FUNZIONE INVERSA DEL FATTORIALE.

Risposta a una domanda comparsa su QUORA francese:

Comment puis-je résoudre x! = 40320 mathématiquement ?

I. Fattoriale

Non sono riuscito per ora a identificare una funzione inversa del fattoriale espressa in termini finiti, per cui penso che si debba ricorrere a metodi numerici. Ho il sospetto che se anche esistesse una formula in termini finiti, l'uso sarebbe abbastanza lungo e complicato da far preferire un semplice programma numerico come il mio.

Infatti, i programmi rilevanti in Small Basic non sono difficili da scrivere e, per n>5 danno risultati esatti fino a n circa eguale a 1000000!. Non li ho sperimentati per valori di n molto maggiori, perché il programma Mathematica, con cui calcolavo il $\mathrm{Log}_e(n!)$, a un certo punto si bloccava. In ogni caso il programma che ho sviluppato calcola "l'antifattoriale" di numeri di un milione di cifre fino all'ultima cifra, favorito in questo dal fatto che n è un intero.

Come annunciato, il programma l'ho sviluppato per Small Basic, disponibile in rete gratuitamente. Questo programma ha due limitazioni, delle quali la peggiore è che ho trovato difficile maneggiare la notazione scientifica. Fortunatamente, finché ci si accontenta di calcolare l'antifattoriale di numeri di un milione di cifre, tutto funziona, *purché si immetta direttamente il logaritmo (naturale) del fattoriale.*

Il problema può essere che il logaritmo naturale dell'antifattoriale di numeri di un milione di cifre debba essere calcolato con software specializzata, come Mathematica. Simpatizzo, ma in quanto segue io suppongo che il logaritmo naturale dell'antifattoriale sia noto e sia immesso come unico dato necessario. Naturalmente, non garantisco nessun rigore matematico.

I.1. Approssimazione basata sulla formula di Stirling.

Ciò posto, utilizzo la formula di Stirling, solo i primi cinque termini, ma si potrebbe continuare:

$$\ln(n!) \sim n\ln n - n + \frac{1}{2}\ln(2\pi n) + \frac{1}{12n} - \frac{1}{360\,n^3}$$

La formula può essere scritta nel modo seguente:

$$\ln(n!) \sim n(\,H(n))$$

In cui

$$H(n) = \ \ln n - 1 + \frac{1}{2n}\ln(2\pi n) + \frac{1}{12n^2} - \frac{1}{360\,n^4}$$

Dopodiché, le varie iterazioni di n si susseguono:

$$n_i = \frac{\ln(n_0!)}{H(n_{i-1})}$$

Per H(0) si può utilizzare come "seme" un numero compreso fra 2 e 10.

Il sistema funziona male con n <= 4, perché Small Basic sembra confondersi per motivi a me ignoti. Ma non credo che sia un grande problema.

I.2 Domanda su QUORA(fr): Trovare x dato x! = 40320.

Per rispondere alla domanda su Quora francese (trovare x per x! =40320), è certo più rapido provare con i primi numeri naturali, perché, per essere un fattoriale, 40320 è un numero piuttosto piccolo. Immagino che i valori 3! = 6, 4! = 24, 5! = 120 siano noti a memoria. Bisogna provare solo con 7 e 8 e ci si arriva. Vediamo che cosa si ottiene con la mia approssimazione "asintotica".

Per incominciare: log x! = 10.604603
Seme, o approssimazione F (0) = 2
Si ottiene x (0) = **5.302301**
Da cui : F (x (0)) = 1.00227
x(1) = 10.604603/1.11009= **10.5806**

Risultato del programma (il cui Listing è in appendice)

Input logaritmo naturale del fattoriale: 10.604603
Incidentalmente , il logaritmo decimale del fattoriale, la cui caratteristica ci dà il numero di cifre , è **4,**605…
Vogliamo allora l'"Antifattoriale " di (exp^10.604603)
Input seme > 1: 2
Seme: 2
Zero order: 5,3023015
1 order: H = 1,0022717820478358805792849772; x = 10,580566259515694213117730286
2 order: H = 1,5583728519032299913504300698; x = 6,8049202647804546025821285878

3 order: H = 1,1958202562410637417240456371; x = 8,8680576739304145348895123 84
4 order: H = 1,4105232043478182524173564660; x = 7,5182052782344947363301460506
5 order: H = 1,2755852814211522231037741094; x = 8,313519412975056755463746787
6 order: H = 1,3573549618866209773446435350; x = 7,8126969715131862073176908471
7 order: H = 1,3066789004434996175046212680; x = 8,115691618193799464846022317
8 order: H = 1,3376550048287261324282611500; x = 7,927756380919620972469381084
9 order: H = 1,3185594613466213028539307825; x = 8,042567143062101845236085518
10 order: H = 1,3302699790637575819828799350; x = 7,9717675110307359491024 03315
11 order: H = 1,3230653777015042769882696456; x = 8,0151768602870175112225 36234
12 order: H = 1,3274891213985956990778730233; x = 7,9884669705069706731933 93022
13 order: H = 1,3247695861932399334281695724; x = 8,004865986146771510903 01291
14 order: H = 1,3264402035926967256356720548; x = 7,9947840628451742304341 25524
15 order: H = 1,3254134707887773112629520752; x = 8,0009772299122708426327 49668
16 order: H = 1,3260443063054702224819023877; x = 7,9971709463809593190042 34302
17 order: H = 1,3256566475100248907020568773; x = 7,9995095411157780892734 92033
18 order: H = 1,3258948450000355973716026896; x = 7,9980724263240613848557 00919
19 order: H = 1,3257484746992871186173643558; x = 7,9989554597868867579605 39867
20 order: H = 1,3258384143907578760800985975; x = 7,9984128419396944595330 29959
21 order: H = 1,3257831480784766277175529087; x = 7,9987462620638811795975 88987
22 order: H = 1,3258171077144738122947894849; x = 7,9985413812323449274616 24138
23 order: H = 1,3257962402543075800757008845; x = 7,998667274818850203942 4377
29 order: H = 1,3258090627831725073731178739; x = 7,9985899159103230508068 25942

Risultato arrotondato: 8

Il programma si arresta normalmente a 30 iterazioni, oppure quando il valore assoluto della differenza tra le due ultime approssimazioni, x(i) e z(i-1) è inferiore a 0.0001. Qui vediamo che il risultato non si avvicina ulteriormente a 8. Con cento approssimazioni non si va oltre 7.9985….Poco male, il fattoriale si riferisce solo a numeri interi e ovviamente l'unico intero nelle vicinanze è 8.

Il problema, probabilmente, risiede nel fatto che l'approssimazione di Stirling vale per x molto grandi. Per cui vale la pena provare con numeri più grandi.

I.3 Trovare x dato x ! = 265252859812191058636308480000000.

Proviamo quindi con un numero un po' più eccitante, come
x ! = 265252859812191058636308480000000.

Troviamo con altro programma log (x!) = 74.65823635

Risultati:

Input logaritmo naturale del fattoriale: 74.65823635
Incidentalmente , il logaritmo decimale del fattoriale, la cui caratteristica ci dà il numero di cifre , è 32,42…
Vogliamo allora l'"Antifattoriale " di (exp^74.65823635)
Input seme > 1:4
Seme: 4

Zero order: 18,6645590875
1 order: H = 2,0546590671424444513581366962; x = 36,336070321307534180107689133
2 order: H = 2,6676839816126764808511247469; x = 27,986162103379042140738714889
3 order: H = 2,4242818781934616581317916051; x = 30,796021296679490608345181375
4 order: H = 2,5130554214241924469985099321; x = 29,708153554245879740197325397
5 order: H = 2,4796266074196770622985372909; x = 30,108660766344198894877175396
6 order: H = 2,4920657673177942177720296784; x = 29,958373221568113702325593781
7 order: H = 2,4874164182971607359510820381; x = 30,014369850107223890742273564
8 order: H = 2,4891513110771963016240112400; x = 29,993450385180146470139925257
9 order: H = 2,4885035390610352299171217176; x = 30,001257855622790950340379999
10 order: H = 2,4887453474084980363570824715; x = 29,998342911114134099812933326
11 order: H = 2,4886550744052966532086933985; x = 29,999431065327830557790558124
12 order: H = 2,4886887744514121042976023811; x = 29,999024834456088992910828291
13 order: H = 2,4886761936494776976003319117; x = 29,999176486081411205075094929
29 order: H = 2,4886808902562922375973437301; x = 29,999119872018408291008118068
Numero arrotondato: 30

30 è la risposta esatta, e il suo fattoriale è enorme, con più di trenta cifre, ma 14 iterazioni sono sufficienti a darci un numero che differisce dall'originale per 0.0008.

I.4. Un numero ancora più grande. Trovare x se log(x!)= 1.281551838E+07 (trovato col programma MATHEMATICA).

Quante cifre ha questo mostro? Facile, si prende la caratteristica del logaritmo decimale, che vale
5565708, quindi il numero x! ha 5 milioni e mezzo di cifre.

Input logaritmo naturale del fattoriale: log(x!)= 1.281551838E+07
Vogliamo allora l'"Antifattoriale " di (exp^1.281551838E+07)
Input seme > 1: 10
Seme: 10
Zero order: 1281551,838
1 order: H = 13,063588481294320640068780245; x = 981010,56982547093915723665 07
2 order: H = 12,796346484467575913120588788; x = 1001498,2319801745960361426712
3 order: H = 12,817015487398415580464459433; x = 999883,19375911751654122064 13
4 order: H = 12,815401575406607717815048474; x = 1000009,1143919841291592790051

5 order: H = 12,8155275018972576054216912 51; x = 999999,2882152329465187735319
6 order: H = 12,815517675833879451903593165; x = 1000000,0549463656763500785809
7 order: H = 12,815518442559559577873904048; x = 999999,9951184526901293824773
8 order: H = 12,815518382732098095487696629; x = 999999,9997868132982299815759
9 order: H = 12,815518387400463878060474997; x = 999999,9994225388584096436154
29 order: H = 12,815518387036166548051642911; x = 999999,9994509651240351510398

Numero arrotondato: 1000000, risposta esatta. Il nostro numero con 5.5 milioni di cifre è 1000000!, e per domare questo mostro bastano 10 approssimazioni. Francamente, il fatto che l'approssimazione di Stirling, pur troncata, sia così precisa e il numero di iterazioni necessarie in un programma così semplice sia così basso, mi lascia stupefatto.

II. Funzione Gamma.

Il metodo funziona anche per valori non interi (ma reali e positivi) di n!, cioè per la funzione di Eulero Γ(n+1), ***fino a due decimali***, a partire da n =4 per sicurezza. Non mi si chieda il perché. Naturalmente non si considererà il numero arrotondato dato per ultimo dal programma, che si applica solo se abbiamo a che fare con il fattoriale di un intero.

E se volessimo Γ(1.31)? Qui ci verrebbe in aiuto la relazione funzionale della funzione Gamma:

$$\Gamma\,(z+1) = z\,\Gamma(z),$$

per cui, Γ (4.31)= 4.31 Γ (4.31)= 4.31*3.31 *Γ (2.31) = 4.31*3.31* 2.31* Γ (1.31).

Ma torniamo a un problema a cui il nostro programma si presta.

Supponiamo di sapere che Γ(x+1) = $1.217117692772869 \times 10^{42}$, da cui Log(Γ (x+1))= 96.9051.

Input logaritmo naturale del fattoriale: 96.9051.
Incidentalmente , il logaritmo decimale del fattoriale, la cui caratteristica ci dà il numero di cifre , è 42,085
Vogliamo allora l'"Antifattoriale " di (exp^96.9051)
Input seme > 1:4
Seme: 4

Zero order: 24,226275
1 order: H = 2,2914179702341733533547779461; x = 42,290451265901830739407596724
2 order: H = 2,8106789690818265807210040123; x = 34,477470058295663901991103185
3 order: H = 2,6184575109960402462349110497; x = 37,008467616164631521104441702
4 order: H = 2,6849060514135065300041611104; x = 36,092547800316122048205686879
5 order: H = 2,6613719262749493976324086345; x = 36,411708954800413174611298955
6 order: H = 2,6696363476458953351761015961; x = 36,298988843724584048835973264
7 order: H = 2,6667254336632348422915546687; x = 36,338611683349467125699277137
8 order: H = 2,6677496412480559067404174448; x = 36,324660493502978379464169246
9 order: H = 2,6673891390501815267779936002; x = 36,329569833408895318506357247
10 order: H = 2,6675160125949035515119793566; x = 36,327841910771794770586040258
11 order: H = 2,6674713592278818140644856980; x = 36,32845003743539984652490798
12 order: H = 2,6674870748044052837462338975; x = 36,328236007331211160171712182
29 order: H = 2,6674815437376656702179865435; x = 36,328311334524518534421324723

Troviamo un valore x = 36.3283 etc., ma io ero partito dalla funzione Γ (z+1) = Γ (37.3291)

Dobbiamo ricordare che x! = Γ (z+1). Però, come avevo annunciato, l'approssimazione non va oltre due cifre decimali. Aumentando il numero di iterazioni, il risultato, per le prime due cifre decimali, non cambia. Probabilmente le cose migliorano aggiungendo termini alla formula di Stirling, che ne ha infiniti…

III. Listing:

```
TextWindow.WriteLine("Inverse factorial  (daino equinoziale)")
start:
TextWindow.Write("Input natural logarithm of factorial: ")
lfac= TextWindow.Read( )
TextWindow.WriteLine("Incidentally, the decimal logarithm, giving the
number of figures, is "+ lfac/Math.NaturalLog(10.))
TextWindow.WriteLine("Inverse factorial  of "+ lfac)
TextWindow.Write("Input seed > 1 ")
seed= TextWindow.Read( )
TextWindow.WriteLine("Seed: "+ seed)
x0 = lfac/seed
TextWindow.WriteLine("Zero order: "+ lfac/seed)
Counter =1
x=x0
While (Counter <30)
  x1= lfac/(Math.NaturalLog(x)-1 + (0.5*Math.NaturalLog(x) +0.921894)/x
+1/(12*Math.Power(x,2))-1/(360*Math.Power(x,4)))
  If (Math.Abs(x-x1)< 0.00001) Then
    Counter = 29
    EndIf
 x=x1
  TextWindow.WriteLine(Counter + " order: "+ x)
  Counter = Counter +1

EndWhile
TextWindow.WriteLine("Round number: "+ Math.Round(x))
Goto start
```