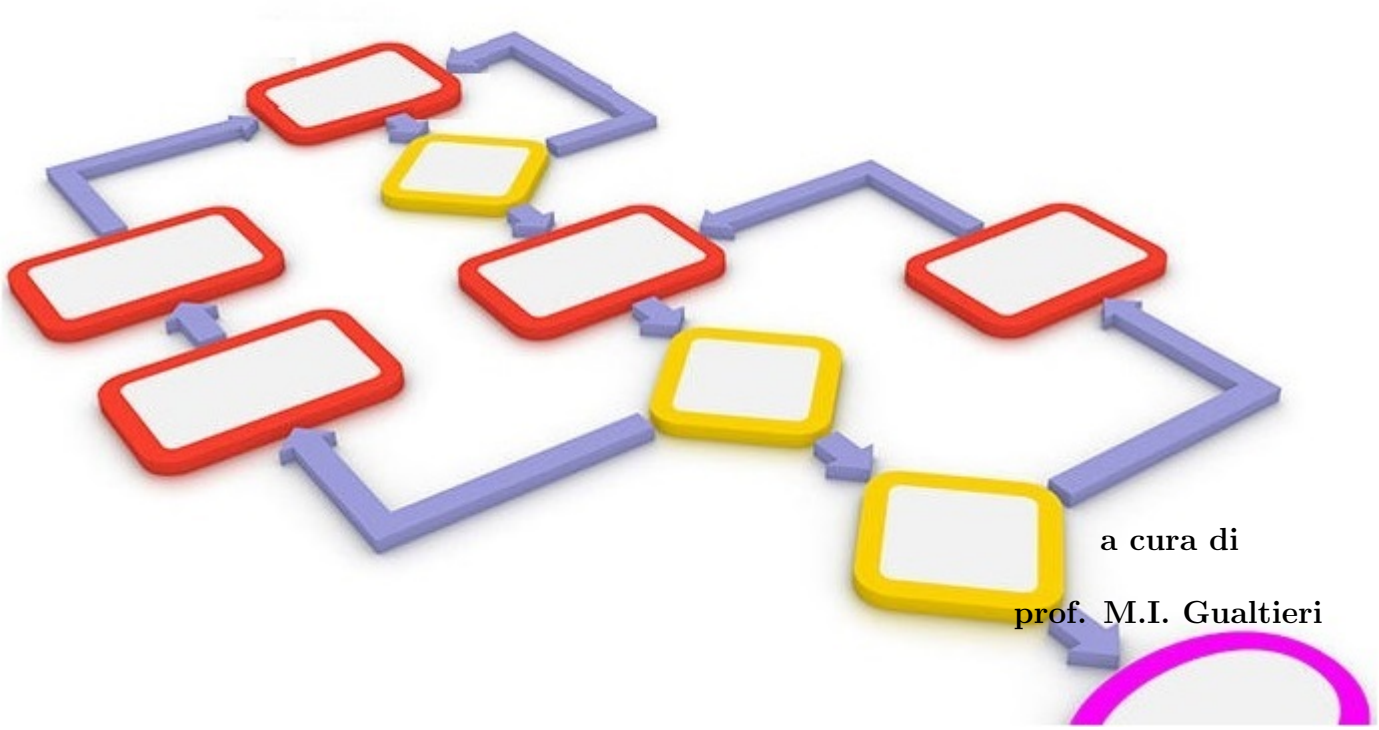
Corso di Laurea in Matematica

a.a. 2020-2021

Dispense del corso di

Laboratorio di Programmazione e Calcolo

Schemi di Calcolo



a cura di

prof. M.I. Gualtieri

Indice

Capitolo 1

Modelli discreti

1.1 Equazioni alle differenze lineari del primo ordine

Schema di soluzione

$$a_1 y_{n+1} - a_0 y_n = \overline{b} \quad \longrightarrow \quad y_{n+1} - a y_n = b$$

La soluzione è data da

$$y_n = \overline{y}_n + y_n^\star$$

con

$\overline{y}_n$	soluzione dell'equazione omogenea associata
$y_n^\star$	soluzione particolare dell'equazione completa.

$$\overline{y}_n = ca^n$$

$$y_n^\star = \begin{cases} \dfrac{b}{1-a} & a \neq 1 \\ nb & a = 1 \end{cases}$$

quindi in definitiva

$$y_n = \begin{cases} ca^n + \dfrac{b}{1-a} & a \neq 1 \\ c + nb & a = 1 \end{cases}$$

Se si impone la condizione iniziale y_0

$$y_n = \begin{cases} a^n \left(y_0 - \dfrac{b}{1-a} \right) + \dfrac{b}{1-a} & a \neq 1 \\ y_0 + nb & a = 1. \end{cases}$$

Schema di calcolo equazioni I ordine (per ricorrenza)

1. **input:** $a_1,\ a_0,\ \overline{b},\ n,\ y_0$
2. $a = -\dfrac{a_0}{a_1}$
3. $b = \dfrac{\overline{b}}{a_1}$
4. **per** k **da 0 a n-1**
5. $\quad y_{k+1} = b + a\ y_k$
6. **fine**

Schema di calcolo equazioni I ordine (soluzione generale 1)

1. **input:** $a_1,\ a_0,\ \overline{b},\ n,\ y_0$
2. $a = -\dfrac{a_0}{a_1}$
3. $b = \dfrac{\overline{b}}{a_1}$
4. $\overline{y}_n = y_0\ a^n$
5. **se** $a = 1$ **allora**
6. $\quad y_n^\star = n\ b$
7. **altrimenti**
8. $\quad y_n^\star = \dfrac{b}{1-a}$
9. **fine**
10. $y_n = \overline{y}_n + y_n^\star$

Schema di calcolo equazioni I ordine (soluzione generale 2)

1. **input:** $a_1,\ a_0,\ \overline{b},\ n,\ y_0$
2. $a = -\dfrac{a_0}{a_1}$
3. $b = \dfrac{\overline{b}}{a_1}$
4. **se** $a = 1$ **allora**
5. $\quad y_n = y_0 + n\ b$
6. **altrimenti**
7. $\quad y_n = a^n \left(y_0 - \dfrac{b}{1-a} \right) + \dfrac{b}{1-a}$
8. **fine**

1.2 Equazioni alle differenze lineari a coefficienti costanti del secondo ordine

Schema di soluzione

$$a_2 y_{n+2} + a_1 y_{n+1} + a_0 y_n = b$$

La soluzione è data da

$$y_n = \overline{y}_n + y_n^\star$$

con

$\overline{y}_n$	soluzione dell'equazione omogenea associata
$y_n^\star$	soluzione particolare dell'equazione completa.

Si cercano le soluzioni dell'equazione caratteristica

$$a_2\lambda^2 + a_1\lambda + a_0 = 0$$

$$\Delta = a_1^2 - 4a_0a_2, \qquad \begin{array}{llll} 2 \text{ soluzioni} & \lambda_{1,2} = \frac{-a_1 \pm \sqrt{\Delta}}{2a_2} & \Delta > 0 \\ 1 \text{ soluzione} & \lambda = \frac{-a_1}{2a_2} & \Delta = 0 \end{array}$$

(non consideriamo il caso $\Delta < 0$)

$$\overline{y}_n = \begin{cases} c_1\lambda_1^n + c_2\lambda_2^n & \Delta > 0 \\ c_1\lambda^n + c_2 n\lambda^n & \Delta = 0 \end{cases}$$

$$y_n^\star = \begin{cases} \dfrac{b}{a_2 + a_1 + a_0} & a_2 + a_1 + a_0 \neq 0 \\ \dfrac{b}{2a_2 + a_1} n & a_2 + a_1 + a_0 = 0, \quad 2a_2 + a_1 \neq 0 \\ \dfrac{b}{4a_2 + a_1} n^2 & a_2 + a_1 + a_0 = 0, \quad 2a_2 + a_1 = 0 \end{cases}$$

quindi in definitiva

$$y_n = \begin{cases} c_1\lambda_1^n + c_2\lambda_2^n + \dfrac{b}{a_2 + a_1 + a_0} & \Delta > 0, \quad a_2 + a_1 + a_0 \neq 0 \\ c_1\lambda_1^n + c_2\lambda_2^n + \dfrac{b}{2a_2 + a_1} n & \Delta > 0, \quad a_2 + a_1 + a_0 = 0, \quad 2a_2 + a_1 \neq 0 \\ c_1\lambda_1^n + c_2\lambda_2^n + \dfrac{b}{4a_2 + a_1} n^2 & \Delta > 0, \quad a_2 + a_1 + a_0 = 0, \quad 2a_2 + a_1 = 0 \\ & \\ c_1\lambda^n + c_2 n\lambda^n + \dfrac{b}{a_2 + a_1 + a_0} & \Delta = 0, \quad a_2 + a_1 + a_0 \neq 0 \\ c_1\lambda^n + c_2 n\lambda^n + \dfrac{b}{2a_2 + a_1} n & \Delta = 0, \quad a_2 + a_1 + a_0 = 0, \quad 2a_2 + a_1 \neq 0 \\ c_1\lambda^n + c_2 n\lambda^n + \dfrac{b}{4a_2 + a_1} n^2 & \Delta = 0, \quad a_2 + a_1 + a_0 = 0, \quad 2a_2 + a_1 = 0 \end{cases}$$

Le costanti c_1 e c_2 si calcolano imponendo le condizioni iniziali

Schema di calcolo equazioni II ordine (per ricorrenza)

1. **input:** $a_2,\ a_1,\ a_0,\ b,\ n,\ y_0,\ y_1$
2. **per** k **da 0 a n-1**
3. $$y_{k+2} = \frac{1}{a_2}\Big[b - a_1\ y_{k+1} - a_0\ y_k\Big]$$
4. **fine**

Schema di calcolo equazioni II ordine (soluzione generale 2)

1. **input:** $a_2,\ a_1,\ a_0,\ b,\ n,\ y_0,\ y_1$
2. **se** $a_2 + a_1 + a_0 \neq 0$
3. $$y_n^\star = y_0^\star = y_1^\star = \frac{b}{a_2 + a_1 + a_0}$$
4. **altrimenti se** $2a_2 + a_1 \neq 0$
5. $$y_n^\star = n\,\frac{b}{2a_2 + a_1}, \quad y_0^\star = 0, \quad y_1^\star = \frac{b}{2a_2 + a_1}$$
6. **altrimenti**
7. $$y_n^\star = n^2\,\frac{b}{4a_2 + a_1}, \quad y_0^\star = 0, \quad y_1^\star = \frac{b}{4a_2 + a_1}$$
8. **fine**
9. $$\Delta = a_1^2 - 4a_2a_0$$
10. **se** $\Delta > 0$
11. $$\lambda_1 = \frac{-a_1 + \sqrt{\Delta}}{2\,a_2}$$
12. $$\lambda_2 = \frac{-a_1 - \sqrt{\Delta}}{2\,a_2}$$
13. **si risolve il sistema**
$$\begin{pmatrix} 1 & 1 \\ \lambda_1 & \lambda_2 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} y_0 - y_0^\star \\ y_1 - y_1^\star \end{pmatrix}$$
14. $$y_n = c_1\lambda_1^n + c_2\lambda_2^n + y_n^\star$$
15. **altrimenti se** $\Delta = 0$
16. $$\lambda = \frac{-a_1}{2\,a_2}$$
17. **si risolve il sistema**
$$\begin{pmatrix} 1 & 0 \\ \lambda & \lambda \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} y_0 - y_0^\star \\ y_1 - y_1^\star \end{pmatrix}$$
18. $$y_n = c_1\lambda^n + c_2 n\lambda^n + y_n^\star$$
19. **altrimenti**
20. $\Delta < 0$ radici immaginarie
21.
22. **fine**

Capitolo 2

Metodi numerici

2.1 Metodo di Archimede

Metodo di Archimede con numero di lati assegnato

1. **input:** $nmax$ numero di lati richiesto;
2. **porre** $n = 6$;
3. **porre** $l = 1$;
4. **calcolare** $\pi_i = \dfrac{nl}{2}$;
5. **calcolare** $\pi_c = \dfrac{2\pi_i}{\sqrt{4 - l^2}}$;
6. **ripetere fintanto che** $n < nmax$
   - 6.1 **porre** $l = \dfrac{l}{\sqrt{2 + \sqrt{4 - l^2}}}$;
   - 6.2 **raddoppiare il numero di lati** $n = 2n$;
   - 6.3 **calcolare** $\pi_i = \dfrac{nl}{2}$;
   - 6.4 **calcolare** $\pi_c = \dfrac{2\pi_i}{\sqrt{4 - l^2}}$;
7. **calcolare** $err = \pi_c - \pi_i$;
8. **output:** π_i approssimazione per difetto di π;
   π_c approssimazione per eccesso di π;
   err errore stimato.

Metodo di Archimede con precisione assegnata

1. **input:** ε precisione;
2. **porre** $n = 6$;
3. **porre** $l = 1$;
4. **calcolare** $\pi_i = \dfrac{nl}{2}$;
5. **calcolare** $\pi_c = \dfrac{2\pi_i}{\sqrt{4 - l^2}}$;
6. **ripetere fintanto che** $\pi_c - \pi_i > \varepsilon$
   - 6.1 **porre** $l = \dfrac{l}{\sqrt{2 + \sqrt{4 - l^2}}}$;
   - 6.2 **raddoppiare il numero di lati** $n = 2n$;
   - 6.3 **calcolare** $\pi_i = \dfrac{nl}{2}$;
   - 6.4 **calcolare** $\pi_c = \dfrac{2\pi_i}{\sqrt{4 - l^2}}$;
7. **output:** π_i approssimazione per difetto di π;
   π_c approssimazione per eccesso di π;
   n numero di lati necessari per raggiungere la precisione.

2.2 Calcolo di $\sin x$

Algoritmo A per il calcolo di $\sin t$

1. **input:** t angolo, n tale che 2^n sia il numero di suddivisioni;
2. **calcolare** la corda iniziale $s = \dfrac{2t}{2^n}$;
3. **ripetere** per n volte
   - **3.1. calcolare** la corda relativa all'angolo doppio $s = s\sqrt{4 - s^2}$;
4. **porre** $sent = \dfrac{s}{2}$;
5. **output:** $sent \approx \sin t$.

Algoritmo T per il calcolo di $\sin t$

1. **input:** t angolo, n tale che 3^n sia il numero di suddivisioni;
2. **calcolare** la corda iniziale $s = \dfrac{2t}{3^n}$;
3. **ripetere** per n volte
   - **3.1. calcolare** la corda relativa all'angolo triplo $s = 3s - s^2$;
4. **porre** $sent = \dfrac{s}{2}$;
5. **output:** $sent \approx \sin t$.

Capitolo 3

Calcolo numerico di polinomi

3.1 Algoritmo di Horner

Algoritmo di Horner base

1. **input:** n grado del polinomio, $a_j,\ j = 0, \ldots, n$ coefficienti, z punto in cui valutare il polinomio;
2. **porre** $b_0 = a_0$;
3. **ripetere per** $j = 1, \ldots, n$

   6.1 **calcolare** $b_j = b_{j-1} z + a_j$;
4. **output:** $b_n = P(z)$

Algoritmo di Horner per il calcolo delle derivate

1. **input:** n grado del polinomio, $a_j,\ j = 0, \ldots, n$ coefficienti, z punto in cui valutare il polinomio;
2. **porre** $b_{i,0} = a_0,\ i = 0, \ldots, n$;
3. **ripetere per** $i = 0, \ldots, n-1$

   6.1 **ripetere per** $j = 1, \ldots, n-j$

   6.1.1 **porre** $b_{i,j} = b_{i,j-1} z + b_{i-1,j}$
4. **porre** $p_j = j!\, b_{j,n-j},\ j = 0, \ldots, n$
5. **output:** $p_j = P^{(j)}(z),\ j = 0, \ldots, n$

3.2 Zeri di funzioni

Metodo di bisezione

1. **input:** f funzione, a, b estremi intervallo, ε tolleranza;
2. **porre** $err = \dfrac{b-a}{2}$;
3. **ripetere fintanto che** $err > \varepsilon$
   - 3.1 **porre** $c = \dfrac{a+b}{2}$;
   - 3.2 **se** $f(a)\, f(c) > 0$;
     porre $a = c$;
     altrimenti
     porre $b = c$;
   - 3.3 **calcolare** $err = \dfrac{b-a}{2}$;
4. **output:** $z = \dfrac{a+b}{2}$

Metodo della falsa posizione

1. **input:** f funzione, a, b estremi intervallo, ε tolleranza;
2. **porre** $err = \varepsilon + 1$;
3. **porre** $x_0 = a,\ x_1 = b,\ t = a,\ k = 1$;
4. **ripetere fintanto che** $err > \varepsilon$
   - 4.1 **calcolare** $x_{k+1} = x_k - f(x_k)\, \dfrac{x_k - t}{f(x_k) - f(t)}$;
   - 4.2 **porre** $err = |x_{k+1} - x_k|$;
   - 4.3 **se** $f(x_{k+1})\, f(a) < 0$;
     porre $t = a$;
     altrimenti
     porre $t = b$;
   - 4.5 **porre** $k = k + 1$;
5. **output:** $z = x_k$

Metodo della secante

1. **input:** f funzione, a, b estremi intervallo, ε tolleranza, $itmax$ numero massimo di iterazioni;
2. **porre** $err = \varepsilon + 1$;
3. **porre** $x_0 = a,\ x_1 = b,\ k = 1$;
4. **ripetere fintanto che** $err > \varepsilon$ e $k < itmax$
   4.1 **calcolare** $x_{k+1} = x_k - f(x_k)\,\dfrac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}$;
   4.2 **porre** $err = |x_{k+1} - x_k|$;
   4.3 **porre** $k = k + 1$;
5. **se** $k \geq itmax$;
   output: *precisione non raggiunta*
   altrimenti
   output: $z = x_k$

Metodo di Newton

1. **input:** $f,\ f'$ funzione e derivata prima, x_0 punto iniziale, ε tolleranza, $itmax$ numero massimo di iterazioni;
2. **porre** $err = \varepsilon + 1$;
3. **porre** $k = 0$;
4. **ripetere fintanto che** $err > \varepsilon$ e $k < itmax$
   4.1 **calcolare** $x_{k+1} = x_k - \dfrac{f(x_k)}{f'(x_k)}$;
   4.2 **porre** $err = |x_{k+1} - x_k|$;
   4.3 **porre** $k = k + 1$;
5. **se** $k \geq itmax$;
   output: *precisione non raggiunta*
   altrimenti
   output: $z = x_k$