

CoCoA-5 - Bug #1574

ApproxSolveTF

03 Feb 2021 17:46 - John Abbott

Status:	New	Start date:	03 Feb 2021
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	bug	Estimated time:	0.00 hour
Target version:	CoCoA-5.4.2	Spent time:	0.75 hour
Description			
ApproxSolveTF is not documented. Maybe that is a good thing...			
<pre>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]; Pts := ApproxSolveTF(L); indent([[FloatStr(coord) coord in pt] pt in Pts]); [["0.0000", "0.0000", "0.0000"], ["0.0000", "0.0000", "0.0000"], ["0.0000", "0.0000", "0.0000"], ["0.0000", "0.0000", "0.0000"], ["0.99998", "0.0089742", "111.43"], ["0.0000", "0.0000", "0.0000"], ["0.0000", "0.0000", "0.0000"]]</pre>			
I'm not quite sure what happened there. ApproxSolve can handle this (if I increase the internal precision enough): it finds these solutions:			
<pre>[["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"]]</pre>			
Related issues:			
Related to CoCoA-5 - Bug #1216: RationalSolve: gives wrong answer		Closed	07 Aug 2018
Related to CoCoA-5 - Bug #1573: ApproxSolve: very imprecise		Closed	30 Jan 2021

History

#1 - 03 Feb 2021 17:48 - John Abbott

I have set the target version as 5.3.6, but maybe it could be postponed?

It is a bit strange that the solution [0,0,0] appears several times. Not sure what that means.

Maybe these fns (**ApproxSolve** and **ApproxSolveTF**) should automatically check that the solutions looks "plausible" but substituting into the orig polys, and verifying that the values obtained are reasonably small (whatever that might mean).

#2 - 03 Feb 2021 17:51 - John Abbott

- *Description updated*

#3 - 03 Feb 2021 22:39 - John Abbott

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

#4 - 16 Feb 2021 14:04 - John Abbott

The code seems to be a bit convoluted. It'll take time to work out what it is supposed to be doing... not today!

#5 - 24 Sep 2021 22:26 - John Abbott

- *Related to Bug #1573: ApproxSolve: very imprecise added*

#6 - 24 Sep 2021 22:29 - John Abbott

Probably ApproxSolveTF should have an accuracy-check-loop the same was as was done for ApproxSolve: the heuristic is that the final value of each poly should be at most $10^{(-6)}$ times that max value of a term in the poly (so that heuristically there is "good cancellation").

Of course, if you think of a better heuristic, let us know!