

Alcuni metodi iterativi per la ricerca di radici di funzioni

Gionata Massi

Indice

1	Il problema	1
2	Esistenza delle radici	2
2.1	Teorema degli zeri	2
3	Metodi numerici	3
3.1	Metodi diretti e metodi iterativi	3
3.2	Ordine di convergenza	3
3.3	Criteri di arresto	3
4	Metodo di bisezione	4
4.1	Convergenza	4
4.2	Codifica in JavaScript	4
4.3	Esempi	4
5	Iterazioni di punto fisso	6
5.1	Approssimazioni con rette	6
5.2	Metodo delle corde	7
5.2.1	Convergenza	7
5.2.2	Codifica in JavaScript	7
5.2.3	Esempi	7
5.3	Metodo delle secanti	10
5.3.1	Convergenza	10
5.3.2	Codifica in JavaScript	10
5.3.3	Esempi	10
5.4	Metodo delle tangenti	10
5.4.1	Convergenza	10
5.4.2	Codifica in JavaScript	12
5.4.3	Esempi	12
5.4.4	Esempio: algoritmo del reciproco	12
5.5	Altri metodi	14
5.5.1	Metodo di Steffensen	14
5.5.2	Metodo di Muller	15

1 Il problema

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

Usualmente consideriamo funzioni continue in \mathbb{R} o almeno in un intervallo $[a, b] \subseteq \mathbb{R}$ 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).

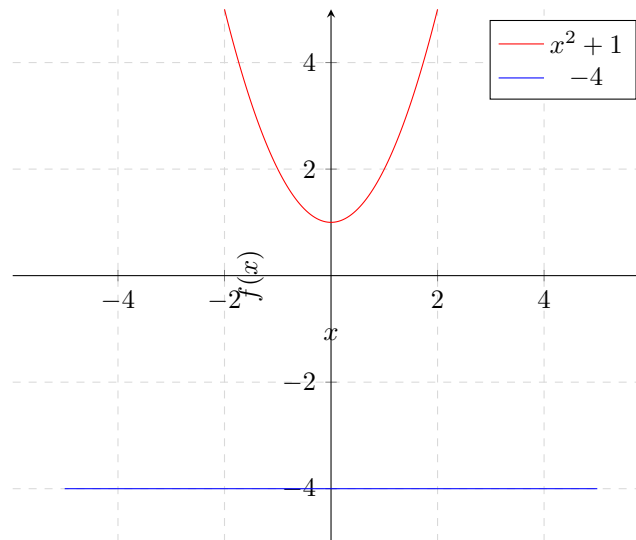


Figura 1: Funzioni che non intersecano l'asse $y = 0$

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

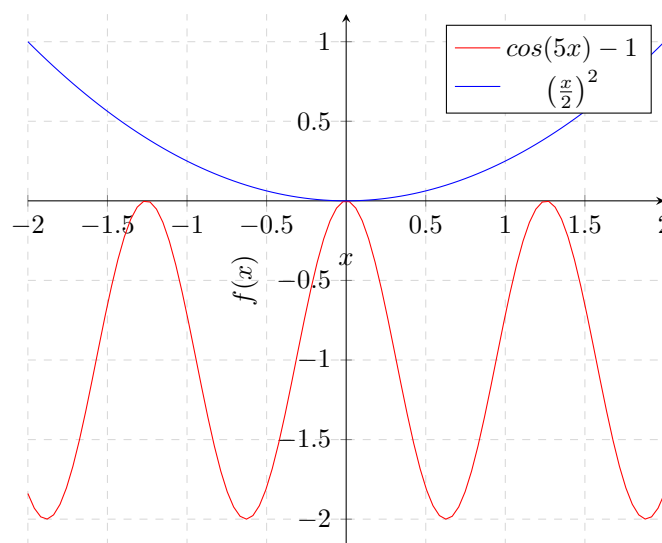


Figura 2: Funzioni che intersecano l'asse $y = 0$ in un estremo

2.1 Teorema degli zeri

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 2.1 (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).

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.

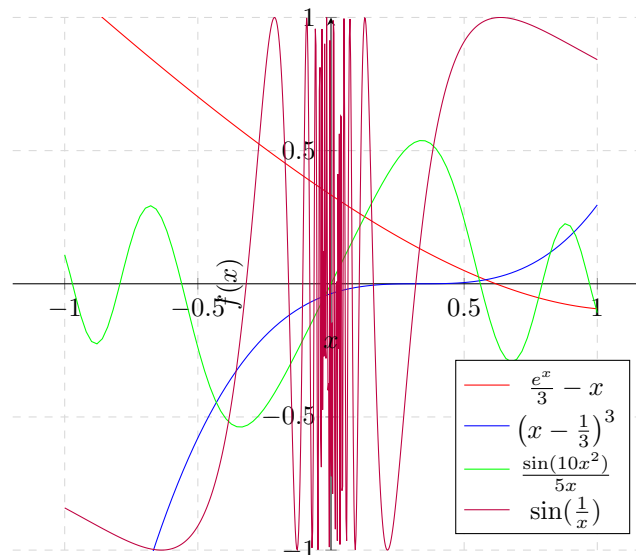


Figura 3: Funzioni che assumono valori opposti agli estremi -1, 1

3 Metodi numerici

3.1 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.

3.2 Ordine di convergenza

Sia $x_k, k = 0, 1, \dots$ una successione convergente al valore α .

Definizione 3.1 (Errore all'iterazione k -esima). L'errore per elemento della successione in posizione k è il valore non negativo $\varepsilon_k = |x_k - \alpha|$.

Definizione 3.2 (Ordine di convergenza). Se esistono un numero reale $p \geq 1$ e una costante reale positiva C tale che $\lim_{k \rightarrow +\infty} \frac{\varepsilon_{k+1}}{\varepsilon_k^p} = C$ allora la successione $x_k, k = 0, 1, \dots$ ha ordine di convergenza p e costante d'errore C .

Si noti che nella definizione di ordine di convergenza, la costante C è positiva, ossia strettamente maggiore di zero.

3.3 Criteri di arresto

I metodi iterativi sviluppano una successione di valori che deve essere troncata per produrre un risultato. Si possono usare vari criteri per arrestare un metodo numerico in base a vincoli sull'approssimazione della soluzione cercata e sulle risorse di calcolo (essenzialmente il tempo di calcolo) da impiegare. I criteri di arresto comunemente adottati sono i seguenti:

Tolleranza sull'approssimazione della funzione $|f(x_k)| < \tau_y$.

Tolleranza assoluta sull'approssimazione di α $|x_{k+1} - x_k| < \tau_x$.

Tolleranza relativa sull'approssimazione di α $\frac{|x_{k+1} - x_k|}{|x_{k+1}|} < \tau_r$.

Numero di iterazioni $k < k_M$.

k	a_k	b_k	x_k	s_{a_k}	s_{b_k}	s_k	$ b_k - a_k $	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 $n_{\max} = 10$

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 $k > n_{\max}$, ove n_{\max} 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

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

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.

4.1 Convergenza

Il metodo è sempre convergente e si può calcolare l'ordine considerando l'errore come l'ampiezza dell'intervallo di incertezza: $\varepsilon_k = \frac{|b-a|}{2^k}$. Il rapporto tra due errori successivi per $p = 1$ vale

$$\frac{\varepsilon_{k+1}}{\varepsilon_k^p} = \frac{\frac{|b-a|}{2^{k+1}}}{\frac{|b-a|}{2^k}} = \frac{1}{2} =$$

L'ordine di convergenza è uno con costante d'errore vale un mezzo.

4.2 Codifica in JavaScript

Si veda il listato 1.

4.3 Esempi

Si vedano le tabb. 1, 2 e 3 e la fig. 4.

```

1  /**
2  *
3  * @param {Function} f una funzione continua in [a, b]
4  * @param {Number} a l'estremo sinistro dell'intervallo di incertezza
5  * @param {Number} b l'estremo destro dell'intervallo di incertezza
6  * @param {Number} e1 tolleranza su asse delle ascisse
7  * @param {Number} e2 tolleranza su asse delle ordinate
8  * @param {Number} nmax numero massimo di iterazioni
9  */
10 const bisezione = (f, a, b, e1 = 1e-16, e2 = 1e-16, nmax = 10) => {
11   let f_a = f(a);
12   let f_b = f(b);
13   let s_a = Math.sign(f_a);
14   let s_b = Math.sign(f_b);
15   if (s_a === s_b) {
16     throw "Segni concordi nei due estremi.";
17   }
18   let x;
19   for (let iter = 0; iter < nmax && b - a >= e1; iter++) {
20     x = a + (b - a) / 2;
21     let f_x = f(x);
22     if (Math.abs(f_x) < e2) {
23       return x;
24     }
25     let s_x = Math.sign(f_x);
26     if (s_a === s_x) {
27       a = x;
28     } else {
29       b = x;
30     }
31   }
32   return x;
33 };

```

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	$ b_k - a_k $	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 $n_{\max} = 10$

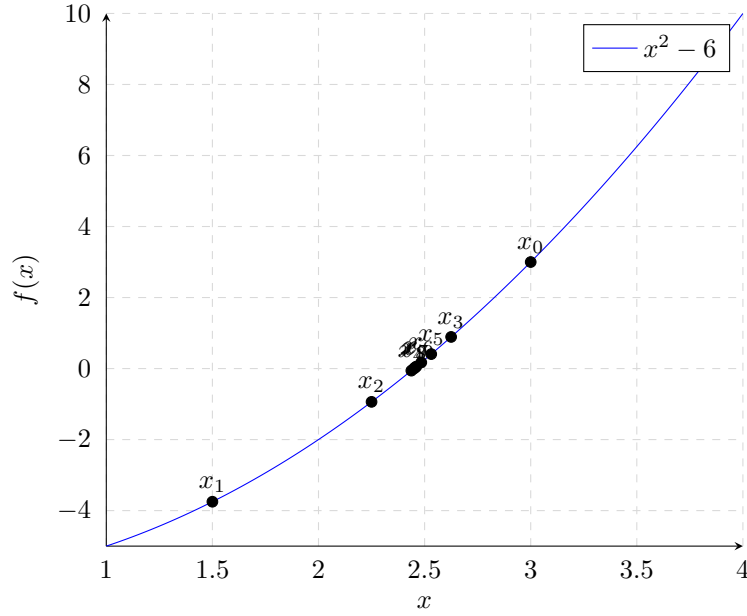


Figura 4: Successione delle soluzioni del metodo dicotomico applicato a $x^2 - 6$ nell'intervallo $[0, 6]$

k	a_k	b_k	x_k	s_{a_k}	s_{b_k}	s_k	$ b_k - a_k $	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 $n_{\max} = 10$

5 Iterazioni di punto fisso

In generale si può costruire un metodo iterativo cercando un punto fisso di una funzione $\Phi(x)$, costruita in modo che si annulli nel 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:

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

5.1 Approssimazioni con rette

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

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

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

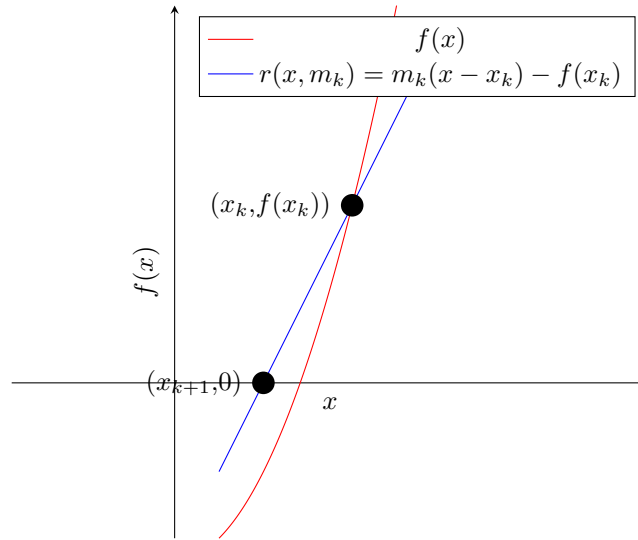


Figura 5: Approssimazione di una funzione con una retta passante per $(x_k, f(x_k))$

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

Risolvendo per

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

si ha

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

5.2 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

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

L'iterata ha equazione:

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

5.2.1 Convergenza

Il metodo non necessariamente converge. Può oscillare (vedi tab. 7) o divergere.

5.2.2 Codifica in JavaScript

Si veda il listato 2.

5.2.3 Esempi

Si vedano le tabb. 4, 5 e 6 e la fig. 6.

```

1  /**
2  *
3  * @param {Function} f una funzione continua in un intorno di x
4  * @param {Number} a una stima iniziale della radice
5  * @param {Number} x una seconda stima della radice
6  * @param {Number} e1 tolleranza su asse delle ascisse
7  * @param {Number} e2 tolleranza su asse delle ordinate
8  * @param {Number} nmax numero massimo di iterazioni
9  */
10 const corde = (f, a, x, e1 = 1e-16, e2 = 1e-16, nmax = 10) => {
11   const f_a = f(a);
12   let err = e1 + 1; // permette di entrare nel ciclo
13   for (let iter = 0; iter < nmax && err >= e1; iter++) {
14     let f_x = f(x);
15     let inv_m_k = (a - x) / (f_a - f_x);
16     let xp = x;
17     x = xp - f_x * inv_m_k;
18     err = Math.abs(x - xp);
19     if (Math.abs(f_x) < e2) {
20       return x;
21     }
22   }
23   return x;
24 };

```

Codice sorgente 2: Descrizione in JavaScript del metodo delle corde

k	a_k	b_k	x_k	s_{a_k}	s_{b_k}	s_k	$ b_k - a_k $	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 4: Metodo delle corde applicato a $x^2 - 6$ nell'intervallo $[0, 6]$ con $nmax = 10$

k	x_k	f_k	m_k	$ x_{k+1} - x_k $
0	3.1	0.0415806624	-0.9995480586	0.041599463
1	3.141599463	$-6.8093626611 \cdot 10^{-6}$	-0.999431461	$6.8132362516 \cdot 10^{-6}$
2	3.1415926497	$3.8735904598 \cdot 10^{-9}$	-0.9994315273	$3.875793908 \cdot 10^{-9}$
3	3.1415926536	$-2.2034481946 \cdot 10^{-12}$	-0.9994315273	$2.2049029269 \cdot 10^{-12}$
4	3.1415926536	$1.4547323095 \cdot 10^{-15}$	-0.9994315273	$1.3322676296 \cdot 10^{-15}$
5	3.1415926536	$1.2246467991 \cdot 10^{-16}$	-0.9994315273	0

Tabella 5: Metodo delle corde applicato a $\sin(x)$ nell'intervallo $[3, 3.2]$ con $nmax = 10$

k	x_k	f_k	m_k	$ x_{k+1} - x_k $
0	0.5	0.0452392119	-0.7060771687	0.0640712006
1	0.5640712006	0.0020832911	-0.7108561397	0.0029306789
2	0.5670018795	0.0000959245	-0.7110776753	0.0001349002
3	0.5671367796	$4.4165623789 \cdot 10^{-6}$	-0.7110878784	$6.2109937645 \cdot 10^{-6}$
4	0.5671429906	$2.0334709361 \cdot 10^{-7}$	-0.7110883482	$2.8596600421 \cdot 10^{-7}$
5	0.5671432766	$9.3624932251 \cdot 10^{-9}$	-0.7110883698	$1.3166427171 \cdot 10^{-8}$
6	0.5671432898	$4.3106729297 \cdot 10^{-10}$	-0.7110883708	$6.0620775066 \cdot 10^{-10}$
7	0.5671432904	$1.9847234967 \cdot 10^{-11}$	-0.7110883709	$2.7911117861 \cdot 10^{-11}$
8	0.5671432904	$9.1371354927 \cdot 10^{-13}$	-0.7110883709	$1.2849721287 \cdot 10^{-12}$
9	0.5671432904	$4.2077452633 \cdot 10^{-14}$	-0.7110883709	$5.9174887213 \cdot 10^{-14}$

Tabella 6: Metodo delle corde applicato a $e^{e^{-x}} - x$ nell'intervallo $[0, 1]$ con $n_{\max} = 10$

k	x_k	f_k	m_k	$ x_{k+1} - x_k $
0	2	3	2	1.5
1	0.5	-0.75	0.5	1.5
2	2	3	2	1.5
3	0.5	-0.75	0.5	1.5
4	2	3	2	1.5
5	0.5	-0.75	0.5	1.5
6	2	3	2	1.5
7	0.5	-0.75	0.5	1.5
8	2	3	2	1.5
9	0.5	-0.75	0.5	1.5

Tabella 7: Metodo delle corde applicato a $x^2 - 1$ nell'intervallo $[0, 2]$ con $n_{\max} = 10$

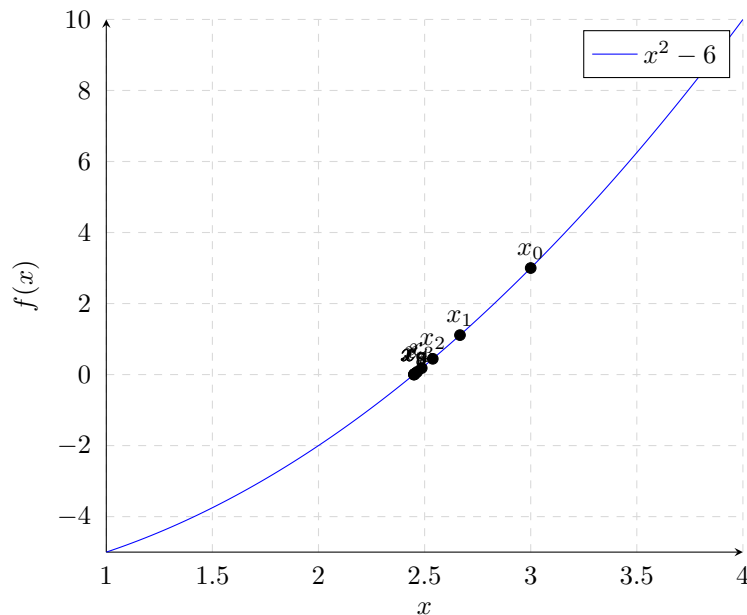


Figura 6: Successione delle soluzioni del metodo delle corde applicato a $x^2 - 6$ nell'intervallo $[0, 6]$

5.3 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

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

5.3.1 Convergenza

Il metodo non necessariamente converge. Può oscillare o divergere.

5.3.2 Codifica in JavaScript

5.3.3 Esempi

Si vedano gli esempi nelle tabb. 8, 9 e 10 e la fig. 7.

5.4 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:

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

5.4.1 Convergenza

Il metodo non necessariamente converge. Può oscillare o divergere.

```

1  /**
2  *
3  * @param {Function} f una funzione continua in un intorno di x
4  * @param {Number} a una stima iniziale della radice
5  * @param {Number} x una seconda stima della radice
6  * @param {Number} e1 tolleranza su asse delle ascisse
7  * @param {Number} e2 tolleranza su asse delle ordinate
8  * @param {Number} nmax numero massimo di iterazioni
9  */
10 const secanti = (f, a, x, e1 = 1e-16, e2 = 1e-16, nmax = 10) => {
11   let f_a = f(a);
12   let err = e1 + 1; // permette di entrare nel ciclo
13   for (let iter = 0; iter < nmax && err >= e1; iter++) {
14     let f_x = f(x);
15     let inv_m_k = (a - x) / (f_a - f_x);
16     let xp = x;
17     a = x;
18     f_a = f_x;
19     x = xp - f_x * inv_m_k;
20     err = Math.abs(x - xp);
21     if (Math.abs(f_x) < e2) {
22       return x;
23     }
24   }
25   return x;
26 };

```

Codice sorgente 3: Descrizione in JavaScript del metodo delle secanti

k	a_k	b_k	x_k	s_{a_k}	s_{b_k}	s_k	$ b_k - a_k $	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 8: Metodo delle secanti applicato a $x^2 - 6$ nell'intervallo $[0, 6]$ con $n_{\max} = 10$

k	x_k	f_k	m_k	$ x_{k+1} - x_k $
0	3.1	0.0415806624	-0.9995480586	0.041599463
1	3.141599463	$-6.8093626611 \cdot 10^{-6}$	-0.999431461	$6.8132362516 \cdot 10^{-6}$
2	3.1415926497	$3.8735904598 \cdot 10^{-9}$	-0.9994315273	$3.875793908 \cdot 10^{-9}$
3	3.1415926536	$-2.2034481946 \cdot 10^{-12}$	-0.9994315273	$2.2049029269 \cdot 10^{-12}$
4	3.1415926536	$1.4547323095 \cdot 10^{-15}$	-0.9994315273	$1.3322676296 \cdot 10^{-15}$
5	3.1415926536	$1.2246467991 \cdot 10^{-16}$	-0.9994315273	0

Tabella 9: Metodo delle secanti applicato a $\sin(x)$ nell'intervallo $[3, 3.2]$ con $n_{\max} = 10$

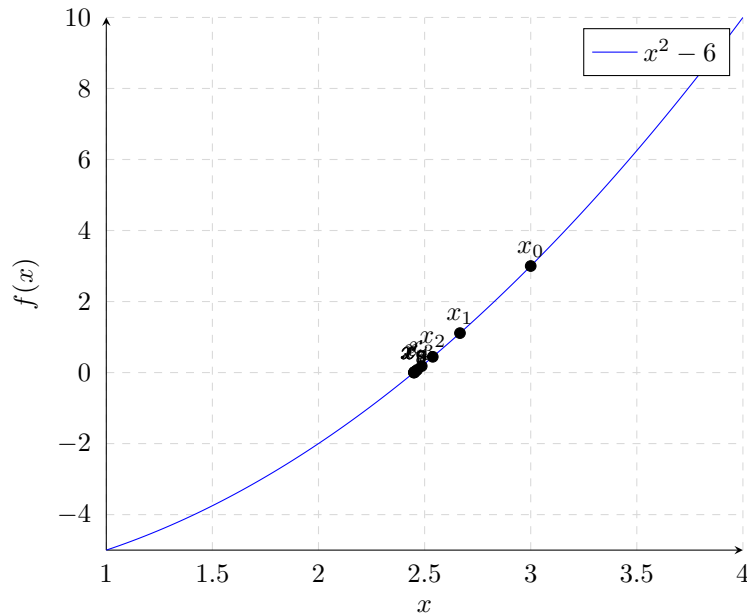


Figura 7: Successione delle soluzioni del metodo delle secanti applicato a $x^2 - 6$ nell'intervallo $[0, 6]$

k	x_k	f_k	m_k	$ x_{k+1} - x_k $
0	0.5	0.0452392119	-0.7060771687	0.0640712006
1	0.5640712006	0.0020832911	-0.7108561397	0.0029306789
2	0.5670018795	0.0000959245	-0.7110776753	0.0001349002
3	0.5671367796	$4.4165623789 \cdot 10^{-6}$	-0.7110878784	$6.2109937645 \cdot 10^{-6}$
4	0.5671429906	$2.0334709361 \cdot 10^{-7}$	-0.7110883482	$2.8596600421 \cdot 10^{-7}$
5	0.5671432766	$9.3624932251 \cdot 10^{-9}$	-0.7110883698	$1.3166427171 \cdot 10^{-8}$
6	0.5671432898	$4.3106729297 \cdot 10^{-10}$	-0.7110883708	$6.0620775066 \cdot 10^{-10}$
7	0.5671432904	$1.9847234967 \cdot 10^{-11}$	-0.7110883709	$2.7911117861 \cdot 10^{-11}$
8	0.5671432904	$9.1371354927 \cdot 10^{-13}$	-0.7110883709	$1.2849721287 \cdot 10^{-12}$
9	0.5671432904	$4.2077452633 \cdot 10^{-14}$	-0.7110883709	$5.9174887213 \cdot 10^{-14}$

Tabella 10: Metodo delle secanti applicato a $e^{e^{-x}} - x$ nell'intervallo $[0, 1]$ con $n_{\max} = 10$

5.4.2 Codifica in JavaScript

Si vedanil listato 4.

5.4.3 Esempi

Si vedano le tabb. ??, ?? e ?? e la fig. 8.

5.4.4 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

```

1  /**
2   *
3   * @param {Function} f una funzione continua e derivabile in un intorno di x
4   * @param {Number} x il valore x_0
5   * @param {Function} f1 la derivata prima di f
6   * @param {Number} e1 tolleranza su asse delle ascisse
7   * @param {Number} e2 tolleranza su asse delle ordinate
8   * @param {Number} nmax numero massimo di iterazioni
9   */
10 const tangenti = (f, x, f1, e1 = 1e-16, e2 = 1e-16, nmax = 10) => {
11   let err = e1 + 1; // permette di entrare nel ciclo
12   for (let iter = 0; iter < nmax && err >= e1; iter++) {
13     let f_x = f(x);
14     if (Math.abs(f_x) < e2) {
15       return x;
16     }
17     let f1_x = f1(x);
18     let xp = x;
19     x = xp - f_x / f1_x;
20     err = Math.abs(x - xp);
21   }
22   return x;
23 };

```

Codice sorgente 4: Descrizione in JavaScript del metodo delle tangenti

k	x_k	f_k	f'_k	$ x_{k+1} - x_k $
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 11: Metodo delle tangenti applicato a $x^2 - 6$ con stima iniziale 3 e $n_{\max} = 10$

k	x_k	f_k	f'_k	$ x_{k+1} - x_k $
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 12: Metodo delle tangenti applicato a $\sin(x)$ con stima iniziale 3,1 e $n_{\max} = 10$

k	x_k	f_k	f'_k	$ x_{k+1} - x_k $
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 13: Metodo delle tangenti applicato a $e^{-x} - x$ con stima iniziale 0,5 e $n_{\max} = 10$

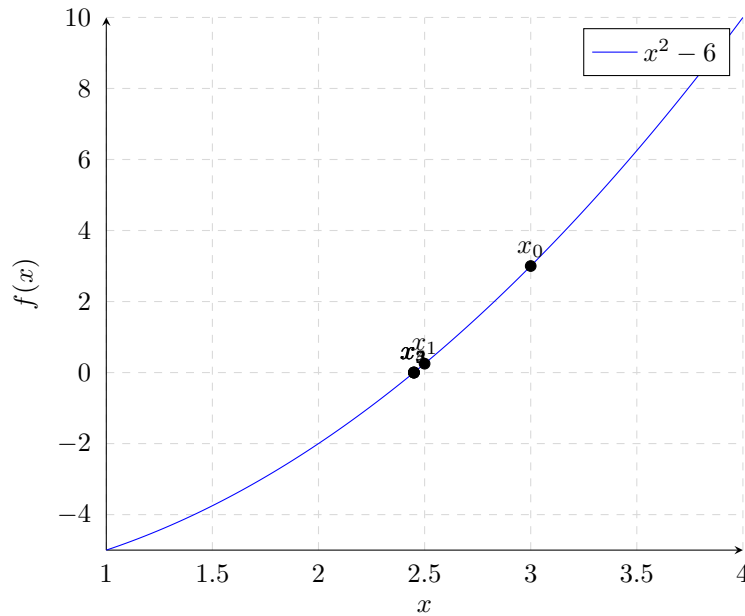


Figura 8: Successione delle soluzioni del metodo delle tangenti applicato a $x^2 - 6$ nell'intervallo $[0, 6]$

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

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.

5.5 Altri metodi

5.5.1 Metodo di Steffensen

Le derivate sono complesse da calcolare quando non sono note in forma esplicita ma è possibile ottenere un metodo di ordine di convergenza due applicando più valutazioni della funzione senza ricorrere alla derivazione.

$$x_{k+1} = x_k - \frac{f(x_k)}{g(x_k)}; \quad g(x_k) = \frac{f(x_k + f(x_k)) - f(x_k)}{f(x_k)}$$

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

Si noti che la valutazione della funzione f in x_k è molto vicina a zero. Per apprezzare la funzione g approssimi f' riscriviamo $g(x_k)$ con $h = f(x_k)$:

$$g(x_k) = \frac{f(x_k + h(x_k)) - f(x_k)}{f(x_k)}$$

5.5.2 Metodo di Muller

Un altro modo di approssimare la funzione f è quella di usare una parabola. Nel metodo di Muller si stimano tre punti iniziali e si determina la parabola passante per essi. Dalla soluzione dell'equazione di secondo grado si ottengono due punti che si usano come nuove stime. Risolvere le equazioni secondo grado, però, può portare a soluzioni inesistenti.