

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

Wolfram Koepf
Andreas Bernig
Herbert Melenk
ZIB Berlin

Revised by Francis Wright
QMW London
E-mail: F.J.Wright@Maths.QMW.ac.uk

17 January 1999

1 Introduction

The REDUCE package TRIGSIMP is a useful tool for all kinds of problems related to trigonometric and hyperbolic simplification and factorization. There are three operators included in TRIGSIMP: `trigsimp`, `trigfactorize` and `triggcd`. The first is for simplifying trigonometric or hyperbolic expressions and has many options, the second is for factorizing them and the third is for finding the greatest common divisor of two trigonometric or hyperbolic polynomials. This package is automatically loaded when one of these operators is used.

2 Simplifying trigonometric expressions

As there is no normal form for trigonometric and hyperbolic expressions, 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 operator `trigsimp` in order to influence the transformations. It is possible to decide whether or not a rational expression involving trigonometric and hyperbolic functions vanishes.

To simplify an expression `f`, one uses `trigsimp(f[,options])`. For example:

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

1

The possible options (where * denotes the default) are:

1. `sin*` or `cos`;
2. `sinh*` or `cosh`;
3. `expand*`, `combine` or `compact`;
4. `hyp`, `trig` or `expon`;
5. `keepalltrig`;
6. `tan` and/or `tanh`;
7. target arguments of the form *variable / positive integer*.

From each of the first four groups one can use at most one option, otherwise an error message will occur. Options can be given in any order.

The first group fixes the preference used while transforming a trigonometric expression:

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

2
sin(x)

```
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 transformation. With the default, `expand`, an expression is transformed to use only simple variables as arguments:

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

$$2*\cos(x)*\cos(y)*\sin(x) - 2*\sin(x) *\sin(y) + \sin(y)$$

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

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

$$\frac{\sin(x - y) + \sin(x + y)}{2}$$

With `compact`, the REDUCE operator `compact` [2] is applied to `f`. This often leads to a simple form, but in contrast to `expand` one does not get a normal form. For example:

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

$$\cos(x)^{40} *\sin(x)^{40}$$

With an option from the fourth group, the input expression is transformed to trigonometric, hyperbolic or exponential form respectively:

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

$$- \sinh(i*x)*i$$

```
trigsimp(sinh(x), expon);
```

$$\frac{e^{2*x} - 1}{2*e^x}$$

```
trigsimp(e^x, trig);
```

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

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

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

$$\frac{- (\tan(x) + \tan(y))}{\tan(x)*\tan(y) - 1}$$

Alternatively, the options `tan` and/or `tanh` can be given to convert the output to the specified form as far as possible:

```
trigsimp(tan(x+y), tan);
```

$$\frac{- (\tan(x) + \tan(y))}{\tan(x)*\tan(y) - 1}$$

By default, the other functions used will be `cos` and/or `cosh`, unless the other desired functions are also specified in which case this choice will be respected.

The final possibility is to specify additional target arguments for the trigonometric or hyperbolic functions, each of which should have the form of a variable divided by a positive integer. These additional arguments are treated as if they had occurred within the expression to be simplified, and their denominators are used in determining the overall denominator to use for each variable in the simplified form:

```
trigsimp(csc x - cot x + csc y - cot y, x/2, y/2, tan);
```

$$\tan\left(\frac{x}{2}\right) + \tan\left(\frac{y}{2}\right)$$

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

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

$$\frac{\cos(4*x) - 4*\cos(2*x) + 3}{8}$$

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

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

$$\left(\frac{1}{x}\right)^n + e^{n*x}$$

```
trigsimp(ws, hyp);
```

$$2*\cosh(n*x)$$

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

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

```
trigsimp(ws, combine);
```

$$\sinh(a*n + p)$$

The `trigsimp` operator can be applied to equations, lists and matrices (and compositions thereof) as well as scalar expressions, and automatically maps itself recursively over such non-scalar data structures:

```
trigsimp( { sin(2x) = cos(2x) } );
```

$$\{2*\cos(x)*\sin(x) = -2*\sin(x)^2 + 1\}$$

3 Factorizing trigonometric expressions

With `trigfactorize(p,x)` one can factorize the trigonometric or hyperbolic polynomial p in terms of trigonometric functions of the argument x .

The output has the same format as that from the standard REDUCE operator `factorize`. For example:

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

$$\{\{2,1\}, \{\sin(\frac{x}{2}), 1\}, \{\cos(\frac{x}{2}), 1\}\}$$

If the polynomial is not coordinated or balanced [1], the output will equal the input. In this case, changing the value for `x` can help to find a factorization, e.g.

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

$$\{\{\cos(x) + 1, 1\}\}$$

```
trigfactorize(1+cos(x), x/2);
```

$$\{\{2,1\}, \{\cos(\frac{x}{2}), 2\}\}$$

The polynomial can consist of both trigonometric and hyperbolic functions:

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

$$\{\{4,1\}, \{\sinh(x), 1\}, \{\cosh(x), 1\}, \{\sin(x), 1\}, \{\cos(x), 1\}\}$$

The `trigfactorize` operator respects the standard REDUCE `factorize` switch `nopowers` – see the REDUCE manual for details. Turning it on gives the behaviour that was standard before REDUCE 3.7:

```
on nopowers;
```

```
trigfactorize(1+cos(x), x/2);
```

$$\{2, \cos(\frac{x}{2}), \cos(\frac{x}{2})\}$$

4 GCDs of trigonometric expressions

The operator `triggcd` is essentially an application of the algorithm behind `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 trigonometric polynomials and `x` is the argument to use. For example:

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

$$\cos\left(\frac{x}{2}\right)$$

```
triggcd(sin(x), 1+cos(x), x);
```

1

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

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

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

5 Further Examples

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

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

$$\frac{\sin(x)*\sin(y)}{\cos(x)*\cos(y)}$$

```
trigsimp(ws, combine);
```

$$\frac{\cos(x - y) - \cos(x + y)}{\cos(x - y) + \cos(x + y)}$$

```
trigsimp((sin(x-a)+sin(x+a))/(cos(x-a)+cos(x+a)));
```

$$\frac{\sin(x)}{\cos(x)}$$

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

0

```
trigsimp(sec(a-b), keepalltrig);
```

$$\frac{\csc(a)*\csc(b)*\sec(a)*\sec(b)}{\csc(a)*\csc(b) + \sec(a)*\sec(b)}$$

```
trigsimp(tan(a+b), keepalltrig);
```

$$\frac{-(\tan(a) + \tan(b))}{\tan(a)*\tan(b) - 1}$$

```
trigsimp(ws, keepalltrig, combine);
```

$\tan(a + b)$

Some difficult expressions can be simplified:

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

$$\begin{aligned} & (-4*\cos(x)^5 - 4*\cos(x)^4 - 20*\cos(x)^3*\sin(x)^2 + 12*\cos(x)^3 \\ & - 24*\cos(x)^2*\sin(x)^2 + 20*\cos(x)^2 - 15*\cos(x)*\sin(x)^4 \\ & + 12*\cos(x)*\sin(x)^2 + 8*\cos(x)^4 - 15*\sin(x)^4 + 16*\sin(x)^2) / \\ & (16*\sqrt{\cos(x) + 1}) \end{aligned}$$


```

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

on rationalize;
trigsimp(ws);

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

off rationalize;
load_package taylor;

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

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

- (cos(a)*sin(b) + cos(b)*sin(a))*x2

+ -----*x3
  2*(-cos(a)*cos(b) + sin(a)*sin(b))
  3

+ -----*x4 + 0(x5)
  cos(a)*sin(b) + cos(b)*sin(a)
  3

trigsimp(ws, combine);

  sin(a - b) + sin(a + b)
  ----- + cos(a + b)*x2 - sin(a + b)*x2
          2

- -----*x3 + -----*x4 + 0(x5)
  2*cos(a + b)      3      sin(a + b)      4      5
  3                  3

```

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

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

$$\begin{aligned} & \frac{2}{(\cos(x) * x - \cos(x) * \sin(x) - 2 * \cos(y) * \log(\cos(x)) * \sin(y) \\ & + \sin(x) * x) / 2} \end{aligned}$$

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

$$\begin{aligned} & \frac{(\cos(x) * \sin(x) - 2 * \cos(y) * \log(\tan(\frac{x}{2}) + 1) * \sin(y) \\ & + 2 * \cos(y) * \log(\tan(\frac{x}{2})) * \sin(y) + x) / 2}{2} \end{aligned}$$

Without the package, the integration fails, and in the second case one does not receive an answer for many hours.

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

```
{{2*cos(x)*sin(x),1},
```

$$\left\{ \cos\left(\frac{y}{2}\right) - \sin\left(\frac{y}{2}\right), 2 \right\},$$

$$\left\{ \cos\left(\frac{y}{2}\right) + \sin\left(\frac{y}{2}\right), 2 \right\}$$

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

$$\left\{ \sin(y)^2 + x, 1 \right\}, \left\{ \sin(y)^2 - x, 1 \right\}$$

```
trigfactorize(sin(x)*sinh(x), x/2);
```

```
{{4,1},
```

```

      x
{sinh(---),1},
      2

```

```

      x
{cosh(---),1},
      2

```

```

      x
{sin(---),1},
      2

```

```

      x
{cos(---),1}}
      2

```

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

```

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

```

```
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.