Solving systems of algebraic equations with

axiom™

Marc Moreno Maza

University of Western Ontario

ACA 2004

Outline

- The main features of AXIOM.
- A brief history.
- Some celebrated AXIOM packages.
- Solving systems of algebraic equations with AXIOM.
- Recent developments at UWO.

```
Z := Integer
   (1)  Integer
                                                  Type: Domain
                                                  Time: 0 sec

Q := Fraction Z
   (2)  Fraction Integer
                                                  Type: Domain
                                                  Time: 0 sec

P := Polynomial Q
   (3)  Polynomial Fraction Integer
                                                  Type: Domain
                                                  Time: 0 sec

K := Fraction P
   (4)  Fraction Polynomial Fraction Integer
                                                  Type: Domain
                                                  Time: 0 sec
```

```
M := SquareMatrix(4, K)
   (5)  SquareMatrix(4,Fraction Polynomial Fraction Integer)
                                                    Type: Domain
                                                    Time: 0 sec

quaternion(a: K, b: K, c: K, d: K): M ==
   matrix ([[d ,  a ,  b ,  c],
            [-a ,  d ,  -c ,  b],
            [-b ,  c ,  d ,  -a],
            [-c ,  -b ,  a ,  d ]])

   Function declaration quaternion : (Fraction Polynomial Fraction
      Integer,Fraction Polynomial Fraction Integer,Fraction Polynomial
      Fraction Integer,Fraction Polynomial Fraction Integer) ->
      SquareMatrix(4,Fraction Polynomial Fraction Integer) has been
      added to workspace.
```

```
q := quaternion(1,2,3,x/y)
        + x                 +
        | -    1    2    3 |
        | y                 |
        |                   |
        |      x            |
        |- 1   -   - 3   2 |
        |      y            |
   (7)  |                   |
        |           x       |
        |- 2   3    -    - 1|
        |           y       |
        |                   |
        |                x |
        |- 3  - 2   1    - |
        +                y +
recip(q)
```

```
     +   1              1  2         1  2           3  2  +
     |  -- x y         -- y          - y           -- y   |
     |  14             14            7             14     |
     |----------  - ----------  - ----------  - ----------|
     | 2    1  2     2    1  2     2    1  2     2    1  2|
     |y  + -- x     y  + -- x     y  + -- x     y  + -- x |
     |     14            14            14            14   |
     |                                                    |
     |    1  2        1              3  2          1  2   |
     |   -- y        -- x y         -- y           - y    |
     |   14          14             14             7      |
     |----------   ----------    ----------   - ----------|
     | 2    1  2    2    1  2     2    1  2      2    1  2|
     |y  + -- x    y  + -- x     y  + -- x      y  + -- x |
     |     14           14            14             14   |
(8)  |                                                    |
     |   1  2           3  2        1              1  2   |
     |   - y           -- y        -- x y         -- y    |
     |   7             14          14             14      |
     |----------  - ----------   ----------    ---------- |
     | 2    1  2     2    1  2    2    1  2     2    1  2 |
     |y  + -- x     y  + -- x    y  + -- x     y  + -- x  |
     |     14            14           14            14    |
     |                                                    |
     |    3  2        1  2            1  2        1       |
     |   -- y         - y            -- y        -- x y   |
     |   14           7              14          14       |
     |----------   ----------   - ----------   ---------- |
     | 2    1  2    2    1  2      2    1  2    2    1  2 |
     |y  + -- x    y  + -- x      y  + -- x    y  + -- x  |
     +     14           14             14           14    +
```

```
Type: Union(SquareMatrix(4,Fraction Polynomial Fraction Integer),...)
                Time: 0.05 (EV) + 0.02 (OT) + 0.01 (GC) = 0.08 sec

quaternion(1,2,3,q)
   Cannot convert from type SquareMatrix(4,Fraction Polynomial Fraction
     Integer) to Fraction Polynomial Fraction Integer for value

+ x             +
| -    1    2    3 |
| y              |
|                |
|      x         |
|- 1   -   - 3   2 |
|      y         |
|                |
|           x    |
|- 2   3    -   - 1|
|           y    |
|                |
|               x |
|- 3  - 2   1    - |
+               y +
```

- Axiom is a general purpose Computer Algebra system.
- It defines a strongly typed, mathematically correct type hierarchy.
- It has a programming language (similar to that of ALDOR) and a built-in compiler.
- AXIOM source programs are compiled to LISP. Then, AXIOM compiled programs are called in an interpreter.
- The main concern of the AXIOM algebraic libraries is genericity.

```
++ Description: PolynomialFactorizationByRecursion(R,E,VarSet,S)
++ is used for factorization of sparse univariate polynomials over
++ a domain S of multivariate polynomials over R.
PolynomialFactorizationByRecursion(R,E, VarSet:OrderedSet, S): public ==
 private where
  R:PolynomialFactorizationExplicit
  E:OrderedAbelianMonoidSup
  S:PolynomialCategory(R,E,VarSet)
```

The main features of AXIOM

- Lack of emphasis on specialized non-generic algorithms e.g. generic Gaussian elimination is used over small prime fields.
- No access to machine types and no dense univariate polynomials.
- Consequently modular algorithms (polynomial gcd and factorization) are inefficient.
- AXIOM is difficult to interface with external packages.
- But things were tried there for the first time! In addition, things are changing now!
- In addition, ALDOR and its libraries were initially developed to extend AXIOM and solve these difficulties.

- The `Scratchpad` project (AXIOM) was developed since 1971 by IBM under the direction of **Richard Jenks.**
- **Barry Trager** was key to the technical direction of the project. Scratchpad developed over a 20 year stretch and was basically considered as a research platform for developing new ideas in computational mathematics.
- In the 1990s, the `Scratchpad` project was renamed to AXIOM sold to the **Numerical Algorithms Group (NAG)** and became a commercial system. For a variety of reasons it never became a financial success and NAG withdrew it from the market in October, 2001.
- NAG agreed to release Axiom as **free software.** The basic motivation was that Axiom represents something different from other programs in a lot of ways. In its current state it represents about 30 years and 300 man-years of research work. **Tim Daily** plays a key role in this open AXIOM today.

Cyril Alberga
Roy Adler
Richard Anderson
George Andrews

Henry Baker
Stephen Balzac
Yurij Baransky
David R. Barton
Gerald Baumgartner
Gilbert Baumslag
Fred Blair
Mark Botch
Alexandre Bouyer
Peter A. Broadbery
Martin Brock
Manuel Bronstein
Florian Bundschuh
William Burge

Robert Caviness
Bruce Char
Cheekai Chin
David V. Chudnovsky
Gregory V. Chudnovsky
Josh Cohen
Don Coppersmith
George Corliss
Robert Corless
Gary Cornell

Timothy Daly Sr.
Timothy Daly Jr.
James H. Davenport
Jean Della Dora
Michael Dewar
Claire DiCrescendo
Sam Dooley
Lionel Ducos
Martin Dunstan

Brian Dupee	Dominique Duval	
Robert Edwards	Lars Erickson	
Richard Fateman	Bertfried Fauser	Stuart Feldman
Brian Ford	Albrecht Fortenbacher	George Frances
Timothy Freeman	Korrinn Fu	
Marc Gaetano	Rudiger Gebauer	Patricia Gianni
Holger Gollan	Teresa Gomez-Diaz	Laureano Gonzalez-Vega
Stephen Gortler	Johannes Grabmeier	Matt Grayson
James Griesmer	Vladimir Grinberg	Oswald Gschnitzer
Steve Hague	Vilya Harvey	Satoshi Hamaguchi
Martin Hassner	Henderson	
Pietro Iglio		

Richard Jenks

Grant Keady	Tony Kennedy	Klaus Kusche
Bernhard Kutzler		

Larry Lambe	Frederic Lehobey	Michel Levaud
Rudiger Loos	Michael Lucks	Richard Luczak

Camm Maguire	Bob McElrath	Michael McGettrick
Ian Meikle	David Mentre	Victor S. Miller
Gerard Milmeister	Mohammed Mobarak	H. Michael Moeller
Michael Monagan	Marc Moreno Maza	Scott Morrison
Mark Murray		

William Naylor	C. Andrew Neff	John Nelder
Godfrey Nolan	Arthur Norman	

Michael O'Connor

Julian A. Padget	Bill Page	Susan Pelzel
Michel Petitot	Didier Pinchon	

Claude Quitte

Norman Ramsey	Michael Richardson	Renaud Rioboo
Jean Rivlin	Simon Robinson	Michael Rothstein

Philip Santas	Alfred Scheerhorn	William Schelter
Gerhard Schneider	Martin Schoenert	Marshall Schor
Fritz Schwarz	Nick Simicich	William Sit
Jonathan Steinbach	Christine Sundaresan	Robert Sutor
Moss E. Sweedler	Eugene Surowitz	

James Thatcher	Dylan Thurston	Barry Trager

Themos T. Tsikas

Bernhard Wall	Stephen Watt	Juergen Weiss
M. Weller	Mark Wegman	James Wen
Thorsten Werther	John M. Wiley	Berhard Will
Clifton J. Williamson	Shmuel Winograd	Robert Wisbauer
Waldemar Wiwianka	Knut Wolf	

David Yun

Richard Zippel	Bruno Zuercher	Dan Zwillinger

The Dynamical closure implements the ***constructible closure*** of a given field K. It can be viewed as ***the smallest* extension field** K_c of K which is **constructibly closed** (i.e. we may add parameters to K_c, set algebraic constraints on them, and stay in K_c.) These algebraic constraints need not to define a prime ideal. Indeed, The **D5 principle** is used in order to split the computations into cases, when this becomes necessary. The AXIOM implementation was developed by **Dominique Duval and Teresa Gómez Díaz** from 1984 to 1996. It is available from Dominique Duval's web page.

The Real Closure implements the ***real closure*** of an ordered field. A ***real closed field*** is a real field that has no nontrivial real algebraic extension. The AXIOM implementation is developed by **Renaud Rioboo** since 1991 and was integrated in the AXIOM distribution for the release 2.2 in 1999. It is used in the AXIOM implementation of Collins CAD (also by Renaud Rioboo).

```
RN := Fraction Integer
   (1)  Fraction Integer
                                       Type: Domain
                                       Time: 0.06 (IN) + 0.06 (OT) = 0.12 sec
CL:= DCCLOS RN

   (2)  DynamicConstructibleClosure Fraction Integer
                                       Type: Domain
                                       Time: 0.01 (IN) + 0.01 (OT) = 0.02 sec
CC:= DCCASE RN
   (3)  DynamicConstructibleCase Fraction Integer
                                       Type: Domain
                                       Time: 0.01 (OT) = 0.01 sec
M:= Matrix CL
   (4)  Matrix DynamicConstructibleClosure Fraction Integer
                                       Type: Domain
                                       Time: 0.02 (IN) + 0.02 (OT) = 0.04 sec
```

```
rango(x:NNI):NNI ==
  a:CL:= newElement('a)
  xc:CL:= x::CL
  m:M:= [[a+xc,1::CL],[1::CL,1::CL]]
  rank(m)

   Function declaration rango : NonNegativeInteger -> NonNegativeInteger
   has been added to workspace.
                        Type: Void
                        Time: 0.05 (IN) + 0.01 (OT) + 1.08 (GC) = 1.14 sec

allCases(rango,0)$DCTRL(CC,NNI,NNI)
   (12)
   [value is 2 in case a = 0, value is 1 in case a = 1,
    value is 2 in case a /= 0 , a /= 1]
Type: List DynamicClause(DynamicConstructibleCase Fraction Integer,NonNegativeInteger)
```

Triangular decompositions provide one of the best symbolic tools for decomposing radical polynomial ideals (and algebraic varieties) into unmixed components or irreducible components. They rely also on the *D5 Principle* and on polynomial gcds over direct product of fields **that I will discuss tomorrow**.

Modular Methods for these polynomial gcds are currently developed and implemented at ORCCA (UWO) in AXIOM, ALDOR and MAPLE. They are designed to provide a great speed up for the triangular decomposition libraries in this software.

Efficient Polynomial Arithmetic is currently being developed by **Xin Li** (UWO) for AXIOM: **dense polynomial**, **improved Lisp support** for modular arithmetic.

The open AXIOM around Tim Daily to be detailed below.

AXIOM-**MONET Services** by **E. Smirnova** and **S.M. Watt**.

- In [Lazard, 1992] a subset

$$T = \{T_1, \ldots T_n\} \subseteq \mathbf{k}[x_1 < \cdots < x_n]$$

is a triangular set if for $i = 1 \cdots n$

$$T_i = 1\, \mathbf{x_i^{d_i}} + a_{i-1}\, \mathbf{x_i^{d_i - 1}} + \cdots + a_1\, \mathbf{x_i} + a_0$$

with

$$a_{i-1}, \ldots, a_1, a_0 \in \mathbf{k}[x_1, \ldots, x_{i-1}].$$

- Algorithmic properties. Let $p \in \mathbf{k}[x_1, \ldots, x_n]$ with $\mathsf{deg}(p, x_n) > 0$ and define $\mathcal{I} = (T)$.
  - One can decide whether $p \in \mathcal{I}$. Indeed T is a Gr. basis of $\mathcal{I}$ w.r.t. $x_1 < \cdots < x_n$.
  - One can decide whether $p^{-1} \mod \mathcal{I}$ exists. Indeed, ...

  (1) For $n = 1$ by computing $\gcd(p, T_1)$. This may split T_1 into 2 factors.

(2) For $n = 2$ one can try to compute $\gcd(p, T_2)$ modulo T_1 (by running the Eucl. algo as if $\mathbf{k}[x_1]/(T_1)$ was a field).

If p is not invertible mod. $\mathcal{I}$ then triang. sets T', T'' appear s.t. $\sqrt{\mathcal{I}} \;=\; \sqrt{\mathcal{I}'} \cap \sqrt{\mathcal{I}''}$.

- Generalization to non-monic T_i's

(3) For $n = 3$

$$T \;=\; \left\{ \begin{array}{r} \mathbf{x_1^{d_1}} + \cdots + a_1\,\mathbf{x_1} + a_0 \\ h_2\,\mathbf{x_2^{d_2}} + \cdots + b_1\,\mathbf{x_2} + b_0 \\ h_3\,\mathbf{x_3^{d_3}} + \cdots + c_1\,\mathbf{x_3} + c_0 \end{array} \right.$$

is a triangular set if h_2^{-1} exists modulo T_1 and h_3^{-1} exists modulo $(\{T_1, T_2\}) : {h_2}^{\infty}$. Here $\mathcal{I}$ changes to $(T) : (h_2\,h_3)^{\infty}$.

(n) $\mathcal{I}$ changes to $\mathbf{Sat}(T) = (T) : (h_2\,\cdots\,h_n)^{\infty}$.

- The subset $T \subseteq \mathbf{k}[x_1 < \cdots < x_n]$ is a regular chain if
  - either $T = \emptyset$
  - or $T = T' \cup \{t\}$ where T' is a regular chain and
    * $t = h_t\, \mathbf{x_i^{d_i}} + a_{i-1}\, \mathbf{x_i^{d_i - 1}} + \cdots + a_1\, \mathbf{x_i} + a_0$
    * $T', \{h_t\, a_{i-1}, \ldots, a_1, a_0\} \subseteq \mathbf{k}[x_1, \ldots, x_{i-1}]$.
    * h_t is regular modulo $\mathbf{Sat}(T')$.
- Every prime ideal $\mathcal{P}$ in a polynomial ring $\mathbf{k}[x_1, \ldots, x_n]$ is given by a regular chain. That is, there exists a regular chain such that by $\mathcal{P} = \mathbf{Sat}(T)$.
- Algorithmic properties. One can decide whether $p \in \mathbf{k}[x_1, \ldots, x_n]$
  - belongs to $\mathbf{Sat}(T)$ (by pseudo-division)
  - belongs to $\sqrt{\mathbf{Sat}(T)}$ (by gcd computation)
  - is regular w.r.t. $\mathbf{Sat}(T)$ (by gcd computation)

- Notations
  - Let h be the product of the initials of T and s be the product of the separants of T. Define $m = \#T$.
  - Let $X = \{x_1, \ldots, x_n\}$, A the set of the $x \in X$ s.t. x is the greatest variable of some $t \in T$. Define $B = X \setminus A$.
- Reduction to dimension zero.
  - For every prime $\mathcal{P}$ associated with $\mathbf{Sat}(T)$ we have $\mathsf{dim}(\mathcal{P}) = n - m$ and $\mathcal{P} \cap \mathbf{k}[B] = \{0\}$.

  $\Rightarrow$ T can be viewed as a triangular set in $\mathbf{k}(B)[A]$.
- Reduction to product of fields.
  - For every prime $\mathcal{P}$ associated with $\mathcal{J} = (T) : s^\infty$ we have

  $$\mathsf{dim}(\mathcal{P}) = n - m \quad \text{and} \quad \mathcal{P} \cap \mathbf{k}[B] = \{0\}.$$

  Moreover $\mathcal{J}$ is radical (Lazard's Lemma).

- The common zeros of every polynomial system can be decomposed into finitely many triangular sets.
- For $x > y > a > b > c > d > e > f > g > h > i$ consider

$$\mathcal{P} = \begin{cases} ax + by - c \\ dx + ey - f \\ gx + hy - i \end{cases}$$

$$\Updownarrow$$

$$T = \begin{cases} gx + hy - i \\ (hd - eg)\, y - id + fg \\ (ie - fh)\, a + (ch - ib)\, d + (fb - ce)\, g \end{cases}$$

- We have $\mathcal{P} = \mathbf{Sat}(T)$ and

- $$\begin{aligned}\mathbf{V}(\mathcal{P}) &= \mathbf{W}(T) \cup \mathbf{W}\left\{\begin{array}{l} dx + ey - f \\ hy - i \\ (ie - fh)\, a + (-ib + ch)\, d \\ g \end{array}\right. \\ &\cup \mathbf{W}\left\{\begin{array}{l} gx + hy - i \\ (ha - bg)\, y - ia + cg \\ hd - eg \\ ie - fh \end{array}\right. \\ &\cup \mathbf{W}\left\{\begin{array}{l} x \\ (hd - eg)\, y - id + fg \\ fb - ce \\ ie - fh \end{array}\right. \quad \cup \cdots\end{aligned}$$

  where $\mathbf{W}(T)$ denotes the zeros of T that do not cancel its initials. Note that we have $\overline{\mathbf{W}(T)} = \mathbf{V}(\mathbf{Sat}(T))$.

- For $F \subseteq \mathbf{k}[X]$ one can compute decompositions of the form
  - $\mathbf{V}(F) = \cup_{i=1}^{\ell} \overline{\mathbf{W}(T_i)}$ or
  - $\mathbf{V}(F) = \cup_{i=1}^{\ell} \mathbf{W}(T_i)$.

- (Lazard, 1991), (Kalkbrener, 1993), (Moreno Maza, 2000)

Efforts are underway to

- develop a better **user interface**,
- make it useful as a **teaching tool**,
- integrate **additional mathematical libraries**,
- rebuild the algebra in a **literate programming style**,
- develop an **Axiom Journal** with refereed submissions,
- develop a Wiki site for interactive **web development**.