}$ *della differenza tra y fra* $x_2 = 3.5555$ *e* $x_1 = 3.5554$ *utilizzando un'aritmetica finita che lavora con arrotondamento in base 10 con 4 cifre per la mantissa normalizzata. Se ne calcoli quindi il corrispondente errore relativo e la maggiorazione di esso che si ottiene utilizzando il numero di condizionamento della somma algebrica.*
Soluzione:
Prima si calcola un'approssimazione di x_1 e x_2 poi si calcola un approssimazione $\tilde{y}$ della differenza y.

$$\tilde{x_1} = 3.555$$

$$\tilde{x_2} = 3.556$$

$$\tilde{y} = \tilde{x_1} - \tilde{x_2} = 0.001$$

$$y = x_2 - x_1 = 0.0001$$

L'errore relativo è uguale a: $\varepsilon_y = \frac{0.001-0.0001}{0.0001} = 9$
Per la maggiorazione serve $max\{|\varepsilon_{x_1}|, |\varepsilon_{x_2}|\}$

$$\varepsilon_{x_1} = \frac{\tilde{x_1} - x_1}{x_1} = -1.125 \cdot 10^{-4}$$

$$\varepsilon_{x_2} = \frac{\tilde{x_2} - x_2}{x_2} = 1.406 \cdot 10^{-4}$$

Quindi:

$$k = \frac{|-3.5554| + |3.5555|}{|3.5555 - 3.5554|} \cdot 1.406 \cdot 10^{-4} = 9.9979254$$

1.2.10 Esercizio 10

Dimostrare che il numero razionale 0.1 *(espresso in base 10) non può essere un numero di macchina in un'aritmetica finita che utilizza la base 2 indipendentemente da come viene fissato il numero m di bit riservati alla mantissa. Dare una maggiorazione del corrispondente errore relativo di rappresentazione supponendo di utilizzare l'aritmetica finita binaria che utilizza l'arrotondamento e assume $m = 7$.*

CAPITOLO 2

Radici di una equazione

2.1 Esercizi del libro

2.1.1 Esercizio 2.1

Definire una procedura iterativa basata sul metodo di Newton per determinare $\sqrt{a}$, per un assegnato $a > 0$. Costruire una tabella dell'approssimazioni relativa al caso $a = x_0 = 2$ (Comparare con la tabella 1.1)

2.1.2 Esercizio 2.2

Generalizzare il risultato del precedente esercizio, derivando una procedura iterativa basata sul metodo di Newton per determinare $\sqrt[p]{a}$ per un assegnato $a > 0$

2.1.3 Esercizio 2.3

In analogia con quanto visto nell'Esercizio 2.1, definire una procedura iterativa basata sul metodo delle secanti per determinare $\sqrt{a}$. Confrontare con l'esercizio 1.4.

2.1.4 Esercizio 2.4

Discutere la convergenza del metodo di Newton, applicato per determinare le radici dell'equazione (2.11) in funzione della scelta del punto iniziale x_0

2.1.5 Esercizio 2.5

Comparare il metodo di Newton (2.9), il metodo di Newton modificato (2.16) ed il metodo di accellerazione di Aitken (2.17), per approsimare gli zeri delle funzioni:

$$f_1(x) = (x-1)^{10}, \qquad f_2(x) = (x-1)^{10}e^x$$

per valori decrescenti della tolleranza `tolx`. Utilizzare, in tutti i casi, il punto iniziale $x_0 = 10$.

2.1.6 Esercizio 2.6

È possibile, nel caso delle funzioni del precedente esercizio utilizzare il metodo di bisezione per determinare lo zero?

2.1.7 Esercizio 2.7

Costruire una tabella in cui si comparano, a partire dallo stesso punto iniziale $\texttt{x}_0 = 0$*, 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*

$$f(x) = x - \cos x$$

2.1.8 Esercizio 2.8

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

2.1.9 Esercizio 2.9

Quali controlli introdurreste, negli algoritmi 2.4-2.6, al fine di rendere più "robuste" le corrispondenti iterazioni?

CAPITOLO 3

Capitolo 3

3.1 Esercizi del libro

3.1.1 Esercizio 3.1

Riscrivere gli Algoritmi 3.1-3.4 in modo da controllare che la matrice dei coefficenti sia non singolare.

3.1.2 Esercizio 3.2

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

3.1.3 Esercizio 3.3

Dimostrare che il prodotto di due matrici triangolari inferiori a diagonale unitaria è a sua volta una matrice triangolare inferiore a diagonale unitaria.

3.1.4 Esercizio 3.4

Dimostrare che la matrice inversa di una matrice triangolare inferiore è as ua volta triangolare inferiore. Dimostrare inoltre che, se la matrice ha diagonale unitaria, tale è anche la diagonale della sua inversa.

3.1.5 Esercizio 3.5

Dimostrare i lemmi 3.2 e 3.3.

3.1.6 Esercizio 3.6

Dimostrare che il numero di flop richiesti dall'algoritmo 3.5 è dato da (3.25)

3.1.7 Esercizio 3.7

Scrivere una function matlab che implementi efficientemente l'algoritmo 3.5 per calcolare la fattorizzazione LU di una matrice.

3.1.8 Esercizio 3.8

Scrivere una function Matlab che, avendo in ingresso la matrice A riscritta dall'algoritmo 3.5, ed un vettore x contenente i termini noti del sistema lineare (3.1), ne calcoli efficientemente la soluzione.

3.1.9 Esercizio 3.9

Dimostrare i lemmi 3.4 e 3.5

3.1.10 Esercizio 3.10

Completare la dimostrazione del Teorema 3.6

3.1.11 Esercizio 3.11

Dimostrare che , se A è non singolare, le matrici A^TA e AA^T sono sdp.

3.1.12 Esercizio 3.12

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

3.1.13 Esercizio 3.13

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

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

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

$$x^TAx = x^TA_sx$$

3.1.14 Esercizio 3.14

Dimostrare la consistenza delle formule (3.29).

3.1.15 Esercizio 3.15

Dimostrare che il numero di flop richiesti dall'algoritmo 3.6 è dato da 3.30

3.1.16 Esercizio 3.16

Scrivere una function Matlab che implementi efficientemente l'algoritmo di fattorizzazione LDL^T per matrici sdp

3.1.17 Esercizio 3.17

Scrivere una funzione Matlab che, avendo in ingresso la matrice A prodotta dalla precedente funziona, contenente la fattorizzazione LDL^T della matrice sdp originaria, ed un vettore di termini noti, x, calcoli efficientemente la soluzione del corrispondente sistema lineare.

3.1.18 Esercizio 3.18

Utilizzare la function dell'esercizio 3.16 per verificare che la matrice

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 \\ 1 & 2 & 1 & 1 \\ 1 & 2 & 1 & 2 \end{pmatrix}$$

non è sdp.

3.1.19 Esercizio 3.19

Dimostrare che, al passo i-esimo di eliminazione di Gauss con pivoting parziale, si ha $a_{k_i i}^{(i)} \neq 0$, se A è non singolare.

3.1.20 Esercizio 3.20

Con riferimento alla matrice A definita nella (3.24), qual è la matrice di permutazione P che rende PA fattorizzatile LU? Chi sono, in tal caso, i fattori L e U?

3.1.21 Esercizio 3.21

Scrivere una function Matlab che implementi efficientemente l'algoritmo di fattorizzazione LU con pivoting parziale.

3.1.22 Esercizio 3.22

Scrivere una funzione matlab che, avendo in ingresso la matrice A prodotta dalla precedente function, contenente la fattorizzazione LU della matrice permutata, il vettore p contenente l'informazione relativa alla corrispondente matrice di permutazione, ed un vettore di termini noti, x, calcoli efficientemente la soluzione del corrispondente sistema lineare.

3.1.23 Esercizio 3.23

Costruire alcuni esempi di applicazione delle funzioni degli esercizi 3.21 e 3.22

3.1.24 Esercizio 3.24

Le seguenti istruzioni MATLAB *sono equivalenti a risolvere il dato sistema 2×2 con l'utilizzo del pivoting e non, rispettivamente. Spiegarne il differente risultato ottenuto. Concludere che l'utilizzo del pivoting migliora, in generale, la prognosi degli errori in aritmetica finita.*

```
A = [eps 1;1 0], b = [1; 1/4]
A\b
L = [1 0; 1/eps 1], U = [eps 1; 0 -1/eps]
L*U-A
U\(L\b)
```

3.1.25 Esercizio 3.25

Si consideri la seguente matrice bidiagonale inferiore

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

Calcolare $k_\infty(A)$. Confrontate il risultato con quello fornito dalla ***function cond*** *di* MATLAB. *Dimostrare, e verificare, che $k_\infty(A) = k_1(A)$.*

3.1.26 Esercizio 3.26

Si consideri i seguenti vettori di $\mathbb{R}^{10}$,

$$\underline{b} = \begin{pmatrix} 1 \\ 101 \\ \vdots \\ 101 \end{pmatrix}, \quad \underline{c} = 0.1 \cdot \begin{pmatrix} 1 \\ 101 \\ \vdots \\ 101 \end{pmatrix},$$

ed i seguenti sistemi lineari

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

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

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

Confrontare questi vettori con quelli calcolato dalle seguenti due serie di istruzioni MATLAB,

```
b=[1 101*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+[1 101*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, alla luce di quanto visto in Sezione 3.7.

3.1.27 Esercizio 3.27

Dimostrare che il numero di `flop` *richiesti dall'Algoritmo di fattorizzazione QR di Householder è dato da 3.60.*

3.1.28 Esercizio 3.28

Definendo il vettore $\underline{\hat{v}} = \frac{\underline{v}}{v_1}$*, verificare che* ***beta****, come definito nel seguente algoritmo per la fattorizzazione QR di Householder, corrisponde alla quantità* $\frac{2}{\underline{\hat{v}}^T \underline{\hat{v}}}$

3.1.29 Esercizio 3.29

Scrivere una ***function*** MATLAB *che implementi efficientemente l'algoritmo di fattorizzazione QR, mediante il metodo di Householder (vedi l'Algoritmo descritto nell'Esercizio precedente).*

3.1.30 Esercizio 3.30

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* $\underline{b}$*, calcoli efficientemente la soluzione del sistema lineare sovradeterminato.*

3.1.31 Esercizio 3.31

Utilizzare le ***function*** *degli Esercizi 3.29 e 3.30 per calcolare la soluzione ai minimi quadrati di (3.51), ed il corrispondente residuo, nel caso in cui*

$$A = \begin{pmatrix} 3 & 2 & 1 \\ 1 & 2 & 3 \\ 1 & 2 & 1 \\ 2 & 1 & 2 \end{pmatrix}, \qquad \underline{b} = \begin{pmatrix} 10 \\ 10 \\ 10 \\ 10 \end{pmatrix}.$$

3.1.32 Esercizio 3.32

Calcolare i coefficienti della equazione della retta

$$r(x) = a_1 x + a_2,$$

che meglio approssima i dati prodotti dalle seguenti istruzioni MATLAB*:*

```
x     = linspace(0,10,101)'
gamma = 0.1
y     = 10*x+5+(sin(x*pi))*gamma
```

Riformulare il problema come minimizzazione della norma Euclidea di un corrispondente vettore residuo. Calcolare la soluzione utilizzando le ***function*** *sviluppate negli Esercizi 3.29 e 3.30, e confrontarla con quella ottenuta dalle seguenti istruuzioni* MATLAB*:*

```
A = [x x.^0]
a = A\y
r = A*a-y
```

Confrontare i risultati che si ottengono per i seguenti valori del parametro ***gamma****:*

$$0.5, 0.1, 0.05, 0.01, 0.005, 0.001.$$

Quale è la soluzione che si ottiene nel limite $\boldsymbol{gamma} \to 0$*?*

3.1.33 Esercizio 3.33

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 (3.63) per calcolarne il punto stazionario.*

3.1.34 Esercizio 3.34

Uno dei metodi di base per la risoluzione di equazioni differenziali ordinarie, $y'(t) = f(t, y(t))$*,* $t \in [t_0, T]$*,* $y(t_0) = y_0$ *(problema di Cauchy di prim'ordine), è il metodo di Eulero implicito,*

$$y_n = y_{n-1} + hf_n, \qquad n = 1, 2, \ldots, N \equiv \frac{T - t_0}{h},$$

in cui $y_n \approx y(t_n)$*,* $f_n = f(t_n, y_n)$*,* $t_n = t_0 + nh$*. Utilizzare questo metodo nel caso in cui* $t_0 = 0$*,* $T = 10$*,* $N = 100$*,* $y_0 = (1, 2)^T$ *e, se* $y = (x_1, x_2)^T$*,*
$f(t, y) = (-10^3 x_1 + \sin x_1 \cos x_2, -2x_2 + \sin x_1 \cos x_2)^T$*. Utilizzare il metodo (3.64) per la risoluzione dei sistemi nonlineari richiesti.*

CAPITOLO 4

Capitolo 4. Approssimazione di funzioni

4.1 Esercizi del libro

4.1.1 Esercizio 1

Sia $f(x) = 4x^2 - 12x + 1$. *Determinare* $p(x) \in \Pi_4$ *che interpola* $f(x)$ *sulle ascisse* $x_i = i, i = 0, \ldots 4$.

4.1.2 Esercizio 2

Dimostrare che il seguente algoritmo,

```
p = a(n+1)
for k = n:-1:1
        p = p*x +a(k)
end
```

valuta il polinomio (4.4) nel punto x, se il vettore a contiene i coefficienti del polinomio $p(x)$ *(Osservare che in* MATLAB *i vettori hanno indice che parte da 1, invece che da 0).*

4.1.3 Esercizio 3

Dimostrare il lemma 4.1

4.1.4 Esercizio 4

Dimostrare il lemma 4.2

4.1.5 Esercizio 5

Dimostrare il lemma 4.4

4.1.6 Esercizio 6

*Costruire una **function** MATLAB che implementi in modo efficiente l'Algoritmo 4.1*

4.1.7 Esercizio 7

*Dimostrare che il seguente algoritmo, che riceve in ingresso i vettori x e f prodotti dalla **function** dell'Esercizio 4.6, valuta il corrispondente polinomio interpolante di Newton in un punto xx assegnato.*

```
p = f(n+1)
for k = n:-1:1
        p = p*(xx-x(k)) +f(k)
end
```

*Quale è il suo costo computazionale? Confrontarlo con quello dell'Algoritmo 4.1. Costruire, quindi, una corrispondente **function** MATLAB che lo implementi efficientemente (complementare la possibilità che xx sia un vettore)*

4.1.8 Esercizio 8

*Costruire una **function** MATLAB che implementi in modo efficiente l'algoritmo del calcolo delle differenze divise per il polinomio di Hermite.*

4.1.9 Esercizio 9

Si consideri la funzione

$$f(x) = (x-1)^9.$$

*Utilizzando le **function** degli Esercizi 4.6 e 4.8, valutare i polinomi interpolanti di Newton e di Hermite sulle ascisse*

$$0, 0.25, 0.5, 0.75, 1,$$

per x=linspace(0,1,101). Raffigurare, quindi, (e spiegare) i risultati.

4.1.10 Esercizio 10

Quante ascisse di interpolazione equidistanti sono necessarie per approssimare la funzione $\sin(x)$ sull'intervallo $[0, 2\pi]$, con un errore di interpolazione inferiore a 10^{-6}?

4.1.11 Esercizio 11

*Verificare sperimentalmente che, considerando le ascisse di interpolazione equidistanti (**??**) su cui si definisce il polinomio $p(x)$ interpolante $f(x)$, l'errore $||f-p||$ diverge, al crescere di n, nei seguenti due casi:*

1. *esempio di Runge*:
$$f(x) = \frac{1}{1+x^2}, \qquad [a,b] \equiv [-5,5];$$
2. *esempio di Bernstein*:
$$f(x) = |x|, \qquad [a,b] \equiv [-1,1].$$

4.1.12 Esercizio 12

Dimostrare che, se $x \in [-1, 1]$, allora:

$$\tilde{x} \equiv \frac{a+b}{2} + \frac{b-a}{2} x \in [a, b].$$

Viceversa, se $\tilde{x} \in [a, b]$, allora:

$$x \equiv \frac{2\tilde{x} - a - b}{b-1} \in [-1, 1].$$

Concludere che è sempre possibile trasformare il problema di interpolazione (4.1)-(4.2) in uno definito sull'intervallo $[-1, 1]$, e viceversa.

4.1.13 Esercizio 13

Dimostrare le proprietà dei polinomi di Chebyshev di I specie (4.24) elencate nel Teorema 4.9.

4.1.14 Esercizio 14

Quali diventano le ascisse di Chebyshev (4.26), per un problema definito su un generico intervallo $[a, b]$?

4.1.15 Esercizio 15

Utilizzare le ascisse di Chebyshev (4.26) per approssimare gli esempi visti nell'Esercizio 4.11, per $n = 2, 4, 6, \ldots, 40$.

4.1.16 Esercizio 16

Verificare che la fattorizzazione LU della matrice dei coefficienti del sistema tridiagonale (4.40) è dato da:

$$L = \begin{pmatrix} 1 & & & \\ l_2 & 1 & & \\ & \ddots & \ddots & \\ & & l_{n-1} & 1 \end{pmatrix}, \qquad U = \begin{pmatrix} u_1 & \xi_1 & & \\ & u_2 & \ddots & \\ & & \ddots & \xi_{n-2} \\ & & & u_{n-1} \end{pmatrix},$$

con

$$\begin{aligned} & u_1 = 2, \\ & l_i = \frac{\varphi_i}{u_{i-1}}, \\ & u_i = 2 - l_i \xi_{i-1}, \qquad i = 2, \ldots, n-1. \end{aligned}$$

*Scrivere una **function** MATLAB che implementi efficientemente la risoluzione della (4.40).*

4.1.17 Esercizio 17

*Generalizzare la fattorizzazione del precedente Esercizio 4.16 al caso della matrice dei coefficienti del sistema lineare (4.41). Scrivere una corrispondente **function** MATLAB che risolva efficientemente questo sistema.*

4.1.18 Esercizio 18

Scrivere una ***function*** MATLAB *che, noti gli* $\{m_i\}$ *in (4.32), determini l'espressione, polinomiale a tratti, della spline cubica (4.36).*

4.1.19 Esercizio 19

Costruire una ***function*** MATLAB *che implementi le spline cubiche naturali e quelle definite dalle condizioni not-a-knot.*

4.1.20 Esercizio 20

Utilizzare la ***function*** *dell'Esercizio 4.19 per approssimare, su partizioni (4.28) uniformi con* $n = 10, 20, 30, 40$*, gli esempi proposti nell'Esercizio 4.11.*

4.1.21 Esercizio 21

Interpretare la retta dell'Esercizio 3.32 come retta di approssimazione ai minimi quadrati dei dati.

4.1.22 Esercizio 22

È noto che un fenomeno ha un decadimento esponenziale, modellizzato come

$$y = \alpha \cdot e^{-\lambda t},$$

in cui α e λ sono parametri positivi e incogniti. Riformulare il problema in modo che il modello sia di tipo polinomiale. Supponendo inoltre di disporre delle seguenti misure,

t_1	0	1	2	3	4	5	6	7	8	9	10
y_i	5.22	4.00	4.28	3.89	3.53	3.12	2.73	2.70	2.20	2.08	1.94

calcolare la stime ai minimi quadrati dei due parametri incogniti. Valutare il residuo e raffigurare, infine, i risultati ottenuti.

CAPITOLO 5

Formule di quadratura

5.1 Esercizio 1

Calcolare il numero di condizionamento dell'integrale

$$\int_0^{e^{21}} \sin\sqrt{x}\, dx.$$

Questo problema è ben condizionato o è malcondizionato?
Soluzione:

$$I(f) = \int_0^{e^{21}} \sin\sqrt{x}\, dx$$

Poichè $\kappa = b - a$, dove $[a, b]$ è l'intervallo d'integrazione, in questo caso si ha che $\kappa = e^{21} - 0$ ed il problema risulta mal condizionato.

5.2 Esercizio 2

Derivare, dalla (5.5), i coefficienti della formula dei trapezi (5.6) e della formula di Simpson (5.7).
Soluzione:
Per generare i coefficenti si usa la formula (5.5) del libro.

- Per la formula dei trapezi si usa $n = 1$ ed i coefficenti risultano così:

$$c_{1,1} = \int_0^1 \frac{t-0}{1-0} dt = \int_0^1 t\, dt = \frac{t^2}{2}|_0^1 = \frac{1}{2}$$

$$c_{0,1} = 1 - c_{1,1} = \frac{1}{2}$$

- Per la formula di Simpson si usa $n = 2$ ed i coefficenti risultano così:

$$c_{1,2} = \int_0^2 \frac{t-0}{1-0}\frac{t-2}{1-2}dt = \int_0^2 t(t-2)dt = \left[\frac{t^3}{3} - t^2\right]_0^2 = \frac{4}{3}$$

$$c_{22} = \int_0^2 \frac{t-0}{2-0}\frac{t-1}{2-1}dt = \int_0^2 \frac{t(t-1)}{2}dt = \left[\frac{t^3}{6} - \frac{t^2}{4}\right]_0^2 = \frac{1}{3}$$

$$c_{0,2} = 1 - c_{1,2} - c_{2,2} = \frac{1}{3}$$

5.3 Esercizio 3

Verificare, utilizzando il risultato del Teorema (5.2) le (5.9) e (5.10).
Soluzione:
Per $n = 1$ si ha che:

$$\nu_n = \int_0^1 \prod_0^1 = \int_0^1 t(t-1)dt = \int_0^1 t_2 - t\, dt = \left[\frac{t^3}{3} - \frac{t^2}{2}\right]_0^1 = -\frac{1}{6}$$

da cui:

$$E_1(f) = -\frac{1}{6}\frac{f^{(2)}}{2!}(\xi)(b-a)^3 = -\frac{1}{12}f^{(2)}(\xi)(b-a)^3, \qquad \xi \in [a,b].$$

5.4 Esercizio 4

*Scrivere una **function** MATLAB che implementi efficientemente la formula dei trapezi composita (5.11).*
Soluzione:

```
function [result] = trapeziComposita(fun, a, b, n)
    h = (b-a)/n;
    result = 0;
    for i=1:n-1
        result = result+fun(a+i*h);
    end
    result = (h/2)*(2*result + fun(a) + fun(b));
end
```

5.5 Esercizio 5

*Scrivere una **function** MATLAB che implementi efficientemente la formula di Simpson composita (5.13).*
Soluzione:

```
2
function [result] = simpsonComposita(fun, a, b, n)
4     h = (b-a)/n;
      result = fun(a)-fun(b);
6     for i=1:n/2
          result = result + 4*fun(a+(2*i-1)*h)+2*fun((a+2*i*h));
8     end
      result = result*(h/3);
10 end
```

5.6 Esercizio 6

Implementare efficientemente in MATLAB *la formula adattativa dei trapezi.*
Soluzione:

```
function [result, points] = trapeziAdattativa(fun, a, b, tol)
2     [result, points] = trapeziAdattativaRicorsiva(fun, a, b, tol, 3);
end
4
function [result, points] = trapeziAdattativaRicorsiva(fun, a, b, tol
    , points)
6     h = (b-a)/2;
      m = (a+b)/2;
8     int1 = h*(feval(fun, a) + feval(fun, b));
      result = int1/2 + h*feval(fun, m);
10    err = abs(result-int1)/3;
      if err>tol
12        [rsx, pointsx] = trapeziAdattativaRicorsiva(fun, a, m, tol/2,
              1);
          [rdx, pointdx] = trapeziAdattativaRicorsiva(fun, m, b, tol/2,
              1);
14        result = rsx+rdx;
          points = points+pointsx+pointdx;
16    end
end
```

5.7 Esercizio 7

Implementare efficientemente in MATLAB *la formula adattativa di Simpson.*
Soluzione:

```
1
3 function [result, points] = simpsonAdattativa(fun, a, b, tol)
      [result, points] = simpsonAdattativaRicorsiva(fun, a, b, tol, 5);
5 end
```

```
function [result, points] = simpsonAdattativaRicorsiva(fun, a, b, tol
    , points)
    h = (b-a)/6;
    m = (a+b)/2;
    m1 = (a+m)/2;
    m2 = (m+b)/2;
    int1 = h*(feval(fun, a) +4*feval(fun, m) + feval(fun, b));
    result = int1/2 + h*(2*feval(fun, m1) + 2*feval(fun, m2) -feval(
        fun, m));
    err = abs(result-int1)/15;
    if err>tol
        [rsx, pointsx] = simpsonAdattativaRicorsiva(fun, a, m, tol/2,
            1);
        [rdx, pointdx] = simpsonAdattativaRicorsiva(fun, m, b, tol/2,
            1);
        result = rsx+rdx;
        points = points+points+pointdx;
    end
end
```

5.8 Esercizio 8

Come è classificabile, dal punto di vista del condizionamento, il seguente problema?

$$\int_{\frac{1}{2}}^{100} -2x^{-3}\cos\left(x^{-2}\right)\,\mathrm{d}x \equiv \sin\left(10^{-4}\right) - \sin(4)$$

Soluzione:

5.9 Esercizio 9

Utilizzare le ***function*** *degli Esercizi 5.4 e 5.6 per il calcolo dell'integrale*

$$\int_{\frac{1}{2}}^{100} -2x^{-3}\cos\left(x^{-2}\right)\,\mathrm{d}x \equiv \sin\left(10^{-4}\right) - \sin(4),$$

indicando gli errori commessi. Si utilizzi $n = 1000, 2000, \ldots, 10000$ *per la formula dei trapezi composita e* $tol = 10^{-1}, 10^{-2}, \ldots, 10^{-5}$ *per la formula dei trapezi adattativa (indicando anche il numero di punti).*
Soluzione:

Trapezi Composita		
n	I	$E_1^{(n)}$
1000	6.6401e-001	9.2897e-002
2000	7.3077e-001	2.6131e-002
3000	7.4507e-001	1.1836e-002
4000	7.5020e-001	6.7007e-003
5000	7.5260e-001	4.3009e-003
6000	7.5391e-001	2.9914e-003
7000	7.5470e-001	2.1998e-003
8000	7.5522e-001	1.6853e-003
9000	7.5557e-001	1.3321e-003
10000	7.5582e-001	1.0793e-003

Trapezi adattativa			
tol	I	$E_1^{(n)}$	points
1.0e-001	7.5143e-001	5.4696e-003	159
1.0e-002	7.5563e-001	1.2676e-003	471
1.0e-003	7.5657e-001	3.3005e-004	1567
1.0e-004	7.5684e-001	6.5936e-005	4851

5.10 Esercizio 10

Utilizzare le ***function*** *degli Esercizi 5.5 e 5.7 per il calcolo dell'integrale*

$$\int_{\frac{1}{2}}^{100} -2x^{-3}\cos\left(x^{-2}\right)\ \mathrm{d}x \equiv \sin\left(10^{-4}\right) - \sin(4),$$

indicando gli errori commessi. Si utilizzi $n = 1000, 2000, \ldots, 10000$ *per la formula di Simpson composita e* $tol = 10^{-1}, 10^{-2}, \ldots, 10^{-5}$ *per la formula di Simpson adattativa (indicando anche il numero di punti).*
Soluzione:

Simpson composita		
n	I	$E_1^{(n)}$
1000	7.0132e-001	5.5580e-002
2000	7.5303e-001	3.8753e-003
3000	7.5617e-001	7.2977e-004
4000	7.5668e-001	2.2403e-004
5000	7.5681e-001	9.0209e-005
6000	7.5686e-001	4.3062e-005
7000	7.5688e-001	2.3094e-005
8000	7.5689e-001	1.3479e-005
9000	7.5689e-001	8.3892e-006
10000	7.5690e-001	5.4921e-006

Simpson Adattativa			
tol	I	$E_1^{(n)}$	points
1.0e-001	7.5701e-001	1.1164e-004	49
1.0e-002	7.5671e-001	1.9384e-004	65
1.0e-003	7.5690e-001	4.8068e-006	93
1.0e-004	7.5688e-001	1.7808e-005	181
1.0e-005	7.5690e-001	4.8337e-006	309

CAPITOLO 6

Calcolo del Google Pagerank

6.1 Esercizio 1

[Teorema di Gershgorin] Dimostrare che gli autovalori di una matrice $A = (a_{ij}) \in \mathbb{C}^{nn}$ *sono contenuti nell'insieme*

$$\mathcal{D} = \bigcup_{i=1}^{n} \mathcal{D}_i, \qquad \mathcal{D}_i = \left\{ \lambda \in \mathbb{C} : |\lambda - a_{ii}| \leq \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}| \right\}, \quad i = 1, \ldots, n.$$

Soluzione:
Sia $\lambda \in \sigma A$ autovalore e sia $\underline{x}$ il suo corrispettivo autovettore. Quindi per $i = 1, \ldots, n$ si ha $\underline{e}_i^T A\underline{x} = \lambda \underline{e}_i^T \underline{x}$. Ponendo i tale che $|x_i| \geq |x_j|$ $(x_i \neq 0)$ si ottiene
$\lambda = \frac{\sum_{j=1}^{n} a_{i,j} x_j}{x_i} = a_{i,i} + \sum_{j=1\ j\neq i}^{n} a_{i,j} \frac{x_j}{x_i}$ indi per cui $\lambda - a_{i,i} = \sum_{j=1\ j\neq i}^{n} a_{i,j} \frac{x_j}{x_i}$. Portando tutto inv alore assoluto si ha:

$$|\lambda - a_{i,i}| = \left| \sum_{j=1\ j\neq i}^{n} a_{i,j} \frac{x_j}{x_i} \right| \leq \sum_{j=1\ j\neq i}^{n} \left| a_{i,j} \frac{x_j}{x_i} \right| \leq \sum_{j=1\ j\neq i}^{n} |a_{i,j}|$$

poiché $\left| \frac{x_j}{x_i} \right| \leq 1$ in quanto $|x_i| \geq |x_j|$. Segue $\lambda \in \bigcup_{i=1}^{n} \mathcal{D}_i$ da cui $\sigma A \subseteq \mathcal{D}$.

6.2 Esercizio 2

Utilizzare il metodo delle potenze per approssimare l'autovalore dominante della matrice

$$A_n = \begin{pmatrix} 2 & -1 & & \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{pmatrix} \in \mathbb{R}^{n \times n},$$

per valori crescenti di n. Verificare numericamente che questo è dato da $2\left(1+\cos\frac{\pi}{n+1}\right)$.
Soluzione:
La matrice A_n è ovviamente una M-Matrice in quanto si può scrivere come:

$$A_n = 2(I_n - B_n)$$

In questo caso si ha:

$$B_n = \begin{pmatrix} 0 & 1/2 & & \\ 1/2 & 0 & \ddots & \\ & \ddots & \ddots & 1/2 \\ & & 1/2 & 0 \end{pmatrix} \in \mathbb{R}^{n\times n}.$$

Numericamente, applicando il metodo delle potenze, si ottengono i seguenti risultati:

Tolleranza $tol = 10^{-5}$			
n	λ_1	Risultato atteso	Errore
10	3.9189	3.9190	0.0001
15	3.9614	3.9616	0.0002
20	3.9774	3.9777	0.0003
25	3.9853	3.9854	0.0002
30	3.9891	3.9897	0.0006
35	3.9921	3.9924	0.0003
40	3.9929	3.9941	0.0012
45	3.9938	3.9953	0.0015
50	3.9947	3.9962	0.0015

Provando con una tolleranza di 10^{-7} si perviene al seguente risultato:

Tolleranza $tol = 10^{-7}$			
n	λ_1	Risultato atteso	Errore
5	3.7321	3.7321	0.0000
10	3.9190	3.9190	0.0000
15	3.9616	3.9616	0.0000
20	3.9777	3.9777	0.0000
25	3.9854	3.9854	0.0000
30	3.9897	3.9897	0.0000
35	3.9924	3.9924	0.0000
40	3.9941	3.9941	0.0000
45	3.9953	3.9953	0.0000
50	3.9962	3.9962	0.0000

6.3 Esercizio 3

Dimostrare i Corollari 6.2 e 6.3.
Soluzione:
Corollario 6.2 Sia $A = \alpha(I - B) = M - N$ con $0 \le N \le \alpha B$ quindi
$M = \alpha I - \alpha B + N = \alpha I - \alpha B + \alpha Q = \alpha(I - (B - Q))$ con $\alpha Q = N \le \alpha B$ e $0 \le Q \le B$. Visto che $0 \le B - Q \le B$, per il Lemma 6.2, $\rho(B - Q) \le \rho(B) \le 1$. Per quanto detto sopra M è una M-matrice; lo splitting, quindi, risulta regolare e infatti $M^{-1} \ge 0$ e $B - Q \ge 0$.

Corollario 6.3 Presa $A = M - N$ M-matrice con $M = diag(A)$ e considerando la matrice $B = A - M$ che avrà elementi nulli sulla diagonale e i restanti gli elementi di A. Visto che $A = \alpha(I - B) = \alpha I - \alpha B = M - N$, si può dire che $\alpha B = N$ di conseguenza lo splitting risulta regolare e il metodo è convergente.

6.4 Esercizio 4

Dimostrare il Teorema 6.9.
Soluzione: A è una M-matrice, quindipuò essere scritta come $A = \alpha(I - B)$ con $B \geq 0$ e $\rho(B) < 1$. Per ipotesi sappiamo che, $A = D - L - U$ quindi si può dire che

$$D = \alpha(I - diag(B))$$

e

$$(L + U) = \alpha(B - diag(B))$$

.
La matrice $(L + U) = \alpha(B - diag(B))$ risulta > 0 perché $B > 0 \rightarrow (B - diag(B)) > 0$. La matrice D ha elementi positivi sulla diagonale: supponiamo per assurdo $a_{i,i} \leq 0$: l'i-esima riga di A, negativa, è data da $A\underline{e}_i \leq \underline{0}$ dato che A è una M-matrice risulta $\underline{e}_i \leq A^{-1}\underline{0} = \underline{0}$ assurdo poiché $\underline{e}_i \geq \underline{0}, \forall i$.

6.5 Esercizio 5

Tenendo conto della (6.10), riformulare il metodo delle potenze (6.11) per il calcolo del Google pagerank come metodo iterativo definito da uno splitting regolare.
Soluzione: Il problema del *Google pagerank* è $S(p)\hat{\underline{x}} = \hat{\underline{x}}$ dove $S(p) = pS + (1 - p)\underline{v}\,\underline{e}^T$. Sostituendo, risulta

$$(pS + (1 - p)\underline{v}\,\underline{e}^T)\hat{\underline{x}} = \hat{\underline{x}} \quad \Rightarrow \quad (I - pS)\hat{\underline{x}} = (1 - p)\underline{v}\,\underline{e}^T\hat{\underline{x}}$$

Dato che $\underline{v} = \frac{1}{n}\underline{e}$ ed $e^T\hat{\underline{x}} = 1$ si ricava

$$(I - pS)\hat{\underline{x}} = \frac{1 - p}{n}\underline{e}.$$

Si può infine definire il seguente metodo iterativo:

$$I\underline{x}_{k+1} = pS\underline{x}_k + \frac{1 - p}{n}\underline{e}.$$

Il metodo è convergente in quanto la matrice di iterazione ha raggio spettrale minore di 1: $\rho(I^{-1}pS) = \rho(pS) < 1$ dato che $p \in (0, 1)$ e $\rho(S) = 1$.

6.6 Esercizio 6

Dimostrare che il metodo di Jacobi converge asintoticamente in un numero minore di iterazioni, rispetto al metodo delle potenze (6.11) per il calcolo del Google pagerank. Soluzione: La matrice di iterazione di Jacobi ha raggio spettrale minore di 1 mentre, nel calcolo del *Google pagerank*, l'autovalore dominante è $\lambda = 1$ quindi il raggio spettrale di tale matrice è esattamente 1. Poiché il raggio spettrale della matrice di iterazione del metodo di Jacobi è minore del raggio spettrale della matrice del *Google pagerank*, il metodo di Jacobi converge in asintoticamente in un numero minore di iterazioni, rispetto al metodo delle potenze per il calcolo del *Google pagerank*.

6.7 Esercizio 7

Dimostrare che, se A è diagonale dominante, per riga o per colonna, il metodo di Jacobi è convergente.
Soluzione: Nel metodo di Jacobi si ha $A = M - N = D - (L + U)$ quindi

$$M = \begin{pmatrix} a_{1,1} & & & \\ & \ddots & & \\ & & \ddots & \\ & & & a_{n,n} \end{pmatrix} \text{ e } N = \begin{pmatrix} 0 & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{n-1,n} \\ a_{n,1} & \dots & a_{n,n-1} & 0 \end{pmatrix}.$$

Si ha

$$B = M^{-1}N = \begin{pmatrix} \frac{1}{a_{1,1}} & & & \\ & \ddots & & \\ & & \ddots & \\ & & & \frac{1}{a_{n,n}} \end{pmatrix} \begin{pmatrix} 0 & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{n-1,n} \\ a_{n,1} & \dots & a_{n,n-1} & 0 \end{pmatrix} =$$

$$= \begin{pmatrix} 0 & \frac{a_{1,2}}{a_{1,1}} & \dots & \frac{a_{1,n}}{a_{1,1}} \\ \frac{a_{2,1}}{a_{2,2}} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \frac{a_{n-1,n}}{a_{n-1,n-1}} \\ \frac{a_{n,1}}{a_{n,n}} & \dots & \frac{a_{n,n-1}}{a_{n,n}} & 0 \end{pmatrix},$$

per il Teorema di Gershgorin risulta

$$\mathcal{D}_i = \left\{ \lambda \in \mathbb{C} \,:\, |\lambda - b_{i,i}| \leq \sum_{j=1\ j\neq i}^{n} |b_{i,j}| \right\} = \left\{ \lambda \in \mathbb{C} \,:\, |\lambda| \leq \sum_{j=1\ j\neq i}^{n} \left| \frac{a_{i,j}}{a_{i,i}} \right| \right\};$$

supposta A a diagonale dominante per righe, $|a_{i,i}| \geq \sum_{j=1,\ j\neq i}^{n} |a_{i,j}|$, risulta
$|\lambda| \leq \frac{1}{|a_{i,i}|} \sum_{j=1\ j\neq i}^{n} a_{i,j} < 1$. Ogni $\mathcal{D}_i$ è centrato in 0 e ha raggio minore di 1 quindi
$\mathcal{D} = \bigcup_{i=1}^{n} \mathcal{D}_i$ è centrato in 0 e ha raggio pari al raggio massimo dei $\mathcal{D}_i$ ma sempre minore di 1.
Dato che $\lambda(A) = \lambda(A^T)$, il risultato vale anche se A è a diagonale dominante per colonne; il metodo di Jacobi è dunque convergente per matrici a diagonale dominante.

6.8 Esercizio 8

Dimostrare che, se A è diagonale dominante, per riga o per colonna, il metodo di Gauss-Seidel è convergente.
Soluzione: Nel metodo di Gauss-Seidel si ha $A = M - N = (D - L) - U$; sia $\lambda \in \sigma(M^{-1}1N)$ quindi λ è tale che $\det(M^{-1}N - \lambda I) = \det(M^{-1}(N - \lambda M)) = \det(M^{-1}) \det(N - \lambda M) = 0$.
Dato che, per definizione di splitting, $\det(M^{-1}) \neq 0$ deve risultare
$\det(N - \lambda M) = 0 = \det(\lambda M - N)$; sia $H = \lambda M - N$ matrice singolare e supponiamo, per assurdo, $|\lambda| \geq 1$. Risulta

$$H = \begin{cases} \lambda a_{i,j} & \text{se } i \geq j, \\ a_{i,j} & \text{altrimenti,} \end{cases};$$

quindi H è a diagonale dominante ma

$$\sum_{j=1\ j\neq i}^{n} |h_{i,j}| \leq |\lambda| \sum_{j=1\ j\neq i}^{n} |a_{i,j}| < |\lambda||a_{i,i}| = |h_{i,i}|.$$

Si ha una contraddizione poiché le matrici a diagonale dominanti sono non signolari, dunque $|\lambda| < 1$ e quindi il metodo di Gauss-Seidel è convergente per matrici a diagonale dominante.

6.9 Esercizio 9

Se A è sdp, il metodo di Gauss-Seidel risulta essere convergente. Dimostrare questo risultato nel caso (assai più semplice) in cui l'autovalore di massimo modulo della matrice di iterazione sia reale.
(Suggerimento: considerare il sistema lineare equivalente

$$(D^{-\frac{1}{2}}AD^{-\frac{1}{2}})(D^{\frac{1}{2}}\underline{x}) = (D^{-\frac{1}{2}}\underline{b}), \qquad D^{\frac{1}{2}} = diag(\sqrt{a_{11}}, \ldots, \sqrt{a_{nn}}),$$

la cui matrice dei coefficienti è ancora sdp ma ha diagonale unitaria, ovvero del tipo $I - L - L^T$. Osservare quindi che, per ogni vettore reale $\underline{v}$ di norma 1,si ha:
$\underline{v}^T L\underline{v} = \underline{v}^T L^T \underline{v} = \frac{1}{2}\underline{v}^T(L + L^T)\underline{v} < \frac{1}{2}$.)
Soluzione:
Scriviamo il sistema $A\underline{x} = \underline{b}$ nella forma equivalente

$$\left(D^{-1/2}AD^{-1/2}\right)\left(D^{1/2}\underline{x}\right) = \left(D^{-1/2}\underline{b}\right)$$

con $D = diag(\sqrt{a_{1,1}}, \ldots, \sqrt{a_{n,n}})$. La matrice $C = \left(D^{-1/2}AD^{-1/2}\right)$ ha diagonale unitaria:

$$c_{i,i} = d_i^{-1}a_{i,i}d_i^{-1} = \frac{1}{\sqrt{a_{i,i}}}a_{i,i}\frac{1}{\sqrt{a_{i,i}}} = a_{i,i};$$

inoltre è ancora sdp e scrivibile come $C = I - L - L^T$.
Poiché C è sdp risulta $\underline{v}^T A\underline{v} > 0, \forall \underline{v} \neq \underline{0}$ cioè

$$\underline{v}^T\underline{v} > \underline{v}^T L\underline{v} + \underline{v}^T L^T\underline{v} \quad \Rightarrow \quad \underline{v}^T L\underline{v} = \underline{v}^T L^T\underline{v} < \frac{1}{2}.$$

Sia $|\lambda| = \rho(M_{GS}^{-1}N_{GS}) = \rho\left((I - L)^{-1}L^T\right)$ assunto reale e $\underline{v}$ il corrispondente autovettore, dunque

$$(I - L)^{-1}L^T = \lambda\underline{v} \quad \Rightarrow \quad \lambda\underline{v} = L^T\underline{v} + \lambda L\underline{v}$$

ovvero $\lambda = \underline{v}^T L\underline{v} + \lambda\underline{v}^T L\underline{v} = (1 + \lambda)\underline{v}^T L)\underline{v}$ da cui

$$\frac{\lambda}{1 + \lambda} = \underline{v}^T L\underline{v} < \frac{1}{2} \quad \Rightarrow \quad -1 < \lambda < 1.$$

Segue $\rho(M_{GS}^{-1}N_{GS}) = |\lambda| < 1$.

6.10 Esercizio 10

Con riferimento ai vettori errore (6.16) e residuo (6.17) dimostrare che, se

$$||\underline{r_k}|| \leq \varepsilon ||\underline{b}||, \tag{6.1}$$

allora

$$||\underline{e_k}|| \leq \varepsilon k(A) ||\underline{\hat{x}}||,$$

dove $k(A)$ denota, al solito, il numero di condizionamento della matrice A. Concludere che, per sistemi lineari malcondizionati, anche la risoluzione iterativa (al pari di quella diretta) risulta essere più problematica.

Soluzione:

Posto $\underline{e}_k = \underline{x}_k - \underline{\hat{x}}$ e $\underline{r}_k = A\underline{x}_k - \underline{b}$, risulta

$$A\underline{e}_k = A(\underline{x}_k - \underline{\hat{x}}) = A\underline{x}_k - A\underline{\hat{x}} = A\underline{x}_k - b = \underline{r}_k.$$

Segue, passando alle norme,

$$\begin{aligned} ||\underline{e}_k|| = \left|\left|A^{-1}\underline{r}_k\right|\right| \leq \left|\left|A^{-1}\right|\right| \cdot ||\underline{r}_k|| \leq \left|\left|A^{-1}\right|\right| \cdot \varepsilon ||\underline{b}|| \leq \\ \leq \left|\left|A^{-1}\right|\right| \cdot ||A|| \cdot ||\underline{\hat{x}}|| = \varepsilon \kappa(A) ||\underline{\hat{x}}||, \end{aligned} \tag{6.2}$$

ovvero

$$\frac{||\underline{e}_k||}{||\underline{\hat{x}}||} \leq \varepsilon \kappa(A).$$

La risoluzione iterativa, come la risoluzione diretta, di sistemi lineari è ben condizionata per $\kappa(A) \approx 1$ mentre risulta malcondizionata per $\kappa(A) \gg 1$.

6.11 Esercizio 11

Calcolare il polinomio caratteristico della matrice

$$\begin{pmatrix} 0 & \dots & 0 & \alpha \\ 1 & \ddots & & 0 \\ & \ddots & \ddots & \vdots \\ 0 & & 1 & 0 \end{pmatrix} \in \mathbb{R}^{n \times n}.$$

Soluzione:

Il polinomio caratteristico è dato del determinante della matrice $A - \lambda I$:

$$\begin{aligned} \det(A - \lambda I) = & \det \begin{pmatrix} -\lambda & \dots & 0 & \alpha \\ 1 & \ddots & & 0 \\ & \ddots & \ddots & \vdots \\ 0 & & 1 & -\lambda \end{pmatrix} = \\ = & (-1)^n \lambda^n + (-1)^{n+1} \alpha = (-1)^n (\lambda^n - \alpha). \end{aligned} \tag{6.3}$$

Le radici di tale polinomio sono $\lambda = \sqrt[n]{\alpha}$.

6.12 Esercizio 12

Dimostrare che i metodi di Jacobi e Gauss-Seidel possono essere utilizzati per la risoluzione del sistema lineare (gli elementi non indicati sono da intendersi nulli)

$$\begin{pmatrix} 1 & & & -\frac{1}{2} \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{pmatrix} \underline{x} = \begin{pmatrix} \frac{1}{2} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{R}^n,$$

la cui soluzione è $\underline{x} = (1, \ldots, 1)^T \in \mathbb{R}^n$. *Confrontare il numero di iterazioni richieste dai due metodi per soddisfare lo stesso criterio di arresto (6.19), per valori crescenti di* n *e per tolleranze* ε *decrescenti. Riportare i risultati ottenuti in una tabella* (n/ε).

Soluzione:

La matrice è una M-matrice:

$$A = \begin{pmatrix} 1 & & & -1/2 \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{pmatrix} = I - B \text{ con } B = \begin{pmatrix} 0 & & & 1/2 \\ 1 & \ddots & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{pmatrix} > 0.$$

Si dimostra che $\rho(B) < 1$ calcolando gli autovalori della matrice B:

$$\det(B - \lambda I) =$$

$$\det \begin{pmatrix} -\lambda & & & 1/2 \\ 1 & \ddots & & \\ & \ddots & \ddots & \\ & & 1 & -\lambda \end{pmatrix} = -\lambda \det \begin{pmatrix} -\lambda & & \\ 1 & \ddots & \\ & \ddots & \ddots \\ & & 1 \end{pmatrix}_{(n-1)\times(n-1)} +$$

$$+(-1)^{n+1}\frac{1}{2}\det \begin{pmatrix} 1 & -\lambda & & \\ & \ddots & \ddots & \\ & & \ddots & -\lambda \\ & & & 1 \end{pmatrix}_{(n-1)\times(n-1)} = -\lambda(-\lambda)^{n-1} + (-1)^{n+1}\frac{1}{2} =$$

$$= (-1)^{n-2}\lambda^n + (-1)^{n+1}\frac{1}{2} = (-1)^n\frac{1}{2}(2\lambda^n - 1).$$

Quindi $\det(B - \lambda I) = 0$ se e solo se $2\lambda^n - 1 = 0$ se e solo se $\lambda = 2^{-1/n}$. Poiché $|\lambda| < 1$, $\rho(B) < 1$ quindi A è una M-matrice ed è possibile risolvere il sistema lineare tramite i metodo iterativi di Jacobi e Gauss-Seidel in quando lo splitting è regolare ed A converge.
Implementando il criterio d'arresto $||\underline{r}_k|| \leq \varepsilon||\underline{b}||$, si ha convergenza nel numero di iterazioni riportate nelle seguenti tabelle:

Iterazioni del metodo di Jacobi

$n\backslash\varepsilon$	10^{-1}	10^{-2}	10^{-3}	10^{-14}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
5	25	34	54	74	85	105	124	139	157	171
10	50	72	116	145	181	211	250	276	304	342
15	87	125	180	230	284	326	379	430	465	524
20	95	175	239	298	370	433	512	573	628	702
25	128	217	299	375	468	549	643	720	800	885
30	162	265	378	475	570	659	762	870	964	1066
35	199	311	436	533	662	775	905	1014	1120	1249
40	214	384	494	635	755	889	1037	1157	1299	1429
45	252	423	556	703	853	1020	1165	1327	1448	1607
50	299	458	622	787	963	1108	1290	1473	1630	1789

Iterazioni del metodo di Gauss-Seidel

$n\backslash\varepsilon$	10^{-1}	10^{-2}	10^{-3}	10^{-14}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
5	3	7	10	13	17	20	22	26	30	33
10	1	6	10	13	16	20	23	26	27	33
15	4	6	9	12	17	19	21	27	27	33
20	2	5	10	12	15	20	23	27	28	33
25	4	4	10	12	16	20	24	23	28	33
30	1	7	7	13	17	19	23	27	29	33
35	2	7	10	11	17	19	23	26	30	32
40	1	7	9	12	17	20	18	27	30	25
45	1	3	9	13	15	20	23	27	30	32
50	2	3	10	7	15	19	23	27	27	31

Si nota come il numero di iterazioni richieste dal metodo di Gauss-Seidel sia molto minore del numero richiesto dal metodo di Jacobi, per ogni valore della tolleranza.