

# Summary of New Features in Magma V2.27

May 2022

## 1 Introduction

This document provides a terse summary of the new features released as part of Magma versions V2.27 (May 2022).

A small number of new features were exported in patch releases prior to the main release of V2.27 in May 2022 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 V2.26-x.

## 2 Highlights

### Algebraic Geometry

- *Schemes*
  - The function `JacobianMatrix` (arising out the function `IsSingular`) has been greatly sped up for certain types of schemes.

### Arithmetic Geometry

- *Curves over Finite Fields*
  - A number of optimisations have been made to parts of the Tuitman algorithm for computing zeta functions which substantially improve the performance in many cases.

## Arithmetic Fields

- *Algebraic Number Fields*

- *Maximal orders* : The computation of all the primes at which an order is not maximal can be an expensive computation when computing maximal orders of number fields due to the expense of factoring large integers. This factorization expense can be reduced by attempting to compute polynomial GCDs over modular rings in a similar way to the Dedekind test combined with coprime factorization which reduces the size of integers to be factored (Sircana, 2020). This approach can also be used in computing maximal orders of orders defined as an extension of another order. While this prefactorization of the discriminant has been available in Magma since V2.26, improvements to the implementation have been made since then, including excluding relative orders defined by a non monic or non integral polynomials from this approach.

Another approach to reducing the integer factorization cost in computing maximal orders is implemented using the `Ramification` and `Discriminant` parameters to the `MaximalOrder` intrinsic. However, more care needed to be taken when using these parameters with orders defined by a non monic or non integral polynomial.

- *Subfields* : It is now possible to test whether a number field is a subfield of another without computing a maximal order. This means that testing whether 2 number fields are isomorphic can now also be done without computing a maximal order. There are a number of algorithms available for computing subfields and the algorithm used for fields of degree over 50 has been changed as it has become faster to compute all roots of the polynomial involved than lifting one root. Computing whether a field is a subfield also installs an embedding of the subfield into the superfield if the field is indeed a subfield. Embeddings can also be added using the `Embed` intrinsic. It is now checked that the embeddings added using this intrinsic as well as by `IsSubfield` and `IsIsomorphic` are consistent with any embeddings previously installed, unless the new `Overwrite` parameter is set to `true` when using `Embed`.

- *Valuation Rings*

- Valuation rings can be constructed over algebraic number fields and function fields allowing localizations to be computed at ideals of such fields.

## Basic Rings

- *Integer Ring*
  - Parallel versions of the ECM and MPQS algorithms for integer factorisation are now available. These are selected by first calling `SetNthreads(k)`; (to select  $k$  threads) and optionally the `StartWorkers` procedure (to select worker nodes) before general integer factorisation is called. (The MPQS algorithm also no longer uses files to communicate between workers.)

## Commutative Algebra

- *Gröbner Bases*
  - The parallel version of the Faugere  $F_4$  algorithm (selected by `SetNthreads`) has been greatly improved in the linear algebra phase, while the symbolic reduction and critical pair management phases have been parallelised for the first time. This includes the cases of ideals defined over  $\text{GF}(q)$  for  $2 < q < 2^{30}$  where  $q$  is not a power of 2 or over  $\mathbf{Q}$ . New record timings for a 4-core Intel Core i7-7700 CPU (3.60GHz) include the following (for computing the *grevlex* Gröbner basis in each case):
    - \* The Cyclic-9 roots ideal over  $\text{GF}(32003)$ : 10.2 secs (4 cores) or 29.3 secs (1 core);
    - \* The Cyclic-10 roots ideal over  $\text{GF}(32003)$ : 601.2 secs (4 cores) or 1957.1 secs (1 core).
    - \* The Cyclic-9 roots ideal over  $\mathbf{Q}$ : 144.3 secs (4 cores) or 286.1 secs (1 core), using 750MB for either computation;
    - \* The Cyclic-10 roots ideal over  $\mathbf{Q}$ : 5.2 hours (4 cores) or 16.1 hours (1 core), using 46GB for either computation.
  - The main strategy for the Faugere  $F_4$  algorithm has been extended so that by default a limit on the number of critical pairs is now dynamically chosen for each new step (so the pairs limit varies throughout the algorithm). This leads to major speedups for computing Gröbner bases of many types of input ideal for which the previous default strategy of no pairs limit was not optimal.
  - The Faugere  $F_4$  has been improved so that the setup of higher-degree critical pairs is delayed as much as possible. As a result, if the final Gröbner basis is trivial or has very small degree polynomials only, the critical pair management is often greatly sped up. The low-level subalgorithm for critical pairs management (used in both the Faugere  $F_4$  and Buchberger algorithm) has also been significantly sped up itself in general.

- The second pass reduction in the linear algebra phase of the Faugere  $F_4$  algorithm (which sometimes can be quite expensive) has been significantly sped up.
- The linear algebra phase of the Faugere  $F_4$  algorithm has a new representation for computations over  $\text{GF}(p)$  for  $p < 256$ . This yields significant memory savings when the Gröbner basis is large, together with a moderate speedup.
- New functions are provided to create dense MinRank or HFE multivariate polynomial systems.

- *Ideal Theory*

- The function `MinimalBasis` for an ideal of a multivariate polynomial ring has been improved in the search for a grading to make the input homogeneous if it is not already. The new function `HomogeneousWeightsSearch` also does the search for weights.
- The general strategy for primary decomposition of positive-dimensional ideals has been improved.

## Group Theory

- *Classical Groups*

- In recent years fast methods have been implemented for determining the conjugacy classes of elements in most classical matrix groups. In this version of Magma the coverage of fast conjugacy class construction has been extended to include the following types of group:-
  - \* Conformal classical groups.
  - \* The projective versions of the classical groups of these types:-  
 $GL, SL$   
 $Sp$   
 $GU, SU$   
 $GO+, SO+, Omega+$   
 $GO-, SO-, Omega-$   
 $GO, SO, Omega.$

- *Finite Soluble Groups*

- New routines for construction of finite soluble groups are much faster when the group order has large prime divisors.
- A new algorithm for automorphism groups of finite nilpotent groups has been implemented.

- *Databases of Groups*

- The small groups database has been expanded to include all groups with order a product of at most 4 primes. This implements the catalogue of Dietrich, Eick and Pan.

## Lattices and Quadratic Forms

- *Lattices*
  - An effective parallel version of close vector enumeration in lattices was completed. (V2.26-4)
  - The FP-based LLL algorithm has been parallelised (selected by `SetNthreads`) so that non-trivial speedups are achieved for lattices with medium dimension and large integer entries (occurring in several computations in number theory and in modular reconstruction algorithms).

## Linear Algebra and Module Theory

- *Linear Algebra over  $\mathbf{Z}$* 
  - The algorithm for final saturation when computing the nullspace of a matrix over  $\mathbf{Z}$  has been improved.
  - The general algorithm for saturation of an integral lattice has been improved.

## Representation Theory

- *KG-Modules*
  - The computation of irreducible  $G$ -modules in characteristic zero for a pc-group  $G$  has been made more stable.
  - The main algorithm for the computation of irreducible  $G$ -modules in characteristic zero has been improved, particularly in the case that the composition length of  $G$  is non-trivial.

## System

- *Parallelism*
  - There is a new procedure `StartWorkers` to start up Magma worker jobs remotely without having to start up Magma jobs on remote machines manually. This is useful for the internal distributed algorithms (such as code/lattice enumeration, integer factorisation). As a result, the `-w` command line option is no longer needed unless more manual control is desired.

## 3 Aggregates and Mappings

Bug Fixes:

- Error messages which may arise from some preimage computations for rule maps have been improved. (V2.26-7)

## 4 Algebraic Geometry

### 4.1 Schemes

New Features:

- The function `JacobianMatrix` (arising out the function `IsSingular`) has been greatly sped up for certain types of schemes.

### 4.2 Algebraic Curves

Bug Fixes:

- A problem with unstable curves input to `TernaryFormPotentiallyUnstablePrimes` and intrinsics such as `MinimizeReduce` which call it has been fixed by A.-S. Elsenhans. (V2.26-9)

## 5 Arithmetic Geometry

### 5.1 Curves Over Finite Fields

New Features:

- A number of optimisations have been made to parts of the Tuitman algorithm for computing zeta functions which substantially improve the performance in many cases.

Bug Fixes:

- A bug in the Tuitman algorithm for computing zeta functions related to precision has been fixed. (V2.26-7)

## 6 Arithmetic Geometry (Modular Forms)

### 6.1 Modular Forms

Bug Fixes:

- The function `eq` for modular form elements has been made consistent with `IsZero`.
- The function `Newforms` has been sped up for moderately-size cusp form spaces.

#### Bug Fixes:

- `ClassicalModularPolynomial(1)` has been fixed so it returns the symmetric polynomial  $x - y$ . (V2.26-10)
- An occasional crash in computing the dimension of Hilbert cusp form spaces has been fixed. (V2.26-11)
- A slowdown in computing Hecke operators of cusp form spaces has been fixed. (V2.26-11)

## 7 Arithmetic Fields (Global)

### 7.1 Algebraic Number Fields

New Features:

- The `Localization` of an order of a number field at an ideal of that order can now be computed. A valuation ring is returned.
- Whether warnings regarding expensive class group computations are printed can now be controlled using `SetPrintClassGroupWarnings`.

Changes and Removals:

- The efficiency of the Montes algorithm has been improved in one place.

Bug Fixes:

- A bug has been fixed when computing a `ClassGroup` of a degree 2 field. (V2.26-4)
- The prefactorization of discriminants when computing a `MaximalOrder` over the integers has been improved. (V2.26-4, V2.26-7)
- The computation of first roots of elements of number fields is now handled as a trivial case. (V2.26-7)
- A bug when inverting elements of low-degree number fields where the defining polynomial has large integer coefficients has been fixed. (V2.26-7)
- The test for a number field being a subfield of another (intrinsic `IsSubfield`) no longer involves computing a maximal order. (V2.26-7)
- The test for two number fields being isomorphic (intrinsic `IsIsomorphic`) no longer involves computing a maximal order. (V2.26-7)
- An improvement has been made when `UsePowerProduct` is `true` when computing `PicardGroup`. The expansion of large powers is further avoided. (V2.26-7)
- A check that the unit rank is at most 10 has been added to `ExceptionalUnits`. (V2.26-7)
- A crash in `MaximalOrder` for number fields which involve a non-monic or non integral defining polynomial has been fixed. (V2.26-9)
- The embeddings of number fields using `Embed`, `subset`, `IsSubfield` and `IsIsomorphic` have been improved. Now, once an embedding has been added, no embedding that is inconsistent with it can be added, except using `Embed` with the new `Overwrite` parameter set to `true`. (V2.26-10)
- `Magma` level printing of towers of algebraic number fields has been fixed. (V2.26-11)
- The algorithm used to compute `IsSubfield` for fields of degree more than 50 has been changed. Previously a root was computed by lifting but this showed to be unnecessarily expensive when for such degrees all roots can be computed in much less time. (V2.26-12)
- A crash has been fixed in `MaximalOrder` when the `Ramification` parameter is specified and the number field is defined by a non-monic or non integral polynomial. (V2.26-12)
- $p$ -adic rings constructed as a completion of the integers or rationals will now have default precision given by the `Precision` parameter, consistent with completions of number fields. (V2.26-12)



### 7.1.1 Quadratic Fields

Bug Fixes:

- `IsQuadratic` applied to an order of degree 2 which is not a radical extension has been fixed. (V2.26-11)
- Non-exact division of elements is now possible in orders of degree 2 fields which are not radical extensions. (V2.26-11)

## 7.2 Algebraically Closed Fields

Bug Fixes:

- A crash in the `Roots` function for polynomials defined over algebraically closed fields defined over function fields of small characteristic has been fixed.
- A crash involving algebraically closed fields defined over a small finite field has been fixed.

## 7.3 Abelian Extensions of Number Fields

Bug Fixes:

- A crash when computing the `NumberField` of a `FldAb` constructed using an invalid map has been improved to an error message. (V2.26-4)

## 7.4 Algebraic Function Fields

New Features:

- The `Localization` of an order of a function field at an ideal of that order can now be computed. A valuation ring is returned.

Changes and Removals:

- Series rings constructed as a completion of a rational function field will now have default precision given by the `Precision` parameter, consistent with completions of function fields. (V2.26-12)
- The efficiency of the Montes algorithm as been improved in one place.

Bug Fixes:

- A crash in `NormEquation` for function fields has been fixed. (V2.26-8)
- A small improvement has been made to the prime choosing when computing `Subfields` of a prime characteristic algebraic function field represented as an extension of another algebraic function field (V2.26-9) and as an extension of a rational function field.
- The embeddings between fields in a tower and the direct extension between 2 of these. (V2.26-10)

## 7.5 Galois Groups

### Changes and Removals:

- Efficiency has been improved when computing `GaloisSplittingField` or `SolveByRadicals` for polynomials over a prime characteristic rational function field.

### Bug Fixes:

- A precision has been increased when computing `GaloisGroups` of polynomials over characteristic 0 rational function fields to improve the accuracy of results. (V2.26-7)

## 8 Arithmetic Fields (Local)

### 8.1 $p$ -adic Rings and their Extensions

New Features:

- Exact  $p$ -adic rings and fields can be constructed and extended because of the inclusion of Christopher Doris' C package. These structures of type `RngXPad` and `FldXPad` were first documented in V2.26-6. Such structures and their elements can be approximated to as much precision as the objects they depend on allow, some to arbitrary precision. They are represented “lazily”, knowing how they are computed but not being approximated unless necessary. In addition to some expected  $p$ -adic invariants, a `RamificationPolygon` and `DiscriminantValuation` can be computed for such exact extensions.
- Elements of exact  $p$ -adic rings and fields (`RngXPadElt`, `FldXPadElt`) have some additional invariants which apply to them but not inexact elements. These invariants mitigate the infinite nature of the `Valuation` invariant for exact zeros. A `WeakValuation` can be obtained and valuation compared using `ValuationEq` and similar invariants where `Eq` is replaced by `Ne`, `Ge`, `Gt`, `Le` or `Lt`.
- A specific type of polynomial ring over exact  $p$ -adic rings and fields is provided (`RngUPolXPad`) in addition to being able to construct the usual polynomial rings over these exact rings and fields. Polynomials can be converted to `ExactPolynomials` to which `IsWeaklyZero` and `IsWeaklyEqual` can be applied. The `Factorization` of exact polynomials can be computed as well as their `Roots`. Further to the construction of a `NewtonPolygon`, `RamificationResidualPolynomials` can be calculated.

Bug Fixes:

- The printing of elements of precision 1 fields has been fixed so that the `0()` term uses the correct uniformizer name instead of 0. (V2.26-4)
- The comment attached to `SetPrecision` has been adjusted to clarify that it is `DefaultPrecisions` which are changed rather than the actual precisions of a ring or field and its coefficient rings. (V2.26-5)
- An improvement recently made to `Factorization` of polynomials has been fixed. (V2.26-5)

### 8.2 Series Rings

Bug Fixes:

- Improvements have been made to the `Factorization` of inseparable polynomials over series rings and extensions of series rings. It is also now possible to take  $p$ -th power roots of elements of extensions of series rings of characteristic  $p$ . (V2.26-10)

### 8.3 Valuation Rings

New Features:

- Valuation rings can now be constructed over algebraic number fields and function fields. These valuation rings have the same functionality as the existing valuation rings, including Euclidean operations on elements. Matrices can be constructed over these valuation rings and `HermiteForms` of these matrices can be computed.

## 9 Basic Rings and Fields

### 9.1 Integer Ring

New Features:

- Parallel versions of the ECM and MPQS algorithms for integer factorisation are now available. These are selected by first calling `SetNthreads(k)`; (to select  $k$  threads) and optionally the `StartWorkers` procedure (to select worker nodes) before general integer factorisation is called. (The MPQS algorithm also no longer uses files to communicate between workers.)

Changes:

- The command-line argument `-q` to Magma is no longer supported/needed, and the procedure `MPQS(n, S)` (where  $S$  specified a directory to communicate between workers) has been removed, since the algorithm no longer uses files.

### 9.2 Real and Complex Fields

Bug Fixes:

- A problem with printing polynomials over complex fields with imaginary numbers has been fixed.
- An inaccuracy in root finding over complex fields arising from computing reduced modules of hyperelliptic curves has been fixed. (V2.26-7)

### 9.3 Univariate Polynomial Rings

Bug Fixes:

- A hang in polynomial factorisation arising when testing isomorphism of hyperelliptic curves has been fixed. (V2.26-7)

### 9.4 Multivariate Polynomial Rings

Bug Fixes:

- An incorrect result when computing  $n$ -th roots of recursively defined polynomials has been fixed.

### 9.5 Local Multivariate Polynomial Rings

Bug Fixes:

- A crash in `NormalForm(f, S)` where  $S$  is a set or sequence of local multivariate polynomials has been fixed.

## 9.6 Rational Function Fields

Changes:

- The function `Evaluate(f, e)` for rational function field elements has been improved to avoid confusing automatic coercion of  $e$ . (V2.26-4)

# 10 Coding Theory

## 10.1 General Linear Codes

New Features:

- The intrinsic `StandardForm` now works for codes defined over prime-powered residue rings, not just  $\mathbf{Z}_4$ .

## 10.2 Linear Codes over Finite Fields

Bug Fixes:

- The function `EuclideanDecoding(C, v)` has been fixed for a code  $C$  constructed as `AlternantCode(A, Y, r)`, where  $A$  contains a zero element.
- A crash when testing equivalence of two linear codes has been fixed. (V2.26-10)

# 11 Commutative Algebra

## 11.1 Ideal Theory and Gröbner Bases

New Features:

- The parallel version of the Faugere  $F_4$  algorithm (selected by `SetNthreads`) has been greatly improved in the linear algebra phase, while the symbolic reduction and critical pair management phases have been parallelised for the first time. This includes the cases of ideals defined over  $\text{GF}(q)$  for  $2 < q < 2^{30}$  where  $q$  is not a power of 2 or over  $\mathbf{Q}$ . New record timings for a 4-core Intel Core i7-7700 CPU (3.60GHz) include the following (for computing the *grevlex* Gröbner basis in each case):
  - The Cyclic-9 roots ideal over  $\text{GF}(32003)$ : 10.2 secs (4 cores) or 29.3 secs (1 core);
  - The Cyclic-10 roots ideal over  $\text{GF}(32003)$ : 601.2 secs (4 cores) or 1957.1 secs (1 core).
  - The Cyclic-9 roots ideal over  $\mathbf{Q}$ : 144.3 secs (4 cores) or 286.1 secs (1 core), using 750MB for either computation;
  - The Cyclic-10 roots ideal over  $\mathbf{Q}$ : 5.2 hours (4 cores) or 16.1 hours (1 core), using 46GB for either computation.
- The main strategy for the Faugere  $F_4$  algorithm has been extended so that by default a limit on the number of critical pairs is now dynamically chosen for each new step (so the pairs limit varies throughout the algorithm). This leads to major speedups for computing Gröbner bases of many types of input ideal for which the previous default strategy of no pairs limit was not optimal; that previous default behaviour (with no pairs limit for each step) can now be selected by setting the parameter `PairsLimit` to 0.
- The Faugere  $F_4$  has been improved so that the setup of higher-degree critical pairs is delayed as much as possible. As a result, if the final Gröbner basis is trivial or has very small degree polynomials only, the critical pair management is often greatly sped up. The low-level subalgorithm for critical pairs management (used in both the Faugere  $F_4$  and Buchberger algorithm) has also been significantly sped up itself in general.
- The second pass reduction in the linear algebra phase of the Faugere  $F_4$  algorithm (which sometimes can be quite expensive) has been significantly sped up.
- The linear algebra phase of the Faugere  $F_4$  algorithm has a new representation for computations over  $\text{GF}(p)$  for  $p < 256$ . This yields significant memory savings when the Gröbner basis is large, together with a moderate speedup.
- New function `MinRankSystem(K, n, k, r)` to create a multivariate polynomial system corresponding to a random instance of the square  $(n, k, r)$ -MinRank problem over a finite field  $K$ .
- New function `HFESystem(q, n, D)` to create a multivariate polynomial system corresponding to a random instance of the Hidden Field Equations (HFE) system over  $\text{GF}(q)$  with  $n$  variables and secret degree  $D$ .
- The function `MinimalBasis` for an ideal of a multivariate polynomial ring has been improved in the search for a grading to make the input homogeneous if it is not already. The new function `HomogeneousWeightsSearch` also does the search for weights.
- The general strategy for primary decomposition of positive-dimensional ideals has been improved.

Bug Fixes:

- A crash in the symbolic phase of the  $F_4$  algorithm in the sparse monomial case has been fixed. (V2.26-2)
- A crash in the version of the  $F_4$  algorithm that uses the sparse monomial representation when there are more than 255 variables has been fixed. (V2.26-7)
- A bug in primary decomposition (arising from `PrimeComponents` applied to a scheme) has been fixed. (V2.26-5)
- A crash in `Dimension` for ideals with a grading containing zeros has been fixed. (V2.26-7)

## 11.2 Affine Algebras

Bug Fixes:

- The functions `VectorSpace`, `Dimension`, `MatrixAlgebra` and `Algebra` applied to an affine algebra  $Q = I/J$  have been fixed to include the case where  $I$  is a proper ideal. The same is now true for quotients of free algebras. (V2.26-7)
- A crash in `Image` for homomorphisms between modules defined over affine algebras has been fixed. (V2.26-7)

## 11.3 Invariant Theory

Bug Fixes:

- An bug causing an incorrect (non-minimal) result in `FundamentalInvariants` has been fixed. (V2.26-9)
- A bug causing an occasional incorrect result when computing the relation ideal of an invariant ring has been fixed (V2.27-1).

# 12 Geometry

## 12.1 Convex Polytopes and Polyhedra

Bug Fixes:

- A bug in `InteriorPoints` that could miss a point with one-dimensional polytopes having a rational endpoint and an integral endpoint was fixed (V2.27-1).

## 13 Groups

### 13.1 Classical Groups

New Features:

- In recent years fast methods have been implemented for determining the conjugacy classes of elements in most classical matrix groups. In this version of Magma the coverage of fast conjugacy class construction has been extended to include the following types of group:-
  - Conformal classical groups.
  - The projective versions of the classical groups of these types:-
    - $GL, SL$
    - $Sp$
    - $GU, SU$
    - $GO+, SO+, Omega+$
    - $GO-, SO-, Omega-$
    - $GO, SO, Omega.$
- The intrinsic `ClassicalGroupType(G)` and the algorithms which compute the conjugacy classes of a classical group are not restricted to the standard copies.

### 13.2 Databases of Groups

New Features:

- The small groups database has been expanded to include all groups with order a product of at most 4 primes. This implements the catalogue of Dietrich, Eick and Pan.



## 14 Lattices and Quadratic Forms

### 14.1 Lattices

New Features:

- An effective parallel version of close vector enumeration in lattices was completed. (V2.26-4)
- The FP-based LLL algorithm has been parallelised (selected by `SetNthreads`) so that non-trivial speedups are achieved for lattices with medium dimension and large integer entries (occurring in several computations in number theory and in modular reconstruction algorithms).
- New functions `ShortVector`, `ShortestVector`, `CloseVector`, `ClosestVector` to return a single non-zero short/close/shortest/closest vector in a lattice.

Bug Fixes:

- A bug in `CloseVectors` where it would sometimes miss close vectors when the lattice basis had small entries and the input vector had very large entries has been fixed.

## 15 Linear Algebra and Module Theory

### 15.1 Matrices

New Features:

- The algorithm for final saturation when computing the nullspace of a matrix over  $\mathbf{Z}$  has been improved.
- The algorithm for computing the nullspace of a large dense matrix over  $\mathbf{Q}$  has been improved.
- The general algorithm for saturation of an integral lattice has been improved.

Bug Fixes:

- A bug in `InvariantFactors` for matrices over small finite fields has been fixed. (V2.26-9)
- A bug where constructing random matrices over the finite field  $\text{GF}(2, 30)$  occasionally gave invalid entries has been fixed. (V2.26-11)

### 15.2 Sparse Matrices

New Features:

- The function `ChangeRing` for sparse matrices now allows the map argument (as for dense matrices).

Bug Fixes:

- The procedure `SetEntry` has been fixed to extend the matrix to the correct number of rows when the last argument is zero and the existing row is zero. (V2.26-10)

## 16 Linear Associative Algebras

### 16.1 Associative Algebras

Bug Fixes:

- The error checking of the index to `.` has been improved. (V2.26-4)

### 16.2 Basic Algebras

Bug Fixes:

- A bug in `BasicAlgebraOfEndomorphismAlgebra` where the dimension of the resulting algebra was wrong has been fixed. Reported by B. Sambale. (V2.26-11)
- A crash in `BasicAlgebraOfGroupAlgebra` has been fixed. (V2.26-11)

### 16.3 Matrix Algebras

Bug Fixes:

- A crash in `Centre` for matrix algebras with large numbers of generators has been fixed. (V2.26-12)

### 16.4 Finitely Presented Associative Algebras

Bug Fixes:

- A bug in reduction of the final Groebner basis computed for some types of ideal of a free algebra has been fixed. (V2.26-11)

## 17 Parallelism

New Features:

- New procedure `StartWorkers` to start up Magma worker jobs remotely without having to start up Magma jobs on remote machines manually. This is useful for the internal distributed algorithms (such as code/lattice enumeration, integer factorisation). As a result, the `-w` command line option is no longer needed unless more manual control is desired. Note that the main `magma` driving script should be updated too.

## 18 Representation Theory

### 18.1 $K[G]$ -Modules

New Features:

- The computation of irreducible  $G$ -modules in characteristic zero for a pc-group  $G$  has been made more stable.
- The main algorithm for the computation of irreducible  $G$ -modules in characteristic zero has been improved, particularly in the case that the composition length of  $G$  is non-trivial.

Bug Fixes:

- A crash in `AHom` for trivial action has been fixed.
- A slowdown when deleting large sequences of  $G$ -modules has been fixed.
- A bug has been fixed in `IrreducibleModules(G, RationalField())` for a PC group  $G$ , where isomorphic modules were returned.
- A bug in `DirectSumDecomposition` for  $G$ -modules where  $G$  was an unconditioned PC group has been fixed. (V2.26-10)

### 18.2 Modules over Algebras

Bug Fixes:

- A crash in `DirectSumDecomposition(M)` for an  $A$ -module  $M$  has been fixed. (V2.26-12)

## 19 System

### 19.1 GPU support

Bug Fixes:

- A crash in large matrix multiplication over  $\text{GF}(5)$  on V100 GPUs has been fixed. (V2.26-4)