Summary of New Features in Magma V2.14

October 2007

1 Introduction

This document provides a terse summary of the new features installed in Magma for release version V2.14 (October 2007).

Previous releases of Magma were: V2.13 (July 2006), V2.12 (June 2005), V2.11 (May 2004), V2.10 (April 2003), V2.9 (May 2002), V2.8 (July 2001), V2.7 (June 2000), V2.6 (November 1999), V2.5 (July 1999), V2.4 (December 1998), V2.3 (January 1998), V2.2 (April 1997), V2.1 (October 1996), V2.01 (June 1996) and V1.3 (March 1996).

2 Summary

The Integers

- *Primality Proving:*
    - The original version of the Elliptic Curve Primality Prover (ECPP) of F. Morain has been upgraded. In particular, it now uses Morain's 1998 tables of Weber polynomials which enable the primality of much larger integers to be established.

Finite Fields

- *Arithmetic*
    - Extension fields are now defined by sparse lexicographically minimal polynomials when possible (and when Conway polynomials are not available), leading to big speedups of arithmetic in moderate to high degree extension fields. Databases of such polynomials have been constructed for characteristic 2 up to degree $120,000$; for characteristic 3 up to $50,000$; for characteristics 5 and 7 up to degree $20,000$; and for characteristics p, $11 \leq p \leq 101$, up to degrees at least $1,500$.
    - A new packed representation for finite fields of characteristic 3 has been introduced giving large speedups for fields of higher degree, in particular. In addition, a fast irreducibility test for polynomials over GF(3) has been devised. This algorithm can run 10 times faster in the case of sparse polynomials. Factoring a polynomial over GF(3^{103}) is now 10 to 15 times faster than for V2.13. (The speed-up comes from the more efficient field arithmetic alone, not improvements in factorization.)

  - Fast Frobenius maps, based on linear algebra.

- *Factorisation*

  - The Magma implementation of the Berlekamp algorithm now uses a sparse matrix datastructure when the polynomial is sparse leading to less memory usage and reduced execution times, particularly over GF(2). Sparse polynomials of degree $100,000$ over GF(2) can be factored in a few seconds.
  - For polynomials whose coefficients lie in a subfield, factorisation and root finding have been sped up enormously.

- *Isomorphism/Embedding*

  - Magma now uses an algorithm due to Eric Rains for constructing isomorphisms between fields and embedding subfields in larger fields. For example, embedding GF(2^{1000}) in GF(2^{2000}) now takes 0.3 seconds, compared to 20 minutes with V2.13.

- *Norm/Trace/Hilbert90*

  - A new deterministic algorithm to solve norm equations in finite fields of relative degree 2 has been added. It is usually faster than the randomised standard algorithm.
  - A fast deterministic algorithm for Hilbert-90 equations over finite fields has been added. Given a finite field F of order q and an element $a \in F$, the Hilbert 90 equation $x^q x^{-1} = a$ can be solved for x in some extension of F. This is the fundamental result in the Galois cohomology of finite fields.

Polynomial Rings

- *Arithmetic*

  - Multiplication of polynomials over finite fields of characteristic 2 has been greatly improved. For polynomials over GF(2) this is achieved through use of the Cantor algorithm which has better complexity than the Karatsuba algorithm. For polynomials over high degree fields GF(2^n), $n > 1$, the Kronecker segmentation expansion method is now used. Multiplication of degree 1000 polynomials over GF(2^{1000}) is now about 10 times faster than in V2.13 on 64-bit processors.
  - The multivariate GCD algorithm has been extended to take advantage of the important case in which the quotient of one of the inputs by the GCD is a low-degree polynomial.

- *Factorisation*

  - Coppersmith's method for finding small roots of univariate polynomials modulo an integer has been implemented. This implementation uses the new fpLLL package of Damien Stehlé.
  - Factorisation of univariate polynomials over small finite fields has been completely overhauled, leading to very significant speedups. This is particularly significant for fields of characteristic 2 and for very sparse polynomials over small fields of any characteristic.
  - Separate to the previous item, testing irreducibility of polynomials over finite fields has been greatly improved through use of a sieving method.
  - In V2.14, for the first time factorisation of polynomials over a class of power series rings is supported. This was achieved by extending S. Pauli's p-adic factorisation method, which is actually an algorithm for the factorisation of polynomials over local fields.

Linear Algebra

- *Matrix Arithmetic*
  - Matrix multiplication has been greatly improved in the case that one of the input matrices is sparse and the other dense.
  - The multiplication of matrices over prime finite fields has been optimised by precomputing the inverse of the modulus.
  - The code for the multiplication of matrices over GF(2) now uses Intel SSE2 instructions when supported.
- *Linear Algebra over Finite Fields*
  - General speed-ups have been achieved for linear algebra over finite fields of characteristic 2.
- *Linear Algebra over Euclidean Rings*
  - The calculation of the Smith normal form and the determinant of a dense matrix defined over an euclidean ring has been significantly improved through replacing the Havas-Holt-Rees algorithm formerly used by an asymptotically-fast recursive echelonisation algorithm.

Lattices and Quadratic Forms

- *Lattices*
  - Simon's variant of LLL reduction for indefinite forms has been efficiently implemented in Magma as part of a new package of Damien Stehlé, and works readily in dimensions greater than 100.
  - The automorphism group and theta series of a lattice is now stored and can be asserted via attributes.
  - A new function ThetaSeriesLimited(L, n) takes a time limit and returns the contribution to the first n coefficients of the theta series of a lattice L found by lattice enumeration within the specified CPU time.
- *Quadratic Forms*
  - Given a quadratic form F in an arbitrary number of variables, Mark Watkins has used Denis Simon's ideas as the basis of an algorithm for finding a large (totally) isotropic subspace of F. The subspace found is frequently maximal isotropic.

Global Arithmetic Fields

- *Number Fields*
  - The fast algorithm of Bosma and Stevenhagen for computing the 2-part of the ideal class group of a quadratic field has been implemented.
  - It is now possible to compute the integral closure of more general rings, such as the closure of $Z[x]$ (instead of the more customary $Q(x)$). This is an example of recognising the existence of non-trivial valuations in the base field. Applications of this technique arise in areas such as inverse Galois theory (over Q) and the computation of minimal models for schemes.
  - The unit group of a suborder is canonically a subgroup of the full unit group of the maximal order, typically having very large index. In V2.14 we include code that implements a very efficient approach to computing this subgroup.
  - A new algorithm of C. Fieker and W. de Graaf has been implemented which finds the Z-lattice of all dependencies that exist between the roots of a polynomial or even a set of arbitrary algebraic integers.
  - A routine is provided for finding a simple representative modulo nth powers of a number field element.
- *Galois Theory*
  - Prior to this release, the calculation of Galois groups of polynomials defined over number fields has been limited to polynomials of degree at most 23. In this release, the new Fieker-Klüners algorithm has been extended so as to apply to polynomials defined over absolute extensions of **Q**. This allows the computation of Galois groups for polynomials of arbitrary large degrees (at least in theory) defined over number fields.
  - In a previous release, machinery was provided for the computation of arbitrary subfields of the normal closure based on the use of the Galois correspondence. In V2.14, the techniques are extended to allow computation of towers of relative extensions of number fields corresponding to descending chains of subgroups. This has applications to such problems as the solvability of equations by radicals and the computation of splitting fields.
  - One of the most celebrated results in mathematics states that an equation is soluble by means of radicals if and only if its Galois group is soluble. A constructive version of this theorem has been installed in Magma. More precisely, given a polynomial over Q with solvable Galois group, we find a representation of its roots in a radical tower.
  - Given a polynomial f over the integers, code has been developed which exploits Magma's ability to find Galois groups in order to efficiently compute the splitting field of f (and representations of its roots).

- *Class Field Theory*
  - Drinfeld modules of rank 1 may be viewed as realising the CM-theory for global function fields. In particular, they can be used to find explicit defining equations for abelian extensions. In V2.14 a new algorithm is included that is capable of finding explicit algebraic descriptions of images under this Drinfeld module for arbitrary fields (in principle). Previous methods applied only to elliptic or hyperelliptic curves.
  - A celebrated theorem in class field theory due to Grunwald and Wang asserts the existence of a global (cyclic) field with given local degrees. Theoretical applications arise, for example, in the theory of algebras where the theorem guarantees the existence of a minimal degree splitting field satisfying given local conditions. A constructive version of the theorem has been implemented: given (finitely many) local degrees, it produces a cyclic number field interpolating the given local data.
- *Galois Cohomology*
  - It is important to be able to recognise whether an element of the second cohomology group of the Galois group of a number field acting on the multiplicative group is trivial (ie. an element of the first cohomology group). (It can be thought of as a generalisation of a norm equation.) Applications occur in the theory of central simple algebras (the relative Brauer group of a field) and in representation theory. In V2.14 a new algorithm of C. Fieker determines whether a given 2-cochain is trivial, and if so, finds a corresponding 1-cochain.

Local Arithmetic Fields

- *Unramified Fields*
  - The Frobenius map (GaloisImage) for unramified extensions of Q_p with default bases has been rewritten for increased speed.
  - An implementation of an algorithm of Harley for efficient Teichmüller lifts in unramified extensions of Q_p has been added.

Algebras

- *Matrix Algebras*
  - A very efficient algorithm for computing the unit group and Jacobson radical of a matrix algebra defined over a finite field has been developed and implemented by P. Brooksbank and E. O'Brien.
- *Quaternion Algebras*
  - For definite quaternion algebras over number fields, the conjugacy classes of maximal orders, and the 2-sided ideal class group of a maximal order can now be computed. A much faster routine for determining unit groups of orders is provided. An alternative algorithm for the left or right ideal classes, that makes use of the other new features, has also been implemented. This machinery was developed by Markus Kirschmer.
- *Reductive Lie Algebras*
  - The construction of twisted reductive Lie algebras is now supported. This makes it possible allows us to construct a wider range of Lie algebras over non-algebraically closed fields. For example, over a real field, this allows the construction of the unitary Lie algebra.
  - It is now possible to compute standard bases for reductive Lie algebras over finite fields.
- *Nilpotent Lie Algebras*
  - A database of all nilpotent Lie algebras of dimension up to 6 over fields of odd characteristic has been implemented by Willem de Graaf. Given any such algebra it is possible to identify it in the database.
- *Algebras over Euclidean rings*
  - One may now create arbitrary quotients of finite dimensional algebras given in terms of structure constants which are defined over euclidean rings (including rings with zero divisors, such as residue class rings). The quotient algebras may have both free and torsion parts. Only free algebras and their subalgebras over such rings were previously supported.
  - As a consequence of the previous item, Lie rings (that is, Lie algebras over defined over an euclidean ring, usually $\mathbf{Z}$) are now supported. In particular, it is possible to compute a basis and multiplication table for finitely presented Lie rings having finite dimension. The implementation handles both homogeneous and nonhomogeneous relations. A variant of the algorithm is provided which can find nilpotent quotients of finitely presented Lie rings. This implementation was undertaken by W. de Graaf.

Finite Groups

- *Arithmetic*
  - A variation of the Product Replacement Algorithm for generating random elements of a group due to H. Bäärnhielm and C. Leedham-Green has been installed. The new algorithm, known as Prospector, is designed so as to minimise the length of the words.
  - The evaluation of straight line programs (SLPs) in a group has been revised following ideas of H. Bäärnhielm to retain fewer intermediate results in memory during the evaluation. This can greatly reduce memory use, for example, when evaluating homomorphisms. The new scheme is particularly effective when evaluating SLPs produced by the product replacement algorithm.
- *Monte-Carlo Structure Algorithms*
  - A number of Monte-Carlo algorithms for investigating the structure of large groups have been implemented by E. O'Brien and others. Perhaps the most important of these algorithms is an implementation of the well-known algorithm of J. Bray for computing the centraliser of an involution.
- *Maximal Subgroups*
  - The maximal subgroups of the classical groups of dimension not exceeding 12 have been determined by J. Bray, D. Holt and C. Roney-Dougal and the corresponding Magma code has been written by D. Holt and C. Roney-Dougal. The maximal subgroups are given in the natural representation of the given classical group.
  - The maximal subgroups of the twisted groups ${}^2B_2(q)$ (more commonly known as the Suzuki groups $Sz(q)$) and the maximal subgroups of the twisted exceptional groups ${}^2G_2(q)$ (small Ree groups) have been determined by Henrik Bäärnhielm whose Magma implementation is included in V2.14. Again, the maximal subgroups are produced in the natural matrix representation.
- *Sylow Subgroups*
  - The Sylow subgroups of the family of exceptional groups ${}^2F_4(q)$ (large Ree groups) have been determined by Henrik Bäärnhielm and his Magma implementation is released in V2.14. The Sylow subgroups of the families $Sz(q)$ and ${}^2G_2(q)$ were released in V2.13. The Sylow subgroups are produced in the natural matrix representation.
  - The Sylow subgroups of the classical groups were already included in V2.13.

- *Conjugacy Classes*
  - Machinery for computing with element conjugacy in the linear, unitary and symplectic families of classical groups has been implemented by S. Haller and S. Murray. In particular, functions are provided for determining representatives of each class, calculating the corresponding centralisers, determining the class in which an arbitary element lies and constructive conjugation of elements in the respective groups.
  - The conjugacy classes of elements for the simple groups of Suzuki have been implemented by H. Bäärnhielm.
- *Constructive Recognition*
  - It is now possible to to perform constructive recognition on both the large and small Ree groups (that is, ${}^2F_4(q)$ and ${}^2G_2(q)$) in various matrix representations using a package developed by Hendrik Bäärnhielm.
  - Constructive recognition of $U_3(q)$ and $U_4(q)$ has been implemented by Peter Brooksbank.
- *Databases*
  - The Small Groups database has been agumented by code that will enumerate all groups of any square-free order. This code was developed by Bettina Eick and Eamonn O'Brien.
  - The Magma version of the Atlas database of matrix and permutation representations of simple groups and simple groups with decorations has been updated to roughly correspond to the current contents of R. Wilson's Atlas web site. The number of groups contained in the new version of the database is approximately 700 compared to 300 in the previous Magma version.
- *Group Cohomology*
  - The package has been extended in a number of ways. The more important changes include the calculation of 1-coboundaries and 2-coboundaries, the restriction of cohomology to a subgroup, calculations with corestriction and coboundary maps, and having the extension functions return the projection and injection maps.

Finitely-Presented Groups

- *Nilpotent Quotient Algorithm*
  - The latest version (2.2) of Werner Nickel's Nilpotent Quotient program has been installed in Magma by Bill Unger and Michael Vaughan-Lee. This version uses combinatorial collection and so is often much faster than the version included in earlier releases of Magma. It also contains expanded functionality including an ability to handle identical relations.

Lie Groups

- *General*
  - In a major project, A. Cohen, S. Haller and S. Murray have designed and implemented a practical version of Lang's algorithm for connected reductive groups of Lie type. Among various applications this can be used to compute twisted tori.
  - Some conjugation functions for groups of Lie type are provided: Conjugation of a semisimple element into a torus; conjugation of any element into a Borel subgroup.
- *Finite Groups of Lie Type*
  - A conformal group is the group of matrices that preserve a given bilinear or quadratic form up to a constant. Constructions are provided for the conformal groups corresponding to the forms defining the classical groups.
  - Fast machinery for solving element conjugacy problems in most families of classical groups has been developed. In particular, it is possible to determine (a) a representative element from each conjugacy class, (b) the centraliser of any element in the group, and (c) test any pair of elements for conjugacy. With the exception of (a), the algorithms used are polynomial-time.

Representation Theory

- *Characters and Block Theory*
  - It is now possible to determine the table of rational characters of a finite group.
  - An algorithm has been implemented for computing the p-blocks of the table of ordinary characters for a finite group. It is also possible to construct the defect group of a p-block.
- *Ordinary Representations of Finite Groups*
  - A key problem when constructing an ordinary irreducible representation of a group is to determine its Schur index, that is, the degree of a minimal field for the representation taken over the field generated by its character values. The first practical algorithm for this was developed in 2006 by G. Nebe and W. Unger. This version of Magma contains an implementation of the algorithm.
  - The problem of writing a given (absolutely irreducible) representation over as small a field as possible (or over an "arbitrary" user defined field) is a key problem in representation theory. A new method due to C. Fieker and based on Galois cohomology has been implemented. This method will find a minimal subfield that affords a given representation. If this field is not "small enough" then a constructive version of the Grunwald-Wang theorem is used to find a minimal degree splitting field.
- *Representations of Lie Groups*
  - An extensive package for computing the combinatorial properties of highest weight representations of a Lie algebra has been written by Dan Roozemond. For example, given two highest weight representations, we can compute the decomposition of their tensor product into highest weight representations. The aim is to provide functionality in Magma equivalent to that in the now defunct `LiE` package.

Commutative Algebra

- *Gröbner Bases*
  - The F4 algorithm is now used for computing with ideals having fixed bases. Thus the coordinate matrix for a Gröbner basis is now found much more quickly.
  - A new algorithm, based on Faugere F4 techniques, has been introduced to reduce a sequence of polynomials modulo another sequence of polynomials or an ideal. This is important for the efficient computation of the secondary invariants of a finite group.
  - The memory management in the F4 algorithm has been improved so that less memory is used when there are extremely large ultrasparse matrices; the time is also significantly reduced in such cases.
  - The method used to calculate Gröbner bases over algebraic number fields (including cyclotomic and quadratic fields) has been greatly improved.
- *Ideals and Modules*
  - The primary decomposition and radical algorithms have been improved by heuristics to quickly determine whether or not the ideal is prime or radical (thus catching common cases quickly).
  - Modules over multivariate polynomial rings have been completely revised. The former embedded and reduced types have been merged into a single type, which supports the features of both previous types. Any sub- or quotient module may be defined in embedded or reduced form, and such modules may be mixed.
  - Full support is provided for gradings and homogeneous modules for the first time.
- *Invariant Theory*
  - The computation of primary invariants has been improved by the use of Faugere F4 techniques.
  - A new algorithm designed and implemented by G. Kemper for computing the secondary invariants in the non-modular case has been implemented. This algorithm is very much faster than the previous one.
  - Invariant rings of reductive algebraic groups can be computed in Magma for the first time using Derksen's algorithm among others. The Magma code was developed by G. Kemper.
  - Algorithms have been implemented for computing invariant fields (these include Derksen's algorithm). The Magma code was developed by G. Kemper.

Algebraic Geometry

- *General Schemes*
  - The code for computing images under general maps as been rewritten for increased speed.
  - Basic attributes of schemes (such as non-singularity, irreducibility and the singular subscheme) are now stored to avoid expensive recomputation.
  - For schemes over algebraic function fields, it is now possible to test whether the scheme is locally soluble over a completion of the base field.
- *Algebraic Curves*
  - A package has been developed which, given an algebraic curve X, and a subgroup G of the automorphism group of X, computes the quotient of X by G as an algebraic curve.

Arithmetic Geometry

- *Conics*
  - The algorithm by J. Cremona and M. van Hoeij for finding points on plane conics over rational function fields has been installed. (Code was written by John Cremona and David Roberts).
- *Elliptic Curves over Finite Fields*
  - A canonical lift method has been implemented to provide fast point counting for curves over finite fields in small, odd characteristic. This case was not covered by the fast point counting machinery previously installed in Magma.
  - A much more efficient version of the Weil pairing has been coded while the Tate, Eta and Ate pairings have been implemented for the first time in Magma. In each case Miller's algorithm is used. This project was undertaken by F. Vercauteren.
- *Elliptic Curves over the Rationals*
  - A new function `MordellWeilShaInformation` is provided as an easy interface to all the relevant Magma machinery.
  - A new algorithm by Steve Donnelly for computing the Cassels-Tate pairing on the 2-Selmer group of an elliptic curve over $\mathbf{Q}$ has been programmed.
  - For curves admitting 2-isogenies, a routine for lifting "descent via 2-isogenies" to full 2-descent has been provided.

  - An 8-descent routine is now provided. The algorithm and implemention are by Sebastian Stamminger.

- *Elliptic Curves over p-adic Fields*
  - Versions of some routines for obtaining local information about elliptic curves over the rationals now work in the case of curves defined directly over p-adic fields. These include computation of conducter, Tamagawa numbers (Tate's algorithm) and minimal models, and also computation of the root number.

- *Elliptic Curves over Number Fields*
  - Root numbers of a curve over a number field may be efficiently computed, in full generality, using an algorithm of T. Dokchitser and V. Dokchitser. The implementation was undertaken by T. Dokchitser.

- *Elliptic Curves over Rational Function Fields*
  - Two-descent machinery has been added in characteristic 2: a routine for computing the two-isogeny Selmer groups is provided for non-supersingular curves.
  - A full two-descent routine is available in odd characteristic for curves without 2-torsion, representing the elements of the 2-Selmer group as hyperelliptic curves. A separate routine for descent via 2-isogenies has been contributed by David Roberts.
  - Minimization and point-searching are available for the 2-covering curves (written by David Roberts).

- *Hyperelliptic Curves over Finite Fields*
  - For curves defined over finite fields of characteristic 2, Kedlaya's algorithm for point counting has been implemented by F. Vercauteren. Around genus 4 the Kedlaya algorithm out-performs the Mestre canonical lift approach currently used.

- *Plane Curves over Finite Fields*
  - An efficient implementation of Diem's algorithm for computing discrete logarithms for points on the Jacobian of a general plane curve C over $GF(q)$ having small genus was developed by Jasper Scholten. The complexity of this method is $O(q^{2-2/(d-1)})$, where d the degree of C.

Modular Arithmetic Geometry

- *Modular Curves*
  - Code has been developed by Mark Watkins for finding models of modular curves $X_0(N)$ and their quotients by Atkin-Lehner involutions.
- *Modular Forms*
  - A major revision of the modular forms package is being undertaken. In particular, very considerable improvements to efficiency for the main routines in the package have been achieved (in particular, computation of newforms and of q-expansion bases).
  - Modular forms of weight one are now included in the package. A special feature is the direct computation of those forms associated to dihedral Galois representations. This was adapted from code provided by Kevin Buzzard.
  - Modular forms of half-integral weight are also included in the package. The functionality now available includes q-expansions bases of the spaces, and basic operations.
- *Arithmetic Fuchsian Groups*
  - A module for working with arithmetic Fuchsian groups has been developed by John Voight. The module provides support for determining basic invariants of a Fuchsian group Γ and for computing a fundamental domain for Γ. Specialised algorithms are provided for triangle groups.

Coding Theory

- *Linear Codes over Fields*
  - A database of best known linear codes (BLKC) over GF(5), GF(7), GF(8) and GF(9) constructed by M. Grassl, is included in Magma for the first time.
  - The existing database of best known linear codes (BLKC) over GF(2), GF(3) and GF(4) has been upgraded by M. Grassl to include some newly discovered better codes. Compared to previous released versions of the Magma BKLC database, 1308 codes over $GF(2)$, 102 codes over $GF(3)$ and 160 codes over $GF(4)$ have been improved, and the maximal length for codes over $GF(3)$ and $GF(4)$ has been increased to 243 and 256, respectively.

System and Language

- *Associative Arrays*
  - A new type for associative arrays has been introduced.

3 Documentation

New chapters in the Handbook for V2.14 (with their chapter numbers) are:

- Associative Arrays (13).

4 Language and System Features

New Features:

- When an intrinsic is printed, its signatures are now sorted alphabetically by type.
- A new string type for binary data has been created. Using this type it is possible to read and write binary data files.
- One may now use {`[T]`} to specify a set OR sequence with elements of type T for an argument of an intrinsic signature.

5 Aggregates

5.1 Associative Arrays (New)

New Features:

- A new type for associative arrays has been introduced. One may index an array with objects from an arbitrary structure.

6 Groups

6.1 Finite Groups

New Features:

- New functions ClassesData and ClassRepresentative have been added for accessing information stored in a group on its conjugacy classes. The function ClassesData returns a list of pairs giving element order and size of each conjugacy class. If a representative of the class is wanted the ClassRepresentative function will return this representative. This saves the construction of all class representatives when only a few are wanted. This can mean considerable improvement in time and space requirements when working with conjugacy classes of large finite groups. The older functions ConjugacyClasses/Classes is still supported, but we advise using this function only for smaller groups.

Bug Fixes:

- The AbelianBasis function has been fixed in the cases of matrix and permutation groups to give results that agree with the handbook description of the function and with the results when applied to a pc-group.
- A number of bugs in the computation of subgroups when conditions are imposed on the subgroups returned have been fixed.

6.2 Permutation Groups

Changes:

- The data structure behind the GSet types has been altered to be an indexed set. In many applications this will save half the time that used to go in GSet construction.
- Permutation group action on linear codes has been improved to speed up enumeration of orbits of code words under such an action.

6.3 Matrix Groups Over General Rings

New Features:

- Algorithms for computing the soluble radical and radical quotient of a matrix group over the integers modulo n have been installed. These groups now have access to structural algorithms such as Subgroups, CompositionFactors, ChiefFactors, AutomorphismGroup and many others.

6.4 Matrix Groups over Finite Fields

Bug Fixes:

- The base point selection algorithm for matrix groups is now more tightly controlled to avoid massive use of time and space in searching for good base points.

New Features:

- The membership test for classical matrix groups over finite fields has been greatly improved in efficiency.
- Intrinsics for constructing the conformal unitary and conformal symplectic groups have been added.

6.5 Finite Soluble Groups

Changes:

- The algorithm for computing the centralizer in H of an element not in H has been made more efficient.

6.6 Finitely Presented Groups

Changes:

- The latest version (2.2) of Werner Nickel's Nilpotent Quotient program has been installed in Magma by Bill Unger and Michael Vaughan-Lee. This version uses combinatorial collection and so is often much faster than the version included in earlier releases of Magma. It also contains expanded functionality including a ability to handle identical relations.

  There are some changes to the algorithm parameters. The badly named "Metabelian" parameter has been renamed "Nickel" in honour of the author of the package. The default value of the "SemigroupOnly" parameter has been changed to true. The new parameter "NumberOfFreeVariables" controls which generators are treated as identical generators for the duration of the computation of the quotient.

7 Basic Rings

7.1 Integer Ring

New Features:

– Large factors computed via `ECM` or `MPQS` are now stored so that subsequent factorization of the same integer is fast. The procedure `StoreFactor` allows one to add specific integers to this list to help.

– `ECPP` has been partially upgraded and now works with Morain's more extensive precomputed data tables from 1998. This allows the verification of primality of larger numbers for which the implementation would have previously failed.

7.2 Polynomial Rings

New Features:

– Multiplication of polynomials over finite fields of characteristic 2 has been greatly improved.

  - For polynomials over $GF(2)$ this is achieved through use of the Cantor algorithm which has better complexity than the Karatsuba algorithm.

  - For high degree finite fields of characteristic 2 the Kronecker segmentation expansion method is now used. Multiplication of degree 1000 polynomials over $GF(2^{1000})$ is now about 10 times faster than in V2.13.

– Factorization of univariate polynomials over small finite fields has been completely overhauled, leading to every significant speedups. This is particularly significant for fields of characteristic 2 and for very sparse polynomials over small fields of any characteristic.

– Independent of the above, the irreducibility test for polynomials over finite fields has been greatly improved through use of a sieving method.

– A proper squarefree factorization algorithm is now provided for all applicable base fields, handling inseparability when applicable.

– An asymptotically-fast half-gcd-like algorithm is now used for resultant computation.

– The computation of the inverse of a rational polynomial modulo another rational polynomial has been sped up greatly (is now asymptotically-fast in both the degree and bit size).

– Factorization of polynomials over function fields of characteristic zero has been sped up, particularly in the case where there are non-trivial repeated factors.

– The multivariate GCD algorithm has been extended to take advantage of the important case in which the quotient of one of the inputs by the GCD is a low-degree polynomial.

– New function `Evaluate(L, P)` to evaluate a sequence of polynomials L at one fixed sequence P of points.

– The ability to factorise polynomials over a ring is one of the basic tools for obtaining an understanding of the extensions of that ring. In V2.14 for the first time we support a factorisation of polynomials over certain power series rings. This was achieved by extending S. Pauli's p-adic factorisation method, which is actually an algorithm for the factorisation of polynomials over local fields.

7.3 Finite Fields

Changes:

- The greater use of sparse irreducible polynomials means that field extensions now have possibly different defining polynomials.

New Features:

- Extension fields are now defined by sparse lexicographically minimal polynomials when possible (and when Conway polynomials are not available), leading to big speedups of arithmetic in moderate to high degree extension fields. Databases of such polynomials have been constructed for the following fields:
  - GF(2): up to degree 120,000.
  - GF(3): up to degree 50,000.
  - GF(4), GF(5), GF(7): up to degree 20,000.
  - GF(q), $9 \leq q \leq 127$: up to degree 1,000 or more.
- A new packed representation for finite fields of characteristic 3 has been introduced giving large speedups for fields of higher degree, in particular. In addition, a fast irreducibility test for polynomials over GF(3) has been devised. This algorithm can run 10 times faster in the case of sparse polynomials.
- Magma now uses an algorithm due to Eric Rains for constructing isomorphisms between fields and embedding subfields in larger fields. For example, embedding GF(2^{1000}) in GF(2^{2000}) now takes 0.3 seconds, compared to 20 minutes with V2.13.
- For polynomials whose coefficients lie in a subfield, factorization and root finding have been sped up enormously.
- The Magma implementation of the Berlekamp algorithm now uses a sparse matrix datastructure when the polynomial is sparse leading to less memory usage and speedups, particularly over $GF(2)$. Sparse polynomials of degree $100,000$ over $GF(2)$ can be factored in a few seconds.
- An improved test for irreducibility of polynomials over $GF(2)$ has been installed. The calculation of the trace of an element in a field of characteristic field 2 has also been sped up.
- An intrinsic `Frobenius` is now provided for the fast computation of the Frobenius image of a matrix or other object (thus speeding up p^k-th powers and p^k-th roots, where p is the characteristic).
- It is possible for the first time to compute k-th roots of elements in finite fields, for arbitrarily large integer k.
- The database of Conway polynomials has been greatly expanded.

8 Linear Algebra and Module Theory

8.1 Matrices

New Features:

- Matrix multiplication has been greatly improved in the case that one of the input matrices is sparse and the other dense.
- Matrix multiplication has been sped up over prime finite fields by the use of precomputation of the inverse of the modulus.
- Dense matrix multiplication over GF(2) now uses Intel SSE2 instructions when supported.
- Linear algebra over finite fields of characteristic 2 has been sped up in general.
- The calculation of the Smith normal form and the determinant of a dense matrix defined over an euclidean ring has been greatly improved through replacing the Havas-Holt-Rees algorithm by an asymptotically-fast recursive echelon algorithm.
- Echelonization and nullspace computation for sparse matrices over the rational field have been improved.

9 Commutative Algebra

9.1 Ideal Theory and Gröbner Bases

New Features:

- The F_4 algorithm is now used for computing with ideals having fixed bases. Thus the coordinate matrix for a GB is now found much more quickly.
- A new function `NormalForm`(L, G), based on Faugere F_4 techniques, is now provided to reduce a sequence of polynomials L modulo another sequence of polynomials G (or an ideal). This is important for the efficient computation of the secondary invariants of a finite group.
- The memory management in the F_4 algorithm has been improved so that less memory is used when there are extremely large ultrasparse matrices; the time is significantly reduced in such cases.
- The main strategy to compute the GB of an ideal has been improved through the introduction of various preprocessing techniques.
- Computation of GBs over algebraic number fields (including cyclotomic and quadratic fields) has been greatly improved.
- Computation of GBs over rational function fields with a small number of indeterminates has been improved.
- The primary decomposition and radical algorithms have been improved by heuristics to quickly determine whether or not the ideal is prime or radical (thus catching common cases quickly).
- The computation of the Hilbert series of an ideal has been improved by more efficient selection of a suitable GB.

9.2 Modules over Affine Algebras

New Features:

- Modules over multivariate polynomial rings have been completely revised. The old separate embedded and reduced types have been merged into a single type, which supports the features of both previous types. Any sub- or quotient module may be defined in embedded or reduced form, and such modules may be mixed.
- Full support is provided for gradings and homogeneous modules for the first time.
- The computation of syzygies and resolutions has been sped up.

9.3 Invariant Theory

New Features:

- Algorithms have been implemented for computing invariant rings of reductive groups. Derksen's algorithm for reductive groups is among those implemented.
- Algorithms have been implemented for computing invariant fields (these include Derksen's algorithm).
- A new algorithm of G. Kemper for computing the secondary invariants in the non-modular case has been implmented. This algorithm is very much faster than the previous one.
- The primary invariants algorithm has been improved by use of Faugere F_4 techniques.

10 Extensions of Rings

10.1 Algebraic Number Fields

New Features:

- `NumberField` for parents of places and divisors of number fields has been included.
- Homomorphisms of orders of number fields can now be applied to ideals of orders of number fields when the codomain is also an order.
- The support for infinite places of number fields has been improved, in particular it is now easy to get access to all real places and test for positivitiy at any or all real places.
- The computation of Galois group for polynomials of degree > 23 over extensions of $\mathbf{Q}$ is now supported. The functionality to compute arbitray subfields of a normal closure has been extended to compute arbitrary subfield towers correspoding to subgroup chains.
- The computation of splitting fields for polynomials over $\mathbf{Q}$ can now be based on the computation of the Galois group to gain efficiency.
- For polynomials over $\mathbf{Z}$ with solvable Galois group, it is now possible to express the roots as radicals.
- Given a 2-cocycle with values in the multiplicative group of a number field, it is now possible to determine if this cocycle splits, and in this case, to compute a 1-cochain to verify this.
- Linear dependecies of arbitrary elements of the splitting field of a polynomial can be compututed.
- A routine `NiceRepresentativeModuloPowers` for number field elements has been provided.

Bug Fixes:

- As a result of the re-write of the computation of Galois groups over number fields, a number of bugs in this module have been removed.

10.2 Quadratic Fields

New Features:

- The 2-part of the class group can now be computed using the method of Bosma and Stevenhagen.

10.3 Abelian Extensions

New Features:

- New functionality to exploit the Galois-module properties of an abelian extension of a normal field have been added. In particular it is possible to create the corresponding cohomology module and thus compute the low-dimensional cohomology groups explicitly.
- A function to directly enumerate absolutely normal subfields of some suitable abelian extension has been provided.

10.4 Local Fields

Removals and Changes:

- The slow `GaloisImage` for unramified extensions of Q_p without cyclotomic or Gaussian normal bases has been reimplemented for greater speed using modular composition.

New Features:

- Given a monic polynomial with coefficients in $\mathbf{Z}$, it is now possible to compute a p-adic splitting field.
- In a local ring L, with residue field k and ring of integers O_L, the Teichmüller lift of a non-zero element $u \in k$ is the unique root of unity $\hat{u} \in O_L$ of order prime to the characteristic of k that reduces to u modulo the maximal ideal. When O_L is a fixed precision p-adic quotient ring or unramified extension of such, there is now a fast implementation of `TeichmuellerLift` using the iterative method of Harley.

Bug Fixes:

- `XGCD` of polynomials over local rings has been fixed.

10.5 Algebraic Function Fields

Removals and Changes:

- The implementation of `Expand` and the application of the map returned from `Completion` has been improved to use quadratic newton lifting. The new implementation has been seen to be up to 200 times faster.
- The default precision of the series ring which is the `Completion` of an order of a function field has been changed to 20 to be consistent with the series rings which are returned as the completions of function fields themselves.
- For global function fields, class field theory based on Drinfeld modules is now accessible. In particular, this means that for any place, the corresponding rank-1 module can be explicitly determined with coeffcients in the Hilbert-class field. In the place is of degree 1 as well, then the module will be sign-normalizsed and integral.

New Features:

- The computation of more general integral closures is now possible. In particular, this means that valuations of the coefficient ring can be included in the computations. For example, the closure of $\mathbf{Z}[x]$ in a function field defined over $\mathbf{Q}(x)$ can be computed.
- Homomorphisms of orders of function fields can now be applied to ideals of orders of function fields so long as the codomain is an order.

10.6 Series Rings

Removals and Changes:

- `Factorization` of polynomials over series rings has been improved and is now available over laurent series fields and rings with infinite precision.

11 Lattices and Quadratic Forms

11.1 Lattice Reduction

New Features:

- Simon's variant of LLL-reduction for indefinite forms has been efficiently implemented in Magma as part of a new package of Damien Stehlé, and works readily in dimensions greater than 100.
- The automorphism group and theta series of a lattice is now stored and can be asserted via attributes.
- A new function ThetaSeriesLimited(L, n) takes a time limit and returns the contribution to the first n coefficients of the theta series of a lattice L found by lattice enumeration within the specified CPU time.
- Given a quadratic form F in an arbitrary number of variables, Mark Watkins has used Denis Simon's ideas as the basis of an algorithm for finding a large (totally) isotropic subspace of F.

12 Algebras

12.1 Finite Dimensional Algebras

Changes:

- The function `LieAlgebra`, taking a set/sequence of relations in an FP-Lie algebra, has had its return values changed.

New Features:

- One may now create arbitrary quotients of associative algebras defined over general euclidean rings (including rings with zero divisors, such as residue class rings). Only free algebras and their subalgebras over such rings were supported before.
- Quotient algebras over euclidean rings may have both free and torsion parts and this underlying structure may be accessed via the new function `Moduli`.
- Lie Rings are now supported; these are Lie algebras defined over a euclidean ring.

12.2 Quaternion Algebras

New Features:

- The `ConjugacyClasses` of maximal orders or Eichler orders in definite quaternion algebras can now be computed.
- A routine `IsConjugate` (or `IsIsomorphic`) is provided for quaternion orders.
- Representatives of the `TwoSidedIdealClasses`, and the `TwoSidedIdealClassGroup` can be computed for an order in a definite quaternion algebra; the `Support` of the representatives can be specified by the user.
- The `Factorization` of a two-sided ideal (in a definite quaternion order) can be obtained.

- An alternative algorithm for the `LeftIdealClasses` and `RightIdealClasses` is provided, which is more efficient in many cases; the `Support` of the representatives can be specified by the user.
- A more sophisticated `UnitGroup` algorithm, using results about the structure of the unit group, has been implemented for orders in definite quaternion algebras.
- An intrinsic `NormOneGroup` is included.

Bug fixes:

- A number of errors in enumeration of units and right ideal classes for orders in definite quaternion algebras over number fields have been corrected.
- The `pMatrixRing` routine is now stable and available in all cases.
- Trivial bugs in `Embed` have been fixed.

12.3 Orders in Associative Algebras

Removals and Changes:

- Creation of two sided ideals of orders of associative algebras has been improved. Checks have also been added for all ideal creations from a basis (matrix or pseudo matrix).
- `Algebra` of an order will now always return the algebra the order was created from.

New Features:

- `AlgebraOverFieldOfFractions` has been provided for orders over number rings to return the algebra over the field of fractions of the number ring rather than the number field (if the order was created of an algebra over a number field).
- The following routines and operations are now included: `Generators` of an order or ideal as a module over its base ring, `CommutatorIdeal`, “`meet`” and conjugation of an order by an element.

12.4 Matrix Algebras

New Features:

- A very efficient algorithm for computing the unit group and Jacobson radical of a matrix algebra defined over a finite field has been developed and implemented by P. Brooksbank and E. O’Brien.

12.5 Characters of Finite Groups

New Features:

- A utility `RationalCharacterTable` has been installed to compute the table of rational characters of a finite group. Note that it is the character values that are rational. These characters will be afforded by rational representations if and only if the appropriate Schur index is 1. The algorithm starts by computing the character table of the given group.

- An algorithm has been implemented for computing the p-blocks of the table of ordinary characters for a finite group. The algorithm used is very fast once the table of ordinary characters has been constructed. It relies on computing inner products over p-regular elements of the group. The computations are performed in a finite field for speed. The intrinsic function is named `Blocks`. Another utility `DefectGroup` will return a defect group of a p-block.

Bug Fixes:

- The intrinsic function `CharacterTable` now applies to finite groups of type `GrpAb`. A serious memory problem when computing the characters of an abelian group has also been fixed.

12.6 Ordinary Representations of Finite Groups (New)

New Features:

- A key problem when constructing an ordinary irreducible representation tion of a group is to determine its Schur index, that is, the degree of a minimal field for the representation taken over the field generated by its character values. The first practical algorithm for this was developed in 2006 by G. Nebe and W. Unger. This version of Magma contains an implementation of the algorithm. The algorithm works with characters of the group and its subgroups, and fields generated by character values to determine the Schur index of the character over all completions of the rationals. C. Fieker has provided a routine that uses this data to compute the Schur index of the character over a given number field.

  The algorithm first computes strong upper bounds for the local Schur indices, then uses the Brauer-Witt Theorem and a search through subgroups of the group to reduce to considering quasi-elementary groups. Schur indices are then computed by calculations with values of Brauer characters, degrees of field extensions and, in the 2-adic case, further reduction to a case considered by U. Riese & P. Schmid. The upper bounds used are sufficiently strong that in many cases the subgroup search, which may be very time-consuming, is not necessary.

  The names of the new intrinsic functions are `SchurIndex`, for the Schur index over a particular number field, and `SchurIndices` for the indices over all completions of a number field.

- The problem of writing a given (absolutely irreducible) representation over as small a field as possible (or over an “arbitrary” user defined field) is a key problem in representation theory. A new method due to C. Fieker and based on Galois cohomology has been implemented. This method will find a minimal subfield that affords a given representation. If this field is not “small enough” then a constructive version of the Grunwald-Wang theorem is used to find a minimal degree splitting field.

13 Lie Theory

13.1 Coxeter Groups as FP Groups

New Features:

- Functions for computing transversals and double coset representatives of minimal length with respect to standard parabolic subgroups have been installed. The algorithms were supplied by R. Howlett.

13.2 Groups of Lie Type

New Features:

- Twisted groups of Lie type can now be constructed.

14 Algebraic Geometry

14.1 Schemes

Removals and Changes:

- IsLocallySolvable has been disabled for schemes in weighted projective spaces (except special cases), as this was not working correctly.
- The subset relation is deprecated; IsSubscheme should be used instead.
- IsLinearScheme has been changed to IsLinear.

New Features:

- The computation of images of maps between ordinary projective schemes has been rewritten to take better advantage of the bi-homogeneity in both sets of variables of the natural graph. This involves a saturation but avoids the expensive computation of an elimination Groebner basis and is much faster in most cases. NB: For efficiency, the new image computation is only used when the domain of the map is known to be irreducible, based on either the relevant attribute or the domain being a curve with a function field. In the non-irreducible case, multiple saturations may be necessary.
- Basis attributes of schemes are now stored for use in later computations. For certain types of curves (such as conics and elliptic curves), attributes may be set on creation. They can be set (but not reset) by the user; the attributes are Irreducible, Reduced, GeometricallyIrreducible, GeometricallyReduced, Nonsingular and SingularSubscheme.
- The generic case of IsLocallySolvable has been adapted to work for schemes over algebraic function fields.
- A function HeightOnAmbient has been added, which computes the height in projective space of a point on any scheme defined over a number field.
- For a scheme defined over a number field, the RestrictionOfScalars to a subfield can be computed.
- PointSearch may now be called for affine schemes.

Bug Fixes:

- A bug in RationalPointsByFibration has been fixed. The default for the UseHypersurface parameter has been changed to false.
- A bug that was causing errors in Is(Non)Singular and SingularSubscheme for the empty scheme has been fixed.
- A bug in GenericPoint for curves and ambients, which caused crashs over certain basefields (particularly function fields), has been fixed.

14.2 Algebraic Curves

New Features:

- For G an automorphism group of algebraic curve C of genus >= 2, a function to compute the quotient curve C/G, `CurveQuotient`, is now available. This uses a combination of techniques, mainly relying on Magma's function fields but also using Invariant theory when the quotient is of genus 0 or 1. When the quotient has genus >= 2, it is returned as a canonical curve, a non-split geometrically hyperelliptic curve in P^3 or a hyperelliptic model. In the genus 0 or 1 case, it is returned in a small degree projective normal embedding.

15 Arithmetic Geometry

15.1 Rational Curves and Conics

New Features:

- The algorithm by J. Cremona and M. van Hoeij for finding points on plane conics over rational function fields has been installed. (Code was written by John Cremona and David Roberts).

Bug Fixes:

- A bug in `Parametrization` for rational curves, which occasionally caused an invalid map to be returned, has been fixed.

15.2 Elliptic Curves

New Features:

- Functions have been provided to calculate the `FormalGroupLaw` of an ellliptic curve, and the `FormalGroupHomomorphism` associated to an isogeny.

15.2.1 Elliptic Curves over the Rational Field

Removals and Changes:

- `HeegnerPointsOverClassField` has been renamed `HeegnerPoints`.

New Features:

- An interface function `MordellWeilShaInformation` is provided; this calls descent routines and analytic routines to obtain all available information about the Mordell-Weil group and the Tate-Shafarevich group of a given curve.
- A new algorithm by Steve Donnelly for computing the `CasselsTatePairing` on the 2-Selmer group of an elliptic curve over **Q** has been programmed; the input is pair of 2-coverings of the kind obtained from `TwoDescent`. This provides the same information as FourDescent regarding the rank and the Tate-Shafarevich group of a curve; however, the new algorithm does not require any class group or norm equation computations.
- For curves admitting 2-isogenies, a routine is provided for performing higher descent on 2-coverings corresponding to 2-isogenies (in other words, extending to a full 2-descent).
- A `EightDescent` routine is now provided; it determines all locally soluble 8-coverings lying above a given 4-covering, and presents these as intersections of three quartics in P^3. The algorithm and implemention are by Sebastian Stamminger.
- The machinery for Heegner points has been extended: `HeegnerPoints` and `HeegnerForms` now work in greater generality, and the argument to `HeegnerForms` may be simply a level N. The `HeegnerTorsionElement` attached to an Atkin-Lehner involution may be obtained.
- The `ManinConstant` relative to $X_0(N)$ can be computed.
- The `FaltingsHeight` of a curve can be computed.
- A function `TwoTorsionOrbits` is provided, returning the Galois orbits of non-trivial 2-torsion points.

15.2.2 Elliptic Curves over Number Fields

New Features:

- Root numbers of a curve over a number field may be efficiently computed, in full generality, using an algorithm of T. Dokchitser and V. Dokchitser. The implementation was undertaken by T. Dokchitser.
- `HeightPairingMatrix` now also works over number fields.

15.2.3 Elliptic Curves over p-adic Fields

New Features:

- Arithmetic can now be performed on p-adic points on elliptic curves.
- Versions of some routines for obtaining local information about elliptic curves over the rationals now work in the case of curves defined directly over p-adic fields. These include computation of conducter, Tamagawa numbers (Tate's algorithm) and minimal models, and also computation of the root number.

15.3 Elliptic Curves over Finite Fields

New Features:

- A canonical lift method has been implemented to provide fast point counting for curves over finite fields in small, odd characteristic p. It is particularly fast in the cases where the modular curve $X_0(p)$ is of genus 0 when a modular parameter of level p or p^2 is lifted. When $X_0(p)$ is elliptic or hyperelliptic, an adaptation is used, devised by M. Harrison, which lifts nice Weierstrass coordinates. In the general case, the j-invariant is lifted using the classical modular polynomial. This case was not covered by the fast point counting machinery previously installed in Magma.
- A much more efficient version of the Weil pairing has been coded using Miller's algorithm (by F. Vercauteren).
- The Tate, Eta and Ate pairings have been implemented. In each case Miller's algorithm is used. This project was undertaken by F. Vercauteren.

15.4 Elliptic Curves over Function Fields

Bug Fixes:

- Several bugs in `TorsionSubgroup` have been fixed.

New Features:

- In characteristic 2, a routine `TwoIsogenySelmerGroups` computes the Selmer groups of the Frobenius isogeny and its dual, for all ordinary elliptic curves defined over a rational function field $k(t)$ with k finite.
- In odd characteristic, a function `TwoDescent` is now available for curves without 2-torsion, returning hyperelliptic curves of degree 4 representing the nontrivial elements of the `TwoSelmerGroup`.

- In characteristic at least 5, for curves admitting 2-isogenies, a 2-isogeny descent routine can be invoked by calling MordellWeilGroup with Al set to Descent. (Contributed by David Roberts.)
- Minimization of 2-covering curves (in characteristic at least 5) can be done using QuarticMinimize. (Contributed by David Roberts.)
- The function Points efficiently searches for rational points up to a given height bound on hyperelliptic curves (in particular 2-covering curves). The function PointsQI does likewise for curves given as intersections of two quadrics in P^3. Both functions use a lattice reduction method. (Contributed by David Roberts.)
- Functions RankBound and RankBounds provide a simple interface to the relevant machinery.
- Functions IsLinearlyDependent and IndependentGenerators are now available for curves over function fields.

15.5 Genus One Models

New Features:

- IsEquivalent, Minimise and Reduce are now available for models of degree 2, 3 or 4 (interfacing existing functionality in some cases).
- A function InverseTransformation has been added.
- The RamificationPoints, TwoSelmerElement and TwoTorsionMatrices of a model of degree 2 can be obtained.
- The RelativeSelmerElement can be computed for a degree 4 model covering a given degree 2 model.

Bug fixes:

- A sign error in the conversion function HyperellipticCurve has been corrected.

15.6 Modular Curves

New Features:

- The function X0NQuotient computes a reasonable model of the modular curve $X_0(N)$, or its quotient by one or more Atkin-Lehner involutions. The model returned is in general (a reduced system of equations for) the the canonical embedding of the curve; otherwise in low genus it is an elliptic or hyperelliptic curve. The function handles large examples, such as $X_0(997)$ of genus 82.

15.7 Modular Forms

Removals and Changes:

- Computation of Newforms and qExpansionBasis for many weights and levels is now much faster, due to revisions of package code and speed-ups in some number field operations and linear algebra.
- For some small levels, the qExpansionBasis is now obtained efficiently by taking products of forms of low weight.

- The `PrecisionBound` is now sharper for many spaces.
- The behaviour of `EisensteinSeries` has changed again: the function lists the series corresponding to only one Galois representative of each orbit of characters, unless the new optional parameter `AllCharacters` is set to true.
- The intrinsic `DeleteAllAssociatedData` (superceding `DisownChildren`) reclaims most of the memory occupied by a space of modular forms (which may not be automatically reclaimed by the memory manager).

New Features:

- Half-integral weight forms are included in the package. The functionality available in this release includes basic arithmetic, `CuspidalSubspace` and `qExpansionBasis`.
- For weight 1/2, the function `WeightOneHalfData` returns the basis described by Serre-Stark.
- Weight one forms can now be computed (code adapted from an original version by Kevin Buzzard). Available functionality includes basic arithmetic, `CuspidalSubspace`, `EisensteinSubspace`, `EisensteinSeries`, `Dimension`, `qExpansionBasis` and Hecke operators.
- Weight one eigenforms associated to dihedral Galois representations are computed directly.
- Hecke operators are enabled for spaces with a quadratic Dirichlet character.
- General Atkin-Lehner operators w_q are now implemented for Eisenstein spaces.
- Functions `CuspidalProjection` and `EisensteinProjection` are provided for convenience.

15.7.1 Dirichlet Characters

Removals and Changes:

- `GaloisConjugacyRepresentatives` no longer returns repeats.
- `DirichletCharacterFromValuesOnUnitGenerators` is available as an intrinsic.
- More caching is done to speed up certain operations, such as `IsOdd/IsEven`.

15.8 Modular Symbols

Removals and Changes:

- The behaviour of `qIntegralBasis` has changed, now including all oldforms (for spaces with more than one Dirichlet character).
- The intrinsic `DeleteAllAssociatedData` (superceding `DisownChildren`) reclaims much of the memory occupied by a space of modular symbols (which may not be automatically reclaimed by the memory manager).

15.9 Arithmetic Fuchsian Groups and Shimura Curves

Features:

- A Fuchsian group can be created from an order in a quaternion algebra over a totally real number field (ramified at exactly one infinite place).
- Basic invariants of a Fuchsian group (genus, signature, volume) can be computed.
- A finite presentation of a Fuchsian group can be computed, and the word problem can be solved.
- A package treating the geometry of the unit disc under the hyperbolic metric is provided, supporting calculation of geodesics, distance, angles and volumes.
- The action of a Fuchsian group on the unit disc is implemented.
- A fundamental domain for the action of a Fuchsian group on the unit disc can be computed.
- Triangle groups can be created (as Fuchsian groups).
- CM points and their j-invariants can be computed for triangle groups.