Truncated exponential

`expo.1.n`

February 19, 1998

Ring: $\mathbb{R}[x]$
Equations:

$$\sum_{i=0}^{n} \frac{x^i}{i!}$$

Supplied by D.A. Bini (Oct. 1996).

1 Characteristics

- The polynomial is obtained by truncating the series of e^x.
- The roots of this polynomial are located along a curve.
- Large values of n, say $n = 500, 1000, 2000$, constitute an interesting test for polynomial root-finders.

2 Problem

Newton-isolate all the roots.

2.1 Solution

Algorithm based on simultaneous iterations (Aberth's method) with cluster analysis (see [?]).

Program in Fortran 90 based on the multiprecision package MPFUN, by D. Bailey.

Machine used: IBM Risc 6000.

Time for Newton-isolating all the roots: for $n = 20$, 1.9 seconds; for $n = 50$, 12.8 seconds; for $n = 100$, 168 seconds.

Dario Andrea Bini, Dipartimento di Matematica, via Buonarroti 2, 56127 Pisa, ITALY bini@dm.unipi.it October 15, 1996

2.2 Solution

Here is an expr‘erimentation, for exp100, with the MuPAD solver on a 300Mhz Pentium Pro, using Scho"nhage's algorithm.

```
################################################################################

   *----*    MuPAD 1.3  ---  Multi Processing Algebra Data Tool
  /|   /|
 *----* |    Copyright (c) 1992-96 by B. Fuchssteiner, Automath
 | *--|-*    University of Paderborn.  All rights reserved.
 |/   |/
 *----*      Demo version, please register with
             MuPAD-distribution@uni-paderborn.de

>> PRETTY_PRINT:=FALSE:
>> p:=_plus(x^j/j!$j=0..100):
>> st:=time():
>> float(hold(solve)(p,x));
{- 2.808924458594931245886620218882313111954933313322270e1 + 7.675541135315\
92348425822994307916362158204847528000*I, - 2.888369044926197552623276894l\
38438821590369393390150e1 + 6.999308702824276349664627686202542641448902078\
21790e-1*I, - 2.872516257146332973109760396030381621216298534166640e1 + 3.4\
...
- 7.230215130236114979156052389875518765095488545130600 \
- 3.501470286809142451452683156088578363291596163139700e1*I, 9.831248714635\
741546391488917559680407077076170498600 - 4.112016090358764175637155992820 3\
17561555858886372850e1*I}

>> time()-st;
1001000
```

See here the complete solution.

Paul Zimmerman, POLKA, INRIA Lorraine, Technopole de Nancy-Brabois,
615 rue du Jardin Botanique, BP 101, F-54600 Villers-les-Nancy Paul.Zimmermann@loria.fr
February 9 1998

References

[1] D. Bini. Numerical computation of polynomial zeros by means of aberth's method. *Numerical Algorithms*, 1997. (to appear).

[2] D. Bini and B. Mourrain. Polynomials test suite. 1996.