Summary of New Features in Magma V2.23

July 2017

1 Introduction

This document provides a terse summary of the new features released as part of Magma versions V2.23 (July 2017).

A small number of new features were exported in patch releases prior to the main release of V2.23 in July 2017 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 V2.22-x.

Recent releases of Magma were: V2.22 (May 2016), V2.21 (December 2014), V2.20 (December 2013), V2.19 (December 2012), V2.18 (December 2011), V2.17 (December 2010), V2.16 (November 2009), V2.15 (December 2008), V2.14 (October 2007).

2 Highlights

Arithmetic Geometry

- *Hyperelliptic Curves*
  - A new version of the code for height computations of hyperelliptic curves and Jacobians has been provided by Jan-Steffen Mueller. In particular, the `Regulator` and `HeightPairingMatrix` intrinsics now have a direct `UseArakelov` option.
- *Curves Over Finite Fields*
  - Moritz Minzlaff has contributed code for computing zeta functions of *superelliptic* curves of the form $y^a = h(x)$. This employs a p-adic cohomology method, generalizing Kedlaya's algorithm. It is described in Minzlaff's thesis "Frobenius-stable lattices in rigid cohomology of curves" (2013, TU Berlin). The implementation incorporates the $O(\sqrt{p})$ improvement due to Harvey. This means it is the best method currently available in Magma when the characteristic is large relative to the genus.
  - Jan Tuitman has contributed code for zeta functions of general curves. Tuitman's algorithm is a generalization of Kedlaya's algorithm, which allows it to compute the p-adic cohomology of any curve, given a lift of the curve to characteristic zero that satisfies mild p-adic conditions.

  - An implementation of algorithms of Castryck and Tuitman finds a *good* lift for any curve of genus 5 or less. Although there is currently no general algorithm for finding good lifts in higher genus, a trivial lift works in the case of many curves.

- *Hypergeometric Motives*

  - New functionality for Jacobi motives has been provided. Also it is now possible to compute the Bezoutian of a hypergeometric motive. This continues the project of controlling the bad Euler factors at wild primes.

Arithmetic Fields

- *Rational Function Fields*

  - A univariate rational function $f(x)$ defined over a field can be expressed as a composition of rational functions using code contributed by Jonas Szutkoski. More generally, all non-equivalent complete decompositions of $f(x)$ can be found. The decompositions are found using a new algorithm due to Luiz Allem, Juliane Capaverde, Mark van Hoeij and Jonas Szutkoski.

- *Galois Groups*

  - Galois groups of reducible polynomials over $\mathbf{Q}(t)$ can now be constructed.

Basic Rings and Fields

- *The Complex Field*

  - A new implementation of a Newton-based algorithm has been developed for computing roots of univariate polynomials over the complex field $\mathbf{C}$. Compared to the previous implementation, the new implementation is more robust and typically much faster (often by a factor of 10 or more).

Coding Theory

- *Linear Codes over Finite Fields*

  - An update of the tables of best known linear codes (BKLCs) over small finite fields has been provided by Markus Grassl (Max Planck Institute for the Science of Light, Erlangen). Compared ith the previous release of the Magma BKLC database, 1201 missing codes have been added, and the lower bound on the minimum distance has been improved for 5,155 codes. The database now contains a total of 127,278 codes. The tables contain every best known linear code over $GF(2)$ of length up to 256, and all best known linear codes over $GF(4)$ of length up to 100. The tables also contain a high proportion of the best known linear codes for the fields $GF(q)$ up to length n for the following values of q and n: $q = 3, n = 243$; $q = 4, n = 256$; $q = 5, n = 130$; $q = 7, n = 100$; $q = 8, n = 130$; $q = 9, n = 130$.

Combinatorial Theory

- *Graph Theory*

  - The version of the Brendan McKay's nauty package has been updated to V2.5r9. This package has very efficient algorithms for computing the automorphism group of a graph or digraph. It can also produce a canonical labelling for a graph thus solving the isomorphism problem.
  - Adolfo Piperno has also developed a package, known as Traces, for computing automorphism groups and canonical labellings for graphs. In this release of Magma, its version of Traces has been updated to V2.1. The Traces package performs much better for some types of graphs than nauty and overall the two packages complement one another. Magma allows the user to select which one is used in a given application.

- *Hadamard Matrices*

  - The algorithm used in Magma for testing equivalence of Hadamard matrices reduces the problem to one of determining the canonical labelling for a graph. This is then solved using either nauty or Traces and so the Hadamard equivalence intrinsic allows the user to select the method. It was found that when testing equivalence of Hadamard matrices of order 256 Traces was dramatically faster than nauty and so the default Hadamard equivalence test uses Traces.
  - The same holds for the computation of the automorphism groups of Hadamard matrices.

Commutative Algebra

- *Gröbner Bases*
  - The dense F_4 Gröbner basis algorithm has been sped up significantly and memory usage has also been reduced for several types of inputs over finite fields. Based on these improvements, for the first time, a 300-variable HFE cryptosystem over GF(2) with secret degree 100 can be solved using this algorithm. The computation was done on a 3.1GHz Xeon E5-2687W taking about 410,000 seconds (using one core) or in about 47,000 seconds with the same single core and a K40 Tesla GPU, with memory use of 330GB in both cases.
  - The Gröbner Walk order change algorithm has been improved in various ways. The algorithm is generally faster when changing to a lexicographical Gröbner basis with many variables. Changing order for positive-dimensional ideals defined over number fields is also faster now.
  - Major improvements have been made to the primary decomposition algorithm both in the zero- and positive-dimension cases.
  - The main algorithm for intersection of ideals has been greatly sped up for several important classes of ideals. Also the critical algorithm for computing colon ideals has been improved, especially for the case in which the polynomial ring possesses a grading.

- *The Variety of an Ideal*
  - A new algorithm has been developed for determining the variety of a polynomial ideal over the real or complex field. The new algorithm is much faster and more robust than the previous algorithm.

Group Theory

- *Automatic Groups*
  - Given an automatic group A, code has been provided by Derek Holt which attempts to verify that A is word hyperbolic.
- *Classical Groups*
  - The machinery for producing the maximal subgroups of the finite classical groups contains generic Magma code for calculating the eight categories of *geometric* types. In this release Derek Holt has extended the code to produce the maximal subgroups of non-geometric type C9 for all families of classical groups of degree up to 17 (previously the limit was 12). So *all* maximal subgroups will now be returned for degree up to and including 17.
  - Intrinsics are provided for finding the Borel subgroup, extended Weyl group, maximal parabolic subgroups, Siegel unipotent radical, and Siegel parabolic subgroup for a symplectic group. These were developed by Don Taylor.
- *Finitely-Presented Groups*
  - The first installment of a package that attempts to establish that a finitely-presented group F is infinite is included in this release. The code searches for epimorphisms ϕ of F and uses the Holt-Plesken cohomological criterion to test for non-finiteness of the kernel of ϕ. The code is very effective for many classes of groups. For example, it is able to show that each of the groups

    $$\langle x, y | x^2, y^3, (xy)^7, (x, y)^n \rangle$$

    for n = 12, ..., 100, is infinite in a total of 90 minutes.
  - The L_2-quotient algorithm computes, for a 2-generator finitely-presented group, all quotients isomorphic to $PSL(2, q)$ or $PGL(2, q)$. Until now it has been somewhat limited in its range of application. However, recent improvements to the commutative algebra machinery, together with some modifications to Sebastin Jambor's L_2-quotient code has greatly extended the class of groups for which the procedure succeeds. The performance of the L_3-quotient program has also been improved.
  - Given a presentation P for a finitely-presented group G, Magma can now determine the values of the rational numbers i, j and k for which the small cancellation conditions $T(i)$, $C(j)$ and $C'(k)$ are satisfied by P. A group G satisfying sufficiently strong conditions is known to be word-hyperbolic.
  - A related intrinsic attempts to prove, using techniques due to Richard Parker based on the curvature of van Kampen diagrams for G, that the finitely-presented group G is word-hyperbolic. Code for both procedures was written by Derek Holt.

- *Finite Groups*
  - Machinery has been developed by Derek Holt to determine minimal degree permutation representations of the full automorphism groups of finite simple groups.
  - Three specialised subgroup functions have been introduced: list all subgroups of a small group (not just conjugacy class representatives), list the minimal overgroups of a subgroup H of G, and list those subgroups of G which lie between a subgroup H and G.
- *Matrix Groups*
  - For general matrix groups having a base and strong generating set (BSGS), improved algorithms have been introduced for finding a composition series, computing the normalizer of a subgroup, and testing subgroups for conjugacy.
  - The Soluble Radical (SR) approach is the basis for many structure algorithms for permutation groups and is currently being developed for the case of matrix groups where it is necessary to use the composition tree approach rather than BSGS. These algorithms are commonly known as LMG algorithms. Much work has been invested in improving reliability and speeding up the LMG algorithms. However, this approach is still at the experimental stage.
  - If a *good* base can be found then BSGS methods are usually the fastest way of computing with quite large matrix groups. Code written by Derek Holt uses the LMG machinery (see above) to find a satisfactory base if one exists. The procedure is frequently successful and so effectively extends the applicability of BSGS algorithms.

Lattices and Quadratic Forms

- *Lattices*
  - A new package for computations with lattices over number fields has been installed by Markus Kirschmer (Aachen) with contributions from his student David Lorch. This complements the previous Lorentzian package derived from work of Gael Collinet. New functionality included genus computations and local isometry tests (at primes ideals).

Linear Algebra and Module Theory

- *Linear Algebra over Finite Fields*
  - A new algorithm has been developed to compute the transpose of a matrix over GF(2) on an NVIDIA Tesla GPU. This yields non-trivial speedups for several linear algebra routines which require transposing matrices (in particular, the dense F_4 Gröbner basis algorithm).
  - The algorithms for computing rank and determinant of sparse matrices have been improved, in particular, for the case where matrices have easily detected dependencies. As a consequence, a large body of homology computations have been sped up by these improvements.
- *R-Modules*
  - The general algorithm for computing Hom modules or endomorphism rings of A-modules or G-modules has been greatly sped up in general, particularly for modules of large dimension with long composition length. For example, for some typical endomorphism ring computations for modules over GF(2) with dimension over a thousand and having long composition length, the algorithm is now more than 10 times faster than previously.

Representation Theory and Cohomology

- *Characters of Finite Groups*
  - The library of ATLAS group character tables available in Magma has been extended to 487 groups. The stored characters each have their Schur indices stored with them.
- *KG-Modules*
  - The algorithm for computing the irreducible QG-modules, Q the rational field, for a finite group G has been improved in various ways.
  - A database of irreducible QG-modules, Q the rational field, for moderately-sized simple groups G will be exported for the first time in the release. The database now contains most of the irreducible QG-modules of degree up to 1000 for ATLAS groups.
  - New intrinsics are provided to efficiently compute the dimension of first cohomology group of G-modules using sparse matrix techniques. These intrinsics are particularly efficient in special cases such as permutation modules.
- *Cohomology*

– Due to a major improvement in the linear algebra, the computation of the dimension of the first cohomology group $H^1(G, M)$ of a permutation module M for a finitely presented group G is now much faster. It is now practical to compute this dimension for modules of dimension up to 100,000 and beyond. Using a similar approach, fast methods are also available for determining the dimension of $H^1(G, T)$ where T is the exterior square or symmetric square of a permutation module M for G.

3 Language and System Features

New Features:

- A slowdown in the memory manager when there were many free blocks of similar size has been fixed. This has led to major speedups.
- A major slowdown in the structure handling machinery (when very many similar structures of the same type were present) has been fixed. This has led to significant speedups for some computations, in particular in the Large Matrix Groups package.
- New function `GetProfile` which returns whether the profiler is currently turned on.
- A break or continue statement is now allowed from within a try-catch statement. (V2.22-9)

Bug Fixes:

- Errors caused by invoking a `require` statement during a constructor no longer silently disappear. (This could happen, for instance, during set creation when the `eq` intrinsic for the type has a `require` in it.) (V2.22-8)
- A problem with `clear` where identifiers had wrong values has been fixed. (V2.22-9)

4 Aggregates and Mappings

4.1 Mappings

New Features:

- Automorphisms now store their properties of being injective and surjective homomorphisms.
- A map constructed as the composition of two maps storing the property of being homomorphisms now also stores the property of being a homomorphism and similarly for maps constructed from two injective or surjective maps.

5 Algebraic Geometry

5.1 Algebraic Curves

Bug Fixes:

- A crash with `PointSearch` for cubic models has been fixed. (2.22-4)

5.2 Algebraic Surfaces

Bug Fixes:

- A major slowdown in `FormallyResolveProjectiveHypersurface` for some kinds of inputs has been fixed. (V2.22-3)
- A bug has been fixed in the computation of the adjoints of surfaces. (V2.22-3)

6 Arithmetic Geometry

6.1 Elliptic Curves

6.1.1 Elliptic Curves over the Rational Field

Bug Fixes:

- A problem with `GaloisRepresentation` was fixed. (2.22-9)

6.1.2 Elliptic Curves over Number Fields

Bug Fixes:

- A problem with precision in `AnalyticRank` of an elliptic curve over a number field has been remedied. (2.22-6)
- A problem with `AnalyticRank` for elliptic curves over number fields with even degree was fixed. (2.22-7)
- A problem with `EulerFactor` for elliptic curves over number fields with nontrivially intersecting ramification was fixed. This also appeared in EulerFactor for an elliptic curve twisted by an Artin representation, giving a result in a cyclotomic field rather than a complex field. (2.22-7)

6.2 Curves Over Finite Fields

New Features:

- In `ZetaFunction(C)`, automatic algorithm selection now makes better choices between all the available algorithms.
- Minzlaff has contributed code for computing zeta functions of *superelliptic* curves of the form $y^a = h(x)$. This employs a p-adic cohomology method, generalizing Kedlaya's algorithm. It is described in Minzlaff's thesis "Frobenius-stable lattices in rigid cohomology of curves" (2013, TU Berlin). The implementation incorporates the $O(\sqrt{p})$ improvement due to Harvey. This means it is the best method currently available in Magma for large characteristics. Minzlaff's implementation is selected automatically where appropriate, or can be selected by the user by specifying `Al:="Minzlaff"` in `ZetaFunction(C)`.
- Jan Tuitman has contributed code for zeta functions of general curves. Tuitman's algorithm is a generalization of Kedlaya's algorithm, which allows it to compute the p-adic cohomology of any curve, given a lift of the curve to characteristic zero that satisfies mild p-adic conditions. Tuitman's implementation is selected automatically where appropriate, or can be selected by the user by specifying `Al:="Tuitman"` in `ZetaFunction(C)`. The lower level function `ZetaFunction(f, p)` is the more direct way to call Tuitman's algorithm, taking as input a *good* lift to characteristic zero (defined by a polynomial f).
- The intrinsic `GonalityPreservingLift(C)` returns a *good* lift (in the sense of Tuitman's algorithm) for an arbitrary curve C of genus 5 or less, using algorithms of Castryck and Tuitman.

6.3 Hyperelliptic Curves and Jacobians

Bug Fixes:

- A longstanding bug when counting points on genus 1 hyperelliptic curves has been fixed (the twisting parameter was applied twice). (2.22-4)
- Corrections and improvements have been made to the code for computing heights of points on Jacobians of hyperelliptic curves by its author, Steffen Mueller.
- A bug in `ShiodaInvariants` has been fixed. (2.22-9)

6.4 Hypergeometric Motives

New Features:

- The `Bezoutian` of a hypergeometric motive datum is now available. (2.22-8)

Bug Fixes:

- A problem with the computation of the conductor in the identification of the Grossencharacter of a Jacobi motive over $\mathbf{Q}(\zeta_4)$ was fixed. (2.22-7)
- The guesses for wild Euler factors have been arranged to avoid errors. (2.22-8)

6.5 L-Series

Changes:

- The `PHV` (`PrintHodgeVector`) parameter has been added to a number of `HodgeStructure` intrinsics. (2.22-8)

Bug Fixes:

- A bug with the `TateTwist` of a Grossencharacter with nontrivial Dirichlet component has been fixed. (2.22-7)
- An error with tensor product L-functions has been fixed. (2.22-8)
- The L-series of a Hilbert modular form has been corrected in some cases, though some errors with (narrow) class group distinctions still exist. (2.22-8)

7 Arithmetic Fields

7.1 Dirichlet and Hecke Characters

Changes:

- Dirichlet and Hecke character evaluations that are 1 or -1 are now returned in the integers. (2.22-8)
- The `RootNumbers` intrinsic for a Hecke or Grossencharacter now has a vararg `AA` that, if set, returns the result as an associative array. (2.22-8)
- Non-primitive characters over **Q** now include their modulus when printed. (2.22-9)

Bug Fixes:

- Equality check for Grossencharacters over totally real fields has been fixed. (2.22-8)
- A bug with `RootNumber` at a finite place for a Grossencharacter over a totally real field was fixed. (2.22-8)
- A bug in `HeckeCharacterGroup` of an abelian number field was fixed. (2.22-9)

7.2 Algebraic Number Fields

7.2.1 General Number Fields

New Features:

- Standard names for ideals (used in the LMFDB database, and elsewhere) are implemented by `LMFDBLabel, LMFDBIdeal`.
- The operator `!!` can now be applied to integers to map them to ideals of an order of a number field in the same way it can be applied to ideals of an order to coerce them to an ideal of another order. (V2.22-9)

Changes and Removals:

- The parameter `Order` to the intrinsic `Order` taking a sequence of elements has been renamed to `IsBasis` to reflect that it means that the sequence of elements are assumed to be a basis of the resulting order.
- `OptimizedRepresentation` of an order of a number field now returns an order, not necessarily maximal, with the same discriminant as the input order. A bug was also fixed.
- `HilbertSymbol` for number fields and prime ideals over 2 has been sped up considerably. (V2.22-5)

Bug Fixes:

- Two kinds of crashes in the sieve routine within `ClassGroup` have been fixed. (V2.22-6)
- The composition of automorphisms of number fields has been fixed. (V2.22-3)

7.2.2 Quadratic Fields

Changes:

– The process behind coercing an element of a quadratic field into a complex number has changed resulting in substantial speed up.

Bug Fixes:

– A crash in `BiquadraticResidueSymbol` when the result is zero has been fixed.

7.2.3 Cyclotomic Fields

Changes:

– The process behind coercing an element of a cyclotomic field into a complex number has changed resulting in substantial speed up.

– `IsRootOfUnity` has been corrected for the case of cyclotomic fields using the sparse representation. (V2.22-5)

7.3 Characters and Artin Representations

Bug Fixes:

– A precision-based bug when computing local data at bad primes was fixed. (2.22-7)

7.4 Rational Function Fields

New Features:

– The intrinsic `Decomposition` can be applied to a univariate rational function f to construct a sequence of rational functions f_i such that $f = f_r(\ldots(f_1(t)))$. The decomposition code has been contributed by Jonas Szutkoski. Part of the code in this package was originally developed by Peter Fleischmann.

Changes:

– An efficient version of `IsSquare` has been added for elements of rational function fields. (V2.22-6)

7.5 Algebraic Function Fields

New Features:

– It is now possible to construct quotients of orders of function fields by ideals of those orders using the `quo< | >` constructor. The `Modulus` of the quotient can be retrieved. Arithmetic operations `+, -, *, ^` and `/` are available for elements of these quotients as well as testing for equality and `IsZero`, `IsOne`, `IsMinusOne` and `IsUnit`. An `Eltseq` of these elements is provided.

– The operator `!!` can now be applied to polynomials and elements of valuation rings to map them to ideals of an order of a function field in the same way it can be applied to ideals of an order to coerce them to an ideal of another order. (V2.22-9)

Changes and Removals:

– The parameter `Order` to the intrinsic `Order` taking a coefficient ring and a sequence of elements has been renamed to `IsBasis` to reflect that it means that the sequence of elements are assumed to be a basis of the resulting order.

– The algorithm choice for computing a 2-element representation of an ideal has been improved. (V2.22-4)

– Coercion of elements in an exact constant field of a function field into the function field has been fixed for relative extensions. (V2.22-3)

7.6 Galois Groups

New Features:

– It is now possible to compute Galois groups of reducible polynomials over $\mathbf{Q}(t)$ as well as Galois groups of irreducible polynomials over $\mathbf{Q}(t)$ and Galois groups of reducible polynomials over global fields which have been previously available.

– The intrinsic `GaloisProof` can now be applied to `GaloisGroup` computations over $\mathbf{Q}(t)$.

Changes and Removals:

– Improvements have been made to the computation of Galois groups. (V2.22-7)

Bug Fixes:

– A bug in `SolveByRadicals` for non-monic polynomials has been fixed. (V2.22-7)

– An infinite loop has been eliminated in the `GaloisGroup` and `Subfields` computations for an algebraic function field defined as an extension of another algebraic function field by a polynomial with denominators.

7.7 p-adic Rings and their Extensions

Bug Fixes:

– An ancient problem with factorization over the p-adics was fixed. The problem was that an incorrect test was being used to test for squarefreeness of an auxiliary polynomial (the discriminant was used, when this could be 0 to the ambient precision without the polynomial itself having a repeated factor). This would cause occasional infinite loops. (2-22.5)

– A crash involving HenselLift was changed to give a runtime error. (2.22-7)

– A bug with coercion sometimes failing for elements between two p-adic quotient rings with the same (large) p was fixed. (2.22-9)

– A problem with NormEquation of a low-precision element in a higher precision ring was fixed. (2.22-10)

7.8 Series Rings

New Features:

– The `hom` constructor is now supported for Laurent series rings.

8 Basic Rings and Fields

8.1 Finite Fields

New Features:

– The new procedure `SetZechLimit(L)` is now available, which allows one to set the limit L so that the Zech representation will be used for any non-prime finite field with size at most L (default is 2^{20} and max is 2^{30}). The corresponding function `GetZechLimit()` returns the current limit. Note that for a finite field of size q with the Zech representation, tables of size $8q$ bytes are needed, so creating fields of larger size will increase the memory used.

Bug Fixes:

– A crash when trying to compute a logarithm in a large finite field for which the data is not available has been fixed. (V2.22-4)

8.2 Polynomial Rings

New Features:

– A new implementation of a Newton-based algorithm has been developed for computing roots of univariate polynomials over the complex field $\mathbf{C}$. Compared to the previous implementation, the new implementation is more robust and typically much faster (often by a factor of 10 or more). (The old algorithm can be selected by setting the parameter `Al` to `"Schonhage"`.) Repeated roots are now handled rigorously and furthermore, if $f \in \mathbf{Z}[x]$ or $f \in \mathbf{Q}[x]$ and a complex root r of f is real, then the complex root which is returned for r will now always have an exact zero as its imaginary part.

– The algorithm for computing roots of polynomials over the real field has been improved. A more accurate answer is returned when a root is very close to zero.

– The algorithms to test squareness of both univariate and multivariate polynomials have been significantly sped up.

– The algorithm for GCD of polynomials over number fields has been sped up by use of LLL reconstruction methods.

– The algorithm for evaluation of a multivariate polynomial f at a sequence of points in a general ring S has been greatly sped up in two ways: (1) by caching evaluated powers of variables; (2) by faster handling of the case in which the polynomial has a denominator. This improves several types of computations in commutative algebra and algebraic geometry.

– The multivariate polynomial factorisation algorithm has been improved for homogeneous inputs. In particular, when the input is a homogeneous bivariate polynomial, the computation has been dramatically sped up.

– Factorisation of polynomials now works fully for fields of fractions of affine algebras over rational function fields.

Bug Fixes:

– A crash in testing whether a multivariate polynomial is square has been fixed.

– A crash in factoring polynomials over polynomial quotient rings over finite fields (which are themselves finite fields) has been fixed. (V2.22-4)

– A crash in multivariate factorization over the integer ring has been fixed. (V2.22-5)

– The previously omitted function `SquarefreeFactorisation` has been added as a synonym for `SquarefreeFactorization`. (V2.22-9)

– A crash in multivariate resultant over the integers has been fixed. (V2.22-9)

– A crash in polynomial multiplication of very high-degree multivariate polynomials has been fixed. (V2.22-10)

– A potential crash in `SmallRoots` has been fixed. (V2.22-10)

– Incorrect scaling of exponents in the output of bivariate polynomial for certain cases has been fixed. (V2.22-10)

– A crash in polynomial multiplication of very high-degree multivariate polynomials has been fixed. (V2.22-10)

9 Coding Theory

9.1 General Linear Codes

Bug Fixes:

– `IsQuasiCyclic(C, d)` now gives an error instead of crashing if the degree is zero. An error is also given if the degree is equal to the length of the code. (2.22-6)

9.2 Linear Codes over Finite Fields

New Features:

– The tables of best known linear codes (BKLCs) over small finite fields have been updated by Markus Grassl. Some 1201 missing codes have been added, and the lower bound on the minimum distance has been improved for 5,155 codes. These new tables can be accessed using the usual intrinsics, for example, the intrinsic `BKLC`.

Changes and Removals:

– The intrinsics used to access the tables of best known linear codes expect that the finite field argument is the *standard* finite field, that is the field defined by `GF`(q). If a different field is given then the results are unpredictable. In this release this argument is checked and an error is raised if a non-standard field is given.

10 Combinatorial Theory

10.1 Graph Theory

New Features:

- The Magma version of Brendan McKay's nauty package for computing automorphism groups and canonical labellings of graphs has been updated to V2.5r9.
- Adolfo Piperno's package Traces (V2.1) for computing automorphism groups and canonical labellings of graphs is now available in Magma. It can be accessed via the intrinsics `AutomorphismGroup`, `CanonicalGraph`, `EdgeGroup` and `IsIsomorphic` by setting the parameter `Al` to `Traces` on any of these intrinsics.

10.2 Hadamard Matrices

New Features:

- The intrinsic `HadamardCanonicalForm`, which produces a canonical form for Hadamard matrices with respect to equivalence classes, reduces the problem to one of determining the canonical labelling for a graph. The Traces algorithms of Adolfo Piperno are much faster for Hadamard matrices of larger degree and is now the default method for Hadamard equivalence. The user may select between nauty and Traces by setting the parameter `Al`.
- Likewise the intrinsic `IsHadamardEquivalent`$(H1, H2)$, which uses the canonical form to determine if Hadamard matrices $H1$ and $H2$ are equivalent, also allows the user to specify which graph canonical labelling method is to be used via the parameter `Al`,
- The intrinsic `HadamardAutomorphismGroup` is often faster for larger matrices if Traces is used and so it is now the default for this intrinsic. Again the user can choose between the three methods, Leon, nauty and Traces by setting the parameter `Al`.

10.3 Difference Sets

Changes:

- The intrinsic `IsDifferenceSet` which tests whether a set of group elements belonging to group G is a difference set for G has been significantly sped up, especially when G is a soluble group defined by a PC-presentation.

11 Commutative Algebra

11.1 Ideal Theory and Gröbner Bases

New Features:

- A new algorithm for computing the variety of a polynomial ring ideal over the real field or complex field which is much faster and more robust than the previous algorithm.
- The dense F_4 Gröbner basis algorithm has been sped up significantly and memory usage has also reduced for several types of inputs over finite fields. Based on these improvements, for the first time, a 300-variable HFE cryptosystem over GF(2) with secret degree 100 can be solved with this algorithm on a 3.1GHz Xeon E5-2687W in about 410,000 seconds (using one core) or in about 47,000 seconds with the same single core and a K40 Tesla GPU, with 330GB used in both cases.
- The Gröbner basis machinery now does a non-trivial search to attempt to detect whether an input ideal is already a GB w.r.t. some monomial order, and if so, a GB order change algorithm is then used (this yields a massive speedup in several important classes of inputs).
- The algorithm for computing normal forms or membership testing of polynomials w.r.t. an ideal has been greatly improved in the case that an input polynomial has high degree.
- Membership testing for multivariate polynomial ideals has been improved in the case that the input polynomial has high degree. This particularly improves the case when the Gröbner basis of an ideal has `grevlex` order but the polynomial tested for membership comes from ideals defined with the `lex` order.
- The main algorithm for intersection of ideals has been greatly sped up for several important classes of ideals.
- The critical algorithm for computing colon ideals has been improved, especially for the case in which the polynomial ring possesses a grading.
- The primary decomposition algorithm has had some major improvements, both in the zero- and positive-dimension cases.
- The Gröbner Walk order change algorithm has been improved in various ways. The algorithm is generally faster when changing to a lexicographical Gröbner basis with many variables. Order change of positive-dimensional ideals defined over number fields is now faster too.

Bug Fixes:

- An error when computing a truncated-degree Gröbner basis over a finite field of size 2^k ($k > 1$) has been fixed. (V2.22-2)
- A problem with linear factors in reducible polynomials has been fixed. (V2.22-4)
- Crashes in Gröbner basis computation over the rational field have been fixed. (V2.22-4)
- A crash in the Gröbner basis machinery when computing whether a curve is singular has been fixed. (V2.22-5)
- A bug in the `FGLM` parameter handling for the function `GroebnerBasis` has been fixed. (V2.22-9)

11.2 Affine Algebras

New Features:

- Various operations for fields of fractions of affine algebras have been sped up. (V2.22-4)

Bug Fixes:

- A crash with fields of fractions of affine algebras was fixed. (V2.22-4)

12 Groups

12.1 Automatic Groups

New Features:

– The intrinsic `IsHyperbolic`(A) attempts to prove that the automatic group A is word hyperbolic. If it is unable to show that the group is hyperbolic, no conclusion can be drawn.

Bug Fixes:

– The Magma level printing of an automatic group has been fixed to record the word differences, which were previously omitted. Should a user need to compute word differences for an automatic group written out previously, the `ReconstructWordDifferences` function can be used.

12.2 Classical Groups

A number of new intrinsics are available for symplectic groups. In the following descriptions, F is a finite field and G is the symplectic group $Sp(n, q)$ or $Sp(n, F)$.

New Features:

– New intrinsics `BorelSp`(n, q) and `BorelSp`(n, F): In the symplectic group G a Borel subgroup is the normaliser of a Sylow p-subgroup, where p is the characteristic of G. Alternatively a Borel subgroup can be obtained as follows:

```
G := Sp(n,q);
P := ClassicalSylow(G,p);
N := ClassicalSylowNormaliser(G,P);
```

But in this case the group N is not guaranteed to be a subgroup of the maximal parabolic subgroups or the Siegel parabolic subgroup returned by the following intrinsics.

– New intrinsics `ExtendedWeylGroupSp`(n, q) and `ExtendedWeylGroupSp`(n, F): If T is the standard torus in the symplectic groups G (i.e., the subgroup of diagonal matrices), the extended Weyl group is a subgroup E of G that normalises T such that $ET/T \simeq W$, the Weyl group of G.

– New intrinsics `MaximalParabolicsSp`(n, q) and `MaximalParabolicsSp`(n, F): The m maximal parabolic subgroups of G that contain the standard Borel subgroup, where $m = n/2$. They are obtained by adjoining $m - 1$ of the m generators of the extended Weyl group to the Borel subgroup.

– New intrinsics `SiegelParabolic`(n, q) and `SiegelParabolic`(n, F): A Siegel parabolic subgroup of a symplectic group is the stabiliser of maximal isotropic subspace of the underlying symplectic geometry.

– New intrinsics `SiegelUnipotentRadical`(n, q) and `SiegelUnipotentRadical`(n, F): The unipotent radical of the standard Siegel parabolic subgroup.

Changes:

– The intrinsic `ClassicalMaximals` has been upgraded by Derek Holt to produce the maximal subgroups of the non-geometric type C9 for all families of classical groups of degree up to 17. Previously, the C9 maximals were only available up to degree 12.

Bug Fixes:

– A number of bugs in the intrinsic `ClassicalSylow` which efficiently computes the Sylow p-subgroups of a classical group have been fixed by Derek Holt. These bugs were in the original code and so some of the functionality has had to be performed using much slower generic code. So for the first time users can now take full advantage of the fast classical Sylow subgroup algorithms. Other intrinsics that may be affected are `ClassicalSylowConjugation` and `ClassicalSylowNormaliser`.

12.3 Finite Groups

New Features:

– The new intrinsic `AutomorphismGroupSimpleGroup`, developed by Derek Holt, returns a permutation representation of the full automorphism group of a simple group. Where possible the intrinsic returns a permutation representation of minimal degree.

– Three specialised subgroup functions have been introduced:

`AllSubgroups`: list all subgroups of a small group (not just conjugacy class representatives);

`MinimalOvergroups`(G, H): list conjugacy class representatives of the minimal overgroups of a subgroup H of G;

`IntermediateSubgroups`(G, H): list conjugacy class representatives of those subgroups of G which lie between H and G.

Changes:

– The `Subgroups` function has been improved in efficiency for groups having a non-trivial soluble radical.

12.4 Finitely-Presented Groups

New Features:

– A new intrinsic `IsInfiniteFPGroup`(F), attempts to establish that a finitely-presented group F is infinite. The code searches for epimorphisms ϕ of F and uses the Holt-Plesken cohomological criterion to test for non-finiteness of the kernel of ϕ. This version of the intrinsic works best on groups F which are perfect or which have small abelian quotients. The application of the Holt-Plesken theorem to an epimorphism ϕ requires a knowledge of the QH modules, where H is the image of ϕ. A database containing such modules for ATLAS groups of degree up to 1000 is available as an optional download. The user of `IsInfiniteFPGroup` is strongly advised to have this database available as it can save a great deal of runtime. The intrinsic has parameters `L2Quot`, `SimQuot`, `LISub` and `LINSub` that allow the user to control the type of epimorphisms to be used.

– The effectiveness of the L2-quotient intrinsic `L2Quotient` has been vastly improved. A new parameter, `SingleInfinite`, for the intrinsic causes the code to return immediately an infinite quotient or quotient family is found.

– Given a finitely-presented group G with presentation P, the intrinsic `SmallCancellationConditions` determines the values i, j, k' for which the small cancellation conditions $T(i)$, $C(j)$ and $C'(k)$ are satisfied by P.

– The intrinsic `RSym`(F) attempts to prove that the finitely-presented group F is word-hyperbolic using a approach due to Richard Parker based on the curvature of van Kampen diagrams for F. Both this code and the code for determining small cancellation conditions were implemented by Derek Holt.

Changes:

– The intrinsic `DicyclicGroup` now returns a group of type `GrpGPC`. This form of the group makes more functionality available for working with the group.

– The `SimpleQuotients` function now has a parameter setting the search for permutation quotients by degree of the permutation group, rather than by order.

– The parameters for `L2Quotient` and `L3Quotient` have been renamed (to have initial capital letter) to be consistent with the rest of Magma

Bug Fixes:

– Certain internal data structures have been updated thereby eliminating various crashes.

– Bugs in the `Simplify` function, in particular when the `Iterations` parameter is set, have been fixed. The function now respects the value given to the iterations parameter in all cases.

12.5 Matrix Groups Over Finite Fields

New Features:

– An LMG function `LMGBase` has been added to find a good base for a matrix group over a finite field. Assigning this base to the group will enable fast matrix group computations (compared to other LMG functions).

– The algorithm for computing the soluble radical of a matrix group over a finite field may now call LMG code when the original BSGS methods fail. This will reduce the number of failures in this fundamental structural algorithm.

– The *Composition Tree* (CT) project, headed by Eamonn O'Brien and Charles Leedham-Green, has developed an alternative to the BSGS method for computing with large matrix groups defined over finite fields. An update of the Composition Tree (CT) package has been provided by Eamonn O'Brien. The main difference between this version and the previous version exported in Magma is the elimination of a number of bugs.

– This release includes new LMG algorithms for finding a base and for defining a homomorphism. Major improvements have been made to both LMG machinery and some of the underlying Magma kernel machinery. This results in much faster execution times for some classes of long running jobs.

– The intrinsic `IrreducibleSubgroups` which produces the conjugacy classes of irreducible subgroups of the linear groups $GL(2, q)$ and $GL(3, q)$ has been extended by Barry Hurley (Auckland) to include the soluble subgroups of $GL(3, q)$. Thus, the lists of irreducible subgroups are now complete for characteristics equal to or greater than 5.

12.6 Matrix Groups over Infinite Fields

Changes:

- A refinement has been made to the bound used for deciding finiteness.

Bug Fixes:

- A bug causing a crash which occurs for certain integral matrix group input to `IsSolubleByFinite` has been fixed.

12.7 Permutation Groups

New Features:

- The new intrinsic `AffineSplitExtension` constructs a permutation group P from a G-module M. The group P has a regular normal subgroup isomorphic to M and P/M is isomorphic to the action of G on M.
- The function `AutomorphismGroupSimpleGroup` returns a permutation representation of the full automorphism group of the named simple group.

Changes:

- When computing the derived series and soluble residual of a primitive non-affine permutation group, group orders are considered to avoid verification of the soluble residual. Such verification can be time-consuming, and the O'Nan-Scott classification may render verification unnecessary.

Bug Fixes:

- Some crashes caused by `RandomSchreier` followed by `Verify`, when the random Schreier was incomplete, have been fixed. In general however an incomplete random Schreier can cause problems that Magma cannot trap or fix. Be aware of potential crashes when using random Schreier.

12.8 Databases of Groups

Changes:

- The `SmallGroup` function for groups of order p^7 has been changed to speed-up getting a single group from the list.

13 Lattices and Quadratic Forms

13.1 Lattices

New Features:

- A new package for computations with lattices over number fields has been installed by Markus Kirschmer (Aachen) with contributions from his student David Lorch. This complements the previous Lorentzian package derived from work of Gael Collinet. New functionality included genus computations and local isometry tests (at primes ideals).

  The main new intrinsics are `GenusRepresentatives`, `Neighbours` Kand `IteratedNeighbours` which allow easy access to the genus information, while `JordanDecomposition`, `IsLocallyIsometric`, and `GenusSymbol` (at prime ideals) allow more direct access to the underlying information.
- The `NaturalAction` parameter for `IsIsometric` of totally definite number field lattices has had its restriction about ambient inner products removed. (2.22-5)
- The `WittInvariants` and `HasseMinkowskiInvariants` now have a parameter `AA` which, when set, returns the result as an associative array. (2.22-8)

Bug Fixes:

- A problem with crashes in lattices with standard basis and inner product has been remedied. This often showed up in number field computations. (2.22-3)
- A problem with verbose printing has been fixed. (2.22-3)
- A problem with BKZ in high dimension has been fixed. (2.22-3)
- A crash with BKZ when the number of columns exceeded the rank was fixed. (2.22-4)
- A bug in the `AutomorphismGroup` function has been fixed. (2.22-5)
- Bugs arising from computing `Coordinates` for vectors in lattices over a real field have been fixed. (2.22-7)
- The Hash of a lattice is no longer computed from the user basis, as there is no canonical form that is easy to compute. This fixes bugs with sets of lattices and equality-checking. (2.22-8)
- The eq operator for lattices of different ranks now returns `false`, rather than an incompatibility error. (2.22-8)
- A bug with internal precision in BKZ was remedied. (2.22-9)
- A problem with the lengths/norms and number of `ShortVectors` was fixed. (2.22-10)

14 Lie Theory

14.1 Reflection Groups

Changes and Removals:

- The complex Cartan matrices and associated root data for the primitive unitary reflection groups have been revised to ensure that the generators returned by the intrinsic `ShephardTodd`(n) satisfy the relations encoded by the Broué–Malle–Rouquier diagrams.

Bug Fixes:

- A bug has been fixed in the the code for the intrinsic `HighestCoroot`.

15 Linear Algebra and Module Theory

15.1 Matrices

New Features:

- A new algorithm has been developed to compute the transpose of a matrix over GF(2) on an NVIDIA Tesla GPU. This yields non-trivial speedups for several linear algebra routines which require transposing matrices (in particular, the dense F_4 Gröbner basis algorithm).
- The algorithms for computing rank and determinant of sparse matrices have been improved, in particular, for the case where matrices have easily detected dependencies. In particular, a large body of homology computations have been sped up by these improvements.

Bug Fixes:

- An occasional crash in echelonisation of matrices over medium prime finite fields has been fixed. (V2.22-2)
- Some superfluous printing in the sparse Hermite normal form algorithm has been fixed. (V2.22-3)
- The function `Submatrix(A, I, J)` where I and J are sequences has been fixed for the case in which there are zero rows or columns in the result. (V2.22-3)

15.2 Sparse Matrices

New Features:

- The intrinsic `SparseMonomialMatrix` creates a sparse monomial matrix from a permutation.

15.3 R-Modules

New Features:

- The general algorithm for computing Hom modules or endomorphism rings of A-modules or G-modules has been greatly sped up in general, particularly for modules of large dimension with long composition length. For example, for some typical endomorphism ring computations for modules over GF(2) with dimension over a thousand and having long composition length, the algorithm is now more than 10 times faster than previously.

Bug Fixes:

- A hang in computing an endomorphism ring of a general A-module in characteristic zero (the result of a maximal order computation) has been fixed. (V2.22-2)

15.4 Numerical Linear Algebra

Bug Fixes:

- A problem with images of complex matrices was fixed. (2.22-4)

16 Linear Associative Algebras

16.1 Associative Algebras

New Features:

- It is now possible to create sets from collections of orders of associative algebras.

17 Representation Theory and Cohomology

17.1 Character Theory

New Features:

– The function `SchurIndicesBounds` has been added for the case in which a group's character table is defined, but no group is defined. This will determine bounds on the p-adic Schur indices of the character using information from the full character table.

Changes:

– The library of ATLAS group character tables available in Magma has been extended to 487 groups. The Schur index of each characters is stored with it.

Bug Fixes:

– The group theory algorithms used by `LMGCharacterTable` were re-organised to make proper use of the LMG functionality. This avoids crashes when BSGS functions are used on a matrix group where LMG is needed.

17.2 $K[G]$-Modules

New Features:

– An optional database of irreducible rational representations may be downloaded from the database download webpage (tar file `RepRat.tar.gz`). The following intrinsic functions `IrreducibleModules` and `DBIrreducibleQGModules` use the database when it is present.

– The algorithm for computing the irreducible QG-modules, Q the rational field, for a finite group G has been improved in various ways. The corresponding intrinsic is `IrreducibleModules`(G, Q).

– The intrinsic `DBIrreducibleQGModules`(A), where A is a string giving the ATLAS name of a near-simple group G returns the irreducible QG-modules, Q the rational field, for moderate sized groups G. The intrinsic `DBHasIrreducibleModules`(A) returns true if the rational irreducibles of the group with ATLAS name A are stored in the database. The database contains most of the irreducible QG-modules of degree up to 1000 for ATLAS groups.

17.3 Cohomology

New Features:

– Let F be a finitely presented group, let ϕ be an epimorphism of F onto the finite group H and let HM be a permutation module of H over the ring R that can be lifted to an F-module FM. The intrinsic `H1Dimension`(F, ϕ, R) determines the dimension of the first cohomology group $H^1(F, FM)$. Similar intrinsics are provided for the exterior square and symmetric square of the module FM.

18 System

Bug fixes:

- Formal sets no longer crash when using a predicate which is a single function call that has multiple return values, such as `IsSquare`. (2.22-6)