

## CoCoA-5 - Bug #724

### RationalSolve: wrongly complains about non zero-dim even in finite char

02 Jun 2015 15:53 - John Abbott

<b>Status:</b>	Closed	<b>Start date:</b>	02 Jun 2015
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	John Abbott	<b>% Done:</b>	100%
<b>Category:</b>	enhancing/improving	<b>Estimated time:</b>	1.44 hour
<b>Target version:</b>	CoCoA-5.4.0	<b>Spent time:</b>	1.45 hour
<b>Description</b>			
I ran the example given in issue <a href="#">#723</a> which tries to compute all rational solutions over a finite field of a homogeneous system which is not 1-dimensional. But RationalSolve gave an error even though the number of <b>rational</b> solutions is necessarily finite.			
Is this a bug?			
<b>Related issues:</b>			
Related to CoCoA-5 - Bug #1216: RationalSolve: gives wrong answer		<b>Closed</b>	<b>07 Aug 2018</b>
Related to CoCoA-5 - Slug #1270: RationalSolve: use MinPolyQuot instead of elim		<b>Closed</b>	<b>05 Apr 2019</b>

### History

#### #1 - 02 Jun 2015 15:55 - John Abbott

Is it just enough to skip the check for zero-dimensionality in this case?  
(if so, that's a really simple fix)

#### #2 - 02 Jun 2015 17:06 - John Abbott

I have disabled the zero-dimensionality check if the coeff ring is a finite field.

In one simple test that seems to be enough for the homog case; it was not enough for the inhomog case -- I had to change the result of an elim because if all values are possible then the result of the elim is the zero ideal (yet the code expected there to be at least one generator).

#### #3 - 02 Jun 2015 22:43 - John Abbott

- Status changed from New to In Progress

- % Done changed from 0 to 10

It is not quite as easy as I had hoped. Every so often zero polynomials pop up in awkward places (e.g. to be factorized for their rational roots). The code needs to be reviewed more completely to be sure that it can cope with non zero-dim systems when the coeffs are in a finite field (i.e. when the zero poly has a finite number of rational roots).

Note: maybe I should add a fn which computes the rational roots of a univariate polynomial, and which gives all finite field elems if the polynomial is zero.

#### #4 - 26 Oct 2018 15:59 - John Abbott

The example from [#723](#) still fails with the following error:

```
--> ERROR: Value must be non-zero
--> [CoCoALib] factor(x)
--> WHERE: at line 102 (column 23) of RationalPoints.cpkg5
-->   LinearFacs := [g in factor(f).factors | deg(g) = 1];
-->               ^^^^^^^^^^
```

The stack unwinding gave:

```
CONTEXT: function RationalRoots at line 102 of RationalPoints.cpkg5
CALLED BY: function RationalAffinePointsLoop at line 76 of RationalPoints.cpkg5
  CALLED BY: function RationalProjectivePointsLoop at line 136 of RationalPoints.cpkg5
    CALLED BY: function RationalProjectivePointsLoop at line 135 of RationalPoints.cpkg5
      CALLED BY: function RationalProjectivePointsLoop at line 135 of RationalPoints.cpkg5
        CALLED BY: function RationalProjectivePointsLoop at line 135 of RationalPoints.cpkg5
          CALLED BY: function RationalProjectivePointsLoop at line 135 of RationalPoints.cpkg5
            CALLED BY: function RationalProjectivePointsLoop at line 135 of RationalPoints.cpkg5
              CALLED BY: function RationalProjectivePointsLoop at line 135 of RationalPoints.cpkg5
                CALLED BY: function RationalProjectivePointsLoop at line 135 of RationalPoints.cpkg5
                  ...
                    CALLED BY: function RationalProjectivePointsLoop at line 135 of RationalPoints.cpkg5
                      CALLED BY: function RationalProjectivePointsLoop at line 135 of RationalPoints.cpkg5
                        CALLED BY: function RationalProjectivePointsLoop at line 135 of RationalPoints.cpkg5
                          CALLED BY: function RationalProjectivePointsLoop at line 135 of RationalPoints.cpkg5
                            CALLED BY: function RationalProjectivePointsLoop at line 135 of RationalPoints.cpkg5
                              CALLED BY: function RationalProjectivePointsLoop at line 135 of RationalPoints.cpkg5
                                CALLED BY: function RationalProjectivePointsLoop at line 135 of RationalPoints.cpkg5
                                  CALLED BY: function RationalProjectivePoints at line 121 of RationalPoints.cpkg5
                                    CALLED BY: function RationalSolve at line 56 of RationalPoints.cpkg5
                                      called at top-level
```

**#5 - 05 Apr 2019 20:14 - John Abbott**

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

**#6 - 05 Apr 2019 20:19 - John Abbott**

- Related to Slug #1270: RationalSolve: use MinPolyQuot instead of elim added

**#7 - 12 Feb 2021 00:16 - John Abbott**

- Target version changed from CoCoA-5.?.? to CoCoA-5.4.0

**#8 - 12 Feb 2021 12:19 - John Abbott**

- Status changed from *In Progress* to *Resolved*
- Assignee set to *John Abbott*
- % Done changed from 10 to 60

The example from issue [#723](#) is way too slow to act as a test case -- do we have a more sane test case? How about the one below? I suspect that this is now resolved.

```
/**/ use ZZ/(3)[x,y,z];
/**/ L := [x, x*y];
/**/ solns := RationalSolveHomog(L);
record[ProjectivePts := [[0, 1]], indets := [x, y]]
```

**#9 - 19 Mar 2021 12:43 - John Abbott**

- Status changed from *Resolved* to *Closed*
- % Done changed from 60 to 100
- Estimated time set to 1.44 h

Added a test to exbugs.cocoa5.

Closing.