TRIGSIMP
A REDUCE Package for the Simplification and Factorization of Trigonometric and Hyperbolic Functions

Wolfram Koepf
Andreas Bernig
Herbert Melenk
ZIB Berlin
email: Koepf@ZIB-Berlin.de

1 Introduction

The REDUCE package TRIGSIMP is a useful tool for all kinds of trigonometric and hyperbolic simplification and factorization. There are three procedures included in TRIGSIMP: trigsimp, trigfactorize and triggcd. The first is for finding simplifications of trigonometric or hyperbolic expressions with many options, the second for factorizing them and the third for finding the greatest common divisor of two trigonometric or hyperbolic polynomials.

To start the package it must be loaded by:

```
1: load trigsimp;
```

2 REDUCE operator `trigsimp`

As there is no normal form for trigonometric and hyperbolic functions, the same function can convert in many different directions, e.g. $\sin(2x) \leftrightarrow 2\sin(x)\cos(x)$. The user has the possibility to give several parameters to the procedure `trigsimp` in order to influence the direction of transformations. The decision whether a rational expression in trigonometric and hyperbolic functions vanishes or not is possible using the (default) `expand` direction.

To simplify a function `f`, one uses `trigsimp(f[,options])`. Example:

```
2: trigsimp(sin(x)^2+cos(x)^2);

1
```

Possible options are (* denotes the default):

1. `sin` (*) or `cos`
2. `sinh` (*) or `cosh`
3. `expand` (*) or `combine` or `compact`
4. `hyp` or `trig` or `expon`
5. `keepalltrig`

From each group one can use at most one option, otherwise an error message will occur. The first group fixes the preference used while transforming a trigonometric expression:

```
3: trigsimp(sin(x)^2);

      2
sin(x)

4: trigsimp(sin(x)^2,cos);

         2
 - cos(x)  + 1
```

The second group is the equivalent for the hyperbolic functions. The third group determines the type of transformations. With the default `expand`, an expression is written in a form only using single arguments and no sums of arguments:

```
5: trigsimp(sin(2x+y));

                                   2
2*cos(x)*cos(y)*sin(x) - 2*sin(x) *sin(y) + sin(y)
```

With `combine`, products of trigonometric functions are transformed to trigonometric functions involving sums of arguments:

```
6: trigsimp(sin(x)*cos(y),combine);
```

```
 sin(x - y) + sin(x + y)
-------------------------
            2
```

With compact, the REDUCE operator compact [2] is applied to f. This leads often to a simple form, but in contrast to expand one doesn't get a normal form. Example for compact:

```
7: trigsimp((1-sin(x)^2)^20*(1-cos(x)^2)^20,compact);

      40       40
cos(x)  *sin(x)
```

With the fourth group each expression is transformed to a trigonometric, hyperbolic or exponential form:

```
8: trigsimp(sin(x),hyp);

 - sinh(i*x)*i

9: trigsimp(sinh(x),expon);

  2*x
 e    - 1
----------
      x
   2*e

10: trigsimp(e^x,trig);

     x          x
cos(---) + sin(---)*i
     i          i
```

Usually, tan, cot, sec, csc are expressed in terms of sin and cos. It can be sometimes useful to avoid this, which is handled by the option keepalltrig:

```
11: trigsimp(tan(x+y),keepalltrig);

  - (tan(x) + tan(y))
----------------------
  tan(x)*tan(y) - 1
```

It is possible to use the options of different groups simultaneously:

```
12: trigsimp(sin(x)^4,cos,combine);

 cos(4*x) - 4*cos(2*x) + 3
---------------------------
             8
```

Sometimes, it is necessary to handle an expression in different steps:

```
13: trigsimp((sinh(x)+cosh(x))^n+(cosh(x)-sinh(x))^n,expon);

  2*n*x
 e      + 1
------------
     n*x
    e

14: trigsimp(ws,hyp);

2*cosh(n*x)

15: trigsimp((cosh(a*n)*sinh(a)*sinh(p)+cosh(a)*sinh(a*n)*sinh(p)+
    sinh(a - p)*sinh(a*n))/sinh(a));

cosh(a*n)*sinh(p) + cosh(p)*sinh(a*n)

16: trigsimp(ws,combine);

sinh(a*n + p)
```

3 REDUCE operator `trigfactorize`

With `trigfactorize(p,x)` one can factorize the trigonometric or hyperbolic polynomial `p` with respect to the argument x. Example:

```
17: trigfactorize(sin(x),x/2);

        x        x
{2,cos(---),sin(---)}
        2        2
```

If the polynomial is not coordinated or balanced [1], the output will equal the input. In this case, changing the value of the second argument can help to find a factorization:

```
18: trigfactorize(1+cos(x),x);

{cos(x) + 1}

19: trigfactorize(1+cos(x),x/2);

        x        x
{2,cos(---),cos(---)}
        2        2
```

The polynomial can consist of both trigonometric and hyperbolic functions:

```
20: trigfactorize(sin(2x)*sinh(2x),x);

{4, cos(x), sin(x), cosh(x), sinh(x)}
```

4 REDUCE operator `triggcd`

The operator `triggcd` is an application of `trigfactorize`. With its help the user can find the greatest common divisor of two trigonometric or hyperbolic polynomials. It uses the method described in [1]. The syntax is: `triggcd(p,q,x)`, where p and q are the polynomials and x is the smallest unit to use. Example:

```
21: triggcd(sin(x),1+cos(x),x/2);

     x
cos(---)
     2

22: triggcd(sin(x),1+cos(x),x);

1
```

The polynomials p and q can consist of both trigonometric and hyperbolic functions:

```
23: triggcd(sin(2x)*sinh(2x),(1-cos(2x))*(1+cosh(2x)),x);

cosh(x)*sin(x)
```

5 Further Examples

With the help of the package the user can create identities:

```
24: trigsimp(tan(x)*tan(y));

 sin(x)*sin(y)
---------------
 cos(x)*cos(y)

25: trigsimp(ws,combine);

 cos(x - y) - cos(x + y)
-------------------------
 cos(x - y) + cos(x + y)

26: trigsimp((sin(x-a)+sin(x+a))/(cos(x-a)+cos(x+a)));

 sin(x)
--------
 cos(x)

27: trigsimp(cosh(n*acosh(x))-cos(n*acos(x)),trig);

0

28: trigsimp(sec(a-b),keepalltrig);

  csc(a)*csc(b)*sec(a)*sec(b)
-------------------------------
 csc(a)*csc(b) + sec(a)*sec(b)

29: trigsimp(tan(a+b),keepalltrig);

  - (tan(a) + tan(b))
----------------------
  tan(a)*tan(b) - 1

30: trigsimp(ws,keepalltrig,combine);

tan(a + b)
```

Some difficult expressions can be simplified:

```
31: df(sqrt(1+cos(x)),x,4);

                              4             2       2            2
(sqrt(cos(x) + 1)*( - 4*cos(x)  - 20*cos(x) *sin(x)  + 12*cos(x)

                      2                        4           2
     - 4*cos(x)*sin(x)  + 8*cos(x) - 15*sin(x)  + 16*sin(x) ))/(16

          4           3           2
   *(cos(x)  + 4*cos(x)  + 6*cos(x)  + 4*cos(x) + 1))

32: trigsimp(ws);

 sqrt(cos(x) + 1)
------------------
        16

33: load taylor;

34: taylor(sin(x+a)*cos(x+b),x,0,4);

cos(b)*sin(a) + (cos(a)*cos(b) - sin(a)*sin(b))*x

                                   2
 - (cos(a)*sin(b) + cos(b)*sin(a))*x

    2*( - cos(a)*cos(b) + sin(a)*sin(b))   3
 + --------------------------------------*x
                    3

    cos(a)*sin(b) + cos(b)*sin(a)   4      5
 + -------------------------------*x  + O(x )
                 3

35: trigsimp(ws,combine);

sin(a - b) + sin(a + b)                                2    2*cos(a + b)   3
------------------------- + cos(a + b)*x - sin(a + b)*x  - --------------*x
            2                                                     3

    sin(a + b)   4      5
 + ------------*x  + O(x )
        3
```

Certain integrals whose calculation was not possible in REDUCE (without preprocessing), are now computable:

```
36: int(trigsimp(sin(x+y)*cos(x-y)*tan(x)),x);

       2                                                             2
 cos(x) *x - cos(x)*sin(x) - 2*cos(y)*log(cos(x))*sin(y) + sin(x) *x
----------------------------------------------------------------------
                                   2

37: int(trigsimp(sin(x+y)*cos(x-y)/tan(x)),x);

                                    x  2
(cos(x)*sin(x) - 2*cos(y)*log(tan(---)  + 1)*sin(y)
                                    2

                       x
  + 2*cos(y)*log(tan(---))*sin(y) + x)/2
                       2
```

Without the package, the integration fails, in the second case one doesn't receive an answer for many hours.

```
38: trigfactorize(sin(2x)*cos(y)^2,y/2);

{2*cos(x)*sin(x),

      y          y
 cos(---) + sin(---),
      2          2

      y          y
 cos(---) + sin(---),
      2          2

      y          y
 cos(---) - sin(---),
      2          2

      y          y
 cos(---) - sin(---)}
      2          2

39: trigfactorize(sin(y)^4-x^2,y);
```

```
          2                2
{ - sin(y)  + x, - (sin(y)  + x)}

40: trigfactorize(sin(x)*sinh(x),x/2);

        x        x         x         x
{4,cos(---),sin(---),cosh(---),sinh(---)}
        2        2         2         2

41: triggcd(-5+cos(2x)-6sin(x),-7+cos(2x)-8sin(x),x/2);

       x        x
2*cos(---)*sin(---) + 1
       2        2

42: triggcd(1-2cosh(x)+cosh(2x),1+2cosh(x)+cosh(2x),x/2);

        x  2
2*sinh(---)  + 1
        2
```

References

[1] Roach, Kelly: Difficulties with Trigonometrics. Notes of a talk.

[2] Hearn, A.C.: COMPACT User Manual.