Summary of New Features in Magma V2.16

November 2009

1 Introduction

This document provides a terse summary of the new features installed in Magma for release version V2.16 (November 2009).

Previous releases of Magma were: V2.15 (December 2008), V2.14 (October 2007), V2.13 (July 2006), V2.12 (June 2005), 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

Algebraic Geometry

- *Coherent Sheaves*
  - A new package is included providing functionality for working with coherent sheaves on ordinary projective schemes. These are naturally represented by graded modules over the polynomial ring, that is, the coordinate ring of the ambient of the base scheme.
  - There are a number of basic constructors of sheaves, including one for the canonical (dualising) sheaf of an equidimensional, locally Cohen-Macaulay scheme. Further construction operations include tensors, direct sums, tensor powers, Homs and duals.
  - The initial focus, in terms of functionality, apart from the computation of important cohomological invariants of varieties, has been on invertible sheaves (or divisors) and the explicit computation of their associated rational maps into projective space. The map is computed from the "global section submodule" of the sheaf, which in turn comes from the maximal module.
  - For a base scheme X and an effective (Cartier) divisor D on X defined as a closed subscheme by an ideal I, there is code to compute an invertible sheaf corresponding to the class of D. Here, computing the explicit divisor map is essentially the same as computing the Riemann-Roch space if X is a variety. In fact, the Riemann-Roch space can be recovered during the computation of the associated sheaf.

  - A test for isomorphism of sheaves enables linear equivalence of effective Cartier divisors to be tested. Other properties which can be determined include local freeness, and whether the maximal module of the sheaf is arithmetically Cohen-Macaulay.

- *Algebraic Surfaces*

  - For a curve defined over a global field, one may calculate a *regular model* of the associated arithmetic surface, locally at a given prime. From this, one may obtain information such as the component group of the Jacobian at that prime.
  - Parametrization of degree 5 Del Pezzo surfaces has been added. The routines for parametrization of degree 7 and 8 surfaces and for rational scrolls have been updated. The code was provided by Josef Schicho (RISC, Linz).

- *Toric Varieties*

  - Magma V2.16 contains the first stage of a large new package for toric geometry being developed by Gavin Brown, Jaroslaw Buczynski and Alexander Kasprzyk. It incorporates both the combinatorial and Cox ring approaches.
  - The package includes code for cones, fans and polytopes in a rational vector space. Standard operations and constructions are provided including the definition of structures, duality, and lattice point counting within finite polytopes.
  - Toric varieties are defined via fans. Combinatorial tests are provided for the usual geometric properties of the toric variety: singularity, completeness, projectivity. Standard fan-based constructions such as weighted blow-ups of toric subsets are included.
  - Support is provided for working with torus-invariant divisors and divisor-class groups. This includes arithmetic of divisors, equivalence tests, computation of the canonical divisor and the construction of graded cones of the union of Riemann-Roch spaces for all multiples of a divisor.
  - The Cox ring of a toric variety T may be computed. This allows T to be used as a very general single or multi-graded ambient space. Definition of arbitrary closed subschemes of T via homogeneous ideals in the Cox ring is supported. Conversely, Cox rings can be made as abstract objects and the corresponding toric variety and its combinatorics deduced.
  - The basic components of the minimal model program for toric varieties are incorporated, including extremal contractions, generalised flips and an explicit tour of the chambers of the mobile cone (in the sense of Mori dream spaces).
  - The package is integrated with the existing Magma scheme structures, using the Cox ring as the ambient coordinate ring. Many of the basic scheme operations work for subschemes defined via the Cox ring.

Arithmetic Geometry

- *Elliptic Curves*
  - The routine for determining the set of S-integral points on the curve has been replaced with a new implementation. In addition to being reliable, there are a number of new ideas which lead to improved efficiency.
  - The Cassels-Tate pairing between elements of the 2-Selmer group has now implemented for elliptic curves defined over number fields.
  - A new implementation of "elliptic curve Chabauty" based on the Mordell-Weil sieve has been included.
  - An algorithm based on the Mazur-Tate algorithm has been implemented to compute the p-adic height of a point on an elliptic curve over the rationals.
  - A routine is included to compute the automorphism group of an elliptic curve over an arbitrary field. It is returned as an abstract group together with a map to the group of actual automorphisms.
  - Some speedups have been introduced for point counting over small prime fields, in particular, for fields of cardinality less than 2^{30}.
- *Hyperelliptic Curves*
  - Two-cover-descent has been implemented by Nils Bruin for hyperelliptic curves. This is two-descent on a hyperelliptic curve, over the rationals or a number field. This is not the same as two-descent on the Jacobian, and it yields more precise information about rational points on the curve. One obtains the "2-Selmer set" of the curve (which is defined to be the 2-Selmer group of an abelian variety).
- *L-Functions*
  - Many new types of L-functions have been added, together with utility functions for working with them. The most prominent new L-functions are for Hecke characters, and Hecke Grössencharacters, which are important in the development of Tate's thesis. This appears to be the first general implementation of L-functions for Hecke Grössencharacters.
  - The `TensorProduct` intrinsic has been widened in its application. The new intrinsic is quite powerful, and has been used to identify (numerically, via the functional equation) the spinor L-function for various Siegel modular forms.
  - A related addition is the construction of symmetric powers. These have been implemented in the simplest cases (for degree 1 L-functions), and for elliptic curves over the rationals. The tensor power construction is of interest in studies of the Sato-Tate conjecture, as it can compute the exact Euler factor at bad primes for any symmetric power of an elliptic curve over the rationals.

Arithmetic Geometry (Modular Forms)

- *Arithmetic Fuchsian Groups*
  - The fundamental domain routine has been further improved, and a function has been added for solving the word problem.
- *Modular Symbols*
  - A routine has been added to decide whether two given newforms are twists of each other.
- *Hilbert Modular Forms*
  - The existing package has been extensively reworked resulting in better performance and greater reliability. The restriction to squarefree level has been lifted, one can now obtain a NewSubspace relative to any level. A general procedure has been implemented for obtaining new (and old) spaces of a given space from knowledge of dimensions and the Hecke action on spaces of lower level. The field over which Hecke operators are expressed has been changed to the natural one, determined by the Galois structure of the weight (it is $\mathbf{Q}$ in parallel weight).
- *Modular Forms Over Imaginary Quadratic Fields*
  - A new package is included that computes modular forms over arbitrary imaginary quadratic fields. The method involves the 'Sharbly' complex and Voronoi polyhedra, and was developed in practice by Paul Gunnels and Dan Yasaki. These techniques made it possible to automate calculations which in previous implementations had to be done separately for individual fields. The current version computes Hecke operators (at ideals that have odd order in the class group) on spaces of cusp forms of weight 2 with trivial character.
- *Admissible Representations of $GL_2(Q_p)$*
  - A new package, developed by Jared Weinstein, treats local Langlands theory for GL_2 over $\mathbf{Q}$. Starting with a newform in a space of classical cusp forms, and a prime p, one can construct the local component at p of the associated automorphic representation. This is an admissible representation of $GL_2(Q_p)$. One can compute key features of this, such as principal series parameters or a cuspidal inducing datum. Furthermore, the local Langlands correspondence associates to this a Galois representation on the absolute Galois group of $\mathbf{Q}_p$. One can compute (the restriction to inertia of) that Galois representation.

Commutative Algebra

- *Polynomial Rings*
  - Multivariate polynomial multiplication and division has been made much faster, using a simple implementation of the heap-based algorithms of Monagan and Pearce.
  - The factorization of multivariate polynomials has been further optimised for some classes of input (in particular for polynomials over the integers with more than 2 variables).
  - Bivariate factorization has been improved via better use of deflation techniques.
- *Gröbner Bases*
  - An interface between Magma and the SAT solver Minisat has been developed, allowing one to apply SAT methods when solving polynomial systems over $GF(2)$.
  - When applying the F_4 algorithm to systems of polynomial equations defined over $GF(2)$, an early termination criterion is used based on the occurrence of linear polynomials. This often yields a non-trivial speedup in the hardest step.
- *Ideals and Modules*
  - Improvements to the algorithm for constructing the minimization of a free resolution has led to significant speed-ups.
  - The computation of the primary decomposition of an ideal has been improved leading to significant speed-ups in some cases.
  - The calculation of the colon ideal of an ideal is now significantly faster for some inputs. Being able to compute the colon ideal rapidly is critical to many key calculations in algebraic geometry.
  - The computation of a minimal basis for a non-homogeneous ideal or module has been improved through use of automatic homogenization.
  - The algorithm of Eisenbud and Sturmfels has been implemented for computing a maximal regular sequence of elements inside an ideal I of a polynomial ring over a field. It is designed to produce a sequence of reasonably sparse polynomials.
  - For an ideal I of an affine algebra R, given by an ideal in the polynomial ring of which R is a quotient, code is provided to construct the polynomial ideal J whose quotient algebra is isomorphic to the Rees algebra $R(I)$ of I over R.
- *Differential Rings*
  - Routines for the factorisation of linear differential operators over differential Laurent series rings have been implemented by Alexa van der Waall. Both coprime index 1 factorisation and LCLM factorisation are supported.

Global Arithmetic Fields

- *Number Fields*
  - Dirichlet and Hecke characters, including Hecke Grössencharacters in some cases, have been implemented as the duals of `RayResidueRing` and `RayClassGroup`, and should allow group operations. This appears to be the first known general implementation for Hecke Grössencharacters and their L-functions; the only previous code was in PARI/GP and was only for finite order Hecke characters in special cases (largely Hilbert characters).
  - A new algorithm for the computation of the subgroup of K^* generated by a set of elements has been implemented. This enables one to conveniently work in subgroups of the multiplicative group.
- *Algebraic Function Fields*
  - A new algorithm has been implemented for the computation of p-maximal and maximal orders in Artin–Schreier extensions. The same techniques have also been applied to compute the prime splitting in those cases. In particular, when constructing arithmetic-geometric codes, those two new algorithms improve performance by several orders of magnitude.
- *Galois Theory*
  - Following the model of the highly successful implementation of the Galois groups over $\mathbf{Q}$ which allow the computation of Galois groups of (reducible) polynomials of arbitrary degree, a similar algorithm for Galois groups of function fields in positive characteristic has been implemented.
  - Support for the use of complex approximations in the computation of Galois groups over $\mathbf{Q}$ has been added.
  - A new algorithm for computing invariants for intransitive groups yields a reduction of computation time of several orders of magnitude for reducible polynomials.
  - A generic, field independent algorithm for the computation of subfields following new ideas of Klüners and van Hoeij has been implemented. While the new algorithm does not result in improved performance over $\mathbf{Q}$, it is generic and thus, for the first time, allows computation of subfields of global function fields.

Group Theory

- *Finite Groups*
  - New algorithms for constructing all subgroups from maximal subgroups have been implemented, bringing speed improvements, and the ability to handle groups with larger abelian chief factors than the previous method allowed. With this machinery it is straightforward to construct the 111,004 conjugacy classes of subgroups of the simple group, Fischer Fi_{22}, which has order 6,456,175,165,400.
  - Magma includes a database containing information about almost simple groups G, where $S \leq G \leq \mathrm{Aut}(S)$ and S is a simple group. The groups G that are included in the database are those associated with S such that $|S|$ is less than 16000000, as well as M_{24}, HS, J_3, McL, $Sz(32)$ and $L_6(2)$. The groups in the database are defined on *standard generators* which can be used to create an isomorphism between an almost simple group in some arbitrary representation and the "standard" version of it stored in the database. The database was originally conceived by Derek Holt with a major extension by Volker Gebhardt and sporadic additions by Bill Unger.
  - Black-box recognition is available for the first time for the classical groups $SU(4, q)$. The existing black-box recognition for the families $SU(3, q)$ and $Sp(4, q)$, q even, has been upgraded to include rewriting algorithms. The recognition is performed using Brooksbank's algorithm and has been implemented by Peter Brooksbank.
- *Matrix Groups*
  - A package developed by Alla Detinko, Dane Flannery and Eamonn O'Brien allows the user to determine whether or not a matrix group defined over a rational function field is finite.
  - The intrinsic `SubgroupLattice` has now been implemented for finite matrix groups for which a base and strong generating set can be found. The intrinsic `Subgroups` (which determines the conjugacy classes of subgroups) was installed for matrix groups in an earlier release.
  - A database of the maximal finite irreducible subgroups of $Sp_{2n}(\mathbf{Q})$ for $1 \leq i \leq 11$ (constructed by Markus Kirschmer) is included.

- *Finitely Presented Groups*
  - Some machinery has been developed by Derek Holt for computing with subgroups of a free group. For most operations the subgroups are allowed to have finite or infinite index. The functions include index in the free group, free generators for a subgroup, element membership and enumeration of short elements. It is also possible to determine the intersection of two subgroups of a free group.
  - The automorphism group of a free group may be computed (Derek Holt).
  - As a result of experiments which involved trying to establish whether certain fp-groups are finite or infinite, revisions were made to key tools such as the p-quotient and Todd-Coxeter functions. These changes have resulted in Magma being able to resolve a significantly higher proportion of examples. It is now possible to settle (without any human intervention) the question for all but 11 instances of the 13,646 distinct one-relator quotients of the modular group where the additional relator has length 36.
  - The newly-published Plesken-Fabianska algorithm for finding infinite $PSL(2, K)$-quotients of a finitely presented group has been implemented as part of V2.16.
  - The `Homomorphisms` function has been extended by Derek Holt so that it is now possible to search for homomorphisms from an fp-group into a (small) soluble group given by a power-commutator presentation.
  - A simple function has been provided which tries to construct the regular representation of a finite fp-group and then search for a permutation representation having much smaller degree. This function has been successfully applied to groups of order up to 600,000,000.

Lattices

- *Lattice Reduction*
  - Functions have been developed by Damien Stehlé for computing Hermite-Korkine-Zolotarev reduced bases of lattices. HKZ-reduction is an alternative to LLL-reduction. It is significantly more expensive to obtain, but it provides lattice bases of much better quality (i.e., shorter and basis vectors that are closer to being mutually orthogonal).
- *Lattice Enumeration*
  - It is now possible to prune the tree that is explored during the enumeration of short vectors. Although it may result in some vectors being missed, it can make the computations faster by factors higher than 100 if a small probability of an incorrect output is acceptable.
- *Automorphism Group*
  - An improved algorithm for computing the automorphism group of an integral lattice has been developed. The algorithm can handle lattices having a much larger number of vectors of minimal norm than its predecessor. The result is that it is much faster than the old algorithm and can handle significantly larger lattices. For instance, it is able to compute the automorphism group of some of the easier lattices of dimension 48 in the Sloane-Nebe database. A similar algorithm for determining isometry of a pair of lattices will be provided shortly.
- *Database*
  - A new version of the Sloane-Nebe database has been constructed. This version contains the Θ-series series and automorphism groups for most of the lattices. A number of errors in the original have been corrected.

Representation Theory

- *Splitting G-modules and A-modules*
  - A new Meataxe algorithm has been developed for splitting general A-modules, where A is a finite dimensional matrix algebra defined over the rational field. This yields an effective algorithm for decomposing a module into indecomposable summands. If the module is a G-module for some group G, extensive use is also made of character theory. Representations associated with characters having non-trivial Schur indices are properly handled. The difficult problem of splitting homogeneous modules (direct sums of the same indecomposable) is handled by decomposing the endomorphism ring of the module via a maximal order. Schur indices are properly handled. Modules having dimensions in the several hundreds are routinely split into indecomposable modules.
  - A method for extracting a particular G-module from a large degree permutation module defined over a number field has been implemented. The algorithm is based on Nickerson's "Split-P" condensation method. The new feature is the use of the Michler-Weller algorithm for character values of constituents of a permutation module to identify an appropriate condensed vector to spin in the uncondense stage.
  - Tools for constructing the condensation of permutation modules, tensor products and induced modules over fields of either characteristic zero or characteristic p are included in the release.
- *Irreducible Rational Representations*
  - An effective algorithm has been developed for computing irreducible $\mathbf{Q}[G]$-modules for a finite group G. Character theory is used to identify a (reducible) module M that contains the desired module. The Meataxe described above is then used to split the module M thereby yielding the required irreducible module. Condensation is applied to reduce the dimensions of the modules that have to be split.
  - Lattice-based techniques have been developed that control the growth of coefficients at every stage.
  - A variant of the algorithm can determine all irreducible $\mathbf{Q}[G]$-modules. The machinery has been used to construct irreducible $\mathbf{Q}[G]$-modules having dimensions well over a thousand.
- *Integral Representations*
  - A constructive version of the Jordan-Zassenhaus theorem for determining all classes of non-equivalent integral representations over a number field has been implemented (in the case where the representations are absolutely irreducible over the field).

3 Language and System Features

New Features:

- The memory manager has been extended so that on Linux systems, better use is made of the `mmap()` system call. This causes less memory fragmentation and generally ensures that freed memory is returned to the operating system dynamically (within a single function call). This means that Magma may use significantly less memory than before for some inputs. The new use of `mmap()` also fixes some problems occurring in some newer versions of Linux (involving randomly mapped shared system libraries).

4 Aggregates

4.1 Records

New Features:

- There is now proper handling for printing of records with circular references.

5 Algebraic Geometry

5.1 Schemes

New Features:

- A new type of scheme map, a scheme graph map of type `MapSchGrph`, has been introduced as an alternative to the current `MapSch` maps. These are produced by certain intrinsics in the new coherent sheaves module and there is also a basic construction intrinsic for general use. They are currently only available for maps between ordinary projective schemes.

  Graph maps are defined intrinsically by the closure of the graph G of a rational map $X \to Y$. For computational ease, we take G as a subscheme of the product of the ambients of X and Y - a product projective space. Functionally, it is defined by a bihomogenous ideal in a polynomial ring with $n + m + 2$ variables, where n (resp. m) is the dimension of the ambient of X (resp. Y).

  There is a simple basic intrinsic `SchemeGraphMap` for the construction of such a map by the user. The arguments are the domain X, the codomain Y and an ideal I defining the graph in an $n + m + 2$ variable polynomial ring P as described above. P must have the grevlex ordering. I must be large enough to define the graph pointwise as a scheme. A naturally-defined I will often not be the maximal defining ideal, but the intrinsic automatically saturates it with respect to a suitable domain variable (unless the user indicates that this has already occurred via a `Saturated` parameter) which is functionally all that is required. This is a rather primitive constructor with only minimal checking on the input data. Graph maps are more naturally constructed and returned from functions such as `DivisorMap(S)`, where `S` is an invertible sheaf.

  Graph maps have most of the functionality of `MapSch` maps including `IsInvertible` and `Expand`. The major difference currently is that it is not possible to ask for the image or preimage under a graph map of a point in a pointset over a proper extension of the base field. Graph maps can be composed, but not mixed with `MapSch` maps. The graph map format has some advantages over that of `MapSch` for a number of function calls. A graph map is automatically maximally defined, so `Extend` and alternative equations are unnecessary. Computation of images of subschemes of the domain or of the inverse of a map go, in one way or another, through the graph of the map, so it is more efficient to already have it in graph form. For an invertible graph map, separate inverse equations are not required. It is only necessary to record that it is invertible (and saturate by a codomain variable) and consider the reverse of the graph.

  There is a function `SchemeGraphMapToSchemeMap` that converts a graph map f into an equivalent `MapSch`. If f is known invertible, this also computes inverse defining polynomials. It should be noted that for maps between complicated schemes, this often produces a `MapSch` with extremely high degree defining polynomials and a large base scheme where it is not defined. In such cases, the original `MapSchGrph` is a functionally much more efficient representation.

Bug Fixes:

- A memory leak in `Saturate` has been removed. (V2.15-14)
- Bugs, which caused points to be missed, in `PointSearch` have been fixed. The first involved erroneously choosing primes of bad reduction, and the second failed to reduce a lattice modulo a prime power when many derivatives vanished. Singular points were also being ignored in some cases.

5.2 Sheaves

New Features:

- A new package is included providing functionality for working with coherent sheaves on ordinary projective schemes. These are naturally represented by graded modules over the polynomial ring, that is, the coordinate ring of the ambient of the base scheme.
- The first major task that the package deals with is the computation of the maximal (separated) graded module attached to the sheaf starting from the defining module. The aim was to do this efficiently in reasonable generality. The maximal module is the direct sum of global sections of all Serre twists of the sheaf and is needed for several applications.
- The basic assumption is that the exact support of the sheaf S - a subscheme of the base scheme - has all irreducible components of the same dimension > 0 and that S has no non-generic associated points on this support. The implementation computes the maximal module via a double dual calculation treating the defining module as a module over the polynomial ring giving a linear Noether normalisation of the coordinate ring of the exact support of S.
- There are a number of basic constructors of sheaves, including one for the canonical (dualising) sheaf of an equidimensional, locally Cohen-Macaulay scheme. Further construction operations include tensors, direct sums, tensor powers, Homs and duals.
- The initial focus, in terms of functionality, as well as the computation of important cohomological invariants of varieties, has been on invertible sheaves (or divisors) and the explicit computation of their associated rational maps into projective space. There is an intrinsic `DivisorMap` for this, which also returns the image of the map. The map is computed from the "global section submodule" of S, which in turn comes from the maximal module. It is naturally computed and returned as a `MapSchGrph`, the new type of scheme graph map. This gives a method of computing important maps like canonical, anticanonical or adjunction maps on general varieties.
- For a base scheme X and an effective (Cartier) divisor D on X defined as a closed subscheme by an ideal I, `DivisorToSheaf` computes an invertible sheaf corresponding to the class of D. Here, computing the explicit divisor map is essentially the same as computing the Riemann-Roch space if X is a variety. In fact, the Riemann-Roch space can be recovered during the computation of the associated sheaf in a usually more compact form than from later computation with the divisor map of the sheaf. Thus we provide a `RiemannRochBasis` intrinsic that returns a basis in explicit form (as a sequence of polynomials on the ambient and a denominator) as well as the associated sheaf. The computation relies on the fact that, for appropriate $r > 0$, the invertible sheaf of D is isomorphic to the rth Serre twist of the ideal sheaf defining a complementary divisor to D in rH where H is a hyperplane divisor.
- There is some basic functionality for homomorphisms between sheaves on the same base scheme, kernels, images etc.
- There are tests intrinsics `IsLocallyFree`, which tests for local freeness and also returns the degree, `IsArithmeticallyCohenMacaulay`, which tests whether the maximal module of the sheaf S on X is a Cohen-Macaulay module as a graded module over the coordinate ring of X (if S is the structure sheaf of X, this just tests whether X is arithmetically Cohen-Macaulay in its current projective embedding), and `IsIsomorphic` for whether two sheaves on the same X are isomorphic.
- The type for a coherent sheaf is `ShfCoh` and for a sheaf homomorphism `ShfHom`.

5.3 Algebraic Surfaces

New Features:

- For a curve C defined over a global field F (the rationals, a number field or a univariate function field), and a prime p of the ring of integers $\mathbf{Z}_F$, one may obtain a `RegularModel` of the associated arithmetic surface. The model has generic fibre C, and is regular on its special fibre above p; however it is not necessarily a minimal model.

  The routine returns an object of type `CrvRegModel` which stores a number of patches that define the model, as well as the components of the special fibre and other data. From this object, one may access information of interest such as the `ComponentGroup` of the Jacobian. One may also access equations for the patches of the model, and

  The current implementation imposes some extra restrictions on which curves and fields are allowed; this will improve in subsequent releases. Also, additional functions to extract information from regular models may be added on request.
- The new intrinsic `ParametrizeDegree5DelPezzo` is provided for parametrizing a degree 5 Del Pezzo surface (that may be singular, i.e., degenerate) anti-canonically embedded in P^5. This is also linked to the general rational hypersurface parametrization routine `ParametrizeProjectiveHypersurface`, plugging that special case gap. The parametrization routines for degree 7 and 8 Del Pezzos have been updated for speed and efficiency and similarly for the parametrization of Rational Scrolls.

5.4 Toric Varieties

New Features:

- Magma V2.16 contains the first stage of a large new package for toric geometry being developed by Gavin Brown, Jaroslaw Buczynski and Alexander Kasprzyk. It incorporates both the combinatorial and Cox ring approaches.
- The package includes code for cones, fans and polytopes in a rational vector space. Standard operations and constructions are provided including the definition of structures, duality, and lattice point counting within finite polytopes.
- Toric varieties are defined via fans. Combinatorial tests are provided for the usual geometric properties of the toric variety: singularity, completeness, projectivity. Standard fan-based constructions such as weighted blow-ups of toric subsets are included.
- Support is provided for working with torus-invariant divisors and divisor-class group. This includes arithmetic of divisors, equivalence tests, computation of the canonical divisor and the construction of graded cones of the union of Riemann-Roch spaces for all multiples of a divisor.
- The Cox ring of a toric variety T may be computed. This allows T to be used as a very general single or multi-graded ambient space. Definition of arbitrary closed subschemes of T via homogeneous ideals in the Cox ring is supported. Conversely, Cox rings can be made as abstract objects and the corresponding toric variety and its combinatorics deduced.
- The basic components of the minimal model program for toric varieties are incorporated, including extremal contractions, generalised flips and an explicit tour of the chambers of the mobile cone (in the sense of Mori dream spaces).
- The package is integrated with the existing Magma scheme structures, using the Cox ring as the ambient coordinate ring. Many of the basic scheme operations work for subschemes defined via the Cox ring.

6 Arithmetic Geometry

6.1 Rational Curves and Conics

Bug Fixes:

- The `IsotropicSubspace` command in dimension 3 now uses conic code.

6.2 Elliptic Curves

6.2.1 General Elliptic Curves

Removals and Changes:

- The functions `FormalLog`, `FormalGroupLaw` and `FormalGroupHomomorphism` now consistently use the standard choice of parameter $T = -x/y$ (as in Silverman). Previously some of them used $T = x/y$.
- For an elliptic curve (with type `CrvEll`), `AutomorphismGroup` and `Automorphisms` previously returned automorphisms of the underlying curve, in the category of curves rather than elliptic curves.

New Features:

- A function `AutomorphismGroup` determines the automorphisms of an elliptic curve over its base field (which may be any field that is adequately supported in Magma). It returns an abstract group (an abelian or polycyclic group), and also a map sending group elements to concrete automorphisms.
- The `FormalLog` has been made more efficient (using Newton iteration).

6.2.2 Elliptic Curves over the Rational Field

New Features:

- A new implementation of `SIntegralPoints` is included in Magma V2.16-2, replacing the flawed one that has been there for some years. In addition to correctness, significant improvements in performance will be noticed. There are a number of innovations in the method for reducing the bound. The special case of `IntegralPoints`, which had already been mostly rewritten and made reliable in Magma V2.14, has also benefited significantly from these improvements.
- The ability to compute the p-adic height of a point on a curve over the rationals has been added. Similarly, the p-adic regulator can now be used. These rely on the `EisensteinTwo` function, which computes the relevant Eisenstein series.

Bug fixes:

- A long standing bug in the rank computation has been fixed. This bug would occasionally cause the upper bound on the rank to be incorrectly reported as one or even two less than it should be. It could arise when there were “small” two-coverings with hard to find points, and “large” two-coverings with easy to find points.
- The `pAdicEllipticLogarithm` routine has been replaced with a correct one, which obtains answers to a precision which may be specified as an optional argument.
- A bug whereby `CasselsTatePairing` was occasionally unable to find local points has been fixed.

6.2.3 Elliptic Curves over Number Fields

New Features:

- The `CasselsTatePairing` between elements in the 2-Selmer group of an elliptic curve is now also implemented for curves over a number field. This uses the same algorithm as for curves over **Q**. The most expensive step is to solve a conic defined over the number field.
- A new implementation of the method known as "elliptic curve Chabauty" developed by Nils Bruin is included. This uses a combination of Mordell-Weil sieving and Chabauty's method.

6.2.4 Elliptic Curves over Finite Fields

New Features:

- Some speedups have been made for point counting over small prime fields, in particular of size less than 2^{30}. This was a combination of using alternative code, and switching from the old floating-point method (optimised for SPARCs) to one using 64-bit integers.

6.3 Hyperelliptic Curves

New Features:

- `HyperellipticCurveFromIgusaClebsch` now works over number fields.
- A function `TwoCoverDescent` performs descent by 2-covers on a hyperelliptic curve over a number field. It returns the "2-Selmer set" of the curve. (This is not the same as 2-descent on the Jacobian of the curve.) If the 2-Selmer set is empty, the curve has no rational points. It can be applied to hyperelliptic curves of degree 4, providing a way to do 4-descent on elliptic curves over number fields.

6.4 L-Series

New Features:

- Many new types of L-functions have been added, together with utility functions for working with them. The most prominent new L-functions are for Hecke characters, and Hecke Grössencharacters. These are important in the development of Tate's thesis.
- The machinery for tensor products has been improved, so that they can be computed in more situations of arithmetic interest.
- Complementary to tensor products are symmetric powers, which have been implemented in the simplest cases (for degree 1 L-functions), and for elliptic curves over the rationals. Although general machinery exists for symmetric powers, it is not yet very robust.
- Utility functions that provide access to information associated with an L-functions, such as `EulerFactor`, have been added.
- Internally, some of the computations with products of L-functions have also been changed.

Bug fixes:

- A problem that arises when applying `TensorProduct` to LSeries has been fixed. The "weight" of the tensor product should now be correct for $GL(2)$ (for instance, when tensoring two elliptic curves, or more generally two modular forms).

7 Arithmetic Geometry (Modular Forms)

7.1 Arithmetic Fuchsian Groups and Shimura Curves

New Features:

- The computation of a `FundamentalDomain` has been sped up by optimizing certain parameters.
- A function `WordProblem` has been added that expresses an element of a Fuchsian group G as a word in the generators of G.

7.2 Modular Forms

Removals and Changes:

- Some speed-ups have been achieved by replacing inefficient code, for instance in coercions and in `AtkinLehnerOperator`.

Bug fixes:

- A bug with `EisensteinProjection` of a modular form that was already in an `EisensteinSubspace` has been fixed.

7.3 Modular Symbols

New Features:

- A function `IsTwist` has been added, that determines whether two given newforms (specified as spaces of modular symbols) are twists of each other (by a Dirichlet character of p-power conductor, for a given prime p). Also, the function `IsMinimalTwist` determines whether a given newform is a twist of another one with lower level.
- Several low-level functions for manipulating modular symbols have been massively sped up by more careful coding. This mainly improves certain operations on spaces with more than one Dirichlet character.
- Low-level functions `ModularSymbolApply` and `ModularSymbolRepresention` can now be called at the user level.

7.4 Hilbert Modular Forms

Changes:

- The package has been extensively revised. The code (for NewSubspace in particular) is now organised more logically (and correctly), avoiding premature computation, caching spaces appropriately, and so on.
- Many speed-ups have been achieved in the "definite" case, by means of some additional tricks in the precomputation phase, and by more careful coding throughout.

- The field over which Hecke operators are expressed is now the natural one, determined by the Galois structure of the weight (it is always $\mathbf{Q}$ in parallel weight).

New Features:

- The dimension of spaces is now computed by a "formula": in the definite case this involves a sum of class numbers, and in the indefinite case is basically the genus formula for Fuchsian groups.
- The restriction to squarefree level has been lifted; the `NewSubspace` relative to any level can now be obtained. A general procedure has been implemented for obtaining new (and old) spaces of a given Hecke module. This works by determining the dimensions, and the Hecke action, for the relevant spaces of lower level, and using that information to split off the oldspaces.
- In the "indefinite" case, code has been developed to use the fundamental domain of a single Fuchsian group of some discriminant D to compute spaces whose level is a multiple of D. This is a huge improvement compared with directly computing fundamental domains of the Eichler orders with discriminant equal to the desired levels. The precomputed fundamental domain may be reused when creating a new space by setting the optional argument `QuaternionOrder`. The computation of the fundamental domain, which remains the bottleneck, has itself been improved.
- In the "indefinite" case, the computation of Hecke operators has been greatly sped up by use of new techniques for finding a generator of a principal ideal. In particular, the reduction algorithm utilizes knowledge of the fundamental domain of the Fuchsian group. The function `IsPrincipal(I, Gamma)` calls this code.
- A parallelized version of the precomputation in the "definite case" is available on request.

Bug Fixes:

- Various minor bugs have been fixed.

7.5 Modular Forms Over Imaginary Quadratic Fields

Features:

- This new package computes spaces of modular forms over an arbitrary imaginary quadratic field (referred to as *Bianchi modular forms*). The function `BianchiCuspForms` creates the space of cuspidal forms of weight 2 with given level and trivial character.
- These spaces have type `ModFrmBianchi`. The relevant verbose flag is `Bianchi`.
- The computation of the space involves an expensive precomputation phase which depends only on the field. Essentially, this consists of determining the classes of perfect forms over the field. The results of this phase are returned by `VoronoiData`, and this can later be passed in when creating another space over the same field, to avoid repeating the precomputation.
- The space that is computed internally contains some Eisenstein series; however, these are recognised by their eigenvalues and quotiented out.
- The `HeckeOperator` T_I can be computed on these spaces, for an ideal I that has odd order in the class group.
- The `NewSubspace` and its `NewformDecomposition` can be computed (using generic code for dealing with Hecke modules that is also used for Hilbert modular forms).
- The functionality is likely to be extended in subsequent releases.

7.6 Admissible Representations of $GL_2(Q_p)$

Features:

- This new package deals with local components of the automorphic representation associated to a cuspidal newform. The newform is specified by giving (one component of the `NewformDecomposition` of) a space of modular symbols. Given this and a prime p, the function `LocalComponent` constructs the associated admissible representation of $GL_2(Q_p)$.
- These objects have type `RepLoc`. The relevant verbose flag is likewise `RepLoc`.
- One may compute the `Conductor` of an admissible representation, and whether it `IsMinimal` with respect to twisting.
- One may determine whether the representation is in the principal series, or is supercuspidal. In the first case, one may compute the `PrincipalSeriesParameters`; or otherwise a `CuspidalInducingDatum`.
- The package also computes the local `GaloisRepresentation` that corresponds to an admissible representation under the local Langlands correspondence. (More precisely, the Galois representation is described by returning its restriction to the inertia subgroup of a finite extension of Q_p.)

8 Arithmetic Fields (Global)

8.1 Ring of Integers

New Features:

- `ContinuedFraction` of a rational number is now coded at C level.
- A new function `ConvergentsSequence` returns the first n continued fraction convergents, for given n (previously they could only be obtained individually).

8.2 Dirichlet Characters

Changes:

- `Exponent` of a Dirichlet group now returns the exponent of that group, rather than the exponent of the full group $(\mathbf{Z}/N)^\times$.

New Features:

- `Conductor` has been coded in an efficient way.
- Elements of a Dirichlet group are now cached, instead of being endlessly recreated.

8.3 Algebraic Number Fields

Changes:

- Support for the use of complex approximations in the computation of Galois groups over $\mathbf{Q}$ has been added.
- A new algorithm to compute invariants for intransitive groups yields a reduction of computation time of several orders of magnitude for Galois groups of reducible polynomials.
- Removed an unnecessary restriction on orders being maximal when testing whether ideals are prime.
- The `MaximalOrder` computation in Kummer extensions has been sped up by avoiding the construction of intermediate p-maximal orders. (V2.15-3)
- The application of the residue field map to inputs with denominators has been improved. (V2.15-6)
- Some expensive computations with orders have been avoided which has considerable effect in maximal order computations. (V2.15-6)
- The factorisation of ideals is now returning an ordered sequence.

New Features:

- An implementation of Dirichlet and Hecke characters for number fields has been added. Dirichlet characters are on elements of the field, while Hecke characters are on ideals. These are implemented as the dual group of `RayResidueRing` and `RayClassGroup` respectively. The structures have been given their Magma types, allowing substructures, pullbacks, etc., on the groups of characters. A preliminary implementation of Hecke Grössencharacters is also now available, at least for those of type A_0 over CM fields. This appears to be the first general implementation for computing with Hecke Grössencharacters and their L-functions.

- Computing a prime decomposition of a prime which totally ramifies in a Kummer extension now uses a similar algorithm to that implemented for computation of p-maximal and maximal orders.
- A new algorithm for the computation of the subgroup of K^* generated by a set of elements has been implemented. This allows convenient access to the subgroups of the multiplicative group.

Bug Fixes:

- A problem with `QuadraticClassGroupTwoPart` in some cases was fixed.
- A bug in certain computations with trivial Ray class groups has been fixed.

8.4 Algebraic Function Fields

Removals and Changes:

- Removed an unnecessary restriction on orders being maximal when testing whether ideals are prime and in some operations on prime ideals.
- `GaloisGroup` of a global function field has been reimplemented.
- The `MaximalOrder` computation in Kummer extensions has been sped up by avoiding the construction of intermediate p-maximal orders. (V2.15-3)
- The application of the residue field map to inputs with denominators has been improved. (V2.15-6)
- Some expensive computations with orders have been avoided which has considerable effect in maximal order computations. (V2.15-6)

New Features:

- A new algorithm has been implemented for the computation of p-maximal and maximal orders in Artin–Schreier extensions.
- Computing a prime decomposition of a prime which totally ramifies in an Artin–Schreier or a Kummer extension now uses a similar algorithm to that implemented for computation of p-maximal and maximal orders.
- The algorithm of Klüners and van Hoeij for computing subfields of a generic field has been implemented and is used to compute the `Subfields` of a global function field.
- `InfiniteDivisor`, `FiniteDivisor`, `FiniteSplit` of a divisor have been added.
- A `sub` constructor has been added for function fields.

Bug Fixes:

- The `ext<>` constructor for infinite orders of function fields has been fixed.
- It is now possible to take the `Valuation` of a rational function at a place, after the fix of an input check. (V2.15-3)
- The computation of p-maximal orders of function fields which are a direct Kummer extension of a rational function field has been fixed. (V2.15-6)
- A problem with product representations has been fixed. (V2.15-7)
- Some bugs resulting in incorrect answers from `Expand` and the application of completion maps have been fixed. (V2.15-8, V2.15-10)

9 Arithmetic Fields (Local)

9.1 Series Rings

Removals and Changes:

- Some improvements have been made to `Factorization` of polynomials over series rings.
- The integer ring or the field of fractions is now remembered by a series ring.

New Features:

- Some of the basic arithmetic functions have been improved when the result has high precision.
- Extensions of series rings can be constructed having unbounded precision.
- For elements of extensions of series rings `IsUnit` has been added.

Bug Fixes:

- Some precision handling has been improved for extensions of series rings.
- The construction of an unramified extension by a polynomial not over the coefficient ring of the series ring has been fixed. (V2.15-11)
- The computation of roots of a polynomial over a series ring with low precision has been fixed. (V2.15-13)
- The coercion of rational functions into series rings over the integer ring has been fixed. (V2.15-13)

9.2 General Local Fields

Bug Fixes:

- A bug in the creation of a local field as an extension of a ramified extension which was an extension of an unramified extension has been fixed. (V2.15-10)
- Bugs in the `RamifiedRepresentation` of a local field have been fixed.

10 Basic Rings and Fields

10.1 Real and Complex Fields

Removals and Changes:

- Magma nows uses MPC V0.6 and MPFR V2.4.1.
- `BesselFunction` now uses MPFR V2.4.1 instead of the MP reals.
- `Max` and `Min` of real numbers now use MPFR max and min functions directly.
- For real numbers `Dilog` now uses MPFR V2.4.1 instead of PARI.
- For complex numbers the functions `Arg`, `Log`, `Sin`, `Cos`, `Tan`, `Sinh`, `Cosh`, `Tanh` all now use MPC V0.6.

New Features:

- `GetMPFRVersion`, `GetMPCVersion` and `GetGMPVersion` have been added.
- `BesselFunctionSecondKind` has been added from MPFR V2.4.1.

Bug Fixes:

- The intrinsics `Dilog` and `Polylog` have been made available again for complex numbers. (V2.15-3)
- The trignometric functions `Tan` and `Cot` for complex numbers have been corrected. (V2.15-13)

11 Coding Theory

11.1 Linear Codes over Finite Rings

New Features:

- A package, developed by J. Pernas, J. Pujol and M. Villanueve, that extends the existing facilities in Magma for codes over the ring Z_4 is included.
- Constructions are given for families of codes, including Hadamard codes and Reed-Muller codes: `HadamardCodeZ4`, `ExtendedPerfactCodeZ4`, `Reed-MullerCodeZ4`, `Reed-MullerCodeQRZ4`, `Reed-MullerCodeLRMZ4`, and `Reed-MullerCodeMZ4`.
- Another group of functions produce new Z_4-codes by modifying in some way the code words of given Z_4-codes. This includes the functions `PlotkinSum`, `QuaternaryPlotkinSum`, `BQPlotkinSum`, `DoublePlotkinSum`, and `DualKronecker`.
- Functions are provided which compute the rank and dimension of the kernel of any code over Z_4. The functions are:- `SpanZ2CodeZ4`, `KernelZ2CodeZ4`, `DimensionOfSpanZ2`, `DimensionOfKernelZ2`.

12 Commutative Algebra

12.1 Polynomial Rings

New Features:

- The intrinsic `Valuation` of a polynomial at another polynomial has been added.

12.2 Multivariate Polynomial Rings

New Features:

- Multivariate polynomial multiplication and division has been sped up greatly, using a simple implementation of the heap-based algorithms of Monagan and Pearce.
- Multivariate polynomial factorization has been greatly sped up for some classes of input (in particular for polynomials over the integers with more than 2 variables).
- Bivariate factorization has been improved via better use of deflation techniques.

12.3 Ideal Theory and Gröbner Bases

New Features:

- The pair handling in the F_4 algorithm has been improved, leading to speedups in computing Groebner bases for some inputs.
- The new parameter `ReversePairs` has been introduced for `Groebner` and related functions. It is relevant when setting `PairsLimit` to a particular value, so as to choose the subset of pairs in opposite order rather than by default.
- The intrinsic `ColonIdeal` is now much faster for certain inputs.
- New function `MinimalDecomposition` to minimize a decomposition of an ideal.
- The function `AbsoluteAlgebra` now accepts affine algebras over finite fields.
- New function `RegularSequence` computes a maximal regular sequence in an ideal of a polynomial ring over a field. The algorithm used is that of Eisenbud and Sturmfels. It is designed to produce a sequence of reasonably sparse polynomials.
- New function `ReesIdeal`. For an ideal I of an affine algebra R, given by an ideal in the polynomial ring of which R is a quotient, this function returns a polynomial ideal J whose quotient algebra is isomorphic to the Rees algebra $R(I)$ of I over R. There is also a “flattening” option that quotients out by a-torsion for a a given non-zero divisor of R.

12.4 Differential Rings

New Features:

- Given a linear differential operator L over a differential Laurent series ring, the intrinsic `Factorisation` produces a factorisation of L. Both coprime index 1 factorisation and LCLM factorisation are supported.

13 Graph Theory

13.1 Graph Databases

New Features:

- Several of the collections of graphs found on Brendan McKay's webpage `http://cs.anu.edu.au/ bdm/data/graphs.html` have been included among the optional databases. Specifically, the following new familes are included:-

  - *Simple graphs*: All graphs on 2–10 vertices.
  - *Connected simple graphs:* All graphs on 2–10 vertices.
  - *Eulerian graphs:* All graphs on 2–12 vertices.
  - *Connected Eulerian graphs:* All graphs on 3–11 vertices.
  - *Connected planar graphs:* A collection of non-isomorphic connected planar graphs on 1–11 vertices.
  - *Self-complementary graphs:* All graphs on 4, 5, 8, 9, 12, 13, 16, and 17 vertices; An incomplete list of graphs on 20 vertices.

  Additionally, the database of *strongly regular graphs* found on McKay's web page has been available for several years.

14 Groups

14.1 Finite Groups

Changes:

- The `MaximalSubgroups` functions for groups of type `GrpAb` and `GrpPC` have been changed to return a sequence of records, as this function does for permutation and matrix groups.

New Features:

- The function `CohomologyGroup` has been sped up significantly.
- New algorithms for constructing all subgroups from maximal subgroups have been implemented, with speed improvements, and the ability to handle groups with larger abelian chief factors than the previous method allowed.

Bug Fixes:

- A number of bugs in the filtering code of the `Subgroups` family of functions have been fixed. Generally these bugs caused subgroup classes to be missed, rather than causing a crash.

14.2 Permutation Groups

Changes:

- Constructions of `PSO` and `POmega` (and their plus/minus forms) have been changed so that they return subgroups of the corresponding `PGO`.
- The `ExtraSpecialGroup` function now returns a minimal degree permutation representation, rather than the regular representation previously returned.
- The `FPGroup` function has had restrictions on the group order lifted to $2^{30} - 1$, equal to the current limit for the degree of a permutation group.

New Features:

- The test for maximality of a subgroup of a permutation group now uses an algorithm which exploits Magma's ability to construct maximal subgroups in large groups.
- Blocks action code has been modernised, along with use of the minimal block finder, to make better use of the fast minimal blocks algorithm. This has impacted important routines such as finding the soluble radical and Sylow subgroups, the basic test `IsAltsym`, and degree reduction code. Dealing with high degree transitive and imprimitive groups, such as are produced from fp-groups using the Todd-Coxeter, has been much improved.
- Testing for regular groups has been modernised, and more use is made of the regularity property, particularly when the group order or a presentation is required.
- A new operation on permutations has been installed: `TensorProduct`. This is equivalent to converting the two permutations to matrices, taking tensor product of matrices, and converting back to a permutation, but without making a matrix.

14.3 Matrix Groups

New Features:

- Construction of orbits for a matrix group has been modernised to use new matrix action code. This improves both the `Orbit` function and construction of base and strong generating set.
- `RandomSchreierBounded` function installed. Construction stops if a basic orbit length exceeds the given bound. Intended for deciding whether or not to use BSGS methods for a matrix group.
- Improved algorithms for computing the soluble radical of matrix groups over the integers and rationals (with a BSGS) have been included.
- The intrinsic `SubgroupLattice` has now been implemented for finite matrix groups for which a base and strong generating set can be found. The intrinsic `Subgroups` (which determines the conjugacy classes of subgroups) was installed for matrix groups in an earlier release.
- Generators for the special and general linear groups over the integers have been included.
- A package developed by Alla Detinko, Dane Flannery and Eamonn O'Brien allows the user to determine whether or not a matrix group defined over a rational function field is finite. If it is finite then functions are provided which return the order and an isomorphic matrix group defined over a finite field.
- Black-box recognition is available for the first time for the classical groups $SU(4, q)$. The existing black-box recognition for the families $SU(3, q)$ and $Sp(4, q)$, q even, has been upgraded to include rewriting algorithms. The recognition is performed using Brooksbank's algorithm and has been implemented by Peter Brooksbank.
- A database of the maximal finite irreducible subgroups of $Sp_{2n}(\mathbf{Q})$ for $1 \leq i \leq 11$ (constructed by Markus Kirschmer) is included.

14.4 Finite Soluble Groups

Changes:

- The `MaximalSubgroups` functions for groups of type `GrpAb` and `GrpPC` have been changed to return a sequence of records, as this function does for permutation and matrix groups.

New Features:

- The p-group function `Omega` has been revised to improve performance in the case in which the group is not abelian.

Bug Fixes:

- A number of low-level problems in the pc-group module have been addressed. These include crashing when many pc-groups were created and deleted, many instances of repeated unpacking of data, and slow speed of pc-group constructors.
- A bug in the function `IsConditioned` has been fixed. The bug caused some presentations that were not conditioned to be treated as a conditioned, and may have been the cause of many problems.

14.5 Finitely Presented Groups

New Features:

- An implementation of the Plesken-Fabianska algorithm for determining whether a fp-group has $PSL(2, K)$, K an algebraic extension of $\mathbf{Q}$, as a quotient is included.
- The `Homomorphisms` function has been extended so that it is now possible to search for homomorphisms from a finitely-presented group into a (small) soluble group given by a power-commutator presentation (pc-group). Code supplied by D.F. Holt.
- Two functions for converting *finite* finitely presented groups of moderate cardinality to other types have been added. These are `PermutationGroup` and `PCGroup`.
- Machinery is now provided that allows a user to define automorpisms and automorphism groups for finitely-presented groups.

Bug Fixes:

- The `IsNormal` command for fp-groups has been corrected to use the inverses of the group generators as well as the generators when conjugating, possibly correcting results for infinite groups.

14.6 Finitely Presented Abelian Groups

Changes:

- The `MaximalSubgroups` functions for groups of type `GrpAb` and `GrpPC` have been changed to return a sequence of records, as this function does for permutation and matrix groups.

14.7 Groups Defined by Rewrite Systems

Bug Fixes:

- A bug where each call to `RWSGroup` and `RWSMonoid` left an empty file in the tmp directory has been fixed.

14.8 Automorphism Groups

New Features:

- Machinery is now provided that allows a user to define automorpisms and automorphism groups for finitely-presented groups.

14.9 Databases of Groups

New Features:

- Magma includes a database containing information about almost simple groups G, where $S \leq G \leq \mathrm{Aut}(S)$ and S is a simple group. The groups G that are included in the database are those associated with S such that $|S|$ is less than 16000000, as well as M_{24}, HS, J_3, McL, $Sz(32)$ and $L_6(2)$. The groups in the database are defined on *standard generators* which can be used to create an isomorphism between an almost simple group in some arbitrary representation and the "standard" version of it stored in the database.
- A database of the maximal finite irreducible subgroups of $Sp_{2n}(\mathbf{Q})$ for $1 \leq i \leq 11$ (constructed by Markus Kirschmer) is included.

15 Lattices

New Features:

- Functions that compute HKZ-reduced bases of matrices, Gram matrices and lattices have been implemented. A Hermite-Korkine-Zolotarev reduced basis starts with a shortest non-zero lattice vector, and orthogonally to the first basis vector the remaining vectors are themselves HKZ-reduced. HKZ-reduction is a very strong notion of reduction, providing bases of much better quality than LLL-reduction. It is however much more expensive to obtain. HKZ-reducing a lattice may allow the user to solve problems on a given lattice more easily, enumerating short and close vectors being two natural examples. The functions `GaussReduce` and `GaussReduceGram` are restrictions of the HKZ functions in dimension 2.
- A new `SetVerbose`("HKZ", b) flag allows the user to obtain information during the computation of an HKZ-reduced basis.
- The function `EnumerationCostArray` provides a priori information on the efficiency of the executions of the functions `Minimum`, `CentreDensity`, `CenterDensity`, `KissingNumber`, `ShortVectors`, `ShortVectorsMatrix`, `ShortestVectors`, `ShortestVectorsMatrix` and `ThetaSeries`. The information is more precise than that provided by the function `EnumerationCost` provided in the previous release, as `EnumerationCostArray`(L, u) gives a heuristic evaluation of the size of each layer of the tree to be visited during the enumeration of vectors within the prescribed norm u, rather than the sum of the sizes of the layers.
- A new `Prune` option has been added for the functions `Minimum`, `CentreDensity`, `CenterDensity`, `EnumerationCost`, `KissingNumber`, `ShortVectors`, `ShortVectorsMatrix`, `ShortestVectors`, `ShortestVectorsMatrix` and `ThetaSeries`. The `Prune` option is also available for the new functions `EnumerationCostArray` and `HKZ`. It allows the user to finely prune the tree to be considered during the enumeration. The output may not be correct anymore, but by using the `EnumerationCostArray` function, the user can heuristically estimate the running-time speed-up and the likeliness of an incorrect output.
- The function `ReconstructLatticeBasis` takes as input an arbitrary basis of a lattice and a full rank set of short linearly independent vectors. It returns a basis of the lattice that is not much longer than the full-rank set of linearly independent vectors.
- An improved algorithm for computing the automorphism group of an integral lattice has been developed. The algorithm can handle lattices having a much larger number of vectors of minimal norm than its predecessor. The result is that it is much faster than the old algorithm and can handle significantly larger lattices. For instance, it is able to compute the automorphism group of some of the easier lattices of dimension 48 in the Sloane-Nebe database. A similar algorithm for determining isometry of a pair of lattices is also provided.
- A new version of the lattice database, with slightly different functionality, is now available. The main feature is the addition and checking of many more automorphism groups, and similarly with Θ-series. A few new lattices have been added, and some duplicates have been removed. The information about Hermitian bases has not been included, but can be added if users request the Magma group to do so.

Bug Fixes:

- Two local solubility glitches in dimension 4 for `IsotropicSubspace` have been fixed.
- Another local solubility problem was also fixed, and a failure to minimize in some cases (particularly dimension 6) were also fixed.
- A bug with the 2-adic genus of a lattice was fixed.

16 Lie Theory

16.1 Coxeter Groups/Reflection Groups

New Features:

– A very efficient algorithm for computing the growth function of a Coxeter group has been designed by Bob Howlett and implemented by Bob and Bill Unger.

– The degrees of the fundamental invariants can now be computed for all complex reflection groups, not just real groups (`BasicDegrees`). Basic codegrees can also be computed. The algorithm is due to Lehrer and Taylor.

– Reflection groups can now be defined over additional number field types, for example, `FldQuad`.

16.2 Groups of Lie Type

New Features:

– Frobenius maps can be evaluated efficiently for groups over finite fields.

– The order of a twisted finite group of Lie type can be computed.

16.3 Representations

New Features:

– Direct sum decompositions can now be determined for representations of Lie algebras and groups of Lie type, and for modules over Lie algebras.

17 Linear Algebra and Module Theory

17.1 Matrices

New Features:

- Matrix multiplication over rational function fields has been greatly sped up.
- Matrix multiplication (and related operations) for matrices over GF(3) and GF(4) sped up by use of SSE instructions where applicable.
- New functions `ReverseRows` and `ReverseColumns` to reverse the rows/columns of a matrix.
- New function `AddScaledMatrix(A, s, B)` to compute $A + s \cdot B$ efficiently for matrices A, B and scalar s.
- New function `TraceOfProduct(A, B)` to compute `Trace(A*B)` efficiently.
- The algorithms for multiplying dense vectors by sparse matrices have been improved.

17.2 Vector Spaces

New Features:

- The procedure `Include`, which includes a vector in a vector space in place, has been greatly improved in efficiency.

17.3 R-Modules

Changes:

- The function `RSpaceWithBasis` now works over general euclidean rings (not just fields).
- New function `IsPermutationModule` for A-modules.
- New function `CentreOfEndomorphismRing` for A-modules, which computes the centre of the endomorphism ring of a module (and is generally much faster than computing the endomorphism ring and then its centre for some rings).

17.4 Modules Hom(U, V)

New Features:

- There is now better support for elements of matrix spaces, considered as maps on vector spaces, etc.

18 Linear Associative Algebras

18.1 Orders in Associative Algebras

Bug Fixes:

- A bug in `RepresentationMatrix` of elements of orders of algebras has been fixed.
- Orders of algebras over number fields, that have been constructed from a ring and a sequence of algebra elements, now store coefficient ideals bringing them into line with other orders.
- An error in `IsUnit` for an element (e.g. a matrix or a polynomial) over an order has been fixed.

18.2 Matrix Algebras

New Features:

- A new algorithm is used to compute a Z-basis of a maximal order of a central simple algebra over Z (as a sequence of rational matrices).
- A faster version of the algorithm for computing the centre of a matrix algebra has been incorporated.
- The implementation of the Carlson-Matthews algorithm for computing a presentation of a matrix algebra now runs considerably faster.

18.3 Quaternion Algebras

Changes:

- In `RightIdealClasses`, when the optional argument `Support` is specified, precisely this support is now used. (Previously it was enlarged by the prime divisors of the discriminant of the order.)
- The functions `Embed`, for embedding a quadratic field or order in the algebra, and `pMatrixRing`, for identifying a specified *order* with a matrix ring, are now implemented for algebras over $\mathbf{Q}$.

Bug fixes:

- `RightIdealClasses` is now correctly implemented for Eichler orders.

18.4 Quantum Groups

Changes:

- The type `AlgPBW` and those inheriting from `AlgPBW` no longer inherit from the type `GenMPolB`. They still inherit from `Rng` and `AlgInfD`.

18.5 Finitely Presented Associative Algebras

Changes:

- The parameter `MaximumTime` for `QuotientModule` now uses the reals based on the MPFR reals. (V2.15-3)

19 Representation Theory

19.1 Modules over Algebras

New Features:

- A new Meataxe algorithm has been developed for splitting general A-modules, where A is a finite dimensional matrix algebra defined over the rational field. This yields an effective algorithm for decomposing a module into indecomposable summands. If the module is a G-module for some group G, extensive use is also made of character theory. Representations associated with characters having non-trivial Schur indices are properly handled. The difficult problem of splitting homogeneous modules (direct sums of the same indecomposable) is handled by decomposing the endomorphism ring of the module via a maximal order. Modules having dimensions in the several hundreds are routinely split into indecomposable modules. Such modules are created via such functions as `GModule` and `RModule`, as for modules over finite fields.

19.2 $K[G]$-Modules

New Features:

- An algorithm has been developed for computing irreducible $\mathbf{Q}[G]$-modules for a finite group G. Given a rational character of G, the algorithm proceeds by locating a (reducible) module that contains the desired module. Then using the Meataxe described above, the module M is split thereby yielding the required irreducible module. Use is made of condensation to reduce the dimensions of the modules that have to be split. The algorithm controls the growth of coefficients at every stage, thus returning modules whose actions are usually defined by matrices with very small integral entries. A variant of the algorithm is provided which determines all irreducible $\mathbf{Q}[G]$-modules for G. The machinery has been used to construct irreducible $\mathbf{Q}[G]$-modules having dimension well over a thousand in favourable circumstances. New functions:
  - `IrreducibleModules(G, RationalField())` to compute all or some irreducible modules for G over $\mathbf{Q}$ (with many options).
  - `RationalCharacterTable(G)` to compute the table of irreducible rational characters for G.
  - `GModule(chi, RationalField())`: compute irreducible module for given irreducible rational character.
- A specialised method for splitting a large-degree permutation module to obtain a specific irreducible has been included. The algorithm is a combination of the Michler-Weller algorithm for determining character values of constituents of a permutation representation, together with Nickerson's "Split-P" condensation method. The character values are used to identify the correct module to be uncondensed to obtain the G-module affording the given character. The results of Michler and Weller allow an algorithmic search for the right vector to spin, as opposed to Nickerson's heuristic approach.

19.3 Character Theory

New Features:

- The function `RationalCharacterTable(G)` returns the table of irreducible rational characters.
- The intrinsic `CharacterTable` now applies to finite groups of type `GrpAb`.

20 Topology

20.1 Simplicial Complexes

New Features:

- A very basic module for defining and computing with simplicial complexes developed by M. Johansson is released for the first time in V2.16.
- The module supports creation of simplicial complexes from lists of faces, as well as a few preprogrammed complex types. Standard techniques for modifying and combining simplicial complexes are available.
- The most important operation supported is the calculation of the (reduced) homology of a simplicial complex with coefficients in a designated ring. This in turns allows the Euler characteristic to be determined.