Summary of New Features in Magma V2.18

December 2011

1 Introduction

This document provides a terse summary of the new features installed in Magma for release in version V2.18 (December 2011). A small number of new features were exported in patch releases subsequent to the main release of V2.17 in December 2010 and these are also listed here for completeness. Only significant bugfixes are noted here – for a more complete list of bugfixes the reader should consult the patch release change log for V2.17-x.

Recent previous releases of Magma were: V2.17 (December 2010), V2.16 (November 2009), V2.15 (December 2008), V2.14 (October 2007), V2.13 (July 2006).

2 Highlights

Algebraic Geometry

- *Schemes*
  - New functionality for finding isolated points (zero-dimensional subschemes) on fairly general schemes has been added. This includes functions to first to attempt to reduce the complexity of a system via linear substitution and resultants, and then to lift a local point to a global point (via Newton's method) once one is found. Notable applications of this method, including the computation of a polynomial defined over a quartic field (constructed by N. D. Elkies) whose monodromy group appears to be M_{23} are given in the Handbook.
- *Curves*
  - A. J. Wilson has contributed a package to compute the Log Canonical Threshold of a single point or of all singular points on a curve. This invariant contains important geometric information about the differentials with poles along an effective $\mathbf{Q}$-Cartier divisor of a normal variety.
  - A package for computing minimal degree maps to $\mathbf{P}^1$ and smallest degree plane models for general curves of genus less than 7 is nearing completion. These 'gonal' maps and small degree models may require a finite extension of the base field. The package will be added to the V2.18 export distribution in a patch release early in 2012.

Arithmetic Fields

- *Global Fields*
  - An algorithm based on the Self Initializing Quadratic Sieve due to Mike Jacobson (*Math. Comp., 1999*) for generating relations in quadratic orders has been implemented by Jean-Francois Biasse. For fields with discriminants having more than 20 decimal digits, this method is significantly faster than the standard strategy of generating relations via lattice enumeration in a large number of lattices. It enables computation of class groups for orders with discriminants having up to 90 decimal digits on a standard processor.
  - An analogue of the number field sieve has been implemented by Jean-Francois Biasse for the calculation of class groups for low degree number fields. This generalizes the approach of Jacobson for quadratic fields. For fields of low degree and large discriminant, the new implementation is much faster. Work is still underway to fully optimise the sieve-based routine, particularly in the non-quadratic case, and to make the best choice of changeover points with respect to the old algorithm.
  - A new method for unit group computations (which arise as a necessary step in class group computations) due to Claus Fieker is used. In this method, p-adic approximations of the logarithms of the Archimedian valuations are used instead of floating point approximations, and it provides a better way of controlling loss of precision in calculations with units.
  - A new algorithm due to Jürgen Klüners and Mark van Hoeij for finding all subfields of finite extensions of $\mathbf{Q}$ has been installed. For fields having many subfields, this can be much faster than the existing algorithm. For example, the new algorithm is twice as fast as the old algorithm in finding the 58 subfields of a degree 60 field. In the case of a degree 32 field, the new algorithm finds its 373 subfields in about 200 seconds as compared to more than 8 hours using the existing algorithm.
- *Galois Theory*
  - The new Galois group algorithm of Claus Fieker and Jürgen Klüners has been extended so that it can now determine the Galois group for a (reducible) polynomial over any global arithmetic field.
  - The implementation of the Fieker-Klüners Galois group algorithm has been improved in a number of ways. In particular, improvements to the code for computing maximal subgroups for larger permutation groups enables the calculation of Galois groups in much larger degree in some cases.

- *Databases of Fields*
  - Databases of number fields having degrees 2 through 9 are now available in Magma. In the case of degree 2, the enumeration lists all 607,925 fields having (absolute value of the) discriminant less than a million. For fields of degree 3, all fields having discriminant d such that $-10^6 < d \leq 2000000$, and all $C3$ fields with $d < 117527609329$ are included. For degrees greater than 3 the databases include fields having small discriminants (small absolute value), as well as various other fields that may be of interest. Stored with each number field is the discriminant, the signature, the Galois group, the class number and the class group. This database was constructed by the Pari and KANT groups.
  - A database of function fields corresponding to low degree extensions of the base finite field is also provided. For a given finite field K, we list the degrees n for which all function fields of degree n having base field K are tabulated. GF(2): 2, 3; GF(3): 3; GF(4): 2, 3, 5; GF(5): 2, 3, 4, 8; GF(7): 9; GF(11): 3, GF(13): 3. Stored with each function field are the genus, the number of degree one places, the class number, and the class group. This database was constructed by Claus Fieker. (Note that while both the number fields and function fields have previously been available as Magma databases, through an oversight, they were neither announced nor was documentation available prior to this release).

Arithmetic Geometry

- *Elliptic Curves: General*
  - The intrinsic IsLinearlyIndependent for determining whether a set of points on an elliptic curves is linearly independent over the integers has been rewritten to avoid linear algebra over the reals (with heights) as much as possible. The new version is not only more robust but it is also faster.
  - A much improved algorithm has been installed for computing complex elliptic logarithms.
  - Rigorous lower bounds on the height of non-torsion points can be calculated.
- *Elliptic Curves Over* **Q**
  - The extension of John Cremona's database for elliptic curves curves over **Q** having conductor up to 200,000 has been installed.
  - The machinery for performing descent on elliptic curves over **Q** has been expanded. New features include an algorithm of Creutz for doing full 9-descent in the general case, algorithms for 5- and 7-isogeny descents due to Fisher, and algorithms for second 5- and 7-isogeny descents due to Creutz, and a routine for 3-descent-by-isogeny (first and second descent) written by Creutz.
- *Elliptic Curves Over Number Fields*
  - The routine for 2-descent on elliptic curves over number fields has been reimplemented in a more careful way. It now succeeds for a much larger range of input, where previously it either failed to terminate or returned answers involving huge numbers.
  - Minimisation and reduction algorithms for 2-coverings of elliptic curves (as obtained from 2-descent) have been implemented for the first time over number fields. The algorithms were developed by Donnelly and Fisher.
  - The point search routine for elliptic curves over number fields has been substantially improved.
  - A search routine is provided for finding elliptic curves with specified conductors, over number fields (or over **Q**). If some traces of Frobenius are known for the desired curves, the routine is able to take advantage of this knowledge.
- *Hyperelliptic Curves*
  - A package for computing canonical heights on hyperelliptic curves has has been contributed by Steffan Mueller. This is implemented for curves of any genus, and for curves defined over **Q**, number fields and some global function fields.

  - The point search routine for hyperelliptic curves over number fields has been substantially improved.

- *Cyclic Covers of* $\mathbf{P}^1$
  - New algorithms for computing the set of rational points on a curve over a number field of the form $y^q = f(x)$ have been included. Current functionality allows one to search for points, test local solubility and disprove the existence of rational points by computing descent obstructions via an algorithm of Mourao generalising techniques used in the case $q = 2$ by Bruin and Stoll.

- *Algebraic Surfaces*
  - For Del Pezzo surfaces over various types of base field, the Galois action on the set of exceptional curves, and on the geometric Picard group, can be calculated as a permutation group acting on a lattice.
  - Minimisation and reduction routines for Del Pezzo surfaces of degree 3 (cubic surfaces) and degree 4 have been developed by Elsenhans.
  - Isomorphism of cubic surfaces can be determined using code provided by Elsenhans.
  - A collection of tools that obtain bounds on the Picard rank of a surface has been provided by Elsenhans.

Arithmetic Geometry (Modular Forms)

- *Modular Curves*
  - A collection of precomputed data and functions giving nice geometric models and useful modular information for the $X_0(N)$ modular curves that classify elliptic curves with cyclic N-isogenies has been developed by Michael Harrison. The initial database will contain all $X_0(N)$ of genus 0, elliptic, hyperelliptic and non-hyperelliptic of genus up to 6. This covers all $N < 60$ and about half of the N between 60 and 80. For elliptic and hyperelliptic curves, the models are minimal Weierstrass models. For non-hyperelliptic cases, the models are either smooth curves in P^3 or, more usually, minimal degree plane models. The chosen models are integral, given by sparse defining polynomials with small coefficients. Furthermore, they are chosen to have nice reduction properties at primes not dividing N. In addition to the curve models, the database contains a variety of information including automorphisms, projection maps between curves of related levels, cusps and non-cuspidal rational points, standard modular functions such as the j-invariant, and forms such as E_4 and E_6 as meromorphic k-differential forms.
  - A package for the computation of curves of genus > 1 that are images of $X_0(N)$ and $X_1(N)$ (also referred to as modular curves) has been provided by Enrique González-Jiménez. This includes the computation of all hyperelliptic and genus 3 non-hyperelliptic new modular curves of given level, computation of curves related to specific modular abelian subvarieties of the new-part of $J_1(N)$ and some functions to compute not-necessarily-new modular curves.

- *Modular Forms*
  - Code developed by Allan Lauder for computing the characteristic series of Hecke operators on spaces of overconvergent modular forms has been included.

- *Hilbert Modular Forms*
  - Atkin-Lehner operators now be computed (in both the algorithms). Degeneracy maps can be computed in the "definite" algorithm.
  - Computation of new subspaces and decomposition into newforms now uses more efficient techniques (in both algorithms).
  - The "indefinite" algorithm has been greatly improved: fields of arbitrary class number are handled, a new approch to the Hecke precomputation saves substantial work, and the code performs more efficiently overall.

Associative Algebras

- *Quaternion Algebras*
  - The functionality provided for quaternion algebras over $\mathbf{F}_q(x)$ has been extended so that it is now the same as for algebras defined over the rationals or a number field. In particular, it is now possible to construct algebras and orders with given discriminant.
- *Algebras with Involution*
  - A package for working with algebras equipped with an involution (*-algebras) has been developed by Peter Brooksbank and James Wilson. The algebras may be constructed either from a group algebra or as the algebra of adjoints of a system of forms. Constructors are provided for producing any simple *-algebra over a finite field. For algebras defined over a field of odd characteristic, their structure may be determined. Further, the group of isometries of a system of forms over a field of odd characteristic may be computed.
- *Clifford Algebras*
  - A package for working with Clifford algebras has been developed by Don Taylor. At present the emphasis has been on algebras over finite fields. A range of operations that relate to the form of the algebra are provided.

Coding Theory

- *Linear Codes over Finite Fields*
  - A multi-threaded version of the algorithm for computing the minimum weight or words of given length for a linear code over GF(2) is now available (where POSIX threads are supported). When a number of cores are available, this leads to much reduced wall-clock time.

Combinatorial Theory

- *Matrices*

  - Functions have been added to determine the automorphism group of a matrix corresponding to applying the same permutation to both rows and columns. This generalises an existing function which was restricted to 0–1 matrices. The corresponding test for isomorphism of matrices is also included.

Commutative Algebra

- *Multivariate Polynomial Rings*

  - Factorization of multivariate polynomials has been sped up, particularly for polynomials defined over number fields and function fields in characteristic 0.

- *Gröbner Basis*

  - The Faugère F_4 algorithm for computing a Gröbner basis is now significantly faster for ideals defined over $\mathrm{GF}(2^k)$ for $k > 1$. Over $\mathrm{GF}(2^2)$, the linear algebra phase is up to 100 times faster for the dense systems which can arise in algebraic attacks, while for $20 < k \leq 64$ a new special packed representation is also used, leading to non-trivial speed ups.
  - A basic multi-threaded algorithm has been added for the linear algebra phase of the Faugère F_4 algorithm in the case that the base field is $\mathrm{GF}(p)$, where p is a prime with $2 < p < 2^{23.5}$. Running on an POSIX-enabled machine having a number of cores available, leads to a much reduced wall-clock time when the matrices which arise are very large.
  - The Faugère F_4 algorithm is now much faster when used to construct a Gröbner bases for an ideal of a polynomial ring defined over a number field.

- *Ideal Arithmetic*

  - The algorithm used to find the primary decomposition of an ideal has been improved with the result that decompositions in positive dimension have been sped up.

Group Theory

- *Permutation Groups: Backtrack Algorithms*
  - Ordered partition stacks have been implemented with their own type and Magma intrinsics. They implement the data structure described very briefly in Jeff Leon's 1997 paper. They can be used as the basis for implementing various backtrack searches in the Magma language (e.g., lattice automorphism group).
  - It is now possible to compute stabilisers of unordered partitions of the permutation domain of a group. Likewise it is possible to determine conjugacy of unordered partitions. This case was not included in Leon's library.
  - The algorithms used for normalizer and conjugacy testing of subgroups in a permutation group have been upgraded to those given by J. Leon (1997). This ordered partition stack backtrack search is a major improvement over the algorithm due to G. Butler that it replaces.
  - All other backtrack algorithms (ordered partition stabiliser, set stabiliser, centraliser and intersection) which were formerly computed using Leon's routines are now calculated using a new implementation by Bill Unger of partition-refinement backtrack searching. While not necessarily faster than the Leon code, the new implementation avoids many of the memory limitations that arose from the use of the Leon code.

- *Permutation Groups: Subgroup Structure*
  - An improved algorithm has been installed to compute the maximal subgroups of the alternating and symmetric groups. In particular, the need to test conjugacy of subgroups has been removed. The result is very much faster code that allows the determination of maximals up to degree at least 1000.
  - A general revision of the code to determine maximal subgroups of a permutation group has resulted in much improved runtimes.

- *Permutation Groups: Databases*
  - The Butler/McKay/Hulpke transitive group databases for degrees 2-31 have been revised so as to store the groups more compactly and to provide much faster recognition.
  - The transitive groups database has been extended to include the Cannon-Holt database of the 2,801,324 transitive groups of degree 32. Identification of the transitive degree 32 groups is also supported.
  - The existing database of primitive permutation groups has been extended to include the primitive groups of orders between 2501 and 4095 which have been classified recently by Coutts, Quick, and Roney-Dougal.

- *Classical Groups*

  - Efficient methods based on the geometry of quadratic spaces are now used to compute conjugacy classes in those classical groups which lie between the conformal group and the derived group.
  - A package which computes the normaliser of a linear group using an algorithm based on a theorem of Aschbacher has been included for the first time. This allows normaliser calculation within much larger groups that can be handled using backtrack search. This was developed by Hannah Coutts and it generalises work of Colva Roney-Dougal.
  - A package developed by Peter Brooksbank and James Wilson for computing intersections of pairs of classical groups has been installed.

Lattices and Quadratic Forms

- *Forms over Finite Fields*

  - The machinery for bilinear, quadratic and sesqui-linear forms has been revised and greatly extended. In particular, the changes ensure consistency and compatibility with the $*$-algebras package. Other features are summarised in the following points.
  - The standard invariants invariants of the bilinear and quadratic form may be calculated: Witt index, Arf invariant, Dickson invariant, spinor norm.
  - Isometries may be factored into products of reflections.
  - The Wall form of an isometry may be calculated.

- *Forms over* $\mathbf{Q}$

  - The algorithm used for calculating isotropic subspaces of forms defined over $\mathbf{Q}$ has been improved. For even-dimensional input, the new version typically returns a space that has dimension larger by one than the subspace found by Simon's original algorithm.

- *Integral Lattices*
  - Isometry testing of lattices (with and without extra forms) has been greatly improved by using the same methods as have been previously adopted for computing automorphism groups of lattices. These methods use the backtrack search ideas of Plesken and Souvignier, complemented by the ordered partition stack methods of Leon.
  - A function for identifying the theta series of an integral lattice as a modular form is provided for the first time. Various techniques are employed to minimise the amount of lattice enumeration needed to uniquely determine the theta series in the relevant finite dimensional space of modular forms.

Linear Algebra

- *Over Finite Fields*
  - The algorithm for multiplication of matrices over $\mathrm{GF}(2^k)$ for $k > 4$ has been sped up greatly (fast algorithms based on packed representations have been present for $k \leq 4$ in previous versions). As a consequence of this, several critical linear algebra algorithms for matrices over such fields are now faster (particularly in the case of larger matrices).
- *Over* $\mathbf{Z}$ *and* $\mathbf{Q}$
  - Following the introduction of some new ideas, the p-adic algorithm for computing the nullspace of an integer or rational matrix has been greatly sped up This has led to a speedup of several algorithms for integer or rational matrices.
  - The algorithm for computing the Hermite form of an integer matrix has been sped up (independent of the improvements to the nullspace algorithm above). In particular, a better strategy is employed to handle the different classes of input matrices.
  - The algorithm to compute the Smith form (or elementary divisors) of an integer matrix has been sped up, as has the related algorithm for computing the saturation of an integer lattice.

Lie Theory

- *Lie Algebras*
  - Code is provided which makes it possible to test isomorphism in certain classes of Lie algebra. These include semisimple algebras and low dimensional nilpotient and soluble algebras.
  - The robustness and performance of the computation of split toral subalgebras and Chevalley bases have been improved.
  - Improvements have been made to the recognition procedure for reductive Lie algebras.
  - Code for computing nilpotent orbits in simple Lie algebras and for theta-groups has been developed by Willem de Graaf.
  - A facility is provided for computing with Lie algebras generated by extremal elements. The functionality includes computing bases (either over fields, or a basis of the Lie ring over $\mathbf{Z}$), multiplication tables, and associated varieties.
  - It is now possible to determine the semisimple subalgebras of a simple Lie algebra using a database constructed by Willem de Graaf.
- *Kac-Moody Algebras*
  - The first stage of a project to support computing with Kac-Moody (affine) Lie algebras is included in V2.18. The current machinery includes tools to construct such an algebra and to perform arithmetic with its elements.
- *Representation Theory*
  - A W-graph is a combinatorial structure that encodes a matrix representation of a Weyl group W, or more generally, a matrix representation for the Iwahori-Hecke algebra associated to W. The W-graphs for types A_n, E_6, E_7 and E_8 have been supplied by Bob Howlett.
  - Functions have been provided that construct the matrices giving the action of a Hecke algebra representation obtained from a W-graph.
  - The Kazhdan-Lustzig polynomials are now computed more efficiently. Some related procedures have been improved.
  - Code has been contributed by Robert Zeier for studying the irreducible simple subalgebras of the Lie algebra $su(k)$. In particular, all of its irreducible simple subalgebras can be determined.

3 Documentation

New Handbook Chapters:

- Chapter 85: "Algebras with Involution"
- Chapter 98: "Kac-Moody Lie Algebras"
- Chapter 119: "Elliptic Curves over **Q** and Number Fields". This chapter consists of the relevant material formerly in the Chapter "Elliptic Curves".
- Chapter 125: "Small Modular Curves"

Deleted Handbook Chapters:

- The former chapter, "Universal Enveloping Algebras" has been removed and its material included in Chapter 97: "Lie Algebras".

4 Language and System Features

New Features:

- Magma on Mac OS X now uses `mmap` for memory allocation, which results in reduced memory usage in certain circumstances. (Note that nothing has changed for Magma on the Linux operating system: it continues to use `mmap` as before).
- New procedure `SetNthreads(n)` to specify that n parallel threads should be used when possible (for platforms where POSIX threads are supported). Currently, the algorithms which support multi-threading are the coding theory minimum-weight algorithm over GF(2), and the Faugère F_4 Gröbner basis algorithm over medium-prime finite fields.

Bug Fixes:

- A new scheme for garbage collection was released in V2.17. This scheme was designed to overcome Magma's previous inability to delete certain attributes defined in package code. It also addressed memory problems in code for arithmetic fields and other areas. Over the past year a number of bugs resulting from this change have been reported and fixed. At the time of release of V2.18 (Dec, 2011), such reports have become very infrequent. However, users are urged to report every bug they see – no matter how trivial.

4.1 Packages and Attributes

Bug Fixes:

- A bug has been fixed where `import` would sometimes incorrectly report that a package file had not been attached. (V2.17-6)

5 Algebraic Geometry

5.1 Schemes

New Features:

- New functionality for finding isolated points (zero-dimensional subschemes) on fairly arbitrary schemes has been added. This includes functions to first try to reduce the complexity of a system via linear substitution and resultants, and then to lift a local point to a global point (via Newton's method) once one is found.

  New intrinsics include `LinearElimination`, `IsolatedPointsFinder` and `IsolatedPointsLifter`.

- A function `RepresentativePoints` for zero-dimensional schemes, which returns one point in each conjugacy class of geometric points, has been added.

Changes and Removals:

- The normalization of the defining equations of a map into a scheme with one non-trivial grading has been improved to remove some common factors. In particular this affects maps into hyperelliptic curves or their ambients. (V2.17-3)
- The `PointSearch` function now tries to handle singularities more effectively. In addition, the case where the input is an affine scheme has been modified to use the standard projective closure.

Bug Fixes:

- A bug in the application of a `hom` from a function field of a scheme has been fixed. (V2.17-6)

5.2 Algebraic Curves

New Features:

- A.J. Wilson has contributed a package of intrinsics to compute the Log Canonical Threshold of various effective divisors D on a curve C. This invariant contains important geometric information about the differentials with poles along an effective $\mathbf{Q}$-Cartier divisor D of a normal variety V.
- `LogCanonicalThresholdAtOrigin` gives the value for the divisor equal to the single point O on a plane affine curve.
- `LogCanonicalThreshold` or `LogCanonicalThresholdOverExtension` give the values for a divisor D on a general curve C, when D is either a single point or the sum of all singular points on C.

6 Arithmetic Geometry

6.1 Rational Curves and Conics

Bug Fixes:

– A bug with ParametrizationMatrix when given an integral conic was fixed. (V2.17-13).

6.2 Elliptic Curves

6.2.1 General Elliptic Curves

Bug Fixes:

– A few minor bugs in the function `EllipticCurve(C, p)` which caused runtime errors for some inputs have been fixed.

6.2.2 Elliptic Curves over the Rational Field

New Features:

– New intrinsics `NineDescent` and `NineSelmerSet` allow one to perform 9-descents on generic elliptic curves over $\mathbf{Q}$.

– New intrinsics `pIsogenyDescent` and `FakeIsogenySelmerSet` have been added. These allow one to perform first and second p-isogeny descents on elliptic curves with a rational p-torsion point. Taken together these make full 5- and 7-descents practical for elliptic curves with a point of order 5 or 7.

– A new intrinsic `ThreeDescentByIsogeny` provides an alternative to `ThreeDescent` for elliptic curves which admit a 3-isogeny. The new algorithm factors the 3-descent into first and second 3-isogeny descents, resulting in significant speed-ups for larger examples.

Changes and Removals:

– The `IsLinearlyIndependent` intrinsic now uses an algorithm based on images mod 2 rather than height pairings.

– In `ThreeDescent`, the step of trivializing a degree 3 cyclic algebra is now done using Tom Fisher's code.

Bug fixes:

– A bug with local heights of torsion points (involving infinite valuations) was fixed (V2.17-11).

– A bug with local heights at infinity when the discriminant was negative was fixed (V2.17-11).

– Allow QuarticReduce (of 2-covers) to work on cubic inputs more readily. Reported by T. A. Fisher (V2.17-8).

– Fixed a bug with Reduction of a GenusOneModel in the Windows version. A work-around for avoiding indefinite LLL was no longer needed, and could occasionally run into precision problems (V2.17-6).

6.2.3 Elliptic Curves over Number Fields

New Features:

- An intrinsic `MordellWeilShaInformation` is provided, similar to the existing function of the same name for curves over **Q**. This is a meta-function which uses all relevant Magma machinery to obtain as much information as possible about the Mordell-Weil group and the Tate-Shafarevich group of a given elliptic curve.
- The `TwoDescent` routine has been rewritten, and is now capable of handling a much larger range of input. (Here the difficulty is mainly dependent on the discriminant of the base field and the conductor of the curve).
- New algorithms for global minimisation and reduction of 2-coverings are implemented. These are applied to the output of `TwoDescent` when the optional flag `MinRed` is set (and sometimes by default). They may also be applied directly by the user. When no global minimal model exists (due to class group obstructions), it returns a model that is nearly minimal, in the sense that the extra factor appearing in the discriminant is chosen to have small norm. The primes dividing the extra factor may be specified via the optional argument `ClassGroupPrimes`.
- A function `ReducedModel` is provided to do reduction of Weierstrass models of elliptic curves over number fields.

Changes and Removals:

- The `IsLinearlyIndependent` intrinsic now uses an algorithm based on images mod 2 rather than height pairings.
- The routine `TorsionSubgoup` has been made far more efficient.
- Obvious improvements have been made in the computation of the `Conductor`.

Bugs:

- A runtime error in RootNumber (due to insufficient p-adic precision) has been fixed (V2.17-6).

6.3 Genus One Models

New Features:

- The functions `Minimise` and `Reduce` are now implemented for the additional case of models of degree 2 over number fields. They call the new algorithms for global minimisation and reduction of 2-coverings of elliptic curves, which are described elsewhere in these notes.

Changes:

- The type `TransG1` for transformations between genus one models, which was introduced in V2.17, is now used consistently by all functions dealing with transformations.

6.4 Hyperelliptic Curves

Changes and Removals:

- `RationalPoints` now returns all points instead of only one in each $+/-$ pair (in order to match the convention used for other base fields).
- `Involution` now works for points over extensions (V2.17-13).

New Features:

- Functionality has been introduced for dealing with general cyclic covers of the projective line, i.e., curves of the form $y^q = f(x)$, which are a natural way to generalise hyperelliptic curves. Currently a cyclic cover is represented in Magma simply by the polynomial f and the integer q. These routines are implemented for curves defined over $\mathbf{Q}$ and over number fields.
- The function `RationalPoints` can be used to search for points with x-coordinates in a specified range. This is based on the same sieve routine as for hyperelliptic curves.
- The functions `HasPoint` and `HasPointsEverywhereLocally` test local solubility, either at a given prime, or at all places of the base field.
- The routines `qCoverDescent` and `qCoverPartialDescent` generalize `TwoCoverDescent` (for hyperelliptic curves) to the case of general cyclic covers of the projective line.

Bug fixes:

- The `LSeries` of a hyperelliptic curve now first transforms the curve into a `MinimalWeierstrassModel` (rather than a `ReducedModel`). This will solve some problems at $p = 2$, though others still remain.
- A bug in `TwoSelmerGroup` present in the case of odd genus hyperelliptic curves of even degree has been corrected. The local two torsion subgroups were incorrectly computed in some cases (where the polynomial factors locally in a particular way).
- A major bug which caused `TwoSelmerGroup` and `TwoCoverDescent` to be extremely slow or never finish for many inputs has been corrected. The bug was in the code for systematically searching for local points.

7 Arithmetic Geometry (Modular Forms)

7.1 Modular Curves

New Features: **Small Modular Curves Database**

- A Small Modular Curves database has been added that contains simple geometric models for $X_0(N)$ of low level along with associated modular-geometric information.
- The cases so far covered are those where $X_0(N)$ is genus 0, elliptic or hyperelliptic as well as non-hyperelliptic cases of genus less than 7 (there are a few genus 5 and 6 cases not yet added).
- For elliptic and hyperelliptic cases, a minimal Weierstrass model is given. For genus 3 and 4 nonhyperelliptic, a canonical model in $\mathbf{P}^2$ or $\mathbf{P}^3$ is given. For genus 5 and 6 non-hyperelliptic, a smallest degree (6 in every case) singular plane projective model is given. The models over $\mathbf{Q}$ or a field extension may be obtained with the `SmallModularCurve` intrinsic.
- The package contains intrinsics to recover the natural projection maps from $X_0(N) \to X_0(M)$, when $M|N$, that correspond to the map $z \mapsto z$ as quotients of the upper half-plane. Also, when $r|(N/M)$ the "r-projection" maps that correspond to $z \mapsto rz$.
- There are intrinsics to return all of the Atkin-Lehner involutions, the automorphisms that correspond to $z \mapsto z + (1/r)$ on the upper half-plane and an extra "non-matrix" automorphism in the $N = 37$ and 63 cases ($N = 108$ hasn't been added yet). There are also intrinsics to compute and return the full automorphism group (avoiding the generic curve automorphism machinery) over cyclotomic fields (or the group generated by the above special automorphisms in the genus 0 and 1 cases).
- Cusps and non-cuspidal rational points (when genus > 0) are stored and can be returned as points or places.
- The j-invariant function and the forms $E_2^{(N)}$, E_4 and E_6 can be obtained as rational functions, differentials or k-differentials on the models.
- For a non-cuspidal point or place on the model, a moduli representative consisting of an elliptic curve E with cyclic subgroup C of order N or a cyclic N-isogeny $E \to F$ between elliptic curves can be computed.
- There is an intrinsic to return the q-expansions of the coordinate functions of the models to any precision as well as one to return a string giving expressions for these modular functions/forms in terms of basic types like eta-products or theta series.
- There is a new package provided by Enrique González-Jiménez for the computation of modular curves over $\mathbf{Q}$ of genus > 1 which are the images of $X_0(N)$ or $X_1(N)$.

New Features: **Curves as Images of $X_0(N)$ or $X_1(N)$**

- There is an intrinsic `NewModularHyperellipticCurves` that computes all new modular hyperelliptic curves of level N for $\Gamma_1(N)$ or $\Gamma_0(N)$ and a `NewModularHyperellipticCurve` to compute the new modular hyperelliptic curve corresponding to a given modular abelian subvariety of $J_1(N)$ or $J_0(N)$ if it exists.
- There is an intrinsic `NewModularNonHyperellipticCurvesGenus3` that computes all new modular non-hyperelliptic genus 3 curves of level N for $\Gamma_1(N)$ or $\Gamma_0(N)$ and `NewModularNonHyperellipticCurveGenus3` to compute the new modular genus 3 non-hyperelliptic curve corresponding to a given modular abelian subvariety of $J_1(N)$ or $J_0(N)$ if it exists.

– There are also intrinsics `ModularNonHyperellipticCurveGenus3` and `ModularHyperellipticCurve` to compute whether a given modular abelian subvariety of $J_1(N)$ or $J_0(N)$ leads to a not-necessarily-new modular curve which is hyperelliptic or non-hyperelliptic of genus 3. In the hyperelliptic case, the curve returned is modular but may be associated to a different modular abelian variety to the one input.

Changes and Removals:

– Some inefficiencies in the code for producing q-expansions of modular forms have been mitigated.

– The functions `DeleteAllAssociatedData` and `DisownChildren` that enabled deletion (by cleaning up circular references) are deprecated and should not be used. They are obselete since the new system of memory management was introduced in V2.17. The functions were provided for spaces of modular forms or symbols and for Dirichlet characters.

7.2 Hilbert Modular Forms

New Features:

– For spaces computed using the "definite" algorithm, Atkin-Lehner operators are implemented for divisors of the level that are coprime to the discriminant of the quaternion order used. Degeneracy maps in the "downward" direction are also implemented, as maps from a space to itself, for primes divisors of the level satisfying the same condition.

– For spaces using the "indefinite" algorithm, Atkin-Lehner operators are implemented.

– The `NewSubspace` is now computed using degeneracy maps (in one or other direction) for *all* spaces that use the "definite" algorithm. (Previously for non-squarefree levels, naive black-box code was used, which was completely unable to handle large dimensional spaces.)

– The "indefinite" algorithm is implemented with no restriction on the class group.

– Computation of Hecke operators with the "indefinite" algorithm has been improved in many ways. In particular, a new technique in the Hecke precomputation step greatly reduces the number of calls to ideal-principalization needed (using knowledge of the unit group).

– Improvements to the code used for computing subspaces and decomposition into newforms are made as part of ongoing development of the package.

– Facility is provided for obtaining only newforms of specified degrees (for instance, very small degrees) rather than the full decomposition.

– Some additional utility functions are included.

Changes and Removals:

– The algorithm can be specified by the user ("Definite" or "Indefinite"), as an easy alternative to specifying the `QuaternionOrder`.

Bug Fixes:

– A typographical error caused `NewSubspace(M, P)` to compute the wrong space. However, `NewSubspace(M)` was unaffected by this.

7.3 Bianchi Modular Forms

Bug Fixes:

- `NewSubspace`, `NewformDecomposition` and related functions available for Hilbert modular forms are supposed to work in the same way for Bianchi modular forms; however they had become incompatible.

8 Arithmetic Fields (Global)

8.1 Dirichlet and Hecke Characters

New Features:

- The intrinsic `SetTargetRing` can now be used to set the target ring of a character, so that it takes values in a ring other than in a `CyclotomicField`.

8.2 Algebraic Number Fields

New Features:

- Subfields of extensions of $\mathbf{Q}$ can be computed using the newer algorithm of Klüners and van Hoeij. Using p-adic methods this algorithm will be more efficient for fields with large numbers of subfields.
- A new algorithm for computing class groups of number fields (suitable for fields of large discriminant) is included. This is a sieve algorithm, due to Jacobson for quadratic fields and Biasse for general number fields, and implemented by Biasse. The new algorithm can be selected by the user (by setting `Al := "Sieve"`); it may also be selected by default where appropriate. The choice of algorithm does not affect the usage of the `ClassGroup` functions: there is no visible difference to the arguments, optional parameters, or output. In particular, it should be noted that by default all class group computations are done unconditionally, using the Minkowski bound.
- For conditional class group computations (where GRH is assumed), an improved bound is used instead of the Bach bound, due to Belabas et al. (*Math. Comp., 2008*).

Changes and Removals:

- The computation of maximal orders of radical extensions at critical primes has been improved (V2.17-2).
- A more efficient invariant is now used in some special cases in `GaloisSubgroup`. (V2.17-3)
- `RootOfUnity` has been improved for large cyclotomic fields. (V2.17-10)
- Prime selection for the computation of `Subfields` has been improved. (V2.17-10)
- Some multiplications of ideals has been improved. (V2.17-11)
- The `Al` parameter for `MaximalOrder` has been fixed so that setting it to `"Round2` or `"Round4"` will ensure that the specified algorithm is used rather than the special radical or Artin–Schreier algorithms, if applicable. (V2.17-12)
- Maximal order computations in radical extensions have been improved. (V2.17-12)
- The function `pSelmerGroup` can now be called for prime powers (however in that case the map is only implemented in the easy direction).

Bug Fixes:

- The computation of `Subfields` of a relative number field has been fixed. (V2.17-4)
- The computation of `GaloisGroup`s of reducible polynomials has been fixed. (V2.17-4)

– Some memory management in Class group computations has been fixed. (V2.17-4)

– Some `MaximalOrder` computations for radical extensions have been fixed. (V2.17-5, 12)

– A fix has been made to the preimage computation of class group maps for quadratic fields. (V2.17-9)

– Fixes have been made to coercion involving cyclotomic fields. (V2.17-10)

– A fix has been made to elements of global arithmetic fields. (V2.17-10)

– Computation of Absolute orders has been fixed. (V2.17-10)

– A fix has been made to the coercion of elements involving non-simple fields. (V2.17-10)

– Ensure that `IsSubfield` always installs an embedding when returning `true`, there was one missing between cyclotomic fields. (V2.17-11)

– Checking whether elements are in ideals of non-simple relative orders has been fixed. (V2.17-11)

– Computing a 2 element representation of an ideal when the 1st generator is 1 has been fixed. (V2.17-11)

– Computing a 2 element representation of some fractional ideals has been fixed. (V2.17-11)

– A bug causing `HilbertSymbol` to return incorrect answers has been fixed (V2.17-10).

– In the function `pSelmerGroup`, the `Integral` option was not working.

8.3 Algebraic Function Fields

New Features:

– Galois groups of reducible polynomials over function fields having prime characteristic can now be computed.

– The intrinsic `Embed` is now available to give an embedding of one function field into another.

Changes and Removals:

– The computation of maximal orders of radical extensions at critical primes has been improved (V2.17-2).

– Maximal order computations in Artin–Schreier extensions have been improved. (V2.17-8)

– The construction of function fields using polynomials over non-fields has been fixed and expanded. (V2.17-8)

– Powering of elements of coefficient rings of extensions has been optimized by using the coefficient ring for the powering instead of the extension which is the parent. (V2.17-12)

– The computation of infinite equation orders has been improved resulting in defining polynomials with smaller coefficients and larger orders. (V2.17-12)

– The computation of resultants of polynomials over orders of function fields now uses the subresultant algorithm. (V2.17-12)

– The `Al` parameter for `MaximalOrder` has been fixed so that setting it to `"Round2` or `"Round4"` will ensure that the specified algorithm is used rather than the special radical or Artin–Schreier algorithms, if applicable. (V2.17-12)

– Maximal order computations in radical extensions have been improved. (V2.17-12)

– The `Evaluate` intrinsic applied to an element in the product representation at a place has been sped up. (V2.17-12)

Bug Fixes:

– The `Check` parameter to `ext<FldFunRat | >` has been enabled. (V2.17-6)

– Some `MaximalOrder` computations for radical extensions have been fixed. (V2.17-5)

– `Places` of degree of a function field in relative extension representation has been improved to return large degree places over small degree places. These were previously missed. (V2.17-10)

– A fix has been made to the coercion of elements involving non-simple fields. (V2.17-10)

– Checking whether elements are in ideals of non-simple relative orders has been fixed. (V2.17-11)

– Computing a 2 element representation of an ideal when the 1st generator is 1 has been fixed. (V2.17-11)

– Computing a 2 element representation of some fractional ideals has been fixed. (V2.17-11)

– A crash in denominator handling of elements of non-simple relative orders has been fixed. (V2.17-11)

– Computing an expansion at a prime of an element of a non-simple field has been fixed. (V2.17-11)

– `IsSimple` has been fixed for Function fields. (V2.17-11)

– Multivariate factorization and primary decomposition over non-simple function fields has been improved. (V2.17-11)

– Some fixes have been made to linear extensions of function fields. (V2.17-11)

9 Arithmetic Fields (Local)

9.1 p-adic Rings and their Extensions

Changes and Removals:

– `PrincipalUnitGroupGenerators` has been added for `RngPadRes`. (V2.17-5)

Bug Fixes:

– A coercion in which precision loss occurred has been corrected. (V2.17-5)

– `HenselLift` of a root of a polynomial over a p-adic field to a default precision no longer returns roots to the wrong precision. (V2.17-7)

9.2 Series Rings

Changes and Removals:

- Finding multiple roots of polynomials over series rings has been improved (V2.17-3).
- Taking the `Reverse` of a series over a non-field has been disallowed. (V2.17-3)
- Some precision handling in extensions of series fields has been improved when precision of an element is infinite. (V2.17-8)

Bug Fixes:

- A digit of precision loss when calculating the `Reverse` of a series has been fixed. (V2.17-4)
- A check has been added in Log to check whether the logarithm of the first coefficient can be taken. (V2.17-8)

10 Basic Rings and Fields

10.1 Integer Ring

Changes and Removals:

- New function `GetStoredFactors` to return a sequence containing the currently stored integers used in integer factorization.

10.2 Finite Fields

New Features:

- A new memory-saving packed representation has been introduced for GF(2^k) when $20 < k \leq 32$. Fast polynomial and matrix operations over such fields have also been implemented, as well as similar support in the F_4 Gröbner basis algorithm.
- A recently constructed table of all irreducible trinomials over GF(2) for all degrees less than 100,000 has been added to the collection of Magma databases.

10.3 General Rings

New Features:

- The intrinsic `IsRootOfUnity` now works for a wider class of rings.

10.4 Polynomial Rings

New Features:

- The algorithm for factoring polynomials over algebraic number fields has been improved for some classes of polynomials.

11 Coding Theory

11.1 Linear Codes over Finite Fields

New Features:

- A multi-threaded version of the algorithm for computing the minimum weight or words of given length for a linear code over GF(2) is now available (where POSIX threads are supported). When a number of cores are available, this leads to much reduced wall-clock time. Specifying the number of threads can be done via the global `SetNthreads` procedure or via the new parameter `Nthreads` for the functions `MinimumWeight` and `Words`.

12 Combinatorial Theory

New Features:

– Functions have been added to determine the automorphism group of a matrix corresponding to applying the same permutation to both rows and columns. This generalises an existing function which was restricted to 0–1 matrices. The corresponding test for isomorphism of matrices is also included.

13 Commutative Algebra

13.1 Multivariate Polynomial Rings

New Features:

– Factorization of multivariate polynomials has been sped up, particularly for polynomials defined over number fields and function fields in characteristic 0.

Bug Fixes:

– The testing of equality of homomorphisms between algebras which do not inherit from `AlgGen` has been fixed. (V2.17-9)

13.2 Ideal Theory and Gröbner Bases

New Features:

– The Faugère F_4 algorithm for computing a Gröbner basis is now significantly faster for ideals defined over GF(2^k) for $k > 1$. Over GF(2^2), the linear algebra phase is up to 100 times faster for the dense systems which can arise in algebraic attacks, while for $20 < k \leq 64$ a new special packed representation is also used, leading to non-trivial speed ups.

– A basic multi-threaded algorithm has been added for the linear algebra phase of the Faugère F_4 algorithm in the case that the base field is GF(p), where p is a prime with $2 < p < 2^{23.5}$ (and where POSIX threads are supported). When a number of cores are available, this leads to much reduced wall-clock time when the matrices which arise are very large. Specifying the number of threads can be done via the global `SetNthreads` procedure or via the new parameter `Nthreads` to functions/procedures which compute a Gröbner basis.

– The Faugère F_4 algorithm has been greatly sped up for ideals over a number field.

– The computation of a Gröbner basis for an ideal defined by fixed basis (via `IdealWithFixedBasis`) has been improved significantly.

– The algorithm to compute the primary decomposition of an ideal has been sped up in positive dimension.

– The algorithm for the computation of the minimal polynomial of an element of an affine algebra over a function field has been sped up.

14 Groups

14.1 Finite Groups

New Features:

– The permutation and matrix group function `Subgroups` has had an extra parameter, `IndexEqual`, added. The pc-group version of `Subgroups` already had this parameter. This brings them closer together.

Changes:

– P. Brooksbank's package for constructive recognition of SU3, SU4 and Sp4 in even characteristic has been updated.

– The `MaximalSubgroups` function has been improved by eliminating, as far as possible, hard normalizer and centralizer computations.

– Recognition of alternating and symmetric groups by the `MaximalSubgroups` function has been improved in number of special cases, as has constructing their maximal subgroups.

– The warning message about large numbers of groups from the `SmallGroups` command is now completely disabled by setting the parameter Warning to false. Up till now the parameter was ignored for over 100,000 groups.

– Parameters for `LieType` and `RecogniseSU4` are altered to improve performance for certain types of input.

14.2 Permutation Groups

New Features:

– New algorithms have been implemented for computing normalizers and subgroup conjugacy. These algorithms are based on J. Leon's 1997 ordered partition stack backtrack searches. They are a great improvement on the algorithms previously used.

– The `Stabilizer` and `IsConjugate` functions are extended to apply to unordered partitions of the permutation group domain.

– The `Stabilizer` and `IsConjugate` functions have extra optional parameters to allow the user to supply subgroups of the stabilizers of the structures, and, in the case of conjugacy testing, to specify which stabilizer groups should be computed before the conjugacy test is started.

Changes and Removals:

– Backtrack searches for element centralizer, element conjugacy, group intersection, and set and partition stabilizer and conjugacy now use new routines that replace the code of J. Leon that has been used up until now. The algorithms are substantially the same as Leon used. The new implementation will increase reliability and maintainability of these functions.

– The implementation of the *Jellyfish* family of algorithms has been improved in the light of some experience with their use. They are now used internally in computing normal subgroup structure of a primitive group, and in the `IsAltsym` test at the suggestion of R. Beals.

14.3 Matrix Groups – General

Changes:

- The amount of set-up time for `Random` applied to a high degree matrix group has been reduced on the advice of E. O'Brien.

14.4 Matrix Groups Over Finite Fields

New Features:

- The function `FactoredClassicalGroupOrder` has been added at the request of E. O'Brien.

Changes:

- The selection of base points in `RandomSchreier` has been fixed to use the equivalent of the function `GoodBasePoints`. This change was suggested by E. O'Brien.
- A new version of the matrix group `CompositionTree` code has been installed. Code supplied by E. O'Brien.
- The orders of minus type, and all conformal, orthogonal groups are now stored with the group on creation. Omission reported by E. O'Brien.
- The performance of the function `UnipotentStabiliser` for matrix groups has been improved.
- Code for `UnitaryForm` has been improved by D. Holt.
- Code for `IsGLConjugate` has been improved by C. Roney-Dougal.

14.5 Matrix Groups Over Infinite Fields

Changes and Removals:

- A new optional parameter `ExtDegree` has been added for intrinsic `CongruenceImage`. If the matrix group G is defined over a (rational) function field of positive characteristic, the congruence image will be defined over extension of the coefficient field of (at least) this degree.

14.6 Classical Groups

New Features:

- Efficient methods based on the geometry of quadratic spaces are now used to compute conjugacy classes in those classical groups which lie between the conformal group and the derived group.
- The code for finite conformal groups has been revised to ensure that the group order is computed by formula and then stored. This provides an essential speedup for groups of large dimension and/or field.
- Code that constructs matrix generators for spin groups over finite fields has been installed.

– A package which computes the normaliser of a linear group using an algorithm based on a theorem of Aschbacher has been included for the first time. This was developed by Hannah Coutts and it generalises work of Colva Roney-Dougal.

– A package developed by Peter Brooksbank and James Wilson for computing intersections of pairs of classical groups has been installed.

– The machinery for black-box recognition of the groups $SU3$, $SU4$ and $Sp4$ in even characteristic have been improved.

Changes:

A number of small fixes relating to the code producing the maximal subgroups of a classical group have been made. Some of the changes relate to errors in the code and some are changes coming from minor corrections to the Bray/Holt/Ronay-Dougal classification. Here is a summary of the more noteworthy changes.

– A bug in the code for $C9$ maximal subgroups for $O^+(8, q^3)$ of type $3D_4(q)$ has been corrected.

– A bug that led to an error in the non-degenerate $C1$ maximal subgroups of $O^-(12, q)$ has been corrected.

– A subgroup of $Sp(12, q)$ of type $U(3, 4)$, originally listed as a $C9$ maximal, has been deleted.

– The $C9$ maximal subgroups of $SU(9, 2)$ of type $L(2, 19)$ are now listed as novelty maximals.

– The $C9$ maximal subgroups of $SU(9, q)$ ($q = 2, 8 \bmod 15,\ q > 2$) of type $3.A_6.2_3$ are now included.

– A subgroup of $O(11, 11)$ of type $L(2, 11)$, originally listed as a $C9$ maximal, is now omitted.

– The list of $C9$ maximal subgroups of $O^+(12, q)$ of type $2.M_{12}$ has been corrected after noting that they are only maximal for $p = 1, 23 \pmod{24}$.

– One of the imprimitive novelty maximal subgroups of $Sp(4, 4)$ has been removed.

14.7 Finite Soluble Groups

Changes:

– When all `Subgroups` of a pc-group are computed, they are now stored with the group.

– The algorithm for subgroups of pc-groups where there is a low index limit has been improved to eliminate more cases more quickly.

– Some low level arithmetic routines have been changed to increase speed, particularly when evaluating homomorphisms.

Bug Fixes:

– A bug when using sets of pc-groups with universe specified as `PowerGroup(G)` has been fixed. Using this feature greatly improves lookup time when creating a set of subgroups of a fixed group G.

14.8 Finitely Presented Groups

New Features:

- The `ToddCoxeter` function has an extra parameter, `LowerBound`. Enumeration will be considered complete when the coset table has no holes and has `LowerBound` rows. This may avoid tracing all relators at all cosets as the final act of the enumeration. The permutation and matrix group STCS algorithms use this feature.

Changes and Removals:

- The epimorphisms returned by the `SolubleQuotient` and `PCGroup` intrinsics have been extended to support inverse images.

14.9 Databases of Groups

New Features:

- The database of transitive groups has been extended to include all degree 32 transitive groups, as determined by Cannon and Holt. The `TransitiveGroupIdenitfication` routine has been extended to deal with degree 32 groups.
- The database of primitive groups has been extended to include all primitive groups of degree up to 4095, as determined by Coutts, Quick and Roney-Dougal. The groups were supplied to Magma by Hannah Coutts. The `PrimitiveGroupIdenitfication` routine has been extended to deal with degrees up to 4095.

Changes:

- The transitive group database access and identification have been significantly improved, in terms of both speed and memory usage.
- A new database type for the transitive group databases has been added. This provides faster access for multiple groups than the old process approach, the use of which is now recommended against.

15 Lattices

15.1 Lattices

New Features:

- The `IsotropicSubspace` intrinsic (for an indefinite symmetric Gram matrix) has been improved, and for even-dimensional input typically returns a space that is one dimension larger than that given by Simon's original algorithm.
- The algorithm used for testing isometry of lattices has been upgraded to the same methods as now used for computing automorphism groups, i.e. a backtrack search (as Plesken and Souvignier) using ordered partition stack methods (Leon). The optional parameters for the `IsIsometric` function have been changed. The `Depth` and `BacherSCP` parameters are now ignored. The `Stabilizer` parameter, which was ignored, has now been removed. The `Generators` parameter is replaced by `LeftGenerators` and `RightGenerators`, to supply generators of the left and right automorphism groups respectively. There are new parameters `LeftVectors` and `RightVectors` which may be used to override the standard selection algorithm for the lattice vectors that will be permuted as the search progresses. Note that neither or both of the vectors parameters may be set, setting just one will result in a runtime error.
- The routine `ThetaSeriesModularForm` returns the theta series of an integral lattice as a modular form (belonging to the space given by `ThetaSeriesModularFormSpace`). The weight of the modular form is equal to half the lattice dimension, so it can be a form of half-integral weight. The function is useful when the dimension of the modular forms space is quite small; this is frequently the case for "interesting" lattices arising in arithmetic constructions. Once the theta series is identified as a modular form, it becomes feasible to compute many coefficients (far more easily than by short vector enumeration).

Changes and Removals:

- The `HasIsometricEmbedding` intrinsic function has been withdrawn. The function almost never did what the handbook description promised.

16 Lie Theory

16.1 Root Data

Changes and Removals:

- The intrinsics `IsAdjoint` and `IsSimplyConnected` have been changed so that their behaviour corresponds with the standard concepts of "adjoint" and "simply connected" as defined by for example Demazure (*SGA3 Expose XXI*, Def. 6.2.6) and Springer (*Linear Algebraic Groups*, Section 8.1.11). In particular this means that if R is a non-semisimple root datum, then `IsAdjoint(R)` and `IsSimplyConnected(R)` will always return `false`. The previous behaviour of these functions is available as `IsWeaklyAdjoint` and `IsWeaklySimplyConnected`.
- The output of `CartanName` no longer contains any trailing whitespace.

Bug fixes:

- Two bugs in `DirectSum` for toral root data have been fixed. (V2.17-2, V2.17-7)
- A bug in `SimplyConnectedVersion` and `AdjointVersion` has been fixed. (V2.17-9)

16.2 Reflection Groups

Changes and Removals:

- The function `TransversalWds` for Coxeter groups has been sped up. Additionally, it has been renamed to `TransversalWords`. (V2.17-3)

16.3 Lie Algebras

New Features:

- Code is provided which makes it possible to test isomorphism in certain classes of Lie algebra. These include semisimple algebras and low dimensional nilpotient and soluble algebras.
- The robustness and performance of the computation of split toral subalgebras and Chevalley bases have been improved.
- Improvements have been made to the recognition procedure for reductive Lie algebras.
- Code for computing nilpotent orbits in simple Lie algebras and for theta-groups has been developed by Willem de Graaf.
- It is now possible to determine the semisimple subalgebras of a simple Lie algebra using a database constructed by Willem de Graaf.
- A facility is provided for computing with Lie algebras generated by extremal elements. The functionality includes computing bases (either over fields, or a basis of the Lie ring over $\mathbf{Z}$), multiplication tables, and associated varieties.
- The first stage of a project to support computing with Kac-Moody (affine) Lie algebras is included in V2.18. The current machinery includes tools to construct such an algebra and to perform arithmetic with its elements.

Changes and Removals:

- Support for `SplittingCartanSubalgebra`, `IsSplitToralSubalgebra` and `SplitToralSubalgebra` for matrix Lie algebras has been added. (V2.17-8)

Bug Fixes:

- A crash in `LieAlgebra` for a finitely presented Lie algebra has been fixed. (V2.17-2)
- The intrinsic `LieAlgebra` for sets or sequences of elements of finitely presented Lie algebras now returns a map (rather than a function) as fourth argument, improving robustness.
- A bug in the homomorphism returned by the `quo<...>` constructor for finitely presented Lie algebras has been fixed.
- Some small issues in the computation of Chevalley bases have been resolved. (V2.17-3)
- A bug in the map returned by the `quo<...>` constructor for matrix Lie algebras has been fixed. (V2.17-3)
- A bug in `MatrixLieAlgebra` that would cause too few elements to be returned has been fixed. (V2.17-7)

16.4 Groups of Lie Type

Bug Fixes:

- A bug in `Generators` was fixed that would cause an incomplete set of generators to be returned in some special cases. (V2.17-6)
- Argument checking for `GroupOfLieType` has been improved. (V2.17-11)

16.5 Representations

New Features:

- Code has been contributed by Robert Zeier for studying the irreducible simple subalgebras of the Lie algebra $su(k)$. In particular, all of its irreducible simple subalgebras can be determined.
- The intrinsic `WZWFusion` has been added, enabling computation of fusion rules using the Kac-Walton formula.
- The weights and multiplicities of a weight multiset of type `LieRepDec` may now be more easily accessed via the new functions `Multiplicity` and `Multiset`.
- The function `Branch` for weight multisets now takes an optional parameter `Virtual` to allow for "virtual decompositions" of representations.

Changes and Removals:

- The interface to the Littlewood-Richardson tensor procedures for weight multisets has been improved and a number of checks have been added. `LittlewoodRichardsonTensor` now takes either partitions (or partition multisets) and returns a partition multiset; or takes two weight multisets and returns a weight multiset; or it takes a root datum and two highest weights and returns a weight multiset. (V2.17-8)

Bug Fixes:

- Various problems arising in the calculation of the adjoint and standard representations of a Lie algebra or Lie group have been fixed.
- The Kazhdan-Lustzig polynomials are now computed more efficiently and some related procedures have been improved. This applies in particular to `BruhatLessOrEqual`, `BruhatDescendents`, `KLPolynomial` and `RPolynomial`. Moreover, these methods now apply to elements of Coxeter groups presented as permutation groups, finitely presented groups and matrix groups. (V2.17-3)
- A bug in `RestrictionMatrix` has been fixed. (V2.17-3)
- A bug in `HighestWeightRepresentation` has been fixed. (V2.17-4)
- A number of bugs in `RowReductionHomomorphism` and `Inverse` for representations of groups of Lie type have been fixed. (V2.17-6)
- A crash in `SymmetricPower` has been fixed. (V2.17-7)
- A number of memory leaks in the algorithms for calculation with representation decompositions have been eliminated. (V2.17-7)
- A bug in `HighestWeightModule` and `HighestWeightRepresentation`, occurring in certain cases due to incompatibility of root orders, has been fixed. (V2.17-7)

17 Linear Algebra and Module Theory

17.1 Matrices

New Features:

- The algorithm for multiplication of matrices over $\mathrm{GF}(2^k)$ for $k > 4$ has been sped up greatly (fast algorithms based on packed representations have been present for $k \leq 4$ in previous versions). As a consequence of this, several critical linear algebra algorithms for matrices over such fields are now faster (particularly in the case of larger matrices).
- Following the introduction of some new ideas, the p-adic algorithm for computing the nullspace of an integer or rational matrix has been greatly sped up This has led to a speedup of several algorithms for integer or rational matrices.
- The algorithm for computing the Hermite form of an integer matrix has been sped up (independent of the improvements to the nullspace algorithm above). In particular, a better strategy is employed to handle the different classes of input matrices.
- The algorithm to compute the Smith form (or elementary divisors) of an integer matrix has been sped up, as has the related algorithm for computing the saturation of an integer lattice.

18 Linear Associative Algebras

18.1 Quaternion Algebras

New Features:

- The functionality provided for quaternion algebras over $\mathbf{F}_q(x)$ is now the same as for algebras over the rationals or number fields. In particular, it is now possible to construct algebras and orders with given discriminant.
- Intrinsics to compute the Gorenstein closure of a quaternion order has been added.

Bug fixes:

- A bug causing `HilbertSymbol` to return incorrect answers has been fixed (V2.17-10).
- The `IsEichler` intrinsic has been rewritten and should now give correct output.

18.2 Orders of Associative Algebras

Bug Fixes:

- The construction of sets of ideals of orders of associative algebras has been fixed. (V2.17-9)

18.3 Matrix Algebras

Bug Fixes:

- The testing of equality of homomorphisms between algebras which do not inherit from `AlgGen` has been fixed. (V2.17-9)

19 Representation Theory

19.1 Character Theory

New Features:

- It is now possible to conjugate a character of group G by an element of the automorphism group of G.

Changes:

- After experience with gained from working with a number of large and difficult groups, the algorithm used to compute the table of ordinary irreducible characters has been modified to improve performance.