On Numerical Aspects of Pseudo-Complex Powers in $\mathbb{R}^3$

Carla Cruz[a] M.I. Falcão[b] H.R. Malonek[a]

[a]Departamento de Matemática and CIDMA, Universidade de Aveiro, Portugal
[b]Departamento de Matemática e Aplicações and CMAT, Universidade do Minho, Portugal

Abstract

In this paper we consider a particularly important case of 3D monogenic polynomials that are isomorphic to the integer powers of one complex variable (called *pseudo-complex powers* or *pseudo-complex polynomials*, PCP). The construction of bases for spaces of monogenic polynomials in the framework of Clifford Analysis has been discussed by several authors and from different points of view. Here our main concern are numerical aspects of the implementation of PCP as bases of monogenic polynomials of homogeneous degree k. The representation of the well known Fueter polynomial basis by a particular PCP-basis is subject to a detailed analysis for showing the numerical efficiency of the use of PCP. In this context a modification of the Eisinberg-Fedele algorithm for inverting a Vandermonde matrix is presented.

Information

Keywords:
Pseudo-complex powers, monogenic polynomials, Vandermonde matrix.

Original publication:
Lecture Notes in Computer Science **8579**, (2014), 1-16
DOI: 10.1007/978-3-319-09144-0_1
http://link.springer.com

1 Introduction

To understand the meaning and significance of *pseudo-complex powers* in higher dimensional Euclidean spaces, we briefly explain some steps from the recent past that led us to this subject.

The authors initiated recently ([7, 8, 9, 21]) the use of PCP as building blocks for Clifford-valued homogeneous polynomials in the Clifford Algebra $C\ell_{0,n}$. The main differences to other approaches (cf. [20]) consist in fixing the PCP-basis of monogenic polynomials of homogeneous degree k by using a recursively (with respect to k) defined algorithm for the choice of a parameter set among all vectors in $\mathcal{S}^n$. This means, that in the 3D-case the parameter set is related to the primitive roots of unity and can be visualized in the complex plane. Therefore PCP opened the way to some new insights into the connection of Hypercomplex Function Theory and Number Theory.

We stressed in [7], by presenting two examples of different choices of the parameter set, also some computational advantages of 3D-PCP in comparison with the generalized powers generated by the well known Fueter variables. Among other problems, we discussed in [9] the applications of PCP to some problems of combinatorial nature. In [21] we studied their role in 3D Appell systems and extended the obtained results to the 4D quaternionic case in [8].

All studies were mainly motivated by two observations. The fact that, apart from the concrete parameter choice, the PCP obey for a fixed homogeneous degree k all the same structure, isomorphic to the structure of z^k of the complex variable z. This leads immediately to essentially reduced numerical costs for their construction. From the other side, it led us to interesting studies about different parameter choices and its closed connection with the primitive roots of unity as mentioned before. Moreover, the freedom of the parameter choice implies

that PCP can play an important role for the application of bijective methods from enumerative and algebraic combinatorics (see [9]).

The goal of the present paper is now a detailed discussion of different numerical aspects of 3D-PCP. After introducing in Sect. 2 the necessary basic notations and results of Hypercomplex Function Theory needed for use in the subsequent sections, we revisit in Sect. 3 the theory of PCP by presenting and proving some of their properties. In Sect. 4 we propose an efficient algorithm to obtain PCP which takes advantages of their relation to the integer powers of one complex variable. This means that the algorithm uses, in its most demanding steps, only complex arithmetic. A modification of the Eisinberg-Fedele algorithm for inverting a Vandermonde matrix is presented in Sect. 5, in order to obtain an efficient procedure to derive the series expansion of any reduced quaternion valued monogenic function in terms of PCP.

2 Basic Notations and Results

Let $\{1, \boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3\}$ be an orthonormal basis of the Euclidean vector space $\mathbb{R}^4$ with the (quaternionic) product given according to the multiplication rules

$$\boldsymbol{e}_1^2 = \boldsymbol{e}_2^2 = \boldsymbol{e}_3^2 = -1, \;\; \boldsymbol{e}_1\boldsymbol{e}_2 = -\boldsymbol{e}_2\boldsymbol{e}_1 = \boldsymbol{e}_3.$$

This non-commutative product generates the well known algebra of real quaternions $\mathbb{H}$. For a quaternion of the form $x = x_0 + \boldsymbol{e}_1 x_1 + \boldsymbol{e}_2 x_2 + \boldsymbol{e}_3 x_3$, the conjugate of x is $\bar{x} = x_0 - \boldsymbol{e}_1 x_1 - \boldsymbol{e}_2 x_2 - \boldsymbol{e}_3 x_3$, the scalar part of x is $\mathrm{Sc}\, x := x_0 = \frac{1}{2}(x + \bar{x})$ and the vector part of x is $\underline{x} := x_1\boldsymbol{e}_1 + x_2\boldsymbol{e}_2 + x_3\boldsymbol{e}_3 = \frac{1}{2}(x - \bar{x})$. The norm $|x|$ of x is defined by $|x|^2 = x\bar{x} = \bar{x}x = x_0^2 + x_1^2 + x_2^2 + x_3^2$ and it immediately follows that each non-zero $x \in \mathbb{H}$ has an inverse given by $x^{-1} = \frac{\bar{x}}{|x|^2}$.

Considering the subset

$$\mathcal{A} := \mathrm{span}_{\mathbb{R}}\{1, \boldsymbol{e}_1, \boldsymbol{e}_2\} \subset \mathbb{H},$$

the real vector space $\mathbb{R}^3$ can be embedded in $\mathcal{A}$ by the identification of each element $\boldsymbol{x} = (x_0, x_1, x_2) \in \mathbb{R}^3$ with the so-called *paravector* or *reduced quaternion* $x = x_0 + x_1\boldsymbol{e}_1 + x_2\boldsymbol{e}_2 \in \mathcal{A}$.

In what follows we consider $\mathbb{H}$-valued functions f defined in some open subset $\Omega \subset \mathbb{R}^3$, which are continuously real differentiable.

To call attention to its relation to the complex Wirtinger derivatives, we use for a generalized Cauchy-Riemann operator in $\mathbb{R}^3$ the notation

$$\overline{\partial} := \frac{1}{2}(\partial_0 + \partial_{\underline{x}}), \quad \partial_0 := \frac{\partial}{\partial x_0}, \quad \partial_{\underline{x}} := \boldsymbol{e}_1\frac{\partial}{\partial x_1} + \boldsymbol{e}_2\frac{\partial}{\partial x_2}.$$

Functions f satisfying the equation $\overline{\partial} f = 0$ (resp. $f\overline{\partial} = 0$) are called *left monogenic* (resp. *right monogenic*). We suppose that f is hypercomplex differentiable in Ω in the sense of [15] and [20], i.e. has a uniquely defined areolar derivative f' in each point of Ω. Then f is real differentiable and f' can be expressed by real partial derivatives as $f' = \partial f$ where analogously to the generalized Cauchy-Riemann operator we use $\partial := \frac{1}{2}(\partial_0 - \partial_{\underline{x}})$ for the conjugate Cauchy-Riemann operator. Since a hypercomplex differentiable function belongs to the kernel of $\overline{\partial}$, it follows that in fact $f' = \partial_0 f = -\partial_{\underline{x}} f$ similar to the complex case.

A second structure of $\mathbb{R}^3$ different from that given by the set of paravectors $\mathcal{A}$ consists in the following isomorphism:

$$\mathbb{R}^3 \cong \mathcal{H}^2 = \{\vec{z} = (z_1, z_2) : z_k = x_k - x_0\boldsymbol{e}_k, \; x_0, x_k \in \mathbb{R}; \; k = 1, 2\},$$

where the variables z_1 and z_2 are reduced quaternions, usually called *Fueter variables*. For the next step we apply

Definition 1 *For $a_k \in \mathbb{H}$ $(k = 1, \ldots, n)$, the symmetric $\times$-product is defined by*

$$a_1 \times a_2 \times \cdots \times a_n = \frac{1}{n!} \sum_{\pi(i_1, \ldots, i_n)} a_{i_1} a_{i_2} \ldots a_{i_n} \tag{1}$$

where the sum runs over **all** *permutations of $(i_1, \ldots, i_n)$*

together with the following convention: if the factor a_j occurs μ_j-times in (1), we briefly write

$$\underbrace{a_1 \times \cdots \times a_1}_{\mu_1} \times \cdots \times \underbrace{a_n \times \cdots \times a_n}_{\mu_n} = {a_1}^{\mu_1} \times \cdots \times {a_n}^{\mu_n} = \vec{a}^{\,\mu} \tag{2}$$

where $\mu = (\mu_1, \ldots, \mu_n)$, and set parentheses if the powers are understood in the ordinary way (see [20]).

It can be proved that the symmetric products $z_1^{\mu_1} \times z_2^{\mu_2}$ are monogenic functions of homogeneous degree $\mu_1 + \mu_2$ which take their values in $\mathcal{A}$ and are $\mathbb{H}$-linearly independent (see [19, 20]). We list now several other essential and well known properties of the so-called *generalized powers* (GP) of degree k.

Theorem 1 ([4, 19]) *The generalized powers $\mathcal{G}_s^k := {z_1}^{k-s} \times {z_2}^s,\ s = 0, \ldots, k$, form a basis of the $\mathbb{H}$-linear space of homogeneous monogenic $\mathbb{H}$-valued polynomials of degree k.*

Theorem 2 ([4, 19]) *If $f : \Omega \subset \mathbb{R}^3 \to \mathbb{H}$ is a monogenic function, then the general form of the Taylor series of f in the neighborhood of the origin is given by*

$$f(z_1, z_2) = \sum_{k=0}^{\infty} \frac{1}{k!} \sum_{s=0}^{k} \binom{k}{s} \frac{\partial^k f(0)}{\partial x_1^{k-s} \partial x_2^s} \mathcal{G}_s^k. \tag{3}$$

As is indicated in [16], the polynomials $\mathcal{G}_s^k$ satisfy the recursion formula

$$\begin{aligned} &\mathcal{G}_0^0 = 1, \\ &\mathcal{G}_s^k = \frac{1}{k} \left\{ (k-s) z_1 \mathcal{G}_s^{k-1} + s\, z_2 \mathcal{G}_{s-1}^{k-1} \right\}, \quad s = 0, \ldots, k. \end{aligned} \tag{4}$$

and applying convention (2), the following binomial formula (see, e.g. [20]) can be derived:

$$(\alpha z_1 + \beta z_2)^k = \sum_{s=0}^{k} \binom{k}{s} \alpha^{k-s} \beta^s {z_1}^{k-s} \times z_2^s, \quad \text{with} \quad \alpha, \beta \in \mathbb{R}. \tag{5}$$

3 Pseudo-Complex Powers

The construction of bases for spaces of monogenic polynomials in the framework of Clifford Analysis has been discussed by several authors and from different points of view. During the last decade, numerous papers with emphasis on theoretical questions like their orthogonality or their relationship to Appell systems were published [1, 3, 10, 13, 14, 18].

Here our main concern are numerical aspects of the implementation of special bases of monogenic polynomials of homogeneous degree k. Concretely, in this section we are going to study a set of polynomials of the form

$$\mathcal{Z}_s^k(x) = (x_0 + y_s \boldsymbol{i}_s)^k, \tag{6}$$

where

$$y_s = \alpha_s x_1 + \beta_s x_2 \tag{7}$$

and

$$\boldsymbol{i}_s = \alpha_s \boldsymbol{e}_1 + \beta_s \boldsymbol{e}_2 \tag{8}$$

for the particular choice

$$\beta_s = (s+1)\alpha_s = \frac{s+1}{\sqrt{1 + (s+1)^2}};\ s = 0, \ldots, k. \tag{9}$$

We observe that since $\alpha_s^2 + \beta_s^2 = 1$, it follows that $\boldsymbol{i}_s^2 = -1$ and therefore one can prove that the above polynomials are monogenic polynomials isomorphic to the complex powers. To underline this fact we call such polynomials *pseudo-complex powers* or *pseudo-complex polynomials* (PCP).

In the recent past, pseudo-complex powers have been considered by several authors in different contexts. In [5] it was proved that homogeneous monogenic polynomials of the form

$$H_{(a_i,b_i)}^k = (a_i z_1 + b_i z_2)^k, a_i, b_i \in \mathbb{R},\ i = 0, \ldots, k$$

form a basis of the $\mathbb{H}$-linear space of homogeneous monogenic $\mathbb{H}$-valued polynomials of degree k if and only if $a_i^2+b_i^2=1$ and $a_ib_j-a_jb_i\neq 0,\ i,j=0,...,k$. In [6], a complete set of pseudo-complex polynomials, having prescribed properties was constructed. Later on, aspects of combinatorial nature were considered in [9] and some computational aspects related to the explicit constructions of PCP were already discussed in [7]. The idea of this paper is to study, in detail, PCP of the form (6)-(9), from a theoretical and computational point of view.

For the sake of completeness and better understanding, we list and prove the main properties of the PCP needed in this work.

Proposition 1 *The PCP (6)-(9) can be written, in terms of the Fueter variables z_1 and z_2, as*

$$\mathcal{Z}_s^k(x)=(\alpha_s z_1+\beta_s z_2)^k(\boldsymbol{i}_s)^k. \tag{10}$$

Proof: Since $\mathcal{Z}_s^k=(\mathcal{Z}_s^1)^k$ and

$$\begin{aligned}\mathcal{Z}_s^1=(\alpha_s z_1+\beta_s z_2)\boldsymbol{i}_s&=\big(\alpha_s(x_1-x_0\boldsymbol{e}_1)+\beta_s(x_2-x_0\boldsymbol{e}_2)\big)\boldsymbol{i}_s\\&=(x_0(-\boldsymbol{i}_s)+x_1\alpha_s+x_2\beta_s)\boldsymbol{i}_s=x_0+(x_1\alpha_s+x_2\beta_s)\boldsymbol{i}_s\end{aligned}$$

the result follows at once. □

Remark 1 *Last property reveals that $\mathcal{Z}_s^k=H^k_{(\alpha_s,\beta_s)}(\boldsymbol{i}_s)^k$ and therefore one can conclude that the set $\{\mathcal{Z}_s^k\}_{s=0}^k$ is also a basis of the $\mathbb{H}$-linear space of homogeneous monogenic $\mathbb{H}$-valued polynomials of degree k.*

Consider now the vectors $\mathcal{Z}$ and $\mathcal{G}$ containing the $k+1$ PCP and GP of degree k, respectively, i.e.

$$\mathcal{Z}=\begin{pmatrix}\mathcal{Z}_0^k\\ \mathcal{Z}_1^k\\ \vdots\\ \mathcal{Z}_k^k\end{pmatrix}\qquad\text{and}\qquad\mathcal{G}=\begin{pmatrix}\mathcal{G}_0^k\\ \mathcal{G}_1^k\\ \vdots\\ \mathcal{G}_k^k\end{pmatrix}.$$

Proposition 2 *The polynomials $\mathcal{Z}_s^k$ and $\mathcal{G}_s^k$ are linked by the relation*

$$\mathcal{Z}=D_1V(1,2,\ldots,k+1)D_2\mathcal{G},$$

where D_1 and D_2 are the diagonal matrices of order $k+1$

$$D_1=\operatorname{diag}\left(\alpha_0^k\boldsymbol{i}_0^k,\ldots,\alpha_j^k\boldsymbol{i}_j^k,\ldots,\alpha_k^k\boldsymbol{i}_k^k\right),\quad D_2=\operatorname{diag}\left(\tbinom{k}{0},\ldots,\tbinom{k}{j},\ldots,\tbinom{k}{k}\right) \tag{11}$$

and $\mathcal{V}_n:=V(1,2,\ldots,n)$ is the Vandermonde matrix of order n associated with the equidistant nodes $1,2,\ldots,n$, i.e.

$$\mathcal{V}_n=V(1,2,\ldots,n)=\begin{pmatrix}1&1&1&\ldots&1\\ 1&2&2^2&\ldots&2^{n-1}\\ \vdots&\vdots&\vdots&\ddots&\vdots\\ 1&n&n^2&\ldots&n^{n-1}\end{pmatrix}. \tag{12}$$

Proof: The use of Proposition 1 and the binomial formula (5) lead easily to the relation

$$\mathcal{Z}=\begin{pmatrix}\binom{k}{0}\alpha_0^k\boldsymbol{i}_0^k&\binom{k}{1}\alpha_0^{k-1}\beta_0\boldsymbol{i}_0^k&\cdots&\binom{k}{k}\beta_0^k\boldsymbol{i}_0^k\\ \binom{k}{0}\alpha_1^k\boldsymbol{i}_1^k&\binom{k}{1}\alpha_1^{k-1}\beta_1\boldsymbol{i}_1^k&\cdots&\binom{k}{k}\beta_1^k\boldsymbol{i}_1^k\\ \vdots&\vdots&\ddots&\vdots\\ \binom{k}{0}\alpha_k^k\boldsymbol{i}_k^k&\binom{k}{1}\alpha_k^{k-1}\beta_k\boldsymbol{i}_k^k&\cdots&\binom{k}{k}\beta_k^k\boldsymbol{i}_k^k\end{pmatrix}\mathcal{G}$$

or equivalently

$$\mathcal{Z} = \begin{pmatrix} \boldsymbol{i}_0^k & & & \\ & \boldsymbol{i}_1^k & & \\ & & \ddots & \\ & & & \boldsymbol{i}_k^k \end{pmatrix} \begin{pmatrix} \alpha_0^k & \alpha_0^{k-1}\beta_0 & \cdots & \beta_0^k \\ \alpha_1^k & \alpha_1^{k-1}\beta_1 & \cdots & \beta_1^k \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_k^k & \alpha_k^{k-1}\beta_k & \cdots & \beta_k^k \end{pmatrix} \begin{pmatrix} \binom{k}{0} & & & \\ & \binom{k}{1} & & \\ & & \ddots & \\ & & & \binom{k}{k} \end{pmatrix} \mathcal{G}.$$

Since $\alpha_s \neq 0,\ s = 0, \ldots, k$ one can write

$$\mathcal{Z} = \begin{pmatrix} \boldsymbol{i}_0^k & & & \\ & \boldsymbol{i}_1^k & & \\ & & \ddots & \\ & & & \boldsymbol{i}_k^k \end{pmatrix} \begin{pmatrix} \alpha_0^k & & & \\ & \alpha_1^k & & \\ & & \ddots & \\ & & & \alpha_k^k \end{pmatrix} \begin{pmatrix} 1 & \frac{\beta_0}{\alpha_0} & \cdots & (\frac{\beta_0}{\alpha_0})^k \\ 1 & \frac{\beta_1}{\alpha_1} & \cdots & (\frac{\beta_1}{\alpha_1})^k \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \frac{\beta_k}{\alpha_k} & \cdots & (\frac{\beta_k}{\alpha_k})^k \end{pmatrix} D_2\mathcal{G}.$$

Thus

$$\mathcal{Z} = D_1 V(\tfrac{\beta_0}{\alpha_0}, \ldots, \tfrac{\beta_k}{\alpha_k}) D_2 \mathcal{G}$$

and the results follows now from $\frac{\beta_s}{\alpha_s} = s + 1$ (cf. (9)). □

4 Numerical Aspects

We focus now on the implementation details of PCP of degree k. Our aim here is to show that, from the computational point of view, these polynomials can be a *good* alternative to GP, since they can be computed by a less time-consuming algorithm. It is clear that the computational time and effort needed to carry out each polynomial depend on numerous parameters, such as the software used, the structure and complexity of the polynomials, the technique/algorithm implemented for their calculation, among others.

The numerical experiments reported in this work were obtained by the use of the package [12] which endow the Mathematica `Quaternions` package with the ability to perform operations on symbolic expressions involving quaternion-valued functions. All simulations have been performed in Mathematica 9.0 (64-bit) on a computer with Intel Xeon E5607 4C 2.26GHz/1066Mhz/8MB processors and 64GB of RAM.

Programming recursively makes it easier to write simple and elegant programs to obtain GP via (4). However, when the degree k increases, computing GP in terms of recursion reveals one of the potential drawbacks of recursive programs. The issue is that for the computation of GP of degree k, the value of some of the GP of lower degree will be needed many times. This means that the time required to compute GP grows exponentially with k.

We point out that PCP can also be computed by means of a recursive formula (see (6)-(9)), namely

$$\mathcal{Z}_s^k = \mathcal{Z}_s^{k-1}\mathcal{Z}_s^1, \quad s = 0, \ldots, k, \qquad \mathcal{Z}_s^0 = 1, \qquad \mathcal{Z}_s^1 = x_0 + y_s \boldsymbol{i}_s. \tag{13}$$

Both recursion solutions (4) and (13) are natural but inefficient, since many identical recursive calls are made during any given calculation, i.e. both problems exhibit *overlapping subproblems*. This issue can be observed in Fig. 1, where the CPU time, in seconds, is presented for both algorithms and for several values of k. These values correspond to the CPU time required to constructing *all* the $k + 1$ linearly independent polynomials of degree k of the form (4) and (13).

Problems having the overlapping subproblems property are almost always solved using *dynamic programming*, a catch-all term for any algorithm in which the definition of a function is extended as the computation proceeds (see [24]). Dynamic programming is a technique for avoiding the repeated computation of the same values in a recursive program. Each value computed is immediately stored. If the value is needed again, it is not computed but simply looked up in memory.

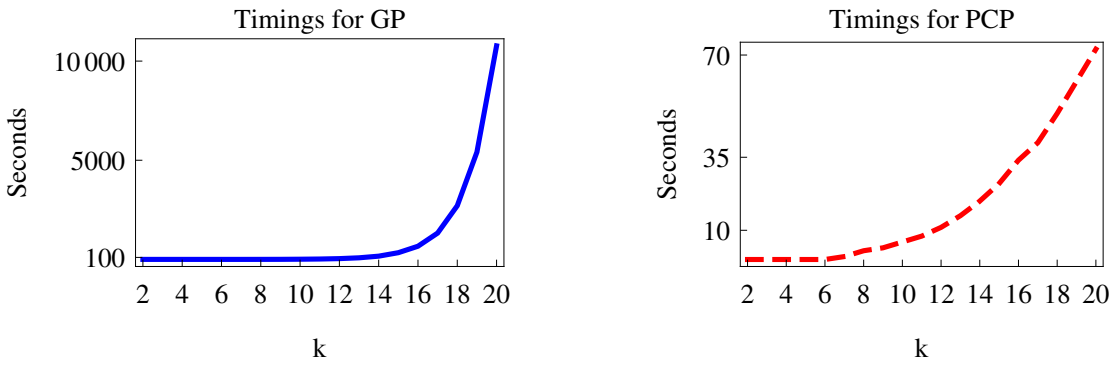

Figure 1: CPU time: GP versus PCP - recursion algorithm ($k = 2, \ldots, 20$).

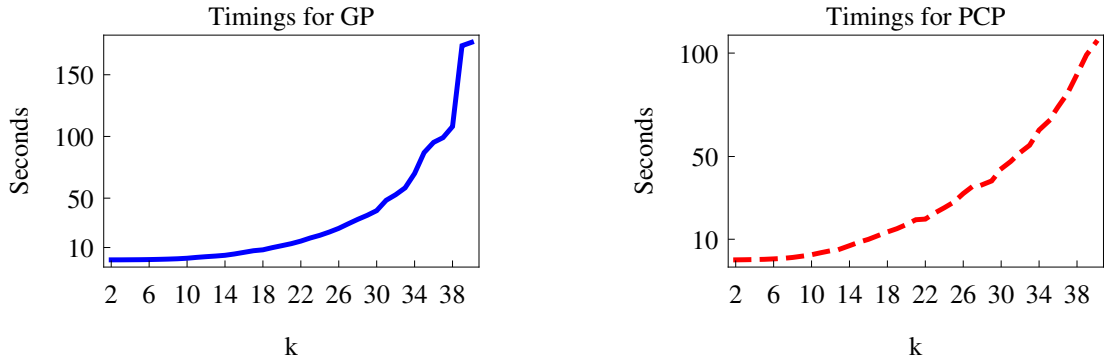

Figure 2: CPU time: GP versus PCP - recursion with result caching ($k = 2, \ldots, 40$).

The ability to add rules to a function as the function executes, makes result caching very easy to implement in Mathematica ([24]). In Fig. 2 we present the costs of implementing (4) and (13) in this system, by using the aforementioned *memorization* technique.

As we can observe, recursion with caching result reduces the overall running time considerably. On the other hand, recursive solutions to problems have the well known serious drawback of using large amount of memory. In addition, this solution does not take any advantage of the simple structure of the PCP. In what follows we propose a non-recursive algorithm to compute PCP which takes into account the isomorphism between PCP and complex powers. In concrete, we obtain the PCP by means of the following procedure:

Algorithm for obtaining PCP using complex arithmetic

1. `Compute, by using complex arithmetic,` $(x_0 + yi)^k, \quad x_0, y \in \mathbb{R}$.
2. `Replace the real parameter` y `by`
$$y \leftarrow \alpha_s x_1 + \beta_s x_2;$$
3. `Replace the imaginary unit` i `by the unit vector`
$$i \leftarrow \alpha_s \boldsymbol{e}_1 + \beta_s \boldsymbol{e}_2; \quad s = 0, 1, \ldots, k.$$

We remark the fact that the most demanding operation is performed in Step 1 and is done in complex arithmetic. As a consequence, the time consumed for obtaining polynomials of a certain degree is significantly

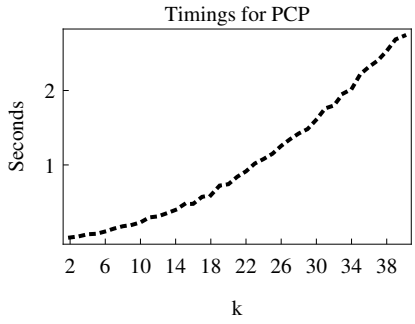

Figure 3: CPU time for PCP - complex arithmetic ($k = 2, \ldots, 40$).

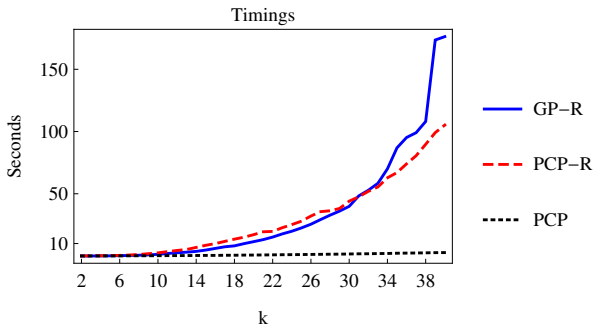

Figure 4: Comparison between GP and PCP with result caching (GP-R and PCP-R) with PCP with complex arithmetic (PCP) ($k = 2, \ldots, 40$).

reduced. Fig. 3 illustrates the performance of the algorithm. Once more, the CPU time concerns the effort needed to computing all the $k+1$ polynomials of a given degree k.

For the sake of better visibility, we compare in Fig. 4 the recursion and complex approach to obtain PCP and the recursion algorithm for computing GP.

5 Applications

In this section we will see how to express each generalized powers $\mathcal{G}_s^k$ as a linear combination of pseudo-complex powers $\mathcal{Z}_0^k,\ \mathcal{Z}_1^k, \ldots, \mathcal{Z}_k^k$. As a consequence it will be possible to obtain, from Theorem 2, the series expansion of any reduced quaternion valued monogenic function in terms of PCP.

Recalling Proposition 2 in Sect. 3, one can write

$$\mathcal{G} = \mathcal{D}_2^{-1} W(1, 2, \ldots, k+1) \mathcal{D}_1^{-1} \mathcal{Z}, \tag{14}$$

where $\mathcal{D}_1$ and $\mathcal{D}_2$ are the diagonal matrices (11) and $W(1, 2, \ldots, k+1) =: \mathcal{W}_{k+1}$ is the inverse of the Vandermonde matrix $\mathcal{V}_{k+1}$ in (12).

It is well known that Vandermonde matrices are often quite ill-conditioned and standard numerically stable methods fail, in general, to compute accurately the entries of their inverses (cf. [11]). For this reason, the inversion of Vandermonde matrices has received so much attention, particularly in the search for fast and accurate algorithms. In this context, we have to mention here the well known Traub [23], Parker [22] and Björck and Pereyra [2] algorithms which make use of the special structure of a Vandermonde matrix to rapidly compute its inverse. More recently, Eisinberg and Fedele in their paper [11], proposed an explicit formula for the inverse of Vandermonde matrices which is particularly useful in the case of equidistant nodes in $[1, n]$, as is the case of $\mathcal{V}_n$ in (12). Their algorithm is based on the following essential result.

Theorem 3 *Let $\mathcal{V}_n$ be a Vandermonde matrix of the form (12) and $\mathcal{W}_n$ be its inverse. The generic element $w_n(i,j)$ of $\mathcal{W}_n$ is*

$$w_n(i,j) = (-1)^{i+j}\Phi(n,j)\sum_{r=0}^{n-i}(-1)^r j^r \sigma(n,n-i-r), \tag{15}$$

where

$$\Phi(n,j) = \frac{1}{(n-j)!(j-1)!} \tag{16}$$

and

$$\sigma(n,s) = \begin{bmatrix} n+1 \\ n+1-s \end{bmatrix}. \tag{17}$$

Here $\begin{bmatrix} a \\ b \end{bmatrix}$ denotes the usual [1] (unsigned) Stirling numbers of the first kind [17].

Proof: The result follows from [11, Theorem 1], using the closed expression of $\sigma(n,s)$ and $\Phi(n,j)$ given by formulas (32) and (33) presented in the same reference. □

The algorithm proposed in [11] which is, in fact, a modification of [22], can now be rewritten as follows.

Eisinberg and Fedele algorithm for inverting $\mathcal{V}_n$

1. `Compute, by means of (16),` $\phi(n,j),\ j=1,2,\ldots,n$;
2. `Compute, by means of (17),` $\sigma(n,s),\ s=0,1,\ldots,n$;
3. `Construct the recursive function`

$$\begin{aligned}
&\psi(1,j) = (-1)^{1+j}\frac{\sigma(n,n)}{j},\ j=1,2,\ldots,n\\
&\psi(2,j) = -\psi(1,j)\sum_{\substack{r=1\\ r\neq j}}^{n}\frac{1}{r},\ j=1,2,\ldots,n\\
&\psi(n,j) = (-1)^{n+j},\ j=1,2,\ldots,n\\
&\psi(i-1,j) = j\psi(i,j) - (-1)^{i+j}\sigma(n,n+1-i),\ i=n,n-1,\ldots,2,\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad j=1,2,\ldots,n;
\end{aligned}$$

4. `Compute the` j`−th column` $[\psi(i,j)\phi(n,j)],\ j=1,2,\ldots,n.$

We point out that the problem of obtaining a series expansion similar to (3) but involving PCP requires the inversion of several Vandermonde matrices $\mathcal{V}_n$, for $n=1,2,\ldots$. This demanding need provided motivation to develop a recursive algorithm to compute all the inverses of $\mathcal{V}_n$ which, as we will see later on, is competitive comparing with the aforementioned Eisinberg and Fedele algorithm. In this context, the following result plays an important role, since it allows to obtain a recursive procedure for computing $\mathcal{W}_n$.

Proposition 3 *The matrix $\mathcal{W}_n = \mathcal{V}_n^{-1}$ has the following structure*

$$\mathcal{W}_n = \left(\begin{array}{c|c} \mathcal{M}_{n-1}\mathcal{W}_{n-1}\mathcal{D}_{n-1} & \boldsymbol{\Sigma}_{n-1} \\ \hline \boldsymbol{\Phi}_{n-1} & \Phi(n,n) \end{array}\right), \tag{18}$$

where

[1] $\begin{bmatrix} a \\ b \end{bmatrix}$ gives the number of permutations of a elements that contain exactly b cycles and are related with the signed Stirling numbers $s(a,b)$ by $s(a,b) = (-1)^{a-b}\begin{bmatrix} a \\ b \end{bmatrix}$.

- $\mathcal{M}_{n-1}$ *is the lower bidiagonal matrix whose elements* m_{ij} *are given by*

$$m_{ij} = \begin{cases} n, & i = j \\ -1, & i = j+1 \end{cases} \qquad i,j = 1,\ldots,n-1,$$

- $\mathcal{D}_{n-1}$ *is the diagonal matrix*

$$\mathcal{D}_{n-1} = \mathrm{diag}(\tfrac{1}{n-1}, \tfrac{1}{n-2}, \ldots, \tfrac{1}{2}, 1),$$

- $\mathbf{\Sigma}_{n-1}$ *is the* $(n-1)-$*column vector whose elements are given by*

$$(-1)^{n+i}\sigma(n-1, n-i)\Phi(n,n),\ i = 1,\ldots,n-1. \tag{19}$$

- $\mathbf{\Phi}_{n-1}$ *is the row vector whose elements are*

$$(-1)^{n+j}\Phi(n,j),\ j = 1,\ldots,n-1. \tag{20}$$

Proof: The well known properties of the Stirling numbers of the first kind (see e.g. [17])

$$\begin{bmatrix} k \\ k \end{bmatrix} = 1, \quad \begin{bmatrix} n \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ n \end{bmatrix} = 0,\ n > 0, \quad \begin{bmatrix} n+1 \\ k \end{bmatrix} = n\begin{bmatrix} n \\ k \end{bmatrix} + \begin{bmatrix} n \\ k-1 \end{bmatrix},$$

allow to derive the following properties of $\sigma(n,s)$ in (17):

$$\begin{aligned} &\sigma(n,0) = 1, \\ &\sigma(n,s) = 0, \quad \text{for } s > n \text{ or } s < 0, \\ &\sigma(n,s) = \sigma(n-1,s) + n\sigma(n-1,s-1) \end{aligned} \tag{21}$$

and these together with (15)-(17) lead to the following relation between the elements $w_n(i,j)$ of $\mathcal{W}_n$ and $w_{n-1}(i,j)$ of $\mathcal{W}_{n-1}$:

$$w_n(i,j) = \frac{n}{n-j}w_{n-1}(i,j) - \frac{1}{n-j}w_{n-1}(i-1,j), \tag{22}$$

for $i = 1,\ldots,n$ and $j = 1,\ldots,n-1$, with the convention $w_n(0,j) := 0$.

Denoting by $\widetilde{w}_n(i,j)$ the elements of the matrix in the right-hand side of (18), observe that (22) implies that

$$\widetilde{w}_n(i,j) = \frac{nw_{n-1}(i,j) - w_{n-1}(i-1,j)}{n-j} = w_n(i,j),\ i,j = 1,\ldots,n-1.$$

This means that the elements of $\mathcal{M}_{n-1}\mathcal{W}_{n-1}\mathcal{D}_{n-1}$ coincide with the elements in the first $n-1$ rows and $n-1$ columns of $\mathcal{W}_n$.

Last row of $\mathcal{W}_n$ follows directly from the expression of $w_n(n,j)$ in (15). Since relations (21) imply that

$$\sum_{r=0}^{n-i}(-1)^r n^r \sigma(n, n-i-r) = \sigma(n-1, n-i),$$

the expression of $\mathbf{\Sigma}_{k-1}$ can be obtained easily. □

Based on this result, we propose the following algorithm to obtain the inverse of a Vandermonde matrix of the form (12).

Modified algorithm for inverting $\mathcal{V}_n$

1. `Compute, explicitly, first and last rows of` $\mathcal{W}_n$`:`

$$w_n(1,j) = (-1)^{j+1}\binom{n}{j}; \tag{23}$$

$$w_n(n,j) = \frac{(-1)^{n+j}}{(n-j)!(j-1)!}; \tag{24}$$

`for` $j = 1,\ldots,n$.

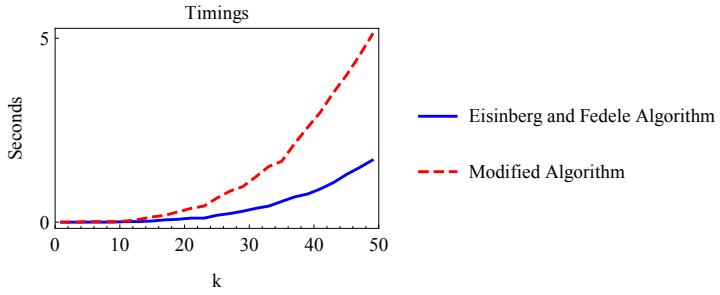

Figure 5: CPU time for inverting $V(1,\ldots,k)$.

2. `Compute recursively:`

$$w_n(i,n) = w_n(i+1,n) + \frac{(-1)^{n-i}}{(n-1)!}\begin{bmatrix} n+1 \\ i+1 \end{bmatrix}; \tag{25}$$

$$w_n(i,j) = \frac{n}{n-j}w_{n-1}(i,j) - \frac{1}{n-j}w_{n-1}(i-1,j),\ j=1,\ldots,n-1, \tag{26}$$

`for` $i = n-1,\ldots,2.$

Remark 2 *The expressions* (24) *and* (26) *come at once from* (20) *and* (22)*, respectively. Using the result*

$$\sum_{r=0}^{n}(-1)^{n-r}j^r\begin{bmatrix} n \\ r \end{bmatrix} = 0,\ \textit{for } j<n,$$

(see e.g. [17]) together with (17) *gives*

$$\sum_{r=0}^{n-1}(-1)^r j^r \sigma(n,n-1-r) = \frac{n!}{j}$$

and this, in turn, yields

$$w_n(1,j) = (-1)^{1+j}\Phi(n,j)\sum_{r=0}^{n-1}(-1)^r j^r\sigma(n,n-1-r) = (-1)^{1+j}\binom{n}{j}.$$

Finally, (25) *follows from* (19) *and* (21).

At this stage our goal is to provide an experimental comparison between Eisinberg and Fedele algorithm and the modified algorithm proposed in this work for inverting a Vandermonde matrix $\mathcal{V}_n$.

All the numerical results were obtained by the use of *Mathematica*, working with infinite precision. Figures 5 and 6 compare the performance of this new algorithm with the one proposed by Eisinberg and Fedele. We underline that the CPU time presented on Fig. 5 concerns the effort needed to computing the inverse $\mathcal{W}_n$, for a given n. When the goal is to compute all the inverses of order less or equal than n we observe in Fig. 6 that the modified algorithm is, in fact, very competitive comparing with the Eisinberg and Fedele algorithm.

We have now the necessary tools to express a reduced quaternion valued monogenic function in a series expansion of PCP. First we note that (14) together with (23)-(26) lead easily to the following result.

Proposition 4 *Each generalized power of degree k can be written as*

$$\mathcal{G}_s^k = \sum_{t=0}^{k} a_{s,t}\mathcal{Z}_t^k,\ s=0,\ldots,k,$$

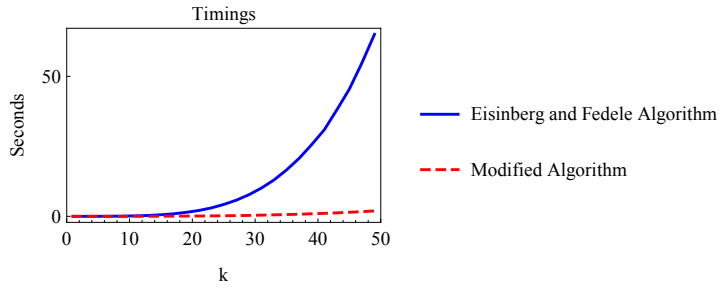

Figure 6: CPU time for computing $\{V(1), V(1,2), \ldots, V(1,\ldots,k)\}$.

where

$$a_{s,t} = \binom{k}{s}^{-1} (-\boldsymbol{e}_1 - (t+1)\boldsymbol{e}_2)^k w_{k+1}(s+1, t+1)$$

with $w_{k+1}(s+1,t+1)$ *given by* (23)-(26).

Finally, one can use this last result together with the Taylor expansion (3) to conclude:

Proposition 5 *Any monogenic function* $f : \Omega \subset \mathbb{R}^3 \to \mathbb{H}$ *can be written as a series of the form*

$$f(x) = \sum_{k=0}^{\infty} \sum_{t=0}^{k} c_{k,t} \mathcal{Z}_t^k,$$

where the coefficients $c_{k,t}$ *are given by*

$$c_{k,t} = \frac{(-\boldsymbol{e}_1 - (t+1)\boldsymbol{e}_2)^k}{k!} \sum_{s=0}^{k} \frac{\partial^k f(0)}{\partial x_1^{k-s} \partial x_2^s} w_{k+1}(s+1, t+1).$$

Remark 3 *We point out that* $(-\boldsymbol{e}_1 - (t+1)\boldsymbol{e}_2)^k$*, depending on the parity of* k*, is a real number (if* k *even) or a pure vector (if* k *odd). More precisely,*

$$(-\boldsymbol{e}_1 - (t+1)\boldsymbol{e}_2)^k = (-1)^{\frac{k}{2}} (t^2 + 2t + 2)^{\lfloor \frac{k}{2} \rfloor} a_{k,t},$$

where $\lfloor . \rfloor$ *is the floor function and* $a_{k,t}$ *is*

$$a_{k,t} = \begin{cases} 1, & k \text{ even} \\ \boldsymbol{e}_1 + (t+1)\boldsymbol{e}_2, & k \text{ odd} \end{cases}.$$

6 Final Remarks

The algorithm proposed in this work allows to obtain the inverse of Vandermonde matrices in equally spaced nodes explicitly and, from the computational point of view, efficiently. Combining this fact with the performance revealed by the complex algorithm, presented in Sect. 4, for computing PCP of the form (6)-(9) we can conclude that these PCP can become, in the future, an attractive and competitive tool for applications, in particular, in the framework of approximation problems.

Acknowledgements

This work was supported by Portuguese funds through the CIDMA - Center for Research and Development in Mathematics and Applications, the Research Centre of Mathematics of the University of Minho and the Portuguese Foundation for Science and Technology ("FCT - Fundação para a Ciência e a Tecnologia"), within projects PEst-OE/MAT/UI4106/2014 and PEstOE/MAT/UI0013/2014.

References

[1] Álvarez-Peña, C., Porter, M.: Appell bases for monogenic functions of three variables. Adv. in Appl. Clifford Algebras 23, 547–560 (2013)

[2] Björck, A., Pereyra, V.: Solution of Vandermonde systems of equations. Mathematics of Computation 24, 893–903 (1970)

[3] Bock, S., Gürlebeck, K.: On a generalized Appell system and monogenic power series. Math. Methods Appl. Sci. 33(4), 394–411 (2010)

[4] Brackx, F., Delanghe, R., Sommen, F.: Clifford analysis. Pitman, Boston-London-Melbourne (1982)

[5] Cação, I., Gürlebeck, K., Malonek, H.: Special monogenic polynomials and L_2-approximation. Adv. Appl. Clifford Algebras 11(S2), 47–60 (2001)

[6] Cação, I., Malonek, H.: On complete sets of hypercomplex Appell polynomials. In: Simos, T.E., Psihoyios, G., Tsitouras, C. (eds.) AIP Conference Proceedings. vol. 1048, pp. 647–650 (2008)

[7] Cruz, C., Falcão, M.I., Malonek, H.R.: On pseudo-complex bases for monogenic polynomials. In: Sivasundaram, S. (ed.) AIP Conference Proceedings, vol. 1493, pp. 350–355 (2012)

[8] Cruz, C., Falcão, M.I., Malonek, H.R.: On the structure of generalized Appell sequences of paravector valued homogeneous monogenic polynomials. In: Simos, T.E., Psihoyios, G., Tsitouras, C. (eds.) AIP Conference Proceedings. vol. 1479, pp. 283–286 (2012)

[9] Cruz, C., Falcão, M.I., Malonek, H.R.: Monogenic pseudo-complex power functions and their applications. Mathematical Methods in the Applied Sciences (2013), doi:10.1002/mma.2931

[10] Delanghe, R., Lávička, R., Souček, V.: The Gelfand-Tsetlin bases for Hodge-de Rham systems in Euclidean spaces. Math. Methods Appl. Sci. 7(35), 745–757 (2012)

[11] Eisinberg, A., Fedele, G.: On the inversion of the Vandermonde matrix. Applied Mathematics and Computation 174, 1384–1397 (2006)

[12] Falcão, M.I., Miranda, F.: Quaternions: A Mathematica package for quaternionic analysis. In: Murgante, B., Gervasi, O., Iglesias, A., Taniar, D., Apduhan, B. (eds.) Lecture Notes in Computer Science, vol. 6784, pp. 200–214. Springer-Verlag Berlin Heidelberg (2011)

[13] Falcão, M.I., Cruz, J., Malonek, H.: Remarks on the generation of monogenic functions. 17th Inter. Conf. on the Appl. of Computer Science and Mathematics on Architecture and Civil Engineering, Weimar (2006)

[14] Falcão, M.I., Malonek, H.: Generalized exponentials through Appell sets in $\mathbb{R}^{n+1}$ and Bessel functions. In: Simos, T.E., Psihoyios, G., Tsitouras, C. (eds.) AIP Conference Proceedings. vol. 936, pp. 738–741 (2007)

[15] Gürlebeck, K., Malonek, H.: A hypercomplex derivative of monogenic functions in $\mathbb{R}^{n+1}$ and its applications. Complex Variables Theory Appl. 39, 199–228 (1999)

[16] Gürlebeck, K., Habetha, K., Sprößig, W.: Holomorphic functions in the plane and n-dimensional space. Birkhäuser Verlag, Basel (2008)

[17] Knuth, D.E.: The art of computer programming, vol. 1. Addison Wesley, Reading MA (1968)

[18] Lávička, R.: Canonical bases for sl(2,c)-modules of spherical monogenics in dimension 3. Archivum Mathematicum Tomus 46, 339–349 (2010)

[19] Malonek, H.: Power series representation for monogenic functions in $\mathbb{R}^{n+1}$ based on a permutational product. Complex Variables, Theory Appl. 15, 181–191 (1990)

[20] Malonek, H.: Selected topics in hypercomplex function theory. In: Eriksson, S.L. (ed.) Clifford algebras and potential theory, pp. 111–150. 7, University of Joensuu (2004)

[21] Malonek, H.R., Falcão, M.I.: On paravector valued homogeneous monogenic polynomials with binomial expansion. Advances in Applied Clifford Algebras 22(3, SI), 789–801 (2012)

[22] Parker, F.D.: Mathematical Notes: Inverses of Vandermonde matrices. Amer. Math. Monthly 71(4), 410–411 (1964)

[23] Traub, J.F.: Associated polynomials and uniform methods for the solution of linear problems. SIAM Rev. 8, 277–301 (1966)

[24] Wagner, D.B.: Dynamic programming. The Mathematica J. 5, 42–51 (1995)