Advanced Math Improvements in Maple 2026

Maple 2026 introduces important enhancements to the core mathematics engine, increasing the range of problems that can be solved symbolically and improving reliability in advanced computations.

Major Advances in Linear Recurrence Solving

Maple 2026 can now fully solve over 94% of the 55,979 entries in the Online Encyclopedia of Integer Sequences (OEIS) that satisfy a linear recurrence relation.

This advance reflects substantial new research in linear difference equations, implemented through major enhancements to the LREtools package and significant extensions to rsolve. Maple now solves broad new classes of homogeneous and inhomogeneous recurrences, including many second-, third-, and fourth-order cases that were previously out of reach.

The new functionality in LREtools also provides additional tools for working with linear difference operators and underpins the expanded capabilities of rsolve. A more detailed overview of the new commands and algorithms is available on the dedicated rsolve and LREtools page.

Improvements to Assumption Handling and Logical Inference

The assume facility, including the command **assume** for declaring assumptions and the commands **is** and **coulditbe** for reasoning under those assumptions, forms a core part of Maple's symbolic reasoning engine. These commands underpin logical inference across simplification, equation solving, property testing, and many internal algorithms.

Maple 2026 includes substantial improvements to the robustness, correctness, and scope of reasoning under assumptions.

Strengthened Reasoning Under Assumptions

Maple 2026 improves logical inference in a variety of common and subtle cases. For example:

```
> is(x<=x^2+1+sin(x)^2) assuming x::real;
```

true

```
> is(-I*(-2+3*X^2+2*(1-X^2)^2)^(1/2),real) assuming X > 0;
```

false

This expression simplifies to -2; **is** now recognizes that is an integer:

```
> is(-cosh(1)^2/sinh(1)^2-(cosh(1)^2-2)/sinh(1)^2, integer);
```

true

```
> is(u in {0,1}) assuming u::integer, u in RealRange(0,1);
```

true

Logical Disjunctions and Relational Assumptions

Logical disjunctions and relational assumptions are now evaluated with greater precision.

The following disjunction is valid only when the variables are real. By default, it evaluates to false:

```
> is(Or(b <= c,c <= a, a <= b));
```

false

Under the additional assumption that the variables are real, the result is true:

```
> is(Or(b <= c,c <= a, a <= b)) assuming real;
```

true

Similarly, the following inequality is now evaluated more carefully (it's only true if x is also known to be real):

```
> is(y <= z) assuming x + y <= x + z;
```

false

Improved Interpretation of Assumptions

Certain structural assumptions are now interpreted more accurately:

```
> assume(n <= n);
```

```
> getassumptions(n);
```

$\{n\sim::real\}$

```
> assume(n-1 < n);
```

```
> getassumptions(n);
```

$$\{n\sim{::}real\}$$

```
> n := 'n':
```

Logically inconsistent assumptions now generate more appropriate error messages:

```
> assume(n <= n-1);
```

Error, (in assume) the assumed property or properties cannot be satisfied

Reasoning, using **is** now better takes into account assumptions involving **signum**:

```
> is(0 < x) assuming signum(x) = 1;
```

$$true$$

```
> is(x < 0) assuming signum(x) = -1;
```

$$true$$

Improved Logical Consistency

These next three examples previously returned false in cases where the truth value could not be determined. This has now been corrected, and they appropriately return FAIL:

```
> is(M, diagonal) assuming M::AndProp(triangular, symmetric);
```

$$FAIL$$

```
> is(f, constant) assuming f::AndProp(monotonic, EvenMap);
```

$$FAIL$$

```
> is(x, real) assuming x::And(realcons, Not(infinity));
```

$$FAIL$$

Improved Robustness and Special Function Handling

Reasoning with **is** has been made more robust in computationally sensitive cases such as expressions involving large integer powers:

```
> is(exp(ln(99/100)/x^9) < 0) assuming 0 < x, x < 1/10;
```

$$false$$

Support for reasoning involving the LambertW function has also been extended, leading to improved results in **coulditbe** and **odetest**:

```
> coulditbe(-1/LambertW(-exp(-x)) <= 0) assuming x::real;
```

$$false$$

```
> ode := diff(y(x),x)*(x-ln(diff(y(x),x))) = 1:
```

```
> maple_sol := y(x) = -1/LambertW(-exp(-x))*(1+LambertW(-exp(-x))*x+
  LambertW(-exp(-x))^2)+c__1:
```

```
> odetest(maple_sol,ode) assuming x::real;
```

$$0$$

Enhancements to simplify

Maple 2026 includes several improvements to simplify, expanding its ability to produce more compact, mathematically natural forms across trigonometric, logarithmic, and radical expressions.

Euler's Formula and Exponential Forms

simplify has long performed the conversions such as:

- `exp(I*x) + exp(-I*x) -> 2*cos(x)`
- `exp(I*x) - exp(-I*x) -> 2*I*sin(x)`

and, beginning in Maple 2025, applied similar transformations to hyperbolic functions.

In Maple 2026, simplify now also performs the reverse transformation — applying Euler's formula to convert appropriate combinations of trigonometric or hyperbolic trig functions into exponential form when this yields a more compact result:

```
> map(simplify, [cos(x) + I*sin(x), cos(x) - I*sin(x), cosh(x) +
  sinh(x), cosh(x) - sinh(x)]);
```

$$\left[e^{I x}, e^{-I x}, e^{x}, e^{-x}\right]$$

As a consequence, certain integrals are now solved directly or returned in more compact form:

```
> int(sqrt(z)*sqrt(cosh(z)-sinh(z)),z=0..infinity);
```

$$\sqrt{2}\,\sqrt{\pi}$$

```
> int((x+sinh(x)+cosh(x))/(cosh(x)-sinh(x)),x);
```

$$\frac{(e^x)^2}{2} + e^x x - e^x$$

Improved Logarithmic Simplification

Logarithmic simplification has been significantly strengthened.

When the sign of the argument is known, logarithmic terms are now combined more effectively:

```
> simplify(ln(a)-ln(-a)) assuming a < 0;
```

$$\mathrm{I}\,\pi$$

```
> simplify(ln(a)-ln(-a)) assuming csgn(I*a) = 1;
```

$$-\mathrm{I}\,\pi$$

```
> simplify(ln(x)+ln(1/x)-2*I*Pi) assuming x < 0;
```

$$0$$

Logarithms involving complex numeric factors are now simplified by factoring out the real component:

```
> simplify(ln(2*I*z)-ln(I*z));
```

$$\ln(2)$$

```
> simplify(ln((2/3+2/3*I)*z)-ln((1+I)*z));
```

$$\ln(2) - \ln(3)$$

In the following two examples, the available assumptions allow **`simplify`** to combine logarithmic terms more completely:

```
> simplify(ln(a/x)-ln(a)+ln(x)) assuming x < 0;
```

$$\ln(-a) - \ln(a) + \mathrm{I}\,\pi$$

```
> simplify(ln(a/x)-ln(a)+ln(x)) assuming x < 0, a < 0;
```

$$0$$

In addition, **ln** now automatically simplifies perfect powers of rational numbers:

```
> ln(9/4);
```

$$2\ln\left(\frac{3}{2}\right)$$

Additional Refinements

Several refinements improve robustness and completeness of simplification:

The following example now simplifies to 0 under real assumptions:

```
> simplify(x*(x - 4*exp(x/2) + 2) + x*sqrt((-8*x - 16)*exp(x/2) +
  x^2 + 4*x + 16*exp(x) + 4)) assuming x::real;
```

$$0$$

Simplification with respect to radicals now gives the following result in a single call. A prior issue that prevented complete simplification in some cases has been corrected:

```
> simplify(((-alpha^2*x^2+R^2+x^2)/(-alpha^2*x^2+R^2))^(1/2)/(R^4/(-
  alpha^2*x^2+R^2+x^2)^2/(-alpha^2*x^2+R^2))^(1/2), radical)
  assuming R>0,alpha>0,x*alpha<R,x>=0;
```

$$\frac{\left(-\alpha^2 x^2 + R^2 + x^2\right)^{3/2}}{R^2}$$

Enhancements to combine

Maple 2026 includes improvements to **combine** that strengthen its ability to apply identities reliably and return more compact, mathematically natural forms. These updates are most noticeable in expressions involving inverse trigonometric functions and logarithms, and they also help downstream results in **simplify**.

Improved Combination of Inverse Trigonometric Expressions

combine makes better use of some identities involving arctan. For exmaple, the following simplification was already supported:

```
> combine(arctan(z)-arctan(1/z)) assuming z>0;
```

$$\arctan\left(\frac{z}{2} - \frac{1}{2z}\right)$$

In Maple 2026, this identify is applied more flexibly. **combine** (and consequently **simplify**) can also use the same identity more flexibly, allowing any two of the related terms to be combined into the third:

```
> {combine, simplify}(arctan(1/2*z-1/2/z)+arctan(1/z)) assuming z > 0;
```

$$\{\arctan(z)\}$$

The same functionality is used in this more complicated example involving **exp(x)** and **sinh(x)**:

```
> expr := arctan(exp(x))-arctan(sinh(x)):
```

```
> combine(expr) assuming x::real;
```

$$\arctan\left(\frac{e^x - \sinh(x)}{e^x \sinh(x) + 1}\right)$$

```
> simplify(expr) assuming x::real;
```

$$\arctan(e^{-x})$$

Improved Combinations of Logarithms

combine is now better at combining logarithm expressions under appropriate assumptions:

```
> combine( ln(x) + ln(1/x) - 2*I*Pi ) assuming x<0;
```

$$0$$

```
> combine(2*ln(x)-2*ln(y)) assuming x<0, y<0;
```

$$\ln\left(\frac{x^2}{y^2}\right)$$

```
> combine(ln(x) + ln(y)) assuming x<0, y<0;
```

$$\ln(x\,y) + 2\,\mathrm{I}\,\pi$$

These improvements can also lead to substantially improved results from **simplify** when **combine** is applied first:

```
> expr := combine(3200*ln(-exp(-t/3200) + 1) - 3200*ln(1 - exp(t/3200))) assuming t > 0;
```

$$expr := \ln\left(\frac{\left(e^{-\frac{t}{3200}} - 1\right)^{3200}}{\left(-1 + e^{\frac{t}{3200}}\right)^{3200}}\right) - 3200\, I\, \pi$$

```
> simplify(expr) assuming t > 0;
```

$$-t - 3200\, I\, \pi$$

Reduced Introduction of Radicals

combine also now avoids introducing radicals unless necessary. For example:

```
> combine(2*ln(2) + ln(7) + ln(Pi));
```

$$\ln(28\, \pi)$$

```
> combine(3*ln(2)-2*ln(3),ln);
```

$$\ln\left(\frac{8}{9}\right)$$

```
> combine(-I*ln(2)+I*ln(-1/2*I)+I/2*ln(1-I));
```

$$-\frac{I}{2}\left(\ln(-8 - 8\, I) + 2\, I\, \pi\right)$$

Enhancements to evalc

The **evalc** command evaluates complex-valued expressions symbolically. In Maple 2026, its handling of **RootOf** expressions has been corrected to preserve mathematical equivalence.

In previous versions, **evalc** would split **RootOf** of a polynomial with real coefficients into an expression of the form **x + I*y**, where **x** and **y** were themselves **RootOf** objects. However, this transformation was not equivalent to the original input and typically represented a larger set of values than intended. For example:

```
> evalc(RootOf(x^3-x^2));
```

$$RootOf(\_Z^3 - \_Z^2)$$

In Maple 2026, such expressions now return unchanged, ensuring correctness and consistency.

The previous functionality is now available through **evala** using **RealPart(R)** and **ImaginaryPart(R)**:

```
> evala(RealPart(RootOf(x^5+1)));
```

$$RootOf\left(4\_Z^3+2\_Z^2-3\_Z-1\right)$$

```
> evala(ImaginaryPart(RootOf(x^5+1)));
```

$$RootOf\left(16\_Z^5-20\_Z^3+5\_Z\right)$$

In addition, **evalc** no longer produces an error for symbolic powers of zero:

```
> evalc(0^x);
```

$$0^x$$

Enhancements to Differential Equation Solvers

Maple 2026 includes numerous improvements to dsolve, pdsolve, odetest, and related routines. These enhancements improve robustness, correctness, and the compactness of returned solutions across a range of ordinary and partial differential equations.

Improved Handling of Abel ODEs

Several refinements affect Abel equations:

- Certain Abel ODEs (when the second absolute invariant is zero) are now recognized as unsolvable much more quickly:

```
> dsolve(diff(y(x),x)=x+y(x)^3, y(x));
```

- This call to dsolve now returns without giving an error:

```
> dsolve([x^2+3*x*diff(y(x),x)=y(x)^3+2*y(x), y(1)=1],[Abel]);
```

- This Abel ODE now has a slightly more compact solution due to an improved formula:

```
> dsolve(x^2+3*x*diff(y(x),x) = y(x)^3+2*y(x), [Abel]);
```

$$y(x)=-\frac{1}{6}\left(x^{2/3}\left(\sqrt{3}+3\tan\left(RootOf\left(3x^{4/3}-4\_Z\sqrt{3}-2\ln\left(\frac{4}{3\left(\tan(\_Z)^2+1\right)}\right)-4\ln\left(\frac{\sqrt{3}}{2}+\frac{\tan(\_Z)}{2}\right)-2\ln(3)+12c_1\right)\right)\right)\sqrt{3}\right)$$

Improved Verification and Testing

- The following odetest call now returns quickly:

```
> odetest(y(x) = (-1/2*exp(RootOf(-exp(_Z)^2*ln(x*(exp(_Z)-2))+2*
  _C7*exp(_Z)^2+_Z*exp(_Z)^2+4*exp(_Z)*ln(x*(exp(_Z)-2))-8*_C7*exp
  (_Z)-4*exp(_Z)*_Z-2*exp(_Z)-4*ln(x*(exp(_Z)-2))+8*_C7+4*_Z+6))
  ^2+3*exp(RootOf(-exp(_Z)^2*ln(x*(exp(_Z)-2))+2*_C7*exp(_Z)^2+_Z*
  exp(_Z)^2+4*exp(_Z)*ln(x*(exp(_Z)-2))-8*_C7*exp(_Z)-4*exp(_Z)*
  _Z-2*exp(_Z)-4*ln(x*(exp(_Z)-2))+8*_C7+4*_Z+6))-6+2*(exp(RootOf(-
  exp(_Z)^2*ln(x*(exp(_Z)-2))+2*_C7*exp(_Z)^2+_Z*exp(_Z)^2+4*exp(_Z)
  *ln(x*(exp(_Z)-2))-8*_C7*exp(_Z)-4*exp(_Z)*_Z-2*exp(_Z)-4*ln(x*
  (exp(_Z)-2))+8*_C7+4*_Z+6))^2-6*exp(RootOf(-exp(_Z)^2*ln(x*(exp
  (_Z)-2))+2*_C7*exp(_Z)^2+_Z*exp(_Z)^2+4*exp(_Z)*ln(x*(exp(_Z)-2))
  -8*_C7*exp(_Z)-4*exp(_Z)*_Z-2*exp(_Z)-4*ln(x*(exp(_Z)-2))+8*_C7+4*
  _Z+6))+9)^(1/2))/(1/2*exp(RootOf(-exp(_Z)^2*ln(x*(exp(_Z)-2))+2*
  _C7*exp(_Z)^2+_Z*exp(_Z)^2+4*exp(_Z)*ln(x*(exp(_Z)-2))-8*_C7*exp
  (_Z)-4*exp(_Z)*_Z-2*exp(_Z)-4*ln(x*(exp(_Z)-2))+8*_C7+4*_Z+6))^2
  -3*exp(RootOf(-exp(_Z)^2*ln(x*(exp(_Z)-2))+2*_C7*exp(_Z)^2+_Z*exp
  (_Z)^2+4*exp(_Z)*ln(x*(exp(_Z)-2))-8*_C7*exp(_Z)-4*exp(_Z)*_Z-2*
  exp(_Z)-4*ln(x*(exp(_Z)-2))+8*_C7+4*_Z+6)))*x, x^2-2*x*y(x)+5*y(x)
  ^2 = (x^2+2*x*y(x)+y(x)^2)*diff(y(x),x));
```

$$0$$

- This call to odetest now returns properly:

```
> ode:=(y(x)-x*diff(y(x),x))/(x^2+y(x)^2)^(1/2) = m:

> sol:=ln(x) = -1/m*arcsinh(y(x)/x)+_C2:

> odetest(sol,ode,y(x)) assuming integer;
```

$$\mathrm{signum}(x)\, m - m$$

Student and PDE Improvements

- **Student:-ODEs:-Solve** now correctly returns the expected solution:

```
> Student:-ODEs:-Solve({diff(y(x),x,x) = y(x), y(0) = 1, y(1) = 2},y
  (x));
```

$$y(x) = (2\sinh(x) - \sinh(-1 + x))\operatorname{csch}(1)$$

- **pdsolve** now works correctly when **assumingusesAssume** is enabled:

```
> Physics:-Setup(assumingusesAssume = true):
```

```
> pdsolve({diff(u(r, t), t) = k*diff(u(r, t), r$2), u(r,0)=r*f(r), u
  (0,t)=0,u(a,t)=a*phi(t)}, u(r, t));
```

$$u(r,t) = \left(\sum_{n1=1}^{\infty} \frac{2\sin\left(\frac{\pi\, n1\, r}{a}\right) e^{-\frac{k\pi^2 n1^2 t}{a^2}} \int_0^a r\,(f(r) - \phi(0))\sin\left(\frac{\pi\, n1\, r}{a}\right) \mathrm{d}r}{a}\right) + \int_0^t \left(\sum_{n1=1}^{\infty} \frac{2\sin\left(\frac{\pi\, n1\, r}{a}\right) e^{-\frac{k\pi^2 n1^2 (t-\tau)}{a^2}} (-1)^{n1} \left(\frac{\mathrm{d}}{\mathrm{d}\tau}\phi(\tau)\right) a}{\pi\, n1}\right) \mathrm{d}\tau + r\,\phi(t)$$

Reduction and Simplification Improvements

- **DETools:-reduce_order** now produces more compact basis results free of unnecessary integrals and **DESols**:

```
> ODE := diff(y(x),x,x) -(exp(2*x)+exp(x)-x^2-1)/(-x+exp(x))*diff(y
  (x),x) + exp(x)*(x*exp(x)-x^2+x-1)/(-x+exp(x))*y(x) = 0:
```

```
> particular_solution := y(x) = exp(1/4*x^2+1/2*sinh(x)+1/2*cosh(x))
  *cosh(1/4*x^2-1/2*exp(x)):
```

```
> DEtools:-reduce_order(ODE, particular_solution, output = basis);
```

$$\left[e^{\frac{x^2}{4} + \frac{\sinh(x)}{2} + \frac{\cosh(x)}{2}} \cosh\left(\frac{x^2}{4} - \frac{e^x}{2}\right), -\frac{\sinh\left(\frac{x^2}{4} - \frac{e^x}{2}\right) e^{\frac{e^x}{2} + \frac{x^2}{4}}}{2} \right]$$

- Improvements in trigonometric simplification lead to more compact **dsolve** results:

```
> simplify(dsolve(diff(diff(y(x),x),x)*sin(x)^2 = 2*y(x)));
```

$$y(x) = -\mathrm{I}\cot(x)\ln\left(e^{2\,\mathrm{I}\,x}\right) c_8 + c_1\cot(x) - 2\,c_8$$

Improvements in SumTools

Maple 2026 expands the functionality of the **SumTools** package with new access functions for working with unevaluated and inert sums.

The following functions have been added to facilitate structured manipulation and analysis of sums:

- **GetSummand**,
- **GetVariable**,
- **GetRange**,
- **GetOptions**,
- GetParts

These commands allow programmatic extraction of the key components of a summation expression, making it easier to inspect, transform, and manipulate symbolic sums in a systematic way.

Updates to fsolve

The **fsolve** command now accepts the option **maxinit** to specify the number of initial points to be attempted using the Newton solver for multivariate problems. The default value has been increased from 20 to 30, improving the likelihood of locating solutions without requiring manual adjustment.

The following example succeeds with the new default setting:

```
> eqs:={20-1/450*(z+150)*(x/y)^(1/2)=0,100000-1000000000/3/Pi*
  (1/x/y)^(1/2)=0,59/3-1/450*z=0};
```

$$eqs := \left\{20 - \frac{(z+150)\sqrt{\frac{x}{y}}}{450} = 0,\ 100000 - \frac{1000000000\sqrt{\frac{1}{x\,y}}}{3\,\pi} = 0,\ \frac{59}{3} - \frac{z}{450} = 0\right\}$$

```
> fsolve(eqs, {x=0..10^6,y=0..10^6,z=0..10^6});
```

$$\{x = 1061.032954,\ y = 1061.032954,\ z = 8850.000002\}$$

If **maxinit** is set too low, a solution may be missed:

```
> fsolve(eqs, {x=0..10^6,y=0..10^6,z=0..10^6}, maxinit=10);
```

$$fsolve\left(\left\{20 - \frac{(z+150)\sqrt{\frac{x}{y}}}{450} = 0,\ 100000 - \frac{1000000000\sqrt{\frac{1}{x\,y}}}{3\,\pi} = 0,\ \frac{59}{3} - \frac{z}{450} = 0\right\}, \{x, y,\right.$$

$$z\}, \{x = 0 ..1000000, y = 0 ..1000000, z = 0 ..1000000\}, maxinit = 10\Big)$$

This update provides finer control over the solver’s search strategy while improving robustness in the default configuration.

Improvements to Polynomial System Solving

The Groebner[RationalUnivariateRepresentation] command introduces a new sub-algorithm that uses parallel computation and is significantly faster for large polynomial systems. In addition to improved performance, it often produces representations with smaller coefficients. See the Performance Improvements page for benchmark results and comparisons.

This new sub-algorithm is now used internally by RootFinding:-Isolate when computing certified numerical solutions of polynomial systems, improving both efficiency and scalability for these computations.