Summary of New Features in Magma V2.22

May 2016

1 Introduction

This document provides a terse summary of the new features released as part of Magma versions V2.22 (May 2016).

A small number of new features were exported in patch releases prior to the main release of V2.22 in May 2016 and these are also listed here for completeness. Only significant bugfixes are noted here – for a more complete list of bugfixes the reader should consult the patch release change log for V2.21-x.

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

All timings below are for a 3.2GHz Intel Xeon E5-1650 processor, unless otherwise indicated.

2 Highlights

Algebraic Geometry

- *Schemes*
  - The package for working with isolated singularities that are analytically isomorphic to a hypersurface singularity has been expanded to cover most corank 3 families and more corank 2 families. Included are functions for classification, computation of normal forms and explicit transformation to normal form following Arnold's classification scheme. (V2.21-4)
- *Surfaces*
  - Desingularisation by local blow-up, an alternative to formal desingularisation, has been developed for surfaces. It is faster than formal desingularisation in certain cases, thereby making feasible some computations that were previously impossible. Blow-up desingularisation also provides additional information, not available with formal desingularisation. (V2.21-4)
  - Major speedups have been achieved in the main function for formally resolving projective hypersurfaces and in the computations of adjoints, and the other algorithms which depend heavily on this. In particular, computations over non-trivial number fields have been greatly sped up.

Arithmetic Geometry

- *Elliptic Curves Over* $\mathbf{Q}$ *and Number Fields*

  - The functions for computing the Mordell-Weil group, rank, etc, for the group of rational points on an elliptic curve over $\mathbf{Q}$ or a number field have undergone a major internal change. These functions now make use of the full power of the machinery in Magma: all the descent methods, and other (analytic) methods, are automatically used where appropriate. A related function provides information about the Tate-Shafarevich group. (V2.21-4)
  - An algorithm for computing a rank bound for a curve over $\mathbf{Q}$ with a 2-isogeny, by performing several higher descent steps, has been contributed by Tom Fisher. He has also contributed an implementation of 5-descent. (V2.21-4)

- *K3-Surfaces*

  - An algorithm implemented by Stephan Elsenhans computes the zeta-function of a (possibly singular model of a K3-surface $w^2 = f_6(x, y, z)$ with a sextic form $f_6 \in F_p[X, Y, Z]$. It has been used in practice for smooth models and for primes p up to 151. In the case of a singular model, even larger primes can be treated.
  - Using the zeta-function algorithm above, the standard method for computing upper bounds of the geometric Picard rank of a K3-surface over $\mathbf{Q}$ produces the exact rank in almost all cases.

Arithmetic Fields

- *Number Fields*

  - A new class group algorithm is now mostly in place. It is often much faster for higher degree number fields and is more robust. For example, the class group of fields of degree 24 and discriminant up to 105 digits can be computed in around one day. It can also handle all the cyclotomic fields of degree 72. (V2.21-4)

- *Galois Groups*

  - The recently developed Fieker-Klüners algorithm for finding the Galois group of a polynomial has been improved in several different ways. We believe that it is now capable of computing the Galois group of most polynomials of degree up to 100. For degree 20 polynomials over $\mathbf{Q}$ with moderately sized coefficients, the group will usually be found in less than one second.

Associative and Nonassociative Algebras

- *Associative Algebras*

  - It is now possible to compute maximal orders of algebras over function fields.

- *Non-Associative Algebras*

  A number of constructions for non-associative algebras have been implemented by Josh Malgione and James Wilson using tools from their multilinear algebra package (see below).

  - Constructions are provided for certain alternative algebras including composition and octonion algebras.
  - Similar tensor-based constructions are also provided for special and exceptional Jordan algebras.
  - The tensor approach is also used as the basis for a mechanism to compute certain invariants for any algebra. These invariants include the centre, centroid, left-right- and mid-nucleus and the derivation algebra.

Coding Theory

- *Linear Codes over* $\mathbf{Z}_4$

  A package for linear codes over $\mathbf{Z}_4$, contributed to Magma in 2009, by the Combinatoric, Coding and Security Group (CCSG) at the Universitat Autònoma de Barcelona has been considerably extended and a current version is distributed with Magma V2.22. The new features include:-

  - Computation of automorphism groups, whose elements act on coordinate positions only, for Hadamard $\mathbf{Z}_4$ codes, and for certain extended perfect $\mathbf{Z}_4$ codes.
  - Functionality for working with the information space and information set of a $\mathbf{Z}_4$ code.
  - Coset decoding, syndrome decoding, lifted decoding and permutation decoding algorithms for $\mathbf{Z}_4$ codes.

- *Linear Codes over* $\mathbf{F}_q$

  - The CCSG group has contributed code for permutation decoding for linear codes over $\mathbf{F}_q$.

Combinatorial Theory

- *Hadamard Matrices*
  - The database of Hadamard matrices has been extended with the addition of 29,613 inequivalent Hadamard matrices of degree 256 constructed from a list of all regular subgroups of order 256 in the affine symplectic group $ASp(8, 2)$.

Commutative Algebra

- *Univariate Polynomial Arithmetic*
  - New fast fraction-free methods have been introduced for computing extended GCDs of polynomials.
  - The calculation of GCDs and factorisation over number fields has been improved.
  - The computation of roots of polynomials over some types of large finite field has been improved.
- *Multivariate Polynomial Arithmetic*
  - The multiplication of dense polynomials has been improved by using Kronecker substitution, thereby exploiting FFT-based multiplication.
- *Gröbner Bases*
  - A major speedup has been achieved in the Gröbner basis algorithm for polynomial rings over GF(p) and GF(2^k) (the two cases have separate implementations). The speedup is due to a non-trivial improvement in the linear algebra phase of the algorithm.
  - A very major speedup has been achieved in the Gröbner basis algorithm for polynomial rings over $\mathbf{Q}$. This is due to a new algorithm which is considerably faster for most large inputs. In particular, the challenge problem Cyclic-9 runs 20 times faster than the previous algorithm, while challenge problem Cyclic-10 can now be solved for first time in about 30 hours.
  - A very major speedup has been achieved in the Gröbner basis algorithm for polynomial rings over number fields and rational function fields $K(x)$. In both cases new modular algorithms using LLL-construction are employed. For the first time number fields given as relative extensions can be handled.
  - A basic version of an algorithm for the Dixon resultant is included in the release. Related improvements to the computation of determinants over multivariate polynomial rings and function fields are also part of the release.

- *Ideal Arithmetic*

  - Improvements have been made to the algorithm used to compute the primary decomposition of an ideal in the positive dimension case (particularly when working over a characteristic p field).

Geometry

- *Incidence Geometry*

  - String C-groups are groups generated by involutions that satisfy an intersection property and which have a linear Coxeter diagram. There exists a one-to-one correspondence between string C-groups and abstract regular polytopes. These are also coset geometries that are thin, residually connected and flag-transitive. Machinery has been implemented by Dimitri Leemans for working with string C-groups. Among other things this provides a nice way to go from a group G with set of generators S such that (G, S) is a string C-group to the corresponding coset geometry and vice-versa.
  - C^+-groups are a new concept that that is used to study those chiral geometries that are known as hypertopes. Some initial machinery is now provided in Magma and this will be extended over time. In the past the focus was on flag-transitive geometries but now there is more and more interest in geometries that, while not flag-transitive, are close to being so. Chiral geometries are an example. Among other things, the B-diagram of a C^+-group can be computed.

Group Theory

For additional group-related facilities please see the section on *Representation Theory*.

- *Finitely-Presented Groups*
  - Derek Holt's package for finitely-generated subgroups of free groups has been expanded. It now includes algorithms for calculating normalizers and centralizers of subgroups and also testing conjugacy of subgroups. (V2.21-4)
  - Given a finitely-presented group G, the L_3-quotient algorithm of Sebastian Jambor constructs all quotients isomorphic to $\mathrm{PSL}(2,q)$ or $\mathrm{PGL}(2,q)$, simultaneously for all prime powers q.
  - An implementation by Sebastian Jambor of an algorithm of Plesken and Fabianska establishes the existence or non-existence of an infinite quotient of a two-generator finitely-presented group lying in $PSL_2(K)$ for K a field of characteristic zero.
  - Given a finitely-presented group G on two generators, the L_3U_3-quotient algorithm of Sebastian Jambor computes all quotients of G which are isomorphic to some $\mathrm{PSL}(3,q)$, $\mathrm{PGL}(3,q)$, $\mathrm{PSU}(3,q)$, or $\mathrm{PGU}(3,q)$, simultaneously for all prime powers q.
- *Matrix Groups*
  - An update of CompositionTree (CT) package has been provided by Eamonn O'Brien. It includes the following new features:-
    - Given an absolutely irreducible defining characteristic module of dimension at most d^2 over a finite field for a classical group of dimension d, code prepared by Brian Corr and Eamonn O'Brien reconstructs the natural module.
    - Improved code for classical constructive recognition of orthogonal groups in even characteristic prepared by Heiko Dietrich.
    - A new implementation of the algorithm for rewriting the elements of a classical group as words in standard generators undertaken by Csaba Schneider.
  - The Soluble Radical (SR) approach is the basis for many structure algorithms for permutation groups and is currently being developed for CT matrix groups. This release includes new CT+SR algorithms for normal subgroups, low index subgroups, coset actions and character tables. The SR algorithms in Magma often apply when other techniques fail. For example, the character table of the group $2^{1+22}.Co_2$ in a degree 1025 representation over GF(2) can be computed using these methods. SR methods are being developed by D. Holt, W. Unger and J. Cannon. (V2.21-4)

  - Conjugacy classes for symplectic groups, extended symplectic groups and orthogonal groups over finite fields of odd characteristic are now computed using their associated invariants. This is a significant speed improvement. (An extended symplectic group is a subgroup of the conformal symplectic group that contains the symplectic group.) The generic intrinsic that tests a pair of elements for conjugacy will shortly be changed to use these invariants in the case of symplectic and orthogonal groups over finite fields of odd characteristic. This will result in huge reductions in CPU time when performing conjugacy tests in larger degree groups.

- *Permutation Groups*

  - Structural algorithms for permutation groups that depend upon computing the socle have been extended to apply to groups having degrees up to 10^9 (the previous limit was 10^7). The algorithms affected include those for socle, chief series, simplicity tests, composition series, and composition factors. (V2.21-4)
  - A fast algorithm due to Frobenius is provided to determine the number of double cosets HxK of G.
  - A backtrack algorithm has been developed to find a canonical element of a double coset, given any element of the double coset. This makes it possible to test membership and equality of double cosets. A version of this algorithm which returns the coset representatives in blocks rather than as a single sequence has been provided.
  - The database of transitive groups in Magma has been extended by Derek Holt to include all transitive groups of degree less than 48 (previously was 33). (V2.21-4)

- *p-Groups*

  - The Small Groups database has been extended to include the groups of order 3^8, as computed by Mike Vaughan-Lee and communicated by Eamonn O'Brien. This data will need to be downloaded separately to access these groups.

- *Automatic Groups*

  - A database containing automatic structures for the fundamental groups of 5,389 of the first 5,800 manifolds in the Hodgson-Weeks census of small hyperbolic 3-manifolds is released for the first time. (There are a total of 11,031 manifolds in the census). Having the automatic structure for a fundamental group G defines a normal form and normalisation algorithm for the elements of G. An automatic structure can be read into Magma which can then perform elementary group theoretic operations on the fundamental group. The automatic structures were constructed by John Cannon and Derek Holt.

L-Functions

- *L-Series*
  - A new package for Jacobi sum motives has been added. The ability to identify the L-series with that of a Hecke Grössencharacter is a principal feature, which additionally allows local root numbers to be computed. See the Hypergeometric Motives and Hecke Grössencharacter sections for more information.
  - A new type `LSerMot` has been introduced that expedites certain types of L-series computations. In particular, the amount of (internal) precision required when invoking weighting functions (incomplete Mellin transforms) is now computed dynamically. The principal gain is when checking the functional equation, which for L-functions of higher degree or weight is now faster and more robust.
- *Hypergeometric Motives*
  - In conjunction with David Roberts and Fernando Rodriguez-Villegas, a package for computing with Jacobi sum motives is now available. One can also compute with Kummer twists of such motives. As with all motivic L-functions, the capacity to compute the Euler factors is a primary necessity. Here a p-adic method is used, while a complex method is also available for degree 1 primes. These methods also allow a Jacobi sum motive to be identified as a Grössencharacter, making Weil's famous theorem effectively computable. There is also functionality for basic arithmetic with such motives (adding, tensoring, scaling).
- *Hecke Grössencharacters*
  - Computations of local root numbers of Grössencharacters have been added. These can be combined to get the global root number, and then used to verify L-series calculations.

Lattices and Quadratic Forms

- *Lattices over Number Fields*
  - In conjunction with Gael Collinet, and also Markus Kirschmer with his student David Lorch, development of a package for computing with lattices over number fields has commenced. Basic functionality such as inner products, sublattices, and orthogonal complements are all available, and rely on the Dedekind module machinery. For totally definite lattices, computing automorphism groups and doing isometry tests are also available, as is enumeration of short vectors.

– Similar routines are being developed for Lorentzian lattices, which are lattices having one indefinite place. Automorphism groups and isometry between lattices can be determined using the data of a lattice and a timelike vector in it. Future work will be undertaken with the goal of performing cohomological computations with such lattices.

Linear Algebra and Module Theory

- *Linear Algebra over Finite Fields*

  A number of speedups have been implemented. These include:-

  - A version of Strassen multiplication exploiting GPUs for matrices over fields of characteristic 2.
  - Classical matrix multiplication exploiting GPUs for matrices over fields of characteristic 3, 5, and 7.
  - Matrix multiplication using CPUs for matrices over fields of characteristic 3.

- *Linear Algebra over* **Z** *and* **Q**

  - For matrices **Z** and **Q**, a major improvement to the nullspace algorithm has been achieved through the use of LLL reconstruction.
  - For matrices over **Z**, improvements have been made to the Hermite, saturation and determinant algorithms.

- *Linear Algebra over Number Fields*

  - Modular algorithms have been introduced for many critical matrix operations over number fields. At the same time the algorithms have been generalised to work for number fields given as relative extensions. The modular algorithms will provide significant speedups for the many areas of Magma which use linear algebra over number fields.

- *Multilinear Algebra*

  A large multilinear algebra package has been contributed by Josh Maglione and James Wilson (Colorado State U). The package provides tools for computing with tensors and multilinear maps.

  - Standard tensors can be constructed from direct data, user-provided functions, and derived from other algebraic objects such as the product in nonassociative algebras, the commutator operator in groups, or systems of forms.
  - The package includes tools to construct subtensors, ideals, and quotients of a tensor space as well as to compute invariants such as derivations, adjoints, nuclei, and centroids.
  - Standard tensor and cotensor space constructions are included such as the exterior and symmetric cotensor spaces.
  - Users can modify the underlying category of their tensors and tensor spaces, for example, to implement duality and triality. Standard categories are Albert's homotopism category, the cohomotopism category, and the adjoint category.

  - The package includes functions to produce standard exceptional tensors including octonions, composition algebras, and exceptional central simple Jordan algebras, and composition algebras.

- *Numerical Linear Algebra (over Real and Complex Fields)*

  - Routines for computing eigenvalues and the singular value decomposition have been added. These use the previously implemented QR-type decomposition, and then iterative methods. First the numerical Hessenberg form, and then the numerical Schur form, are computed, and the eigenvalues can readily be obtained from the latter. With the singular value decomposition, an intermediate bidiagonal form is first computed and then the numerical SVD.

Representation Theory

- *Basic Algebras*

  A number of additional functions have been added to the package for basic algebras by Jon Carlson.

  - Code has been provided to compute the quiver and its relations for a basic algebra.
  - The machinery for computing automorphisms of a basic algebra has been augmented with the addition of code to find the group of inner automorphisms.
  - A database of basic algebras for the p-modular group algebras of some of the smaller groups that are catalogued in the *Atlas of Finite Groups* has been constructed by Jon Carlson. For each group G included in the database and each prime p dividing the order of G, the goal is to store the basic algebra for each p-block of the group algebra $K[G]$.
  - A similar library contains the basic algebras of a small collection of Schur algebras $S(n, r)$.

- *KG-Modules*

  - It is now possible to test a KG-module M for being projective. The algorithm, due to Jon Carlson, deduces the result by considering the restriction of M to a Sylow p-subgroup for an appropriate prime p.

- *Characters of Finite Groups*

  - A major extension of the character table machinery allows calculations to be performed with the table of complex characters for a group G when G is absent. This has two benefits. Firstly, it allows computations with character tables in the case in which G has no representation of reasonable degree. Secondly, it makes it possible to construct compact databases of character tables.
  - A database of character tables for groups that appear in the *Atlas of Finite Groups* is in the process of being constructed. The current version of the database forms part of the V2.22 release. This comprises 350 of the approximately 430 character tables that are listed in the Atlas. Additional character tables will be added from time-to-time.
  - A mechanism is provided for writing a character table to a file and for subsequently reading it back in.

3 Documentation

New Handbook Chapters:

- Lattices over Number Fields.
- Multilinear Algebra.
- Non-associative Algebras.
- Linear Codes Over the Integer Residue Ring $\mathbf{Z}_4$.

Rearranged Handbook Chapters:

- The Chapter "Linear Codes Over Finite Rings" has been split into two chapters:
  - The chapter "Linear Codes Over Finite Rings" now covers machinery that applies generally to codes over finite rings, with the main emphasis on codes over Galois rings.
  - A new chapter "Linear Codes Over the Integer Residue Ring $\mathbf{Z}_4$" comprises material previously in the above chapter which applies only to linear codes over $\mathbf{Z}_4$ as well as some new material on $\mathbf{Z}_4$-codes.

3.1 Types and Structures

New Features:

- One may now define `sub`, `quo` and `ext` constructors for a user-defined type T. These are specified by supplying intrinsics `SubConstructor(X::T, t::Tup)` `QuoConstructor(X::T, t::Tup)` `ExtConstructor(X::T, t::Tup)` respectively, where the argument X is of type T (the user-defined type), and the argument y takes a tuple which contains all the objects on the right-hand-side of the constructor when it is called.

4 Algebraic Geometry

4.1 Schemes

New Features:

- The new intrinsic `LocalBlowUp` returns the result as a sequence of affine patches rather than globally embedding it.

Changes and Removals:

- Error checking in `InverseDefiningPolynomials` for a map between schemes which is a composition of maps between schemes has been improved. (V2.21-7)
- It is now again possible to construct points over non-fields having some common factor in their coordinates so long as the GCD of necessary groups of coordinates is not a zero divisor. This common factor is removed from the coordinates. (V2.21-4)

Bug Fixes:

- A bug involving length 0 schemes constructed from a ring to be the coordinate ring using `Spec` has been fixed. The ring is now converted to be of the expected type. A crash was seen in the use of `EmptySubscheme`. (V2.21-4)

4.2 Algebraic Curves

Bug Fixes:

- An internal error in `GapNumbers` given a divisor and a place of a curve has been fixed. (V2.21-10)

4.3 Algebraic Surfaces

New Features:

In addition to the older functionality for formal desingularisation of hypersurfaces in $\mathbf{P}^3$ in characteristic zero, a more general desingularisation routine has been developed that desingularises by blowing up. It has the current restriction that it only applies to surfaces with point singularities (the singular subscheme is of dimension zero) but this will be removed in due course.

- `DesingulariseSurfaceByBlowUp` and `ResolveSingByBlowUp` are the main blow-up desingularisation intrinsics.
- Accessing the data produced by the blow-up resolution: `NumberOfBlowUpDivisors`, `SingularPoint`, `BlowUpDivisor`, `BlowUpDivisorAllPatches`.
- Blow-up desingularisation intrinsics that don't depend on the ambient space: `IntersectionMatrix`, `Multiplicities`, `MultiplicitiesAndIntersections`, `LinearSystemDivisorRestriction`.

– Blow-up resolution intrinsics that involves canonical divisors on the desingularisation: `DifferentialMultiplicities`, `FirstChernClassOfDesingularization`, `CanonicalIntersection`.

– General desingularisation intrinsic which can choose either formal desingularisation or blow-up resolution: `ResolveSingularSurface`.

– Major speedups have been achieved in the main function for formally resolving projective hypersurfaces and in the computations of adjoints, and consequently in the other algorithms which depend heavily on this (such as `ArithmeticGenusOfDesingularization`).

– In particular, computations with schemes defined over non-trivial number fields have been significantly sped up.

5 Arithmetic Geometry

5.1 Elliptic Curves

Bug Fixes:

– The `MinimalQuadraticTwist` intrinsic when applied to a rational whose numerator has a prime-power divisor p^e with $6|e$ and $p > 3$ congruent to 3 mod 4 could return a curve with a different j-invariant. (V2.21-5)

– A bug with `HeegnerPoint` when computing more than 2^{30} primes was fixed. (V2.21-8)

5.1.1 Elliptic Curves over Number Fields

Bug Fixes:

– A problem with computing Euler factors at good primes (over a number field) which divide the denominator of the a-invariants has been fixed. (V2.21-3)

– An incorrect answer with `Order` of a torsion point over a number field has been fixed. The error was that the p-power torsion was not being bounded correctly in some cases. (V2.21-11)

5.1.2 Elliptic Curves over Finite Fields

Bug Fixes:

– A bug in the `ReducedTatePairing` has been fixed. (V2.21-11)

5.1.3 $K3$-Surfaces

New Features:

– A new intrinsic `WeilPolynomialOfDegree2K3Surface` developed by Stephan Elsenhans computes the zeta-function of a (possibly singular model of a) K3-surface $w^2 = f_6(x, y, z)$ with a sextic form $f_6 \in F_p[X, Y, Z]$.

– Using the above intrinsic `WeilPolynomialOfDegree2K3Surface`, the standard method for computing upper bounds of the geometric Picard rank of a K3-surface over $\mathbf{Q}$ produces the exact rank in almost all cases.

6 Arithmetic Fields (Global)

6.1 Algebraic Number Fields

New Features:

- Functionality for quotients of orders of number fields represented as an extension of another number field have been expanded. A `Random` intrinsic and an iterator have been provided. (V2.21-5) An `XGCD` intrinsic for elements is now included.
- `WeakApproximation` is now available for ideals of orders of number fields represented as an extension of another number field. (V2.21-6)

Changes and Removals:

- Coercion between number fields and their orders has been improved.
- A number of improvements have been made to the computation of `Subfields` of a number field. For hard examples when the number field is a direct extension of **Q** there is a speed up by about a factor of 3. The verbose printing for such computations has been improved.
- The determination of whether an ideal of a non-maximal order is a principal ideal (`IsPrincipal`) has been rewritten using a Picard group approach rather than a lattice approach.
- Compatibility has been improved for orders of number fields. This allows the comparison of orders having the same coefficient ring using `eq`. (V2.21-5)
- The `CoveringStructure` of two sets of places is now a group of divisors which all places in those sets can be coerced into. (V2.21-11)
- The error checking in `LLL` and `LLLBasisMatrix` when the input is an ideal of an order of a number field has been improved to give a runtime error when the number field is not a direct extension of the rational field. (V2.21-2)
- The computation of a `Kernel` of a matrix over an order has been improved.

Bug Fixes:

- `ChineseRemainderTheorem` taking an ideal, a sequence of infinite places, an element and a sequence of signs has been fixed when the element given is in the ideal. (V2.21-11)
- A bug in the computation of a `UnitGroup` of an order of a number field has been fixed. (V2.21-7)
- The `meet` of 2 number fields which are defined as extensions of different coefficient fields has been fixed. (V2.21-3)
- The application of an automorphism to a fractional ideal of an order of a quadratic field has been fixed. (V2.21-2)
- A problem with `MinimalPolynomial` was fixed, when the polynomial that was found had degree 0 (and thus trivial factorisation). (V2.21-2)
- A problem with `Discriminant` in abelian fields (relative discriminant formula) has been fixed. (V2.21-8)

6.1.1 Cyclotomic Fields

Bug Fixes:

- A fix has been made to `CyclotomicUnits`. (V2.21-6)

6.2 Algebraic Function Fields

New Features:

- `WeakApproximation` has been implemented for ideals of orders represented in a relative representation. (V2.21-6)

Changes and Removals:

- Error checking for non simple fields in intrinsics using Cartier representation has been improved. (V2.21-12)
- `UnderlyingRing` can now be applied to extensions of infinite degree. (V2.21-11)
- The `CoveringStructure` of two sets of places is now a group of divisors into which all places in those sets can be coerced. (V2.21-11)
- `Automorphisms` has been fixed for non-simple extensions. (V2.21-11)
- Error checking has been improved in `Automorphisms` of an algebraic function field over a given coefficient field. This intrinsic is currently only for function fields whose constant field is the rational field. (V2.21-9)
- Computations of `Subfields` have had some efficiency improved by not checking known irreducible factors for irreducibility when they are used in the calculation of subfields to define a function field. (V2.21-6)
- The computation of the `GaloisGroup` of a polynomial has been improved by choosing a prime which has been checked for suitability for use with all subfields involved in the calculation.
- The default precision used by the `Completion` of a function field has been fixed for subsequent calls to `Completion` so that the `Precision` parameter of the first call to `Completion` is not used as the default precision for all calls to `Completion`. (V2.21-4)

Bug Fixes:

- Printing of elements of non-simple fields has been fixed in some cases. (V2.21-6)
- The application of the `Completion` mapping and use of `Expand` have been fixed for non-simple relative extensions. (V2.21-11)
- A crash in `WronskianOrders` has been fixed. (V2.21-6)
- A bug in `ExactConstantField` computations for some function fields has been fixed. As a result more exact constant fields will be able to be computed and less errors will occur. (V2.21-4)

6.3 Galois Groups

Changes and Removals:

- Various improvements have been made to the computation of the `GaloisGroup` of a number field.
- The computation of a `GaloisGroup` of a polynomial over a characteristic p function field may now use msum polynomials.
- The computation of the `GaloisGroup` of a polynomial (especially a reducible polynomial) has been improved by choosing a prime which has been checked for suitability for use with all subfields involved in the calculation. (V2.21-6)
- The code performing verbose printing during `GaloisGroup` computations has been rewritten.

Bug Fixes:

- A fix has been made to `GaloisProof`. (V2.21-6)

7 Arithmetic Fields (Local)

7.1 p-adic Rings and their Extensions

New Features:

- Various speed improvements have been made to p-adic `Gamma`. (V2.21-6)
- The `RootNumber` of a p-adic field extension has been added. (V2.21-8)

Changes and Removals:

- A zero of a p-adic field now has infinite valuation.
- More error checking has been added to the computation of `Roots` of a polynomial over a local ring or field. (V2.21-6)
- `Gamma` of a 2-adic integer now has its output precision calculated more precisely. (V2.21-6)

Bug Fixes:

- `IsWildlyRamified` for extensions of p-adic fields has been corrected. (V2.21-8)
- An incorrect increase in default precision of a `Completion` of a number field, seen when computing preimages using the completion mapping has been fixed. (V2.21-6)
- A precision problem with p-adic determinants has been fixed. (V2.21-9)
- A problem with precision when computing determinants of p-adic matrices was fixed. (V2.21-10)

7.2 Series Rings

Changes:

- Computing `Roots` of polynomials over series rings over inexact rings have been disallowed. (V2.21-4)
- The computation of `Roots`, to a given precision, of polynomials over series rings over inexact rings have been disallowed. (V2.21-5)

Bug Fixes:

- The `IsSquare` intrinsic for series over the rationals whose constant coefficient (after removing the valuation) was not 1 could give wrong answers due to `Log`/`Exp` not working. The same problem can also occur with `IsPower` more generally. (V2.21-5)
- The `IsPower` intrinsic for series has been improved to consider scaling by a leading coefficient that is not 1. (V2.21-9)

7.3 General Local Fields

New Features:

- Precision handling with respect to the `RamifiedRepresentation` has been improved.

8 Basic Rings and Fields

8.1 Finite Fields

New Features:

– A major general speedup has been achieved in the case that many default finite fields of the same characteristic are created in succession. At the same time, a crash which would sometimes arise when creating a large lattice of finite fields of the same characteristic has been fixed.

8.2 Real and Complex Fields

Bug Fixes:

– `MinimalPolynomial` had a bug when the best approximation for the given parameters was the 1-polynomial. (V2.21-4)

9 Coding Theory

9.1 Linear Codes over Finite Fields

New Features:

– Permutation decoding is now available using `IsPermutationDecodeSet` and `PermutationDecode`. The intrinsics `PDSetSimplexCode` and `PDSetHadamardCode` are also available.

Changes and Removals:

– The intrinsic `Decode` has been replaced by `EuclideanDecoding` and `SyndromeDecoding`.

9.2 Linear Codes over Finite Rings

Version 2.0 of the *Linear Codes over* $\mathbf{Z}_4$ package developed by the Combinatoric, Coding and Security Group (CCSG) at the Universitat Autònoma de Barcelona has been installed for the V2.22 release. It extends the facilities available in Magma for $\mathbf{Z}_4$-codes in various ways.

New Features:

– Some new basic intrinsics include: `MinRowsGeneratorMatrix`, `KernelCosetRepresentatives` and `CosetRepresentatives`.

– Some intrinsics are provided for working with the information space and information sets of $\mathbf{Z}_4$-codes: `InformationSpace`, `InformationSet` and `IsInformationSet`.

– The intrinsics `SyndromeSpace`, `Syndrome` and `CosetLeaders` compute the syndrome space and coset leaders (representatives).

– Tools are provided to construct automorphism groups of certain classes of $\mathbf{Z}_4$-codes. The relevant intrinsics are `PermutationGroupHadamardCodeZ4`, `PermutationGroupHadamardCodeZ4Order`, `PermutationGroupExtendedPerfectCodeZ4`, `PermutationGroupExtendedPerfectCodeZ4Order` and `PAutExtendedPerfectCodeZ4Order`.

– Coset decoding, syndrome decoding, lifted decoding and permutation decoding are provided for $\mathbf{Z}_4$-codes by means of the intrinsics `CosetDecode`, `SyndromeDecode`, `LiftedDecode`, `IsPermutationDecodeSet`, `PermutationDecode`, `PDSetHadamardCodeZ4` and `PDSetKerdockCodeZ4`.

9.3 Hadamard Matrices

New Features:

– The database of Hadamard matrices has been extended with the addition of 29,613 inequivalent Hadamard matrices of degree 256 constructed from a list of all regular subgroups of order 256 in the affine symplectic group $ASp(8, 2)$.

10 Commutative Algebra

10.1 Polynomial Rings

New Features:

- The XGCD algorithm has been greatly sped up for polynomials over function fields by using a new fast fraction free method.
- The algorithms for both GCD and factorisation over univariate polynomials over ANFs have been greatly improved.
- The algorithm for the computation of roots of univariate polynomials over some types of large finite field has been improved.

Bug Fixes:

- A bug causing an incorrect result when powering extremely high degree binary polynomials in quotient rings has been fixed.

10.2 Multivariate Polynomial Rings

New Features:

- Multiplication of dense polynomials has been greatly improved by using Kronecker substitution (thus exploiting FFT-based multiplication)
- Certain base arithmetic operations are sped up greatly when the input polynomials have certain patterns which are recognised.

Changes:

- `Numerator` and `Denominator` of a multivariate polynomial now consider the polynomial as a rational function and return the polynomial itself and the 1 polynomial respectively. The intrinsics `CoefficientNumerator` and `CoefficientDenominator` have been added to return the LCM of the denominators of the coefficients and the product of that denominator with the polynomial. (V2.21-6)

Bug Fixes:

- A crash in `Roots` for polynomials over number fields has been fixed.
- `div:=` has been fixed to make it consistent with `div`.

10.3 Ideal Theory and Gröbner Bases

New Features:

- A significant speedup has been introduced into the linear algebra phase of the F_4 algorithm for computing Gröbner bases. For example, over a small-prime finite field, the speedup for computing the Cyclic-10 GB is about a factor of 4 and the speedup for computing the Katsura-12 GB is about a factor of 20. (Initially released in 2.21-4, but with further improvements in V2.22.)
- The standard F_4 Gröbner basis algorithm has been greatly sped up for ideals defined over GF(2^k) ($k > 1$). This gives major improvements to typical GB computations in the 'Descent' phase of Joux's method for discrete logarithms in finite fields of characteristic 2. For example, a typical Descent computation over GF(2^9) with 15 variables takes 2676 seconds in V2.22, compared with 16331 seconds in V2.21 (a speedup factor of 6.1).
- A new asymptotically-fast modular algorithm has been developed for computing Gröbner Bases of ideals defined over the rational field **Q**. This yields dramatic speedups for some large inputs. For example, the Cyclic-9 ideal GB over **Q** is computed in V2.22 in 390 seconds (about 19 times faster than for V2.21) and the Cyclic-10 ideal GB over **Q** is computed in V2.22 in about 30 hours (which was not practically computable previously).
- A new asymptotically-fast modular algorithm has been developed for computing Gröbner Bases of ideals defined over algebraic number fields. The new modular algorithm handles fields of arbitrary degree (while the previous modular algorithm was only applicable for degree up to 5). Also, the new modular algorithm is applicable to fields defined as relative extensions. (Initially released in 2.21-4, but with further improvements in V2.22.)
- The dense variant of the F_4 Gröbner basis algorithm has gained a moderate speed improvement.
- The computation of Gröbner bases over rational function fields $K(x)$ has been improved.
- The Wiedemann algorithm (used in the function `Variety`) has been improved in general, particularly when the computation is over base fields which are not prime finite fields.
- There have been more improvements made to the primary decomposition algorithm for ideals having positive dimension (particularly in the characteristic p case).

Changes:

- The deprecated parameter `Digits` has been removed from the `Variety` and `VarietySequence` intrinsics.

Bug Fixes:

- A crash in the computation of varieties over the complex field has been fixed.
- A crash in the primary decomposition algorithm for ideals defined over function fields has been fixed.

11 Geometry

11.1 Incidence Geometry

Machinery for working with C-groups and C^+-groups has recently been developed by Dimitri Leemans.

New Features:

- Intrinsics testing for properties of C-groups: `HasIntersectionProperty`, `HasStringProperty`, `IsCGroup`, `IsStringCGroup`.
- Intrinsics transferring to and from C-groups: `CosetGeometryFromCGroup`, `CosetGeometryToCGroup`.
- Intrinsic for constructing the Coxeter diagram of a C-group: `CoxeterDiagram`.
- Intrinsics testing for properties of C^+-groups: `HasIntersectionPropertyPlus`, `IsCPlusGroup`.
- Intrinsic constructing the coset geometry from a C^+-group: `CosetGeometryFromCPlusGroup`.
- Intrinsic constructing the B-diagram of of a C^+-group: `BDiagram`.

12 Groups

12.1 Classical Groups

New Features:

The conjugacy class representatives for symplectic groups, extended symplectic groups and orthogonal groups over finite fields of odd characteristic are now computed using their associated invariants. This is a significant speed improvement. (An extended symplectic group ($ExpSp$) is a subgroup of the conformal symplectic group (CSp) that contains the symplectic group (Sp).) Some of the available intrinsics are listed below. The reader should consult the Handbook for full details and a more complete list.

- Moving between the class invariant and an element of a symplectic group (Sp may be replaced by Csp or $ExtSp$): `ConjugacyInvariantSp` and `RepresentativeMatrixSp`.
- Constructing the conjugacy class invariants of a symplectic group (Sp may be replaced by Csp or $ExtSp$): `ClassInvariantsSp`.
- Constructing the class representatives of a symplectic group: (Sp may be replaced by Csp or $ExtSp$): `ClassesSp`.
- Constructing the class representatives and invariants of a symplectic group: `ClassRepresentativesSp`.
- Finding the order of the centraliser of a symplectic group element given its invariant: (Sp may be replaced by Csp or $ExtSp$): `CentraliserOrderSp`.
- Moving between the class invariant and an element of an orthogonal group: `ConjugacyInvariantO` and `RepresentativeMatrixO`.
- Finding the order of the centraliser of an orthogonal group element given its invariant: `CentraliserOrderO`.
- Constructing the conjugacy class invariants of an orthogonal group: `ClassInvariantsGO`, ClassInvariantsGOPlus, ClassInvariantsGOMinus.
- Constructing the class representatives of an orthogonal group: `ClassesGO`, `ClassesGOPlus` and `ClassesGOMinus`.
- Constructing the class representatives and invariants of an orthogonal group: `ClassRepresentativesGO`, `ClassRepresentativesGOPlus`, and `ClassRepresentativesGOMinus`.

12.2 Finitely Presented Groups

New Features:

- The following new intrinsics for free groups have been implemented by Derek Holt:
  - Centralizer of an element: `FSCentraliser`
  - Conjugacy of two elements: `FSIsConjugate`
  - Centralizer of a subgroup: `FSCentraliser`
  - Normalizer of a subgroup: `FSNormaliser`
  - Conjugacy of two subgroups: `FSIsConjugate`

– An algorithm of S. Jambor that constructs all quotients isomorphic to $\mathrm{PSL}(2,q)$ or $\mathrm{PGL}(2,q)$: `L2Quotients`.

– An implementation by S. Jambor of an algorithm of Plesken and Fabianska that searches for an infinite quotient of a 2-generator finitely-presented group lying in $PSL_2(K)$ for K a field of characteristic zero: `HasInfinitePSL2Quotient`.

– The $\mathrm{L_3U_3}$-quotient algorithm of S. Jambor which computes all quotients of a finitely-presented 2-generator group G which are isomorphic to some $\mathrm{PSL}(3,q)$, $\mathrm{PGL}(3,q)$, $\mathrm{PSU}(3,q)$, or $\mathrm{PGU}(3,q)$, simultaneously for all prime powers q: `L3Quotients`.

Bug Fixes:

– Unwanted printing in the `NilpotentQuotient` intrinsic has been turned off.

12.3 Matrix Groups Over Finite Fields

An update of the Composition Tree (CT) package has been provided by Eamonn O'Brien. Some noteworthy features of the update include the following:-

– Let G be a classical group of degree d over a finite field of characteristic p. Given an absolutely irreducible $K[G]$-module M over a field of characteristic p and having degree less than d^2, code written by Brian Corr and Eamonn constructs the natural module for G. The relevant intrinsics are `RecogniseSmallDegree` and `SmallDegreePreimage` which generalise and replace the following eight intrinsics [which apply only to special linear groups]:-

  – `RecogniseAlternatingSquare`, `AlternatingSquarePreimage`

  – `RecogniseSymmetricSquare`, `SymmetricSquarePreimage`

  – `RecogniseAdjoint`, `AdjointPreimage`

  – `RecogniseDelta`, `DeltaPreimage`

– Improved code for constructive recognition of orthogonal groups in even characteristic. The code was developed by Heiko Dietrich.

– A new implementation of the algorithm for rewriting the elements of a classical group as words in standard generators undertaken by Csaba Schneider. The intrinsic is `ClassicalRewriteNatural`.

The Soluble Radical (SR) approach to designing structure algorithms for a large matrix group over a finite field where the Composition Tree datastructure is used has been extended to the construction of further objects:-

– The normal subgroups: `LMGNormalSubgroups`.

– All subgroups of index less than a given bound: `LMGLowIndexSubgroups`.

– The action on the cosets of a subgroup: `LMGCosetAction`, `LMGCosetImage`, and `LMGCosetActionInverseImage`.

– The table of complex characters: `LMGCharacterTable`.

12.4 Permutation Groups

New Features:

- Structural algorithms for permutation groups that depend upon computing the socle have been extended to apply to all permutation groups. The theory behind the socle algorithm for a primitive group has been extended to degree 2^{32}, which exceeds the current degree limit for a permutation group of $2^{30} - 1$. The algorithm for imprimitive groups has also been improved. The algorithms affected include those for socle, chief series, simplicity tests, composition series, and composition factors. (V2.21-4)
- The `IsAltsym` routine now uses an improved probabilistic test for a transitive group to contain the alternating group. At high degree, this test uses many fewer random group elements than the previous test.

13 L-Functions

13.1 Dirichlet and Hecke Characters

New Features:

- The creation intrinsics `DirichletCharacter` and `HeckeCharacter` can now take a list as an argument instead of a tuple. (V2.21-3)
- The intrinsic `GrossenTwist` has been expanded to include twists by Hecke characters of the same modulus, and also returns a kernel corresponding to characters that are trivial on the given data. (V2.21-3)
- The `TateTwist` intrinsic can now be used to get a Hecke character corresponding to twisting by the norm, which internally is of the type `GrossenChar`. This can be done over any field (not just CM). (V2.21-6)
- Intrinsics for the local `Components` of characters have now been added, and allow a place or prime ideal to be specified. (V2.21-6)
- Intrinsics for the local `RootNumbers` of Hecke and Grössencharacters have now been added, and allow a place or prime ideal to be specified. The global root number can also be obtained in this manner. (V2.21-6)
- A new intrinsic `QuadraticCharacter` has been added to allow the user to obtain the Hecke character corresponding to a quadratic extension. (V2.21-6)

Changes:

- The `Extend` intrinsics for Dirichlet and Hecke characters (and groups) have been modified to eliminate the extraneous second return value (there is not really a kernel involved). The `Extend` intrinsic for groups thus now returns a group of the same size. (V2.21-2)
- Dirichlet and Hecke characters should now retain their ambient subgroups when multiplied by elements in the same subgroup. (V2.21-6)
- The printing with `Grossencharacter`s has been changed. (V2.21-6)

Bug Fixes:

- A problem with `HeckeCharacterGroup` of an abelian extension was fixed, taking into account infinite places. (V2.21-2)
- A bug with the `Domain` of a Hecke character (or its ambient group) has been fixed. (V2.21-4)
- A problem with assigning names to the generators of a Dirichlet group has been fixed. (V2.21-5)
- A problem with a Grossencharacter of negative weight has been fixed. (V2.21-6)

13.2 L-Series

New Features:

- An intrinsic for the `ImaginaryTwist` of a Hodge structure has been added. (V2.21-6)
- A new type `LSerMot` has been introduced that expedites certain types of L-series computations. In particular, the amount of (internal) precision required when invoking weighting functions (incomplete Mellin transforms) is now computed dynamically. The principal gain is when checking the functional equation, which for L-functions of higher degree or weight is now faster and more robust. (V2.21-4).
- Functionality for Jacobi sum motives is now available.

Changes:

- The convention for `RootNumber` of a `HodgeStructure` has been changed, taking the reciprocal of that given by Deligne. The code for Artin representations has been changed to reflect this. (V2.21-6)
- The output of `CriticalPoints` (of a Hodge structure) has been modified to be symmetric, and the intrinsic is now documented. (V2.21-6)
- The `EulerFactor`s for an L-series that had been `Translate`d were always computed to maximal precision, rather than to the amount of terms needed. (V2.21-8)
- Some internal changes were made so that `LCfRequired` is more consistent with `Translate`d L-series. (V2.21-8)

Bug Fixes:

- A problem with equality of L-series in the case of modular forms with different embeddings has been avoided, by having such equality always be false. (V2.21-4)
- A bug with the `Integral` vararg to `EulerFactor` was fixed. (V2.21-4)
- A bug with the `EulerFactor` of a Hilbert modular form at a prime which splits into different degree ideals was fixed (the relevant sequence could not hold both polynomials and power series). (V2.21-7)

13.3 Hypergeometric Motives

New Features:

- A complementary package for Jacobi sum motives and their Kummer twists has now been added. The most notable feature of this is the ability to identify these motives with a `Grossencharacter`, following Weil. (V2.21-6)

Bug Fixes:

- An incorrect answer for `EulerFactor` at a tame prime when 1 was in the cyclotomic data for the alpha's has been corrected. (V2.21-6)
- The `ComplexEvaluation` intrinsic for Jacobi motives was fixed to properly include the Tate-twisting factor. (V2.21-7)
- A bug with insufficient precision with `Grossencharacter` of a Jacobi motive has been fixed. (V2.21-7)
- Problems with identifying the correct infinity-type for the `Grossencharacter` of a Jacobi motive have been remedied. (V2.21-7)

13.4 Artin Representations

Bug Fixes:

– A bug when computing the `EulerFactor` of an Artin representation at a unramified place which is highly ramified in the field was fixed. (V2.21-4)

14 Lattices

14.1 Lattices over Z and Q

New Features:

– `BKZ` now works for matrices whose rows are dependent. (V2.21-11)

Changes and Removals:

– The `StepLimit` vararg in many routines can now be a 64-bit integer.

Bug Fixes:

– A crash on specific (rather rare) inputs for real matrices of low precision was fixed. The problem was that an iterative test was used, and the precision in such examples increased above the ambient. (V2.21-4)

– A problem with the `TimeLimit` vararg in `LLL` was fixed. (V2.21-8)

– An occasional error with precision at the 53-bit cusp was fixed in the `LLL` routine. (V2.21-10)

– The `InnerProductMatrix` of a lattice over a real field is now of the correct degree. (V2.21-11)

– `ClosestVectors` has been fixed in a case where the inner product matrix has large entries. (V2.21-11)

14.2 Lattices over Number Fields

New features:

– A new package for computing with lattices over number fields has been added. Much of it has been taken from work of Gael Collinet, and Markus Kirschmer and his student David Lorch have also been of assistance. All the underlying computations are done via the Dedekind module machinery.

  In the case of totally definite lattices, the functionality includes short vector enumeration and automorphism group and isometry testing. Similarly, in the Lorentzian case (one indefinite place), the automorphism group of a lattice with a given timelike vector can be determined, and the same for isometries.

15 Linear Algebra and Module Theory

15.1 Matrices

New Features:

- Matrix multiplication over finite fields of characteristic 3 has been improved in general.
- The nullspace algorithm for matrices over **Z** or **Q** has been improved.
- The integer matrix Hermite Normal Form, Saturation and Determinant algorithms have been improved in general.
- The algorithm for computing the Hermite Normal Form (HNF) of a sparse integer matrix has been greatly improved, especially when the resulting HNF has non-trivial diagonal entries before the last entry. For example, for a particular sparse full-rank 798 by 738 matrix A which occurs in a typical class group computation, the HNF of A is now computed in 0.8 seconds, compared with 11.7 seconds in V2.21.
- There has been a general improvement to the asymptotically-fast modular algorithms for several critical matrix operations for matrices defined over algebraic number fields. This also includes handling of fields given as relative extensions for the first time.
- Computation of the kernel of a matrix over an order of a Dedekind domain is now supported.
- The GPU-based matrix multiplication in characteristic 2 now uses the Strassen method.
- Fast matrix multiplication in characteristic 3, 5 and 7 is now supported for Tesla GPUs.
-

Changes:

- The `IsPositiveDefinite` intrinsic for real matrices is now more robust, and will give a runtime error if definiteness cannot be reliably determined. Similarly with `IsNegativeDefinite`. The corresponding intrinsics for semi-definiteness have been restricted to integral or rational input.

15.2 Modules over Dedekind Domains

New Features:

- The new package with lattices over number fields uses the Dedekind module machinery extensively.

Changes:

- Compatibility of modules over Dedekind domains has been improved. This, in particular, allows more flexibility in the combination of inputs to + and meet. (V2.21-9)

Bug Fixes:

- Some bugs in computing a `Minimal` sequence of `Generators` for a module with dimension 0 have been fixed. (V2.21-4)

15.3 Multilinear Algebra

Below we provide a sample of the functionality for the new multilinear algebra package. Please see the Handbook for a full description of all intrinsics.

New Features:

- Tensors of type `TenSpcElt` can be constructed using `Tensor`. Bilinear tensors can also be constructed using, for example, `CommutatorTensor` and `AssociatorTensor`. Tensors can be constructed from other tensors using operations such as `AlternatingTensor` and `(Anti)symmetricTensor`.
- Various operations with tensors are available. Tensors can be added, multiplied by scalars and `Compress`ed. Their `Domain` and `Codomain` can be accessed and their `Valence` computed. Properties such as `IsCovariant`, `IsContravariant`, `IsNondegenerate` and `IsSymmetric`, among others can be tested.
- Invariants such as `Radical`, `AdjointAlgebra`, `Discriminant` and `Pfaffian` can be obtained as well as `Centroid` and `DerivationAlgebra`. Nuclei can be computed using `Left`, `Right` and `MidNucleus`.
- Tensors can be used to define Heisenberg algebras and groups: `HeisenbergAlgebra`, `HeisenbergGroup`.
- Tensors belong to a tensor space. These parents can be constructed using a number of `TensorSpace` intrinsics for which a number of operations such as `Generators`, `Random`, `Valence` and `Frame` are available.
- All tensors have a category. Tensor categories such as `HomotopismCategory` and `AdjointCategory` can be constructed. Operations such as `Valence` and `Arrows` are available for tensor categories. Categorical operations such as `Subtensor(Space)` can be applied to tensors and tensor spaces.
- Homotopisms can be constructed between tensors. The following intrinsics are provided: `Homotopism`, `Domain`, `Codomain`, `Kernel` and `Image`.

15.4 Numerical Linear Algebra

New Features: The range of facilities for numerical linear algebra has been substantially expanded and are summarised in the points below:

- Intrinsics `NumericalRank`, `NumericalIsConsistent` and `NumericalSolution` provide for the solution of systems of linear equations over a real or complex field.
- Intrinsics `NumericalInverse` and `NumericalPseudoinverse` provide for the construction of an inverse for a real or complex matrix.
- Intrinsics `NumericalEigenvalues` and `NumericalEigenvectors` find eigenvalues and eigenvalues of a real or complex matrix.
- Intrinsics `NumericalImage` and `NumericalKernel` compute the kernel and image of a real or complex matrix considered as a mapping.
- Intrinsic `NumericalSignature` computes the signature of a real symmetric matrix.
- The intrinsic `NumericalSingularValueDecomposition` constructs the the singular value matrix for a matrix over a real or complex field.
- The intrinsics `NumericalBidiagonalForm`, `NumericalHessenbergForm`, and `NumericalSchurForm` transform a real or complex matrix into the indicated special form.

16 Linear Associative and Non-associative Algebras

16.1 General Associative Algebras

New Features:

- Maximal orders of an associative algebra defined over a function field can now be computed. This functionality is available through the intrinsics `MaximalOrderFinite` and `MaximalOrderInfinite` which compute an order over an extension of a polynomial ring and an over a valuation ring, respectively.
- A `MaximalOrder` can also be computed for an order of an associative algebra over a function field.

16.2 Clifford Algebras

New Features:

- Improvements have been made to the generator assignment code for Clifford algebras and the assignment of print names via `AssignNames`. If `Q` is an $n \times n$ quadratic form, the construction

  ```
  C¡x1, x2, ..., xn¿:= CliffordAlgebra(Q)
  ```

  creates a Clifford algebra of rank n and assigns the identifiers $\texttt{x}_1$, $\texttt{x}_2$, ..., $\texttt{x}_n$ to its generators, namely the basis elements of the underlying quadratic space of `C`.

16.3 Non–associative Algebras

New Features:

- The `Center`, `Centroid`, `Left`, `Mid` and `RightNucleus` as well as the `DerivationAlgebra` can be computed for all algebras.
- Whether an algebra has an involution can be determined by `IsStarAlgebra` and its involution returned by `Star`.
- The `GenericMinimalPolynomial`, `GenericTrace` and `GenericNorm` can all be obtained for power associative algebras as well as `GenericTracelessSubspaceBasis`.
- The construction of `CompositionAlgebra`s, `OctonionAlgebra`s and `SplitOctonionAlgebra`s is now possible. It is also now possible to compute a `JordanSpinAlgebra` and an `ExceptionalJordanCSA` (exceptional Jordan central simple algebra).

17 Representation Theory

17.1 Character Theory

New Features: A major extension of the character table machinery allows calculations to be performed with the table of complex characters for a group G when G is absent.

A database of character tables for groups that appear in the *Atlas of Finite Groups* is in the process of being constructed. The current version of the database forms part of the V2.22 release. This comprises 360 of the approximately 430 character tables that are listed in the Atlas. Additional character tables will be added from time-to-time.

- The parent structure for such characters without a group is created using `R := CharacterRing(Q)`, where `Q` is a sequence of pairs, `<o,n>`, where `o` is the order of the elements in the conjugacy class, and `n` is the length of the class. The first pair must be `<1,1>`, denoting the identity class of G.
- The call `ClassesData(R)` applied to a character ring returns a sequence `Q` as above. New attributes for a character ring are `Group` and `PowerMap`. A statement such as `assigned R‘Group` may be used to test whether or not a character ring has an attached group. The function call `PowerMap(R)` will compute the power map of `R` if `R` has an attached group.
- To create an element of such an `R`, let `v` be a sequence of cyclotomic field elements, with length equal to the length of `ClassesData(R)`. Then `R!v` will be the element of `R` taking value `v[i]` on the ith class.
- For any character ring, elements of the character ring can be added and multiplied together, and inner products of two elements may be taken. Functions such as `Decomposition` may also be used.
- A power map may be assigned to a character ring as `R‘PowerMap := PM`, where `PM` is a sequence of sequences of integers. If `d` denotes `ClassesData(R)`, then the length of `PM` is the length of `d`, and the length of `PM[i]` is `d[i,1]`, the element order of the ith class. The value of `PM[i,j]` is the image of class i under power j.
- If the character ring has a power map assigned, or has a group assigned, then functions such as `Indicator`, `Schur`, and `Symmetrization` may be applied to the elements of the ring.
- There are new access functions for character rings. They are `GroupOrder`, `GroupFactoredOrder`, `ClassesData`, `NumberOfClasses` and `PowerMap`.
- The `KnownIrreducibles` function returns the sequence of known norm 1 characters in the character ring. If all such characters are known, the `CharacterTable` function does the same. However, this latter will attempt to compute the full character table if the ring has an attached group, where the first will make no such attempt. If the full character table is known, then the `StructureConstant` function may be used.
- To read a character table from the new database, use a call like `CT := CharacterTable("Ly")`. The string names the group, using the same names as the `ATLASGroup` command, in this case Lyon's sporadic group. The result, `CT`, is a sequence of characters giving the full character table of the named group, as computed by the `CharacterTable` command.
- The call `Universe(CT)` will give the character ring. This will have attribute `PowerMap` assigned, but `Group` not assigned. Each character in `CT` is internally flagged as a character of norm 1, and has its Frobenius-Schur indicator stored within it.
- The full set of names of the the available character tables is obtained by `CharacterTableNames()`. There are currently 362 names in this set. We will be extending this collection.

17.2 $K[G]$-Modules

New Features:

- The new intrinsic `IsProjective` determines whether or not a $K[G]$-module is projective.
- The algorithm to compute `Hom` modules for general A-modules over algebraic number fields has been improved.

17.3 Basic Algebras

New Features:

- Intrinsics `Quiver` and `QuiverAndRelations` construct the quiver and its relations for a basic algebra.
- The new intrinsic `InnerAutomorphismGroup` constructs the group of inner automorphisms for a basic algebra.
- A small library of basic algebras for the p-modular group algebras of some of the smaller Atlas groups has been constructed by Jon Carlson. This library may be accessed using the intrinsics `BasicAlgebraFromGroup` and `BasicAlgebraGroupNames`.
- A second library contains a collection of basic algebras for Schur algebras and can be accessed using the intrinsic `BasicAlgebraFromSchur`.

18 System

18.1 I/O handling

New Features:

- Using sequence indexing on large strings (greater than 2GB) now works properly.
- `Read` has been slightly sped up for larger files.
- Some unnecessary memory duplication when calling `eval` has been removed.