The GaussForHomalg Package Manual

Gauss Functionality for homalg

Version 2009.11.09

November 2009

Simon Goertzen

Simon Goertzen
— Email: simon.goertzen@rwth-aachen.de
— Homepage: http://wwwb.math.rwth-aachen.de/goertzen/
— Address: Lehrstuhl B für Mathematik
RWTH Aachen
Templergraben 64
52062 Aachen
(Germany)

Abstract

This document explains the primary uses of the GaussForHomalg package. Included in this manual is a documented list of the most important methods and functions you will need.

Copyright

Acknowledgements

Many thanks to Mohamed Barakat and the Lehrstuhl B für Mathematik at RWTH Aachen University in general for their support. It should be noted that GaussForHomalg is dependant on the GAP homalg package by M. Barakat [Bar09], as well as the Gauss package by myself [Gör08]. This should be clear as GaussForHomalg presents a link between these two packages. This manual was created with the help of the GAPDoc package by M. Neunhöffer and F. Lübeck [LN08].

Contents

Chapter 1

Introduction

1.1 Overview over this manual

Chapter 1 is concerned with the technical details of installing and running this package. Chapter 2 explains how GaussForHomalg works and what you need to know to extend homalg with your own rings. Also included in this manual is a documented list of the most important methods and functions for this linking process. Anyone interested in source code should just check out the files in the `gap/pkg/GaussForHomalg/gap/` folder (→ Appendix A).

1.2 Installation of the GaussForHomalg Package

To install this package just extract the package's archive file to the GAP `pkg/` directory. By default the GaussForHomalg package is not automatically loaded by GAP when it is installed. You must load the package with `LoadPackage("GaussForHomalg");` before its functions become available. Please, send me an e-mail if you have any questions, remarks, suggestions, etc. concerning GaussForHomalg. Also, I would like to hear about applications of this package.
Simon Goertzen

Chapter 2

Usage

If you are just interested in using the Gauss package with homalg, you do not need to know much about GaussForHomalg, as it will work in the background, telling homalg which functions to call.

However, you might be interested in writing your own XyzForHomalg, enabling homalg to assist you with your computations. For this purpose, I will provide an overview of the GaussForHomalg code. Please note that Gauss is a GAP package, therefore this is not a reference guide for the package RingsForHomalg, which utilizes the IO-stream functionality of IO_ForHomalg to send commands to external computer algebra systems. If you wish to tie an external system to homalg, RingsForHomalg is the better reference package.

The file for all low-level operations is `GaussTools.gi`. Like all "Tools" files it just includes one global variable `CommonHomalgTableForGaussTools`, which is a record of functions with homalg matrices as arguments. The return values of the GaussForHomalg tools are documented in 3 and have to be the same for each tools table.

In this particular case, the file also includes the following code:

```
if IsBound( HOMALG.OtherInternalMatrixTypes ) then
    Add( HOMALG.OtherInternalMatrixTypes, IsSparseMatrix );
else
    HOMALG.OtherInternalMatrixTypes := [ IsSparseMatrix ];
fi;
```

This is a specialty to explain to homalg that Gauss introduces a new matrix type in GAP. Usually, there should not be a need for this.

The next "general" file is `GaussBasic.gi`. This includes the basic functions based on [BR08], again stored in the global record `CommonHomalgTableForGaussBasic`. Preceding this record are some small methods to make sure GaussForHomalg works with sparse as well as with dense matrices - just like above, these should not be neccessary in general.

In `GaussForHomalg.gi` the methods for matrix entry manipulation are installed.

Finally, we come to the most important files, making sense of the basic functions and tools defined above. Depending on the functionality (especially concerning function names) of the system you will need different files for different rings. In this case, functionality for $\mathbb{Z}/n\mathbb{Z}$ is stored in `GaussFQI.gi` (Finite Quotients of the Integers), while the Rationals have their own file, `GaussRationals.gi`. Note that both files include only one method, `CreateHomalgTable`, using method selection to create the correct table. Depending on the properties of the ring, the basic functions are loaded and some more "specific" functions can be defined, in this case for example the function `RowReducedEchelonForm`

(3.2.5), both in a one- and a two-argument version. The tools should be universal enough to be loaded for every possible ring. If it is neccessary to overwrite a tool, this is the place to do it. An example for this could be `Involution` (3.1.5), which is generally just a matrix transposition, but could be overwritten to be a true involution when creating the homalg table for noncommutative rings.

Chapter 3

GaussForHomalg methods and functions

3.1 The Tools

Please note that there are more tool functions you can define, GaussForHomalg just provides homalg with a sufficient subset. This varies with the type and complexity of the rings you want to define. On the other hand, `ImportMatrix` (3.1.4) is a function specifically designed for GaussForHomalg.

3.1.1 ZeroMatrix

◊ `ZeroMatrix(C)` (function)

Returns: a sparse matrix

This returns a sparse matrix with the same dimensions and base ring as the homalg matrix `C`.

3.1.2 IdentityMatrix

◊ `IdentityMatrix(C)` (function)

Returns: a sparse matrix

This returns a sparse $n \times n$ identity matrix with the same ring as the homalg matrix `C`, n being the number of rows of `C`.

3.1.3 CopyMatrix

◊ `CopyMatrix(C)` (function)

Returns: a sparse matrix

This returns a sparse matrix which is a shallow copy of the sparse matrix stored in the `Eval` attribute of the homalg matrix `C`.

3.1.4 ImportMatrix

◊ `ImportMatrix(M, R)` (function)

Returns: a sparse matrix

This returns the sparse version of the GAP matrix `M` over the ring `R`. It prevents homalg from calling sparse matrix algorithms on dense GAP matrices. Note that this is not a ”standard” tool but neccessary because of the new data type.

3.1.5 Involution

◊ Involution(*M*) (function)

Returns: a sparse matrix

This returns a sparse matrix which is the transpose of the sparse matrix stored in the Eval attribute of the homalg matrix *M*.

3.1.6 CertainRows

◊ CertainRows(*M*, *plist*) (function)

Returns: a sparse matrix

This returns the rows in *plist* of the sparse matrix stored in the Eval attribute of the homalg matrix *M* as a new matrix.

3.1.7 CertainColumns

◊ CertainColumns(*M*, *plist*) (function)

Returns: a sparse matrix

This returns the columns in *plist* of the sparse matrix stored in the Eval attribute of the homalg matrix *M* as a new matrix.

3.1.8 UnionOfRows

◊ UnionOfRows(*A*, *B*) (function)

Returns: a sparse matrix

This returns the sparse matrix created by concatenating the rows of the sparse matrices stored in the Eval attributes of the homalg matrices *A* and *B*.

3.1.9 UnionOfColumns

◊ UnionOfColumns(*A*, *B*) (function)

Returns: a sparse matrix

This returns the sparse matrix created by concatenating the columns of the sparse matrices stored in the Eval attributes of the homalg matrices *A* and *B*.

3.1.10 DiagMat

◊ DiagMat(*e*) (function)

Returns: a sparse matrix

This method takes a list *e* of homalg matrices and returns the sparse block matrix of the matrices stored in the Eval attributes of the matrices in *e*.

3.1.11 KroneckerMat

◊ KroneckerMat(*A*, *B*) (function)

Returns: a sparse matrix

This returns the sparse Kronecker matrix of the matrices stored in the Eval attributes of the homalg matrices *A* and *B*.

3.1.12 Compose

◊ Compose(A, B) (function)

Returns: a sparse matrix

This returns the matrix product of the sparse matrices stored in the Eval attributes of the homalg matrices A and B.

3.1.13 NrRows

◊ NrRows(C) (function)

Returns: an integer

This returns the number of rows of the sparse matrix stored in the Eval attribute of the homalg matrix C.

3.1.14 NrColumns

◊ NrColumns(C) (function)

Returns: an integer

This returns the number of columns of the sparse matrix stored in the Eval attribute of the homalg matrix C.

3.1.15 IsZeroMatrix

◊ IsZeroMatrix(C) (function)

Returns: TRUE or FALSE

This returns TRUE if the sparse matrix stored in the Eval attribute of the homalg matrix C is a zero matrix, and FALSE otherwise.

3.1.16 IsDiagonalMatrix

◊ IsDiagonalMatrix(C) (function)

Returns: TRUE or FALSE

This returns TRUE if the sparse matrix stored in the Eval attribute of the homalg matrix C is a diagonal matrix, and FALSE otherwise.

3.1.17 ZeroRows

◊ ZeroRows(C) (function)

Returns: a list

This returns the list of zero rows of the sparse matrix stored in the Eval attribute of the homalg matrix C.

3.1.18 ZeroColumns

◊ ZeroColumns(C) (function)

Returns: a list

This returns the list of zero columns of the sparse matrix stored in the Eval attribute of the homalg matrix C.

3.2 The Basic Functions and homalg table creation

3.2.1 DecideZeroRows

◊ DecideZeroRows(A, B) (function)

Returns: a homalg matrix

This returns the homalg matrix you get by row reducing the homalg matrix A with the homalg matrix B.

3.2.2 DecideZeroRowsEffectively

◊ DecideZeroRowsEffectively(A, B, T) (function)

Returns: a homalg matrix M

This returns the homalg matrix M you get by row reducing the homalg matrix A with the homalg matrix B. The transformation matrix is stored in the void homalg matrix T as a side effect. The matrices satisfy $M = A + T * B$.

3.2.3 SyzygiesGeneratorsOfRows

◊ SyzygiesGeneratorsOfRows(M) (function)

Returns: a homalg matrix

This returns the row syzygies of the homalg matrix M, again as a homalg matrix.

3.2.4 RelativeSyzygiesGeneratorsOfRows

◊ RelativeSyzygiesGeneratorsOfRows(M, N) (function)

Returns: a homalg matrix

The row syzygies of M are returned, but now the computation interpretes the rows of the homalg matrix N as additional zero relations.

3.2.5 RowReducedEchelonForm

◊ RowReducedEchelonForm(M[, U]) (function)

Returns: a homalg matrix N

If one argument is given, this returns the triangular basis (reduced row echelon form) of the homalg matrix M, again as a homalg matrix. In case of two arguments, still only the triangular basis of M is returned, but the transformation matrix is stored in the void homalg matrix U as a side effect. The matrices satisfy $N = U * M$.

3.2.6 CreateHomalgTable

◊ CreateHomalgTable(R) (function)

Returns: a homalg table

This returns the homalg table of what will become the homalg ring R (at this point R is just a homalg object with some properties for the method selection of CreateHomalgTable). This method includes the needed functions stored in the global variables CommonHomalgTableForGaussTools and CommonHomalgTableForGaussBasic, and can add some more to the record that will become the homalg table.

3.3 Matrix entry manipulation

This is just support for the sparse matrix data type.

3.3.1 GetEntryOfHomalgMatrix

◊ GetEntryOfHomalgMatrix(*M*, *r*, *c*, *R*) (method)

Returns: *M*[*r*,*c*]

If the Eval attribute of the homalg matrix *M* over the homalg ring *R* is sparse, this calls the corresponding Gauss command GetEntry.

3.3.2 SetEntryOfHomalgMatrix

◊ SetEntryOfHomalgMatrix(*M*, *r*, *c*, *e*, *R*) (method)

Returns: nothing

If the Eval attribute of the homalg matrix *M* over the homalg ring *R* is sparse, this calls the corresponding Gauss command GetEntry, to achieve *M*[*r*,*c*]:=*e*.

3.3.3 AddToEntryOfHomalgMatrix

◊ AddToEntryOfHomalgMatrix(*M*, *r*, *c*, *e*, *R*) (method)

Returns: nothing

If the Eval attribute of the homalg matrix *M* over the homalg ring *R* is sparse, this calls the corresponding Gauss command AddToEntry, to achieve *M*[*r*,*c*] := *M*[*r*,*c*] + *e*.

Chapter 4

Example

4.1 HomHom

The following example is taken from Section 2 of [BR06].

The computation takes place over the ring $R = \mathbb{Z}/2^8\mathbb{Z}$, which is directly supported by the package Gauss.

Here we compute the (infinite) long exact homology sequence of the covariant functor $Hom(Hom(-,\mathbb{Z}/2^7\mathbb{Z}),\mathbb{Z}/2^4\mathbb{Z})$ (and its left derived functors) applied to the short exact sequence

$$0 \longrightarrow M_- = \mathbb{Z}/2^2\mathbb{Z} \xrightarrow{\alpha_1} M = \mathbb{Z}/2^5\mathbb{Z} \xrightarrow{\alpha_2} \_M = \mathbb{Z}/2^3\mathbb{Z} \longrightarrow 0\ .$$

Example

```
gap> R := HomalgRingOfIntegers( 2^8 );
<A homalg internal ring>
gap> Display( R );
Z/256Z
gap> M := LeftPresentation( [ 2^5 ], R );
<A cyclic left module presented by an unknown number of relations for a cyclic\
 generator>
gap> Display( M );
Z/256Z/< ZmodnZObj(32,256) >
gap> M;
<A cyclic left module presented by 1 relation for a cyclic generator>
gap> _M := LeftPresentation( [ 2^3 ], R );
<A cyclic left module presented by an unknown number of relations for a cyclic\
 generator>
gap> Display( _M );
Z/256Z/< ZmodnZObj(8,256) >
gap> _M;
<A cyclic left module presented by 1 relation for a cyclic generator>
gap> alpha2 := HomalgMap( [ 1 ], M, _M );
<A "homomorphism" of left modules>
gap> IsMorphism( alpha2 );
true
gap> alpha2;
<A homomorphism of left modules>
gap> Display( alpha2 );
   1
```

```
the map is currently represented by the above 1 x 1 matrix
gap> M_ := Kernel( alpha2 );
<A cyclic left module presented by yet unknown relations for a cyclic generato\
r>
gap> alpha1 := KernelEmb( alpha2 );
<A monomorphism of left modules>
gap> seq := HomalgComplex( alpha2 );
<An acyclic complex containing a single morphism of left modules at degrees
[ 0 .. 1 ]>
gap> Add( seq, alpha1 );
gap> seq;
<A sequence containing 2 morphisms of left modules at degrees [ 0 .. 2 ]>
gap> IsShortExactSequence( seq );
true
gap> seq;
<A short exact sequence containing 2 morphisms of left modules at degrees
[ 0 .. 2 ]>
gap> Display( seq );
-------------------------
at homology degree: 2
Z/256Z/< ZmodnZObj(4,256) >
-------------------------
  24

the map is currently represented by the above 1 x 1 matrix
------------v------------
at homology degree: 1
Z/256Z/< ZmodnZObj(32,256) >
-------------------------
   1

the map is currently represented by the above 1 x 1 matrix
------------v------------
at homology degree: 0
Z/256Z/< ZmodnZObj(8,256) >
-------------------------
gap> K := LeftPresentation( [ 2^7 ], R );
<A cyclic left module presented by an unknown number of relations for a cyclic\
 generator>
gap> L := RightPresentation( [ 2^4 ], R );
<A cyclic right module on a cyclic generator satisfying an unknown number of r\
elations>
gap> triangle := LHomHom( 4, seq, K, L, "t" );
<An exact triangle containing 3 morphisms of left complexes at degrees
[ 1, 2, 3, 1 ]>
gap> lehs := LongSequence( triangle );
<A sequence containing 14 morphisms of left modules at degrees [ 0 .. 14 ]>
gap> ByASmallerPresentation( lehs );
<A non-zero sequence containing 14 morphisms of left modules at degrees
[ 0 .. 14 ]>
gap> IsExactSequence( lehs );
false
```

```
gap> lehs;
<A non-zero left acyclic complex containing
14 morphisms of left modules at degrees [ 0 .. 14 ]>
gap> Assert( 0, IsLeftAcyclic( lehs ) );
gap> Display( lehs );
--------------------------
at homology degree: 14
Z/256Z/< ZmodnZObj(4,256) >
--------------------------
   4

the map is currently represented by the above 1 x 1 matrix
------------v------------
at homology degree: 13
Z/256Z/< ZmodnZObj(8,256) >
--------------------------
   6

the map is currently represented by the above 1 x 1 matrix
------------v------------
at homology degree: 12
Z/256Z/< ZmodnZObj(8,256) >
--------------------------
   2

the map is currently represented by the above 1 x 1 matrix
------------v------------
at homology degree: 11
Z/256Z/< ZmodnZObj(4,256) >
--------------------------
   4

the map is currently represented by the above 1 x 1 matrix
------------v------------
at homology degree: 10
Z/256Z/< ZmodnZObj(8,256) >
--------------------------
   6

the map is currently represented by the above 1 x 1 matrix
------------v------------
at homology degree: 9
Z/256Z/< ZmodnZObj(8,256) >
--------------------------
   2

the map is currently represented by the above 1 x 1 matrix
------------v------------
at homology degree: 8
Z/256Z/< ZmodnZObj(4,256) >
--------------------------
   4
```

```
the map is currently represented by the above 1 x 1 matrix
------------v------------
at homology degree: 7
Z/256Z/< ZmodnZObj(8,256) >
--------------------------
   6

the map is currently represented by the above 1 x 1 matrix
------------v------------
at homology degree: 6
Z/256Z/< ZmodnZObj(8,256) >
--------------------------
   2

the map is currently represented by the above 1 x 1 matrix
------------v------------
at homology degree: 5
Z/256Z/< ZmodnZObj(4,256) >
--------------------------
   4

the map is currently represented by the above 1 x 1 matrix
------------v------------
at homology degree: 4
Z/256Z/< ZmodnZObj(8,256) >
--------------------------
   6

the map is currently represented by the above 1 x 1 matrix
------------v------------
at homology degree: 3
Z/256Z/< ZmodnZObj(8,256) >
--------------------------
   2

the map is currently represented by the above 1 x 1 matrix
------------v------------
at homology degree: 2
Z/256Z/< ZmodnZObj(4,256) >
--------------------------
   8

the map is currently represented by the above 1 x 1 matrix
------------v------------
at homology degree: 1
Z/256Z/< ZmodnZObj(16,256) >
--------------------------
   1

the map is currently represented by the above 1 x 1 matrix
------------v------------
at homology degree: 0
Z/256Z/< ZmodnZObj(8,256) >
```

```
-------------------------
```

Appendix A

An Overview of the GaussForHomalg package source code

Filename	Content
`GaussForHomalg.gi`	Methods for matrix entry manipulation
`GaussTools.gi`	Tools for matrix operations
`GaussBasic.gi`	The "Basic" Operations ($\rightarrow$ [BR08] and [Bar09])
`GaussFQI.gi`	homalg Table for finite quotients of $\mathbb{Z}$: $\mathbb{Z}/\langle p^n \rangle$
`GaussRationals.gi`	homalg Table for the rationals $\mathbb{Q}$

Table: *The GaussForHomalg package files.*

References

[Bar09] M. Barakat. *The homalg package – GAP4*, 2007-2009. http://homalg.math.rwth-aachen.de/index.php/core-packages/homalg-package. 2, 18

[BR06] M. Barakat and D. Robertz. *homalg*: First steps to an abstract package for homological algebra. In *Proceedings of the X meeting on computational algebra and its applications (EACA 2006), Sevilla (Spain)*, pages 29–32, 2006. http://homalg.math.rwth-aachen.de/maple/homalg_eaca06.pdf. 13

[BR08] M. Barakat and D. Robertz. homalg – A Meta-Package for Homological Algebra. *J. Algebra Appl.*, 7(3):299–317, 2008. arXiv:math.AC/0701146. 6, 18

[Gör08] S. Görtzen. *GAP Package Gauss*, 2007-2008. http://wwwb.math.rwth-aachen.de/goertzen/Gauss. 2

[LN08] F. Lübeck and M. Neunhöffer. *GAP Package GAPDoc*, 2007-2008. http://www.math.rwth-aachen.de/~Frank.Luebeck/GAPDoc. 2

Index