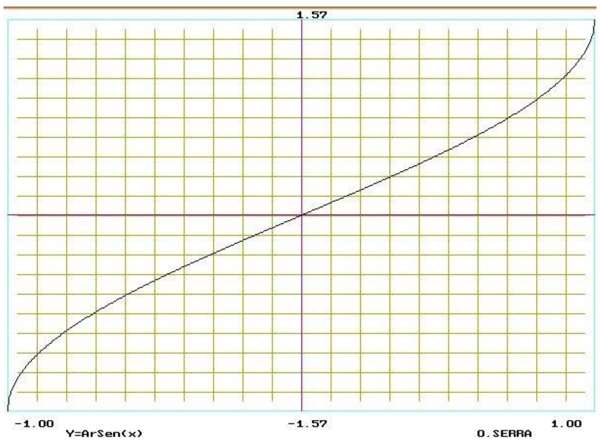
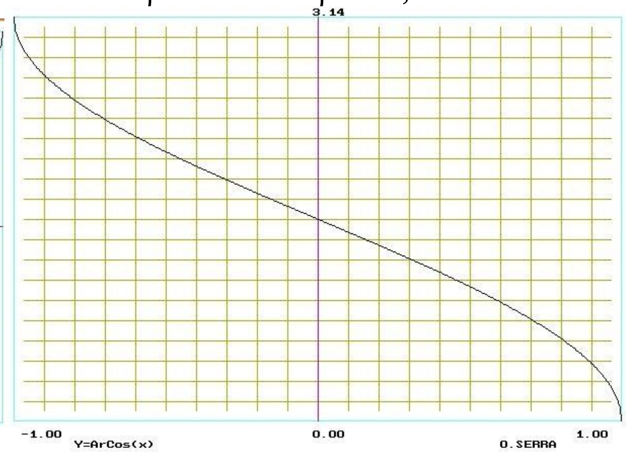
Ottavio Serra

Funzioni circolari inverse.

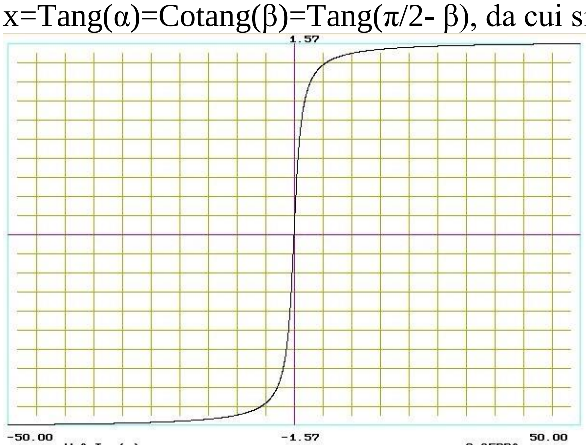
In questo articolo mostrerò come esprimere le funzioni circolari inverse in funzione dell'arcoseno e perché è più utile questa scelta rispetto a quella di esprimere le altre in funzione dell'arcotangente, come accade nei linguaggi usuali di programmazione, quali BASIC, PACAL, DELPHI, VISUALBASIC. Dimostro prima tre proprietà.

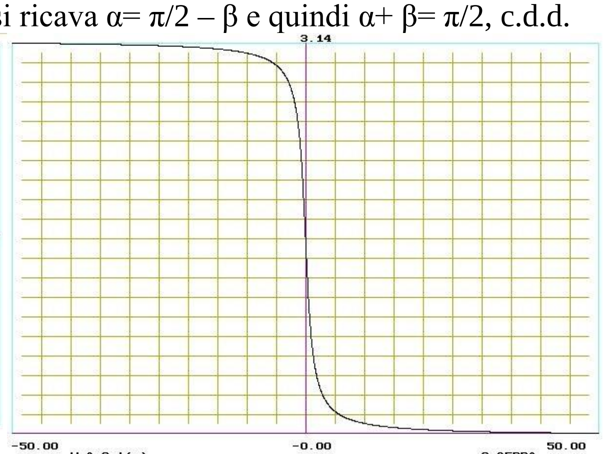
1) **ArcSen(x)+ArcCos(x)=π/2.**
   Infatti, posto α=ArcSen(x), β=ArcCos(x), segue
   x=Sen(α)=Cos(β)=Sen(π/2 – β), da cui si ricava α= π/2 – β e infine α+ β= π/2, c.d.d.





2) **ArcTang(x)+ArcCotang(x)=π/2.**
   Infatti, posto α=ArcTang(x), β=ArcCotang(x), segue
   x=Tang(α)=Cotang(β)=Tang(π/2- β), da cui si ricava α= π/2 – β e quindi α+ β= π/2, c.d.d.





3) **ArcTang(x)+ArcTang(1/x)=π/2 se x>0, -π/2 se x<0.**
   Questa identità si dimostra così: posto β=ArcTang(1/x), segue Tang(β)=1/x, Cotang(β)=x e poi si procede (quasi) come nel caso **2)** precedente. Si noti che sia la funzione somma del caso 2) sia quella di questo caso 3) hanno derivata zero, però mentre nel caso 2) l'annullarsi della derivata nel dominio garantisce che la funzione è costante, ora non lo garantisce: spiegare il perché.

4) **Esprimo ora** ArcoSeno, ArcoCoseno, ArcoCotangente in funzione di ArcoTangente.
   Posto y=ArcTan(x), sarà ArcCotang(x)= π/2 - y.
   Poi, da y=ArcSen(x) si ricava Sen(y)=x e infine

$y = ArcSen(x) = ArcTang\dfrac{x}{\sqrt{1-x^2}}$. *E qui succede il guaio, se x=1.*

(Sarà poi ArcCos(x)= π/2-ArcSen(x), non applicabile se x=1, oppure

$ArcCos(x) = ArcTang\dfrac{\sqrt{1-x^2}}{x}$; *qui il guaio succede, se x=0*).

5) **Conviene perciò esprimere le funzioni circolari inverse in funzione dell'arcoseno** (Nella mia "**Unit**" ho chiamato questa funzione *ArSen*; le altre *ArCos*, *ArTan* e *ArCot*).

Si tratta di trovare una formula per l'arcotangente (il resto è ovvio).

Posto y=ArTan(x), ho Tang(y)=x, $Sen^2(y)=x^2(1- Sen^2(y))$, $Sen^2(y).(1+x^2)=x^2$ e infine

$Sen(y) = \dfrac{x}{\sqrt{1+x^2}}$, da cui si ricava $y \equiv ArTan(x) = ArSen\dfrac{x}{\sqrt{1+x^2}}$.

Risulterà poi ArCos(x)= π/2-ArSen(x) e ArCot(x)= π/2-ArTan(x).

Come si vede, non si hanno denominatori tendenti a zero, al variare di x.

6) **Tutto sta nel disporre di un algoritmo** per calcolare la funzione ArSen(x) in modo efficiente.

Il modulo che ho realizzato, implementato in Pascal e in Delphi è il seguente:

```
Function arsen(x:reale):reale; {Serra}
 const eps=1e-18; {10^-18}
 var d,c,e:reale;
Begin d:=x;c:=sqrt(1-x*x);
 repeat c:=sqrt((1+c)/2);d:=d/c;e:=d/c
 until abs(e-d)<eps;
 arsen:=(e+d)/2
End;
```

Per giustificare l'algoritmo, si faccia riferimento alla figura seguente Fig. 1):

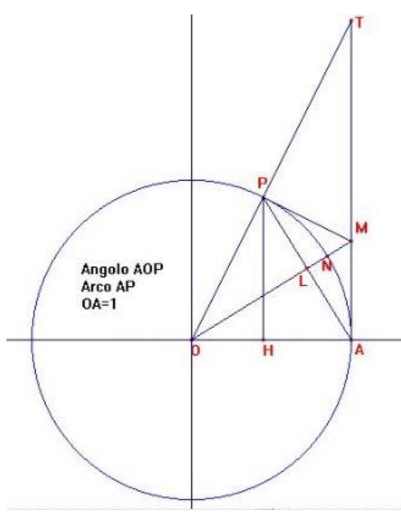


Fig. 1

Nel cerchio di centro O e raggio OA=1 si consideri l'angolo α=AOP, la cui misura in radianti è uguale all'ampiezza dell'arco corrispondente AP.

Se α è misurato in radianti, la lunghezza dell'arco AP sarà l=rα, uguale ad α, se r=1.

Noto, allora, PH=Senα=x, (la variabile d del modulo, inizialmente =x), sarà $Cos\alpha = OH = \sqrt{1-x^2}$ (la variabile c del modulo) e $AT = Tang\alpha = \dfrac{x}{\sqrt{1-x^2}}$.

Nel ciclo l'angolo AOP viene bisecato con ON, c assume il valore OL=Cos(α/2), (formula di bisezione), d=AP=2LP=2Sen(α/2), AT viene sostituito da AM+MP=2AM=2Tang(α/2) =e (la variabile "e" del modulo rappresenta un valore per eccesso, come "d" rappresenta un valore per difetto dell'arco AP).
L'arco AP= α =ArSen(x) viene approssimato ripetutamente dalle successioni di valori {d}ed {e} (per difetto e per eccesso) fino a quando la differenza e-d, in valore assoluto, risulta minore di $\varepsilon = 10^{-18}$. Vedi Fig. 2.

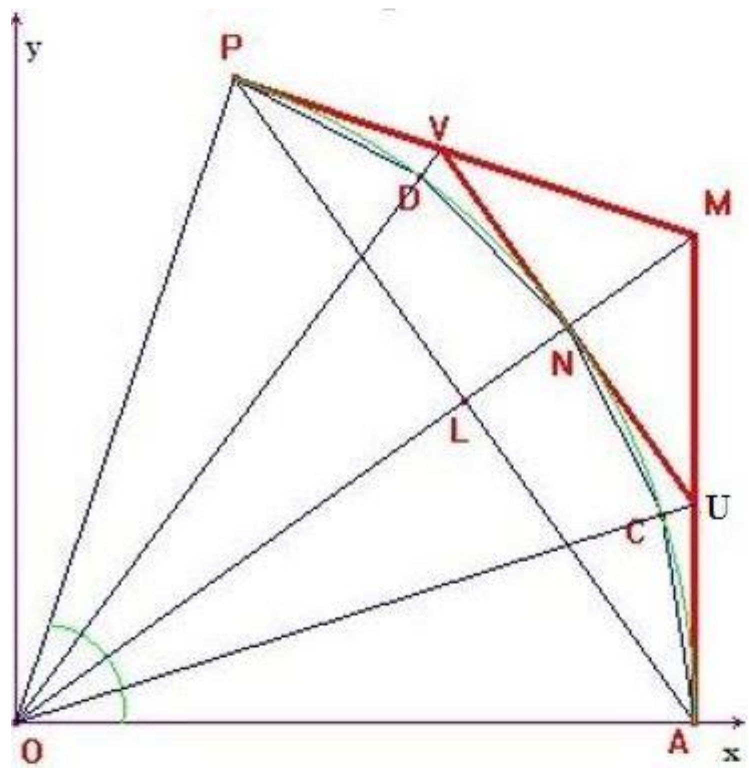


Fig. 2

L'arco AP è approssimato, alla prima iterazione, dalle spezzate AN+NP (per difetto) e AM+MP (per eccesso); alla seconda iterazione da AC+CN+ND+DP (per difetto) e AU+UN+NV+VP (per eccesso) e così via, per bisezione successiva degli angoli.

Il metodo è una generalizzazione del famoso algoritmo col quale Archimede approssima la lunghezza L della circonferenza con i perimetri di poligoni regolari inscritti e circoscritti. Come si sa, Archimede si ferma ai poligoni di 96 lati, ottenendo, per π=L/(2r), la disuguaglianza $3+\dfrac{10}{71} < \pi < 3+\dfrac{10}{70}$ *. Con la notazione decimale, 3,141< π < 3,143.*

Si veda anche il mio articolo "*Trigonometria nel biennio*" pubblicato sul 2° Annuario del Liceo Scientifico Scorza di Cosenza, anno 1989-90, nel quale vengono anche implementati seno e coseno senza Taylor.

7) Io non conosco i motivi per cui nei linguaggi di programmazione che ho citato in precedenza si sia preferito implementare nella libreria di sistema l'arcotangente, anziché l'arcoseno, che, come abbiamo visto, è più stabile per calcolare le altre funzioni circolari inverse; né conosco l'algoritmo utilizzato. Ho realizzato, perciò, una mia versione, sulla falsariga del modulo che ho utilizzato per l'arcoseno.
Si tratta di calcolare l'arco AP, ovvero l'angolo α=AOP, dato in input il valore x di AT (vedi Fig. 1). L'arco AP viene successivamente approssimato con la spezzata AM+MP =2AM=2Tang(α/2), poi con la spezzata AU+UN+NV+VP =4AU=4Tang(α/4) e così via (vedi Fig. 2).
Riporto qui sotto la porzione di codice:

```
 Procedure ArcTgSerra(x:reale;var t,s:reale);
  var t0:reale;n:naturale;Neg,Complemento:boolean;
 begin
  Neg:=x<0;If Neg then x:=-x;Complemento:=x>1;
 if complemento then x:=1/x;
 t:=x;n:=1;
 repeat t0:=t;n:=2*n;
  t:=t0/(Sqrt(1+t0*t0)+1);
until t0-t<eps; s:=t/Sqrt(1+t*t);
t:=n*t;s:=n*s;if complemento then
begin t:=Pi/2-t;s:=Pi/2-s
end;
if NEG then
begin t:=-t;s:=-s end;
end;TextColor(11);
WRITELN('   N° cicli=',n);TextColor(15);
end;
```

{**NOTA**: La variabile *Neg* è utilizzata per riportarci al 1° quadrante, la variabile *Complemento* per riportarci ad angoli minori di 45° sfruttando l'identità dimostrata al punto **3).** La variabile *t* inizializzata con x è la tangente dell'angolo corrente ottenuto con n bisezioni; *s* rappresenta il seno che approssima l'arco $\alpha/2^n$ per difetto, come t lo approssima per eccesso. L'arco (angolo) α alla fine sarà approssimato per difetto e per eccesso da $s.2^n$ e $t.2^n$. La variabile ε (*eps*) è dichiarata e assegnata in input nel blocco programma (*MAIN*).
Ho implementato Arcoseno e arcotangente in Turbo Pascal 7 e in Delphi.}