Herbert Melenk (ed.)

The Complexity Barrier in REDUCE a Case Study

Abstract

The power of the symbolic math system REDUCE for solving large and difficult problems in science and engineering is demonstrated by a set of model problems. These include algebraic equation solving, formal variable elimination, formal power series, symbolic treatment of differential equations and applications from theoretical physics.

Contents

1 Introduction

The symbolic math system REDUCE [5] has been developed over the last 30 years. From the beginning, the development of REDUCE has been directed towards the solution of real problems in science and engineering. Originally most applications came from theoretical physics, but it soon became clear that application oriented symbolic computation can be exploited in many scientific and technical disciplines. Consequently the number of applications and specific packages has grown over the years. Nevertheless physics, as well as mechanical and electrical engineering continue to be the main domains of REDUCE.

Some of the currently available symbolic math systems are primarily directed towards use as an extended mathematical note pad, or even as a tool for mathematical education. In contrast to these, REDUCE focuses on the leading edge problem solving aspect and consequently concentrates radically on supplying modern and efficient computation methods and algorithms which are needed to solve large problems.

The intent of this collection of case studies is to demonstrate the specific power of REDUCE for the solution of complicated symbolic computational tasks in science and engineering. These cases mark the leading edge of symbolic methods. Unfortunately this collection cannot represent the power and elegance of the REDUCE algebraic language: most of the advanced problem solutions in the field require extensive algebraic programming where the powerful elementary operations of REDUCE such as polynomial arithmetic, algebraic equation solving etc. are used as black boxes – a well known review of algebraic systems [17] stressed this point by a heading "Power – if you can tame it". Fortunately this aspect is covered by recent books [12, 18, 6].

Last but not least it should be pointed out here that during the 80's and 90's the REDUCE development has grown to a global effort with contributions from many countries in several continents, including a significant European contribution.

Examples and descriptions of this collection have been contributed by

Marc Ganter (University of Washington, USA),
Guy Grebot (Queen Mary and Westfield College, London, England),
Jarmo Hietarinta (University of Turku, Finland),
David Hartley (GMD Bonn, Germany),
Friedrich W. Hehl (University of Cologne, Germany),
Rainer Schöpf (Konrad-Zuse-Zentrum, Berlin, Germany),
Franz Schunck (University of Cologne, Germany),
Duane Storti (University of Washington, USA),
Thomas Wolf (Queen Mary and Westfield College, London, England)

The editor wishes to express his thanks to all of them.

The following **REDUCE** packages are used in the examples:

- **Crack**: Coputation with differential equations and equation systems (A. Brand, T. Wolf)
- **Dfpart**: Computation with generic partial differentials (H. Melenk)
- **Excalc**: Exterior Calculus, differential forms (E. Schrüfer)

- **Gentran**: Program code generation for FORTRAN, C, PASCAL (B. Gates)
- **Groebner**: Algorithms for polynomial systems built on Gröbner bases (H. Melenk, M. Möller, W. Neun)
- **Invbase**: Involutive bases for polynomial systems (A. Zharkov)
- **Roots**: Roots for polynomials (S. Kameny)
- **Scope**: Source code optimization (H. v. Hulzen)
- **Taylor**: Algebra with truncated power series (R. Schöpf)

The documentation of these packages is found in the REDUCE delivery system or the REDUCE network library.

The computations were performed on a DEC Alpha AXP (OSF-1) machine or on equivalent systems (DEC Alpha AXP PC (Windows NT), HP-Apollo 9000/730 (HP-UX)). For comparison with other systems: these machines perform the REDUCE standard test in about 1.5 seconds[1].

[1] A personal computer with an AT architecture and Intel 486 DX2/66 chip needs about 2.6 seconds for the same sequence. When computing with big integers, e.g. for large polynomial computations such as Gröbner bases, the 64 bit DEC AXP processors are significantly faster than a comparable 32 bit machine

2 Algebraic equation systems

2.1 Constant quantum Yanq-Baxter equations in 2 dimensions

Problem:

The the constant quantum Yang-Baxter equation [7, 8] $R^{k_1k_2}_{j_1j_2}R^{l_1k_3}_{k_1j_3}R^{l_2l_3}_{k_2k_3} = R^{k_2k_3}_{j_2j_3}R^{k_1l_3}_{j_1k_3}R^{l_2l_2}_{k_1k_2}$ leads to a set of 64 quadratic equations in 16 variables

$$
\begin{gathered}
\{-abk + acf - b^2p + bhk + c^2p - cfl + df^2 + dhp - dk^2 - dlp, \\
ach - adf + bcf - bck - bdp + bhm + c^2u - cfn + dfh + dhu - dkm - dnp, \\
a^2b - a^2c - abh - abm + acg + acl + b^2f - b^2u + bhl - c^2k + c^2q - chl + dfg + dhq - dkm - dlu, \\
a^2d - ac^2 + acj + acn - adh - bcm + bdf - bdu + bhn - c^2m + c^2v - chn + dfj + dhv - dm^2 - dnu, \\
-abl + adk - b^2q + bcf - bck + bjk + cdp - cgl + dfg + djp - dkl - dlq, \\
-adg + adm + bcg + bch - bcl - bcm - bdq + bjm + cdu - cgn + dgh + dju - dlm - dnq, \\
-a^2d + ab^2 - abj - abn + adl + b^2g - b^2v + bcg + bjl - cdk + cdq - cjl + dg^2 + djq - dkn - dlv, \\
abd - acd - adj + adn + bcj - bcn + bdg - bdv + bjn - cdm + cdv - cjn + dgj + djv - dmn - dnv, \\
abp - akl - bfk + bku - blp + cfk + cmp + dfp + dpu - flm - k^2n - lnp, \\
bcp + bmu + chk - ckl + cmu - dfk + dhp - dlp + du^2 - fmn - kmn - n^2p, \\
abk - ack + b^2p - bhk + cgk - ckm + cmq + dgp + dqu - hlm - kmn - lnu, \\
adk + amn + bdp + bnu - c^2k + cjk - clm - cm^2 + cmv - dhk + djp - dlu + duv - hmn - m^2n - n^2u, \\
a^2l - abk + abq + akn - al^2 - bgk - bkm + bkv - blq + cfl + cnp + dfq + dpv - glm - kln - lnq, \\
acl + amn - bck + bcq - bm^2 + bmv + chl - cl^2 + cnu - dgk + dhq - dlq + duv - gmn - lmn - n^2q, \\
abl - adk + b^2q - bjk + cgl + cnq + dgq - dkm + dqv - jlm - kn^2 - lnv, \\
adl + an^2 + bdq + bnv - cdk + cjl - cln + cnv - djk + djq - dlv - dm^2 + dv^2 - jmn - mn^2 - n^2v, \\
-a^2f + a^2k + afl + afm - agk - ahk - bf^2 - bgp + ck^2 + cmp + f^2n - fhl + hkl + hnp - jk^2 - jlp, \\
-acf + ack + cfl - cgk + ckm + cmu - df^2 - dgp + hlm + hnu - jkm - jnp, \\
abk - acf - bfh + bfl - bgu - chk + ckl + cmq + fgn - h^2l + hl^2 + hnq - jkm - jlu, \\
adk + ajm - c^2f - cgm - chm + ckn + cmv - dfh + dfl - dgu + fjn - h^2n + hln + hnv - jm^2 - jnu, \\
-abf + abk - bfg + bfm - bgq - bhk + dk^2 + dmp + fgn - ghl - jlq + jnp, \\
-bcf + bck - dfg - dgq + dkm + dmu - jnq + jnu, \\
-adf - agn + b^2k - bfj + bgl + bgm - bgv - dhk + dkl + dmq + g^2n - hjl - jkn + jl^2 - jlv + jnq, \\
bdk + bjm - cdf - cgn - dfj + dgl - dgv - dhm + dkn + dmv + gjn - hjn + jln - jmn, \\
-a^2p + ak^2 - akq - aku + alp - bfp - bpq + fkm - flu + fnp + k^2m - k^2v + klu - lpv + m^2p + npu, \\
-acp - amu + ck^2 - ckq + clp - dfp - dpq - fnu + hkm + hnp + km^2 - kmv + lmu + m^2u - npv + nu^2, \\
-acp - amq - bhp + bk^2 + blp - bqu - cku + gkm + gnp - hlu + klm - kmv + l^2u - luv + m^2q + nqu, \\
-c^2p - cmq - cmu - dhp + dk^2 + dlp - dqu - hnu + jkm + jnp + kmn + lnu, \\
-abp + akl - bgp - bku - bq^2 + flm + fnq - glu + k^2n - lqv + mnp + npv, \\
-bcp - bmu + ckl - dgp - dq^2 - gnu + hlm + hnq + kmn + mnu - nqv + nuv, \\
-adp - anq - bjp + bkl + blq - bqv - dku + glm + gnq - jlu + kln - knv + l^2v - lv^2 + mnq + nqv, \\
-cdp - cnq - djp + dkl + dlq - dmu - dqv + jlm + jnq - jnu + kn^2 + lnv, \\
-acp + afh - bfk - bgp + cfk - cfq + chp - dkp - dpq + f^2j + ghk + hjp, \\
-a^2h + acf - acu - afj + ah^2 - bhk - bjp + cfg + cfm - cfv + chu - dku - dpv + fhj + ghm + hju, \\
abf - acf + bfg - bfm - bgu - c^2p + cfl - dmp - dqu + fgj + ghl + hjq, \\
-ach + adf - bhm - bju - c^2u + cfn + dfg - dmu - duv + fj^2 + ghn + hjv, \\
-bcp + bfh - bfl - bgq - cgq + dfk + dhp - dlp - dq^2 + fgj + gjk + j^2p, \\
-abh - agj + bcf - bcu + bh^2 - bhl - bjq + cg^2 - cgv + dfm + dhu - dlu - dqv + ghj + gjm + j^2u,
\end{gathered}
$$

$-adf-agj+b^2f-bfn+bg^2+bgh-bgv-cdp-cjq+dfl+dhq-dnp-dqv+g^2j+gjl+j^2q,$

$-adh-aj^2+bdf+bhj-bhn-bjv-cdu-cjv+dfn+dg^2+dhv-dnu-dv^2+gj^2+gjn+j^2v,$

$agp-amp-fgk+fhk+fjp-fkl+fkm-fmq+gku-glp+hmp+jpu-knp-npq,$

$-ahk-amu+cfk+cgp-fjk+fm^2-fmv+gmu+h^2k+hjp-hkl+hmu-jlp+ju^2-knu-npv,$

$bfk+bgp-cfk-cmp+gjp+jqu-mnp-nqu,$

$-chk-cmu+dfk+dgp+fmn+gnu-hlm+j^2p-jlu+juv-mnu-nuv,$

$afl+agq-bfk-bmp+fhl+fjq+fkn-fl^2-g^2k+gkv-glq-gmq+hnp+jpv-lnp-nq^2,$

$-bhk-bmu+cfl+cgq+fmn-gjk+h^2l+hjq-hl^2+hnu-jlq+juv-lnu-nqv,$

$bfl+bgq-dfk-dmp+ghl-gjk+gjq+hnq-jmq+jqv-n^2p-nqv,$

$dfl+dgq-dhk-dmu+fn^2+gnv+hjl-hln+hnv-j^2k+j^2q-jlv-jmv+jv^2-n^2u-nv^2,$

$a^2p-af^2+afq+afu-ahp+ckp+cpu-f^2g+f^2v-fgk-fhq-g^2p+hkq+hpv-jkp-jpq,$

$acp-afh+cfq+cmp+cu^2-f^2j-ghk-gjp+hmq+huv-jku-jpv,$

$abp+agu+bfq-cf^2-chp+clp+cqu-fgh-fgm+fgv-g^2u-h^2q+hlq+hqv-jmp-jqu,$

$adp+aju-cfh-chu+cnp+cuv+dfq-fhj+fjv-ghm-gju-h^2v+hnq+hv^2-jmu-juv,$

$abp+agq-bf^2+bfu-bhp+dkp+dpu-fg^2-fgl+fgv-g^2q-ghq+jkq-jlp+jpv-jq^2,$

$bcp-bfh+cgq+dmp+du^2-fgj-ghl-gjq-jlu+jmq-jqv+juv,$

$b^2p+bgq+bgu-df^2-dhp+dlp+dqu-fgj-fgn-hjq+jlq-jnp,$

$bdp+bju-dfh+dgq-dhu+dnp+duv-fj^2-ghn-hjv+jnq-jnu,$

$-afp+akp+apq-apu-fgp-fkq+fku+fpv-fqu-gpq+kmp-kpv+kqu+mpu-pqv+puv,$

$-ahp-au^2+ckp+cpq-fjp-fuv-hkq+hku+hpv-jpq-kuv+m^2p+mqu+mu^2-pv^2+u^2v,$

$bkp+bpq-cfp-cpu-fmq-ghp+gku+gpv-gqu-hqu+lmp+lqu-mpv+mqu,$

$-chp-cu^2+dkp+dpq-hjp-hmq-huv+jku+jpv-jqu+mnp+nqu,$

$alp+aq^2-bfp-bpu-flq+flu+fqv-g^2p-gq^2-gqu+knp+kqv-lpv+npu+pv^2-q^2v,$

$-bhp-bu^2+clp+cq^2-gjp-guv-hlq+hlu+hqv-jq^2-luv+mnp+mqv+nu^2-qv^2+uv^2,$

$blp+bq^2-dfp-dpu-fnq-gjp+glu-jqu+lnp+lqv-npv+nqu,$

$-dhp+dlp+dq^2-du^2-hnq-j^2p+jlu-juv+n^2p+nqv\}$

where the $R^{i_1 i_2}_{i_3 i_4}$ are encoded as elements of the matrix

$$\begin{pmatrix} a & b & c & d \\ f & g & h & j \\ k & l & m & n \\ p & q & u & v \end{pmatrix} .$$

The complete system cannot be solved automatically with fully general unspecified parameters.

Solution:

J. Hietarinta has broken the problem into smaller parts by symmetry analysis and by exploiting invariants "by hand". This approach leads to a set of specializations of the above matrix, e.g.

$$\begin{pmatrix} 1 & b & c & 0 \\ 1-u & m & h & j \\ 1-u & l & m & n \\ 0 & u & u & v \end{pmatrix} .$$

After substituting these special settings into the full system, a factorizing Gröbner basis can be computed from the resulting system which gives immediate access to the solutions.

When the automatic variable sequence optimization of the REDUCE Gröbner package is used, many of these computations are very fast.

The most complicated one starts from

$$\begin{pmatrix} a & b & 1 & 0 \\ k & g & h & 0 \\ k & l & 1 & 0 \\ p & q & u & v \end{pmatrix}$$

and leads to the factored Gröbner basis

$$\begin{array}{l}
\{\{k^2 + p, -k^2 + u, -k + v - 1, g - k - 1, q, a - k, h + k, l, b\}, \\
\{-2k\,l - 2l^2 + p, k\,l + l^2 + u, -k - l + v - 1, g - 1, k\,l + l^2 + q, a - k - l + 1, h - l, b - 1\}, \\
\{b\,k - k\,l - k + p, -b^2 - b\,k + k\,l + 2k + l^2 + 2l + u + 1, -k - l + v - 1, -b + g, \\
b^2 - k - l^2 - 2l + q - 1, a + b - k - l, b + h - l - 1, -b^2 + b\,l + k + l + 1\}, \\
\{k^2 + k\,l + p, k\,l + k + l^2 + 2l + u, -k - l + v - 1, -b + g, -k^2 - 2k\,l - k - l^2 - 2l + q, \\
a + b - k - l, b + h - l - 1, b\,k + b\,l + l\}, \\
\{p, h\,l + u, -l + v - 1, g + h - l - 1, -h\,l + q, a - l - 1, b + 1, k\}\}
\end{array}$$

This computation needs 51 seconds. The total evaluation of all 40 subproblems takes 187 seconds.

2.2 Elimination of variables

Problem:

In a context of modeling physical objects using implicit functions, an object is represented as a tri–variate polynomial function $F(x, y, z)$. Further, the surface of the object is taken as the iso-value surface at $F(x, y, z) = 0$. Find symbolic solution for the locus of points which satisfy the condition $F(x, y, z) = 0$ and $\partial F/\partial y = 0$. Here the functions F are (sparse) polynomials with rational coefficients.

A simple example of one of this type:

$F = ((-\frac{1}{4} + x)^2 + y^2 + z^2)^N + ((16 * x^2 + 16 * y^2 + (\frac{1}{4}) * z^2))^{-N}$

This function F represents an object that somewhat resembles an apple which has been cored where the core is off center.

Compute $\partial F/\partial y$, then eliminate the y variable from F and $\partial F/\partial y$. For $N = 1$, we obtain the following factored result:

$(28 + 32x - 63 * z^2) * (92 + 32x - 63 * z^2) * (64 - 960 * x^2 - 512 * x^3 + 1024 * x^4 - 15 * z^2 - 8 * x * z^2 + 1040 * x^2 * z^2 + 16 * z^4)$

This represents the solution locus of $F = 0$ and $\partial F/\partial y = 0$.

The problem is easily solved with most computer algebra systems for $N = 1$ and $N = 2$ using Gröbner Basis and up to $N = 4$ using resultants. One key point is one must factor (if possible) the functions before variable elimination.

Solution:

The Gröbner package of REDUCE implements a special term order for groupwise variable elimination, where two groups of variables compete. A Gröbner basis computed under this odering has the elimination property [2] with respect to the first variable group, but the complexity of the computation is significantly lower than a full lexicographical Gröbner basis, which is normally used for variable elimination. In the above case the first group has only one element $\{y\}$ and the second group is $\{x, z\}$, such that the “lower” part of a Gröbner basis under this term ordering is free of y and describes exactly the intersection of the polynomial ideal with $\mathbf{Z}(x, z)$.

```
    n := 4;  %  degree parameter
    load groebner;
    F:= ((-1/4 + x)^2 + y^2 + z^2)^N +
        ((16*x^2 + 16*y^2 + (1/4)*z^2))^(-N);
    vars := {y,x,z};
    torder gradlexgradlex,1; % stepped order with step after 1st variable

    write ‘‘===============================
’’,n,‘‘ ===========’’;
    fy := df(f,y);
    vars := {y,x,z};
        % factorization of input
    f0 := factorize num f; f1 := factorize num fy;
    write k,`` factors: '',length f0,``
'',length f1;
        % compute the elimination polynomials
    s := for each f00 in f0 collect
      for each f11 in f1 collect
         first reverse groebner({f00,f11},vars);

    write s;
```

Computing time of this sequence for varying N

N	factors of F	factors of FF	time
1	1	4	0.3s
2	1	2	0.6s
3	2	4	1.8s
4	1	2	20.5s
5	2	4	13.2s
6	2	3	52.4s
7	2	4	1 min 16s
8	1	2	21 min 40 s
9	3	4	4 min 13 s
10	2	3	30 min 36 s
11	2	4	17 min 41 s
12	2	3	45 min 49 s
13	2	4	54 min 03 s
14	2	3	6 h 41 min
15	4	4	1 h 11 min

The case $N = 16$ did not terminate within 12 hours.

Remarks:

1. The less factors F has, the higher is the degree of the Gröbner problem and the more time is needed.

2. REDUCE is also able to compute the elimination by resultants; however, the approach using Gröbner basis with the special elimination term order is significantly faster.

2.3 Steady state of Lotka-Volterra equation

Problem:

The following algebraic equation system has been derived [14] as steady state solution of a Lotka-Volterra equation describing a neural network. The system has n variables x_i and one free parameter c.

$$\left\{ 0 = f_j = 1 - x_j * (c + \sum_{i \neq j} x_i^2),\ j = 1 \cdots n \right\}$$

Solution:

The system can easily be solved in full generality for $n \leq 3$. For $n = 4$ the system is too complicated for an automatic solution. We therefore decompose the polynomial ideal over $Z[c][x_1, x_2, x_4, x_4]$ first by a factoring **Gröbner** basis computation with total degree term order. The resulting partial ideals are simple enough for an automatic processing by the REDUCE **solve** operator.

Input:

```
f1 := 1-c*x1+x1*x2^2+x1*x3^2 + x1*x4^2;
f2 := 1-c*x2+x2*x1^2+x2*x3^2 + x2*x4^2;
f3 := 1-c*x3+x3*x1^2+x3*x2^2 + x3*x4^2;
f4 := 1-c*x4+x4*x1^2+x4*x2^2 + x4*x3^2;

  % decompose ideal under total degree term order
torder revgradlex;
basis1 := groebnerf({f1,f2,f3,f4}, {x1,x2,x3,x4});

  % compute the solutions for each branch
result := for each bas in basis1 join solve(bas,{x1,x2,x3,x4});
```

The result is[2]:

$\{x_4 = \text{root\_of}\left(-3x^3 + xc - 1, x\right), x_1 = x_4, x_3 = x_4, x_2 = x_4\}$,

$\{x_4 = \text{root\_of}\left(6x^4 - 5x^2c + x + c^2, x\right), x_2 = \left(6x_4^3 - 3x_4c + 1\right)/c, x_1 = x_4, x_3 = x_4\}$,

$\{x_4 = \text{root\_of}\left(3x^6 - 4x^4c + x^2c^2 + 2xc + 1, x\right)$,
$x_2 = \left(-3x_4^5c + 3x_4^4 + 4x_4^3c^2 - 4x_4^2c - x_4c^3 + x_4 - c^2\right)/2$,
$x_1 = x2,\ x_3 = x_4\}$,

$\{x_4 = \text{root\_of}\left(3x^6 - 4x^4c + x^2c^2 + 2xc + 1, x\right)$,
$x_3 = \left(-3x_4^5c + 3x_4^4 + 4x_4^3c^2 - 4x_4^2c - x_4c^3 + x_4 - c^2\right)/2$,
$x_1 = x_3,\ x_2 = x_4\}$,

[2] compressed by hand for reduction of multiple solutions arising from the different computations branches

$$\begin{aligned}&\{x_4 = \mathrm{root\_of}\left(3x^6 - 4x^4c + x^2c^2 + 2xc + 1, x\right),\\&x_3 = \left(-3x_4^5c + 3x_4^4 + 4x_4^3c^2 - 4x_4^2c - x_4c^3 + x_4 - c^2\right)/2,\\&x_1 = x_4,\ x_2 = x_3\},\end{aligned}$$

$$\begin{aligned}&\{x_4 = \mathrm{root\_of}\left(-2x^3 + xc + 1, x\right),\\&x_2 = \left(-x_4 + \sqrt{-7x_4^2 + 4c}\right)/2,\ x_1 = \left(-x_4 - \sqrt{-7x_4^2 + 4c}\right)/2,\\&x_3 = x_4\},\end{aligned}$$

$$\begin{aligned}&\{x_4 = \mathrm{root\_of}\left(-2x^3 + xc + 1, x\right),\\&x_2 = \left(-x_4 - \sqrt{-7x_4^2 + 4c}\right)/2,\ x_1 = \left(-x_4 + \sqrt{-7x_4^2 + 4c}\right)/2,\\&x_3 = x_4\},\end{aligned}$$

$$\begin{aligned}&\{x_4 = \mathrm{root\_of}\left(-3x^4c + 3x^3 + x^2c^2 + 4xc + 4, x\right),\\&x_3 = \left(3x_4^3c^2 - 9x_4^2c - x_4c^3 + 6x_4 - 2c^2\right)/12,\\&x_1 = x_3,\ x_2 = x_3\},\end{aligned}$$

$$\begin{aligned}&\{x_4 = \mathrm{root\_of}\left(2x^6 - 3x^4c - 5x^3 + x^2c^2 + 4xc + 4, x\right),\\&x_3 = \left(-2x_4^5c + 4x_4^4 + 3x_4^3c^2 - x_4^2c - x_4c^3 - 6x_4 - 2c^2\right)/4,\\&x_1 = -x_3,\ x_2 = x_3\},\end{aligned}$$

$$\begin{aligned}&\{x_4 = \mathrm{root\_of}\left(2x^6 - 3x^4c - 5x^3 + x^2c^2 + 4xc + 4, x\right),\\&x_3 = \left(-x_4 + \sqrt{2x_4^5 - 3x_4^3c - 4x_4^2 + x_4c^2 + 4c}\right)/2,\\&x_1 = \left(-2x_4^5c + 4x_4^4 + 3x_4^3c^2 - x_4^2c - x_4c^3 - 6x_4 - 2c^2\right)/4,\\&x_2 = \left(-x_4 - \sqrt{2x_4^5 - 3x_4^3c - 4x_4^2 + x_4c^2 + 4c}\right)/2\},\end{aligned}$$

$$\begin{aligned}&\{x_4 = \mathrm{root\_of}\left(2x^6 - 3x^4c - 5x^3 + x^2c^2 + 4xc + 4, x\right),\\&x_3 = \left(-x_4 - \sqrt{2x_4^5 - 3x_4^3c - 4x_4^2 + x_4c^2 + 4c}\right)/2,\\&x_1 = \left(-2x_4^5c + 4x_4^4 + 3x_4^3c^2 - x_4^2c - x_4c^3 - 6x_4 - 2c^2\right)/4,\\&x_2 = \left(-x_4 + \sqrt{2x_4^5 - 3x_4^3c - 4x_4^2 + x_4c^2 + 4c}\right)/2\}\end{aligned}$$

Remark: Some of the polynomials inside the $root\_of$ expressions above have zeros which can be expressed explicitly by surds[3]. However these are too big for practical use.

The total computing time is 9 hours 25 minutes.

[3] REDUCE returns explicit forms for the roots only if these are "small" or if the user asks for them expliclty.

3 Series expansions

3.1 Verification of a finite difference formula

Problem:

Given the partial differential expression

$$DD = \partial_x\Big(a(x,y)\partial_x g(x,y)\partial_y g(x,y)\partial_y f(x,y)\Big) + \partial_y\Big(a(x,y)\partial_x g(x,y)\partial_y g(x,y)\partial_x f(x,y)\Big)$$

and a formula $FD = FD(a, f, g, x, y, dx, dy)$ which is an arithmetic combination of x, y, dx, dy and of function expressions of a, f and g with arguments from x, y, $x + dx$ and $y + dy$. FD is claimed to be a second order finite difference approximation for the differential equation $DD = 0$. Check automatically whether this is true.

Solution:

Using the packages **DFPART** and **TAYLOR** we introduce the generic function symbols $a(x, y)$, $f(x, y)$ and $g(x, y)$ and declare their partial derivatives commutative. Then we compute in terms of the symbolic partial derivatives of a, f and g

1. the fully expanded formula for DD
2. the second order Taylor approximation of FD in dx and dy at the point $(dx, dy) = (0, 0)$.

The claim is true if both forms are identical up to an error term of magnitude $O(dx^2)+O(dy^2)$.

And indeed both computations lead to the same result

$2\,a(x,y)\,f_{xy}(x,y)\,g_x(x,y)\,g_y(x,y)\;+\;a(x,y)\,f_x(x,y)\,g_{xy}(x,y)\,g_y(x,y)\,+$

$a(x,y)\,f_x(x,y)\,g_x(x,y)\,g_{yy}(x,y)\;+\;a(x,y)\,f_y(x,y)\,g_{xx}(x,y)\,g_y(x,y)\,+$

$a(x,y)\,f_y(x,y)\,g_{xy}(x,y)\,g_x(x,y)\;+\;a_x(x,y)\,f_y(x,y)\,g_x(x,y)\,g_y(x,y)\,+$

$a_y(x,y)\,f_x(x,y)\,g_x(x,y)\,g_y(x,y)$

such that their difference vanishes.

REDUCE program:

```
   % declare generic functions with commuting derivatives
generic_function a(x,y), f(x,y), g(x,y);
dfp_commute a(x,y), g(x,y), f(x,y);

   % the differential formula
dd := dfp(a(x,y)*dfp(g(x,y),x)*dfp(g(x,y),y)*dfp(f(x,y),y),x) +
      dfp(a(x,y)*dfp(g(x,y),x)*dfp(g(x,y),y)*dfp(f(x,y),x),y) ;


   % the finite difference formula
```

```
fd := a(x+dx,y+dy)*f(x+dx,y+dy)*g(x+dx,y+dy)^2/(32*dx^2*dy^2)
     +a(x+dx,y)*f(x+dx,y+dy)*g(x+dx,y+dy)^2/(32*dx^2*dy^2)
     +a(x,y+dy)*f(x+dx,y+dy)*g(x+dx,y+dy)^2/(32*dx^2*dy^2)
     +a(x,y)*f(x+dx,y+dy)*g(x+dx,y+dy)^2/(32*dx^2*dy^2)
     -f(x,y)*a(x+dx,y+dy)*g(x+dx,y+dy)^2/(32*dx^2*dy^2)
     -   ...
```

and so on for more than 150 lines. Instead of printing the full original form we give an equivalent C program in the next paragraph. The REDUCE evaluation continues with

```
   % computing the taylor approximation
texp := taylor (fd, dx, 0, 1, dy, 0, 1);
factor dx,dy;
result := taylortostandard texp; % convert series to polynomial
   % compare expressions
dd - result;
```

with final result 0. The total evaluation takes 8 seconds. The computing time is almost totally spent in the two dimensional Taylor approximation. The computation of an approximation in the next higher degree in dx and dy takes 34 seconds; that result has 379 terms.

3.2 Program generation

Problem:

Generate a reasonable program for the numeric evaluation of the finite difference form fd in the previous example.

Solution:

The formula fd had been given in a fully expanded form. This form is inadequate for a direct numeric evaluation for example in a program in the language **C**, because function evaluations with the same arguments are repeated many times. The packages **GENTRAN** and **SCOPE** allow us to convert such an expanded expression (–sequence) into an optimized program source in some target language. For fd we receive the following **C** program: [4].

```
fd(x,y)
{
  g2=x+dx; g3=y+dy; g1=a(g2,g3); g4=f(g2,g3); g7=g(g2,g3);
  g184=dy*dx; g79=g184*g184; g8=32.0*g79; g5=(g7*g7)/g8;
  g9=a(g2,y); g10=a(x,g3); g11=a(x,y); g12=f(x,y); g13=g(x,y);
  g77=g13*g13; g15=16.0*g79; g16=g(g2,y); g36=g16*g16; g17=g4/g8;
  g18=g(x,g3); g71=g18*g18; g19=g4/g15; g20=g1/g8; g23=y-dy;
  g22=a(g2,g23); g24=f(g2,g23); g27=g(g2,g23); g25=(g27*g27)/g8;
  g28=a(x,g23); g30=g24/g8; g31=g(x,g23); g74=g31*g31; g32=g24/g15;
  g33=g22/g8; g35=g36/g15; g42=x-dx; g41=a(g42,g3); g43=f(g42,g3);
  g46=g(g42,g3); g44=(g46*g46)/g8; g47=a(g42,y); g49=g(g42,y);
  g65=g49*g49; g50=g43/g8; g51=g43/g15; g52=g41/g8; g54=a(g42,g23);
  g55=f(g42,g23); g58=g(g42,g23); g56=(g58*g58)/g8; g60=g55/g8;
  g61=g55/g15; g62=g54/g8; g64=g65/g15; g141=(g49/g15)*g12;
  g106=g31*g28; g138=(g58/g15)*g13; g112=g18*g10; g137=(g46/g15)*g13;
  g136=(g16/g15)*g12; g133=(g27/g15)*g13; g131=(g7/g15)*g13;
  g157=g71/g15-g44-g5; g158=g74/g15-g56-g25; g159=g20*g12-(g17*g11);
  g160=g33*g12-(g30*g11); g161=(g9/g8)*g12+(g47/g8)*g12;
  g162=g52*g12-(g50*g11); g163=g62*g12-(g60*g11); g177=g28+g47;
  g164=-g177-g54; g175=g47+g10; g165=-g175-g41; g167=g9+g11;
  g168=-g22-g28; g170=-g1-g10; g171=-(g9*g4)-(g10*g4)-(g11*g4);
  g172=-g11-g47; g179=g36/g8+g65/g8; g180=-(g28*g24)-(g24*g11);

  t0=(g49*g47+g49*g11)*(g51*g18+g61*g31)+(g16*g9+g16*g11)*(g19*g18+g32
    *g31)+(g167-g170)*(g131*g12+g77*g17)+(g167-g168)*(g133*g12+g77*g30)
    +(-g18-g31)*(g136*g9+g136*g11)+g165*(g137*g43+g65*g50+g71*g50)+(g11
    -g165)*(g137*g12+g77*g50)+g164*(g138*g55+g65*g60+g74*g60)+(g11-g164
    )*(g138*g12+g77*g60)+(g170-g9)*(g36*g17+g71*g17)+(g168-g9)*(g36*g30
    +g74*g30)+g5*(-g171-(g12*g1))+g25*(g24*g9-g180)+g77*g12*(-g20-g33-(
```

[4] GENTRAN writes one assignment per line; the text has been compacted by hand. Declarations have been omitted.

```
    g9/g15)-g52-g62-(g47/g15)-(g10/g15)-(g28/g15))+g141*g31*(g172-g28)+
    g141*g18*(g172-g10)+g106*(g32*g16+g61*g49)+g49*g31*(g61*g54-((g54/
    g15)*g12))+g55*g11*(g56-g138)+g112*(g19*g16+g51*g49)+g49*g18*(g51*
    g41-((g41/g15)*g12));
  fd=t0+g43*g11*(g44-g137)+g31*g16*(g32*g22-((g22/g15)*g12))+g133*g24*
    (-g22-g9)+g18*g16*(g19*g1-((g1/g15)*g12))+g171*g131+g4*g1*(g5-g131)
    +g180*g133+g136*(-g112-g106)+g12*g11*(g158+g157-(g77/(8.0*g79))+g64
    +g35)+g28*g12*(g179+g158)+g12*g10*(g179+g157)+g47*g12*(g64-g44-g56)
    +g56*g54*(g55-g12)+g177*g56*g55+g44*g41*(g43-g12)+g175*g44*g43+g12*
    g9*(g35-g5-g25)+g25*g22*(g24-g12)+g36*(g160+g159)+g71*(g162+g161+
    g159)+g74*(g163+g161+g160)+g65*(g163+g162);
  return(fd);
}
```

3.3 Expansion of elliptic functions

Problem:

Given the differentation rules and values at $x = 0$ for the elliptic functions $\mathrm{sn}(x, k)$, $\mathrm{cn}(x, k)$, $\mathrm{dn}(x, k)$, expand sn and cn with respect to k about 0 to sixth order and show that (in this order) [1]

$$\mathrm{sn}(x, k)^2 + \mathrm{cn}(x, k)^2 = 1$$

Defintions [11]:

- Jacobi's elliptic functions $\mathrm{sn}(x, k)$, $\mathrm{cn}(x, k)$, $\mathrm{dn}(x, k)$. The modulus and complementary modulus are denoted by K and K!', usually written as k and k' respectively.
- epsilon(x, k) is the incomplete elliptic integral of the second kind usually written as $\mathrm{E}(x|k)$.
- KK(k) is the complete elliptic integral of the first kind usually written as K(k) (note: K(k)=arcsn(1,k)).
- KK!'(k) is the complementary complete integral usually written mathematically as K'(k) (note: K'(k) = K(k')).
- EE(k) is the complete elliptic integral of the second kind usually written as E(k).
- EE!'(k) is the complementary complete integral usually written as E'(k) (note: E'(k)=E(k')).

Solution:

Definition of the calculus as a rule set (rewriting rules):

```
operator sn, cn, dn, epsilon;
elliptic_rules := {
  % Differentiation rules for basic functions
   df(sn(~x,~k),~x)     => cn(x,k)*dn(x,k),
   df(cn(~x,~k),~x)     => -sn(x,k)*dn(x,k),
   df(dn(~x,~k),~x)     => -k^2*sn(x,k)*cn(x,k),
   df(epsilon(~x,~k),~x)=> dn(x,k)^2,

  % k-derivatives
   df(sn(~x,~k),~k)  => (k*sn(x,k)*cn(x,k)^2
     -epsilon(x,k)*cn(x,k)*dn(x,k)/k)/(1-k^2) + x*cn(x,k)*dn(x,k)/k,
   df(cn(~x,~k),~k)  => (-k*sn(x,k)^2*cn(x,k)
     +epsilon(x,k)*sn(x,k)*dn(x,k)/k)/(1-k^2) - x*sn(x,k)*dn(x,k)/k,
   df(dn(~x,~k),~k)  => k*(-sn(x,k)^2*dn(x,k)
     +epsilon(x,k)*sn(x,k)*cn(x,k))/(1-k^2) - k*x*sn(x,k)*cn(x,k),
   df(epsilon(~x,~k),~k) => k*(sn(x,k)*cn(x,k)*dn(x,k)
     -epsilon(x,k)*cn(x,k)^2)/(1-k^2) -k*x*sn(x,k)^2,
  % parity properties
   sn(-~x,~k) => -sn(x,k),         cn(-~x,~k) => cn(x,k),
   dn(-~x,~k) => dn(x,k),          epsilon(-~x,~k) => -epsilon(x,k),
   sn(~x,-~k) => sn(x,k),          cn(~x,-~k) => cn(x,k),
   dn(~x,-~k) => dn(x,k),          epsilon(~x,-~k) => epsilon(x,k),

  % values at x=0
   sn(0,~k) => 0,                  cn(0,~k) => 1,
   dn(0,~k) => 1,                  epsilon(0,~k) => 0,

  % special cases of modulus
   sn(~x,0) => sin(x),             cn(~x,0) => cos(x),
   dn(~x,0) => 1,                  epsilon(~x,0) => x,

   sn(~x,1) => tanh(x),            cn(~x,1) => 1/cosh(x),
   dn(~x,1) => 1/cosh(x),          epsilon(~x,1) => tanh(x)
}$

let elliptic_rules;
```

Compute the formal Taylor approximations in k of order 6:

```
tsn := taylor(sn(x,k),k,0,6);
tcn := taylor(cn(x,k),k,0,6);
```

$$\begin{aligned}
&tsn := sin(x) + cos(x)\,(cos(x)^2\,x + cos(x)\,sin(x) + sin(x)^2\,x - 2\,x)/4\,k^2 + \\
&\quad (cos(x)^5\,x - 2\,cos(x)^4\,sin(x)\,x^2 + 5\,cos(x)^4\,sin(x) - 10\,cos(x)^3\,sin(x)^2\,x \\
&\quad + 6\,cos(x)^3\,x - 4\,cos(x)^2\,sin(x)^3\,x^2 + cos(x)^2\,sin(x)^3 + \\
&\quad 8\,cos(x)^2\,sin(x)\,x^2 + 4\,cos(x)^2\,sin(x) - 11\,cos(x)\,sin(x)^4\,x + \\
&\quad 30\,cos(x)\,sin(x)^2\,x - 16\,cos(x)\,x - 2\,sin(x)^5\,x^2 + 8\,sin(x)^3\,x^2 - \\
&\quad 8\,sin(x)\,x^2)/64\,k^4 + \\
&\quad (- 6\,cos(x)^7\,x^3 + 17\,cos(x)^7\,x - 99\,cos(x)^6\,sin(x)\,x^2 + 21\,cos(x)^6\,sin(x) - \\
&\quad 18\,cos(x)^5\,sin(x)^2\,x^3 - 71\,cos(x)^5\,sin(x)^2\,x + 36\,cos(x)^5\,x^3 - 18\,cos(x)^5\,x \\
&\quad - 135\,cos(x)^4\,sin(x)^3\,x^2 - 133\,cos(x)^4\,sin(x)^3 + 324\,cos(x)^4\,sin(x)\,x^2 + \\
&\quad 172\,cos(x)^4\,sin(x) - 18\,cos(x)^3\,sin(x)^4\,x^3 - 13\,cos(x)^3\,sin(x)^4\,x + \\
&\quad 72\,cos(x)^3\,sin(x)^2\,x^3 - 156\,cos(x)^3\,sin(x)^2\,x - 72\,cos(x)^3\,x^3 + \\
&\quad 160\,cos(x)^3\,x + 27\,cos(x)^2\,sin(x)^5\,x^2 - 118\,cos(x)^2\,sin(x)^5 + \\
&\quad 176\,cos(x)^2\,sin(x)^3 - 108\,cos(x)^2\,sin(x)\,x^2 + 32\,cos(x)^2\,sin(x) - \\
&\quad 6\,cos(x)\,sin(x)^6\,x^3 + 75\,cos(x)\,sin(x)^6\,x + 36\,cos(x)\,sin(x)^4\,x^3 - \\
&\quad 498\,cos(x)\,sin(x)^4\,x - 72\,cos(x)\,sin(x)^2\,x^3 + 888\,cos(x)\,sin(x)^2\,x + \\
&\quad 48\,cos(x)\,x^3 - 384\,cos(x)\,x + 63\,sin(x)^7\,x^2 - 324\,sin(x)^5\,x^2 + \\
&\quad 540\,sin(x)^3\,x^2 - 288\,sin(x)\,x^2)/2304\,k^6 + O(k^7)
\end{aligned}$$

$$\begin{aligned}
&tcn := cos(x) + sin(x)\,(- cos(x)^2\,x - cos(x)\,sin(x) - sin(x)^2\,x + 2\,x)/4\,k^2 + \\
&\quad (- 2\,cos(x)^5\,x^2 - 5\,cos(x)^4\,sin(x)\,x - 4\,cos(x)^3\,sin(x)^2\,x^2 -
\end{aligned}$$

$$\begin{array}{l}
7\,\mathit{cos}(x)^3\,\mathit{sin}(x)^2\,+\,8\,\mathit{cos}(x)^3\,x^2\,+\,2\,\mathit{cos}(x)^2\,\mathit{sin}(x)^3\,x\,+\,2\,\mathit{cos}(x)^2\,\mathit{sin}(x)\,x\\
-\,2\,\mathit{cos}(x)\,\mathit{sin}(x)^4\,x^2\,-\,3\,\mathit{cos}(x)\,\mathit{sin}(x)^4\,+\,8\,\mathit{cos}(x)\,\mathit{sin}(x)^2\,x^2\,-\\
4\,\mathit{cos}(x)\,\mathit{sin}(x)^2\,-\,8\,\mathit{cos}(x)\,x^2\,+\,7\,\mathit{sin}(x)^5\,x\,-\,22\,\mathit{sin}(x)^3\,x\,+\,16\,\mathit{sin}(x)\,x)/\\
64\,k^4+\\
(\,-\,9\,\mathit{cos}(x)^7\,x^2\,+\,6\,\mathit{cos}(x)^6\,\mathit{sin}(x)\,x^3\,-\,71\,\mathit{cos}(x)^6\,\mathit{sin}(x)\,x\,+\\
135\,\mathit{cos}(x)^5\,\mathit{sin}(x)^2\,x^2\,-\,66\,\mathit{cos}(x)^5\,\mathit{sin}(x)^2\,-\,36\,\mathit{cos}(x)^5\,x^2+\\
18\,\mathit{cos}(x)^4\,\mathit{sin}(x)^3\,x^3\,-\,\mathit{cos}(x)^4\,\mathit{sin}(x)^3\,x\,-\,36\,\mathit{cos}(x)^4\,\mathit{sin}(x)\,x^3+\\
18\,\mathit{cos}(x)^4\,\mathit{sin}(x)\,x\,+\,297\,\mathit{cos}(x)^3\,\mathit{sin}(x)^4\,x^2\,+\,61\,\mathit{cos}(x)^3\,\mathit{sin}(x)^4-\\
720\,\mathit{cos}(x)^3\,\mathit{sin}(x)^2\,x^2\,-\,208\,\mathit{cos}(x)^3\,\mathit{sin}(x)^2\,+\,252\,\mathit{cos}(x)^3\,x^2+\\
18\,\mathit{cos}(x)^2\,\mathit{sin}(x)^5\,x^3\,+\,31\,\mathit{cos}(x)^2\,\mathit{sin}(x)^5\,x\,-\,72\,\mathit{cos}(x)^2\,\mathit{sin}(x)^3\,x^3-\\
24\,\mathit{cos}(x)^2\,\mathit{sin}(x)^3\,x\,+\,72\,\mathit{cos}(x)^2\,\mathit{sin}(x)\,x^3\,+\,56\,\mathit{cos}(x)^2\,\mathit{sin}(x)\,x+\\
153\,\mathit{cos}(x)\,\mathit{sin}(x)^6\,x^2\,+\,91\,\mathit{cos}(x)\,\mathit{sin}(x)^6\,-\,684\,\mathit{cos}(x)\,\mathit{sin}(x)^4\,x^2-\\
212\,\mathit{cos}(x)\,\mathit{sin}(x)^4\,+\,900\,\mathit{cos}(x)\,\mathit{sin}(x)^2\,x^2\,-\,32\,\mathit{cos}(x)\,\mathit{sin}(x)^2-\\
288\,\mathit{cos}(x)\,x^2\,+\,6\,\mathit{sin}(x)^7\,x^3\,-\,39\,\mathit{sin}(x)^7\,x\,-\,36\,\mathit{sin}(x)^5\,x^3\,+\,318\,\mathit{sin}(x)^5\,x\\
+\,72\,\mathit{sin}(x)^3\,x^3\,-\,672\,\mathit{sin}(x)^3\,x\,-\,48\,\mathit{sin}(x)\,x^3\,+\,384\,\mathit{sin}(x)\,x)/2304\,k^6+\\
O(k^7)
\end{array}$$

Final computation of $tsn^2 + tcn^2$:

```
taylorcombine(tsn^2 + tcn^2);
```

$$\mathit{cos}(x)^2\,+\,\mathit{sin}(x)^2\,+\,O(k^7)$$

The total computing time is 5.3 seconds.

4 Differential equations

4.1 Solving of killing equations

Problem:

Solve the killing equations for the static spherically symmetric metrics with constant spacial curvature. These metrics include,for example,the well known Schwarzschild metric describing a black hole or a spherical symmetric star.

$$ds^2 = c(r)\,dt^2 - a(r)\,dr^2 - k^2\,(d\theta^2 + \sin(\theta)\,d\phi^2). \tag{1}$$

The equations have been generated by the computer algebra system `cLaSSI/ShEEp` for general relativity which can be coupled directly with REDUCE. The equations are solved by the REDUCE package `CRACK`.

Solution:

```
  % parameters for crack

lisp(genint_:=nil)$
lisp(print_:=200);

  % Data input.

depend c,r$
depend a,r$
depend v0,tt,r,h,p$
depend v1,tt,r,h,p$
depend v2,tt,r,h,p$
depend v3,tt,r,h,p$

LISTOFFUNS := LIST(v0,v1,v2,v3)$

depend c,r$
depend a,r$
depend v0,tt,r,h,p$
depend v1,tt,r,h,p$
depend v2,tt,r,h,p$
depend v3,tt,r,h,p$

listoffuns := list(v0,v1,v2,v3)$

kileqs :=
list(2*df(v0,tt)+c^(-1)*v1*df(c,r,-a^(1/2)*c^(-1/2)*df(v1,tt)+a^(-1/2)*c^(
1/2)*df(v0,r),c^(1/2)*k^(-1)*df(v0,h)-c^(-1/2)*k*df(v2,tt),c^(1/2)*k^(-1)*(sin(h))^(
-1)*df(v0,p)-c^(-1/2)*k*sin(h)*df(v3,tt),-2*df(v1,r)-a^(-1)*v1*df(a,r),-a^(1
/2)*k^(-1)
*df(v1,h)-a^(-1/2)*k*df(v2,r),-a^(1/2)*k^(-1)*(sin(h))^(-1)*df(v1,p)-a^(-1/2
)*k*sin(h
```

```
)*df(v3,r),-2*df(v2,h),-sin(h)*df(v3,h)-(sin(h))^(-1)*df(v2,p),-2*v2*cos(h)*
(sin(h))^(-1
)-2*df(v3,p),1/2*a^(1/2)*c^(-1)*df(v1,tt,2+1/2*a^(-1/2)*df(v0,tt,r)+1/2*a^(-
1/2)*c^(
-1)*v1*df(c,r,2)+1/2*a^(-1/2)*c^(-1)*df(c,r)*df(v0,tt)-1/4*a^(-1/2)*c^(-2)*v
1*(df(c,r)
)^2-1/4*a^(-3/2)*c^(-1)*v1*df(a,r)*df(c,r,1/2*c^(-1/2)*df(v1,tt,r)-1/4*c^(-3
/2)*df(c
,r)*df(v1,tt)+1/2*a^(-1)*c^(1/2)*df(v0,r,2+1/4*a^(-1)*c^(-1/2)*df(a,r)*df(v1
,tt)+3/4*
a^(-1)*c^(-1/2)*df(c,r)*df(v0,r)-1/4*a^(-2)*c^(1/2)*df(a,r)*df(v0,r),1/2*a^(
1/2)*c^(
-1/2)*k^(-1)*df(v1,tt,h)+1/2*a^(-1/2)*c^(1/2)*k^(-1)*df(v0,r,h)+1/2*a^(-1/2)
*c^(-1/2
)*k^(-1)*df(c,r)*df(v0,h),1/2*a^(1/2)*c^(-1/2)*k^(-1)*(sin(h))^(-1)*df(v1,tt
,p)+1/2*a
^(-1/2)*c^(1/2)*k^(-1)*(sin(h))^(-1)*df(v0,r,p)+1/2*a^(-1/2)*c^(-1/2)*k^(-1)
*(sin(h)
)^(-1)*df(c,r)*df(v0,p),1/2*k^(-1)*df(v0,tt,h)+1/2*c^(-1)*k*df(v2,tt,2)+1/4*
c^(-1)*k^
(-1)*df(c,r)*df(v1,h)-1/4*a^(-1)*c^(-1)*k*df(c,r)*df(v2,r),1/2*a^(-1/2)*c^(1
/2)*k^(-1)
*df(v0,r,h)+1/2*a^(-1/2)*c^(-1/2)*k*df(v2,tt,r)+1/4*a^(-1/2)*c^(-1/2)*k^(-1)
*df(c,r)*df
(v0,h)-1/4*a^(-1/2)*c^(-3/2)*k*df(c,r)*df(v2,tt),1/2*c^(1/2)*k^(-2)*df(v0,h,
2)+1/2*c^(
-1/2)*df(v2,tt,h),-1/2*c^(1/2)*k^(-2)*cos(h)*(sin(h))^(-2)*df(v0,p)+1/2*c^(1
/2)*k^(-2
)*(sin(h))^(-1)*df(v0,h,p)-1/2*c^(-1/2)*cos(h)*df(v3,tt)+1/2*c^(-1/2)*(sin(h
))^(-1)*df
(v2,tt,p),1/2*k^(-1)*(sin(h))^(-1)*df(v0,tt,p)+1/2*c^(-1)*k*sin(h)*df(v3,tt,
2)+1/4*c^(
-1)*k^(-1)*(sin(h))^(-1)*df(c,r)*df(v1,p)-1/4*a^(-1)*c^(-1)*k*sin(h)*df(c,r)
*df(v3,r),
1/2*a^(-1/2)*c^(1/2)*k^(-1)*(sin(h))^(-1*df(v0,r,p)+1/2*a^(-1/2)*c^(-1/2)*k*
sin(h)*
df(v3,tt,r)+1/4*a^(-1/2)*c^(-1/2)*k^(-1)*(sin(h))^(-1)*df(c,r)*df(v0,p)-1/4*
a^(-1/2)*
c^(-3/2)*k*sin(h)*df(c,r)*df(v3,tt),-1/2*c^(1/2)*k^(-2)*cos(h)*(sin(h))^(-2)
*df(v0,p)+
1/2*c^(1/2)*k^(-2)*(sin(h))^(-1)*df(v0,h,p)+1/2*c^(-1/2)*cos(h)*df(v3,tt)+1/
2*c^(-1/2
)*sin(h)*df(v3,tt,h),1/2*c^(1/2)*k^(-2)*cos(h)*(sin(h))^(-1)*df(v0,h)+1/2*c^
(1/2)*k^(
-2)*(sin(h))^(-2)*df(v0,p,2)+1/2*c^(-1/2)*cos(h)*(sin(h))^(-1)*df(v2,tt)+1/2
*c^(-1/2)*df
(v3,tt,p),-1/2*a^(1/2)*c^(-1/2)*k^(-1)*df(v1,tt,h)+1/2*a^(-1/2)*c^(-1/2)*k*d
f(v2,tt,r
)-1/4*a^(-1/2)*c^(-1/2)*k^(-1)*df(c,r)*df(v0,h)-1/4*a^(-1/2)*c^(-3/2)*k*df(c
,r)*df(v2
,tt),-1/2*k^(-1)*df(v1,r,h)+1/2*a^(-1)*k*df(v2,r,2)-1/4*a^(-1)*k^(-1)*df(a,r
)*df(v1,h)
```

```
-1/4*a^(-2)*k*df(a,r)*df(v2,r),-1/2*a^(1/2*k^(-2)*df(v1,h,2)+1/2*a^(-1/2)*df
(v2,r,h),
1/2*a^(1/2)*k^(-2)*cos(h)*(sin(h))^(-2)*df(v1,p)-1/2*a^(1/2)*k^(-2)*(sin(h))
^(-1)*df
(v1,h,p)-1/2*a^(-1/2)*cos(h)*df(v3,r)+1/2*a^(-1/2)*(sin(h))^(-1)*df(v2,r,p),
-1/2*a^(1/2
)*c^(-1/2)*k^(-1)*(sin(h))^(-1)*df(v1,tt,p)+1/2*a^(-1/2)*c^(-1/2)*k*sin(h)*d
f(v3,tt,r
)-1/4*a^(-1/2)*c^(-1/2)*k^(-1)*(sin(h))^(-1)*df(c,r)*df(v0,p)-1/4*a^(-1/2)*c
^(-3/2)*
k*sin(h)*df(c,r)*df(v3,tt),-1/2*k^(-1)*(sin(h))^(-1)*df(v1,r,p)+1/2*a^(-1)*k
*sin(h)*df(
v3,r,2)-1/4*a^(-1)*k^(-1)*(sin(h))^(-1)*df(a,r)*df(v1,p)-1/4*a^(-2)*k*sin(h)
*df(a,r)*df
(v3,r),1/2*a^(1/2)*k^(-2)*cos(h)*(sin(h))^(-2)*df(v1,p)-1/2*a^(1/2)*k^(-2)*(
sin(h))^
(-1)*df(v1,h,p)+1/2*a^(-1/2)*cos(h)*df(v3,r+1/2*a^(-1/2)*sin(h)*df(v3,r,h),-
1/2*a^(1/2
)*k^(-2)*cos(h)*(sin(h))^(-1)*df(v1,h)-1/2*a^(1/2)*k^(-2)*(sin(h))^(-2)*df(v
1,p,2)+1/
2*a^(-1/2)*cos(h)*(sin(h))^(-1)*df(v2,r)+1/2*a^(-1/2)*df(v3,r,p),c^(-1/2)*co
s(h)*df(v3
,tt)+1/2*c^(-1/2)*sin(h)*df(v3,tt,h)-1/2*c^(-1/2)*(sin(h))^(-1)*df(v2,tt,p),
a^(-1/2)*cos
(h)*df(v3,r)+1/2*a^(-1/2)*sin(h)*df(v3,r,h)-1/2*a^(-1/2)*(sin(h))^(-1)*df(v2
,r,p),3/2*k
^(-1)*cos(h)*df(v3,h)+1/2*k^(-1)*cos(h)*(sin(h))^(-2)*df(v2,p)+1/2*k^(-1)*si
n(h)*df(v3,
h,2)-1/2*k^(-1)*(sin(h))^(-1)*df(v2,h,p),-k^(-1)*v2+k^(-1)*cos(h)*(sin(h))^(
-1)*df(v3
,p)+1/2*k^(-1)*df(v3,h,p)-1/2*k^(-1)*(sin(h))^(-2)*df(v2,p,2),-1/2*a^(-1)*c^
(-1)*v1*df
(c,r,3)-a^(-1)*c^(-1)*df(c,r,2)*df(v0,tt)-a^(-1)*c^(-1)*df(c,r,2)*df(v1,r)+1
/2*a^(-1)
*c^(-2)*v1*df(c,r)*df(c,r,2)+1/2*a^(-1)*c^(-2)*(df(c,r))^2*df(v0,tt)+1/2*a^(
-1)*c^(-
2)*(df(c,r))^2*df(v1,r)-1/4*a^(-1)*c^(-3)*v1*(df(c,r))^3+1/4*a^(-2)*c^(-1)*v
1*df(a,r
)*df(c,r,2)+1/4*a^(-2)*c^(-1)*v1*df(a,r,2)*df(c,r)+1/2*a^(-2)*c^(-1)*df(a,r)*df(c,r)*df
(v0,tt)+1/2*a^(-2)*c^(-1)*df(a,r)*df(c,r)*df(v1,r)-1/4*a^(-3)*c^(-1)*v1*(df(
a,r))^2*df
(c,r),-1/2*a^(-1/2)*c^(-1)*k^(-1)*df(c,r,2)*df(v1,h)+1/4*a^(-1/2)*c^(-2)*k^(
-1)*(df(
c,r))^2*df(v1,h)+1/4*a^(-3/2)*c^(-1)*k^(-1)*df(a,r)*df(c,r)*df(v1,h),-1/2*a^
(-1/2)*c^
(-1)*k^(-1)*(sin(h))^(-1)*df(c,r,2)*df(v1,p)+1/4*a^(-1/2)*c^(-2)*k^(-1)*(sin
(h))^(-1
)*(df(c,r))^2*df(v1,p)+1/4*a^(-3/2)*c^(-1*k^(-1)*(sin(h))^(-1)*df(a,r)*df(c,
r)*df(v1
,p),1/2*a^(-1)*c^(-1/2)*k^(-1)*df(c,r,2)*df(v0,h)-1/4*a^(-1)*c^(-3/2)*k^(-1)
*(df(c,r)
)^2*df(v0,h)-1/4*a^(-2)*c^(-1/2)*k^(-1)*df(a,r)*df(c,r)*df(v0,h),1/2*a^(-1)*
```

```
c^(-1/2)*
k^(-1)*(sin(h))^(-1)*df(c,r,2)*df(v0,p)-1/4*a^(-1)*c^(-3/2)*k^(-1)*(sin(h))^
(-1)*(df
(c,r))^2*df(v0,p)-1/4*a^(-2)*c^(-1/2)*k^(-1)*(sin(h))^(-1)*df(a,r)*df(c,r)*d
f(v0,p),c
^(-1/2)*k^(-1)*sin(h)*df(v3,tt),-c^(-1/2)*k^(-1)*df(v2,tt),a^(-1/2)*k^(-1)*s
in(h)*df
(v3,r),-a^(-1/2)*k^(-1)*df(v2,r),-2*k^(-2*v2*cos(h)*(sin(h))^(-1)-2*k^(-2)*d
f(v2,h)-
2*k^(-2)*df(v3,p),-1/2*a^(-1)*c^(-3/2)*df(c,r,3)*df(v1,tt)+a^(-1)*c^(-5/2)*d
f(c,r)*df
(c,r,2)*df(v1,tt)-1/2*a^(-1)*c^(-7/2)*(df(c,r))^3*df(v1,tt)+3/4*a^(-2)*c^(-3
/2)*df(a,
r)*df(c,r,2)*df(v1,tt)+1/4*a^(-2)*c^(-3/2)*df(a,r,2)*df(c,r)*df(v1,tt)-1/2*a
^(-2)*c^(-
5/2)*df(a,r)*(df(c,r))^2*df(v1,tt)-1/2*a^(-3)*c^(-3/2)*(df(a,r))^2*df(c,r)*d
f(v1,tt),-
1/2*a^(-3/2)*c^(-1)*v1*df(c,r,4)-a^(-3/2)*c^(-1)*df(c,r,3)*df(v0,tt)-3/2*a^(
-3/2)*c^
(-1)*df(c,r,3)*df(v1,r)+a^(-3/2)*c^(-2)*v1*df(c,r)*df(c,r,3)+a^(-3/2)*c^(-2)
*v1*(df(c,
r,2))^2+2*a^(-3/2)*c^(-2)*df(c,r)*df(c,r,2)*df(v0,tt)+3*a^(-3/2)*c^(-2)*df(c
,r)*df(c,
r,2)*df(v1,r)-5/2*a^(-3/2)*c^(-3)*v1*(df(c,r))^2*df(c,r,2)-a^(-3/2)*c^(-3)*(
df(c,r))^
3*df(v0,tt)-3/2*a^(-3/2)*c^(-3)*(df(c,r))^3*df(v1,r)+a^(-3/2)*c^(-4)*v1*(df(
c,r))^4+
3/4*a^(-5/2)*c^(-1)*v1*df(a,r)*df(c,r,3)+a^(-5/2)*c^(-1)*v1*df(a,r,2)*df(c,r
,2)+1/4*a^
(-5/2)*c^(-1)*v1*df(a,r,3)*df(c,r)+3/2*a^(-5/2)*c^(-1)*df(a,r)*df(c,r,2)*df(
v0,tt)+9/4*
a^(-5/2)*c^(-1)*df(a,r)*df(c,r,2)*df(v1,r)+1/2*a^(-5/2)*c^(-1)*df(a,r,2)*df(
c,r)*df(v0
,tt)+3/4*a^(-5/2)*c^(-1)*df(a,r,2)*df(c,r)*df(v1,r)-a^(-5/2)*c^(-2)*v1*df(a,
r)*df(c,r)
*df(c,r,2)-1/2*a^(-5/2)*c^(-2)*v1*df(a,r,2*(df(c,r))^2-a^(-5/2)*c^(-2)*df(a,
r)*(df(c
,r))^2*df(v0,tt)-3/2*a^(-5/2)*c^(-2)*df(a,r)*(df(c,r))^2*df(v1,r)+1/2*a^(-5/
2)*c^(-3
)*v1*df(a,r)*(df(c,r))^3-5/4*a^(-7/2)*c^(-1)*v1*(df(a,r))^2*df(c,r,2)-5/4*a^
(-7/2)*c^
(-1)*v1*df(a,r)*df(a,r,2)*df(c,r)-a^(-7/2)*c
^(-1)*(df(a,r))^2*df(c,r)*df(v0,tt)-3/2*a^
(-7/2)*c^(-1)*(df(a,r))^2*df(c,r)*df(v1,r)+
1/2*a^(-7/2)*c^(-2)*v1*(df(a,r))^2*(df(c,r
))^2+a^(-9/2)*c^(-1)*v1*(df(a,r))^3*df(c,r),-1/2*a^(-1)*c^(-1)*k^(-1)*df(c,r
,3)*df(
v1,h)+a^(-1)*c^(-2)*k^(-1)*df(c,r)*df(c,r,2)*df(v1,h)-1/2*a^(-1)*c^(-3)*k^(-
1)*(df(c
,r))^3*df(v1,h)+3/4*a^(-2)*c^(-1)*k^(-1)*df(a,r)*df(c,r,2)*df(v1,h)+1/4*a^(-
2)*c^(-1
)*k^(-1)*df(a,r,2)*df(c,r)*df(v1,h)-1/2*a^(-2)*c^(-2)*k^(-1)*df(a,r)*(df(c,r
))^2*df(v1
,h)-1/2*a^(-3)*c^(-1)*k^(-1)*(df(a,r))^2*df(c,r)*df(v1,h),-1/2*a^(-1)*c^(-1)
*k^(-1)
*(sin(h))^(-1)*df(c,r,3)*df(v1,p)+a^(-1)*c^(-2)*k^(-1)*(sin(h))^(-1)*df(c,r)
*df(c,r,2)
*df(v1,p)-1/2*a^(-1)*c^(-3)*k^(-1)*(sin(h))^(-1)*(df(c,r))^3*df(v1,p)+3/4*a^
(-2)*c^
(-1)*k^(-1)*(sin(h))^(-1)*df(a,r)*df(c,r,2*df(v1,p)+1/4*a^(-2)*c^(-1)*k^(-1)
*(sin(h)
)^(-1)*df(a,r,2)*df(c,r)*df(v1,p)-1/2*a^(-2)*c^(-2)*k^(-1)*(sin(h))^(-1)*df(
a,r)*(df(
c,r))^2*df(v1,p)-1/2*a^(-3)*c^(-1)*k^(-1*(sin(h))^(-1)*(df(a,r))^2*df(c,r)*d
f(v1,p)
,-1/2*a^(-1)*c^(-1)*k^(-1)*df(c,r,3)*df(v1,h)+a^(-1)*c^(-2)*k^(-1)*df(c,r)*d
f(c,r,2)
*df(v1,h)-1/2*a^(-1)*c^(-3)*k^(-1)*(df(c,r))^3*df(v1,h)+3/4*a^(-2)*c^(-1)*k^
(-1)*df
(a,r)*df(c,r,2)*df(v1,h)+1/4*a^(-2)*c^(-1)*k^(-1)*df(a,r,2)*df(c,r)*df(v1,h)
-1/2*a^(-2
)*c^(-2)*k^(-1)*df(a,r)*(df(c,r))^2*df(v1,h)-1/2*a^(-3)*c^(-1)*k^(-1)*(df(a,
r))^2*df
(c,r)*df(v1,h),-1/2*a^(-1)*c^(-1)*k^(-1)*(sin(h))^(-1)*df(c,r,3)*df(v1,p)+a^
(-1)*c^(
-2)*k^(-1)*(sin(h))^(-1)*df(c,r)*df(c,r,2)*df(v1,p)-1/2*a^(-1)*c^(-3)*k^(-1)
*(sin(h))
^(-1)*(df(c,r))^3*df(v1,p)+3/4*a^(-2)*c^(-1)*k^(-1)*(sin(h))^(-1)*df(a,r)*df
(c,r,2)*
df(v1,p)+1/4*a^(-2)*c^(-1)*k^(-1)*(sin(h)^(-1)*df(a,r,2)*df(c,r)*df(v1,p)-1/
2*a^(-2)
*c^(-2)*k^(-1)*(sin(h))^(-1)*df(a,r)*(df(c,r))^2*df(v1,p)-1/2*a^(-3)*c^(-1)*k^(-1)*
(sin(h))^(-1)*(df(a,r))^2*df(c,r)*df(v1,p),1/2*a^(-3/2)*c^(-1/2)*k^(-1)*df(c
,r,3)*df(
v0,h)-a^(-3/2)*c^(-3/2)*k^(-1)*df(c,r)*df(c,r,2)*df(v0,h)+1/2*a^(-3/2)*c^(-5
/2)*k^(-
1)*(df(c,r))^3*df(v0,h)-3/4*a^(-5/2)*c^(-1/2)*k^(-1)*df(a,r)*df(c,r,2)*df(v0
,h)-1/4*a^
(-5/2)*c^(-1/2)*k^(-1)*df(a,r,2)*df(c,r)*df(v0,h)+1/2*a^(-5/2)*c^(-3/2)*k^(-
1)*df(a,r
)*(df(c,r))^2*df(v0,h)+1/2*a^(-7/2)*c^(-1/2)*k^(-1)*(df(a,r))^2*df(c,r)*df(v
0,h),1/2*
a^(-3/2)*c^(-1/2)*k^(-1)*(sin(h))^(-1)*df(c,r,3)*df(v0,p)-a^(-3/2)*c^(-3/2)*
k^(-1)*
(sin(h))^(-1)*df(c,r)*df(c,r,2)*df(v0,p)+1/2
*a^(-3/2)*c^(-5/2)*k^(-1)*(sin(h))^(-1)*(
df(c,r))^3*df(v0,p)-3/4*a^(-5/2)*c^(-1/2)*k^(-1)*(sin(h))^(-1)*df(a,r)*df(c,
r,2)*df(v0
,p)-1/4*a^(-5/2)*c^(-1/2)*k^(-1)*(sin(h))^(-1)*df(a,r,2)*df(c,r)*df(v0,p)+1/
2*a^(-5/2)
*c^(-3/2)*k^(-1)*(sin(h))^(-1)*df(a,r)*(df(c,r))^2*df(v0,p)+1/2*a^(-7/2)*c^(
-1/2)*k^
(-1)*(sin(h))^(-1)*(df(a,r))^2*df(c,r)*df(v0 ,p))$
```

```
res:=first crack(kileqs,{},listoffuns,{})$

% further simplification step

let int(df(a,r)*df(c,r)/(sqrt(a)*a*c),r)=-2*df(c,r)/(sqrt(a)*c)
 +2*int((c*df(c,r,2)-df(c,r)^2)/(c^2*sqrt(a)),r);

res;
```

The final result:

- Explicit Solutions:
  $v_3 = \frac{\cos h \cos pc_{126} + \cos h \sin pc_{125}}{\sin h}$,

  $v_2 = -\cos pc_{125} + \sin pc_{126}$,

  $v_1 = 0$,

  $v_0 = -c_{134}$
- Function symbols: $c_{126}, c_{125}, c_{134}$
- Special cases. Any of the following expressions represents a special case for the parameters/parametric functions a, c, k, h to be investigated individually:

$\frac{d^2a}{dr^2}\frac{d\,c}{d\,r}ac^2 - 2\left(\frac{d\,a}{d\,r}\right)^2\frac{d\,c}{d\,r}c^2 + 3\frac{d\,a}{d\,r}\frac{d^2c}{dr^2}ac^2 - 2\frac{d\,a}{d\,r}\left(\frac{d\,c}{d\,r}\right)^2 ac - 2\frac{d^3c}{dr^3}a^2c^2 + 4\frac{d^2c}{dr^2}\frac{d\,c}{d\,r}a^2c - 2\left(\frac{d\,c}{d\,r}\right)^3 a^2 = 0$,
$k = 0$, $\cos^2\, h - 1 = 0$, $\frac{d\,c}{d\,r} = 0$, $\sqrt{a} = 0$, $\sqrt{c} = 0$

The total computing time is 24 seconds.

4.2 Symmetry analysis of a PDE system

Problem:

Analyze the symmetries of the Karpman equations [10] which play a role in plasma physics. In their real form they are [3]

$$\rho,_t + w_1\rho,_z + \tfrac{1}{2}\left[s_1(2\rho,_x\phi,_x + 2\rho,_y\phi,_y + \rho\phi,_{xx} + \rho\phi,_{yy}) + s_2(2\rho,_z\phi,_z + \rho\phi,_{zz})\right] = 0$$

$$\phi,_t + w_1\phi,_z - \tfrac{1}{2}\left[s_1\left(\frac{\rho,_{xx}}{\rho} + \frac{\rho,_{yy}}{\rho} - \phi,_x^2 - \phi,_y^2\right) + s_2\left(\frac{\rho,_{zz}}{\rho} - \phi,_z^2\right)\right] + a_1\nu = 0$$

$$\nu,_{tt} - (w_2)^2(\nu,_{xx} + \nu,_{yy} + \nu,_{zz}) - 2a_2\rho(\rho,_{xx} + \rho,_{yy} + \rho,_{zz}) - 2a_2(\rho,_x^2 + \rho,_y^2 + \rho,_z^2) = 0$$

with the three functions ρ, ϕ, ν of four variables t, x, y, z and with constant parameters a_i, s_i, w_i.

Solution:

Using the package `CRACK`. At first, preliminary symmetry conditions are derived and solved successively for each of the three equations. The solution of the preliminary conditions for the first equation requires 31 integrations and states that all ξ^i are functions of t, x, y, z only. The preliminary conditions of the second equation do not provide new constraints. The result from the preliminary conditions for the third equation is that ν is a function of t, x, y, z only. This needs 4 integrations. The full conditions for the first equation are derived which requires a further 77 integrations to solve. The remaining full conditions used to solve the second equation requires 6 integrations. Finally, the full conditions for the last equation are solved with three more integrations.

The general solution for the symmetry generators then reads

$$\begin{array}{rcl@{\qquad}rcl}
\xi^x &=& -yc_1 + c_2 & \eta^r &=& 0 \\
\xi^y &=& xc_1 + c_3 & \eta^\phi &=& a_1c_6t^2 + a_1c_7t + c_8 \\
\xi^z &=& c_4 & \eta^v &=& -c_6t - c_7 \\
\xi^t &=& c_5 & & &
\end{array}$$

with constants $c_1, \ldots, c_8$. The corresponding 8 symmetry generators are

$$\begin{array}{ll}
X_1 = \partial_x & X_5 = y\partial_x - x\partial_y \\
X_2 = \partial_y & X_6 = \partial_\phi \\
X_3 = \partial_z & X_7 = a_1t\partial_\phi - \partial_\nu \\
X_4 = \partial_t & X_8 = a_1t^2\partial_\phi - 2t\partial_\nu.
\end{array}$$

The corresponding input is

```
depend r,x,y,z,tt;
depend f,x,y,z,tt;
depend v,x,y,z,tt;
```

```
de := {{df(r,tt)+w1*df(r,z)
        + s1*(df(r,x)*df(f,x)+df(r,y)*df(f,y)+r*df(f,x,2)/2+r*df(f,y,2)/2)
        + s2*(df(r,z)*df(f,z)+r*df(f,z,2)/2),

        df(f,tt)+w1*df(f,z)
        - (s1*(df(r,x,2)/r+df(r,y,2)/r-df(f,x)**2-df(f,y)**2) +
           s2*(df(r,z,2)/r-df(f,z)**2))/2 + a1*v,

        df(v,tt,2)-w2**2*(df(v,x,2)+df(v,y,2)+df(v,z,2))
        - 2*a2*r*(df(r,x,2)+df(r,y,2)+df(r,z,2))
        - 2*a2*(df(r,x)**2+df(r,y)**2+df(r,z)**2)},

       {r,f,v},  {x,y,z,tt}};
mo := {0, nil, nil};
LIEPDE(de,mo);
```

with the final result

```
The symmetries are:$

xi_x=(c192*y + c193)/(s1*s2)$

xi_y=( - c192*x + c194)/(s1*s2)$

xi_z=c190/(s1*s2)$

xi_tt= - c202$

eta_r=0$

eta_f=(c190*w1 - c204*s2*tt**2 - c205*s2*tt - c206*s2)/(s1*s2**2)$

eta_v=(2*c204*tt + c205)/(a1*s1*s2)$

with c206  c205  c204  c194  c193  c192  c190  c202 free.
```

The total computing time for this example is 9 minutes 12 seconds.

5 Theoretical physics applications

5.1 Non-commutative Algebra

Problem:

Define a calculus fo a non–commutative algebra based on a system of axioms [6]: Consider, in Cartesian coordinates, the position and the momentum operators x_i and p_i of a quantum mechanical system with $x^2 = x_1^2 + x_2^2 + x_3^2$ and $p^2 = p_1^2 + p_2^2 + p_3^2$; define the commutators $[x_i, p_j] = i\bar{h}\delta_{ij}$. Calculate $[x^2, p_j]$ and $[x_i, p^2]$.

Solution:

```
operator x,p;        % subscribed quantities x,p
operator comm;       % comm(a,b) represents [a,b]
operator delta;      % delta function
noncom x,p,comm;     % declaration of non-commutativity
antisymmetric comm; % declare comm(b,a) = -comm(a,b)

let {delta(~a,~b) => if a=b then 1 else 0};

let{ comm(x(~a),x(~b)) => 0,
     comm(p(~a),p(~b)) => 0,
     comm(x(~a),p(~b)) => hbar*i*delta(a,b),
     comm(~a+~b,~c)    => comm(a,c) + comm(b,c),
     comm(~a^2,~b)     => a*comm(a,b) + comm(a,b)*a
   };

x2 := for k:=1:3 sum x(k)^2;
p2 := for k:=1:3 sum p(k)^2;

for k:=1:3 collect comm(x2,p(k));
for k:=1:3 collect comm(p2,x(k));
```

The result of the last two statements is after a few milliseconds

```
{2*hbar*i*x(1),
 2*hbar*i*x(2),
 2*hbar*i*x(3)}
{ - 2*hbar*i*p(1),
  - 2*hbar*i*p(2),
  - 2*hbar*i*p(3)}
```

5.2 Solution of Einstein's field equation

Problem:

Check whether a specific metric of Robinson–Trautman type is a solution of Einstein's field equations of general relativity theory [16, 9].

Solution:

The following program has been written by J. Dermott McCrea (Dublin) [5] In the first part of the program a general metric with two covariant indices (gdd = g–down–down) is introduced. In terms of gdd, the contravariant metric (guu = g–up–up), the Christoffel symbols (chrisddd, chrisddu), and the Ricci tensor (riccidd) are determined. Moreover, the 22–component of the left hand side of Einstein's field equations *with* cosmological constant is calculated. In the second part the explicit form of the covariant components of the metric is given and information provided on some functions and their derivatives. As a result of the program, the tensor component lamric22 should turn out to be zero. The REDUCE package `EXCALC` is used.

```
off exp$
array gdd(3,3);
operator x;
x(0):=u$ x(1):=r$ x(2):=theta$ x(3):=phi$
gdd(0,0):=g00$ gdd(0,1):=g01$
gdd(1,1):=g11$ gdd(2,2):=g22$ gdd(3,3):=g33$
depend g00,u,r;
depend g01,u,r;
depend g11,u,r;
depend g22,u,r;
depend g33,u,r,theta;

array ag(2,2), bg(3,3),guu(3,3)$
for i:=0:3 do for j:=i+1:3 do
gdd(j,i):=gdd(i,j)$
for k:=0:3 do for l:=k:3 do begin
integer ii$
for i:=0:2 do begin
if i>=k then ii:=i+1 else ii:=i$
for j:=0:2 do
if j>=l then ag(i,j):=gdd(ii,j+1)
else ag(i,j):=gdd(ii,j) end$
bg(k,l):=bg(l,k):=ag(0,0)*(ag(1,1)*ag(2,2)-ag(1,2)*ag(2,1))
         -ag(0,1)*(ag(1,0)*ag(2,2)-ag(1,2)*ag(2,0))
          +ag(0,2)*(ag(1,0)*ag(2,1)-ag(1,1)*ag(2,0))
end$
gdet:=gdd(0,0)*bg(0,0)-gdd(0,1)*bg(0,1)+gdd(0,2)*bg(0,2)
      -gdd(0,3)*bg(0,3)$

subdet:=gdet$
depend pdt,u,r,theta;
gdet:=pdt;
```

[5] Dermott McCrae died in 1993. The present program has been transmitted by F. W. Hehl.

```
for k:=0:3 do for l:=k:3 do
guu(k,l):=guu(l,k):=(-1)**(k+l)*bg(l,k)/gdet$
clear ag,bg;
``GUU
computed'';

array
chrisddd(3,3,3),chrisddu(3,3,3)$
for i:=0:3 do for j:=i:3
do
 <<for k:=0:3
do
    chrisddd(j,i,k) :=
chrisddd(i,j,k):=

(df(gdd(j,k),x(i))+df(gdd(k,i),x(j))-df(gdd(i,j),x(k)))/2$
for k:=0:3 do chrisddu(j,i,k):= chrisddu(i,j,k)
:=
               for p := 0:3 sum guu(k,p)*chrisddd(i,j,p) >>$

%ricdd(i,j):=ricdd(j,i):=for n:=0:3 sum
%(df(chrisddu(i,j,n),x(n))-df(chrisddu(n,j,n),x(i))+for k:=0:3 sum
%(chrisddu(n,k,n)*chrisddu(i,j,k)-chrisddu(i,k,n)*chrisddu(n,j,k)))$
ricdd22:=for n:=0:3 sum
(df(chrisddu(2,2,n),x(n))-df(chrisddu(n,2,n),x(2))+for k:=0:3 sum
(chrisddu(n,k,n)*chrisddu(2,2,k)-chrisddu(2,k,n)*chrisddu(n,2,k)))$
lam0:=1/2; m0:=1;   %HERE
lamric22:=ricdd22-lam0*gdd(2,2)$
lamric22:=num(lamric22)$
off exp;
depend f1,u;
depend f2,u,r;
depend f3,u,r;
depend psi,u,r;
depend uu,u,r;
depend vv,u,r;
depend ww,u,r;
let df(psi,u)=-lam0*u/6, df(psi,r)=lam0*r/6;
g00:=(1/psi**2)*(-vv**4+ww*f2**2/f3**2)$
g01:=(ww/psi**2)*r*f1*f2/f3**2$
g11:=(1/psi**2)*(vv**4+ww*r**2*f1**2/f3**2)$
g22:=(vv**4/psi**2)*r**2$
g33:=g22*sin(theta)**2$
lamric22:=lamric22$
lamric22:=num(lamric22)$
let df(f1,u)=-lam0/6, df(f2,u)=sqrt(lam0/3)*(sqrt(lam0/3)*u/2-1),
    df(f2,r)=lam0*r/6, df(f3,u)=-df(f2,u), df(f3,r)=lam0*r/6;
lamric22:=lamric22$
vv:=1+m0*psi/(2*r);
uu:=(1-m0*psi/(2*r))/(1+m0*psi/(2*r));
ww:=vv**4-uu**2$
lamric22:=lamric22$
```

```
on exp;
lamric22:=lamric22$
length(lamric22);
%u:=1/3; r:=3; theta:=pi/2; %HERE

psi:=1+(lam0/12)*(r**2-u**2)$
let sin(theta)**2=1-cos(theta)**2;
lamric22:=lamric22$

f1:=sqrt(lam0/3)-lam0*u/6$
f2:=lam0*r**2/12+(sqrt(lam0/3)*u/2-1)**2$
f3:=lam0*r**2/12-(sqrt(lam0/3)*u/2-1)**2$
%
lamric22:=lamric22$
pdt:=subdet;
lamric22:=lamric22;
```

The final result is a polynomial of degree 40 in r and u which is too big to be printed here. The computing time is 6 hours 30 minutes.

5.3 Computing symmetries of a field equation

Problem:

Check the symmetries of a field equation for the axial vector component of the torsion in a vacuum within an unusual unified theory of gravitational and hadronic interactions (see [4] for details). The equations are

$\Delta K - \frac{d\delta K}{2} + *(K^d K) = 0$

where K is a 1-form, d is the exterior derivative, * is the Hodge dual operator, and δ is the adjoint of d under *. It is well known that the first two terms are invariant under the conformal group of transformations: the question is whether the third, non-linear term spoils this invariance.

Solution:

Use the package **EXCALC** to convert the equation from the simple geometrical form above into (messy) component equations. Then change the notation (PARTDF → DF) before giving the equations to **CRACK** with the most general linear symmetry ansatz. The computation shows that the conformal invariance is preserved. A similar calculation without the constraining ansatz eventually produces the same set of symmetries.

The problem input is

```
% Base space
coframe e(t) = d t, e(x) = d x, e(y) = d y, e(z) = d z
  with signature 1,-1,-1,-1;
factor e;

% Axial torsion 1-form
pform kt=0,kx=0,ky=0,kz=0;
fdomain kt=kt(t,x,y,z),kx=kx(t,x,y,z),ky=ky(t,x,y,z),kz=kz(t,x,y,z);
kk := kt*e(t) + kx*e(x) + ky*e(y) + kz*e(z);

% Define derivative operators
procedure delta(f)$ # d # f$

procedure laplacian(f)$ -(d delta f + delta d f)$

% Now the equations
eqns := {laplacian kk - d delta kk/2  + #(kk^d kk)};


% Extract the coefficient equations
eqns := for each q in eqns join
          foreach f in {e(t),e(x),e(y),e(z)} collect coeffn(q,f,1);


% A procedure to convert to the notation understood by CRACK
```

```
symbolic operator partdftodf;
symbolic procedure partdftodf l;
% converts expressions using @ (PARTDF) into ones using DF.

aeval reval subst('df,'partdf,l);

% Convert the equations to use the DF operator
neweqns := partdftodf eqns;
```

Evaluation with the linear ansatz:

```
ansatz:={xi_t=t*mtt+x*mtx+y*mty+z*mtz+vt,
         xi_x=t*mxt+x*mxx+y*mxy+z*mxz+vx,
         xi_y=t*myt+x*myx+y*myy+z*myz+vy,
         xi_z=t*mzt+x*mzx+y*mzy+z*mzz+vz,
         eta_kt=ky*nktky+kz*nktkz+kt*nktkt+kx*nktkx+wkt,
         eta_kx=kt*nkxkt+kx*nkxkx+ky*nkxky+kz*nkxkz+wkt,
         eta_ky=kt*nkykt+kx*nkykx+ky*nkyky+kz*nkykz+wkt,
         eta_kz=kt*nkzkt+kx*nkzkx+ky*nkzky+kz*nkzkz+wkt};
functions:={mt,mtx,mty,mtz,vt,
            mxt,mxx,mxy,mxz,vx,
            myt,myx,myy,myz,vy,
            mzt,mzx,mzy,mzz,vz,
            nktky,nktkz,nktkt,nktkx,wkt,
            nkxkt,nkxkx,nkxky,nkxkz,wkt,
            nkykt,nkykx,nkyky,nkykz,wkt,
            nkzkt,nkzkx,nkzky,nkzkz,wkt};
symm:=liepde({neweqns,{kt,kx,ky,kz},{t,x,y,z}},{0,ansatz,functions});
```

Evaluation with the general ansatz:

```
symm:=liepde({neweqns,{kt,kx,ky,kz},{t,x,y,z}},{0,nil,nil});
```

The intermediate PDE system under investigation is :

$$0 = 3\frac{\partial^2 kt}{\partial t^2} - 2\frac{\partial^2 kt}{\partial x^2} - 2\frac{\partial^2 kt}{\partial y^2} - 2\frac{\partial^2 kt}{\partial z^2} - \frac{\partial^2 kx}{\partial t\partial x} - 2\frac{\partial kx}{\partial y}kz + 2\frac{\partial kx}{\partial z}ky - \frac{\partial^2 ky}{\partial t\partial y} + 2\frac{\partial ky}{\partial x}kz - 2\frac{\partial ky}{\partial z}kx - \frac{\partial^2 kz}{\partial t\partial z} - 2\frac{\partial kz}{\partial x}ky + 2\frac{\partial kz}{\partial y}kx$$

$$0 = \frac{\partial^2 kt}{\partial t\partial x} - 2\frac{\partial kt}{\partial y}kz + 2\frac{\partial kt}{\partial z}ky + 2\frac{\partial^2 kx}{\partial t^2} - 3\frac{\partial^2 kx}{\partial x^2} - 2\frac{\partial^2 kx}{\partial y^2} - 2\frac{\partial^2 kx}{\partial z^2} + 2\frac{\partial ky}{\partial t}kz - \frac{\partial^2 ky}{\partial x\partial y} - 2\frac{\partial ky}{\partial z}kt - 2\frac{\partial kz}{\partial t}ky - \frac{\partial^2 kz}{\partial x\partial z} + 2\frac{\partial kz}{\partial y}kt$$

$$0 = \frac{\partial^2 kt}{\partial t\partial y} + 2\frac{\partial kt}{\partial x}kz - 2\frac{\partial kt}{\partial z}kx - 2\frac{\partial kx}{\partial t}kz - \frac{\partial^2 kx}{\partial x\partial y} + 2\frac{\partial kx}{\partial z}kt + 2\frac{\partial^2 ky}{\partial t^2} - 2\frac{\partial^2 ky}{\partial x^2} - 3\frac{\partial^2 ky}{\partial y^2} - 2\frac{\partial^2 ky}{\partial z^2} + 2\frac{\partial kz}{\partial t}kx - 2\frac{\partial kz}{\partial x}kt - \frac{\partial^2 kz}{\partial y\partial z}$$

$$0 = \frac{\partial^2 kt}{\partial t\partial z} - 2\frac{\partial kt}{\partial x}ky + 2\frac{\partial kt}{\partial y}kx + 2\frac{\partial kx}{\partial t}ky - \frac{\partial^2 kx}{\partial x\partial z} - 2\frac{\partial kx}{\partial y}kt - 2\frac{\partial ky}{\partial t}kx + 2\frac{\partial ky}{\partial x}kt - \frac{\partial^2 ky}{\partial y\partial z} + 2\frac{\partial^2 kz}{\partial t^2} - 2\frac{\partial^2 kz}{\partial x^2} - 2\frac{\partial^2 kz}{\partial y^2} - 3\frac{\partial^2 kz}{\partial z^2}$$

The resulting symmetry relations are

$\xi_t = -c_{140}z + c_{290}y + c_{304}x + c_{305} + c_{309}t$

$\xi_x = -c_{229}y - c_{230} + c_{304}t + c_{309}x + c_{321}z$

$\xi_y = c_{229}x + c_{262}z + c_{290}t + c_{309}y + c_{342}$

$\xi_z = -c_{140}t - c_{262}y + c_{309}z - c_{321}x + c_{334}$

$\eta_{kt} = c_{140}kz - c_{290}ky - c_{304}kx - c_{309}kt$

$\eta_{kx} = -c_{229}ky - c_{304}kt - c_{309}kx + c_{321}kz$

$\eta_{ky} = c_{229}kx + c_{262}kz - c_{290}kt - c_{309}ky$

$\eta_{kz} = c_{140}kt - c_{262}ky - c_{309}kz - c_{321}kx$

with free parameters c_{321}, c_{290}, c_{229}, c_{230}, c_{140}, c_{262}, c_{304}, c_{305}, c_{309}, c_{334} and c_{342}.

The total computing time is 48 minutes.

References

[1] A. Barnes: private communication to R. Schöpf.

[2] B. Buchberger; Groebner Bases, An Algorithmic Method in Polynomial Ideal Theory, in: N. K. Bose, Progress, directions and open problems in multidimensional systems theory, Reidel(1985)

[3] B. Champagne, W. Hereman and P. Winternitz: The computer calculation of Lie point symmetries of large systems of differential equation, Comp. Phys. Comm. 66, 319-340 (1991)

[4] W. D. Drechsler: Class Quant Grav 6(1989)623-657

[5] A. C. Hearn: REDUCE 3.5 User's Manual, The Rand Corporarion, CA (1993)

[6] F. W. Hehl, V. Winkelmann, H. Meyer: Computer Algebra, ein Kompaktkurs über die Anwendung von REDUCE (in German), Springer(1992). English translation in: D. Stauffer, F.W. Hehl, N. Ito, V. Winkelmann, J.G. Zabolitzky: Computer Simulation and Computer Algebra - Lectures for Beginners, Springer (1993)

[7] J. Hietarinta: Solving the constant quantum Yang-Baxter equation in 2 Dimensions with massive use of factorizing Gröbner basis computations, Proceedings of the International Symposium on Symbolic and Algebraic Computation (ISSAC) 1992, pp 350-357, ACM Press

[8] J.Hietarinta: Solving the two-dimensional constant quantum Yang-Baxter equation, J. Math. Phys. 34, 1725-1756 (1993)

[9] d'Inverno, Introducing Einstein's Relativity, Oxford University Press (1992)

[10] V.I. Karpman, Phys. Lett. A 136, 216 (1989)

[11] D.F. Lawden: Elliptic Functions and Applications, Springer (1989)

[12] M. MacCallum, F. Wright: Algebraic Computing with REDUCE, Oxford University Press (1991)

[13] Dermott McCrae died in 1993. The present program has been transmitted by F. W. Hehl.

[14] V. W. Noonburg: A Neural Network Modeled by an Adaptive Lotka-Volterra System, SIAM J. Appl. Math 49(6) (1989)

[15] G. Reimer, private communication to H. M. Möller

[16] Sexl, Urbantke: Gravitation und Kosmologie, BI, Mannheim (1983)

[17] B. Simon: Symbolic Math Software, PC Magazine, August 1992

[18] Johannes Ueberberg, Einführung in die Computeralgebra mit REDUCE, BI-Wissenschaftsverlag, Mannheim(1992)