Summary of New Features in Magma V2.12

July 2005

1 Introduction

This document provides a terse summary of the new features installed in Magma for release version V2.12 (July 2005).

Previous releases of Magma were: V2.11 (May 2004), V2.10 (April 2003), V2.9 (May 2002), V2.8 (July 2001), V2.7 (June 2000), V2.6 (November 1999), V2.5 (July 1999), V2.4 (December 1998), V2.3 (January 1998), V2.2 (April 1997), V2.1 (October 1996), V2.01 (June 1996) and V1.3 (March 1996).

2 Summary

Groups

- *Finitely-Presented Groups:* A new algorithm due to David Firth and Derek Holt computes all normal subgroups of a finitely presented group up to a specified index n for $n \leq 100000$.

- *Finite Groups:* The `Subgroups` family of functions has been improved by implementing new algorithms for lifting subgroups through the soluble radical. This makes working with groups with larger abelian chief factors possible. The `Subgroups` functions have also been extended to apply to matrix groups (with BSGS) over a field or integral domain. Finally, the lattice of all subgroups of a group is now computed by repeated use of the new maximal subgroup machinery and is much faster than the previous algorithm. (This was introduced following experiments by Dimitri Leemans).

- *Finite Groups:* The database of maximal subgroups and automorphism groups for almost simple groups has been extended. This database is being constructed firstly, by determining all maximals generically for each classical family in each dimension and, secondly, by storing a description of the maximals for a particular group in the case of exceptional groups and sporadic simple groups. The following groups are included for the first time: $L_5(q)$ for all prime powers q, $L_6(3)$, $L_7(3)$; $U_3(q)$ for prime powers $q \leq 25$; $U_4(q)$ for prime powers $q \leq 7$; $Sp_4(q)$ for all odd prime powers q and even $q \leq 16$, $Sp_{10}(2)$. In particular, the database includes all simple groups having a permutation representation of degree less than 1000. The expansion of the maximal subgroups database automatically expands the range of applicability of other key group structure functions such as subgroups, low index subgroups, automorphism groups, character tables and representations.

- *Permutation Groups:* A greatly improved algorithm has been introduced for computing the conjugacy classes of a large non-soluble permutation group G. The new algorithm reduces the problem to that of determining the classes of an almost simple group. Once the classes of the non-abelian simple composition factors of G are known, the algorithm proceeds to lift these first to the radical quotient and then to the whole group in accord with the TF-group philosophy employed by Cannon and Holt in other contexts. In the case of an almost simple group, special algorithms are used for almost simple groups having A_n or $PSL(2, q)$ as socle while the classes of other groups are found using an inductive algorithm developed by Greg Butler. The new algorithm can, if required, also compute the power maps and centralisers of the class representatives simultaneously. This takes advantage of the partial information computed during the classes search and so is much faster than computing them after the classes are already known.

- *Matrix Groups:* Conjugacy classes of elements are now found using a lifting algorithm in the case of a matrix group (with BSGS) over a field or integral domain. The new conjugacy class algorithm for permutation groups (outlined above) is used to compute classes of the TF-quotient, and then the classes are lifted through the soluble radical. Lifting methods are now used to find centralizers and test element conjugacy in these matrix groups. These methods can be much faster than the pure backtrack search used previously.

- *Matrix Groups:* Improvements have been made to BSGS operations, including using a base from a known supergroup and encouraging shallow Schreier trees. This has improved the speed of such fundamental operations as membership testing, finding a BSGS using the Schreier–Todd-Coxeter method, and homomorphism evaluation.

- *Matrix Groups:* Eamonn O'Brien has supplied constructions of families of low degree irreducible matrix groups (Flanneryt & O'Brien).

- *p-groups:* Magma contains O'Brien & Vaughan-Lee's constructions of groups of order p^7. These may be accessed using the new intrinsics `SearchPGroups` and `CountPGroups`.

- *Automorphism Groups:* More support is provided for working with the automorphism group (of a group) represented as a set of mappings. In particular, it is now possible to test pairs of elements of the automorphism group for equality and to create subgroups.

Basic Rings

- *Ring of Integers:* Multiplication of large integers (having 2^{17} bits or more) has been greatly sped up by a new implementation of the FFT-based Schoenhage-Strassen algorithm, which is up to 2.3 times faster than GMP. Paul Zimmermann's efficient GMP-ECM is now included for integer factorization.

- *Polynomial Rings:* Polynomial multiplication and factorization has been greatly sped up over various types of rings.

- *Real and Complex Numbers*: Support for real and complex numbers has been greatly enhanced in this release, by replacing the existing package with a new implementation based upon MPFR and MPC (which are in turn based upon GMP). These libraries have been designed to extend the semantics of the ANSI/IEEE-754 standard for double-precision floating-point arithmetic to real numbers of arbitrary precision. As a result, the rounding behaviour of algorithms based on MPFR is precise and thus significantly easier to analyze. At the same time, the developers of MPFR have expended great effort on making the library efficient; users of Magma can on the whole expect significant performance improvements in code that rely on the reals.

Linear Algebra and Module Theory

- *Matrices:* Portions of the ATLAS (Automatically Tuned Linear Algebra Software) library are now used on selected platforms, providing speedups of fundamental matrix operations over finite fields.

Commutative Algebra

- *Modules and Homology:* The Chain complex type has been generalized to modules over multivariate polynomial rings and an implementation of the La Scala resolution algorithm is included.

Extensions of Rings

- *Cyclotomic Fields:* The sparse representation of cyclotomic fields underwent big changes in this release, to match the behaviour of the dense cyclotomic fields where possible. In particular this means that the intrinsic `RootOfUnity` now returns the "same" root in both representations and the mapping between the sparse and the dense representation has changed.

- *Function Fields:* The algorithm of Florian Heß to compute isomorphisms between function fields or plane curves has been implemented in this release.

Representation Theory

- The algorithm used for computing the character table of a finite group has been changed to a new algorithm devised by Bill Unger. The previous algorithm (Dixon-Schneider) is still used for PC-groups and small groups. The new algorithm is used for larger permutation and matrix groups. Unger's algorithm induces characters from elementary subgroups and splits them using LLL, and has shown itself capable of coping with much larger groups than Magma's Dixon-Schneider implementation.

- The `CharacterDegrees` intrinsic has been extended to apply to all PC-groups (Conlon's algorithm), not just p-groups. This has meant a change in the output format. The new format is a sequence of pairs $[\langle d_1, c_1\rangle, \ldots]$, where each c_i is the number of characters having degree d_i.

Lie Theory

- *Coxeter groups:* The Geck-Pfeiffer algorithm for computing conjugacy classes in finite Coxeter groups has been implemented. This algorithm works for finite Coxeter groups given by permutations or by a presentation.

- *Groups of Lie type:* The algorithms for multiplying elements in groups of Lie type have been speeded up considerably, as have the algorithms for computing representations and their inverses. The Galois cohomology of a groups of Lie type can now be computed using an algorithm due to Haller. Generators for twisted tori of finite groups of Lie type can now be constructed.

- *Lie algebras:* The Cartan-type and twisted classical-type Lie algebras can now be constructed over a finite field.

Algebraic Geometry

- *Curves and Function Fields:* The `Crv` type has now been extended to all schemes of dimension 1 and the function fields of integral curves have been reimplemented. All previous plane curve functionality has been extended to the new type.

- *Scheme Saturation:* Previously in Magma, errors would occasionally result because the defining ideal for a projective scheme was not the largest such ideal (ie, not *saturated*). A saturation mechanism for projective schemes has been added to solve this problem. Saturation only occurs when required for certain computations.

Arithmetic Geometry

- *Jacobians of Hyperelliptic Curves:* The explicit group law has been extended to hyperelliptic curves with two rational points at infinity. Hence, arithmetic is now implemented for all Jacobians except those where the curve has no rational points at infinity and odd genus.

Coding Theory

- *Quantum Stabilizer Codes:* A new module is included in Magma for constructing and computing with quantum stabilizer codes, which are an essential component of quantum computing. Quantum stabilizer codes are represented by symplectic self-orthogonal additive codes, and so build on top of the additive codes package released in version V2.11. Important constructions such as CSS codes are included, as well as cyclic and quasi-cyclic quantum codes. An adaption of the Zimmermann minimum weight algorithm is used to calculate not only the minimum weight of a quantum code, but also to simultaneously test its purity. A database is included of the best known quantum codes of length up to 35. A basic package is also available for computing with quantum states in a Hilbert space, as well as mappings to the error group.

3 Removals and Changes

This section lists the most important changes in Version 2.12. Other minor changes are listed in the relevant sections.

– The `.dat` data files have been removed from the packages directory (the `.sig` files remain). Consequently, the packages directory is now architecture independent so may be shared across different architectures, although it is recommended that it be installed separately on local disks to avoid slow startups of Magma.

– The function `Roots` applied to a univariate polynomial over a real field now returns only the real roots over the given real field (previously, the result was over a complex field).

4 Documentation

New chapters in the Handbook for V2.12 (with their chapter numbers) are:

– Quantum Groups (89).

– Universal Enveloping Algebras (90).

– L-Functions (107).

– Quantum Codes (121).

5 Language and System Features [HB 1–6]

New Features:

– The `.dat` data files have been removed from the packages directory (the `.sig` files remain). Consequently, the packages directory is now architecture independent so may be shared across different architectures, although it is recommended that it be installed separately on local disks to avoid slow startups of Magma.

6 Groups

6.1 Permutation Groups [HB 18]

Changes:

- Evaluating the class map of a permutation group has been made more efficient.
- The conjugacy classes of a permutation group may now be asserted to be an ordinary sequence of 3-tuples, not necessarily one constructed by a call to `ConjugacyClasses`.

New Features:

- A greatly improved algorithm has been introduced for computing the conjugacy classes of a large non-soluble permutation group G. See Summary pages of this document for more details.

Bug fixes:

- A bug which caused a crash when constructing a projective unitary group over a field of size greater than 2^{20} has been fixed.
- A bug that occurred when finding the complements of a normal subgroup of a permutation group has been fixed.
- Maximal subgroups of $PSL(11, 2)$ corrected to include two classes of M_{24}, conjugate under an outer automorphism. They do not extend to maximal subgroups of the automorphism group. Also, a bug in making maximal subgroups of $PSp(4, p) : 2$, where $p = 1$ mod 12, has been fixed. Bugs reported by Colva Roney-Dougal.
- A bug in computing class matrices of permutation groups has been fixed. This bug may have caused character table computations to go wrong.
- Crashes when using the `!!` operator have been fixed.
- Two classes of maximal subgroups of Alt(574) have been added. They are primitive, isomorphic to PSL(2,41), and do not extend to maximal subgroups of Sym(574). Bug reported by Colva Roney-Dougal.

6.2 General Matrix Groups [HB 19]

Changes:

- For matrix groups with BSGS over a field or integral domain, the algorithms used for conjugacy classes, element centralizer and element conjugacy testing have been changed to lifting algorithms, which compute the answer in a permutation representation of the group over its soluble radical, and then lift through the radical.
- The order of a matrix group with a single generator is now determined by the order of the generator rather than by STCS. Suggested by Derek Holt
- Matrix group STCS has been modified to add extra strong generators to force shallow schreier trees (much as the random schreier does). This will hasten membership testing and homomorphism evaluation, and may improve STCS performance along the way.

New Features:

- Conjugacy classes of elements are now found using a lifting algorithm in the case of a matrix group (with BSGS) over a field or integral domain. The new conjugacy class algorithm for permutation groups (outlined above) is used to compute classes of the TF-quotient, and then the classes are lifted through the soluble radical. Lifting methods are now used to find centralizers and test element conjugacy in these matrix groups. These methods can be much faster than the pure backtrack search used previously.
- Eamonn O'Brien has supplied constructions of families of low degree irreducible matrix groups (Flannery & O'Brien). The relevant new intrinsics are `GL2IrreducibleSubgroups`, `GL2IrreducibleSolubleSubgroups` and `GL3IrreducibleSolubleSubgroups`.
- Michael Downward and Eamonn O'Brien's functions for sporadic simple groups are included in V2.12. The relevant intrinsics are `StandardGenerators`, `StandardPresentation`, `MaximalSubgroups`, `Subgroups`, `GoodBasePoints`, `SubgroupsData`, `BSGS` and `RandomSchreier`. The `GoodBasePoints` intrinsic uses the algorithm of O'Brien & Wilson. One of the arguments to each of these intrinsics is a string giving the name of the sporadic simple group you are working on.

Bug fixes:

- A bug with `FPGroupStrong` for matrix groups, where the presentation found did not necessarily contain a presentation for each basic stabilizer (and so wasn't a strong presentation), has been fixed.
- A bug with `FPGroupStrong`, where the presentation found did not necessarily contain a presentation for each basic stabilizer (and so wasn't a strong presentation), has been fixed. Bug reported by Derek Holt.
- A bug which sometimes caused `GModule` and `ChiefSeries` to go into an infinite loop has been fixed.
- A bug in the setting of universes of matrix group orbits has been rectified. Bug reported by Eric Rains.
- Bugs in construction of generators by the `ChevalleyGroup` intrinsic for families 2A2, 2G2 and 2F4 with large fields (size $> 2^{20}$) have been fixed. Bugs reported by Henrik Baarnhielm.
- A bug with the map returned by `PCGroup` of a trivial matrix group has been fixed. Bug reported by Mark Stather.

New Features:

- Constructive recognition of a matrix group isomorphic to $(P)SL(2, q)$ in defining characteristic. This implements the algorithm of Conder, Leedham-Green and O'Brien.

Bug Fixes:

- A memory management problem that sometimes caused a crash when doing arithmetic in GPC-groups has been fixed. Bug reported by Mark Stather.

6.3 Finite Soluble Groups [HB 20]

Changes:

- Data structures used for some orbit-stabilizer calculations have been modernized. This has given speed-ups to `Centralizer`, `ClassMap`, and subgroup intersection.
- The verbose flag `Classes` now gives information on the progress of the PC-group class table algorithms.

New Features:

- The intrinsic `CharacterDegrees` now applies to all finite soluble groups. The intrinsic uses Conlon's algorithm. The implementation was supplied by Derek Holt.

Bug Fixes:

- A bug in the construction of direct products of PC-groups that sometime caused crashes, particularly when forming products of a sequence of groups, has been fixed. Bug reported by Kasper Andersen. The homomorphisms associated with the direct product have also been corrected. Bug reported by Mark Stather.
- A bug where the `IsConjugate` function for subgroups of GrpPC sometimes returned an incorrect conjugating element (along with a true result) has been fixed. The problem only applied to conjugation of subgroups, not elements.
- A bug with testing conjugacy of elements of PC-groups, when the elements are not in the group where the conjugating element is sought, has been fixed.
- The sorting of class tables of PC-groups has been fixed so that the conjugacy classes are reliably sorted by element order and then class size. This resolves a bug in `PowerMap` that occurred when this failed for groups with large classes.

6.4 Finite p-groups

Changes:

- The intrinsic `CharacterDegrees` has changed its return value. As the intrinsic now applies to general PC-groups, the returned sequence has the form of a sequence of pairs $[\ldots\langle d_i, c_i\rangle,\ldots]$, where each c_i is the number of characters having degree d_i. The previous algorithm, with unchanged return values, is available under the name `CharacterDegreesPGroup`.

New Features:

- The new intrinsics `SearchPGroups` and `CountPGroups` allow searches through p-groups in the small groups database as well as groups of order p^7.

6.5 Databases of Groups [HB 26]

New Features:

- The database of primitive permutation groups has been extended to degree 2499. The data was supplied by Colva Roney-Dougal. The data for the affine groups may be accessed to get all irreducible subgroups of $GL(n, p)$ where p is prime and $p^n \leq 2499$ using the new `IrreducibleMatrixGroup` intrinsic.

Bug Fixes:

- A memory management problem that allowed the orders of ATLAS groups to be corrupted has been fixed. Bug reported by Markus Grassl.
- Primitive groups database - Added a group of degree 574 which is not in the Dixon & Mortimer list. The group is isomorphic to $PSL(2, 41)$, with point stabilizer A_5. The automorphism group does not have a corresponding primitive representation. Bug reported by Colva Roney-Dougal.

6.6 Finitely Presented Groups [HB 28]

New Features:

- A new `LowIndexNormalSubgroups` intrinsic has been supplied by David Firth & Derek Holt. This will find the lattice of normal subgroups of a finitely-presented group up to given index $n \leq 10^5$.

Bug Fixes:

- Crashes when applying simplification to free finitely presented groups have been fixed.
- A bug with with finding the coset image of a finitely presented group on itself has been fixed. Bug reported by Derek Holt.
- A bug in the Homomorphisms intrinsic for has been fixed. The bug caused a crash when an FP-group with 32 generators was input. Bug reported by Eamonn O'Brien.
- Memory handling in the `pQuotient` function has been improved for the Mac version. Problem reported by Mike Newman.

6.7 Generic Abelian Groups

Bug Fixes:

- A bug when constructing generic abelian groups over an aggregate of integers has been fixed.

6.8 Finitely Presented Abelian Groups

New Features:

- An intrinsic `CosetIntersection` which finds the intersection of the cosets associated with two subgroups of a finitely presented abelian group has been provided.

7 Basic Rings

7.1 Integer Ring [HB 37]

New Features:

- Multiplication of large integers (having 2^{17} bits or more) has been greatly sped up by a new implementation of the FFT-based Schoenhage-Strassen algorithm. The new implementation is up to 2.3 times faster than GMP; see `http://tinyurl.com/9657h` for comparative tables.
- The LIP-ECM code for factoring integers has been replaced by Paul Zimmermann's much more efficient GMP-ECM.
- The GMP-ECM package also provides fast $p-1$ and $p+1$ integer factorisation algorithms. This is the first time the $p+1$ method has been available in Magma.
- New functions `ECMOrder` and `ECMFactoredOrder` which, given a prime factor p returned by ECM and the corresponding sigma seed, return the order or factored order of the successful elliptic curve mod p.
- Some extra functionality for dealing with prime numbers has been added. In particular, a `PrimesUpTo` function returns the primes up to a given limit, and `PrimesInInterval` returns the primes in an interval. Finally, the `NthPrime` function returns the nth prime.

7.2 Real and Complex Fields [HB 42]

Removals and Changes:

- The function `Roots` applied to a univariate polynomial over a real field now returns only the real roots over the given real field (previously, the result was over a complex field).

New Features:

- The previous real and complex arithmetic (based on MP and Pari) has been replaced by code based on MPFR and MPC. Note that a few functions (such as root finding) still call the old Pari code internally.
- The Uspenksy algorithm is now used to compute the real roots of a polynomial defined over a real field.
- Intrinsic `WeberF1` has been added.

7.3 Univariate Polynomial Rings [HB 40]

New Features:

- Polynomial multiplication over most types of finite fields and the integers has been completely reorganised, resulting in great speedups in general. In particular:
  - A new direct polynomial FFT Schoenhage-Strassen algorithm has been implemented, which is applicable to polynomials with large coefficients. An extension of this, handling input degrees not close to powers of 2, has also been developed.

  - Polynomials with small coefficients are now multiplied by the Kronecker-Schoenhage method (also known as segmentation), which maps the problem to large integer multiplication, thus benefiting from the new very fast integer multiplication mentioned above.
  - Multiplication of polynomials over power series and finite field extensions has also been sped up, via a similar expansion technique.
  - As an example, on an Opteron 150 (2.4GHz) Magma V2.12 can multiply two polynomials, each having degree 1000 and 1000-bit integer coefficients, in 0.0125 seconds.

- Polynomial factorization has been sped up, particularly in the van Hoeij factor combination algorithm.

Bug Fixes:

- `Discriminant` has been fixed so that the result agrees with the product formula (Cohen, ACCANT, 3.3.3) in characteristic p for a polynomial of degree a multiple p (so the diagram commutes if one computes first in characteristic zero and then reduces mod p).

8 Linear Algebra and Module Theory

8.1 Matrices [HB 43]

New Features:

- Major improvements have been made to the fundamental matrix algorithms over small finite fields. In particular, matrix multiplication over GF(2) has been rewritten from scratch, and is up to 10 times faster than previously.
- Portions of the ATLAS (Automatically Tuned Linear Algebra Software) library are now used on selected platforms. This provides a speedup of up to a factor 3 in fundamental linear algebra algorithms over machine-int-size prime finite fields, and characteristic zero algorithms based on these.

9 Commutative Algebra [HB 91–93

New Features:

- The pair selection algorithm for the Buchberger and F_4 algorithms has been sped up.
- New monomial order `"grevlexw"` (graded-reverse-lexicographical with weights) to allow weighted orders which work better with ideals which are homogeneous w.r.t. the weights.
- The existing type for module complexes has been greatly extended for Magma V2.12 to support modules over multivariate polynomial rings and affine algebras. A large amount of homological algebra over such rings will be supported, including operations on free resolutions and chain maps, and computation of homology.
- Fundamental to the new homology machinery is a recently developed algorithm for constructing free resolutions, which combines the ideas of the La Scala/Stillman resolution algorithm with the Faugere F4 Groebner basis algorithm. Consequently, Magma V2.12 will allow free resolutions to be computed for the first time, which have been impossible hitherto.

- The field of fractions of an arbitrary affine algebra over a field is now supported as a basic type in V2.12, and the scheme and curve machinery has been modified to exploit the new type in a uniform way. (Recall that the existing function field machinery only applies in the case of plane curves.) Affine algebras which are not domains are also supported, leading to "rings of total fractions".
- A new multivariate sparse interpolation algorithm is available in V2.12 for the first time. The algorithm greatly speeds up the computation of GCDs and resultants in many variables.

10 Extensions of Rings

10.1 Algebraic Number Fields [HB 53]

Changes:

- The number fields code has been extensively revised so as to use the MPFR and MPC packages for performing real and complex arithmetic. Formerly, this was done using the MP package of Richard Brent. Among many improvements, the MPFR arithmetic is often much faster.
- Bounds for the class group computation may now be specified by means of a user-defined function.
- Functions to drive the class group computation step-by-step have been added. In particular, a user is now able to "store" a class group computation and re-create it later. Furthermore, a function to check the completeness of a factorbasis in a given range has been added. Together, these two features allow for a efficient distributed verification of class groups.
- Infinite places for relative extensions of number fields are now supported.

10.2 Cyclotomic Fields [HB 49]

Removals and Changes:

- Sparse cyclotomic fields have been changed to more close resemble the dense representation, in particular, this means a different element is now returned from the `RootOfUnity` function and the maps between the dense and the sparse representation have changed.

10.3 Local Fields

New Features:

- Unit Groups: There are new functions for the computation of unit groups of local rings and local fields. They return an abelian group and a map whose inverse solves the discrete logarithm problem. The algorithms used are described in the context of the multiplicative group of residue class rings in the paper: Florian Hess, Sebastian Pauli, and Michael Pohst: *Computing the Multiplicative Group of Residue Class Rings*, Mathematics of Computation **72**, Number 243, 2003
- Enumeration of Extensions: Krasner's formulae for the number of extensions of a given degree and discriminant have been implemented. There is also rudimentary support for the enumeration of all extensions of given degree and discriminant (unramified, tamely ramified, degree p, degree p^m and normal).

- Class fields over local fields are computed using a new algorithm due to Sebastian Pauli. They are constructed as towers of extensions of degree p for the wildly ramified part over the tamely ramified part. The implementation includes many support functions for the computation of norm groups and solving norm equations.

Bug Fixes:

- Several bugs relating to the factorisation of polynomials over local fields and the computation of roots have been fixed.

10.4 Algebraic Function Fields [HB 53]

A type for field of fractions of orders of algebraic function fields has been introduced.

Removals and Changes:

- `FieldOfFractions` returns the new field of fractions type rather than the function field of the order. The function field can be accessed using `FunctionField`.
- The `Basis` of an order of a function field now returns a sequence of elements in the field of fractions of the order rather than the function field. The basis of an order as function field elements can still be gained using `Basis(O, FunctionField(O))`.
- The ideal arithmetic in function fields has been changed to reflect optimizations already included for the number fields.
- `RepresentationMatrix`, `MinimalPolynomial` etc may now be calculated for function field elements relative to specified rings.
- Support for expansion of degree > 1 places has been added.

New Features:

- Fields of Fractions for orders of function fields have been introduced. Simple functionality for these fields and their elements is available. The order which is the integers of one of these fields can be accessed using `Order`. Basis elements of orders can also be accessed using the `.` operator which returns an element in the field of fractions.

11 Lattices and Quadratic Forms

11.1 Lattices

New Features:

- A more general version of Simon's `IndefiniteLLLGram` has been implemented. This is used internally in various functions with conics and elliptic curves. However, it uses an integer-based method, and thus can be quite slow with large entries. It is hoped that the next release will include a robust floating-point method.

11.2 Binary Quadratic Forms

Removals and Changes:

- `Reduction` now returns the reduction of the quadratic form, and the matrix that effects the reduction.

12 Algebras

12.1 Group Algebras

Changes:

- The algorithm used by `JacobsonRadical` has been updated. See subsection 12.3 for details.

12.2 Finite Dimensional Algebras [HB 67]

Changes:

- The algorithm used by `JacobsonRadical` on associative algebras has been updated. See subsection 12.3 for details.

12.3 Matrix Algebras [HB 68]

Changes:

- The Meataxe algorithm is no longer used by default for `JacobsonRadical` where over fields in which p-th roots can be taken. The new algorithm is that of Cohen, Ivanyos and Wales. The Meataxe algorithm is used when the `Al` parameter is set to `"Meataxe"` or in situations where the Cohen, Ivanyos, Wales algorithm does not apply.
- New package of Jon Carlson and G. Matthews to compute presentations of matrix algebras.

13 Representation Theory

13.1 Modules over An Algebras [HB 73]

New Features:

- An implementation of the Dixon algorithm for finding the irreducible representation affording a given character of a finite group.
- Techniques for finding a field of minimal degree which realises a given representation.
- Permutation condensation and tensor condensation of modules over finite fields.
- Higher symmetric and exterior powers for G-modules.

14 Homological Algebra

14.1 Complexes

- Chain complex type generalized to modules over multivariate polynomial rings.
- Homomorphisms.
- Free resolutions, Hom modules.
- Homology.
- New algorithm for computing free resolutions, using a combination of of the La Scala/Stillman resolution algorithm and the Faugere F4 algorithm.

15 Lie Theory

15.1 Coxeter Groups [HB 83]

Changes:

- Arithmetic operations for finitely presented Coxeter groups have been implemented in the C kernel replacing the previous package code, resulting in large speed-ups for computations with Coxeter groups.
- A dramatically faster method for computing the left and right descent sets of an element of a finitely presented Coxeter group (suggested by Derek Holt) has been installed. The time taken for an example reported by Tim Honeywell dropped from 3.5 days to 80 seconds.

15.2 Coxeter Groups as Permutation Groups [HB 84]

BugFixes:

- The returned element from `TransversalElt` has been made to be of type GrpPermCoxElt, where before it could have been either that or GrpPermElt. Bug reported by Markus Grassl.

15.3 Groups of Lie Type [HB 86]

Removals and Changes:

- The application of the maps returned by `StandardRepresentation` and `RowReductionHomomorphism` has been made more efficient.

16 Algebraic Geometry

16.1 Schemes [HB 94]

Removals and Changes:

- A new type of function field for both ambients and curves has been introduced. This type is an interface on the old type and the old type function fields can be retrieved for advanced usage.
- `IsAmbientFunction` and `IsAmbientRationalFunction` have been removed. They are no longer necessary with the new implementation of function fields of schemes.
- All function fields know the scheme they are a function field of. Because of this it has been possible to reduce the number of arguments some intrinsics require. The `Restriction` intrinsic and the `ProjectiveMap` intrinsics now require one less argument than before.
- Saturation of a homogeneous ideal with respect to certain redundant ideals leads to the largest ideal defining the same subscheme of ambient projective spaces. The defining ideal stored for a projective scheme is now saturated when necessary for certain calculations. This has fixed the problem that led to occasional incorrect results in the intrinsics `Dimension`,`IsReduced`,`IsIrreducible`, `Prime/PrimaryComponents`, and `f(X)` (map images).
- The basic `Scheme`, `Curve` and `Cluster` constructions have a new Boolean parameter `Saturated`, which allows the user to specify that the given defining equations generate a saturated ideal. This will guarantee that Magma does no extra work in trying to saturate the ideal at a later stage.
- The order and types of arguments to `Parametrization` has been changed and signatures are now more specialized.

New Features:

- Intrinsic `ArithmeticGenus` has been extended from schemes in ordinary projective space to all projective schemes bar rational scrolls.
- Intrinsic `HasAffinePatch` added. This tells the user whether the nth affine patch of a scheme can be created (mainly relevant for weighted projective schemes).
- `f(p)` and `p @ f` can now be used for evaluation of functions at points.
- The intrinsics `ProjectiveFunction` and `RestrictionToPatch` have been added to convert function field elements to elements of the field of fractions of the coordinate ring of (an affine patch of) the scheme of the function field.
- Intrinsics `Saturate` and `IsSaturated` have been added. The first saturates the defining ideal of a projective scheme, if this has not already been done (see Summary). The second returns whether the current defining ideal for a scheme has been saturated or not.

Bug Fixes:

- `ProjectiveClosure` has been fixed for multi-graded ambient spaces.
- Bugs in `Reduction` for Linear Systems have been fixed.

16.2 Algebraic Curves [HB 95]

Removals and Changes:

- The general curve type `Crv` has been generalised to include all 1-dimensional schemes. For integral (reduced and irreducible) curves, all of the old (integral) plane curve functionality has been extended.
- There is also a new implementation of function fields of curves. There is still an algebraic function field in the background, but the new function field is now the main user interface to rational functions on curves. The conversion between the new function field elements and old-style scheme functions (as used in scheme maps, for example) has been greatly simplified and is now automatic in most cases. We have also extended the curve interface to include curve analogs of several algebraic function field intrinsics which were previously unavailable. See subsection 16.1 above.
- Some signatures taking divisor groups or sets of places have been replaced by signatures taking a curve argument instead or only one argument if the curve can be deduced from the second argument.
- `ProjectiveCurve` for function fields of curves and for places and divisors and their parent structures has been removed. The intrinsic `Curve` will return the projective curve stored on the function field.

New Features:

- An algorithm `PointsElkies` (also callable via `Points`) has been added to search for points on a plane curve. There is a special version `PointsCubicModel` for cubic models, and additionally there is `PointsQI` for point-searching on the intersection of two quadrics in $\P^3$.
- `IsGeometricallyHyperelliptic/IsHyperelliptic` have been added. The first function determines whether a general algebraic curve is hyperelliptic over the algebraic closure of its base field and, if so, also returns a 2-1 scheme map from the curve to a plane conic or the projective line. The second function determines whether the curve is hyperelliptic over the base field (has Weierstrass form) and, if so, returns a hyperelliptic model in `CrvHyp` and a scheme isomorphism from the curve to the model.

Bug Fixes:

- Several bugs in `Adjoints` and `AdjointLinearSystem` have been fixed.

16.3 Resolution Graphs and Splice Diagrams

Bug Fixes:

- Several bugs in `ResolutionGraph` have been fixed.

17 Arithmetic Geometry

17.1 Rational Curves and Conics [HB 98]

Removals and Changes:

- Simon's algorithm for finding points on conics has been folded into the general `IndefiniteLLL` machinery. Much of the functionality to deal with certificates in the Cremona-Rusin method has been eliminated.

17.2 Elliptic Curves [HB 99]

17.2.1 Elliptic Curves over the Rational Field

New Features:

- The intrinsic IsIsogenous has been added for rational elliptic curves; this returns a boolean and a map between the curves. Also, the underlying machinery for isogenies should now be compatible with schemes.
- A new intrinsic FrobeniusTraceDirect has been added to aid the computation of the trace of Frobenius of a rational elliptic curve modulo p. In particular, the hassle of having to create the finite field and coerce the curve into it is avoided.

Removals and Changes:

- The CasselsPairing function has been removed.
- The EllipticExponential (that is, the Weierstrass $\wp$-function) is now computed (for large precisions) using a Newton iteration on the EllipticLogarithm.
- The FourDescent function has been almost completely re-written. Much of the functionality has changed. In particular, most of the intrinsics introduced in V2.11 have been renamed or removed.
- The HeegnerPoint function has been almost completely re-written. In particular, the algorithm now chooses auxiliary discriminant in a much better manner, and can interact with isogenies and descents to find points faster.

Bug Fixes:

- A precision problem reported by N. Bruin with AnalyticRank has been fixed. This naturally affects ConjecturalRegulator also.

17.2.2 Elliptic Curves over an Algebraic Number Field

New Features:

- Intrinsic HasComplexMultiplication has been added. Determines whether the curve has complex multiplication and, if so, also returns the discriminant of the CM order.

17.3 Hyperelliptic Curves [HB 100]

17.3.1 Jacobians of Hyperelliptic Curves

New Features:

- The group law has been extended to hyperelliptic curves with two rational infinite points. Hence arithmetic is now available for the Jacobians of all hyperelliptic curves except those of odd genus with no rational points at infinity. The extension uses similar algorithms to the normal (one point at infinity) case but with extra work needed to keep track of infinite points in the semi-reduced divisors. Additionally, in the odd genus case, reduced divisors of degree $g + 1$ must be allowed and a unique representative of the corresponding divisor class is chosen which depends on fixing a particular point at infinity as the "default" (this is chosen at creation time).

Bug Fixes:

- `ReducedBasis` has been fixed.

17.4 Modular Curves

New Features:

- Intrinsic `WeberClassPolynomial` now returns the minimal polynomial of an appropriate Weber function for any negative discriminant D not congruent to 3 mod 8. It also returns a rational function R such that the corresponding j-invariants are given by $R(\tau)$ where τ are the roots of the Weber polynomial.

18 Incidence Structures

19 Coding Theory

19.1 Linear Codes over Finite Fields [HB 118]

New Features:

- An improvement to the minimum weight algorithm has been developed by Greg White which uses known automorphisms of a code. This method is particularly useful in computing the minimum weight of quasi-cyclic codes, resulting in calculations between 2 and 7 times faster.
- A new algorithm for collecting the minimum word set of a linear code has been developed by Greg White, based on the fast Zimmermann minimum weight algorithm. For example on a $[145, 30, 46]$ binary code, the new method computes the result over 40 times faster than the old methods. The minimum word set is a major first step in the calculation of the automorphism group, and so makes the calculation of the automorphism group of much larger codes now feasible.
- The method for computing the minimum word set has also been generalised for computing sets of words of arbitrary weight. This method is most effective for weights near to the minimum weight, and Magma automatically decides which method will be the fastest.
- An adaptation of the minimum weight algorithm for calculating a partial weight distribution has been developed by Greg White. For example on a $[110, 34, 27]$ binary code, computing the first 4 non-zero values of the weight distribution using the new algorithm takes around 2% of the time taken to compute the full weight distribution.
- The Best Known Linear Codes database has been updated, thanks to the contribution of Markus Grassl, with many of the previous bounds being surpassed by new codes. The database over $GF(2)$ has 905 improved codes, the database over $GF(3)$ has 290 improved codes, and the database over $GF(4)$ has 186 new codes. Further, the 40 missing codes (out of $5,150$) has been reduced to only 14.

19.2 Quantum Stabilizer Codes (New) [HB 121]

A new package for quantum stabilizer codes has been included in this release. Given an additive code C over $GF(q^2)$ which is self-orthogonal to a defined symplectic inner product, then the wordset of $C^{\perp}\backslash C$ defines a quantum stabilizer code over $GF(q)$, where $C^{\perp}$ is the symplectic dual of C.

New Features:

- Invariants of a quantum code, such as ambient space, alphabet, various numerical invariants, the stabilizer and normalizer codes, can be accessed.
- An adaption of the minimum weight algorithm for additive codes is used interchangeably with the weight distribution and MacWilliams transformation, depending on dimension of the code.
- Calculation of the weight distribution of the component stabilizer code, as well as of the word set which defines the error correcting properties of the quantum code.
- The purity of a quantum stabilizer code is automatically detected during the calculation of its minimum weight, and so is easily determined.
- Constructios for CSS codes, cyclic codes and quasi-cyclic codes are provided, as well as constructing codes from graphs.
- Functions for creating new quantum codes from existing ones, such as extending, shortening, puncturing and taking direct sums.
- A database of all best known quantum codes (in terms of minimum weight) is provided for all dimensions and up to length 35, including self-dual codes. These codes come from the tables compiled by Markus Grassl.
- Calculation of the automorphism group of a quantum code, using the underlying software package of J. Leon.
- Facilities exist to map a quantum stabilizer code into its corresponding abelian subgroup of the error group.
- A first module is included for creating and computing with quantum states. Available using both dense and sparse representations, there are standard facilities for computing with states, as well as some basic unitary transformations available.

19.3 Linear Codes over Finite Rings [HB 119]

New Features:

- An adaption of the Zimmermann minimum weight algorithm has been developed by Greg White to calculate the minimum Lee weight of a code over $\mathbf{Z}_4$. The runtimes for the new algorithm are dramatically faster than calculating the full weight distribution. For example with the length 32 Reed Muller code over $\mathbf{Z}_4$, the full Lee weight distribution takes over $2,800$ times longer to calculate than the minimum Lee weight.
- Faster methods for collecting the set of words of minimum Lee weight as well as words of specified Lee weight, as described for Linear codes over finite fields.