Calcolo di e

Per una conoscenza storica e generale di questo numero trascendente denominato Numero di Eulero (o di Nepero) si rimanda ai riferimenti. In questa sede ci dedicheremo invece a calcolarne il valore con un qualsiasi numero di decimali esatti sino ad un limite di 50000 utilizzando le formule indicate qui di seguito.
Dai testi di matematica si ha per questo numero la seguente formula:

$$\boldsymbol{e} = \lim \left(1+\frac{1}{n}\right)^{n} \qquad \text{per} \quad n \to \infty$$

Sviluppando $\boldsymbol{e}$ sotto forma di serie si ha :

$$\boldsymbol{e} = 1 + \sum_{n=1}^{\infty} \frac{1}{n!} = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \ldots\ldots\ldots\ . + \frac{1}{n!} + \ldots\ldots\ldots$$

Per un calcolo effettivo di $\boldsymbol{e}$ i termini della serie devono essere considerati in numero finito, dipendente questo numero da quante cifre decimali esatte di $\boldsymbol{e}$ si vogliono ottenere. Nella formula occorre sostituire quindi il segno di infinito con un numero N ben definito ed il meno elevato possibile compatibilmente con tutte le cifre che si vogliono ottenere . Useremo pertanto la seguente formula:

$$\boldsymbol{e} = 1 + \sum_{n=1}^{N} \frac{1}{n!} = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \ldots\ldots\ldots\ \ldots + \frac{1}{(N-1)!} + \frac{1}{N!} \qquad (1e)$$

Risulta però più opportuno utilizzare al posto della (1e) la seguente espressione di $\boldsymbol{e}$:

$$\boldsymbol{e} = 1+\frac{1}{1}\cdot\left(1+\frac{1}{2}\cdot\left(1+\frac{1}{3}\cdot\left(1+\frac{1}{4}\cdot\left(1+\frac{1}{5}\cdot\left(1+\frac{1}{6}\cdot\ldots\ldots\ldots\ldots\ldots\cdot\left(1+\frac{1}{(N-1)}\cdot\left(1+\frac{1}{N}\right)\right)\right)\right)\right)\right)\right) \qquad (2e)$$

che a prima vista appare più complicata della (1e), ma che è la stessa messa sotto altra forma. L'espressione (2e) si presta invece meglio ad essere programmata con un algoritmo di tipo iterativo veramente semplice e nello stesso tempo efficace, da sviluppare e trattare con operazioni aritmetiche a precisione multipla .
Utilizzeremo infatti non la Base di numerazione (Radix) 10, ma una base più elevata 10^{G} (con G scelto tra i valori 5,6,7) per accelerare l'esecuzione dei calcoli. Dalla formula indicata si può osservare che vengono implicate in effetti solo due tipi di *operazioni aritmetiche elementari* (*divisione e addizione*) che possono essere effettuate con l'aritmetica a doppia precisione disponibile. Nei riguardi dell'operazione di divisione si vuole precisare che per essa non si presentano particolari difficoltà di programmazione purché si mantengano adeguatamente piccoli tutti i valori numerici dei vari divisori.
Per rispettare l'obiettivo di calcolare tutte la cifre esatte richieste, nel tempo più breve possibile si dovrà realizzare un programma che persegua i seguenti obiettivi:

individuare il minimo numero N di termini della serie in grado di determinare il valore di $\boldsymbol{e}$ con tutte e sole le cifre decimali esatte richieste attraverso le seguenti considerazioni: nella formula (1e) compare la sommatoria col termine generico in valore assoluto sempre più piccolo man mano che il valore dell'indice n aumenta. Assumendo la somma di un certo numero di termini della (1e) come valore approssimato di $\boldsymbol{e}$, l'errore che si commette è minore in valore assoluto del primo termine della sommatoria che si trascura.
Basterà allora vedere quale termine della sommatoria è in valore assoluto minore del numero fissato della voluta approssimazione. Se si vuole pertanto ottenere il valore di $\boldsymbol{e}$ con DE cifre decimali esatte,

il che significa che l'approssimazione di ***e*** deve essere tale da dare luogo ad un errore minore di $\frac{1}{10^{DE}}$. Ciò equivale ad imporre che il termine generico $\frac{1}{n!}$ della (3) sia in valore assoluto non
$\frac{1}{10^{DE}}$. Ciò equivale ad imporre che il termine generico $\frac{1}{n!}$ della (3) sia in valore assoluto non
superiore a $\frac{1}{10^{DE}}$. Deve vale pertanto la seguente relazione $\frac{1}{n!} \le \frac{1}{10^{DE}}$ da cui $n! \ge 10^{DE}$.
Il valore minimo cercato di n sarà dato quindi da:

$$\boldsymbol{N\,!} = 10^{DE} \qquad \text{o} \quad \text{anche} \quad \mathbf{log}\,\boldsymbol{N\,!} = \mathbf{DE} \qquad (3e)$$

(con log = logaritmo in base10).

Per trovare il valore di N si possono seguire due metodi:

1° metodo

si ricorre alla formula di Stirling: $\log N\,! \approx (N - 0.5) \cdot \log N - N \cdot \log e + 0.5 \cdot \log 2\boldsymbol{\pi}$
e si impone che sia $(N - 0.5) \cdot \log N - N \cdot \log e + 0.5 \cdot 2\boldsymbol{\pi} = \mathrm{DE}$
isolando N si ottiene una espressione del tipo $N = f(N)$

$$\text{dove } f(N) = \frac{DE - 0.5 \cdot \log N + 0.5 \cdot \log 2\pi}{\log N - \log e}$$

utilizzando il metodo delle sostituzioni successive partendo da un valore iniziale opportuno N_0 si costruisce per iterazione una successione di valori:

$$N_1 = f(N_0)\ ,\ N_2 = f(N_1)\ ,\ N_3 = f(N_2)\ldots\ldots N_{k-1} = f(N_{k-2})\ ,\ N_k = f(N_{k-1})$$

per la presente $f(N)$ si può osservare che, pur partendo da un valore N_0 positivo praticamente qualsiasi, questa successione converge verso un valore tale per cui i due valori N_{k-1} *e* N_k diventano praticamente uguali; si avrà pertanto $N = N_k$.
Poiché N deve essere un intero ci si ferma nelle sostituzioni allorché si trova che la parte intera di N_k e quella di N_{k-1} coincidono ed il valore effettivo di N sarà il seguente $N = \lfloor N_k \rfloor + 1$

2° metodo :

poiché log k! = log 1 + log 2 +log3 + ….. log (k-1) + log k
si ricorre ad un algoritmo iterativo del tipo ad esempio
DO: k = k + 1 : x = x + log k : LOOP UNTIL x ≥ DE

All'uscita del ciclo iterativo, in corrispondenza dell'ultimo valore ottenuto di k, che indicheremo con *ko*, risulterà evidentemente che $\log ko! \ge DE$; tenendo presente la (3e) si ricava pertanto N = ko (dove *ko* è l'ultimo valore assunto da *k* alla fine del loop)
Il secondo metodo non implica formule complicate ed è anch'esso ugualmente efficace e veloce anche se in effetti il numero delle iterazioni è maggiore del numero delle sostituzioni successive del primo metodo.
Questo 2° metodo è quello che risulta utilizzato nei programmi dedicati al calcolo di **e** .
Qualche esempio: per avere il valore di **e** con *DE* = 100 cifre decimali esatte è ridondante considerare 1000 od anche100 termini nella serie (1e) ; calcolando il valore di N con uno qualsiasi dei due metodi si trova che è sufficiente un numero N di termini pari a 70.
Per avere 1000 decimali esatti sarà sufficiente porre N = 450 e per 10000 decimali esatti N = 3249
Occorre poi dare un certo margine per il calcolo del valore di N considerando al posto della (3e) la seguente relazione:

$$\boldsymbol{N\,!} = 10^{DE+G} \qquad \text{e quindi} \qquad \mathbf{log}\,\boldsymbol{N\,! = DE + G}$$

Il ciclo iterativo per trovare il valore di N sarà pertanto il seguente
DO: *n* = *n* + 1 : *n* = *n* + log *n* : LOOP UNTIL x ≥ DE +G
con N = *no* (dove *no* è l'ultimo valore assunto da *n* alla fine del loop)

L 'unico vettore implicato nei calcoli in aritmetica a precisione multipla in Base 10^G dovrà essere dimensionato con un numero di elementi pari a $\left\lfloor \frac{DE+G}{G} \right\rfloor +1$.

Si avranno pertanto i seguenti valori per i necessari termini N della serie
per DE = 100 N = 73 ; per DE = 1000 N= 452 ; per DE =10000 N = 3251

In calce a questo articolo viene riportato il listato di un programma in linguaggio Qbasic relativo al calcolo del numero di Eulero (o di Nepero). E' inoltre disponibile sempre su questo sito l'Eseguibile del Programma in questione per il calcolo effettivo del valore numerico di e con tutte le cifre decimali desiderate.

Programma in Qbasic relativo al calcolo del valore esatto di e con tutti i decimali desiderati

```
REM ------- NUMERO DI EULERO e CON TUTTI I DECIMALI ESATTI DESIDERATI ---
CLS : DEFDBL A-Z:
10 INPUT "QUANTE CIFRE DECIMALI ESATTE VUOI PER  e "; DE
If DE > 50000 then print " valore troppo grande: ridigita " : GOTO 10
PRINT
PRINT "scegliere G per la Base di numerazione 10^G tra i valori 5, 6 ,7"
INPUT "G"; G
PRINT : t1 = TIMER: P = 10 ^ G: DG = DE + G
DO: n = n + 1: x = x + LOG(n) / LOG(10): LOOP UNTIL x > DE + G
no = n: PRINT "Numero di termini della serie necessari: N ="; no
M = INT(DG / G) + 1:  IF DE / G <> INT(DE / G) THEN M = M + 1
'PRINT "numero elementi del vettore: M="; M
DIM C(M + 1): C(1) = 1
REM -------INIZIO CALCOLI TERMINI SOMMATORIA---------------
90 FOR H = n TO 1 STEP -1
        FOR k = 1 TO M
           Q = INT(C(k) / H): r = C(k) - Q * H
           C(k) = Q: C(k + 1) = C(k + 1) + r * P
        NEXT k
     C(1) = C(1) + 1
   NEXT H
PRINT
REM -------- PRESENTAZIONE RISULTATI ----------------------
f$ = "": ab$ = STR$(C(2)): lx = LEN(ab$)
IF G = 7 THEN PRINT "   e = 2.";
IF G = 6 THEN PRINT "e = 2.";
IF G = 5 THEN PRINT "e= 2.";
PRINT SPC(G - lx); ab$; SPC(1);
FOR k = 3 TO M - 1
  x$ = "": C$ = STR$(C(k)): la = LEN(C$) - 1: r$ = RIGHT$(C$, la)
  s = G - la: IF s = 0 THEN B$ = x$ + r$: GOTO X90
  z$ = "0": x$ = STRING$(s, z$): B$ = x$ + r$
X90: PRINT B$; SPC(1);
NEXT k
CE = G * (M - 2): PRINT : PRINT : PRINT "Nø DECIMALI DI e:"; CE
t2 = TIMER - t1: PRINT : PRINT "TEMPO DI CALCOLO:"; t2; "sec."
END
```