Calcolo simbolico

Symbol is the head associated with a symbol.

Operazioni aritmetiche

x + y + z represents a sum of terms.

x - y is equivalent to x + (-1 * y).

x*y*z or x y z represents a product of terms.

x/y or Divide[x, y] is equivalent to x y^-1.

x^y gives x to the power y.

Max[x1, x2, ...] yields the numerically largest of the xi.
Max[{x1, x2, ...}, {y1, ...}, ...] yields the largest element of any of the lists.

Min[x1, x2, ...] yields the numerically smallest of the xi.
Min[{x1, x2, ...}, {y1, ...}, ...] yields the smallest element of any of the lists.

Sum[f, {i, imax}] evaluates the sum of f with i running from 1 to imax.
Sum[f, {i, imin, imax}] starts with i = imin.
Sum[f, {i, imin, imax, di}] uses steps di.
Sum[f, {i, imin, imax}, {j, jmin, jmax}, ...] evaluates a multiple sum.

Product[f, {i, imax}] evaluates the product of f with i running from 1 to imax.
Product[f, {i, imin, imax}] starts with i = imin.
Product[f, {i, imin, imax, di}] uses steps di.
Product[f, {i, imin, imax}, {j, jmin, jmax}, ...] evaluates a multiple product.

Espressioni algebriche

Apart[expr] rewrites a rational expression as a sum of terms with minimal denominators.
Apart[expr, var] treats all variables other than var as constants.

Cancel[expr] cancels out common factors in the numerator and denominator of expr.

Coefficient[expr, form] gives the coefficient of form in the polynomial expr.
Coefficient[expr, form, n] gives the coefficient of form^n in expr.

Collect[expr, x] collects together terms involving the same power of x.
Collect[expr, {x1, x2, ...}] collects together terms that involve the same powers of x1, x2, ...

Denominator[expr] gives the denominator of expr.

Distribute[f[x1, x2, ...]] distributes f over Plus appearing in any of the xi.
Distribute[expr, g] distributes over g.
Distribute[expr, g, f] performs the distribution only if the head of expr is f.

Expand[expr] expands out products and positive integer powers in expr.
Expand[expr, patt] avoids expanding elements of expr which do not contain terms matching the pattern patt.

ExpandAll[expr] expands out all products and integer powers in any part of expr.
ExpandAll[expr, patt] avoids expanding parts of expr which do not contain terms matching the pattern patt.

Exponent[expr, form] gives the maximum power with which form appears in expr.
Exponent[expr, form, h] applies h to the set of exponents with which form appears in expr.

Numerator[expr] gives the numerator of expr.

PowerExpand[expr] expands nested powers, powers of products, logarithms of powers, and logarithms of products.
PowerExpand[expr, {x1, x2, ...}] expands expr with respect to the x1.

Simplify[expr] performs a sequence of transformations on expr, and returns the simplest form it finds.

Together[expr] puts terms in a sum over a common denominator, and cancels factors in the result.

Polinomi

PolynomialQ[expr, var] yields True if expr is a polynomial in var, and yields False otherwise.
PolynomialQ[expr, {var1, ...}] tests whether expr is a polynomial in the vari.

CoefficientList[poly, var] gives a list of coefficients of powers of var in poly, starting with power 0.
CoefficientList[poly, {var1, var2, ...}] gives a matrix of coefficients of the vari.

Decompose[poly, x] decomposes a polynomial, if possible, into a composition of simpler polynomials.

Factor[poly] factors a polynomial over the integers.

InterpolatingPolynomial[data, var] gives a polynomial in the variable var which provides an exact fit to a list of data. The data can have the forms {{x1, f1}, {x2, f2}, ...} or {f1, f2, ...}, where in the second case, the xi are taken to have values 1, 2, ... The fi can be replaced by {fi, dfi, ddfi, ...}, specifying derivatives at the points xi.

PolynomialGCD[poly1, poly2, ...] gives the greatest common divisor of the polynomials poly1, poly2, ...

PolynomialLCM[poly1, poly2, ...] gives the least common multiple of the polynomials poly1, poly2, ...

PolynomialQuotient[p, q, x] gives the quotient of p and q, treated as polynomials in x, with any remainder dropped.

PolynomialRemainder[p, q, x] gives the remainder from dividing p by q, treated as polynomials in x.

Resultant[poly1, poly2, var] computes the resultant of the polynomials poly1 and poly2 with respect to the variable var.

Proposizioni

True is the symbol for the Boolean value true.

False is the symbol for the Boolean value false.

!expr is the logical NOT function. It gives False if expr is True, and True if it is False.

e1 && e2 && ... is the logical AND function. It evaluates its arguments in order, giving False immediately if any of them are False, and True if they are all True.

e1 || e2 || ... is the logical OR function. It evaluates its arguments in order, giving True immediately if any of them are True, and False if they are all False.

lhs === rhs yields True if the expression lhs is identical to rhs, and yields False otherwise.

lhs =!= rhs yields True if the expression lhs is not identical to rhs, and yields False otherwise.

Relazioni aritmetiche

lhs == rhs returns True if lhs and rhs are identical.

lhs != rhs returns False if lhs and rhs are identical.

x >= y yields True if x is determined to be greater than or equal to y. x1 >= x2 >= x3 yields True if the xi form a non-increasing sequence.

x <= y yields True if x is determined to be less than or equal to y. x1 <= x2 <= x3 yields True if the xi form a non-decreasing sequence.

x < y yields True if x is determined to be less than y. x1 < x2 < x3 yields True if the xi form a strictly increasing sequence.

x > y yields True if x is determined to be greater than y. x1 > x2 > x3 yields True if the xi form a strictly decreasing sequence.

Equazioni

MainSolve[eqns] is the underlying function for transforming systems of equations. Solve and Eliminate call it. The equations must be of the form lhs == rhs. They can be combined using && and ||. MainSolve returns False if no solutions to the equations exist, and True if all values of variables are solutions. MainSolve rearranges the equations using certain directives.
MainSolve[eqns, vars, elim, rest] attempts to rearrange the equations eqns so as to solve for the variables vars, and eliminate the variables elim. The list rest can be included to specify the elimination order for any remaining variables.

Eliminate[eqns, vars] eliminates variables between a set of simultaneous equations.

Reduce[eqns, vars] simplifies the equations eqns, attempting to solve for the variables vars. The equations generated by Reduce are equivalent to eqns, and contain all the possible solutions. Any variable in eqns but not vars is regarded as a parameter.
Reduce[eqns] treats all variables encountered as vars above.
Reduce[eqns, vars, elims] simplifies the equations, trying to eliminate the variables elims.

Solve[eqns, vars] attempts to solve an equation or set of equations for the variables vars. Any variable in eqns but non vars is regarded as a parameter.
Solve[eqns] treats all variables encountered as vars above.
Solve[eqns, vars, elims] attempts to solve the equations for vars, eliminating the variables elims.

Sostituzioni

Replace[expr, rules] applies a rule or list of rules in an attempt to transform the entire expression expr.

expr /. rules applies a rule or list of rules in an attempt to transform each subpart of an expression expr.

expr //. rules repeatedly performs replacements until expr no longer changes.

lhs -> rhs represents a rule that transforms lhs to rhs.

lhs :> rhs represents a rule that transforms lhs to rhs, evaluating rhs only when the rule is used.
lhs :> rhs /; test represents a rule which applies only if the evaluation of test yields True.

Costanti

E is the exponential constant e (base of natural logarithms), with numerical value 2.71828....

I represents the imaginary unit Sqrt[-1].

Pi is pi, with numerical value 3.14159....

Degree gives the number of radians in one degree. It has a numerical value of Pi/180.

Infinity is a symbol that represents a positive infinite quantity.

ComplexInfinity represents a quantity with infinite magnitude, but undetermined complex phase.

DirectedInfinity[ ] represents an infinite numerical quantity whose direction in the complex plane is unknown.
DirectedInfinity[z] represents an infinite numerical quantity that is a positive real multiple of the complex number z.

Indeterminate is a symbol that represents a numerical quantity whose magnitude cannot be determined.

Funzioni elementari

Exp[z] is the exponential function.

Log[z] gives the natural logarithm of z (logarithm to base E).
Log[b, z] gives the logarithm to base b.

Cos[z] gives the cosine of z.

Sin[z] gives the sine of z.

Tan[z] gives the tangent of z.

Sec[z] gives the secant of z.

Csc[z] gives the cosecant of z.

Cot[z] gives the cotangent of z.

ArcCos[z] gives the arc cosine of the complex number z.

ArcSin[z] gives the arc sine of the complex number z.

ArcTan[z] gives the inverse tangent of z.
ArcTan[x, y] gives the inverse tangent of y/x where x and y are real, taking into account which quadrant the point (x, y) is in.

ArcSec[z] gives the arc secant of the complex number z.

ArcCsc[z] gives the arc cosecant of the complex number z.

ArcCot[z] gives the arc cotangent of the complex number z.

Sinh[z] gives the hyperbolic sine of z.

Cosh[z] gives the hyperbolic cosine of z.

Tanh[z] gives the hyperbolic tangent of z.

Sech[z] gives the hyperbolic secant of z.

Csch[z] gives the hyperbolic cosecant of z.

Coth[z] gives the hyperbolic cotangent of z.

ArcSinh[z] gives the inverse hyperbolic sine of the complex number z.

ArcCosh[z] gives the inverse hyperbolic cosine of the complex number z.

ArcTanh[z] gives the inverse hyperbolic tangent of the complex number z.

ArcSech[z] gives the inverse hyperbolic secant of the complex number z.

ArcCsch[z] gives the inverse hyperbolic cosecant of the complex number z.

ArcCoth[z] gives the inverse hyperbolic cotangent of the complex number z.

Sqrt[z] gives the square root of z.

Abs[z] gives the absolute value of the real or complex number z.

Arg[z] gives the argument of the complex number z.

Re[z] gives the real part of the complex number z.

Im[z] gives the imaginary part of the complex number z.

Conjugate[z] gives the complex conjugate of the complex number z.

Calcolo differenziale

f' represents the derivative of a function f of one argument.
Derivative[n1, n2, ...][f] is the general form, representing a function obtained from f by differentiating n1 times with respect to the first argument, n2 times with respect to the second argument, and so on.

D[f, x] gives the partial derivative of f with respect to x.
D[f, {x, n}] gives the nth partial derivative with respect to x.
D[f, x1, x2, ...] gives a mixed derivative.

Dt[f, x] gives the total derivative of f with respect to x.
Dt[f] gives the total differential of f.
Dt[f, {x, n}] gives the nth total derivative with respect to x.
Dt[f, x1, x2, ...] gives a mixed total derivative.

DSolve[eqn, y[x], x] solves a differential equation for the functions y[x], with independent variable x.
DSolve[{eqn1, eqn2, ...}, {y1[x1, ...], ...}, {x1, ...}] solves a list of differential equations.

Integrate[f,x] gives the indefinite integral of f with respect to x.
Integrate[f,{x,xmin,xmax}] gives the definite integral.
Integrate[f,{x,xmin,xmax},{y,ymin,ymax}] gives a multiple integral.

Limit[expr, x->x0] finds the limiting value of expr when x approaches x0.

Series[f, {x, x0, n}] generates a power series expansion for f about the point x = x0 to order (x - x0)^n.
Series[f, {x, x0, nx}, {y, y0, ny}] successively finds series expansions with respect to y, then x.

Vettori e matrici

VectorQ[expr] gives True if expr is a list, none of whose elements are themselves lists, and gives False otherwise.
VectorQ[expr, test] gives True only if test yields True when applied to each of the elements in expr.

MatrixQ[expr] gives True if expr is a list of lists that can represent a matrix, and gives False otherwise.
MatrixQ[expr, test] gives True only if test yields True when applied to each of the matrix elements in expr.

a.b.c or Dot[a, b, c] gives products of vectors, matrices and tensors.

Det[m] gives the determinant of the square matrix m.

DiagonalMatrix[list] gives a matrix with the elements of list on the leading diagonal, and 0 elsewhere.

Dimensions[expr] gives a list of the dimensions of expr.
Dimensions[expr, n] gives a list of the dimensions of expr down to level n.

CharacteristicPolynomial[m, x] gives the characteristic polynomial defined by the square matrix m and the variable x. The result is normally equivalent to Det[m - x IdentityMatrix[Length[m]]].

Eigenvalues[m] gives a list of the eigenvalues of the square matrix m.

Eigenvectors[m] gives a list of the eigenvectors of the square matrix m.

IdentityMatrix[n] gives the n X n identity matrix.

Inverse[m] gives the inverse of a square matrix m.

LinearSolve[m, b] gives the vector x which solves the matrix equation m.x==b.

Minors[m, k] gives a matrix consisting of the determinants of all k X k submatrices of m.

RowReduce[m] gives the row-reduced form of the matrix m.