Complementi di Matematica e Calcolo Numerico A.A. 2018-2019

Laboratorio 9 - Equazioni non lineari

Data $f : \mathbb{R} \longrightarrow \mathbb{R}$ determinare $\alpha \in \mathbb{R}$ tale che

$$f(\alpha) = 0$$

Le soluzioni di questo problema vengono dette **radici** o **zeri** di f

Teorema degli zeri

Sia $f : \mathbb{R} \longrightarrow \mathbb{R}$, $f \in C^0([a,b])$, $f(a)f(b) < 0$, allora esiste $\alpha \in [a,b]$ tale che $f(\alpha) = 0$.

Nelle ipotesi del Teorema degli zeri, per determinare una radice α di f possiamo utilizzare la funzione predefinita di Matlab `fzero`.

Sintassi: `alfa = fzero(f,[a,b])`
input:

f è la funzione definita con @

$[a,b]$ sono gli estremi di un intervallo contenente la radice cercata che soddisfino $f(a) * f(b) < 0$

output:

`alfa` approssimazione della radice calcolata

`fzero` come gli altri metodi numerici che permettono di approssimare radici di funzioni sono di tipo iterativo, ovvero a partire da un valore iniziale x_0 producono una successione di approssimazioni che sotto opportune ipotesi converge a una radice di f.
Pertanto è possibile utilizzare `fzero` dando in input, al posto dell'intervallo $[a, b]$, un solo valore `x0`, a partire dal quale l'algoritmo cercherà la radice di f:

```
alfa = fzero(f, x0)
```

Attenzione: La convergenza dipende dalla scelta del punto iniziale `x0`. Se `x0` non viene scelto in maniera opportuna, l'algoritmo potrebbe non convergere o, in caso di più radici, potrebbe convergere ad una radice diversa da quella cercata!

Osservazione: Nel caso si utilizzi `fzero` non si incorre in questi problemi utilizzando la modalità precedente ed avendo l'accortezza di scegliere come $[a, b]$ un intervallo che contenga solo la radice voluta.

Osservazione: In caso si volessero trovare più radici della stessa funzione è necessario ripetere la procedura per ogni singola radice.

Onde localizzare ogni radice e scegliere un intervallo $[a, b]$ che la contenga è utile tracciare preliminarmente un grafico della funzione f. Uno studio preliminare della funzione per localizzare le eventuali radici sarà sempre utile anche utilizzando altri metodi.

Limitazioni

- La funzione `fzero` definisce uno zero come un punto in cui la funzione assegnata attraversa l'asse x. Punti in cui la funzione tocca ma non attaversa l'asse x non sono considerati zeri validi. Ad esempio la parabola $f(x) = x^2$ ha una radice doppia in 0 e quindi tocca ma non attraversa l'asse x pertanto `fzero` non è in grado di determinare tale radice di f.

```
>> f=@(x) x.^2;
>> x=fzero(f,[-1,1])
??? Error using ==> fzero at 293
The function values at the interval endpoints
 must differ in sign.

>> x=fzero(f,0.005)
Exiting fzero: aborting search for an interval
containing a sign change because NaN or Inf
function value encountered during search.
(Function value at -1.55333e+154 is Inf.)
Check function or try again with a different
starting value.
x =
   NaN
```

- La funzione `fzero(f,x0)` cerca di individuare punti in un intorno di x_0 in cui la f cambia segno, se la funzione assegnata è continua un tale punto corrisponde ad una radice di f altrimenti `fzero` può ritornare un punto di discontinuità anziché uno zero di f. Ad esempio

```
>> f=@(x) tan(x);
>> alfa=fzero(f,1)
alfa =
    1.5708
```

Si osservi che entrambe le limitazioni osservate sono strettamente legate al venire meno di qualche ipotesi del teorema degli zeri sul quale infatti si basa il metodo implementato in `fzero`.

Esercizio 1. Eseguire il grafico delle seguenti funzioni negli intervalli specificati ed in seguito, con la funzione `fzero`,trovarne le radici:

a. $f(x) = e^{-x} - \sin(x) \qquad x \in [-1, 5]$

b. $f(x) = (x^3 - 3x + 2)e^x, \qquad x \in [-3, 1.5]$

Metodo di Bisezione

`fzero` implementa una "versione sofisticata" del semplicissimo metodo di bisezione che a sua volta si basa sul teorema degli zeri ed è cosi' definito:
Sia $[a, b]$ un intervallo in cui siano soddisfatte le ipotesi del teorema degli zeri

- inizializzazione:

$$k = 1,\ a^{(1)} = a,\ b^{(1)} = b,$$
$$\text{calcolo } x^{(1)} = (a^{(1)} + b^{(1)})/2$$
$$\text{pongo } \text{err}^{(1)} = (b^{(1)} - a^{(1)})/2,$$

- finchè $\text{err}^{(k)} >$ `toll` itero le operazioni seguenti:
  se $f(x^{(k)}) = 0$, stop
  se $f(a^{(k)}) \cdot f(x^{(k)}) < 0 \rightarrow a^{(k+1)} = a^{(k)}, b^{(k+1)} = x^{(k)}$
  se $f(a^{(k)}) \cdot f(x^{(k)}) > 0 \rightarrow a^{(k+1)} = x^{(k)}, b^{(k+1)} = b^{(k)}$
  calcolo $x^{(k+1)} = (a^{(k)} + b^{(k)})/2$,
  pongo $\text{err}^{(k+1)} = \text{err}^{(k)}/2$
  aggiorno $k = k + 1$

dove `toll` è la precisione voluta. Si noti che per l'errore vale $|x^{(k)} - \alpha| < (b^{(k)} - a^{(k)})/2$ e che ad ogni iterazione l'ampiezza del sottointervallo si dimezza.
Il metodo converge sempre, non è quindi necessario fissare un numero massimo di iterazioni consentite.

Costruire una function Matlab che implementi il metodo sopra descritto con la seguente sintassi.

`[x, nit] = bisezione(f, a, b, toll)`
input:

f funzione definita con `@`

a, b con $a < b$: estremi di un intervallo contenente la radice cercata che soddisfino $f(a)f(b) < 0$

`toll` precisione richiesta

output:

`x` approssimazione della radice calcolata

`nit` numero iterazioni effettuate

Un esempio di possibile implementazione si trova nel file **bisezione.m** scaricabile dalla pagina web del corso

Attenzione: Come nel caso di `fzero`, punti in cui la funzione tocca ma non attaversa l'asse x non sono zeri calcolabili con la funzione `bisezione` in quanto avvivicinandoci a un tale punto non siamo più nelle ipotesi del Teorema degli zeri.
Ad esempio `bisezione` non è in grado di determinare la radice doppia della parabola $f(x) = x^2$.

Esercizio 2. Si consideri il problema della ricerca degli zeri α_1 e α_2 (con $\alpha_1 < \alpha_2$) della funzione non lineare

$$f(x) = e^x - x^2 - \sin(x) - 1, \quad -2 \leq x \leq 2.$$

1. Tracciare un grafico della funzione nell'intervallo considerato. Localizzare graficamente gli zeri di $f(x) = 0$ eventualmente con l'aiuto dello zoom.

2. Il metodo di bisezione è applicabile per calcolare tutti gli zeri?

3. Applicare il metodo, quando possibile, utilizzando il programma `bisezione` con tolleranza `eps=1e-8` e considerando un opportuno intervallo di partenza.

Esercizio 3. Si considerino le funzioni dell'Esercizio 1 e si determini, ove possibile, le radici negli intervalli di definizione sopraindicati utilizzando il metodo di bisezione con tolleranza `1e-5` e scegliendo in modo opportuno gli intervalli di partenza.

Metodo di Newton

Costruire una function Matlab che, dati dall'utente:

- una funzione f
- una funzione df (derivata di f)
- un punto iniziale x_0
- una tolleranza TOL
- un numero massimo di iterazioni NMAX,

trovi uno zero della funzione $f(x)$ usando il metodo di **Newton**

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \qquad k = 0, 1, 2, 3, ...$$

Il programma deve fermarsi qualora almeno una delle seguenti condizioni sia soddisfatta

- il modulo della differenza tra il valore x_k calcolato nel passo corrente e il valore x_{k-1} calcolato nel passo precedente è inferiore a TOL
- il numero totale di iterazioni effettuate è maggiore o uguale a NMAX.

ovvero

```
[x,nit]=newton(f,df,x0,toll,nitmax)
```

Esercizio 4. Si applichi il metodo di Newton alla funzione

$$f(x) = x^2 - 7$$

con $x_0 = 10$, TOL$= 10^{-12}$ e NMAX=200.

Esercizio 5. Si applichi il metodo di Newton alla funzione

$$f(x) = \sin(x^2)$$

con $x_0 = 1/2$, TOL=10^{-12}, NMAX=200.
Si osservi che il metodo di Newton è in grado di calcolare la radice multipa $\alpha = 0$ di f

Esercizio 6. Eseguire il grafico delle seguenti funzioni negli intervalli specificati ed in seguito trovarne le radici con il metodo di Newton scegliendo tolleranza $1e-6$ e numero massimo di iterazioni pari a 100:

a. $f(x) = \sin(e^x), \qquad x \in [0, 2.5]$

b. $f(x) = e^{-x}\sin(x), \qquad x \in [-1, 5]$

c. $f(x) = (x-1)^2(x+2), \qquad x \in [-3, 3]$

Nel caso c. la funzione f ha una radice doppia e una semplice, si confronti il numero di iterazioni effettuate dal metodo di Newton per calcolare ciascuna di esse. Quale radice viene calcolata più velocemente?

Esercizio 7. Combinazione di bisezione e Newton
Applicare sia il metodo di bisezione (con la scelta di intervallo $[-15, 20]$) che il metodo di Newton (con punto iniziale 20), entrambi con $TOL = 10^{-12}$, alla seguente funzione

$$f(x) = \text{arctg}(x)\ . \tag{1}$$

Commentare i risultati. Sono soddisfacenti in entrambi i casi?
Si scriva un programma che usi in successione il metodo di bisezione (toll $= 0.001$) e quello di Newton (toll $=$ 1e-12) passando a Newton come valore di innesco il valore ottenuto con le bisezioni, e lo si applichi alla funzione $f(x) = \text{arctg}(x)$ sull'intervallo $[-15, 20]$.

Esercizio 8. Di riepilogo

Si consideri la funzione

$$f(x) = x - \frac{1}{\sin(x) + 2}.$$

1. Si approssimi la radice α di f nell'intervallo $[0, 5]$ utilizzando la function di Matlab `fzero`.
2. Si approssimi la radice α di f nell'intervallo $[0, 5]$ utilizzando il metodo di Bisezione con tolleranza `1e-8`.
3. Si approssimi la radice α di f nell'intervallo $[0, 5]$ utilizzando il metodo di Newton con tolleranza `1e-8` e numero massimo di iterazioni 200 a partire dal punto iniziale $x_0 = 2$.
4. Considerando come soluzione esatta la radice trovata con `fzero`, calcolare gli errori di approssimazione commessi utilizzando gli altri due metodi.

Esercizio 9. Di riepilogo

Si consideri la funzione

$$f(x) = 2x^2 - \frac{1}{\cos(x^2) + 3}.$$

1. Si approssimi la radice α di f nell'intervallo $[0, 5]$ utilizzando la function di Matlab `fzero`.

2. Si utilizzi il metodo di bisezione nell'intervallo $[0, 5]$ con tolleranza `1e-2` per determinare il punto iniziale x_0 da passare al metodo di Newton per approssimare la radice α di f nell'intervallo $[0, 5]$. Si scelga per il metodo di Newton tolleranza `1e-6` e numero massimo di iterazioni 200. Sia $\tilde{\alpha}$ l'approssimazione calcolata.

3. Considerando come soluzione esatta α la radice trovata con `fzero`, calcolare l'errore di approssimazione commesso $|\alpha - \tilde{\alpha}|$.