Appendice A

Il calcolo numerico di π

Il problema del calcolo delle cifre di π è stato oggetto di studio sin dall'antichità. Il valore di π ha attirato l'attenzione di molti matematici, dai tempi di Archimede ai nostri giorni e sono state utilizzate diverse metodologie che hanno condotto ad approssimazioni sempre piú accurate del numero π [1].

Di seguito riportiamo alcuni risultati classici poiché riteniamo utile, dal punto di vista didattico, evidenziare come risultino ancora attuali alcune questioni di calcolo numerico dell'antichità.

Uno dei primi risultati significativi è dovuto ad Archimede (250 a.c.) che riuscì a dimostrare che:

$$3.140845704225352112... = \frac{223}{71} = 3 + \frac{10}{71} < \pi < 3 + \frac{1}{7} = \frac{22}{7} = 3.142857142857142857...$$

approssimando la lunghezza della circonferenza di raggio unitario con il perimetro dei poligoni inscritti e circoscritti, arrivando ad utilizzare poligoni fino a 96 lati.

A partire da Leibniz (1688 d.c), il calcolo delle cifre di π si basa sull'approssimazione della funzione arcotangente. Attualmente, negli algoritmi numerici, il modo più accurato per calcolare un'approssimazione di π secondo la precisione del sistema è

$$\pi = 4 \cdot arctg(1)$$

[1] L'interesse che accomuna i matematici antichi e quelli moderni, nei confronti del calcolo dei decimali di π, è legato sia all'uso che ne hanno fatto le civiltà nel corso dei secoli (basti pensare che già i costruttori egizi stabilirono l'inclinazione delle pareti della piramide di Cheope in modo tale che il rapporto tra l'altezza e la larghezza della base corrispondesse a quello esistente tra il raggio e la circonferenza di un cerchio), che a quello che se ne fa attualmente in numerosi campi, tra i quali, ad esempio, la crittografia; d'altra parte, lo stimolo a determinare sempre più cifre di quello sviluppo infinito che rappresenta lo stesso π, è legato anche alle problematiche ancora aperte, che riguardano l'espressione e la sequenza dei decimali del numero. Ci si chiede, ad esempio, se ciascuna cifra si ripete infinitamente spesso, o se π è *semplicemente normale in base 10*, nel senso che ogni sua cifra appare ugualmente spesso, in senso asintotico, nel suo sviluppo decimale, oppure se è *normale in base 10*, cioè se ogni blocco di cifre di una fissata lunghezza appare ugualmente spesso, in senso asintotico, nel suo sviluppo decimale, o anche se π è *normale*, intendendo con ciò, che ogni blocco di cifre di una fissata lunghezza appare ugualmente spesso, in senso asintotico, nella sua rappresentazione in una qualsiasi base.

che fa uso dell'approssimazione della funzione arcotangente, considerata come funzione elementare (*built-in*) del compilatore.
Nella tabella seguente riportiamo una breve sintesi dello **sviluppo storico** del calcolo delle cifre di π:

Autore	**Anno**	**Risultato**	**Cifre corrette**
Re Salomone	975 a.c	3.0	1
Archimede	250 a.c.	3.14	3
Aryabhata	499 d.c.	3.1416	4
Fibonacci	1220	3.141818	4
Viete	1593	3.141592653	9
Newton	1665	3.1415926535897932	16
Sharp	1699	3.(72 cifre)	71
Machin	1706	3.(100 cifre)	100
Eulero	1755	3.(20 cifre)	20
Vega	1789	3.(140 cifre)	126
Vega	1794	3.(137 cifre)	136
Ferguson	1946	3.(700 cifre)	620

Tra gli autori indicati in tabella vale la pena evidenziare Eulero, che adottò per primo la lettera greca π e che ottenne il risultato in solo un'ora di calcoli!
Il valore ottenuto da Ferguson nel Luglio del 1946 rappresenta l'ultimo calcolo delle cifre di π eseguito senza usare il calcolatore.
Nella tabella seguente si riportano, invece, alcuni dei risultati ottenuti mediante l'uso del calcolatore.

Autore	**Anno**	**Cifre corrette**	**Macchina**
Ferguson	1947	710	Desk calculator
Ferguson e Wrench	1947	808	Desk calculator
Shants e Wrench Jr	1961	100 265	IBM 7090
Guilloud e Filliatre	1966	250 000	IBM 7030
Guilloud e Dichampt	1967	500 000	CDC 6600
Kanada e Ushiro	1983	10 013 395	Hitachi S-810/20
Kanada et al.	1987	134 214 700	NEC SX-2
Kanada e Takahashi	1999	206 158 430 000	Hitachi SR8000
Kanada et al.	2002	1 241 100 000 000	Hitachi SR8000/MP

In particolare il risultato raggiunto da Y. Kanada e Y. Ushiro nel 1983 è il primo ottenuto mediante *calcolatore parallelo*. Dal 1980 Yasumasa Kanada è uno dei maggiori esponenti, tra gli interessati al calcolo di π con un elevato numero di cifre. La maggior parte dei suoi calcoli è stata realizzata utilizzando calcolatori paralleli ed è basata su moderni algoritmi iterativi.

Le prime 151 cifre del numero π calcolate con un elaboratore IBM 704 sono riportate qui di seguito [2]:

$$
\begin{aligned}
\pi = 3.&141592653589793238562643383279\\
&592307816406286208998628034825\\
&342117067982148086513282306647\\
&093844609550582231725359408128\ldots
\end{aligned}
$$

A.1 *Metodo I - Archimede 240 a.C.*

Si consideri una circonferenza di centro l'origine e raggio unitario. Detta A_n l'area del poligono regolare con 2^n lati inscritto nella circonferenza, si ha:

$$\pi = \lim_{n\to\infty} A_n$$

Il metodo proposto da Archimede è quello di approssimare il valore di π con A_n.

Come si può anche notare dalla Figura A.1, per $n = 2$, A_2, l'area del quadrato inscritto nella circonferenza può essere calcolata come 4 volte l'area del triangolo AOB.
Quindi:

$$A_2 = 4 \cdot \frac{AO \cdot BH}{2} = 2^2 \cdot \frac{\sin(\beta_2)}{2}$$

essendo $AO = 1$ perchè raggio di una circonferenza unitaria, $\beta_2 = A\hat{O}B$ e
$BO = BH = \sin(\beta_2)$.
Per $n = 3$, A_3, l'area dell'ottagono inscritto nella circonferenza, A_3 può essere calcolata come 8 volte l'area del triangolo AOB, come illustrato in Figura A.2.

[2] L'ultimo risultato relativo al calcolo delle cifre di π risale al Settembre 2002, quando il Prof. Yasumasa Kanada dell'Università di Tokyo, *Information Technology Center*, insieme ad altri nove collaboratori, calcolò π con 1.241.100.000.000 cifre decimali, superando di più di sei volte il numero delle cifre del loro stesso *Guinness World Record*, costituito da 206.158.430.000 cifre decimali e raggiunto nel 1999. Il calcolo richiese circa 602 ore e fu realizzato con un calcolatore Hitachi (Hitachi SR8000).

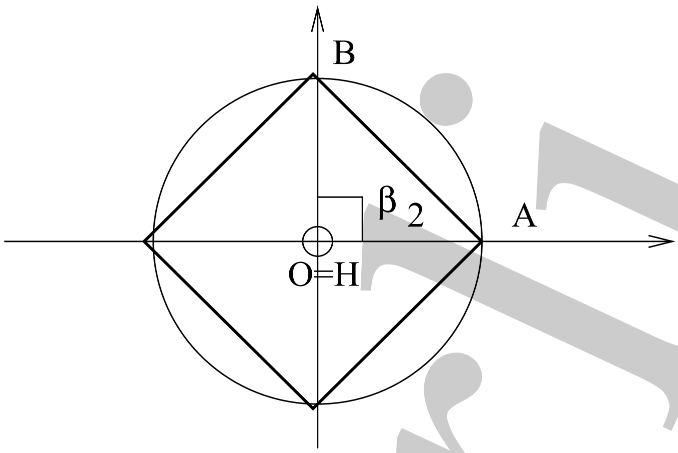

Figura A.1: Quadrato inscritto nella circonferenza di raggio unitario e centro l'origine. L'area del quadrato è $A_2 = 4 \cdot \frac{AO \cdot OB}{2}$.

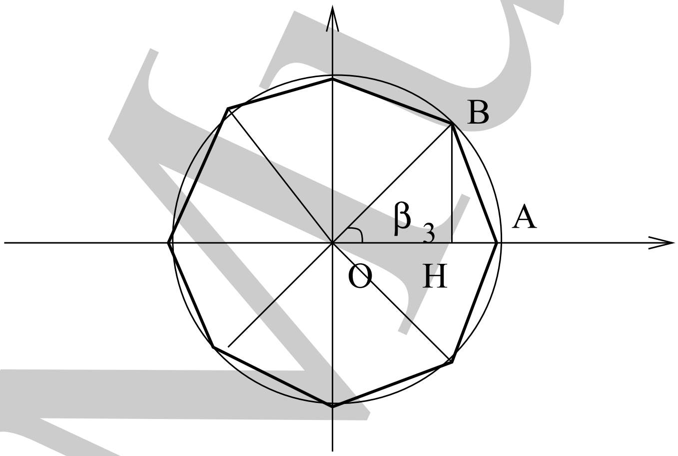

Figura A.2: Per $n = 3$ si ottiene un ottagono inscritto nella circonferenza. L'area dell'ottagono è $A_3 = 8 \cdot \frac{AO \cdot BH}{2}$.

Quindi:

$$A_3 = 8 \cdot \frac{AO \cdot BH}{2} = 2^3 \cdot \frac{1 \cdot \sin(\beta_3)}{2} \qquad (AO = 1,\ \beta_3 = A\hat{O}B)$$

In generale, per un fissato n, l'area del poligono con 2^n lati, A_n, è 2^n volte l'area dei triangoli che lo compongono, e quindi:

$$A_n = \frac{2^n AO \cdot BH}{2} = 2^{n-1} \sin(\beta_n) \qquad (AO = 1,\ \beta_n = A\hat{O}B) \tag{A.1}$$

Posto:

$$b_n = 2^{n-1} \tag{A.2}$$

la (A.1) diventa:

$$A_n = b_n \cdot \sin(\beta_n) \tag{A.3}$$

Il problema del calcolo di A_n è ricondotto al calcolo di $\sin(\beta_n)$.
Nel metodo proposto da Archimede, il valore di $\sin(\beta_n)$ viene calcolato a partire da quello di $\sin(2\beta_n)$ attraverso la formula di bisezione degli angoli:

$$\sin(\beta_n) = \sqrt{\frac{1 - \cos(2\beta_n)}{2}} = \sqrt{\frac{1 - \sqrt{1 - \sin^2(2\beta_n)}}{2}} \tag{A.4}$$

dove $0 \leq \beta_n < \pi/2$. Posto $s_n = \sin(\beta_n)$ si ha:

$$s_2 = \sin(\beta_2) = 1$$

e, in generale:

$$\sin(2\beta_n) = \sin\left(2\frac{2\pi}{2^n}\right) = \sin(\beta_{n-1}) = s_{n-1}$$

essendo $\beta_n = \frac{2\pi}{2^n}$. Dalla (A.4) si ottiene la formula ricorrente:

$$s_2 = 1, \qquad s_n = \sqrt{\frac{1 - \sqrt{1 - s_{n-1}^2}}{2}}\ , \ n > 2$$

Inoltre:

$$b_2 = 2, \qquad b_i = 2 \times b_{i-1}, \ \ i > 2$$

pertanto la (A.3), diventa:

$$A_n = b_n \cdot s_n = 2 \cdot b_{n-1} \cdot \sqrt{\frac{1 - \sqrt{1 - s_{n-1}^2}}{2}}$$

In conclusione, una stima di π si ottiene attraverso approssimazioni successive ottenute dallo schema iterativo seguente:

$$\begin{cases} b_2 = & 2\,, \quad s_2 = 1 \\ b_i = & 2 \cdot b_{i-1} \\ s_i = & \sqrt{\frac{1-\sqrt{1-s_{i-1}^2}}{2}}, \quad i > 2 \\ A_i = & b_i \cdot s_i, \quad i > 2 \end{cases} \tag{A.5}$$

A.1.1 Analisi della stabilità dell'algoritmo

Volendo implementare lo schema iterativo descritto dalla (A.5) in un sistema aritmetico a precisione finita $\Im$, bisogna tenere conto della propagazione dell'errore di *round-off*. Consideriamo, quindi, i primi i passi dell'algoritmo di Archimede:

$$\boxed{\begin{array}{l}
A_2 = b_2 \cdot s_2 = 2 \cdot 1 = 2 \\
A_3 = b_3 \cdot s_3 = 2b_2 \cdot \sqrt{\frac{1-\sqrt{1-s_2^2}}{2}} = 2^2 \cdot \sqrt{\frac{1}{2}} = 2^2 \cdot \frac{\sqrt{2}}{2} = 2 \cdot \sqrt{2} \\
A_4 = b_4 s_4 = 2b_3 s_4 = 2^3 \cdot \sqrt{\frac{1-\sqrt{1-s_3^2}}{2}} = 2^3 \cdot \sqrt{\frac{1-\sqrt{1-\frac{1}{2}}}{2}} = 2^2 \cdot \sqrt{2-\sqrt{2}} \\
A_5 = b_5 s_5 = 2b_4 s_5 = 2^4 \cdot \sqrt{\frac{1-\sqrt{1-s_4^2}}{2}} = 2^3 \cdot \sqrt{\frac{1-\sqrt{1-\frac{2-\sqrt{2}1}{4}}}{2}} = 2^2 \cdot \sqrt{2-\sqrt{2-\sqrt{2}}} \\
\vdots \\
A_i = b_i \cdot s_i = 2b_{i-1} \cdot s_i = 2^{i-1} \cdot \sqrt{\frac{1-\sqrt{1-s_{i-1}^2}}{2}} = 2^{i-2} \cdot \underbrace{\sqrt{2-\sqrt{2-...-\sqrt{2}}}}_{i-2 \text{ radicali innestati}} \quad , i > 2
\end{array}}$$

Si osserva che al passo i-esimo si moltiplica il termine precedente per $\mu = 2^{i-2}$ che è, quindi, anche il fattore di amplificazione dell'errore di round-off. Lo stesso fattore, in base 10, è:

$$\mu \simeq 10^{0.301 \cdot (i-2)}$$

Segue, allora, che l'algoritmo descritto è **instabile**.

A.1.2 Analisi dell'errore di troncamento analitico

Per dare una stima dell' errore di troncamento che si commette arrestando al passo n il processo iterativo descritto, si utilizza lo sviluppo in serie di Mc Laurin della funzione $\sin(x)$.
Lo sviluppo in serie di Mc Laurin della funzione $\sin(x)$ è dato da :

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots \tag{A.6}$$

Ricordando la (A.1) si ha:

$$A_n = 2^{n-1} \sin \frac{\pi}{2^{n-1}}$$

e quindi:

$$A_n = 2^{n-1} \left[\frac{\pi}{2^{n-1}} - \frac{1}{6} \frac{\pi^3}{8^{n-1}} + \frac{1}{120} \frac{\pi^5}{32^{n-1}} - \dots \right] = \pi - \frac{1}{6} \frac{\pi^3}{4^{n-1}} + \frac{1}{120} \frac{\pi^5}{16^{n-1}} - \dots$$

per l'errore di troncamento analitico si ha:

$$|e_n| = \mathcal{O}\left(\left|\frac{\pi^3}{6}\cdot\frac{1}{4^{n-1}}\right|\right) = \mathcal{O}\left(\left|\frac{2}{3}\pi^3\cdot\frac{1}{4^n}\right|\right) \approx \mathcal{O}\left(21\cdot\frac{1}{4^n}\right) = \mathcal{O}\left(\frac{1}{4^n}\right)$$

e:

$$|e_{n+1}| < \frac{1}{4}|e_n|$$

A.2 *Metodo II - Viete 1593*

Si consideri una circonferenza C di centro l'origine e raggio unitario. Indicata con $l(C)$ la lunghezza della circonferenza C, si ha:

$$l(C) = 2\cdot\pi$$

e:

$$\pi = \lim_{n\to\infty}\frac{P_n}{2}$$

essendo P_n il perimetro del poligono regolare con 2^n lati inscritto nella circonferenza. Il metodo proposto da Viete è quello di approssimare il valore di π con $P_n/2$. Per ogni fissato valore di n, si indichi con a_n la lunghezza del lato del poligono di 2^n lati e con p_n il semiperimetro dello stesso poligono, si ottiene:

$$p_n = \frac{2^n a_n}{2} = 2^{n-1}a_n \tag{A.7}$$

Ricaviamo una relazione che lega il lato a_n del poligono di 2^n lati con il lato a_{n+1} del poligono con 2^{n+1} lati.
Per $n = 2$, consideriamo i triangoli AKC e AOB, come illustrato nella Figura A.3. Gli angoli $A\widehat{K}C$ ed $A\widehat{O}B$ sono due angoli, rispettivamente alla circonferenza ed al centro, che insistono sullo stesso arco $(\widehat{AC})$ [3] per cui:

$$A\widehat{K}C = \frac{A\widehat{O}B}{2},\quad A\widehat{O}B = \frac{\pi}{4} \Rightarrow A\widehat{K}C = \frac{\pi}{8}$$

L'angolo $C\widehat{A}K$ è uguale all' angolo $A\widehat{K}C$, essendo angoli alla base di un triangolo isoscele (ACK); per cui se $\frac{a_2}{2}$ è un cateto del triangolo ABC, a_3 ne è l'ipotenusa, si ha:

[3] Ricordiamo il seguente risultato di geometria:

Teorema A.2.1. *In una circonferenza, l'angolo alla circonferenza è la metà del corrispondente angolo al centro che insiste sullo stesso arco.*

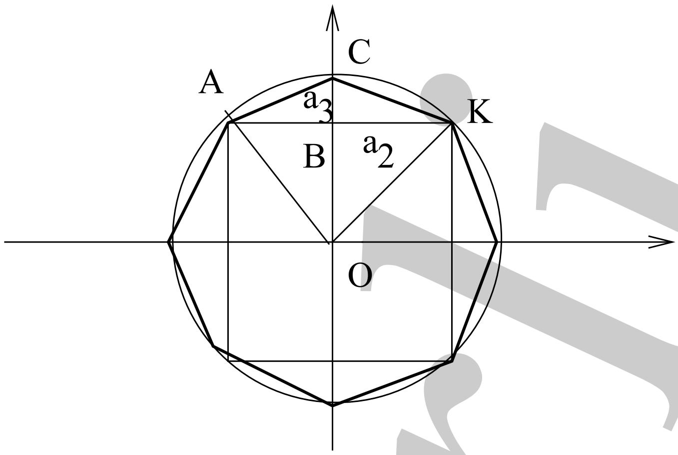

Figura A.3: Metodo di Viete per il calcolo di π. a_2 è il lato del quadrato inscritto e a_3 è il lato dell'ottagono inscritto.

$$a_3 \cdot \cos\left(\frac{\pi}{8}\right) = a_3 \cdot \cos\left(\frac{\pi}{2^3}\right) = \frac{a_2}{2}.$$

In generale, la relazione che lega il lato a_n del poligono di 2^n lati al lato a_{n+1} del poligono con 2^{n+1} lati è la seguente:

$$a_{n+1} \cdot \cos\left(\frac{\pi}{2^{n+1}}\right) = \frac{a_n}{2} \tag{A.8}$$

da cui si ottiene:

$$a_n = 2 \cdot a_{n+1} \cdot \cos\left(\frac{\pi}{2^{n+1}}\right) \tag{A.9}$$

Posto $\beta_i = \frac{\pi}{2^i}$ si ha quindi:

$$p_i = 2^{i-1}a_i = 2^{i-1} \cdot 2 \cdot \cos(\beta_{i+1}) \cdot a_{i+1} = 2^i \cos(\beta_{i+1}) \cdot a_{i+1} = p_{i+1} \cdot \cos(\beta_{i+1}) = p_{i+1} \cdot c_{i+1}$$

da cui:

$$p_{i+1} = \frac{p_i}{c_{i+1}}$$

Riferendoci sempre alla Figura A.3 il triangolo OAB è isoscele e retto in B e l'angolo $O\widehat{A}B = \frac{\pi}{4}$, per cui:

$$\frac{a_2}{2} = \cos\left(\frac{\pi}{4}\right) \quad \Rightarrow \quad a_2 = 2 \cdot 1 \cdot \frac{\sqrt{2}}{2} = \sqrt{2}$$

e:

$$p_2 = 2 \cdot a_2 = 2 \cdot \sqrt{2}$$

Ponendo $c_i = \cos(\beta_i)$ ed utilizzando la formula di bisezione per il coseno:

$$\cos(\beta_i) = \sqrt{\frac{1+\cos(2\beta_i)}{2}} = \sqrt{\frac{1+\cos(\beta_{i-1})}{2}}$$

$$c_1 = \cos\left(\frac{\pi}{2}\right) = 0$$

e:

$$c_i = \sqrt{\frac{1+c_{i-1}}{2}}, \quad i \geq 2$$

In conclusione, resta individuata la seguente formula ricorrente:

$$\begin{cases} p_2 = & 2\cdot\sqrt{2} \\ c_2 = & \sqrt{\frac{1}{2}} = \frac{\sqrt{2}}{2} \\ p_{i+1} = & \frac{p_i}{\sqrt{\frac{1+c_i}{2}}}, \qquad i \geq 2 \end{cases}$$

A.2.1 Analisi della stabilità dell'algoritmo

Essendo:

$$\frac{\sqrt{2}}{2} < c_i = \cos\left(\frac{\pi}{2^i}\right) < 1, \qquad i \geq 2$$

e quindi

$$1 < \frac{1}{c_i} < \sqrt{2}$$

cioè il fattore di amplificazione dell'errore nella formula di ricorrenza

$$p_{i+1} = \frac{p_i}{c_{i+1}}$$

è limitato e quindi il metodo di Viete per il calcolo di π è **stabile**.

A.2.2 Analisi dell'errore di troncamento analitico

Per determinare una stima dell'errore di troncamento analitico commesso ad ogni passo n si osserva che:

$$a_n = 2\cdot\sin\frac{2\pi}{2^{n+1}} \;\Rightarrow\; p_n = 2^{n-1}a_n \;=\; 2^n\sin\frac{\pi}{2^n}$$

Utilizzando lo sviluppo in serie di Taylor della funzione seno, calcolato in $\frac{\pi}{2^n}$, si ha:

$$p_n \;=\; 2^n\left[\frac{\pi}{2^n} - \frac{1}{6}\frac{\pi^3}{8^n} + \frac{1}{120}\frac{\pi^5}{32^n} - \ldots\right] = \pi - \frac{1}{6}\frac{\pi^3}{4^n} + \frac{1}{120}\frac{\pi^5}{16^n} - \ldots$$

da cui:

$$|e_n| < \left| \frac{1}{6}\frac{\pi^3}{4^n} + \frac{1}{120}\frac{\pi^5}{8^n} - \dots \right| = \mathcal{O}\left(\frac{1}{6}\pi^3 \cdot \frac{1}{4^n}\right) = \mathcal{O}\left(\frac{1}{4^n} \cdot 5.1677\dots\right) = \mathcal{O}\left(\frac{1}{4^n}\right)$$

A.3 *Metodo III - Leibniz 1688*

Il valore di π è ottenuto dallo sviluppo in serie di Mc Laurin della funzione $arctg(x)$ [4]:

$$arctg(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots$$

Per $x = 1$ si ha, infatti:

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots$$

indicata con q_i la somma parziale dei primi i addendi è possibile scrivere la formula ricorrente seguente:

$$q_1 = 1, \quad q_i = q_{i-1} + \frac{(-1)^{i-1}}{2i-1} \tag{A.10}$$

dove $i = 2, \dots, n-1$.

A.3.1 Analisi della stabilità dell'algoritmo

In un sistema aritmetico a precisione finita $\Im$, l'errore di round-off E_n, nel calcolo di una somma di n numeri x_i, si può stimare attraverso la formula seguente:

$$|E_n| \leq (n-1)u \sum_{i=1}^{n} |x_i| + \mathcal{O}(u^2), \quad \text{con} \quad \textit{u massima accuratezza}$$

e, quindi, cresce linearmente con il numero degli addendi, n. Il metodo di Leibniz per il calcolo di π è, dunque, **stabile**.

[4]La funzione $arctg(\cdot)$ fornita dal compilatore di un linguaggio di programmazione evoluto (Fortran, C, C++,...) non viene calcolata facendo uso dello sviluppo in serie di Mc Laurin (cfr. **Capitolo 8**).

A.3.2 Analisi dell'errore di troncamento analitico

Utilizzando la formula ricorrente espressa nella (A.10), si ricava che l'errore di troncamento analitico è:

$$|e_n| = \mathcal{O}((2n)^{-1})$$

A.4 *Metodo IV - Integrazione Numerica*

Il valore di π è calcolato utilizzando la formula di quadratura trapezoidale.
Infatti la funzione che definisce l'arco di circonferenza unitaria nel primo quadrante è:

$$f(x) = \sqrt{1 - x^2}$$

Per cui integrando $f(x)$ in $[0, 1]$ si ottiene $\frac{\pi}{4}$.

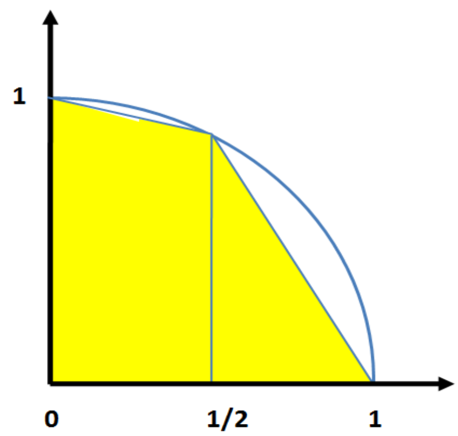

Figura A.4: Metodo di integrazione numerica.

Utilizzando la formula di quadratura trapezoidale composita per il calcolo numerico dell'integrale si ottiene:

$$\frac{\pi}{4} \simeq s_n = \frac{1}{2^n}[1 + 2b_1 + 2b_2 + \cdots + 2b_{n-1}]$$

dove:

$$b_i = \sqrt{1 - \frac{i^2}{n^2}}$$

A.4.1 Analisi della stabilità dell'algoritmo

In un sistema aritmetico a precisione finita $\Im$, l'errore di round-off E_n, nel calcolo della somma s_n, si può stimare nel modo seguente:

$$|E_n| \leq \frac{1}{2^{n-1}}(n-1)u\sum_{i=1}^{n-1}|b_i| + \mathcal{O}(u^2)$$

e, quindi, cresce linearmente con n, numero degli addendi. Il metodo di integrazione numerica è, dunque, **stabile**.

A.4.2 Analisi dell'errore di troncamento analitico

L'errore di troncamento analitico che si commette è:

$$|e_n| < k \cdot n^{-3/2}$$

essendo k una costante positiva.

A.5 Algoritmi implementati in FORTRAN

Di seguito vengono proposti alcuni programmi in Fortran relativi agli algoritmi descritti nel paragrafo precedente. I test hanno l'obiettivo di mettere in evidenza gli effetti della propagazione dell'errore sul risultato finale. Come vedremo, utilizzando il metodo di Archimede, per $n = 7$ il risultato è accurato a 4 cifre significative, ma già per $n = 14$, il risultato calcolato è completamente errato. Ciò a causa dell'instabilità dell'algoritmo. Con il metodo di Viete, invece, l'errore introdotto si mantiene limitato e si ottiene un valore di π corretto a 8 cifre significative per $n = 16$. Infine, se si utilizza il metodo di Leibniz, sebbene l'algoritmo sia stabile, a causa della lenta velocità di convergenza dello sviluppo in serie di McLaurin, solo per $n = 500000$ si ottiene un risultato accurato a 6 cifre significative.

Programma Fortran per il calcolo di π con il metodo di Archimede

```
====================================================
    program  CalcoloMetodoArchimede
C Fase  di dichiarazione delle variabili.
     integer n,step
     real  p,bi,si,error,pi,error1,n1
C Stampa a video del problema da risolvere
  write(*,*)    '   _________________________________________    '
  write(*,*)    '    Calcolo di PI                               '
  write(*,*)    '    (1)   pi=b(i-1)s(i-1)                       '
  write(*,*)    '    (2)   s(i)=sqrt(1-sqrt(1-s(i-1)2̂)/2)         '
  write(*,*)    '                                                '
  write(*,*)    '   _________________________________________    '

     write(*,*) 'Inserire n'
     read*,n
     print*,n
C Fase di  Calcolo
     bi=2.
     si=1.
     step=0
     p=0

5    continue
     p=bi*si
     bi=2*bi
     si=sqrt((1-sqrt(1-si**2))/2)
     step=step+1
C Terminazione della fase di Calcolo.
     if(step.lt.n) goto 5
     n1=n
     error=4**n1
     error=21*1/error
     pi=4*atan(1.)
     error1=abs(pi-p)/pi
     print*, 'PI calcolato=',p
     print*, 'Errore analitico stimato < di',error
C Confronto con la funzione di libreria
     print*, 'PI mediante funzioni built-in =',pi
     print*, 'Errore relativo',error1
     stop
     end
==================================================
```

Prova di esecuzione

Consideriamo alcune prove di esecuzione dell'algoritmo proposto da Archimede.

```
=======================================================
______________________________________
Calcolo di PI
(1)   pi=b(i-1)s(i-1)
(2)   s(i)=sqrt(1-sqrt(1-s(i-1)2̂)/2)
______________________________________
Inserire n
7
  PI calcolato=  3.1412857
  Errore analitico stimato < di  0.0012817383
  PI mediante funzione built-in=  3.1415927
  Errore relativo  0.000097747594
=======================================================

=======================================================
______________________________________
Calcolo di PI
(1)   pi=b(i-1)s(i-1)
(2)   s(i)=sqrt(1-sqrt(1-s(i-1)2̂)/2)
______________________________________
Inserire n
14
  PI calcolato=  2.828427(*)
  Errore analitico stimato < di  7.8231096E-8
  PI mediante funzioni built-in=  3.1415927
  Errore relativo  0.099683724
=======================================================
```

(*) Il risultato della seconda esecuzione è del tutto errato. Dunque, nel sistema aritmetico del calcolatore, che esegue le operazioni in registri a doppia precisione, per $n = 14$ l'errore di *round off* è del 100% (algoritmo instabile).

Eseguendo l'implementazione in doppia precisione l'instabilità dell'algoritmo comporta un errore di *round off* del 100% per $n = 34$; si ottengono, infatti, i risultati seguenti:

```
=======================================================
______________________________________
Calcolo di PI
(1)   pi=b(i-1)s(i-1)
(2)   s(i)=sqrt(1-sqrt(1-s(i-1)2̂)/2)
______________________________________
```

```
 Inserire n
 7
   PI calcolato=  3.141277250932773
   Errore analitico stimato < di  0.00128173828125
   PI mediante funzione built-in=  3.141592653589793
   Errore relativo  0.00010039578385817655
=======================================================

=======================================================
```

```
 Calcolo di PI
 (1)   pi=b(i-1)s(i-1)
 (2)   s(i)=sqrt(1-sqrt(1-s(i-1)^2)/2)
```

```
 Inserire n
 14
   PI calcolato=  3.1415926343386995
   Errore analitico stimato < di  7.82310962677002E-8
   PI mediante funzioni built-in=  3.141592653589793
   Errore relativo  6.127813415313911E-9
=======================================================

=======================================================
```

```
 Calcolo di PI
 (1)   pi=b(i-1)s(i-1)
 (2)   s(i)=sqrt(1-sqrt(1-s(i-1)^2)/2)
```

```
 Inserire n
 34
   PI calcolato=  4.
   Errore analitico stimato < di  7.115076756936123E-20
   PI mediante funzioni built-in=  3.141592653589793
   Errore relativo  0.2732395447351627
=======================================================
```

Programma Fortran per il calcolo di π con il metodo di Viete

```
================================================
      program  CalcoloMetodoViete
C Fase  di dichiarazione delle variabili.
```

```
      integer n,step
      real  p,bi,si,error,pi,error1,n1
C Stampa a video del problema da risolvere
   write(*,*)    '                                    '
   write(*,*)    '   ________________________________ '
   write(*,*)    '   Calcolo di PI                    '
   write(*,*)    '   (1)   pi=p(i-1)/ci               '
   write(*,*)    '   (2)   c(i)=sqrt(1-c(i-1))/2 )    '
   write(*,*)    '   ________________________________ '

      write(*,*) 'Inserire n'
      read*,n
C Fase di  Calcolo
      ci=0.
      per=2.
      step=1
      p=2.

5    continue

      step=step+1
      ci=sqrt((1+ci)/2)
      p=per/ci
      per=p
C Terminazione della fase di Calcolo.
       if(step.lt.n) goto 5
       n1=n
       error=4**n1
       print*,'errore=',error
       error=5.2*1/error
       pi=4*atan(1.)
       error1=abs(pi-p)/pi
       print*, 'PI calcolato=',p
       print*, 'Errore analitico stimato < di',error
C Confronto con la funzione di libreria
         print*, 'PI mediante funzioni built-in =',pi
         print*, 'Errore relativo',error1
         stop
         end
```
==

Prova di esecuzione

Consideriamo alcune prove di esecuzione dell'algoritmo proposto da Viete.
Per $n = 7$ si ottiene un'accuratezza di 4 cifre significative corrette e per $n = 16$ il risultato ha 8 cifre significative corrette.

```
=================================================
_______________________________
 Calcolo di PI
 (1)   pi=p(i-1)/ci
 (2)   c(i)=sqrt(1-c(i-1))/2 )
_______________________________
 Inserire n
           7
   PI calcolato=   3.1412773
   Errore analitico stimato < di  0.0003173828
   PI mediante funzioni built-in =   3.1415927
   Errore relativo  0.00010040378
=================================================
```

```
=================================================
_______________________________
 Calcolo di PI
 (1)   pi=p(i-1)/ci
 (2)   c(i)=sqrt(1-c(i-1))/2 )
_______________________________
 Inserire n
          16
   PI calcolato=   3.1415927
   Errore analitico stimato < di  1.2107193E-9
   PI mediante funzioni built-in =   3.1415927
   Errore relativo  0.
=================================================
```

Programma Fortran per il calcolo di π con il metodo di Leibniz

```
================================================
      program  CalcoloMetodoLeibniz

C Fase  di dichiarazione delle variabili.

   integer z
```

```
   real  p,i,error,pi,error1,n1,step,n

C Stampa a video del problema da risolvere

  write(*,*)   '                                              '
  write(*,*)   '  ____________________________________________ '
  write(*,*)   '   Calcolo di PI                              '
  write(*,*)   '    Sviluppo in serie di Taylor di arctg(x)   '
  write(*,*)   '  ____________________________________________ '
  write(*,*)   '                                              '

   write(*,*) 'Inserire ordine dello sviluppo'
   read*,n

C Fase di  Calcolo

    qi=1.
    step=1

5   continue

    step=step+1
    i=(-1)**(step-1)
    i=i/(2*step-1)
    qi=qi+i

C Terminazione della fase di Calcolo.

     if(step.le.n-1) goto 5
     p=4*qi
     n1=n
     error=1/n1
     error=2*error
     pi=4*atan(1.)
     error1=abs(pi-p)/pi
     print*, 'PI calcolato=',p
     print*, 'Errore analitico stimato < di',error

C Confronto con la funzione di libreria

     print*, 'PI mediante funzioni built-in =',pi
     print*, 'Errore relativo',error1
     stop
     end
================================================================
```

Prova di esecuzione

Consideriamo alcune prove di esecuzione dell'algoritmo proposto da Leibniz.

```
=======================================================================
___________________________________________
 Calcolo di PI
  Sviluppo in serie di Taylor di arctg(x)
___________________________________________
 Inserire ordine dello sviluppo
 70
   PI calcolato= 3.1273072
   Errore analitico stimato < di 0.028571429
   PI mediante funzioni built-in = 3.1415927
   Errore relativo 0.0045472365
=======================================================================

=======================================================================
___________________________________________
 Calcolo di PI
  Sviluppo in serie di Taylor di arctg(x)
___________________________________________
 Inserire ordine dello sviluppo
 200
   PI calcolato= 3.1365926
   Errore analitico stimato < di 0.01
   PI mediante funzioni built-in = 3.1415927
   Errore relativo 0.0015915858
=======================================================================

=======================================================================
___________________________________________
 Calcolo di PI
  Sviluppo in serie di Taylor di arctg(x)
___________________________________________
 Inserire ordine dello sviluppo
 700
   PI calcolato= 3.140165
   Errore analitico stimato < di 0.0028571428
   PI mediante funzioni built-in= 3.1415927
   Errore relativo 0.00045443524
=======================================================================

=======================================================================
```

```
 Calcolo di PI
  Sviluppo in serie di Taylor di arctg(x)

 Inserire ordine dello sviluppo
 500000
   PI calcolato= 3.141594
   Errore analitico stimato < di 0.000004
   PI mediante funzioni built-in = 3.1415927
   Errore relativo 3.7945495E-7
==========================================================
```

A causa della lenta convergenza dello sviluppo in serie di Taylor, solo con 500000 passi si ottengono 6 cifre significative corrette.

Bibliografia

[1] http://numbers.computation.free.fr/Constants/constants.html

[2] http://www.geocities.com/hjsmithh/Pi/index.html

[3] http://www.super-computing.org/

[4] O'Connor J. J. , Robertson E. F. -*A chronology of Pi* - September 2000, World Wide Web site at the adress:
http://www-groups.dcs.st-and.ac.uk/~history/HistTopics/Pi_chronology.html

[5] O'Connor J. J. , Robertson E. F. -*A history of Pi* - August 2001, World Wide Web site at the adress:
http://www-groups.dcs.st-and.ac.uk/~history/HistTopics/Pi_through_the_ages.html

[6] Sebah P. , Gourdon X. - π *and its computation through the ages* - April 2003, World Wide Web site at the adress:
http://numbers.computation.free.fr/Constants/Pi/piCompute.html