CS855: Parallel Symbolic Computation and Algebraic Geometry

Marc Moreno Maza

February 19, 2008

- Our main concern: Symbolic Computations.
- Our main tool: Parallel Computations.
- Our challenge: Solving Geometrical Problems.

- Typical symbolic computations are
  - solving univariate equations, such as $x^2 + x - 1 = 0$ by radicals,
  - factoring polynomials,
  - computing anti-derivatives of functions,
  - solving systems of linear (or non-linear) equations.
- Most of them are *high-school* problems, but:
  - the algorithms that we have learned in *high-school* can only handle *toy examples*.
  - there is a real need for handling large input data,
  - symbolic computations can involve intermediate huge intermediate data,
  - hence we need sophisticated algorithms and implementation techniques.

- Consider the system S (Barry Trager).

$$-x^5 + y^5 - 3y - 1 = 5y^4 - 3 = -20x + y - z = 0$$

We solve it for $z < y < x$.

- The exact solution is given below and has 9114 characters in total.

$$
\begin{array}{l}
- \\
114741279465692560074688619671388225994546322534047768700511994762226192690048901447618534394846710571230976934651913810508137045627329171252933709324791305415981639539600782016547479165073205735746803568230 40x - \\
17712605050082028621028540517021898341445070419214009122128543579469609331953356418583965018969358502883886989734236570248748901689518870061195207164038985884472656250000000000000z^{19} + \\
69934941672556438770604195551612193972977183106616813730136104734331616752952150977397654681986297393686589921444591865048249366533039077286429321101822204589843750000000000000000z^{18} + \\
4698033057372004369628572309403845943516901456096080945793282669881686485390936578666175235967213427460258099049103158338144237309897808986338911669057392838188681643681014342240 3125z^{17} - \\
3624577949980872265230642371971182386814553874346853792171708143077531532237850295577589142064921396560471587508817062087363711020822852702099485603348451121370559283022414296875000 00z^{16} + \\
182558840983144129257028601685384373297644771129092120128266359787322504095639220690574114668770499695595754316489285270876520168173545566353240056056199018421773934609375000000000000000z^{15} + \\
151384178460667251183582226588998788962467225266512277813388396930460206274093549761989465144274545813617140475288968551583152633110556311235297962315832575993037109375000000000000000000z^{14} - \\
443943358739034775586223820376199033996055435130191939848508110344015397674352445829758618270875644685197947702714342116764170792252091206631407369181847043216814116202609460618176641575 00z^{13} - \\
2398894638319738859704396544591592407731579470289955844307815442694326841805687077917675761917871130339864691191892935328273879968419204084615510666231818228388108541438912801652512500000000z^{12} + \\
27383396627989971288277129673535208075787121561611954126243384593168535690807541301547194521196228628235306090721309448057226671580596955987717404360616501191967139744465869750000000000000000z^{11} - \\
152371339486589977786933953445963421265232316881028589410282951401496074779560518480664573334972022843566414271419963537397962504790186241746199898415346140743494777559960937500000000000000000z^{10} + \\
4856391347410632777061560951110896275634940887029344611985724298328089928128704127659741470395314284711097054354196541536981977178916401637191689258348488593206522226647424422365687058076340 4750z^{9} + \\
1827709014752692114620308283759341810040325817543392095814567632394138225663551675690804005364380128824994168903072684742345947539419919102239134713365445883558417936978864823477746231323750000 00z^{8} + \\
30919129613095072997366859536802112563524969324865875138127923901717040322453163109045163040345690230109046354154120866101551210169691772309725826487133681798609204575827504904539025000000000000 0z^{7} - \\
6838688396641645490945090868618366582490420637673970853279869471018348887091817749546675847593376908651767451450243404728169565283801174082247345442363224813502338696761842209062500000000000000000 0z^{6} - \\
748156823800707525930652056310913558181154201465607063798861710733037765053357306037655291256264679716332849646414825289875704231589234990170319064622665507210419586793334912471724140809554018456514 20z^{5} - \\
1546080455275692923387543379737978438247137018552307587682361742927801505920906300566302345120640667639872360668265714450771926419524242832766806629792833533866149283795717893909570155702706314500000 00z^{4} - \\
12469538581957864228527528797540201566899450220047706509464051559860111513017516706370534366523919321363133021570560629259477339924222850862780138188159578900786902041841785279144486011017500000000000 000z^{3} - \\
66152659857188245320424888024222967738184293737891699176976594293187674688484864881423871033576765065422479234984449961643090211269389726039235842212801812250522818211946896210922615625000000000000000 0z^{2} +
\end{array}
$$

$$
\begin{gathered}
573598714920124956474610718803150703376812978417179178775576117319500000077857129232958889104193427114987500929833686714791341712743162700766075396541379832681132358445310329142895528887477470724804102079717z+\\
239787108649287987286424755607482454864690786827841184696976286133386057573817722098997859322480446751288360706756986017238407677469601203785469989972240351121714956603188344506951151541954650850823956500000\\
-\\
5737063973284628003734430983569411299727316126702388435025599738111309634502445072380926719742335528561548846732595690525406852281366458562646685466239565270799081976980039100827373958253660286787340178411152y-\\
177126050500820286210285405170218983414450704192140091221285435794696093319533564185839650189693585028838869897342365702487489016895188700611952071640389858844726562500000000000000z^{19}+\\
6993494167255643877060419555161219397297718310661681373013610473433161675295215097739765468198629739368658992144459186504824936653303907728642932110182220458984375000000000000000z^{18}+\\
46980330573720043696285723094038459435169014560960809457932826698816864853909365786661752359672134274602580990491031583381442373098978089863389116690573928381886816436810143422403125z^{17}-\\
3624577949980872265230642371971182386814553874346853792171708143077531532237850295577589142064921396560471587508817062087363711020822852702099485603348451121370559283022414296875000000z^{16}+\\
182558840983144129257028601685384373297644771129092120128266359787322504095639220690574114668770499695595754316489285270876520168173545566353240056056199018421773934609375000000000000z^{15}+\\
151384178460667251183582265889987889624672252665122778133883969304602062740935497619894651442745458136171404752889685515831526331105563112352979623158325759930371093750000000000000000z^{14}-\\
44394335873903477558622382037619903399605543513019193984850811034401539767435244582975861827087564468519794770271434211676417079225209120663140736918184704321681411620260946061817664157500z^{13}-\\
239889463831973885970439654459159240773157947028995584430781544269432684180568707791767576191787113033986469119189293532827387996841920408461551066623181822838810854143891280165251250000000z^{12}+\\
273833966279899712882771296735352080757871215616119541262433845931685356908075413015471945211962286282353060907213094480572266715805969559877174043606165011919671397444658697500000000000000000z^{11}-\\
152371339486589977786933953445963421265232316881028589410282951401496074779560518480664573334972022843566414271419963537397962504790186241746199898415346140743494777559960937500000000000000000z^{10}+\\
48563913474106327770615609511108962756349408870293446119857242983280899281287041276597414703953142847110970543541965415369819771789164016371916892583484885932065222266474244223656870580763404750z^{9}+\\
182770901475269211462030828375934181004032581754339209581456763239413822566355167569080400536438012882499416890307268474234594753941991910223913471336544588355841793697886482347774623132375000000z^{8}+\\
309191296130950729973668595368021125635249693248658751381279239017170403224531631090451630403456902301090463541541208661015512101696917723097258264871336817986092045758275049045390250000000000000z^{7}-\\
683868839664164549094509086861836658249042063767397085327986947101834888709181774954667584759337690865176745145024340472816956528380117408224734544236322481350233869676184220906250000000000000000000z^{6}-\\
74815682380070752593065205631091355818115420146560706379886171073303776505335730603765529125626467971633284964641482528987570423158923499017031906462266550721041958679333491247172414080955401845651420z^{5}-\\
154608045527569292338754337973797843824713701855230758768236174292780150592090630056630234512064066763987236066826571445077192641952424283276680662979283353386614928379571789390957015570270631450000000z^{4}-\\
124695385819578642285275287975402015668994502200477065094640515598601115130175167063705343665239193213631330215705606292594773399242228508627801381881595789007869020418417852791444860110175000000000000000z^{3}-\\
661526598571882453204248880242229677381842937378916991769765942931876746884848648814238710335767650654224792349844499616430902112693897260392358422128018122505228182119468962109226156250000000000000000z^{2}-\\
107682408337843898832379553790426595918634253059664726983856491630963372387378005133782870040125741167383743425882337749343515393483155498593150082576694398775839252693580939841866937888557953929915761435z+\\
239787108649287987286424755607482454864690786827841184696976286133386057573817722098997859322480446751288360706756986017238407677469601203785469989972240351121714956603188344506951151541954650850823956500000\\
-\ 3125z^{20}-9375z^{16}-40000000000z^{15}-2015999988750z^{12}-1560000000000z^{11}+\\
1920000000000000000z^{10}-12165125356800006750z^{8}-14745602232000000000z^{7}-\\
6528000000000000000z^{6}-4096000000000000000000000z^{5}-\\
169869086392333478399979752z^{4}-1415576715264030240000000z^{3}-\\
5898238732800000000000000z^{2}-12288000000000000000000000z-\\
61953036192319828787324416000243
\end{gathered}
$$

- An approximate solution is given below

```
===========================================================
solution 1 :
t :  0.00000000000000E+00   0.00000000000000E+00
m : 1
the solution for t :
 x :  8.05224664844877E-01   5.92969846725325E-01
 y : -2.50273775222733E-01   9.68175106804426E-01
 z :  6.86409502730791E-01   7.27215232624336E-01
== err :  0.000E+00 = rco :  1.000E+00 = res :  0.000E+00 ==
solution 2 :
t :  0.00000000000000E+00   0.00000000000000E+00
m : 1
the solution for t :
 x :  8.05224664844877E-01   5.92969846725325E-01
 y : -9.68175106804426E-01  -2.50273775222733E-01
 z :  6.86409502730791E-01   7.27215232624336E-01
== err :  0.000E+00 = rco :  1.000E+00 = res :  0.000E+00 ==
solution 3 :
t :  0.00000000000000E+00   0.00000000000000E+00
m : 1
the solution for t :
 x :  8.05224664844877E-01   5.92969846725325E-01
 y :  2.50273775222733E-01  -9.68175106804426E-01
 z :  6.86409502730791E-01   7.27215232624336E-01
== err :  0.000E+00 = rco :  1.000E+00 = res :  0.000E+00 ==
solution 4 :
t :  0.00000000000000E+00   0.00000000000000E+00
m : 1
the solution for t :
```

```
 x :  8.05224664844877E-01   5.92969846725325E-01
 y :  9.68175106804426E-01   2.50273775222733E-01
 z :  6.86409502730791E-01   7.27215232624336E-01
== err :  0.000E+00 = rco :  1.000E+00 = res :  0.000E+00 ==
solution 5 :
t :  0.00000000000000E+00   0.00000000000000E+00
m : 1
the solution for t :
 x : -3.15119730967720E-01   9.49051924372335E-01
 y : -2.50273775222733E-01   9.68175106804426E-01
 z :  6.86409502730791E-01   7.27215232624336E-01
== err :  0.000E+00 = rco :  1.000E+00 = res :  0.000E+00 ==
solution 6 :
t :  0.00000000000000E+00   0.00000000000000E+00
m : 1
the solution for t :
 x : -3.15119730967720E-01   9.49051924372335E-01
 y : -9.68175106804426E-01  -2.50273775222733E-01
 z :  6.86409502730791E-01   7.27215232624336E-01
== err :  0.000E+00 = rco :  1.000E+00 = res :  0.000E+00 ==
solution 7 :
t :  0.00000000000000E+00   0.00000000000000E+00
m : 1
the solution for t :
 x : -3.15119730967720E-01   9.49051924372335E-01
 y :  2.50273775222733E-01  -9.68175106804426E-01
 z :  6.86409502730791E-01   7.27215232624336E-01
== err :  0.000E+00 = rco :  1.000E+00 = res :  0.000E+00 ==
solution 8 :
t :  0.00000000000000E+00   0.00000000000000E+00
m : 1
```

```
the solution for t :
 x : -3.15119730967720E-01   9.49051924372335E-01
 y :  9.68175106804426E-01   2.50273775222733E-01
 z :  6.86409502730791E-01   7.27215232624336E-01
== err :  0.000E+00 = rco :  1.000E+00 = res :  0.000E+00 ==
solution 9 :
t :  0.00000000000000E+00   0.00000000000000E+00
m : 1
the solution for t :
 x : -9.99979369108651E-01  -6.42350037472706E-03
 y : -2.50273775222733E-01   9.68175106804426E-01
 z :  6.86409502730791E-01   7.27215232624336E-01
== err :  0.000E+00 = rco :  1.000E+00 = res :  0.000E+00 ==
solution 10 :
t :  0.00000000000000E+00   0.00000000000000E+00
m : 1
the solution for t :
 x : -9.99979369108651E-01  -6.42350037472706E-03
 y : -9.68175106804426E-01  -2.50273775222733E-01
 z :  6.86409502730791E-01   7.27215232624336E-01
== err :  0.000E+00 = rco :  1.000E+00 = res :  0.000E+00 ==
solution 11 :
t :  0.00000000000000E+00   0.00000000000000E+00
m : 1
the solution for t :
 x : -9.99979369108651E-01  -6.42350037472706E-03
 y :  2.50273775222733E-01  -9.68175106804426E-01
 z :  6.86409502730791E-01   7.27215232624336E-01
== err :  0.000E+00 = rco :  1.000E+00 = res :  0.000E+00 ==
solution 12 :
t :  0.00000000000000E+00   0.00000000000000E+00
```

```
m : 1
the solution for t :
 x : -9.99979369108651E-01  -6.42350037472706E-03
 y :  9.68175106804426E-01   2.50273775222733E-01
 z :  6.86409502730791E-01   7.27215232624336E-01
== err :  0.000E+00 = rco :  1.000E+00 = res :  0.000E+00 ==
solution 13 :
t :  0.00000000000000E+00   0.00000000000000E+00
m : 1
the solution for t :
 x : -3.02901507190103E-01  -9.53021865930664E-01
 y : -2.50273775222733E-01   9.68175106804426E-01
 z :  6.86409502730791E-01   7.27215232624336E-01
== err :  0.000E+00 = rco :  1.000E+00 = res :  0.000E+00 ==
solution 14 :
t :  0.00000000000000E+00   0.00000000000000E+00
m : 1
the solution for t :
 x : -3.02901507190103E-01  -9.53021865930664E-01
 y : -9.68175106804426E-01  -2.50273775222733E-01
 z :  6.86409502730791E-01   7.27215232624336E-01
== err :  0.000E+00 = rco :  1.000E+00 = res :  0.000E+00 ==
solution 15 :
t :  0.00000000000000E+00   0.00000000000000E+00
m : 1
the solution for t :
 x : -3.02901507190103E-01  -9.53021865930664E-01
 y :  2.50273775222733E-01  -9.68175106804426E-01
 z :  6.86409502730791E-01   7.27215232624336E-01
== err :  0.000E+00 = rco :  1.000E+00 = res :  0.000E+00 ==
solution 16 :
```

```
t :  0.00000000000000E+00   0.00000000000000E+00
m : 1
the solution for t :
 x : -3.02901507190103E-01  -9.53021865930664E-01
 y :  9.68175106804426E-01   2.50273775222733E-01
 z :  6.86409502730791E-01   7.27215232624336E-01
== err :  0.000E+00 = rco :  1.000E+00 = res :  0.000E+00 ==
solution 17 :
t :  0.00000000000000E+00   0.00000000000000E+00
m : 1
the solution for t :
 x :  8.12775942421597E-01  -5.82576404792269E-01
 y : -2.50273775222733E-01   9.68175106804426E-01
 z :  6.86409502730791E-01   7.27215232624336E-01
== err :  0.000E+00 = rco :  1.000E+00 = res :  0.000E+00 ==
solution 18 :
t :  0.00000000000000E+00   0.00000000000000E+00
m : 1
the solution for t :
 x :  8.12775942421597E-01  -5.82576404792269E-01
 y : -9.68175106804426E-01  -2.50273775222733E-01
 z :  6.86409502730791E-01   7.27215232624336E-01
== err :  0.000E+00 = rco :  1.000E+00 = res :  0.000E+00 ==
solution 19 :
t :  0.00000000000000E+00   0.00000000000000E+00
m : 1
the solution for t :
 x :  8.12775942421597E-01  -5.82576404792269E-01
 y :  2.50273775222733E-01  -9.68175106804426E-01
 z :  6.86409502730791E-01   7.27215232624336E-01
== err :  0.000E+00 = rco :  1.000E+00 = res :  0.000E+00 ==
```

```
solution 20 :
t :  0.00000000000000E+00   0.00000000000000E+00
m : 1
the solution for t :
 x :  8.12775942421597E-01  -5.82576404792269E-01
 y :  9.68175106804426E-01   2.50273775222733E-01
 z :  6.86409502730791E-01   7.27215232624336E-01
== err :  0.000E+00 = rco :  1.000E+00 = res :  0.000E+00 ==
```

Consider another system of equations (Rose).

- $7y^4 - 20x^2 = 0,$
- $2160x^2z^4 + 1512xz^4 + 315z^4 - 4000x^2 - 2800x - 490 = 0,$
- $-10080000x^4z^3 - 28224000x^3z^3 - 15288000x^2z^3 - 1978032xz^3 - 180075z^3 - 23520000x^4yz^2 - 41395200x^3yz^2 - 26726560x^2yz^2 - 7727104xyz^2 - 852355yzz + 40320000x^6y^2z + 28800000x^5y^2z + 21168000x^3y^2z + 4939200x^2y^2z + 347508xy^2z + 67200000x^5y^3 + 94080000x^4y^3 + 40924800x^3y^3 + 2634240x^2y^3 - 2300844xy^3 - 432180y^3 = 0.$

Its solution consists of two components. One is:

- $548033463712323993600000000000000000000 * x^32 + 2333056745518179287040000000000000000000 * x^31 - 2226041095342245543936000000000000000000 * x^30 - 3790391477535038767104000000000000000000 * x^29 - 1305289515920841108357120000000000000000 * x^28 - 2683206421164416760702566400000000000000 * x^27 - 3848388720869152005040373760000000000000 * x^26 - 4009175679465270287674638336000000000000 * x^25 - 2960548468428654517092876288000000000000 * x^24 -$

$1298888648464449888212484096000000000000000 * x^{2}3 +$
$114561491838721185563956740096000000000000 * x^{2}2 +$
$81904020695138658077291115970560000000000 * x^{2}1 +$
$9013025778409579141347219918028800000000 * x^{2}0 +$
$679905821802410780888723538051072000000 * x^{1}9 +$
$40115498555156633343975255535779840000 0 * x^{1}8 +$
$1749259143999785821516198545929011200000 * x^{1}7 + 3305191208952212959053697481126707200 0 *$
$x^{1}6 - 2930472345357336293117016660812759040 0 * x^{1}5 -$
$38754561454772375365752789716930396160 * x^{1}4 - 26594578416681394692866717715849019392 *$
$x^{1}3 - 1314713018970901973899923612961374208 0 * x^{1}2 -$
$5058107255180177091775777126427033600 * x^{1}1 - 1558972754249315266527324764709316608 *$
$x^{1}0 - 388791989377015748494145767326228480 * x^{9} - 782829141214917302737492051057612 80 *$
$x^{8} - 125538597299813657224687341103779 84 * x^{7} - 155883945805759926431891523890112 0 *$
$x^{6} - 14207451168396615041891868575616 0 * x^{5} - 846532701905096189539080787046 4 * x^{4} -$
$21768463689533556626130612560 0 * x^{3} + 837437834259590753904215375 0 * x^{2} +$
$761038687880048862537750000 * x + 137454646239245673597656 25 = 0,$

- $7 * y^{4} - 20 * x^{2} = 0,$
- $283168745717760000000000000 * x^{2}2 + 805008291397632000000000000 * x^{2}1 +$
$899470110228480000000000000 * x^{2}0 + 1491677593731072000000000000 * x^{1}9 +$
$351081568319569920000000000 0 * x^{1}8 + 4877613724871688192000000000 * x^{1}7 +$
$3814948995421726310400000000 * x^{1}6 + 1498415397695432294400000000 * x^{1}5 +$
$96574870970597376000000000 * x^{1}4 + 339548392207472394240000000 * x^{1}3 +$
$102682091100307886899200000 * x^{1}2 - 265000751049684374323200000 * x^{1}1 -$
$620378401409915004702720000 * x^{1}0 - 611394874196233136486400000 * x^{9} -$
$379121528771643391035187200 * x^{8} - 165907531210801696135034880 * x^{7} -$
$53621839648134286638380544 * x^{6} - 12961877796408012932261760 * x^{5} -$
$2324798379975294721233120 * x^{4} - 302173771156655400996672 * x^{3} - 27264000562672589474640 *$
$x^{2} - 1566802525349007426900 * x - 44873357946792493125) * z + (471947909529600000000000000 *$
$x^{2}1 + 1665301909340160000000000000 * x^{2}0 + 2475384443397734400000000000 * x^{1}9 +$
$2945734044077260800000000000 * x^{1}8 + 4831655616696287232000000000 * x^{1}7 +$

$$
\begin{aligned}
&7860175213153458585600000000 * x^{1}6 + 9439939808104071168000000000 * x^{1}5 + \\
&8012618067480436408320000000 * x^{1}4 + 4702845020394747985920000000 * x^{1}3 + \\
&1574342255114247035289600000 * x^{1}2 - 382837453156615646085120000 * x^{1}1 - \\
&1181421093339053372252160000 * x^{1}0 - 1154747438950292537462784000 * x^{9} - \\
&750722120601129703329587200 * x^{8} - 356747173557691035923937280 * x^{7} - \\
&127596426656909424014499328 * x^{6} - 34656751075408356933969280 * x^{5} - \\
&7097071266874137980687200 * x^{4} - 1066495809914660472049024 * x^{3} - \\
&111172187565430329162160 * x^{2} - 7185580909304773168900 * x - 217207466976968421625) * y = 0
\end{aligned}
$$

The other one is:

- $20 * x + 7 = 0,$
- $20 * y^{4} - 7 = 0,$
- $z = 0.$

No! For the following kinds of problems, they are the only way:

- Computing canonical forms, or deciding whether two input systems have the same set of solutions.
- Looking for real solutions (and discarding those that a have a non-zero imaginary part).
- Cryptography, auto-correcting codes.

- The concerns of *pure mathematicians*, especially before the 60's, were mainly existence and uniqueness of solutions of polynomial systems.
- From the 60's, computers have opened new doors and brought new problems.
- As we have seen before, there is a complexity challenge in performing symbolic computations.
- Symbolic (or exact) computations may require a lot computer resources and special techniques to manage them. This is a job for computer scientists.

$$W = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \quad H = \begin{vmatrix} H_0 \\ H_1 \\ H_2 \\ H_3 \end{vmatrix} \quad P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad u = \begin{vmatrix} 1 \\ z_1 \\ z_2 \\ z_1 z_2 \end{vmatrix}$$

$$D(z_1, z_2) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & z_1 & 0 & 0 \\ 0 & 0 & z_2 & 0 \\ 0 & 0 & 0 & z_1 z_2 \end{bmatrix} \quad R_i = \begin{bmatrix} cos\alpha_i & -sin\alpha_i & 0 & 0 \\ sin\alpha_i & cos\alpha_i & 0 & 0 \\ 0 & 0 & cos\beta_i & -sin\beta_i \\ 0 & 0 & sin\beta_i & cos\beta_i \end{bmatrix}$$

$$H = A_0 A_1 \cdots A_K u \;\; \text{avec} \;\; A_0 = R_0 W P, \; A_i = D(z_1, z_2) P W R_i W P.$$

Given K, compute the maximum N such that $S(K, N)$ possesses real roots. Then solve $S(K, N_{max})$.

$$S(K, N) = \Big\{ \frac{\partial^k H_i}{\partial^{k_1} z_1 \partial^{k_2} z_2}(+/-1, +/-1) = 0$$
$$i = 1, 2, 3, 4 \;\; 0 \le k = k_1 + k_2 \le N \Big\}.$$

Compute the complex numbers $A_2, \ldots, A_k$ with module 1 such that:

$N=2$ $\quad \sum_{k=1}^{K} A_k = 0$

$N=3$ $\quad \sum_{k=2}^{K} \frac{1}{A_k} \left(\sum_{j=1}^{k-1} A_j - \sum_{j=k+1}^{K} A_j \right) = -\frac{1}{4}$

$N=4$ $\quad \sum_{k=2}^{K-1} \frac{1}{A_k} \left(\sum_{j=k+1}^{K} A_j \right) \left(\sum_{j=1}^{k-1} A_j \right) = -\frac{1}{8}$

$N=5$ $\quad \sum_{i=4}^{K} \sum_{j=2}^{i-1} \frac{1}{A_i A_j} \left(\sum_{k=1}^{j-1} A_k - \sum_{i+1}^{K} A_k \right) \left(\sum_{k=j+1}^{i-1} A_k \right) = \frac{4K-7}{32}$

Every equation $f(A_1, A_2, \ldots) = 0$ leads to $f(\frac{1}{A_1}, \frac{1}{A_2}, \ldots) = 0$.

K	N_{max}	complex dim
5	3	2
6	3	2
7	4	1
8	5	1

Involved software: GB, RealSolving, AXIOM.
Involved people: F. Moreau de Saint-Martin, F. Rouillier,
J.-C. Faugère, M. Moreno Maza.

$$\begin{cases} (2u_5 + 2u_4 + 2u_3 + 2u_2 + 1)\, U_6 + 2 \\ u_5 U_5 - 1, u_4 U_4 - 1 \\ u_3 U_3 - 1, u_2 U_2 - 1 \\ 2u_6 + 2u_5 + 2u_4 + 2u_3 + 2u_2 + 1 \\ \left(\left(128{u_2}^3 + 512{u_2}^2 + 512u_2\right) {u_3}^5 + \cdots\right) u_5 + \cdots \\ \left(\left(20{u_2}^2 + 56u_2 + 32\right) {u_3}^2 + \cdots\right) {u_4}^2 + \cdots \end{cases}$$

A *conjugate gradient method* is used to compute the *free variables*.
In 1997 we obtained the most regular (known at that date)
orthogonal centro-symetric wavelets (with a Sobolev coefficient 2.11).

The corresponding filter bank lead to a patent in image compression.

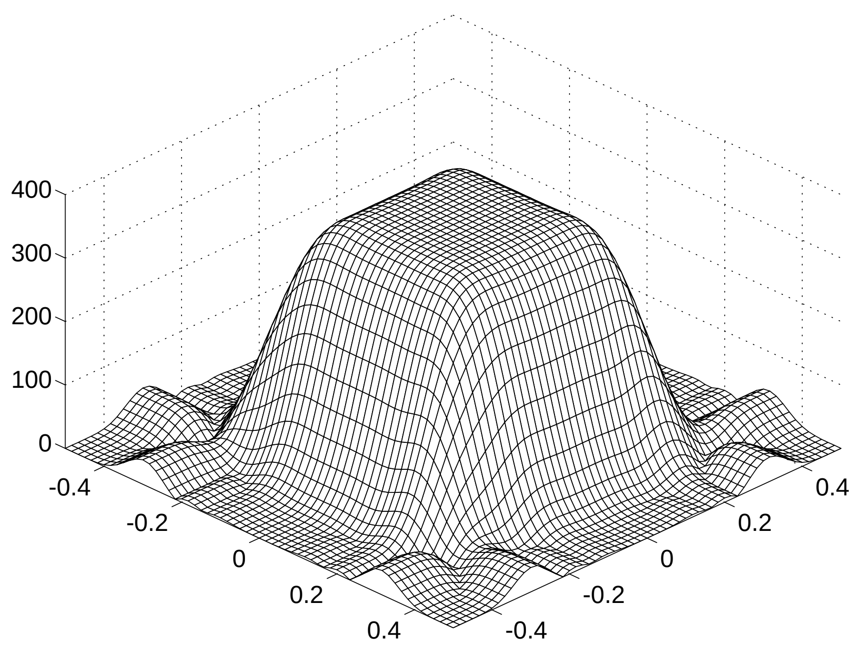


Figure 1: The scale function of the $S(8,5)$ problem.

We look for systems of coupled evolution differential equations that possess *solitons*. Solitons are traveling waves that recover their shape after collision.

$$\begin{cases} u_t &= u_{xxxxx} - 10uu_{xxx} + 30vv_{xxx} - 25u_xu_{xx} \\ & \quad +45v_xv_{xx} + 20u^2u_x - 30v^2u_x - 60uvv_x \\ v_t &= -9v_{xxxxx} + 10vu_{xxx} + 30uv_{xxx} + 35v_xu_{xx} \\ & \quad +45u_xv_{xx} - 20uvu_x - 20u^2v_x - 30v^2v_x \end{cases}$$

- These systems are extremely rare and there is no algorithms to construct them.
- However we can select candidates in (very) large classes.
- Here's the class where the above system was found.

$$\begin{cases} u_t &= a_1 u_{xxxxx} + a_2 v_{xxxxx} \\ & +a_3 u u_{xxx} + a_4 v u_{xxx} + a_5 u v_{xxx} + a_6 v v_{xxx} \\ & +a_7 u_x u_{xx} + a_8 v_x u_{xx} + a_9 u_x v_{xx} + a_{10} v_x v_{xx} \\ & +a_{11} u^2 u_x + a_{12} u v u_x + a_{13} v^2 u_x \\ & +a_{14} u^2 v_x + a_{15} u v v_x + a_{16} v^2 v_x \\ v_t &= b_1 u_{xxxxx} + b_2 v_{xxxxx} \\ & +b_3 u u_{xxx} + b_4 v u_{xxx} + b_5 u v_{xxx} + b_6 v v_{xxx} \\ & +b_7 u_x u_{xx} + b_8 v_x u_{xx} + b_9 u_x v_{xx} + b_{10} v_x v_{xx} \\ & +b_{11} u^2 u_x + b_{12} u v u_x + b_{13} v^2 u_x \\ & +b_{14} u^2 v_x + b_{15} u v v_x + b_{16} v^2 v_x \end{cases}$$

To study such class we consider a system of algebraic equations which can only be solved **symbolically.**

- However this system has typically 120 unknowns and 500 equations.
- Mikhail Foursov (ENST, France) and myself developed special techniques for these systems.
- Our computations allowed a complete classification of the systems generalizing those of Korteweg-de Vries, Sawada-Kotera and Kaup-Kupershmidt.
- We found 2 new systems possessing solitons.

$$\left\{ \begin{array}{rcl} u_t & = & u_{xxx} + 2vu_x + uv_x \\ v_t & = & uu_x \end{array} \right. \tag{1}$$

$$\left\{ \begin{array}{rcl} u_t & = & 4u_{xxx} + 3v_{xxx} + 4uu_x + vu_x + 2uv_x \\ v_t & = & 3u_{xxx} + v_{xxx} - 4vu_x - 2uv_x - 2vv_x \end{array} \right. \tag{2}$$

- Single-processor machine cannot keep evolving following Moore's' law.
- Moreover, some symbolic computations may require much more memory resources than we have on an usual PC.
- Therefore, for significant improvements we need to use parallel computations.
- However, we should try
  - to distribute the work evenly among the processors (or machines),
  - to keep communication costs within acceptable limits.
- Indeed, parallel programs may run fast on some architectures and surprisingly slowly on others!

- Let $T(n, p)$ the time elapsed
  - from the start of a parallel computation to the moment where the last processor finishes,
  - on an input data of size n,
  - and using p processors.
- $T(n, p)$ takes into account
  - computational steps (such as adding, multiplying, swapping variables),
  - routing (or communication) steps (such as transferring and exchanging information between processors).
- **Example:** parallel search of an item x
  - in an unsorted input file with n items,
  - in a shared memory with p processors,
  - where any cell can be accessed by only processor at a time.

Broadcasting x costs at least $O(\log(p))$, leading to

$$T(n,p) = O(\log(p)) + O(n/p).$$

- Let $E(n,p)$ be the *parallel efficiency*, or *speedup per processor*, that is

$$E(n,p) = \frac{SU(n)}{pT(n,p)}$$

where $SU(n)$ is a lower bound for a sequential execution. Observe that we have $SU(n) \leq p\,T(n,p)$ and thus $E(n,p) \leq 1$.

- Reasons for inefficiency:
  - large communication latency compared to computational performances (it would be better to calculate locally rather than remotely)
  - too big overhead in synchronization, poor coordination, poor load distribution (processors must wait for dependent data),
  - lack of useful work to do (too many processors for too little work).

- Performance and scalability of parallel algorithms, PRAM and BSP models.
- Parallel complexity theory: circuit complexity.
- Parallel complexity theory: communication complexity.
- Routine algorithms.
- Broadcast algorithms.
- Parallel dense matrix operations.
- Solving systems of linear equations in parallel.
- Parallel computations of GCDs.
- Triangular decomposition algorithms.
- Applications.
- Project presentations.

- Familiarity with *linear algebra*.
- Familiarity with *complexity analysis of algorithms*.
- No background in symbolic computations or parallel computations is required.

One research project based on articles and leading to a

- a report and
- a presentation in class.

Here's a first example of an algorithm that is *naturally parallel*. As for the *Karatsuba* algorithm, the trick relies on a smart use of distributivity:

(1) If $n = 1$ the matrices A et B are of the form (a) et (b) where $a, b \in \mathbf{Z}$, such that $A\,B$ equals $(a\,b)$.

(2) If $n > 1$ the matrices A and B can be decomposed as

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \ B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix},$$

where A_{ij} and B_{ij} are square matrices of order $n/2$. Them, one computes the following 8 sums.

$$\begin{aligned} S_1 &= A_{21} + A_{22} & T_1 &= B_{12} - B_{11} \\ S_2 &= S_1 - A_{11} & T_2 &= B_{22} - T_1 \\ S_3 &= A_{11} - A_{21} & T_3 &= B_{22} - B_{12} \\ S_4 &= A_{12} - S_2 & T_4 &= T_2 - B_{21} \end{aligned}$$

(3) Next, one computes the following 7 matrix products by recursive calls to the algorithm.

$$\begin{aligned} P_1 &= A_{11}\, B_{11} & P_5 &= S_1\, T_1 \\ P_2 &= A_{12}\, B_{21} & P_6 &= S_2\, T_2 \\ P_3 &= S_4\, B_{22} & P_7 &= S_3\, T_3 \\ P_4 &= A_{22}\, T_4 & & \end{aligned}$$

(4) Finally, one computes the following 7 sums:

$$
\begin{aligned}
U_1 &= P_1 + P_2 & U_5 &= U_4 + P_3 \\
U_2 &= P_1 + P_6 & U_6 &= U_3 - P_4 \\
U_3 &= U_2 + P_7 & U_7 &= U_3 + P_5 \\
U_4 &= U_2 + P_5
\end{aligned}
$$

and one returns:

$$
\begin{pmatrix} U_1 & U_5 \\ U_6 & U_7 \end{pmatrix}.
$$

Given a polynomial system: $\begin{cases} f1 = x^3 - 3x^2 + 2x = 0 \\ f2 = 2yx^2 - x^2 - 3yx + x = 0 \\ f3 = zx^2 - zx = 0 \end{cases}$

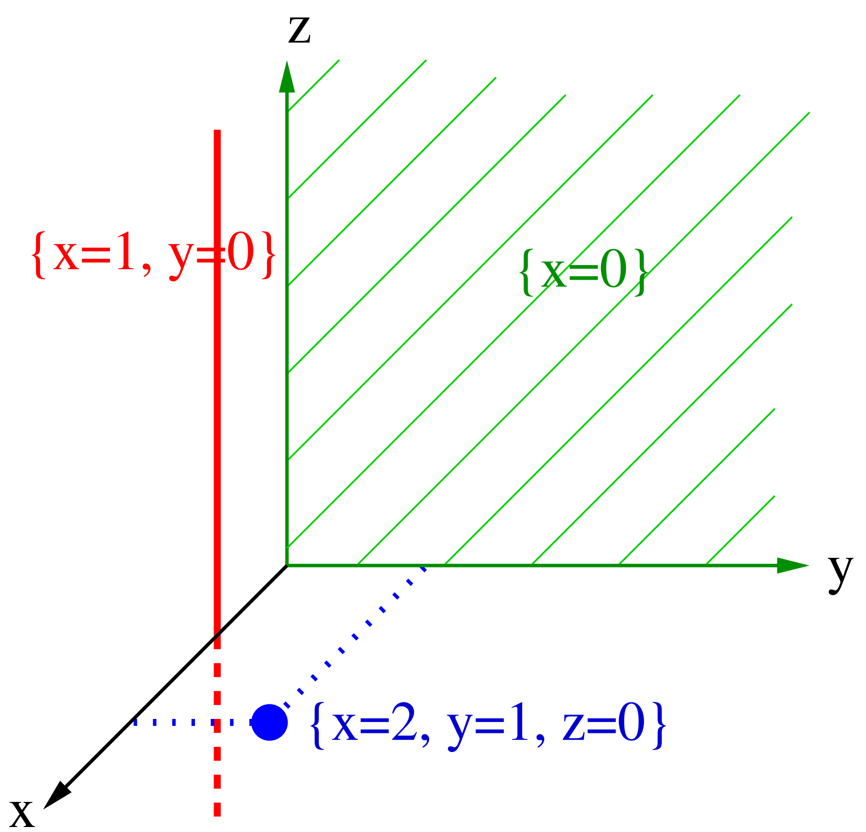
It <u>has lex $(x > y > z)$ Gröbner basis</u> :

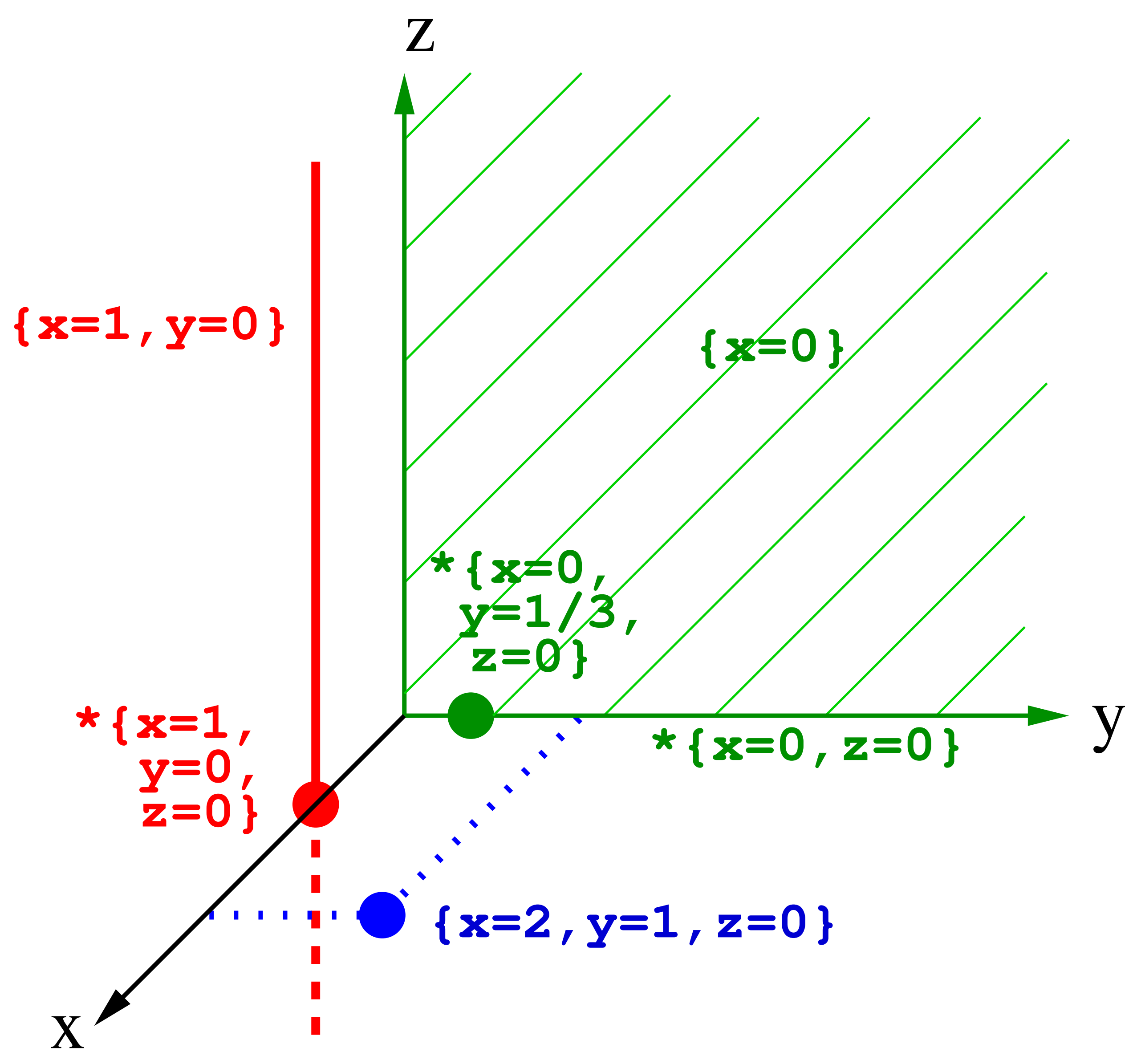
$$\begin{cases} x^2 - xy - x = 0 \\ -xy + xy^2 = 0 \\ zxy = 0 \end{cases}$$

<u>and TRIAngular DEcomposition</u> :

$$\begin{cases} x = 0 \end{cases} \quad \text{or} \quad \begin{cases} y = 0 \\ x = 1 \end{cases} \quad \text{or} \quad \begin{cases} z = 0 \\ y = 1 \\ x = 2 \end{cases}$$

```
f1 = x(x-1)(x-2)
f2 = x((x-1)(y-1)+(x-2)y)
f3 = x(x-1)z
```




z
{x=1,y=0}
{x=0}
*{x=0,
y=1/3,
z=0}
*{x=1,
y=0,
z=0}
y
*{x=0,z=0}
{x=2,y=1,z=0}
x

The removal of redundant components is a central issue.

- (Wu Wen Tsun, 1987) First method. Produce many redundant components without being able to remove them.
- (Kalkbrener, 1991) Produces less redundant components, but those which are produced are still expensive to remove.
- (Lazard, 1991) Checking that a component is contained in another is easier, but these checks may happen quite late. So many intermediate computations can be useless.
- (Moreno Maza, 1999) Good management of the computation: redundant branches are easy to cut and can be cut at an early stage.

This latter algorithm has three (sequential) implementations:
- in the Computer Algebra System AXIOM, since 1998
- in the Aldor library, since 1999
- in the Computer Algebra System Maple (available in Maple 10, this summer)

```
f1 = x^3 - 3x^2 + 2x
f2 = 2yx^2 - x^2 - 3yx + x
f3 = zx^2 - zx             [x > y > z]
```

Task0 (T0): `[{f1,f2,f3},{}]` phase 0

- A *task* is a pair $[F, T]$ where F is a system to solve and T is a regular chain (= **a solved system**, e.g. a set of algebraic constraints).
- A task represents a computation which goes on.
- With a system F to solve we associate the *initial* task $[F, \emptyset]$.
- The output will be a list of the form $[[\emptyset, T_1], \ldots, [\emptyset, T_s]]$.
- The expect output here:
  $[[\emptyset, \{x\}], [\emptyset, \{x-1, y\}], [\emptyset, \{x-2, y-1, z\}]$.

```
f1 = x^3 - 3x^2 + 2x
f2 = 2yx^2 - x^2 - 3yx + x
f3 = zx^2 - zx            [x > y > z]
```

Task0 (T0): `[{f1,f2,f3},{}]` phase 0

- First we choose T0 to transform (process), here,
  - select $f3$ (it has the smallest variable z and the smallest total degree)
  - $f3 = x(x-1)z$ by factorization
  - then we split into three cases: $x = 0$, $x = 1$, and $z = 0$.
  - then evaluating $f1$ and $f2$ w.r.t each case, we get three tasks in phase 1.

```
f1 = x^3 - 3x^2 + 2x
f2 = 2yx^2 - x^2 - 3yx + x
f3 = zx^2 - zx              [x > y > z]
```

T0: `[{f1,f2,f3},{}]` phase 0

T1: `[{},{x}]` **T2:** `[{y},{x-1}]` **T3:** `[{f1,f2},{z}]` phase 1

```
f1 = x^3 - 3x^2 + 2x
f2 = 2yx^2 - x^2 - 3yx + x
f3 = zx^2 - zx              [x > y > z]
```

T0: `[{f1,f2,f3},{}]` phase 0

T1: `[{},{x}]` **T2:** `[{y},{x-1}]` **T3:** `[{f1,f2},{z}]` phase 1

- T1 is done
- We choose T3 to transform, here,
  - check $f2$, which factorizes into $x(2yx - x - 3y + 1)$
  - then T3 changes to T4 and T5 in phase 2.

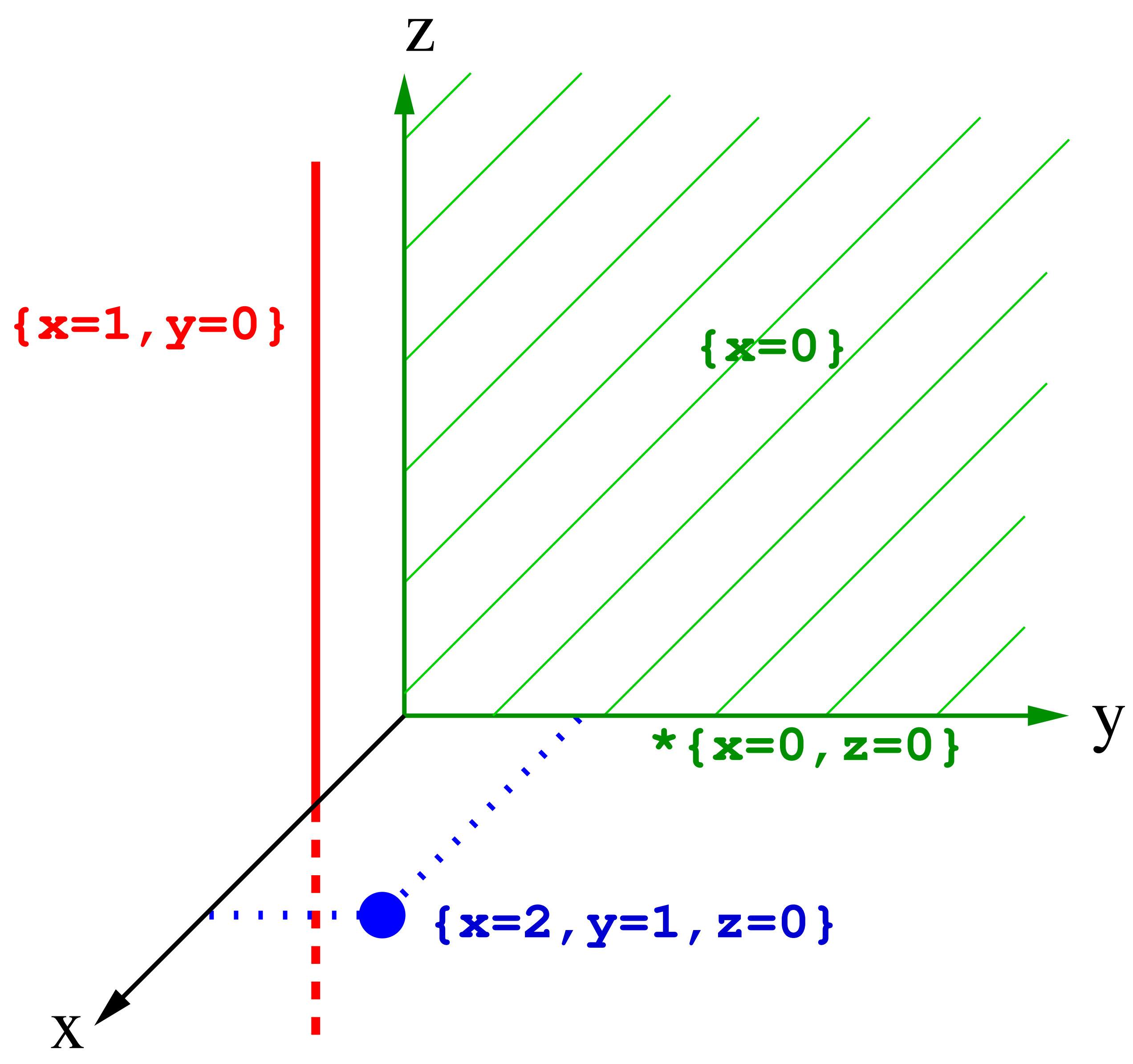
```
f1 = x^3 - 3x^2 + 2x
f2 = 2yx^2 - x^2 - 3yx + x
f3 = zx^2 - zx              [x > y > z]
```

T0: [{f1,f2,f3},{}] phase 0

T1: [{},{x}] T2: [{y},{x-1}] T3: [{f1,f2},{z}] phase 1

phase 2

T4: [{},{x,z}] T5: [{f1},{f2/x = 2yx-x-3y+1,z}]


z
{x=1,y=0}
{x=0}
y
*{x=0,z=0}
{x=2,y=1,z=0}
x

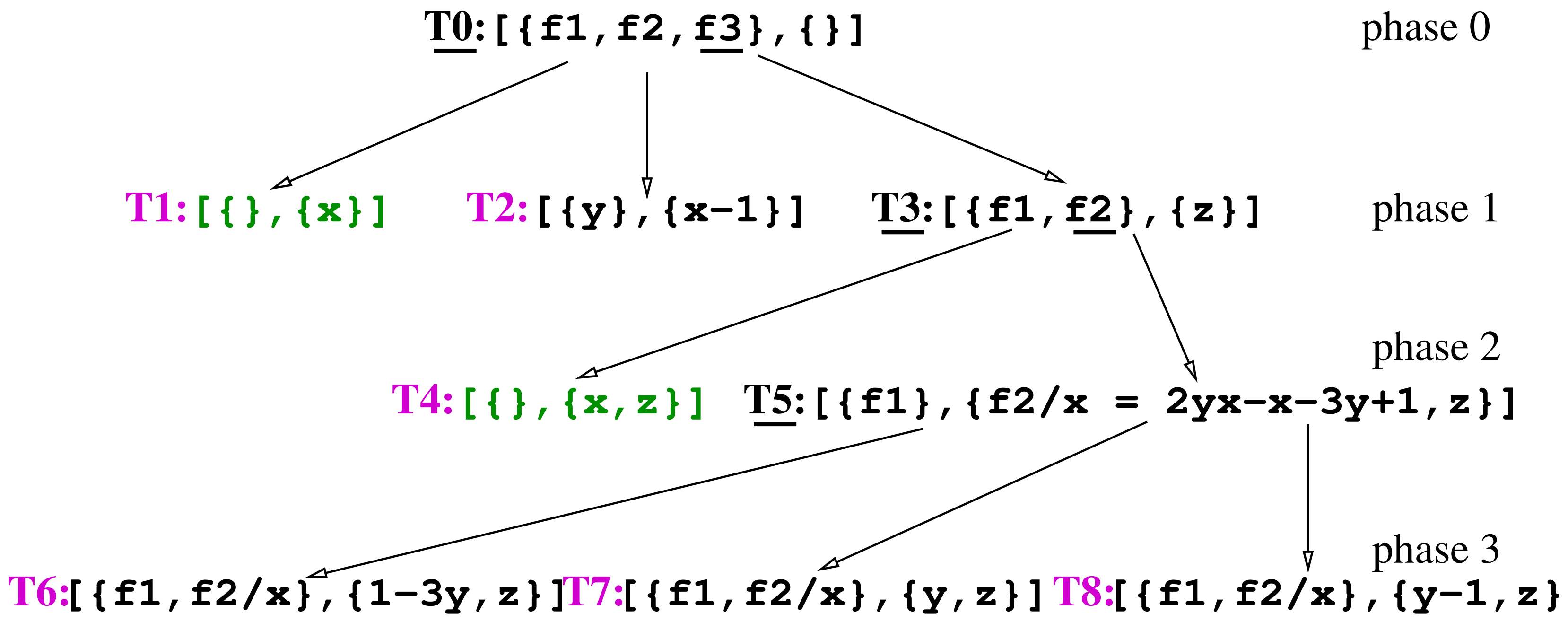
- Choose T5, we need to compute the resultant of $f1$ and $f2/x$.
- This leads to compute the determinant of a special matrix (Sylvester):

$$\mathbf{res} = \mathbf{det}\begin{pmatrix} 1 & -3 & 2 & 0 \\ 2y-1 & 1-3y & 0 & 0 \\ 0 & 2y-1 & 1-3y & 0 \\ 0 & 0 & 2y-1 & 1-3y \end{pmatrix}$$

$res = 3y^3 - 4y^2 + y = (3y-1)(y-1)y$

- Then T5 changes to three tasks in each case where $3y - 1 = 0$, $y - 1 = 0$, and $y = 0$.
- T5 is a very expensive task! (hard to achieve by hand).
- It could have intermediate expression swell and cause memory overflow.

```
f1 = x^3 - 3x^2 + 2x
f2 = 2yx^2 - x^2 - 3yx + x
f3 = zx^2 - zx              [x > y > z]
```



```
f1 = x^3 - 3x^2 + 2x
f2 = 2yx^2 - x^2 - 3yx + x
f3 = zx^2 - zx                [x > y > z]
```

T0: [{f1,f2,f3},{}] phase 0

T1: [{},{x}] T2: [{y},{x-1}] T3: [{f1,f2},{z}] phase 1

phase 2

T4: [{},{x,z}] T5: [{f1},{f2/x = 2yx-x-3y+1,z}]

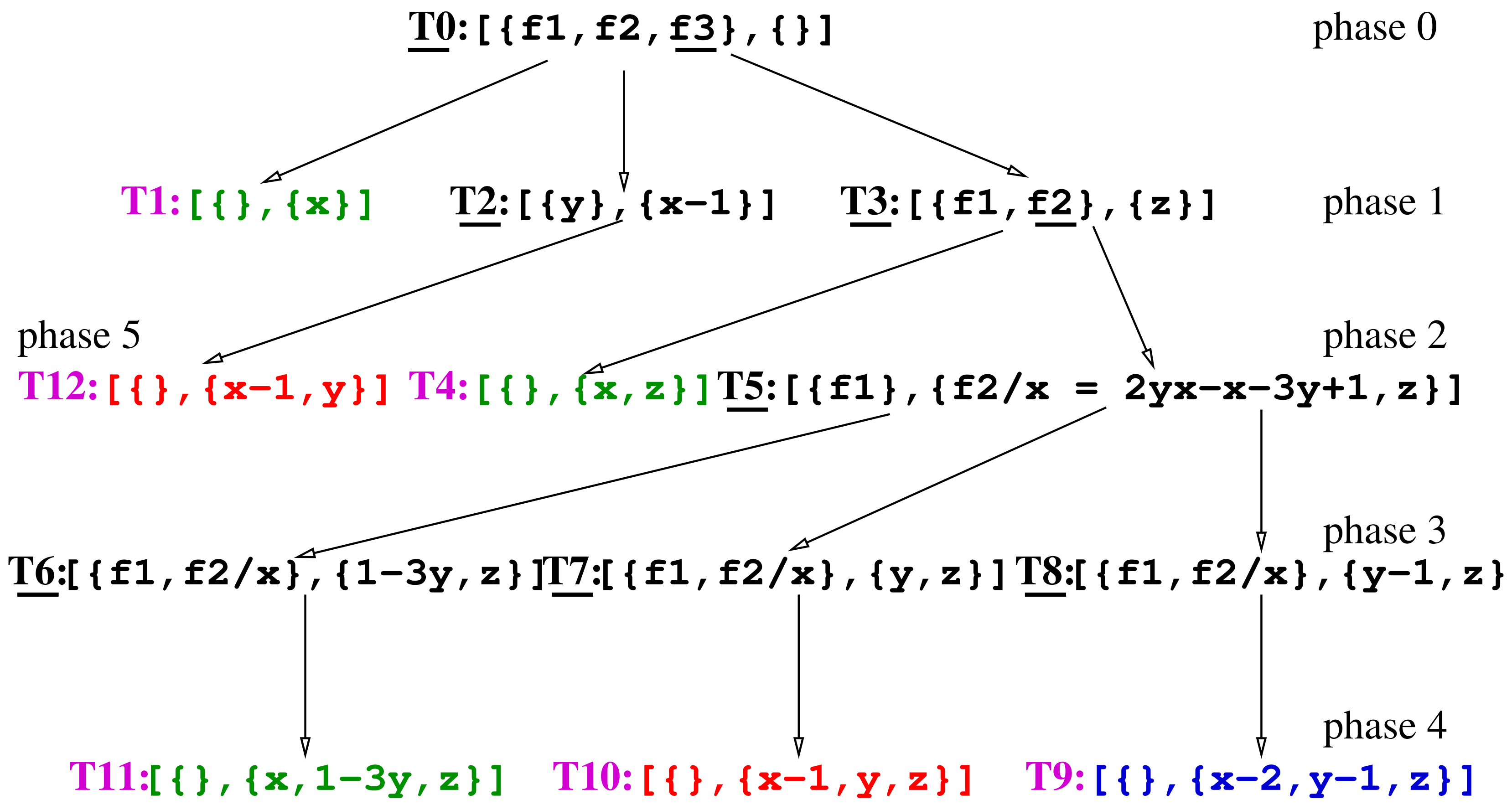
phase 3

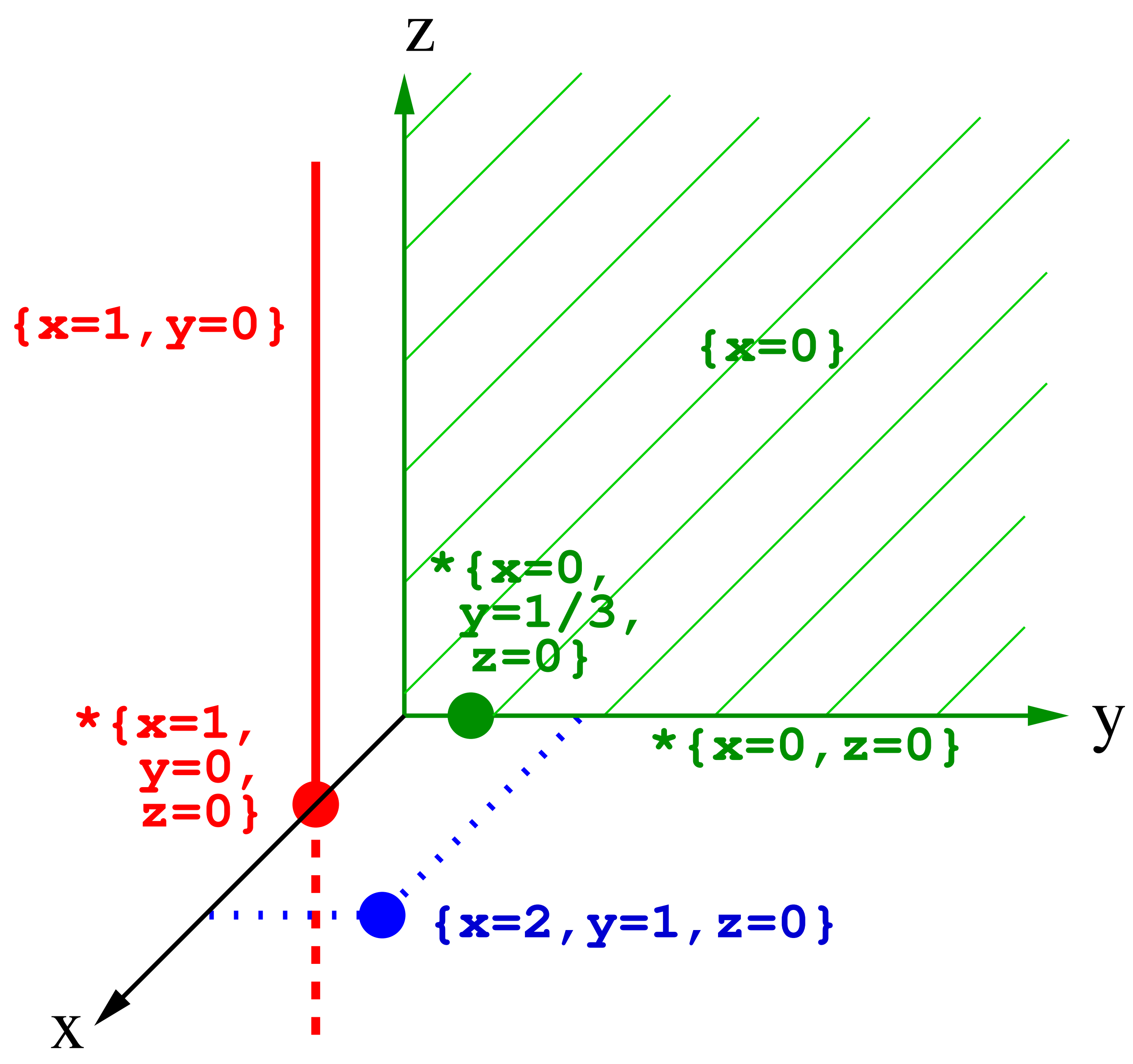
T6: [{f1,f2/x},{1-3y,z}] T7: [{f1,f2/x},{y,z}] T8: [{f1,f2/x},{y-1,z}]

phase 4

T11: [{},{x,1-3y,z}] T10: [{},{x-1,y,z}] T9: [{},{x-2,y-1,z}]

```
f1 = x^3 - 3x^2 + 2x
f2 = 2yx^2 - x^2 - 3yx + x
f3 = zx^2 - zx              [x > y > z]
```




z
{x=1,y=0}
{x=0}
*{x=0,
y=1/3,
z=0}
*{x=1,
y=0,
z=0}
y
*{x=0,z=0}
{x=2,y=1,z=0}
x