CoCoA-5 - Slug #907

ApproxSolve very slow on this example

14 Jul 2016 09:24 - John Abbott

Status:	Closed	**Start date:**	14 Jul 2016
Priority:	Normal	**Due date:**	
Assignee:	Anna Maria Bigatti	**% Done:**	100%
Category:	enhancing/improving	**Estimated time:**	7.70 hours
Target version:	CoCoA-5.2.4	**Spent time:**	7.80 hours

Description

While "playing" with a demo of sparse squares, I noticed that ApproxSolve is strangely slow on this example (while being much faster on other similar examples produced during the run)

```
use QQ[a[0..7],x];
sys := [
  a[6]^2 +2*a[5]*a[7] +2*a[3] +2*a[4],
  2*a[5]*a[6] +2*a[4]*a[7] +2*a[2] +2*a[3],
  a[5]^2 +2*a[4]*a[6] +2*a[3]*a[7] +2*a[1] +2*a[2],
  2*a[4]*a[5] +2*a[3]*a[6] +2*a[2]*a[7] +2*a[0] +2*a[1],
  2*a[3]*a[4] +2*a[2]*a[5] +2*a[1]*a[6] +2*a[0]*a[7],
  a[3]^2 +2*a[2]*a[4] +2*a[1]*a[5] +2*a[0]*a[6],
  2*a[2]*a[3] +2*a[1]*a[4] +2*a[0]*a[5],
  a[2]^2 +2*a[1]*a[3] +2*a[0]*a[4],
  2*a[1]*a[2] +2*a[0]*a[3],
  a[0]*x -1
 ];
solns := ApproxSolve(sys);
[[FloatStr(z) | z in SolnPt] | SolnPt in solns];
indent([[FloatStr(eval(f,pt)) | pt in solns] | f in sys]);
```

2016-07 added ApproxSolveTF using TwinFloats

Related issues:

Related to CoCoALib - Feature #565: FloatApprox for TwinFloat values?	**In Progress**	**22 May 2014**
Related to CoCoA-5 - Slug #1392: ApproxSolve: another slow example	**Closed**	**13 Jan 2020**

History

#1 - 14 Jul 2016 09:25 - John Abbott

On my old MacBook with CoCoA-5.1.5 this computation took more than 1 hour (then I stopped it).

#2 - 19 Jul 2016 08:16 - Anna Maria Bigatti

I've added the improvements we talked about at ICMS Berlin.
Now this example is almost instant :-) :-)
There are a few thing to settle (how to extract a RAT out of a TwinFloat which is not rational)
but it ready to be played with!
Use ApproxSolve2.
If it looks satisfactory on other examples we should write this in a paper!!

#3 - 19 Jul 2016 14:26 - John Abbott

- Related to Feature #565: FloatApprox for TwinFloat values? added

#4 - 21 Jul 2016 21:57 - John Abbott

- Status changed from New to In Progress

- % Done changed from 0 to 10

I have just tried the examples again (after Anna revised ApproxSolve yesterday or this morning). It still seems to be quite slow... It certainly wasn't "almost instant": I gave up waiting for the result.

:-(

#5 - 21 Jul 2016 21:59 - Anna Maria Bigatti

John Abbott wrote:

> I have just tried the examples again (after Anna revised ApproxSolve yesterday or this morning). It still seems to be quite slow... It certainly wasn't "almost instant": I gave up waiting for the result.
>
> :-(

Try ApproxSolve2

#6 - 22 Jul 2016 07:27 - Anna Maria Bigatti

- Description updated

#7 - 22 Jul 2016 11:43 - John Abbott

OK. ApproxSolve2 solves the problem in about 0.8s (on my old MacBook... it'll be faster on the new computer, if Dell ever decides to let me actually have it!)

I do not like the name ApproxSolve2.

Any idea why ApproxSolve is so slow?

#8 - 22 Jul 2016 12:17 - Anna Maria Bigatti

John Abbott wrote:

> OK. ApproxSolve2 solves the problem in about 0.8s (on my old MacBook... it'll be faster on the new computer, if Dell ever decides to let me actually have it!)

not bad!

> I do not like the name ApproxSolve2.

Needs also refinement and testing ;-)

Any idea why ApproxSolve is so slow?

gbasis lex. I think we should use FGLM!

#9 - 22 Jul 2016 13:21 - John Abbott

Do you really need lex? I vaguely recall using several eliminations to compute the various polynomials of the form x[j] - poly(x[n]).

There ought to be a quick(ish) way of getting these polys...

#10 - 07 Apr 2017 15:09 - Anna Maria Bigatti

- Description updated

#11 - 05 Oct 2017 10:21 - John Abbott

This is another example where ApproxSolve is too slow (in CoCoA-5.2.2).
The example came from "playing with" palindromic sparse squares -- preparation for Cameroon.

```
use QQ[a[1..5],nz];
//SetVerbosityLevel(100);
L := [a[3]^2 +2*a[2]*a[4] +4*a[1]*a[5] +2*a[4],
  a[1]*a[3] +a[2]*a[4] +(1/2)*a[4]^2 +2*a[3]*a[5] +a[2],
  a[2]^2 -2*a[2]*a[4] -a[4]^2 -4*a[3]*a[5] -2*a[2] +2*a[4],
  a[1]*a[2] +a[3],
  a[2]*a[3] +a[3]*a[4] +2*a[4]*a[5] +a[1] -a[3],
  a[2]*a[4]^2 +120*a[2]*a[4] +14*a[4]^2 -120*a[1]*a[5] +152*a[3]*a[5] +16*a[5]^2 -96*a[1]*nz +96*a[5]*nz +128*
a[2] +104*a[4] -88,
  a[3]*a[4]^2 -528*nz^3 +663*a[1]*a[4] +511*a[3]*a[4] +1748*a[2]*a[5] +1156*a[4]*a[5] -658*a[2]*nz +2362*a[4]*
nz -3888*a[1] -1626*a[3] +3172*a[5] +4429*nz,
  a[4]^3 +(-11054/17)*a[2]*a[4] +(-1381/17)*a[4]^2 +(10104/17)*a[1]*a[5] +(-14324/17)*a[3]*a[5] +(-1472/17)*a[
5]^2 +(8240/17)*a[1]*nz +(-8224/17)*a[5]*nz +(-96/17)*nz^2 +(-11602/17)*a[2] +(-8844/17)*a[4] +8386/17,
  a[1]*a[4]*a[5] +(1/34)*a[2]*a[4] +(53/68)*a[4]^2 +(148/17)*a[1]*a[5] +(-3/17)*a[3]*a[5] +(-20/17)*a[5]^2 +(9
2/17)*a[1]*nz +(-94/17)*a[5]*nz +(12/17)*nz^2 +(-149/68)*a[2] +(-203/34)*a[4] -31/34,
  a[1]*a[4]^2 +24*nz^3 +(-69/2)*a[1]*a[4] +(-89/4)*a[3]*a[4] -70*a[2]*a[5] +(-107/2)*a[4]*a[5] +25*a[2]*nz -99
*a[4]*nz +(697/4)*a[1] +(271/4)*a[3] -129*a[5] +(-403/2)*nz,
  a[3]*nz +1,
  a[2]*a[4]*nz +(2/5)*a[1]*a[4] +(1/10)*a[3]*a[4] +(1/5)*a[4]*a[5] +(4/5)*a[2]*nz +(-7/10)*a[1] +(-1/10)*a[3]
+(-6/5)*a[5],
  a[4]*a[5]*nz +(1/2)*a[1]*nz +(-1/2)*a[2] +(-1/2)*a[4] +1/2,
  a[4]^2*nz +(-4/5)*a[1]*a[4] +(-1/5)*a[3]*a[4] +(-2/5)*a[4]*a[5] +(2/5)*a[2]*nz +(-3/5)*a[1] +(1/5)*a[3] +(-8
/5)*a[5],
  a[1]*a[4]*nz +(-137/17)*a[2]*a[4] +(-52/17)*a[4]^2 +(-92/17)*a[1]*a[5] +(-164/17)*a[3]*a[5] +(40/17)*a[5]^2
+(-48/17)*a[1]*nz +(52/17)*a[5]*nz +(-24/17)*nz^2 +(-87/17)*a[2] +(67/17)*a[4] +150/17,
  a[1]*a[5]*nz +(-1/5)*a[1]*a[4] +(-1/20)*a[3]*a[4] +(-1/10)*a[4]*a[5] +(-2/5)*a[2]*nz +(1/2)*a[4]*nz +(7/20)*
a[1] +(-1/5)*a[3] +(3/5)*a[5],
  a[2]*a[5]*nz +(-9/34)*a[2]*a[4] +(-1/68)*a[4]^2 +(-40/17)*a[1]*a[5] +(-7/17)*a[3]*a[5] +(-24/17)*a[5]^2 +(-1
2/17)*a[1]*nz +(13/17)*a[5]*nz +(-6/17)*nz^2 +(-53/68)*a[2] +(-9/34)*a[4] +29/17,
  a[3]*a[4]*a[5] +(3317/34)*a[2]*a[4] +(837/68)*a[4]^2 +(-1438/17)*a[1]*a[5] +(2187/17)*a[3]*a[5] +(232/17)*a[
5]^2 +(-1176/17)*a[1]*nz +(1172/17)*a[5]*nz +(24/17)*nz^2 +(1719/17)*a[2] +(1242/17)*a[4] -1289/17,
  a[2]*a[4]*a[5] -48*nz^3 +(307/5)*a[1]*a[4] +(223/5)*a[3]*a[4] +154*a[2]*a[5] +(526/5)*a[4]*a[5] +(-286/5)*a[
2]*nz +210*a[4]*nz +(-1756/5)*a[1] +(-1441/10)*a[3] +(1394/5)*a[5] +403*nz,
  a[4]^2*a[5] +216*nz^3 +(-1358/5)*a[1]*a[4] +(-2079/10)*a[3]*a[4] -714*a[2]*a[5] +(-2349/5)*a[4]*a[5] +(1339/
5)*a[2]*nz -963*a[4]*nz +(15903/10)*a[1] +(6639/10)*a[3] +(-6456/5)*a[5] +(-3623/2)*nz,
  a[2]*a[5]^2 +(71/8)*a[2]*a[4] +(5/4)*a[4]^2 -8*a[1]*a[5] +(21/2)*a[3]*a[5] +2*a[5]^2 +(-13/2)*a[1]*nz +6*a[5
]*nz +9*a[2] +(53/8)*a[4] -25/4,
  a[1]*a[5]^2 +3*nz^3 +(-309/80)*a[1]*a[4] +(-211/80)*a[3]*a[4] +(-17/2)*a[2]*a[5] +(-143/20)*a[4]*a[5] +(131/
40)*a[2]*nz +(-103/8)*a[4]*nz +(107/5)*a[1] +(333/40)*a[3] +(-337/20)*a[5] +(-407/16)*nz,
  a[3]*a[5]^2 -27*nz^3 +(2651/80)*a[1]*a[4] +(4363/160)*a[3]*a[4] +(373/4)*a[2]*a[5] +(4673/80)*a[4]*a[5] +(-1
409/40)*a[2]*nz +(987/8)*a[4]*nz +(-32011/160)*a[1] +(-13653/160)*a[3] +(6691/40)*a[5] +(3619/16)*nz,
  a[2]*nz^2 +(29/68)*a[2]*a[4] +(29/68)*a[4]^2 +(-234/17)*a[1]*a[5] +(-18/17)*a[3]*a[5] +(-120/17)*a[5]^2 +(-1
71/34)*a[1]*nz +(147/34)*a[5]*nz +(-13/17)*nz^2 +(-105/136)*a[2] +(23/34)*a[4] +60/17,
  a[5]^2*nz +(1/5)*a[1]*a[4] +(-3/40)*a[3]*a[4] +(1/4)*a[2]*a[5] +(-3/20)*a[4]*a[5] +(-1/10)*a[2]*nz +(-19/40)
```

```
*a[1] +(-17/40)*a[3] +(-11/10)*a[5] +(-1/4)*nz,
  a[4]*nz^2 +(-235/136)*a[2]*a[4] +(-41/68)*a[4]^2 +(-110/17)*a[1]*a[5] +(-49/17)*a[3]*a[5] +(-32/17)*a[5]^2 +
(-217/68)*a[1]*nz +(74/17)*a[5]*nz +(-8/17)*nz^2 +(-65/68)*a[2] +(95/68)*a[4] +217/68,
  a[5]^3 +9*nz^3 +(-87/8)*a[1]*a[4] +(-597/64)*a[3]*a[4] +(-261/8)*a[2]*a[5] +(-591/32)*a[4]*a[5] +(49/4)*a[2]
*nz +(-337/8)*a[4]*nz +(4333/64)*a[1] +(1899/64)*a[3] +(-927/16)*a[5] +(-1203/16)*nz,
  a[4]*a[5]^2 +(-2739/68)*a[2]*a[4] +(-701/136)*a[4]^2 +(1047/34)*a[1]*a[5] +(-898/17)*a[3]*a[5] +(-82/17)*a[5
]^2 +(452/17)*a[1]*nz +(-450/17)*a[5]*nz +(-12/17)*nz^2 +(-1379/34)*a[2] +(-485/17)*a[4] +517/17,
  a[1]*nz^2 -6*nz^3 +(389/40)*a[1]*a[4] +(347/80)*a[3]*a[4] +(29/2)*a[2]*a[5] +(517/40)*a[4]*a[5] +(-101/20)*a
[2]*nz +(45/2)*a[4]*nz +(-3719/80)*a[1] +(-1337/80)*a[3] +(459/20)*a[5] +(411/8)*nz,
  a[1]^2 +(-137/17)*a[2]*a[4] +(-52/17)*a[4]^2 +(-92/17)*a[1]*a[5] +(-164/17)*a[3]*a[5] +(40/17)*a[5]^2 +(-48/
17)*a[1]*nz +(52/17)*a[5]*nz +(-24/17)*nz^2 +(-191/34)*a[2] +(67/17)*a[4] +150/17,
  a[5]*nz^2 -6*nz^3 +(181/20)*a[1]*a[4] +(21/5)*a[3]*a[4] +(29/2)*a[2]*a[5] +(253/20)*a[4]*a[5] +(-211/40)*a[2
]*nz +(97/4)*a[4]*nz +(-1801/40)*a[1] +(-673/40)*a[3] +(241/10)*a[5] +(101/2)*nz,
  nz^4 +(-62855/3264)*a[2]*a[4] +(-5807/816)*a[4]^2 +(-5381/204)*a[1]*a[5] +(-3407/136)*a[3]*a[5] +(-137/102)*
a[5]^2 +(-2955/544)*a[1]*nz +(11791/816)*a[5]*nz +(-3395/272)*nz^2 +(-43787/3264)*a[2] +(15263/1632)*a[4] +133
93/544];
I := ideal(L);
IsZeroDim(I);
mp1 := MinPolyQuot(a[1],I,a[1]);
mp2 :=MinPolyQuot(a[2],I,a[2]);
mp3 := MinPolyQuot(a[3],I,a[3]);
mp4 := MinPolyQuot(a[4],I,a[4]);
mp5 := MinPolyQuot(a[5],I,a[5]);

ApproxSolns := ApproxSolve(L); --> takes too long!
```

#12 - 01 Mar 2018 21:18 - Anna Maria Bigatti

- Assignee set to Anna Maria Bigatti

- Target version changed from CoCoA-5.?.? to CoCoA-5.2.4

- % Done changed from 10 to 80

- Estimated time set to 10.00 h

greatly improved, now part of test-ApproxSolve

#13 - 31 Jul 2018 12:11 - Anna Maria Bigatti

- Status changed from In Progress to Feedback

- % Done changed from 80 to 90

I tested the examples, they are now quite fast.
I suggest closing this.
(controlling the error in the evaluation is another problem, and should be dealt in another issue)

#14 - 31 Jul 2018 12:14 - Anna Maria Bigatti

- Description updated

#15 - 06 Aug 2018 16:22 - John Abbott

- Status changed from Feedback to Closed

- % Done changed from 90 to 100

- Estimated time changed from 10.00 h to 7.70 h

Just tested the two examples. They are acceptably fast now.
Closing.

#16 - 14 Jan 2020 08:27 - Anna Maria Bigatti

- Related to Slug #1392: ApproxSolve: another slow example added