UNIVERSITÀ DEGLI STUDI FIRENZE

FLORE
Repository istituzionale dell'Università degli Studi di Firenze

Formalizing basic quaternionic analysis

Questa è la Versione finale referata (Post print/Accepted manuscript) della seguente pubblicazione:

Original Citation:

Formalizing basic quaternionic analysis / Andrea Gabrielli; Marco Maggesi. - STAMPA. - 10499:(2017), pp. 225-240. (Intervento presentato al convegno ITP 2017 - Interactive Theorem Proving 2017 tenutosi a Brasilia (Brasile) nel 26 - 29 settembre 2017) [10.1007/978-3-319-66107-0_15].

Availability:

This version is available at: 2158/1090335 since: 2021-03-23T23:04:54Z

Publisher:

Springer

Published version:

DOI: 10.1007/978-3-319-66107-0_15

Terms of use:

Open Access

La pubblicazione è resa disponibile sotto le norme e i termini della licenza di deposito, secondo quanto stabilito dalla Policy per l'accesso aperto dell'Università degli Studi di Firenze (https://www.sba.unifi.it/upload/policy-oa-2016-1.pdf)

Publisher copyright claim:

(Article begins on next page)

03 July 2024

Formalizing basic quaternionic analysis

Andrea Gabrielli[1] and Marco Maggesi[2]

[1] University of Florence, Italy
andrea.gabrielli@unifi.it
[2] University of Florence, Italy
marco.maggesi@unifi.it
http://www.math.unifi.it/~maggesi/

Abstract We present a computer formalization of quaternions in the HOL Light theorem prover. We give an introduction to our library for potential users and we discuss some implementation choices.
As an application, we formalize some basic parts of two recently developed mathematical theories, namely, slice regular functions [5] and PH-curves [3].

1 Introduction

Quaternions are a well-known and elegant mathematical structure which lies in the intersection of algebra, analysis and geometry. They have a wide range of theoretical and practical applications from mathematics, physics to CAD, computer animations, robotics, signal processing and avionics.

Arguably, a computer formalization of quaternions can be useful, or even essential, for further developments in pure mathematics or for a wide class of applications in formal methods.

In this paper we present a formalization of quaternions in the HOL Light theorem prover. The aim of this paper is to give a quick introduction of our library to potential users and to discuss some implementation choices.

The structure of the paper is in two parts. First we describe the core of our library, which is already available in the HOL Light distribution.

Next, we outline two applications to recently developed mathematical theories which should serve as further examples and as a testbed for our work. More precisely, we give the basic definition and some basic theorems about slice-regular quaternionic functions (Section 6) and pythagorean-hodograph curves (Section 7).

2 Background and related work

The HOL Light theorem prover furnishes an extensive library about multivariate analysis [6] and complex analysis [7] which has been constantly and steadily extended over the years by Harrison, the main author of the system.

* This work has been supported by GNSAGA-INdAM and MIUR.

Our aim is to try to further improve this work by adding contributions in (hyper)complex analysis. One previous work along this line was the proof of the Cartan fixed point theorems [1] by Ciolli, Gentili and the second author of this paper.

In a broader context, quaternions are one of the simplest examples of geometric algebra (technically, real Clifford algebra). In this respect, we mention two recent related efforts. Fuchs and Théry [4] devise an elegant inductive data structure for formalizing geometric algebra. More recently, Ma et al. [8], provide a formalization in HOL Light of Conformal Geometric Algebra. In principle, these contribution can be integrated with our work, but at the present stage, we focused entirely on the specific case of quaternions.

3 The core library

Quaternions were "invented" by Hamilton in 1843. From their very inception, they was meant to represent, in an unified form, both scalar and vector quantities. Informally, a quaternion q is expressed as a formal combination

$$q = a + b\,\mathbf{i} + c\,\mathbf{j} + d\,\mathbf{k} \in \mathbb{H} \qquad a, b, c, d \in \mathbb{R}$$

where $\mathbf{i}, \mathbf{j}, \mathbf{k}$ are *imaginary units.* A product structure is induced by the following identities

$$\begin{aligned} \mathbf{ij} &= \mathbf{k} = -\mathbf{ji} \\ \mathbf{jk} &= \mathbf{i} = -\mathbf{kj} \\ \mathbf{ki} &= \mathbf{j} = -\mathbf{ik} \\ \mathbf{i}^2 &= \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1 \end{aligned}$$

which build the set $\mathbb{H}$ of quaternions into a skew field.

It turns to be useful to consider a number of different possible decompositions for a quaternion q, as briefly sketched in the following schema (here $\mathbb{I} = \mathbb{R}^3$ can be interpreted, depending on the context, as the imaginary part of $\mathbb{H}$ or the 3-dimensional space):

$$\begin{aligned} q &= \underbrace{a}_{\operatorname{Re} q} + \underbrace{b\,\mathbf{i} + c\,\mathbf{j} + d\,\mathbf{k}}_{\operatorname{Im} q} && \in \mathbb{H} = \mathbb{R} \oplus \mathbb{I} \\ &= \underbrace{a}_{\text{scalar}} + \underbrace{b\,\mathbf{i} + c\,\mathbf{j} + d\,\mathbf{k}}_{\text{3d-vector}} && \in \mathbb{R}^4 = \mathbb{R} \oplus \mathbb{R}^3 \\ &= \underbrace{a + b\,\mathbf{i}}_{z\in\mathbb{C}} + \underbrace{(c + d\,\mathbf{i})}_{w\in\mathbb{C}}\,\mathbf{j} && \in \mathbb{H} \simeq \mathbb{C} \oplus \mathbb{C} \end{aligned}$$

For the sake of consistency, whenever possible, our development mimics Harrison's formalization of complex numbers present in the HOL Light standard library [7]. Following this path, we define the data type `` `:quat` ``of quaternions as an alias for the type of 4-dimensional vectors `` `:real^4` ``. This approach has a fundamental benefit from the fact that the we inherit immediately from the general theory of euclidean spaces the appropriate metric, topology, analytical and real-vector space structure.

A set of auxiliary constants for constructing and destructing quaternions are defined to setup a suitable abstraction barrier. They are listed in Table 3.

Table 1. Constructors and destructors for the `` `:quat` `` datatype

Constant name	Type	Description
`Hx`	`:real->quat`	Embedding $\mathbb{R} \to \mathbb{H}$
`ii`, `jj`, `kk`	`:quat`	Imaginary units $\mathbf{i}, \mathbf{j}, \mathbf{k}$
`quat`	`:real#real#real#real->quat`	Generic constructor
`Hv`	`:real^3->quat`	Embedding $\mathbb{R}^3 \to \mathbb{H}$
`Re`	`:quat->real`	Real component
`Im1`, `Im2`, `Im3`	`:quat->real`	Imaginary components
`HIm`	`:quat->real^3`	Imaginary part
`cnj`	`:quat->quat`	Conjugation
`real`	`:quat->bool`	Whether a quaternion is real

This can be summarized with the following theorem

```
QUAT_EXPAND
|- !q. q = Hx(Re q) + ii*Hx(Im1 q) + jj*Hx(Im2 q) + kk*Hx(Im3 q)
```

which is the quaternionic analogous of the following theorem for complex numbers

```
COMPLEX_EXPAND
|- !z. z = Cx(Re z) + ii*Cx(Im z)
```

With these notations in place, the multiplicative structure can be expressed with an explicit formula

```
let quat_mul = new_definition
  `p * q =
   quat
   (Re p *  Re q - Im1 p * Im1 q - Im2 p * Im2 q - Im3 p * Im3 q,
    Re p * Im1 q + Im1 p *  Re q + Im2 p * Im3 q - Im3 p * Im2 q,
    Re p * Im2 q - Im1 p * Im3 q + Im2 p *  Re q + Im3 p * Im1 q,
    Re p * Im3 q + Im1 p * Im2 q - Im2 p * Im1 q + Im3 p *  Re q)`;;
```

the inverse of a quaternion is defined analogously. Moreover, we also provide axiliary theorems that reduce the already defined additive and metric structure in the same language, e.g.,

```
quat_add
|- p + q =
   quat(Re p + Re q,Im1 p + Im1 q,Im2 p + Im2 q,Im3 p + Im3 q)

quat_norm
|- norm q =
   sqrt(Re q pow 2 + Im1 q pow 2 + Im2 q pow 2 + Im3 q pow 2)
```

Notice that several notations (`Re`, `ii`, `cnj`, `real`, ...) overlap in the complex and quaternionic case and, more generally, with the ones of other number systems (`+`, `*`, ...). HOL Light disposes of an overloading mechanism that uses the type inference to select the right constant associated to a given notation.

3.1 Computing with quaternions

After settling the basic definitions, we define a simple automated procedure for proving quaternionic algebraic identities which consists in just two steps: (1) rewrite the expression in real components, (2) use an automated procedure for the real field (essentially one involving polynomial normalization, elimination of fractions and Gröbner Basis):

```
let SIMPLE_QUAT_ARITH_TAC =
  REWRITE_TAC[QUAT_EQ; QUAT_COMPONENTS; HX_DEF;
              quat_add; quat_neg; quat_sub; quat_mul;
              quat_inv] THEN
  CONV_TAC REAL_FIELD;;
```

This approach, although very crude, allows us to prove directly nearly 60 basic identities, e.g.,

```
let QUAT_MUL_ASSOC = prove
 (`!x y z:quat. x * (y * z) = (x * y) * z`,
  SIMPLE_QUAT_ARITH_TAC);;
```

and it is also occasionally useful to prove ad hoc identities in the middle of more complex proofs. In this way, we quickly bootstrap a small library with the essential algebraic results which needed for building more complex procedures and theorems.

Next, we provide a conversion `RATIONAL_QUAT_CONV` for evaluating literal algebraic expression. This is easily assembled from elementary conversions for each basic algebraic operation (`RATIONAL_ADD_CONV`, `RATIONAL_MUL_CONV`, ...) using the well-know mechanism of higher-order conversionals. For instance, the computation

$$\left(1 + 2\mathbf{i} - \frac{1}{2}\mathbf{k}\right)^3 = -\frac{47}{4} - \frac{5}{2}\mathbf{i} + \frac{5}{8}\mathbf{k}$$

is performed with the command

```
# RATIONAL_QUAT_CONV
  `(Hx(&1) + Hx(&2) * ii - Hx(&1 / &2) * kk) pow 3`;;
val it : thm =
  |- (Hx(&1) + Hx(&2) * ii - Hx(&1 / &2) * kk) pow 3 =
     -- Hx(&47 / &4) - Hx(&5 / &2) * ii + Hx(&5 / &8) * kk
```

Finally, we implement a procedure for simplifying quaternionic polynomial expressions. HOL Light provides a general procedure for polynomial normalization, which unfortunately works only for commutative rings. Hence we are forced to code our own solution. In principle, our procedure can be generalized to work with arbitrary (non-commutative) rings. However, at the present stage, it is hardwired to the specific case of quaternions. To give an example, the computation

$$(p+q)^3 = p^3 + q^3 + pq^2 + p^2q + pqp + qp^2 + qpq + q^2p$$

can be done with the command

```
# QUAT_POLY_CONV `(x + y) pow 3`;;
val it : thm =
  |- (p + q) pow 3 =
     p pow 3 + q pow 3 + p * q pow 2 + p pow 2 * q +
     p * q * p + q * p pow 2 + q * p * q + q pow 2 * p
```

4 The geometry of quaternions

One well-know and simple fact, which makes quaternions useful in several physical and geometrical applications, is that the quaternionic product encodes both the scalar and the vector product. More precisely, if q_1 and q_2 are purely imaginary quaternions then we have

$$q_1 q_2 = -\underbrace{\langle q_1, q_2\rangle}_{\substack{\text{scalar}\\\text{product}}} + \underbrace{q_1 \wedge q_2}_{\substack{\text{vector}\\\text{product}}} \in \mathbb{R} + \mathbb{I}$$

which can be easily verified by direct computation.

Moreover, it is possible to use quaternions to encode orthogonal transformations. We briefly recall the essential mathematical construction. For $q \neq 0$, define the conjugation map

$$\begin{aligned} c_q &: \mathbb{H} \longrightarrow \mathbb{H} \\ c_q(x) &:= q^{-1}\, x\, q \end{aligned}$$

An elementary remark is that the product of quaternions corresponds to the composition of the respective conjugation map:

$$c_{q_1} \circ c_{q_2} = c_{q_1 q_2}$$

One important special case is when q unitary, i.e., $\|q\| = 1$ for which we have $q^{-1} = \bar{q}$ and thus

$$c_q(x) = \bar{q}\, x\, q$$

Now, we are ready to state some basic results, which we have formalized in our framework.

Proposition 1. *If $q^2 = -1$ and $-c_q$ is the reflection w.r.t. $q^\perp$.*

Here is the corresponding statement proved in HOL Light

```
REFLECT_ALONG_EQ_QUAT_CONJUGATION
|- !v. ~(v = vec 0)
       ==> reflect_along v = \x. --HIm (inv (Hv v) * Hv x * Hv v)
```

The theorem of Cartan-Dieudonné asserts that any orthogonal transformation $f\colon \mathbb{R}^n \longrightarrow \mathbb{R}^n$ is the composition of at most n reflections. Using this and the previous proposition we get the following result.

Proposition 2. *Any orthogonal transformation $f : \mathbb{R}^3 \longrightarrow \mathbb{R}^3$ is of the form*

$$f = c_q \quad \textit{or} \quad f = -c_q, \qquad \|q\| = 1.$$

The corresponding formalization is the following

```
ORTHOGONAL_TRANSFORMATION_AS_QUAT_CONJUGATION
|- !f. orthogonal_transformation f
         ==> (?q. norm q = &1 /\
                  ((!x. f x = HIm (inv q * Hv x * q)) \/
                   (!x. f x = --HIm (inv q * Hv x * q))))
```

5 Quaternionic Analysis

Passing from algebra to analysis, we need to prove a series of technical results about the analytical behaviour of the algebraic operations. To give an idea, here we report a statement about the uniform continuity of the quaternionic inverse $q \mapsto q^{-1}$.

```
UNIFORM_LIM_QUAT_INV
|- !net P f l b.
     (!e. &0 < e
          ==> eventually (\x. !n. P n ==> norm (f n x - l n) < e)
                         net) /\
     &0 < b /\
     eventually (\x. !n. P n ==> b <= norm (l n)) net
     ==> (!e. &0 < e
              ==> eventually
                  (\x. !n. P n
                           ==> norm (inv (f n x) - inv (l n)) < e)
                  net)
```

To explain the precise meaning of this statement, we should give more details about the net topology implemented in HOL Light, which cannot be done here.

We conducted a systematic formalization of behaviour of algebraic operation from the point of view of limits and continuity, which brought us to prove more than fifty such theorems overall. Some of them are indeed trivial. For instance, the uniform continuity of the product is a trivial consequence of a more general results already available on bilinear maps. Some are less immediate and forced us to dive into a technical $\epsilon\delta$-reasoning.

Next, we considered the differential structure. Given a function $f : \mathbb{R}^n \to \mathbb{R}^m$ we denote by $\mathrm{D}f_{x_0}(v)$ or $\frac{\mathrm{d}}{\mathrm{d}x} f(x)|_{x_0}(v)$ the (Frechét) differential of f in x_0 applied to the vector v. When the differential exists, it is the linear function from $\mathbb{R}^n$ to $\mathbb{R}^m$ that "best" approximates the variation of f in a neighborhood of x_0, i.e.,

$$f(x) - f(x_0) \approx \mathrm{D}f_{x_0}(x - x_0).$$

In HOL Light, the ternary predicate `(f has_differential f') (at x0)` is used to asserts that f is differentiable at x_0 and $f' = \mathrm{D}f_{x_0}$

We compute the differential of the basic quaternionic operations. Notice that, if f is a quaternionic valued function, the differential $\mathrm{D}f_{a_0}(x)$ is a quaternion (in the modern language of Differential Geometry this is the natural identification of the tangent space $T_{f(a_0)}\mathbb{H} \simeq \mathbb{H}$).

For instance, given two differentiable functions $f(q)$ and $g(q)$, the differential of their product in q_0 is

$$\frac{\mathrm{d}\big(f(q)g(q)\big)}{\mathrm{d}q}|_{q_0}(x) = f(q_0)\,\mathrm{D}g_{q_0}(x) + \mathrm{D}f_{q_0}(x)g(q_0).$$

In our formalism, the previous formula becomes the following theorem:

```
QUAT_HAS_DERIVATIVE_MUL_AT
|- !f f' g g' q.
     (f has_derivative f') (at q) /\ (g has_derivative g') (at q)
     ==> ((\x. f x * g x) has_derivative
          (\x. f q * g' x + f' x * g q)) (at q)
```

One consequence that will be useful later, is the following formula for the power:

$$\frac{\mathrm{d}q^n}{\mathrm{d}q}|_{q_0}(x) =$$

that is, the HOL theorem

```
QUAT_HAS_DERIVATIVE_POW
|- !q0 n.
     ((\q. q pow n) has_derivative
      (\h. vsum (1..n) (\i. q0 pow (n - i) * h * q0 pow (i - 1))))
     (at q0)
```

which is easily shown by induction using the differential of the product.

Finally, a straightforward but important observation for the next section is the following. Let $R_p\colon \mathbb{H} \to \mathbb{H}$ be the right multiplication by the quaternion p, i.e., $R_p(x) = xp$. Since R_p is $\mathbb{R}$-linear, we have $\mathrm{D}R_p = R_p$, that is

```
|- !net p. ((\q. q * p) has_derivative (\q. q * p)) net
```

6 Slice-regular functions

7 Pythagorean-Hodograph curves

Pythagorean-hodograph (PH) curves provide significant computational advantages for computer-aided design (CAD) and robotics applications since, among other things, their arc length can be computed precisely, i.e., without numerical quadrature, and their offsets are rational curves.

Planar and spatial Pythagorean–hodograph curves are characterized by different approaches since Pythagorean polynomial triples and quadruples involve disparate algebraic structures. An algebraic model for planar PH curves is based on the properties of the complex numbers while spatial PH curves can be described by quaternions. In our work we deal with spatial PH curves using our HOL Light formalization of the quaternion algebra.

7.1 Basic definition and the spatial Hermite interpolation problem

A parametric polynomial curve $\mathbf{r}(t) = (x(t), y(t), z(t))$ in $\mathbb{R}^3$ is PH if and only if its hodograph satisfies the pithagorean condition, i.e., exists a polynomial $\sigma(t)$ such that

$$|\mathbf{r}'(t)|^2 = x^2(t) + y^2(t) + z^2(t) = \sigma^2(t) \tag{1}$$

In other words a curve $\mathbf{r}(t)$ is PH if and only if it is polynomial and the norm of its hodograph $|\mathbf{r}'(t)|$ is also polynomial. So, the formal definition in HOL Light is the following.

```
let is_ph_curve = new_definition
 `is_ph_curve r <=>
  vector_polynomial_function r /\
  real_polynomial_function
   (\t. norm (vector_derivative r (at t)))`;;
```

Such a curves, in the spatial case, can be succinctly expressed in terms of the algebra of quaternions in fact, it is well know [3] that, regarding $\mathbf{r}(t) = x(t)\mathbf{i} + y(t)\mathbf{j}z(t)\mathbf{k}$ as a pure vector in $\mathbb{H}$, condition 1 holds if and only if exists a quaternionic polynomial $A(t)$ such that

$$\mathbf{r}'(t) = A(t)\mathbf{u}\bar{A}(t) \tag{2}$$

where $\mathbf{u}$ is any fixed unit vector and $\bar{A}(t)$ is the usual quaternionic conjugate of $A(t)$. We prove formally the sufficient condition for a spatial curve to be PH involving quaternions.

```
QUAT_PH_CURVE : thm =
  |- !r A u.
         (!t. r differentiable at t) /\
         u pow 2 = --Hx (&1) /\
         vector_polynomial_function A /\
         (!t. vector_derivative r (at t) = A t * u * cnj (A t))
         ==> is_ph_curve r
```

In order to work always inside the type `` `:quat` ``[3] we consider vectors in $\mathbb{R}^3$ (pure vectors) as quaternions with real part equal to zero. Unit vectors are, from this point of view, quaternions such that its square is equal to minus one.

Note also that we have to take as hypothesis that the curve is differentiable at every point even if we know that its hodograph is polynomial. It happens because HOL Light admits only total function so the derivative of non-differentiable functions is also defined but, in this case, it is an unknown object so we can't work with it. The other implication of the previous theorem, even if is true, is more difficult to formalize because involves formal properties of polynomials (as division and factorization) that are very hard to implement.

Specifying $A(t)$ in the Bernstein form

$$A(t) = \sum_{i=0}^{m} A_i b_i^m(t) \tag{3}$$

with $A_i \in \mathbb{H}$ and $b_i^m(t) = \binom{m}{i}(1-t)^{m-i}t^i$ we obtain PH cubics or PH quintics if $A(t)$ has degree 1 or 2, i.e. if it is of the following forms respectively.

-
$$A(t) = A_0(1-t) + A_1 t \tag{4}$$
-
$$A(t) = A_0(1-t)^2 + A_1(1-t)t + A_2 t^2 \tag{5}$$

Since PH curves have many good properties, from a computational and applicative point of view, interpolation of first-order spatial Hermite data by PH cubics or quintics is a very common problem. The question is: given the initial and final point $\{\mathbf{p}_i, \mathbf{p}_f\}$ and derivatives $\{\mathbf{d}_i, \mathbf{d}_f\}$, can we found a PH cubic or a PH quintic that interpolates this data set? In other words, can we found a quaternionic polynomial $A(t)$ of the form 4 or 5 such that the PH curve defined by the hodograph 2 interpolates the given data set?

7.2 Solutions of the equation $A\mathbf{u}\bar{A} = \mathbf{d}$

The computation of interpolant PH cubic or quntic, of a given data set, involves in both cases equation of the form

$$A\mathbf{u}\bar{A} = \mathbf{d} \tag{6}$$

[3] HOL Light doesn't have subtypes so we can't consider spatial vectors as quaternions because an element can't be of different types simultaneously.

where $\mathbf{u}$ is a unit vector and $\mathbf{d}$ is any non-zero vector not aligned with $-\mathbf{u}$. It turns out that the latter has no a unique solution but involves a one-parameter family of solution (see [2], Section 2). The general form of these solutions is

$$A = \sqrt{|\mathbf{d}|}\mathbf{n}\exp(\phi\mathbf{u}) \tag{7}$$

where $\exp(\phi\mathbf{u}) = \cos(\phi)+\sin(\phi)\mathbf{u}$, $\mathbf{n} = \frac{\mathbf{u}+\frac{\mathbf{d}}{|\mathbf{d}|}}{|\mathbf{u}+\frac{\mathbf{d}}{|\mathbf{d}|}|}$ and ϕ is a free angular parameter. We prove formally a sufficient condition for a quaternion to be solution of 7.

```
QUAT_ROTATION_QUAT_SOLUTIONS : thm =
  |- !u d A t.
         u pow 2 = --Hx (&1) /\
         (!a. ~(u = Hx a * d)) /\
         ~(d = Hx (&0)) /\
         Re d = &0 /\
         A = Hx (sqrt (norm d)) *
             inv (norm (u + inv (norm d) % d)) %
             (u + inv (norm d) % d) *
             (Hx (cos t) + Hx (sin t) * u)
         ==> A * u * cnj A = d
```

Note that the universal quantification over `t:real` gives the expected one-parameter family of solutions.

7.3 PH cubic interpolant

It is know that, given a data set $\{\mathbf{p}_i, \mathbf{p}_f, \mathbf{d}_i, \mathbf{d}_f\}$, the ordinary cubic interpolant is expressed, in the Bézier form, as

$$\mathbf{r}(t) = \mathbf{p}_i + b_0^3(t) + (\mathbf{p}_i + \frac{1}{3}\mathbf{d}_i)b_1^3(t) + (\mathbf{p}_f - \frac{1}{3}\mathbf{d}_f)b_2^3(t) + \mathbf{p}_f b_3^3(t) \tag{8}$$

and its hodograph is

$$\mathbf{r}'(t) = \mathbf{d}_i b_0^2(t) + \mathbf{w}b_1^2(t) + \mathbf{d}_f b_2^2(t) \tag{9}$$

with $\mathbf{w} = 3(\mathbf{p}_f - \mathbf{p}_i) - (\mathbf{d}_i + \mathbf{d}_f)$.

Since a cubic curve is a PH curve if and only if its hodograph is of the form $\mathbf{r}'(t) = A(t)\mathbf{u}\bar{A}(t)$ for some quaternionic polynomial $A(t)$ of the form 4, i.e. in the Bézier form

$$\mathbf{r}'(t) = A_0\mathbf{u}\bar{A}_0 b_0^2(t) + \frac{1}{2}(A_0\mathbf{u}\bar{A}_1 + A_1\mathbf{u}\bar{A}_0)b_1^2(t) + A_1\mathbf{u}\bar{A}_1 b_2^2(t) \tag{10}$$

we have that 8 is PH if the above hodograph agrees with 10. It turns out that this happens if the following conditions hold (see [2], Section 5):

$$\mathbf{w}\cdot(\delta_i - \delta_f) = 0 \tag{11}$$

and

$$\left(\mathbf{w} \cdot \frac{\delta_i + \delta_f}{|\delta_i + \delta_f|}\right)^2 + \frac{(\mathbf{w} \cdot \mathbf{z})^2}{|\mathbf{z}|^4} = |\mathbf{d}_i||\mathbf{d}_f| \tag{12}$$

with $\delta_i = \frac{\mathbf{d}_i}{|\mathbf{d}_i|}$, $\delta_f = \frac{\mathbf{d}_f}{|\mathbf{d}_f|}$ and $\mathbf{z} = \frac{\delta_i \times \delta_f}{|\delta_i \times \delta_f|}$. The above conditions 11 and 12 are therefore sufficient for the ordinary Hermite interpolant cubic to be PH so we have the following formal theorem.

```
PH_INTER_CUBIC
|- !Pf Pi di df:real^4.
     let w = Hx(&3) * (Pf - Pi) - (di + df) in
     let n = \v. Hx(inv(norm v)) * v in
     let z = Hx(inv (norm (n di + n df))) *
             Hv(HIm(n di) cross HIm(n df)) in
     let r = \t. bernstein 3 0 (drop t) % Pi +
                 bernstein 3 1 (drop t) % (Pi + Hx(&1 / &3) * di) +
                 bernstein 3 2 (drop t) % (Pf - Hx(&1 / &3) * df) +
                 bernstein 3 3 (drop t) % Pf in
     Re Pf = &0 /\ Re Pi = &0 /\ Re di = &0 /\ Re df = &0 /\
     ~(Hx(&0) = di) /\ ~(Hx(&0) = df) /\ (!a. ~(n di = Hx a * df))
     ==>
     pathstart r = Pi /\ pathfinish r = Pf /\
     pathstart (\t. vector_derivative r (at t)) = di /\
     pathfinish (\t. vector_derivative r (at t)) = df /\
     (w dot (n di - n df) = &0 /\
      (w dot (n (n di + n df))) pow 2 +
      inv(norm z) pow 4 * (w dot z) pow 2 =
      norm di * norm df
      ==> is_ph_curve r)`
```

7.4 PH quintic interpolant

As regard PH quintics things are very different. Integration of the hodograph $\mathbf{r}'(t) = A(t)\mathbf{u}\bar{A}(t)$ gives a generic PH quintic, in the Bézier form,

$$\mathbf{r}(t) = \sum_{i=0}^{5} \mathbf{p}_i b_i^5(t) \tag{13}$$

with control points

- $\mathbf{p}_1 = \mathbf{p}_0 + \frac{1}{5} A_0 \mathbf{u} \bar{A}_0$
- $\mathbf{p}_2 = \mathbf{p}_1 + \frac{1}{10}(A_0 \mathbf{u} \bar{A}_1 + A_1 \mathbf{u} \bar{A}_0)$
- $\mathbf{p}_3 = \mathbf{p}_2 + \frac{1}{30}(A_0 \mathbf{u} \bar{A}_2 + A_1 \mathbf{u} \bar{A}_1 + A_2 \mathbf{u} \bar{A}_0)$
- $\mathbf{p}_4 = \mathbf{p}_3 + \frac{1}{10}(A_1 \mathbf{u} \bar{A}_2 + A_2 \mathbf{u} \bar{A}_1)$
- $\mathbf{p}_5 = \mathbf{p}_4 + \frac{1}{5} A_2 \mathbf{u} \bar{A}_2.$

and, unlike cubics, the right choice of the coefficients of $A(t)$ gives always an Hermite interpolant. More precisely we obtain particular a two-parameter family of interpolants (see [2], Section 6). Unfortunately, for reasons of space, we can't show the explicit expression of A_0, A_1 and A_2.

So, also in this case, we prove formally a sufficient condition for a generic PH quintic to be interpolant of a given data set but, unfortunately for reasons of space, we can't report the theorem (only the statement involves about 40 code lines while the proof about 400 code lines) but the all code is available online at the link $\cdots$. However the proof is essentially an algebraic manipulation on quaternions so our framework has been very useful to automate many calculations that was implicit also in the informal proof. Interesting further developments are about polynomials, as syntactic objects, in order to prove formally the reverse of the theorem 2 so to use quaternions to characterize completely spatial PH curves.

8 Conclusions

References

1. Ciolli, G., Gentili, G., Maggesi, M.: A certified proof of the cartan fixed point theorems. Journal of Automated Reasoning 47(3), 319–336 (2011)
2. Farouki, R.T., Giannelli, C., Manni, C., Sestini, A.: Identification of spatial ph quintic hermite interpolants with near-optimal shape measures. Computer Aided Geometric Design 25(4), 274 – 297 (2008)
3. Farouki, R.: Pythagorean-Hodograph Curves: Algebra and Geometry Inseparable. Geometry and Computing, Springer Berlin Heidelberg (2009)
4. Fuchs, L., Théry, L.: Implementing geometric algebra products with binary trees. Advances in Applied Clifford Algebras 24(2), 589–611 (2014)
5. Gentili, G., Stoppato, C., Struppa, D.: Regular Functions of a Quaternionic Variable. Springer Monographs in Mathematics, Springer Berlin Heidelberg (2013)
6. Harrison, J.: A HOL theory of Euclidean space. In: Hurd, J., Melham, T. (eds.) Theorem Proving in Higher Order Logics, 18th International Conference, TPHOLs 2005. Lecture Notes in Computer Science, vol. 3603, pp. 114–129. Springer-Verlag, Oxford, UK (August 2005)
7. Harrison, J.: Formalizing basic complex analysis. In: Matuszewski, R., Zalewska, A. (eds.) From Insight to Proof: Festschrift in Honour of Andrzej Trybulec. Studies in Logic, Grammar and Rhetoric, vol. 10(23), pp. 151–165. University of Białystok (2007), http://mizar.org/trybulec65/
8. Ma, S., Shi, Z., Shao, Z., Guan, Y., Li, L., Li, Y.: Higher-order logic formalization of conformal geometric algebra and its application in verifying a robotic manipulation algorithm. Advances in Applied Clifford Algebras 26(4), 1305–1330 (2016)