```
> restart:
> libname;
  with(SINGULARPLURALlink);
```

```
"C:\Maple11/Cliffordlib", "C:\Maple11/lib", "C:\Brachey.Troy/TNB",
    "C:\Maple11/SINGULARPLURALlinklib"
SINGULARPLURALlink 0.3 beta (May 15, 2008) says Hello...
3 function(s) exported
===>If you find this packages useful, please let us know about your derived work.
===>You can contact us at http://math.tntech.edu/rafal/ or http://clifford.physik.
uni-konstanz.de/~fauser/
===>This package requires CLIFFORD/Bigebra for Maple available from http://math.tn
tech.edu/rafal/
===>Singular:Plural requires nctools.lib and clifford.lib libraries to be installe
d. Consult http://www.singular.uni-kl.de/
Clifford package with 84 functions loaded...
Increase verbosity by infolevel[`function`]=val -- use online help > ?Bigebra[help
]
Bigebra package with 33 functions loaded...
                    [PLURALforClink, PLURALforGlink, SINGULARlink, linkversion]
```

```
> linkversion();
```

```
                  +++++++++++++++++++++++++++++++++++++++++++
    SINGULARPLURALlink - A Maple 11 Package for Linking Maple with Singular and Plural
         Last revised: May 15, 2008 (Source file: SINGULARPLURALlink_M11_0515.mws)
                Copyright 2008 by Rafal Ablamowicz (*) and Bertfried Fauser ($)

                      (*) Department of Mathematics, Box 5054
                 Tennessee Technological University, Cookeville, TN 38505
                     tel: USA (931) 372-3662, fax: USA (931) 372-6353
                              rablamowicz@tntech.edu
                             http://math.tntech.edu/rafal/

                  ($) Universit"at Konstanz, Fachbereich Physik, Fach M678
                              78457 Konstanz, Germany
                           Bertfried.Fauser@uni-konstanz.de
                        http://clifford.physik.uni-konstanz.de/~fauser/
                             http://math.tntech.edu/rafal/

   This package requires CLIFFORD/Bigebra for Maple available from http://math.tntech.edu/rafal/
Singular:Plural requires nctools.lib and clifford.lib libraries to be installed. Consult http://www.s\
    ingular.uni-kl.de

+++++++++++This is SINGULARPLURALlink for Maple 11 version 0.3 with drp+++++++\
```

+++++

>

Package **SINGULARPLURALlink** provides an interface between Maple and Singular with Plural for the purpose of computing commutative Groebner basis in Singular and non commutative Groebner bases in non commutative algebras, e.g., Grassmann, Clifford, etc., in its Plural extension. For now, the link permits only non commutative Groebner bases computation in Grassmann and Clifford algebras of a quadratic form. Input from Maple is sent to Singular:Plural, and results are returned to Maple.

NOTE: In order to compute with Grassmann or Clifford algebras in Maple, download and install package CLIFFORD/Bigebra by Ablamowicz and Fauser from http://www.math.tntech.edu/rafal/

NOTE: In order to compute with Grassmann or Clifford algebras in Singular:Plural, make sure that your Singular:Plural library located in C:\cygwin\usr\share\Singular\LIB directory contains two Singular:Plural libraries "nctools.lib" and "clifford.".

-- Procedure SINGULARlink provides an interface between Maple and Singular for the purpose of computing commutative Groebner basis in Singular and returning result back to Maple. It has been succsessfully tested with Maple 8, 9, 9.5, 10, 11; and 3-0-4. It can be easily modified for the use of other commands in Singular than "std" and "groebner". This procedures is the same as the one posted on Singular's web page

http://www.mathematik.uni-kl.de/ftp/pub/Math/Singular/misc/

as SINGULARlink2.mws, dated 05-Aug-2006. The only difference now is that this procedure that has been successfully tested, is now part of a package (Maple module) called "SINGULARPLURALlink".

**Usage: Procedure SINGULARlink uses the following input arguments:**

SINGULARlink(Id,C,tord,elimorder,input_for_Singular,input_for_Maple,wait);
SINGULARlink(Id,C,tord,elimorder,input_for_Singular,input_for_Maple,wait,'p');

- Id - type "list(polynom)" - a list of polynomials [f1, f2,..., fn] that generate ideal Id, that is, Id = <f1, f2, ..., fn>.
- C - type "nonnegint" - non negative characteristic for the ground field of polynomial coefficients, set 0 for the rationals
- tord - type "symbol" - one of total orders used by Singular/Plural: lp, dp, rp, Dp, ls, ds, Ds, or drp

NOTE: In this version of the link, tord can really be anything as long as Singular:Plural can understand it. For example, in Maple we define a new order drp in n noncommutng variables (x(1..n)) as (a(1:n),rp).

This will be then equivalent to Deg[InvLex] order or a degree order such that x^\alpha > x^\beta if |\alpha| > |\beta|, or, in case of equality |\alpha| = |\beta|, if the right-most nonzero entry in $\alpha - \beta$ is positive.

This order has no special name in Singular:Plural, but we call it for short 'drp'.

- elimorder - type "list" - list of polynomial variables that are to be eliminated in the order in which they appear in the list. This list must be identical to the list of polynomial variables.
- input_for_Singular - type "string" - path to a directory where input file In.txt for Singular will be written to
- input_for_Maple - type "string" - path to a directory where output file Out.txt from Singular for Maple will be written to
- wait - type "nonnegint" or "symbol" - when used with a positive integer it will pass on that integer to "groebner(Id,wait)" as its wait parameter; when used as symbol 'infty', procedure "groebner(Id)" will be invoked

'p' - (optional) type symbol 'd' - for saving files to the disk; 't' for displaying In.txt to the terminal only (see examples below)

-- Procedure PLURALforGlink provides an interface between Maple and Plural for the purpose of computing non commutative Groebner basis in Grassmann algebra using Plural and returning result back to Maple. It has been succsessfully tested with Maple 11; and Singular 3-0-3 and 3-0-4.

**Usage: Procedure PLURALforGlink uses the following input arguments:**

PLURALforGlink(G,C,tord,vars,input_for_Singular,input_for_Maple,wait);
PLURALforGlink(G,C,tord,vars,input_for_Singular,input_for_Maple,wait,'p');

- G - type "list::({clipolynom,climon,clibasmom})" - a list of Clifford or Grassmann polynomials [f1, f2,..., fn] that generate ideal G that is, G = <f1, f2, ..., fn>, in a Grassmann algebra.
- C - type "nonnegint" - non negative characteristic for the ground field of polynomial coefficients, set 0 for the rationals
- tord - type "symbol" - one of total orders used by Singular/Plural: lp, dp, rp, Dp, ls, ds, Ds, or drp:

NOTE: In this version of the link, tord can really be anything as long as Singular:Plural can understand it. For example, in Maple we define a new order drp in n noncommutng variables (x(1..n)) as (a(1:n),rp).

This will be then equivalent to Deg[InvLex] order or a degree order such that x^\alpha > x^\beta if |\alpha| > |\beta|, or, in case of equality |\alpha| = |\beta|, if the right-most nonzero entry in $\alpha - \beta$ is positive.

This order has no special name in Singular:Plural, but we call it for short 'drp'.

- vars - type "list" - list of generators eg, eq,e2,e3,..., e9, of the Grassmann algebra. This list must be identical or larder than the list of variables appearing in the polynomials generating G.
- input_for_Singular - type "string" - path to a directory where input file In.txt for Singular will be

written to
- input_for_Maple - type "string" - path to a directory where output file Out.txt from Singular for Maple will be written to
- wait - type "nonnegint" or "symbol" - when used with a positive integer it will pass on that integer to "groebner(Id,wait)" as its wait parameter; when used as symbol 'infty', procedure "groebner(Id)" will be invoked
'p' - (optional) type symbol 'd' - for saving files to the disk; 't' for displaying In.txt to the terminal only (see examples below)

-- Procedure PLURALforClink provides an interface between Maple and Plural for the purpose of computing non commutative Groebner basis in Clifford algebra Cl(Q) of a quadratic form which could be degenerate, using Plural and returning result back to Maple. It has been succsessfully tested with Maple 11; and Singular 3-0-4.

**Usage: Procedure PLURALforClink uses the following input arguments:**

PLURALforClink(G,C,tord,vars,B,input_for_Singular,input_for_Maple,wait);
PLURALforClink(G,C,tord,vars,B,input_for_Singular,input_for_Maple,wait,'p');

- G - type "list::({clipolynom,climon,clibasmom})" - a list of Clifford or Grassmann polynomials [f1, f2,..., fn] that generate ideal G that is, G = <f1, f2, ..., fn>, in a Grassmann algebra.
- C - type "nonnegint" - non negative characteristic for the ground field of polynomial coefficients, set 0 for the rationals
- tord - type "symbol" - one of total orders used by Singular/Plural: lp, dp, rp, Dp, ls, ds, Ds, or drp
NOTE: In this version of the link, tord can really be anything as long as Singular:Plural can understand it. For example, in Maple we define a new order drp in n noncommutng variables (x(1..n)) as (a(1:n),rp).
This will be then equivalent to Deg[InvLex] order or a degree order such that x^\alpha > x^\beta if |\alpha| > |\beta|, or, in case of equality |\alpha| = |\beta|, if the right-most nonzero entry in $\alpha - \beta$ is positive.
This order has no special name in Singular:Plural, but we call it for short 'drp'.
- vars - type "list" - list of generators eg, eq,e2,e3,..., e9, of the Grassmann algebra. This list must be identical or larder than the list of variables appearing in the polynomials generating G.
- B - type "diagmatrix", a diagonal matrix that defines a quadratic form Q for Cl(Q). The size of this matrix must but at least equal to the largest index in generators appearing in the list vars
- input_for_Singular - type "string" - path to a directory where input file In.txt for Singular will be written to
- input_for_Maple - type "string" - path to a directory where output file Out.txt from Singular for Maple will be written to

- wait - type "nonnegint" or "symbol" - when used with a positive integer it will pass on that integer to "groebner(Id,wait)" as its wait parameter; when used as symbol 'infty', procedure "groebner(Id)" will be invoked
'p' - (optional) type symbol 'd' - for saving files to the disk; 't' for displaying In.txt to the terminal only (see examples below)

**Comments:**

1. Global variable "dirpathSexe" must be defined (see below) with a path to bash.exe (version 3-0-4) to execute Singular 3-0-4.
2. The transfer directories "input_for_Singular" and "input_for_Maple" need not be the same. If either one does not exist, it will be created. These directories are not automatically deleted when Singular is finished computing unless appropriate lines system(cat("rm -r ",pathS)); and system(cat("rm -r ",pathM)); are unremarked. Be VERY CAREFUL when you unremark these lines as you may accidently delete directories you wanted to keep!
3. The eighth parameter 'p' is optional and can only take values 't' and 'd'. When 't' is used, the input script for Singular is displayed to the terminal only: No transfer directory is created and nothing is written to the hard disk. When 'd' is used, the input file In.txt is created, written to the hard disk, then Singular is invoked to execute it and write its output as file Out.txt. Then, Out.txt file is read in and displayed to the terminal/screen. When this parameter is not used, the procedure behaves like if it were used with the parameter 'd'.
4. The output of this procedure is the Groebner basis returned by Singular's "groebner" command. One can easily use "std" command instead.
5. Some error messaging is built in.
6. No warranties of any kind can be made. Permission is granted to use this procedure and modify it at will, although I would appreciate an acknowledgment.
7. A complete interface module allowing a Maple user to use other command from Singular could easily be written.

**References**

[1] T. Stokes, Groebner Bases in Exterior Algebra, J. Automated Reasoning 6 (1990), 233--250

[2] J.L. Bueso, J.Gomez-Torrecillas and A. Verschoren, "Algorithmic Methods in Non-Commutative Algebra - Applications to Quantum Groups", Kluwer Academic Publishers, Dordrecht-Boston-London, 2003

[3] R. Ab{\l}amowicz and B. Fauser, `CLIFFORD' - Maple 11 package for Clifford algebra computations, ver. 11, http://math.tntech.edu/rafal/cliff11/, 2008.

[4] R. Ab{\l}amowicz and B. Fauser, `GfG' - Maple 11 package for \Grobner Bases for Grassmann Algebras, ver. 0.5, http://math.tntech.edu/rafal/GfG/, 2008.

[5] T. Brachey, `TNB - A Maple 11 Package for Computing Groebner bases in Grassmann algebras`, ver. delta 0.3 (March 28, 2008), http://www.math.tntech.edu/rafal/TNB0328/TNB_delta_03_28.zip

[6] ViktorLevandovskyy, Manipulations within Clifford Algebras in Singular:Plural, Private Communication, 2006

Rafal Ablamowicz
Department of Mathematics
Tennessee Technological University, Box 5054
Cookeville, TN 38501
Email: rablamowicz@tntech.edu
Tel: (931) 372-3662
Fax: (931) 372-6353

Cookeville, May 15, 2008

First, we define some paths and directories:

```
>
```

1. Procedure **'linkversion()'** returns information about the current version of the link,s authors' names, etc.

```
> linkversion();
```

```
                    +++++++++++++++++++++++++++++++++++++++++++
 SINGULARPLURALlink - A Maple 11 Package for Linking Maple with Singular and Plural
   Last revised: May 15, 2008 (Source file: SINGULARPLURALlink_M11_0515.mws)
          Copyright 2008 by Rafal Ablamowicz (*) and Bertfried Fauser ($)

                    (*) Department of Mathematics, Box 5054
              Tennessee Technological University, Cookeville, TN 38505
                 tel: USA (931) 372-3662, fax: USA (931) 372-6353
                           rablamowicz@tntech.edu
                          http://math.tntech.edu/rafal/

            ($) Universit"at Konstanz, Fachbereich Physik, Fach M678
                          78457 Konstanz, Germany
                        Bertfried.Fauser@uni-konstanz.de
                    http://clifford.physik.uni-konstanz.de/~fauser/
```

*http://math.tntech.edu/rafal/*

*This package requires CLIFFORD/Bigebra for Maple available from http://math.tntech.edu/rafal/ Singular:Plural requires nctools.lib and clifford.lib libraries to be installed. Consult http://www.s\
ingular.uni-kl.de*

*++++++++++++This is SINGULARPLURALlink for Maple 11 version 0.3 with drp+++++++\
+++++*

```
>
```

First, we define some paths and directories:

```
> #dirpathSexe:="C:/cygwin/usr/local/Singular/2-0-3/ix86-Win/Singula
  r.exe": #A path to Singular executable VERSION 2-0-3
  dirpathSexe:="C:/cygwin/bin/bash -l C:/cygwin/bin/Singular";
  #A path to Singular executable VERSION 3-0-4
  transfer_directoryM:="C:/transferM":      #A user-defined path to
  a directory where Singular should write its output file Out.txt
  for Maple.
  transfer_directoryS:="C:/transferS":      #A user-defined path to
  a directory where Maple should write its input file In.txt for
  Singular.
  input_for_Maple:=transfer_directoryM;     #A path to a directory
  where Singular will write its output file Out.txt for Maple.
  input_for_Singular:=transfer_directoryS;  #A path to a directory
  where Maple will write its input file In.txt for Singular.
```

*dirpathSexe* := "C:/cygwin/bin/bash -l C:/cygwin/bin/Singular"

*input_for_Maple* := "C:/transferM"

*input_for_Singular* := "C:/transferS"

```
>
```

**Example 1:** To test option 't' in the procedure **SINGULARlink**. When used with option  't', the procedure outputs a script file In.txt containing input for Singular to the terminal/screen. It does not write it to the hard disk nor it invokes Singular to execute it:

```
> f:=2*x^10+4*x^9;
  g:=4*y-8-x;
  h:=f*g^2+z^4;
  k:=-5*x^10+x^9*z^2;
  m:=x*y*z;
  L:=map(expand,[f,g]);
  vars:=sort(convert(indets(L),list));
```

```
N:=nops(vars);
```

$$f := 2\,x^{10} + 4\,x^{9}$$

$$g := 4\,y - 8 - x$$

$$h := (2\,x^{10} + 4\,x^{9})\,(4\,y - 8 - x)^{2} + z^{4}$$

$$k := -5\,x^{10} + x^{9}\,z^{2}$$

$$m := x\,y\,z$$

$$L := [2\,x^{10} + 4\,x^{9}, 4\,y - 8 - x]$$

$$vars := [x, y]$$

$$N := 2$$

```
> SINGULARlink([f,g,h,k,m],0,lp,[x,y,z],input_for_Singular,input_for
  _Maple,'infty','t');
```

*Trying to create and display input file for Singular... proceeding...*

```
ring R = 0,(x,y,z),lp;
ideal I =
2*x^10
+ 4*x^9
,
4*y
+ -8
+ -x
,
32*x^10*y^2
+ -160*x^10*y
+ -16*x^11*y
+ 192*x^10
+ 36*x^11
+ 2*x^12
+ 64*x^9*y^2
+ -256*x^9*y
+ 256*x^9
+ z^4
,
-5*x^10
+ x^9*z^2
,
x*y*z
;
short=0;
option(redSB);
ideal J = groebner(I);
write(":w C:/transferM/Out.txt",J);
quit;
```

```
>
```

**Example 2:** To test option 'd' in the above procedure. When used with option 'd', the procedure creates and saves file In.txt file containing input for Singular to the hard disk. It then invokes Singular to execute it and write its output file Out.txt for Maple. It then reads and displays the file Out.txt to the terminal/screen. Option 'infty' means that Maple will wait for Singular to complete its computation infinitely long or, as long as it takes for Singular to complete its computation. If instead 'infty' some positive integer is entered, say 30, then Maple will wait 30 CPU seconds for the output.

```
> f:=2*x^10+4*x^9;
  g:=4*y-8-x;
  h:=f*g^2+z^4;
  k:=-5*x^10+x^9*z^2;
  L:=map(expand,[f,g]);
  vars:=sort(convert(indets(L),list));
  N:=nops(vars);
```

$$f := 2\,x^{10} + 4\,x^{9}$$

$$g := 4\,y - 8 - x$$

$$h := (2\,x^{10} + 4\,x^{9})\,(4\,y - 8 - x)^{2} + z^{4}$$

$$k := -5\,x^{10} + x^{9}\,z^{2}$$

$$L := [2\,x^{10} + 4\,x^{9}, 4\,y - 8 - x]$$

$$vars := [x, y]$$

$$N := 2$$

```
> SINGULARlink([f,g,h],0,dp,[x,y,z],input_for_Singular,input_for_Map
  le,'infty','d');
```

*Transfer directory for Maple already exists... proceeding... waiting for Singular results...*

*Transfer directory for Singular already exists... proceeding... waiting for Singular results...*

$$[x - 4\,y + 8, z^{4}, 2\,y^{10} - 39\,y^{9} + 342\,y^{8} - 1776\,y^{7} + 6048\,y^{6} - 14112\,y^{5} + 22848\,y^{4} - 25344\,y^{3} + 18432\,y^{2} - 7936\,y + 1536]$$

When neither option 't' or 'd' is used, the deafult is to execute Singular:Plural.

```
> SINGULARlink([f,g,h],0,dp,[z,y,x],input_for_Singular,input_for_Map
  le,'infty');
```

*Transfer directory for Maple already exists... proceeding... waiting for Singular results...*

*Transfer directory for Singular already exists... proceeding... waiting for Singular results...*

$$[4\,y - 8 - x, z^{4}, x^{10} + 2\,x^{9}]$$

```
> SINGULARlink([f,g,h],0,lp,[z,y,x],input_for_Singular,input_for_Map
  le,30);
```

*Transfer directory for Maple already exists... proceeding... waiting for Singular results...*

*Transfer directory for Singular already exists... proceeding... waiting for Singular results...*

$$[x^{10} + 2\,x^{9}, 4\,y - 8 - x, z^{4}]$$

```
>
```

**Example 3**. Testing some intended error messages:

```
> SINGULARlink([f,g,h],0,LP,[z,y,x],input_for_Singular,input_for_Map
  le,30);
Warning, third argument, total order, must be one of lp, dp, rp, Dp, ls, ds, or
```

```
Ds. Order drp only makes sense with noncommuting variables! Proceeding anyway...
```

*Transfer directory for Maple already exists... proceeding... waiting for Singular results...*

*Transfer directory for Singular already exists... proceeding... waiting for Singular results...*

*Singular has not returned any results... increase 'wait' time or make it 'infty'*

$[\,]$

```
> SINGULARlink([f,g,h],0,lp,[z,y,X],input_for_Singular,input_for_Map
  le,30);
Error, (in SINGULARPLURALlink:-SINGULARlink) extra variables found in polynomials
that do not appear in elimination list
```

```
> SINGULARlink([f,g,h],0,lp,[z,y,x],input_for_Singular,input_for_Map
  le,'infty','p');
Error, (in SINGULARPLURALlink:-SINGULARlink) last optional argument needs to be
't' (for displaying to terminal) or 'd' (for saving to disk)
```

```
> SINGULARlink([f,g,h],-1,lp,[z,y,x],input_for_Singular,input_for_Ma
  ple,30);
Error, invalid input: SINGULARPLURALlink:-SINGULARlink expects its 2nd argument,
C, to be of type nonnegint, but received -1
```

```
>
```

2. Procedure **PLURALforGlink** formats input for PLURAL to compute a Groebner basis in an ideal I = <F> = <f1,f2,...,fn> in a Grassmann algebra ER. The ideal is defined in PLURAL as an ideal I generated by the said polynomials, for example, as follows:

ring r = 0, (e1,e2,e3,e3,e4,e5,e6,e7,e8,e9), dp;
def ER = Exterior();
setring ER;
poly f1=......;
poly f2=.....;
..................
poly fn=.....;
ideal I=f1,f2,....,fn;
ideal GB=std(I);
GB;.

We prepare to work with Grassmann algebra $\bigwedge_9$.

```
> B:=diag(0$9):
  MonOrderS:=Deg[Lex]:### same as Plural's dp monomial order
  eval(Clifford:-makealiases(9,'ordered')):
  e12;
```

$e12$

**Example 4:** Simple computation of a Groebner basis in a Grassmann algebra. Let I <f1,f2> be an ideal in $\bigwedge_4.$

```
>
```

```
> f1,f2:=2*e1we2+e2-4*e3we4,e1;
  F:=[f1,f2];
  vars:=[seq(e||i,i=1..9)];
  wedge(f1,f2);
```

$$f1, f2 := 2\, e12 + e2 - 4\, e34, e1$$

$$F := [2\, e12 + e2 - 4\, e34, e1]$$

$$vars := [e1, e2, e3, e4, e5, e6, e7, e8, e9]$$

```
Cliplus has been loaded. Definitions for type/climon and type/clipolynom now inclu
de &C and &C[K]. Type ?cliprod for help.
```

$$-e12 - 4\, e134$$

First, we show a typical input file In.txt prepared for Plural.

Note that in CLIFFORD/Bigebra package, that has been loaded by the link, a Grassmann monomial built out of, for example, e1 and e2 is denoted by e1we2 whereas the wedge product is computed as wedge(e1,e2). In general, the wedge product of any two Grasssmann polynomials p1 and p2 is computed with wedge(p1,p2) where wedge is a procedure in the CLIFFORD/Bigebra package. For more information type ?Clifford once you have installed CLIFFORD/Bigebra on your system. Furthermore, note that in Singular:Plural, the wedge product of e1 and e2 is just written as e1*e2 which is different than e2*e1: That is, the star \`*\` implicitly is non-commutative and represents the wedge product in the Grassmann algebra defined as a ring ER with the command Exterior().

Procedure PLURALforGlink translates Maple input, like Grassmann polynomials with wedge product, into polynomials in the exterior algebra ER with the wedge product denoted by \`*\`. That is why in in the input file below, the first polynomial is written as 2*e1*e2+e2-4*e3*e4. Then, before Plural's output is displayed back in Maple, it needs to be translated back into input that CLIFFORD/Bigebra understand.

Note also that a special library package "nctools.lib" is loaded by Singular:Plural.

```
> PLURALforGlink(F,0,dp,vars,input_for_Singular,input_for_Maple,'inf
  ty','t');
```

*Trying to create and display input file for Singular... proceeding...*

```
LIB "nctools.lib";
ring R = 0,(e1,e2,e3,e4,e5,e6,e7,e8,e9),dp;
def ER = Exterior();
setring ER;
ideal I =
2*e1*e2
+ e2
+ -4*e3*e4
,
e1
;
short=0;
option(redSB);
ideal GB = std(I);
write(":w C:/transferM/Out.txt",GB);
quit;
```

Next, we actually compute a Groebner basis for the ideal I generated by the above polynomials. This basis can be compared with the output from Troy Brachey's TNB package for Maple available from

*http://math.tntech.edu/rafal/*

REMEMBER: Deg[Lex] is the same as Plural's dp monomial order whereas Deg[InvLex] order is the same as (a(1:n),rp), called shortly drp.

```
> GB1:=PLURALforGlink(F,0,dp,vars,input_for_Singular,input_for_Maple
,'infty','d');
nops(GB1);
```

*Transfer directory for Maple already exists... proceeding... waiting for Singular results...*

*Transfer directory for Singular already exists... proceeding... waiting for Singular results...*

$$GB1 := [e1, 4\,e34 - e2, e24, e23]$$

$$4$$

Thus, GB1 is a Groebner basis for the ideal I = <f1,f2> for the dp order.

**Example 5.** We change order in the above example to drp = (a(1:n),rp) which is the same as Deg[InvLex] order in CLIFFORD/Bigebra. This is a graded rp order such that x^\alpha > x^\beta if |\alpha| > |\beta|, or, in case of equality |\alpha| = |\beta|, if the right-most nonzero entry in $\alpha - \beta$ is positive. **This order has no special name in Singular:Plural**, but we call it for short 'drp' as it is a global order based on Plurals' rp order with the added gradation.

```
> PLURALforGlink(F,0,drp,vars,input_for_Singular,input_for_Maple,'in
fty','t');
```

*Trying to create and display input file for Singular... proceeding...*

```
LIB "nctools.lib";
ring R = 0,(e1,e2,e3,e4,e5,e6,e7,e8,e9),(a(1:9),rp);
def ER = Exterior();
setring ER;
ideal I =
2*e1*e2
+ e2
+ -4*e3*e4
,
e1
;
short=0;
option(redSB);
ideal GB = std(I);
write(":w C:/transferM/Out.txt",GB);
```

```
quit;
```

```
>
> PLURALforGlink(F,0,drp,vars,input_for_Singular,input_for_Maple,'in
  fty','d');
```

*Transfer directory for Maple already exists... proceeding... waiting for Singular results...*

*Transfer directory for Singular already exists... proceeding... waiting for Singular results...*

$$[e1, e23, e24, 4\, e34 - e2]$$

```
>
```

**Example 6:** Simple computation of a Groebner basis in a Grassmann algebra. Let I = <f1,f2> be an ideal in $\bigwedge_6.$ We use monomial order drp = Deg[InvLex].

```
>
> f1,f2:=e56-e23,e45-e13;
  F:=[f1,f2];
  vars:=[seq(e||i,i=1..6)];
  wedge(f1,f2);
```

$$f1, f2 := e56 - e23, e45 - e13$$

$$F := [e56 - e23, e45 - e13]$$

$$vars := [e1, e2, e3, e4, e5, e6]$$

$$-e2345 - e1356$$

```
> PLURALforGlink(F,0,drp,vars,input_for_Singular,input_for_Maple,'in
  fty','t');
```

*Trying to create and display input file for Singular... proceeding...*

```
LIB "nctools.lib";
ring R = 0,(e1,e2,e3,e4,e5,e6),(a(1:6),rp);
def ER = Exterior();
setring ER;
ideal I =
e5*e6
+ -1*e2*e3
,
e4*e5
+ -1*e1*e3
;
short=0;
option(redSB);
ideal GB = std(I);
write(":w C:/transferM/Out.txt",GB);
quit;
```

```
>
> PLURALforGlink(F,0,drp,vars,input_for_Singular,input_for_Maple,'in
  fty','d');
```

*Transfer directory for Maple already exists... proceeding... waiting for Singular results...*

*Transfer directory for Singular already exists... proceeding... waiting for Singular results...*

$$[e45 - e13, e56 - e23, e134, e135, e235, e136 - e234, e236]$$

```
>
```

3. Procedure **PLURALforClink** formats input for PLURAL to compute a Groebner basis in an ideal I = <F> = <f1,f2,...,fn> in a Clifford algebra Cl(Q) of a quadratic form Q which may be degenerate.

Notice that this time, once an appropriate Clifford algebra is defined in Plural, the asterisk `*` denotes Clifford product in the Clifford algebra Cl(Q). Thus, procedure **PLURALforClink** translates Maple input which, as default, is expressed as Grassmann polynomials with the wedge product, into polynomials in the Clifford algebra with the wedge product denoted by `*`. That is why in in the input file below, the second polynomial is written as 3*e1*e2+e1*e2. Then, before Plural's output is displayed back in Maple, it is reduced NF modulo the defining relations in the Clifford algebra. Then, it is written to a file Out.txt, read into Maple, and translated back into input that CLIFFORD/Bigebra understand.

We prepare to work with a Clifford algebra Cl(R^2) by defining a bilinear form B which is diagonal, that is, Q = B.

**Example 7:** (Following Example 4.1 from ViktorLevandovskyy, Private Communication, 2006)

We compute a Groebner basis for the ideal F in Clifford algebra Cl(R^2) w.r.t. to the order dp=Deg[Lex].

```
> with(linalg):
  B:=diag(1,1);
  f1:=e1+2*e2;
  f2:=3*e1+cmul(e1,e2);
  F:=[f1,f2];
  clidata([2,0]);
  vrs:=[e1,e2];
```

$$B := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$f1 := e1 + 2\, e2$$

$$f2 := 3\, e1 + e12$$

$$F := [e1 + 2\, e2,\ 3\, e1 + e12]$$

$$\left[ real,\ 2,\ simple,\ \frac{Id}{2} + \frac{e1}{2},\ [Id,\ e2],\ [Id],\ [Id,\ e2] \right]$$

$$vrs := [e1,\ e2]$$

Showing an input file for Singular:Plural:

```
> PLURALforClink(F,0,dp,vrs,B,input_for_Singular,input_for_Maple,'in
  fty','t');
```

*Trying to create and display input file for Singular... proceeding...*

```
LIB "clifford.lib";
ring R = 0,(e1,e2),dp;
option(redSB);
option(redTail);
matrix M[2][2];
M[1,1]=2;M[2,2]=2;
clifAlgebra(M);
qring Q =twostd(clQuot);
ideal I =
e1
+ 2*e2
,
3*e1
+ e1*e2
;
short=0;
ideal GB = std(I);
if(size(GB)>=1){int j=1; for ( j=1; j <= size(GB); j++) {GB[j]=NF(GB[j],std(0));}}
;
write(":w C:/transferM/Out.txt",GB);
quit;
```

```
> PLURALforClink(F,0,dp,vrs,B,input_for_Singular,input_for_Maple,'in
  fty','d');
```

*Transfer directory for Maple already exists... proceeding... waiting for Singular results...*

*Transfer directory for Singular already exists... proceeding... waiting for Singular results...*

$$[Id]$$

**Thus, the above does show that the ideal is the entire algebra Cl(R^2) since it contains the identity element Id. However, we can check that as follows:**

**Example 8:** Following Example 4.1 from ViktorLevandovskyy, Private Communication, 2006, but using drp order defined above.

We compute a Groebner basis for the ideal F in Clifford algebra Cl(R^2) w.r.t. to the order drp=Deg[InvLex].

```
> with(linalg):
  B:=diag(1,1);
  f1:=e1+2*e2;
  f2:=3*e1+cmul(e1,e2);
  F:=[f1,f2];
  clidata([2,0]);
  vrs:=[e1,e2];
```

$$B := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$f1 := e1 + 2\, e2$$

$$f2 := 3\, e1 + e12$$

$$F := [e1 + 2\, e2, 3\, e1 + e12]$$

$$\left[real, 2, simple, \frac{Id}{2}+\frac{e1}{2}, [Id, e2], [Id], [Id, e2]\right]$$

$$vrs := [e1, e2]$$

Showing an input file for Singular:Plural:

```
> PLURALforClink(F,0,drp,vrs,B,input_for_Singular,input_for_Maple,'infty','t');
```

*Trying to create and display input file for Singular... proceeding...*

```
LIB "clifford.lib";
ring R = 0,(e1,e2),(a(1:2),rp);
option(redSB);
option(redTail);
matrix M[2][2];
M[1,1]=2;M[2,2]=2;
clifAlgebra(M);
qring Q =twostd(clQuot);
ideal I =
e1
+ 2*e2
,
3*e1
+ e1*e2
;
short=0;
ideal GB = std(I);
if(size(GB)>=1){int j=1; for ( j=1; j <= size(GB); j++) {GB[j]=NF(GB[j],std(0));}}
;
write(":w C:/transferM/Out.txt",GB);
quit;
```

```
> F:=PLURALforClink(F,0,drp,vrs,B,input_for_Singular,input_for_Maple,'infty','d');
```

*Transfer directory for Maple already exists... proceeding... waiting for Singular results...*

*Transfer directory for Singular already exists... proceeding... waiting for Singular results...*

$$F := [Id]$$

**Thus, the above does show that the ideal is the entire algebra Cl(R^2) since it contains only identity element Id.**

```
>
```

**Example 9** (Following Example 4.2 from ViktorLevandovskyy, Private Communication, 2006)

We compute a Groebner basis for the ideal F in Clifford algebra Cl(R^3) w.r.t. to the order dp=Deg[Lex].

```
> B:=diag(1,1,1);
  f1:=3*Id-e1;
  f2:=cmul(e1,e3)+e2;
  F:=[f1,f2];
```

```
clidata([3,0]);
vrs:=[e1,e2,e3];
```

$$B := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$f1 := 3\, Id - e1$$

$$f2 := e13 + e2$$

$$F := [3\, Id - e1, e13 + e2]$$

$$\left[ complex, 2, simple, \frac{Id}{2} + \frac{e1}{2}, [Id, e2, e3, e23], [Id, e23], [Id, e2] \right]$$

$$vrs := [e1, e2, e3]$$

```
> PLURALforClink(F,0,dp,vrs,B,input_for_Singular,input_for_Maple,'in
  fty','t');
```

*Trying to create and display input file for Singular... proceeding...*

```
LIB "clifford.lib";
ring R = 0,(e1,e2,e3),dp;
option(redSB);
option(redTail);
matrix M[3][3];
M[1,1]=2;M[2,2]=2;M[3,3]=2;
clifAlgebra(M);
qring Q =twostd(clQuot);
ideal I =
3*1
+ -1*e1
,
e1*e3
+ e2
;
short=0;
ideal GB = std(I);
if(size(GB)>=1){int j=1; for ( j=1; j <= size(GB); j++) {GB[j]=NF(GB[j],std(0));}}
;
write(":w C:/transferM/Out.txt",GB);
quit;
```

```
> PLURALforClink(F,0,dp,vrs,B,input_for_Singular,input_for_Maple,'in
  fty','d');
```

*Transfer directory for Maple already exists... proceeding... waiting for Singular results...*

*Transfer directory for Singular already exists... proceeding... waiting for Singular results...*

$$[Id]$$

```
>
```

**Example 10** We compute a Groebner basis for the ideal F in Clifford algebra Cl(R^3) w.r.t. to the order drp=Deg[InvLex].

Notice that the ideal I = <F> = <f1,f2,f3,f4> = Cl(R^3)f is a spinor ideal in Cl(R^3) generated by a primitive idempotent f=f1.

```
> B:=diag(1,1,1);
  f1:=1/2*(Id+e1);
  f2:=cmul(e2,f1);
  f3:=cmul(e3,f1);
  f4:=cmul(e23,f1);
  F:=[f1,f2,f3,f4];
  clidata([3,0]);
  vrs:=[e1,e2,e3];
```

$$B := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$f1 := \frac{Id}{2} + \frac{e1}{2}$$

$$f2 := -\frac{e12}{2} + \frac{e2}{2}$$

$$f3 := -\frac{e13}{2} + \frac{e3}{2}$$

$$f4 := \frac{e23}{2} + \frac{e123}{2}$$

$$F := \left[\frac{Id}{2} + \frac{e1}{2}, -\frac{e12}{2} + \frac{e2}{2}, -\frac{e13}{2} + \frac{e3}{2}, \frac{e23}{2} + \frac{e123}{2}\right]$$

$$\left[complex, 2, simple, \frac{Id}{2} + \frac{e1}{2}, [Id, e2, e3, e23], [Id, e23], [Id, e2]\right]$$

$$vrs := [e1, e2, e3]$$

```
> PLURALforClink(F,0,drp,vrs,B,input_for_Singular,input_for_Maple,'i
  nfty','t');
```

*Trying to create and display input file for Singular... proceeding...*

```
LIB "clifford.lib";
ring R = 0,(e1,e2,e3),(a(1:3),rp);
option(redSB);
option(redTail);
matrix M[3][3];
M[1,1]=2;M[2,2]=2;M[3,3]=2;
clifAlgebra(M);
qring Q =twostd(clQuot);
ideal I =
1/2*1
+ 1/2*e1
,
-1/2*e1*e2
+ 1/2*e2
,
-1/2*e1*e3
+ 1/2*e3
```

```
,
1/2*e2*e3
+ 1/2*e1*e2*e3
;
short=0;
ideal GB = std(I);
if(size(GB)>=1){int j=1; for ( j=1; j <= size(GB); j++) {GB[j]=NF(GB[j],std(0));}}
;
write(":w C:/transferM/Out.txt",GB);
quit;
```

```
> F:=PLURALforClink(F,0,drp,vrs,B,input_for_Singular,input_for_Maple
  ,'infty','d');
```

*Transfer directory for Maple already exists... proceeding... waiting for Singular results...*

*Transfer directory for Singular already exists... proceeding... waiting for Singular results...*

$$F := [e1 + Id]$$

Notice, as it should not be a surprise, that the Groebner basis for the left ideal is, up to a scalar coefficient, the idempotent f1 as S = Cl(R^3)f1 = <f1>.

```
>
```

**Example 11** (following example 4.3 from ViktorLevandovskyy, Private Communication)

We compute a Groebner basis for the ideal F in Clifford algebra Cl(B) w.r.t. to the order dp=Deg[Lex]. However, in this case B=0, hence the Clifford algebra Cl(B) = Grassmann algebra. We can also use here procedure PLURALforGlink.

```
> B:=diag(0,0,0); #Grassmann algebra
  f1:=Id+e1+cmul(e2,e3);
  f2:=cmul(e1,e3)+cmul(e1,e2,e3);
  f3:=e1-2*e3;
  F:=[f1,f2,f3];
  vrs:=[e1,e2,e3];
```

$$B := \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$f1 := Id + e1 + e23$$

$$f2 := e13 + e123$$

$$f3 := e1 - 2\,e3$$

$$F := [Id + e1 + e23, e13 + e123, e1 - 2\,e3]$$

$$vrs := [e1, e2, e3]$$

```
> PLURALforClink(F,0,dp,vrs,B,input_for_Singular,input_for_Maple,'in
  fty','t');
```

*Trying to create and display input file for Singular... proceeding...*

```
LIB "clifford.lib";
ring R = 0,(e1,e2,e3),dp;
option(redSB);
option(redTail);
```

```
matrix M[3][3];
M[1,1]=0;M[2,2]=0;M[3,3]=0;
clifAlgebra(M);
qring Q =twostd(clQuot);
ideal I =
1
+ e1
+ e2*e3
,
e1*e3
+ e1*e2*e3
,
e1
+ -2*e3
;
short=0;
ideal GB = std(I);
if(size(GB)>=1){int j=1; for ( j=1; j <= size(GB); j++) {GB[j]=NF(GB[j],std(0));}}
;
write(":w C:/transferM/Out.txt",GB);
quit;
```

```
> PLURALforGlink(F,0,dp,vrs,input_for_Singular,input_for_Maple,'infty','t');
```

```
Trying to create and display input file for Singular... proceeding...
LIB "nctools.lib";
ring R = 0,(e1,e2,e3),dp;
def ER = Exterior();
setring ER;
ideal I =
1
+ e1
+ e2*e3
,
e1*e3
+ e1*e2*e3
,
e1
+ -2*e3
;
short=0;
option(redSB);
ideal GB = std(I);
write(":w C:/transferM/Out.txt",GB);
quit;
```

```
>
```

```
> G1:=PLURALforClink(F,0,dp,vrs,B,input_for_Singular,input_for_Maple,'infty','d');
```

```
Transfer directory for Maple already exists... proceeding... waiting for Singular results...
Transfer directory for Singular already exists... proceeding... waiting for Singular results...
G1 := [Id]
```

```
> G2:=PLURALforGlink(F,0,dp,vrs,input_for_Singular,input_for_Maple,'infty','d');
```

```
Transfer directory for Maple already exists... proceeding... waiting for Singular results...
```

*Transfer directory for Singular already exists... proceeding... waiting for Singular results...*

$$G2 := [Id]$$

```
>
```

**Example 12** Following example 4.3 from ViktorLevandovskyy, Private Communication, 2006, but for drp order.

We compute a Groebner basis for the ideal F in Clifford algebra Cl(B) w.r.t. to the order dp=Deg[Lex]. However, in this case B=0, hence the Clifford algebra Cl(B) = Grassmann algebra. We can also use here procedure **PLURALforGlink**.

```
> B:=diag(0,0,0); #Grassmann algebra
  f1:=Id+e1+cmul(e2,e3);
  f2:=cmul(e1,e3)+cmul(e1,e2,e3);
  f3:=e1-2*e3;
  F:=[f1,f2,f3];
  vrs:=[e1,e2,e3];
```

$$B := \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$f1 := Id + e1 + e23$$

$$f2 := e13 + e123$$

$$f3 := e1 - 2\, e3$$

$$F := [Id + e1 + e23, e13 + e123, e1 - 2\, e3]$$

$$vrs := [e1, e2, e3]$$

```
> PLURALforClink(F,0,drp,vrs,B,input_for_Singular,input_for_Maple,'i
  nfty','t');
```

*Trying to create and display input file for Singular... proceeding...*

```
LIB "clifford.lib";
ring R = 0,(e1,e2,e3),(a(1:3),rp);
option(redSB);
option(redTail);
matrix M[3][3];
M[1,1]=0;M[2,2]=0;M[3,3]=0;
clifAlgebra(M);
qring Q =twostd(clQuot);
ideal I =
1
+ e1
+ e2*e3
,
e1*e3
+ e1*e2*e3
,
e1
+ -2*e3
;
short=0;
```

```
ideal GB = std(I);
if(size(GB)>=1){int j=1; for ( j=1; j <= size(GB); j++) {GB[j]=NF(GB[j],std(0));}}
;
write(":w C:/transferM/Out.txt",GB);
quit;
```

```
> PLURALforClink(F,0,drp,vrs,B,input_for_Singular,input_for_Maple,'i
  nfty','d');
  PLURALforGlink(F,0,drp,vrs,input_for_Singular,input_for_Maple,'inf
  ty','d');
```

*Transfer directory for Maple already exists... proceeding... waiting for Singular results...*

*Transfer directory for Singular already exists... proceeding... waiting for Singular results...*

$$[Id]$$

*Transfer directory for Maple already exists... proceeding... waiting for Singular results...*

*Transfer directory for Singular already exists... proceeding... waiting for Singular results...*

$$[Id]$$

```
>
```

**Example 13** (following example 4.4 from ViktorLevandovskyy, Private Communication)

We compute a Groebner basis for the ideal F in Clifford algebra Cl(B) w.r.t. to the order dp=Deg[Lex] whene B=0, that is, Cl(B) = Exterior Algebra. However, ideal is smaller.

```
> B:=diag(0,0,0); #Grassmann algebra
  f1:=cmul(e1,e2,e3)+cmul(e2,e3);
  f2:=e1-2*e3;
  F:=[f1,f2];
  vrs:=[e1,e2,e3];
```

$$B := \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$f1 := e123 + e23$$

$$f2 := e1 - 2\,e3$$

$$F := [e123 + e23,\ e1 - 2\,e3]$$

$$vrs := [e1,\ e2,\ e3]$$

```
> PLURALforClink(F,0,dp,vrs,B,input_for_Singular,input_for_Maple,'in
  fty','t');
```

*Trying to create and display input file for Singular... proceeding...*

```
LIB "clifford.lib";
ring R = 0,(e1,e2,e3),dp;
option(redSB);
option(redTail);
matrix M[3][3];
M[1,1]=0;M[2,2]=0;M[3,3]=0;
clifAlgebra(M);
qring Q =twostd(clQuot);
```

```
ideal I =
e1*e2*e3
+ e2*e3
,
e1
+ -2*e3
;
short=0;
ideal GB = std(I);
if(size(GB)>=1){int j=1; for ( j=1; j <= size(GB); j++) {GB[j]=NF(GB[j],std(0));}}
;
write(":w C:/transferM/Out.txt",GB);
quit;
```

```
> PLURALforGlink(F,0,dp,vrs,input_for_Singular,input_for_Maple,'infty','t');
```

*Trying to create and display input file for Singular... proceeding...*

```
LIB "nctools.lib";
ring R = 0,(e1,e2,e3),dp;
def ER = Exterior();
setring ER;
ideal I =
e1*e2*e3
+ e2*e3
,
e1
+ -2*e3
;
short=0;
option(redSB);
ideal GB = std(I);
write(":w C:/transferM/Out.txt",GB);
quit;
```

```
> PLURALforClink(F,0,dp,vrs,B,input_for_Singular,input_for_Maple,'infty','d');
```

*Transfer directory for Maple already exists... proceeding... waiting for Singular results...*

*Transfer directory for Singular already exists... proceeding... waiting for Singular results...*

$$[e1 - 2\,e3, e23]$$

We can convert the above basis expressed in Grassmann basis to Clifford unevaluated basis using command 'Cliplus:-cliexpand'.

Note: Package 'Cliplus' is an extension of CLIFFORD/Bigebra. It loads automatically with CLIFFORD.:

```
> map(Cliplus:-cliexpand,%);
```

$$[e1 - 2\,e3, e2 \mathbin{\&C} e3]$$

```
> map(Cliplus:-clieval,%);
```

$$[e1 - 2\,e3, e23]$$

```
> PLURALforGlink(F,0,dp,vrs,input_for_Singular,input_for_Maple,'infty','d');
```

*Transfer directory for Maple already exists... proceeding... waiting for Singular results...*

*Transfer directory for Singular already exists... proceeding... waiting for Singular results...*

$$[e1 - 2\ e3, e23]$$

```
> map(Cliplus:-cliexpand,%);
```

$$[e1 - 2\ e3, e2\ \&C\ e3]$$

```
> map(Cliplus:-clieval,%);
```

$$[e1 - 2\ e3, e23]$$

We can convert the above basis expressed in Grassmann basis to Clifford unevaluated basis using command 'Cliplus:-cliexpand'. The result is the same when using PLURALforClink procedure.

Additional examples:

**Example 14:**

We continue with this algebra Cl(2,0):

```
> B:=diag(1,1);
  eval(makealiases(2)):
  clid:=clidata([2,0]);
  N:=nops(clid[5]);
```

$$B := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$clid := \left[ real, 2, simple, \frac{Id}{2} + \frac{e1}{2}, [Id, e2], [Id], [Id, e2] \right]$$

$$N := 2$$

```
> f:=clid[4];
  f:=eval(f);
  for i from 1 to N do
        g||i:=clid[5][i];
  end do;
  for i from 1 to N do
      f||i:=cmul(g||i,f);
  end do;
  F:=[f1,f2];
```

$$f := \frac{Id}{2} + \frac{e1}{2}$$

$$f := \frac{Id}{2} + \frac{e1}{2}$$

$$g1 := Id$$

$$g2 := e2$$

$$f1 := \frac{Id}{2} + \frac{e1}{2}$$

$$f2 := -\frac{e12}{2} + \frac{e2}{2}$$

$$F := \left[\frac{Id}{2} + \frac{e1}{2}, -\frac{e12}{2} + \frac{e2}{2}\right]$$

```
> vrs:=[seq(e||k,k=1..maxindex(F))];
```

$$vrs := [e1, e2]$$

```
> PLURALforClink(F,0,dp,vrs,B,input_for_Singular,input_for_Maple,'infty','t');
```

*Trying to create and display input file for Singular... proceeding...*

```
LIB "clifford.lib";
ring R = 0,(e1,e2),dp;
option(redSB);
option(redTail);
matrix M[2][2];
M[1,1]=2;M[2,2]=2;
clifAlgebra(M);
qring Q =twostd(clQuot);
ideal I =
1/2*1
+ 1/2*e1
,
-1/2*e1*e2
+ 1/2*e2
;
short=0;
ideal GB = std(I);
if(size(GB)>=1){int j=1; for ( j=1; j <= size(GB); j++) {GB[j]=NF(GB[j],std(0));}}
;
write(":w C:/transferM/Out.txt",GB);
quit;
```

```
> GB:=PLURALforClink(F,0,dp,vrs,B,input_for_Singular,input_for_Maple,'infty','d');
  ff:=op(GB)/2;
```

*Transfer directory for Maple already exists... proceeding... waiting for Singular results...*

*Transfer directory for Singular already exists... proceeding... waiting for Singular results...*

$$GB := [e1 + Id]$$

$$ff := \frac{Id}{2} + \frac{e1}{2}$$

```
> 'ff' = cmul(ff,ff);
```

$$ff = \frac{Id}{2} + \frac{e1}{2}$$

```
>
```

**Example 15:** We continue with this algebra Cl(3,1):

```
> B:=diag(1,1,1,-1);
  eval(makealiases(4)):
```

```
clid:=clidata([3,1]);
N:=nops(clid[5]);
```

$$B := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

$$clid := \left[ real, 4, simple, \text{'}Clifford{:}\text{-}cmulQ\text{'}\left(\frac{Id}{2} + \frac{e1}{2}, \frac{Id}{2} + \frac{e34}{2}\right), [Id, e2, e3, e23], [Id], [Id, e2, e3, e23] \right]$$

$$N := 4$$

```
> f:=clid[4];
  f:=eval(f);
  for i from i to N do
        g||i:=clid[5][i];
  end do;
  for i from 1 to N do
      p||i:=cmul(g||i,f);
  end do;
  F:=[p1,p2,p3,p4];
```

$$f := Clifford{:}\text{-}cmulQ\left(\frac{Id}{2} + \frac{e1}{2}, \frac{Id}{2} + \frac{e34}{2}\right)$$

$$f := \frac{Id}{4} + \frac{e1}{4} + \frac{e34}{4} + \frac{e134}{4}$$

$$g3 := e3$$

$$g4 := e23$$

$$p1 := \frac{Id}{4} + \frac{e34}{4} + \frac{e1}{4} + \frac{e134}{4}$$

$$p2 := -\frac{e12}{4} + \frac{e2}{4} + \frac{e234}{4} - \frac{e1234}{4}$$

$$p3 := -\frac{e13}{4} - \frac{e14}{4} + \frac{e3}{4} + \frac{e4}{4}$$

$$p4 := \frac{e23}{4} + \frac{e123}{4} + \frac{e124}{4} + \frac{e24}{4}$$

$$F := \left[ \frac{Id}{4} + \frac{e34}{4} + \frac{e1}{4} + \frac{e134}{4}, -\frac{e12}{4} + \frac{e2}{4} + \frac{e234}{4} - \frac{e1234}{4}, -\frac{e13}{4} - \frac{e14}{4} + \frac{e3}{4} + \frac{e4}{4}, \frac{e23}{4} + \frac{e123}{4} + \frac{e124}{4} + \frac{e24}{4} \right]$$

```
> vrs:=[seq(e||k,k=1..maxindex(F))];
```

$$vrs := [e1, e2, e3, e4]$$

We proceed to compute a left Groebner basis in the spinor ideal S = Cl(3,1)f1.

```
> PLURALforClink(F,0,dp,vrs,B,input_for_Singular,input_for_Maple,'infty','t');
```

*Trying to create and display input file for Singular... proceeding...*

```
LIB "clifford.lib";
ring R = 0,(e1,e2,e3,e4),dp;
option(redSB);
option(redTail);
matrix M[4][4];
M[1,1]=2;M[2,2]=2;M[3,3]=2;M[4,4]=-2;
clifAlgebra(M);
qring Q =twostd(clQuot);
ideal I =
1/4*1
+ 1/4*e3*e4
+ 1/4*e1
+ 1/4*e1*e3*e4
,
-1/4*e1*e2
+ 1/4*e2
+ 1/4*e2*e3*e4
+ -1/4*e1*e2*e3*e4
,
-1/4*e1*e3
+ -1/4*e1*e4
+ 1/4*e3
+ 1/4*e4
,
1/4*e2*e3
+ 1/4*e1*e2*e3
+ 1/4*e1*e2*e4
+ 1/4*e2*e4
;
short=0;
ideal GB = std(I);
if(size(GB)>=1){int j=1; for ( j=1; j <= size(GB); j++) {GB[j]=NF(GB[j],std(0));}}
;
write(":w C:/transferM/Out.txt",GB);
quit;
```

```
> GB:=PLURALforClink(F,0,dp,vrs,B,input_for_Singular,input_for_Maple,'infty','d');
  g:=op(GB)/4;
```

*Transfer directory for Maple already exists... proceeding... waiting for Singular results...*

*Transfer directory for Singular already exists... proceeding... waiting for Singular results...*

$$GB := [e13 + e14 - e3 - e4]$$

$$g := \frac{e13}{4} + \frac{e14}{4} - \frac{e3}{4} - \frac{e4}{4}$$

Notice that this element is nilpotent:

```
> cmul(g,g);
```

$$0$$

However, as Cl(3,1) is a simple Artinian algebra, any of its left and right ideals is generated by an idempotent. It turns out that in this case g = -cmul(e3,f):

```
> e3f:=-cmul(e3,f);
```

$$e3f := \frac{e13}{4}+\frac{e14}{4}-\frac{e3}{4}-\frac{e4}{4}$$

```
> g-e3f;
```

$$0$$

```
> cmul(e3,f,e3,f);
```

$$0$$

```
> -cmul(e3,g)=f;
```

$$\frac{Id}{4}+\frac{e34}{4}+\frac{e1}{4}+\frac{e134}{4}=\frac{Id}{4}+\frac{e34}{4}+\frac{e1}{4}+\frac{e134}{4}$$

**Example 16:** We continue with a semisimple algebra.

```
>
> all_sigs(1..9, 'real', 'semisimple');
```

$$[[0, 7], [1, 0], [1, 8], [2, 1], [3, 2], [4, 3], [5, 4], [9, 0]]$$

```
> clidata([2,1]);
```

$$\left[ real, 2, semisimple, 'Clifford{:}-cmulQ'\left(\frac{Id}{2}+\frac{e1}{2}, \frac{Id}{2}+\frac{e23}{2}\right), [Id, e2], [Id], [Id, e2]\right]$$

```
> B:=diag(1,1,-1);
  eval(makealiases(4)):
  clid:=clidata([2,1]);
  N:=nops(clid[5]);
```

$$B := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

$$clid := \left[ real, 2, semisimple, 'Clifford{:}-cmulQ'\left(\frac{Id}{2}+\frac{e1}{2}, \frac{Id}{2}+\frac{e23}{2}\right), [Id, e2], [Id], [Id, e2]\right]$$

$$N := 2$$

```
> f:=clid[4];
  f:=eval(f);
  frev:=gradeinv(f);
  for i from i to N do
        g||i:=clid[5][i];
  end do;
```

```
for i from 1 to N do
    p||i:=cmul(g||i,f);
end do;
F:=[p1,p2];
```

$$f := Clifford\text{:-}cmulQ\left(\frac{Id}{2}+\frac{e1}{2}, \frac{Id}{2}+\frac{e23}{2}\right)$$

$$f := \frac{Id}{4}+\frac{e1}{4}+\frac{e23}{4}+\frac{e123}{4}$$

$$frev := \frac{Id}{4}-\frac{e1}{4}+\frac{e23}{4}-\frac{e123}{4}$$

$$p1 := \frac{Id}{4}+\frac{e23}{4}+\frac{e123}{4}+\frac{e1}{4}$$

$$p2 := -\frac{e13}{4}-\frac{e12}{4}+\frac{e2}{4}+\frac{e3}{4}$$

$$F := \left[\frac{Id}{4}+\frac{e23}{4}+\frac{e123}{4}+\frac{e1}{4}, -\frac{e13}{4}-\frac{e12}{4}+\frac{e2}{4}+\frac{e3}{4}\right]$$

```
> vrs:=[seq(e||k,k=1..maxindex(F))];
```

$$vrs := [e1, e2, e3]$$

```
> GB:=PLURALforClink(F,0,dp,vrs,B,input_for_Singular,input_for_Maple
,'infty','d');
```

*Transfer directory for Maple already exists... proceeding... waiting for Singular results...*

*Transfer directory for Singular already exists... proceeding... waiting for Singular results...*

$$GB := [e12+e13-e2-e3]$$

```
> g:=op(GB)/4;
```

$$g := \frac{e12}{4}+\frac{e13}{4}-\frac{e2}{4}-\frac{e3}{4}$$

```
> cmul(g,g);
```

$$0$$

```
> e2f:=-cmul(e2,f);
```

$$e2f := \frac{e12}{4}+\frac{e13}{4}-\frac{e2}{4}-\frac{e3}{4}$$

```
> g-e2f;
```

$$0$$

```
>
>
>
>
```

End

Cookeville, May 15, 2008