Appunti per le lezioni di Calcolo Numerico

Prof. Flavio Sartoretto

14 marzo 2000

Indice

Capitolo 1

Introduzione.

F. Sartoretto
Calcolo Numerico

Testi adottati:

- G. Gambolati, *Lezioni di Metodi Numerici per Ingegneria e Scienze Applicate*, Cortina, Padova, 1994.
- F. Sartoretto e M. Putti, *Fortran per applicazioni numeriche*, Edizioni Libreria Progetto, Padova, 2000.

Prerequisiti:

- Analisi I, con cenni sui numeri complessi.
- Analisi II: risoluzione di equzioni differenziali ordinarie lineari a coefficienti costanti, calcolo di derivate parziali.
- Algebra delle matrici.

Notazioni:

- $a \gg b$ significa *"a" é molto piú grande di "b".*
- $a \ll b$ significa *"a" é molto piú piccolo di "b".*
- $a \simeq b$ significa *"a" é approssimativamente uguale a "b".*
- $f(x) = O(g(x))$ **per** $x \to a$ (si legge $f(x)$ *é di ordine "O-grande" di* $g(x)$ *per* $x \to a$), significa che la funzione

$$\left|\frac{f(x)}{g(x)}\right|$$

é limitata per $x \to a$, ossia esiste un intorno I_a di a ed esiste $M \in \Re^+$ t.c. $(\forall x \in I_a)\ |f(x)/g(x)| \leq M$.

- $f(x) = o(g(x))$ **per** $x \to a$ (si legge $f(x)$ *é di ordine "o-piccolo" di* $g(x)$ *per* $x \to a$), significa che

$$\lim_{x \to a} \left| \frac{f(x)}{g(x)} \right| = 0.$$

Esempi:

- $\sin(x) = O(1)$ per $x \to \infty$. Infatti $|\sin(x)/1| \leq 1$. Notare che *non esiste* il $\lim_{x\to\infty} \sin(x)/1$.
- $2(n+1)n = O(n^2)$ per $n \to +\infty$. Infatti $\lim_{n\to+\infty} \left|2(n+1)n/n^2\right| = 2$.
- $n(n+1) = o(n^k)$, $k > 2$, per $n \to +\infty$. Infatti se $k > 2$ risulta $\lim_{n\to+\infty} \left|(n+1)n/n^k\right| = 0$.

1.1 Concetti di base.

Calcolo Numerico:

Disciplina che tenta di dare una risposta numerica ad un problema matematico usando un elaboratore digitale.

Obiettivo del Calcolo Numerico é

Trovare gli algoritmi che risolvono un problema matematico • nel minor tempo, • con il minimo impiego di risorse di calcolo, • con la massima accuratezza possibile.

Capitolo 2

Rappresentazione dei numeri.

2.1 Pentium FDIV flaw

```
Eseguendo sul Pentium l' operazione

 EXA := (5-1/1000000)/(15-1/1000000);

Si ottiene il valore 0.333329, con un errore relativo:

 (EXA-0.333329)/EXA =   1.287461e-5

Eseguendo la stessa operazione su un 486 si ottiene 0.33333329.
L' errore relativo e':

 (EXA-33333329/100000000)/EXA =

                              1666671
                        - ---------------
                          499999900000000
ossia circa

                      - 3.333343e-9
```

2.2 Esempi di errori di arrotondamento.

La rappresentazione con un numero finito di cifre dá luogo ad errori. Ecco i valori ottenuti calcolando le quantità $xis = x_i^{(s)} = x_{i-1} + h$, $x_0^{(s)} = x_0$, $xip =$

$x_i^{(p)} = x_0 + ih$, $i = 1, \ldots, n$. Il valore iniziale esatto è $x_0 = 1.3$.

```
Risultati ottenuti sul VAX con VAX FORTRAN
i      xis                  xip
0  1.29999995231628418  1.29999995231628418
1  1.34999990463256836  1.34999990463256836
2  1.39999985694885254  1.39999997615814209
3  1.44999980926513672  1.44999992847442627
4  1.49999976158142090  1.50000000000000000
5  1.54999971389770508  1.54999995231628418
6  1.59999966621398926  1.60000002384185791
7  1.64999961853027344  1.64999997615814209
8  1.69999957084655762  1.69999992847442627
9  1.74999952316284180  1.75000000000000000
```

2.3 Schemi ricorrenti.

Esercizio 2.3.1 Per calcolare i valori dei polinomi ortogonali $T_n^*(x)$, $n = 2, \ldots, 20$, per $x = 2$ e $x = 1/2$, si vuole utilizzare lo schema ricorrente

$$\begin{cases} \quad T_{n+1}^*(x) & = (-2+4x)T_n^*(x) - T_{n-1}^*(x) \\ T_0^*(x) = 1, & T_1^*(x) = x \end{cases}$$

Supponiamo che T_0^* e T_1^* siano noti con errori ϵ_0 e ϵ_1 che siano uguali in modulo.

Nel caso $x = 2$ lo schema si puó ritenere stabile?

Nel caso $x = 1/2$ lo schema si puó ritenere stabile?

Soluzione.

Le seguenti tabelle riassumono i risultati che si ottengono calcolando alcuni termini della formula ricorrente. Caso $x = 2$.

$$\begin{pmatrix} \epsilon_0 = \epsilon_1 & \epsilon_0 = -\epsilon_1 \\ \epsilon_0 & -\epsilon_0 \\ 5\,\epsilon_0 & -7\,\epsilon_0 \\ 29\,\epsilon_0 & -41\,\epsilon_0 \\ 169\,\epsilon_0 & -239\,\epsilon_0 \end{pmatrix}$$

Caso $x = 1/2$.

$$\begin{pmatrix} \epsilon_0 = \epsilon_1 & \epsilon_0 = -\epsilon_1 \\ \epsilon_0 & -\epsilon_0 \\ -\epsilon_0 & -\epsilon_0 \\ -\epsilon_0 & \epsilon_0 \\ \epsilon_0 & \epsilon_0 \\ \epsilon_0 & -\epsilon_0 \\ -\epsilon_0 & -\epsilon_0 \\ -\epsilon_0 & \epsilon_0 \\ \epsilon_0 & \epsilon_0 \end{pmatrix}$$

Quindi nel caso $x = 2$ lo schema **non** é stabile, lo é nel caso $x = 1/2$.

Capitolo 3

Risoluzione di equazioni non lineari.

Esercizio 3.0.2 Calcolare con uno scarto relativo minore o uguale a 10^{-3} il valore di ξ t.c.

$$ln(\xi) = cos(\xi) \tag{3.1}$$

Risolvere questo problema

1. determinando un intervallo in cui cade la soluzione.
2. Eseguendo una iterazione con il metodo della bisezione, in modo da ottenere una stima $\bar{x_0}$ della soluzione,
3. eseguendo un numero opportuno di iterazioni con lo schema di Newton-Raphson, dopo aver posto $x_0 = \bar{x_0}$.

Confrontare il numero di iterazioni effettuate con il metodo di Newton-Raphson con la stima che si ottiene usando la relazione asintotica sugli errori.

Arrotondare i risultati intermedi a 5 cifre decimali.

Soluzione: schizzando le funzioni $y = \ln(x)$ e $z = \cos(x)$ si vede facilmente (cf figura 3.1) che esiste una sola soluzione dell' equazione data, che cade ad esempio nell'intervallo $I = [1, \pi/2]$.

Trovare il valore ξ che soddisfa l' equazione equivale a trovare lo zero della funzione

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

Applicando il metodo di bisezione all' intervallo I, posto r_i = stima della soluzione all' i-esima iterazione, $e_i = |(t_i - s_i)/(2r_i)|$, otteniamo:

```
i   s_i       t_i       r_i       e_i
0  1.0000    1.5708    1.2854    0.2220
1  1.2854    1.5708    1.4281    0.0999
```

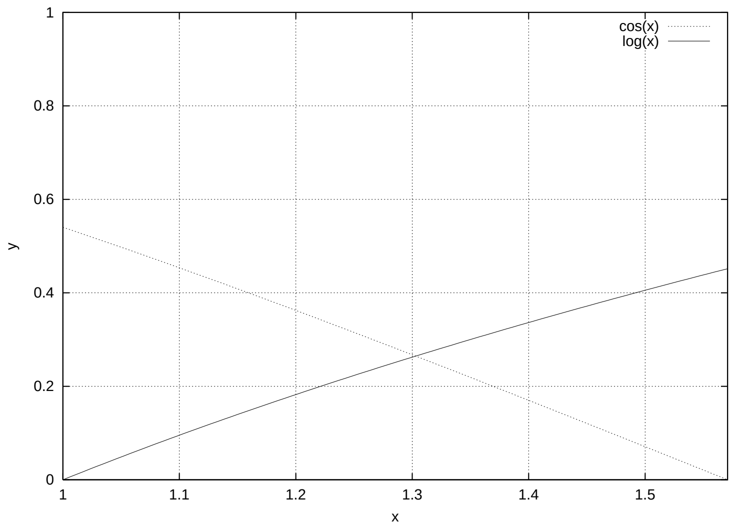


Figura 3.1: Grafico delle due funzioni.

Posto ora $x_0 = 1.4281$, $e_i = |(x_i - x_{i-1})/x_i|$ e usando lo schema di Newton-Raphson, otteniamo:

```
i  x_i       e_i
0  1.4281      ?
1  1.3014    0.0974
2  1.3030    0.0012
3  1.3030         0
```

Poiché lo scarto relativo é decrescente e alla terza iterazione é minore della tolleranza richiesta, alla terza iterazione si ha l' approssimazione richiesta, che, arrotondata alla quinta cifra decimale é $x = 1.3030$.

La relazione asintotica tra gli errori stabilisce che

$$|\epsilon_{n+1}| = \left|\frac{f''(\xi)}{2f'(\xi)}\epsilon_n^2\right| = |A||\epsilon_n^2|$$

che in questo caso diventa

$$|\epsilon_{n+1}| \simeq \left|(-9.3655E-2)\epsilon_n^2\right|$$

Ora

$$|\epsilon_n| = \left|A^{(2^n-1)}\epsilon_0^{(2^n)}\right|$$

perció deve essere

$$\left|\frac{\epsilon_n}{\xi}\right| = \left|\frac{A^{(2^n-1)}\epsilon_0^{(2^n)}}{\xi}\right| \leq 10^{-3}$$

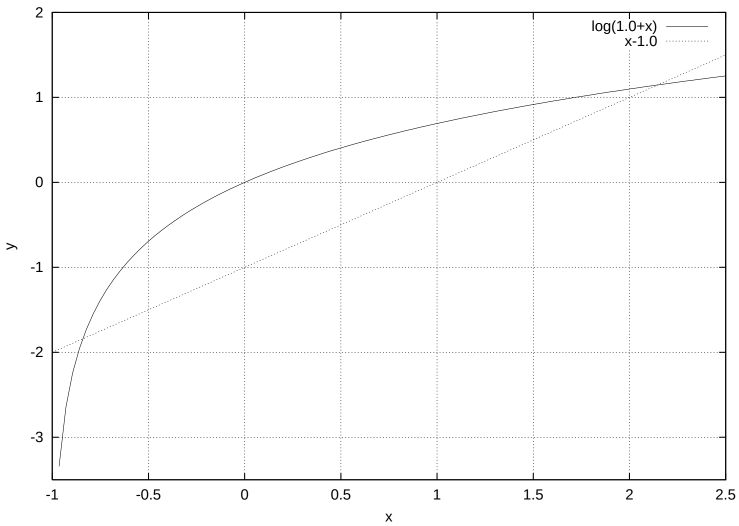


Figura 3.2: Grafico delle funzioni $y = x - 1$ e $y = \ln(1 + x)$.

ossia, passando ai logaritmi,

$$2^n(\log(|A|) + \log(|\epsilon_0|)) - log(|A|) \le \log(|\xi|10^{-3})$$

ed essendo $\epsilon_0 = -1.2510E - 1$, si ottiene $n \ge 2$, che ben si confronta con il numero di iterazioni realmente effetuate.

Esercizio 3.0.3 Consideriamo lo schema del punto fisso

$$\begin{aligned} x_{n+1} &= ln(1 + x_n) + 1, \quad n = 0, 1, \dots && (3.2) \\ x_0 &= 2 && (3.3) \end{aligned}$$

1. Provare che questo schema converge ad un valore ξ.
2. Usando la stima a priori sull' errore, valutare il numero di iterazioni k necessario perché x_k sia affetto da un errore relativo $\epsilon_k \le 0.003$.
3. Usando la stima a posteriori sull' errore, eseguire il numero n di iterazioni necessario perché l' errore relativo sia $\epsilon_n \le 0.003$.

Arrotondare i risultati intermedi a 5 cifre decimali.

Soluzione: Se lo schema converge, deve convergere alle soluzioni di

$$x = g(x) = ln(1 + x) + 1$$

che si vede (cf figura 3.2) sono una negativa ξ^-, e una positiva ξ.

Studiando la funzione, si puó vedere che é $2 < \xi < 2.5$. Essendo $g'(x) = 1/(1 + x)$, posto ad esempio $I = [2, 3]$ risulta

- $g(I) \subset I$
- $g(x)$ é contrattiva in I, dato che

$$m = \max_{x \in I} |g'(x)| = \frac{1}{3} < 1.$$

Dato che $x_0 \in I$, lo schema converge a ξ.

Si ha $x_0 = 2$, $x_1 = g(x_0) \simeq 2.0986$. La stima a priori sull' errore ci dice che

$$\left| \frac{\xi - x_n}{\xi} \right| \leq \left| \frac{\xi - x_n}{2} \right| \leq \frac{1}{2} \left| \frac{m^n}{1-m} \right| |x_1 - x_0| \simeq (7.3950E-2)(1/3)^n$$

quindi deve essere

$$(7.3950E-2)(1/3)^n \leq 0.003$$

ossia, passando ai logaritmi,

$$n \geq \frac{\ln(0.003/7.3950E-2)}{\ln(1/3)} \simeq 2.9171$$

ossia $n \geq 3$.

Utilizzando la stima a posteriori

$$\epsilon_i = \left| \frac{\xi - x_i}{\xi} \right| \leq \frac{|\xi - x_i|}{2} \leq (1/2) \frac{m}{1-m} |x_i - x_{i-1}| = \texttt{eps(i)}$$

Otteniamo i risultati

```
i  xi       eps(i)
0  2.0000     ?
1  2.0986   2.4650E-2
2  2.1310   8.0855E-3
3  2.1413   2.5959E-3
```

ed essendo $\epsilon_3 < 0.003$, ci possiamo fermare alla terza iterazione e il risultato é $\xi = 2.1413$.

Capitolo 4

Risoluzione di sistemi lineari.

4.1 Matrici speciali.

Tipo matrice	Proprietá	Elementi in	Autovalori	Autovettori
Hermitiana	$H^+ = H$	$\mathcal{C}$	reali	ortonorm.*
Hermitiana e definita positiva	$H^+ = H$ $u^+ H u > 0$	$\mathcal{C}$	reali positivi	ortonorm.*
Simmetrica	$H^T = H$	$\Re$	reali	ortonorm.*
Simmetrica e definita positiva	$H^T = H$ $u^+ H u > 0$	$\Re$	reali positivi	ortonorm.*
Antisimmetrica	$H^+ = -H$	$\mathcal{C}$	nulli o con parte reale nulla	ortonorm.*
Antisimmetrica e reale	$H^+ = -H$	$\Re$	nulli o con parte reale nulla + a coppie di compl. coniug.	ortonorm.*
Unitaria	$U^+ U = I = U U^+$ $U^+ = U^{-1}$	$\mathcal{C}$	$\|\lambda\| = 1$	?
Ortogonale	$U^T U = I = U U^T$ $U^T = U^{-1}$	$\Re$	$\|\lambda\| = 1$ $\lambda = e^{\pm i\phi}$	?

* = Gli autovettori *possono essere scelti in modo da* essere ortonormali.

4.2 Decomposizione a valori singolari

Mediante matrici ortogonali è possibile diagonalizzare una qualsiasi matrice $A \in \mathbb{R}^{m \times n}$.[1] Vale il seguente

[1] Sezione sviluppata da M. Lizza

Teorema 4.2.1 *(SVD) Sia $A \in \mathbb{R}^{m\times n}$. Esistono due matrici ortogonali $U \in \mathbb{R}^{m\times m}$ e $V \in \mathbb{R}^{n\times n}$ tali che*

$$U^T AV = \Sigma = diag(\sigma_i). \tag{4.1}$$

$\Sigma \in \mathbb{R}^{m\times n}$ è una matrice diagonale con elementi σ_i tali che se r è il rango di A,

$$\sigma_1 \geq \ldots \geq \sigma_r > \sigma_{r+1} = \ldots = \sigma_n = 0. \tag{4.2}$$

Per la dimostrazione, si veda [10]. Le quantitá σ_i vengono chiamate i *valori singolari di A*. $A^T A$ è una matrice simmetrica semi-definita positiva, in quanto per ogni vettore $x \in \mathbb{R}^n$

$$x^T \left(A^T A\right) x = \left(x^T A^T\right) (Ax) = (xA)^T (Ax) = \|Ax\|_2^2 \geq 0. \tag{4.3}$$

I suoi autovalori λ_i sono, pertanto, reali non negativi e risulta $\lambda_i = \sigma_i^2$. Risultato analogo vale per la matrice AA^T. Le colonne u_i di U sono dette *vettori singolari sinistri*, le colonne v_i di V *vettori singolari destri* e rappresentano, rispettivamente, gli autovettori di AA^T e di $A^T A$. Infatti, $AA^T U = (U\Sigma V^T)(U\Sigma V^T)^T U = U\Sigma V^T V^T \Sigma U^T U = U\Sigma^2$ e $A^T AV = (U\Sigma V^T)^T (U\Sigma V^T)V = V\Sigma U^T U\Sigma V^T V = V\Sigma^2$.

Corollario 4.2.1 *Ogni matrice $A \in \mathbb{R}^{m\times n}$ ammette una fattorizzazione*

$$A = U\Sigma V^T, \quad \Sigma = diag(\sigma_i), \tag{4.4}$$

dove $U \in \mathbb{R}^{m\times m}$ e $V \in \mathbb{R}^{n\times n}$ sono matrici ortogonali e $\Sigma \in \mathbb{R}^{m\times n}$ è una matrice diagonale tale che $\sigma_1 \geq \ldots \geq \sigma_n \geq 0$.

Si puó dimostrare che [10]

$$\|A\|_F = \left(\sigma_1^2 + \ldots + \sigma_r^2\right)^{1/2}, \quad \|A\|_2 = \sigma_1.$$

Infatti $\|A\|_F^2 = \text{traccia}(A^T A) = \text{traccia}(V\Sigma^2 V^T) = \sum_i \lambda_i(V\Sigma^2 V^T) = \sum_i \lambda_i(\Sigma^2) = \sum_i \sigma_i^2$. Inoltre $\|A\|_2^2 = \rho(A^T A) = \sigma_1^2$.

La SVD di una matrice ha interessanti proprietá, alcune condivise da altre fattorizzazioni, come la fattorizzazione QR con pivoting sulle colonne, altre peculiari della SVD.

4.2.1 Un esempio

Consideriamo la matrice quadrata

$$A = \begin{vmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{vmatrix}$$

Il suo rango è 2. La decomposizione a valori singolari $A = U\Sigma V^T$ è:

$$U = \begin{vmatrix} -0.6280 & 0.3251 & -0.7071 \\ -0.4597 & -0.8881 & 0 \\ -0.6280 & 0.3251 & 0.7071 \end{vmatrix}$$

$$\Sigma = \begin{vmatrix} 2.1753 & 0 & 0 \\ 0 & 1.1260 & 0 \\ 0 & 0 & 0 \end{vmatrix}$$

$$V = \begin{vmatrix} -0.5774 & 0.5774 & -0.5774 \\ -0.2113 & -0.7887 & -0.5774 \\ -0.7887 & -0.2113 & 0.5774 \end{vmatrix}$$

Le matrici U e V sono ortonormali. Osserviamo che Σ ha due elementi non nulli, pari al rango di A. Costruiamo la matrice

$$A_1 = U_1 \Sigma_1 V_1^T = |2.1753| \cdot \begin{vmatrix} -0.6280 \\ -0.4597 \\ -0.6280 \end{vmatrix} \cdot \begin{vmatrix} -0.5774 & 0.5774 & -0.5774 \end{vmatrix}$$

Risulta

$$\|A - A_1\|_2 = 1.1260 = \sigma_2.$$

Come vedremo più avanti, questo risultato ha portata generale.

4.2.2 Spazi vettoriali

Dalle equazioni (4.1) e (4.4) si ottiene

$$\begin{cases} AV = U\Sigma \\ A^T U = V\Sigma^T \end{cases},$$

ossia

$$\begin{cases} Av_i = \sigma_i u_i \\ A^T u_i = \sigma_i v_i \end{cases}, \quad i = 1, \ldots, \min\{m, n\}.$$

Dalla (4.2) si ricava che

$$\begin{array}{ll} \mathit{span}\,\{v_{r+1}, \ldots, v_n\} = \ker(A), & \mathit{span}\,\{u_{r+1}, \ldots, u_m\} = \mathrm{range}(A)^\perp, \\ \mathit{span}\,\{u_1, \ldots, u_r\} = \mathrm{range}(A), & \mathit{span}\,\{v_1, \ldots, v_r\} = \ker(A)^\perp, \end{array}$$

dove $\ker(A) = \{x \text{ t.c. } Ax = 0\}$, $\mathrm{range}(A) = \{y \text{ t.c. } y = Ax\}$, $S^\perp = \{y \text{ t.c. } x^T y = 0\, \forall x \in S\}$.

4.2.3 Rango di matrici

La nozione di rango è di fondamentale importanza nell'algebra lineare. Determinare numericamente il rango non è un problema banale. Piccole perturbazioni lo modificano sensibilmente. Vale il seguente

Teorema 4.2.2 (Eckart-Young) *Sia $A = U\Sigma V^T$. Se $k < r = rank(A)$ e*

$$A_k = \sum_{i=1}^{k} \sigma_i \cdot u_i \cdot {v_i}^T, \tag{4.5}$$

allora

$$\min_{rank(B)=k} \|A - B\|_2 = \|A - A_k\|_2 = \sigma_{k+1}\,. \tag{4.6}$$

Per la dimostrazione, si veda [10].

In altri termini, A_k è la migliore approssimazione di rango k della matrice A. Se $\sigma_k > \sigma_{k+1}$ essa è unica. Il teorema di Eckart-Young vale anche per la norma di Frobenius $\|\cdot\|_F$.

È possibile stimare l'errore relativo commesso quando la matrice A, di rango $r > k$, viene sostituita con una matrice di rango k:

$$e_k = \frac{\|A - A_k\|_2}{\|A\|_2} = \frac{\sigma_{k+1}}{\sigma_1} \tag{4.7}$$

Definizione 4.2.1 *Sia $\epsilon > 0$. L' ϵ-rango, r_ϵ, della matrice A è:*

$$r_\epsilon = \min_{\|A-B\|_2 \leq \epsilon} rank(B).$$

Dal Teorema 4.2.2 segue che r_ϵ è pari al numero di valori singolari strettamente maggiori di ϵ, ossia

$$\sigma_1 \geq \ldots \geq \sigma_{r_\epsilon} > \epsilon > \sigma_{r_\epsilon+1} = \ldots = \sigma_n = 0\,.$$

Le relazioni (4.5) valgono anche quando si mette A_r al posto di A, $r = \mathrm{rank}(A)$.

Vediamo alcune applicazioni della decomposizione a valori singolari.

4.2.4 Sistemi sottodeterminati

Sia $A \in \mathbb{R}^{m\times n}$, $m < n$, di rango r, e sia $b \in \mathbb{R}^m$. Il sistema lineare $Ax = b$ è *sottodeterminato*: ha più incognite che equazioni, perciò ha infinite soluzioni, oppure nessuna. Sostituendo ad A la sua SVD e ponendo $y = V^T x$, $c = U^T b$, si ottiene il sistema equivalente

$$\Sigma y = c,$$

la cui soluzione y ha componenti

$$\begin{cases} \sigma_j\, y_j = c_j, & j \leq m, \sigma_j \neq 0, \\ 0\, y_j = c_j, & j \leq m, \sigma_j = 0. \end{cases}$$

Il secondo insieme di equazioni è vuoto se $r = m$. Quindi il sistema ha soluzione se e solo se $c_j = 0$ quando $\sigma_j = 0$. Se esistono, le soluzioni y sono

$$\begin{cases} y_j = c_j/\sigma_j, & j \leq m, \sigma_j \neq 0, \\ y_j = \text{arbitrario}, & j \leq m, \sigma_j = 0, \text{ opp. } m \leq j \leq n. \end{cases}$$

Infine, le soluzioni del sistema lineare originario sono i vettori $x = Vy$.

4.2.5 Sistemi sovradeterminati

Sia $A \in \mathbb{R}^{m\times n}$, $m > n$, di rango r, e sia $b \in \mathbb{R}^m$. Il sistema lineare $Ax = b$ è *sovradeterminato*: ha più equazioni che incognite. Procedendo come nel caso precedente, abbiamo:

$$\begin{cases} \sigma_j y_j = c_j, & j \le n, \sigma_j \neq 0, \\ 0\, y_j = c_j, & j \le n, \sigma_j = 0, \\ 0 = c_j, & n < j \le m. \end{cases}$$

Se $c_j = 0$ quando $\sigma_j = 0$, oppure $j > n$, esistono soluzioni y e sono

$$\begin{cases} y_j = c_j/\sigma_j, & j \le n, \sigma_j \neq 0, \\ y_j = \text{arbitrario}, & j \le n, \sigma_j = 0. \end{cases}$$

Data la norma euclidea di vettore, $\|\cdot\|_2$, calcoliamo una soluzione “alternativa” minimizzando la quantitá

$$\|r\|_2{}^2 = \|Ax - b\|_2{}^2.$$

$\|r\|_2^2$ è minima sse $\partial\|r\|_2^2/\partial x_i = 0$, $i = 1, \ldots, n$. Queste relazioni equivalgono al sistema (vedi la sezione 6.1)

$$A^T A x = A^T b, \tag{4.8}$$

che è detto *sistema delle equazioni normali*. L'accuratezza della soluzione numerica risente molto del condizionamento di A, che peggiora in $A^T A$. Utilizzando la SVD è possibile risolvere il problema di minimizzazione senza passare per il sistema lineare (4.8). Sia $A = U\Sigma V^T$. Le matrici ortogonali preservano la norma euclidea, ossia se H è ortogonale, $\|Hv\|_2^2 = v^T H^T H v = v^T v = \|v\|_2^2$.

$$\|r\|_2{}^2 = \|U^T\left(AVV^T x - b\right)\|_2{}^2 = \|\Sigma V^T x - U^T b\|_2{}^2$$

da cui, ponendo $y = V^T x$ e $c = U^T b$, otteniamo

$$\|r\|_2{}^2 = \|\Sigma y - c\|_2{}^2.$$

Quindi $\|r\|_2^2 = 0$ se le componenti delle soluzioni y sono

$$\begin{cases} y_j = c_j/\sigma_j, & j \le n, \sigma_j \neq 0, \\ y_j = \text{arbitrario}, & j \le n, \sigma_j = 0. \end{cases}$$

Le soluzioni di (4.8) si ottengono dalla relazione $x = Vy$. Se A non ha rango massimo, esistono infinite soluzioni. Di solito si pone $y_j = 0$ per ogni j tale che $\sigma_j = 0$, ottenendo così la soluzione x di norma minima, dato che $\sum_i y_i^2 = \|y\|_2^2 = \|Vy\|_2^2 = \|x\|_2^2$.

4.2.6 Filtraggio del rumore

Molti problemi nel campo dell' elaborazione del segnale utilizzano un modello di tipo vettoriale, in cui le misurazioni vengono organizzate in una matrice $A \in \mathbb{R}^{m\times n}$ [6]. Le colonne $a_i \in \mathbb{R}^m$ sono $a_i = s_i + r_i$, dove s_i e r_i rappresentano rispettivamente la componente del segnale e quella del rumore. Il modello assume che l'errore possa essere separato dal dato, e che la componente rumore risieda in un sottospazio ortogonale a quello del segnale.

In assenza di rumore si può calcolare il rango di A. Purtroppo esso viene di solito sovrastimato a causa degli errori (non sistematici) di misurazione che "disturbano" i dati della matrice. Supponiamo che s_i risieda in uno spazio vettoriale di dimensione k. Calcoliamo la SVD di A. Dalle equazioni (4.5) risulta $s_i \in span\{u_1, \ldots, u_k\}, \quad r_i \in span\{u_{k+1}, \ldots, u_n\}$. Possiamo costruire la matrice A_k e stimare le due componenti (segnale e rumore) sfruttando il Teorema (4.2.2).

4.2.7 Compressione di immagini

Una immagine puó essere rappresentata mediante una matrice I *(bitmap)* di dimensioni $m \times n$ [8]. Supponiamo $m > n$. Il generico elemento, $I_{i,j}$, definisce il colore dell'elemento di immagine (*pixel*). Per le immagini in bianco e nero $I_{i,j}$ è un valore di luminositá (tipicamente compreso nell'intervallo $[\,0, 255\,]_{\mathbb{N}}$ cosicché può essere memorizzato in un *byte*). Figure anche semplici richiedono grandi matrici per essere rappresentate. Per memorizzare e/o trasmettere una matrice–immagine bisogna comprimerla. Gli algoritmi più efficienti per la compressione di immagini sono quelli che comportano una certa perdita di dettaglio (*lossy algortihms*).

Utilizzando la *decomposizione a valori singolari (SVD)* [10] è possibile definire una rappresentazione di dimensione ridotta che preserva la maggior parte della struttura del modello. Una volta calcolata la decomposizione a valori singolari $I = U\Sigma V^T$, si costruiscono le approssimazioni $I_k = U_k\Sigma_k {V_k}^T$. Sia $U = [u_1|\ldots|u_m]$, $V = [v_1|\ldots|v_n]$, $\Sigma = diag(\sigma_1, \ldots, \sigma_n)$, $\sigma_1 \geq \ldots \geq \sigma_r > \sigma_{r+1} = \ldots = \sigma_n = 0$. Poniamo $\Sigma_k = diag(\sigma_1, \ldots, \sigma_k)$, $U_k = [u_1|\ldots|u_k]$, $V_k = [v_1|\ldots|v_k]$. Calcoliamo la matrice ridotta di rango k

$$I_k = U_k\Sigma_k {V_k}^T. \tag{4.9}$$

Ci chiediamo se queste approssimazioni possono rimuovere i dettagli presenti mantenendo l'"essenza" dell'immagine.

Esaminiamo due immagini. I dati relativi sono riportati nella Tabella 4.1. Nelle Tabelle 4.2 e 4.3 sono riportati i risultati delle prove effettate. Il parametro di maggior interesse è il numero di complessivo di valori (detti anche *celle* o *pixel*), necessari per memorizzare le matrici U_k, Σ_k e ${V_k}^T$. Nelle Tabelle 4.4 e 4.5 vengono mostrate alcune delle immagini ottenute con le approssimazioni. Ad esempio si vede che per k=30 si riduce di nove volte l'occupazione di memoria in *gatlin* (cf tabelle 4.3 e 4.5) e si ha un'approssimazione di qualitá accettabile. Per dimensioni minori, la qualitá dell'immagine peggiora considerevolmente ma

Immagine	Dimensione	Pixel	Rango
earth.eps	250×257	64250	250
gatlin.eps	480×640	320720	480

Tabella 4.1: Caratteristiche delle immagini considerate

k	Celle
125	79000
62	35278
32	17248
15	7830
10	5170

Tabella 4.2: Risultati quantitativi di compressione *(earth.eps)*

k	Celle
240	326400
120	148800
60	70800
30	34500
15	17025

Tabella 4.3: Risultati quantitativi di compressione *(gatlin.eps)*

è ancora possibile farsi un'idea dell'aspetto "globale", soprattutto per quanto riguarda le regioni in cui compaiono linee orizzontali e/o verticali.

Il fattore di compressione non è paragonabile a quello offerto da algoritmi di compressione lossy come **JPEG** [13]. E' confrontabile con quello offerto da **GIF** [13], che però non è lossy. Abbiamo comunque un interessante esempio di come la decomposizione a valori singolari possa essere teoricamente utilizzata per comprimere immagini.

4.3 Esercizio sull' eliminazione di Gauss.

Esercizio.

Risolvere il sistema lineare

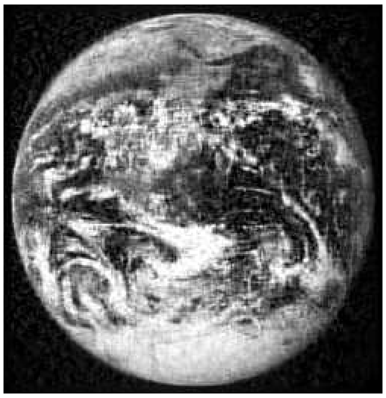
$$\begin{aligned} 10^{-2}x + y &= 1 \\ x + y &= 2 \end{aligned}$$

con il metodo di eliminazione di Gauss senza pivoting,

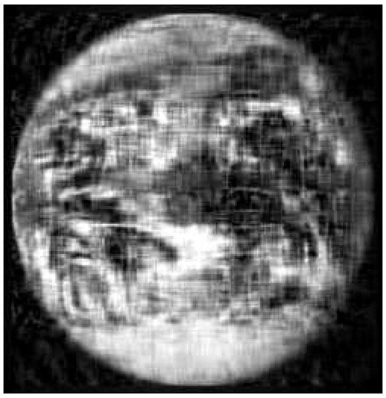
a) in aritmetica esatta, ottenendo cosíla soluzione **s**.

earth.eps

$k = 62$



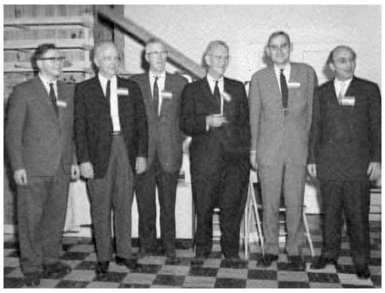
$k = 32$



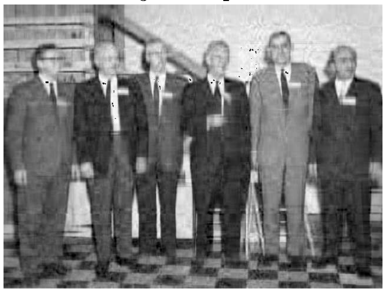
$k = 15$

Tabella 4.4: Risultati qualitativi di compressione *(earth.eps)*

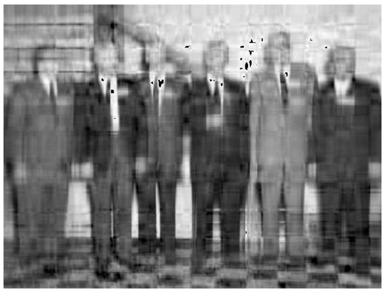
gatlin.eps



$k = 60$



$k = 30$



$k = 15$

Tabella 4.5: Risultati qualitativi di compressione *(gatlin.eps)*

b) In aritmetica a tre cifre decimali con virgola mobile normalizzata.

c) Come nel caso precedente, ma scambiando la prima riga con la seconda.

Nei casi b) e c) calcolare inoltre

$$\mathbf{e}_r = \frac{||\mathbf{e}||_2}{||\mathbf{s}||_2} \tag{4.10}$$

Caso a) I risultati sono:

```
  matrice A        vett. b
  0.01    1          1
  1       1          2
Eliminazione di Gauss
  0.01    1          1
  0     -99        -98
```

La soluzione è quindi: $\{x = 100/99, y = 98/99\}$.

Caso b) Si ottiene

```
      Matrice A          Vettore b
  1.00e-002  1.00e+000  1.00e+000
  1.00e+000  1.00e+000  2.00e+000
Eliminazione di Gauss
  1.00e-002  1.00e+000  1.00e+000
          0 -9.90e+001 -9.80e+001
Calcolo di y
  1.00e-002  1.00e+000  1.00e+000
          0  1.00e+000  9.90e-001
Calcolo di x
  1.00e+000          0  1.00e+000
          0  1.00e+000  9.90e-001
```

La soluzione è quindi: $\{x = 1.00E + 0, y = 9.90E - 1\}$. $\mathbf{e}_r \simeq 7.1420e - 3$.

Caso c) Si ottiene

```
      Matrice A          Vettore b
  1.00e+000  1.00e+000  2.00e+000
  1.00e-002  1.00e+000  1.00e+000
Eliminazione di Gauss
  1.00e+000  1.00e+000  2.00e+000
          0  9.90e-001  9.80e-001
```

```
Calcolo di y
  1.00e+000  1.00e+000  2.00e+000
          0  1.00e+000  9.90e-001
Calcolo di x
  1.00e+000          0  1.01e+000
          0  1.00e+000  9.90e-001
```

La soluzione è quindi: $\{x = 1.01E + 0, y = 9.90E - 1\}$. $\mathbf{e}_r \simeq 1.4143e - 4$.

Come si vede, nel caso c) il risultato é piú accurato e corrisponde ad una eliminazione con pivoting.

4.4 Costo del metodo di Jacobi

Consideriamo il metodo di Jacobi (J) per risolvere il sistema lineare di ordine n

$$\mathbf{Ax} = \mathbf{b}. \tag{4.11}$$

Stimiamo il costo computazionale $C[J]$ di tale metodo, inteso come numero di operazioni floating point da effettuare, ponendo $C[\oplus] = C[\otimes] = C[\oslash] = 1$, dove $\oplus$, $\otimes$, $\oslash$ sono la somma, moltiplicazione e divisione floating point.

Lo schema di Jacobi può essere messo nella forma [9]

$$\mathbf{x}_{k+1} = \mathbf{E}\mathbf{x}_k + D^{-1}\mathbf{b} = \mathbf{E}\mathbf{x}_k + \mathbf{c}, \tag{4.12}$$

La matrice di iterazione è $E = -D^{-1}(L + U)$.

Il costo di una iterazione consta di un prodotto matrice–vettore P e di una somma di vettori S. Assumendo di eseguire il prodotto P con l'algoritmo standard, il costo è di n^2 moltiplicazioni e $n^2 - n$ addizioni. Il costo di una somma di vettori è di n addizioni. Quindi il costo di una iterazione è $C[J_1(n)] = O(2n^2)]$.

Supponiamo di eseguire m iterazioni per ottenere un' approssimazione $\mathbf{x}_m$ tale che $\|\mathbf{e}_m\|/\|\mathbf{e}_0\| < \tau$, essendo τ una tolleranza prefissata, $\mathbf{e}_i$ l'errore alla i-esima iterazione. Allora $C[J(n)] = m \cdot O(2n^2)$.

Sia $\|\cdot\|$ la norma euclidea di vettori. Assumiamo che la matrice $\mathbf{A}$ sia *simmetrica*. Allora possiamo stimare m [9],

$$m > \frac{\log \tau}{\log \rho(\mathbf{E})} = m_0.$$

Assumiamo $m = m_0$. Dovendo esprimere $\rho(\mathbf{E})$ in funzione di n, consideriamo un caso particolare. Consideriamo la matrice tridiagonale $\tilde{\mathbf{A}}$ che ha gli elementi tutti nulli, tranne

$$\tilde{\mathbf{A}}_{ij} = \left\{ \begin{array}{rll} 8 & se & i = j, \\ -1 & se & |i - j| = 1 \end{array} \right. , \quad i, j = 1, \ldots, n.$$

Gli autovalori di $\tilde{\mathbf{A}}$ sono [9]

$$\lambda_{\tilde{\mathbf{A}}}(k) = 8 - \frac{\sin((n-1)k\pi/(n+1))}{\sin(nk\pi/(n+1))} = 8 - \mu(k), \quad k = 1, \ldots, n. \tag{4.13}$$

Poniamo $\mathbf{A} = \tilde{\mathbf{A}} + 10^{-6}$. Grazie alla disuguaglianza di Wielandt-Hoffman [14], si ha

$$\max_j \left| \lambda_j(\tilde{\mathbf{A}}) - \lambda_j(\mathbf{A}) \right| < 10^{-6}.$$

Perciò $\rho(\mathbf{E}) = \rho(-\mathbf{D}^{-1}(\mathbf{L}+\mathbf{U})) = \rho(-\mathbf{D}^{-1}(\mathbf{A}-\mathbf{D})) = \rho((1/8)\mathbf{A} - I) \simeq \max_k |(1/8)\mu(k)| = (1/8)\max_k |\mu(k)| = (1/8)\mu(n)$. Ora

$$\lim_{n\to\infty} (1/8)|\mu(n)| = 0.25. \tag{4.14}$$

Una stima asintotica è quindi $m_0(\tau) \simeq \log\tau / \log 0.25$.

Supponiamo di volere la stessa accuratezza di uno schema diretto, in doppia precisione, perciò fissiamo $\tau = 10^{-16}$. Allora $m_0 = -16/\log 0.25 \simeq 26.575$. Quindi poniamo $\bar{m} = 27$, ottenendo $C[J(n)] = \bar{m} \cdot O(2n^2) = 27 \cdot O(2n^2) = O(54n^2)$.

Capitolo 5

Interpolazione.

5.1 Operatori lineari.

Siano S, Z spazi di funzioni.

Un operatore é un funzionale

$$P : S \to Z. \tag{5.1}$$

Sia $\mathcal{D}(P) \subseteq S$ il dominio di P, ossia l' insieme delle funzioni per le quali P é definito e sia $\mathcal{R}(P) \subseteq Z$ la sua immagine o codominio.

Esempio: $D : C^0 \to C^0$ definito come

$$Df = \frac{df}{dx} = f' \tag{5.2}$$

é un operatore. Risulta $\mathcal{D}(D) = C^1$.

P, P_1, P_2, P_3 siano operatori t.c. $\mathcal{D}(P) \subseteq \mathcal{D}(P_1) \cap \mathcal{D}(P_2) \cap \mathcal{D}(P_3) = S \neq \emptyset$, $\mathcal{R}(P_1) \subseteq \mathcal{D}(P_2)$ e sia $f \in S$, α uno scalare. Le operazioni tra operatori si definiscono come

$$(P_1 + P_2)f = P_1 f + P_2 f \tag{5.3}$$
$$(P_2 P_1)f = P_2(P_1 f) \tag{5.4}$$
$$(\alpha P)f = \alpha(Pf) \tag{5.5}$$

Indichiamo l' identitá con 1: $1f = f, \forall f \in \mathcal{D}(P)$.

Se esiste un operatore P_D^{-1} t.c.

$$PP_D^{-1} = 1, \tag{5.6}$$

P_D^{-1} viene chiamato inverso destro di P.

Se esiste un operatore P_S^{-1} t.c.

$$P_S^{-1} P = 1, \tag{5.7}$$

P_S^{-1} viene chiamato inverso sinistro di P.

Si puó dimostrare che se esistono P_D^{-1} e P_S^{-1}, allora

$$P_S^{-1} = P_D^{-1} = P^{-1} \tag{5.8}$$

Definizione 5.1.1 Un operatore L é detto lineare se

- $\mathcal{D}(L)$ é uno spazio lineare,
- per tutti gli $f, g \in \mathcal{D}(L)$ e per tutti gli scalari α, β risulta

$$L(\alpha f + \beta g) = \alpha L f + \beta L g.$$

Capitolo 6

Approssimazione.

6.1 Approssimazione di funzioni.

Il problema dell' approssimazione di funzioni si può schematizzare nel seguente modo.

E' data una funzione $f(x)$ che vogliamo analizzare nell'intervallo $I = [a, b]$. Sia $S = \{a_0\phi_0(x) + \ldots + a_n\phi_n(x)\}$, a_i sono scalari, lo spazio lineare generato dalle funzioni $\phi_0(x), \ldots, \phi_n(x)$. Si vogliono calcolare i parametri $c_0, \ldots, c_n$, della funzione $\tilde{f}(x) = c_0\phi_0(x) + \ldots + c_n\phi_n(x)$ tale che

$$d(f, \tilde{f}) = \min_{g \in S} d(f, g) \tag{6.1}$$

dove $d(\cdot, \cdot)$ è una opportuna misura di "distanza" tra funzioni.

La scelta delle funzioni $\phi_i(x)$ è importante. Spesso si pone $\phi_i(x) = x^i$, in modo che S è lo spazio dei polinomi di grado minore o uguale a n e in tal caso si parla di **approssimazioni polinomiali**.

Tuttavia se è noto che la funzione da approssimare ha certe caratteristiche, sarà conveniente scegliere le funzioni $\phi_i(x)$ che meglio rappresentano queste caratteristiche. Ad esempio in alcuni casi si prende $\phi_k(x) = e^{kx}$.

La scelta della "misura di distanza" è anch'essa importante e varia a seconda del tipo di modello da cui scaturisce f. Siano $x_0, \ldots, x_n$ valori dati e sia

$$I = [a, b], \quad a = min\{x_0 \ldots, x_n\}, \quad b = max\{x_0 \ldots, x_n\},$$

il più piccolo intervallo che li contiene. Se f è nota nei punti $x_0, \ldots, x_n$, ossia sono noti i valori $y_i = f(x_i)$, $i = 1, \ldots, n$, una scelta é

$$d_1(f, g) = \sum_{i=0}^{n} |f(x_i) - g(x_i)|. \tag{6.2}$$

In tal caso, se $\phi_i(x) = x^i$ e S è lo spazio dei polinomi di grado minore o uguale a n, si parla di problema della **interpolazione polinomiale**. Si dimostra che

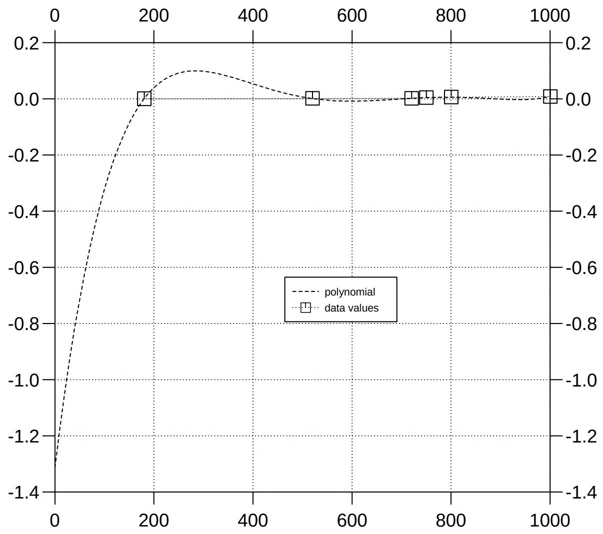


Figura 6.1: I polinomi interpolatori di grado elevato presentano forti oscillazioni.

il polinomio $\tilde{f}(x)$ che rende minima la distanza (6.2) è unico ed è il polinomio per cui $d_1(f, \tilde{f}) = 0$, ossia $\tilde{f}(x_i) = y_i$, $i = 1, \ldots, n$.

Vengono definite anche distanze in cui entrano le derivate della funzione, ad esempio

$$d_1^{(1)}(f, g) = \sum_{i=0}^{n} (|f(x_i) - g(x_i)| + |f'(x_i) - g'(x_i)|). \tag{6.3}$$

e in tal caso si parla di interpolazione di Hermite.

Se si usa la distanza d_1 ma al posto dei polinomi si usano funzioni razionali del tipo

$$R_{n,m}(x) = \frac{P_n(x)}{Q_m(x)} = \frac{a_0 + \ldots + a_n x^n}{b_0 + \ldots + b_m x^m} \tag{6.4}$$

abbiamo la cosiddetta **approssimazione di Padé**. In questo caso non è detto che vi sia sempre soluzione al problema, ossia che esista una funzione $R_{n,m}(x)$ tale che

$$R_{n,m}(x_i) = y_i, \quad i = 0, 1, \ldots, n + m \tag{6.5}$$

Se questo problema ha soluzione, la funzione che si ottiene é particolarmente efficace nel rappresentare funzioni che hanno delle singolaritá, che sono invece mal approssimate da polinomi.

Un altro caso importante è quello della **interpolazione polinomiale a tratti**: mostreremo che non è conveniente interpolare punti con polinomi di

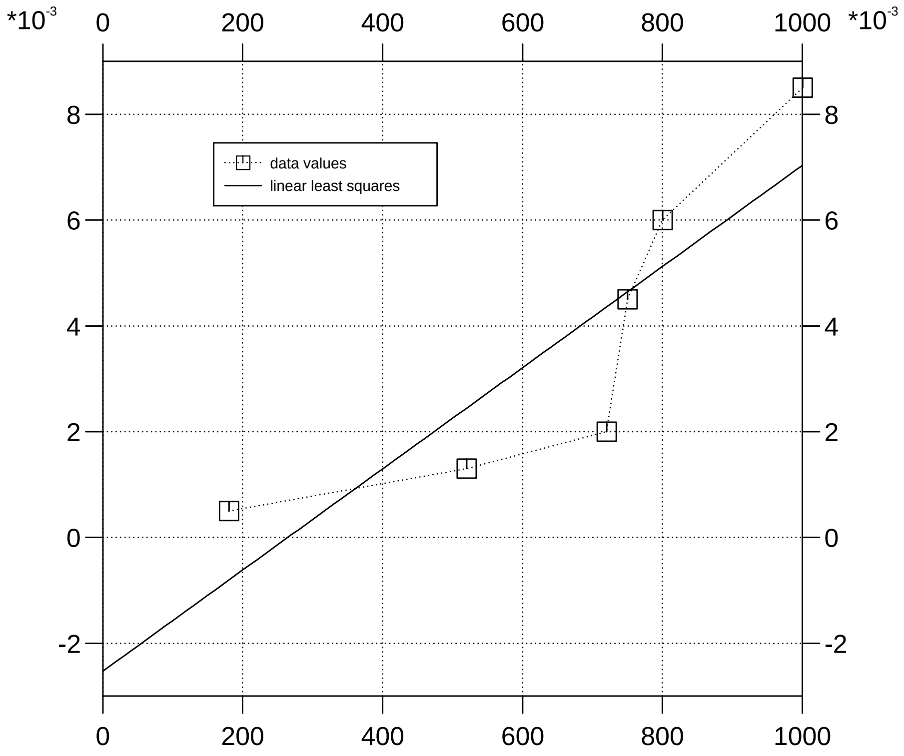


Figura 6.2: Esempio di approssimazione ai minimi quadrati.

grado elevato. Se i punti sono molti, conviene interpolarli (ossia calcolare la funzione che rende uguale a zero la distanza (6.2)) con polinomi *continui a tratti*, ossia funzioni che sono polinomi opportunamente "raccordati" per dare vita a una funzione definita in tutto l' intervallo I.

Se $d = d_2$, dove

$$d_2(f, g) = \sum_{i=0}^{n} (f(x_i) - g(x_i))^2, \tag{6.6}$$

il problema di calcolare la funzione g che minimizza d_2 in uno spazio di dimensione minore di n, viene detto **approssimazione discreta ai minimi quadrati**. Se S è uno spazio di polinomi, viene detta approssimazione **polinomiale** discreta ai minimi quadrati. Notare che se $dim(S) = n$, allora la soluzione è quella per la quale $d_2(f, g) = 0$, ossia il polinomio *interpolatore*!

Se $d = d_2$, ma

$$S = \{1, \cos(x), \sin(x), \cos(2x), \sin(2x), \ldots, \cos(kx), \sin(kx)\},$$

si parla di approssimazione **trigonometrica** discreta ai minimi quadrati.

Figura 6.3: figura 6.9 del testo del prof. G. Gambolati.

Se $d = d_2^{(c)}$, dove

$$d_2^{(c)} = \int_a^b (f(x) - g(x))^2 dx \tag{6.7}$$

si parla di approssimazione **continua** ai minimi quadrati.

Infine se $d = d_\infty$ dove

$$d_\infty(f, g) = \sup_{x \in [a,b]} |f(x) - g(x)|, \tag{6.8}$$

si parla di **approssimazione uniforme** o **approssimazione di Chebychev** in $[a, b]$. Se S è uno spazio di polinomi, si parla di approssimazione **polinomiale** uniforme e la soluzione è detta *polinomio di Chebychev.*

Il problema dell' approssimazione di Chebychev è in generale più difficile da risolvere di quello dell' approsimazione ai minimi quadrati, sia nel caso continuo che in quello discreto. In teoria l' approssimazione di Chebychev sarebbe preferibile perchè fornisce una stima *buona su tutto l' intervallo*, mentre la stima con la distanza $d_2^{(c)}$ può fornire una soluzione $\tilde{f}_2^{(c)}(x)$ che differisce molto da $f(x)$ in un ristretto sottointervallo di $[a, b]$.

6.2 L' algoritmo di Neville.

Siano $x_0, x_1, < \ldots, x_n$ ascisse date e sia $I = [a, b]$, $a = \min_{i=0,\ldots,n}\{x_i\}$, $b = \max_{i=0,\ldots,n}\{x_i\}$, il piú piccolo intervallo che contiene le ascisse date.

L' algoritmo di Neville serve per calcolare in un punto dato $\bar{x} \in I$ i valori $P_0(\bar{x}), \ldots, P_n(\bar{x})$ dei polinomi

Polinomio	grado	punti interpolati
$P_0(\bar{x})$	0	(x_0, y_0)
$P_1(\bar{x})$	1	$(x_0, y_0), (x_1, y_1)$
...	...	...
$P_n(\bar{x})$	n	$(x_0, y_0), \ldots, (x_n, y_n)$

La tecnica consiste nel costruire la tabella

$$
\begin{array}{lllll}
x_0 & y_0 = T_{00}(\bar{x}) & & & \\
x_1 & y_1 = T_{10}(\bar{x}) & T_{11}(\bar{x}) & & \\
x_2 & y_2 = T_{20}(\bar{x}) & T_{21}(\bar{x}) & T_{22} & \\
\ldots & \ldots & \ldots & \ldots & \\
x_n & y_n = T_{n0}(\bar{x}) & T_{n1}(\bar{x}) & \ldots & T_{nn}
\end{array}
$$

Gli elementi della tabella si calcolano tramite le relazioni

$$
\begin{aligned}
T_{i,0} &:= y_i, \quad 1 \leq k \leq i, \quad i = 0, 1, \ldots, n \\
T_{i,k} &:= \frac{1}{x_i - x_{i-k}} det \begin{pmatrix} T_{i,k-1} & \bar{x} - x_i \\ T_{i-1,k-1} & \bar{x} - x_{i-k} \end{pmatrix}.
\end{aligned}
$$

Si puó dimostrare che $P_i(\bar{x}) = T_{ii}(\bar{x})$.

L' algoritmo di Aitken (riportato nel testo di G. Gambolati) é una variante dell' algoritmo di Neville.

Capitolo 7

Integrazione numerica.

Esercizio 7.0.1 Calcolare le approssimazioni del valore di

$$I = \int_0^1 x^2 dx \tag{7.1}$$

che si ottengono usando gli schemi di integrazione

1. dei trapezi semplice,
2. dei trapezi composta, con $n = 2$ suddivisioni.
3. di Gauss-Legendre a tre punti.

Calcolare le stime dei valori assoluti degli errori commessi, chiamiamoli rispettivamente E_1, E_2 e E_3, usando le stime teoriche e confrontarli con gli errori effettivamente generati.

Soluzione.

n	pesi	ascisse
1	2	0
2	1	$\pm 0.57735\,02692$
3	5/9	$\pm 0.77459\,66692$
	8/9	0
4	0.34875 48451	$\pm 0.86113\,63116$
	0.65214 51549	$\pm 0.33998\,10436$
5	0.23692 68851	$\pm 0.90617\,98459$
	0.47862 86705	$\pm 0.53846\,93101$
	0.56888 88889	0

Tabella 7.1: Pesi ed ascisse per lo schema di integrazione di Gauss-Legendre.

Usando la regola dei trapezi semplice abbiamo:

$$I \simeq 1\frac{0+1}{2} = 1/2.$$

Usando la regola dei trapezi composta, con $n = 2$ risulta:

$$I \simeq \frac{1}{2}(\frac{0+1}{2} + \frac{1}{4}) = 3/8.$$

Poiché l' intervallo di integrazione non é $[-1, 1]$, non possiamo applicare direttamente la formula di Gauss-Legendre, tuttavia ponendo $x = \frac{1}{2}y + \frac{1}{2}$, abbiamo

$$I = \frac{1}{8}\int_{-1}^{1}(y+1)^2 dx.$$

Consultando una tabella con i pesi e le ascisse di Gauss-Legendre e usando lo schema a 3 punti otteniamo

$$I = \frac{1}{8}(0.5556(-0.7746+1)^2 + 0.8889(0+1) + 0.5556(0.7746+1)^2) \simeq 0.3333.$$

Per quanto riguarda gli errori, risulta

$$E_1 \leq \left|\frac{(b-a)^3}{12} f"(\xi)\right| = 1/6,$$

$$E_2 \leq \left|\frac{(b-a)^3}{122^2} f"(\xi)\right| = 1/24,$$

e infine

$$E_3 = 0$$

perché $f^{(5)}(x) = 0$.

I valori esatti degli errori sono uguali a quelli stimati.

7.1 Estrapolazione e integrazione di Gauss

Estrapolazione di
Romberg

$$T(h) = \frac{b-a}{n}\left(\frac{f_0+f_n}{2} + \sum_{i=1}^{n} f_i\right) \tag{7.2}$$

$$f \in \mathcal{C}^{m+2}[a, b]$$

$$T(h) = \int_a^b f(x)dx + \tau_1 h^2 + \tau_2 h^4 + \ldots + \tag{7.3}$$

$$\tau_m h^{2m} + \alpha_{m+1}(h) h^{2m+2} \tag{7.4}$$

- $|\alpha_{m+1}(h)| \leq M_{m+1}$ per ogni $h = (b-a)/n$.
- τ_k non dipende da h.

$$T(0) = \int_a^b f(x)dx \tag{7.5}$$

$$h_0 = b - a, \quad h_1 = h_0/n_1, \quad h_2 = h_0/n_2, \ldots \tag{7.6}$$

$$n_1 < n_2 < n_3 < \ldots$$

$$T_{i,0} = T(h_i), \quad i = 1, \ldots, m$$

$$\tilde{T}_{m,m}(h) = a_0 + a_1 h^2 + a_2 h^4 + \ldots + a_m h^{2m}$$

polinomio interpolatore

$$\tilde{T}_{m,m}(h_i) = T(h_i), \quad i = 0, \ldots, m$$

Estrapolare: $\tilde{T}_{m,m}(0) \simeq T(0)$.
Integrazione di Romberg: $h_i := \frac{b-a}{2^i}$.
Neville:

$$\begin{array}{llll} h_0 & T(h_0) = T_{0,0} & & \\ h_1 & T(h_1) = T_{1,0} & T_{1,1} & \\ h_2 & T(h_2) = T_{2,0} & T_{2,1} & T_{2,2} \\ & \ldots & & \end{array}$$

$$\begin{aligned} T_{i,0} &= T(h_i), \\ T_{i,k} &= T_{i,k-1} + \frac{T_{i,k-1} - T_{i-1,k-1}}{\left[\frac{h_{i-k}}{h_i}\right]^2 - 1} \end{aligned}$$

Può essere che estrapolazione = "vecchie formule". Esempio:
$h_0 = b - a$, $h_1 = (b-a)/2$.
$T_{0,0} = \frac{b-a}{2}(f(a) + f(b)) \leftarrow$ trapezi
$T_{1,1} = \frac{b-a}{2}\frac{1}{3}(f(a) + 4f((a+b)/2) + f(b)) \leftarrow$ Simpson

Integrazione di Gauss

$$I(f) := \int_a^b f(x)\omega(x)dx$$

- $\omega(x) \geq 0$ misurabile su $[a, b]$.
- Esistono tutti i momenti $\mu_k = \int_a^b x^k \omega(x) dx$, $k = 0, 1, \ldots$.
- $\int_a^b \omega(x) dx > 0$.

Calcolare

$$I(f) \simeq \tilde{I}(f) := \sum_{i=1}^{n} w_i f(x_i)$$

... in modo che l' uguaglianza sia esatta fino al maggior grado possibile.

$\mathcal{P}_j$:= { polinomi di grado $\leq$ j }

$\bar{\mathcal{P}}_j$:= { polinomi monoici di grado $\leq$ j }

Prodotto scalare

$$(f, g) = \int_a^b f(x) g(x) \omega(x) dx$$

$L^2[a, b]$ = insieme delle funzioni "a quadrato sommabile".

$(f, g) = 0$: f e g sono ortogonali.

Teorema. Per ogni $k \in \mathcal{N}$, esistono polinomi $p_j \in \mathcal{P}_k$, $j = 0, 1, \ldots, n$, tali che

$$(p_i, p_j) = \delta_{ij}$$

Questi polinomi sono:

$$p_0(x) = 1, \quad p_{-1}(x) = 0,$$

$$p_{i+1}(x) = (x - \delta_{i+1}) p_i(x) - \gamma_{i+1}^2 p_{i-1}(x), \quad i \geq 0$$

- $\delta_{i+1} := (x\, p_i, p_i)/(p_i, p_i)$, $i \geq 0$.
- $\gamma_{i+1}^2 := \begin{cases} 0 & i = 0, \\ (p_i, p_i)/(p_{i-1}, p_{i-1}) & i \geq 1 \end{cases}$

Dato che se $p \in \mathcal{P}_j$ allora $p = \sum_{i=0}^{j} c_i\, p_i$, risulta $(p, p_k) = 0$ per ogni $p \in \mathcal{P}_{k-1}$.

Teorema. Le radici x_i, $i = 1, \ldots, n$ di p_n sono

- reali e semplici,
- contenute nell' intervallo $[a, b]$.

Teorema. La matrice

$$A := \begin{pmatrix} p_0(t_1) & \dots & p_0(t_n) \\ p_{n-1}(t_1) & \dots & p_{n-1}(t_n) \end{pmatrix}$$

è non singolare se i punti t_i sono distinti.

Quindi il sistema lineare

$$A^T \begin{pmatrix} 0 \\ c_1 \\ \dots \\ c_{n-1} \end{pmatrix} = f_i$$

ha una e una sola soluzione, ossia esiste $p(x)$ t.c. $p(x) = \sum_{i=0}^{n-1} c_i f_i(x)$, $p(t_i) = f_i$.

A viene chiamata *Matrice di Haar* e $det(a) \neq 0$ viene chiamata *condizione di Haar.*

$p_0, \dots, p_n$ soddisfano la condizione di Haar: si dice che formano un *sistema di Chebychev.*

Teorema. Siano $p_0, \dots, p_n$ polinomi ortogonali. Siano $x_1, \dots, x_n$ le radici di p_n e sia $w = (w_1, \dots, w_n)^T$ la soluzione del sistema lineare

$$\begin{pmatrix} p_0(x_1) & \dots & p_0(x_n) \\ p_{n-1}(x_1) & \dots & p_{n-1}(x_n) \end{pmatrix} \begin{pmatrix} w_1 \\ \dots \\ w_n \end{pmatrix} \begin{pmatrix} (p_0, p_0) \\ 0 \\ 0 \end{pmatrix} \tag{7.7}$$

allora i valori w_i (detti *pesi*) sono maggiori di zero e

$$\begin{aligned} \textstyle\int_a^b f(x)\omega(x)dx \qquad &= \textstyle\sum_{i=1}^n w_i p(x_i) \\ &\text{per ogni } p \in \mathcal{P}_{2n-1}. \end{aligned} \tag{7.8}$$

Viceversa: se w_i, x_i soddisfano (7.8) allora $p_n(x_i) = 0$ e $w = (w_1, \dots, w_n)$ soddisfa (7.7).

NON esistono $2n$ numeri w_i, x_i tali che (7.8) valga per ogni $p \in \mathcal{P}_{2n}$.

Esempio 7.1.1 $\omega(x) = 1$, intervallo $= [-1, 1]$:
polinomi di Legendre
formula ricorrente:

$$p_{-1} = 0, \quad p_0 = 1$$

$$p_{n+1}(x) = \frac{2n+1}{n+1}\, x\, p_n(x) - \frac{n}{n+1}\, p_{n-1}(x)$$

definizione alternativa:

$$p_k(x) = \frac{k!}{(2k)!} \frac{d^k}{dx^k} (x^2-1)^k$$

Come si calcolano i w_i e gli x_i?

$$J = \begin{pmatrix} \delta_1 & \gamma_1 & 0 & & & \\ \gamma_1 & \delta_2 & \gamma_2 & 0 & & \\ & & \dots & & & \\ 0 & & & \gamma_{n-2} & \delta_{n-1} & \gamma_{n-1} \\ & 0 & & 0 & \gamma_n & \delta_n \end{pmatrix} \tag{7.9}$$

$$det(J - xI) = p_n(x)$$

$$p_n(x) = 0 \text{ sse } x \in \lambda(J)$$

I w_i si possono calcolare a partire da $V(J)$.

Teorema. Sia $f \in \mathcal{C}^{2n}[a, b]$,
esiste $\xi \in [a, b]$ tale che

$$\int_a^b f(x)\omega(x)dx - \sum_{i=1}^{n} w_i f(x_i) = \frac{f^{(2n)}(\xi)}{(2n)!}(p_n, p_n)$$

Capitolo 8

Equazioni differenziali ordinarie.

8.1 Cenni sulla risolubilitá delle ODE.

Si chiama *problema di Cauchy* il problema di trovare una funzione y continua e derivabile nell' intervallo $I_0 = [x_0, x_1]$ tale che

$$\begin{aligned} y'(x) &= f(x, y(x)), \quad x \in I_0 && (8.1) \\ y(x_0) &= y_0 && (8.2) \end{aligned}$$

La condizione (8.2) si chiama **condizione di Cauchy** e anche **condizione iniziale**.

Una funzione y che verifica la (8.1) viene chiamata un **integrale** dell' equazione differenziale.

La equazione (8.1) viene detta **del primo ordine** perché in essa interviene solo la derivata prima di y.

Vale il

Teorema (Cauchy-Lipschitz): se la funzione f é continua su $I_0 \times \Re$ ed esiste una costante reale positiva L tale che

$$|f(x, y) - f(x, z)| \leq L|y - z|, \tag{8.3}$$

per tutte le coppie (x, y), $(x, z) \in I_0 \times \Re$, allora il problema (8.1)+(8.2) ammette una ed una sola soluzione.

Per poter risolvere numericamente un problema differenziale, non basta che la soluzione esista e sia unica, bisogna anche che essa sia ben condizionata. Per studiare il condizionamento del problema, dobbiamo studiare il comportamente della soluzione quando i dati cambiano. A questo proposito, sia $\psi(x)$ una funzione continua in I_0, β ed ϵ numeri reali. Come problema *perturbato* consideriamo il seguente:

trovare una funzione $y_\epsilon \in C^1(I_0)$ tale che

$$y'_\epsilon(x) = f(x, y_\epsilon(x)) + \epsilon\psi(x), \quad x \in I_0 \tag{8.4}$$
$$y_\epsilon(x_0) = y_0 + \epsilon\beta \tag{8.5}$$

Vale la

Proposizione: se valgono le condizioni del teorema di Cauchy-Lipschitz, allora il problema (8.4)+(8.5) ammette una ed una sola soluzione e per tutti gli $x \in I_0$ risulta

$$|y(x) - y_\epsilon(x)| \leq |\epsilon\beta| e^{L|x-x_0|} + \int_{x_0}^{x} |\epsilon\psi(s)| ds \tag{8.6}$$

La relazione (8.6) implica la *dipendenza continua* della soluzione dai dati, una caratteristica molto importante dal punto di vista della risolubilitá numerica.

Si dice che un problema é **ben posto** se la sua soluzione *esiste*, *é unica* e *dipende con continuitá dai dati*. Un problema differenziale é numericamente risolubile se é non solo ben posto, ma anche *ben condizionato*. La relazione (8.6) mostra che il problema in esame é ben condizionato quando la quantitá

$$\max_{x \in I_0} e^{L|x-x_0|} \tag{8.7}$$

non é "troppo grande".

Consideriamo infine il problema della *regolaritá*, ossia del grado di derivabilitá, della soluzione in dipendenza dalla regolaritá dei dati. L' ordine di convergenza dei metodi numerici dipende infatti dalla regolaritá della soluzione. Vale il

Teorema: sia $y \in C^1(I_0)$ una soluzione del problema (8.1)+(8.2). Se f é derivabile p volte nel punto $(x, y(x))$, allora nello stesso punto la soluzione y é derivabile $(p+1)$ volte e si ha

$$y^{(p+1)}(x) = f^{(p)}(x, y(x)).$$

Esercizio 8.1.1 Si vuole integrare numericamente il problema differenziale

$$y' + y = 0, \quad x \in [0, 1] \tag{8.8}$$
$$y(0) = y_0 = 1 \tag{8.9}$$

Calcolare le soluzioni approssimate $\bar{y}^{(1)}$ e $\bar{y}^{(2)}$ che si ottengono usando rispettivamente gli schemi alle differenze:

$$(1 + h)\bar{y}_{n+1} - \bar{y}_n = 0 \tag{8.10}$$

$$\bar{y}_{n+2} - 4\bar{y}_{n+1} + (3 - 2h)\bar{y}_n = 0 \tag{8.11}$$

con passo $h = 1/2$.

Calcolare le norme 1 dei vettori degli errori relativi che si commettono utilizzando i due schemi.

Soluzione.

Tramite le relazioni ricorrenti, é immediato calcolare le approssimazioni richieste. L' unico problema é il valore di $\bar{y}_1^{(2)}$. Poniamo $\bar{y}_1^{(2)} := \bar{y}_1^{(1)}$. Otteniamo cosí:

x	exp(-x)	y1	y2
0	1.0000	1.0000	1.0000
0.5000	0.6065	0.6667	0.6667
1.0000	0.3679	0.4444	0.6667

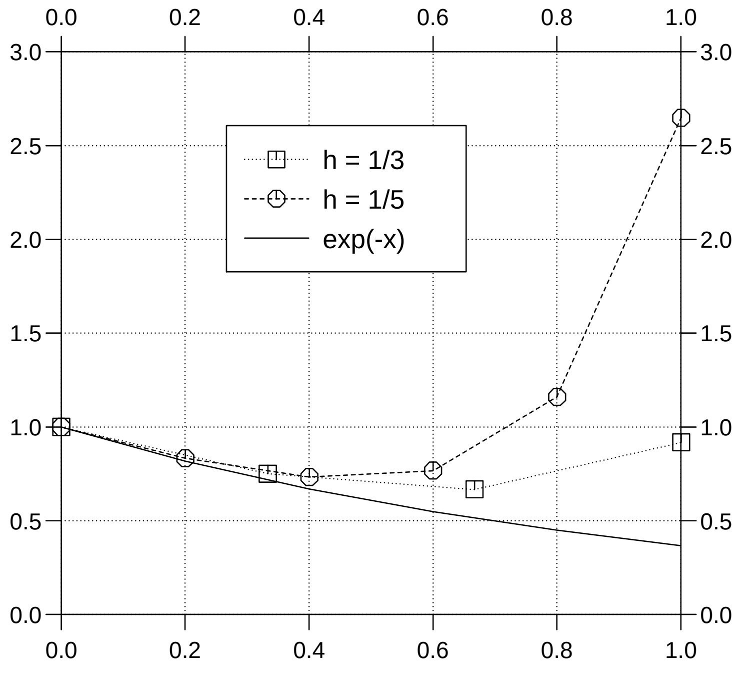
dove `y1` $:= \bar{y}^{(1)}$, `y2` $:= \bar{y}^{(2)}$.

La soluzione esatta del problema é $y(x) = exp(-x)$. Le norme 1 dei vettori degli errori relativi

$$\left\| e^{(j)} \right\|_1 = \sum_{i=0}^{2} \left| \frac{y(x_i) - \bar{y}_i^{(j)}}{y(x_i)} \right|$$

sono quindi: $\left\| e^{(1)} \right\|_1 \simeq 0.3072$, $\left\| e^{(2)} \right\|_1 \simeq 0.9114$.

Schema instabile di ordine 2.



Come si vede, l' errore relativo generato con lo schema del secondo ordine é piú elevato, per il fatto che la soluzione ha una componente spuria che cresce rapidamente. E' interessante vedere che cosa accade quando si usa tale schema e si diminuisce il passo di integrazione. In figura si puó vedere il comportamento della soluzione esatta e di quelle approssimate ottenute con passi $h = 1/3$ e $h = 1/5$: le soluzioni dell' equazione alle differenze crescono molto rapidamente e non convergono alla soluzione esatta.

Esercizio 8.1.2 Approssimare la soluzione del problema differenziale

$$\begin{aligned} y' + y &= x^2, \quad x \in [0,1] & (8.12)\\ y(0) &= y_0 = 3 & (8.13)\end{aligned}$$

utilizzando lo schema di Eulero.

Analizzare la stabilitá e la convergenza dello schema alle differenze che si ottiene.

Soluzione.

Lo schema di Eulero é

$$\bar{y}_{n+1} = \bar{y}_n + hf(x_n, y_n).$$

In questo caso abbiamo il problema:

$$y' = f(x,y) = -y + x^2,$$

l' equazione alle differenze risultante é

$$\bar{y}_{n+1} = \bar{y}_n + h(-\bar{y}_n + x_n^2),$$

ossia, dato che $x_n = x_0 + nh = nh$:

$$\bar{y}_{n+1} - (1-h)\bar{y}_n = n^2h^3.$$

La soluzione generale di questa equazione alle differenze é

$$\bar{y}_n^* = c(1-h)^n + (hn)^2 - 2(hn) + 2 - h,$$

che, tenendo conto delle condizioni iniziali, si particolarizza in

$$\bar{y}_n = (1+h)(1-h)^n + (hn)^2 - 2(hn) + 2 - h.$$

La soluzione del problema differenziale é invece

$$y = e^{-x} + x^2 - 2x + 2.$$

E' facile vedere che per $h \to 0$ e $n \to \infty$ in modo che $nh = x=$ cost., si ha

$$\lim_{h\to 0}(1+h)(1-h)^n = e^{-x}, \quad \lim_{h\to 0}\left[(hn)^2 - 2(hn) + 2 - h\right] = x^2 - 2x + 2,$$

ossia lo schema é convergente:

$$\lim_{h \to 0} \bar{y}_n = y(x).$$

Per quanto riguarda la stabilitá, se $\bar{y}_k$ é affetto da un errore ϵ_k, abbiamo:

$$\bar{y}_{n+1} + \epsilon_{n+1} - (1-h)(\bar{y}_n + \epsilon_n) = n^2 h^3,$$

da cui l' errore soddisfa alla relazione

$$\epsilon_{n+1} - (1-h)\epsilon_n = 0.$$

Quindi

$$\epsilon_n = \epsilon_0 (1-h)^n,$$

e lo schema é stabile a condizione che $|1-h| \leq 1$, cioé $0 \leq h \leq 2$.

8.2 Runge-Kutta per sistemi di ODE.

Ecco un esempio di applicazione di uno schema del tipo Runge-Kutta (lo schema di Heun) per risolvere un sistema di equazioni differenziali.

Risolvere il problema differenziale:

$$y" + 2y' + y = x + 1, \quad x \in [0,1] \tag{8.14}$$
$$y(0) = 0, \quad y'(0) = 0. \tag{8.15}$$

La soluzione di questo problema è $y = e^{-x} + x - 1$.

L' equazione (8.14) è equivalente al sistema di equazioni

$$\begin{cases} y' &= z = f_1(x,y,z) \\ z' &= -2z - y + x + 1 = f_2(x,y,z) \end{cases} \tag{8.16}$$

Lo schema di Heun si scrive:

$$\begin{aligned} \bar{y}_{n+1} &= \bar{y}_n + \frac{1}{2}k_1 + \frac{1}{2}k_2 && (8.17) \\ k_1 &= hf(x_n, \bar{y}_n) && (8.18) \\ k_2 &= hf(x_n + h, \bar{y}_n + k_1). && (8.19) \end{aligned}$$

La sua ovvia generalizzazione ad un sistema di due equazioni in due incognite è

$$\begin{aligned} \bar{y}_{n+1} &= \bar{y}_n + \frac{1}{2}k_{y,1} + \frac{1}{2}k_{y,2} && (8.20) \\ \bar{z}_{n+1} &= \bar{z}_n + \frac{1}{2}k_{z,1} + \frac{1}{2}k_{z,2} && (8.21) \\ k_{y,1} &= hf_1(x_n, \bar{y}_n, \bar{z}_n) && (8.22) \\ k_{z,1} &= hf_2(x_n, \bar{y}_n, \bar{z}_n) && (8.23) \\ k_{y,2} &= hf_1(x_n + h, \bar{y}_n + k_{y,1}, \bar{z}_n + k_{z,1}) && (8.24) \\ k_{z,2} &= hf_2(x_n + h, \bar{y}_n + k_{y,1}, \bar{z}_n + k_{z,1}) && (8.25) \end{aligned}$$

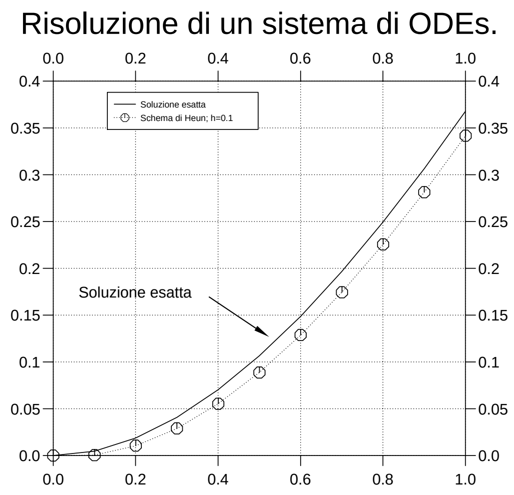


Figura 8.1: Sistema di ODE risolto con lo schema di Heun.

dove i termini $k_{y,*}, k_{z,*}$ indicano *incrementi riguardanti* y *oppure* z.

Applicando questa generalizzazione al sistema (8.16) otteniamo le relazioni

$$\bar{y}_{n+1} = \bar{y}_n + \frac{1}{2}k_{y,1} + \frac{1}{2}k_{y,2} \tag{8.26}$$

$$\bar{z}_{n+1} = \bar{z}_n + \frac{1}{2}k_{z,1} + \frac{1}{2}k_{z,2} \tag{8.27}$$

$$k_{y,1} = h\bar{z}_n, \tag{8.28}$$

$$k_{z,1} = h(-2\bar{z}_n - \bar{y}_n + x_n + 1) \tag{8.29}$$

$$k_{y,2} = h(\bar{z}_n + k_{z,1}) \tag{8.30}$$

$$k_{z,2} = h(-2(\bar{z}_n + k_{z,1}) - (\bar{y}_n + k_{y,1}) + (x_n + h) + 1) \tag{8.31}$$

La figura (8.1) mostra l' andamento della componente $\bar{y}_n$ della soluzione di questo sistema di equazioni alle differenze, usando un passo $h = 0.1$ e imponendo le condizioni iniziali (8.15).

8.3 Risoluzione di un BVP con shooting.

E' dato il problema con valori al contorno (BVP)

$$y" + 2y' = 4x + 3e^x, x \in [0,1] \tag{8.32}$$

$$y(0) = 1, \quad y(1) = e. \tag{8.33}$$

Vogliamo calcolare un' approssimazione numerica della sua soluzione, usando un metodo di *shooting*. Esso consiste nel risolvere i problemi

$$y" + 2y' = 4x + 3e^x, x \in [0,1] \tag{8.34}$$

$$y(0) = 1, \quad y'(0) = s \tag{8.35}$$

per diversi valori del parametro s: ad ogni valore corrisponderà una soluzione $y^*(x;s)$ che in generale *non* soddisfa alla condizione al contorno in $x=1$, ossia risulta $y^*(1;s) \neq y(1)$. Risolvere il problema con le condizioni al contorno (8.33) equivale a calcolare la soluzione s dell' equazione

$$\Phi(s) = y^*(1;s) - y(1) = 0. \tag{8.36}$$

Si può determinarla con qualsiasi schema per calcolare le soluzioni di un' equazione non lineare. Useremo lo schema della regula falsi

$$s_{k+1} = s_k - \Phi(s_k)/\tfrac{\Phi(s_k)-\Phi(s_{k-1})}{s_k-s_{k-1}} \tag{8.37}$$

$$k = 1,2,\ldots, \quad s_0, s_1 \quad \text{valori dati} \ . \tag{8.38}$$

Per integrare numericamente l' equazione (8.32) usiamo lo schema alle differenze che si ottiene usando le approssimazioni

$$y'_n = \frac{\Delta y_n}{h} + O(h), \tag{8.39}$$

$$y"_n = \frac{\Delta^2 y_n}{h^2} + O(h), \tag{8.40}$$

e scrivendo l' equazione (8.32) nel punto $x_n = nh$, ossia lo schema

$$\bar{y}_{n+2} + 2(h-1)\bar{y}_{n+1} + (1-2h)\bar{y}_n = 4h^2 x_n + 3h^2 e^{x_n}. \tag{8.41}$$

Stimiamo ora i due valori iniziali di s. Poniamo ad esempio

$$s_0 = \frac{\bar{y}_1^{+\delta} - y(0)}{h}, \quad s_1 = \frac{\bar{y}_1^{-\delta} - y(0)}{h}, \tag{8.42}$$

$h = 1/2$, dove i valori $\bar{y}_1^{+\delta}$, $\bar{y}_1^{-\delta}$ sono stati ricavati dalla relazione (8.41) ponendo $h = 1/2$, $\bar{y}_0 = y(0) = 1$, $\bar{y}_2^{\pm\delta} = y(1) \pm \delta$, essendo δ un valore arbitrario, diciamo $\delta = 1$.

Utilizzando questi due valori iniziali e applicando lo schema (8.37) otteniamo i risultati:

x	y	y0	y1	y2	yb
0	1.0000	1.0000	1.0000	1.0000	1.0000
0.1000	1.0152	0.4037	0.8037	1.0458	1.0000
0.2000	1.0614	-0.0434	0.6766	1.1125	1.0300
0.3000	1.1399	-0.3639	0.6121	1.2030	1.0912
0.4000	1.2518	-0.5757	0.6051	1.3201	1.1847
0.5000	1.3987	-0.6926	0.6520	1.4662	1.3121
0.6000	1.5821	-0.7254	0.7503	1.6438	1.4747
0.7000	1.8038	-0.6822	0.8984	1.8554	1.6743
0.8000	2.0655	-0.5689	1.0956	2.1034	1.9126
0.9000	2.3696	-0.3899	1.3417	2.3901	2.1917
1.0000	2.7183	-0.1479	1.6374	2.7183	2.5137

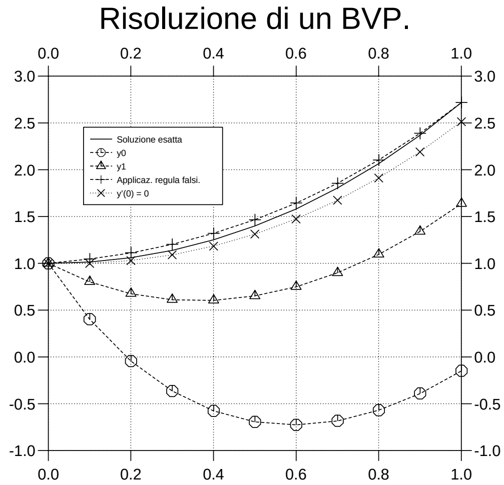


Figura 8.2: Applicazione del metodo di shooting con uso della regula falsi.

dove `y` è la soluzione esatta $y = x^2 - x + e^x$ del problema dato, `yi` $= \bar{y}^{(i)} = \{\bar{y}_j(s_i) = \bar{y}(x_j; s_i), \quad j = 1, \ldots n = 1/h\}$ sono le soluzioni dello schema (8.41), ottenute ponendo di volta in volta

$$\bar{y}_1(s_i) = y_0 + hs_i, \tag{8.43}$$

dove s_0 e s_1 sono dati dalla (8.42) e s_2 quello ottenuto applicando la regula falsi (8.37). Notare come basta una sola iterazione con lo schema della regula falsi per ottenere una buona approssimazione della soluzione cercata. Se si calcola anche $\bar{y}^{(3)}$ si vede che $\bar{y}^{(2)} = \bar{y}^{(3)}$: non si guadagna nulla procedendo oltre la prima iterazione.

`yb` $= \bar{y}(x_n; 0)$ è la soluzione numerica che si ottiene imponendo la condizione iniziale esatta $y'(0) = 0$. E' interessante notare come $\bar{y}^{(2)}$ sia piú accurata di $\bar{y}(x_n; 0)$: conoscere esattamente la condizione iniziale sulla derivata è meno vantaggioso che conoscere il valore della funzione nell' estremo finale dell' intervallo.

La figura (8.2) fornisce uno schizzo delle curve che si ottengono plottando i valori trovati.

8.4 BVP e sistemi alle DF

E' dato il problema con valori al contorno (BVP)

$$y" + 2y' = 4x + 3e^x, x \in [0, 1] \tag{8.44}$$

$$y(0) = 1, \quad y(1) = e, \tag{8.45}$$

che ha la soluzione

$$y = x^2 - x + e^x. \tag{8.46}$$

Si può determinare una soluzione numerica di questo problema usando uno schema alle differenze finite. Usiamo lo schema alle differenze

$$\bar{y}_{n+2} + 2(h-1)\bar{y}_{n+1} + (1-2h)\bar{y}_k = 4h^2 x_n + 3h^2 e^{x_n} \tag{8.47}$$

$x_n = nh$, che si ottiene dalla equazione (8.44) impiegando note approssimazioni dell' operatore differenziale. Posto $h = 0.25$, abbiamo $\bar{y}_0 = y(0) = 1$, $\bar{y}_4 = y(1) = e$, e scrivendo le relazioni (8.47) per $n = 0, 1, 2$, otteniamo un sistema lineare di 3 equazioni in tre incognite

$$\mathbf{As} = \mathbf{b} \tag{8.48}$$

dove

```
    [ 2 * (h - 1)   1              0          ]
A = [ (1 - 2 * h)   2 * (h - 1)    1          ]
    [ 0             (1 - 2 * h)    2 * (h - 1)]

     [ -1.5000    1.0000         0 ]
A =  [  0.5000   -1.5000    1.0000 ]
     [       0    0.5000   -1.5000 ]

    [ (3 * h*h - (1 - 2 * h) )                        ]
b = [ (4 * h*h *   h + 3 * h*h * exp(  h)  )          ]
    [ (4 * h*h * 2*h + 3 * h*h * exp(2*h) - exp(1) ) ]

    [ -0.3125 ]
  = [  0.3033 ]
    [ -2.2841 ]
```

Risolvendo questo sistema lineare otteniamo il risultato

```
        x         y        yn
        0    1.0000    1.0000
   0.2500    1.0965    1.2673
   0.5000    1.3987    1.5884
   0.7500    1.9295    2.0522
   1.0000    2.7183    2.7183
```

8.5 Strato limite

8.5.1 Formulazione del problema

Si vuole risolvere numericamente il problema [11]

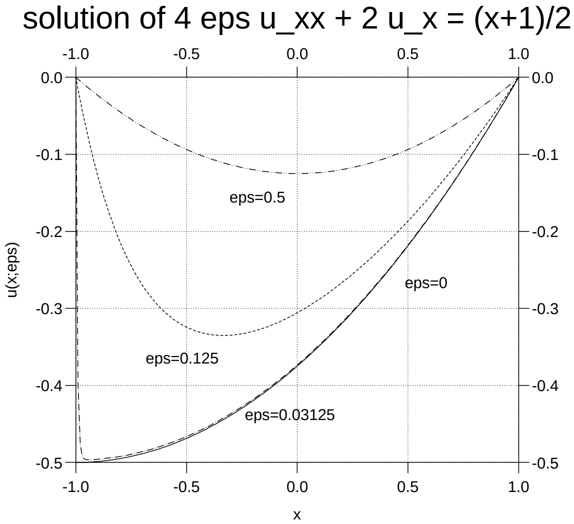


Figura 8.3: Soluzioni del problema per alcuni valori di ϵ.

$$\begin{cases} 4\epsilon u_{xx} + 2u_x &= (x+1)/2, \quad x \in (-1,1), \quad \epsilon > 0, \\ u(-1) = u(1) &= 0 \end{cases} \tag{8.49}$$

La sua soluzione analitica è

$$\begin{aligned} u \quad &= \quad -\frac{(4\,\epsilon - 1)\; e^{\frac{1}{2\,\epsilon} - \frac{x}{2\,\epsilon}}}{2\,e^{\frac{1}{\epsilon}} - 2} + \\ &+ \quad \frac{x^2 + (1 - 4\,\epsilon)\; x + 8\,\epsilon^2 - 2\,\epsilon}{4} - \\ &+ \quad \frac{\left(4\,\epsilon^2 - 3\,\epsilon + 1\right)\; e^{\frac{1}{\epsilon}} - 4\,\epsilon^2 - \epsilon}{2\,e^{\frac{1}{\epsilon}} - 2} \end{aligned}$$

Quando $\epsilon \to 0$, la soluzione di questa equazione ha uno *strato limite* o *boundary layer* in $x = -1$. Il significato di questa affermazione si puó capire guardando le figure 8.3 e 8.4.

8.5.2 Risoluzione numerica

Definiamo una griglia in [0,1] ponendo

$$x_i = -1 + ih, i = 0, 1, \ldots, N$$

N intero positivo dato, $h = 2/N$.

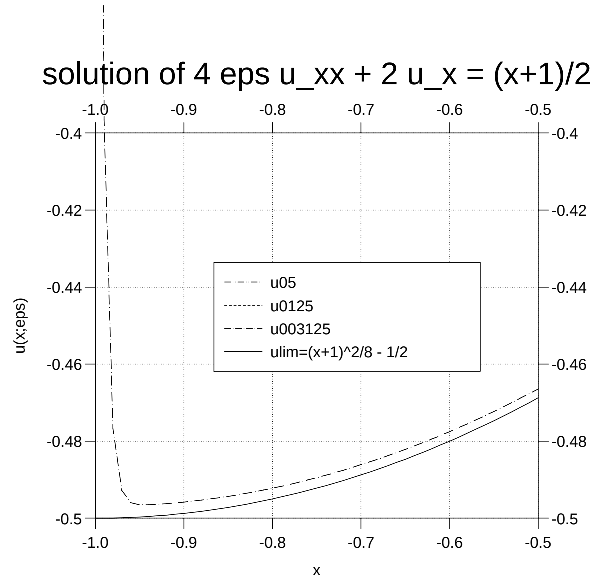


Figura 8.4: Dettaglio della figura precedente.

Lo schema alle differenze finite (FD) del *secondo ordine centrato* (*centered FD*) é definito come

$$4\epsilon\frac{u_{j+1}-2u_j+u_{j-1}}{h^2}+2\frac{u_{j+1}-u_{j-1}}{2h}=\frac{x_j+1}{2}, \quad j=1,2,\ldots,N-1. \tag{8.50}$$

Lo schema del *primo ordine upwind* (*Upwind FD*) come

$$4\epsilon\frac{u_{j+1}-2u_j+u_{j-1}}{h^2}+2\frac{u_{j+1}-u_j}{h}=\frac{x_j+1}{2}, \quad j=1,2,\ldots,N-1. \tag{8.51}$$

Quando $\epsilon = O(1)$, l' errore di troncamento di (8.50) è $O(h^2)$, mentre quello di (8.51) è $O(h)$.

Sia $u(x)$ la soluzione esatta e $\tilde{u}(x)$ quella approssimata. Poniamo

$$e(x)=u(x)-\tilde{u}(x), \quad E=\max_{0\le j\le N}|e(x_j)|. \tag{8.52}$$

La figura 8.5 mostra $\log_{10} E$ in funzione di $-\log_{10}\epsilon$.

La figura 8.6 mostra $\log_{10}|e|$ per alcuni valori di ϵ.

8.5.3 Conclusioni

Notiamo le seguenti caratteristiche dei due metodi.

Differenze centrali:

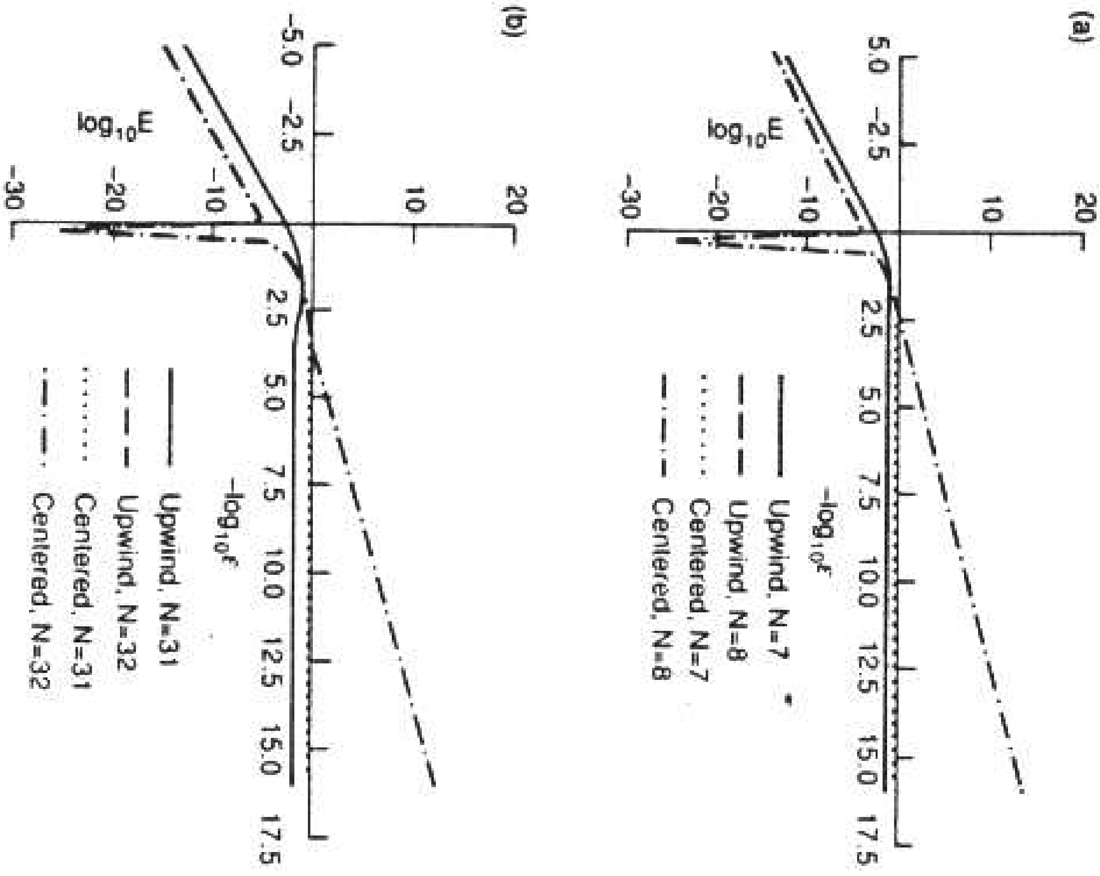


Figura 8.5: $\log_{10} E$ in funzione di $-\log_{10} \epsilon$.

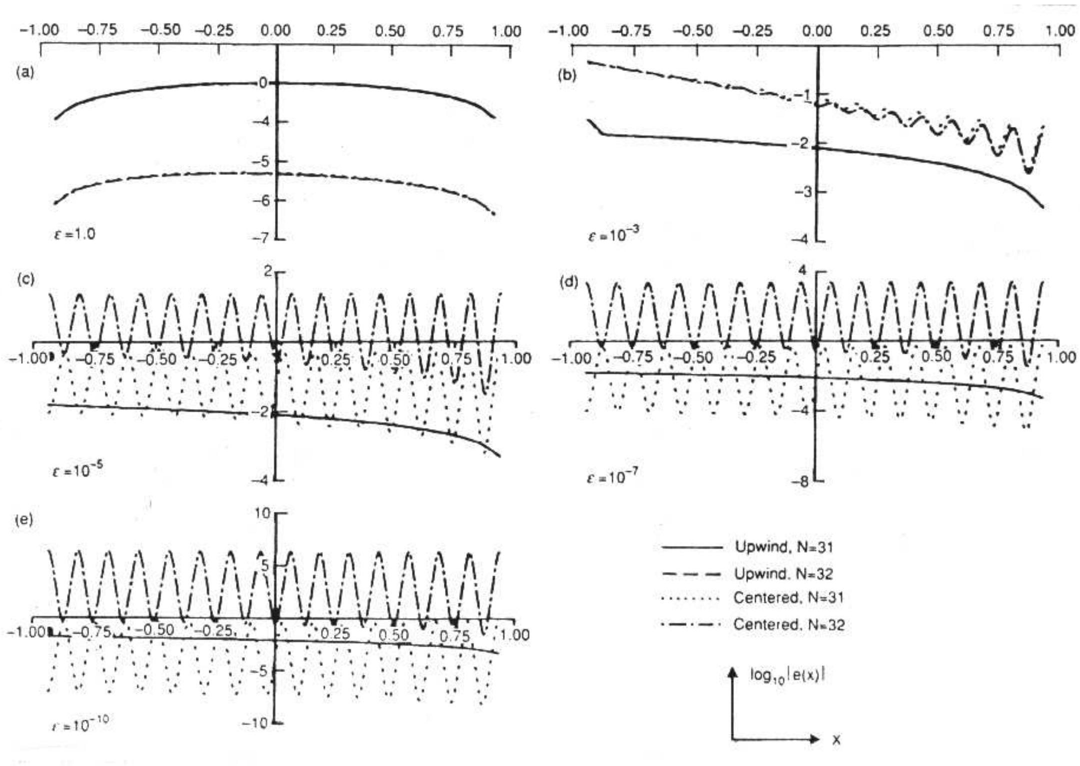


Figura 8.6: $\log_{10}|e|$ in funzione di ϵ.

- la soluzione calcolata oscilla quando $\epsilon \to 0$;
- errore e soluzione sono limitati per N dispari e illimitati per N pari;
- $E = O(h^2)$ quando $\epsilon = O(1)$;

Differenze Upwind:

- la soluzione calcolata *non* oscilla quando $\epsilon \to 0$;
- i comportamenti dell' errore e della soluzione non dipendono dal fatto che N sia pari o dispari;
- $E = O(h)$ quando $\epsilon = O(1)$;

Appendice A

Progetto di un contenitore

A.1 Il problema

La legge di stato di un gas perfetto è data dalla eq.:

$$pV = nRT \tag{A.1}$$

dove p è la pressione, V è il volume, n è il numero di grammi-molecole, T è la temperatura assoluta ed R è la costante universale dei gas.
L'eq. (A.1) è valida per pressioni e temperature comprese in un intervallo limitato ed inoltre alcuni gas la soddisfano meglio di altri.
Per i gas reali l'eq. (A.1) è stata corretta da van der Waals nella relazione:

$$\left(p + \frac{n^2 a}{V^2}\right)\left(\frac{V}{n} - b\right) = RT \tag{A.2}$$

dove le costanti a e b sono determinate empiricamente per ogni specifico gas.
Volendo progettare dei contenitori per l'anidride carbonica(CO_2) e l'ossigeno (O_2) per diverse combinazioni dei valori di p e T, è necessario calcolare accuratamente con la (A.2) il volume

$$v = V/n \tag{A.3}$$

occupato da una grammo-molecola.
Sono assegnati i seguenti dati:

$$\mathrm{R} = 0.082054\ \mathrm{l}\cdot\ \mathrm{atm}/(\mathrm{K}\cdot\ \mathrm{mol})$$

$$\left\{\begin{array}{ll} a & = 3.592\ \mathrm{atm\ l^2/mol^2} \\ & \\ b & = 0.04267\ \mathrm{l/mol} \end{array}\right. \quad \text{per } \mathrm{CO_2}$$

$$\left\{\begin{array}{ll} a & = 1.360\ \mathrm{atm\ l^2/mol^2} \\ & \\ b & = 0.03183\ \mathrm{l/mol} \end{array}\right. \quad \text{per } \mathrm{O_2}$$

Le pressioni e le temperature di progetto sono:

p(atm)	1	1	1	10	10	10	100	100	100
T(K)	300	500	700	300	500	700	300	500	700

A.2 Compiti

Scrivere un programma in *FORTRAN* che, per una assegnata tabella di valori di p e T e per ogni specifico gas, calcola con la (A.1) il valore v_0 iniziale e trova il valore finale v_{fin} risolvendo la (A.2) con lo schema di Newton-Raphson. Stampare per i due gas assegnati e per ogni valore di p e T di progetto, v_0, v_{fin} ed il numero di iterazioni richieste per ottenere v_{fin}.

Appendice B

Modelli di crescita

B.1 Introduzione

I modelli di crescita di popolazioni assumono che la densità di una popolazione p sia proporzionale alla popolazione esistente, ossia

$$\frac{dp}{dt} = kp, \tag{B.1}$$

dove k non dipende dalla concentrazione di nutrimento disponibile. Anche quando il nutrimento non scarseggia, la crescita in ambiente chiuso viene limitata dalle sostanze di rifiuto che gli esseri viventi producono. Questi prodotti inibiscono la crescita quando la densità raggiunge un valore massimo p_m. In questa situazione, la (B.1) si modifica nella

$$\frac{dp}{dt} = Kp(p_m - p). \tag{B.2}$$

Prendiamo come esempio una popolazione di cellule in un liquido di coltura. Se la dimensione di p è $[p]$ = cellule/litro, e il tempo è misurato in giorni, K ha dimensione $[K] = (litri/cellule)/giorno$. Posto $p(0) = p_0$, la soluzione di (B.2) è:

$$p(t) = \frac{p_m}{1 + (p_m/p_0 - 1)\exp(-Kp_m t)}. \tag{B.3}$$

Questa funzione viene chiamata *modello di crescita logistica*, o semplicemente *logistica*. Il suo andamento è esemplificato in figura B.1.

Come esempio di applicazione del modello logistico, consideriamo il problema di stimare l'evoluzione del mercato di un certo tipo di elaboratori. All'istante $t = 0$ sono stati venduti pochi elaboratori, diciamo $p_0 = 10$. È noto che dopo $t = 60$ settimane, ne sono stati venduti 15.000 e che K vale 2×10^{-6}. Si vuole calcolare il numero di elaboratori venduto, p_T, dopo $T = 90$ settimane. Poniamo $[p]$ = numero elaboratori, misuriamo il tempo in settimane, allora $[K]$ = 1/(elaboratori · settimane). Sostituendo le informazioni disponibili per $t = 60$ nella eq. (B.3), si ottiene una equazione non lineare, che va risolta nella variabile p_m. Sostituendo p_m nella (B.3), si può infine calcolare $p_T = p(T)$.

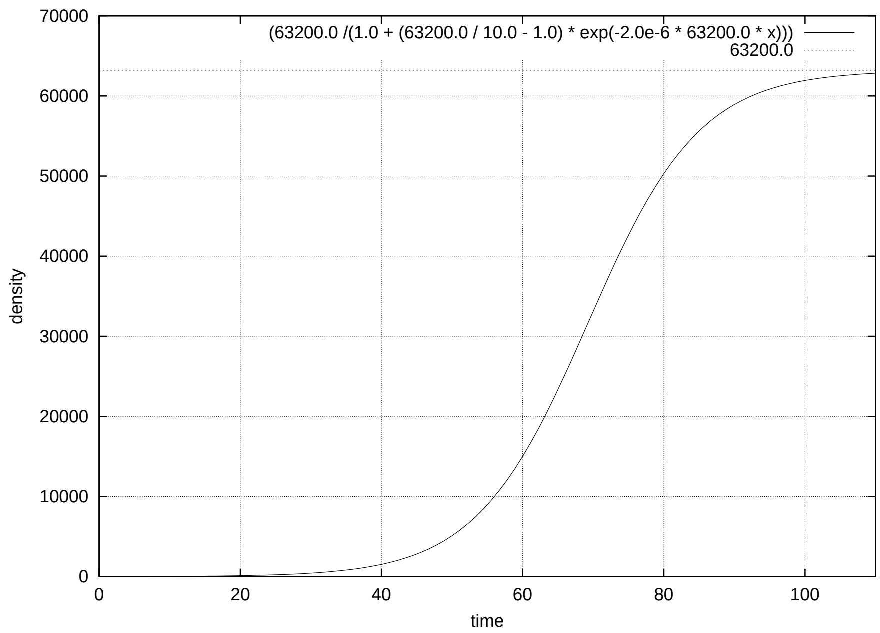


Figura B.1: Curva logistica.

B.2 Compiti

Calcolare la quantità p_m,

- usando lo schema di Newton-Rapson,
- usando lo schema della secante variabile,
- usando lo schema della secante fissa.

Sia

$$s_k = |(p_{k+1} - p_k)/p_{k+1}|$$

lo scarto relativo alla k-esima iterazione. Arrestare le iterazioni quando $s_k < \tau$, $\tau = 10^{-2}, 10^{-4}, 10^{-6}$. Calcolare i corrispondenti valori di $p(T)$.

Tabulare il numero di iterazioni effettuate, le approssimazioni intermedie e gli scarti relativi.

Tabulare e graficare le soluzioni numeriche ottenute.

Appendice C

Flusso di calore

C.1 Il problema

In molti campi dell'ingegneria si pone il problema di determinare la temperatura in corpi solidi. In problemi stazionari e per solidi omogenei ed isotropi la temperatura T è governata dall'equazione di Laplace:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0$$

Le derivate seconde possono approssimarsi con schemi alle differenze finite (si veda il numero 9.10 del testo "Metodi Numerici" di G. Gambolati). Su una griglia regolare come quella della figura C.1.

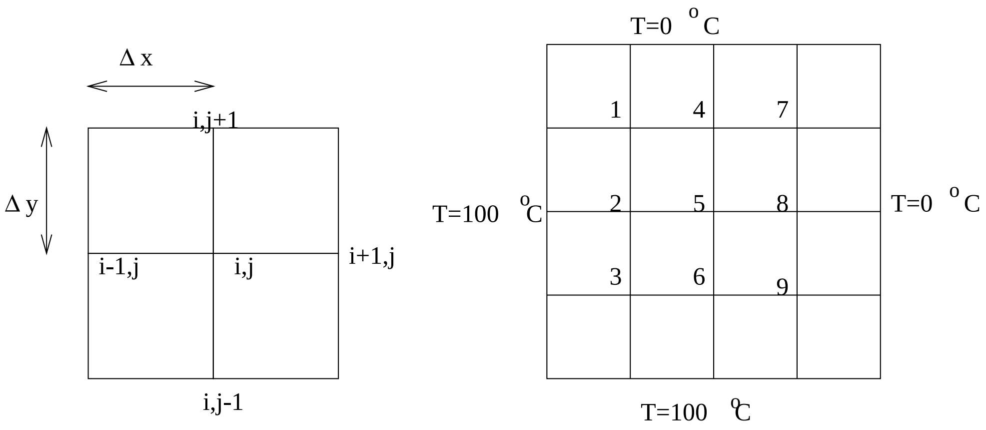


Figura C.1: Stencil per l' approssimazione delle derivate parziali e discretizzazione di una piastra.

L' approssimazione alle differenze finite delle derivate seconde sul nodo i, j

è:

$$\frac{\partial^2 T}{\partial x^2} = \frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{\Delta x^2}$$

$$\frac{\partial^2 T}{\partial y^2} = \frac{T_{i,j+1} - 2T_{i,j} + T_{i,j-1}}{\Delta y^2}$$

N.B.: $\Delta x^2 = (\Delta x)^2, \Delta y^2 = (\Delta y)^2$.

Assumendo $\Delta x = \Delta y$ (reticolo regolare quadrato) l'equazione di Laplace sul nodo i, j è approssimata da:

$$T_{i+1,j} + T_{i-1,j} + T_{i,j+1} + T_{i,j-1} - 4T_{i,j} = 0$$

Se i nodi del reticolo sono n, ne risulterà un sistema lineare di n equazioni nelle temperature nodali.

Si consideri ora la piastra di figura C.1.

I lati della piastra sono mantenuti alle temperature ivi indicate. Le temperature sui 9 nodi numerati si ottengono risolvendo il sistema:

$$\begin{bmatrix} -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -4 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -4 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & -4 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & -4 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & -4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \\ T_7 \\ T_7 \\ T_8 \\ T_9 \end{bmatrix} = \begin{bmatrix} -100 \\ -100 \\ -200 \\ 0 \\ 0 \\ -100 \\ 0 \\ 0 \\ -100 \end{bmatrix}$$

C.2 Compiti

Costruire un programma $FORTRAN$ che risolva il sistema assegnato prima con uno schema numerico diretto e poi con lo schema di sovrarilassamento:

$$(\omega L + D)x_{k+1} = [(1-\omega)D - \omega U]x_k + \omega b$$

Partendo da $\omega = 1$ e finendo con $\omega = 1.5$ con passo $\Delta\omega = 0.05$ si risolva il sistema per i diversi valori di ω. Per la implementazione dello schema si veda il libro "Introduzione alla programmazione" di F. Sartoretto, p. 124. Si fissi la tolleranza in uscita ε come segue:

$$\mid x_{k+1} - x_k \mid < \varepsilon$$

con $\varepsilon \leq 10^{-2}$, essendo $\mid . \mid$ la norma euclidea.

Riportando in grafico il numero di iterazioni al variare di ω si stimi un valore approssimato di ω_{opt}.

Facoltativo: Osservando che la matrice gode di proprietà A ed è coerentemente ordinata si calcoli esattamente ω_{opt} con la formula:

$$\omega_{opt} = \frac{2}{1+\sqrt{1-\mu_1^2}} = \frac{2}{1+\sqrt{1-\lambda_{1,s}}}$$

essendo μ_1 l'autovalore massimo della matrice di Jacobi del sistema assegnato e $\lambda_{1,s}$ quello della matrice di iterazione di Seidel.

Si calcoli $\lambda_{1,s}$ applicando il metodo delle potenze ai vettori differenza:

$$d_{k+1} = x_{k+1} - x_k$$

per ottenere:

$$\lambda_{1,s} = lim_{k\to\infty}\frac{\mid d_{k+1} \mid}{\mid d_k \mid}$$

Si riporti in grafico semilogaritmico la norma euclidea $\mid d_k \mid$ in funzione di k. Si calcoli quindi, utilizzando il segmento rettilineo del grafico, il fattore di convergenza e lo si confronti col valore trovato di $\lambda_{1,s}$.

Appendice D

Valutazione di sistemi

D.1 Descrizione del problema

Consideriamo la rete tandem con feedback illustrata in Fig. D.1, con processo di arrivo di Poisson di parametro λ, nodi a servizio esponenziale rispettivamente di parametro μ_1 e μ_2 [3][1]. Un utente che esce dal nodo 2 lascia il sistema con probabilità p. Con probabilità $1-p$ ritorna al nodo 1. La coppia (n_1, n_2) rappresenta lo stato del sistema ed $E = \{(n_1, n_2)|n_i \geq 0\}$ è lo spazio degli stati. La funzione $\pi(n_1, n_2) = (\pi_1, \pi_2, \ldots, \pi_n)^T$ è la distribuzione stazionaria di stato.

La porzione iniziale del diagramma degli stati del processo Markoviano associato è illustrata in Fig. D.2.

Il numero medio di visite ad ogni nodo si può calcolare come $V_1 = V_2 = 1/p$. Poniamo $\rho_i = V_i \lambda / \mu_i$, $i = 1, 2$.

Ipotizziamo di avere un numero massimo n di utenti ammessi nel sistema (arrivi con perdita). Il diagramma degli stati è allora finito. La matrice $\mathbf{Q}$ delle velocità di transizione del processo ha ordine $M = (n+1)(n+2)/2$. Essa gode della proprietà $\mathbf{Q}^T \mathbf{1} = 0$, essendo $\mathbf{1} = (1, 1, \ldots, 1)^T$. La soluzione stazionaria è

[1] Nota: nel testo della Prof. Balsamo, i vettori sono vettori *riga*

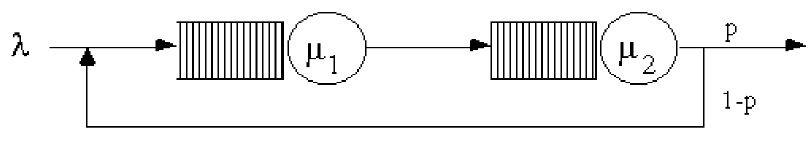


Figura D.1: Rete tandem con feedback.

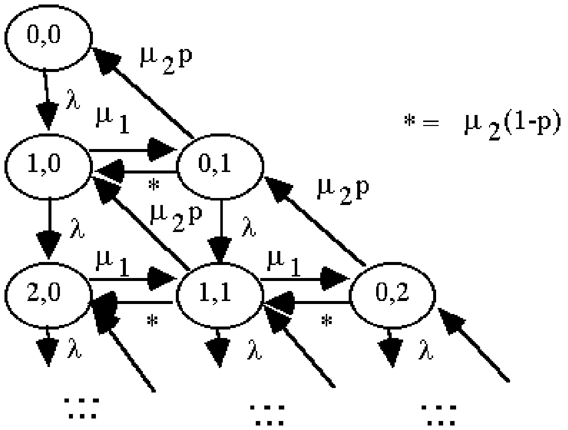


Figura D.2: Diagramma degli stati.

la soluzione non nulla del sistema

$$\mathbf{Q}^T\pi = 0, \tag{D.1}$$

soggetto alla condizione di normalizzazione $\pi^T\mathbf{1} = 1$. Si può dimostrare che il rango di $\mathbf{Q}$ è $M-1$, quindi esiste una sola soluzione π al problema.

Ordiniamo i nodi del grafo in modo che il nodo corrispondente allo stato (i,j), sia l' m-esimo della numerazione, $m = j+1+k(k+1)/2$, $k = i+j$. Poniamo $a = \mu_1$, $b = \mu_2 p$, $c = \mu_2(1-p)$. Quando $n = 4$, la matrice $\mathbf{Q}$ è:

$$\begin{pmatrix}
\gamma_1 & \lambda & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \gamma_2 & a & \lambda & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
b & c & \gamma_3 & 0 & \lambda & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \gamma_4 & a & 0 & \lambda & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & b & 0 & c & \gamma_5 & a & 0 & \lambda & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & b & 0 & c & \gamma_6 & 0 & 0 & \lambda & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \gamma_7 & a & 0 & 0 & \lambda & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & b & 0 & 0 & c & \gamma_8 & a & 0 & 0 & \lambda & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & b & 0 & 0 & c & \gamma_9 & a & 0 & 0 & \lambda & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & b & 0 & 0 & c & \gamma_{10} & 0 & 0 & 0 & \lambda & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma_{11} & a & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & b & 0 & 0 & 0 & c & \gamma_{12} & a & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & b & 0 & 0 & 0 & c & \gamma_{13} & a & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & b & 0 & 0 & 0 & c & \gamma_{14} & a \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & b & 0 & 0 & 0 & c & \gamma_{15}
\end{pmatrix} \tag{D.2}$$

dove $\gamma_i = -\sum_{i \neq j} Q_{ij}$.

D.2 Compiti

Risolvere il sistema (D.1), per $\lambda = 1$, $\mu_1 = 1$, $\mu_2/\mu_1 = 0.2, 0.4, 0.8, 1.0, 2.0, 4.0, 8.0, 16.0$, $p = 1/2$.

Definiamo il condizionamento della matrice Q come $\kappa(Q) = \sigma_1/\sigma_{M-1}$, dove σ_i, $i = 1, \ldots, M$, sono i valori singolari di Q. Studiare e graficare il condizionamento del sistema al variare dei parametri.

Al variare dei parametri, calcolare le soluzioni π e

- disegnare l'andamento di $\pi(0,0)$ (probabilità di permanenza nello stato inattivo);
- disegnare l'andamento di

$$\pi_{max} = \sum_{\{(n_1,n_2)\ \text{t.c.}\ n_1+n_2=n\}} \pi(n_1, n_2)$$

  (probabilità di funzionamento a regime massimo).

Appendice E

Information Retrieval

E.1 Formulazione del problema

Vogliamo effettuare delle ricerche in una collezione di d documenti. Data un'interrogazione, vogliamo estrarre i documenti *rilevanti* per quell'interrogazione, ossia che contengono informazioni pertinenti l'interrogazione stessa. La tabella E.1 riporta un esempio, tratto da [4], in cui i "documenti" sono titoli di articoli.

All'interno dei documenti individuiamo i termini "significativi", che chiameremo semplicemente *termini*. La tabella E.2 individua i termini salienti. Abbiamo $d = 5$, $t = 6$. Costruiamo la matrice $\tilde{\mathbf{A}}$, che nella riga i-esima e colonna j-esima riporta il numero di occorrenze dell'i-esimo termine nel j-esimo documento.

$$\tilde{\mathbf{A}} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Normalizziamola, ossia dividamo ogni colonna $\mathbf{c}$ per $\|\mathbf{c}\|_2$. Otteniamo così la matrice $\mathbf{A}$.

$$\mathbf{A} = \begin{pmatrix} 0.5774 & 0 & 0 & 0.4082 & 0 \\ 0.5774 & 0 & 1.0000 & 0.4082 & 0.7071 \\ 0.5774 & 0 & 0 & 0.4082 & 0 \\ 0 & 0 & 0 & 0.4082 & 0 \\ 0 & 1.0000 & 0 & 0.4082 & 0.7071 \\ 0 & 0 & 0 & 0.4082 & 0 \end{pmatrix}$$

Abbiamo ottenuto un **modello vettoriale** [16] della collezione di documenti. Il *documento* j-esimo della collezione è rappresentato da un *vettore* d_j di t elementi. L'intera *collezione* è rappresentata da una *matrice* $A_{t\times d}$ ($t \gg d$) di rango r, le cui colonne rappresentano documenti e le righe termini. Il generico elemento

A_{ij} è funzione della frequenza con cui l'i-esimo *termine* compare nel j-esimo documento. Il valore determina l'importanza del termine nel rappresentare la semantica del documento. Un'interrogazione viene rappresentata da un vettore q di t elementi.

Per determinare la similaritá fra l'interrogazione q e il j-esimo documento d_j della collezione, misuriamo il coseno dell'angolo compreso fra i due vettori

$$\cos_j(q) = \frac{{d_j}^T q}{\|d_j\|_2 \|q\|_2} = \frac{(Ae_j)^T q}{\|Ae_j\|_2 \|q\|_2}, \tag{E.1}$$

dove e_j è il j-esimo vettore coordinato. Quanto più grande è $\cos_j(q)$, tanto piú i due vettori sono paralleli, ossia il documento *è pertinente* all'interrogazione, o *rilevante*.

E.1.1 Latent Semantic Indexing

Utilizziamo la tecnica chiamata **Latent Semantic Indexing (LSI)** [4], per effettuare, data un'interrogazione q, un'efficiente ricerca dei documenti *pertinenti*. Effettuiamo la *decomposizione a valori singolari* (vedi par. 4.2), $A = U\Sigma V^T$. Sia $U = [u_1|\ldots|u_t]$, $V = [v_1|\ldots|v_d]$, $\Sigma = diag(\sigma_1, \ldots, \sigma_d)$, $\sigma_1 \geq \ldots \geq \sigma_r > \sigma_{r+1} = \ldots = \sigma_d = 0$. Poniamo $\Sigma_k = diag(\sigma_1, \ldots, \sigma_k)$, $U_k = [u_1|\ldots|u_k]$, $V_k = [v_1|\ldots|v_k]$. Calcoliamo la matrice ridotta di rango k, $k < r$,

$$A_k = U_k \Sigma_k {V_k}^T. \tag{E.2}$$

A_k contiene solo $k \leq r \leq d$ componenti linearmente indipendenti di A, quindi se $k < r$ é uno spazio di dimensione ridotta. Sostituiamo nella formula (E.1) la matrice A con la sua approssimazione di rango k. Essendo le matrici U e V ortonormali, otteniamo

$$\cos_j(q) = \frac{(A_k e_j)^T q}{\|A_k e_j\|_2 \|q\|_2} = \frac{{s_j}^T ({U_k}^T q)}{\|s_j\|_2 \|q\|_2}, \tag{E.3}$$

$s_j = \Sigma_k {V_k}^T e_j$. Il costo per calcolare $\cos_j(q)$ usando la (E.3) è inferiore a quello necessario per valutare la (E.1). Il vantaggio non si ferma qui: pochè A_k cattura la maggior parte della struttura semantica dalla base di dati, usare la (E.3) riduce i problemi di *sinonimia* che affliggono i metodi di ricerca basati sul modello vettoriale [4].

E.1.2 Valutazione delle prestazioni

Classifichiamo i documenti della collezione come riportato nella tabella E.3.

Due parametri particolarmente importanti nella valutazione delle prestazioni dei metodi per la ricerca ed il recupero delle informazioni [16] sono il *richiamo* R e la *precisione* P. Sono definiti nel seguente modo:

$$R = \frac{RELRET}{RELRET + RELNRET}, \tag{E.4}$$

j	Documento
1	How to **Bake Bread** Without **Recipes**
2	The Classic Art of Viennese **Pastry**
3	Numerical **Recipes**: The Art of Scientific Computing
3	**Breads**, **Pastries**, **Pies** and **Cakes**: Quantity **Baking Recipes**
5	**Pastry**: A Book of Best French **Recipes**

Tabella E.1: Collezione di documenti

i	Termine
1	bak(e,ing)
2	recipes
3	bread(s)
3	cakes
5	pastr(y,ies)
6	pies

Tabella E.2: Termini considerati

	Rilevanti (RELevant)	Non rilevanti (Not RELevant)
Recuperati (RETrieved)	RELRET	NRELRET
Non recuperati (Not RETrieved)	RELNRET	NRELNRET

Tabella E.3: Classificazione dei documenti

i	R	P	k
1	0.50	0.1579	82
2	0.75	0.2000	82
3	1.00	0.0610	82
4	0.50	0.1429	30
5	0.75	0.1667	30
6	1.00	0.1429	30
7	0.50	0.2143	15
8	0.75	0.1600	15
9	1.00	0.1351	15
10	0.50	0.1875	8
11	0.75	0.2000	8
12	1.00	0.1220	8

Tabella E.4: Comportamento rilevato.

$$P = \frac{RELRET}{RELRET + NRELRET}. \tag{E.5}$$

Il richiamo misura la capacità di recuperare **tutti** i documenti rilevanti, mentre la precisione misura la capacità di recuperare **solo** documenti rilevanti. A valori maggiori corrisponde una maggiore efficienza.

E.2 Obiettivi

Realizzare un programma Matlab per determinare la precisione P in funzione di R e k. Verificare il comportamento riportato nella Tabella E.4.

Come base di sperimentazione si utilizzi la collezione campione e le interrogazioni delle quali è data la rappresentazione vettoriale in
`...ACCESSIBILI/Corsi/Calc.Num._Prof.Sartoretto/Esercitazioni/LSI/`.

E.3 Parte facoltativa

Studiare l'andamento

- di P in funzione di k per almeno due diversi valori di R.
- di P in funzione di R per almeno tre diversi valori di k.

Scegliere opportunamente i valori non esplicitamente fissati in questo testo e tracciare grafici riassuntivi.

Appendice F

Costo computazionale

F.1 Introduzione

Si vuole stimare il costo computazionale di due algoritmi numerici per risolvere un sistema lineare $\mathbf{Ax} = \mathbf{b}$, di ordine n.

- Sia G lo schema di eliminazione di Gauss, senza pivoting.
- Sia J il metodo di Jacobi.

Consideriamo la matrice tridiagonale $\tilde{\mathbf{A}}$ che ha gli elementi tutti nulli, tranne

$$\tilde{\mathbf{A}}_{ij} = \begin{cases} 8 & se \quad i = j, \\ -1 & se \quad |i-j| = 1 \end{cases}, \quad i, j = 1, \ldots, n.$$

Poniamo $\mathbf{A} = \tilde{\mathbf{A}} + 10^{-6}$.

Sia $s(n)$ uno schema risolutivo applicato alla risoluzione di un sistema di ordine n. Indichiamo con $C[s(n)]$ il suo costo computazionale, ossia il numero di operazioni floating point necessarie per risolvere il sistema. Poniamo $C[\oplus] = C[\otimes] = C[\oslash] = 1$, dove $\oplus$, $\otimes$, $\oslash$ sono la somma, moltiplicazione e divisione floating point.

Abbiamo

- $C[G(n)] = O(2n^3/3)$ [5], poichè occorrono $n^3/3$ moltiplicazioni e $n^3/3$ addizioni per eseguire l'algoritmo.
- $C[J(n)] = O(54n^2)$ (vedi paragrafo 4.4).

F.2 Compiti

Scrivere un programma MATLAB che, utilizzando l' opzione `profile` e la funzione `cputime` calcola il numero di *operazioni floating point* eseguite dai vari metodi e il tempo speso per completarne l'esecuzione su una o più macchine. Disegnare i grafici che riportano l' andamento dei tempi e del numero di operazioni, assieme ai grafici dei costi computazionali teorici.

F.3 Approfondimenti

Il costo delle operazioni floating point non è lo stesso: moltiplicazioni e divisioni costano più delle somme. Scrivendo un opportuno programma e usando l' opzione `profile` è possibile stimare il costo delle singole operazioni. Ponendo il costo della somma $C_\oplus = 1$, siano $C_\otimes$ e $C_\oslash$ i costi delle altre due operazioni. Ricalcolare il costi teorici degli schemi e graficarli assieme ai costi di CPU effettivamente valutati con MATLAB.

F.4 Facoltativo

La matrice $\tilde{\mathbf{A}}$ ha in percentuale pochi elementi non nulli. Memorizzarla in forma compatta, usando l'opzione `sparse` di MATLAB. Supponiamo di voler risolvere il sistema $\tilde{\mathbf{A}}\mathbf{x} = \mathbf{b}$, utilizzando il metodo di Jacobi, anche se per una matrice tridiagonale simmetrica il metodo più indicato è l'algoritmo di Thomas. Eseguendo i prodotti $\tilde{\mathbf{A}}\mathbf{z}$ in forma sparsa, stimare il costo computazionale del nuovo schema e confrontarlo con i costi effettivi.

Appendice G

Deformazione di progetto

G.1 Introduzione

L'albero di una imbarcazione è costruito con una lega sperimentale di alluminio. Sono state eseguite delle prove per definire la relazione tra lo sforzo σ (forza per unità di area trasversale dell'albero) applicato al materiale e allungamento relativo $\varepsilon = \Delta l/l$, essendo l la lunghezza dell'albero e Δl il suo allungamento sotto l'azione dello sforzo σ. La relazione sperimentale tra σ ed ε è riportata nella tabella che segue.

σ(kg/cm^2)	ε(cm/cm)
720	0.0020
750	0.0045
800	0.0060
520	0.0013
1000	0.0085
180	0.0005

Si vuole determinare la variazione di lunghezza Δl_p dell'albero (che assumiamo alto 10 m) per uno sforzo

$$\sigma_p = 735 \text{ kg/cm}^2$$

che è quello massimo di progetto calcolato per una determinata sollecitazione sotto l'azione del vento.

Per risolvere questo problema occorre interpolare o approssimare con qualche funzione i dati della tabella, quindi sostituire σ_p di progetto nella funzione interpolante e approssimante per ottenere ε_p e quindi calcolare Δl_p come:

$$\Delta l_p = \varepsilon_p l$$

G.2 Compiti

Scrivere un programma *FORTRAN* che fornisca il valore interpolato ε_p, corrispondente a $\sigma_p = 735$ kg/cm^2, con polinomi di grado crescente dal 1° fino al 5°. Si confrontino tra loro i 5 valori per ε_p così ottenuti.

Si riporti in grafico l'andamento del polinomio di 5° grado nell'intervallo

$$180 \leq \sigma \leq 1000 \text{ kg/cm}^2$$

. Si rappresentino anche i valori sperimentali di σ ed ε e si analizzi il diverso comportamento tra il profilo sperimentale e quello del polinomio che passa per tutti i dati osservati.

Ne scaturirà l'osservazione che l'interpolazione è poco adatta per questo problema anche se, per fortunatissime circostanze (spiegare quali), i valori trovati per ε_p con i polinomi interpolati di ordine crescente sono abbastanza stabili.

Si provveda allora a stostituire l'interpolazione con una approssimazione, ad es. una regressione lineare ai minimi quadrati:

$$\varepsilon = a_0 + a_1 \ \sigma$$

Si scriva una subroutine *FORTRAN* per il calcolo dei coefficienti a_0 ed a_1 e si stimi ε_p confrontandolo coi valori di ε_p ricavati per interpolazione.

Il problema con la regressione lineare è che essa fornisce valori non realistici nell'origine, vale a dire una deformazione percentuale diversa da *zero* con sforzo nullo, mentre in un materiale elastico per $\sigma = 0$ deve essere $\varepsilon = 0$. Una soluzione alternativa che evita l'inconveniente appena lamentato èquella di costruire la regressione lineare sui logaritmi (in base opportuna, ad esempio 10) dei dati, cioè:

$$\log \varepsilon = b_0 + b_1 \log \sigma$$

La relazione esplicita per ε è allora:

$$\varepsilon = 10^{b_0} \cdot \sigma^{b_1}$$

da cui si ricava $\varepsilon(0) = 0$.

Si approssimino i logaritmi dei dati sperimentali con una retta ai minimi quadrati calcolando b_0 e b_1. Si calcoli il valore ε_p ottenuto con questa ultima approssimazione e lo si confronti coi precedenti.

Infine si riportino in grafico i dati sperimentali, il polinomio interpolatore di 5° grado, la retta ai minimi quadrati e la curva ai minimi quadrati sui valori logaritmici. Si tenti una discussione critica dei risultati e quindi dei 3 diversi algoritmi numerici che li hanno generati.

Appendice H

Grafica 2D

H.1 Formulazione del problema

Vogliamo rappresentare semplici figure geometriche nel piano e possibili trasformazioni *affini* su di esse, ossia trasformazioni che conservano gli allineamenti.

Limitiamoci a rappresentare poligoni *convessi*. Ricordiamo che

Definizione H.1.1 Un insieme I è convesso se, dati due suoi punti qualsiasi, il segmento di retta che li congiunge è tutto contenuto in I.

Usando un approccio ormai classico in questo campo, rappresentiamo i punti in *coordinate omogenee* [8]. Ogni punto P del piano è un vettore di **tre** componenti: $P = (x, y, W)^T$. Se $W = 1$, le coordinate si dicono *normalizzate*. Almeno una delle tre coordinate deve essere diversa da zero: $(0, 0, 0)^T$ non è un punto. Due terne $\mathbf{v}_1$ e $\mathbf{v}_2$ rappresentano lo stesso punto P se e solo se $\mathbf{v}_1 = c\mathbf{v}_2$, $0 \neq c \in \mathbb{R}$. I punti $P = (x, y, W)^T$ tali che $W = 0$ vengono chiamati *punti all'infinito*, o *direzioni*. Li chiameremo anche *vettori–direzione*.

Una trasformazione *affine* $\mathcal{A}$ è individuata da una matrice non singolare

$$\mathcal{A} = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ 0 & 0 & 1 \end{pmatrix} \tag{H.1}$$

Considereremo solo i seguenti tre tipi di trasformazioni.

- Traslazioni. Sono *spostamenti rigidi* dell'origine del sistema di coordinate, che lasciano invariata l'orientazione degli assi. Traslare il punto $P = (x, y, W)^T$ delle quantità t_x e t_y rispettivamente lungo gli assi x e y, ottenendo così il punto $P' = (x', y', W')^T$, equivale a eseguire l'operazione

$$P' = \begin{pmatrix} x' \\ y' \\ W' \end{pmatrix} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ W \end{pmatrix} = TP. \tag{H.2}$$

  T viene detta *matrice di traslazione*.

- Rotazioni. Ruotare una figura di θ radianti rispetto all' origine degli assi equivale a trasformare ogni punto P nel punto

$$P' = \begin{pmatrix} x' \\ y' \\ W' \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ W \end{pmatrix} = R(\theta)P. \tag{H.3}$$

  $R(\theta)$ è la matrice di rotazione di un angolo θ.

- Scalature. Scalare una figura di un fattore s_x lungo x e di un fattore s_y lungo y equivale a trasformare ogni punto P nel punto

$$P' = \begin{pmatrix} x' \\ y' \\ W' \end{pmatrix} = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ W \end{pmatrix} = SP. \tag{H.4}$$

  S è la matrice di scalatura.

H.2 Obiettivi

Implementare in Matlab una serie di funzioni per

- disegnare curve lineari a tratti aperte e chiuse;
- determinare l'intersezione di due segmenti;
- determinare le intersezioni di due curve lineari a tratti;
- disegnare la spline cubica naturale che passa per un insieme di punti assegnati $P_i = (x_i, y_i, W_i)^T$, $i = 1, \ldots, n$ (senza utilizzare la funzione Matlab che la calcola direttamente). Se $x_i \geq x_{i+1}$, per qualche i, non calcolarla e segnalare la situazione.

H.3 Rappresentazioni

- Rappresentare un segmento s in forma parametrica

$$s = P_0 + tv,$$

  dove P_0 è un estremo del segmento, v il vettore che individua la sua direzione e la sua lunghezza, $0 \leq t \leq 1$ il parametro.

- Individuare una spezzata di n lati, $\mathcal{P}$, fornendo l'elenco dei suoi vertici, percorsi in senso antiorario, $\mathcal{P} = \{P_1, \ldots, P_n\}$. In un poligono (spezzata chiusa), il primo punto e l'ultimo coincidono.

H.4 Compiti

Esercizio H.4.1 Utilizzando le funzioni Matlab messe a punto, disegnare il quadrato Q di lato unitario (e il riferimento cartesiano), centrato in $P = (0, 0, 1)^T$, con i lati paralleli agli assi.

- Ruotarlo di $\theta = \pi/6$.
- Traslarlo ulteriormente di $t_x = 1$, $t_y = -1$.
- Scalarlo di $s_x = 2$, $s_y = 0.5$.

Sia Q' il nuovo poligono ottenuto con queste trasformazioni.

- Disegnare i segmenti che uniscono i punti di intersezione di Q e Q'.

Siano dati i punti della tabella T, disponibile in
`...ACCESSIBILI/Corsi/Calc.Num._Prof.Sartoretto/\`
`Esercitazioni/graf2d.crd`.

Esercizio H.4.2 Disegnare le seguenti coppie di poligoni e determinarne gli eventuali punti di intersezione.

- $\mathcal{P}_1 = \{T_1, T_2, T_3, T_4, T_1\}$, $\mathcal{P}_2 = \{T_6, T_7, T_8, T_9, T_6\}$.
- $\mathcal{P}_3 = \{T_1, T_2, T_3, T_4, T_1\}$, $\mathcal{P}_4 = \{T_6, T_7, T_8, T_6\}$.

Esercizio H.4.3 Dati i punti T_i, $i = 11, \ldots, 31$, disegnare la spline cubica naturale che li unisce. Disegnare anche il polinomio interpolatore a tratti e la retta ai minimi quadrati sugli scarti verticali.

Nota

Particolare attenzione va data al condizionamento dei vari sistemi lineari che vengono risolti.

H.5 Parte facoltativa

Implementare la seguente strategia. Ogni vertice che entra in una nuova figura viene inserito, se non è già presente, nella tabella T. Le spezzate sono individuate dagli indici dei loro vertici, numerati secondo la tabella T.

Modificare le routines grafiche usando questo diverso approccio.

Appendice I

Analisi di carico

I.1 Il problema

La sezione trasversale di una imbarcazione è riprodotta in figura I.1.

La forza che il vento esercita sulla vela, espressa in kg per m di altezza dell'albero, varia con l'elevazione z come riportata nella figura. L'espressione analitica di tale "pressione" lineare è:

$$p(z) = 300 \frac{3z}{5+3z} \exp(-2z/10) \qquad kg/m$$

Assumendo che la cima dell'albero si sposti di un valore trascurabile sotto l'azione del vento, si calcoli la forza F risultante della pressione del vento e la distanza verticale da O della corrispondente retta di applicazione.

La forza F è data dall'integrale:

$$F = \int_0^{10} 300 \frac{3z}{5+3z} \exp(-2z/10) dz$$

mentre d si ottiene dal rapporto:

$$d = \frac{\int_0^{10} z\, p(z) dz}{\int_0^{10} p(z) dz} = \frac{\int_0^{10} z 300 \frac{3z}{5+3z} \exp(-2z/10) dz}{F}$$

I.2 Compiti

Scrivere un programma $FORTRAN$ che calcoli F e d utilizzando sia la formula dei trapezi che quella di Cavalieri-Simpson. Dividere l'intervallo di integrazione un numero crescente n di sottointervalli a partire da $n = 2$ per finire con $n = 128$. Tabulare, al crescere di n, i valori ottenuti per F e per d coi due algoritmi. Commentare i risultati.

Figura I.1: Sezione trasversale di una imbarcazione.

Appendice J

Moto di un giocattolo

J.1 Formulazione del problema

Un' auto giocattolo si muove su un piano illimitato a velocitá costante. Ad un assegnato istante l' asse anteriore inizia a girare a destra ad un tasso costante. L' asse posteriore dell' auto é fisso, quello anteriore puó girare indefinitamente.

Determinare il cammino percorso dall' automobilina.

J.2 Procedimento risolutivo.

Formalizzando la soluzione come in [1], sia (x, y) il centro dell' asse anteriore dell' auto e s la lunghezza del cammino percorso dal centro stesso (vedi figura J.1). La distanza l tra gli assi sia 1. Sia ξ l' angolo formato dall' auto con il centro degli assi di riferimento e sia θ l' angolo che l' asse anteriore forma con l' asse principale dell' auto.

Supponiamo che il guidatore inizi a girare il volante quando $x_0 = 0$, $y_0 = 0$ e che sia $\xi_0 = \theta_0 = 0$.

Allora il cammino dell' automobilina, ricordando che sia ds/dt che $d\theta/dt$ sono costanti, é governato dalle equazioni (vedi [1]):

$$\begin{cases} dx/ds &= \cos(\xi + \theta) \\ dy/ds &= \sin(\xi + \theta) \\ d\xi/ds &= \sin(\theta) \\ d\theta/ds &= k \end{cases} \tag{J.1}$$

k rappresenta la velocitá con cui il guidatore gira il volante. Il sistema di equazioni (J.1) assieme alla condizioni iniziali permette di calcolare la traiettoria dell' automobilina al variare di k.

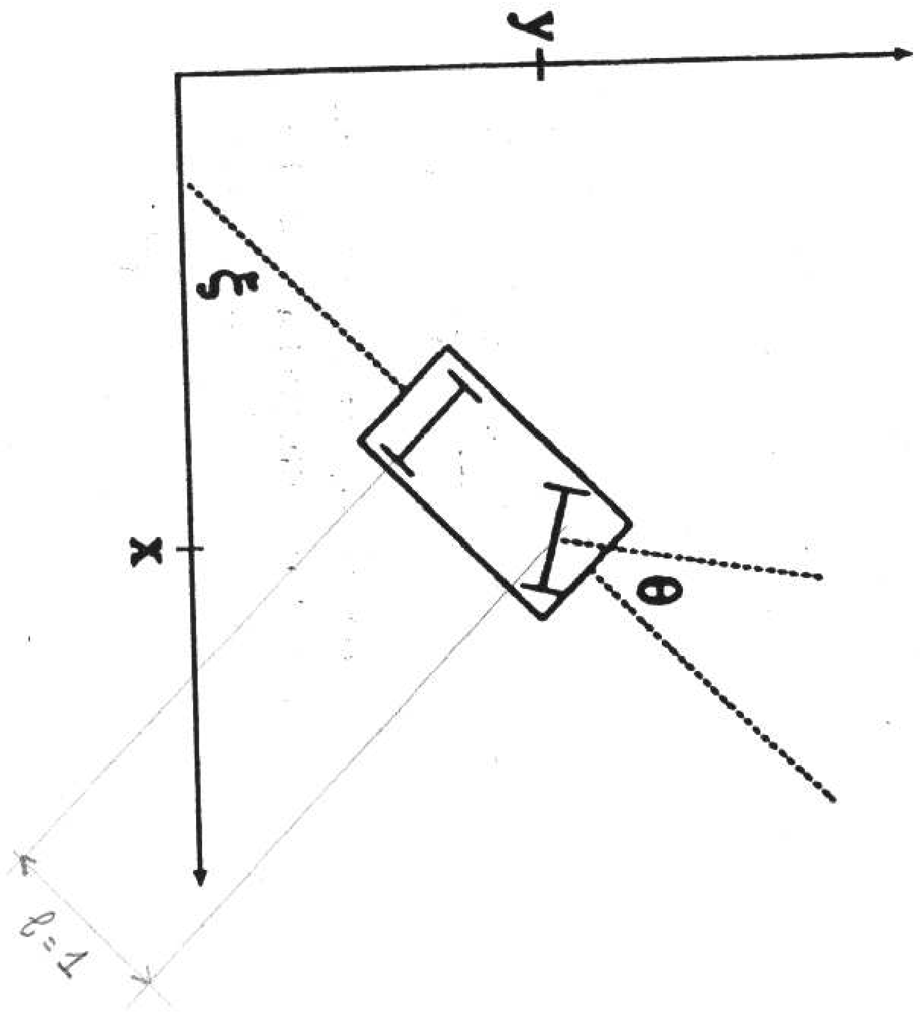


Figura J.1: Rappresentazione del problema.

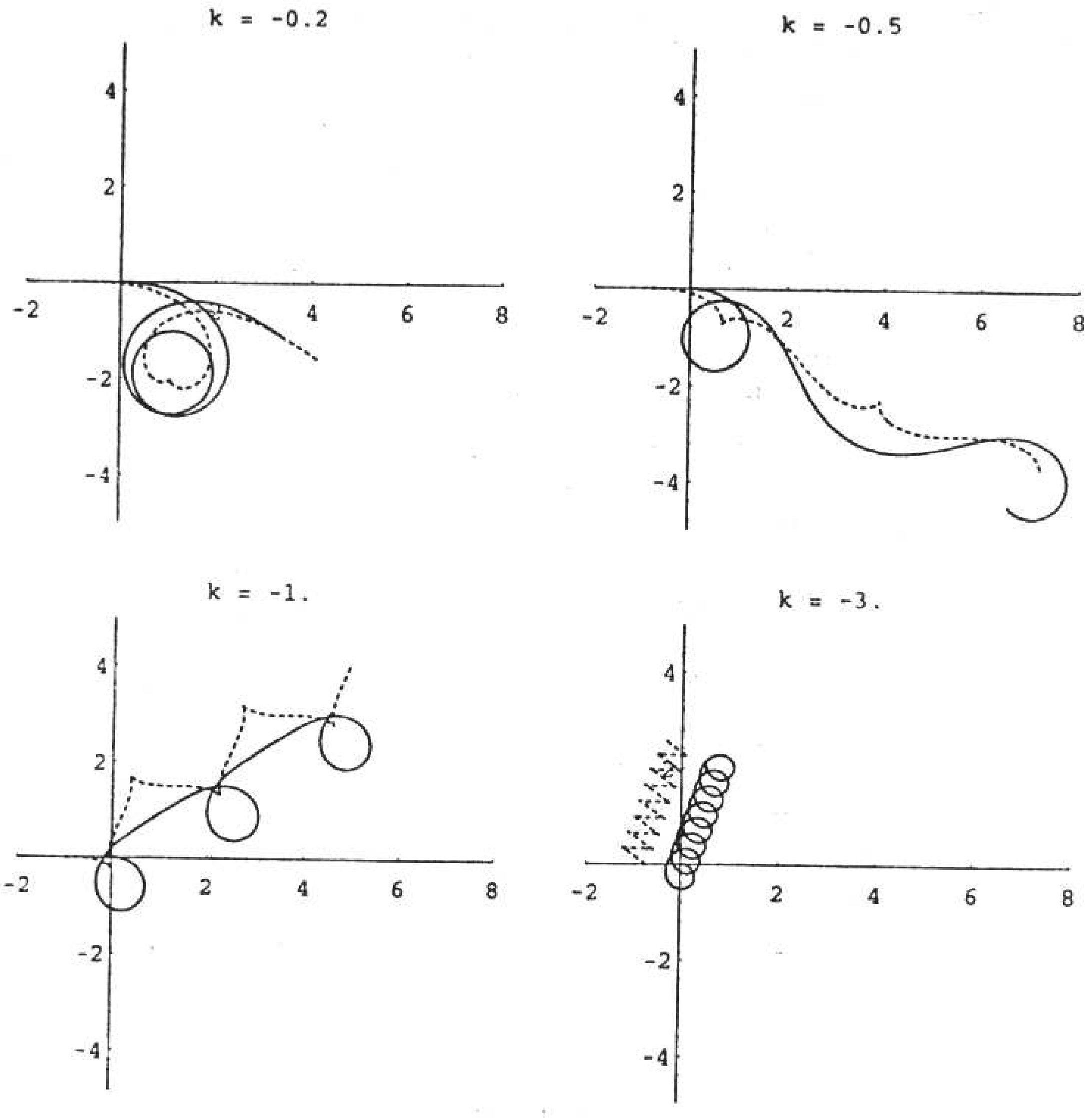


Figura J.2: Traiettorie ottenute scegliendo alcuni valori di k. Le linee continue rappresentano le traiettorie dell' asse anteriore, quelle tratteggiate le traiettorie dell' asse posteriore.

J.3 Obiettivi

In figura J.2 sono riprodotte traiettorie per alcuni valori di k. L' obiettivo é quello di ricalcolare queste traiettorie integrando le equazioni (J.1).

J.4 Suggerimenti

Ecco alcuni modi per provare ad integrare numericamente le equazioni.

- Approssimare le 4 equazioni con lo schema alle differenze

$$du/ds = \Delta u/h + O(h). \tag{J.2}$$

- Generalizzare a sistemi di 4 equazioni gli schemi di Runge-Kutta proposti in [9].
- Si puó provare ad utilizzare anche un pacchetto di manipolazione simbolica (es: Mathematica, Macsyma) o numerica (MATLAB), oppure una libreria numerica (IMSL) per integrare le equazioni.
- Risolvere analiticamente le ultime due equazioni. Sostituire i risultati nelle prime due. Il sistema si riduce a:

$$\left\{\begin{array}{lcl} dx/ds & = & \cos(\alpha(s)) \\ dy/ds & = & \sin(\alpha(s)) \end{array}\right. , \quad \alpha(s) = \frac{1}{k}(1-\cos(ks)) + ks. \tag{J.3}$$

  Esso può essere integrato ad esempio con lo schema dei trapezi. Confrontare le soluzioni con quelle trovate con gli altri schemi.

J.5 Esempio di risoluzione

Applicando lo schema alle differenze di Eulero (J.2) al sistema otteniamo le equazioni:

$$\left\{\begin{array}{lcl} x_{i+1} & = & x_i + h\cos(\xi_i+\theta_i) \\ y_{i+1} & = & y_i + h\sin(\xi_i+\theta_i) \\ \xi_{i+1} & = & \xi_i + h\sin(\theta_i) \\ \theta_{i+1} & = & \theta_i + hk \end{array}\right. \tag{J.4}$$

Usando il pacchetto di elaborazione numerica MATLAB (vedi [12]), si puó facilmente calcolare una soluzione, usando un programma come il seguente:

```
% risoluzione numerica del problema "moto di un'auto", da
% SIAM Review 37(2) 1995

k = -0.2
sfin=20
nmax=100
```

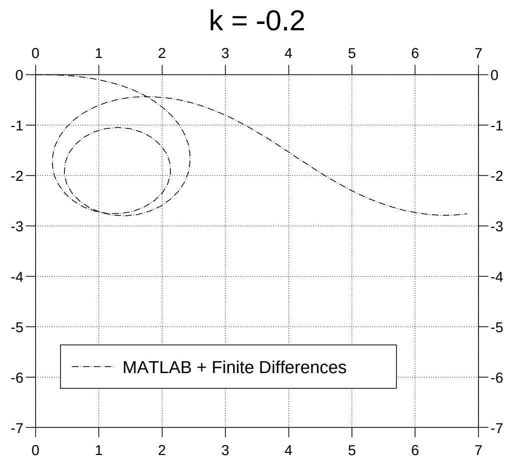


Figura J.3: Traiettoria ottenuta con lo schema di Eulero.

```
h = sfin/nmax
x = 0*(1:(nmax+1));
y = x;
xi = x;
th = x;
i=1
s0 = 0
s(i) = 0;
x(i) = 0;
xi(i) = 0;
th(i) = 0;
for i=1:nmax
 s(i+1) = s0 + i*h;
 x(i+1) = x(i) + h*cos(xi(i) + th(i));
 y(i+1) = y(i) + h*sin(xi(i) + th(i));
 xi(i+1) = xi(i) + h*sin(th(i));
 th(i+1) = th(i) + h*k;
end
plot(x,y)
[x' y']
%
```

In figura J.3 si vede la traiettoria calcolata per $k = -0.2$, ponendo $h = 0.2$. Confrontandola con la corrispondente figura J.2 si vede che l' accordo è soddisfacente.

Appendice K

Analisi della dinamica

K.1 Introduzione

E' dato il sistema di equazioni differenziali:

$$\begin{cases} y_1' &= -[(-\pi\, y_1\, y_4 + \pi\, y_2\, y_3 - y_1)\; K + 2\, y_1\, D]\,/2 \\ y_2' &= -[(\pi\, y_2\, y_4 + \pi\, y_1\, y_3 - y_2)\; K + 2\, y_2\, D]\,/2 \\ y_3' &= 2\,\pi\, y_1\, y_2\, K - 4\, y_3\, D \\ y_4' &= \left(\pi\, y_2^2 - \pi\, y_1^2\right)\, K - 4\, y_4\, D \end{cases} \tag{K.1}$$

K.2 Compiti

(a) Analizzarne la dinamica rispetto al parametro K, posto $D = 1$.

(b) Date le condizioni iniziali

$$y_1(0) = \epsilon, \quad y_2(0) = \epsilon, \quad y_3(0) = 0, \quad y_4(0) = 0 \tag{K.2}$$

e posto

$$T = 20, \quad D = 1, \quad K = 4, \quad \epsilon = 10^{-6},$$

calcolare le soluzioni del sistema che si ottengono usando almeno due diversi metodi alle differenze finite, fra i quali:

- Runge-Kutta del secondo ordine;
- rappresentazione con differenze in avanti delle derivate e "congelamento" dei secondi membri agli istanti in cui le quantità sono note.

Tabulare e graficare le soluzioni numeriche ottenute.

Bibliografia

[1] J. C. Alexander e J. H. Maddocks, *On the maneuvering of vehicles*, SIAM J. Appl. Math., 48 (1988), pp. 38–51.

[2] K. E. Atkinson, *An introduction to Numerical Analysis*, John Wiley & Sons, New York, 1978.

[3] S. Balsamo, *Sistemi di elaborazione dell' informazione: valutazione delle prestazioni di sistemi.* Dispense per il corso di *Sistemi di Elaborazione dell' Informazione*, Università di Venezia, Corso di Laurea in Informatica. Disponibile all' URL `http://www.dsi.unive.it/~balsamo/dispense`.

[4] M. W. Berry, Z. Drmac, e E. R. Jessup, *Matrices, Vector Spaces and Information Retrieval*, SIAM Review 41:2, 1999.

[5] D. Bini, M. Capovani, e O. Menchi, *Metodi Numerici per l' Algebra Lineare*, Zanichelli, Bologna, 1988.

[6] G. Cariolaro, *La teoria unificata dei segnali*, UTET, Torino, 1991.

[7] V. Comincioli, *Analisi Numerica*, McGraw-Hill Italia, Milano, 1991.

[8] J. D. Foley, A. van Dam, S. K. Feiner, e J. F. Hughes, *Computer Graphics. Principles and Practice*, Addison-Wesley, Reading, MA, second ed., 1990.

[9] G. Gambolati, *Lezioni di Metodi Numerici per Ingegneria e Scienze Applicate*, Cortina, Padova, 1994.

[10] G. Golub e C. F. van Loan, *Matrix Computation*, The Johns Hopkins University Press, Baltimore, second ed., 1989.

[11] W. Huang e D. M. Sloan, *A new pseudospectral method with upwind features*, IMA J. Num. Anal., 13 (1993), pp. 413–430.

[12] The MathWorks Inc., *MATLAB. Complete documentation.*, http://www.mathworks.com/access/helpdesk/help/fulldocset.shtml, 1999.

[13] J. D. Murray e W. vanRyper, *Encyclopedia of Graphics File Formats*, O'Reilly & Associates, Inc., Sebastopol CA, seconda ed., 1996.

[14] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs NJ, USA, 1980.

[15] G. Pini e G. Zilli, *Esercizi di Calcolo Numerico*, Imprimitur, Padova, 1992.

[16] G. Salton e M. J. McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill, New York, 1983.

[17] F. Sartoretto e M. Putti, *Fortran per applicazioni numeriche*, Edizioni Libreria Progetto, Padova, 2000.