Alcuni metodi iterativi per la ricerca di radici di funzioni

Gionata Massi

1 Il problema

Data una funzione $f : \mathbb{R} \to \mathbb{R}$, determinare un valore reale α tale che $f(\alpha) = 0$.

Usualmente consideriamo funzioni continue in $\mathbb{R}$ o almeno in un intevallo $[a, b] \subseteq \mathbb{R}$ chiuso e limitato in cui ricercare una radice.

2 Esistenza delle radici

Non tutte le funzioni ammettono radici, ad esempio $x \mapsto k$ e $x \mapsto (x + k)^2$, dove $k \neq 0$ (es: fig. 1).

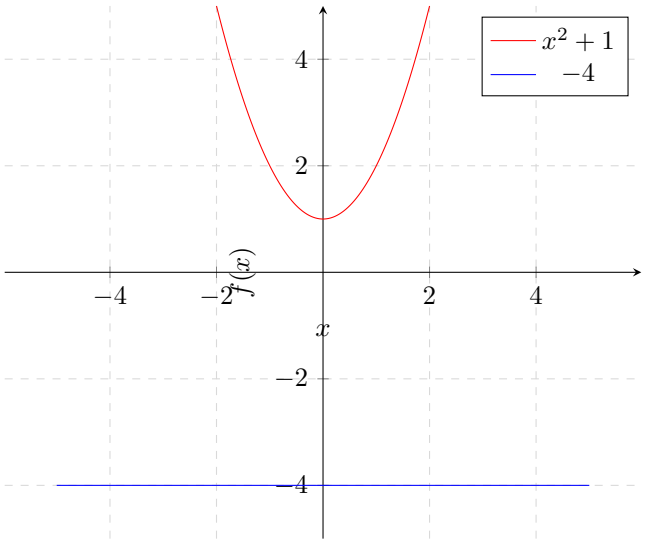


Figura 1: Funzioni che non intersecano l'asse $y = 0$

Altre funzioni hanno radici nei punti di massimo o di minimo locale (es: fig. 2).

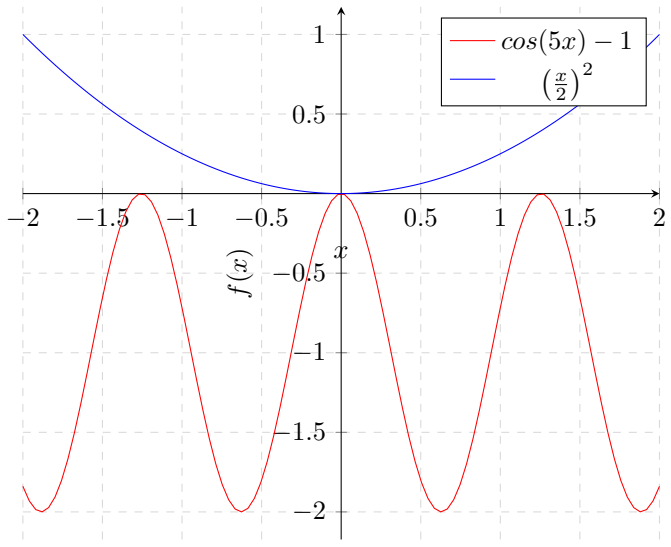


Figura 2: Funzioni che intersecano l'asse $y = 0$ in un estremante

Per essere sicuri che una funzione ammetta almeno una radice richiediamo che la funzione assuma valori positivi e negativi in un certo intervallo e che sia continua.

Teorema 1 (Bolzano) *Se $f(x)$ è una funzione continua sull'intervallo limitato e chiuso $[a, b]$ e $f(a) \cdot f(b) < 0$, allora esiste almeno una radice di $f(x)$ nell'intervallo $[a, b]$.*

Se le ipotesi del teorema sono vere può esistere una sola radice oppure ce ne possono essere in numero finito o anche infinite (fig. 3).

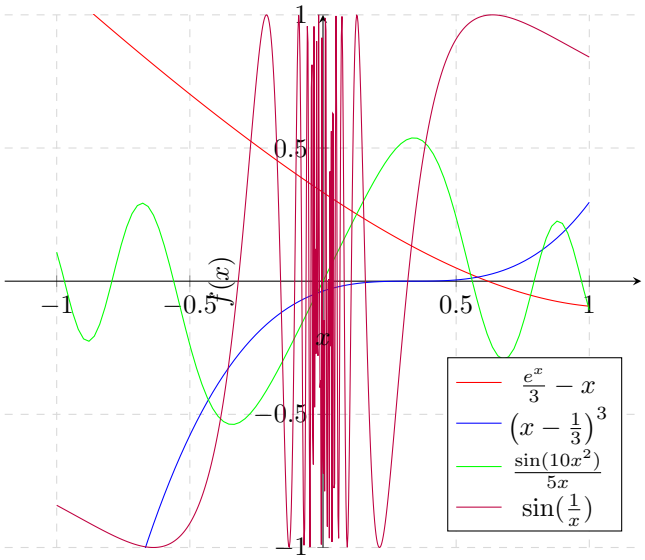


Figura 3: Funzioni che assumono valori opposti agli estremi -1, 1

Un metodo di ricerca delle radici, se convergente, restituirà una sola delle radici. Si intuisce che maggiore è la pendenza della funzione in un intorno della radice, più è facile discriminare la radice. Se invece la pendenza è nulla o quasi, allora il problema si dice mal condizionato.

3 Metodi diretti e metodi iterativi

I **metodi diretti** sono algoritmi che, in assenza di errori di arrotondamento, forniscono la soluzione in un numero finito di operazioni.

I **metodi iterativi** sono algoritmi nei quali la soluzione è ottenuta come limite di una successione di soluzioni. Nella risposta fornita da un metodo iterativo è, quindi, presente usualmente un errore di troncamento.

4 Metodo di bisezione

Sia $f(x)$ una funzione continua sull'intervallo limitato e chiuso $[a, b]$ con $f(a) \cdot f(b) < 0$. L'algoritmo genera una successione di intervalli $[a_k, b_k]$ con $f(a_k) \cdot f(b_k) < 0$ e con $[a_k, b_k] \subset [a_{k-1}\ ,\ b_{k-1}]$ e $|b_k - a_k| = \frac{1}{2}|b_{k-1} - a_{k-1}|$.

Date due tolleranze ϵ_1 , ϵ_2 , l'algoritmo si arresta o quando $|b_k - a_k| \leq \epsilon_1$ o quando $|f(\frac{a_k+b_k}{2})| \leq \epsilon_2$ o infine quando $k >$ nmax, ove nmax è un numero massimo di iterazioni fissato.

Per alleggerire la notazione usiamo s_{a_k} per indicare segno $(f\,(a_k))$, s_{b_k} per segno$(f(b_k))$ e s_k per segno$(f(\frac{a_k+b_k}{2}))$, dove

$$\text{segno}(x) = \begin{cases} -1 & \text{se } x < 0, \\ 0 & \text{se } x = 0, \\ +1 & \text{se } x > 0 \end{cases} \tag{1}$$

Per il calcolo di $\frac{a_k+b_k}{2}$ in virgola si deve usare la formula: $a_k + (b_k - a_k)/2$ in modo da ridurre gli errori di troncamento.

Si vedano il listato 1 e gli esempi nelle tab. .

5 Iterazioni di punto fisso

In generale si può costruire un metodo iterativo cercando un punto fisso di una funzione $\Phi(x)$, costruita in moda che si annulli ne punto desiderato, un valore $\bar{x}$ tale che $\Phi(\bar{x}) = \bar{x}$.

Il punto fisso è calcolato tramite l'applicazione ripetuta della regola di ricorrenza:

$$x_{k+1} = \Phi(x_k)$$

.

```javascript
/**
 *
 * @param {Function} f una funzione continua in [a, b]
 * @param {Number} a l'estremo sinistro dell'intervallo di incertezza
 * @param {Number} b l'estremo destro dell'intervallo di incertezza
 * @param {Number} e1 tolleranza su asse delle ascisse
 * @param {Number} e2 tolleranza su asse delle ordinate
 * @param {Number} nmax numero massimo di iterazioni
 */
const bisezione = (f, a, b, e1 = 1e-16, e2 = 1e-16, nmax = 10) => {
  let f_a = f(a);
  let f_b = f(b);
  let s_a = Math.sign(f_a);
  let s_b = Math.sign(f_b);
  if (s_a === s_b) {
    throw "Segni concordi nei due estremi.";
  }
  let x;
  for (let iter = 0; iter < nmax && b - a >= e1; iter++) {
    x = a + (b - a) / 2;
    let f_x = f(x);
    if (Math.abs(f_x) < e2) {
      return x;
    }
    let s_x = Math.sign(f_x);
    if (s_a === s_x) {
      a = x;
    } else {
      b = x;
    }
  }
  return x;
};
```

Codice sorgente 1: Descrizione in JavaScript del metodo di Bisezione

k	a_k	b_k	x_k	s_{a_k}	s_{b_k}	s_k	$\lvert b_k - a_k \rvert$	f_k
0	0	6	3	−1	1	1	6	3
1	0	3	1.5	−1	1	−1	3	−3.75
2	1.5	3	2.25	−1	1	−1	1.5	−0.9375
3	2.25	3	2.625	−1	1	1	0.75	0.890625
4	2.25	2.625	2.4375	−1	1	−1	0.375	−0.05859375
5	2.4375	2.625	2.53125	−1	1	1	0.1875	0.4072265625
6	2.4375	2.53125	2.484375	−1	1	1	0.09375	0.1721191406
7	2.4375	2.484375	2.4609375	−1	1	1	0.046875	0.0562133789
8	2.4375	2.4609375	2.44921875	−1	1	−1	0.0234375	−0.0013275146
9	2.44921875	2.4609375	2.455078125	−1	1	1	0.01171875	0.0274085999

Tabella 1: Metodo dicotomico applicato a $x^2 - 6$ nell'intervallo $[0, 6]$ con nmax $= 10$

k	a_k	b_k	x_k	s_{a_k}	s_{b_k}	s_k	$\lvert b_k - a_k \rvert$	f_k
0	3	3.2	3.1	1	−1	1	0.2	0.0415806624
1	3.1	3.2	3.15	1	−1	−1	0.1	−0.0084072474
2	3.1	3.15	3.125	1	−1	1	0.05	0.0165918922
3	3.125	3.15	3.1375	1	−1	1	0.025	0.0040926422
4	3.1375	3.15	3.14375	1	−1	−1	0.0125	−0.0021573447
5	3.1375	3.14375	3.140625	1	−1	1	0.00625	0.0009676534
6	3.140625	3.14375	3.1421875	1	−1	−1	0.003125	−0.0005948464
7	3.140625	3.1421875	3.14140625	1	−1	1	0.0015625	0.0001864036
8	3.14140625	3.1421875	3.141796875	1	−1	−1	0.00078125	−0.0002042214
9	3.14140625	3.141796875	3.1416015625	1	−1	−1	0.000390625	$-8.9089102066 \cdot 10^{-6}$

Tabella 2: Metodo dicotomico applicato a $sin(x)$ nell'intervallo $[3, 3.2]$ con nmax $= 10$

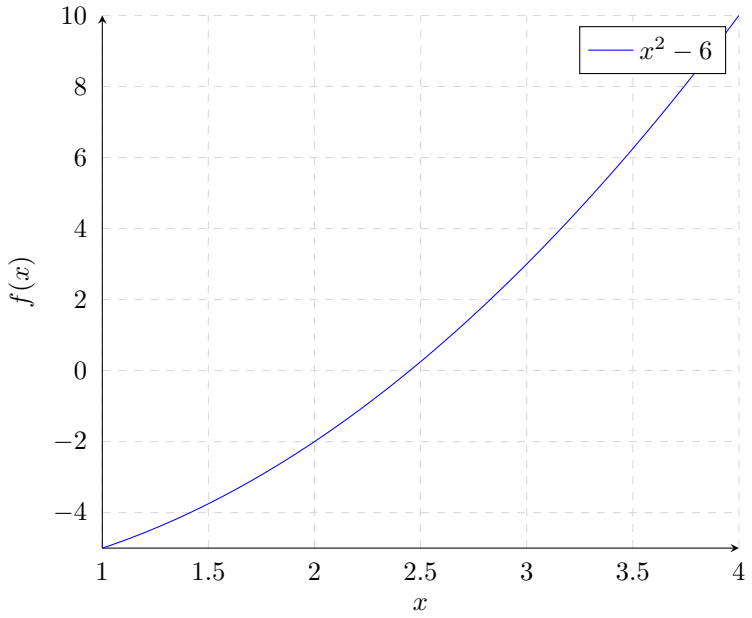


Figura 4: Funzioni che intersecano l'asse $y = 0$ in un estremante

k	a_k	b_k	x_k	s_{a_k}	s_{b_k}	s_k	$\lvert b_k - a_k \rvert$	f_k
0	0	1	0.5	1	−1	1	1	0.0452392119
1	0.5	1	0.75	1	−1	−1	0.5	−0.1264750837
2	0.5	0.75	0.625	1	−1	−1	0.25	−0.0394838005
3	0.5	0.625	0.5625	1	−1	1	0.125	0.0031482702
4	0.5625	0.625	0.59375	1	−1	−1	0.0625	−0.0180982778
5	0.5625	0.59375	0.578125	1	−1	−1	0.03125	−0.0074578482
6	0.5625	0.578125	0.5703125	1	−1	−1	0.015625	−0.0021505284
7	0.5625	0.5703125	0.56640625	1	−1	1	0.0078125	0.0004999324
8	0.56640625	0.5703125	0.568359375	1	−1	−1	0.00390625	−0.0008250321
9	0.56640625	0.568359375	0.5673828125	1	−1	−1	0.001953125	−0.0001624834

Tabella 3: Metodo dicotomico applicato a $e^{e^{-x}} - x$ nell'intervallo $[0, 1]$ con nmax = 10

5.1 Approssimazioni con rette

Una retta è definita da una funzione del tipo $f(x) = mx + q$. Se imponiamo il passaggio per il punto $(x_k, f(x_k))$ della funzione di cui cerchiamo una radice, il fascio di rette sarà:

$$f(x) - f(x_k) = m(x - x_k)$$

.

Possiamo generare delle iterazioni andando a fissare il coefficiente angolare ad ogni iterazione e determinando l'intersezione della retta con l'asse delle ascisse.

$$f(x_{k+1}) - f(x_k) = m_k(x_{k+1} - x_k)$$

Risolvendo per

$$f(x_{k+1}) = 0$$

si ha

$$x_{k+1} = x_k - \frac{f(x_k)}{m_k} \tag{2}$$

5.1.1 Metodo delle corde

Si considerino due punti $a = x_0$ e $b = x_1$ tali da soddisfare le ipotesi del teorema di Bolzano. È possibile costruire una successione che per ogni $k \geq 0$ il punto x_{k+1} sia lo zero della retta passante per il punto $(x_k,\ f(x_k))$ e di coefficiente angolare

$$m_k = \frac{f(a) - f(x_k)}{a - x_k}.$$

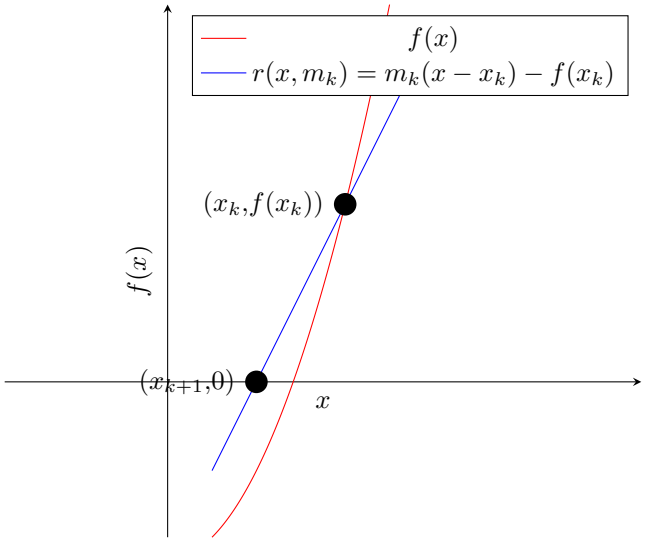


Figura 5: Approssimazione di una funzione con una retta passante per $(x_k,\ f(x_k))$

k	x_k	f_k	f'_k	$\lvert x_{k+1} - x_k \rvert$
0	3	3	6	0.5
1	2.5	0.25	5	0.05
2	2.45	0.0025	4.9	0.0005102041
3	2.4494897959	$2.6030820521 \cdot 10^{-7}$	4.8989795918	$5.3135188693 \cdot 10^{-8}$
4	2.4494897428	$3.5527136788 \cdot 10^{-15}$	4.8989794856	$8.881784197 \cdot 10^{-16}$
5	2.4494897428	$-8.881784197 \cdot 10^{-16}$	4.8989794856	0

Tabella 4: Metodo delle tangenti applicato a $x^2 - 6$ con stima iniziale 3 e nmax = 10

L'iterata ha equazione:

$$x_{k+1} = x_k - f(x_k)\frac{a - x_k)}{f(a) - f(x_k)}.$$

5.1.2 Metodo delle secanti

Dati due punti iniziali x_0 e x_1, si considera la secante passante per i due punti dati.

In generale il coefficiente angolare m_k è calcolato come:

$$m_k = \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}.$$

L'iterata è

$$x_{k+1} = x_k - f(x_k) \cdot \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}.$$

5.1.3 Metodo delle tangenti

Si approssima la funzione $f(x)$ con la retta $r(x) - f(x_k) = f'(x_k)(x - x_k)$ tangente ad essa in $(x_k, f(x_k))$.

L'iterata assume la forma:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'x_k}.$$

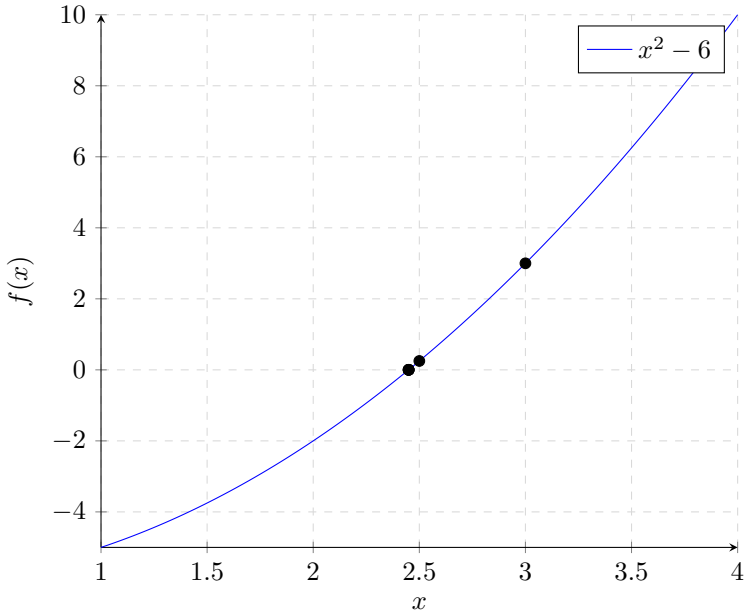
Si vedano il listato 2 e gli esempi.

```
/**
 *
 * @param {Function} f una funzione continua e derivabile in un intorno di x
 * @param {Number} x il valore x_0
 * @param {Function} f1 la derivata prima di f
 * @param {Number} e1 tolleranza su asse delle ascisse
 * @param {Number} e2 tolleranza su asse delle ordinate
 * @param {Number} nmax numero massimo di iterazioni
 */
const tangenti = (f, x, f1, e1 = 1e-16, e2 = 1e-16, nmax = 10) => {
  let err = e1 + 1; // permette di entrare nel ciclo
  for (let iter = 0; iter < nmax && err >= e1; iter++) {
    let f_x = f(x);
    if (Math.abs(f_x) < e2) {
      return x;
    }
    let f1_x = f1(x);
    let xp = x;
    x = xp - f_x / f1_x;
    err = Math.abs(x - xp);
  }
  return x;
};
```

Codice sorgente 2: Descrizione in JavaScript del metodo delle tangenti

k	x_k	f_k	f'_k	$\lvert x_{k+1} - x_k \rvert$
0	3.1	0.0415806624	-0.9991351503	0.0416166546
1	3.1416166546	-0.000024001	-0.9999999997	0.000024001
2	3.1415926536	$4.5633567784 \cdot 10^{-15}$	-1	$4.4408920985 \cdot 10^{-15}$
3	3.1415926536	$1.2246467991 \cdot 10^{-16}$	-1	0

Tabella 5: Metodo delle tangenti applicato a $sin(x)$ con stima iniziale 3,1 e nmax = 10



k	x_k	f_k	f'_k	$\lvert x_{k+1} - x_k \rvert$
0	0.5	0.0452392119	-0.6692957011	0.0675922643
1	0.5675922643	-0.0003045748	-0.6784110106	0.0004489532
2	0.5671433111	$-1.4035108742 \cdot 10^{-8}$	-0.678348491	$2.0690115621 \cdot 10^{-8}$
3	0.5671432904	$-1.1102230246 \cdot 10^{-16}$	-0.6783484881	$1.1102230246 \cdot 10^{-16}$
4	0.5671432904	0	-0.6783484881	0

Tabella 6: Metodo delle tangenti applicato a $e^{e^{-x}} - x$ con stima iniziale 0,5 e nmax = 10

5.1.4 Esempio: algoritmo del reciproco

Si vuole cercare il reciproco del valore ν come $\alpha = \frac{1}{\nu}$ con il metodo delle tangenti.

Per prima cosa occorre trasformare il problema con una funzione che si annulla in $\frac{1}{\nu}$.

Scegliamo

$$f(x) = \nu - \frac{1}{x}$$

che applicata al reciproco di ν produce $f(\frac{1}{\nu}) = \nu - \frac{1}{\frac{1}{\nu}} = \nu - \nu = 0$.

La derivata prima assume la forma $f'(x) = \frac{1}{x^2}$ e

$$\frac{f(x)}{f'(x)} = \frac{\nu - \frac{1}{x}}{\frac{1}{x^2}} = \nu x^2 - x.$$

L'iterata del metodo delle tangenti è:

$$x_{k+1} = x_k - (\nu x_k^2 - x_k) = 2x_k - \nu x_k^2 = x_k \cdot (2 - \nu \cdot x_k).$$

Si noti che per calcolare il reciproco di un numero sono sufficienti le operazioni di moltiplicazione e sottrazione. Per la moltiplicazione occorrono le operazioni primitive di scorrimento e addizione e per la sottrazione quelle di negazione bit a bit e di addizione (basterebbe il solo incremento unitario).

Nota: queste proprietà permettono di realizzare le CPU senza l'operazione di divisione.

5.2 Approssimazioni con parabole

5.2.1 Metodo di Newton del secondo ordine

$$p(x) = ax^2 + bx + c \tag{3}$$