CoCoA-5 - Bug #1573

ApproxSolve: very imprecise

30 Jan 2021 19:48 - John Abbott

Status:	Closed	**Start date:**	30 Jan 2021
Priority:	Low	**Due date:**	
Assignee:	John Abbott	**% Done:**	100%
Category:	enhancing/improving	**Estimated time:**	3.99 hours
Target version:	CoCoA-5.4.2	**Spent time:**	3.95 hours

Description

I have found an example where ApproxSolve gives very imprecise solutions:

```
use QQ[x,y,z];

D := 7;
N := 2^17;
L := [z^D - (N*z-1)^2, y*(z-1/N)-1, x^D - (y^2*x-1)^2];

solns := ApproxSolve(L);
```

Currently it gives solns with z coord about 5*10^677, but the true value is about 111.
Ooops!

We should try to make it more precise...

Related issues:

Related to CoCoA-5 - Support #1252: ShapeLemma: no manual entry	**New**	**05 Mar 2019**
Related to CoCoA-5 - Bug #1171: RealRoots: first point is sometimes wrong?	**Closed**	**03 Apr 2018**
Related to CoCoA-5 - Bug #1216: RationalSolve: gives wrong answer	**Closed**	**07 Aug 2018**
Related to CoCoA-5 - Bug #1574: ApproxSolveTF	**New**	**03 Feb 2021**

History

#1 - 30 Jan 2021 20:17 - John Abbott

- Priority changed from Normal to Low

The example was deliberately chosen to be hard/troublesome. Making the parameters D and/or N larger almost surely makes it worse.

I suppose it would be nice to have a "precision" parameter, so that internal computations of real roots are with precision at least as good as requested.
Right now it is "automatic" (which I suspect just means some fixed value).

#2 - 12 Feb 2021 00:12 - John Abbott

- Status changed from New to Resolved

- Assignee set to John Abbott

- % Done changed from 0 to 50

I have now added an extra loop which checks the values of the polys at the points found.
If the values are not "heuristically small" then the internal precision is increased, and the computation restarted.

This is a fairly mindless "hack", but it works. It is certainly not fast :-(

The example in the description takes a few minutes.
NOTE (2021-09-16) Took about 164s on my Linux computer.

#3 - 12 Feb 2021 00:13 - John Abbott

- Related to Support #1252: ShapeLemma: no manual entry added

#4 - 12 Feb 2021 00:15 - John Abbott

- Related to Bug #1171: RealRoots: first point is sometimes wrong? added

#5 - 16 Sep 2021 13:00 - Anna Maria Bigatti

- Related to Bug #1216: RationalSolve: gives wrong answer added

#6 - 16 Sep 2021 13:04 - John Abbott

- % Done changed from 50 to 60

My hack favours correctness over speed.
Is there a verbosity level which gives useful info? If so, document it (right now the manual is a bit cryptic).

NOTE: may the verbosity should also say that a check indicated poor precision, so a new iteration will happen. Also some indication of time spent on each iteration would be nice.

#7 - 16 Sep 2021 13:08 - Anna Maria Bigatti

- % Done changed from 60 to 50

tested on my Mac, now is gives (takes some time)

```
/**/  indent([ [ FloatStr(coord) | coord in pt ] | pt in solns.AffinePts]);
[
  ["8.7581*10^(-47)",  "-1.0685*10^23",  "0.0000076294"],
  ["8.7581*10^(-47)",  "-1.0685*10^23",  "0.0000076294"],
  ["8.7581*10^(-47)",  "1.0685*10^23",  "0.0000076294"],
  ["8.7581*10^(-47)",  "1.0685*10^23",  "0.0000076294"],
  ["0.99998",  "0.0089742",  "111.43"],
  ["2.6487*10^18",  "1.0685*10^23",  "0.0000076294"],
  ["2.6487*10^18",  "-1.0685*10^23",  "0.0000076294"]
]
```

#8 - 24 Sep 2021 20:34 - John Abbott

- % Done changed from 50 to 80

I have just updated the manual entry for ApproxSolv.
I wonder what ApproxSolveTF does... mmm???

#9 - 24 Sep 2021 22:26 - John Abbott

- Status changed from Resolved to Closed

- % Done changed from 80 to 100

- Estimated time set to 3.99 h

ApproxSolveTF was surprisingly fast on the example given in the description (about 9s, vs 64s using ApproxSolve).
No idea how accurate the answer was -- perhaps ApproxSolveTF should also have a loop like one for ApproxSolve?.

Closing this issue as it has been adequately resolved.

#10 - 24 Sep 2021 22:26 - John Abbott

- Related to Bug #1574: ApproxSolveTF added