Maple Procedures for Simplifying Sine-Cosine Equations

JAIME GUTIERREZ
Departamento de Matemáticas, Estadística y Computación
Facultad de Ciencias, Universidad de Cantabria
Avda, Los Castros S/N, Santander 39071
SPAIN

Abstract:– In this paper we present two MAPLE programs which are designed for performing computations of sine-cosine equations. The two main codes are factoring and decomposing sine-cosine equations modulo the unit circle. We also motivate the interest to work with sine-cosine equations by presenting applications to robot kinematics: solving or simplifying the kinematic polynomial system equations. We include the synopsis of the two main Maple procedures and we also present some simple examples in order to illustrate the performance of our implementation, including the computation time.


1 Introduction

By a sine-cosine equation we will understand a polynomial equality \( f(s, c) = 0 \), with \( f \) in the quotient ring \( K[s, c]/(s^2 + c^2 - 1) \), and where \( K \) is a field of characteristic zero (typically, a numerical field such as the rational numbers field \( \mathbb{Q} \) or the real numbers field \( \mathbb{R} \), or a field of parameters \( \mathbb{Q}(d_1, \ldots, d_m) \)).

Therefore, when we write \( f(s, c) \) we consider, throughout this paper, that this \( s - c \) polynomial expression \( f(s, c) = 0 \), is implicitly univariate in some unknown angle \( \theta \) such that:

\[
s = \sin(\theta), \quad c = \cos(\theta).
\]

We will choose to write \( f(s, c) \) in normal or canonical form: i.e. replacing \( s^2 \) by \( (1 - c^2) \) as much as possible. The result is, then, a polynomial of the form \( A + Bs \), where \( A \) and \( B \) are polynomials in \( c \) only. We remark that if the total degree (as a two-variable polynomial) of \( f(s, c) \) is \( n \), then there are up to \( 2n \) values of the angle \( \theta \) (when properly counted) satisfying the equation.

So, we are interested on simplifying or solving equations of the sort \( f(s, c) = 0 \); and thus, equivalently, for solving or simplifying systems

\[
\begin{align*}
\quad & f(s, c) = 0 \\
\Rightarrow & s^2 + c^2 - 1 = 0.
\end{align*}
\]

2 Motivations

Polynomial systems, where the variables are interpreted as trigonometric functions of unknown angles, are quite ubiquitous, arising, for instance, in electrical networking, in molecular kinematics and in concrete situations, like in tilting effects on a double pendulum. Here, our applications will be taken from the field of robot kinematics. Besides referring to the many situations described in the recent book of [5], we will sketch, for the sake of being self-contained, an example of the role of sine-cosine systems in robotics:

Example

Given a robot arm with six revolute joints, i.e. a 6R robot (see Figure), the inverse kinematics problem is to find the values of the differ-
ent joint angles (with respect to some standard way of measuring them) that place the tip (or hand) of the robot at some desired position and orientation.

So, the inverse kinematics problem amounts to solving a non-linear polynomial system where the unknowns are the sines and cosines:

\[ \{ s_i = \sin(\theta_i), c_i = \cos(\theta_i), i = 1, \ldots, 6 \} \]

of the six joint angles:

\[ \{ \theta_i, i = 1, \ldots, 6 \}. \]

The solution of such systems, in general, is quite involved, and depends on the particular geometry of the robot. After decades of research, a symbolic solution (though not in closed form) for the general 6R manipulator inverse kinematics system has been found (see [6],[7] and [8]). By a clever elimination method it turns out that in this system \( \theta_3 \) can be determined as the solution of a degree-16 polynomial in the tangent of \( \theta_3/2 \); then \( \theta_1 \) and \( \theta_2 \) are found by solving a system of sine-cosine polynomials, linear in these trigonometric functions, with coefficients in \( \theta_3 \). Of course, the determining degree-16 polynomial can be also expressed as a degree 8 polynomial in the sine and cosine of \( \theta_3 \). However, the degree of this solution, together with the complexity of its coefficients, which may contain thousands of terms (see [9]), limits the practical use of this approach.

In practice, the control of a robot requires the solution of the kinematic problem to be of low degree, so that the joint angles can be quickly found. Thus, it is of primordial interest to simplify, when possible, such a univariate sine-cosine equation.

The recent paper by Gutierrez and Recio (1998) contains several methods for solving or simplifying sine-cosine equations. The goal of this paper is the implementation on MAPLE of the algorithms described in [4] for factoring and decomposing sine-cosine equations modulo the unit circle.

3 Factoring and Decomposing Sine-Cosine Equations

3.1 Factoring

Probably the more natural approach to simplification is that of factoring a given s-c equation \( f \) modulo \( s^2 + c^2 - 1 \). Algorithms for factoring bivariate polynomials cannot be directly applied to factor sine-cosine equations, as shown in the next example.

**Example**

The following polynomial:

\[ f(s, c) = 6c^5s - 4c^2 - 2c^4s^2 + cs + 2c^4s - 2c + s^3 + s^2c - s^2 + c^2s + c^3 - s + 1, \]

is irreducible as ordinary bivariate polynomial over the rational number field, but

\[ f(s, c) = g(s, c) h(s, c) \text{ modulo } s^2 + c^2 - 1, \]

where

\[ h(s, c) = -2c^3s + 1, \]

\[ g(s, c) = (-3c - 1 + s)c. \]

Although \( K[s, c]/(s^2 + c^2 - 1) \) is not a unique factorization domain, we can still look for lower degree factors of \( f \). More precisely, factoring \( f \) over an orderable field essentially means: finding sine-cosine polynomials \( g(s, c), h(s, c) \), verifying \( f = gh \) modulo \( s^2 + c^2 - 1 \), plus the conditions: \( \text{deg}(f) > \text{deg}(g) \) and \( \text{deg}(f) > \text{deg}(h) \), in order to avoid trivial factorizations.

The work [4] contains a complete factorization algorithm over fields that do not contain the square root of \(-1\). Basically, the method is based on the half-angle tangent substitution and then factoring univariate polynomial.

3.2 Decomposing

Roughly speaking, a function \( f(x, y) \) can be called decomposable if there is some polynomial \( g(z) \) in a new variable \( z \) and some other function \( h(x, y) \), such that \( f(x, y) = g(h(x, y)) \). Advanced methods for the decomposition of ordinary multivariate polynomials (see [2], [3]) can-
not be directly applied to sine-cosine equations, as shown in the next example.

**Example**
The following polynomial:
\[ f(x, y) = -63y^2 + 60xy - 8y - 20x + 78, \]
can not be written as the composition of two polynomials \( g(x) \) and \( h(x, y) \) such that:
\[ f(x, y) = g(h(x, y)), \]
but
\[ f(s, c) = g(h(s, c)) \text{ modulo } s^2 + c^2 - 1, \]
where \( g(x) = 3z^2 - 4z + 3 \) and \( h(s, c) = 2c + 5s \).

The natural notion of decomposability for \( s \)-\( c \) polynomials \( f(s, c) \) states, therefore, the existence of a standard polynomial \( g(x) \) and of another \( s \)-\( c \) polynomial \( h(s, c) \), such that
\[ f(s, c) = g(h(s, c)) \text{ modulo } s^2 + c^2 - 1. \]
Here, as in the case of factorization, we look for composition factors which are simpler than the given polynomial. The method in [4] for decomposing sine-cosine equations is based on the computation of \( t \)-th approximate root of \( f(s, c) \) and then just solving linear system equations.

**4 Description of Implemented Procedures**

In this section we succinctly describe the main implemented procedures, giving the arguments, calling sequence and a short comment about the codes. The implementation of all steps of the algorithms are straightforward. The programs are based on the algorithm in the paper [4]. The codes are contained in the MAPLE package FRAC (see [1]) which was implemented by Cesar Alonso and it designed for performing computations in the rational function field.

**Procedure : scfacpol**
The code **scfacpol** has as input a bivariate polynomial and three variables \( x, s, c \). The coefficients are multivariate rational functions over the rational number field \( Q(x_1, \ldots, x_n) \).

**ARGUMENTS:** A polynomial \( f \) and variables \( x, s, c \).

**CALLING SEQUENCE:** scfacpol(f,x,s,c).

**SYNOPSIS:** The procedure scfacpol computes a factorization of the normal form \( NF(f) \) of \( f \) modulo the unit circle \( s^2 + c^2 - 1 \), as:
\[ NF[NF(f) = (c + 1)^e NF(g) NF(h)], \]
where \( g \) and \( h \) are in \( Q(x_1, \ldots, x_n)[s, c] \) verifying the conditions \( deg(NF(f)) > deg(NF(g)) \), \( deg(NF(f)) > deg(NF(h)) \) and \( e \) an integer from 0 to the \( deg(NF(f)) \).

The code **scfacpol** has as output the list:
\[ [(c + 1)^e, NF(g), NF(h)] \]
if \( f \) has a such factorization otherwise, returns the empty list.

**Procedure : scdecpol**
The code **scdecpol** has as input a bivariate polynomial and three variables \( x, s, c \). The coefficients are multivariate rational functions over the rational number field \( Q(x_1, \ldots, x_n) \).

**ARGUMENTS:** A polynomial \( f \) and variables \( x, s, c \).

**CALLING SEQUENCE:** scdecpol(f,x,s,c).

**SYNOPSIS:** The procedure scdecpol computes all non-equivalent decomposition of the normal form \( NF(f) \) of \( f \) modulo the unit circle, as:
\[ NF[NF(f) = g(NF(h))], \]
where \( g(x) \) is in \( Q(x_1, \ldots, x_n)[x] \) and \( NF(h) \) is the normal form of \( h \in Q(x_1, \ldots, x_n)[s, c] \) such that \( deg(NF(f)) > deg(NF(h)) \).

The code **scdecpol** has as output a list of lists
\[ [..., [g, NF(h)], ...], \]
if \( f \) has a such decomposition. Otherwise, returns the empty list.

**Procedure : ascdecpol**
The code **ascdecpol** has as input a bivariate polynomial and variables \( x, s, c \). The coefficients are rational numbers.

**ARGUMENTS:** A polynomial \( f \) and variables \( x, s, c \).

**CALLING SEQUENCE:** ascdecpol(f,x,s,c).

**SYNOPSIS:** The procedure ascdecpol computes all non-equivalent decomposition of the normal form \( NF(f) \) of \( f \) modulo the unit circle, as...
where \( g(x) \) is an univariate polynomial in an algebraic extension \( Q(\alpha) \) of \( Q \) and \( NF(h) \) is the normal form of \( h \) with coefficients in \( Q(\alpha) \). with \( \deg(NF(f)) > \deg(NF(h)) \). This procedure uses the Maple command absolute factorization.

The code \texttt{ascdecpol} has as output a list of lists:
\[
[...[g, NF(h)], ...],
\]
if \( f \) has such a decomposition. Otherwise, returns the empty list.

5 Examples

Finally, in this section we show some examples of the performance of our algorithms, showing the time spent in seconds and the "words" in the computations. The examples have been tested in a Power Macintosh 900/300. In the following examples we only show the input sine-cosine polynomial, the list of the components and the computing time.

5.1 Example

The following example is an irreducible numerical sine-cosine polynomial:
\[
f = -2630241 c 5 s - 561087 c 2 + 1055 c 4 s 2 + 225 c s - 211 c 4 s - 280 c + 6330 c 5 s 2 + 1675 c 2 s + 134618 s 2 c 3 + 28385 s - 9495 c 8 s 2 + 325 s 3 + 235 s 2 c - 567 s 2 + 235 c 3 + 567.
\]

We are looking for a factorization of the sine-cosine polynomial \( f \) over the field \( Q(\alpha) \). Then we use the Maple procedure:

\texttt{scfacpol}(f, s, c):
\[
[1, -2025 c 5 s + 1350 c 2 s - 560520 c 2 + 225 c s - 45 c + 28710 s].
\]

Computing Time : 0.58 words 165562.

5.2 Example

The following example is an irreducible sine-cosine polynomial with coefficients over the rational function field \( Q(a,b) \).
\[
f = 70 + 21060 b - 2 b 7 c s + 2 a c 2 - 10530 b 2 s + 256 c 5 a 2 - 35 b s + 8960 a c 3 + 2947200 c 3 a 2 - 4912 a b + 2456 a b + 627536 c 3 a 2 s - 256 c 5 b a + 12636000 b a c - 256 b 7 c a - 1200 c 3 c 2 a - 3 b s - 2 c 2 b + 2695680 c 3 + 2 c 2 b 2 - 2 c 2 b + 42000 sac - 2947200 a c 2 + 1200 c 5 b a - 1200 b 7 c 2 a - 2456 c 2 ab.
\]

We are looking for a factorization of the sine-cosine polynomial \( f \) over the field \( Q(a,b) \). Then we use the Maple procedure:

\texttt{scfacpol}(f, s, c):
\[
[1, -2025 c 5 s + 1350 c 2 s - 560520 c 2 + 225 c s - 45 c + 28710 s].
\]

Computing Time : 0.88 words 139433.

5.3 Example

The following example is an indecomposable numerical sine-cosine polynomial
\[
f = -6582403087232 c 10 s - 278494451200 c s - 11139778048 c 2 s + 1803837606592 c 2 + 154667002112 c 3 + 8274144014624 c 5 s + 8390362928640 c 11 s - 2104862830694 c 9 s - 561173501400 c 7 s - 30715772434788 c 6 s + 173784320704 c 5 - 814657974816 c 4 s + 564955995424 c 3 s - 467765086208 c 5 s + 1011166716448 c 7 s - 368873757728 c 13 s + 2223533920464 c 12 s + 2848858096 c 14 s + 162584215472 c 13 s + 540078984348 c 4 - 11195511740 c 15 - 15034449216 c 14 - 4131288 c 16 s - 107837744 c 15 s - 34216 c 17 s - 3297588 c 17 s - 320833884 c 16 - 3465 c 18 s - 510664163640 c 19 s + 4170037792028 c 18 s - 565600729152 c 17 s + 407323838140 c 12 s - 954071415808 c 11 s - 17021713676.
\]

We want to know if \( f(s,c) \) has a decomposition over the rational field \( Q \).

\texttt{scdecpol}(f, s, c):
\[
[1, 17021713676 - 32447296 x + 15463 x 2, 1/47(47 c 5 - 52 c 8 s - 962844 c 2 - 92034 c 3 + 1991080 c 4 + 147080 c 5 - 979814 c 6 - 55116 c 7 + 870 c 8 + 403400 cs + 16136 c 2 s + 252000 c 3 s + 112080 c 4 s - 647520 c 5 s - 126464 c 6 s - 5316 c 7 s)].
\]

Computing Time : 10.63 words 1921379.
5.4 Example

The following example is an indecomposable sine-cosine polynomial with coefficients in the rational function field \(Q(a, b, m, n)\).

\[
f = -21662 \cdot c + 3938 \cdot es + 114874 \cdot c^2 s - 260864 \cdot c^2 - 121438 \cdot c^3 + 22022 \cdot c^3 b - 547515 \cdot c^5 s + 934 \cdot c^2 s b + 71415 \cdot c^6 b + 12420 \cdot c^4 s b - 6210 \cdot c^5 a^2 - 8078 \cdot c^5 b + 6210 \cdot c^5 b + 135 \cdot c^6 a^2 b^2 + 540 \cdot c^6 a^2 b + 556830 \cdot c^3 s - 12420 \cdot c^4 a^2 s + 142830 \cdot c^5 + 8078 \cdot c^3 a^2 - 647 \cdot c^5 b^2 - 934 \cdot c^2 a^2 s - 22022 \cdot c^2 a^2 s + 3105 \cdot c^5 a^4 s + 155 \cdot c^6 a^4 s + 5001 \cdot n^7 - 71415 \cdot c^4 s - 135 \cdot c^6 a^4 b - 71415 \cdot c^6 a^4 s - 270 \cdot c^4 a^2 s b + 135 \cdot c^5 b^2 s - 90 \cdot c^2 b - 467 \cdot c^4 a^2 b + 934 \cdot c^4 a^2 b + 3105 \cdot c^5 b^2 s + 90 \cdot c^5 a^5 b - 934 \cdot c^3 a^7 b - 934 \cdot c^3 a^2 m^3 + 934 \cdot c^3 b^2 a^5 + 934 \cdot c^5 b^2 a^3 - 22022 \cdot c^3 s a^5 b + 8078 \cdot c^2 s a^5 b + 467 \cdot c^3 s a^5 b + 71820 \cdot c^3 s m^3 - 270 \cdot c^3 s m^3 + 6210 \cdot c^5 a^2 s - 270 \cdot c^5 a^2 s m^3 + 270 \cdot c^3 a^2 m^3 + 270 \cdot c^3 a^2 s b + 6210 \cdot c^2 a^2 s b + 135 \cdot c^5 b^2 a^5 + 135 \cdot c^5 b^2 m^3 - 934 \cdot c^2 a^2 m^3 - 934 \cdot c^3 a^2 b s + 135 \cdot c^5 a^9 b + 135 \cdot c^5 a^4 m^3 - 270 \cdot c^5 a^7 b - 540 \cdot c^4 a^7 b + 135 \cdot c^4 a^{12} b^2 - 540 \cdot c^2 a^2 m^3 + 135 \cdot c^5 a^4 m^3 - 22022 \cdot c^3 s a^5 b + 8078 \cdot c^2 s a^5 b + 467 \cdot c^3 s a^5 b + 71820 \cdot c^3 s m^3 + 270 \cdot c^3 s m^3 + 645 \cdot c^3 m^3 + 54 \cdot c^3 m^3 - 71820 \cdot c^3 b - 270 \cdot c^3 b^2 - 45 \cdot c^5 b^2 - 270 \cdot c^3 a^{10} b^2 + 45 \cdot c^3 a^{15} b - 540 \cdot c^2 a^5 b m^3 - 12420 \cdot c^3 s a^5 b + 3105 \cdot c^3 s a^{10} b^2 + 6210 \cdot c^3 s a^5 b m^3 + 71820 \cdot c^3 a^5 b - 270 \cdot c^3 b m^3 - 270 \cdot c^3 b^2 m^3 - 270 \cdot c^3 b^2 a^{5} s - 6210 \cdot c^4 b^2 s a^5 - 12420 \cdot c^3 s m^3 + 3105 \cdot c^3 s m^3 - 6210 \cdot c^3 s m^3 + 6210 \cdot c^3 s a^5 b m^3 + 540 \cdot c^2 b^2 a^5 - 135 \cdot c^5 b^2 a^{10} + 540 \cdot c^4 b^4 m^3 - 135 \cdot c^5 b^4 m^3 + 135 \cdot c^2 a^5 b m^3 + 135 \cdot c^2 a^{10} b^2 s + 135 \cdot c^3 a^5 b m^3 + 90 \cdot c^3 m^3.
\]

We are interested in a decomposition of the form modulo the unit circle over the field \(Q(a, b, n, m)\).

\[
sedecpol \:
\begin{array}{c}
[b^2 + 5001 \cdot n^7 + (45 \cdot b - 45 \cdot a^2) \cdot x + (-467 \cdot b^2 + 934 \cdot a^2 b - 467 \cdot a^4) \cdot x^2 + (-45 \cdot b^3 + 135 \cdot a^2 b^2 - 135 \cdot a^2 b + 45 \cdot a^6) \cdot x^3; \\
1/(b - a^2)(c^2 b - c^2 a^2 - 23 \cdot cs - ca^5 b + 2 \cdot c - cm^3 - s)]
\end{array}
\]

Computing Time : 6.68 words 557710.

5.5 Example

Finally, this example illustrates an indecomposable sine-cosine numerical polynomial over the rational number field, but it has a decomposition over the algebraic field \(Q(\alpha)\), where \(\alpha\) is a root of the polynomial \(117 \cdot Z - 14 + 14 \cdot Z^2\): \(f = 234 \cdot c^2 + 56 \cdot cs - 190\).

Here, we are looking for a decomposition over an algebraic extension of \(Q\).

\[
\text{asespol} (f, s, c):
\begin{array}{c}
[28 \cdot \text{RootOf}(117 \cdot Z - 14 + 14 \cdot Z^2) + 44 - 28 \cdot \text{RootOf}(117 \cdot Z - 14 + 14 \cdot Z^2) \cdot x^2, \\
-28 \cdot \text{RootOf}(117 \cdot Z - 14 + 14 \cdot Z^2) - 190 + (28 \cdot \text{RootOf}(117 \cdot Z - 14 + 14 \cdot Z^2) + 234) \cdot x^2, \\
c + \text{RootOf}(117 \cdot Z - 14 + 14 \cdot Z^2) \cdot s]
\end{array}
\]

Computing Time : 2.25 words 227308.

6 Conclusions

We have presented two MAPLE implementations for solving or simplifying sine-cosine equations. The author was able to factoring and decomposing sine-cosine equations of eight-degree with a hundreds of digit coefficients highly complex terms within 20 sc of CPU time on an Power Macintosh 900/300. Therefore, we think that our implementation can now be a useful tool for solving sine-cosine equations.

References


