Quaternionic Polynomials in Magma

Introducing the *qPoly* Package

Przemysław Koprowski
Institute of Mathematics · University of Silesia, Poland

ISSAC 2026 Software Presentations
Thursday, July 16th, 2026

Introduction & Package Motivation

- **Goals:**
  - a complete tool-set for MAGMA to handle quaternionic polynomials;
  - up to the level of univariate polynomials over fields;
  - seamless integration and complete documentation.
- **Open source:** distributed under MIT license.
- Latest version: **v26.06**, released 2026.06.21.

`http://www.pkoprowski.eu/qpoly`

Seamless Magma Integration

- **Intuitive Naming:** Retains standard Magma function naming conventions.
- **Non-commutative twist:** `Left`/`Right` qualifiers are explicitly appended when needed (*e.g.*, `EuclideanLeftDivision`).
- **Signature Matching:** Identical input/output specifications ensuring immediate compatibility.

Magma Manual Page

24.8.1 Factorization and Irreducibility

`Factorization(f)`

`Factorisation(f)`

`Al` MonStgElt *Default* : "*Default*"

Given a univariate polynomial f over the ring R, this function returns the factorization of f as a factorization sequence Q, that is, a sequence of pairs, each consisting of an irreducible factor q_i a positive integer k_i (its multiplicity). Each irreducible factor is normalized (see the function `Normalize`), so the expansion of the factorization sequence is the unique canonical associate of f. The function also returns the unit u of R giving the normalization, so $f = u \cdot \prod_i q_i^{k_i}$.

The coefficient ring R must be one of the following: a finite field $\mathbf{F}_q$, the ring of integers $\mathbf{Z}$, the field of rationals $\mathbf{Q}$, an algebraic number field $\mathbf{Q}(\alpha)$, a local ring, or a polynomial ring, function field (rational or algebraic) or finite-dimensional affine algebra (which is a field) over any of the above.

For factorization over very small finite fields, the Berlekamp algorithm is used by default, which depends on fast linear algebra (see, for example, [Knu97, section 4.6.2] or [vzGG99, section 14.8]). For medium to large finite fields, the von zur

qPoly Manual Page

3.6. **Factorization.**

`Factorization( p )`

`Factorisation( p )`

`Nice` BoolElt *Default:* `true`

Let $\mathscr{A}$ be a division quaternion algebra over a number field K. Given a quaternionic polynomial $\wp$, this function returns a factorization of $\wp$ as a sequence of pairs, each consisting of an irreducible factor $\mathscr{q}_i$ and a positive integer k_i being its multiplicity. Each irreducible factor is monic. As a second output value the function returns the leading coefficient $c = \operatorname{lc} \wp$ of $\wp$. Thus, if the output is $\mathscr{L} = \big((\mathscr{q}_1, k_1), \ldots, (\mathscr{q}_n, k_n)\big)$ and c, then

$$\wp = c \cdot \mathscr{q}_1^{k_1} \cdots \mathscr{q}_n^{k_n}.$$

A quaternionic polynomial may have more than one factorization (potentially infinitely many). If the parameter `Nice` is set to `true` (the default), `Factorization` tries (reasonably hard) to find a factorization of $\wp$ with not too big coefficients. This behavior can be disabled by setting the parameter to `false`. The optimization of

The Scope of $\mathscr{qPoly}$ v26.06

- **Basic functions:**
  - norm, trace,
  - Euclidean algorithm,
  - central decomposition,. . . ;
- **Interpolation & roots:**
  - Bray–Whaples polynomials,
  - Lagrange interpolation;
  - Root-finding: `HasRoot`, `Roots`,
  - isolated & spherical root multiplicities;
- **Factorization & similarity:**
  - `Factorization`,
  - certified similarity verification;
- **Ideal arithmetic:**
  - one-side and two-sided ideals,
  - arbitrary handedness,
  - mixed handedness,
  - bounds and closures,. . . ;
- **Resultants & more**

qPoly v26.06 in Numbers

66

Exposed Commands
(plus variants)

22

Manual Pages

21

Included Examples

Live Demonstration – What to Expect

- **Basics:** package loading, extended Euclidean algorithm.
- **Ideals:** intersection of ideals of opposite handedness.
- **Root-finding:** polynomial roots, different multiplicities.
- **Factorization & Similarity**: non-uniqueness of non-commutative factorization, similarity of factors.
- **Practical application:** decomposition of psd rational polynomial into a sum of squares.

```
Magma V2.29-8     Wed Jul 15 2026 21:00:52 on laptopDell [Seed = 3388242734]
Type ? for help.  Type <Ctrl>-D to quit.
> // load the package
> path_to_package := "/home/perry/Dokumenty/Magma/qPoly/";
> AttachSpec(path_to_package cat "qpoly.spec");
```

```
Magma V2.29-8     Wed Jul 15 2026 21:00:52 on laptopDell [Seed = 3388242734]
Type ? for help.  Type <Ctrl>-D to quit.
> // load the package
> path_to_package := "/home/perry/Dokumenty/Magma/qPoly/";
> AttachSpec(path_to_package cat "qpoly.spec");
> // EXAMPLE: Bezout coefficients
> // define the ring of polynomials over a quaternion algebra
> K<t>     := CyclotomicField(7);
> A<i,j,k> := QuaternionAlgebra<K | -t+1, -t+3>;
> Ax<x>    := PolynomialRing(A);
> // take two polynomials
> p := (t - 1)*x^6 + (t^2 - 1)*x^5 + ((2*t^2 - 5*t + 3) + (t^2 - 2*t + 1)*i + j)*x\
^4 + ((-2*t^2 + 2*t) + (2*t - 2)*i + (t + 1)*j - k)*x^3 + (2*t - 3)*j*x^2 + ((t^2 \
- 2*t - 1)*j + k)*x + (-2*t + 2)*j - 2*t*k;
> q := x^5 + ((t + 2) - i)*x^4 + (3*t - 2*i + t*k)*x^3 + (t^2 + (-t + 4)*i + (2*t^\
2 - 2*t)*j + (t^2 + t)*k)*x^2 + ((4*t - 8) + 4*t*i + (t^3 - t)*j - t*k)*x + -4*t +\
 4*i + (-2*t^2 + 2*t)*j - 2*t^2*k;
```

```
Magma V2.29-8     Wed Jul 15 2026 21:00:52 on laptopDell [Seed = 3388242734]
Type ? for help.  Type <Ctrl>-D to quit.
> // load the package
> path_to_package := "/home/perry/Dokumenty/Magma/qPoly/";
> AttachSpec(path_to_package cat "qpoly.spec");

> // EXAMPLE: Bezout coefficients
> // define the ring of polynomials over a quaternion algebra
> K<t>     := CyclotomicField(7);
> A<i,j,k> := QuaternionAlgebra<K | -t+1, -t+3>;
> Ax<x>    := PolynomialRing(A);
> // take two polynomials
> p := (t - 1)*x^6 + (t^2 - 1)*x^5 + ((2*t^2 - 5*t + 3) + (t^2 - 2*t + 1)*i + j)*x\
^4 + ((-2*t^2 + 2*t) + (2*t - 2)*i + (t + 1)*j - k)*x^3 + (2*t - 3)*j*x^2 + ((t^2 \
- 2*t - 1)*j + k)*x + (-2*t + 2)*j - 2*t*k;
> q := x^5 + ((t + 2) - i)*x^4 + (3*t - 2*i + t*k)*x^3 + (t^2 + (-t + 4)*i + (2*t^\
2 - 2*t)*j + (t^2 + t)*k)*x^2 + ((4*t - 8) + 4*t*i + (t^3 - t)*j - t*k)*x + -4*t +\
 4*i + (-2*t^2 + 2*t)*j - 2*t^2*k;

> // compute their greatest common RIGHT divisor
> // and the Bezout coeffs
> r, alpha, beta := XGCRD(p, q); r;
x^2 + (1 + i)*x + -2
```

```
Magma V2.29-8     Wed Jul 15 2026 21:00:52 on laptopDell [Seed = 3388242734]
Type ? for help.  Type <Ctrl>-D to quit.
> // load the package
> path_to_package := "/home/perry/Dokumenty/Magma/qPoly/";
> AttachSpec(path_to_package cat "qpoly.spec");

> // EXAMPLE: Bezout coefficients
> // define the ring of polynomials over a quaternion algebra
> K<t>     := CyclotomicField(7);
> A<i,j,k> := QuaternionAlgebra<K | -t+1, -t+3>;
> Ax<x>    := PolynomialRing(A);
> // take two polynomials
> p := (t - 1)*x^6 + (t^2 - 1)*x^5 + ((2*t^2 - 5*t + 3) + (t^2 - 2*t + 1)*i + j)*x\
^4 + ((-2*t^2 + 2*t) + (2*t - 2)*i + (t + 1)*j - k)*x^3 + (2*t - 3)*j*x^2 + ((t^2 \
- 2*t - 1)*j + k)*x + (-2*t + 2)*j - 2*t*k;
> q := x^5 + ((t + 2) - i)*x^4 + (3*t - 2*i + t*k)*x^3 + (t^2 + (-t + 4)*i + (2*t^\
2 - 2*t)*j + (t^2 + t)*k)*x^2 + ((4*t - 8) + 4*t*i + (t^3 - t)*j - t*k)*x + -4*t +\
 4*i + (-2*t^2 + 2*t)*j - 2*t^2*k;

> // compute their greatest common RIGHT divisor
> // and the Bezout coeffs
> r, alpha, beta := XGCRD(p, q); r;
x^2 + (1 + i)*x + -2

> // check the correctness
> IsZero(p mod r) and IsZero(q mod r);
true
> alpha*p + beta*q eq r;
true
>
```

```
> // EXAMPLE: ideal arithmetic
> // define the ring of polynomials over a quaternion algebra
> K<sqrt7> := QuadraticField(7);
> A<i,j,k> := QuaternionAlgebra<K | -1, -1>;
> Ax<x>    := PolynomialRing(A);
> // take two ideals of opposite handedness
> I := RightIdeal(Ax, (x^2 + 1)*(x + i + k)); I;
Right ideal of Univariate Polynomial Ring in x over Quaternion Algebra with base
ring Quadratic Field with defining polynomial $.1^2 - 7 over the Rational Field,
defined by i^2 = -1, j^2 = -1 generated by x^3 + (i + k)*x^2 + x + i + k
> J := LeftIdeal(Ax, (x + j)*(x^2 + 2)); J;
Left ideal of Univariate Polynomial Ring in x over Quaternion Algebra with base
ring Quadratic Field with defining polynomial $.1^2 - 7 over the Rational Field,
defined by i^2 = -1, j^2 = -1 generated by x^3 + j*x^2 + 2*x + 2*j
```

```
> // EXAMPLE: ideal arithmetic
> // define the ring of polynomials over a quaternion algebra
> K<sqrt7> := QuadraticField(7);
> A<i,j,k> := QuaternionAlgebra<K | -1, -1>;
> Ax<x>    := PolynomialRing(A);
> // take two ideals of opposite handedness
> I := RightIdeal(Ax, (x^2 + 1)*(x + i + k)); I;
Right ideal of Univariate Polynomial Ring in x over Quaternion Algebra with base
ring Quadratic Field with defining polynomial $.1^2 - 7 over the Rational Field,
defined by i^2 = -1, j^2 = -1 generated by x^3 + (i + k)*x^2 + x + i + k
> J := LeftIdeal(Ax, (x + j)*(x^2 + 2)); J;
Left ideal of Univariate Polynomial Ring in x over Quaternion Algebra with base
ring Quadratic Field with defining polynomial $.1^2 - 7 over the Rational Field,
defined by i^2 = -1, j^2 = -1 generated by x^3 + j*x^2 + 2*x + 2*j

> // intersect them
> I meet J;
Two-sided ideal of Univariate Polynomial Ring in x over Quaternion Algebra with
base ring Quadratic Field with defining polynomial $.1^2 - 7 over the Rational
Field, defined by i^2 = -1, j^2 = -1 generated by x^4 + 3*x^2 + 2
```

```
> // EXAMPLE: ideal arithmetic
> // define the ring of polynomials over a quaternion algebra
> K<sqrt7> := QuadraticField(7);
> A<i,j,k> := QuaternionAlgebra<K | -1, -1>;
> Ax<x>    := PolynomialRing(A);
> // take two ideals of opposite handedness
> I := RightIdeal(Ax, (x^2 + 1)*(x + i + k)); I;
Right ideal of Univariate Polynomial Ring in x over Quaternion Algebra with base
ring Quadratic Field with defining polynomial $.1^2 - 7 over the Rational Field,
defined by i^2 = -1, j^2 = -1 generated by x^3 + (i + k)*x^2 + x + i + k
> J := LeftIdeal(Ax, (x + j)*(x^2 + 2)); J;
Left ideal of Univariate Polynomial Ring in x over Quaternion Algebra with base
ring Quadratic Field with defining polynomial $.1^2 - 7 over the Rational Field,
defined by i^2 = -1, j^2 = -1 generated by x^3 + j*x^2 + 2*x + 2*j

> // intersect them
> I meet J;
Two-sided ideal of Univariate Polynomial Ring in x over Quaternion Algebra with
base ring Quadratic Field with defining polynomial $.1^2 - 7 over the Rational
Field, defined by i^2 = -1, j^2 = -1 generated by x^4 + 3*x^2 + 2

> // the reason is:
> IdealBound(I) eq IdealBound(J);
true
>
```

```
> // EXAMPLE: Roots
> // define the ring of polynomials over a quaternion algebra
> A<i,j,k> := QuaternionAlgebra<Rationals() | -1, -1>;
> Ax<x>    := PolynomialRing(A);
> // and a polynomial p
> p        := Ax![ 6+6*i-36*j+24*k, 18-12*i-6*j+30*k,
>             23+5*i-30*j+20*k, 15-10*i-5*j+25*k,
>             16+i-6*j+4*k, 3-2*i-j+5*k, 3 ];
```

```
> // EXAMPLE: Roots
> // define the ring of polynomials over a quaternion algebra
> A<i,j,k> := QuaternionAlgebra<Rationals() | -1, -1>;
> Ax<x>    := PolynomialRing(A);
> // and a polynomial p
> p        := Ax![ 6+6*i-36*j+24*k, 18-12*i-6*j+30*k,
>              23+5*i-30*j+20*k, 15-10*i-5*j+25*k,
>              16+i-6*j+4*k, 3-2*i-j+5*k, 3 ];
> // find the roots of p
> Zp       :=  Roots( p ); Zp;
[ <-i - j, 0, 2>, <i + j - k, 1, 2>, <-1 + i, 1, 0> ]
```

```
> // EXAMPLE: Roots
> // define the ring of polynomials over a quaternion algebra
> A<i,j,k> := QuaternionAlgebra<Rationals() | -1, -1>;
> Ax<x>    := PolynomialRing(A);
> // and a polynomial p
> p        := Ax![ 6+6*i-36*j+24*k, 18-12*i-6*j+30*k,
>             23+5*i-30*j+20*k, 15-10*i-5*j+25*k,
>             16+i-6*j+4*k, 3-2*i-j+5*k, 3 ];
> // find the roots of p
> Zp       :=  Roots( p ); Zp;
[ <-i - j, 0, 2>, <i + j - k, 1, 2>, <-1 + i, 1, 0> ]
> // verify the roots
> [ Evaluate(p, xi[1]) : xi in Zp ];
[ 0, 0, 0 ]
>
```

```
> // EXAMPLE: Factorization
> // define the ring of polynomials over a quaternion algebra
> K<sqrt7> := QuadraticField(7);
> A<i,j,k> := QuaternionAlgebra<K | -1, -1>;
> Ax<x>    := PolynomialRing(A);
> // take a poly given as a product
> r1 := (x^2 - (3+i)*x - k);
> r2 := (x - j);
> r3 := (x^2 + (1-k)*x + (1-k));
> r4 := (k*x - (1+k));
> p := r1*r2*r3*r4;
```

```
> // EXAMPLE: Factorization
> // define the ring of polynomials over a quaternion algebra
> K<sqrt7> := QuadraticField(7);
> A<i,j,k> := QuaternionAlgebra<K | -1, -1>;
> Ax<x>    := PolynomialRing(A);
> // take a poly given as a product
> r1 := (x^2 - (3+i)*x - k);
> r2 := (x - j);
> r3 := (x^2 + (1-k)*x + (1-k));
> r4 := (k*x - (1+k));
> p := r1*r2*r3*r4;
> // factor it and see that the factors are different
> fp, c := Factorization(p); fp; c;
[
    <x^2 + (-3 + i)*x + -k, 1>,
    <x^2 + (1 + 2/3*i + 4/3*j - 1/3*k)*x + 1/3 + 2/3*i + 2/3*j - k, 1>,
    <x + -1 - 2/3*i + 2/3*j + 1/3*k, 1>,
    <x + -j, 1>
]
k
```

```
> // EXAMPLE: Factorization
> // define the ring of polynomials over a quaternion algebra
> K<sqrt7> := QuadraticField(7);
> A<i,j,k> := QuaternionAlgebra<K | -1, -1>;
> Ax<x>    := PolynomialRing(A);
> // take a poly given as a product
> r1 := (x^2 - (3+i)*x - k);
> r2 := (x - j);
> r3 := (x^2 + (1-k)*x + (1-k));
> r4 := (k*x - (1+k));
> p := r1*r2*r3*r4;
> // factor it and see that the factors are different
> fp, c := Factorization(p); fp; c;
[
    <x^2 + (-3 + i)*x + -k, 1>,
    <x^2 + (1 + 2/3*i + 4/3*j - 1/3*k)*x + 1/3 + 2/3*i + 2/3*j - k, 1>,
    <x + -1 - 2/3*i + 2/3*j + 1/3*k, 1>,
    <x + -j, 1>
]
k
> // yet still, it is a valid factorization, just a different one
> // (we are in a non-commutative ring)
> c*FactorisationToPolynomial(fp) eq p;
true
```

```
> // EXAMPLE: Factorization
> // define the ring of polynomials over a quaternion algebra
> K<sqrt7> := QuadraticField(7);
> A<i,j,k> := QuaternionAlgebra<K | -1, -1>;
> Ax<x>    := PolynomialRing(A);
> // take a poly given as a product
> r1 := (x^2 - (3+i)*x - k);
> r2 := (x - j);
> r3 := (x^2 + (1-k)*x + (1-k));
> r4 := (k*x - (1+k));
> p := r1*r2*r3*r4;
> // factor it and see that the factors are different
> fp, c := Factorization(p); fp; c;
[
    <x^2 + (-3 + i)*x + -k, 1>,
    <x^2 + (1 + 2/3*i + 4/3*j - 1/3*k)*x + 1/3 + 2/3*i + 2/3*j - k, 1>,
    <x + -1 - 2/3*i + 2/3*j + 1/3*k, 1>,
    <x + -j, 1>
]
k
> // yet still, it is a valid factorization, just a different one
> // (we are in a non-commutative ring)
> c*FactorisationToPolynomial(fp) eq p;
true
> // verify that the factors are similar
> test, lambda := IsSimilar(r3, fp[2][1] : Isomorphism := true);
> test;
true
```

Application: Sum of Squares Decomposition in $\mathbb{Q}[x]$

Observation

Let $f \in \mathbb{Q}[x]$ be non-negative everywhere, and $\mathcal{A} = \left(\frac{-1,-1}{\mathbb{Q}}\right)$. Then:

$$f \in \sum_4 \square \iff f = N(P)$$

for some $P \in \mathcal{A}[x]$.

The Constructive Algorithm:

1. Factor f over the division ring $\mathcal{A}$ into $f = P\overline{P}$
2. Extract the coordinates of P.

```
> // EXAMPLE: Sum of squares (s.o.s.)
> // take a rational polynomial
> Qx<x> := PolynomialRing(Rationals());
> p := x^6 + 6*x^5 + 8*x^4 - 26*x^3 + 50*x^2 - 22*x + 9; p;
x^6 + 6*x^5 + 8*x^4 - 26*x^3 + 50*x^2 - 22*x + 9
```

```
> // EXAMPLE: Sum of squares (s.o.s.)
> // take a rational polynomial
> Qx<x> := PolynomialRing(Rationals());
> p := x^6 + 6*x^5 + 8*x^4 - 26*x^3 + 50*x^2 - 22*x + 9; p;
x^6 + 6*x^5 + 8*x^4 - 26*x^3 + 50*x^2 - 22*x + 9
> // factor it over (-1,-1 | QQ)
> A<i,j,k> := QuaternionAlgebra<Rationals() | -1, -1>;
> factors := Factorization( PolynomialRing(A) ! p ); factors;
[
    <$.1^3 + (3 - 434/47841*i - 4952/47841*j + 89251/47841*k)*$.1^2 +
        (-107431/47841 + 8047/15947*i - 25963/47841*j - 234545/47841*k)*$.1 +
        45151/15947 + 6952/47841*i + 15520/47841*j + 44296/47841*k, 1>,
    <$.1^3 + (3 + 434/47841*i + 4952/47841*j - 89251/47841*k)*$.1^2 +
        (-107431/47841 - 8047/15947*i + 25963/47841*j + 234545/47841*k)*$.1 +
        45151/15947 - 6952/47841*i - 15520/47841*j - 44296/47841*k, 1>
]
```

```
> // EXAMPLE: Sum of squares (s.o.s.)
> // take a rational polynomial
> Qx<x> := PolynomialRing(Rationals());
> p := x^6 + 6*x^5 + 8*x^4 - 26*x^3 + 50*x^2 - 22*x + 9; p;
x^6 + 6*x^5 + 8*x^4 - 26*x^3 + 50*x^2 - 22*x + 9
> // factor it over (-1,-1 | QQ)
> A<i,j,k> := QuaternionAlgebra<Rationals() | -1, -1>;
> factors := Factorization( PolynomialRing(A) ! p ); factors;
[
    <$.1^3 + (3 - 434/47841*i - 4952/47841*j + 89251/47841*k)*$.1^2 +
        (-107431/47841 + 8047/15947*i - 25963/47841*j - 234545/47841*k)*$.1 +
        45151/15947 + 6952/47841*i + 15520/47841*j + 44296/47841*k, 1>,
    <$.1^3 + (3 + 434/47841*i + 4952/47841*j - 89251/47841*k)*$.1^2 +
        (-107431/47841 - 8047/15947*i + 25963/47841*j + 234545/47841*k)*$.1 +
        45151/15947 - 6952/47841*i - 15520/47841*j - 44296/47841*k, 1>
]
> // the coordinates of any of the two factors form the decomposition into s.o\
.s.
> C := Coordinates(factors[1][1]); C;
[
    x^3 + 3*x^2 - 107431/47841*x + 45151/15947,
    -434/47841*x^2 + 8047/15947*x + 6952/47841,
    -4952/47841*x^2 - 25963/47841*x + 15520/47841,
    89251/47841*x^2 - 234545/47841*x + 44296/47841
]
```

```
> // EXAMPLE: Sum of squares (s.o.s.)
> // take a rational polynomial
> Qx<x> := PolynomialRing(Rationals());
> p := x^6 + 6*x^5 + 8*x^4 - 26*x^3 + 50*x^2 - 22*x + 9; p;
x^6 + 6*x^5 + 8*x^4 - 26*x^3 + 50*x^2 - 22*x + 9
> // factor it over (-1,-1 | QQ)
> A<i,j,k> := QuaternionAlgebra<Rationals() | -1, -1>;
> factors := Factorization( PolynomialRing(A) ! p ); factors;
[
    <$.1^3 + (3 - 434/47841*i - 4952/47841*j + 89251/47841*k)*$.1^2 +
        (-107431/47841 + 8047/15947*i - 25963/47841*j - 234545/47841*k)*$.1 +
        45151/15947 + 6952/47841*i + 15520/47841*j + 44296/47841*k, 1>,
    <$.1^3 + (3 + 434/47841*i + 4952/47841*j - 89251/47841*k)*$.1^2 +
        (-107431/47841 - 8047/15947*i + 25963/47841*j + 234545/47841*k)*$.1 +
        45151/15947 - 6952/47841*i - 15520/47841*j - 44296/47841*k, 1>
]
> // the coordinates of any of the two factors form the decomposition into s.o\
.s.
> C := Coordinates(factors[1][1]); C;
[
    x^3 + 3*x^2 - 107431/47841*x + 45151/15947,
    -434/47841*x^2 + 8047/15947*x + 6952/47841,
    -4952/47841*x^2 - 25963/47841*x + 15520/47841,
    89251/47841*x^2 - 234545/47841*x + 44296/47841
]
> // verification
> &+ [ q^2 : q in C ] eq p;
true
```

qPoly: Factorization and 60+ other tools

Thank you

http://www.pkoprowski.eu/qpoly
przemyslaw.koprowski@us.edu.pl