Summary of New Features in Magma V2.7

June 30, 2000

1 Introduction

This document provides a terse summary of the new features installed in Magma for release version V2.7 (June 30, 2000). Previous releases of Magma are: V2.6 (November 8, 1999), V2.5 (July 7, 1999), V2.4 (December 3, 1998), V2.3 (January 30, 1998), V2.20 (April 18, 1997), V2.10 (October 14, 1996), V2.01 (June 21, 1996), and V1.30 (March 5, 1996); release notes for these versions are found in the document `relprev.dvi`.

All timings given below are for a Sun 400Mhz UltraSPARC unless otherwise indicated.

2 Summary

Algebras

- A module for quaternion algebras has been implemented. Special constructions are implemented for orders over $\mathbf{Z}$, with given ramification and index in a maximal order. The left and right ideal arithmetic of orders in definite quaternion algebras over $\mathbf{Q}$ is treated, permitting enumeration of all one-sided ideal classes.

Algebraic Geometry

- A new package for computing with hyperelliptic curves and their Jacobians has been implemented. (Developed by Michael Stoll [Düsseldorf], with revisions and kernel support by members of the Magma group.)

- A major new initiative in modular forms and modular curves has been undertaken. As the first step, a Magma version of William Stein's Hecke package for computing with modular symbols has been implemented. (Developed by William Stein [Berkeley].) An alternative approach via Brandt modules associated to the ideal theory of quaternion orders has been implemented by David Kohel of the Magma group.

- Elliptic curves can now be created from a plane curve of genus one with given rational point. Functions for transformations of genus one hyperelliptic curves and elliptic curves allow for easy conversion between categories.

- The functionality for computing with elliptic curves has been expanded, including new functions for computing S-integral points, Weil pairings, and group structure over finite fields, and improved algorithms for height and rank computations over $\mathbf{Q}$.

- Resolution graphs and splice diagrams have been introduced as a means of encoding data generated by the resolution machinery. The package includes a resolution function for curves adapted to present its output in resolution graph format.

Coding Theory

- The advanced Zimmermann algorithm for minimum weight computation has been implemented. Several other new coding theory functions have been installed.

Commutative Algebra

- A major revision of the multivariate polynomials module has been achieved. This includes a new internal random-access array representation, a fraction-free coefficient representation and new variable-size monomial representation, yielding improvements in time and space usage. Several other improvements have been done and many bugs and leaks have also been fixed.

- Affine algebras have been expanded in their functionality considerably. Further support has been added for the case for when affine algebras are fields.

Groups

- A new category has been created for the family of groups defined by a polycyclic presentation. Note that such a group may be infinite. Algorithms for element arithmetic and (some) subgroup computations analogous to those for finite soluble groups have been implemented.

- The Smith algorithm for computing the automorphism group of a finite soluble group has been implemented (with improvements) by Mike Slattery.

Incidence Structures

- Categories for Incidence Geometries (in the sense of F. Buekenhout) and Coset Geometries have been implemented by Dimitri Leemans.

Linear Algebra and Module Theory

- A new feature that allows the creation of a matrix without having to predefine its parent structure has been introduced. A matrix may be expressed in a broad range of different formats. This is expected to make working with matrices a great deal easier. In addition, an extensive range of functions for creating matrices having a special structure (block, sparse, scalar, diagonal) has been provided. A new chapter in the Handbook brings together all the operations available for matrices.

- The existing facilities for $K[G]$-modules have been generalised so that the great majority of them apply to modules defined over a (matrix) algebra. As part of the restructuring, this class of modules is described in a new chapter that forms the first chapter in a new series (Handbook Part) relating to Representation Theory.

Rings, Fields and Orders

- A new system has been developed for computing with algebraically closed fields, which have the property that they always contain all the roots of any polynomial defined over them. One can now compute the variety of any zero-dimensional multivariate polynomial ideal over the algebraic closure of its base field.
- The new MONSTER random number generator of G. Marsaglia has been implemented, replacing a weak linear congruential generator.
- Index calculus methods for discrete logarithm computations in prime finite fields have been implemented.
- The algebraic number fields module has been upgraded to correspond to KANT/Kash V2.2. In particular, this provides the user with much improved algorithms for class group and unit group computation.
- The algorithms for computing the Galois group of a number field (more precisely, its splitting field) have been extended by Katharina Geissler so that it is now possible to compute Galois groups for fields having degree up to 20 (previously 15).
- The original KANT/Kash module for function fields has been completely revised and its facilities have been vastly expanded by Florian Hess. In particular, machinery has been introduced for working with places and divisors.
- Local rings and fields have been improved and expanded. A wider range of local rings and fields may be created since inertia rings can now be specified by an inertia polynomial as well as by degree.

3 Removals and Changes

This section lists the most important changes in Version 2.7. Other minor changes are listed in the relevant sections.

- The category of elliptic curves `CurveEll` and their points `CurveEllPt` **has been changed** to `CrvEll` and `CrvEllPt`.
- The procedure `SetSeed` and function `GetSeed` have changed: see the section 'Random Object Generation' below.
- The parameters for the function `ClassGroup` for algebraic number fields and their orders have changed.
- The procedure `Verify` for algebraic number fields and their orders has been removed, since it is no longer necessary.
- The obsolete function `GroebnerWalk` has been removed and the parameters for `Groebner` have changed (see below).
- The function `NullSpace` has been renamed to `Nullspace`, which is more standard. `NullSpace` is supported for backwards compatibility in this version. The function `RowNullspace` has also been renamed to `NullspaceOfTranspose`.
- The basic algebra creation functions `MakeBasicAlgebra` and `pGroupToBasicAlgebra` have been renamed to `BasicAlgebra`.

4 Documentation

The HTML Help Document now has a summary (with links) for each chapter of the Handbook on the opening page for the chapter. Several of the Parts of the Handbook have also been reorganized. New chapters in the Handbook for V2.7 (with chapter numbers) are:

- Reflection Groups (23)
- Polycyclic Groups (27)
- Algebraically Closed Fields (41)
- Affine Algebras (51)
- Matrices (54)
- Binary Quadratic Forms (58)
- Quaternion Algebras (62)
- Modules over Algebras (67)
- Resolution Graphs and Splice Diagrams (74)
- Hyperelliptic Curves (76)
- Modular Forms (77)
- Incidence Geometry (82)
- Pseudo-Random Sequences (84)

The Categories Overview Document (`catv27.dvi`) has also been greatly expanded.

5 Language and System Features

5.1 Random Object Generation [HB 1]

Magma V2.7 contains an implementation of the new MONSTER random number generator due to G. Marsaglia [FSU, Tallahassee, FL]. The Monster generator has period $2^{29430}-2^{27382}$ (approximately 10^{8859}) and passes all of the stringent tests in Marsaglia's *Diehard* test suite (available online at `http://stat.fsu.edu/pub/diehard/`).

The new generator is thus very much better than the previous linear congruential generator, whose period was only $2^{31}-1$ (for which the full period could be easily exhausted).

Changes:

- Because the new Monster generator uses an internal array of machine integers, one 'seed' variable no longer expresses the whole state, so the method for setting or getting the generator state is now by way of a pair of values: (1) the seed for initializing the array, and (2) the number of steps performed since the initialization. In detail:
- The procedure `SetSeed(s, c)` now takes the initial seed s and a step number c to advance to. `SetSeed(s)` is equivalent to `SetSeed(s, 0)`.
- The function `GetSeed()` now returns the initial seed s used to initialize the random-number generator and also the current step c.
- Starting from a given seed s, randomly generating any kind of object using any random functions (including multi-precision integers) will now produce exactly the same result, no matter which machine type is used (including 64-bit machines versus 32-bit machines).

5.2 Types and Structures [HB 1]

New features:

- New function `ISA(T, U)` for types T and U to return whether T ISA U, i.e., whether objects of type T inherit properties of type U.
- New function `ElementType(S)` to return the type of the elements of structure S.
- New functions `CoveringStructure` and `ExistsCoveringStructure` to return the covering structure C of two structures S and T if it exists, so that S and T both embed into C.

5.3 Packages and Attributes [HB 2]

Bug fixes:

- A bug with package `require` statements which would often cause 'Source not available' to be printed erroneously has been fixed.

New features:

- New procedure `ListAttributes` to list the valid attribute field names for structures belonging to a given category.

– New function `GetAttributes` to return the valid attribute field names for structures belonging to a given category as a sorted sequence of strings.

– New verbose flags may now be created by the user within packages by a declare directive such as: `declare verbose MyFlag, 3;` (the 3 gives the maximum level).

6 Aggregates

6.1 Tuples [HB 9]

New features:

– The empty tuple `<>` is now allowed. Similarly, the empty cartesian product `car<>` is now allowed.

– New tuple constructor `<expr(x): x in ... | condition(x)>` as for sequences and sets, so one may build variable-length tuples with ease.

– One may now loop over the entries of a tuple T by `for x in T do ...`, etc.

6.2 Mappings [HB 13]

New features:

– The syntax of the generic-image map constructor has been extended so that a user may now specify both a map and its inverse. The new syntax is:

$$\text{map} < A- > B|x : - > f(x), y : - > g(y) >$$

where $g(y)$ is the inverse map of $f(x)$.

7 Groups

7.1 General Groups [HB 19]

Removals and Changes:

– The library `gps100` has been withdrawn. It is subsumed in the much extended database `SmallGroups` (see below).

New features:

– Various improvements to the `SmallGroups` database, including the `IdentifyGroup` function.

7.2 Permutation Groups [HB 20]

New Features:

– A dynamically constructed disjoint cycle form of a permutation may now be used. E.g., `G![{@ 1, 3,..@},{@ 2, 6,..@}, ...]` is now the same as `G!(1,3,..)(2,6,..)...`. Note that `Cycle(G![{@ 1, 3,..@},{@ 2, 6, ... @}, ...], 1)` will give {@ 1, 3, ... @}.

– The function `IsConjugate` has been extended in the case of permutation groups to allow the testing of whether a pair of indexed sets or a pair of multisets of points of G lie in the same G-orbit.

Bug fixes:

– Bug fixed in `GSet(GrpPerm, SetMulti)`. This signature now generates a runtime error. This avoids the potential problem of things in the same orbit having different multiplicities.

– Bugs fixed in: `DerivedSeries`, `PCGroup`, `SolubleQuotient`, `CompositionFactors`, `IsSimple`, `Cycle`, `CosetAction(GrpAb, GrpAb)`, `Subgroups`, `NormalSubgroups`, `MaximalSubgroups`.

7.3 General Matrix Groups [HB 21]

New features:

– The function `Centralizer(H, K)` for matrix groups H and K is now allowed, provided only that H and K have a common supergroup. (Previously K had to be a subgroup of H.)

Bug fixes:

– A bug in `LineOrbits` which would sometimes include the zero vector has been fixed.

– Bugs fixed in `ConjugacyClasses`.

– Bug in `Eigenspace` fixed.

7.4 Reflection Groups (New) [HB 23]

A package developed by Don Taylor provides for the construction of complex irreducible unitary reflection groups.

New features:

- Construction of all finite irreducible unitary reflection groups over the complex field.

7.5 Finitely Presented Groups [HB 24, 31]

New features:

- New function `AbelianQuotientInvariants(G, n)`, for G a finitely presented group and n an integer. This computes the abelian invariants of the quotient $G/ < G', g^n : \forall g \in G >$. It is particularly fast when n is a small prime.
- Using `LowIndexSubgroups` on a finitely-presented group which does not have relations defined is not possible, and this now causes an error message, rather than crashing or silently giving the wrong results.

7.6 Finitely Generated Abelian Groups [HB 25]

Changes:

- The function `AllHomomorphisms` for abelian groups has been renamed to `Homomorphisms`.

New features:

- The function `Centralizer(H, K)` for abelian groups H and K is now allowed, provided only that H and K have a common supergroup. (Previously K had to be a subgroup of H.)
- The function `DirectSum` has been added to take a sequence of abelian groups, and to return the direct sum, with the appropriate maps.

7.7 Finite Soluble Groups [HB 26]

Changes:

- New function `AutomorphismGroup(G)` to compute the automorphism group of a permutation group G.
- The function `Centralizer(H, K)` for soluble groups H and K is now allowed, provided only that H and K have a common supergroup. (Previously K had to be a subgroup of H.)
- The function `AllHomomorphisms` for soluble groups has been renamed to `Homomorphisms`.

New features:

- If G is a soluble group of order p^n, and v an element of the vector space of dimension n over GF(p), then `G!v` now produces the corresponding element of G.
- A problem in `ExtractGroup(p-quotient process)` that was causing a significant waste of time has been fixed.

7.8 Polycyclic Groups (New) [HB 27]

This new category comprises the family of all groups defined by a polycyclic presentation. Note that such a group may be infinite. Algorithms for element arithmetic and subgroup computations analogous to those for finite soluble groups have been implemented.

7.8.1 Construction and Arithmetic

- Nilpotent quotient of a finitely presented group (W. Nickel's algorithm)
- Direct products
- Product, inverse, conjugate, commutator for elements
- Element normal form and equality testing
- Element order
- Random element generation

7.8.2 Subgroups

- Subgroup and quotient group construction
- Normal closure of a subgroup
- Conjugation of subgroups
- Commutator subgroups
- Test subgroup membership and inclusion
- Test for normal and central subgroups

7.8.3 Structural information

- Test finiteness, group order
- Lower central series, derived subgroup and series
- Test for abelian, nilpotent, soluble
- Nilpotency class
- Hirsch number

8 Rings, Fields and Orders

8.1 Finite Fields [HB 37]

Magma V2.7 contains implementations of two index calculus methods for computing discrete logarithms in prime finite fields: the linear sieve and the Gaussian integer sieve.

The index calculus method applied to an arbitrary field GF(p), where p is a 100-bit prime such that $(p-1)/2$ is prime (the worst case), takes 10 seconds to perform the sieving and about 0.8 seconds to compute an individual logarithm. For a 20-decimal-digit prime p such that $(p-1)/2$ is prime, Magma takes only 1 second for the sieving and about 0.3 seconds to compute an individual logarithm.

New features:

- The arithmetic for extension fields of moderate to large prime fields has been improved.
- Index calculus method using either the Gaussian integer sieve or linear sieve (for prime fields GF(p)).

8.2 Univariate Polynomial Rings [HB 36]

New features:

- The function `Roots` now has a parameter `Max` to specify the maximum number of roots desired (and the function can be more efficient if this maximum is smaller than the actual number).
- New functions `lt`, `le`, `gt`, `ge` for univariate polynomials (uses the total ordering given by comparing degrees, then the comparison algorithm for the coefficients starting from the most significant).

8.3 Algebraic Number Fields [HB 38]

Changes:

- A new version of KANT (Kash V2.2) has been installed. As a consequence, the computation of class groups, units and S-units is much faster.
- The procedure `Verify` for algebraic number fields and their orders has been removed, since it is no longer necessary.
- The parameters for the function `ClassGroup` for algebraic number fields and their orders have changed.
- The function `ResidueClassField` has been changed to take only one argument (a prime). The previous syntax, taking the ring and the prime, will be deleted in a future release.
- The functions `CRT` and `ChineseRemainderTheorem` have been changed to take a sequence of residues of type `RngOrdElt`'s, followed by a sequence of moduli, as for the function over the integers. The previous four argument function will be deleted in a future release.

New features:

- The computation of the Galois group of the splitting field of a number field has been extended up to degree 20.
- Functionality for the rationals has been added for compatibility with orders in number rings.

8.4 Algebraically Closed Fields (New) [HB 41]

Magma V2.7 contains an exciting new system for computing with algebraically closed fields (ACF's), which have the property that they always contain all the roots of any polynomial defined over them. It is of course not possible to construct explicitly the closure of a field, but the system works by automatically constructing larger and larger algebraic extensions of an original base field as needed during a computation, thus giving the illusion of computing in the algebraic closure of the base field.

A similar system was suggested by D. Duval and others (the D5 system[1]), but this has difficulty with the parallelism which occurs when one must compute with several conjugates of a root of a reducible polynomial, leading to situations where a certain expression evaluated at a root is invertible but evaluated at a conjugate of that root is not invertible. The scheme developed for Magma by Allan Steel avoids these problems. Consequently, ACF's behave in the same way as as any other field implemented in Magma; all standard algorithms implemented for generic fields and which use factorization work without change (for example, the Jordan form of a matrix).

The new system avoids factorization over algebraic number fields when possible, and automatically splits the defining polynomials of a field when factors are found within the computation. These factors often arise automatically because of the structure of the algorithm which is computing over the field. The field may also be simplified and expressed as an absolute field if so desired.

Especially significant is also the fact that all the Gröbner basis algorithms work well over ACF's. One can now compute the variety of any zero-dimensional multivariate polynomial ideal over the algebraic closure of its base field. Puiseux expansions of polynomials are now also successfully computed using an algebraically closed field.

Features:

- Automatic extension of the field by the roots of any polynomial over the field, and operations on conjugates of roots
- Basic arithmetic
- All standard algorithms for rings over generic fields work over such fields
- Minimal polynomial
- Simplification of the field
- Construction of the corresponding absolute field together with the isomorphism
- Pruning of useless variables and relations

[1]Della Dora, J., Dicrescenzo, C., Duval, D., *About a new method for computing in algebraic number fields*, Proc. EUROCAL '85, Vol. 2. Lecture Notes in Computer Science (1985), **204**, pp. 289-290.

8.5 Rational Function Fields [HB 42]

New features:

– New parameter `Global` for the functions `FunctionField(R)` and `FunctionField(R, n)` to specify whether the result should be the unique global function field or not.

8.6 Algebraic Function Fields (Expandend) [HB 43]

An algebraic function field F/k (in one variable) over a field k is a field extension F of k such that F is a finite field extension of $k(x)$ for an element $x \in F$ which is transcendental over k.

In V2.7, the original KANT module for function fields has been completely revised and its facilities have been vastly expanded by Florian Hess. In particular, types have been introduced for places and divisors. For simplicity, the complete feature set is outlined below.

8.6.1 General Function Fields

Within Magma, general algebraic function fields can be created by adjoining a root of an irreducible, separable polynomial in $k(x)[y]$ to the rational function field $k(x)$. If k is a finite field, the function field is said to be *global*.

– Arithmetic

– Norm, trace of an element

– Minimal and characteristic polynomials of an element

– Representation matrices of algebraic functions with respect to the field extension $F/k(x)$

– Construction of (finite and infinite) equation orders

– Construction of (finite and infinite) maximal orders (Round 2 algorithm)

8.6.2 Invariants

– The genus

– Determination of the exact constant field via a k-basis

– L-polynomial resp. Zeta-function (global field)

– Determination of subfields

8.6.3 Orders and Fractional Ideals

- Discriminant
- Integral closures resp. maximal orders
- Construction of integral and fractional ideals
- Ideal arithmetic: product, quotient, gcd, lcm
- Determination of whether an ideal is: integral, prime, principal
- Decomposition of primes
- Valuations of order elements and ideals at prime ideals
- Ideal factorization
- Basis reduction (in the global, tamely ramified case)
- Unit rank, independent, fundamental units, regulator (in the global case).

8.6.4 Places

- Zeros and poles of algebraic functions
- Valuation (zero, pole orders) of algebraic functions at places
- Residue class field at a place
- Evaluation of algebraic functions at places
- Decomposition of places of $k(x)$ within $F/k(x)$
- Ramification index, inertia degree
- Bounds on the number of places of degree 1 (global field)
- All places of a prescribed degree (global field)
- Random place of a prescribed degree (global field)

8.6.5 Divisors

- Degree
- Divisor corresponding to a rational function
- Canonical divisor
- Different divisor
- Properties: Effective, positive, principal, special
- Riemann-Roch space $\mathcal{L}(D)$ of a divisor D, given by a k-basis of algebraic functions
- Index of speciality
- Estimate of the number of divisor classes of degree zero (global fields)
- Divisor reduction

8.7 Local (including p-adic) Rings and Fields [HB 45, 46]

Local rings and fields have been improved and expanded since Magma V2.6 for V2.7. A greater range of local rings and fields can be created since inertia rings can now be specified by an inertia polynomial as well as by degree. Some functions for polynomials over local rings and fields have been added providing increased functionality from their action only on polynomials over p-adic rings.

Changes:

- A balanced-mod representation of local elements has been implemented. It has been a huge help in gaining factorizations of polynomials over local rings and fields and has made this less likely to fail due to insufficient precision. From an aesthetic point of view the printing of elements is easier to read since -1 now looks like -1 and not $p^r - 1$. Series printing is also neater in some cases because of this.
- An exact-quotient representation of elements with infinite precision has been implemented. This allows operations with infinite precision elements to retain the infinite precision for longer and avoid some precision loss.
- The spelling of `InertseqPadic` has been changed to `InertseqpAdic`, to be compatible with `pAdicRing`.

New Features:

- The function `Coefficients` has been provided to supply the coefficients of the powers of the uniformizing element (those printed when `SeriesPrinting` is turned on). `Coefficient` has been added as well.
- A precision parameter is now available for the `HenselLift`ing of polynomials. This means that polynomials over an infinite precision ring can be lifted to any given precision. `HenselLift` will now accept a polynomial over a local ring or field and two polynomials whose coefficients can be coerced into that ring or field.
- Polynomial `Factorization` for polynomials over all local rings (and fields) is now available. This addition is accompanied by `Roots` and `HasRoot`.
- An indication of the precision required for `Factorization` can be gained by `SuggestedPrecision`, however, in a few cases this may still not be enough to gain a factorization of the given polynomial.
- Testing whether polynomials over local rings or fields are irreducible can be accomplished by `IsIrreducible`.
- Functions for polynomials over local rings also work for polynomials over local fields. These are functions such as `Factorization`, `Gcd`, `IsIrreducible`, `Roots` and `HenselLift`.
- An optional parameter has been made available for specifying the precision that the results of `Root` and `Sqrt` should be calculated to.
- `Denominator` has been added as a part of the exact quotient implementation to return the denominator of a local ring or field element.
- The function `#` can be used to determine the number of elements in a finite precision local ring. Iteration through the elements in such a local ring using `for` can also be done by will take time proportionate to the number of elements in L and as such is recommended only for "small" local rings (which must have very little precision).

- `GaloisImage` and `MinimalPolynomial`, which were previously only available for local ring elements, are now available for local field elements.
- Whether a local ring or field contains a p-th root of unity can be determined by calling `HasPRoot`.
- It is now possible to extend a local ring or field by an inertial polynomial, (polynomial irreducible over the residue class field). This applies to all local rings, including those which already have an unramified subring different to the p-adic ring. This has led to a greater variety of `LocalRing` construction functions, more options for the right hand side arguments in the `ext` constructors and a second `UnramifiedExtension` signature. It also makes available infinite precision inertia rings and subsequently local rings. However, coercion is not possible between two infinite precision inertia rings.
- Coercion between ramified local rings has been expanded. In addition to coercion between local rings with essentially the same Eisenstein polynomial over compatible inertia rings, it is possible to coerce between two local rings with some link between them. This link may be in the form of one being a direct extension (ramified or unramified) of the other or one being the result of `ChangePrecision` on the other or the two local rings having essentially the same Eisenstein polynomial over compatible inertia rings. A link between two rings may be a composition of direct links. Where direct coercion does not work a chain of coercions using intermediate rings may be possible. Coercion is not possible into or from an infinite precision ramified ring when the infinite precision ring has the larger ramification degree of the rings.

Bug Fixes:

- A bug concerning the characteristic polynomial of a matrix over a p-adic field has been fixed.
- A bug involving coercion into an ideal of a multi–variate polynomial has been fixed.

9 Commutative Algebra

9.1 Multivariate Polynomial Rings [HB 50]

In Magma V2.7, a major revision of the multivariate polynomials module has been achieved. The original linked-list representation of polynomials has been replaced with a more compact random-access array structure, resulting in less memory usage and faster access. A new fraction-free representation for polynomials at the lowest level gives very significant speedups for arithmetic over some fields (particularly the rational field and rational function fields).

A new representation employing variable byte sizes for monomials is also introduced in V2.7, requiring less memory and providing greater speed. The maximum total degree of any monomial has been increased to $2^{30} - 1 = 1073741823$. Monomial overflow is now also rigorously detected.

As a result of the many improvements, Magma now computes the `grevlex` order Gröbner basis for the Cyclic-7 roots ideal in 6.4 minutes and transforms this to the `lex` order Gröbner basis in a further 2.6 minutes (using the p-adic FGLM algorithm mentioned below).

Removals and Changes:

- The obsolete function `GroebnerWalk` has been removed.
- The `Groebner` procedure and related functions have a new parameter `Al` specifying which algorithm should be used for computing a Gröbner basis. The parameter `Walk` is now obsolete and will be removed in the future, but is currently supported for backwards compatibility.

New features:

- New parameter `Global` for the function `PolynomialRing(R, n)` (lexicographical order only) to specify that the result should be the unique global polynomial ring over R with n variables.
- New FGLM algorithm for converting the Gröbner basis of a zero-dimensional ideal from one monomial order to a different order (a fast p-adic method is used in the case of the rational field). This algorithm is automatically invoked internally when applicable.
- New function `GroebnerBasis(L, d)` which computes the degree-d Gröbner basis of the ideal generated by L, which is the truncated Groebner basis obtained by ignoring S-polynomial pairs whose degree is greater than d. This function also works with respect to the grading on the variables of the polynomial ring, if it has been defined this way.
- New parameter `RemoveRedundant` for the procedure `Groebner` to specify whether redundant polynomials in the input (which reduce to zero with respect to the other polynomials) should first be removed.
- New verbose levels have been added. There are now separate verbose flags `Buchberger`, `FGLM`, `GroebnerWalk`, etc., and the verbose flag `Groebner` includes all these flags implicitly.
- The function `Evaluate(f, T)` for multivariate polynomial f may now take a tuple T of n ring elements, where n is the rank of the parent of f. `Evaluate` has also been improved so that it handles automatic coercion better.

– New functions `Exponents` and `Monomial` to convert between a monomial and a sequence of integer exponents.

– New functions `lt`, `le`, `gt`, `ge` for multivariate polynomials (uses the total ordering given by comparing the leading monomials, then leading coefficients, then the same for successive terms).

– New function `SquarefreePart` to return the largest squarefree divisor of a polynomial.

– Multivariate GCD computation has been improved enormously, particularly over the integer ring and rational field.

– One may now compute GCDs of or factorize polynomials defined over any recursively-defined chain of univariate or multivariate polynomial rings or rational function fields of arbitrary depth.

– Primary decomposition of zero-dimensional ideals improved by a new algorithm based on the FGLM algorithm to compute the normal position of an ideal.

– Very many leaks have been removed.

Bug fixes:

– A bug in `PrimaryDecomposition` and `Variety` over finite fields has been fixed. (Reported by Bill Allombert.)

9.2 Affine Algebras (New Chapter) [HB 51]

An affine algebra in Magma is simply the quotient ring of a multivariate polynomial ring $P = K[x_1, \ldots, x_n]$ over a field K by an ideal J of P. Such rings arise commonly in commutative algebra and algebraic geometry. They can also be viewed as generalizations of number fields and algebraic function fields.

The elements of affine algebras are simply multivariate polynomials which are always kept reduced to normal form modulo the ideal of relations. Practically all operations which are applicable to multivariate polynomials are now also applicable to elements of affine algebras, when meaningful.

If the ideal J of relations defining an affine algebra $A = K[x_1, \ldots, x_n]/J$ is *maximal*, then A is a field and since V2.7 may be used with any algorithms in Magma which work over fields. Factorization of polynomials over such affine algebras is now also supported.

New features:

– New constructor `AffineAlgebra<K, ... | ...>` to create an affine algebra without having to create the multivariate polynomial ring first.

– Many functions applicable to multivariate polynomials are now applicable to affine algebra elements (including access functions like `Degree`, etc. and `Evaluate`).

– New linear algebra-based algorithm for computation of the minimal polynomial of an element of an affine algebra (based on FGLM algorithm), including special fast p-adic method over $\mathbf{Q}$.

– General algorithms which work over fields now work over affine algebras which are fields. Most significantly, factorization of both univariate and multivariate polynomials over such affine algebras (of characteristic 0) is now supported.

10 Linear Algebra and Module Theory

The chapters dealing with modules in the Handbook have been rearranged, yielding greater clarity.

10.1 Matrices (New Chapter) [HB 54]

A new feature that allows the creation of a matrix without having to predefine its parent structure has been introduced. A matrix may be expressed in a broad range of different formats. This is expected to make working with matrices a great deal easier. In addition, an extensive range of functions for creating matrices having a special structure (block, sparse, scalar, diagonal) has been provided. A new chapter in the Handbook brings together all the operations available for matrices.

New features:

- New creation function `Matrix` with very many kinds of arguments allowable.
- Several new functions for accessing matrices in easy ways (submatrices, etc.).
- Many matrix functions have been generalized to work for all applicable matrix types.

10.2 Vector Spaces [HB 56]

New features:

- One may now create a vector space with an inner product matrix F (by `VectorSpace(K, n, F)`), so that the functions `Norm` and `InnerProduct`, applied to elements of the space, will be with respect to the inner product matrix F. (The same holds for R-spaces.)

10.3 R-Modules [HB 56]

Changes:

- The function `AbsoluteRepresentation` has been renamed to `AbsolutelyIrreducibleModule`.
- The function `SingleMinimalSubmodule` has been renamed to `MinimalSubmodule`.

The existing facilities for $K[G]$-modules have been generalised so that the great majority of them apply to modules defined over a (matrix) algebra. As part of the restructuring, this class of modules is described in a new chapter that forms the first chapter in a new series (Handbook Part) relating to Representation Theory.

11 Lattices and Quadratic Forms

11.1 Lattices [HB 57]

New features:

– A new type `SymGen` has been created for the genus of a lattice, which builds on the previous machinery for enumeration of genus representatives by the Kneser neighbouring method. Unique local representatives provide rapid computation of equality.

Changes and new features:

– The function `Genus` has been changed to return a datatype for the genus of a lattice (type `SymGen`). The previous functionality of `Genus(L)` is achieved by `GenusRepresenatatives(L)`, or by the expression `Representatives(Genus(L))`.

– The function `SpinorGenus`, introduced in Magma V2.6, has been changed in exactly the same way as `Genus`. The function `SpinorRepresentatives(L)` takes a lattice argument.

11.2 Binary Quadratic Forms (New Chapter) [HB 58]

New features:

– A comprehensive treatment of the class group of nonfundamental quadratic forms.

– Natural homomorphisms from forms of nonfundamental discriminant to those of fundamental discriminant.

– The function `FundamentalQuotient` is the homomorphism of class groups to the forms of fundamental discriminant. The function `QuotientMap(Q1,Q2)` is the homomorphism of forms of discriminant $D_1 = m^2 D_2$ to forms of discriminant D_2.

– Natural coercion (via `!`) from forms of discriminant $D_1 = m^2 D_2$ to forms of discriminant D_2.

– Analysis of the 2-torsion subgroup (function `TwoTorsionSubgroup`) and the enumeration of its elements (function `AmbiguousForms`).

– Analysis of Sylow p-subgroup structure (function `pSubgroup`).

12 Algebras

12.1 Quaternion Algebras (New) [HB 62]

A quaternion algebra is a central, simple algebra of dimension four over a field. A special type for quaternion algebras is released in Magma V2.7. Support for orders over $\mathbf{Z}$ and $k[x]$ and their ideals is provided for quaternions over the rational field $\mathbf{Q}$ or $k(x)$. Special functions for enumeration all ideals in definite quaternion algebras over $\mathbf{Q}$, with connections to modular forms.

Features:

- Arithmetic of elements
- Norm, trace, and conjugation
- Minimal polynomial of elements
- Discriminant and ramified primes
- Creation of prime ideals
- Testing for principal ideals
- Enumeration of left and right ideals of an definite order over $\mathbf{Z}$
- Left and right orders of an ideal in a definite order over $\mathbf{Z}$

13 Algebraic Geometry

13.1 Resolution Graphs and Splice Diagrams (New) [HB 74]

These decorated graphs encode data generated by the resolution machinery. At present they are only attached to resolutions of plane curve singularities, but in due course they may be extended to resolutions of linear systems, surface singularities, etc. The package includes a resolution function for curves adapted to present its output in resolution graph format.

- Creation of resolution graphs and their vertices, either implicitly using resolution routines or Newton Polygons, or explicitly by listing the required data
- Calculation of numerical data associated to blowups, e.g., the canonical class of certain rational surfaces
- The Cartan matrix of a graph and associated calculations such as the contribution to the genus of a plane curve of a singularity having a given graph as its resolution
- Surgery on resolution graphs such as cutting an edge of a graph
- Creation of splice diagrams implicitly and explicitly
- Edge determinants and linking numbers of splice diagrams
- Test for regularity of splice diagrams
- Translation between resolution graphs and splice diagrams

13.2 Elliptic Curves [HB 75]

13.2.1 General Elliptic Curves

Removals and Changes:

- The category of elliptic curves `CurveEll` and their points `CurveEllPt` **has been changed** to `CrvEll` and `CrvEllPt`.
- The function `TwoTorsionPolynomial` has been changed to return the 2-torsion polynomial as a multivariate polynomial. The syntax `DivisionPolynomial(E, 2)` continues to return the univariate polynomial as the first return value.
- The function `TwoTorsionPolynomialMultivariate` is now deprecated and will be removed in a future release.
- The function `IsOrderOfPoint` has been changed to `IsOrder`; the previous syntax is now deprecated and will be removed in a future release.
- The syntax `BaseChange(E, K, j)`, where j is a homomorphism from `BaseRing(E)` to K has been changed to `BaseChange(E,j)`; the previous syntax is now deprecated and will be removed in a future release. Similarly for `BaseExtend`.
- The syntax `EllipticCurve(K,j)`, where j is the j-invariant of a curve over K, has been changed to `EllipticCurve(j)`; the previous syntax is now deprecated and will be removed in a future release.

New features:

– Elliptic curves over any field can be now created with given j-invariant.

– Elliptic curves can now be created from a plane curve C of genus one with a give rational point P (function `EllipticCurve(C,P)`). The transformation from a hyperelliptic curve of genus one, having a unique point at infinity, is achieved with `EllipticCurve(C)`.

– New function `DivisionFunction(E)` to return the division polynomial as an element of the affine coordinate ring of E.

13.2.2 Elliptic Curves over the Rational Field

New features:

– Implementation of the "computing heights of points with little or no factorization" algorithm.

– New functions `EllipticLogarithm` and `pAdicEllipticLogarithm` to compute the elliptic logarithm and p-adic elliptic logarithm of a point, respectively.

– Vastly improved rank calculations for elliptic curves with no two-torsion.

– New functions `IntegralPoints` and `SIntegralPoints` to compute integral points and S-integral points, respectively. New related functions `IntegralQuarticPoints`, `SIntegralQuarticPoints`, `SIntegralLjunggrenPoints` and `SIntegralDesbovesPoints`.

13.2.3 Elliptic Curves over Finite Fields

Removals and Changes:

– The undocumented functions `IsProvenSupersingular` and `IsProbablyOrdinary` have been removed.

New features:

– New function `AbelianGroup` to obtain the group of rational points, with corresponding isomorphism.

– New function `pSubgroup` to obtain the Sylow p-subgroup of the group of rational points.

– New function `SupersingularEllipticCurve` to compute a representative supersingular elliptic curve over a finite field.

13.2.4 Torsion Subgroup Schemes of Elliptic Curves

New features and Changes:

– The function `Subgroup` has been changed to `SubgroupScheme`, to avoid conflict with the new functionality for groups of rational points on an elliptic curve.

– The function `nTorsionSubgroup` has been changed to `TorsionSubgroupScheme`, to avoid conflict with the new functionality for groups of rational points on an elliptic curve.

13.3 Hyperelliptic Curves (New) [HB 76]

Magma V2.7 contains a package for computing with hyperelliptic curves and their Jacobians. This package has been developed by Michael Stoll (Düsseldorf), with revisions and kernel support by members of the Magma group.

Hyperelliptic curves will be provided as a specialisation within the general class of plane curves. The functions reflect their similarity to elliptic curves.

13.3.1 Hyperelliptic Curves

Features:

- Standard quadratic models, simplified models
- Invariants: degree, genus, discriminant
- Determination of points at infinity
- Heights of points on hyperelliptic curves
- Enumeration of all **Q**-rational points up to a given height bound (Elkies-Stahlke-Stoll method)
- Reduction of a hyperelliptic curve from rational field to finite field
- Reduction types and conductors of a genus 2 curve at odd primes
- Zeta functions of hyperelliptic curves over finite fields
- Maps between hyperelliptic curves
- Jacobians of hyperelliptic curves

13.3.2 Jacobians of Hyperelliptic Curves

Features:

- Normal form and addition of points
- Order of a point
- Structure of group of rational points over finite fields
- Height constant for genus 2 curves
- Canonical heights on Jacobians of genus 2 curves (following V. Flynn)
- 2-Selmer rank of the Jacobian of a genus 2 curve (via 2-descent)
- Rational torsion subgroup on the Jacobian of a genus 2 curve
- Rank of the subgroup of the Mordell-Weil group generated by a given set of points in the Jacobian (of a genus 2 curve over **Q**).
- Kummer surface of the Jacobian of a genus 2 curve

13.3.3 Kummer Surfaces

Features:

- Construction from the Jacobian of a genus 2 curve by defining equations
- Points, pseudo addition, and duplication on Kummer surface
- Naive height and canonical height

13.4 Modular Forms (New) [HB 77]

Magma V2.7 includes a preview of a major new initiative in modular forms and modular curves. The basis for the constructions is a finite-rank Hecke module equipped with the action of Hecke operators. The first examples of Hecke modules come from modular symbols. Other constructions include divisor groups of supersingular points on modular curves. A model for the supersingular points is provided either by supersingular elliptic curves following Mestre and Oesterlé or by quaternion ideal theory, the latter having deep connections to the theory of Shimura curves, generalizing modular curves.

13.4.1 Modular Symbols

A new modular symbols package was developed by William Stein (Berkeley) as part of his Ph.D. thesis, and redesigned in a recent visit to the Magma group in Sydney.

Features:

- Construction of spaces of modular symbols of given character, level, and weight.
- Computation of Hecke operators.
- Decomposition into invariant subspaces.
- Computation of invariants of modular abelian varieties.
- Determination of modular elliptic curves of a given level.
- Possibility of user reconstruction, verification, and extension of tables of Cremona.

13.4.2 Dirichlet Characters

Dirichlet characters have been developed in support of the new Hecke module types. Future applications to exponential sums will also be developed.

Features:

- Dirichlet group over a field K as the set of rational characters from $(\mathbf{Z}/N\mathbf{Z})^*$ to $\overline{K}^*$.
- Computation of group generators and enumeration of elements.
- Construction and evaluation Dirichlet characters.
- Invariants: modulus and order.

13.4.3 Hecke Modules of Brandt

A new quaternion algebra package, allowing ideal enumeration, provides the basis for the construction of Brandt modules. This uses the arithmetic of quaternions as a model for the equivalent construction of Mestre and Oesterlé via supersingular elliptic curves.

The Brandt module provide the only known effective means for computing component groups of the Néron model of Jacobians of Shimura curves and modular curves.

Features:

- Construction of Brandt module from the left ideals of an order in a quaternion algebra, together with Hecke operators given by the Brandt matrices.
- Decomposition of the Brandt module under the Hecke algebra.
- Natural inner product structure on the Brandt module.

13.4.4 Classical Modular Forms and Functions (Revised)

The standard collection of modular forms can be created as modular forms.

Features:

- Dedekind η-function
- Eisenstein series
- Theta series
- Evaluation of modular forms on binary quadratic forms
- Evaluation of Weber's f-function, Weber's f_2-function, j-invariant, on binary quadratic forms
- Hilbert class polynomial and Weber class polynomial

14 Incidence Structures

14.1 Incidence Structures and Designs [HB 80]

Changes:

– When creating incidence structures, the names `AffinePlane` and `ProjectivePlane` have been replaced by `FiniteAffinePlane` and `FiniteProjectivePlane` respectively. This avoids clashes with constructions of "ambient" spaces with the functions `AffinePlane(R)` and `ProjectivePlane(R)`, where R can be any ring.

14.2 Incidence Geometry (New) [HB 82]

14.2.1 Incidence Geometries (New)

Magma 2.7 contains facilities for creating and computing with incidence geometries and coset geometries. These have been developed by Dimitri Leemans (Brussels).

Features:

– Creation of incidence geometries
– Set of types, rank
– Diagram, incidence graph, elements
– Residue, truncation
– Properties: flag-transitive geometry, residually connected, firm, thin, thick
– Automorphism group
– Correlation group

14.2.2 Coset Geometries (New)

Features:

– Creation of coset geometries
– Conversion of an incidence geometry to a coset geometry
– Set of types, rank
– Residue, truncation
– Properties: flag-transitive geometry, residually connected, firm, thin, thick
– Borel subgroup
– Maximal parabolic subgroups

15 Coding Theory and Cryptography

15.1 Linear Codes [HB 83]

New features:

- The advanced Zimmermann algorithm for minimum weight computation has been implemented.
- New functions `MinimumWord` and `MinimumWords` to return one word or all words, respectively, of minimum weight from a code.
- New function `VerifyMinimumWeight` to verify that a code has minimum weight at least d for given d.
- New function `WordsOfBoundedWeight` for computing words of bounded weight.
- New function `DirectProduct` to return the direct product of two codes.
- New function `BCHBound` to return the BCH bound for a cyclic code (a lower bound for the minimum weight).
- New code attributes `MinimumWeightWord`, `MinimumWeightUpperBoundWord` and `MinimumWeightRows`.
- Bugs in `ExpurgateCode`, and `QRCode` have been fixed. `QRCode` now returns the correct quadratic residue code!

15.2 Pseudo-Random Sequences (New) [HB 84]

Magma V2.7 provides tools for the creation and analysis of pseudo-random bit sequences. The universe of these sequences is generally GF(2). However, some functions, such as Berlekamp-Massey, apply to sequences defined over arbitrary finite fields.

- Generation of n elements of a Linear Feedback Shift Register sequence (LFSR sequence)
- Next state of a LFSR sequence
- Characteristic polynomial of a LFSR sequence (BerlekampMassey algorithm)
- Shrinking generator
- Random sequence via the RSA random bit generator
- Random sequence via the Blum Blum Shub generator
- Auto-correlation of a binary sequence
- Cross correlation of two binary sequences
- Decimation of a sequence