

Laboratorio Calcolo Numerico

Esercizio 1

Si vuole determinare la soluzione α_2 appartenente all'intervallo $\mathcal{I} = [0.6, 0.8]$ dell'equazione non lineare $f(x) = x^2 - 1 + e^{-x} = 0.$, utilizzando il *Metodo di Newton*.

La funzione $f(x)$ soddisfa le ipotesi del Teorema di convergenza locale del Metodo di Newton per la radice α_2 (Teorema 3.12 del libro) nell'intervallo \mathcal{I} e la sua derivata non si annulla mai nell'intervallo \mathcal{I} .

Prendendo spunto dall'algoritmo di Pag. 106, si scrivano una function Matlab di nome `newtonfun.m` ed uno script di nome `newtonscript.m` che implementino il metodo di Newton.

- La function `newtonfun.m` deve avere come dati in ingresso **sia** la funzione $f \rightarrow \mathbf{f}$, **sia** la *derivata* $f' \rightarrow \mathbf{f1}$, che deve essere preventivamente calcolata a mano, ed ambedue precedentemente generate con il comando `inline`, da stringhe inserite con il comando `input`. Devono essere anche forniti in ingresso, il *valore iniziale* $x_0 \rightarrow \mathbf{x0}$, la *tolleranza* $\varepsilon \rightarrow \mathbf{toll}$ ed il numero massimo di iterazioni consentite $n_{\max} \rightarrow \mathbf{nmax}$;
- la stessa function deve dare come uscita il vettore `xv` che contiene le iterate, approssimazioni successive della soluzione (incluso il valore iniziale x_0), il vettore `fxv` dei residui calcolati nelle corrispondenti iterate, il numero `n` di iterazioni effettuate e una variabile di controllo `flag` che indichi un'eventuale divisione per zero;
- la function dovrà avere la seguente intestazione:

```
function [xv, fxv, n, flag] = newtonfun (f, f1, x0, toll, nmax)
%NEWTONFUN Metodo di Newton
% Uso:
%   [xv, fxv, n, flag] = newtonfun (f, f1, x0, toll, nmax)
%
% Dati di ingresso:
%   f:      funzione
%   f1:     derivata prima
%   x0:     valore iniziale
%   toll:   tolleranza richiesta per il valore assoluto
%           della differenza di due iterate successive
%   nmax:   massimo numero di iterazioni permesse
%
% Dati di uscita:
%   xv:     vettore contenente le iterate
%   fxv:    vettore contenente i corrispondenti residui
%   n:      numero di iterazioni effettuate
%   flag:   Se flag = 1 la derivata prima si e' annullata
```

- lo script `newtonscript.m` deve chiedere all'utente le due stringhe `exprf` e `exprf1` contenenti la funzione e la derivata, il valore iniziale, la tolleranza, il numero n_{\max} ;
- ottenuti gli argomenti di uscita dalla funzione (risultati), lo script deve visualizzare (comando `disp`) l'ultima soluzione approssimata determinata dalla function, il relativo residuo, e l'indice dell'ultima iterata calcolata;
- lo script deve dare un opportuno messaggio d'errore se vengono effettuate $n = n_{\max}$ iterazioni o se `flag` $\neq 0$;

- nello script, alla fine, si preveda anche un grafico che rappresenti (scala logaritmica sull'asse y) il valore assoluto del vettore che contiene i residuo (\mathbf{fxv}).

Si utilizzi lo script tre volte, definendo come valori iniziali i due estremi dell'intervallo ed un valore interno (con tolleranza `tol1 = 1e-8`). **Che cosa si può notare paragonando i vari risultati di Newton ed anche i risultati del Metodo di bisezione ottenuti con la stessa tolleranza?**

Si utilizzi poi nuovamente lo script, definendo come valore iniziale $x_0 = -10$ e stessa tolleranza. **Che cosa si può dedurre dai risultati ottenuti?**

Esercizio 2 Si vuole determinare il volume V occupato da un gas ad una temperatura T e soggetto ad una pressione p . L'equazione di stato del gas (ossia l'equazione che lega p , V e T) è

$$\left[p + \alpha \left(\frac{N}{V} \right)^2 \right] (V - N\beta) = kNT, \quad (1)$$

nella quale α e β sono dei coefficienti che dipendono dallo specifico tipo di gas, N è il numero di molecole di gas contenute nel volume V e k è la cosiddetta *costante di Boltzmann*, che vale $k = 1.3806503 \times 10^{-23} \text{ J} \cdot \text{K}^{-1} (\rightarrow 1.3806503\text{e-23})$. Si ricorda che un joule (1J) corrisponde a $1\text{Pa} \cdot \text{m}^3$.

Indicata la soluzione V cercata come variabile x , si può riscrivere l'equazione (1) come $f(x) = 0$

$$f(x) = \left[p + \alpha \left(\frac{N}{x} \right)^2 \right] (x - N\beta) - kNT.$$

In questa funzione, x è la *variabile*, mentre T , p , N , α e β sono dei semplici *parametri* e k una costante. Si produca uno script Matlab di nome `statogas.m` che

- Assegni il valore di $T \rightarrow \mathbf{T}$, $p \rightarrow \mathbf{p}$, $N \rightarrow \mathbf{nmol}$, $\alpha \rightarrow \mathbf{alpha}$ e $\beta \rightarrow \mathbf{beta}$;
- assegni la costante $k \rightarrow \mathbf{k} = 1.3806503\text{e-23}$;
- crei la funzione $f \rightarrow \mathbf{f}$ con i seguenti comandi (da **copiare** e **incollare** sullo script `statogas.m`:

```
exprf = ['(',num2str(p),'+',num2str(alpha),'.*((',num2str(nmol),...
        './x).^2)).*(x-',num2str(nmol),'*',num2str(beta),')-(',...
        num2str(k,'%15.8e'),'*',num2str(nmol),'*',num2str(T),')'];
f = inline(exprf);
```

- risolva il problema con il *Metodo di Bisezione* utilizzando la function `bisezfun.m`, chiedendo all'utente gli estremi dell'intervallo (di ampiezza 0.06) in cui il metodo va applicato, la tolleranza richiesta ed il numero massimo di iterazioni consentite (per la determinazione dell'intervallo da inserire in fase di esecuzione, si suggerisce di utilizzare preventivamente uno script che disegni la funzione in un certo intervallo);
- visualizzi i valori di T , p , N , α e β assegnati e la soluzione V trovata, indicando anche l'intervallo in cui è stata cercata e la tolleranza richiesta;
- dia un messaggio d'errore nel caso in cui venisse eseguito un numero troppo elevato di iterazioni.

Si usino i seguenti dati:

Per l'*anidride carbonica* (CO_2) i coefficienti α e β della (1) valgono rispettivamente $\alpha = 0.401 \text{ Pa} \cdot \text{m}^6$ ($\rightarrow 0.401$) e $\beta = 42.7 \times 10^{-6} \text{ m}^3$ ($\rightarrow 42.7\text{e-}6$).

Si trovi il volume occupato da 1000 molecole di anidride carbonica poste ad una temperatura $T = 300 \text{ K}$ e ad una pressione $p = 3.5 \times 10^7 \text{ Pa}$ ($\rightarrow 3.5\text{e}7$) utilizzando il metodo di bisezione nell'intervallo prescelto, con un'accuratezza pari a 10^{-10} ($\rightarrow 1\text{e-}10$).

Esercizio 3. Facoltativo

Si desidera calcolare la radice approssimata positiva e **non nulla** della funzione

$$f(x) = 1 - x - e^{-2x}.$$

Si desidera implementare il seguente metodo iterativo (chiamato **secante fissa**)

$$x_{n+1} = x_n - f(x_n) \frac{x_1 - x_0}{f(x_1) - f(x_0)}$$

che permette di determinare la radice reale contenuta nell'intervallo $[x_0, x_1]$.

Si localizzi graficamente, la soluzione reale positiva (non nulla) dell'equazione $f(x) = 0$ e, dopo aver trovato un intervallo sufficientemente piccolo $[x_0, x_1]$ (di ampiezza non maggiore di 0.5) che contenga la soluzione, si salvi la figura (formato ed estensione **.fig** o **.pdf**).

Si deve creare una function di nome **secfis.m** che implementi l'algoritmo relativo a tale metodo, ed uno script, con le seguenti indicazioni:

- La function **secfis.m** deve avere come dati in ingresso la funzione (inline function) $f \rightarrow \mathbf{f}$, precedentemente generata nello script con il comando **inline**, da una stringa inserita con il comando **input**. Devono essere anche forniti in ingresso, i due *valori iniziali* $x_0 \rightarrow \mathbf{x0}$ e $x_1 \rightarrow \mathbf{x1}$, la *tolleranza* $\varepsilon \rightarrow \mathbf{toll}$ per il test di arresto (basato sul valore assoluto della differenza di due iterate successive) ed il numero massimo di iterazioni consentite $n_{\max} \rightarrow \mathbf{nmax}$;
- la stessa function deve dare come uscita il vettore **xv** che contiene le iterate, approssimazioni successive della soluzione (inclusi i valori iniziali x_0 e x_1), il vettore **fxv** dei residui calcolati nelle corrispondenti iterate e il numero **n** di iterazioni effettuate;
- la function dovrà avere la seguente intestazione:

```
function [xv, fxv, n] = secfis (f, x0, x1, toll, nmax)
%SECFIS Metodo della secante fissa per equazione non lineare
%
% Uso:
%   [xv, fxv, n] = secfis(f, x0, x1, toll, nmax)
%
% Dati di ingresso:
%   f:      funzione
%   x0:      prima iterata
%   x1:      seconda iterata
%   toll:    tolleranza richiesta per il valore assoluto
%            tra due iterate successive
%   nmax:    massimo numero di iterate permesse
%
% Dati di uscita:
%   xv:      vettore contenente le iterate
%   fxv:     vettore contenente i corrispondenti residui
%   n:       numero di iterate della successione
```

- Lo script `secfisscript.m` deve chiedere all'utente la stringa `exprf` contenente la funzione, i due valori iniziali, la tolleranza, il numero n_{max} ;
- ottenuti gli argomenti di uscita dalla funzione (risultati), lo script deve visualizzare (comando `disp`) l'ultima soluzione approssimata determinata dalla function, il relativo residuo, e l'indice dell'ultima iterata calcolata;
- lo script deve dare un opportuno messaggio d'errore se vengono effettuate $n = n_{max}$ iterazioni;
- nello script, alla fine, si preveda anche un grafico che rappresenti (scala logaritmica sull'asse y) il valore assoluto del vettore che contiene i residuo (`fxv`).