

### **OBIETTIVO**

Il lavoro che ho preparato per l'esordio del colloquio d'esame consiste nella realizzazione di un programma in Pascal in grado di fornire le soluzioni approssimate di un'equazione di terzo grado avvalendosi del così detto metodo delle tangenti.

### **Introduzione : APPROSSIMARE UN' EQUAZIONE**

La risoluzione in modo esatto di un'equazione *non è sempre possibile*: le equazioni di grado superiore al quarto e le equazioni trascendenti, insieme a tutte quelle equazioni con coefficienti approssimati relative a problemi applicativi, spesso non lo sono. Per risolvere tali equazioni è perciò necessario conoscere i metodi di risoluzione approssimata o numerica.

Sebbene esistano formule risolutive per le equazioni di grado due, tre e quattro, anche per risolvere tali equazioni a volte diviene necessaria la risoluzione approssimata, soprattutto se le rispettive formule risolutive non si ricordano o conoscono. Ecco quindi spiegata la scelta di scrivere un programma in grado di eseguire tale operazione per le equazioni di terzo grado.

I metodi di risoluzione numerica si basano sul fatto che ogni equazione può essere vista come una funzione intersecata con l'asse delle ascisse; in quest'ottica risolvere l'equazione equivale ad individuare i punti di altezza nulla, gli zeri della funzione. Oltre che su questa basilare osservazione, i vari metodi si fondano sulla costruzione mediante l'iterazione di un'operazione di una successione di numeri reali che tende alla soluzione esatta; procedendo in questo modo diviene possibile avvicinarsi quanto ci è necessario alla soluzione voluta.

### **STRUMENTI MATEMATICI**

Per ricercare una soluzione approssimata è necessario determinare un intervallo che contenga solo essa. L'unicità della radice ci viene assicurata da due teoremi, che mi accingo ad enunciare.

#### **Primo teorema di unicità delle radici:**

Se una funzione è continua nell'intervallo limitato e chiuso  $[a;b]$  e i due estremi hanno immagini discordi (cioè  $f(a) \cdot f(b) < 0$ ), in altre parole se le ipotesi del teorema di esistenza degli zeri sono verificate ed inoltre la derivata prima  $f'(x)$  ha segno costante in ogni punto dell'intervallo  $[a;b]$ , allora la curva interseca l'asse delle  $x$  in un solo punto, ammette soltanto uno zero.

#### Ipotesi:

$f$  continua in  $[a;b]$

$f(a) \cdot f(b) < 0$

$f'(x)$  segno cost ( $f$  monotona) in  $[a;b]$

#### Tesi

Esiste  $! c \in ]a;b[ : f(c) = 0$

#### **Secondo teorema di unicità delle radici:**

Se una funzione è continua nell'intervallo limitato e chiuso  $[a;b]$  e i due estremi hanno immagini discordi ed inoltre la derivata seconda  $f''(x)$  ha segno costante in ogni punto dell'intervallo  $[a;b]$ , allora la curva interseca l'asse delle  $x$  in un solo punto.

#### Ipotesi:

$f$  continua in  $[a;b]$

$f(a) \cdot f(b) < 0$

$f''(x)$  cost in  $[a;b]$

#### Tesi

Esiste  $! c \in ]a;b[ : f(c) = 0$

### IL METODO DELLE TANGENTI o di Newton

Questo metodo è applicabile per le equazioni in cui la  $f(x)$  è continua e derivabile almeno due volte in un intervallo  $[a;b]$ . Inoltre all'interno di tale intervallo deve essere possibile stabilirne un altro in cui la derivata seconda della funzione mantenga costante il suo segno.

Un'equazione del tipo  $ax^3+bx^2+cx+d = 0$  soddisfa tali condizioni in quanto il suo dominio è tutto  $\mathbb{R}$ , non presenta punti di discontinuità di alcun genere ed è derivabile due volte.

Per applicare questo metodo bisogna per prima cosa disegnare la curva. Fatto ciò bisogna individuare un intervallo in cui la curva abbia derivata seconda costante in segno, cioè un intervallo in cui rivolga la concavità o sempre verso l'alto o sempre verso il basso.

In seguito si procede nel seguente modo:

1. Tracciare ed individuare algebricamente la retta tangente al grafico della funzione nell'estremo  $(x_i; f(x_i))$  la cui ordinata è concorde con  $f'(x)$ ; Questa retta ha equazione:  
 $y-f(x_i) = f'(x_i) \cdot (x-x_i)$
2. Tale retta interseca l'asse delle  $x$  nel punto di ascissa  $x_1 = x_i - f(x_i)/f'(x_i)$ .

Infatti se

$$y-f(x_i) = f'(x_i) \cdot (x-x_i)$$

$x_1$  è un valore approssimato dell'equazione cercata.

3. Ripetendo il procedimento per il punto  $(x_1; f(x_1))$  si ottiene una nuova e più precisa soluzione approssimata dell'equazione  $f(x) = 0$
4. Ripetendo più volte il procedimento si ottiene una successione di soluzioni approssimate dell'equazione, definite dalla seguente formula di ricorrenza:  
 $x_{n+1} = x_n - f(x_n) / f'(x_n)$
5. Il processo va arrestato quando  $|x_n - x_{n-1}| \leq$  precisione richiesta.

Questo metodo è molto efficiente se  $f'(x)$  è grande in valore assoluto, lo è meno se  $f$  interseca l'asse delle ascisse con una pendenza piccola.

### ANALISI DEL PROBLEMA e creazione del programma:

Considerata l'equazione generica  $ax^3+bx^2+cx+d = 0$ , abbiamo stabilito che nel caso in cui l'utente assegni ad  $a$  un valore pari a zero il programma termini in quanto tale assegnazione renderebbe l'equazione di secondo grado ed in tal caso il metodo di approssimazione risulterebbe di inutile applicazione.

Analizziamo ora separatamente il caso in cui  $a$  sia **positivo** ed il caso in cui sia **negativo**.

#### **Se $a > 0$ allora:**

Il dominio della funzione  $f(x) = ax^3+bx^2+cx+d$  è un sottoinsieme improprio dell'insieme dei numeri reali.

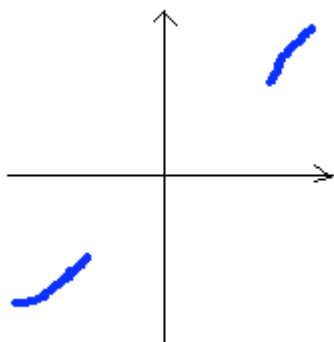
## RELAZIONE: Metodo delle tangenti con il Pascal

---

Dom= $\mathbb{R}$

I limiti agli estremi del campo sono:

$$\lim_{x \rightarrow \pm\infty} ax^3+bx^2+cx+d = \pm\infty$$



La derivata prima è

$$f'(x) = 3ax^2+2bx+c$$

Studiandone il segno è possibile capire dove la funzione cresce e dove cala, perciò poniamo  $f'(x) \geq 0$

La disequazione  $3ax^2+2bx+c \geq 0$  è di secondo grado e pertanto è necessario esaminare il segno del  $\Delta/4$  dell'equazione  $3ax^2+2bx+c=0$  per risolverla.

I casi possibili sono  $\Delta/4 > 0$ ,  $\Delta/4 < 0$  e  $\Delta/4 = 0$ . E' necessario a questo punto analizzarli uno per uno.

- **Se  $\Delta/4 > 0$  allora**

La *derivata prima* è positiva per valori esterni alle radici  $x_{1,2} = -b \pm \sqrt{(b^2-ac)}/3a$

$$f'(x) = 3ax^2+2bx+c > 0 \quad \text{con} \quad x < [-b - \sqrt{(b^2-ac)}/3a] \quad \vee \quad x > [-b + \sqrt{(b^2-ac)}/3a]$$

Questo significa che la funzione è crescente fino ad  $x_1$ , massimo relativo, cala fino ad  $x_2$  e di lì in poi riprende ad essere crescente.

Considerato che la *derivata seconda* ha equazione generica  $f''(x) = 6ax+2b$ , possiamo inoltre dire che il punto di flesso  $x_3$  è **unico**

$$\text{Cerco punto di flesso} \rightarrow \text{pongo } f''(x) = 0 \rightarrow 6ax+2b = 0 \rightarrow x_3 = -b/3a$$

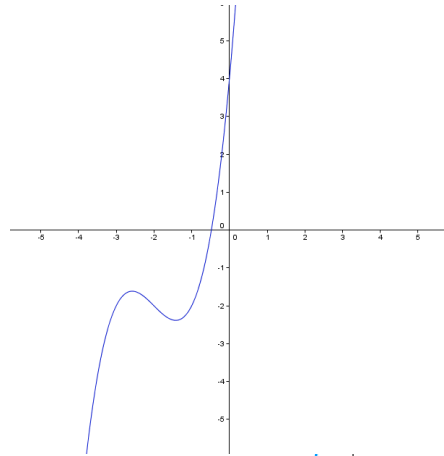
e che si trova ad essere necessariamente **compreso tra** i punti di massimo e minimo. Infatti  $x_1 = [-b - \sqrt{(b^2-ac)}/3a]$ ,  $x_2 = [-b + \sqrt{(b^2-ac)}/3a]$  e  $x_3 = -b/3a$ .

Studiando il segno della  $f''(x)$  scopriamo che la curva volge la concavità verso il basso per valori inferiori ad  $x_3$  e verso l'alto per valori maggiori a  $x_3$  in quanto è positiva per valori superiori a  $-b/3a$

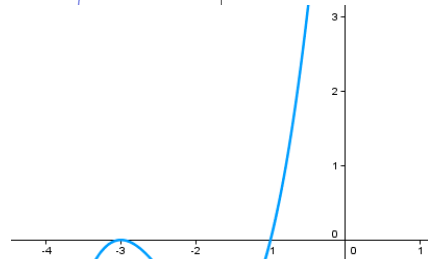
$$f''(x) > 0 \rightarrow 6ax+2b > 0 \rightarrow x > -b/3a$$

I possibili disegni del grafico sono i seguenti a seconda di quali siano i segni delle coordinate dei punti di massimo e di minimo:

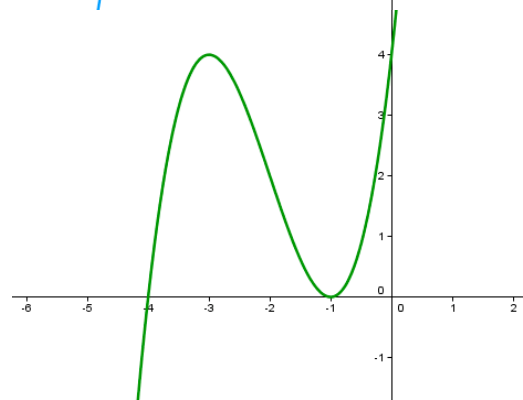
**A**



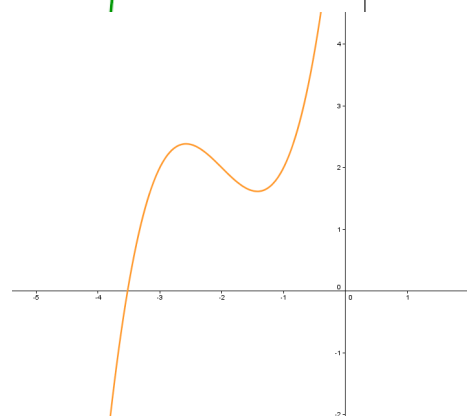
**B**



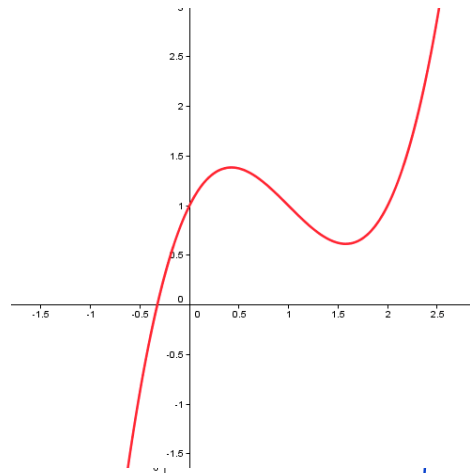
**C**



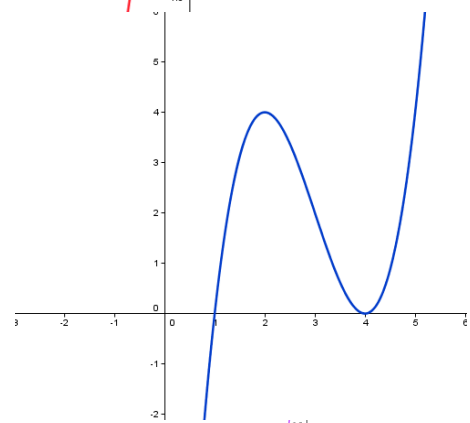
**D**



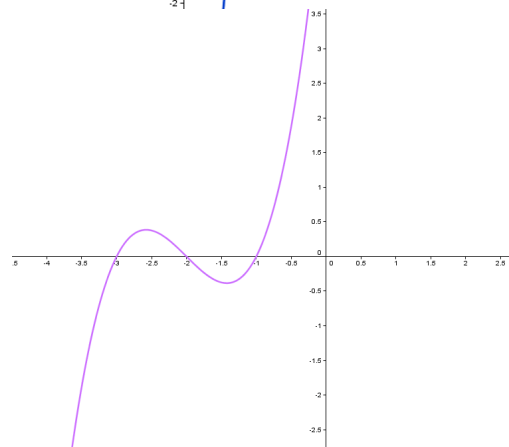
E



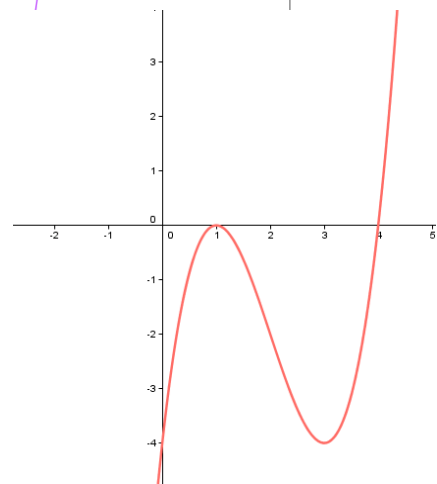
F



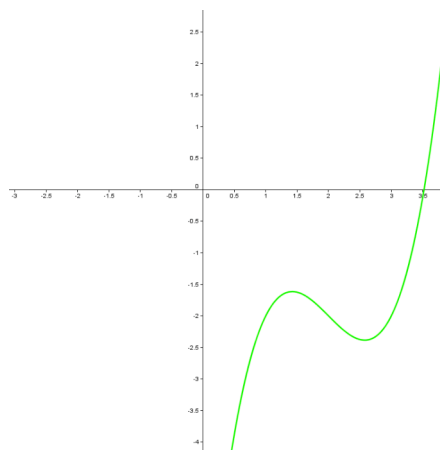
G



H



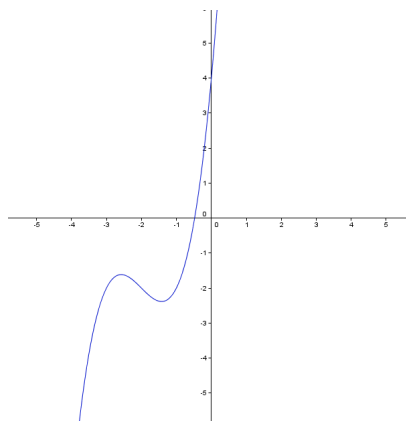
I



Il **caso A** ed il **caso I** sono assimilabili ad un unico caso, che chiameremo, così come ho fatto nel programma vero e proprio, **Caso  $\delta_4 > 0_1$**  in quanto nella ricerca dell'intervallo per attuare il metodo delle tangenti si procede in modo analogo.

Allo stesso modo sono assimilabili il **caso D** con il **caso E** (**Caso  $\delta_4 > 0_2$** ), il **caso C** con il **caso F** (**Caso  $\delta_4 > 0_3$** ) ed il **caso B** con il **caso H** (**Caso  $\delta_4 > 0_4$** ), mentre il **caso G** fa caso a sé e lo chiameremo **Caso  $\delta_4 > 0_5$** .

### **Caso Delta4Positivo\_1:**



In questo caso entrambe le immagini dei punti di massimo e minimo sono negative

$$f(x_1) = y_1 < 0 \quad \text{e} \quad f(x_2) = y_2 < 0$$

e anche il punto di flesso ha immagine negativa.

$$f(x_3) = y_3 < 0$$

Visto che  $x_1$  è massimo relativo e ha ordinata minore di zero e che tra  $x_1$  ed  $x_2$  la funzione cala possiamo dire che nell'intervallo  $]-\infty; x_2[$  la curva non interseca l'asse delle ascisse.

Per il *primo teorema di unicità delle radici* possiamo al contempo affermare che nell'intervallo  $[x_2; +\infty[$  esiste solo uno zero perché la funzione è monotona.

## RELAZIONE: Metodo delle tangenti con il Pascal

---

Per poter applicare il metodo delle tangenti è necessario individuare l'estremo la cui ordinata sia concorde alla derivata seconda.

Visto che la **derivata seconda** è **positiva per  $x > -b/3a$** , dove  $-b/3a$  come già detto è minore di  $x_2$ , l' **estremo ricercato**, che per ovvi motivi non può essere  $+\infty$ , deve avere **immagine positiva**.

Per individuarlo è quindi sufficiente "spostarsi a destra" rispetto a  $x_2$  fino a quando  $f(x)$  non risulta essere maggiore di zero.

Per far fare ciò al processore è stato necessario creare una **funzione**, chiamata **Xinnesco**. Questa funzione è costituita da una parte decisionale che assume come discriminante la variabile *mode*, che può alternativamente essere posta uguale a DX o a SX. In questo caso, poichè nella procedura **Delta4Positivo\_1** ci è necessario "spostarci a destra", ho assegnato a *mode* il valore DX, pertanto, in virtù della condizione dell'IF, si avvierà un ciclo WHILE in cui l'istruzione ripetuta è quella di sommare di volta in volta una quantità fissa K, e per cui il processo si innesca nuovamente solo se  $x$  ha immagine minore di zero.

Per calcolare di volta in volta l'immagine della  $x$  individuata sommando K mi sono servita di una **funzione**, che ho chiamato **F(x)** in grado di restituire il valore della  $y$  sostituendo nell'espressione della curva  $y = ax^3 + bx^2 + cx + d$  il valore di  $x$ .

Una volta individuata una  $x$  con ordinata positiva, il ciclo della funzione Xinnesco si arresta e ci fornisce l'estremo da cui partire ad applicare il metodo di Newton. L'ascissa di tale estremo è stata chiamata, e non solo in questo caso, **xi**.

Facile a questo punto applicare il metodo delle tangenti richiamando una funzione precedentemente creata e basata sul calcolo del punto di intersezione tra l'asse delle ascisse e la tangente passante per il punto della funzione  $xi$ .

Questo ciclo è strutturato in modo da arrestarsi quando  $|x_n - x_{n-1}| \leq$  precisione richiesta. Ogni volta che ricomincia assume come  $xi$  la  $xf$ , cioè la  $x$  calcolata nel "giro" precedente.

La funzione appena spiegata il cui nome è **Xapprossimata**, viene richiamata anche in tutti gli altri casi e a sua volta fa ausilio per funzionare di altre due funzioni. Oltre che servirsi di **F(x)** che calcola il valore della funzione, per calcolare l'**xapprossimata** è necessario conoscere il valore della derivata prima in quel punto e per fare ciò stata opportunamente inserita la funzione **FI(x)**.

*Ecco le funzini sopra citate*

```
function F(x : real) : real;
begin
  F := (a*x*x*x) + (b*x*x) + (c*x) + d;
end;
```

---

```
function FI(x : real) : real;
begin
  FI := (3*a*x*x) + (2*b*x) + c;
end;
```

---

```
function Xinnesco(x : real; mode : integer): real;
var xT : real;
begin
  xT := x;
  if ( mode = SX ) then
    begin
```

## RELAZIONE: Metodo delle tangenti con il Pascal

---

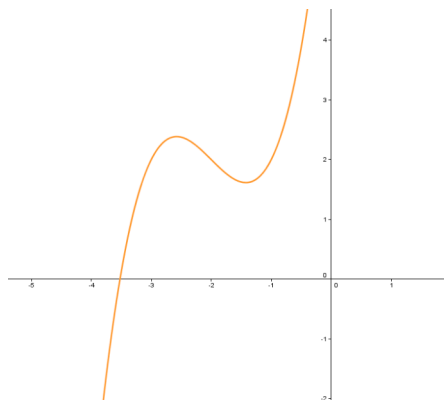
```
while ( F(xT) >= 0 ) do
    xT := xT - incremento;
end
else
begin
    while ( F(xT) <= 0 ) do
        xT := xT + incremento;
    end;
end;
```

---

```
function Xapprossimata(x : real) : real;
begin
    Xapprossimata := x - F(x) / FI(x);
end;
```

---

### Caso Delta4Positivo\_2:



In questo caso i punti di massimo e minimo hanno entrambi ordinata positiva e anche il punto di flesso ha immagine positiva.

$$f(x_1) = y_1 > 0 \quad \text{e} \quad f(x_2) = y_2 > 0 \\ f(x_3) = y_3 > 0$$

Visto che tra  $x_1$  ed  $x_2$  la funzione cala,  $x_2$  è minimo relativo e la sua ordinata è maggiore di 0 e visto che oltre  $x_2$  la funzione è crescente, possiamo dire che nell'intervallo  $]x_1; +\infty[$  la curva non interseca l'asse delle ascisse.

Per il primo teorema di unicità delle radici possiamo al contempo affermare che nell'intervallo  $]-\infty; x_1]$  esiste solo uno zero perché la funzione è monotona.

Per poter applicare il metodo delle tangenti è necessario individuare l'estremo la cui ordinata sia concorde alla derivata seconda.

Visto che la derivata seconda è negativa per  $x < -b/3a$ , dove  $-b/3a$  come già detto è maggiore di  $x_1$ , tale estremo deve avere immagine negativa.

Per individuarlo è sufficiente "spostarsi a sinistra" rispetto a  $x_1$  fino a quando  $f(x)$  non risulta essere minore di zero.

Per far fare ciò al processore è stato necessario richiamare la funzione **Xinnesco**, che assumendo il valore SX come *mode*, esegue in questo caso la parte della procedura costituita di un ciclo WHILE la cui istruzione ripetuta è quella di sottrarre di volta in volta una quantità fissa K; il processo si innesca nuovamente solo se x ha immagine maggiore di zero. Per calcolare di volta in volta l'immagine della x individuata sottraendo K anche in questo caso mi sono servita della funzione F(x).

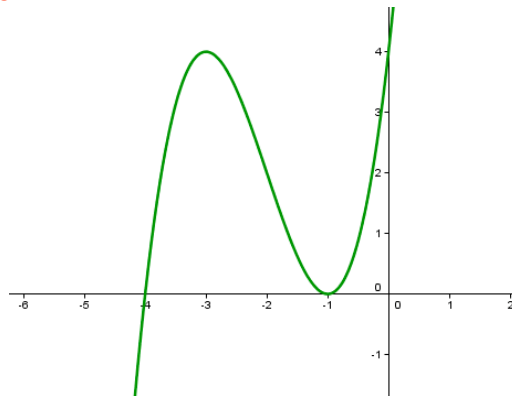
## RELAZIONE: Metodo delle tangenti con il Pascal

---

Una volta individuata una  $x$  con ordinata negativa, cioè  $x_i$ , il ciclo della funzione Xinesco si arresta e ci fornisce l'estremo da cui partire ad applicare il metodo di Newton.

A questo punto richiamando la funzione Xapprossimata il processore elabora e ci fornisce il valore  $x_f$ , la soluzione approssimata.

### Caso Delta4Positivo\_3:



In questo caso il punto di massimo relativo ha ordinata positiva mentre il punto di minimo ha immagine uguale a zero.

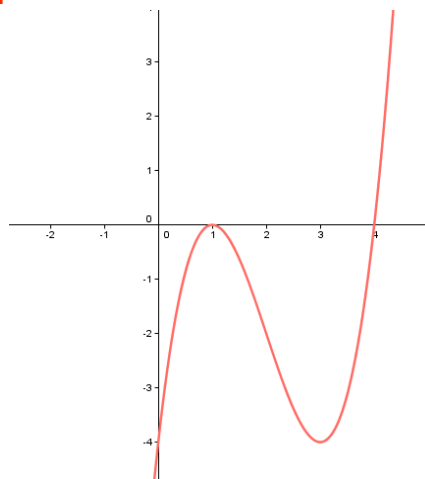
$$f(x_1) = y_1 > 0 \quad \text{e} \quad f(x_2) = y_2 = 0$$

Una soluzione corrisponde quindi al punto di minimo in cui si annulla la derivata seconda, mentre l'altra si ricava attuando un processo analogo a quello usato per individuare la soluzione nel *Caso delta4>0\_2*

Soluzione1 =  $x_2$

Soluzione2 → applico il medesimo metodo di ricerca utilizzato nel *Caso delta4>0\_2*

### Caso Delta4Positivo 0\_4:



## RELAZIONE: Metodo delle tangenti con il Pascal

---

In questo caso il punto di massimo relativo ha ordinata uguale a zero e il punto di minimo invece ha ordinata minore di zero.

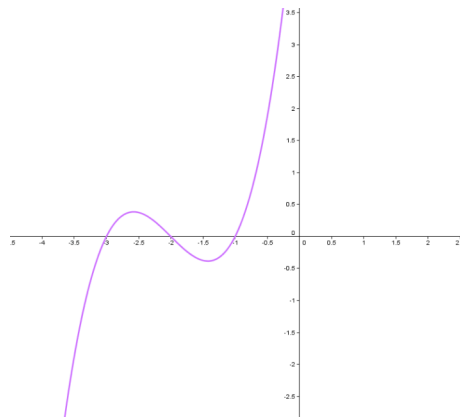
$$f(x_1) = y_1 = 0 \quad \text{e} \quad f(x_2) = y_2 < 0$$

Una soluzione è il punto di minimo, mentre l'altra si ricava attuando un processo analogo a quello usato per individuare la soluzione nel *Caso  $\Delta > 0_1$*

Soluzione1 =  $x_2$

Soluzione2 → applico il medesimo metodo di ricerca utilizzato nel *Caso  $\Delta > 0_2$*

### Caso Delta4Positivo\_5:



In questo caso il massimo relativo ha ordinata maggiore di zero e il punto di minimo invece ha immagine di segno opposto.

$$f(x_1) = y_1 > 0 \quad \text{e} \quad f(x_2) = y_2 < 0$$

Si può osservare come la funzione intersechi la funzione in tre punti distinti.

Per calcolare la **soluzione più piccola** bisogna procedere **come nel Caso  $\Delta > 0_2$**  in virtù del fatto che in  $]-\infty; x_1]$  la funzione è monotona, mentre per conoscere la **soluzione più grande** è necessario operare come nel *Caso  $\Delta > 0_1$* , poiché la funzione si mantiene crescente nell'intervallo  $[x_2; +\infty[$ .

Per giungere ad approssimare la **soluzione compresa** tra quelle sopra citate bisogna invece fare altre osservazioni.

Per individuarla è indispensabile conoscere il punto di flesso e calcolarne l'ordinata al fine di differenziare il processo da svolgere a seconda del segno di quest'ultima.

- Nel caso in cui l'ordinata del punto di flesso fosse pari a zero, l'ascissa del punto di flesso sarebbe soluzione.

$$f(x_3) = y_3 = 0 \rightarrow x_3 \text{ è soluzione}$$

- Nel caso in cui l'ordinata del punto di flesso fosse maggiore di zero, essa sarebbe concorde all'ordinata dell'estremo ricercato al fine di applicare il metodo di approssimazione per valori di  $x$  superiori a  $-b/3a$ .

$$f(x_3) = y_3 > 0 \rightarrow \text{concorde a } f'(x) \text{ per } x > x_3$$

Il meccanismo iterativo di approssimazione deve pertanto esordire da un estremo con ascissa  $x_i = x_3 + K$ ; per individuarlo utilizzo la funzione  $X$  innesco, ponendo  $mode = DX$ .

Una volta individuato sarà sufficiente applicare la funzione  $X$  approssimata a tale valore di  $x$ .

- Nel caso in cui l'ordinata del punto di flesso fosse minore di zero, essa sarebbe concorde all'ordinata dell'estremo ricercato al fine di applicare il metodo di approssimazione, per valori di  $x$  inferiori a  $-b/3a$ .

$$f(x_3) = y_3 > 0 \rightarrow \text{concorde a } f'(x) \text{ per } x > x_3$$

Il meccanismo iterativo di approssimazione deve pertanto esordire da un estremo con ascissa  $x_i = x_3 - K$ ; dove  $x_3$  è l'ascissa del punto di flesso;

per individuarlo utilizzo perciò la funzione  $X$  innesco, con  $mode$  pari a  $SX$ .

Una volta individuato sarà sufficiente applicare la funzione  $X$  approssimata a tale valore di  $x$ .

- Se  $\Delta/4 < 0$

La derivata prima è positiva per qualsiasi valore di  $x$ , perciò la funzione è crescente in tutto il dominio  $\mathbb{R}$ .

$$f'(x) = 3ax^2 + 2bx + c > 0 \quad \text{con} \quad \text{qualsunque } x \in \mathbb{R}$$

Da questo consegue che, per il *primo teorema di unicità delle radici*, la curva interseca l'asse delle ascisse una sola volta.

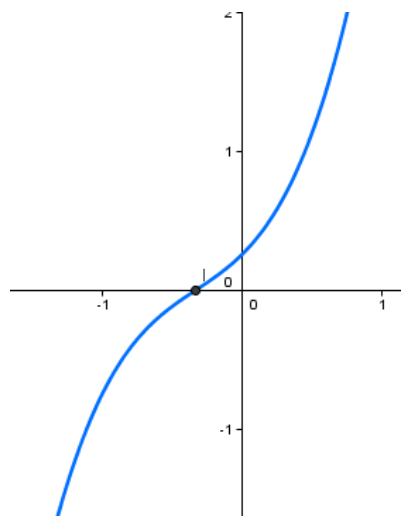
Considerato che la *derivata seconda* ha equazione generica  $f''(x) = 6ax + 2b$ , la funzione ha un unico cambio di concavità in  $x = -b/3a$

Cerco punto di flesso  $\rightarrow$  pongo  $f''(x) = 0 \rightarrow 6ax + 2b = 0 \rightarrow x_3 = -b/3a$

$f''(x) > 0 \rightarrow 6ax + 2b > 0 \rightarrow x > -b/3a \rightarrow$  la concavità verso il basso per valori inferiori ad  $x_3$ , concavità verso l'alto per valori maggiori a  $x_3$

A seconda dell'ordinata del punto di flesso ci si potranno presentare i seguenti casi:

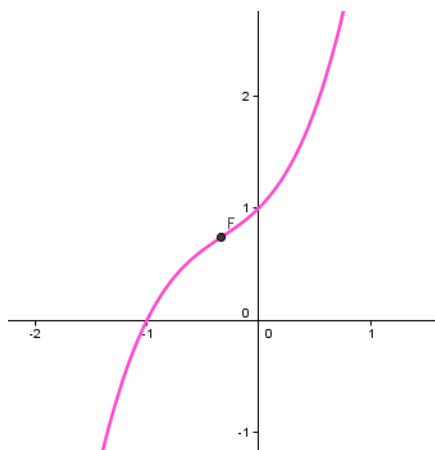
### Caso Delta4Negativo\_1:



In questo caso l'ordinata del punto di flesso è nulla, pertanto, visto che il punto di flesso appartiene alla curva, la sua ascissa è soluzione ricercata dell'equazione.

$$f(x_3) = y_3 = 0 \rightarrow x_3 \text{ è soluzione}$$

### Caso Delta4Negativo\_2:



In questo caso il punto di flesso ha ordinata maggiore di zero;

$$f(x_3) = y_3 > 0$$

Considerato che la funzione è sempre crescente, si può dedurre che il punto intersezione con l'asse delle x (di ordinata pari a zero) ha necessariamente ascissa minore rispetto a  $x_3$ , quella del punto di flesso.

$$f'(x) > 0, f(x_3) = y_3 > 0 \rightarrow \text{soluzione} < x_3$$

Fatte queste considerazioni si può dedurre che l'intervallo ricercato al fine di applicare il metodo delle tangenti sarà  $[x_i; x_3]$ .

Visto che per  $x < x_3$  la concavità è rivolta verso il basso e  $f(x_3) = y_3 > 0$ , sarà  $x_i$ , e non  $x_3$ , il primo valore di x utile a soddisfare la concordanza di segno della sua ordinata con la derivata seconda della funzione.

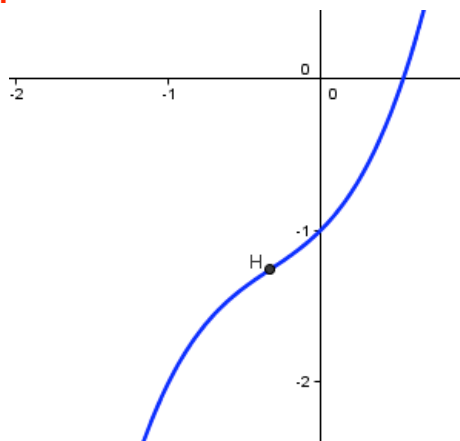
$$f(x_3) = y_3 > 0, f''(x) < 0 \text{ per } x < x_3 \rightarrow x_i = x_3 - K$$

Per individuare  $x_i$  è in altre parole necessario spostarsi a sinistra rispetto  $x_3$  quel tanto che basta affinché l'immagine di x sia negativa.

Al fine di fare ciò risulta nuovamente utile la funzione Xinnessco, con *mode* pari a SX.

A questo punto basta richiamare la funzione Xapprossimata e applicarla alla  $x_i$  trovata: il processore elabora e ci fornisce il valore  $x_f$ , la soluzione approssimata

### Caso Delta4Negativo\_3:



In questo caso il punto di flesso ha ordinata minore di zero;

$$f(x_3) = y_3 < 0$$

Considerato che la funzione è sempre crescente, si può affermare che il punto della curva di ordinata pari a zero ha necessariamente ascissa maggiore rispetto a  $x_3$ , quella del punto di flesso.

$$f'(x) > 0, f(x_3) = y_3 < 0 \rightarrow \text{soluzione} > x_3$$

Fatte queste considerazioni possiamo dire con certezza che l'intervallo ricercato al fine di applicare il metodo delle tangenti sarà  $[x_3; x_i]$ .

Visto che per  $x > x_3$  la concavità è rivolta verso l'alto,  $x_i$  sarà il primo valore di  $x$  utile a soddisfare la concordanza di segno della sua ordinata con la derivata seconda della curva.

$$f(x_3) = y_3 < 0, f''(x) > 0 \text{ per } x > x_3 \rightarrow x_i = x_3 + K$$

Per individuare  $x_i$  è in altre parole necessario spostarsi a destra rispetto a  $x_3$  quel tanto che basta affinché l'immagine di  $x$  sia positiva.

Al fine di fare ciò è nuovamente utilizzabile la funzione Xinnesco con *mode* uguale a DX.

A questo punto basta richiamare la funzione Xapprossimata e applicarla alla  $x_i$  trovata: il processore elabora e ci fornisce il valore  $x_f$ , la soluzione approssimata.

- **Se  $\Delta/4=0$**   
 $3ax^2+2bx+c$  è un quadrato di binomio .

La derivata prima si annulla nel punto soluzione dell'equazione  $3ax^2+2bx+c=0$  ed è sempre positiva per valori maggiori e minori a tale soluzione.

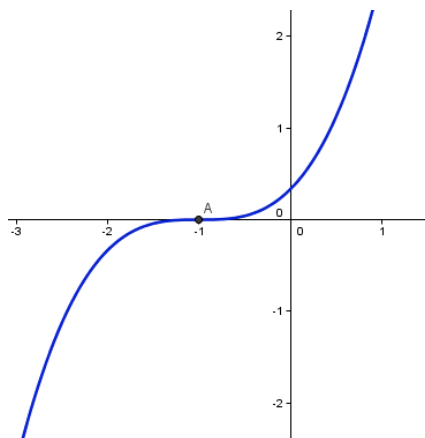
$$f'(x) = 3ax^2 + 2bx + c > 0 \quad \text{con} \quad \text{qualunque } x \in \mathbb{R} - \{ x_{1,2} = -b \pm \sqrt{(b^2 - ac)}/3a \}$$

dove  $x_1 = x_2$

Sapendo ciò, possiamo affermare che il punto  $x_1$  (o  $x_2$ , tanto le due soluzioni coincidono) è di **flesso orizzontale**; le nostre conoscenze infatti corrispondono alle *condizioni sufficienti per i flessi orizzontali*.

A seconda dell'ordinata del punto di flesso orizzontale ci si potranno presentare i seguenti casi:

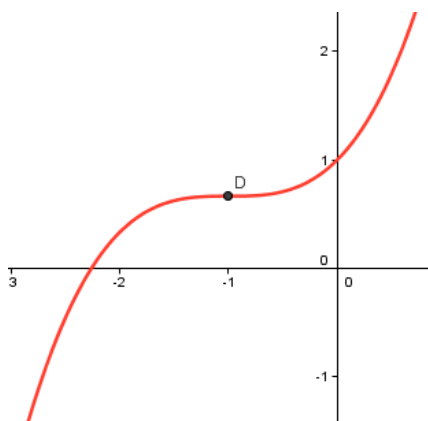
### Caso Delta4Nullo\_1:



In questo caso l'ordinata del punto di flesso orizzontale è nulla.  
Visto che il punto di flesso appartiene alla curva, la sua ascissa è la soluzione ricercata dell'equazione  $ax^3+bx^2+cx+d=0$

$$f(x_{1,2}) = y_{1,2} = 0 \rightarrow x_{1,2} \text{ è soluzione}$$

### Caso Delta4Nullo\_2:



In questo caso il punto di flesso orizzontale ha ordinata maggiore di zero.

$$f(x_1) = y_1 > 0$$

## RELAZIONE: Metodo delle tangenti con il Pascal

---

Considerando che la funzione è sempre crescente, si può dedurre che il punto intersezione con l'asse delle ascisse (di ordinata pari a zero) ha necessariamente ascissa minore rispetto a  $x_{1,2}$ , quella del punto di flesso orizzontale.

$$f'(x) > 0, f(x_{1,2}) = y_{1,2} > 0 \rightarrow \text{soluzione} < x_{1,2}$$

Fatte queste considerazioni appare evidente che l'intervallo ricercato al fine di applicare il metodo delle tangenti sarà  $[x_i; x_1]$ .

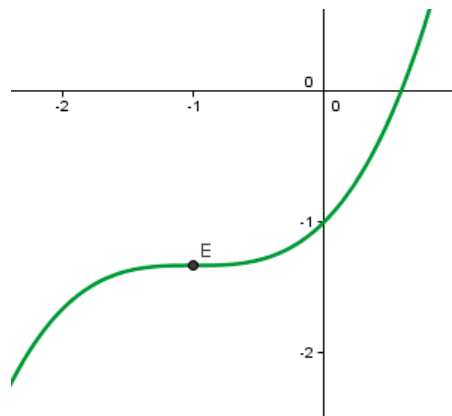
Visto che per  $x < x_3$  la concavità è rivolta verso il basso,  $x_i$  sarà il primo valore di  $x$  utile a soddisfare la concordanza di segno della sua ordinata con la derivata seconda negativa.

$$f(x_1) = y_1 > 0, f''(x) < 0 \text{ per } x < x_3 \text{ con } x_3 = x_1 \rightarrow x_i = x_1 - K$$

Per individuare  $x_i$  è in altre parole necessario spostarsi a sinistra rispetto  $x_1$  quel tanto che basta affinché l'immagine di  $x$  sia negativa.

Al fine di fare ciò risulta opportuna la funzione Xinnessco, *mode* pari a SX.

### Caso Delta4Nullo\_3:



In questo caso il punto di flesso orizzontale ha ordinata minore di zero; considerato anche che la funzione è sempre crescente, si può dedurre che il punto intersezione con l'asse delle ascisse (di ordinata pari a zero) ha necessariamente ascissa maggiore rispetto a  $x_{1,2}$  quella del punto di flesso orizzontale.

$$f'(x) > 0, f(x_{1,2}) = y_{1,2} < 0 \rightarrow \text{soluzione} > x_{1,2}$$

Fatte queste considerazioni appare evidente che l'intervallo ricercato al fine di applicare il metodo delle tangenti sarà  $[x_1; x_i]$ .

Visto che per  $x > x_1$  la concavità è rivolta verso l'alto,  $x_i$  sarà il primo valore  $x$  utile a soddisfare la concordanza di segno della sua ordinata con la derivata seconda.

$$f(x_1) = y_1 < 0, f''(x) > 0 \text{ per } x > x_3 \text{ con } x_3 = x_1 \rightarrow x_i = x_1 + K$$

Per individuare  $x_i$  è in altre parole necessario spostarsi a destra rispetto a  $x_1$  quel tanto che basta affinché l'immagine di  $x$  sia positiva.

Al fine di fare ciò bisogna usare nuovamente la funzione Xinnessco, con *mode* pari a DX.

### Se $a < 0$ allora:

Per analizzare questo caso sarebbe necessario analizzare il problema allo stesso modo in cui si è preceduto per  $a > 0$ .

Tuttavia risulta più immediato e più chiaro e funzionale ai fini della scrittura e lettura del programma ricondurre questo caso a quello analizzato in precedenza.

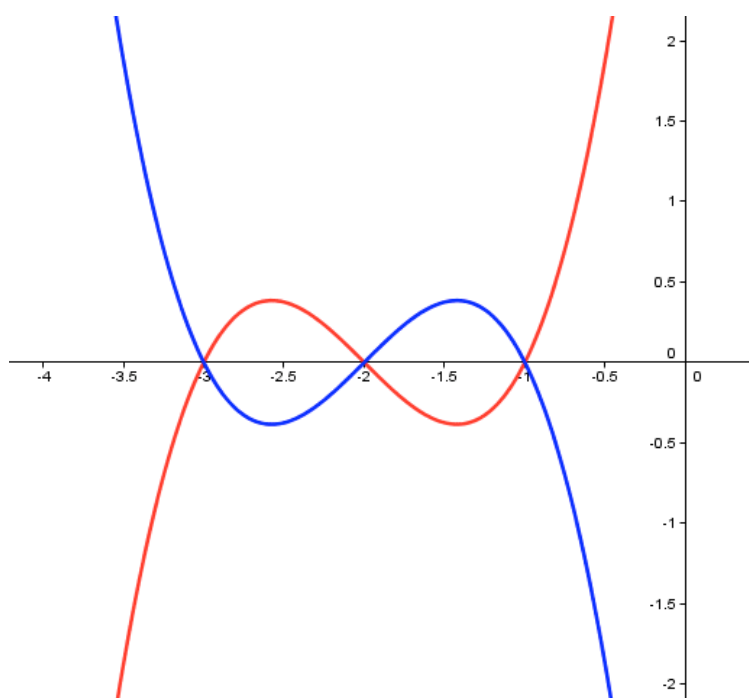
Per farlo è sufficiente individuare la **curva con  $a > 0$  simmetrica rispetto all'asse delle ascisse alla curva con  $a < 0$** ; per ricondurla al caso già analizzato basta moltiplicare per  $-1$  tutti i termini dell'espressione. Tale operazione è facilmente eseguibile dal processore tramite la creazione di una semplice procedura che ho chiamato **inversione\_coefficienti**.

Questo percorso alternativo risulta decisamente più economico in termini di quantità di istruzioni da scrivere nel programma visto che studiare la curva originaria piuttosto che la sua simmetrica è indifferente ai fini dell'individuazione della soluzione: i punti di intersezione con l'asse delle  $x$  non cambiano, le radici sono le stesse.

$$ax^3+bx^2+cx+d = 0$$

$$(ax^3+bx^2+cx+d)(-1) = 0(-1)$$

$$-ax^3-bx^2-cx-d = 0$$



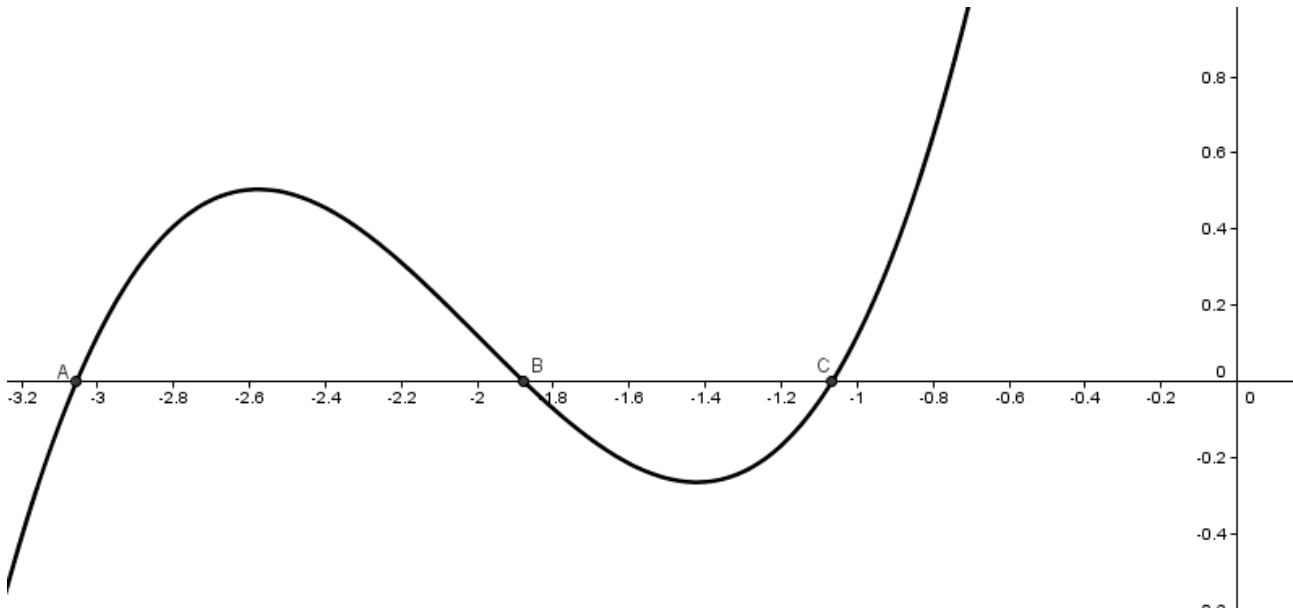
Ecco la procedura che permette di trovare la curva simmetrica:

```
procedure inversione_coefficienti();
begin
  a := (-1) * a;
  b := (-1) * b;
  c := (-1) * c;
  d := (-1) * d;
end;
```

**ESEMPIO**

L'equazione che vogliamo risolvere è:  $x^3+6x^2+11x+6.12$   
La precisione con cui vogliamo ottenere le soluzioni è 0.0001

Ecco il grafico della curva  $f(x)= x^3+6x^2+11x+6.12$



Ecco come ci risponderebbe il programma:

```
C:\Tesina\PrjApprox.exe
Questo programma e' in grado di fornire una soluzione
approssimata di una equazione polinomiale intera di terzo
grado.

Stabiliamo che:
A sia il coefficiente del terzo grado
B sia il coefficiente del secondo grado
C sia il coefficiente del primo grado
D sia il termine noto

Inserisci ora il valore di A (0 se NON vuoi procedere): 1
Inserisci il valore di B : 6
Inserisci il valore di C : 11
Inserisci il valore del termine noto D : 6.12
Inserisci la precisione voluta: 0.0001

Questa e' l' equazione che vuoi risolvere:

      1.00x^3 + 6.00x^2 + 11.00x + 6.12

Caso delta4Positivo_5

La Soluzione1 e':-3.05532
La Soluzione2 e':-1.06648
La Soluzione3 e':-1.06648

Premi INUIO per continuare
```

## **OSSERVAZIONI CONCLUSIVE**

In conclusione di questo lavoro, che ha richiesto un'attenta analisi del problema in tutte le sue sfaccettature e una particolare cura e precisione nello scrivere il testo del programma, ho acquisito una maggior consapevolezza a proposito del complesso lavoro che sta dietro a tutti quei programmi che usiamo abitualmente con tanta facilità, senza soffermarci sulla selva di conoscenze e ragionamenti logici che costituiscono le fondamenta di così fruibili strumenti.

Scrivendo il programma e varando tutte le possibili caratteristiche che la curva  $f(x) = ax^3 + bx^2 + cx + d$  poteva assumere, mi sono inoltre stupita, come non troppo spesso si fa, delle incredibili capacità della mente umana che, esulando dall'ambito informatico di programmazione, è in grado di affrontare un problema di questo tipo dal punto di vista logico in modo più immediato e senza che noi ci avvediamo della complessità di ciò che stiamo facendo per esempio semplicemente attestando che  $a$  è maggiore di zero.

### APPENDICE

Un programma è una sequenza di istruzioni. Una ricetta, una composizione musicale, un lavoro a maglia sono tutti programmi; se intendiamo i programmi in tale accezione, essi esistono da molto tempo prima che fossero inventati i computer. I programmi per computer però sono più lunghi ed intricati di altre specie di programmi e scriverli richiede cura e precisione.

Alcune caratteristiche comuni dei programmi sono:

1. Le istruzioni vengono eseguite in sequenza. A meno che non sia specificato altrimenti, le istruzioni vengono eseguite a partire dalla prima in ordine, una alla volta, fino alla fine. Questo cammino lineare può essere interrotto, deviato e le istruzioni ripetute. Nel caso del programma da me presentato tale possibilità assume particolare importanza in quanto approssimare un'equazione è necessario iterare una certa operazione fintanto che si raggiunge la precisione desiderata.
2. il processo ha un effetto, l'output, nel nostro caso la soluzione dell'equazione;
3. Il processo opera su dati che vengono forniti dall'utente e le quali caratteristiche devono essere preventivamente dichiarate dall'autore del programma.  
Il tipo delle variabili scelte per questo programma è REAL e l'intervallo dei valori consentiti per questo tipo di dati va da  $2,9 \cdot 10^{-39}$  a  $1,7 \cdot 10^{38}$ .
4. Talvolta le istruzioni richiedono una decisione, cioè il processore deve eseguire certe istruzioni piuttosto che altre a seconda che venga o meno soddisfatta una condizione posta dall'autore; in altre parole il processore deve decidere in base ad una condizione. In questo programma ad esempio le operazioni da svolgere si differenziano a seconda che il coefficiente del grado massimo sia positivo o negativo.
5. Come anticipato, spesso è necessario ripetere una o un gruppo di istruzioni più di una volta (si pensi alle procedure non informatiche, ad esempio nel lavoro a maglia). Per far sì che questo avvenga bisogna specificare quante volte eseguirle ("lavora dieci righe") oppure stabilire un criterio che dipenda dallo stato del processo ("lavora fino la fine della riga"). Nel nostro caso il criterio dipendente dallo stato del processo è la precisione della soluzione e l'utente ha la possibilità di stabilirla inserendone il valore nelle fasi iniziali di input.
6. Il programma in sé è un'entità statica, è un testo, ma il processo di esecuzione è dinamico e pertanto non vanno confusi. Scrivere un programma significa individuare tutte le possibili casistiche e le possibili strade da percorrere al fine di giungere all'output desiderato, altra cosa è invece il processo che consiste in ciò che viene eseguito a partire da certi dati.

Il programma consiste essenzialmente in un mezzo mediante cui l'autore comunica con il processore ( se una composizione musicale è l'insieme di istruzioni per eseguire un brano, l'esecutore del brano è il processore, il processore è quell'ente che esegue le istruzioni contenute nel testo ). Tale comunicazione necessita di un linguaggio speciale: musicisti, coreografi e tessitori ne hanno sviluppati di interamente originali per comunicare le loro istruzioni e come loro hanno fatto anche quelli che per primi si sono proposti di realizzare programmi con il computer.

Uno di questi linguaggi è il linguaggio Pascal ed è quello che ho scelto di utilizzare per scrivere il programma oggetto del mio approfondimento. Questo linguaggio, ampiamente usato a scopi didattici, costituisce un importante punto di riferimento nello sviluppo dei linguaggi di programmazione, fu inventato da Niklaus Wirth e fu chiamato così in onore del matematico e filosofo francese Blaise Pascal, inventore della prima macchina calcolatrice automatica in grado di svolgere addizioni e sottrazioni. Questo linguaggio, il primo a

## RELAZIONE: Metodo delle tangenti con il Pascal

---

seguire i concetti di programmazione strutturata, costringe ad un'attenta analisi del problema necessaria alla successiva scrittura del codice.

Per programmazione strutturata si intende quella programmazione che non si limita a realizzare programmi costituiti di liste di istruzioni primitive, comprensibili solo entro un certo grado di complessità anche a chi scrive il programma, ma che si fonda su di una struttura che permette di comprendere significato e conseguenza di ogni istruzione. Le quattro strutture basilari sono la sequenza, la decisione, il ciclo ed la procedura.

La sequenza è un gruppo di istruzioni eseguite una dopo l'altra, la decisione permette di influire con i dati sull'azione del programma ed in Pascal è introdotta dalla parola riservata IF, cui deve seguire una condizione e THEN, subito succeduti dall'istruzione. La struttura ciclo invece viene usata per ripetere una o più istruzioni diverse volte ed in Pascal si può ottenere ad esempio con le seguenti istruzioni riservate:

Repeat

Istruzione

Until <condizione>

Oppure

While <condizione> do

istruzione

Nel primo esempio di ciclo l'istruzione viene ripetuta fino a quando ciò che produce non soddisfa la condizione, mentre il ciclo While si avvia fintanto che una certa condizione viene rispettata.

Infine la procedura è un programma dentro il programma che ci consente di rimpiazzare con un'istruzione sola un gruppo di istruzioni. Tale struttura è particolarmente importante, rende il programma più corto e facile da scrivere e fornisce ad esso una struttura gerarchica. Anche Pascal è dotato di questo importante strumento ed oltre ad esso è fornito anche di uno analogo, la funzione, che supplisce pressoché alle medesime applicazioni, anche se in modo più limitato.

### STRUTTURA DEL PROGRAMMA

Questo programma è costituito di una parte dichiarativa, dove vengono definite tutte le variabili, le procedure e le funzioni che si utilizzeranno e di un corpo principale (main) che contiene le istruzioni di avvio.

Come prima cosa il programma presenta all'utente la spiegazione di ciò che è in grado di fare tramite la procedura *writelniziali()* e successivamente chiede all'utente di inserire i coefficienti dell'equazione tramite la sottoprocedura *inputIniziali()*. Se il valore di *a* risulta diverso da zero il programma procede all'applicazione del metodo delle tangenti chiamando la sottoprocedura *elabora()* mentre in caso contrario si arresta.

La procedura *elabora()* controlla il segno del coefficiente *a* del grado massimo della funzione. Se è positivo avvia tutta l'elaborazione chiamando *elaboraAPositivo()* altrimenti tramite *inversione\_coeficienti()* vengono appunto invertiti i segni dei coefficienti di input per poter utilizzare comunque e sempre *elaboraAPositivo()*. Ciò è motivato dal fatto che il programma non disegnando alcun grafico e che gli zeri di una funzione o della sua opposta sono gli stessi, permette di ottenere un risparmio notevole in termini di scrittura di codice sorgente senza obbligo di dover duplicare con minime differenze (segno di *a*) buona parte del corpo principale.

La sottoprocedura *elaboraAPositivo()* è costituita da una serie di istruzioni di decisione che determinano un diverso percorso esecutivo a seconda del segno del  $\Delta/4$  (delta quarti in quanto il coefficiente del primo grado è pari) della derivata prima della funzione.

Tali sottoprocedure sono:

con  $\Delta/4 > 0$

- Delta4Positivo\_1();*
- Delta4Positivo\_2();*
- Delta4Positivo\_3();*
- Delta4Positivo\_4();*
- Delta4Positivo\_5();*

con  $\Delta/4 < 0$

- Delta4Negativo\_1();*
- Delta4Negativo\_2();*
- Delta4Negativo\_3();*

con  $\Delta/4 = 0$

- Delta4Nullo\_1();*
- Delta4Nullo\_2();*
- Delta4Nullo\_3();*

Non dimentichiamo che tutte queste sottoprocedure si servono internamente di altre funzioni o sottoprocedure come *Xinnesco()* per la ricerca del primo valore di ascissa di innesco del metodo iterativo, *Xapprossimata()* per il calcolo della soluzione approssimata in base alla precisione voluta, *F()* per il valore della funzione in un determinato punto, *F1()* per il valore della derivata prima in un determinato punto, *massimo\_e\_minimo()* per la ricerca appunto di tali stazionari e della procedura *flesso()* per il calcolo del punto dove avviene il cambio di concavità.

La sottoprocedura più complessa in assoluto è *Delta4Positivo\_5()* che ricerca tramite l'ulteriore chiamata a *Delta4Positivo\_5\_sol2()* la terza radice nel caso esista.

Infine il programma fa uso di *nrlnteri()* e di *calcolaNrDecimali()* che permettono di quantificare (nel senso di ottimizzare) il numero dei caratteri di output rispettivamente per la parte intera e per la parte decimale rispettivamente in base alla radice trovata e alla precisione voluta.

## Testo Integrabile Del Programma

```
Program Calcolo_approssimato_di_una_equazione_di_III_grado;
```

```
uses crt;
```

```
var a, b, c, d : real;  
var xi, xf, xii, xff, xiii, xfff : real;  
var x1, x2, x3, y1, y2, y3 : real;  
var precisione, delta4 : real;  
var nrDecimali : integer;
```

```
const incremento = 1;  
const SX = 1;  
const DX = 2;
```

```
(* ----- *)  
function F(x : real) : real;  
begin  
  F := (a*x*x*x) + (b*x*x) + (c*x) + d;  
end;
```

```
(* ----- *)  
function FI(x : real) : real;  
begin  
  FI := (3*a*x*x) + (2*b*x) + c;  
end;
```

```
(* ----- *)  
procedure calcolaNrDecimali();  
begin  
  nrDecimali := 10;  
  
  if ( (precisione >= 0.1) and (precisione <= 0.9) ) then  
    nrDecimali := 2;  
  
  if ( (precisione >= 0.01) and (precisione <= 0.09) ) then  
    nrDecimali := 3;  
  
  if ( (precisione >= 0.001) and (precisione <= 0.009) ) then  
    nrDecimali := 4;  
  
  if ( (precisione >= 0.0001) and (precisione <= 0.0009) ) then  
    nrDecimali := 5;  
  
  if ( (precisione >= 0.00001) and (precisione <= 0.00009) ) then  
    nrDecimali := 6;  
  
  if ( (precisione >= 0.000001) and (precisione <= 0.000009) ) then  
    nrDecimali := 7;  
end;
```

## RELAZIONE: Metodo delle tangenti con il Pascal

---

```
(* ----- *)
function nrlnteri( x: real ) : integer;
begin
  nrlnteri := 10;    (* assegno un valore in ogni caso *)

  if ( abs(x) < 10 ) then
    nrlnteri := 1;

  if ( abs(x) < 100 ) then
    nrlnteri := 2;

  if ( abs(x) < 1000 ) then
    nrlnteri := 3;

  if ( abs(x) < 10000 ) then
    nrlnteri := 4;

  nrlnteri := nrlnteri + 1; (* +1 così tiene conto anche del segno eventuale *)
end;
```

```
(* ----- *)
procedure massimo_e_minimo();
begin
  x1 := ((-1)* b - sqrt( delta4 )) / (3*a);
  x2 := ((-1)* b + sqrt( delta4 )) / (3*a);

  y1 := F(x1);
  y2 := F(x2);
end;
```

```
(* ----- *)
procedure flesso();
begin
  x3 := (((-1)*b) / (3*a) );
  y3 := F(x3);
end;
```

```
(* ----- *)
function Xinnessco(x : real; mode : integer): real;
var xT : real;
begin
  xT := x;
  if ( mode = SX ) then
    begin
      while ( F(xT) >= 0 ) do
        xT := xT - incremento;
      end
    else
      begin
        while ( F(xT) <= 0 ) do
          xT := xT + incremento;
        end
      end
end;
```

## RELAZIONE: Metodo delle tangenti con il Pascal

---

```
    end;

    Xinnesco := xT;
end;

(* ----- *)
function Xapprossimata(x : real) : real;
begin
    Xapprossimata := x - F(x) / FI(x);
end;

(* ----- *)
procedure inversione_coeficienti();
begin
    a := (-1) * a;
    b := (-1) * b;
    c := (-1) * c;
    d := (-1) * d;
end;

(* ----- *)
procedure writelniziali();
begin
    ClrScr;

    Writeln('Questo programma e" in grado di fornire una soluzione');
    Writeln('approssimata di una equazione polinomiale intera di terzo');
    writeln('grado. ');
    writeln();

    Writeln('Stabiliamo che:');
    Writeln('A sia il coefficiente del terzo grado');
    Writeln('B sia il coefficiente del secondo grado');
    Writeln('C sia il coefficiente del primo grado');
    Writeln('D sia il termine noto');
    Writeln( );
    Writeln( );
end;

(* ----- *)
procedure inputiniziali();
begin
    Write('Inserisci ora il valore di A (0 se NON vuoi procedere): ');
    Readln(a);

    If ( a <> 0 ) then
        begin
            Write('Inserisci il valore di B : ');
            Readln(b);

            Write('Inserisci il valore di C : ');
```

## RELAZIONE: Metodo delle tangenti con il Pascal

---

```
    Readln(c);

    Write('Inserisci il valore del termine noto D : ');
    Readln(d);

    Write('Inserisci la precisione voluta: ');
    Readln(precisione);

    precisione := abs( precisione );    (* per essere sicuri nel caso negativo *)

    writeln();
    Writeln('Questa e" l" equazione che vuoi risolvere: ');
    writeln();
    Writeln(' ', a:5:2,'x^3 + ',b:5:2,'x^2 + ',c:5:2,'x + ',d:5:2);

    calcolaNrDecimali; (* in base alla precisione scelta *)
end;
end;
```

```
(* ----- *)
procedure Delta4Positivo_1( x : real );
begin
    writeln( );
    Writeln('Caso delta4Positivo_1');
    writeln();

    xi := Xinnessco( x, DX );
    xf := xi + 2*precisione;

    repeat
        xi := xf;
        xf := Xapprossimata( xi );
    until( abs(xf-xi)< precisione );

    Writeln('La soluzione approssimata e": ', xf:nrInteri(xf):nrDecimali);
end;
```

```
(* ----- *)
procedure Delta4Positivo_2( x : real );
begin
    writeln( );
    Writeln('Caso delta4Positivo_2');
    writeln();

    xi := Xinnessco( x, SX );
    xf := xi - 2*precisione;

    repeat
        xi := xf;
        xf := Xapprossimata( xi );
    until( abs(xf-xi)< precisione );
```

## RELAZIONE: Metodo delle tangenti con il Pascal

---

```
Writeln('La soluzione approssimata e": ', xf:nrInteri(xf):nrDecimali);
end;
```

```
(* ----- *)
procedure Delta4Positivo_3( x : real );
begin
  writeln( );
  Writeln('Caso delta4Positivo_3');
  writeln();

  xi := Xinnessco( x, SX );
  xf := xi - 2*precisione;

  repeat
    xi := xf;
    xf := Xapprossimata( xi );
    (*writeln( 'xi = ', xi:5:2 );
    writeln( 'xf = ', xf:5:2 ); *)
  until( abs(xf-xi)< precisione );

  Writeln('La soluzione approssimata e": ', xf:nrInteri(xf):nrDecimali);
  Writeln('L"altra soluzione e" x=',x2:nrInteri(x2):nrDecimali);
end;
```

```
(* ----- *)
procedure Delta4Positivo_4( x : real );
begin
  writeln();
  Writeln('Caso delta4Positivo_4');
  writeln();

  (* Writeln('La curva interseca l"asse delle ascisse in due punti,uno dei quali e"massimo
relativo ');*)

  (*writeln( ' elaboraPositivo_delta4>_4');
  readln(); *)

  xi := Xinnessco( x, DX );
  xf := xi + 2*precisione;

  repeat
    xi := xf;
    xf := Xapprossimata( xi );
    (*writeln( 'xi = ', xi:5:2 );
    writeln( 'xf = ', xf:5:2 ); *)
  until( abs(xf-xi)< precisione );

  Writeln('La soluzione approssimata e": ', xf:nrInteri(xf):nrDecimali);
  Writeln('L"altra soluzione e" x=',x1:nrInteri(x1):nrDecimali);
```

## RELAZIONE: Metodo delle tangenti con il Pascal

---

end;

```
(* ----- *)
procedure Delta4Positivo_5_sol2(x : real);
begin
  xff := x;
  If ( y3 <> 0 ) then
    begin
      repeat
        xii := xff;
        xff := Xapprossimata( xii );
      until( abs(xff-xii)< precisione );
    end;
end;
```

```
(* ----- *)
procedure Delta4Positivo_5( );
begin
  writeln();
  Writeln('Caso delta4Positivo_5');
  writeln();

  (*soluzione1*)
  xi := Xinnesso( x1, SX );
  xf := xi - 2*precisione;

  repeat
    xi := xf;
    xf := Xapprossimata( xi );
  until( abs(xf-xi)< precisione );

  Delta4Positivo_5_sol2( x3 );  (* soulzione centrale *)

  (*soluzione3*)
  xiii := Xinnesso( x2, DX );
  xfff := xiii + 2*precisione;

  repeat
    xiii := xfff;
    xfff := Xapprossimata( xiii );
  until( abs(xfff-xiii)< precisione );

  Writeln('La Soluzione1 e":', xf:nrInteri(xf):nrDecimali);
  Writeln('La Soluzione2 e":', xff:nrInteri(xff):nrDecimali);
  Writeln('La Soluzione3 e":', xfff:nrInteri(xfff):nrDecimali);
end;
```

```
(* ----- *)
```

---

## RELAZIONE: Metodo delle tangenti con il Pascal

---

```
procedure Delta4Negativo_1();
begin
  writeln();
  Writeln('Caso delta4Negativo_1');
  writeln();

  Writeln('La soluzione e": ', x3:nrInteri(x3):nrDecimali);
end;

(* ----- *)
procedure Delta4Negativo_2( x : real );
begin
  writeln();
  Writeln('Caso delta4Negativo_2');
  writeln();

  xi := Xinnessco( x, SX );

  repeat
    xi := xf;
    xf := Xapprossimata( xi );
  until( abs(xf-xi)< precisione );

  Writeln('La soluzione approssimata e": ', xf:nrInteri(xf):nrDecimali);
end;

(* ----- *)
procedure Delta4Negativo_3( x : real );
begin
  writeln();
  Writeln('Caso delta4Negativo_3');
  writeln();

  xi := Xinnessco( x, DX );

  repeat
    xi := xf;
    xf := Xapprossimata( xi );
    (*writeln( 'xi = ', xi:5:2 );
    writeln( 'xf = ', xf:5:2 ); *)
  until( abs(xf-xi)< precisione );

  Writeln('La soluzione approssimata e": ', xf:nrInteri(xf):nrDecimali);
end;

(* ----- *)
procedure Delta4Nullo_1();
begin
  writeln();
```

## RELAZIONE: Metodo delle tangenti con il Pascal

---

```
Writeln('Caso delta4Nullo_1');
writeln();
```

```
Writeln('La soluzione e": ', x1:nrInteri(x1):nrDecimali);
end;
```

```
(* ----- *)
```

```
procedure Delta4Nullo_2( x : real );
begin
```

```
  writeln();
  Writeln('Caso delta4Nullo_2');
  writeln();
```

```
  xi := Xinnessco( x, SX );
  xf := xi - 2*precisione;
```

```
  repeat
```

```
    xi := xf;
    xf := Xapprossimata( xi );
    (*writeln( 'xi = ', xi:5:2 );
    writeln( 'xf = ', xf:5:2 );      *)
  until( abs(xf-xi)< precisione );
```

```
  Writeln('La soluzione approssimata e": ', xf:nrInteri(xf):nrDecimali);
end;
```

```
(* ----- *)
```

```
procedure Delta4Nullo_3( x : real );
begin
```

```
  writeln();
  Writeln('Caso delta4Nullo_3');
  writeln();
```

```
  xi := Xinnessco( x, DX );
  xf := xi + 2*precisione;
```

```
  repeat
```

```
    xi := xf;
    xf := Xapprossimata( xi );
    (*writeln( 'xi = ', xi:5:2 );
    writeln( 'xf = ', xf:5:2 );      *)
  until( abs(xf-xi)< precisione );
```

```
  Writeln('La soluzione approssimata e": ', xf:nrInteri(xf):nrDecimali);
end;
```

```
(* ----- *)
```

```
procedure elaboraAPositivo();
begin
  delta4 := sqr(b) - 3*a*c;
```

## RELAZIONE: Metodo delle tangenti con il Pascal

---

```
If ( delta4 > 0 ) then
begin
    massimo_e_minimo;
    flesso;

    If ( (y1 < 0) and ( y2 < 0 )) then
        Delta4Positivo_1( x2 );
    If ( (y1 > 0) and ( y2 > 0 )) then
        Delta4Positivo_2( x1 );
    If ( (y1 > 0) and ( y2 = 0 )) then
        Delta4Positivo_3( x1 );
    If ( (y1 = 0) and ( y2 < 0 )) then
        Delta4Positivo_4( x1 );
    If ( (y1 > 0) and ( y2 < 0 )) then
        Delta4Positivo_5();
end;

If ( delta4 < 0 ) then (*la funzione cresce sempre con un cambio di concavità*)
begin
    flesso;
    If ( y3 = 0 ) then
        Delta4Negativo_1;
    If ( y3 > 0 ) then
        Delta4Negativo_2( x3 );
    If ( y3 < 0 ) then
        Delta4Negativo_3( x3 );
end;

If ( delta4 = 0 ) then (*la funzione cresce sempre..flesso orizzontale in x3.. *)
begin
    massimo_e_minimo;
    If ( y1 = 0 ) then
        Delta4Nullo_1;
    If ( y1 > 0 ) then
        Delta4Nullo_2( x1 );
    If ( y1 < 0 ) then
        Delta4Nullo_3( x2 );
end;

writeln( );
writeln( );
writeln( 'Premi INVIO per continuare' );
readln();
end;

(* ----- *)
procedure elabora();
begin
    if ( a < 0 ) then
        inversione_coefficienti;

    elaboraAPositivo;
```

## RELAZIONE: Metodo delle tangenti con il Pascal

---

end;

(\* ----- main ----- \*)

BEGIN

  repeat

    writeIniziali;

    inputIniziali;

    if ( a <> 0 ) then

      elabora;

  until( a = 0 );

END.

**BIBLIOGRAFIA**

- Peter Grogogno, Programmare in Pascal, FrancoMuzzio&c.editore, 1980, consultazione
- Brusamolin Mantovani Ciceri, Elementi di informatica,CEDAM,2002, consultazione
- Bergamini Trifone Barozzi ,Corso base blu di matematica, Zanichelli,2008,consultazione
-