

# The Package CRACK for Solving Large Overdetermined Systems



Thomas Wolf  
 Department of Mathematics, Brock University  
 500 Glenridge Avenue, St. Catharines, Ontario, Canada L2S 3A1  
 Email: twolf@brocku.ca

## General Overview:

The program CRACK is a computer algebra package written in REDUCE for the solution of over-determined systems of algebraic, ordinary or partial differential equations with at most polynomial non-linearity. It is available as part of version 3.8 of the REDUCE system and the latest development version can be downloaded from <http://lie.math.brocku.ca/crack/>.

## Safety Enhancing Measures

### CRACK's safety enhancing measures include

- The ability to backup and re-load the whole session
- The automatic storing of the complete keyboard input during a session with the opportunity to feed this stored input into a new session
- The possibility to impose time restrictions of notoriously slow sub-steps, like factorizations and sometimes the computation and reduction of S-polynomials in Gröbner basis computations
- A method to interrupt an ongoing automatic computation and change it to interactive mode



## Trading Speed for Safety/Memory

### To avoid expression swell, CRACK allows the option to interactively or automatically perform

- Length-reducing versus general Gröbner basis steps
- Substitutions only in shorter equations (i.e. in a sub-system) versus substitutions in the complete system
- Growth bounded substitutions versus general substitutions
- Case-splittings (induced by factorizations, substitutions with potentially vanishing coefficients or ad hoc case distinctions) instead of Gröbner basis steps
- Length reduction steps, aiming at a more sparse system, which cost extra time but have many benefits (allowing shorter substitutions, overdetermined sub-systems,..)

## Detailed Interface with Visualization Aids

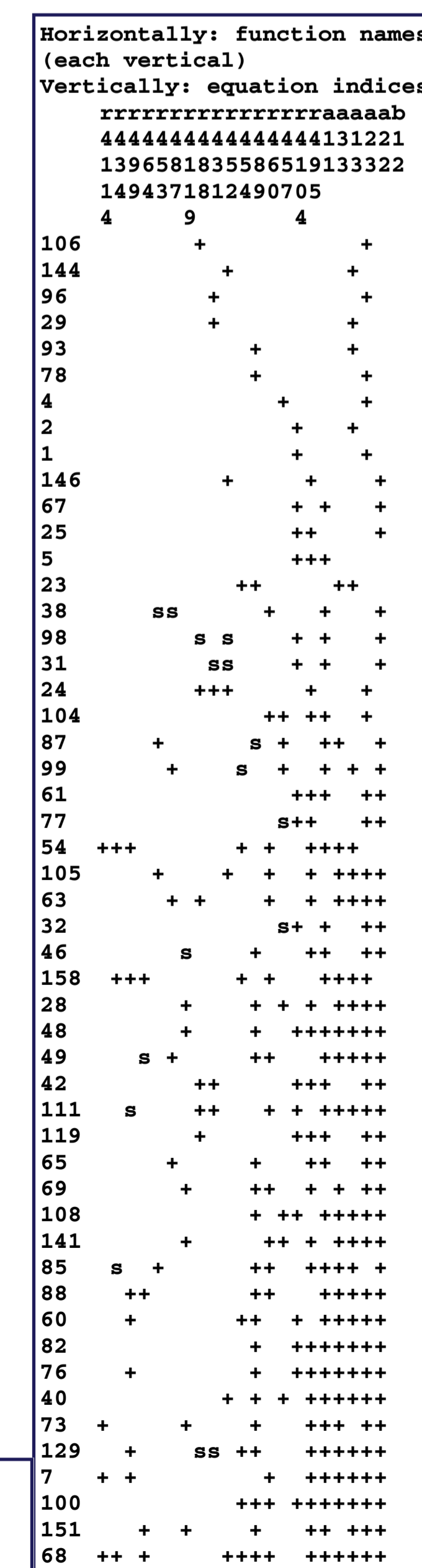
### For inspecting large systems, this interface includes

- for each equation its properties, history and investigations that have already been done with the equation
- a count of the total number of appearances of each unknown
- the occurrence of all derivatives of selected functions in any equation
- a statistical overview of the system
- the determination of not under-determined subsystems
- a listing of all sub- and sub-sub.. cases investigated so far, with their assumptions, number of solutions and number of steps
- graphical displays of size-related measures of the computation done so far
- a matrix display of occurrences of unknown constants and functions in all equations

```
Total number of occurrences of all unknowns in all
equations:
(116 . a22) (110 . a33) (86 . a11) (86 . a23) (81 . a13)
(76 . r460) (69 . r495) (67 . b12) (62 . r457) (22 . r487)
(11 . r452) (4 . r434)
```

```
Total number of equations in which unknowns occur:
(31 . a33) (29 . a22) (27 . r495) (26 . a11) (24 . b12)
(24 . a13) (23 . a23) (23 . r460) (17 . r457) (14 . r487)
```

```
Start : 243 steps
1      0=r448 : 25 steps
11     0=r460 : 37 steps
111    0=r452 : 2 steps
1111   0=r453 : 6 steps
11111  0=r494 : 2 steps
111111 0=r434, 0 soln : 2 steps
111112 0=a13, 1 soln : 3 steps
11112  0=a13 : 7 steps
111121 0=a23, 1 soln : 2 steps
111122 0=a11, 1 soln : 3 steps
11113  0=a23, 1 soln : 6 steps
11114  0=a22 - 3*a33, 1 soln : 5 steps
112    0=a11 - a22 : 5 steps
1121   0=a13 : 3 steps
11211  0=a23, 1 soln : 2 steps
11212  0=a33, 1 soln : 3 steps
1122   0=a23, 1 soln : 6 steps
1123   0=a33, 1 soln : 7 steps
12     0=a13, 1 soln : 4 steps
121    0=a23 : 31 steps
1211   0=a11 - a22, 1 soln : 3 steps
122    0=r452 : 6 steps
1221   0=a33 : 3 steps
12211  0=a11, 0 soln : 2 steps.
12212  0=a11*a22 + b12**2*kap, 1 soln : 1 step
1222   0=a11 - 2*a33, 1 soln : 2 steps
1223   0=a11**2 + b12**2*kap, 0 soln : 2 steps
1224   0=4*a11*a33 - 4*a33**2 + b12**2*kap, 1 soln : 2 steps
2      0=a23, 1 soln
```



The plot to the right shows which unknowns (top row) occur in which equation (left column)

## Algorithmic Extensions

### CRACK's algorithmic extensions include

- the ability to collect and apply syzygies which result as a by-product in the process of computing a differential Gröbner basis to integrate linear PDEs
- the treatment of inequalities: their usage, active collection and derivation, and their constant update in an ongoing reduction process based on newly derived equations
- the capability added by Winfried Neun (ZIB Berlin) to run in a truly parallel mode on a beowulf cluster, recently also ported to 64 bit AMD processors
- post-computation procedures, especially the possibility to merge solutions of parametric linear algebraic systems and to automate the production of web- pages for solutions that are found
- the ability to separate expressions with respect to independent variables which occur polynomially but with variable exponents, leading to automatically investigated case