Demo Maple Worksheet for ResChain Package

Installation

Just add the path where you have this mla file on your computer to the libname variable.
libname ,= "path"
Remember to use \\ instead of \ on windows and that the path should be stored as a string (in quotes).

Alternatively: Add this line to your maple.ini file.

```
> # libname,="C:\\Home\\Packages\\ResChain":
> with(ResChain);
```

[*ResChainBranching*, *ResChainSimple*] **(1)**

```
> Describe(ResChain);

# The purpose of this package is to use a chain of resultants to
eliminate n
# variables from m polynomials. In another word, consider a
system of m
# polynomial equations in n variables and some parameters. The
goal is to
# project the solution set of this system from space of both
variables and
# parameters into the space of only parameters. The result is a
set of
# conditions involving only parameters for the system to have a
solution. This
# is somehow similar to elimination theory via Grobner bases, or
when m = n +
# 1, similar to the Dixon resultant. However, the result with a
chain of
# resultants gives a superset of the Zariski closure of the
projection. The
# other two methods (Grobner basis and the Dixon resultant) give
the Zariski
# closure of the projection which is a better set, but they may
be slower or
# more expensive in some cases. Therefore, there is a trade off
between the
# different approaches.
module ResChain:

    # It eliminates the variables given in the second argument
via a chain of
    # resultants. It returns a list of polynomials after
finishing the
    # elimination.
    ResChainSimple( F::list(polynom), vars::list(symbol) ) ::
list
```

```
    # It eliminates the variables given in the second argument
via a chain of
    # resultants. It returns a list of polynomials after
finishing the
    # elimination.
    ResChainBranching( F::list(polynom), vars::list(symbol) ) ::
list
```

Consider the system of parametric polynomial equations with the polynomials in 'F' and variables in 'vars' and parameters being 'a' and 'b'.

```
> F := [(x^2+y^2-a)*(y-x^2),(x+y)*(x*y-b), (x^3+a*y^2+b)]:
  vars := [y, x]:
```

One can use elimination theory via Grobner basis computation to eliminate variables and find conditions on the parameters for the system to have a solution. Or in another word, projection of the solution set of the system from R^4 with (x, y, a, b) coordinates to R^2 with (a, b) coordinates.

```
> ideal_1 := PolynomialIdeals:-PolynomialIdeal(F):
  ideal_2 := PolynomialIdeals:-EliminationIdeal(ideal_1, {a,b}):
  set_1 := PolynomialIdeals:-Generators(ideal_2);
```

$$\begin{aligned} set\_1 := \{ & 2a^{12}b^7 + 2a^{11}b^8 - 2a^{15}b^3 - 2a^{14}b^4 + 10a^{13}b^5 + 14a^{12}b^6 - 9a^{11}b^7 \\ & - 3a^{10}b^8 + 8a^9b^9 + 3a^{14}b^3 - 11a^{13}b^4 - 25a^{12}b^5 + 47a^{11}b^6 + 90a^{10}b^7 - 31a^9b^8 \\ & - 38a^8b^9 + 10a^7b^{10} - a^{13}b^3 + 14a^{12}b^4 - 22a^{11}b^5 - 77a^{10}b^6 + 87a^9b^7 \\ & + 192a^8b^8 - 47a^7b^9 - 65a^6b^{10} + 8a^5b^{11} - 2a^{11}b^4 + 28a^{10}b^5 - 13a^9b^6 - 95a^8b^7 \\ & + 75a^7b^8 + 177a^6b^9 - 24a^5b^{10} - 24a^4b^{11} + 8a^3b^{12} - 16a^{12}b^2 - 16a^{11}b^3 \\ & + 80a^{10}b^4 + 111a^9b^5 - 46a^8b^6 - 56a^7b^7 - 9a^6b^8 + 32a^5b^9 + 56a^4b^{10} - 8a^3b^{11} \\ & + 24a^{11}b^2 - 88a^{10}b^3 - 200a^9b^4 + 376a^8b^5 + 720a^7b^6 - 239a^6b^7 - 328a^5b^8 \\ & + 64a^4b^9 + 8a^3b^{10} - 8a^{10}b^2 + 112a^9b^3 - 176a^8b^4 - 616a^7b^5 + 696a^6b^6 \\ & + 1536a^5b^7 - 376a^4b^8 - 528a^3b^9 + 64a^2b^{10} - 16a^8b^3 + 224a^7b^4 - 232a^6b^5 \\ & - 888a^5b^6 + 600a^4b^7 + 1416a^3b^8 - 192a^2b^9 - 192ab^{10} + 64b^{11} - 8a^6b^4 \\ & + 208a^5b^5 - 256a^4b^6 - 584a^3b^7 + 256a^2b^8 + 448ab^9 - 64b^{10} + 72a^3b^6 \\ & - 192a^2b^7 - 128ab^8 + 64b^9 - 64b^8 \} \end{aligned} \quad \textbf{(2)}$$

A different approach is to use a chain of resultants as explained in the paper "Resultant Tools for Parametric Polynomial Systems with Application to Population Models, AmirHosein Sadeghimanesh, Matthew England, 2022".

```
> set_2 := ResChainSimple(F, vars);
```

$$set\_2 := \left[a + b - 1, a + b + 1, a^3 - 4a^2 - 2ab - b^2 + 3a - 1, b, a^3 - b^2, a^3 b + 8, a, \right. \quad \textbf{(3)}$$

$$-a^6 b^8 + 9 a^5 b^8 - 18 a^4 b^8 + 6 a^3 b^9 - b^{12} - 9 a^7 b^4 - 6 a^6 b^5 + 3 a^3 b^8 - 18 a^2 b^9$$

$$- 3 a^6 b^4 - 36 a^5 b^5 - 27 a^4 b^6 - 2 a^3 b^7 - 3 b^{10} + a^9 - 21 a^3 b^6 - 18 a^2 b^7 - 3 a^6 b^2$$

$$- 3 b^8 + 3 a^3 b^4 - b^6, a^8 + 8 a^6 b + 24 a^4 b^2 + 32 a^2 b^3 + 16 b^4 - 2 a^3, a^4 - \frac{1}{2} a^3$$

$$+ 4 a^2 b + 4 b^2, a^8 b^4 + 4 a^8 b^3 + 6 a^8 b^2 + 4 a^6 b^4 + 4 a^8 b + 12 a^6 b^3 + a^8 + 12 a^6 b^2$$

$$+ 6 a^4 b^4 + 4 a^6 b - a^5 b^2 + 12 a^4 b^3 + 6 a^4 b^2 - 8 a^3 b^3 + 4 b^4 a^2 - 6 a^3 b^2 + 4 a^2 b^3$$

$$- 4 a^3 b + 8 b^3 a + b^4 - a^3 + 3 a b^2 + b^2, -a^4 b^4 + a^7 - 5 a^5 b^2 - 2 a^4 b^3 + 5 a^3 b^4$$

$$+ 2 a^5 b - a^4 b^2 - 8 a^3 b^3 - 2 b^4 a^2 + 4 a b^5 - b^6 + a^3 b^2 - 2 a^2 b^3 - 3 b^4 a - b^4]$$

Note that set_1 has one polynomial, but that polynomial is reducible.

```
> factor(set_1[1]);
```

$$-b^2 (a + b - 1) (2 a^4 - a^3 + 8 a^2 b + 8 b^2) (-a^4 b^4 + a^7 - 5 a^5 b^2 - 2 a^4 b^3 + 5 a^3 b^4 + 2 a^5 b - a^4 b^2 - 8 a^3 b^3 - 2 b^4 a^2 + 4 a b^5 - b^6 + a^3 b^2 - 2 a^2 b^3 - 3 b^4 a - b^4) (a^3 b + 8) \quad \textbf{(4)}$$

```
> set_3 := [seq(factors(set_1[1])[2][i][1], i=1..numelems(factors
  (set_1[1])[2]))];
```

$$set\_3 := \left[a^3 b + 8, b, a + b - 1, -a^4 b^4 + a^7 - 5 a^5 b^2 - 2 a^4 b^3 + 5 a^3 b^4 + 2 a^5 b - a^4 b^2 - 8 a^3 b^3 - 2 b^4 a^2 + 4 a b^5 - b^6 + a^3 b^2 - 2 a^2 b^3 - 3 b^4 a - b^4, a^4 - \frac{1}{2} a^3 + 4 a^2 b + 4 b^2\right] \quad \textbf{(5)}$$

```
> evalb(`and`(seq(polynomial in set_2, polynomial in set_3)));
```

$$true \quad \textbf{(6)}$$

This shows that all factors of the generator of the elimination ideal are listed among the polynomials in the output of ResChainSimple.

```
> numelems(set_2);
  numelems(set_3);
```

$$12$$
$$5 \quad \textbf{(7)}$$

But ResChainSimple can have extra components. Therefore if Grobner basis computation is feasible, it is better to use elimination via Grobner basis. However, in some cases Grobner basis computation is infeasible on a computer or takes too long time, in that case ResChainSimple may be helpful.

```
> set_4 := ResChainBranching(F, vars);
```

$$set\_4 := \left[a^4 - \frac{1}{2}\,a^3 + 4\,a^2 b + 4\,b^2,\ -a^4 b^4 + a^7 - 5\,a^5 b^2 - 2\,a^4 b^3 + 5\,a^3 b^4 + 2\,a^5 b - a^4 b^2 - 8\,a^3 b^3 - 2\,b^4 a^2 + 4\,a\,b^5 - b^6 + a^3 b^2 - 2\,a^2 b^3 - 3\,b^4 a - b^4,\ a + b - 1,\ a + b + 1,\ b,\ a^3 b + 8,\ a\right] \quad \textbf{(8)}$$

```
> evalb(`and`(seq(polynomial in set_4, polynomial in set_3)));
  evalb(`and`(seq(polynomial in set_2, polynomial in set_4)));
```

$$true$$

$$true \quad \textbf{(9)}$$

This shows that ResChainBranching contains the factors in the result of the elimination with Grobner Basis, and is contained in the result of ResChainSimple.

```
> numelems(set_4);
```

$$7 \quad \textbf{(10)}$$

This is in the middle of the two and takes less time and memory than ResChainSimple.

End of the file.