

Approximate Polynomial Algebra Package in Maple 2021

[New PolynomialTools Subpackage](#)

[GCD and SylvesterMatrix](#)

[CoefficientVector and FromCoefficientVector](#) [Factor and RuppertMatrix](#)

[Divide and ConvolutionMatrix](#)

[References](#)

New PolynomialTools Subpackage

- New in Maple 2021 is the [PolynomialTools:-Approximate](#) subpackage. It includes three main commands for computing approximate operations on polynomials in many variables: **Factor**, **GCD**, and **Divide**, as well as commands for computing the matrices used for these operations.
- These new commands are useful for cases where no exact answer is possible, and a solution is instead found for a nearby problem which does have a solution. Typically, this assumes that some amount of error or noise was introduced into the coefficients of the input polynomial(s) that destroyed algebraic structure, and which these commands attempt to recover.
- The [SNAP](#) package has provided the gcd and division with remainder operations for univariate polynomials but those commands typically require the user to provide a numerical tolerance. These new commands do not take a tolerance as input, but instead always return an approximate solution, even if it has very large backward error following the model in the papers of Gao et al. and Kaltofen et al in the references below.
- This new package supports univariate computations of quotients and gcd under that model (univariate inputs to **Factor** are just sent to the top-level command **factor**).

```
> factor( x^4-3.0*x^2-4.0 );
```

$$(x + 2.000000000000000) (x - 2.000000000000000) (x^2 + 0.9999999999999996) \quad (1)$$

```
> factor( x^4-3.0*x^2-4.0*I );
```

$$(x + 1.85321006568802 + 0.291797565955056 I) (x + 0.627404403732890 - 0.861903714978381 I) (x - 0.627404403732890 + 0.861903714978381 I) (x - 1.85321006568802 - 0.291797565955056 I) \quad (2)$$

```
> SNAP:-QRGCD( x^4-3.0*x^2-4.0, x^2-4.0, x, 10e-10);
```

$$0, 0.2425356250, -0.970142500145332 + 0.242535625036333 x^2 \quad (3)$$

```
> SNAP:-Quotient( x^4-3.0*x^2-4.0, x^2-4.0, x );
```

...

$$1.0000000000000000 x^2 + 1.0000000000000000 \quad (4)$$

- New functionality is described below.

```
> with(PolynomialTools):
```

```
> with(Approximate);
```

[ConvolutionMatrix, Divide, Factor, GCD, RuppertMatrix, SylvesterMatrix] (5)

CoefficientVector and FromCoefficientVector

- The commands [CoefficientVector](#) and [FromCoefficientVector](#) in the main PolynomialTools package were both enhanced to work with multivariate polynomials.

```
> v := CoefficientVector(y^3-y^2*x+(y-2)*x^2+x^3, [x,y]);
```

$$v := \begin{bmatrix} 0 \\ 0 \\ 0 \\ -2 \\ 0 \\ 0 \\ 1 \\ 1 \\ -1 \\ 1 \end{bmatrix} \quad (6)$$

```
> FromCoefficientVector(v, [a, b]);
```

$$a^3 + a^2 b - a b^2 + b^3 - 2 a^2 \quad (7)$$

Divide and ConvolutionMatrix

- Approximate division using a LeastSquares computation on the matrix representing polynomial division.

```
> f := x^2+y^2-1;
```

$$f := x^2 + y^2 - 1 \quad (8)$$

```
> ConvolutionMatrix(f, [x,y], 2);
```

$$\begin{bmatrix}
-1 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 \\
1 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 \\
1 & 0 & 0 & 0 & 0 & -1 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots
\end{bmatrix}$$

(9)

15 × 6 Matrix

```
> infolevel[PolynomialTools] := 1;
```

$$\text{infolevel}_{\text{PolynomialTools}} := 1$$

(10)

```
> d := sqrt(2.0)*(x^2+y^2-1);
```

$$d := 1.414213562 x^2 + 1.414213562 y^2 - 1.414213562$$

(11)

```
> G := sort( (x^2-y^3+1), [x,y]); F := sort( expand( d*G ), [x,y]);
```

$$G := -y^3 + x^2 + 1$$

$$F := -1.414213562 x^2 y^3 - 1.414213562 y^5 + 1.414213562 x^4 + 1.414213562 x^2 y^2 + 1.414213562 y^3 + 1.414213562 y^2 - 1.414213562$$

(12)

```
> ad_8 := Divide( expand( F + 10^(-8)*x*y ), expand(G+10^(-8)*x^2*y), [x,y] );
```

Divide: computed approximate quotient has backward error 2.291288e-08

Divide: computed approximate quotient has backward error 2.291288e-08

$$ad_8 := -1.41421356376777 + 3.10292232917631 \times 10^{-16} x + 4.71404526525276 \times 10^{-9} y + 1.41421356730330 x^2 + 3.33333326578258 \times 10^{-9} xy + 1.41421356200000 y^2$$

(13)

```
> ad_8 := sort( tcoeff(d)*fnormal(expand(ad_8)/(tcoeff(ad_8))), [x,y]);
```

$$ad_8 := 1.414213565 x^2 + 1.414213560 y^2 - 1.414213562$$

(14)

```
> ilog10(norm(expand(d-ad_8),2) / norm(d,2));
```

-9

(15)

GCD and SylvesterMatrix

- Approximate GCD computation using a low-rank approximation of a generalized Sylvester matrix.

```
> d := sqrt(2.0)*(x^2+y^2-1);
```

$$d := 1.414213562 x^2 + 1.414213562 y^2 - 1.414213562 \quad (16)$$

```
> F := sort( expand( d*(x^3-y^3+1) ), [x,y] );
```

$$F := 1.414213562 x^5 + 1.414213562 x^3 y^2 - 1.414213562 x^2 y^3 - 1.414213562 y^5 - 1.414213562 x^3 + 1.414213562 y^3 + 1.414213562 x^2 + 1.414213562 y^2 - 1.414213562 \quad (17)$$

```
> G := sort( expand( d*(x^2-y^3+1) ), [x,y] );
```

$$G := -1.414213562 x^2 y^3 - 1.414213562 y^5 + 1.414213562 x^4 + 1.414213562 x^2 y^2 + 1.414213562 y^3 + 1.414213562 y^2 - 1.414213562 \quad (18)$$

```
> S1 := SylvesterMatrix(F, G, [x,y], 3);
```

$$S1 := \begin{pmatrix} -1.414213562 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1.414213562 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1.414213562 & 0 & 0 & 0 \\ 1.414213562 & 0 & 0 & -1.414213562 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1.414213562 & 0 \\ 1.414213562 & 0 & 0 & 0 & 0 & -1.414213562 \\ -1.414213562 & 1.414213562 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.414213562 & 0 & 0 & 0 \\ 0 & 1.414213562 & 0 & 0 & 0 & 0 \\ 1.414213562 & 0 & 1.414213562 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

- Maximal rank, means $\text{degree}(\text{gcd}(F,G)) < 3$

```
> min(upperbound(S1)) - LinearAlgebra:-Rank(S1);
```

$$0 \quad (20)$$

```
> ad_8 := GCD( expand( F + 10^(-8)*x*y ), expand(G+10^(-8)*x^2*y), [x,y] );
```

GCD: approximate GCD inputs of total degrees 5 and 5 in [x, y]

GCD: approximate GCD determined with an error of 7.484964e-10

$$ad_8 := 0.569494797046587 + 3.40008146937111 \times 10^{-10} x - 1.02787941763836 \times 10^{-9} y \quad (21)$$

$$- 0.569494797402477 x^2 - 1.49456610221387 \times 10^{-10} xy - 0.569494797883116 y^2$$

```
> ad_8 := sort( tcoeff(d)*fnormal(expand(ad_8)/(tcoeff(ad_8))), [x,
y]);
```

$$ad_8 := 1.414213563 x^2 + 1.414213563 y^2 - 1.414213562 \quad (22)$$

```
> ilog10(norm(expand(d-ad_8),2) / norm(d,2));
```

$$-10 \quad (23)$$

Factor and RuppertMatrix

- Approximate factorization computation using a low-rank approximation of a Ruppert matrix.

```
> F := sort( expand( (x^2+y^2-1)*(x^3-y^3+1) ), [x,y]);
```

$$F := x^5 + x^3 y^2 - x^2 y^3 - y^5 - x^3 + y^3 + x^2 + y^2 - 1 \quad (24)$$

```
> R1 := RuppertMatrix( F, [x,y]);
```

$$R1 := \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & \dots \\ -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & \dots \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & \dots \\ 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ -3 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & -1 & 0 & \dots \\ 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & \dots \\ 0 & -3 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots \end{bmatrix} \quad (25)$$

45 × 30 Matrix

- The rank deficiency of the Ruppert matrix is equal to the number of factors.

```
> min(upperbound(R1)) - LinearAlgebra:-Rank(R1);
```

$$2 \quad (26)$$

```
> aF_8 := Factor( expand( F + 10^(-8)*x*y ), [x,y] );
```

```
Factor: approximate factorization input of degree [5, 5] in [x,
y]
```

```
Factor: factorization square-free input of degree [5, 5] in [x,
y]
```

Factor: polynomial has 2 approximate factors at numerical tolerance 10e-9

$$\begin{aligned}
 aF_8 := & -3.51077852332865 \left(-0.553434072558778 + 1.77801496220546 \times 10^{-9} x \right. \\
 & + 7.90820224621849 \times 10^{-10} y + 0.553434076452427 x^2 + 3.77682406621234 \times 10^{-9} xy \\
 & + 0.553434072372128 y^2 \left. \right) \left(-0.514672153977448 - 2.34835389322383 \times 10^{-9} x \right. \\
 & - 5.64898108177015 \times 10^{-9} y - 1.17024438448444 \times 10^{-9} x^2 - 1.49078198461690 \\
 & \times 10^{-9} xy + 2.96989982214706 \times 10^{-9} y^2 - 0.514672156404694 x^3 - 5.26728919650195 \\
 & \times 10^{-10} yx^2 - 2.25599648375540 \times 10^{-9} y^2 x + 0.514672152857353 y^3 \left. \right)
 \end{aligned} \tag{27}$$

> sort(fnormal(expand(aF_8)), [x,y]);

$$\begin{aligned}
 & 1.000000007 x^5 + 1.000000004 x^3 y^2 - 0.9999999990 x^2 y^3 - 0.9999999926 y^5 \\
 & - 0.9999999953 x^3 + 1.000000004 y^3 + 0.9999999999 x^2 + 1.000000001 y^2 \\
 & - 0.9999999951
 \end{aligned} \tag{28}$$

> ilog10(norm(expand(F-aF_8),2) / norm(F,2));

$$-9 \tag{29}$$

References

Gao, S.; Kaltofen, E.; May, J.; Yang, Z.; and Zhi, L. "Approximate factorization of multivariate polynomials via differential equations." **Proceedings of the 2004 International Symposium on Symbolic and Algebraic Computation (ISSAC 2004)**, pp. 167-174. Ed. J. Gutierrez. ACM Press, 2004.

Kaltofen, E.; May, J.; Yang, Z.; and Zhi, L. "Approximate factorization of multivariate polynomials using singular value decomposition." **Journal of Symbolic Computation** Vol. **43(5)**, (2008): 359-376.