

# Computer algebra systems

[en.wikipedia.org/wiki/List\\_of\\_computer\\_algebra\\_systems](http://en.wikipedia.org/wiki/List_of_computer_algebra_systems)

“symbolic math” | computer algebra | symbolic computation | algebraic computation

- Algebra: substitutions, simplification, special functions, ...
- Calculus: derivatives, integrals, limits, series expansions, ...
- Solving algebraic/differential equations

## Maple

[www.maplesoft.com/products/maple/](http://www.maplesoft.com/products/maple/)

Maple 2015 Student Edition ... \$126

Maple 2015 Academic ... \$1560

### Maple Example

$$\frac{dN}{dt} = -\lambda N + P, \quad (1)$$

whose integral from  $t = 0$  to  $t = t_f$  with initial condition  $N(0) = N_0$  and assuming constant  $P$  is

$$N(t_f) = \frac{P}{\lambda} (1 - e^{-\lambda t_f}) + N_0 e^{-\lambda t_f}. \quad (2)$$

```
restart;
ode := diff(N(t),t) = -lambda*N(t) + P;
dsolve({ode, N(0)=0});
simplify(%);
```

## SymPy

Python library for symbolic mathematics

[www.sympy.org](http://www.sympy.org)

[live.sympy.org](http://live.sympy.org)

The lead developers are Ondřej Čertík and Aaron Maurer ([wiki](#))

### SymPy Tutorial

[docs.sympy.org/latest/tutorial/](http://docs.sympy.org/latest/tutorial/)

```
symbols() Function() Eq() Matrix() ImmutableMatrix() Integer() Rational() sympify()
.subs(.) .evalf() lambdify() .equals()
simplify() expand() factor() collect(.) coeff(.) cancel() apart() trigsimp()
expand_trig() powsimp() expand_power_exp() expand_power_base() powdenest()
expand_log() logcombine() factorial(n) binomial(n,k) gamma(z) hyper(...) .rewrite()
expand_func() hyperexpand() combsimp() list_to_frac([])
diff(.) .diff(.) Derivative() .doit() integrate(.,(.,)) Integral() limit(.,)
Limit() .series(.,) .removeO() as_finite_diff(.) finite_diff_weights(.,) apply_finite_diff(.,)
solve(.) roots() dsolve(.)
Matrix() ImmutableMatrix() .shape .row() .col() row_del() col_del() row_insert()
col_insert() .T eye() zeros() ones()
diag() .det() .rref() .nullspace() .eigenvals() .eigenvecs() .diagonalize() .charpoly()
besslj(.) .rewrite()
```

str() srepr() pprint() pretty() latex() print\_mathml() dotprint()

---

### SymPy Example...

```
from sympy import init_session
init_session()
N = Function('N')
t, P, lamda = symbols('t P lamda')
ode = Eq(N(t).diff(t), P - lamda*N(t))
ode
dsolve(ode, N(t), ics = {N(0):0})
```

*Ha! SymPy cannot solve an ODE with ICs...??*