Chapter 17
Symbolic Calculation

This chapter describes symbolic processing in Mathcad. The chapter includes the following sections:

What is symbolic math?

An overview of Mathcad's symbolic math features.

Live symbolic evaluation

Using the symbolic equal sign to perform live symbolic transformations.

Using the Symbolics menu

Using menu commands to perform symbolic transformations.

Symbolic algebra

Manipulating expressions algebraically.

Symbolic calculus

Evaluating indefinite integrals, derivatives, and limits symbolically.

Solving equations symbolically

Algebraic solution of equations or systems of equations.

Symbolic matrix manipulation

Finding the symbolic transpose, inverse, and determinant of a matrix.

Symbolic transforms

Fourier, Laplace and *z*-transforms.

Symbolic optimization

Symbolically simplifying complex expressions before numerically evaluating them.

Using functions and variables

Differences between the symbolic and numerical processors and how they work with variables and functions.

Limits to symbolic processing

Difficulties you may encounter in symbolic processing.

What is symbolic math?

Elsewhere in this *User's Guide*, you've seen Mathcad engaging in *numerical* mathematics. This means that whenever you evaluate an expression, Mathcad returns one or more *numbers*, as shown at the top of Figure 17-1. Although these numbers are quite useful, they may provide little insight into the underlying relationship between the components in an expression.

When Mathcad engages in *symbolic* mathematics, however, the result of evaluating an expression is generally another expression, as shown in the bottom of Figure 17-1. The form of this second expression is to a great extent under your control. You can factor the original expression, integrate it, expand it into a series, and so on. The way you control the form of that second expression is the subject of this chapter.

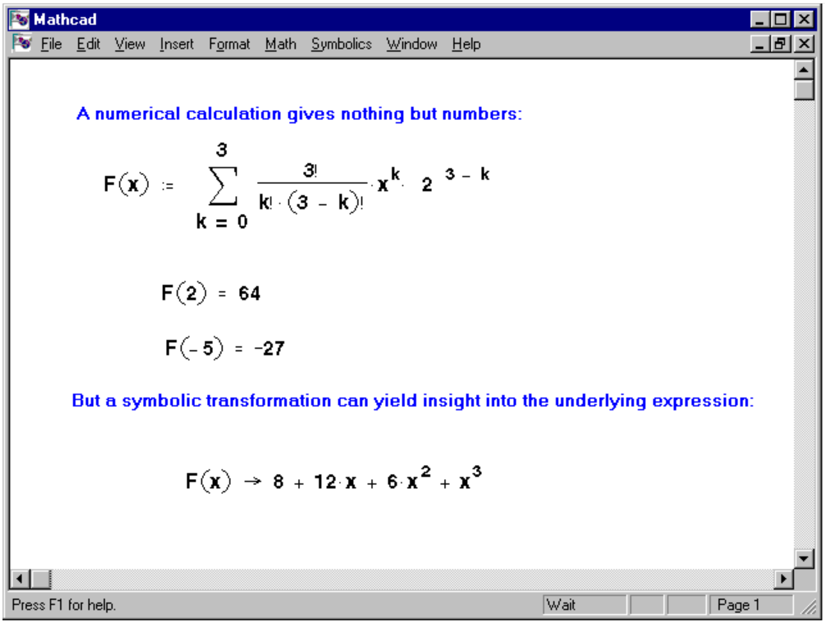


Figure 17-1: A numerical and symbolic evaluation of the same expression.

There are three ways to perform a symbolic transformation on an expression.

- You can use the symbolic equal sign as described in the section "Live symbolic evaluation" on page 359. This method feels very much as if you're engaging in numerical math.
- If you need more control over the symbolic transformation, you can use symbolic keywords with the symbolic equal sign, or you can use commands from the **Symbolics** menu.

- You can make the numerical and symbolic processors work together; the latter simplifying an expression behind the scenes so that the former can work with it more efficiently. This is discussed in the section "Symbolic optimization" on page 394.

Symbolic processing also raises some subtle issues concerning the use of functions and variables. These are described in the section "Using functions and variables" on page 396.

Finally, there are some fundamental limits inherent in computer based symbolic processing generally. These arise because nobody really knows how the human brain does symbolic processing. As a result, nobody really knows how to teach a computer to do it. These limits are discussed in the section "Limits to symbolic processing" on page 401.

Live symbolic evaluation

The symbolic equal sign provides a way to extend Mathcad's live document interface beyond the numerical evaluation of expressions. You can think of it as being analogous to the equal sign "=". Unlike the equal sign, which always gives a number on the right hand side, the symbolic equal sign is capable of giving expressions. You can use it to symbolically evaluate expressions, variables, functions, or programs.

To use the symbolic equal sign:

- Make sure the **Automatic Calculation** command on the **Math** menu has a check-mark beside it. If it doesn't, choose it from the menu.
- Make sure the **Automatic Calculation** command on the **Math** menu has a check-mark beside it.
- Enter the expression you want to evaluate.

$$\frac{d}{dx}\left(x^3 - 2 \cdot y \cdot x\right)$$

- Press [**`Ctrl`**]**.** (the control key followed by a period). Mathcad displays an arrow, "→".

$$\frac{d}{dx}\left(x^3 - 2 \cdot y \cdot x\right) \rightarrow$$

- Click outside the expression. Mathcad displays a simplified version of the original expression. If an expression cannot be simplified further, Mathcad simply repeats it to the right of the arrow.

$$\frac{d}{dx}\left(x^3 - 2 \cdot y \cdot x\right) \rightarrow 3 \cdot x^2 - 2 \cdot y$$

The symbolic equal sign is a live operator just like any Mathcad operator. When you make a change anywhere above or to the left of it, Mathcad updates the result. The symbolic equal sign "knows" about previously defined functions and variables and uses them wherever appropriate. You can force the symbolic equal sign to ignore prior definitions of functions and variables by defining them recursively just before you evaluate them as shown in Figure 17-5.

Figure 17-2 shows some examples of how to use the "→". Note that the "→" only applies to an entire expression. You cannot, for example, use the "→" to transform only part of an expression.

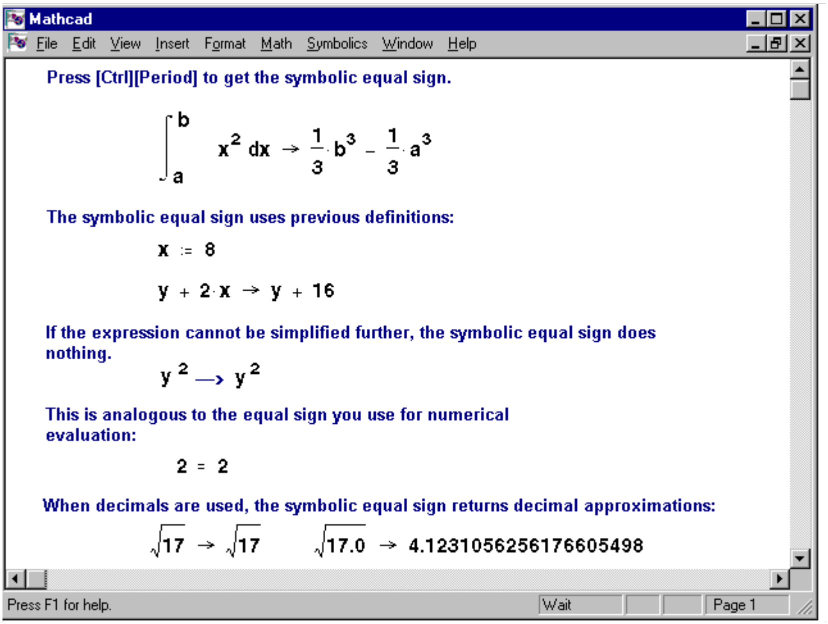


Figure 17-2: Using the symbolic equal sign.

Figure 17-2 also illustrates the fact that the symbolic processor treats numbers containing a decimal point differently from numbers without a decimal point. The general rule is as follows:

- When you send numbers with decimal points to the symbolic processor, any numerical results you get back will be decimal approximations to the exact answer.
- When you send numbers without decimal points to the symbolic processor, any numerical results you get back will be expressed without decimal points whenever possible.

In Figure 17-2, note how $\sqrt{17}$ comes back unchanged since there is no rational square root of 17. But $\sqrt{17.0}$ comes back as a decimal approximation to the irrational number $\sqrt{17}$.

When a symbolic operation gives an approximate decimal answer, this answer is always displayed with 20 significant digits. Although this display is not affected by Mathcad's local or global numerical formats, you can use the **float** keyword described in the next section to control the number of significant digits displayed.

Customizing the symbolic equal sign using keywords

The “→” takes the left-hand side and places a simplified version of it on the right-hand side. By default, it simplifies the left-hand side just as if you had chosen **Evaluate⇒Symbolically** from the **Symbolics** menu (see “Using the Symbolics menu” on page 366).

Of course, exactly what “simplify” means is a matter of opinion. As a result, you can, to a limited extent, control how the “→” transforms the expression by using one of the following keywords. To do so:

- Enter the expression you want to evaluate.

  $(x+y)^3$

- Press [**`Ctrl`**] [**`Shift`**]**`.`** (hold down the control and shift keys and type a period). Mathcad displays a placeholder to the left of the arrow, “→”.

  $(x+y)^3 \ \blacksquare \ \rightarrow$

- Click on the placeholder to the left of the arrow and type any of the keywords from the following table. If the keyword requires any additional arguments, separate the arguments from the keyword with commas.

  $(x+y)^3$ expand $\rightarrow$

- Press [**`Enter`**] to see the result.

  $(x+y)^3$ expand $\rightarrow x^3 + 3\cdot x^2\cdot y + 3\cdot x\cdot y^2 + y^3$

Another way to use a keyword is to enter the expression you want to evaluate and click on a keyword button from the Symbolic Keywords palette. This will insert the keyword, placeholders for any additional arguments, and the arrow, “→”. You just need to press [**`Enter`**] to see the result.

Keyword	**Function**
`complex`	Tells Mathcad to carry out symbolic evaluation in the complex domain. Result will usually be in the form $a+i\cdot b$.
`float,`*`m`*	Tells Mathcad to display a floating point value with *m* places of precision whenever possible. If the argument *m*, an integer, is omitted, the precision is 20.
`simplify`	Simplifies an expression, performing arithmetic, canceling common factors, and using basic trigonometric and inverse function identities.
`expand,`*`expr`*	Expands all powers and products of sums in an expression except for the subexpression *expr*. The argument *expr* is optional. The entire expression is expanded if the argument *expr* is omitted.
`factor,`*`expr`*	Factors an expression into a product, if the entire expression can be written as a product. Factors with respect to *expr*, a

	single radical or a list of radicals separated by commas. The argument *expr* is optional.
solve,*var*	Solves an equation for the variable *var* or solves a system of equations for the variables in a vector *var*.
collect,*var1,var2,...*	Collects like terms with respect to the variables or subexpressions *var1* through *varn*.
coeffs,*var*	Finds coefficients of an expression when it is rewritten as a polynomial in the variable or subexpression *var*.
substitute,*var1=var2*	Replaces all occurrences of a variable *var1* with an expression or variable *var2*. Press [**Ctrl**] **=** for the equal sign.
series,*var=z,m*	Expands an expression in one or more variables, *var*, around the point *z*. The order of expansion is *m*. Arguments *z* and *m* are optional. By default, the expansion is taken around zero and is a polynomial of order six.
convert,parfrac,*var*	Converts an expression to a partial fraction expansion in the variable *var*.
fourier,*var*	Evaluates the Fourier transform of an expression with respect to the variable *var*.
invfourier,*var*	Evaluates the inverse Fourier transform of an expression with respect to the variable *var*.
laplace,*var*	Evaluates the Laplace transform of an expression with respect to the variable *var*.
invlaplace,*var*	Evaluates the inverse Laplace transform of an expression with respect to the variable *var*.
ztrans,*var*	Evaluates the *z*-transform of an expression with respect to the variable *var*.
invztrans,*var*	Evaluates the inverse *z*-transform of an expression with respect to the variable *var*.
assume,*constraint*	Tells Mathcad to impose constraints on one or more variables according to the expression *constraint*.

Note that many of the keywords take at least one additional argument, typically the name of a variable with respect to which you are performing the symbolic operation. Some of the arguments listed with the keywords are optional.

For more information on each of these keywords, see the sections entitled “Symbolic algebra” on page 368, “Symbolic calculus” on page 380, and “Symbolic transforms” on page 390.

Figure 17-3 shows some sample uses of keywords. Note that keywords are case sensitive and must therefore be typed exactly as shown. They are not, however, font sensitive.

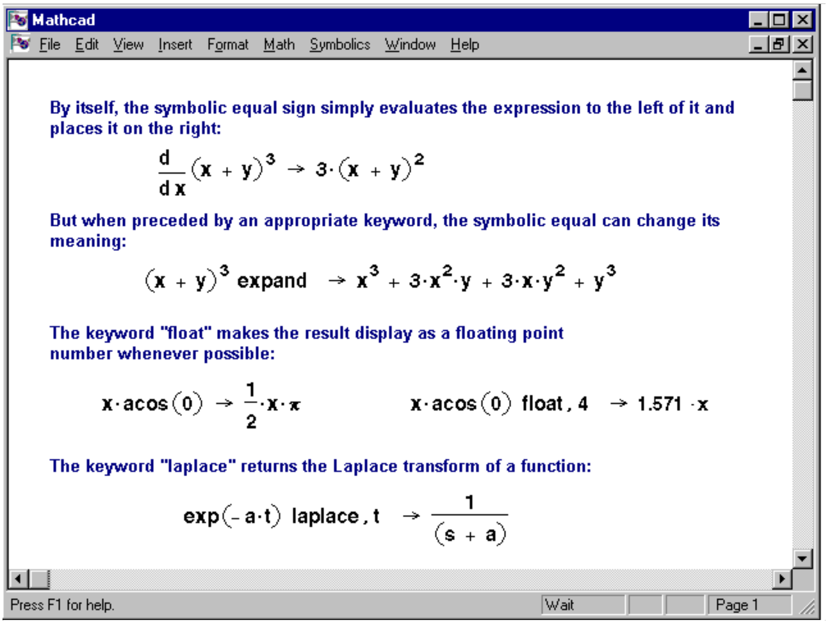


Figure 17-3: Using keywords with a symbolic evaluation sign.

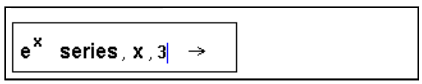
In some cases, you may want to perform two or more types of symbolic evaluation consecutively on an expression. Mathcad allows you to apply several symbolic keywords to a single expression. There are two ways of applying multiple keywords. The method you choose depends on whether you want to see the results from each keyword or only the final result.

To apply several keywords and see the results from each:

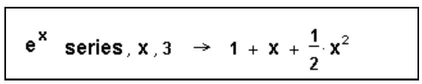
- Enter the expression you want to evaluate.

- Press [**Ctrl**] [**Shift**]**.** (hold down the control and shift keys and type a period). Mathcad displays a placeholder to the left of the arrow, "→".

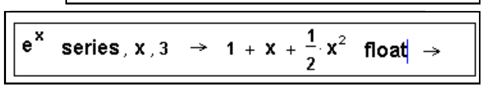
- Enter the first keyword into the placeholder to the left of the arrow, including any comma-delimited arguments the keyword takes.



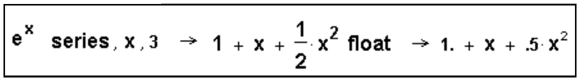
- Press [**Enter**] to see the result from the first keyword.



- Click on the result and press [**Ctrl**] [**Shift**]**.** again. Enter a second keyword into the placeholder.



- Press [**Enter**] to see the result from the second keyword.



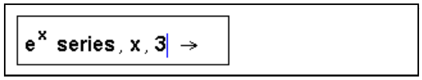
Continue applying keywords to the intermediate results in this manner.

To apply several keywords and see only the final result:

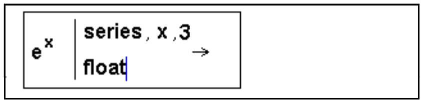
- Enter the expression you want to evaluate.

- Press [**Ctrl**][**Shift**]**.** so that Mathcad displays a placeholder to the left of the arrow, “→”.

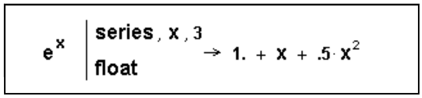
- Enter the first keyword into the placeholder, including any comma-delimited arguments it takes.



- Instead of pressing [**Enter**] to see the result, press [**Ctrl**][**Shift**]**.** again and enter a second keyword into the placeholder. The second keyword is placed immediately below the first keyword.



Continue adding keywords by pressing [**Ctrl**] [**Shift**]**.** after each one. Press [**Enter**] to see the final result.



These two methods of applying multiple keywords to an expression are demonstrated in Figure 17-4.

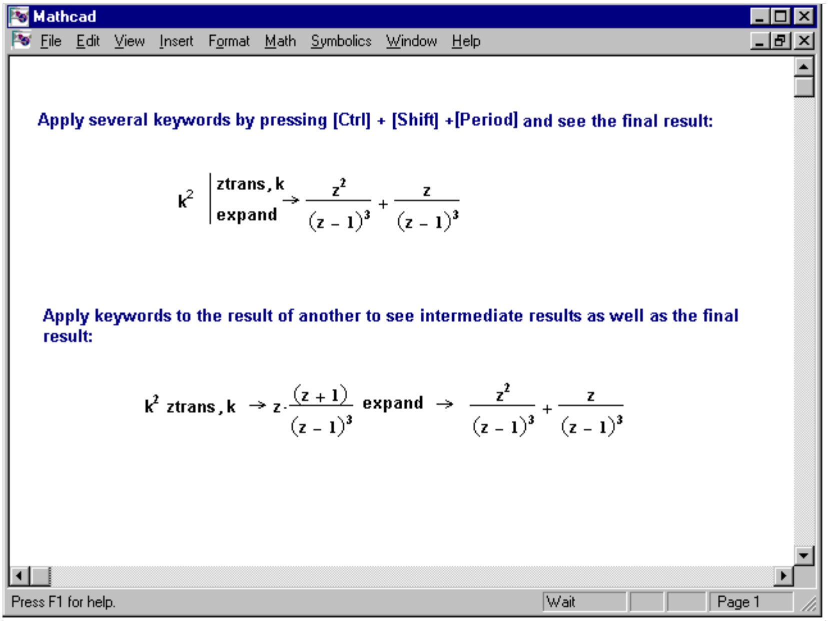


Figure 17-4: Using multiple keywords.

Ignoring previous definitions

When you use the symbolic equal sign to evaluate an expression, Mathcad checks all the variables and functions making up that expression to see if they've been defined earlier in the worksheet. If Mathcad does find a definition, it uses it. Any other variables and functions are evaluated symbolically.

There are two exceptions to this. In evaluating an expression made up of previously defined variables and functions, Mathcad *ignores* prior definitions:

- when the variable has been defined recursively, or
- when the variable has been defined as a range variable.

These are illustrated in Figure 17-5.

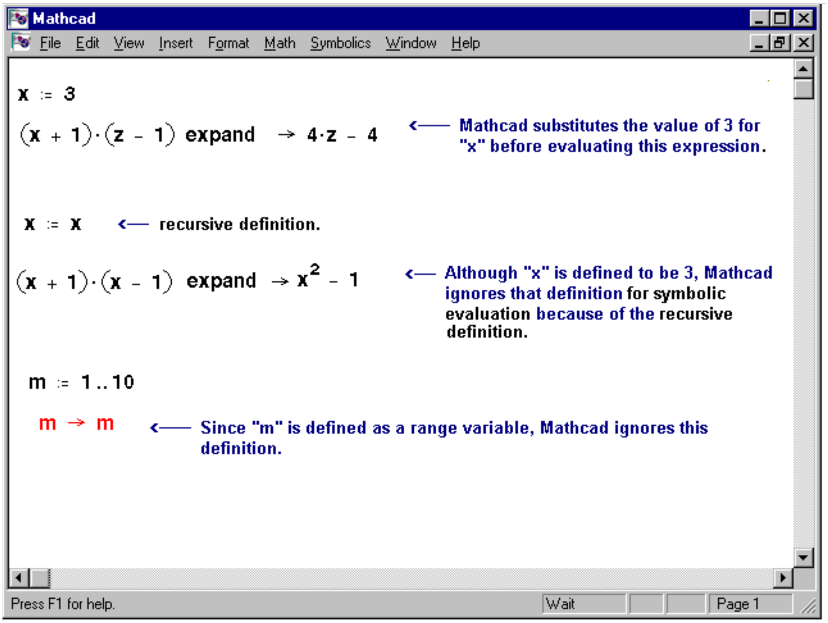


Figure 17-5: Defining a variable in terms of itself makes the symbolic processor ignore any previous definitions.

Using the Symbolics menu

One advantage to using the symbolic equal sign discussed in the last section is that it is “live,” just like the numerical processing in Mathcad. That is, Mathcad checks all the variables and functions making up the expression being evaluated to see if they’ve been defined earlier in the worksheet. If Mathcad does find a definition, it uses it. Any other variables and functions are evaluated symbolically. Later on, whenever you make a change to the worksheet, the results automatically update. This is useful when the symbolic and numerical equations in the worksheet are tied together.

There may be times, however, when a symbolic calculation is quite separate from the rest of your worksheet and does not need to be tied to any previous definitions. In these cases, you can use commands from the **Symbolics** menu. These commands are not live: you apply them on a case by case basis to selected expressions, they do not “know” about previous definitions, and they do not automatically update.

The basic steps for using the **Symbolics** menu are the same for all the menu commands:

- Place whatever math expression you want to evaluate between the two editing lines or, for some commands, click on a variable in the expression.
- Choose the appropriate command from the **Symbolics** menu.

- Mathcad will place the evaluated expression into your document.

For example, to evaluate an expression symbolically using the **Symbolics** menu, follow these steps:

- Enter the expression you want to evaluate.

$$\frac{d}{dx}\left(x^3 - 2 \cdot y \cdot x\right)$$

- Surround the expression with the editing lines.

$$\frac{d}{dx}\left(x^3 - 2 \cdot y \cdot x\right)$$

- Choose **Evaluate⇒Symbolically** from the **Symbolics** menu.
- Mathcad will place the evaluated expression into your worksheet. The location of the result in relation to the original expression depends on the Evaluation Style you've selected, as described below.

$$3 \cdot x^2 - 2 \cdot y$$

The sections "Symbolic algebra" on page 368, "Symbolic calculus" on page 380, and "Solving equations symbolically" on page 385 describe the various **Symbolics** menu commands in detail.

Displaying symbolic results

If you're using the symbolic equal sign, "→", the result of a symbolic transformation will always go to the right of the "→". However, when you use the **Symbolics** menu, you can tell Mathcad to place the symbolic results in one of the following ways:

- The symbolic result can go below the original expression.
- The symbolic result can go to the right of the original expression.
- The symbolic result can simply replace the original expression.

In addition, you can also choose whether or not you want Mathcad to generate text describing what had to be done to get from the original expression to the symbolic result. This text will go between the original expression and the symbolic result, in effect creating a narrative for the symbolic evaluation. These text regions are referred to as "evaluation comments."

To control both the placement of the symbolic result and the presence of narrative text, choose **Evaluation Style** from the **Symbolics** menu to bring up the "Evaluation Style" dialog box. The check box at the top of the dialog box shows whether Mathcad will automatically generate evaluation comments at each step of the evaluation. Click in this box to toggle these comments on or off.

The three option buttons control where symbolic results are placed. These options do the following:

- "Show evaluation steps vertically, inserting lines" is useful when you expect lengthy intermediate results and you want to reserve an entire line for them.

- “Show evaluation steps vertically, without inserting lines” is useful when you want to show two parallel derivations side by side. In this mode, you can position expressions arbitrarily. New answers may, however, overwrite old ones.
- “Show evaluation steps horizontally” is useful if you want to place the symbolic result to the right of the expression being transformed.

Sometimes you don’t care about saving the steps of a derivation. You may just want to transform an expression in place, for example to make a substitution, or to factor the numerator of a fraction. In this case, choose **Evaluation Style** from the **Symbolics** menu and click in the check box for “Evaluate in Place.” This tells Mathcad to replace the old expression with the new one. In this mode, evaluation comments are inappropriate and therefore omitted altogether.

See Figure 17-6 for some examples.

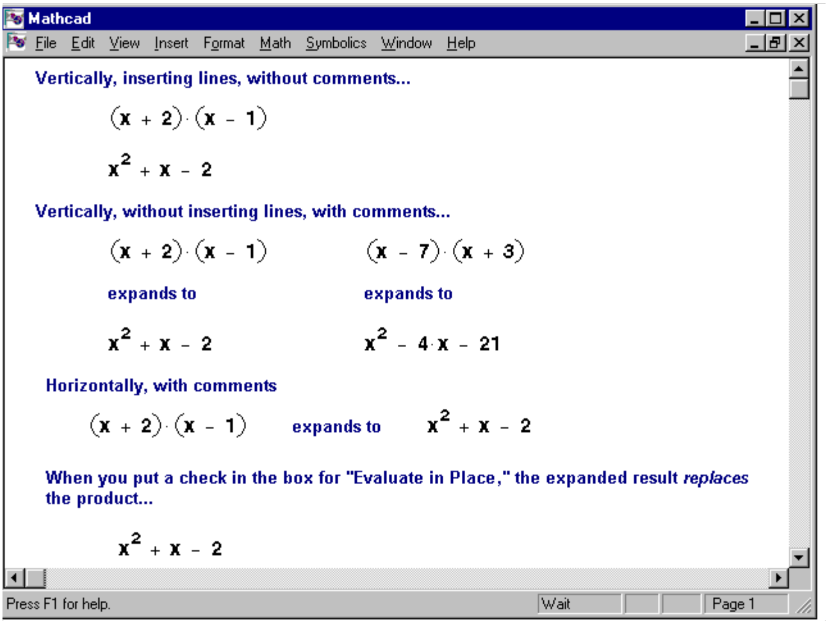


Figure 17-6: Placement of symbolic results and comments.

Symbolic algebra

Mathcad allows you to manipulate an expression algebraically using either keywords and the symbolic equal sign, or menu commands from the **Symbolics** menu. Most of the examples in this section demonstrate “live” symbolic operations using symbolic keywords, but you may apply commands from the **Symbolics** menu to expressions on a case by case basis if you prefer. Keep in mind that, unlike the keyword-modified expressions, expressions modified by commands from the **Symbolics** menu do not update automatically, as described in the section “Using the Symbolics menu” on page 366.

Complex evaluation

When evaluating expressions containing complex numbers, you may want to use the keyword **complex**:

- Enter the expression you want to evaluate.
- Press [**Ctrl**] [**Shift**]**.** (hold down the control and shift keys and type a period). Mathcad displays a placeholder to the left of the arrow, “→”.
- Type **complex** into the placeholder.
- Press [**Enter**] to see the result.

Mathcad will assume all the terms in the expression are written in the form $a + b \cdot i$. The results will also be in this form. Figure 17-7 shows an example.

Another way to evaluate an expression in the complex domain is to enclose the expression between the editing lines and choose **Evaluate⇒Complex** from the **Symbolics** menu.

Floating point evaluation

Ordinarily, the symbolic processor returns results by rearranging variables. Thus, when Mathcad evaluates an expression involving π or *e*, it will usually return another expression involving π or exp(*x*). To force Mathcad to return a number instead, use the keyword **float**:

- Enter the expression you want to evaluate.
- Press [**Ctrl**] [**Shift**]**.** (hold down the control and shift keys and type a period). Mathcad displays a placeholder to the left of the arrow, “→”.
- Type **float** into the placeholder.
- Press [**Enter**] to see the result.

Mathcad by default returns a result with up to 20 digits to the right of the decimal point. To specify a different number of digits for the result, follow the keyword **float** with a comma and an integer between 0 and 250. Figure 17-7 shows an example.

Another way to perform floating point evaluation on an expression is to enclose the expression between the editing lines and choose **Evaluate⇒Floating Point** from the **Symbolics** menu. This brings up a dialog box in which you can specify the number of digits to the right of the decimal point.

Constrained evaluation

When evaluating some expressions, you may want to force Mathcad to make certain assumptions about the variables involved. For example, you might want Mathcad to assume x is real when evaluating the square root of x. To impose constraints on the variables in an expression, use the keyword **`assume`**:

- Enter the expression you want to evaluate.
- Press [**`Ctrl`**][**`Shift`**]**`.`** (hold down the control and shift keys and type a period). Mathcad displays a placeholder to the left of the arrow, "→".
- In the placeholder, type **`assume`** followed by a comma and a constraining equation such as **`x<10`**.
- Press [**`Enter`**] to see the result.

You can also use the **assume** keyword to tell Mathcad to consider a variable to be real or falling in a certain range of real values. To do so, you can use the following "modifiers":

Modifiers for "assume"

`var`**`=real`**	Evaluates the expression on the assumption that the variable *var* is real.
`var`**`=RealRange(`*****`a`*****`,`*****`b`*****`)`**	Evaluates on the assumption that all the indeterminates are real and are between *a* and *b*, where *a* and *b* are real numbers or infinity ([**`Ctrl`**]**`Z`**).

To use a modifier, separate it from the **assume** keyword with a comma. For example, to use "x=real" as a modifier with the **assume** keyword on an expression:

- Enter the expression to simplify.
- Press [**`Ctrl`**][**`Shift`**]**`.`** (hold down the control and shift keys and type a period). Mathcad displays a placeholder to the left of the arrow, "→".
- Enter **`assume,x=real`** into the placeholder (press [**`Ctrl`**]**`=`** for the equal sign).
- Press [**`Enter`**] to see the result.

The last example in Figure 17-7 illustrates how an integral can be made to converge by assuming a variable is positive and greater than 1. Note that in order to specify more than one condition, you simply separate the conditions with a comma.

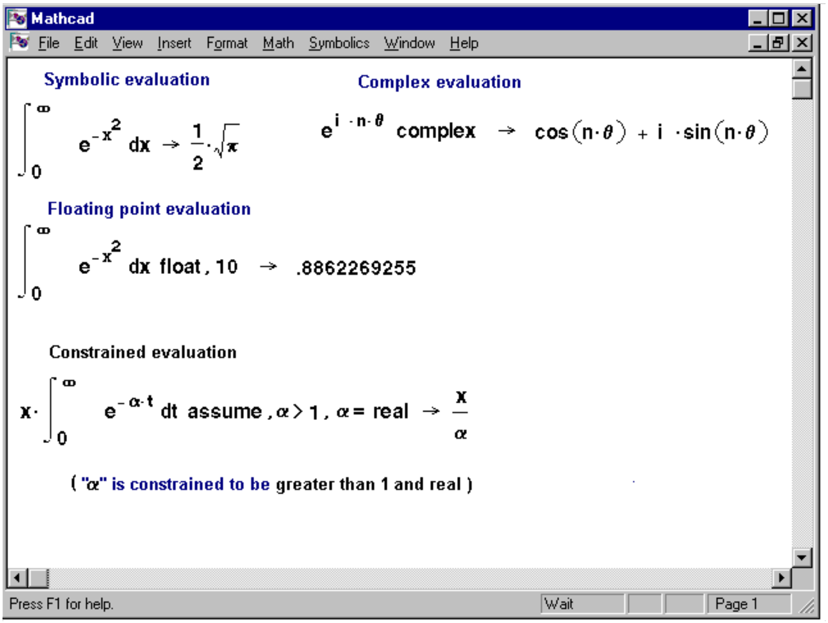


Figure 17-7: Evaluating expressions symbolically.

Simplifying an expression

To force Mathcad to carry out basic algebraic and trigonometric simplification of a selected expression, use the keyword **simplify**:

- Enter the expression you want to evaluate.
- Press [`Ctrl`] [`Shift`]`.` (hold down the control and shift keys and type a period). Mathcad displays a placeholder to the left of the arrow, "→".
- Type `simplify` into the placeholder.
- Press [`Enter`] to see the result.

When the symbolic processor simplifies an expression, it performs arithmetic, cancels common factors, uses basic trigonometric and inverse function identities, and simplifies square roots and powers.

To control the simplification performed by the **simplify** keyword, you can use the following "modifiers":

Modifiers for "simplify"

`assume=real`	Simplifies on the assumption that all the indeterminates in the expression are real.
`assume=RealRange(`*a*`,`*b*`)`	Simplifies on the assumption that all the indeterminates are real and are between *a* and *b*, where *a* and *b* are real numbers or infinity ([`Ctrl`]`Z`).

`trig`	Simplifies a trigonometric expression by applying only the following identities: $\sin(x)^2 + \cos(x)^2 = 1$ $\cosh(x)^2 - \sinh(x)^2 = 1$ It does not simplify the expression by simplifying logs, powers, or radicals.

To use a modifier, separate it from the **simplify** keyword with a comma. For example, to use the “trig” modifier with the **simplify** keyword on an expression:

- Enter the expression to simplify.
- Press [**`Ctrl`**][**`Shift`**]**`.`** (hold down the control and shift keys and type a period). Mathcad displays a placeholder to the left of the arrow, “→”.
- Enter **`simplify,trig`** into the placeholder.
- Press [**`Enter`**] to see the result.

Figure 17-8 shows some examples using the **simplify** keyword with and without additional modifiers.

Note that you can also simplify an expression by placing it between the two editing lines and choosing **Simplify** from the **Symbolics** menu. This method is useful when you want to simplify parts of an expression. Mathcad may sometimes be able to simplify parts of an expression even when it cannot simplify the entire expression. If simplifying the entire expression doesn't give the answer you want, try selecting subexpressions and choosing **Simplify** from the **Symbolics** menu. If Mathcad can't simplify an expression any further, you'll just get the original expression back as the answer.

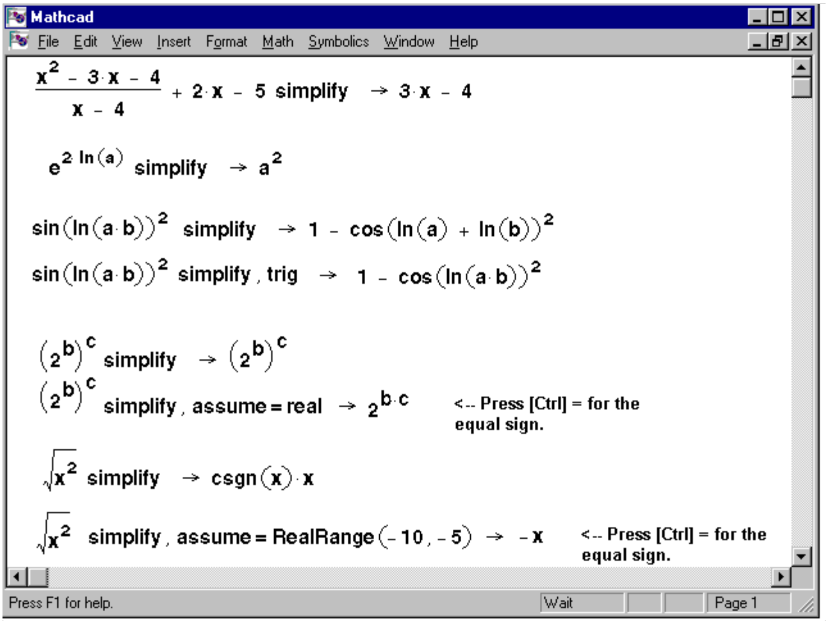


Figure 17-8: Some results of simplifying. Modifiers such as "trig" give more control over the simplification.

In general, when you simplify an expression, the simplified result will have the same numerical behavior as the original expression. However, when the expression includes functions with more than one branch, such as square root or the inverse trigonometric functions, the symbolic answer may differ from a numerical answer. For example, simplifying $\mathrm{asin}(\sin(\theta))$ yields θ, but this equation holds true numerically in Mathcad only when θ is a number between $-\pi/2$ and $\pi/2$.

Expanding an expression

To expand all powers and products of sums in an expression, use the keyword **expand**:

- Enter the expression you want to expand.
- Press [`Ctrl`][`Shift`]`.` (hold down the control and shift keys and type a period). Mathcad displays a placeholder to the left of the arrow, "→".
- Type **`expand`** into the placeholder.
- Press [`Enter`] to see the result.

Mathcad expands all powers and products of sums in the selected expression. If the expression is a fraction, the numerator will be expanded and the expression will be written as a sum of fractions. Sines, cosines and tangents of sums of variables, or integer multiples of variables will be expanded as far as possible into expressions involving only sines and cosines of single variables. If you don't want certain subexpressions to be expanded, follow the **expand** keyword with a comma and the expressions. See Figure 17-10 for some examples.

Another way to expand an expression is to enclose the expression between the editing lines and choose **Expand** from the **Symbolics** menu.

Expanding an expression to a series

To expand an expression to a series, use the keyword **series**:

- Enter the expression you want to expand.
- Press [**Ctrl**] [**Shift**]**.** (hold down the control and shift keys and type a period). Mathcad displays a placeholder to the left of the arrow, "→".
- In the placeholder, type **series** followed by a comma and the variable or expression for which you want to find a series expansion.
- Press [**Enter**] to see the result.

Mathcad will then generate a series of order 6. To specify a different order of expansion, follow the variable of expansion with a comma and an appropriate integer. The order is the order of the error term in the expansion. For example, if Mathcad expands sin(x) to a series in x, it returns an expansion of the sine function in powers of x in which the highest power is x^5. The error is thus $O(x^6)$.

Mathcad will find Taylor series (series in nonnegative powers of the variable) for functions that are analytic at 0, and Laurent series for functions that have a pole of finite order at 0. To develop a series with a center other than 0, the argument to the **series** keyword should be of the form *var*=*z*, where *z* is any real or complex number. For example, **series, x=1** expands around the point x=1. Press [**Ctrl**]**=** for the equal sign.

To expand a series around more than one variable, follow the series keyword with a comma and the variables, separated from each other by commas. The last example in Figure 17-9 shows an expression expanded around x and y.

Figure 17-9 shows some examples of expanded expressions.

Another way to generate a series expansion is to enter the expression and click on a variable for which you want to find a series expansion. Then choose **Variable⇒Expand to Series** from the **Symbolics** menu. A dialog box will prompt you for the order of the series. This command is limited to a series in a single variable; any other variables in the expression will be treated as constants. The results also contain the error term using the *O* notation. Before you use the series for further calculations you will need to delete this error term.

In using the approximations you get from the symbolic processor, keep in mind that the Taylor series for a function may converge only in some small interval around the center. Furthermore, functions like *sin* or *exp* have series with infinitely many terms, while the polynomials returned by Mathcad have only a few terms (how many depends on the order you select). Thus, when you approximate a function by the polynomial returned by Mathcad, the approximation will be reasonably accurate close to the center, but may be quite inaccurate for values far from the center.

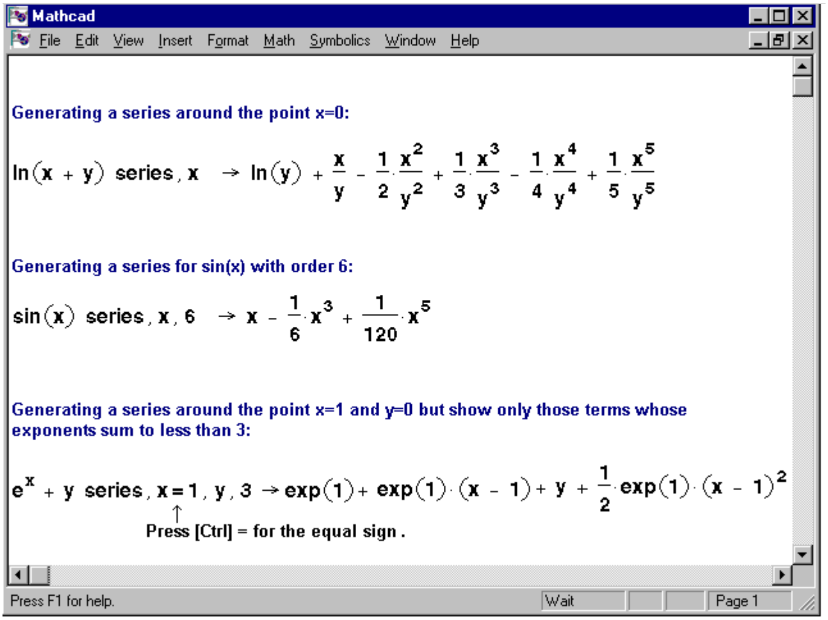


Figure 17-9: Generating a series.

Factoring an expression

To factor an expression, use the keyword **factor**:

- Enter the expression you want to factor.
- Press [`Ctrl`] [`Shift`]`.` (hold down the control and shift keys and type a period). Mathcad displays a placeholder to the left of the arrow, “→”.
- In the placeholder, type `factor`.
- Press [`Enter`] to see the result.

If this expression is a single integer, Mathcad will factor it into powers of primes. Otherwise, Mathcad will attempt to convert the expression into a product of simpler functions. The symbolic processor will combine a sum of fractions into a single fraction and will often simplify a complex fraction with more than one fraction bar.

If you want to factor an expression over certain radicals, follow the **factor** keyword with a comma and the radicals.

When you're simplifying by factoring, you may be able to simplify your expression quite a bit by factoring subexpressions even if the expression taken as a whole can't be factored. To do so, enclose a subexpression between the editing lines and choose **Factor** from the **Symbolics** menu. You can also use this menu command to factor an entire expression, but keep in mind that the **Symbolics** menu commands do not use any previous definitions in your worksheet and do not automatically update.

See the examples in Figure 17-10 for examples of factoring expressions.

Collecting like terms

To simplify an expression by collecting terms containing like powers of a variable:

- Enter the expression.
- Press [**`Ctrl`**] [**`Shift`**] **`.`** (hold down the control and shift keys and type a period). Mathcad displays a placeholder and the arrow, "→".
- In the placeholder, type **`collect`** followed by a comma and the variable or subexpression on which to collect.
- Press [**`Enter`**] to see the result.

The result is a polynomial in the variable or subexpression. The subexpression you select must be a single variable or a built-in function together with its argument.

To collect on more than one variable, follow the **collect** keyword with a comma and the variables on which to collect, separated from each other by commas.

See Figure 17-10 for examples of simplifying expressions by collecting like terms. An alternative method for collecting terms of an expression is to click on a variable in an expression and choose **Collect** from the **Symbolics** menu.

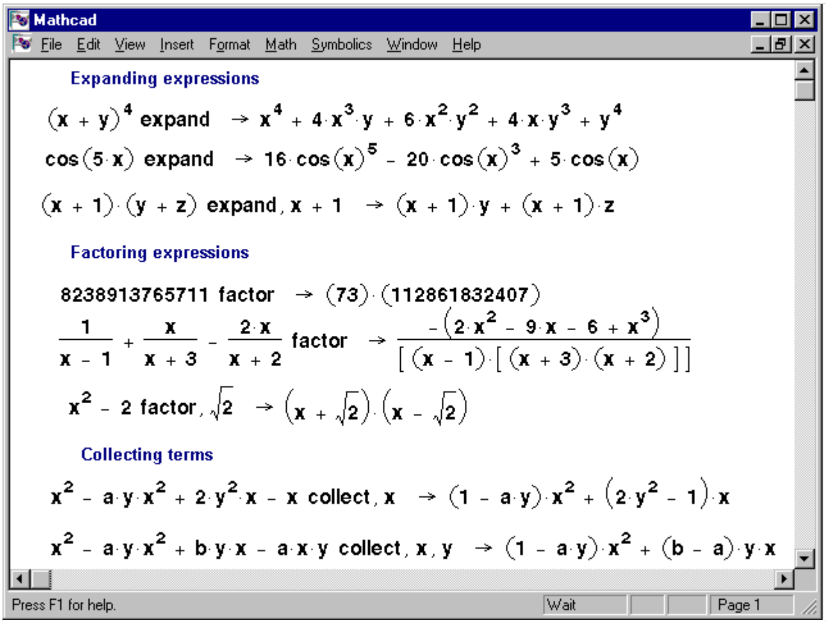


Figure 17-10: Expanding to a polynomial, factoring, and collecting terms.

Partial fraction decomposition

To convert an expression to its partial fraction decomposition, use the keyword **convert**:

- Enter the expression.

- Press [**Ctrl**][**Shift**]**.** (hold down the control and shift keys and type a period). Mathcad displays a placeholder to the left of the arrow, “→”.
- In the placeholder, type **convert,parfrac** followed by a comma and the variable in the denominator of the expression on which to convert.
- Press [**Enter**] to see the result.

The symbolic processor will try to factor the denominator of the expression into linear or quadratic factors having integer coefficients. If it succeeds, it will expand the expression into a sum of fractions with these factors as denominators. All constants in the selected expression must be integers or fractions; Mathcad will not expand an expression that contains decimal points. See Figure 17-11 for some examples.

Another way to convert an expression to a partial fraction is enter the expression and click on a variable in the denominator. Then choose **Variable⇒Convert to Partial Fraction** from the **Symbolics** menu.

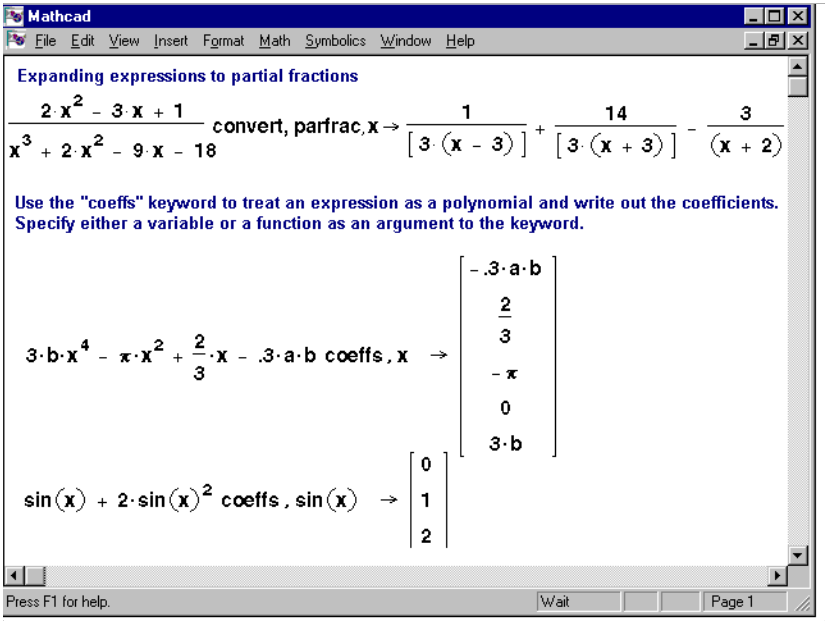


Figure 17-11: Converting expressions to partial fractions and rewriting them as polynomials.

Finding coefficients of a polynomial

Many expressions can be rewritten as polynomials, either in a particular variable or with respect to a subexpression. To force the symbolic processor rewrite an expression as a polynomial and return the coefficients:

- Enter the expression you want to rewrite.
- Press [**Ctrl**][**Shift**]**.** (hold down the control and shift keys and type a period). Mathcad displays a placeholder to the left of the arrow, “→”.

- In the placeholder, type **`coeffs`** followed by a comma and the variable or function in which you want your expression to be regarded as a polynomial.
- Press [**`Enter`**] to see the result.

Mathcad returns a vector containing the coefficients of the equivalent polynomial. The first element of the vector is the constant term and the last element is the coefficient of the highest order term in the expression. Figure 17-11 shows two examples.

Another way to rewrite an expression as a polynomial is to enclose it between the two editing lines and choose **Polynomial Coefficients** from the **Symbolics** menu.

Substituting an expression for a variable

To replace a variable in an expression with another variable or subexpression, use the keyword **substitute**:

- Enter the expression.
- Press [**`Ctrl`**] [**`Shift`**] **`.`** (hold down the control and shift keys and type a period). Mathcad displays a placeholder to the left of the arrow, "→".
- In the placeholder, type **`substitute`** followed by a comma and an expression of the form *var1=var2* where *var1* is a variable and *var2* is a variable or expression. Press [**`Ctrl`**] **`=`** for the equal sign.
- Press [**`Enter`**] to see the result.

Mathcad will replace *var1* with *var2*. If *var1* occurs more than once in the expression you are transforming, Mathcad replaces each occurrence. Figure 17-12 shows some examples.

Note that Mathcad will not substitute a variable for an entire vector or a matrix. You can, however, substitute a scalar expression for a variable that occurs in a matrix. To do so, select the expression that will replace the variable and choose **Copy** from the **Edit** menu. Click on an occurrence of the variable you want to replace and choose **Variable⇒Substitute** from the **Symbolics** menu. You can also use this menu command to perform a substitution in any expression.

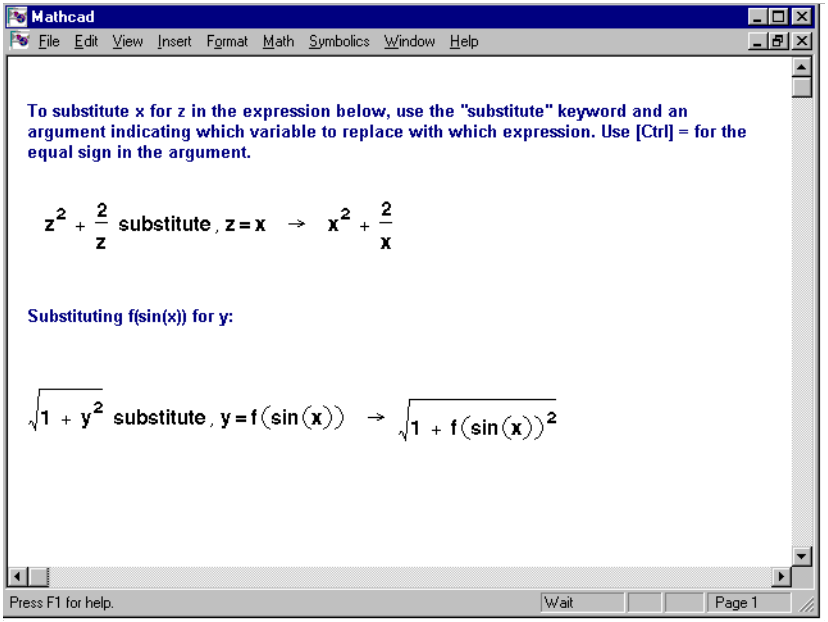


Figure 17-12: Substituting an expression for a variable.

Evaluating a summation

To evaluate a sum symbolically, you can use Mathcad's summation operator and the live symbolic equal sign:

- Create the summation operator by typing [**`Ctrl`**][**`Shift`**]**`4`**.
- Enter the expression you want to sum in the placeholder to the right of the "Σ".
- Enter the index variable and summation range in the placeholders above and below the "Σ" as shown in Figure 17-13.
- Press [**`Ctrl`**]**`.`** (the control key followed by a period). Mathcad displays an arrow, "→".
- Press [**`Enter`**] to see the result.

The procedure is the same for a product over a range, except that you type [**`Ctrl`**][**`Shift`**]**`3`** to get the product operator. If you use numerical limits in a summation or product range, be sure that the upper limit of the range is greater than or equal to the lower limit.

Another way to evaluate a summation is to enclose the summation expression between the editing lines and choose **Evaluate⇒Symbolically** from the **Symbolics** menu.

Figure 17-13 illustrates various results of symbolic evaluation.

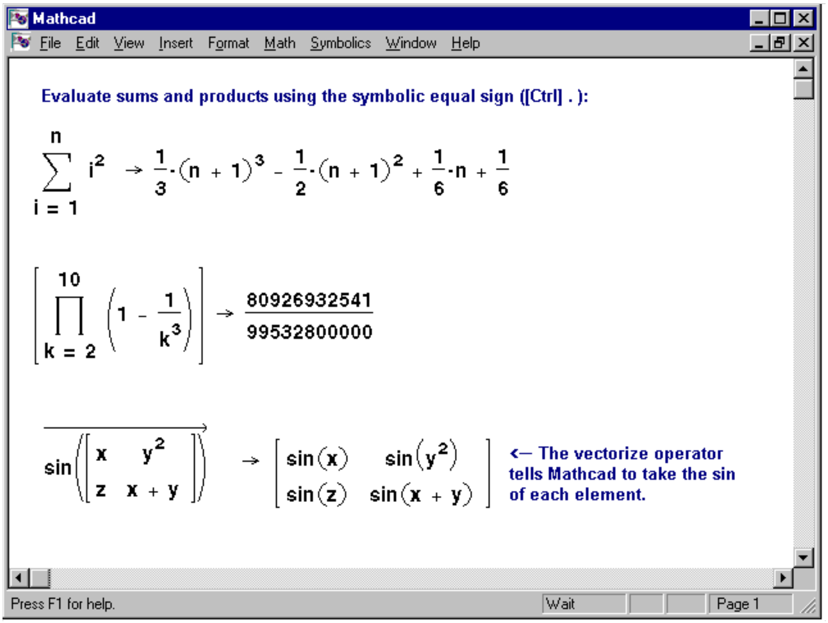


Figure 17-13: Symbolic evaluation of sums, products, and functions.

Symbolic calculus

This section describes how to symbolically evaluate definite and indefinite integrals, derivatives, and limits.

Derivatives

To evaluate a derivative symbolically, you can use Mathcad's derivative operator and the live symbolic equal sign as shown in Figure 17-14:

- Type **?** to create the derivative operator or type [**Ctrl**]**?** to create the higher order derivative operator.
- In the placeholders, enter the expression you want to differentiate and the variable with respect to which you are differentiating.
- Press [**Ctrl**]**.** (the control key followed by a period). Mathcad displays an arrow, "→".
- Press [**Enter**] to see the result.

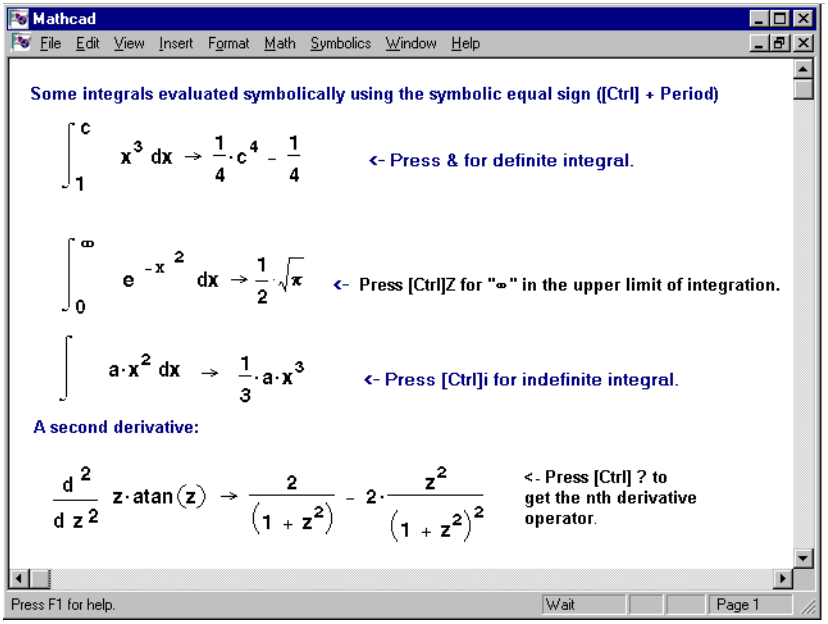


Figure 17-14: Evaluating integrals and derivatives symbolically.

Figure 17-15 shows you how to differentiate an expression without using the derivative operator. The **Symbolics** menu command **Variable⇒Differentiate** differentiates an expression with respect to a selected variable. For example, to differentiate $2 \cdot x^2 + y$ with respect to x:

- Enter the expression.
- Click on the x.
- Choose **Variable⇒Differentiate** from the **Symbolics** menu. Mathcad will display the derivative, $4 \cdot x$.

If you selected the variable y instead of x, you would get the answer 1. Mathcad treats all variables except the one you've selected as constants.

If you've selected neither x nor y the menu command will be gray. Mathcad can't differentiate the expression because you haven't specified a differentiation variable.

If the expression in which you've selected a variable is one element of an array, Mathcad will differentiate only that array element. To differentiate an entire array, differentiate each element individually by selecting a variable in that element and choosing **Variable⇒Differentiate** from the **Symbolics** menu.

Indefinite integrals

Mathcad provides the symbolic indefinite integral operator shown in Figure 17-14. To use this operator:

- Type [`Ctrl`]`I` to insert the indefinite integral operator and its placeholders.

- Fill in the placeholder for the integrand.
- Place the integration variable in the placeholder next to the "*d*." This can be any variable name.
- Press [**`Ctrl`**]**`.`** (the control key followed by a period). Mathcad displays an arrow, "→".
- Press [**`Enter`**] to see the result.

Figure 17-15 shows how to integrate an expression without using the indefinite integral operator. The **Symbolics** menu command **Variable⇒Integrate** integrates an expression with respect to a selected variable. For example, to integrate $2 \cdot x^2 + y$ with respect to *x*:

- Select the *x*.
- Choose **Variable⇒Integrate** from the **Symbolics** menu. Mathcad will display the integral.

The **Variable⇒Integrate** command integrates an expression with respect to a selected variable. If you don't have a variable selected, this command will be gray. Mathcad cannot integrate without knowing the variable of integration.

If the symbolic processor can't find a closed-form indefinite integral, you'll see an appropriate error message. Keep in mind that many simple expressions don't have a closed-form indefinite integral that can be written in terms of polynomials or elementary functions. For example, e^{-x^3} has no elementary integral. If the integral is too big to display, Mathcad puts the answer, in text form, on the clipboard. See the section "Long answers" on page 401 to learn what to do when this happens.

When evaluating an indefinite integral, remember that the answer to an integration problem is not unique. If $f(x)$ is an integral of a given function, so is $f(x) + C$ for any constant *C*. Thus, the answer you get from Mathcad may differ by a constant from the answer you find in tables. If you differentiate a function and then integrate the result, you won't necessarily get the original function back as your answer.

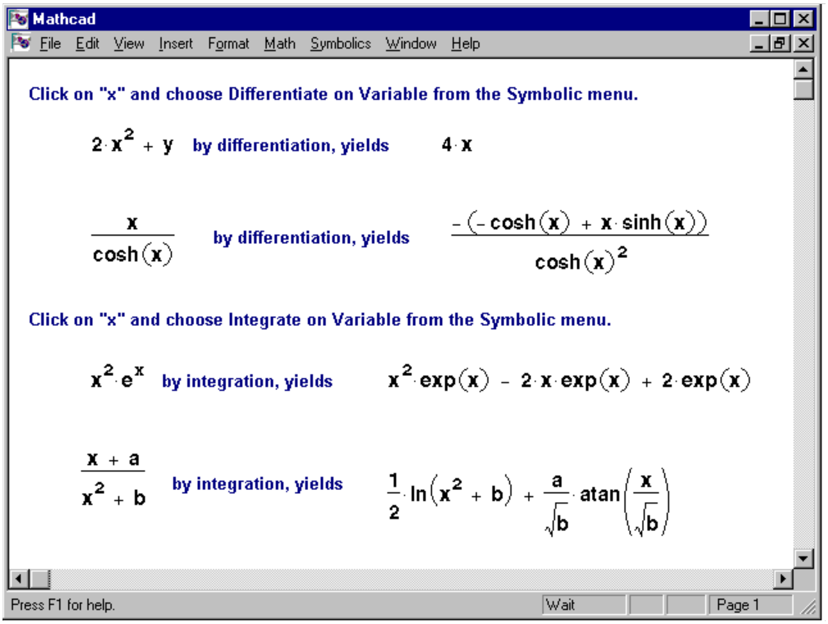


Figure 17-15: Differentiating and integrating expressions.

Definite integrals

To symbolically evaluate a definite integral:

- Type **&** to create the integral operator with its empty placeholders.
- Fill in the placeholders for the limits of integration. These can be variables, constants, or expressions.
- Fill in the placeholder for the integrand.
- Fill in the placeholder next to the "d." This is the variable of integration.
- Press [**Ctrl**]**.** (the control key followed by a period). Mathcad displays an arrow, "→".
- Press [**Enter**] to see the result.

The symbolic processor will attempt to find an indefinite integral of your integrand before substituting the limits you specified. See Figure 17-14 for an example. If the symbolic processor can't find a closed form for the integral, you'll see an appropriate error message.

If the symbolic integration succeeds and the limits of integration are integers, fractions, or exact constants like π, you'll get an exact value for your integral. If the integrand or one of the limits contains a decimal point, the symbolic answer will be a number displayed with 20 significant digits. Use the **float** keyword described in "Floating point evaluation" on page 369 to generate a result with a different number of significant digits.

This answer will in general agree with the answer you get by evaluating the same integral numerically. The symbolic and numerical answers are, however, obtained in very different ways. Mathcad's symbolic processor:

- Finds an indefinite integral.
- Subtracts its value at the lower limit of integration from its value at the upper limit.

The numerical integration routine, on the other hand:

- Samples the integrand at many points in the interval of integration.
- Uses these samples to approximate the integral.

The accuracy of this numerical integration depends on the value you set for the variable TOL and on the smoothness of the function you are integrating.

Of course, many functions do not have a closed form integral, and definite integrals involving these functions can *only* be calculated numerically. Integrals for which the integrand is not smooth (has a discontinuous derivative) might not be evaluated correctly by the symbolic processor. See the section “Integrals” in Chapter 12 for more on Mathcad's numerical integration.

Limits

Mathcad provides three limit operators. These can only be evaluated symbolically. They cannot be evaluated numerically. To use the limit operators:

- Press [**Ctrl**]**L** to create the limit operator. To create operators for limits from the left or right, press [**Ctrl**]**B** or [**Ctrl**]**A**.
- Enter the expression in the placeholder to the right of the “lim.”
- Enter the limiting variable in the left-hand placeholder below the “lim.”
- Enter the limiting value in the right-hand placeholder below the “lim.”
- Press [**Ctrl**]**.** (the control key followed by a period). Mathcad displays an arrow, “→”.
- Press [**Enter**] to see the result.

Mathcad will return a result for the limit. If the limit does not exist, Mathcad returns an error message. Figure 17-16 shows some examples of evaluating limits.

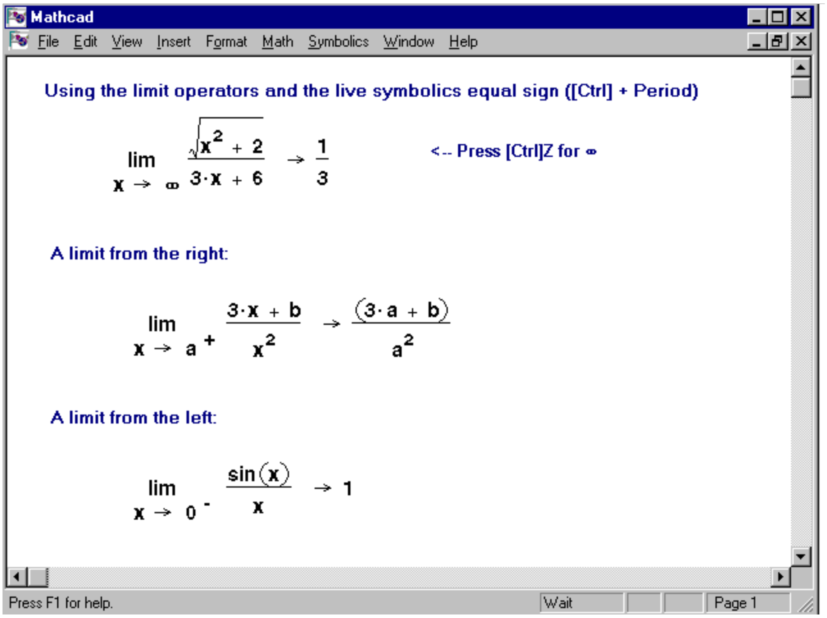


Figure 17-16: Evaluating limits.

Solving equations symbolically

This section discusses how to use either keywords or menu commands from the **Symbolics** menu to symbolically solve an equation for a variable, find the symbolic roots of an expression, and solve a system of equations symbolically. Most of the examples in this section demonstrate "live" solving using symbolic keywords, but you may apply commands from the **Symbolics** menu to expressions on a case by case basis if you prefer. Keep in mind that, unlike the keyword-modified expressions, expressions modified by commands from the **Symbolics** menu do not update automatically, as described in the section "Using the Symbolics menu" on page 366.

Solving equations symbolically is far more difficult than solving them numerically. You may find that the symbolic solver does not give a solution. This may happen for a variety of reasons discussed in "Limits to symbolic processing" on page 401.

Solving an equation for a variable

To solve an equation symbolically for a variable, use the keyword **solve**:

- Type the equation. Make sure you use [**`Ctrl`**]= to create the equal sign.
- Press [**`Ctrl`**][**`Shift`**]**`.`** (hold down the control and shift keys and type a period). Mathcad displays a placeholder to the left of the arrow, "→".

■ In the placeholder, type **`solve`** followed by a comma and the variable for which to solve.

■ Press [**`Enter`**] to see the result.

Mathcad will solve for the variable and insert the result to the right of the “→”. Note that if the variable was squared in the original equation, you may get *two* answers back when you solve. Mathcad displays these in a vector. Figure 17-17 shows an example.

Another way to solve for a variable is to enter the equation, click on the variable you want to solve for in an equation, and choose **Variable⇒Solve** from the **Symbolics** menu.

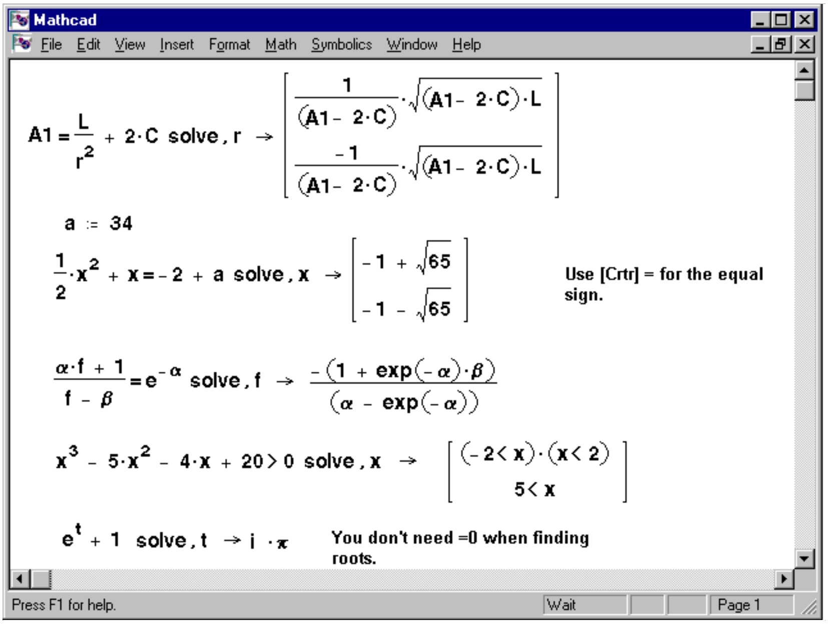


Figure 17-17: Examples of solving equations, solving inequalities, and finding roots.

You can also solve an inequality entered using the symbols <, >, ≤, and ≥. Solutions to inequalities will be displayed in terms of Mathcad boolean expressions. If there is more than one solution, Mathcad places them in a vector. A Mathcad boolean expression such as $x < 2$ has the value 1 if it is true and 0 if it is false. Thus the solution “x is less than 2 and greater than –2” would be represented by the expression $(x < 2) \cdot (-2 < x)$.

Finding the roots of an expression

You can use the **solve** keyword to find the roots of an expression in a manner similar to that of solving an equation in a variable:

■ Type the expression.

- Press [**Ctrl**] [**Shift**]**.** (hold down the control and shift keys and type a period). Mathcad displays a placeholder to the left of the arrow, “→”.
- In the placeholder, type **solve** followed by a comma and the variable for which to solve.
- Press [**Enter**] to see the result.

Note that there is no need to set the expression equal to zero. When Mathcad doesn't find an equals sign, it assumes you mean to set the expression equal to zero. See Figure 17-17 for an example.

Solving a system of equations symbolically: The “solve” keyword

One way to symbolically solve a system of equations is to use the same **solve** keyword used to solve one equation in one unknown. To solve a system of *n* equations for *n* unknowns:

- Press [**Ctrl**] **M** to create a vector having *n* rows and 1 column.
- Fill in each placeholder of the vector with one of the *n* equations making up the system. Make sure you use [**Ctrl**]**=** to create the equals sign.
- Press [**Ctrl**] [**Shift**]**.** (hold down the control and shift keys and type a period). Mathcad displays a placeholder to the left of the arrow, “→”.
- In the placeholder, type **solve** followed by a comma.
- Press [**Ctrl**] **M** to create a vector having *n* rows and 1 column.
- Press [**Enter**] to see the result.

Mathcad displays the *n* solutions to the system of equations to the right of the arrow. Figure 17-18 shows an example.

Solving a system of equations symbolically: Solve block

Another way to solve a system of equations symbolically is to use a solve block, similar to the numerical solve blocks described in Chapter 15, “Solving Equations.”

- Type the word *Given*. This tells Mathcad that what follows is a system of equations. You can type *Given* in any combination of upper and lower case letters, and in any font. Just be sure you don't type it while in a text region or paragraph.
- Now type the equations in any order below the word *Given*. Make sure you press [**Ctrl**]**=** to type “=.”
- Type the *Find* function as appropriate for your system of equations. This function is described “Systems of equations” on page 321. The arguments of the function are the variables for which you are solving.
- Press [**Ctrl**]**.** (the control key followed by a period). Mathcad displays the symbolic equal sign.
- Click outside the *Find* function.

Mathcad displays the solutions to the system of equations to the right of the arrow. If the *Find* function has one argument, Mathcad returns one result. If the *Find* has more than one argument, Mathcad returns a vector of results. For example, Find(x, y) returns a vector containing the expressions for x and y that solve the system of equations. Note that if your system is an overdetermined linear system, the *Find* function will not return a solution. Use the *Minerr* function instead of *Find*. *Minerr* will return an answer that minimizes the errors in the constraints.

Most of the guidelines for solve blocks described in Chapter 15, “Solving Equations,” apply to the symbolic solution of systems of equations. The main difference is that when you solve equations symbolically, you should not enter guess values for the solutions.

Figure 17-18 shows an example of a solve block used to solve a system of equations symbolically. For more information on solve blocks, see Chapter 15, “Solving Equations.”

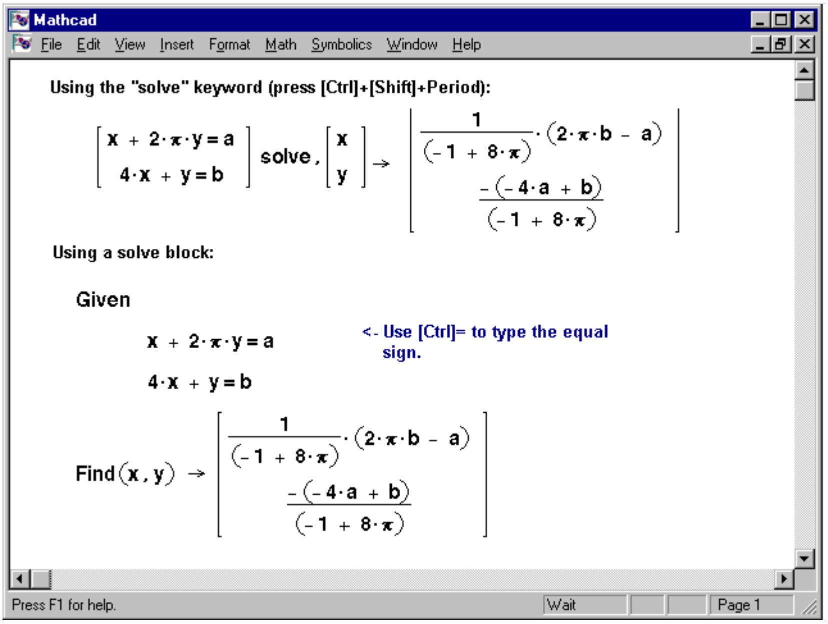


Figure 17-18: Solving a system of equations symbolically.

Symbolic matrix manipulation

This section describes how to find the symbolic transpose, inverse, and determinant of a matrix. The examples in this section demonstrate “live” symbolic matrix manipulation using the matrix operators, described in Chapter 10, “Vectors and Matrices,” and the symbolic equal sign. You may, however, apply the **Matrix** commands from the

Symbolics menu to matrices on a case by case basis if you prefer. Keep in mind that, unlike matrices evaluated with the symbolic equal sign, matrices modified by commands from the **Symbolics** menu do not update automatically, as described in the section “Using the Symbolics menu” on page 366.

Finding the symbolic transpose

To find the symbolic transpose of a matrix:

- Place the entire matrix between the two editing lines by clicking [**`Space`**] one or more times.
- Press [**`Ctrl`**] **`1`** to create the matrix transpose operator.
- Press [**`Ctrl`**]**`.`** (the control key followed by a period). Mathcad displays an arrow, “→”.
- Press [**`Enter`**] to see the result.

Mathcad returns the matrix with its rows and columns swapped to the right of the “→”.

Another way to find the transpose of a matrix is to select the matrix and choose **Matrix⇒Transpose** from the **Symbolics** menu.

Finding the symbolic inverse

To find the symbolic inverse of a square matrix:

- Place the entire matrix between the two editing lines by clicking [**`Space`**] one or more times.
- Press **`^-1`** to indicate matrix inversion.
- Press [**`Ctrl`**]**`.`** (the control key followed by a period). Mathcad displays an arrow, “→”.
- Press [**`Enter`**] to see the result.

Mathcad will return a symbolic representation for the inverse of the selected matrix to the right of the “→”.

Another way to find the inverse of a matrix is to select the matrix and choose **Matrix⇒Invert** from the **Symbolics** menu.

Finding the symbolic determinant

To find the symbolic determinant of a square matrix:

- Place the entire matrix between the two editing lines by clicking [**`Space`**] one or more times.
- Press **`|`** to create the matrix determinant operator.
- Press [**`Ctrl`**]**`.`** (the control key followed by a period). Mathcad displays an arrow, “→”.
- Press [**`Enter`**] to see the result.

Mathcad will return a symbolic representation for the determinant of the selected matrix to the right of the "→". Keep in mind that this is usually a lengthy expression.

Another way to find the determinant of a matrix is to select the matrix and choose **Matrix⇒Determinant** from the **Symbolics** menu.

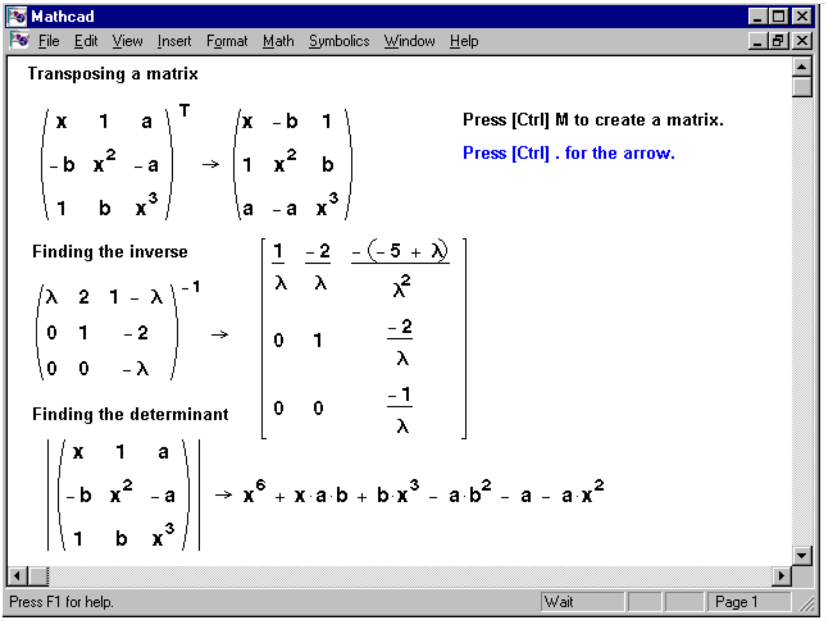


Figure 17-19: Symbolic matrix operations.

Symbolic transforms

This section describes how to perform the Fourier, Laplace, and z-transforms, and their inverses. The examples in this section demonstrate "live" transformations using symbolic keywords, but you may apply the **Transforms** commands from the **Symbolics** menu to expressions on a case by case basis if you prefer. Keep in mind that, unlike keyword-modified expressions, expressions modified by commands from the **Symbolics** menu do not update automatically, as described in the section "Using the Symbolics menu" on page 366.

Figure 17-20 shows some examples of symbolic transforms in Mathcad. Note that the result may contain functions that are recognized by Mathcad's symbolic processor but not by its numeric processor. An example is the function *Dirac* in the middle of Figure 17-20. You'll find numerical definitions for this and other such functions at the end of this chapter and in the on-line Help.

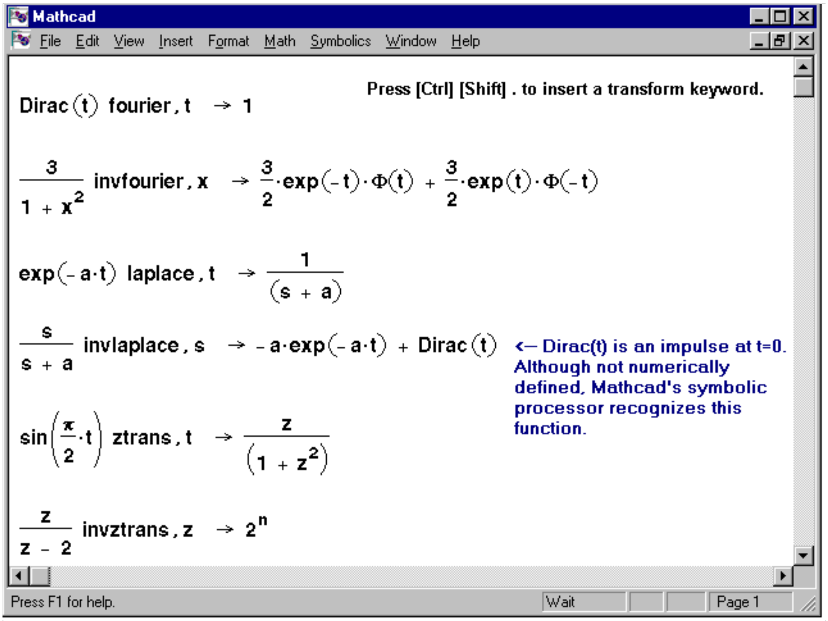


Figure 17-20: Performing symbolic transforms.

When you perform a symbolic transform, Mathcad returns a function in a variable commonly used in the context of the transform. You can substitute a different variable for the one Mathcad returns by using the **substitute** keyword. See the section “Symbolic algebra” on page 368 for information on substituting one variable for another.

Fourier and inverse Fourier transformations

To evaluate the Fourier transform of a function, use the keyword **fourier**:

- Enter the expression to be transformed.
- Press [**Ctrl**] [**Shift**]**.** (hold down the control and shift keys and type a period). Mathcad displays a placeholder to the left of the arrow, “→”.
- In the placeholder, type **fourier** followed by a comma and the transform variable.
- Press [**Enter**] to see the result.

Mathcad returns a function of ω given by:

$$\int_{-\infty}^{+\infty} f(t)e^{-i\omega t}dt$$

where $f(t)$ is the expression to be transformed.

Mathcad returns a function in the variable ω when you perform a Fourier transform since this is a commonly used variable name in this context. If the expression you are transforming already contains an ω, Mathcad avoids ambiguity by returning a function of the variable ωω instead.

Another way to evaluate the Fourier transform of an expression is to enter the expression and click on the transform variable. Then choose **Transform⇒Fourier** from the **Symbolics** menu.

To evaluate the inverse Fourier transform of a function, use the keyword **invfourier**:

- Enter the expression to be transformed.
- Press [**`Ctrl`**] [**`Shift`**]**`.`** (hold down the control and shift keys and type a period). Mathcad displays a placeholder to the left of the arrow, "→".
- In the placeholder, type **`invfourier`** followed by a comma and the transform variable.
- Press [**`Enter`**] to see the result.

Mathcad returns a function of *t* given by:

$$\frac{1}{2\pi}\int_{-\infty}^{+\infty} F(\omega)e^{i\omega t}d\omega$$

where $F(\omega)$ is the expression to be transformed.

Mathcad returns a function in the variable *t* when you perform an inverse Fourier transform since this is a commonly used variable name in this context. If the expression you are transforming already contains a *t*, Mathcad avoids ambiguity by returning a function of the variable *tt* instead.

Another way to evaluate the inverse Fourier transform of an expression is to enter the expression and click on the transform variable. Then choose **Transform⇒Inverse Fourier** from the **Symbolics** menu.

Laplace and inverse Laplace transformations

To evaluate the Laplace transform of a function, use the keyword **laplace**:

- Enter the expression to be transformed.
- Press [**`Ctrl`**] [**`Shift`**]**`.`** (hold down the control and shift keys and type a period). Mathcad displays a placeholder to the left of the arrow, "→".
- In the placeholder, type **`laplace`** followed by a comma and the transform variable.
- Press [**`Enter`**] to see the result.

Mathcad returns a function of *s* given by:

$$\int_{0}^{+\infty} f(t)e^{-st}dt$$

where $f(t)$ is the expression to be transformed.

Mathcad returns a function in the variable *s* when you perform a Laplace transform since this is a commonly used variable name in this context. If the expression you are transforming already contains an *s*, Mathcad avoids ambiguity by returning a function of the variable *ss* instead.

Another way to evaluate the Laplace transform of an expression is to enter the expression and click on the transform variable. Then choose **Transform⇒Laplace** from the **Symbolics** menu.

To evaluate the inverse Laplace transform of a function, use the keyword **invlaplace**:

- Enter the expression to be transformed.
- Press [**`Ctrl`**] [**`Shift`**] **`.`** (hold down the control and shift keys and type a period). Mathcad displays a placeholder to the left of the arrow, "→".
- In the placeholder, type **`invlaplace`** followed by a comma and the transform variable.
- Press [**`Enter`**] to see the result.

Mathcad returns a function of *t* given by:

$$\frac{1}{2\pi}\int_{\sigma - i\infty}^{\sigma + i\infty} F(s)e^{st}dt$$

where *F*(*s*) is the expression to be transformed and all singularities of *F*(*s*) are to the left of the line $\mathrm{Re}(s) = \sigma$.

Mathcad returns a function in the variable *t* when you perform an inverse Laplace transform since this is a commonly used variable name in this context. If the expression you are transforming already contains a *t*, Mathcad avoids ambiguity by returning a function of the variable *tt* instead.

Another way to evaluate the inverse Laplace transform of an expression is to enter the expression and click on the transform variable. Then choose **Transform⇒Inverse Laplace** from the **Symbolics** menu.

z and inverse *z*-transformations

To evaluate the *z*-transform of a function, use the keyword **ztrans**:

- Enter the expression to be transformed.
- Press [**`Ctrl`**] [**`Shift`**] **`.`** (hold down the control and shift keys and type a period). Mathcad displays a placeholder to the left of the arrow, "→".
- In the placeholder, type **`ztrans`** followed by a comma and the transform variable.
- Press [**`Enter`**] to see the result.

Mathcad returns a function of *z* given by:

$$\sum_{n=0}^{+\infty} f(n)z^{-n}$$

where *f*(*n*) is the expression to be transformed.

Mathcad returns a function in the variable *z* when you perform a *z*-transform since this is a commonly used variable name in this context. If the expression you are transforming already contains a *z*, Mathcad avoids ambiguity by returning a function of the variable *zz* instead.

Another way to evaluate the *z*-transform of an expression is to enter the expression and click on the transform variable. Then choose **Transform⇒z-Transform** from the **Symbolics** menu.

To evaluate the inverse *z*-transform of a function, use the keyword **invztrans**:

- Enter the expression to be transformed.
- Press [**`Ctrl`**] [**`Shift`**]**`.`** (hold down the control and shift keys and type a period). Mathcad displays a placeholder to the left of the arrow, "→".
- In the placeholder, type **`invztrans`** followed by a comma and the transform variable.
- Press [**`Enter`**] to see the result.

Mathcad returns a function of *n* given by a contour integral around the origin:

$$\frac{1}{2\pi i}\int_C F(z)z^{n-1}dz$$

where *F*(*z*) is the expression to be transformed and *C* is a contour enclosing all singularities of the integrand.

Mathcad returns a function in the variable *n* when you perform an inverse *z*-transform since this is a commonly used variable name in this context. If the expression you are transforming already contains an *n*, Mathcad avoids ambiguity by returning a function of the variable *nn* instead.

Another way to evaluate the inverse *z*-transform of an expression is to enter the expression and click on the transform variable. Then choose **Transform⇒Inverse z-Transform** from the **Symbolics** menu.

Symbolic optimization

In general, Mathcad's symbolic processor and Mathcad's numerical processor don't communicate with one another. Because of this, it's possible to set up a complicated numerical calculation without knowing that you could have reduced it to an equivalent but much simpler problem by judicious use of the symbolic processor.

You can, however, make the numerical processor ask the symbolic processor for advice before starting what could be a needlessly complex calculation. In effect, the symbolic processor acts like the numerical processor's consultant, examining each expression and recommending a better way to evaluate it whenever possible. It does this for each expression in the worksheet except for those you specifically tell it to ignore.

For example, if you were to evaluate an expression such as:

$$\int_0^u \int_0^v \int_0^w x^2 + y^2 + z^2 dx\, dy\, dz$$

Mathcad would undertake the laborious task of evaluating a numerical approximation of the triple integral even though one could arrive at an exact solution by first performing a few elementary calculus operations.

This happens because by itself, Mathcad's numerical processor does not know enough to simplify before plunging ahead into the calculation. Although Mathcad's symbolic processor knows all about simplifying complicated expressions, these two processors do not consult with each other. To make these two processors talk to each other, choose **Optimization** from the **Math** menu.

Once you've done this, Mathcad's live symbolic processor steps in and simplifies all expressions to the right of a ":=" *before* the numerical processor gets a chance to begin its calculations. It will continue to do so until you choose **Optimization** from the **Math** menu once more to remove the checkmark.

If Mathcad finds a simpler form for the expression, it responds by doing the following:

- It marks the region with a red asterisk.
- It *internally* replaces what you've typed with a simplified form. The expression you typed is left unchanged; Mathcad simply works with an equivalent expression that happens to be better suited for numerical analysis.
- Mathcad evaluates this equivalent expression instead of the expression you specified. To see this equivalent expression, double-click on the red asterisk beside the region.

If Mathcad is unable to find a simpler form for the expression, it places a *blue* asterisk next to it.

In the previous example, the symbolic processor would examine the triple integral and return the equivalent, but much simpler expression:

$$\frac{1}{3}(w^3 vu + wv^3 u + wvu^3)$$

To see this expression in a pop-up window click on the red asterisk with the right mouse button and choose **Show Popup** from the context menu (see Figure 17-21). To dismiss the pop-up, click the close box in the upper right corner.

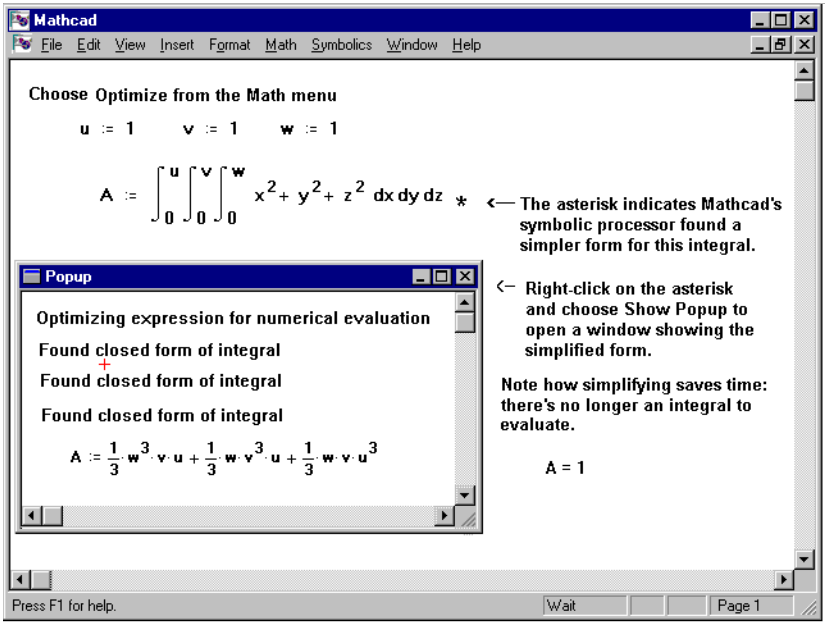


Figure 17-21: A pop-up window showing the equivalent expression that Mathcad actually evaluates.

Since this simplified form contains no integral, Mathcad's numerical processor no longer needs to use a lengthy numerical algorithm to evaluate the integral. This offers two advantages:

- By avoiding time-consuming integration, Mathcad's numerical processor can evaluate the expression far more quickly.
- Mathcad avoids all the computation issues inherent in numerical integration.

There may be times when you don't want Mathcad's symbolic processor to examine a particular equation. You may want to evaluate an expression exactly as you've typed it. To do so, right-click on the expression and choose **Optimize** from the context menu. This procedure disables optimization for that expression.

Using functions and variables

Mathcad's symbolic processor does not treat functions and variables in exactly the same way as its numerical processor. These differences revolve around the answers to the following question:

- Does the symbolic processor "know" that a function or variable is defined elsewhere?

The answer to this depends on two things:

- Is the function or variable built-in or is it defined somewhere on the worksheet?
- Are you using the symbolic equal sign or a menu command?

The next two sections describe what Mathcad does with variables and functions in a symbolic transformation.

A related question is the converse. Symbolic transformations can sometimes return functions and constants which do not exist in Mathcad's list of built-in functions and constants. These are described in the section “Special functions” on page 398.

Built-in functions and variables

As a general rule, built-in functions retain their meanings when used in symbolic transformations provided that it makes sense for them to do so. For example, functions like *sin* and *log* keep their meanings because these have a commonly accepted mathematical meaning. Other functions like *linterp* or *rnd* lack any commonly accepted meaning so Mathcad doesn't attempt to assign one.

Built-in functions that do retain their meanings when used in symbolic calculations include: trigonometric and hyperbolic functions and their inverses; logarithmic and exponential functions; the *Re* and *Im* functions; the *erf* function; the Γ function; the *mod* function; Φ (the Heaviside step function); *max* and *min*; and the *identity* and *eigenvals* functions for matrices.

In general, these functions mean the same thing for both numerical evaluations and symbolic transformations. There are two subtle differences:

- Unlike the numerical *mod* function, the symbolic *mod* function requires an integer modulus, and can accept a polynomial as its first argument.
- Certain of the inverse trigonometric functions use different branches in the complex plane.

As a general rule, built-in constants also retain their meanings when used in symbolic transformations provided that it makes sense for them to do so. The symbolic processor will recognize π, *e* and ∞. Moreover, these will have their exact meanings when used symbolically. When symbolic transformations are involved, there is no need to limit ∞ to 10^{307} or to limit π to only fifteen digits of precision.

Built-in constants lacking an intuitive mathematical meaning are not recognized by the symbolic processor. For example, TOL and ORIGIN will not have their usual meanings in symbolic transformations. They will be treated like any other undefined variable.

Figure 17-22 shows the difference in the way Mathcad treats functions in symbolic transformations. Note that the symbolic processor will recognize and evaluate the *sin* function, but when asked to evaluate $rnd(3)$ the symbolic processor simply returns $rnd(3)$.

User-defined functions and variables

Functions and variables you define yourself *are* recognized by the symbolic processor when you use the symbolic equal sign discussed in the next section. They *are not*, however, recognized when you use **Symbolics** menu commands. Figure 17-22 shows the difference.

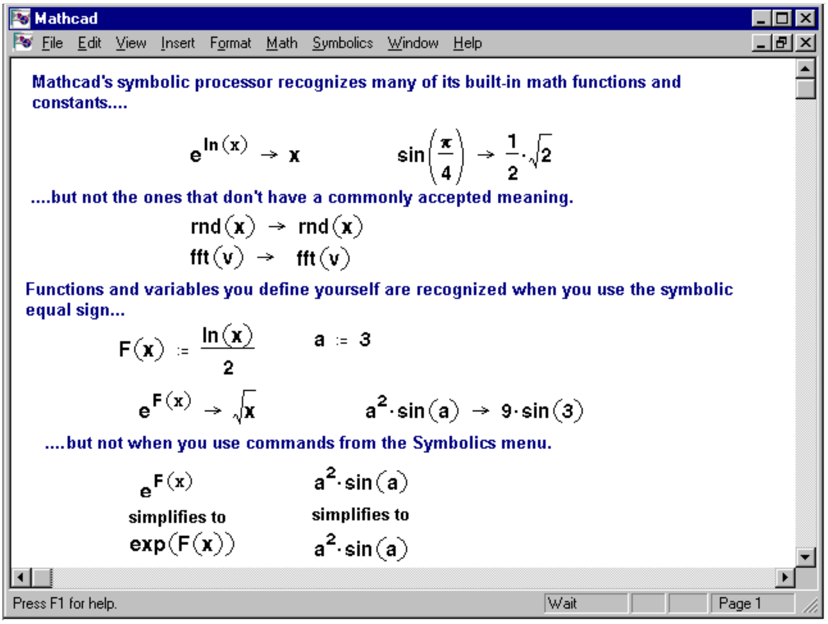


Figure 17-22: The symbolic processor recognizes certain built-in functions. Functions and variables you define yourself are only recognized when you use the symbolic equal sign.

Special functions

A symbolic transformation sometimes comes back in terms of a function which isn't part of Mathcad's list of built-in functions. The list below gives definitions for those special functions. Except for *Ei*, *erf*, and *Zeta*, all of which involve infinite sums, and *W*, you can use these definitions to calculate numerical values.

You can define many of these functions in Mathcad. See the “Special Functions” topic in the Quicksheets of the Resource Center for examples.

γ is Euler's constant, approximately 0.5772156649.

$$\mathrm{Chi}(x) = \gamma + \ln(x) + \int_0^x \frac{\cosh(t) - 1}{t}\, dt$$

$$\mathrm{Ci}(x) = \gamma + \ln(x) + \int_0^x \frac{\cos(t) - 1}{t}\,dt$$

$\mathrm{csgn}(z) = 1$ if $\mathrm{Re}(z) > 0$ or $(\mathrm{Re}(z) = 0$ and $\mathrm{Im}(z) \geq 0)$; -1 otherwise.
Define in Mathcad as: $if(\mathrm{Re}(z) \neq 0, 2\Phi(\mathrm{Re}(z)) - 1, 2\Phi(\mathrm{Im}(z)) - 1)$

$$\mathrm{dilog}(\mathrm{x}) = \int_1^x \frac{\ln(t)}{1-t}\,dt$$

$\mathrm{Dirac}(x) = 0$ if x is not zero. $\int_{-\infty}^{\infty} Dirac(x)dx = 1$

$$\mathrm{Ei}(x) = \gamma + \ln(x) + \sum_{n=1}^{\infty} \frac{x^n}{n \cdot n!} \quad (x > 0)$$

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \sum_{n=0}^{\infty} \frac{(-1)^n z^{2n+1}}{n!(2n+1)} \quad \text{(for complex } z\text{)}$$

$$\mathrm{FresnelC}(x) = \int_0^x \cos\left(\frac{\pi}{2}t^2\right)dt$$

$$\mathrm{FresnelS}(x) = \int_0^x \sin\left(\frac{\pi}{2}t^2\right)dt$$

$$\mathrm{LegendreE}(x, k) = \int_0^x \left(\frac{1 - k^2 \cdot t^2}{1 - t^2}\right)^{1/2} dt$$

$$\mathrm{LegendreEc}(k) = \mathrm{LegendreE}(1, k)$$

$$\mathrm{LegendreEc1}(k) = \mathrm{LegendreEc}(\sqrt{1 - k^2})$$

$$\mathrm{LegendreF}(x, k) = \int_0^x \frac{1}{\sqrt{(1 - t^2)(1 - k^2 \cdot t^2)}}\,dt$$

$$\mathrm{LegendreKc}(k) = \mathrm{LegendreF}(1, k)$$

$$\mathrm{LegendreKc1}(k) = \mathrm{LegendreKc}(\sqrt{1 - k^2})$$

$$\mathrm{LegendrePi}(x, n, k) = \int_0^x \frac{1}{\sqrt{(1 - n^2 \cdot t^2)}\sqrt{(1 - t^2)(1 - k^2 \cdot t^2)}}\,dt$$

$$\text{LegendrePic}(n, k) = \text{LegendrePi}(1, n, k)$$

$$\text{LegendrePic1}(k) = \text{LegendrePic}(n, \sqrt{1-k^2})$$

$$\text{Psi}(n, k) = \frac{d^n}{dx^n}\text{Psi}(x)$$

$$\text{Psi}(x) = \frac{d}{dx}\ln(\Gamma(x))$$

$$\text{Shi}(x) = \int_0^x \frac{\sinh(t)}{t}\,dt$$

$$\text{Si}(x) = \int_0^x \frac{\sin(t)}{t}\,dt$$

$\text{signum}(x) = 1$ if $x = 0$, $x/|x|$ otherwise; calculate in Mathcad as $(x{=}0) + x/|x|$.

$W(x)$ is the principal branch of a function satisfying $W(x) \cdot \exp(W(x)) = x$. $W(n, x)$ is the nth branch of $W(x)$.

$$Zeta(s)\left(\sum_{n=1}^{\infty}\frac{1}{n^s}\right)\ (s > 1)$$

The functions *arcsec*, *arccsc*, *arccot*, *arcsech*, *arcscsh*, *arccoth* can be calculated by taking reciprocals and using the Mathcad built-in functions *acos*, *asin*, etc. For example:

$$\text{arcsec}(x) := \text{acos}\left(\frac{1}{x}\right)$$

The *Psi* function and Γ appear frequently in the results of *indefinite* sums and products. If you use a single variable name rather than a full range in the index placeholder of a summation or product, and you choose **Evaluate Symbolically** or one of the other symbolic evaluation commands, Mathcad will attempt to calculate an indefinite sum or product of the expression in the main placeholder. The indefinite sum of $f(i)$ is an expression $S(i)$ for which

$$S(i+1) - S(i) = f(i)$$

The indefinite product of $f(i)$ is an expression $P(i)$ for which

$$\frac{P(i+1)}{P(i)} = f(i)$$

Limits to symbolic processing

As you work with the symbolic processor, you will undoubtedly discover two things:

- many problems can *only* be solved numerically, and
- many more problems yield such lengthy expressions that you'll wish you *had* solved them numerically.

For a computer, symbolic operations are, in general, much more difficult than the corresponding numerical operations. In fact, if you write down a complicated function at random, the chance is very small that either its roots or its integral can be expressed in a simple closed form. For example, there is no formula for the roots of a general polynomial of degree 5 or higher, even though exact roots can be found for some special cases.

Many deceptively simple-looking functions made up of elementary pieces like powers and roots, exponentials, logs, and trigonometric functions have no closed-form integral that can be expressed in terms of these same functions.

When an equation has several solutions, Mathcad sometimes returns only a partial solution and asks if you want this result placed in the clipboard. If you click “OK,” Mathcad shows a vector containing the solutions found and the word “Root”. In the clipboard, in place of the word “Root” you will see an expression of the form “RootOf (function_of_Z)”. The roots of the indicated function are solutions of the original equation.

As with other symbolic operations, the answers you get depend on whether the constants in your equation contain decimal points. If your constants are pure rational numbers like 1/2 or 4, the symbolic solver will try to find an *exact* solution. For example, the solution to the second equation in Figure 17-17 is exact. But if you had defined *a* to be “34.0” instead of “34”, Mathcad would have given approximate numerical values.

Long answers

Symbolic calculations can easily produce answers so long that they don't fit conveniently in your window. If the answer consists of the sum of several terms, you can reformat such an answer by using Mathcad's “Addition with line break” operator described in the section “List of operators” in Chapter 12.

To break an expression with plus signs:

- Click just to the right of the term that appears immediately before the plus sign at which you want to break the expression.
- Press [**`Space`**] until the all the terms from the first to the selected on are held between the two editing lines.
- Press [**`Del`**]. The plus sign just after the editing lines will disappear.
- Now type [**`Ctrl`**][**`Enter`**] to insert the plus with break.

You can repeat this process if there are several terms connected by plus signs.

Sometimes, a symbolic answer will be so long that you can't conveniently display it in your worksheet. When this happens, Mathcad will ask if you want the answer placed in the clipboard. If you click “OK,” Mathcad places a string representing the answer on the clipboard.

When you examine the contents of the clipboard, you'll see an answer written in a Fortran-like syntax as shown in Figure 17-23. This syntax uses the following conventions:

- The symbols +, –, *, and / stand for the basic arithmetic operations. Exponentiation is denoted by “**”.
- The derivative of $f(x)$ with respect to x is written “diff($f(x)$, x)”. The nth derivative is “diff($f(x)$, $x\$n$)”.
- A “D” also stands for the partial derivative operator. The nth derivative is “(D, n)”. The partial derivative of a function with respect to its nth argument is “$(D[n])$”.
- The integral of $f(x)$ with respect to x is written “int($f(x)$, x)”.
- The summation and product operators appear as “sum( )” and “product( )”.
- The operator “@” denotes function composition. For example, (sin@exp)(x) is the same as sin(exp(x)). A “@@” represents repeated composition, so (f@@2)(x) is the same as $f(f(x))$.
- “RootOf(*equation*)” stands for any root of an algebraic equation. (For example, “RootOf(Z**2 + 1)” represents either i or $-i$.)
- You may see embedded font codes (like “MFNT_03_”) preceding the variable name to indicate the font in which the variable name is to appear.

To insert the answer as text into your Mathcad worksheet:

- Click in an empty area.
- Choose **Paste** from the **Edit** menu.

To save a long clipboard answer as a separate text file, choose **Save As** from the Clipboard's **File** menu.

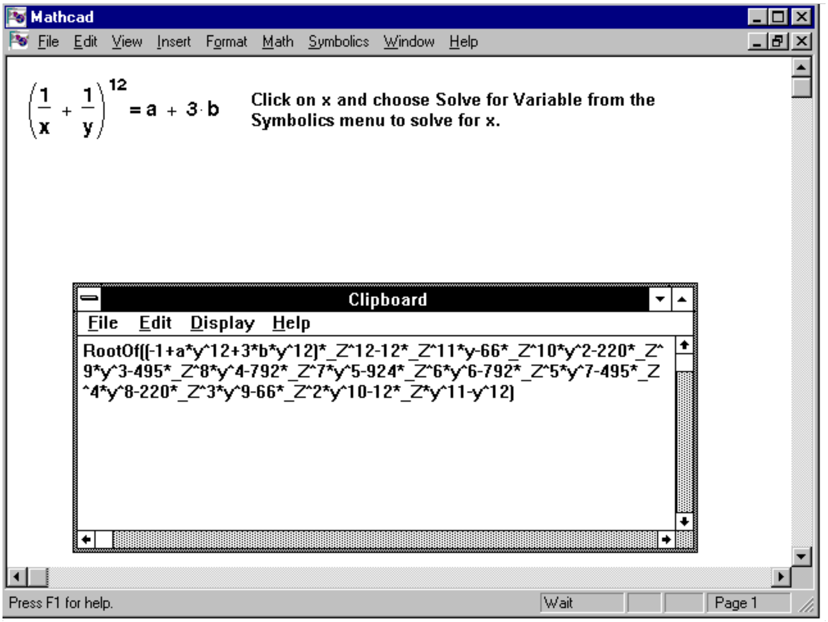


Figure 17-23: A long symbolic answer in the clipboard.