### 1. Ricerca radici di equazioni non lineari - Metodo della bisezione

- -- Espressione:  $x^3 + 4x^2 10 = 0$ .
- -- Grafico per trovare l'intervallo iniziale. (es. provare [-10,10], [-5,5], [0.5,2.5]) [fplot(funz,lims)]
- -- Ricerca in [1, 2], con precisione 10<sup>-5</sup> (calcoli con calcolatrice, 2-3 iterazioni). Compilare tabella (ausilio Matlab):

n	a <sub>n</sub>	b <sub>n</sub>	$p_n$	f(p <sub>n</sub> )

### 2. Alcune funzioni di MATLAB

```
funz = 'sin(2*x)'
fplot(funz, [0, 2*pi] )
lims = [0, 2*pi] - - - fplot(funz, lims)

fnz = inline ('sin(3*x)') - fplot(fnz, lims)
y = pi - a=fnz(y) - feval(fnz, y) - k=feval(fnz, y)

y = (0 : pi/10 : 1) - - k = feval(fnz, y) - plot(y,k)

-- NB problemi con plot(x,y): x = (0 : pi/10 : 1) // y = sin(3*x) // z = x*sin(3*x)
```

### 3. Pseudocodifica

end:

INPUT: a, b, tol, Nmax, espressione

```
1)
       i = 1, an = a, bn = b, FA = f (an)
                                                                  (controlli sui dati)
2) while i <= Nmax
       3) pn = an + (bn-an)/2
         FP = f(pn)
                                                                   | (output parziale)
       4) if f(pn) = 0 OR |(bn-an)| / 2 < tol
               OUTPUT (pn - i)
                                                                   | NB tre condizioni di STOP
               STOP
          endif
       5) incremento i
       6) if FA*FP > 0
                             \rightarrow an = pn
                             \rightarrow FA = FP
                     \rightarrow bn = pn
          else
         endif
```

7) OUTPUT (soluzione non trovata dopo ... iterazioni)

# 4. Operazioni di I/O e gestione dei files in MATLAB

-- Input da pannello principale: R = input ('<string>') → valuta <u>espressione</u> e assegna a R

R = input ('<string>', 's') → acquisisce <u>come string</u> e assegna a R

poi F = inline (R, 'x')

-- Output scritte: (anche prompt per l'input): fprintf(1, ' ...  $\n'$ )

```
es.: fprintf(1,'estremo a dell'intervallo \n')
    a=input(' ')

oppure: a=input('estremo a? \n')
```

-- Forma generale per output formattato, scritte e valori:

# 5. Routine per applicare il metodo della Bisezione

operatori logici: OR: |, AND: & --- valori variabili logiche: true - false (NB lowercase)

- -- Trovare la radice dell'equazione al n.1 nell'intervallo [1, 2], con precisione 10<sup>-5</sup>
  - -- Quante iterazioni sono necessarie?
  - -- Fornire una stima, per ogni singola iterazione, dell'errore di approssimazione commesso.
  - -- Sapendo che la soluzione "esatta" è x = 1.365230013, calcolare l'errore assoluto e relativo ad ogni iterazione; che cosa si osserva? Commentare.

n	a <sub>n</sub>	b <sub>n</sub>	p <sub>n</sub>	f(p <sub>n</sub> )	p <sub>n</sub> -p <sub>n-1</sub>	Errori $(E_{max}, E_a, E_r)$

### 6. Esercizio

Calcolare  $\sqrt{3}$  con una approssimazione di  $10^{-4}$  usando il metodo della bisezione. (Considerare l'espressione  $f(x) = x^2 - 3 = 0$ )

### 7. Esercizio

- -- Espressione:  $x^3 + x 4 = 0$ .
  - a) Determinare il massimo numero di iterazioni necessario per trovare, con il metodo della Bisezione, la soluzione nell'intervallo [1, 4] con una precisione di 10<sup>-3</sup>.
  - b) Trovare questa soluzione approssimata; quante iterazioni sono state effettuate?

(funzione di MATLAB: y = log2(x))