Solving Polynomial Systems Symbolically and in Parallel

Marc Moreno Maza & Yuzhen Xie

Ontario Research Center for Computer Algebra

University of Western Ontario, London, Canada

MITACS - CAIMS, June 18, 2006.

Solving polynomial systems . . .

$$\begin{cases} x^2 + y + z &= 1 \\ x + y^2 + z &= 1 \\ x + y + z^2 &= 1 \end{cases}$$

The output with `phc` the *symb.-num.* software of J. Verschelde:

```
solution 1 :    start residual :  3.968E-12   #iterations : 1   success
 x :  9.99999695984909E-01   4.13938269379988E-07
 y :  3.04015091103714E-07  -4.13938269379988E-07
 z :  3.04015090976779E-07  -4.13938269379988E-07
== err :  2.154E-06 = rco :  1.197E-07 = res :  9.920E-13 = complex regular ==
solution 2 :    start residual :  1.388E-16   #iterations : 1   success
 x :  4.14213562373095E-01   2.35098870164458E-38
 y :  4.14213562373095E-01  -1.67507944992176E-37
 z :  4.14213562373095E-01   1.29304378590452E-37
== err :  7.517E-16 = rco :  6.017E-02 = res :  5.551E-17 = real regular ==
```

```
solution 3 :    start residual :  2.400E-12   #iterations : 1   success
 x :  1.80048038888678E-08   4.29782537417684E-07
 y :  9.99999981995196E-01  -4.29782537417684E-07
 z :  1.80048038262633E-08   4.29782537417684E-07
== err :  1.344E-06 = rco :  7.463E-08 = res :  5.995E-13 = complex regular ==
```

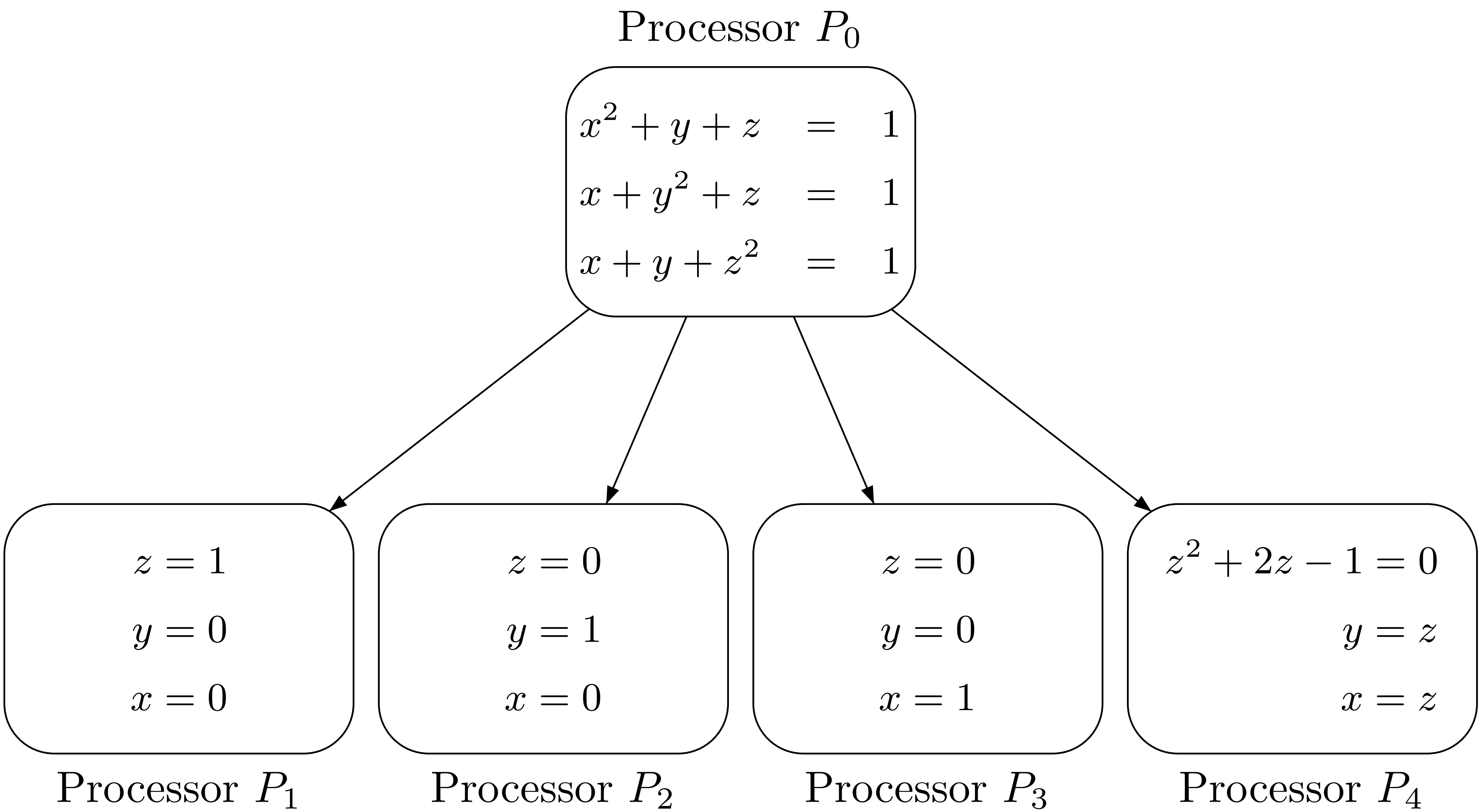
```
solution 4 :    start residual :  9.614E-13   #iterations : 1   success
 x :  1.00000024904061E+00  -3.93267692590196E-08
 y : -2.49040612161639E-07   3.93267692590197E-08
 z : -2.49040612108234E-07   3.93267692590197E-08
== err :  8.657E-07 = rco :  4.806E-08 = res :  2.400E-13 = complex regular ==
solution 5 :    start residual :  2.745E-12   #iterations : 1   success
 x :  3.58839953269127E-07   1.89357516639334E-07
 y :  3.58839953269127E-07   1.89357516639334E-07
 z :  9.99999641160047E-01  -1.89357516639334E-07
== err :  1.645E-06 = rco :  7.071E-08 = res :  6.863E-13 = complex regular ==
solution 6 :    start residual :  1.744E-34   #iterations : 1   success
 x : -2.41421356237309E+00   0.00000000000000E+00
 y : -2.41421356237309E+00   0.00000000000000E+00
 z : -2.41421356237309E+00  -1.00577224408752E-106
== err :  3.611E-35 = rco :  4.142E-01 = res :  6.868E-106 = real regular ==
solution 7 :    start residual :  1.112E-12   #iterations : 1   success
 x : -2.64786238552867E-07  -4.67724648385200E-08
 y : -2.64786238552867E-07  -4.67724648385200E-08
 z :  1.00000026478624E+00   4.67724648385200E-08
== err :  9.341E-07 = rco :  4.530E-08 = res :  2.779E-13 = complex regular ==
solution 8 :    start residual :  2.045E-12   #iterations : 1   success
 x :  1.42636460554469E-07  -3.16738323586431E-07
 y :  9.99999857363539E-01   3.16738323586431E-07
 z :  1.42636460467758E-07  -3.16738323586431E-07
== err :  1.378E-06 = rco :  7.656E-08 = res :  5.117E-13 = complex regular ==
==========================================================================
A list of 8 solutions has been refined :
Number of regular solutions   : 8.
Number of singular solutions  : 0.
Number of real solutions      : 2.
Number of complex solutions   : 6.
Number of clustered solutions : 0.
Number of failures            : 0.
```

Solving polynomial systems symbolically . . .

$$\left\{\begin{array}{l} x^2 + y + z = 1 \\ x + y^2 + z = 1 \\ x + y + z^2 = 1 \end{array}\right. \qquad \underline{\text{has Gröbner basis}} :$$

$$\left\{\begin{array}{r} z^6 - 4z^4 + 4z^3 - z^2 = 0 \\ 2z^2y + z^4 - z^2 = 0 \\ y^2 - y - z^2 + z = 0 \\ x + y + z^2 - 1 = 0 \end{array}\right. \quad \underline{\text{and triangular decomposition}} :$$

$$\left\{\begin{array}{l} z = 1 \\ y = 0 \\ x = 0 \end{array}\right. \bigcup \left\{\begin{array}{l} z = 0 \\ y = 1 \\ x = 0 \end{array}\right. \bigcup \left\{\begin{array}{l} z = 0 \\ y = 0 \\ x = 1 \end{array}\right. \bigcup \left\{\begin{array}{r} z^2 + 2z - 1 = 0 \\ y = z \\ x = z \end{array}\right.$$


Processor P_0
$x^2 + y + z = 1$
$x + y^2 + z = 1$
$x + y + z^2 = 1$
$z = 1$, $y = 0$, $x = 0$
Processor P_1
$z = 0$, $y = 1$, $x = 0$
Processor P_2
$z = 0$, $y = 0$, $x = 1$
Processor P_3
$z^2 + 2z - 1 = 0$, $y = z$, $x = z$
Processor P_4

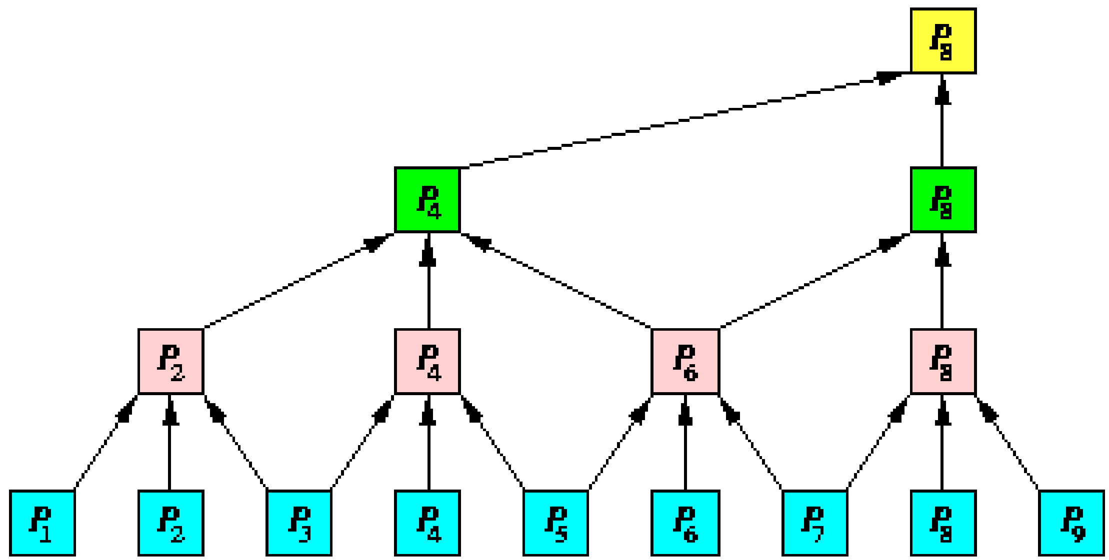
An example of efficient parallelization

Consider a tridiagonal linear system of order n:

$$
\begin{array}{cccccccc}
\cdots & \cdots & \cdots & & & & & \cdots \\
 & a_{i-2}x_{i-2} & +\, b_{i-1}x_{i-1} & +\, c_i x_i & & & = & e_{i-1} \\
 & & a_{i-1}x_{i-1} & +\, b_i x_i & +\, c_{i+1}x_{i+1} & & = & e_i \\
 & & & a_i x_i & +\, b_{i+1}x_{i+1} & +\, c_{i+2}x_{i+2} & = & e_{i+1} \\
 & & & & \cdots & \cdots & \cdots & \cdots
\end{array}
$$

For every even i replacing x_i with $-\frac{e_i - c_{i+1}x_{i+1} - a_{i-1}x_{i-1}}{b_i}$ leads to another tridiagonal system of order $n/2$:

$$
\begin{array}{cccccccc}
\cdots & \cdots & \cdots & & & & & \cdots \\
 & A_{i-3}x_{i-3} & +\, B_{i-1}x_{i-1} & +\, C_{i+1}x_{i+1} & & & = & E_{i-1} \\
 & & A_{i-1}x_{i-1} & +\, B_{i+1}x_{i+1} & +\, C_{i+3}x_{i+3} & & = & E_{i+1} \\
 & & & \cdots & \cdots & \cdots & & \cdots
\end{array}
$$



Observe that, on this example:

- the number of processors, here $p = n$, can be set such that
- the number of parallel steps, here $O(\log n)$, is known and small,
- processors activity (scheduling) is easy to organize,
- data-communication is not intensive.

Why solving non-linear systems is much more difficult?

Let $F \subset K[X]$ with $X = x_1 < \cdots < x_n$ and a coefficient field K. Let d be the maximum (total) degree of a monomial in F.

Let $V(F) \subset \overline{K}^n$ be the zero set of F, where $\overline{K}$ is an algebraically closed field containing K. For instance $K = Q$ and $\overline{K} = C$.

- $V(F)$ may consist of components of **different dimension**: points, curves, surfaces, ...,
- Even if $V(F)$ is finite, it may contain $O(d^n)$ points,
- The idea of *substitution* or *simplification* is much **more complicated** than in the linear case and leads to the notion of a *Gröbner basis*,
- **Large intermediate data**.

What is a Gröbner basis?

• Assume F is a linear system. Then, a solution of F is a **solved** system S for $x_1 < \cdots < x_n$ which reduces to 0 (i.e. cancels) all polynomials in F. Moreover, up to trivial transformations, the set S is unique.

• Now, assume that F is not linear. Then, a Gröbner basis of F is a system B which which reduces to 0 all polynomials in the ideal generated by F. Moreover, up to trivial transformations, the set B is unique.

$$\left\{\begin{array}{l} x^2+y+z=1 \\ x+y^2+z=1 \\ x+y+z^2=1 \end{array}\right. \quad \underline{\text{has Gröbner basis}}: \left\{\begin{array}{r} z^6-4z^4+4z^3-z^2=0 \\ 2z^2y+z^4-z^2=0 \\ y^2-y-z^2+z=0 \\ x+y+z^2-1=0 \end{array}\right.$$

Parallelizing the computation of Gröbner bases

Input: $F \subset K[X]$ and an admissible monomial ordering $\leq$.

Output: G a reduced Gröbner basis w.r.t. $\leq$ of the ideal $\langle F \rangle$ generated by F.

repeat
(S) $B :=$ MinimalAutoreducedSubset(F, $\leq$)
(R) $A :=$ S_Polynomials(F) $\cup F$;
$R :=$ Reduce(A, B, $\leq$)
(U) $R := R \setminus \{0\}$; $F := F \cup R$
until $R = \emptyset$
return B

(Bündgen, Göbel & W. Küchlin, 1994) (Chakrabarti & Yelick, 1993, 1994) (Attardi & Traverso, 1996) (Leykin, 2004)

To go further: triangular decompositions

• The zero set $V(F)$ admits a decomposition (unique when minimal)

$$V(F) = V(F_1) \cup \cdots \cup V(F_e),$$

s.t. $F_1, \ldots, F_e \subset K[X]$ and every $V(F_i)$ **cannot** be decomposed further.

• Moreover, for each $V(F_i)$ the following holds, up to renumbering the variables. If $V(F_i)$ has dimension d, then there exist polynomials $T_{d+1}, \ldots, T_n$ with respective main variables $x_{d+1}, \ldots, x_n$ and respective corresponding leading coefficients $h_{d+1}, \ldots, h_n$ such that

1. $h_{d+1}, \ldots, h_n$ are polynomials in $K[x_1, \ldots, x_d]$,
2. $\sqrt{\langle F_i \rangle} = \langle T_{d+1}, \ldots, T_n \rangle : h^\infty$ where $h = h_{d+1} \cdots h_n$.

Up to technical details, this means that each $V(F_i)$ is the zero set of a polynomial system with a **triangular shape**, called a *regular chain*.

Regular chains for the $V(F_i)$'s form a *triangular decomposition* of $V(F)$.

The characteristic set method

Input: $F \subset K[X]$ and a variable ordering $\leq$.

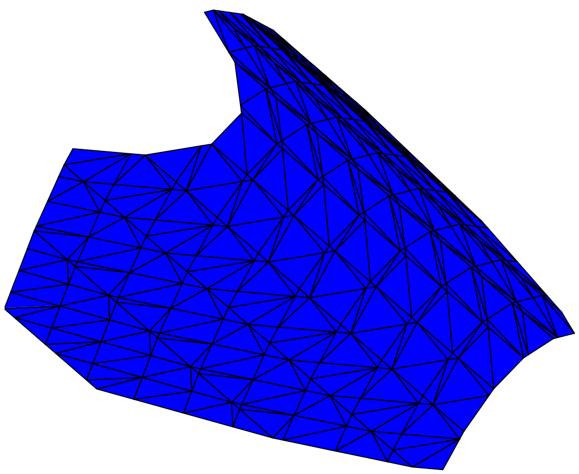
Output: C an autoreduced characteristic set of F (in the sense of Wu).

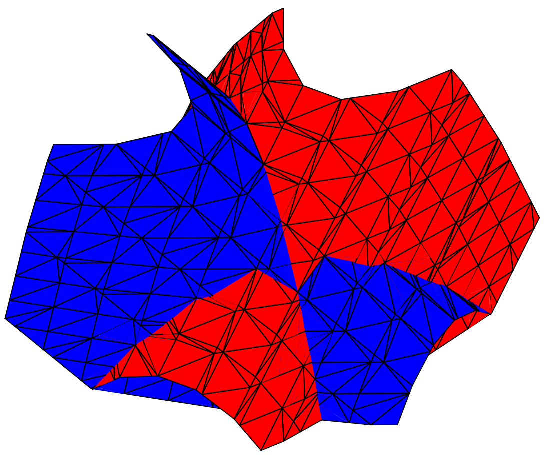
repeat
(S) $B :=$ MinimalAutoreducedSubset$(F, \leq)$
(R) $A := F \setminus B$;
$R :=$ PseudoReduce$(A, B, \leq)$
(U) $R := R \setminus \{0\}$; $F := F \cup R$
until $R = \emptyset$
return B

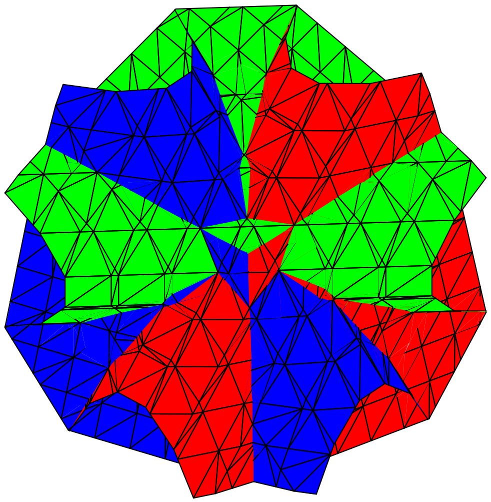
- Repeated calls to this procedure computes a decomposition of $V(F)$.
- Cannot start computing the 2nd component before the 1st is completed.
- (Ajwa, 1998), (Y.W. Wu, W.D. Liao, D.D. Liu & P.S. Wang, 2003) (Y.W. Wu, G.W. Yang, H. Yang, H.M. Zheng & D.D. Liu, 2005)

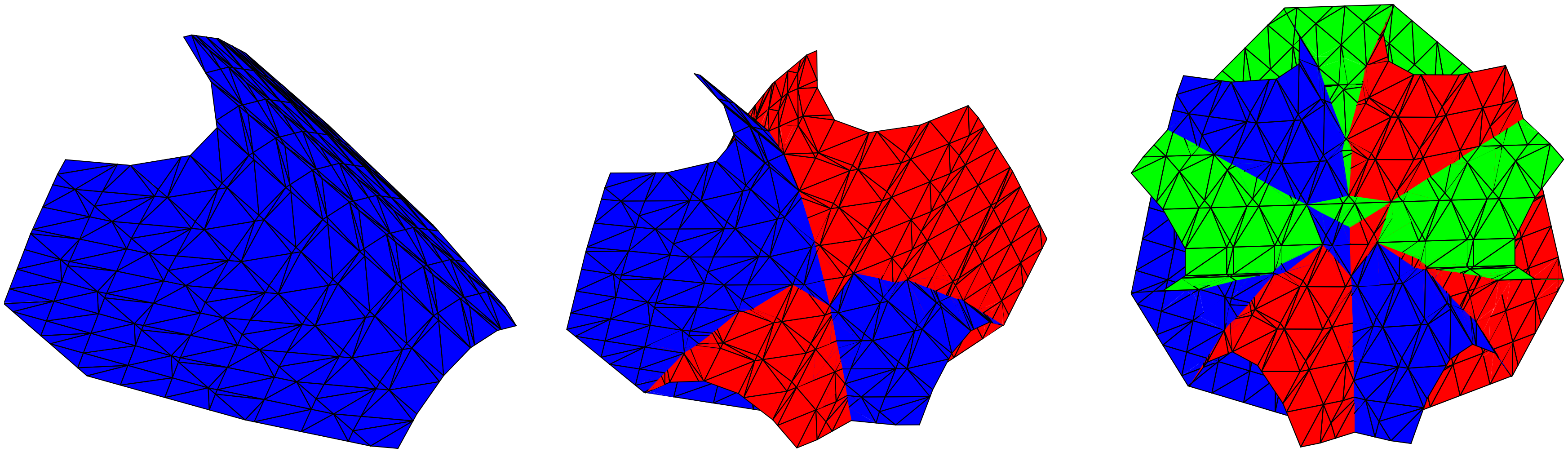
Triangular decompositions: a geometrical approach

$$\left\{ x^2 + y + z = 1 \right. \quad \left\{ \begin{array}{l} x^2 + y + z = 1 \\ x + y^2 + z = 1 \end{array} \right. \quad \left\{ \begin{array}{l} x^2 + y + z = 1 \\ x + y^2 + z = 1 \\ x + y + z^2 = 1 \end{array} \right.$$









$$\left\{\; x^2 + y + z = 1 \right.$$

$$\left\{\begin{aligned} x + y^2 + z &= 1 \\ y^4 + (2z-2)y^2 + y - z + z^2 &= 0 \end{aligned}\right.$$

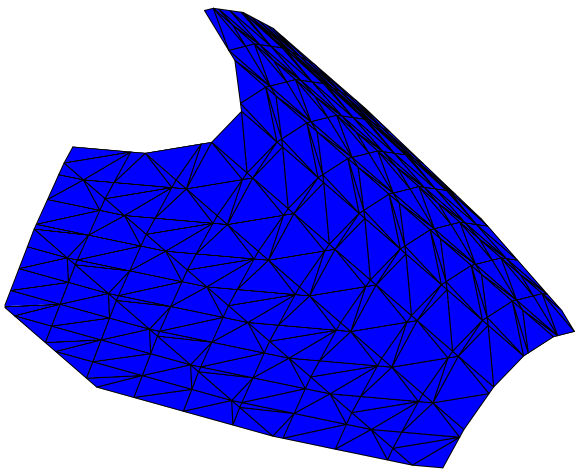
$$\left\{\begin{aligned} x + y \\ y^2 - y = z \end{aligned}\right.$$

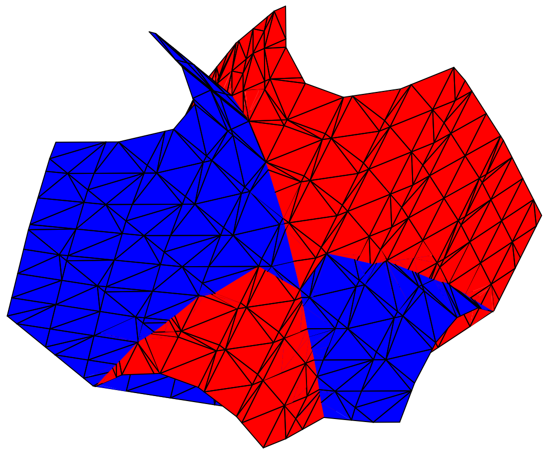
$$\left\{\begin{aligned} 2x + z^2 = 2y + z^2 \\ z^3 + z^2 - 3z = \end{aligned}\right.$$

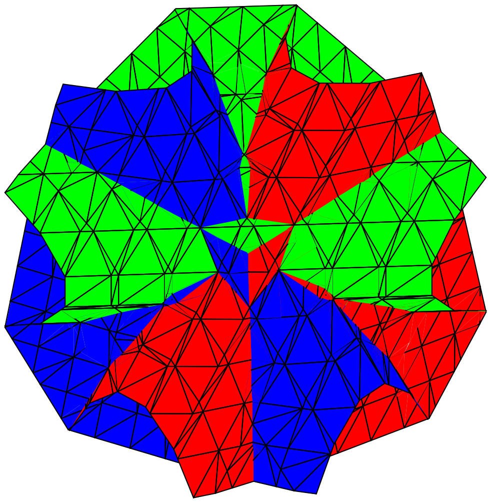
$$\left\{ x^2 + y + z = 1 \right.$$

$$\left\{ \begin{aligned} x^2 + y + z &= 1 \\ x + y^2 + z &= 1 \end{aligned} \right.$$

$$\left\{ \begin{aligned} x^2 + y + z &= 1 \\ x + y^2 + z &= 1 \\ x + y + z^2 &= 1 \end{aligned} \right.$$







$$\left\{ x^2 + y + z = 1 \right.$$

$$\left\{ \begin{aligned} x + y^2 + z &= 1 \\ y^4 + (2z - 2)y^2 + y - z + z^2 &= 0 \end{aligned} \right.$$

$$\left\{ \begin{aligned} x + y &= 1 \\ y^2 - y = z &= 0 \end{aligned} \right.$$

$$\left\{ \begin{aligned} 2x + z^2 = 2y + z^2 &= 1 \\ z^3 + z^2 - 3z &= -1 \end{aligned} \right.$$

Triangular decompositions: a task manager algorithm

A task is any $[F, T]$ where $F, T \subset K[X]$ with T regular chain. It is solved iff $F = \emptyset$ and unsolved otherwise.

Input: $F \subset K[X]$ and a variable ordering $\leq$.

Output: $\mathcal{T}$ a triangular decomposition of $V(F)$ by means of regular chains.

$ToDo := [[F, \emptyset]; \mathcal{T} := [\,]$

repeat

  if $ToDo = \emptyset$ **then break**

(S) $Tasks :=$ Select$(ToDo)$

(R) $Results :=$ LazySolve$(Tasks)$

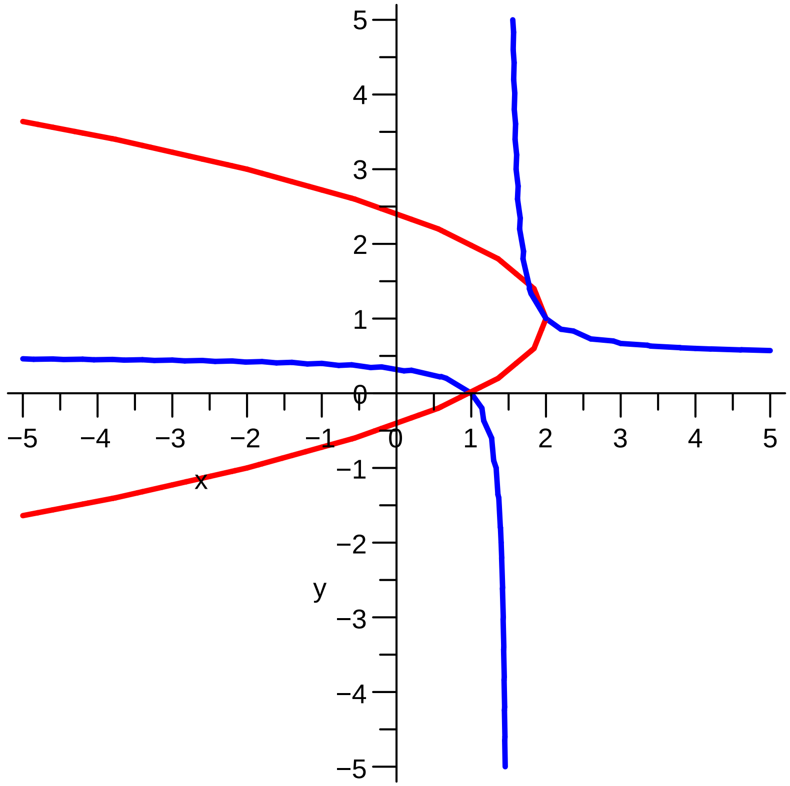
(U) $(ToDo, \mathcal{T}) :=$ Update$(Results, ToDo, \mathcal{T})$

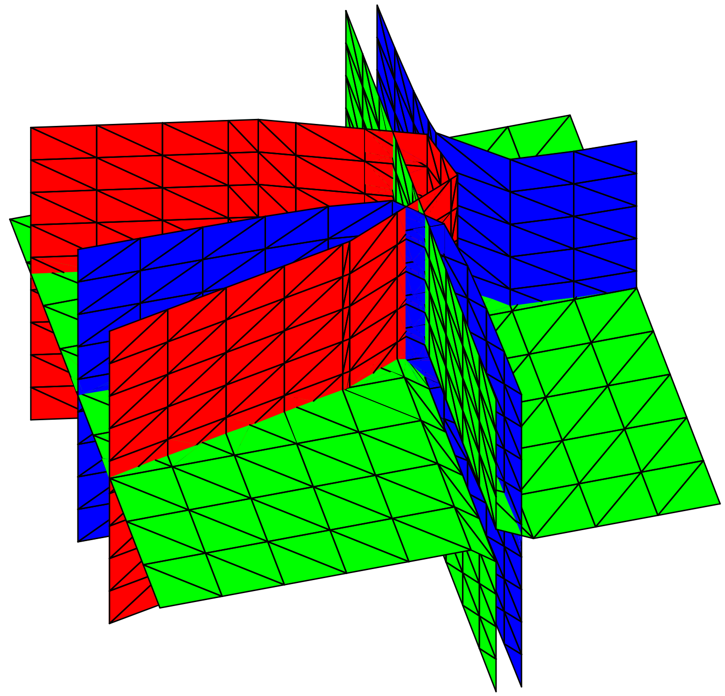
return $\mathcal{T}$

LazySolve$([F, T])$ returns $[F_1, T_1], \ldots, [F_d, T_d]$ which are less unsolved and:

$$V(F) \cap W(T) \ \subseteq \ \cup_{i=1}^{d} Z(F_i, T_i) \ \subseteq \ V(F) \cap \overline{W(T)}.$$

Difficulty 1: redundant and irregular tasks





The red and blue surfaces intersect on the line $x - 1 = y = 0$ contained in the green plane $x = 1$. With the other green plane $z = 0$, they intersect at $(2, 1, 0)$, $(\frac{7}{4}, \frac{3}{2}, 0)$ but also at $x - 1 = y = z = 0$, which is redundant.

Initial task $[\{f_1, f_2, f_3\}, \emptyset]$

$$\begin{aligned} f_1 &= x - 2 + (y-1)^2 \\ f_2 &= (x-1)(y-1) + (x-2)y \\ f_3 &= (x-1)z \end{aligned}$$

$$\begin{aligned} y &= 0 \\ x &= 1 \end{aligned}$$

$$\begin{aligned} x - 1 + y^2 - 2y &= 0 \\ (2y-1)x + 1 - 3y &= 0 \\ z &= 0 \end{aligned}$$

$$\begin{aligned} z &= 0 \\ y &= 0 \\ x &= 1 \end{aligned}$$

$$\begin{aligned} z &= 0 \\ y &= 1 \\ x &= 2 \end{aligned}$$

$$\begin{aligned} z &= 0 \\ 2y &= 3 \\ 4x &= 7 \end{aligned}$$

Difficulty 2: load balancing

- How do splits occur during decompositions? Gien a polynomial ideal $\mathcal{I}$ and polynomials p, a, b, there are two rules:
  - $\mathcal{I} \longmapsto (\mathcal{I} + p, \mathcal{I} : p^{\infty})$.
  - $\mathcal{I} + \langle a\, b \rangle \longmapsto (\mathcal{I} + \langle a \rangle, \mathcal{I} + \langle b \rangle)$.
- The second one is more likely to **split computations evenly**. But geometrically, it means that a component is **reducible**.
- Unfortunately, most polynomial systems $F \subseteq Q[X]$ (both in theory and practice) are **equiprojectable**, that is they can be represented by a single regular chain.
- However, for $F \subseteq Z/pZ[X]$ where p prime, the second rule is more likely to be used.

Key solutions

• We rely on the Triade algorithm (MMM, 2000) for computing triangular decompositions. In this case, LazySolve($(F,T]$) returns $[F_1,T_1],\ldots,[F_d,T_d]$ such that $\boxed{F_i = \emptyset \iff |T_i| = |T|}$ and thus, $\boxed{F_i \neq \emptyset \iff |T_i| > |T|.}$

$\Rightarrow$ We solve completely only in the cases where dimension does not drop and solve lazily the other cases.

$\Rightarrow$ **Computations in lower dimension are delayed toward the end** of the solving process.

• For solving $F \subseteq Q[X]$ we use modular methods (Dahan, MMM, Schost, Wu, Xie, 2005)

- For p big enough, a triangular decomposition of $V(F)$ can be **reconstructed (= merged + lifted)** from one of $V(F \mod p)$.
- The reconstruction is cheap (comparing to the decomposition phasis).
- This modular approach consumes less resources than the direct one.

A parallel scheme

Input: $F \subset K[X]$ and a variable ordering $\leq$.

Output: $\mathcal{T}$ a triangular decomposition of $V(F)$ by means of regular chains.

$ToDo := [[F, \emptyset]; \mathcal{T} := [\,]; d := n;$

repeat

if $ToDo = \emptyset$ **then break**

(1) **let** V be all tasks which can produce solved tasks of diemnsion d

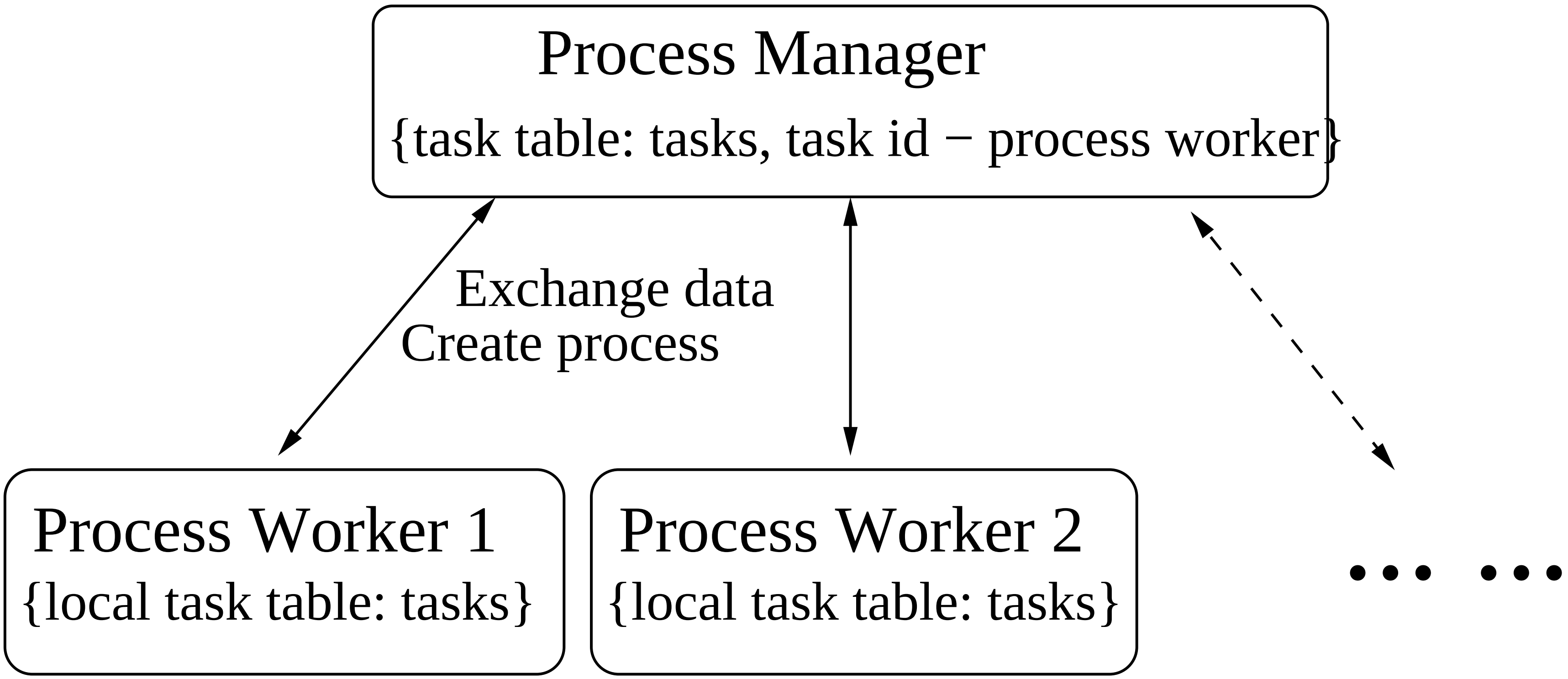
(2) **if** $V \neq \emptyset$ **then**

- lazy-solve these tasks
- update $ToDo$ and $\mathcal{T}$
- go to (1)

(3) **if** $V = \emptyset$ **then** $d := d - 1$ **and** go to (1)

return $\mathcal{T}$

Target implementation



Current implementation

• In Aldor on a 4-processor machine using shared memory for data-communication.

• Only the output components are generated by decreasing order of dimension. (This does not hold yet for the intermediate components)

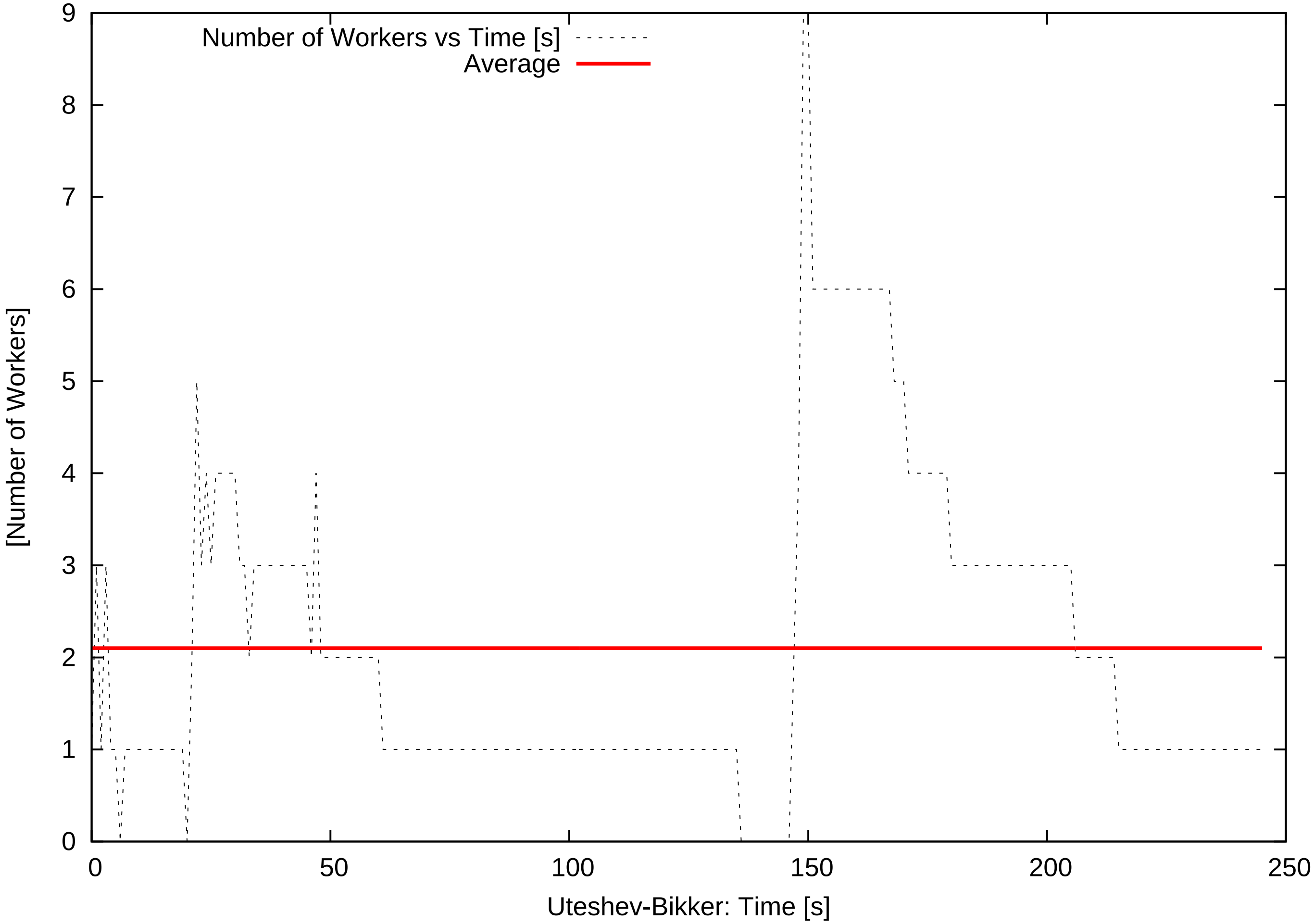
$\Rightarrow$ Hence, we do not implement yet the above parallel scheme, but only an approximation of it.

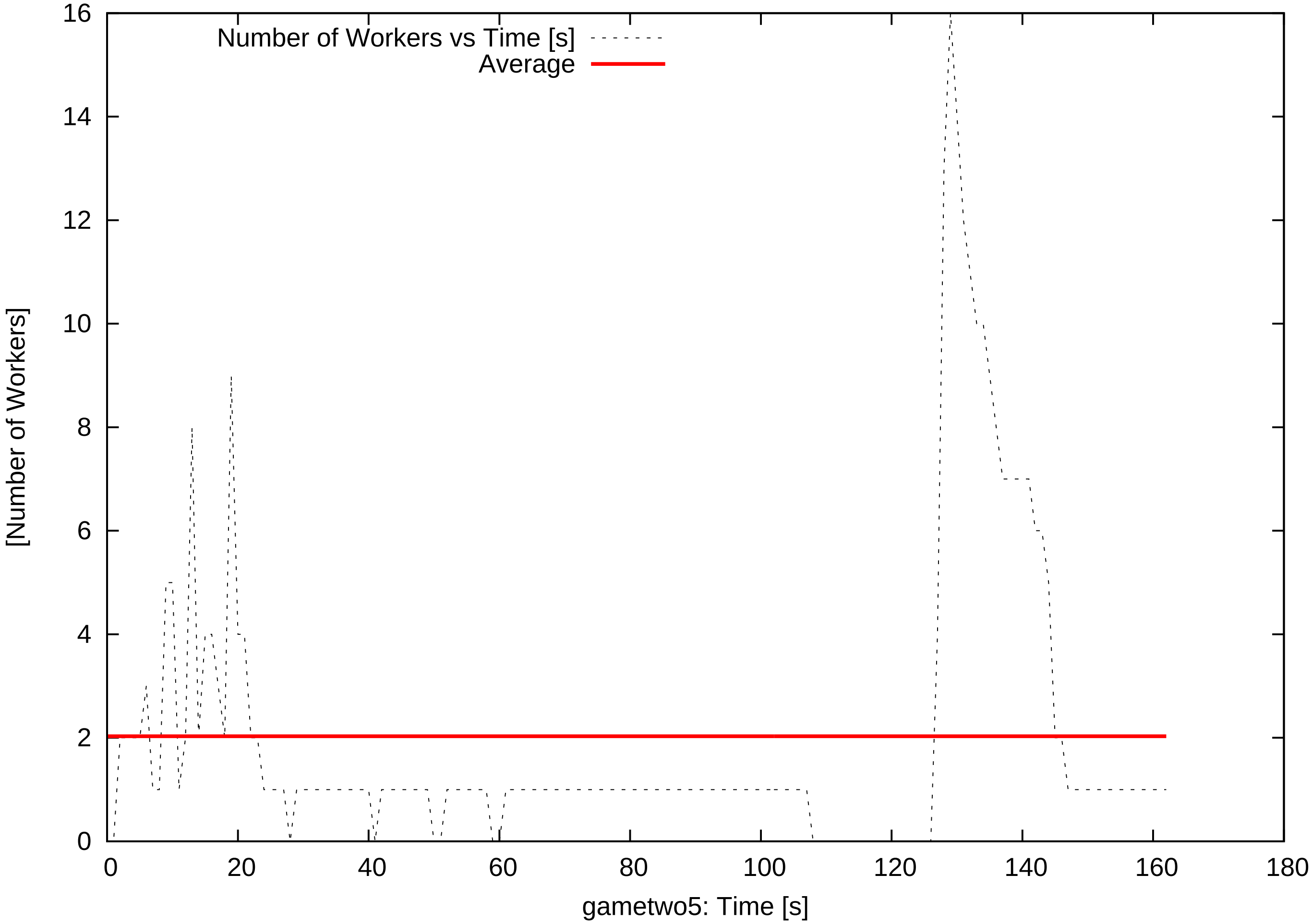
• Splitting (of the 2nd kind) relies only on the *D5 Principle* and univariate polynomial factorization.

• Each *LazySolve* requires to activate a process worker, which terminates after completing this computation.

$\Rightarrow$ Hence, we pay a severe penalty in data-communication and O/S calls w.r.t. our target implementation (work in progress).

Preliminay results





Number of Workers vs Time [s]
Average
[Number of Workers]
gametwo5: Time [s]
0
2
4
6
8
10
12
14
16
0
20
40
60
80
100
120
140
160
180

Work in progress and conclusions

- Combining the Triade algorithm and modular techniques, we have achieved successful **coarse-grain parallelization** of triangular decompositions **based on geometrical information** detected during the solving process.
- Future work:
  - Increasing the average number of working processors (by making use of multivariate factorization)
  - Reducing data-communicatio (with our target implementation scheme).
  - Making use of medium-grain parallelization (by parallelizing our GCDs/resultants).
- **Parallelizing helps removing arbitrary choices.**
- **Modular methods increase opportunities for parallelism.**