# Radiroot

## Roots of a Polynomial by Radicals

## A GAP4 Package

Version 2.1

by

Andreas Distler

Mathematical Institute

University of St Andrews

North Haugh

St Andrews, Fife

KY16 9SS

Scotland, UK

November 2006

# Contents

# 1 Introduction

This package provides functionality to deal with one of the fundamental problems in algebra. The roots of a rational polynomial shall be expressed by radicals. This means one is only allowed to use the four basic operations $(+, -, \cdot, \div)$ and to extract roots. For example, a radical expression for the roots of the polynomial $x^4 - x^3 - x^2 + x + 1$ is

$$\tfrac{1}{4} + \tfrac{1}{4}\sqrt{-3} + \tfrac{1}{2}\sqrt{\tfrac{7}{2} + \tfrac{1}{2}\sqrt{-3}}.$$

There are formulas to solve the general equation $x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0 = 0$ up to degree 4. For higher degrees such formulae do not exist ([Abe26]). It was Évariste Galois (1811 – 1832) who discovered that there exists a radical expression for the roots if and only if the Galois group of the polynomial - initially a permutation group on the roots - is solvable [Gal97]. But the task itself was impractical in his days. This package is the first public tool which provides a practical method for solving a polynomial algebraically. The implementation is based on Galois' ideas and the algorithm is described in [Dis05].

The package can provide the result in various forms. As a default an expression is given in a similar way as in the example above. Alternatively, a file containing the roots might be created which is readable by Maple [MGH+05]. In GAP itself some information deduced during the computation is available.

The user should be aware that radical expressions can get very complicated even for polynomials of small degree. Especially because the algorithm will find an irreducible radical expression. That means one gets a root of the given polynomial for every choice of a value of the radicals in the expression. Moreover it is not the aim of this package to give a simplest expression, in any sense.

In Chapter 2 the methods provided by this package are listed and explained.

Chapter 3 gives details about the info class of this package. See Section 7.4 in the GAP reference manual for general information about info classes.

While the installation of the package follows standard GAP rules the Chapter 4 contains information about external programs required by Radiroot in its default setup.

This package uses the interface to KANT [DFK+97], in the package Alnuth, to factorise polynomials over algebraic number fields. This functionality must be available to use the functions in Radiroot.

# 2 Functionality of the Package

This chapter describes the methods available in the Radiroot package.

## 2.1 Methods for Rational Polynomials

1 ► `IsSeparablePolynomial(` *f* `)`

returns `true` if the rational polynomial *f* has simple roots only and `false` otherwise.

2 ► `IsSolvable(` *f* `)`

► `IsSolvablePolynomial(` *f* `)`

returns `true` if the rational polynomial *f* has a solvable Galois group and `false` otherwise. It signals an error if there exists an irreducible factor with degree greater than 15.

For a rational polynomial *f*

3 ► `SplittingField(` *f* `)`

For a rational polynomial *f*, the smallest algebraic extension of the rationals containing all roots of *f* is returned. The field is constructed with `FieldByPolynomial` (see Creation of number fields in Alnuth).

A matrix field isomorphic to the splitting field will be known after the computation and can be accessed using the attribute `IsomorphicMatrixField`.

```
gap> x := Indeterminate( Rationals, "x" );;
gap> f := UnivariatePolynomial( Rationals, [1,3,4,1] );
x^3+4*x^2+3*x+1
gap> L := SplittingField( f );
<algebraic extension over the Rationals of degree 6>
gap> IsomorphicMatrixField( L );
<rational matrix field of degree 6>
gap> y := Indeterminate( L, "y" );;
gap> g := AlgExtEmbeddedPol( L, x^3+4*x^2+3*x+1 );
y^3+!4*y^2+!3*y+!1
gap> Factors( g );
[ y+((-168/47-535/94*a-253/94*a^2-24/47*a^3-3/94*a^4)),
  y+((336/47+488/47*a+253/47*a^2+48/47*a^3+3/47*a^4)),
  y+((20/47-441/94*a-253/94*a^2-24/47*a^3-3/94*a^4)) ]
gap> FactorsPolynomialAlgExt( L, f );
[ y+((-168/47-535/94*a-253/94*a^2-24/47*a^3-3/94*a^4)),
  y+((20/47-441/94*a-253/94*a^2-24/47*a^3-3/94*a^4)),
  y+((336/47+488/47*a+253/47*a^2+48/47*a^3+3/47*a^4)) ]
```

To factorise a polynomial over its splitting field one has to embed the polynomial first, as seen in the example, or use `FactorsPolynomialAlgExt` (see Alnuth) instead of `Factors`. The primitive element of the splitting field is denoted by `a`.

4 ▶ `IsomorphismMatrixField(` *F* `)`

returns a bijective mapping from the number field $F$ to an isomorphic matrix field.

5 ▶ `RootsAsMatrices(` *f* `)`

gives a list of matrices - one for every distinct root of $f$ - whose minimal polynomial is $f$. The field generated by these matrices is a splitting field of $f$. Using `IsomorphismMatrixField` one can map the matrices to the roots of $f$ in `SplittingField(` *f* `)`.

```
gap> Display(RootsAsMatrices(f)[1]);
[ [   0,   1,   0,   0,   0,   0 ],
  [   0,   0,   1,   0,   0,   0 ],
  [  -1,  -3,  -4,   0,   0,   0 ],
  [   0,   0,   0,   0,   1,   0 ],
  [   0,   0,   0,   0,   0,   1 ],
  [   0,   0,   0,  -1,  -3,  -4 ] ]
gap> MinimalPolynomial( Rationals, RootsAsMatrices(f)[1]);
x^3+4*x^2+3*x+1
gap> FieldByMatrices( RootsAsMatrices(f));
<rational matrix field of degree 6>
gap> iso := IsomorphismMatrixField( L );
MappingByFunction( <algebraic extension over the Rationals of degree
6>, <rational matrix field of degree
6>, function( x ) ... end, function( mat ) ... end )
gap> PreImages( iso, RootsAsMatrices( f ) );
[ (-336/47-488/47*a-253/47*a^2-48/47*a^3-3/47*a^4),
  (-20/47+441/94*a+253/94*a^2+24/47*a^3+3/94*a^4),
  (168/47+535/94*a+253/94*a^2+24/47*a^3+3/94*a^4) ]
```

6 ▶ `GaloisGroupOnRoots(` *f* `)`

calculates the Galois group $G$ of the rational polynomial $f$, which has to be separable, as a permutation group with respect to the ordering of the roots of $f$ given as matrices in `RootsAsMatrices`.

```
gap> GaloisGroupOnRoots(f);
Group([ (2,3), (1,2) ])
```

If you only want to get the Galois group itself it is often better to use the function `GaloisType` (see Chapter 64.11 in the GAP reference manual).

## 2.2 Solving a Polynomial by Radicals

1 ▶ `RootsOfPolynomialAsRadicals(` *f* `[,` *mode* `[,` *file* `] ] )`

computes a solution by radicals for the irreducible, rational polynomial $f$ up to degree 15 if this is possible. That is if the Galois group of $f$ is solvable, and returns `fail` otherwise. If it succeeds the function returns the name of the file, containing the computed information.

The user has several options to specify what happens with the results of the computation. Therefore the optional second argument *mode*, a string, can be set to one of the following values:

```
"dvi"
```

To use this option latex and the dvi-viewer xdvi have to be available. It will cause the irreducible radical expression to appear in a new window. The package uses this option as the default.

"latex"

A LaTeX file is generated, which contains the encoding for the expression by radicals. This gives the user the opportunity to adjust the layout of the individual example before displaying the expression.

"maple"

Generates a file containing the roots of $f$ that can be read into Maple [MGH+05].

"off"

In this mode the function does not actually compute a radical expression but is only called for its side effects. Namely, the attributes `SplittingField`, `RootsAsMatrices` and `GaloisGroupOnRoots` are known for $f$ afterwards. This is slightly more effective than calling the corresponding operations one-by-one.

With the optional third argument *file* the user can specify a file name under which the created file will be stored in the current directory. Depending on the option for *mode* an extension like *.tex* might be added automatically.

The computation may take a very long time and can get unfeasible if the degree of $f$ is greater than 7.

2 ► `RootsOfPolynomialAsRadicalsNC(` *f* `[,` *mode* `[,` *file* `] ] )`

has the advantage that it can be used for polynomials with arbitrary degree. It does essentially the same as `RootsOfPolynomialAsRadicals` except that it runs no test on the input before starting the actual computation. In particular, it may run for a very long time until a non-solvable polynomial is recognized as such.

Detailed examples for these two functions can be found in the next section.

## 2.3 Examples

The function `RootsOfPolynomialAsRadicals` does not generate output inside GAP. Depending on the chosen mode, various kinds of files can be created. As an example the polynomial from the introduction will be considered.

```
gap> g := UnivariatePolynomial( Rationals, [1,1,-1,-1,1] );
x^4-x^3-x^2+x+1
gap> RootsOfPolynomialAsRadicals(g);
"/tmp/tmp.8zkw5B/Nst.tex"
```

will cause a dvi file to appear in a new window:

An expression by radicals for the roots of the polynomial $x^4 - x^3 - x^2 + x + 1$ with the $n$-th root of unity $\zeta_n$ and

$\omega_1 = \sqrt{-3}$,

$\omega_2 = \sqrt{\frac{7}{2} - \frac{1}{2}\omega_1}$,

$\omega_3 = \sqrt{\frac{7}{2} + \frac{1}{2}\omega_1}$,

is:

$\frac{1}{4} - \frac{1}{4}\omega_1 + \frac{1}{2}\omega_2$

If one wants to work with the roots, it might be helpful to use Maple [MGH+05], in which an expression like $2^{(1/2)}$ is valid.

```
gap> RootsOfPolynomialAsRadicals(g, "maple");
"/tmp/tmp.k9aTCz/Nst"
```

will create a file with the following content:

```
w1 := (-3)^(1/2);
w2 := ((7/2) + (-1/2)*w1)^(1/2);
w3 := ((7/2) + (1/2)*w1)^(1/2);

a := (1/4) + (1/4)*w1 + (1/2)*w3;
```

After those computations several attributes are known for the polynomial in GAP.

```
gap> RootsOfPolynomialAsRadicalsNC( g, "off" );
gap> time;
0
gap> SplittingField( g );
<algebraic extension over the Rationals of degree 8>
gap> time;
0
gap> GaloisGroupOnRoots( g );
Group([ (2,4), (1,2)(3,4) ])
gap> time;
0
```

# 3 The Info Class of the Package

The `info` mechanism in GAP allows functions to print information during the computation (see Section 7.4 in the GAP reference manual for general information).

1 ► `InfoRadiroot`

is the info class of this package.

2 ► `SetInfoLevel( InfoRadiroot,` *level* `)`

sets the info level for `InfoRadiroot` to *level*, where *level* has to be an integer in the range 0-4.

The default value for `InfoRadiroot` is 1. Information why a function returns `fail` will be given with this setting.

```
gap> InfoLevel(InfoRadiroot);
1
gap> RootsOfPolynomialAsRadicals(x^5-4*x+2);
#I  Polynomial is not solvable.
fail
```

Setting the info level to a higher value will cause messages to show up during single steps of the computation. On level 2 one gets a rough overview. Those who want to go into the details of the algorithm described in [Dis05] and of the implementation itself will find the information on level 3-4 helpful.

To use the package in silent mode the info level can be given the value 0.

# 4 Installation

## 4.1 Getting and Installing this Package

This package is available at

```
http://www.icm.tu-bs.de/ag_algebra/software/distler/radiroot
```

in form of a gzipped tar-archive. For installation instructions see Chapter 75.1 in the GAP reference manual. Normally you will unpack the archive in the `pkg` directory of your GAP version by typing:

```
bash> tar xfz radiroot.tar.gz        # for the gzipped tar-archive
```

## 4.2 Loading the Package

To use the Radiroot package you have to request it explicitly. This is done by calling

```
gap> LoadPackage("radiroot");
----------------------------------------------------------------------------
Loading  RadiRoot 2.1 (Roots of a Polynomial as Radicals)
by Andreas Distler (a.distler@tu-bs.de).
----------------------------------------------------------------------------
true
```

The `LoadPackage` command is described in Section 75.2.1 in the GAP reference manual.

If you want to load the Radiroot package by default, you can put the `LoadPackage` command into your `.gaprc` file (see Section 3.4 in the GAP reference manual).

## 4.3 Additional Requirements

To use Radiroot the package Alnuth 2.2.2 or higher has to be loaded with its full functionality available. This means in particular that KANT [DFK+97] in version 2.4 or 2.5 need to be installed.

In the standard mode a dvi file is created to display the roots of a polynomial. As default the package uses the command `latex` searched for in your system programs to create the dvi file and the command `xdvi` to start the dvi viewer. If you can not use this settings you will have to change the function `RR_Display` in the file `Strings.gi` in the subdirectory `lib` of the package.

# Bibliography


[Abe26] Niels Henrik Abel. Beweis der Unmöglichkeit, algebraische Gleichungen von höheren Graden als dem vierten allgemein aufzulösen. *J. für die reine und angewandte Math. (Crelle's Journal)*, 1:65–84, 1826.

[DFK+97] M. Daberkow, C. Fieker, J. Klüners, M. Pohst, K. Roegner, and K. Wildanger. Kant V4. *J. Symb. Comput.*, 24:267 – 283, 1997.

[Dis05] Andreas Distler. Ein Algorithmus zum Lösen einer Polynomgleichung durch Radikale. Diplomarbeit, TU Braunschweig, 2005.

`http://www.icm.tu-bs.de/ag_algebra/software/distler/Diplom.pdf`.

[Gal97] Évariste Galois. *Oeuvres Mathématiques d'Évariste Galois*. Gauthier-Villars, Paris, 1897.

[MGH+05] Michael B. Monagan, Keith O. Geddes, K. Michael Heal, George Labahn, Stefan M. Vorkoetter, James McCarron, and Paul DeMarco. *Maple 10 Programming Guide*. Maplesoft, Waterloo ON, Canada, 2005.

# Index

This index covers only this manual. A page number in *italics* refers to a whole section which is devoted to the indexed subject. Keywords are sorted with case and spaces ignored, e.g., "`PermutationCharacter`" comes before "permutation group".