Unit n°101

e^{π} algorithmics

The decimal representation of

$$e^{\pi} = 23.140692632779269005729086367948547380266106242600211993445046409524342350690452783516971997067549229\ldots$$

(Gelfond's constant [1,2,3])
can be computed efficiently using the algorithm consisting of

** the recursion equation

$$z(t+1) = (z(t))^2 + \sqrt{(z(t))^4 - 1} \; ; \; t=0,1,2,3,... \quad (1)$$

in implicit representation

$$\begin{vmatrix} 2(z(t))^2 & z(t+1) \\ z(t+1) & z(t+1) \end{vmatrix} = 1 \; ; \; t=0,1,2,3,\ldots \quad (2)$$

with the initial value

$$z(0) = \frac{\sqrt[4]{2}+1}{\sqrt[4]{2}-1} = 11.5704270... \quad (3)$$

** and the downstream output transformation

$$y(t) = \sqrt[2^t]{2z(t)} \quad (4)$$

$$e^{\pi} = \lim_{t\to\infty} y(t) \quad (5)$$

The execution records for z(t) and y(t)

z(0) = 11.570427015766490403162879294
4732075678854249851 09…

z(1) = 267.74582776889478038756496498
79808002860037333 6729…

z(2) = 143375.65656832664987338822389
866690262137839051458…

z(3) = 4.11131577927974976374834529 75
7837739216964589818 62…* 10^{10}

z(4) = 3.38058348739093121536368966 88
93224386425464551250 4…* 10^{21}

z(5) = 2.28566894304404607831509744 06
50319073458057368217 5…* 10^{43}

y(0) = 23.14069263valid

y(1) = 23.140692632779269005 7valid

y(2) = 23.140692632779269005729086367 9
485473802661 06valid

y(3) = 23.140692632779269005729086367 9
485473802661062426002119934450
464095243423506904527835 1697valid

show

1* per recursion step in equ. (1) we obtain a doubling of valid digits in y(t) $\rightarrow e^{\pi}$,

2* the present algorithm equs (1,3,4) has – despite its very fast convergence – a serious disadvantage: its huge intrinsic numerical disbalance, z(t) increases exponentially and equ. (4) reduces these values exponentially to 23.14…

Problems

1* Modify the algorithm equs (1,3,4) so that the ride on the big dipper 2* is significantly reduced or completely eliminated.

2* Discuss the possibility to invert the algorithm via equ. (2): $y(\infty) \rightarrow z(0)$.

3* Discuss the function $y(\infty) = f(z(0))$.

4* Read Unit 1, p. 12
Convergence acceleration via algorithmic model „Engine + Load“.
Engine = fast converging dynamic input follower, load = slow converging numerical algorithm.
Try to distill from our very fast e^{π} algorithm
** by a suitable normalization
** a high speed dynamic input follower engine.

5* Find a e^{π}-similar algorithm for $\sqrt[\pi]{e}$.

References

[1] Weisstein, Eric W. "Gelfond's Constant."From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/GelfondsConstant.html

[2] https://en.wikipedia.org/wiki/Gelfond's_constant

[3] https://oeis.org/A039661 , http://oeis.org/A058287

Links

Links to Deb 16

http://www.algorithmicschannel-dresden-guilin.net/wordpress/wp-content/uploads/admin/Deb/deb_16.pdf

Links to Unit 60

http://www.algorithmicschannel-dresden-guilin.net/wordpress/wp-content/uploads/Unit-60.pdf