

CoCoALib - Slug #1057

Slug: Polynomial ring constructor slow with (big) matrix ordering

04 May 2017 18:25 - Anna Maria Bigatti

Status:	In Progress	Start date:	04 May 2017
Priority:	High	Due date:	
Assignee:	Anna Maria Bigatti	% Done:	50%
Category:	Improving	Estimated time:	8.00 hours
Target version:	CoCoALib-0.99880	Spent time:	5.70 hours
Description			
I have an example which shows that creating a SparsePolyRing with matrix ordering is slower than default ordering.			
Related issues:			
Related to CoCoA-5 - Slug #1047: NewPolyRing with user defined ordering is sl...		Closed	18 Apr 2017
Related to CoCoALib - Slug #1049: GroebnerFan: slow examples		In Progress	19 Apr 2017
Related to CoCoA-5 - Slug #1068: PolyRing constructor: NewOrdvArith computed ...		In Progress	17 May 2017
Related to CoCoALib - Design #841: NewPolyRing: tidy up the many different ve...		In Progress	21 Jan 2016
Related to CoCoALib - Feature #1197: IsZeroDet: new fn		In Progress	26 Jun 2018
Related to CoCoALib - Slug #1588: ElimMat is slow		New	15 Apr 2021
Related to CoCoALib - Bug #1641: gcd does not recognize univariate input		Closed	20 Dec 2021
Related to CoCoALib - Design #1798: Computing in sub polyring		New	22 Mar 2024

History

#1 - 04 May 2017 18:30 - Anna Maria Bigatti

- % Done changed from 0 to 10

(The example is cvs-ed in the dir IdealsModp)

I had added a ludicrous number of (unused) indeterminates: the computation speed seems good, I believe the cost is in checking the matrix ordering, so I should investigate that part.

If the check is intrinsically slow, I could design a way to pass the matrix from a ring to a new ring (for GB operations) without further checks.

#2 - 04 May 2017 18:31 - Anna Maria Bigatti

- Description updated

#3 - 16 May 2017 15:24 - Anna Maria Bigatti

- Related to Slug #1047: NewPolyRing with user defined ordering is slower than CoCoALib added

#4 - 16 May 2017 15:25 - Anna Maria Bigatti

- Related to Slug #1049: GroebnerFan: slow examples added

#5 - 17 May 2017 10:36 - Anna Maria Bigatti

- Related to Slug #1068: PolyRing constructor: NewOrdvArith computed twice added

#6 - 17 May 2017 10:39 - Anna Maria Bigatti

- Subject changed from Slug: Polynomial ring with matrix ordering to Slug: Polynomial ring constructor slow with (big) matrix ordering

- Status changed from New to In Progress

- % Done changed from 10 to 30

First problem found: NewOrdvArith(ord) ic computed twice.
If ord is defined by M, then many checks are done on M.

#7 - 17 May 2017 18:02 - Anna Maria Bigatti

```
SparsePolyRing NewPolyRing(const ring& CoeffRing, const PPMonoid& PPM)
{
    return NewPolyRing_DMP(CoeffRing, PPM); // the only one for DMP
    // !!! ANNA: but if PPM is PPMonoidOv we could be clever!!!
}
```

#8 - 01 Sep 2017 11:30 - Anna Maria Bigatti

- Related to Design #841: NewPolyRing: tidy up the many different versions added

#9 - 06 Nov 2017 13:57 - John Abbott

- Target version changed from CoCoALib-0.99560 to CoCoALib-0.99600

#10 - 30 Jul 2018 16:11 - Anna Maria Bigatti

- Target version changed from CoCoALib-0.99600 to CoCoALib-0.99650 November 2019

#11 - 09 Apr 2019 10:55 - John Abbott

- Related to Feature #1197: IsZeroDet: new fn added

#12 - 01 Oct 2019 11:27 - John Abbott

- Target version changed from CoCoALib-0.99650 November 2019 to CoCoALib-0.99800

#13 - 15 Apr 2021 10:15 - John Abbott

- Related to Slug #1588: ElimMat is slow added

#14 - 15 Apr 2021 10:23 - John Abbott

Here is a benchmark (since there was none given earlier:

```
/**/ t0 := CpuTime(); M := ElimMat(1..250,500); TimeFrom(t0);
58.731 ---> see issue 1588
/**/ t0 := CpuTime(); P := NewPolyRing(QQ, SymbolRange("x",1,500), M, 0); TimeFrom(t0);
6.403
```

Since we know that M is already a good matrix, it is surprising that NewPolyRing takes more than 6s (on my linux box).
Surely it suffices to check that M is square with integer entries, has non-zero det, and that the first non-zero in each col is positive?

#15 - 04 Feb 2022 22:36 - John Abbott

- Target version changed from CoCoALib-0.99800 to CoCoALib-0.99850

The first line in comment 10 above is still about 50s... too slow!
How long does it take to compute the det of a 500x500 matrix?
We need that IsZeroDet function!

#16 - 08 Mar 2024 18:09 - John Abbott

We now have a first impl of **IsZeroDet**. Hopefully this will help!

#17 - 11 Mar 2024 09:39 - Anna Maria Bigatti

- % Done changed from 30 to 50

John Abbott wrote:

Here is a benchmark (since there was none given earlier:

Much better now with IsZeroDet. On my computer it is

```
/**/ t0 := CpuTime(); M := ElimMat(1..250,500); TimeFrom(t0);  
48.822  
/**/ t0 := CpuTime(); P := NewPolyRing(QQ, SymbolRange("x",1,500), M, 0); TimeFrom(t0);  
1.160
```

But I still want to investigate why it calls twice NewOrdvArith(ord) and IsTermOrdering 3 times.

#18 - 21 Mar 2024 11:33 - Anna Maria Bigatti

- Related to Bug #1641: gcd does not recognize univariate input added

#19 - 22 Mar 2024 09:18 - John Abbott

- Target version changed from CoCoALib-0.99850 to CoCoALib-0.99880

#20 - 22 Mar 2024 09:19 - John Abbott

- Priority changed from Normal to High

- Target version changed from CoCoALib-0.99880 to CoCoALib-0.99850

#21 - 22 Mar 2024 09:26 - Anna Maria Bigatti

- Related to Design #1798: Computing in sub polyring added

#22 - 22 Mar 2024 09:31 - John Abbott

- Target version changed from *CoCoALib-0.99850* to *CoCoALib-0.99880*