Summary of New Features in Magma V2.11

May 2004

1 Introduction

This document provides a terse summary of the new features installed in Magma for release version V2.11 (May, 2004).

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

2 Summary

Groups

- *Finite Groups:* A number of algorithms that perform non-constructive and constructive black-box recognition are available for the first time. These include non-constructive recognition of simple groups; non-constructive recognition of classical groups in their natural representation; constructive recognition of linear groups using the Kantor-Seress algorithm; constructive recognition of the alternating/symmetric group using either Bratus-Pak or Beals *et al.* A datatype and operations for black box groups have been added to support the implementation of group recognition algorithms.

- *Finite Groups:* The database of maximal subgroups and automorphism groups for almost simple groups has been extended. This database is being constructed firstly, by determining all maximals generically for each classical family in each dimension and, secondly, by storing a description of the maximals for a particular group in the case of exceptional groups and sporadic simple groups. The following groups are included for the first time: $L(2,q)$, $L(3,q)$, $L(4,q)$, $L(5,p)$, (q a prime power), $L(d,2)$ for $d \leq 14$, $U(6,2)$, $U(3,9)$ $Sp(8,2)$, $Sp(6,3)$, $O(7,3)$, $O^+(8,2)$, $O^-(8,2)$, $O^+(10,2)$, $O^-(10,2)$, $G_2(4)$, $G_2(5)$, ${}^3D_4(2)$, ${}^2F_4(2)'$, $Sz(32)$, McL, He, Co_3, Co_2, and Fi_{22}. The expansion of the maximal subgroups database automatically expands the range of applicability of other key group structure functions such as subgroups, low index subgroups, automorphism groups, character tables and representations.

- *Finite Groups:* Much of the information in the online *Atlas of Simple Groups* maintained by Robert Wilson at Birmingham is now available as a Magma database. In particular, the list of ordinary and modular representations is available.

- *Permutation Groups:* A number of major improvements to the fundamental algorithms for permutation groups have been made. In the case of many hard examples, these changes will significantly reduce the execution time. Of particular note, the computation of a BSGS in a permutation group will often be much faster in harder cases. Other major speed-ups affect the computation of conjugacy classes of elements and subgroups.

- *Braid Groups:* In 2003, Volker Gebhardt developed a much faster test for conjugacy of elements in a braid group by replacing the earlier notion of *super summit set* by a much more powerful invariant, the *ultra summit set*. Functions for computing ultra summit sets are now included in Magma and both conjugacy testing and conjugacy search now use ultra summit sets instead of super summit sets. While super summit sets in general cannot be computed for products of more than 4 simple elements in a braid group on 8 strings, tests show that ultra summit sets can be computed successfully for elements constructed as product of 1000 random simple elements in a braid group on 100 strings.

- *Automorphisms/Isomorphisms:* It is now possible to determine isomorphism of two finite groups where either can be a permutation group, matrix group or pc-group. For example, it is possible to test whether a matrix group is isomorphic to a permutation group.

- *Finitely Presented Groups:* A Magma database comprising the fundamental groups of the 10,986 small-volume closed hyperbolic manifolds in the Hodgson-Weeks census has been included.

Basic Rings

- *Multivariate Polynomial Rings:* Factorization of bivariate polynomials over all supported rings is now accomplished by a new algorithm which extends van Hoeij's knapsack ideas for $\mathbf{Z}[x]$ to solve the combination problem for $\mathrm{GF}(q)[x, y]$. The new algorithm runs in polynomial time and performs extremely well in practice. General multivariate factorization has been rewritten from scratch and is now reduced to this new bivariate algorithm. Most of the algorithms for multivariate GCD and resultant computation have also been rewritten from scratch. In particular, a faster interpolation algorithm is used for resultants, and there are several new heuristics.

Commutative Algebra

- *Ideal Theory and Gröbner Bases:* A new highly optimized implementation of the Faugère $F4$ algorithm for computing Gröbner bases (GBs) over fields is included in V2.11. The algorithm uses sparse linear algebra and has specialized vector representations for all types of finite fields, and an asymptotically-fast modular algorithm for the rationals. Special techniques are used for systems over GF(2). The Cyclic-7 `grevlex` GB takes 2.2 seconds on an Athlon 2800+ XP PC and the Cyclic-9 `grevlex` GB over **Q** takes 7.5 days on a 750MHz Sunfire v880 (one sequential processor, 11GB memory used). See `http://magma.maths.usyd.edu.au/users/allan/gb/`.

- *Ideal Theory:* The integral closure of the quotient algebra of an ideal in its total ring of fractions can now be computed using a normalisation algorithm included for the first time in V2.11. For ideals in rank 2 polynomial rings, a maximal order in the corresponding function field is computed. For higher rank rings, a general algorithm of de Jong is used. The latter algorithm is often very slow and we plan further improvements. Noether normalisation is also available. Functions for obtaining the equidimensional decomposition of ideals with no embedded primary components have also been added.

Extensions of Rings

- *Algebraic Function Fields*: A new representation of function fields has been implemented (a non-simple extension). A field may be extended by adjoining roots of several different polynomials simultaneously rather than by making a succession of relative extensions. The extension field is a one-step extension of the base field and in certain circumstances may allow much more efficient computation than would be the case with the corresponding relative extension. Fields in this new representation have almost all the functionality of the other representations.

- *p-adic Rings and Fields:* The speed of many operations related to point counting has been greatly improved. In particular, type 1 and 2 Gaussian Normal Bases have been implemented, which give an essentially free Frobenius action without sacrificing multiplication speed.

Algebras

- *Finitely-presented Associative Algebras*: The first non-trivial implementation of these structures has been achieved by extending (where applicable) part of the commutative algebra machinery to noncommutative data structures and algorithms. Noncommutative versions of both the Buchberger and the $F4$ algorithms are provided for computing a noncommutative Gröbner basis of an ideal (if the algorithm should terminate). Consequently, this algorithm may be used to solve the word problem, construct a matrix representation or, if the algebra is finite dimensional, construct an isomorphic structure constant algebra.

Representation Theory

- *Representations of Finite Groups:* An implementation of the Dixon algorithm for finding the irreducible representation affording a given character of a finite group has been coded by Derek Holt. Techniques for finding a field of minimal degree which realises a given representation have also been developed.

- *Representations of Symmetric Groups:* The Symmetrica package is used to provide efficient methods for computing individual characters or character values for the alternating and symmetric groups. Integral, seminormal and orthogonal representations of the symmetric group may also be constructed.

Lie Theory

- *Lie Groups:* It is now possible to construct the highest weight representations and the adjoint representation for groups of Lie type. Functions are provided to determine the centre and also to decompose automorphisms (following Carter).

- *Lie Algebras:* Very efficient machinery has been developed for constructing a Gröbner basis for a finitely presented Lie algebra L by Willem de Graaf and Allan Steel. The GB reduction algorithm used is Magma's generic $F4$ algorithm. At present, if the algebra L is finite dimensional, the GB can be used to construct a Lie algebra defined by structure constants. Alternatively, the GB may be used to construct a nilpotent quotient of L to a designated class.

Differential Rings

- *Differential Rings and Fields:* Machinery has been implemented for defining and working with these structures. The types are based on the existing algebraic function field and affine algebra types. Notable features include the calculation of the differential constant field, extensions of differential rings and fields, and the formation of quotient rings and field of fractions.

- *Differential Operator Rings:* Rings of differential operators can be created over any differential ring or field in such a way that an operator can be applied to elements of the underlying ring or field. Features include the calculation of properties of a place at an operator, the determination of the singular points of an operator, the calculation of the indicial polynomial of an operator and the calculation of all rational solutions of linear differential equations having the forms $L(y) = 0$ and $L(y) = g$.

Algebraic Geometry

- *Schemes:* Machinery for projecting projective schemes isomorphically onto lower dimensional linear subspaces has been implemented. This is applied in a new function which gives a birational embedding of a plane curve as a non-singular projective scheme in 2- or 3-space. The projection of a canonical genus 5 curve over $\mathbf{Q}$ from projective 4-space to 3-space, takes 2 milliseconds on a 750 MHz machine.

- *Hilbert Series and Polarised Varieties:* Magma V2.11 contains a database of 24,099 K3 surfaces that, in a well-defined sense, includes a sketch of all possible examples. As the K3 methods work in many other contexts, tools for calculating subcanonical curves, Fano 3-folds and Calabi–Yau 3-folds are also included. The basic method is to assemble a lot of data, feed it into the Riemann–Roch formula to get a Hilbert series, and then attempt to describe a plausible variety with that Hilbert series.

Arithmetic Geometry

- *Conics:* A new algorithm due to Denis Simon that allows for efficient parametrisation of non-diagonal conics over Q with factorizable discriminant has been implemented. The implementation automatically chooses the best strategy to parametrise a given conic and allows the efficient parametrisation of a much larger class of conics over Q than was previously possible.

- *Elliptic Curves:* The basic machinery for curves over number fields has been considerably expanded. V2.11 includes code written by John Cremona implementing Tate's algorithm for computing local minimal models and Martine Girard's code for determining the heights of a point on an elliptic curve defined over a number field or a function field. Also available is a Magma database of $136, 924, 520$ elliptic curves over $\mathbf{Q}$ with conductors up to 10^8 constructed by William Stein and Mark Watkins.

- *Elliptic Curves:* Analytic methods for elliptic curves over Q have been implemented by Mark Watkins. One can now compute the order of vanishing and special values of the L-function of an elliptic curve. This allows full determination of the data involved in the BSD conjecture for curves over Q. One can also determine the modular degree of an elliptic curve as well as the set of curves isogenous to a given rational elliptic curve. The Heegner point method for the determination of generators of elliptic curves of rank 1 over Q has been implemented by Mark Watkins and Tom Womack.

- *Elliptic Curves:* p-adic canonical lift-based techniques for counting points on elliptic curves over finite fields of characteristic 2 are now included in Magma. When Type I and II GNBs exist, a fast adaptation of the quasi-quadratic algorithm of Lercier and Lubicz by Michael Harrison is used. This mainly applies for fields of size up to about 2^{1000} which include those of cryptographic interest. It combines the good features of Lercier/Lubicz with those of the MSST algorithm. A sample time for a field of size 2^{162} is 0.04 seconds on a 2.1 GHz Athlon. When Type I or II GNBs do not exist, the usual MSST algorithm or, for larger fields, a recursive version by Harley is employed.

- *Hyperelliptic Curves:* It is now possible to compute 2-Selmer groups of arbitrary hyperelliptic curves with a rational Weierstrass point and 2-Selmer groups and 2-isogeny Selmer groups of elliptic curves. Whenever possible, Selmer groups are represented by S-units in a product representation, which avoids coefficient blowup. Assuming GRH, this allows for computation of Selmer groups in truly subexponential time.

- *Hyperelliptic Curves:* Magma routines developed by Nils Bruin allow the application of Chabauty methods to determine the rational points on a curve C over $\mathbf{Q}$ covering an elliptic curve E over a number field with a known Mordell-Weil group. This allows in many cases the computation of an often sharp bound on the number of rational points on a curve that can be mapped non-trivially into certain special abelian varieties. Unramified 2-covers of hyperelliptic curves fit in this category and application of these methods allows for explicit determination of the set of rational points on hyperelliptic curves in many cases.

- *Hyperelliptic Curves:* p-adic techniques have been included for fast point-counting on hyperelliptic curves and their Jacobians over $GF(p^n)$ for small p. For p odd, Kedlaya's algorithm is used with several high-level efficiencies. For $p = 2$, the hyperelliptic Mestre/Lercier/Lubicz algorithm has been adapted in a similar fashion to the elliptic case to produce a fast implementation, particularly well suited to genus 2 and 3 curves. The order of the Jacobian of a genus 2 curve over $Gf(2^{80})$ is found in 1.39 seconds on a 2.1 GHz Athlon machine.

- *Hyperelliptic Curves:* Paul van Wamelen has written a package for computing with the analytic Jacobian of a hyperelliptic curve, that is, with the Jacobian as a complex torus. The analytic Jacobian is constructed in terms of the period matrix of the curve. Mapping of points between the algebraic and analytic Jacobians allows the user to obtain exact algebraic results from inexact analytic computations. The analytic Jacobian can also be used to find maps between different Jacobians including homomorphisms and specialisation such as isomorphisms, isogenies and the endomorphism rings.

- *Modular Abelian Varieties:* William Stein, who developed the packages for modular symbols and modular forms in Magma, has very recently constructed a major package for modular abelian varieties, which are viewed as explicitly given quotients or subvarieties of modular Jacobians. No explicit defining algebraic equations are used in these algorithms, so computations with abelian varieties of large dimension are feasible. The current machinery provides extensive functionality for computing with such abelian varieties, including functions for enumerating and decomposing modular abelian varieties, isomorphism testing, computing exact endomorphism and homomorphism rings, doing arithmetic with finite subgroups and computing information about torsion subgroups and special values of L-functions and Tamagawa numbers.

Incidence Structures

- *Symmetric functions:* The Symmetrica package, developed in Bayreuth by Adalbert Kerber and colleagues has been installed in Magma by Axel Kohnert. The package supports five bases: Schur functions, homogeneous symmetric functions, elementary symmetric functions, monomial symmetric functions and power sum symmetric functions. A wide range of operations are supported including arithmetic, transition matrices for base change, and plethysm. The Magma version incorporates many improvements devised by Kohnert and timings show that Magma is much faster for these operations than the other two symmetric function packages (ACE and SF).

- *Hadamard Matrices:* The previous algorithm used for isomorphism testing has been replaced by a much, much faster idea due to Brendan McKay, a new fast automorphism function based on the same idea is available. We have also constructed the first stage of a database of known Hadamard and skew Hadamard matrices of order up to 100.

- *Graphs:* In addition to vertex labels, the edges of simple graphs may now have either a label, a weight or a capacity. Standard shortest-path algorithms have been implemented for graphs whose edges are weighted.

- *Multigraphs:* Both directed and undirected multigraphs have been installed. They may have multiple edges and/or loops and are implemented using an adjacency list representation. Vertices and edges can be labelled, and edges can be assigned capacities and/or weights. A large number of standard elementary graph operations have been implemented for multigraphs. The following major algorithms are also available: shortest path, maximal flow, planarity, triconnected (the latter two operations involve linear-time algorithms).

Coding Theory

- *Linear Codes over Finite Fields:* An improved algorithm for minimum weight calculation has been introduced (M Grassl and G White). The tables of Best Known Linear Codes (BKLCs) over F_2 and F_4 have been updated to reflect the construction of many improved codes. For the first time a BKLC database over F_3 is released (constructed by M Grassl). The database over F_2 goes up to length 256, (100% filled), over F_3 up to length 100 (100% filled) and over F_4 of length up to 100 (over 99% filled).

- *Additive Codes over Finite Fields:* Given a finite field F and the space of all n-tuples of F, an additive code is a subset of $F^{(n)}$ which is a K-linear subspace for some subfield $K \subseteq F$. Additive codes are of current interest because of their applications to the construction of quantum error-correcting codes. In a new Magma module, basic constructions and cyclic codes are supported as are the important invariants such as weight distribution and weight enumerators. The Zimmermann minimum weight algorithm has been generalised to additive codes by Grassl and White.

3 Removals and Changes

This section lists the most important changes in Version 2.11. Other minor changes are listed in the relevant sections.

- The function `Ideal` for multivariate polynomials has been renamed to `IdealWithFixedBasis` to remove confusion.
- Finitely-presented algebras are now handled by a new type which supports noncommutative Gröbner bases.
- The large degree finite fields GF(2^k) are now constructed by low-term irreducibles (see below), if possible. To obtain the old sparse polynomials, specify the parameter `Sparse` when creating such a field.

4 Documentation

New chapters in the Handbook for V2.11 (with their chapter numbers) are:

- The Magma Profiler (6)
- Black-box Groups (23)
- Differential Rings and Operators (67)
- Finitely Presented Algebras (73)
- Representations of Symmetric Groups (78)
- Finitely Presented Lie Algebras (90)
- Hilbert Series of Polarised Varieties (95)
- Modular Abelian Varieties (104)
- Symmetric Functions (107)
- Multigraphs (109)
- Hadamard Matrices (112)
- Additive Codes (116)

The former chapters on graded rings and K3 surfaces have been removed. Their content appears in the new chapter *Hilbert Series of Polarised Varieties*.

5 Language and System Features [HB 1–6]

New Features:

- A profiler for the Magma language is now included, which gives users the ability to pinpoint performance bottlenecks in their code. The profile graph is accessible as a Magma graph structure, allowing the user to perform any analysis on the data necessary. Many commonly used analyses are provided, and a link into the graph visualization software GraphViz has also been implemented.
- Indexed types have been implemented, allowing more finely grained control over intrinsic selection. For example, the user can now specify an intrinsic which only accepts polynomials over integers as arguments, instead of polynomials over any arbitrary ring. The intrinsic matching logic has been redesigned so that the user also has the ability to override system intrinsics if desired.
- A Windows 2000/XP kernel driver is now provided which adds several missing features to the Windows port: Control-C handling and the Alarm function. With this kernel driver, the Windows port now has no limitations when compared to other platforms.

6 Groups

6.1 Finite Groups [HB 17]

New Features:

- The `IsIsomorphic` intrinsic has been improved by Derek Holt. This may now be applied across three types of finite groups, to permutation, finite matrix and finite soluble groups.
- The stored data necessary for the insoluble composition factors that may be encountered in group structure algorithms has been extended. The list in the database is: All simple groups of order less than 1.6×10^7, plus M_{24}, HS, J_3, McL, $Sz(32)$ and $L_6(2)$. There are also special routines to handle numerous other groups. These include: A_n for $n \leq 999$, $L_2(q)$, $L_3(q)$ and $L_4(q)$ for all q, $L_5(p)$, $S_4(p)$ and $U_3(p)$ for all primes p, $L_d(2)$ for $d \leq 14$, $U_6(2)$, $S_8(2)$, $O_8^{\pm}(2)$, $O_{10}^{\pm}(2)$, $S_6(3)$, $O_7(3)$, $G_2(4)$, $G_2(5)$, ${}^3D_4(2)$, ${}^2F_4(2)'$, Co_2, Co_3, He, Fi_{22}.
- There is a new intrinsic `ElementaryAbelianSeriesCanonical(G)`, where G is a permutation, finite matrix or finite soluble group. This produces a sequence $[S_1, S_2, \ldots S_n]$ where S_1 is the soluble radical of G, $S_n = 1$, and each quotient S_i/S_{i+1} is elementary abelian, which is canonical in the sense that any group isomorphism $G \to H$ will take `ElementaryAbelianSeriesCanonical(G)` to `ElementaryAbelianSeriesCanonical(H)`.
- The handling of subgroup relationships now uses new methods which are much faster when there are many subgroups of a group to be dealt with.
- Black-box groups have been added to support the development of algorithms for black-box recognition of groups.
- Non-constructive recognition of simple groups using a Monte-Carlo algorithm is now available. The method was developed by G. Malle and E. O'Brien and incorporates the algorithm of Babai et al.
- The Kantor-Seress algorithm for black box recognition of linear groups has been installed.
- The Bratus-Pak algorithm for black box recognition of the alternating and symmetric groups has been implemented by Holt.
- The Babai et al algorithm for black-box recognition of the alternating and symmetric groups has been implemented by Roney-Dougal.
- Much of the information in the online Atlas of Simple Groups maintained by Robert Wilson at Birmingham is now available as a Magma database. In particular, the list of ordinary and modular representations is available.

6.2 Permutation Groups [HB 18]

Changes:

- The semantics commands `Verify` and `RandomSchreier` now agree with the same named commands for matrix groups. That is, `RandomSchreier(G)` asserts the group order to be whatever the random schreier run ends with. On the other hand, the `Verify` command ignores any knowledge of the groups BSGS being complete and runs a verification algorithm. WARNING: RandomSchreier may assert an incorrect group order, and, if this happens, any future calculations with this group can go badly wrong. (There may be no sign of problems, there may be a crash, or an infinite loop for example.)

New Features:

- Magma's implementation of the Brownie-Cannon-Sims Verify routine has been upgraded to make use of block systems in the basic orbits and knowledge of a known base for the BSGS being verified. This has greatly improved both `Verify` and `FPGroupStrong` (which is based on this algorithm).
- Construction of conjugacy classes has been improved by revising the extension algorithm for lifting conjugacy classes through the soluble radical. It now uses less memory and runs faster.
- A version of Greg Butler's "Inductive" algorithm for computing conjugacy classes is now installed. This may be invoked using the intrinsic `ClassesInductive`, or by setting parameters `Al` or `TFAl` to `"Inductive"` in the standard `Classes` intrinsic. (The inductive algorithm is not yet part of the default strategy.)
- A new algorithm for computing `DoubleCosetRepresentatives` has been implemented. This refines double cosets along a subgroup chain using orbit-stabilizer methods. The intrinsic `SubgroupChain` gives access to the method used to construct the subgroup chain.
- Subgroup conjugacy and normalizer calculations now use a pre-processing phase where there is a reduction by the orbit structure of the subgroups.
- A function has been provided to determine whether a permutation group of moderate degree can be written as a permutational wreath product `IsWreathProduct`.

6.3 Matrix Groups over Finite Fields (Aschbacher Analysis) [HB 19]

New Features:

- The algorithm of Roney-Dougal for testing conjugacy of subgroups in $GL(n, q)$ has been implemented (`IsGLConjugate`).
- Magma can now determine whether a matrix group modulo scalars has an equivalent representation over a smaller field using an algorithm due to Leedham-Green and O'Brien.
- A matrix group may be tested for being perfect without computing a base and strong generating set by a Monte-Carlo algorithm due to Leedham-Green and O'Brien.
- A function is available to recognize if a group of odd prime degree is contained in the normaliser of an extra-special group. This uses an algorithm of Niemeyer and was implemented in Magma by O'Brien.

6.4 Finite Soluble Groups [HB 20]

Changes:

- Homomorphisms to other group types have been revised to give more flexibility in this area. It is now possible to define homomorphisms by specifying images of an arbitrary generating set of the domain, where before the defined generators had to be used.

New Features:

– The `SmallGroups` commands will now return all groups of order p^4, p^5 or p^6. These groups were contributed by Boris Girnat, Robert McKibbin, M.F. Newman, E.A. O'Brien, and M.R. Vaughan-Lee.

Bug Fixes:

– A bug in the implementation of the collection algorithm has been fixed.

6.5 Databases of Groups [HB 26]

New Features:

– Portions of the online ATLAS of Finite Group Representations are now supplied with Magma. The basic access function is called `ATLASGroup`. This allows access to representations of various "nearly simple" groups as matrix and permutation groups

– All groups of order p^4, p^5, p^6 for any prime p are now available as part of the SmallGroup database. These groups were contributed by Boris Girnat, Robert McKibbin, M.F. Newman, E.A. O'Brien, and M.R. Vaughan-Lee.

– The almost simple groups database now holds information on all simple groups of order less than 1.6×10^7, plus M_{24}, HS, J_3, McL, $Sz(32)$ and $L_6(2)$. Maximal subgroups and automorphism groups can be computed for many more groups, including the simple groups: A_n for $n \leq 999$, $L_2(q)$, $L_3(q)$ and $L_4(q)$ for all q, $L_5(p)$, $S_4(p)$ and $U_3(p)$ for all primes p, $L_d(2)$ for $d \leq 14$, $U_6(2)$, $U_3(9)$, $S_8(2)$, $O_8^{\pm}(2)$, $O_{10}^{\pm}(2)$, $S_6(3)$, $O_7(3)$, $G_2(4)$, $G_2(5)$, ${}^3D_4(2)$, ${}^2F_4(2)'$, Co_2, Co_3, He, Fi_{22}. The code for the families of groups is by Derek Holt and Colva Roney-Dougal.

– A database of fundamental groups of small-volume closed hyperbolic 3-manifolds is now available. The data was prepared by Dunfield & Thurston, based on the manifolds in the Hodgson-Weeks census. The groups are infinite finitely presented groups. The basic access operations are `ManifoldDatabase` and `Manifold`.

6.6 Braid Groups [HB 31]

New Features:

– The intrinsic `Generators` has been provided to return the generators of a group for the current presentation or a given presentation.

– The intrinsics `UltraSummitRepresentative` and `UltraSummitSet` return an element in the ultra summit set of a given element and the whole ultra summit set of a given element respectively. To determine if a given element is contained in its ultra summit set use `IsUltraSummitRepresentative`.

– The process for constructing the ultra summit elements of a given element can be obtained using `UltraSummitProcess`.

– Minimal simple elements result from the new intrinsics `MinimalElementConjugatingToPositive`, `MinimalElementConjugatingToSuperSummit` and `MinimalElementConjugatingToUltraSummit`.

– A simple element and a minimal element can be gained using `Transport` and `Pullback` of an element contained in its super summit set and a simple element.

7 Basic Rings

7.1 Integer Ring [HB 37]

New Features:

- Hexadecimal notation is now supported when reading or writing integers.
- The database of Cunningham factorisations has been updated to the latest version.

7.2 Real and Complex Fields [HB 42]

Removals and Changes:

- The range of inputs to the intrinsics `Eisenstein` and `WeierstrassSeries` has been refined. The precision argument has been made into a parameter in the case of series and removed otherwise. The result from `Eisenstein` will be in a series ring over the field of fractions of the input coefficient ring if one exists and will give an error otherwise if division by zero occurs.
- A new algorithm for the `DedekindEta` function is used which evaluates a sum (as described by Cohen) rather than a product (as was done by the former algorithm). The algorithm for computing `Delta` has been switched so as to use the `DedekindEta` function rather than the the `Eisenstein` function. There have been significant improvements in speed in both cases. It is no longer necessary for a series to be over a field for `Delta` to be applied and the precision argument has been removed.

New Features:

- The classical multivariate θ function has been implemented for column matrices c and z and an element of the Siegel upper half-space.

Bug fixes:

- A bug in each of `Eisenstein` and `WeierstrassSeries` has been fixed.
- A bug in `JacobiTheta` when zero is given as the first argument has been fixed.

7.3 Multivariate Polynomial Rings [HB 41]

Factorization of bivariate polynomials over all supported rings is now accomplished by a new algorithm which extends van Hoeij's knapsack ideas for $\mathbf{Z}[x]$ to solve the hard combination problem for $\mathrm{GF}(q)[x, y]$. The new algorithm runs in polynomial time and performs extremely well in practice. General multivariate factorization is reduced to this new bivariate algorithm, so a combination problem never arises for any number of variables. Shoup's tree Hensel lifting algorithm has also been adapted for power series, making the lifting stages of all kinds of bivariate/multivariate factorization much faster than previously.

Most of the algorithms for multivariate GCD and resultant computation have also been rewritten from scratch. In particular, a faster interpolation algorithm is used for resultants, and there are several new heuristics.

New Features:

- New polynomial-time bivariate factorization algorithm based on generalization of van Hoeij's Knapsack factoring algorithm.
- The algorithms for multivariate factorization have been rewritten from scratch. These includes several new heuristics and are based on the new bivariate algorithm.
- New fast algorithms for multivariate GCD and resultants.

7.4 Finite Fields [HB 39]

Changes:

- Large degree irreducible polynomials over GF(2) (used, for example, when constructing GF(2^k)) are now low-term irreducibles (see below), if possible. To obtain the old sparse polynomials, specify the parameter `Sparse` when creating such a field.

New Features:

- A database of low-term irreducible polynomials over GF(2) (of the form $x^n + g$ where g is of minimal degree) has been constructed up to degree 23000.
- New functions `IrreducibleLowTermGF2Polynomial` and `IrreducibleSparseGF2Polynomial`.

8 Linear Algebra and Module Theory

8.1 Matrices [HB 44]

New Features:

- `lt`, `le`, `gt`, and `ge` are now available for matrices and vectors.
- New function `ZeroMatrix(R, m, n)`.

9 Commutative Algebra

9.1 Ideal Theory and Gröbner Bases, Affine Algebras [HB 49, 50]

Magma now contains the most powerful algorithms available for computing Gröbner bases. Further, in Magma GB computation is available for polynomial rings over all exact fields, for polynomial rings over euclidean rings, for finitely presented associative algebras and for finitely presented Lie algebras.

V2.11 includes a new highly optimized implementation of the Faugère $F4$ algorithm for computing Gröbner bases (GBs) over fields. The algorithm uses sparse linear algebra and has specialized vector representations for all types of finite fields, and an asymptotically-fast modular algorithm for the rationals. Special techniques are also used for solving systems over GF(2).

As an example, the Cyclic-7 `grevlex` GB is computed in 2.2 seconds on an Athlon 2800+ XP PC. We have also computed the Cyclic-9 `grevlex` GB over the rationals in 7.5 days on a 750MHz Sunfire v880 (one sequential processor, 11GB memory used). As far as we are aware, no-one else has successfully computed this Gröbner basis except for J.-C. Faugère (in 18 sequential days on 4 processors). More timings can be found on the webpage `http://magma.maths.usyd.edu.au/users/allan/gb/`.

Removals and Changes:

- The function `Ideal` for multivariate polynomials has been renamed to `IdealWithFixedBasis` to remove confusion (used when specifying a sequence of polynomials with respect to which the `Coordinates` function will return its answer).

New Features:

- New Faugère $F4$ algorithm over fields and new parameters for the `Groebner` procedure.
- Intrinsics `EquidimensionalDecomposition` and `FineEquidimensionalDecomposition` have been added for ideals with no embedded primary components. These are generally faster than primary or radical decompositions.
- Intrinsic `NoetherNormalization` added to give a linear Noether normalization of the quotient algebra of an ideal.
- Intrinsic `Normalization` added to give the integral closure of the quotient algebra of an ideal in its total ring of fractions. For ideals in rank 2 polynomial rings, Magma's function field machinery is used. For higher rank rings, a general algorithm of de Jong is used. The base field of the polynomial ring must be perfect.

10 Extensions of Rings

10.1 Algebraic Number Fields [HB 52]

Changes:

- The checks on the creation of homomorphisms from number fields and fields of fractions of orders have been increased. The codomain of the homomorphism must cover the coefficient ring of the domain if a coefficient map is not given.
- Norm equations over number fields have been completely rewritten including a new, stringent precision analysis for most of the algorithm.
- Element arithmetic has been sped up.

New Features:

- It is now possible to create a `VectorSpace` or `Algebra` representing any algebraic field (not just cyclotomics) over any subfield.
- The map from a number field into a vector space or algebra can be extended to vector and matrix spaces over the number field into a larger dimensional space over a subfield using `InducedMap`.
- Elements can now have product representations.

10.2 Algebraic Function Fields [HB 59]

In this release a new representation of algebraic function fields is provided — non-simple extensions. Algebraic function fields can be created using more than one polynomial and still be a direct extension of the field which was being extended. Fields in this new representation have almost the same level of functionality as their representations as rational extensions. As in the case of the relative representations, Galois groups, series functions and subfields are not yet available. Some functions involving differentials also are not yet available.

Removals and Changes:

- The second argument to `Completion` is now a place rather than an ideal.
- For clarity the names of functions beginning `NumberOfPlaces` have had `OverExactConstantField` appended where applicable. Shorter synonyms have also been provided.
- The expansion of a function field element at a place has been improved to run substantially faster.
- `Poles` and `Zeros` of a function field element have been rewritten more efficiently.
- Element arithmetic has been made more efficient.
- Powering of elements in characteristic p rings and fields has been improved using the fact that $(a+b)^p = a^p + b^p$.

New Features:

- It is now possible to take non–simple extensions of function fields. Such extensions have all the functionality of simple extensions of function fields, except that functions involving series rings, galois groups, subfields and some functions involving differentials are not yet available.
- An order of a function field can be set to be either maximal or non–maximal using the intrinsic `SetOrderMaximal`.
- The calculation of Automorphism groups has been directly provided in `AutomorphismGroup`.
- A function field can be created from an existing function field by extending the constant field.
- Orders of function fields can be created by supplying an order and a sequence of elements of a function field. The order must be a maximal order of the coefficient ring or an order of the field containing the elements whose coefficient ring is maximal.

Bug Fixes:

- A bug in coercion of divisors into divisor groups over an extension of the original field has been fixed.

10.3 Newton Polygons [HB 62]

Removals and Changes:

- Newton polygons created from a polynomial and a prime will now recognise their creation as being from a polynomial and as such will return a `Polynomial` and have `FaceFunctions`. The default faces are consequently the inner faces.

- ValuationsOfRoots replaces NewtonSlopes. This function returns as a sequence of tuples, the valuations of the roots of the polynomial and the number of roots with that valuation.

New Features:

- A parameter Faces can be given to NewtonPolygon to determine which faces will be returned by Faces.
- A polygon can be constructed from a polynomial and a place of an algebraic function field.
- In addition to Slopes, this version makes available the functions InnerSlopes, LowerSlopes, OuterSlopes and AllSlopes are available.

10.4 p-adic Rings and Fields [HB 63]

New Features:

- The speed of many operations related to point counting has been greatly improved.
- Type 1 and 2 Gaussian Normal Bases have been implemented. This provide an essentially free Frobenius action without sacrificing the speed of multiplication.

10.5 Lazy Series Rings [HB 66]

Removals and Changes:

- It is no longer possible to multiply a univariate series by a monomial using $s * n$ where s is a series and n is an integer used as the exponent of the monomial. However, it is still possible to compute this product using $s * [n]$.

11 Differential Rings

11.1 Differential Rings (New) [HB 67]

The Galois theory of linear differential equations is the analogue for linear differential equations of the classical Galois theory of polynomial equations. The natural analogue of the field in the classical case is the differential field which is a field equipped with a derivation. We have undertaken to construct a basic facility for differential fields and rings with the medium term goal of constructing a fast solver for linear differential equations.

Differential rings are formed by adding the functionality of a derivative to an ordinary ring in Magma. Additional functionality is available for rational and algebraic function fields. Differential rings can be used to create differential operators and in a wider perspective to consider topics related to differential galois theory.

New Features:

- Differential rings can be created from any ring given a map from the ring to itself and a specified constant field. A special case is that of the rational function field with the usual derivative. These differential rings look like the ring they are created from and inherit all the functionality of that ring and similarly for their elements.
- The ring underlying a differential ring and the derivation can both be retrieved. Differential rings can be tested for equality and a few other properties.
- Arithmetic of elements, as available in the underlying ring, can be performed. In the same way, elements can be tested for equality and for being one or zero. Of course, elements can also be differentiated with respect to the derivation.
- A differential ring can be created from an existing differential ring by changing the derivation or extending the constant field.
- Differential rings can be extended. The permissible extensions include algebraic, exponential, logarithmic and by adjoining the formal solutions of a linear differential operator.
- Ideals and quotients of differential rings can be formed.
- The Wronskian matrix of a sequence of differential ring elements can be computed.
- Rings of differential operators can be created over any differential ring and that differential ring can be retrieved from the operator ring as well as its derivation. Differential operator rings can be compared to determine equality.
- The usual ring element arithmetic can be performed on differential operators. They can be tested for equality and for being one, zero or monic. Elements can be deconstructed into a sequence of terms and the coefficients of monomials can be extracted. The order of a differential operator can be accessed. An monic differential operator can be formed from a given operator as well as an adjoint operator.
- Operators can be applied to elements of the underlying differential field.
- Euclidean left and right division and GCDs and LCMs of operators is available.
- The companion matrix to a differential operator can be created.
- There are functions which examine the interaction between differential operators and places of the underlying differential field.
- Rational solutions of linear differential equations $L(y) = 0$ and $L(y) = g$, where g is an element of the underlying differential field of the operator L, can be calculated.
- The Newton polygon of an operator at a place can be created and the newton polynomial of a face on this polygon computed.
- Symmetric powers of operators can be taken and operators can be created with the formal roots of a given polynomial as solutions.

12 Algebras

12.1 Finite Dimensional Algebras [HB 68]

New Features:

- The Jacobson radical of associative algebras over rings of characteristic zero can be computed.

12.2 Matrix Algebras [HB 72]

Changes:

- The `JacobsonRadical` of a matrix algebra over a field of characteristic zero has been reviewed and the result is a significant speed up. This intrinsic is now also available for algebras over all fields of characteristic zero instead of only the rational field.

12.3 Finitely Presented Associative Algebras (New Version) [HB 73]

In V2.11 finitely-presented (FP) associative algebras are handled by an extension of the commutative algebra machinery to noncommutative data structures and algorithms, where applicable. These include a noncommutative analogue for Gröbner bases.

Features:

- Construction of free algebras over arbitrary fields.
- Arithmetic.
- Mappings into other associative algebras.
- Definition of left, right, two-sided ideals.
- Noncommutative Gröbner bases of ideals, with specialized algorithms for different coefficient fields (fraction-free methods for the rational field and rational function fields).
- Gröbner bases of ideals over finite fields and rationals, using noncommutative extension of the Faugère $F4$ algorithm.
- Construction of degree-d (truncated) Gröbner bases.
- Normal form of a polynomial with respect to an ideal.
- Construction of FP-algebras as quotient rings.
- Enumeration of the basis of finite-dimensional FP algebras.
- Matrix and structure-constant representations of finite-dimensional FP algebras.
- Construction of a matrix representation (Linton's vector enumerator).

13 Representation Theory

13.1 Modules over Matrix Algebras [HB 74]

New Features:

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

13.2 Representations of Symmetric Groups (New) [HB 78]

New Features:

- Integral representations of a an elements of the symmetric group can be computed using one of three different algorithms. Seminormal and orthogonal representations can also be computed.
- The value of a symmetric group character (defined by a partition) on a specified permutation can be calculated.
- A single character or all characters of the symmetric group can be computed.
- The value of an alternating group character (defined by a partition) on a specified permutation can be calculated.
- A single character or all characters of the alternating group can be computed.

14 Lie Theory

14.1 Groups of Lie Type [HB 89]

New Features:

- Highest weight representations (previously available only for simply-connected semisimple groups).
- Adjoint representations.
- Decomposition of automorphism (see Carter).
- Functions for computing the centre of a groups of Lie type.
- Extension of the base field.
- Coercions for Coxeter groups and groups of Lie type.

14.2 Finitely Presented Lie Algebras (New) [HB 90]

Very efficient machinery has been developed for constructing a Gröbner basis for a finitely presented Lie algebra L by Willem de Graaf and Allan Steel. The GB reduction algorithm used is Magma's generic $F4$ algorithm. At present, if the algebra L is finite dimensional, the GB can be used to construct a Lie algebra defined by structure constants. Alternatively, the GB may be used to construct a nilpotent quotient of L to a designated class.

15 Algebraic Geometry

15.1 Schemes [HB 92]

New Features:

- Intrinsic `ArithmeticGenus` added for ordinary projective varieties.
- To work with the tangent and secant varieties of schemes, `TangentVariety`, `SecantVariety`, `IsInTangentVariety` and `IsInSecantVariety` have been added.
- Intrinsic `IsomorphicProjectionToSubspace` added. This finds a sequence of point projections of a projective scheme of dimension d down to a $2d+1$ linear subspace of its ambient space such that the combined projection is an isomorphism of the scheme onto its image.
- Intrinsic `EmbedPlaneCurveInP3` added. This gives a non-singular projective plane or (3D-)space curve birationally equivalent to the original plane curve.
- The intrinsic `IsAmbient` has been provided for schemes.

15.2 Algebraic Curves [HB 93]

Removals and Changes:

- For clarity the names of functions beginning `NumberOfPlaces` have had `OverExactConstantField` appended where applicable. Shorter synonyms have also been provided.
- For consistency (with function fields) `RiemannRochBasis` of a divisor has been renamed to `Basis`.

15.3 Hilbert Series of Polarised Varieties (New) [HB 95]

New Features:

- The K3 database now contains 24,099 objects (an increase from around 400 in previous versions). In a precise sense, this now represents all possible (infinitely many) Hilbert series that may be associated to a polarised K3 surface.
- The machinery used to generate the database has been lifted to the user level. This includes new datatypes for (the numerical data associated to) polarised singularities and polarised varieties.
- There is a small database of Fano 3-folds. A major point of this work has been to contribute to the (hopefully imminent) classification of Fano 3-folds. This small database is the first step of what may become the comprehensive classification of (Hilbert series of) Fano 3-folds.
- New Riemann–Roch formulas used to compute Hilbert series have been included. This allows one to make graded ring calculations for K3 surfaces, Fano 3-folds, Calabi–Yau 3-folds and curves polarised by a subcanonical divisor. It is possible for a user to generate many families of these varieties interactively—a typical application of these methods is to generate large numbers of examples.
- A facility for users to write and read their own graded ring databases has been added. This makes genuine binary data files together with various indexing methods, as is usual with Magma databases, and it is essential when one generates large numbers of examples.

16 Arithmetic Geometry

16.1 Rational Curves and Conics [HB 96]

New Features:

- The new algorithm of D. Simon for reducing indefinite quadratic forms has been implemented and is now used for finding rational points on conics. In particular, Simon's algorithm is used for the intrinsic `HasRationalPoint`. The Simon algorithm supplements the one previous used (Cremona) and works best on non-diagonal conics whose diagonal form involves integers that are very expensive to factorize.

16.2 Elliptic Curves [HB 97]

Removals and Changes:

- The intrinsic `Periods` has been rewritten.
- The intrinsic `IsogenyMu` has been renamed to `CasselsMap`.
- There are now more parameters to `SelmerGroup` and return values depending on these.
- `RankBound` now takes an elliptic curve and an optional map.

16.2.1 Elliptic Curves over the Rational Field

New Features:

- The twist of a curve with minimal discriminant is computed by the intrinsic `MinimalTwist`.
- All curves isogenous to a given curve over the rationals can be collected by calling `IsogenousCurves`.
- Local and global root numbers can now be gained using `RootNumber`, `AnalyticRank` returns the analytic rank of a curve and `ModularDegree` returns the modular degree.
- The `EllipticExponential` of a complex number can be taken. `EllipticLogarithm` has been extended to complex points.
- A number of intrinsics implementing a four-descent algorithm for elliptic curves are being released for the first time. The main intrinsic is `FourDescent`. Others are `BadPrimes` and `Points`.
- The intrinsic `HeegnerPoint` will find a generator on an elliptic curve.
- To compute the Cassels pairing of two group elements of a 2-Selmer group `CasselsPairing` has been implemented.
- A database of over 100 million elliptic curves over **Q** with conductors up to 10^8 has been constructed.

16.2.2 Elliptic Curves over an Algebraic Number Field

New Features:

- For curves over a number field the functions `Conductor`, `LocalInformation` and `MinimalModel` have been added.
- `TwoSelmerGroup` of elliptic curves has been added.
- `PseudoMordellWeilGroup` and `Chabauty` are now provided.
- `RationalPoints` is now available for curves over number fields.
- The `Height` of points on elliptic curves over number fields can be determined.

16.2.3 Elliptic Curves over an Algebraic Function Field

New Features:

- The `Height` of points on elliptic curves over function fields can be determined.

16.2.4 Elliptic Curves over Finite Fields

New Features:

- Canonical lift algorithm for point counting on curves defined over fields of characteristic two.
- Helper functions `CryptographicCurve` and `ValidateCryptographicCurve` have been added for the generation of cryptographically secure elliptic curves.

16.3 Hyperelliptic Curves [HB 98]

16.3.1 Hyperelliptic Curves

Removals and Changes:

- Intrinsics such as `IspIntegral`, `IspNormal`, `IspMinimal`, `pIntegralModel`, `pNormalModel` and `pMinimalWeierstrassModel` which used to take a "rational place" now take the prime instead.

New Features:

- `RationalPoints` is now available for curves over number fields.

16.3.2 Jacobians of Hyperelliptic Curves

New Features:

- `TwoSelmerGroup` has been added.
- Fast p-adic point-counting methods (Mestre/Lercier/Lubicz in characteristic 2 and Kedlaya in odd characteristic) have been added. These apply to both a hyperelliptic curve C and its Jacobian J over finite fields of small characteristic. Each of the four functions `#C`, `#J`, `EulerFactor(J)`, `ZetaFunction(C)` will use these methods when appropriate.
- Analytic Jacobians can now be created. Their big and small period matrices can be accessed as well as a selection of other invariants. There are several constructions for maps between Jacobians. Voronoi cells can be computed.

16.4 Modular Forms [HB 103]

Removals and Changes:

- `Slopes` now returns a sequence of tuples of slopes and their multiplicities.

16.5 Modular Abelian Varieties (New) [HB 104]

This is a new package that offers some nontrivial functionality for modular abelian varieties, which we view as explicitly given quotients or subvarieties of modular Jacobians. No explicit algebraic defining equations are used in these algorithms, so computations with abelian varieties of large dimension are feasible. Some highlights of the package include:

- Construction of quite general modular abelian varieties, in the sense that arbitrary finite direct sums and quotients may be formed.
- Explicit computation of the group $\mathrm{Hom}(A, B)$ or the ring $\mathrm{End}(A)$, as a subgroup of homology, for modular abelian varieties A, B over $\mathbf{Q}$.
- Computation of kernels, cokernels, and images of homomorphisms of abelian varieties.
- Intersections of subvarieties.
- Computation of discriminants of subgroups of endomorphism rings, such as Hecke algebras.
- A divisor and a multiple of the order of the K-rational torsion subgroup of A.
- The determination of whether or not two modular abelian varieties are isomorphic (in some cases).
- Characteristic polynomial of Frobenius.
- Tamagawa numbers and component group orders (in some cases).
- Computation of all inner and CM twists (not provably correct).
- Computation with torsion points as elements of rational homology.

17 Incidence Structures

17.1 Symmetric Functions (New) [HB 107]

New Features:

- Algebras of symmetric functions can be created with any one of five possible bases: a basis consisting of Schur functions, Elementary, Monomial, Homogeneous or Power Sum functions.
- Elements of algebras of symmetric functions can be created as linear combinations of basis elements (indexed by partitions) or from coercing a polynomial or a scalar.
- Algebras of symmetric functions are rings so there are a number of ring predicates which are available for these algebras also.
- The print style of symmetric functions can be altered using an attribute on the algebra.
- Symmetric functions can be added, subtracted, multiplied and composed (plethysm). They can be tested for homogeneity and equality.
- Symmetric functions can be decomposed into a sequence of basis elements and coefficients thereof. A coefficient corresponding to the basis element of a given partition can be directly accessed. The number of basis elements with non zero coefficient in an element and the degree of an element can also be determined.
- Symmetric functions can be coerced into polynomial rings.
- A Frobenius homomorphism may be applied to symmetric functions. Inner products of symmetric functions can be taken. A set of tableaux for which a Schur function is the generating function can be retrieved and the character of the symmetric group corresponding to a symmetric function can be created.
- Matrices converting from any of the five bases to any other of the five bases can be calculated.

17.2 Graphs [HB 108]

With the introduction of multigraphs, the conventions for labelling edges in simple graphs were extended so that they are consistent across the two graph types.

Changes:

- Edges of a simple graph can be assigned a capacity and/or weight, in addition to a label.
- Flow-based algorithms and shortest-paths algorithms apply to simple graphs (which may be capacitated and/or weighted).
- Graph equality and subgraph testing now take account of the structure of the graph and, if applicable, of the graph support, the vertex/edge labels, and the edge capacities.
- Incremental graph construction (using the `sub<>` constructor, or by adding/removing vertices/edges) retains the support and vertex/edge decoration of the original graph.

New Features:

- Shortest path computation between two vertices in a (weighted) graph.
- Existence of a negative weight cycle in a (weighted) graph.
- All-pairs shortest paths computation.
- Minimum weight tree rooted at some vertex.

17.3 Multigraphs (New) [HB 109]

Multigraphs, both directed and undirected, appear in Magma for the first time. They may have multiple (ie. parallel) edges and/or loops. Multigraphs are represented by means of an adjacency list. In what follows, the generic term "multigraph" will be used whenever we discuss directed and undirected multigraphs.

New Features:

- There is a larger upper-bound on the order of a multigraph (134217722) than on the order of a graph (65535); this made possible by the adjacency list representation.
- Edges are uniquely identified, and one can determine the multiplicity of an edge between any two vertices.
- Vertices and edges can be labelled, and edges can be assigned capacities and/or weights.
- Most basic access functions and predicates that apply to simple graphs are available for multigraphs.
- Maximum flow and shortest-paths algorithms may be applied to a multigraph.
- It is possible to determine if a multigraph is planar, or triconnected (both operations involve linear-time algorithms).
- Incremental graph construction (using the `sub<>` constructor, or by adding/removing vertices/edges) retains the support and vertex/edge decoration of the original graph.

17.4 Hadamard Matrices (New) [HB 112]

New Features:

- A database of Hadamard matrices has been added. The database is created by `HadamardDatabase` and its entries are accessed using `HadamardMatrix` and `HadamardMatricesOfDegree`. Information concerning the database can be accessed through `Degrees`, `NumberOfMatricesOfDegree` and `DegreeRange`.

18 Coding Theory

18.1 Linear Codes over Finite Fields [HB 115]

New Features:

- The Best Known Linear Codes package now includes a database of codes over $GF(3)$. This database is a contribution of Markus Grassl, Karlsruhe. The database is completely full up to length 100, providing $5,150$ codes. The codes of length up to 21 are optimal. The database over $GF(4)$ is now over 99% complete, with only 40 of the $5,150$ codes missing.
- New packing methods, developed by Greg White, have resulted in faster vector addition for small non-binary finite fields. Vector enumeration is the core operation in almost all important computations in coding theory, such as calculating the minimum weight, the weight distribution, or the complete weight enumerator. For the finite fields $GF(3)$, $GF(4)$ and $GF(8)$ there has been a factor of $3-4$ scalar speed up for all of these computations.

- The important minimum weight algorithm has been improved in various ways. This includes the use of a finer incrementing of the lower bound, which allows some computations to finish several steps earlier then before. Since the bulk of the computation is done in the final steps this causes a huge speed up for certain cases. Also, revolving door algorithms have been used in the vector enumeration resulting in about 20% reduction in the number of vector operations.
- The minimum weight algorithm has a new verbose output with several improvements. A useful new feature is an accurate prediction of not only the stage, but also the time at which the algorithm will finish.

18.2 Additive Codes over Finite Fields (New) [HB 116]

A new package for additive codes has been included in this release. Given a finite field F and the space of all n-tuples of F, an *additive code* is a subset of $F^{(n)}$ which is a K-linear subspace for some subfield $K \subseteq F$. As well as the basic constructions, the important class of cyclic codes are available. All significant weight distribution invariants are installed,

New Features:

- Invariants of an additive code, such as ambient space, alphabet, various numerical invariants, the code and dual space, can be accessed.
- Operations on code words can be carried out, e.g. arithmetic, distance and weight.
- Subcodes can be formed.
- An adaptation of the minimum weight algorithm for linear codes has been developed by Markus Grassl and Greg White for computing the minimum weight of additive codes. The runtimes for the computations are comparable to linear cods of the same cardinality.
- Constructions are provided to produce new codes from existing codes.