Chapter 13
Newton's Approach to General Algebraic Equations over Clifford Algebras

Drahoslava Janovská and Gerhard Opfer

Dedicated to Wolfgang Sprößig on the occasion of his 70th birthday

Abstract There is a short section describing how Newton's method works for algebraic problems over Clifford algebras. There are two applications. Zeros of unilateral polynomials over a Clifford algebra in $\mathbb{R}^8$ and solutions of a Riccati equation over all eight Clifford algebras in $\mathbb{R}^4$.

Keywords Clifford algebras · Newton's method · Algebraic equations over Clifford algebras · Riccati equation

Mathematics Subject Classification (2010) Primary 15A66; Secondary 12E10, 1604

13.1 Introduction

Let $p : \mathbb{R}^N \to \mathbb{R}^N$ be a mapping, where $\mathbb{R}^N$ stands for any Clifford algebra. For introduction to Clifford Algebras, see [1], for algebraic and analytic properties of coquaternion algebra, see [6]. Examples for p are polynomials of a general type, including unilateral and bilateral polynomials. Other examples are matrix equations

D. Janovská (✉)
University of Chemistry and Technology, Prague 6, Czech Republic
e-mail: janovskd@vscht.cz

G. Opfer
University of Hamburg, Faculty on Mathematics, Informatics, and Natural Sciences [MIN], Hamburg, Germany
e-mail: gerhard.opfer@uni-hamburg.de

S. Bernstein (ed.), *Topics in Clifford Analysis*, Trends in Mathematics,
https://doi.org/10.1007/978-3-030-23854-4_13

like the Riccati equation

$$p(\mathbf{X}) := \mathbf{A} + \mathbf{B}\mathbf{X} + \mathbf{X}\mathbf{C} + \mathbf{X}\mathbf{D}\mathbf{X} = \mathbf{0}, \tag{13.1.1}$$

where the matrix entries are elements from a Clifford algebra in $\mathbb{R}^N$. If $\mathbf{X}$ is a matrix of size $m \times n$, then, in this case, $p : \mathbb{R}^{mnN} \to \mathbb{R}^{mnN}$. The Riccati equation will be treated separately in the last section. For other examples see also [2]. In order to find the solutions of $p(z) = 0$ by Newton's method, the linear system for h

$$p(z) + p'(z)h = \mathbf{0} \tag{13.1.2}$$

has to be solved where in the beginning, z has to be replaced by an arbitrary *guess* $z \in \mathbb{R}^N$ and after having found h as the solution of (13.1.2) the guess has to be replaced by $z := z + h$. In a paper by Lauterbach and Opfer [5], it was shown that $p'(z)h$ is the linear part of $p(z + h)$ with respect to h. To mention an example let $p(z) = z^2 + a_1 z + a_0$. Then,

$$p(z + h) = (z + h)^2 + a_1(z + h) + a_0 = z^2 + hz + zh + h^2 + a_1 z + a_1 h + a_0$$

and the linear part of this expression with respect to h is

$$p'(z)h = hz + zh + a_1 h. \tag{13.1.3}$$

In all cases and independent of the algebra the linear part always consist of a sum with terms of the form ahb. Since, by definition, $p'(z)h$ is a real, linear mapping $\mathbb{R}^N \to \mathbb{R}^N$ in h, the linear part must have a matrix representation

$$p'(z)h = \mathbf{M}_z h \tag{13.1.4}$$

where $\mathbf{M}_z$ is a real $N \times N$ matrix and h is now a real column vector of length N. The matrix $\mathbf{M}_z$ is called the *Jacobi matrix* of (13.1.2). In comparison with the numerical Jacobi matrix which is constructed by replacing $p'(z)h$ by partial derivatives, the Jacobi matrix we use is exact and in addition easy to compute. Even for general, nonunilateral polynomials the matrix representation of $p'(z)h$ is given in [5, Section 4]. And the underlying algebra is not of large importance. In the MATLAB program given in Table 13.4 one can see how to compute the matrix $\mathbf{M}_z$ given in (13.1.4) for the linear term ahb. Let $y = p(z)$. We call the euclidean $\mathbb{R}^N$ norm $||y||$ the *error* of z. Thus, our aim is to find all z with error zero.

13.2 Application of Newton's Method to Polynomials with Clifford Coefficients

Let p now be a unilateral polynomial in any $\mathbb{R}^N$ algebra. In order to find zeros of p we have to use (13.1.2) together with (13.1.4). As an example we use a Clifford algebra in $\mathbb{R}^8$ which we call C_4 and as examples we will be looking for zeros of a quadratic polynomial and a polynomials of degree 7.

Table 13.1 Multiplication table of C_4 for the canonical unit vectors u_k in $\mathbb{R}^8,\ 1 \le k \le 8$

u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8
u_2	$-u_1$	u_4	$-u_3$	u_6	$-u_5$	u_8	$-u_7$
u_3	$-u_4$	$-u_1$	u_2	u_7	$-u_8$	$-u_5$	u_6
u_4	u_3	$-u_2$	$-u_1$	u_8	u_7	$-u_6$	$-u_5$
u_5	$-u_6$	$-u_7$	u_8	$-u_1$	u_2	u_3	$-u_4$
u_6	u_5	$-u_8$	$-u_7$	$-u_2$	$-u_1$	u_4	u_3
u_7	u_8	u_5	u_6	$-u_3$	$-u_4$	$-u_1$	$-u_2$
u_8	$-u_7$	u_6	$-u_5$	$-u_4$	u_3	$-u_2$	u_1

In order to describe the multiplication rules of the algebra C_4 we abbreviate the eight unit vectors units(k) in $\mathbb{R}^8$ also by $u_k,\ k = 1, 2, \dots, 8$, and the multiplication rules are listed in Table 13.1.

The elements in the C_4 algebra for use in the MATLAB program, which is printed at the end of this section as Table 13.4, page 273, will have the name

$$a = c4([a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8]), a_j \in \mathbb{R}, j = 1, 2, \dots, 8 \qquad (13.2.1)$$

or in short

$$a = c4(h), h \in \mathbb{R}^8.$$

In order to apply the program, an initial guess `xold` is needed. For a systematic search a random guess is very practical. We always used random integer values as components of `xold`. This can be done by the MATLAB command `xold=c4(round(40*(rand(1,8)-0.5)))`, where `rand` is the MATLAB command for random (uniformly distributed) numbers in [0, 1]. The given expression for `xold` produces an integer row vector with eight entries in [−20, 20]. We have applied the program to two examples, a quadratic polynomial and a polynomial of degree 7. In all cases there are several solutions.

Example Let the quadratic polynomial p be defined by

$$p(z) = z^2 + c4([2, 3, 5, 7, 11, 13, 17, 19])z + u_8. \qquad (13.2.2)$$

We found the following four zeros (see Table 13.2) all in less than 20 steps by applying Newton's method. We never found a singular Jacobi matrix. Whether there are more solutions as given, it is an open question.

Example We selected the following polynomial

$$p(z) = u_1 z^7 + u_2 z^6 + u_3 z^5 + u_4 z^4 + u_5 z^3 + u_6 z^2 + u_7 z + u_8. \qquad (13.2.3)$$

Table 13.2 Zeros of p, a quadratic polynomial defined in (13.2.2)

−1.981359548776446,	−2.983544918358173,	−5.012933206449555,	−6.989595272862536,
−10.993692804341773,	−13.005505685207023,	−16.997886278535510,	−18.997172393977728.
−0.018640451223553,	−0.016455081641829,	0.012933206449553,	−0.010404727137464,
−0.006307195658224,	0.005505685207025,	−0.002113721464491,	−0.002827606022272.
8.497172393977728,	−9.997886278535509,	4.005505685207025,	−8.993692804341777,
−8.989595272862536,	−3.987066793550447,	−9.983544918358172,	−8.518640451223554.
−10.497172393977728,	6.997886278535509,	−9.005505685207025,	1.993692804341777,
−2.010404727137465,	−9.012933206449553,	−7.016455081641829,	−10.481359548776446.

Two consecutive lines, separated by a dot . and a small skip define one zero

We can guess already one solution, namely $z = u_4$. If we look at Table 13.1 we see that $u_4^2 = -1 \Rightarrow u_4^4 = 1 \Rightarrow u_4^5 = u_4 \Rightarrow u_4^6 = u_4^2 = -1 \Rightarrow u_4^7 = u_4^3 = -u_4$, thus,

$$p(u_4) = -u_4 - u_2 + u_3u_4 + u_4 - u_5u_4 - u_6 + u_7u_4 + u_8 =$$
$$= -u_4 - u_2 + u_2 + u_4 - u_8 - u_6 + u_6 + u_8 = 0.$$

We applied the given program to (13.2.3) and in less than 40 Newton iterations for each zero we found the solutions given in Table 13.3. It contains 18 zeros of p. These are found by using random integer guesses. Whether there are more zeros than indicated, it is an open problem. We did not encounter one example which did not converge. The convergence behavior was typical for Newton's method. If the error was under a certain limit, say 10^{-2}, then there were only few remaining steps so that almost machine precision was reached. Therefore, in the two Tables 13.2, 13.3 the original MATLAB results are presented. The advantage of using Newton's method for finding zeros of polynomials with arbitrary algebra coefficients is its easy use and its high precision. There is another advantage. If with some other method a zero is computed it is easy to apply few additional Newton steps in order to reduce the error. The disadvantage is, that we do not know whether there are still other, nondiscovered zeros. We note, that the frequently appearing number 0.353553390593274 in Table 13.3 is $\frac{\sqrt{2}}{4}$ (Table 13.4).

13.3 Application to Riccati Equation

In the Riccati equation over a Clifford algebra in $\mathbb{R}^N$, given in (13.1.1), we can choose the sizes of the corresponding matrices as follows:

$$\mathbf{X} : m \times n \Rightarrow \mathbf{A} : m \times n, \ \mathbf{B} : m \times m, \ \mathbf{C} : n \times n, \ \mathbf{D} : n \times m.$$

The linear part of $p(\mathbf{X} + \mathbf{H})$ is easy to compute:

$$p'(\mathbf{X})\mathbf{H} = \mathbf{H}(\mathbf{C} + \mathbf{D}) + (\mathbf{B} + \mathbf{XD})\mathbf{H} = \mathbf{M}_1\mathbf{H} + \mathbf{M}_2\mathbf{H} = (\mathbf{M}_1 + \mathbf{M}_2)\mathbf{H}.$$

In this case the matrices $\mathbf{M}_1, \ \mathbf{M}_2$ are of order mnN and $\mathbf{H}$ is a column vector of dimension mnN. For more details see also [5], Theorem 8.1 and [3]. In [5], also quotations to the relevant literature is given. We will use an example defined in all eight $\mathbb{R}^4$ algebras which have the names and notations:

1. Quaternions, $\mathbb{H}$,
2. Coquaternions, $\mathbb{H}_{\text{coq}}$,
3. Tessarines, $\mathbb{H}_{\text{tes}}$,

Table 13.3 Table of zeros of p, a polynomial of degree 7, defined in (13.2.3)

±(0,	0,	0,	0.923879532511287,
0,	0,	0,	−0.382683432365090).
0.353553390593274,	0.353553390593274,	0,	0.5,
−0.5,	0,	0.353553390593274,	−0.353553390593274.
±(0.382683432365090,	0,	0,	0,
0.923879532511287,	0,	0,	0).
0.191341716182520,	0,	0,	0.961939766255643,
−0.038060233744357,	0,	0,	−0.191341716182520.
0,	0,	0,	1,
0,	0,	0,	0.
−0.191341716182545,	0,	0,	0.038060233744357,
−0.961939766255643,	0,	0,	0.191341716182545.
±(0,	$\sqrt{0.5}$,	0,	0,
0,	0,	0,	$-\sqrt{0.5}$).
0.191341716182545,	0,	0,	0.038060233744357,
0.961939766255643,	0,	0,	0.191341716182545.
−0.191341716182545,	0,	0,	0.961939766255643,
0.038060233744357,	0,	0,	−0.191341716182545.
−0.353553390593272,	0.353553390593274,	0,	0.5,
0.5,	0,	−0.353553390593275,	−0.353553390593272.
±(0.544895106775818,	0.353553390593275,	0,	−0.461939766255644,
0.461939766255644,	0,	0.353553390593272,	−0.162211674410729).
±(-0.162211674410729,	0.353553390593274,	0,	0.461939766255643,
0.461939766255643,	0,	−0.353553390593274,	−0.544895106775819).
−0.353553390593272,	−0.353553390593274,	0,	0.5,
−0.5,	0,	−0.353553390593275,	0.353553390593272.

Two consecutive lines, separated by a dot . and a small skip define one or two zeros

Table 13.4 MATLAB program for computing zeros of polynomials over coefficients from $\mathbb{R}^8$ algebra C_4 by Newton’s method

```
%[xnew,J]=c4_newton(c,xold); c presents the vector of polynomial
%coefficients and xold is an arbitrary initial guess. Computed
%are J, the exact Jacobi matrix at xold, and xnew, the result
%of the application of one Newton step.
%%%
%Basis for the program:
%R. Lauterbach - G. Opfer, AACA 24(2014), pp. 1059 - 1073.
%The polynomials have the form (highest coefficient first)
%%%
%p(x) = c_1x^n + c_2x^{n-1} + ... + c_{n+1}.
%%%
%This programm works in principle for all geometric R^N algebras,
%provided the corresponding algebraic rules have been transmitted
%to MATLAB by a technique called "overloading".
%%%%===============================================================
%function [xnew,J]=c4_newton(c,xold); %J is the exact Jacobi matrix
%  n=length(c)-1;
%  global N;
%  dim=8;
%  N=dim;
%  J=zeros(N,N);
%  for ell=n:-1:1
%    J=J+derivative_of_xpowerj(ell,xold,c(n-ell+1));
%  end;
%  y=Polyval(c,xold);
%  if abs(det(J))>=1e-10
%    h=-J\col(y);
%    xnew=xold+c4(h);
%    else
%    error(‘Jacobi matrix J is near to singular’);
%    %This happens almost never!
%  end;
%
%%%%===============================================================
%function M=derivative_ahb_in_matrixform(a,b); %a,b depend on x
%global N;
%M=[];
%for k=1:N
%M=[M,col(a*units(k)*b)]; %M is real NxN matrix
%end;
%
%%%%===============================================================
%function M=derivative_of_xpowerj(j,x,factor);
%  %M is  real NxN Matrix.
%  %factor stands for polynomial coefficients.
%  global N;
%  M=zeros(N,N);
%  for k=1:j
%    a=factor*x^(k-1); b=x^(j-k);
%    M=M+derivative_ahb_in_matrixform(a,b);
%  end;}
%
%%%%===============================================================
```

Table 13.5 Definition of the coefficients of algebraic Riccati equation

$$\mathbf{A} := \begin{bmatrix} (\ \ 1,-4,-2,\ \ 0), & (-1,\ \ 4,\ \ 3,\ \ 5), & (\ \ 2,-3,\ \ 2,-5) \\ (-1,-1,\ \ 3,\ \ 3), & (\ \ 2,-3,-4,\ \ 0), & (\ \ 4,\ \ 5,\ \ 0,-4) \end{bmatrix};$$

$$\mathbf{B} := \begin{bmatrix} (\ \ 3,-3,\ \ 4,-2), & (-3,-2,\ \ 1,\ \ 0) \\ (-1,\ \ 3,\ \ 1,\ \ 0), & (\ \ 4,-2,\ \ 3,\ \ 3) \end{bmatrix};$$

$$\mathbf{C} := \begin{bmatrix} (-1,\ \ 1,-4,-4), & (\ \ 0,\ \ 3,\ \ 4,-4), & (\ \ 1,\ \ 0,-5,-2) \\ (-3,\ \ 3,-2,\ \ 0), & (-3,\ \ 1,-2,\ \ 2), & (\ \ 2,\ \ 2,\ \ 0,-4) \\ (-3,\ \ 4,-3,\ \ 3), & (\ \ 0,\ \ 5,-4,-1), & (-4,\ \ 5,-5,\ \ 3) \end{bmatrix};$$

$$\mathbf{D} := \begin{bmatrix} (\ \ 5,\ \ 5,-3,\ \ 5), & (-3,\ \ 0,-1,\ \ 1) \\ (\ \ 2,-5,\ \ 3,\ \ 4), & (\ \ 5,-2,\ \ 1,-3) \\ (-2,-5,-4,\ \ 3), & (-4,-2,\ \ 3,-2) \end{bmatrix}.$$

4. Cotessarines, $\mathbb{H}_{\text{cotes}}$,
5. Nectarines, $\mathbb{H}_{\text{nec}}$,
6. Conectarines, $\mathbb{H}_{\text{con}}$,
7. Tangerines, $\mathbb{H}_{\text{tan}}$,
8. Cotangerines, $\mathbb{H}_{\text{cotan}}$.

It should be noted, that the algebras numbered 3, 4, 7, 8 are commutative. About the names and the multiplication rules see [4, 7]. We quote an example from [5].

Example We choose $m = 2, n = 3$ and define the algebraic Riccati equation by $\mathbf{A} \in \mathcal{A}^{2\times 3}, \mathbf{B} \in \mathcal{A}^{2\times 2}, \mathbf{C} \in \mathcal{A}^{3\times 3}, \mathbf{D} \in \mathcal{A}^{3\times 2}$, where the matrix entries are given in Table 13.5.

The data of Table 13.5 are randomly chosen integers in $[-5, 5]$. We choose $\mathbf{X} = \mathbf{0}$ as initial guess with the exception of Algebra 6, where another guess is used. Newton's method converges then in all 8 cases. The solutions are given in Table 13.6. These solutions are not necessarily the only solutions.

Table 13.6 Solutions $\mathbf{X} \in \mathcal{A}^{2\times 3}$ of the algebraic Riccati equation in all eight $\mathbb{R}^4$ algebras

Pos.	Quaternions $\mathbb{H}$	Coquaternions $\mathbb{H}_{\mathrm{coq}}$
x_{11}	$(-0.1045, -0.1495, 0.2660, 0.0712)$	$(0.1544, 0.6186, -0.0883, 0.3512)$
x_{21}	$(0.3071, 0.3399, -0.6350, 0.1765)$	$(0.2732, 0.3379, -0.5506, 0.2582)$
x_{12}	$(0.2731, -0.1913, 0.1003, 0.6167)$	$(0.1612, 0.0891, -0.6692, -0.2445)$
x_{22}	$(0.5842, 0.7970, -0.0900, -0.4788)$	$(0.5916, 1.3237, 0.1198, 0.6950)$
x_{13}	$(0.9956, 0.1306, 0.3851, 0.6174)$	$(-0.1351, 0.2386, -0.6664, 0.8003)$
x_{23}	$(-0.5317, -0.3326, -0.5753, -0.3978)$	$(-1.2738, 0.5379, 0.6102, 1.0119)$

Pos.	Tessarines $\mathbb{H}_{\mathrm{tes}}$	Cotessarines $\mathbb{H}_{\mathrm{cotes}}$
x_{11}	$(-0.8748, -0.1261, 0.0210, 0.0205)$	$(0.0749, 0.5097, 0.8189, -0.7321)$
x_{21}	$(-0.1044, -0.4106, 1.0738, -0.9254)$	$(0.1177, -0.0515, -0.7539, 0.0017)$
x_{12}	$(0.1632, 0.0329, -0.1188, -0.4736)$	$(-1.7670, 0.0836, 0.3282, -1.2972)$
x_{22}	$(0.0282, 0.4299, 0.3035, 0.4502)$	$(0.3680, 0.1663, 0.7493, 1.4654)$
x_{13}	$(-0.6189, 0.5054, -0.8030, -0.4800)$	$(1.0919, -1.0135, -0.5820, 1.0441)$
x_{23}	$(0.8554, -0.0146, 0.2011, -1.1011)$	$(-0.5413, -0.2660, 0.0956, -0.2739)$

Pos.	Nectarines $\mathbb{H}_{\mathrm{nec}}$	Conectarines $\mathbb{H}_{\mathrm{con}}$
x_{11}	$(0.1490, -0.1557, 0.6250, 0.3262)$	$(1.3763, -0.5053, 0.8822, -1.0762)$
x_{21}	$(0.1598, -0.5805, -0.9796, -1.2966)$	$(0.4270, 0.0322, 0.5237, 0.8558)$
x_{12}	$(0.5285, 0.1447, -0.4479, 0.0074)$	$(1.5484, -4.3525, -1.4404, 5.4397)$
x_{22}	$(-0.2607, -0.4621, -0.8051, -0.4240)$	$(-0.5729, 1.4386, 1.0001, -1.7677)$
x_{13}	$(-0.2660, -0.3224, 0.3163, 0.3754)$	$(1.7588, -0.5349, -0.3498, 0.8295)$
x_{23}	$(-0.7931, -0.9394, 0.4466, 0.6899)$	$(-0.3289, 0.4151, -0.0512, -0.3035)$

Pos.	Tangerines $\mathbb{H}_{\mathrm{tan}}$	Cotangerines $\mathbb{H}_{\mathrm{cotan}}$
x_{11}	$(0.0238, 0.1553, -0.6713, 0.3590)$	$(-0.4374, -0.8677, -0.4367, 0.8265)$
x_{21}	$(0.3589, -0.1427, -0.2777, -0.6520)$	$(-0.4075, 0.6499, 0.5927, -0.0469)$
x_{12}	$(-0.1408, -0.0282, 0.2520, 0.8164)$	$(0.2330, -0.6134, -0.4203, 0.2160)$
x_{22}	$(-0.1875, -0.0557, 0.5904, 0.4577)$	$(-0.4898, -0.1886, 0.2013, 0.4882)$
x_{13}	$(0.1099, -0.4139, 0.0090, -0.1696)$	$(0.2155, -0.2357, -0.4028, 0.0678)$
x_{23}	$(-0.1393, -0.2064, 0.0938, 0.0396)$	$(0.2917, 0.1488, 0.3577, -0.2748)$

References

1. D.J.H. Garling, *Clifford Algebras: An Introduction* (Cambridge Univerity Press, Cambridge, 2011), 200 pp
2. K. Gürlebeck, W. Sprössig, *Quaternionic and Clifford Calculus for Physicists and Engineers* (Wiley, Chichester, 1997), 371 pp
3. D. Janovská, G. Opfer, The algebraic Riccati equation for quaternions. Dedicated to Ivo Marek on the occasion of his 80th birthday. Adv. Appl. Clifford Algebras **23**, 907–918 (2013)
4. D. Janovská, G. Opfer, Zeros and singular points for one-sided, coquaternionic polynomials with an extension to other $\mathbb{R}^4$ algebras. Electron. Trans. Numer. Anal. **41**, 133–158 (2014)
5. R. Lauterbach, G. Opfer, The Jacobi matrix for functions in noncommutative algebras. Adv. Appl. Clifford Algebras **24**, 1059–1073 (2014). Erratum: Adv. Appl. Clifford Algebras, 24 (2014), p. 1075
6. A.A. Pogurui, R.M. Rodriguez-Dagnino, Some algebraic and analytic properties of coquaternion algebra. Adv. Appl. Clifford Algebras **20**, 79–84 (2010)
7. B. Schmeikal, Tessarinen, Nektarinen und andere Vierheiten. Beweis einer Beobachtung von Gerhard Opfer. Mitt. Math. Ges. Hamburg **34**, 81–108 (2014)