On the Harmonic and Monogenic Decomposition of Polynomials

F. BRACKX†, D. CONSTALES‡*, A. RONVEAUX§ AND H. SERRA§t
† Seminarie voor wiskundige analyse, St-Pietersnieuwstraat 39, B 9000 Gent, Belgium
‡ Seminarie voor algebra en functionaalanalyse, Galglaan 2, B 900 Gent, Belgium
§ Département de Physique, Facultés universitaires Notre-Dame de la Paix, B 5000 Namur, Belgium

(Received 7 January 1988)

The decomposition of polynomials in terms of spherical harmonics is widely used in various branches of analysis. In this paper we describe a set of REDUCE procedures generating this decomposition and its more general, monogenic, counterpart in Clifford analysis. We then illustrate their use by inverting the Laplacian and the Dirac operator on both Euclidean and Minkowski spaces.

1. Harmonic Polynomials

Let \( \mathcal{P}_k \) denote the space of real-valued homogeneous polynomials of degree \( k \) in the real variables \( (x_1, x_2, \ldots, x_n) \) and \( \mathcal{H}_k \) the subspace of harmonic polynomials of degree \( k \).

As is well known (Vilenkin, 1969), any \( f_k \in \mathcal{P}_k \) may be uniquely decomposed as

\[
 f_k = h_k + r^2 h_{k-2} + \cdots + r^{2[k/2]} h_{k-2[k/2]},
\]

where \( h_k \in \mathcal{H}_k \) and \( r^2 \) stands for \( |x|^2 = x_1^2 + \cdots + x_n^2 \). This result can be elegantly proved following an idea of Stein & Weiss (1971). With the inner product

\[
 \langle f, g \rangle_k = \sum_{|\alpha|=k} a_\alpha f_\alpha g_\alpha, \quad f, g \in \mathcal{P}_k,
\]

where \( f_\alpha \) is the coefficient of \( x_1^{\alpha_1} \cdots x_n^{\alpha_n} \), \( \alpha = (\alpha_1, \ldots, \alpha_n) \) being a multi-index, the space \( \mathcal{P}_k \) is turned into a Hilbert space. Notice that this inner product may also be written as

\[
 \langle f, g \rangle_k = f(\partial) g,
\]

where \( f(\partial) \) is the differential operator obtained by substituting \( \partial_{x_i} \) for \( x_i \) \( (i = 1, \ldots, n) \) in \( f \). It is thus seen at once that for \( f \in \mathcal{P}_{k-2} \) and \( g \in \mathcal{P}_k \),

\[
 \langle |x|^2 f, g \rangle_k = \langle f, \Delta g \rangle_{k-2},
\]

from which it follows that

\[
 \mathcal{P}_k = \mathcal{H}_k \oplus |x|^2 \mathcal{P}_{k-2}.
\]

Indeed, if for some \( g \in \mathcal{P}_k \) and for all \( f \in \mathcal{P}_{k-2} \), \( \langle |x|^2 f, g \rangle_k = 0 \) holds, so will \( \langle f, \Delta g \rangle_{k-2} = 0 \) and \( \Delta g = 0 \), which means that the orthogonal complement of \( |x|^2 \mathcal{P}_k \) is a subspace \( \mathcal{H}_k \). But if \( h = |x|^2 f \in \mathcal{H}_k \cap |x|^2 \mathcal{P}_{k-2} \),

\[
 \langle h, h \rangle_k = \langle |x|^2 f, h \rangle_k = \langle f, \Delta h \rangle_{k-1} = 0,
\]

* Research Assistant of the Belgian National Fund for Scientific Research N.F.W.O.
and hence $h = 0$. So any $f_k \in \mathcal{P}_k$ may be uniquely decomposed as

$$f_k = h_k + r^2 f_{k-2}, \quad h_k \in \mathcal{H}_k, \quad f_{k-2} \in \mathcal{P}_{k-2}$$

and repetition of this argument proves (1). To perform this decomposition for a given homogeneous polynomial $f_k$, we rely on the identity

$$\Delta(r^2 h_{k-2l}) = (2l)(2k - 2l + n - 2)r^{2l-2}h_{k-2l},$$
valid for all $l > 0$ and homogeneous harmonic polynomials $h_{k-2l}$ of degree $k - 2l$. The algorithm can be sketched as follows: if the degree $k < 2$ the decomposition is trivial, $f_k$ being harmonic. Otherwise, we first decompose

$$f_k = 2 \cdot (2k + n - 4)h_{k-2} + \cdots.$$

The REDUCE procedure for obtaining this decomposition of a given homogeneous polynomial $f_k$ is called $HHD(f_k, k, x, r^2)$. Here $x$ is an operator and the polynomial $f_k$ is homogeneous of degree $k$ in $x(1), \ldots, x(n)$; these operator values can be reassigned sensible (kernel) values such as $x, y, z, t$. Of course, $f_k$ can also depend on other parameters. $r^2$ stands for any expression supplied by the user to represent $x_1^2 + \cdots + x_n^2$.

Such a procedure might, for instance, be used to construct Symmetry Adapted Functions (S.A.F.) in Quantum Mechanics. A projection technique, similar to the $HHD$ procedure, was already used (Ronveaux & Saint-Aubin, 1983) to construct cubical harmonics starting from polynomials invariant under a finite subgroup of $O(3)$.

In the examples throughout the paper, REDUCE output has been modified, in that $x^p$ stands for $x(p) x(q) x(r) \ldots$, the multiplication sign is suppressed, fractions are written more neatly and the terms of polynomials are sometimes rearranged. There is an example of the original REDUCE output just after the program examples. Execution times are given for a microcomputer with a 8 MHz 68000 CPU.

EXAMPLE 1.1 (time: 1295 ms).

$$HHD(x^3 + x^2, 3, x, 2, r^2) = a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 = (x^3 + x^2) + a_1 (r^2 (x^3 + x^2)).$$

For a general, not necessarily homogeneous, polynomial $f$, the decomposition is obtained by splitting it in homogeneous parts of decreasing degrees in $x$ and adding the contributions of the previous algorithm when called on each of them. The procedure is called $HD(f, x, n, r^2)$ and gives the unique harmonic decomposition

$$f = \sum_{j=0}^{[m/2]} r^2 h_{m-2j}, \quad \Delta h_{m-2j} = 0. \quad (2)$$

EXAMPLE 1.2 (time: 3280 ms).

$$HD(x_1 + x_2 + x_3, x, 2, r^2) = x_1^2 - x_1^2 x_2 - x_2^2 x_3 + x_3 x_1 x_2 + r^2 (2x_2 + 2x_3).$$

2. Monogenic Polynomials

The concept of monogenicity generalises, in a rather natural manner, to higher dimensions the notion of holomorphicity in the complex plane. It applies to functions defined in the Euclidean space $\mathbb{R}^n$ and taking their values in the Clifford algebra $\mathbb{R}_n$. This associative, but non-commutative, algebra $\mathbb{R}_n$ has as a basis

$$\{e_A = e_{i_1} \cdots e_{i_r} : A \in \mathcal{P} \{1, \ldots, n\}, A = \{i_1, \ldots, i_r\}, 1 \leq i_1 < \cdots < i_r \leq n\}.$$
where \((e_1, \ldots, e_n)\) is an orthonormal basis of \(\mathbb{R}^n\), while the multiplication in \(\mathbb{R}^n\) is governed by the rules
\[ e_i e_j + e_j e_i = -2\delta_{ij} e_0, \]
e_0 being the identity element corresponding to \( A = \phi \). For a study of Clifford algebra we refer to Porteous (1981). REDUCE programs for the calculation in such a Clifford algebra were already developed by Brackx et al. (1987).

A function \( f: \Omega \to \mathbb{R}^n, \) \( \Omega \) open in \( \mathbb{R}^n \), is said to be \((l)\)-monogenic \((l \in \mathbb{N})\) if \( f \) is in \( C^l(\Omega) \) and satisfies \( Df = 0 \) in \( \Omega \), where \( D \) is the Dirac operator
\[ D = \sum_{i=1}^n e_i \partial_{x_i}. \]

If \( l = 1 \) we call such a function monogenic for short. For a detailed study of the monogenic functions we refer to Brackx et al. (1982).

Just as in the case of a holomorphic function of one complex variable, where the real and imaginary parts are harmonic, the components \( f_A \) of an \((l)\)-monogenic function \( f = \sum_A e_A f_A, f_A: \Omega \to \mathbb{R}, \) are \((l+1)/2\)-harmonic, i.e.
\[ \Delta^{(l+1)/2} f_A = 0 \] in \( \Omega \). This is simply due to the fact that \( D \) splits the Laplacian \( \Delta = -\Delta \) and it implies a refinement of the notion of harmonicity, as is illustrated by the following results.

Let \( \mathcal{X}_k \) denote the module of \( \mathbb{R}^n \)-valued homogeneous polynomials of degree \( k \) and \( \mathcal{M}_k \) the submodule of monogenic homogeneous polynomials of degree \( k \). Next, let \( x \) stand for the function
\[ x = \sum_{i=1}^n e_i x_i. \]

**Proposition.** The module \( \mathcal{X}_k \) may be written as the direct sum
\[ \mathcal{X}_k = \mathcal{M}_k \oplus x \mathcal{X}_{k-1}. \]

**Proof.** This is similar to the one of the harmonic decomposition as sketched in section 1. The module \( \mathcal{X}_k \) becomes a Hilbert module for the inner product
\[ \langle f, g \rangle = \sum_{|\alpha| = n} \alpha! f^\alpha g^\alpha, \quad f, g \in \mathcal{X}_k, \]
which also may be written as
\[ \langle f, g \rangle_k = \bar{f} \partial g, \quad (\ast) \]
where the bar denotes the involution defined by the rules \( \bar{\lambda} = \lambda \) if \( \lambda \in \mathbb{R}, \) \( \bar{e}_i = -e_i \) for \( i = 1, \ldots, n \) and \( ab = \bar{a} \bar{b} \) for all \( a, b \in \mathbb{R}^n \). The orthogonal decomposition then follows from the observation \((\ast)\) since, for \( f \in \mathcal{X}_{k-1} \) and \( g \in \mathcal{X}_k \),
\[ \langle xf, g \rangle_k = \bar{x} \partial_x \bar{g} = \bar{f} \partial_x \bar{g} \bar{D}g = -\bar{f} \partial_x \bar{D}g = -\langle f, Dg \rangle_{k-1}. \]

The above proposition leads at once to the following unique decomposition of the homogeneous polynomial \( f_k \in \mathcal{X}_k \):
\[ f_k = m_k + x m_{k-1} + x^2 m_{k-2} + \cdots + x^k m_0, \quad (3) \]
where \( m_j \in \mathcal{M}_j \) for all \( j = 0, \ldots, k \). That this decomposition is indeed a refinement of the harmonic decomposition \((1)\) is shown by the fact that
\[ h_j = m_j + x m_{j-1} \in \mathcal{M}_j \]
and \( x^2 = -r^2 \). Indeed
\[ D(xm_{j-1}) = (2-n-2j)m_{j-1}, \]
and hence
\[ \Delta h_j = -D^2 h_j = 0. \]

To compute the monogenic decomposition of a given homogeneous polynomial \( f_k \) we first perform its harmonic decomposition, and split each \( h_j = h_{k-2j} \) of non-zero degree into \( m_j + x m_{j-1} \) according to the previous formula. An \( h_0 \) obviously does not have to be split—it is already monogenic. For a general polynomial we decompose each homogeneous part separately and add all the results.

The corresponding REDUCE procedures are called \( HMD(f_k, k, x, n, xvec) \) and \( MD(f, x, n, xvec) \). Here, \( x \) has the same meaning as in \( HHD \) and \( HD \) and \( xvec \) stands for any expression the user chooses to represent \( x \). In the decomposition. For some applications this will have to be a non-commutative object.

**Example 2.1** (time: 7020 ms).
\[
HMD((x^3 + x^2)e_0, 3, x, 2, x) = x^3(e_1 + e_2 + e_3) + x^2(e_1 e_2 (x_2 - x_1) - e_0 (x_1 + x_2))
+ x(e_2 (x^2_1 - x^2_2) + e_1 (x^2_2 - x^2_1) + (e_1 + e_2) x_1 x_2)
+ (e_1 e_2 + e_0) x^2_2 + ((- e_1 e_2 + e_0) x^2_1
+ e_1 e_2 (x_1 x_2 - x_1^2 x_2) - e_0 (x_1 x^2_2 + x^2_1 x_2)),
\]

3. **Polyharmonic and Polymonogenic Polynomials**

It was already pointed out that bracketing terms together two by two in the monogenic decomposition (3), the harmonic decomposition (1) is again obtained. In the same order of ideas, when terms are bracketed together \( l \) by \( l \) in (3), a unique decomposition is obtained in terms of \( (l) \)-monogenic polynomials

\[
f_k = v_k + x^l v_{k-1} + \cdots, \quad D^l v_j = 0, \quad \text{for all } j.
\]

Notice that (2l)-monogenicity is equivalent to (l)-harmonicity, in particular for \( l = 2 \), (4) represents a biharmonic decomposition.

In the corresponding REDUCE procedures \( HPMD(f_k, k, x, n, s, l) \) and \( PMD(f, x, n, s, l) \) the degree of monogenicity is incorporated as an argument and \( s \) is a user-supplied symbol for \( x^l \). They work by regrouping the output from the harmonic decomposition (if \( l \) is even) or from the monogenic decomposition (if \( l \) is odd).

**Example 3.1.**
\[
HPMD(x^3_1 + x^3_2, 3, x, 2, x^2, 2) = x^3(-3x_1 - 3x_2) + (\frac{1}{4} (x^3_1 + x^3_2) - \frac{3}{4} (x^2_1 x_2 + x_1 x^2_2)),
\]
time: 1285 ms.

\[
HPMD(f_3, 3, x, 2, x^3, 3) = x^3(\frac{1}{4} e_1 + \frac{1}{4} e_2 + (\frac{1}{8} e_0 - \frac{1}{4} e_1 e_2)x^3_1 + (\frac{1}{8} e_0 + \frac{1}{4} e_1 e_2)x^3_2
+ (\frac{1}{8} e_1 e_2 - \frac{1}{8} e_0) x_1 x^2_2 - (\frac{1}{8} e_1 e_2 + \frac{1}{8} e_0) x^2_1 x_2),
\]
time: 7390 ms.

4. **Generalisation to the Pseudo-metric Case**

The above decomposition may be generalised to the case of the pseudo-Laplacian and pseudo-Dirac operators, where the euclidean distance function

\[
r^2 = \sum_{i=1}^{n} x^2_i
\]
is replaced by its pseudo-euclidean counterpart
\[ \tilde{\rho}^2 = \sum_{i=1}^{n} \beta_i x_i^2, \]
the \( \beta_i \) being real non-zero constants of different signature. If there are \( p \) negative and \( q \) positive \( \beta_i, \) \( p + q = n, \) we put \( a_i = |\beta_i|; \) the corresponding pseudo-Laplacian now reads
\[ \tilde{\Delta} = \sum_{i=1}^{n} \frac{1}{a_i} \frac{\partial^2}{\partial x_i^2} = - \sum_{i=1}^{p} \frac{1}{a_i} \frac{\partial^2}{\partial x_i^2} + \sum_{i=p+1}^{n} \frac{1}{a_i} \frac{\partial^2}{\partial x_i^2}, \]
and it may be shown that all classical formulae, as found in e.g. Vilenkin (1969), about Laplacians of homogeneous polynomials go through. This immediately leads to the following decomposition of the homogeneous polynomial \( f_k: \)
\[ f_k = \tilde{h}_k + \rho^2 \tilde{h}_{k-2} + \cdots + \rho^{2(n/2)} \tilde{h}_{k-2(n/2)}, \]
where \( \tilde{h}_k \) stands for a homogeneous polynomial of degree \( j \) satisfying \( \tilde{\Delta} h_j = 0. \)

The corresponding REDUCE procedures are called \( HGHDF(k, x, p, q, a, s) \) and \( GHD(f, x, p, q, a, s). \) Again, \( s \) is a symbol for \( \tilde{\rho}^2 \) supplied by the user and \( a \) is an operator whose values \( a(i) \) are the \( a_i. \)

**Example 4.1 (time: 2720 ms).**

\[
HGHDF(x_1^2 + x_2^2, 3, x_1, 1, 1, a, \tilde{\rho}^2) = \left[ \tilde{\rho}^2(-\frac{2a_2 x_1}{a_1} + (\frac{2a_2}{a_1} x_2) + \frac{a_2 x_1^2}{a_1} x_1 + \frac{1}{2} a_2 a_1 x_1^2 + \frac{1}{4} a_2 a_1 x_1^2 + \frac{3}{2} a_1 x_2 x_1^2) \right] / (a_1 a_2),
\]

In order to obtain a splitting of the pseudo-Laplacian \( \tilde{\Delta} \) by means of a Clifford–Dirac operator, we introduce the Clifford algebra \( \mathbb{R}_{p,q} \) constructed over \( \mathbb{R}_{p+q} \) with basis \( \{e_1, \ldots, e_p, e_{p+1}, \ldots, e_{p+q} \} \) where
\[
e_i e_j + e_j e_i = 2\delta_{ij} \beta_i.
\]

First notice that for \( x = \sum_{i=1}^{p+q} x_i e_i, \)
\[ x^2 = \frac{p}{\sum_{i=1}^{p} a_i x_i^2 + \sum_{i=p+1}^{p+q} a_i x_i^2} = \tilde{\rho}^2. \]

Next, introducing the pseudodirac operator
\[ \tilde{D} = \sum_{i=1}^{p+q} e_i \frac{1}{a_i} \frac{\partial}{\partial x_i} = - \sum_{i=1}^{p} \frac{a_i}{a_1} \frac{\partial}{\partial x_i} + \sum_{i=p+1}^{p+q} \frac{a_i}{a_1} \frac{\partial}{\partial x_i}, \]

it is at once observed that \( \tilde{D}^2 = \tilde{\Delta}. \)

In a completely similar way, as described in sections 2 and 3, we arrive at the following REDUCE procedures:

(a) \( HGMDF(f_k, k, x, p, q, a, x) = \tilde{h}_k + x \tilde{h}_{k-1} + \cdots + x^k \tilde{h}_0 \) where for all \( j, \tilde{D} \tilde{h}_j = 0; \)
(b) \( GMD(f, x, p, q, a, x) = \tilde{h}_k + x \tilde{h}_{k-1} + \cdots + x^k \tilde{h}_0 \) with \( f \) a polynomial, not necessarily homogeneous;
(c) \( HPGMD(f_k, k, x, p, q, a, x^j, l) = \tilde{v}_k + x \tilde{v}_{k-1} + \cdots, \) where for all \( j, \tilde{D} \tilde{v}_j = 0; \)
(d) \( PGMDF(f, x, p, q, a, x^j, l) = \tilde{v}_k + x \tilde{v}_{k-1} + \cdots, \) for any polynomial \( f. \)

**Example 4.2.**

\[
HPGMDF(x_1^2 + x_2^2, 3, x, 1, 1, a, x^2, 2)
= (x^2(-\frac{3}{2} a_2 x_1 + \frac{3}{2} a_1 x_2) + (\frac{3}{2} a_2 x_1^2 x_1 + \frac{3}{2} a_2 a_1 x_1^2 + \frac{3}{2} a_2 a_1 x_1^2 + \frac{3}{2} a_2 a_1 x_1^2) / (a_1 a_2),
\]
time: 2810 ms.
5. Inversion of the (Pseudo) Laplacian and (Pseudo) Dirac Operator

Using the harmonic decomposition (1) it can be proved that for a given polynomial \( f \) of degree \( m \), there exists a unique polynomial \( g \) of degree \( m+2 \) such that \( \Delta g = f \) and 
\[
g = |x|^2 g',
\]
\( g' \) being some polynomial of degree \( m \).

This inversion of the Laplacian for polynomials is obtained by the REDUCE procedure \( IL(f, x, n) \); it decomposes the homogeneous parts of \( f \) to obtain the harmonics and, dividing them by suitable constants (given in section 1), inverts the Laplacian for each of them.

**Example 5.1 (time: 2525 ms).**

\[
IL(x^3_1 + x^3_2, x, 2) = 5x_2^2 - \frac{1}{64}x_2^2 x_1 + \frac{1}{32}x_2^2 x_1^2 + \frac{3}{32}x_2^2 x_1^3 - \frac{1}{64}x_2 x_1^4 + \frac{1}{32}x_1^4.
\]

Similar results hold for the other operators mentioned in the previous sections. The corresponding REDUCE procedures are:

(a) \( ID(f, x, n) = g_1 \), where \( g_1 \) is the unique polynomial of degree \( m+1 \) for which \( Dg_1 = f \) and 
\[
g_1 = xg_1',
\]
g\( ' \) being some polynomial of degree \( m \);

(b) \( IGL(f, x, p, q, a) = \tilde{g} \), where \( \tilde{g} \) is the unique polynomial of degree \( m+2 \) of the form 
\[
\tilde{g} = r^2 \tilde{g}'
\]
satisfying \( \Delta \tilde{g} = f \);

(c) \( IGD(f, x, p, q, a) = \tilde{g}_1 \), where \( \tilde{g}_1 \) is the unique polynomial of degree \( m+1 \) of the form 
\[
\tilde{g}_1 = xg_1'
\]
satisfying \( \Delta \tilde{g}_1 = f \).

6. The REDUCE Program

REDUCE being well suited to polynomial manipulation, we encountered no difficulty while developing these procedures. Also, REDUCE’s handling of non-commutative objects is quite adequate for the Clifford algebra needed here. To give an idea of what the program looks like, we include the code for \( HHD, HD \) and the other procedures required for them. The interested reader can obtain the full REDUCE program, for a nominal fee, by simple request to the first author. The examples were originally worked out with REDUCE 3.2 installed on a VAX 750.

% Some REDUCE code for the decomposition of polynomials in terms of harmonic polynomials.
% The laplacian in Cartesian coordinates.

\begin{verbatim}
algebraic procedure Lap(expression, coordinate, dimension); for j := 1 : dimension sum dr(expression, coordinate(j), 2);
\end{verbatim}

% Some useful functions to link algebraic and symbolic mode.

\begin{verbatim}
symbolic procedure APlus(x, y); aeval list (plus, x, y);
symbolic procedure ADifference(x, y); aeval list (difference, x, y);
symbolic procedure ATimes(x, y); aeval list (times, x, y);
symbolic procedure AQuotient(x, y); aeval list (quotient, x, y);
\end{verbatim}

% The squared norm in Cartesian coordinates.

\begin{verbatim}
algebraic procedure NormSquared(coordinate, dimension);
for j := 1 : dimension sum coordinate(j)**2;
\end{verbatim}

% Make a polynomial out of a sublist of coefficients (for decreasing powers) and a variable – pastSubList can be nil to specify the whole of subList.

\begin{verbatim}
algebraic procedure MakePoly(coefficients, variable, pastSubList); if pastSubList = nil then
  poly(coefficients, variable)
else
  poly(coefficients, pastSubList)
\end{verbatim}
symbolic procedure SubList2Poly(subList, pastSubList, variable);
begin
    scalar theResult;
    theResult := nil;
    while subList neq pastSubList do
        theResult := APlus(ATimes(variable, theResult), car subList);
        subList := cdr subList;
    return theResult;
end;

% The main decomposition function: it takes a homogeneous polynomial
% and related data to build a list of the harmonics involved in its
% decomposition, this list is in increasing order of degree.
% Only accessible from symbolic mode!

symbolic procedure HomHarmEx(homPoly, degree, coordinate, dimension, normSquared);
if degree < 2
    then list homPoly % this is the trivial case
else
    begin
        scalar workList, count1, count2;
        count1 stands for '2l' in the paper, count2 for '2k - 2l + n - 2'
        if o keypad degree then count1 := degree-1 else count1 := degree;
        count2 := 2*degree + dimension - 2 - count1;
        % call this function on homPoly's Laplacian
        workList := HomHarmEx(Lap(homPoly, coordinate, dimension), degree-2, coordinate, dimension, normSquared);
        % adapt the harmonic polynomials
        for each subList on workList do
            (rplaca(subList, AQuotient(car subList, count1, count2));
            count1 := count1-2;
            count2 := count2+2);
        % compute the highest-degree harmonic and return
        return nconc(workList, list
            ADifference(homPoly, ATimes(SubList2Poly(workList, nil, normSquared), normSquared));
    end;
% The harmonic decomposition of a homogeneous polynomial relying on the
% previous function – accessible from both algebraic and symbolic modes.

symbolic operator HHD;
symbolic procedure HHD(homPoly, degree, coordinate, dimension, normSqrSymbol);
SubList2Poly(HomHarmEx(homPoly, degree, coordinate, dimension, NormSquared(coordinate, dimension)), nil, normSqrSymbol);
The harmonic decomposition of a general polynomial.

```plaintext
algebraic procedure HD(expression, coordinate, dimension, normSqrSymbol);
begin
    scalar leftToDo, currentDegree, theResult;
    leftToDo := num expression;
    theResult := 0;

    % Put a marker to find the homogeneous parts; this marker should not occur
    % anywhere else.
    for j := 1 : dimension do leftToDo := sub(x(j) = ! degreeMarker*x(j), leftToDo);

    % Call the previous procedure on each homogeneous part.
    while (currentDegree := deg(leftToDo, ! degreeMarker)) > 0 do
        theResult := theResult +
            HHD(lcof(leftToDo, ! degreeMarker), currentDegree,
            coordinate, dimension, normSqrSymbol);
        leftToDo := reduct(leftToDo, ! degreeMarker));

    % We treat the constant part separately
    theResult := theResult + leftToDo;
    return theResult/(den expression);
end;
```

An example of the original REDUCE output:

```
HHD(x(1)**3 + x(2)**3, 3, x, 2, r**2);
(x(2)**3 - 3*x(2)**2*x(1) - 3*x(2)*x(1)**2 + 3*x(2)*r**2 + x(1)**3 + 3*x(1)*r**2)/4
```

time: 1295 ms

```
HD(x(1) + x(2) + x(1)**3 + x(2)**3, x, 2, r**2);
(x(2)**3 - 3*x(2)**2*x(1) - 3*x(2)*x(1)**2 + 3*x(2)*r**2 + 4*x(2) + x(1)**3 + 3*x(1)
    *r**2 + 4*x(1))/4
```

time: 3280 ms

References

Constales, D. Automorphisms of monogenic functions (in preparation).