

- > #1. The "Package of Quillen-Suslin" folder contains a Maple package called QuillenSuslin which is used to complete Quillen-Suslin algorithm. Beside it contains a Maple package called Involutive which is used to deal with module operations including the computation of the syzygy module and the Groebner basis for submodules. Refer to <https://who.rocq.inria.fr/Alban.Quadrat/QuillenSuslin/index.html> for more details. #
- #2. Firstly, you need to copy files "Involutive.hdb", "Involutive.ind", "Involutive.lib", "QuillenSuslin.hdb", "QuillenSuslin.ind", "QuillenSuslin.lib" to folder "lib" in the installation directory of Maple, then restart Maple. #
- #3. If 2 doesn't work, use the command line "libname" in maple. Then copy the six file above to this folder. #

```
with(Groebner) :
with(PolynomialIdeals) :
with(LinearAlgebra) :
with(combinat) :
with(Groebner) :
with(ListTools) :
with(Involutive) :
with(QuillenSuslin) :
```

```
> RandMat := proc(n, m, c, d, p, var)
    #Randomly generate a matrix
    #n: row dimension of the matrix
    #m: column dimension of the matrix
    #c: bound of the coefficients
    #d: degree of the polynomials
    #p: non-zero probability
    local M, i, j, roll, a;
    roll := rand(0.0..1.0);
    M := Matrix(n, m);
    for i from 1 to n do
        for j from 1 to m do
            a := roll();
            if (a ≤ p) then
                M[i, j] := randpoly(var, coeffs = rand(-c..c), degree = d)
                + randpoly(var, coeffs = rand(-c..c), degree = 0);
            end if;
        end do;
    end do;
    return M;
end proc;
```

```
RandMat := proc(n, m, c, d, p, var)
```

```
    local M, i, j, roll, a;
    roll := rand(0. . . 1.0);
    M := Matrix(n, m);
    for i to n do
```

(1)

```

    for j to m do
        a := roll( );
        if a <= p then
            M[i, j] := randpoly(var, coeffs=rand( - c.. c), degree = d)
            + randpoly(var, coeffs=rand( - c.. c), degree = 0)
        end if
    end do
end do;
return M
end proc

```

```

end proc

```

```

>

```

```

> IsZLP := proc(M, var)
    #Check if a matrix is ZLP
    #M: a matrix
    #var: variables
    local S, G, n;
    n := RowDimension(M);
    S := MaxMinors(M, n);
    G := Basis(S, tdeg(op(var)));
    if (G[1] = 1) then
        return true;
    else
        return false;
    end if;
end proc;

```

```

IsZLP := proc(M, var)

```

(2)

```

    local S, G, n;
    n := LinearAlgebra:-RowDimension(M);
    S := MaxMinors(M, n);
    G := Groebner:-Basis(S, tdeg(op(var)));
    if G[1] = 1 then return true else return false end if
end proc

```

```

end proc

```

```

> ZLP_Syzygy := proc(F)
    #Computing the generators of the Syz(F) by new method;
    #F: a matrix
    local M, m, l, P, H, Q, w, v, t, p, i, j, k, p0, st;
    #compute all minors of F;
    st := time[real]( );
    M := MaxMinors(F);
    print("Computing Minor using:", time[real]( ) - st, "s");
    st := time[real]( );
    #construct Generators of unimodular matrix
    m := ColumnDimension(F);

```

```

    l := RowDimension(F) :
    P := choose(m, l + 1) :
    H := choose(m, l) :
    #Convert to Table;
    Q := table( ) :
    for i from 1 to nops(H) do
        Q[op(H[i])] := i;
    end do;
    w := [ ] :
    for p in P do
        v := [seq(0, i = 1..m)];
        t := 0;
        for j from 1 to l + 1 do
            p0 := [seq(0, i = 1..l)];
            for k from 1 to j - 1 do
                p0[k] := p[k];
            end do;
            for k from j to l do
                p0[k] := p[k + 1];
            end do;
            if (M[Q[op(p0)]] ≠ 0) then
                t := 1;
            end if;
            v[p[j]] := (-1)j · M[Q[op(p0)]];
        end do;
        if (t = 1) then
            w := [op(w), v];
        end if;
    end do;
    print("Other using:", time[real]() - st, "s");
    return w;
end proc;
ZLP_Syzygy := proc(F)
    local M, m, l, P, H, Q, w, v, t, p, i, j, k, p0, st;
    st := time[real]();
    M := MaxMinors(F);
    print("Computing Minor using:", time[real]() - st, "s");
    st := time[real]();
    m := LinearAlgebra:-ColumnDimension(F);
    l := LinearAlgebra:-RowDimension(F);
    P := combinat:-choose(m, l + 1);
    H := combinat:-choose(m, l);
    Q := table( );
    for i to nops(H) do Q[op(H[i])] := i end do;
    w := [ ];

```

(3)

```

for p in P do
  v := [seq(0, i=1..m)];
  t := 0;
  for j to l+1 do
    p0 := [seq(0, i=1..l)];
    for k to j-1 do p0[k] := p[k] end do;
    for k from j to l do p0[k] := p[k+1] end do;
    if M[Q[op(p0)]] <> 0 then t := 1 end if;
    v[p[j]] := (-1)^j * M[Q[op(p0)]]
  end do;
  if t=1 then w := [op(w), v] end if
end do;
print("Other using:", time[real]() - st, "s");
return w
end proc

```

## Exampe 1

```

> var := [x, y, z];
m := 6;
n := 2;
d := 1;
c := 2;
M := [[2 x + y - 2 z - 4, 2 x + y - 2 z, 2 x + 2 y + z + 2, -2 x - y - z - 1, 2 x
- 2 y + z - 1, -2 x - 2 y + 2 z + 3],
[-2 x + y + 1, 2 x - z + 1, 2 x + 2 y - 2 z - 2, 2 y - 2, x - 2 y + 2 z + 2,
2 x + 2 y - 2 z + 1]];
var := [x, y, z]
m := 6
n := 2
d := 1
c := 2
M := [[2 x + y - 2 z - 4, 2 x + y - 2 z, 2 x + 2 y + z + 2, -2 x - y - z - 1,
2 x - 2 y + z - 1, -2 x - 2 y + 2 z + 3],
[-2 x + y + 1, 2 x - z + 1, 2 x + 2 y - 2 z - 2, 2 y - 2, x - 2 y + 2 z
+ 2, 2 x + 2 y - 2 z + 1]]

```

(1.1)

```

> IsZLP(M, var);
true

```

(1.2)

```

> S := ZLP_Syzygy(M) :
"Computing Minor using:", 0., "s"

```

```
"Other using:", 0.001, "s" (1.3)
```

```
> st := time[real]():
G := SyzygyModule(M%T, var):
print("Computing Syzygy using:", time[real]() - st, "s");
"Computing Syzygy using:", 0.326, "s" (1.4)
```

## Example 2

```
> var := [x, y, z];
m := 6;
n := 2;
d := 2;
c := 2;
M := [[0, 2 x^2 + x z + y^2 - 2 y z - 2 y, -x^2 + 2 y z + z^2 - 2 z + 3, 0, 2 x^2 + x z
+ 2 y^2 - y z + 2 z^2 + 1, -x^2 + y^2 + x + 3],
[2 x^2 + x y - y^2 + 2 y z - z, x^2 + 2 x y + 2 y z + 2 z^2 - z - 2, -2 x y - 2 y^2 + y z
- 1, 0, 2 x z - y^2 + 2 y z + z^2 - x + 1, 2 x y - 2 x z - z + 4]];

var := [x, y, z]
m := 6
n := 2
d := 2
c := 2

M := [[0, 2 x^2 + x z + y^2 - 2 z y - 2 y, -x^2 + 2 z y + z^2 - 2 z + 3, 0, 2 x^2
+ x z + 2 y^2 - z y + 2 z^2 + 1, -x^2 + y^2 + x + 3],
[2 x^2 + x y - y^2 + 2 z y - z, x^2 + 2 x y + 2 z y + 2 z^2 - z - 2, -2 x y
- 2 y^2 + z y - 1, 0, 2 x z - y^2 + 2 z y + z^2 - x + 1, 2 x y - 2 x z - z + 4
]] (2.1)

> IsZLP(M, var);
true (2.2)
```

```
> S := ZLP_Syzygy(M):
"Computing Minor using:", 0., "s"
"Other using:", 0.002, "s" (2.3)
```

```
> st := time[real]():
G := SyzygyModule(M%T, var):
print("Computing Syzygy using:", time[real]() - st, "s");
"Computing Syzygy using:", 6.968, "s" (2.4)
```

## Example 3

```

> var := [x, y, z];
m := 6;
n := 2;
d := 3;
c := 2;
M := [[ -x^2 z - y z^2 - y^2 + x - 2 z, -2 x y z - 2 y^3 - y^2 z + 2 y z^2 - 2 y z - 1, x^2 z
+ x y z + y^2 + z, 0, 2 x^2 y - y^3 + y z^2 + 2 x z + 2 x + z + 1, x z^2 - 2 y^3 + z^3 - 1
],
[x^2 y + 2 y^3 - x z - y + 1, -x^2 z - x y z + y + 2, 0, y^3 + 2 y z^2 - 2 x y - 2 z^2
- 2 z + 1, y^2 z - x^2 + z + 1, x^3 + x y z - y^3 - 2 y z - 2 x]];
var := [x, y, z]
m := 6
n := 2
d := 3
c := 2
M := [[ -x^2 z - y z^2 - y^2 + x - 2 z, -2 x y z - 2 y^3 - y^2 z + 2 y z^2 - 2 z y
- 1, x^2 z + x y z + y^2 + z, 0, 2 x^2 y - y^3 + y z^2 + 2 x z + 2 x + z + 1,
x z^2 - 2 y^3 + z^3 - 1],
[x^2 y + 2 y^3 - x z - y + 1, -x^2 z - x y z + y + 2, 0, y^3 + 2 y z^2 - 2 x y
- 2 z^2 - 2 z + 1, y^2 z - x^2 + z + 1, x^3 + x y z - y^3 - 2 z y - 2 x]]
IsZLP(M, var);
true
S := ZLP_Syzygy(M) :
"Computing Minor using:", 0.002, "s"
"Other using:", 0., "s"
st := time[real]():
G := SyzygyModule(M^%T, var) :
print("Computing Syzygy using:", time[real]() - st, "s");
"Computing Syzygy using:", 105.661, "s"

```

## Example 4

```

> var := [x, y, z];
m := 6;
n := 2;
d := 4;
c := 2;
M := [[ y z^3 + x^2 z + 2 y z - 2 z^2 - z, -x y z^2 - 2 z^4 + 2 x y - 2 x z + y z - 2, -2 y z^3
+ 2 x^2 z - y^3 + x y + 1, 2 x^2 y^2 + 2 y^4 - 2 x^3 + 2 z^2 - 2, 0, -2 x^3 y + x y^2 z
- x z^3 + y^2 z],

```

```

[ 0,  -x2 y z - z4 + 2 x3 - 3,  -2 y2 z2 - 2 z4 - 2 y3 - y2 + z + 1,  2 z4 - 2 x2 z
- x y z - 2 y2 z + 2,  0,  2 x2 z2 + 2 x y2 z - 2 y2 z2 + 2 x2 z + x + 2 ]]
var := [x, y, z]
m := 6
n := 2
d := 4
c := 2
M := [[ y z3 + x2 z + 2 z y - 2 z2 - z,  -x y z2 - 2 z4 + 2 x y - 2 x z + z y
- 2,  -2 y z3 + 2 x2 z - y3 + x y + 1,  2 x2 y2 + 2 y4 - 2 x3 + 2 z2 - 2,  0,
-2 x3 y + x y2 z - x z3 + y2 z],
[ 0,  -x2 y z - z4 + 2 x3 - 3,  -2 y2 z2 - 2 z4 - 2 y3 - y2 + z + 1,  2 z4
- 2 x2 z - x y z - 2 y2 z + 2,  0,  2 x2 z2 + 2 x y2 z - 2 y2 z2 + 2 x2 z + x
+ 2]]
> IsZLP(M, var);
true
> S := ZLP_Syzygy(M) :
"Computing Minor using:", 0.001, "s"
"Other using:", 0., "s"
> st := time[real]( ) :
G := SyzygyModule(M%T, var) :
print("Computing Syzygy using:", time[real]( ) - st, "s");
>

```

(4.1)

(4.2)

(4.3)

## Example 5

```

> var := [x, y, z];
m := 8;
n := 4;
d := 2;
c := 2;
M := [[ 2 x y - 2 y z + x + y - 1,  2 x z + 2 y2 + z2 + y - 4,  x2 - x y + x z - z2
- 4,  x2 + 2 x y + x z - y2 - z2 + x - 1,  x y + z2 + y - 4,  x y - 2 x z + y z
- x - 1,  -2 x y - x z - 2 x - z - 2,  2 x y - 2 x z + y2 - 1],
[ x2 + x y - 2 x z - 2 y2 + 2 y z + y - 2,  0,  -x2 + y2 - 2 y z - 2,  x y + 2 x
- 2 z + 2,  -2 x2 + x y - x z + z2 - z,  x2 + x y - 2 x z + 2 x - 3,  2 x y + y2
- 2 y z + 2 z2 + x + 2,  x2 - x y + 2 x z - y2 - y z - 2],
[ 0,  -x y + 2 x z - 2 y z - 2 x + y - 2 z - 1,  x y + 2 y z - 2 y + 2,  0,  -x y
+ 2 x z - 2 y2 + y z + 2 z2 + y,  x2 - x y + 2 y2 + 2 x - z + 4,  -x y + 3,  -y z
+ 2 z2 + y - 1],
[ -x2 + x y + x z - 2 z2 - x + 1,  y z + 2 z2 - x + y - z,  2 x y + x z + z2,

```

```

-2 x^2 - x z + y^2 - y z + 2 x - 2 z + 1, 2 x z - 2 x + 2 y + 1, -2 x^2 + x z
- 2 y^2 + 2 y + z + 1, 0, -2 x^2 + 2 x y + 2 y^2 - z^2 + 2 x + 2 z - 2]];

```

```

var := [x, y, z]

```

```

m := 8

```

```

n := 4

```

```

d := 2

```

```

c := 2

```

```

M := [[2 x y - 2 y z + x + y - 1, 2 x z + 2 y^2 + z^2 + y - 4, x^2 - x y + x z - z^2
- 4, x^2 + 2 x y + x z - y^2 - z^2 + x - 1, x y + z^2 + y - 4, x y - 2 x z + y z
- x - 1, -2 x y - x z - 2 x - z - 2, 2 x y - 2 x z + y^2 - 1],
[x^2 + x y - 2 x z - 2 y^2 + 2 y z + y - 2, 0, -x^2 + y^2 - 2 y z - 2, x y + 2 x
- 2 z + 2, -2 x^2 + x y - x z + z^2 - z, x^2 + x y - 2 x z + 2 x - 3, 2 x y
+ y^2 - 2 y z + 2 z^2 + x + 2, x^2 - x y + 2 x z - y^2 - y z - 2],
[0, -x y + 2 x z - 2 y z - 2 x + y - 2 z - 1, x y + 2 y z - 2 y + 2, 0,
-x y + 2 x z - 2 y^2 + y z + 2 z^2 + y, x^2 - x y + 2 y^2 + 2 x - z + 4, -x y
+ 3, -y z + 2 z^2 + y - 1],
[-x^2 + x y + x z - 2 z^2 - x + 1, y z + 2 z^2 - x + y - z, 2 x y + x z + z^2,
-2 x^2 - x z + y^2 - y z + 2 x - 2 z + 1, 2 x z - 2 x + 2 y + 1, -2 x^2 + x z
- 2 y^2 + 2 y + z + 1, 0, -2 x^2 + 2 x y + 2 y^2 - z^2 + 2 x + 2 z - 2]]

```

(5.1)

```

> IsZLP(M, var);

```

```

true

```

(5.2)

```

> S := ZLP_Syzygy(M) :

```

```

"Computing Minor using:", 0.004, "s"

```

```

"Other using:", 0.003, "s"

```

(5.3)

```

> st := time[real]( ) :

```

```

SyzygyModule(M^%T, var) :

```

```

print("Computing Syzygy using:", time[real]( ) - st, "s");

```

```

"Computing Syzygy using:", 1064.177, "s"

```

(5.4)

## Example 6

```

> var := [x, y, z];

```

```

m := 8;

```

```

n := 4;

```

```

d := 3;

```

```

c := 2;

```

```

M := [[-2 x^2 y + 2 x y z + 2 y^2 z - 2 y z^2 + z^3 + z y - 1, -2 x^2 y - x y^2 - 2 x z^2
+ 2 z^3 + x z + z - 1, 0, 0, -2 x y z + y^3 - 2 z^3 + 2 x y - 2, x^3 + y^2 z - y^2

```



```

+ 1, 0, -x^2 z + 2 x z^2 + 2 y z^2 - z^3 - x y + 2 z - 1],
[-2 x^3 + y z^2 - z^2 - 2 z - 4, -x y z - 2 y^2 z + y + 2, -2 x^3 - 2 x y^2 - x^2
+ 2 x y - z + 2, x^2 + 2 z y + 2 x - z - 2, 0, -2 x^3 + 2 x z - z^2 + 1, 0,
-2 x y z - y z^2 - x + 2 y - 1],
[0, 0, 2 x y z - y^2 z + 2 y z^2 + y^2 + z y + 2, 0, -2 x^2 y - 2 z^3 - x z - y^2
- 1, 2 x^3 + x y z + y^3 + y^2 z + 2 x^2 - x y + 1, -x^2 z - y z^2 - y^2 + x - 2 z,
-2 x y z - 2 y^3 - y^2 z + 2 y z^2 - 2 z y - 1],
[x^2 z + x y z + y^2 + z, 0, 2 x^2 y - y^3 + y z^2 + 2 x z + 2 x + z + 1, x z^2 - 2 y^3
+ z^3 - 1, x^2 y + 2 y^3 - x z - y + 1, -x^2 z - x y z + y + 2, 0, y^3 + 2 y z^2
- 2 x y - 2 z^2 - 2 z + 1]];

```

```
var := [x, y, z]
```

```
m := 8
```

```
n := 4
```

```
d := 3
```

```
c := 2
```

```

M := [[ -2 x^2 y + 2 x y z + 2 y^2 z - 2 y z^2 + z^3 + z y - 1, -2 x^2 y - x y^2
- 2 x z^2 + 2 z^3 + x z + z - 1, 0, 0, -2 x y z + y^3 - 2 z^3 + 2 x y - 2, x^3
+ y^2 z - y^2 + 1, 0, -x^2 z + 2 x z^2 + 2 y z^2 - z^3 - x y + 2 z - 1],
[ -2 x^3 + y z^2 - z^2 - 2 z - 4, -x y z - 2 y^2 z + y + 2, -2 x^3 - 2 x y^2 - x^2
+ 2 x y - z + 2, x^2 + 2 z y + 2 x - z - 2, 0, -2 x^3 + 2 x z - z^2 + 1, 0,
-2 x y z - y z^2 - x + 2 y - 1],
[0, 0, 2 x y z - y^2 z + 2 y z^2 + y^2 + z y + 2, 0, -2 x^2 y - 2 z^3 - x z - y^2
- 1, 2 x^3 + x y z + y^3 + y^2 z + 2 x^2 - x y + 1, -x^2 z - y z^2 - y^2 + x - 2 z,
-2 x y z - 2 y^3 - y^2 z + 2 y z^2 - 2 z y - 1],
[x^2 z + x y z + y^2 + z, 0, 2 x^2 y - y^3 + y z^2 + 2 x z + 2 x + z + 1, x z^2
- 2 y^3 + z^3 - 1, x^2 y + 2 y^3 - x z - y + 1, -x^2 z - x y z + y + 2, 0, y^3
+ 2 y z^2 - 2 x y - 2 z^2 - 2 z + 1]]

```

(6.1)

```
> IsZLP(M, var);
```

```
true
```

(6.2)

```
> S := ZLP_Syzygy(M) :
```

```
"Computing Minor using:", 0.004, "s"
```

```
"Other using:", 0.003, "s"
```

(6.3)

```
> st := time[real]( ) :
```

```
SyzygyModule(M%T, var) :
```

```
print("Computing Syzygy using:", time[real]( ) - st, "s");
```

## Example 7

```

> var := [x, y];
m := 10;
n := 6;
d := 2;
c := 2;
M := [[2 x^2 - y^2 + 2 x + 1, -y^2 + 2 x - y - 1, 0, 2 x^2 - x y - y^2 + 2 x - y - 3,
-x^2 - x y + 2 y^2 + 2 x - 2 y - 1, 2 x^2 + 2 x y + y^2 - x + 2 y + 1, 2 x y - 2 y^2
+ x - 1, x^2 + 2 x y - y^2 + 2 x + y - 2, 2 x^2 - 2 x y - 2 y^2 - y - 1, x^2 - y - 2
],
[-2 x^2 - 2 x y + y^2 + 2 x - 2 y - 2, 0, 2 x y - 2 y^2 + 2 x - y - 2, 0, x^2
- x y - y + 3, 0, -x^2 - 2 x y - y^2 + x - 2 y - 3, 0, -x^2 - 2 x y + y^2 + x
+ 2, -x^2 + x y + 2 y^2 - x - 2 y - 3],
[2 x^2 - 2 x y - 2 y^2 - 2 x - y - 2, 2 x^2 - 2 x y - x + 1, 0, 0, -2 x^2 - 2 x y
- y^2 + x + 2 y - 3, 2 x^2 + x y - y^2 + x - 2 y + 1, x^2 + x y + y^2 - 2 x, -x^2
+ 2 y^2 - x - y - 2, -x y - x + 2 y, 0],
[0, -x y - y^2 - x - 3, x^2 - x y - 2 y^2 - 2 y - 1, 0, x^2 - x y - 2 y^2 - 2 x
- 2, 2 x^2 - x y - y^2 - 2 x + 2 y, 0, 2 x^2 - x y + y^2 - 2 x + 1, -2 x y + 2 y^2
- y + 1, 0],
[x^2 - x y - y^2 + x - 2 y + 4, 2 x y + 2 y^2 + 2 x + y, -2 x^2 - 2 y^2 - 2 x + 3,
0, -2 x^2 + 2 x y + y^2 - x - 2 y + 2, x^2 - 2 y^2 + x, -2 x^2 + 2 x y + 2 y^2 - x
+ 2 y - 3, 2 y^2 - x + y - 1, x^2 + x y - y^2 - 2 x + 2 y + 3, 0],
[0, x^2 + 2 y^2 + x + y, -x^2 - x y - x + y + 3, -x^2 - 2 y^2 - 2 x + 2 y + 1,
-2 x y - y^2 - 2 x - 2 y + 1, -2 x^2 + 2 x y + 2 y^2 - 2 y, -x^2 + 2 x y - y^2 + y,
-x^2 + x y + 2 y^2 - 2 y - 2, -2 x y - y^2 - x - y + 2, 0]];
var := [x, y]
m := 10
n := 6
d := 2
c := 2
M := [[2 x^2 - y^2 + 2 x + 1, -y^2 + 2 x - y - 1, 0, 2 x^2 - x y - y^2 + 2 x - y - 3,
-3, -x^2 - x y + 2 y^2 + 2 x - 2 y - 1, 2 x^2 + 2 x y + y^2 - x + 2 y + 1,
2 x y - 2 y^2 + x - 1, x^2 + 2 x y - y^2 + 2 x + y - 2, 2 x^2 - 2 x y - 2 y^2 - y
- 1, x^2 - y - 2],
[-2 x^2 - 2 x y + y^2 + 2 x - 2 y - 2, 0, 2 x y - 2 y^2 + 2 x - y - 2, 0, x^2
- x y - y + 3, 0, -x^2 - 2 x y - y^2 + x - 2 y - 3, 0, -x^2 - 2 x y + y^2 + x
+ 2, -x^2 + x y + 2 y^2 - x - 2 y - 3],
[2 x^2 - 2 x y - 2 y^2 - 2 x - y - 2, 2 x^2 - 2 x y - x + 1, 0, 0, -2 x^2
- 2 x y - y^2 + x + 2 y - 3, 2 x^2 + x y - y^2 + x - 2 y + 1, x^2 + x y + y^2
- 2 x, -x^2 + 2 y^2 - x - y - 2, -x y - x + 2 y, 0],
[0, -x y - y^2 - x - 3, x^2 - x y - 2 y^2 - 2 y - 1, 0, x^2 - x y - 2 y^2 - 2 x

```

```

- 2, 2 x^2 - x y - y^2 - 2 x + 2 y, 0, 2 x^2 - x y + y^2 - 2 x + 1, -2 x y
+ 2 y^2 - y + 1, 0],
[x^2 - x y - y^2 + x - 2 y + 4, 2 x y + 2 y^2 + 2 x + y, -2 x^2 - 2 y^2 - 2 x
+ 3, 0, -2 x^2 + 2 x y + y^2 - x - 2 y + 2, x^2 - 2 y^2 + x, -2 x^2 + 2 x y
+ 2 y^2 - x + 2 y - 3, 2 y^2 - x + y - 1, x^2 + x y - y^2 - 2 x + 2 y + 3, 0],
[0, x^2 + 2 y^2 + x + y, -x^2 - x y - x + y + 3, -x^2 - 2 y^2 - 2 x + 2 y + 1,
-2 x y - y^2 - 2 x - 2 y + 1, -2 x^2 + 2 x y + 2 y^2 - 2 y, -x^2 + 2 x y - y^2
+ y, -x^2 + x y + 2 y^2 - 2 y - 2, -2 x y - y^2 - x - y + 2, 0]]

```

```

> IsZLP(M, var);
true (7.2)

```

```

> S := ZLP_Syzygy(M) :
"Computing Minor using:", 0.643, "s"
"Other using:", 0.065, "s" (7.3)

```

```

> st := time[real]( ) :
SyzygyModule(M%T, var) :
print("Computing Syzygy using:", time[real]( ) - st, "s");
"Computing Syzygy using:", 14.200, "s" (7.4)

```

## Example 8

```

> var := [x, y];
m := 10;
n := 6;
d := 3;
c := 2;
M := [[ -2 x y^2 + x y - y^2 - 3, x^2 y + 2 x^2 + x y + 2 y^2 - y - 1, x^3 + 2 x^2 y - 2 x
- y + 4, -2 x^3 + 2 x^2 y + 2 y^3 - x y + y^2 + 2, x^3 - 2 x^2 y - 2 y^2 - x + 2 y
- 2, -x^3 + x^2 y + x y^2 - 2 y^3 - 2 x^2 - x y, -x^3 + 2 x y^2 - x - 2 y - 1, 0, x y
+ y^2 - 2 y - 1, 0],
[x^3 - x^2 y + x^2 + x y + x + 1, x y^2 + x^2 + x y + 2 x, x y^2 + 2 x^2 - 2 x y - 2,
2 y^3 - 2, 0, -2 y^3 + 2 x^2 - 2 y^2 + 2 y + 3, 0, x^3 + 2 x^2 y - 2 x y^2 - y^3
+ 2 x y + 3, 0, 0],
[x^3 - y^3 + x^2 + 2 x y - 2 y^2 - 2 y - 2, 2 x^3 + x y^2 + 2 x y - x + 1, 2 x^2 y
+ 2 y^2 + 2 x - y + 3, 2 x^3 - 2 y^3 + x y + 2, -x^2 y - 2 x y^2 + 2 x y - 2 x - y
+ 1, 0, -x^3 + 2 x y^2 - 2 x^2 + x - 2 y - 2, 2 x^3 + 2 y^3 + x y - 2 y - 1, 0, 0
],
[x^3 - x y^2 + x^2 - 2 x y - y^2 + 2, 0, 0, 2 y^3 - 2 x^2 + 2 y + 2, -x^2 y + x^2 + x y
- 2 y^2 + x - 1, 2 x y^2 - 2 x - 4, 0, x^2 y - y^3 + 2 x^2 + 2 x + 1, -x^3 + 2 x y^2
- x^2 - 2 x - 2, -x^3 - x^2 y - 2 x y - 2 y - 1],
[0, 0, 2 x^3 + x^2 y - x^2 - 2 x - 1, 2 x^2 y + x y^2 + x y + y^2 - 3, -x^2 - 2 x y

```

```

+ 2 y^2 - 2 y - 1, 2 x^3 + 2 x y - y^2 - 2 x - 1, x y^2 + x^2 - 2 y^2 + 2 x - 3, 0,
- 2 x^3 + 2 x y^2 - 2 y^3 + 2 x y + 2 y + 1, - 2 x y^2 - 2 x^2 + 2 y^2 + x - 3],
[0, -x^3 - x^2 y + y^3 - 2 x - 2 y + 1, x^3 + x^2 y + 2 y^3 - 2 x + y - 2, 2 x^2 y
+ 2 x y^2 + 2 x^2 + 2 x - 2 y + 2, - 2 x y^2 - 2 y^3 + 2 x^2 + x y, -x^3 - x y^2 - y^3
- x y + 2 x + y, 2 x y^2 + 2 x y - 2 y^2 + 3, x^3 + 2 y^3 + x^2 - y^2 - 2 y + 1, 2 x^3
+ x^2 - 2 x y + x - 1, - 2 x^3 + x^2 y + 2 x y^2 - 2 y^2 - y]]];

```

```
var := [x, y]
```

```
m := 10
```

```
n := 6
```

```
d := 3
```

```
c := 2
```

```

M := [[ -2 x y^2 + x y - y^2 - 3, x^2 y + 2 x^2 + x y + 2 y^2 - y - 1, x^3 + 2 x^2 y
- 2 x - y + 4, - 2 x^3 + 2 x^2 y + 2 y^3 - x y + y^2 + 2, x^3 - 2 x^2 y - 2 y^2 - x
+ 2 y - 2, -x^3 + x^2 y + x y^2 - 2 y^3 - 2 x^2 - x y, -x^3 + 2 x y^2 - x - 2 y
- 1, 0, x y + y^2 - 2 y - 1, 0],
[x^3 - x^2 y + x^2 + x y + x + 1, x y^2 + x^2 + x y + 2 x, x y^2 + 2 x^2 - 2 x y
- 2, 2 y^3 - 2, 0, - 2 y^3 + 2 x^2 - 2 y^2 + 2 y + 3, 0, x^3 + 2 x^2 y - 2 x y^2
- y^3 + 2 x y + 3, 0, 0],
[x^3 - y^3 + x^2 + 2 x y - 2 y^2 - 2 y - 2, 2 x^3 + x y^2 + 2 x y - x + 1, 2 x^2 y
+ 2 y^2 + 2 x - y + 3, 2 x^3 - 2 y^3 + x y + 2, -x^2 y - 2 x y^2 + 2 x y - 2 x
- y + 1, 0, -x^3 + 2 x y^2 - 2 x^2 + x - 2 y - 2, 2 x^3 + 2 y^3 + x y - 2 y - 1,
0, 0],
[x^3 - x y^2 + x^2 - 2 x y - y^2 + 2, 0, 0, 2 y^3 - 2 x^2 + 2 y + 2, -x^2 y + x^2
+ x y - 2 y^2 + x - 1, 2 x y^2 - 2 x - 4, 0, x^2 y - y^3 + 2 x^2 + 2 x + 1, -x^3
+ 2 x y^2 - x^2 - 2 x - 2, -x^3 - x^2 y - 2 x y - 2 y - 1],
[0, 0, 2 x^3 + x^2 y - x^2 - 2 x - 1, 2 x^2 y + x y^2 + x y + y^2 - 3, -x^2 - 2 x y
+ 2 y^2 - 2 y - 1, 2 x^3 + 2 x y - y^2 - 2 x - 1, x y^2 + x^2 - 2 y^2 + 2 x - 3,
0, - 2 x^3 + 2 x y^2 - 2 y^3 + 2 x y + 2 y + 1, - 2 x y^2 - 2 x^2 + 2 y^2 + x - 3],
[0, -x^3 - x^2 y + y^3 - 2 x - 2 y + 1, x^3 + x^2 y + 2 y^3 - 2 x + y - 2, 2 x^2 y
+ 2 x y^2 + 2 x^2 + 2 x - 2 y + 2, - 2 x y^2 - 2 y^3 + 2 x^2 + x y, -x^3 - x y^2 - y^3
- x y + 2 x + y, 2 x y^2 + 2 x y - 2 y^2 + 3, x^3 + 2 y^3 + x^2 - y^2 - 2 y + 1,
2 x^3 + x^2 - 2 x y + x - 1, - 2 x^3 + x^2 y + 2 x y^2 - 2 y^2 - y]]]

```

(8.1)

```
> IsZLP(M, var);
```

```
true
```

(8.2)

```
> S := ZLP_Syzygy(M) :
```

```
"Computing Minor using:", 0.667, "s"
```

(8.3)

```
| "Other using:", 0.013, "s" (8.3)
```

```
[> st := time[real]( ) :  
  SyzygyModule( $M^{\%T}$ , var) :  
  print("Computing Syzygy using:", time[real]( ) - st, "s");  
  "Computing Syzygy using:", 118.286, "s" (8.4)
```

```
[>
```