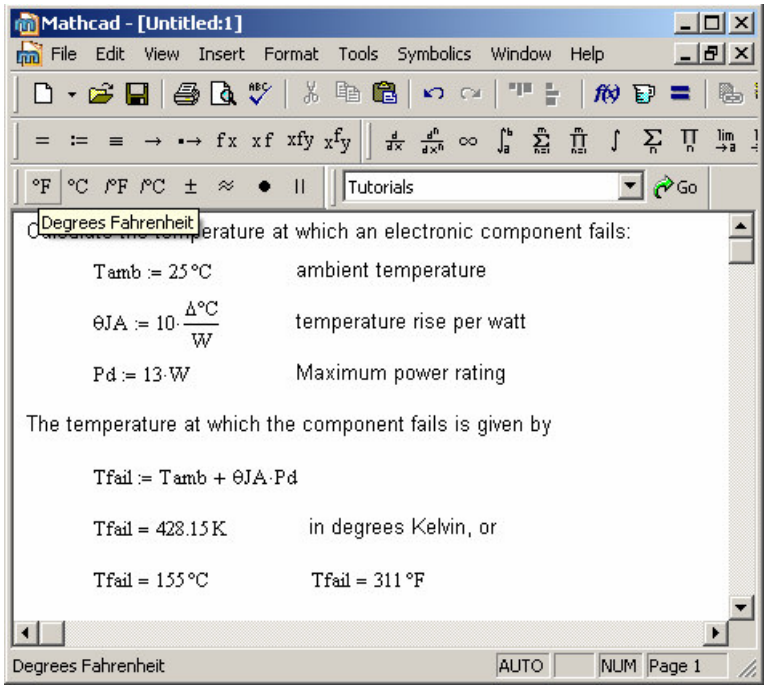

Figure 2: Temperature units allow users to express unit-checked results in common linear-transform scales. The Custom Characters toolbar is also shown, making it easy to enter symbols for degrees, and other common mathematical symbols.

Unit functions in the units placeholder open a whole new way to use the unit analysis functionality in Mathcad. Not only can you carry and balance functional units through calculations, you can display results in almost any conceivable scaling to suit documentation needs.

It is possible to define your own unit functions, using them in expressions with standard unit notation. Just like multiplicative units, the inverse function is used in the units placeholder; users are given a method to define both the forward and inverse functional conversions. Users can define any function/inverse pairs they require: decibels with respect to different base units, special time functions, scaling to a range, and so on. Several useful inverse function pairs are predefined in Mathcad 13 showing how the functionality can be extended. These functions convert common engineering data formats to Mathcad values with associated quantities: Degrees-Minutes-Seconds (DMS), Hours-Minutes-Seconds (hhmmss), and Feet-Inches-Fractions (FIF).

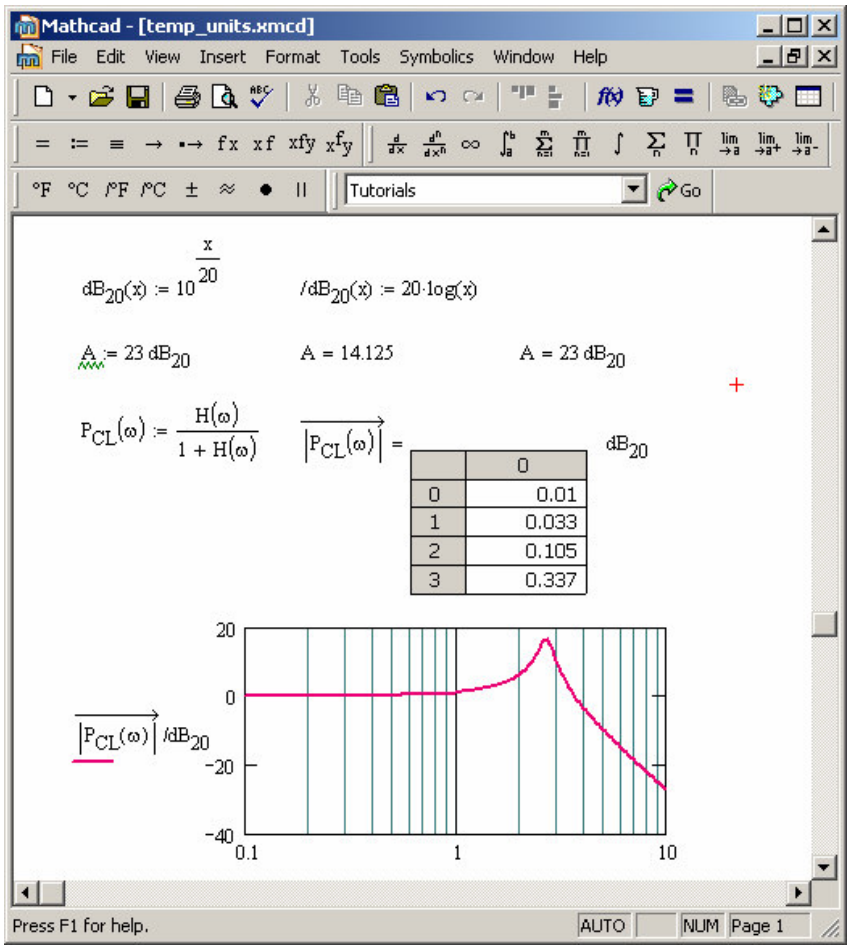

Figure 3: The definition of function and inverse function for decibels applied to inputs, results, and a graph.

NEW! Explicit Calculation

Many certification organizations require reported calculations to show substituted values before results are computed. Mathcad 13 introduces a new symbolic keyword, **explicit**, to do these substitutions. Users can choose to substitute some, all, or none of the defined variables in an expression without reducing the expression numerically. Several of these statements in subsequent calculations can show a step-by-step trail of substitutions, followed by a final numerical evaluation. This functionality greatly improves the ability of a user, auditor, or organization to verify and validate calculations.

Users also have the option to hide the symbolic keywords, or the entire left-hand-side of an expression, allowing them to create print-ready reports with automatic calculations. Symbolic evaluations can be displayed using the standard numerical = sign, to further standardize the presentation of these and other symbolically generated results.

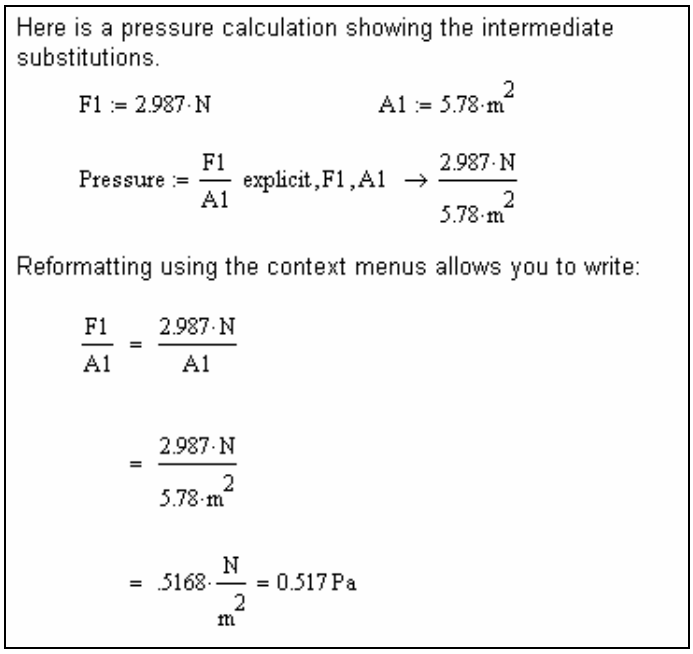

***Figure 4: A series of explicit calculations (keywords and left-hand sides hidden) showing substituted values followed by the final numerical result*.**

Like any symbolic evaluation, explicit calculations can be chained with other Mathcad operations, such as numerical definitions, numerical evaluations, or other symbolic keywords. The explicit keyword without modifiers can be used to mask numerical definitions in live symbolic calculations, which addresses a frequent functionality request.

Metadata Enhancements

Mathcad 12 introduced the calculation management concept of provenance on equations: copying equations from one document into another automatically creates an electronic audit trail of metadata. This information alleviates the "magic number" problem, in which a number of mysterious origin is used in an equation with no justification. Mathcad 13 enhances this functionality by creating provenance on more types of math regions, including results and referenced files. Provenance is also now created when multiple regions are copied and pasted simultaneously.

Regions with provenance and annotations can be shown by selecting **View > Annotations** from the menus. Regions with provenance metadata show up with colored parentheses enclosing the expression. Right-clicking on a region with colored parentheses allows you to view the tracking information.

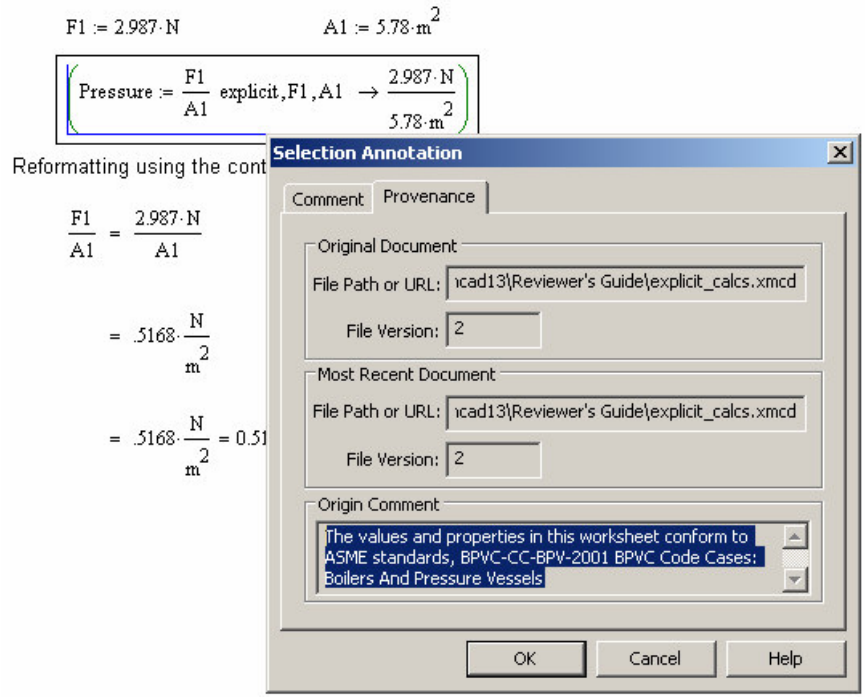

Figure 5: Using and viewing Annotations and Provenance metadata.

Further, documents can carry annotations just as individual regions do. When provenance metadata is created in Mathcad 13, the document annotations are also stored.

User-specified default units

Mathcad 12 introduced a way to customize the default units for a worksheet, so answers would be returned in the most convenient unit for any given quantity. In Mathcad 13, we have created a whole new interface for viewing existing unit settings for base and derived units and creating a custom unit system by adding, modifying and deleting rules for unit simplification.

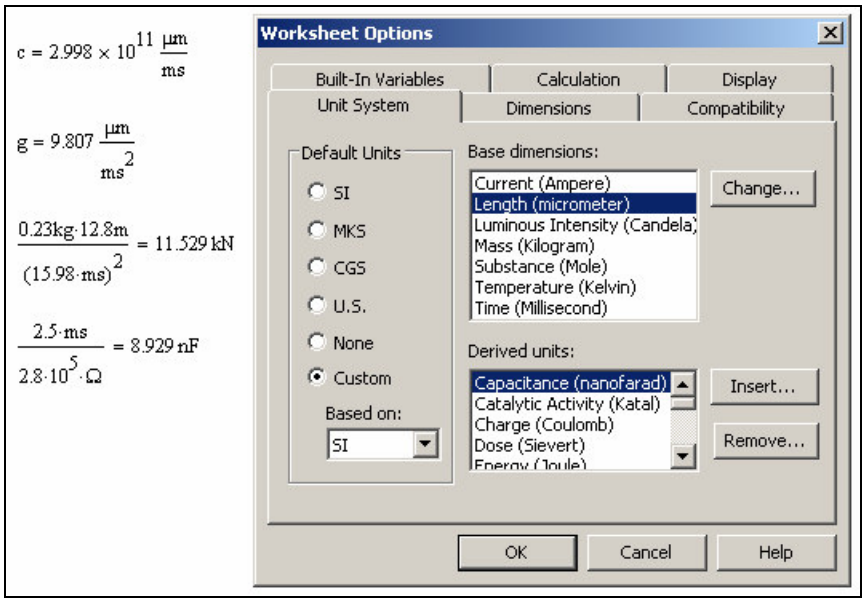

Figure 6: An example of a customized unit set showing the base and derived units used, and some sample automatic unit reductions.

NEW! Expanded 2D Graph trace options

A host of new line and symbol options have been added to 2D graphs in Mathcad 13. These options are designed to let users communicate complex technical information more clearly in visual format. The most important new option is the ability to choose the plotted symbol frequency independently from the total number of points used to create smooth connecting lines. For example, if you need 100 points to create a smooth sinusoidal curve, but you only wish to plot a symbol on every 10^{th} point, change the symbol frequency to 10. This results in a smooth plot with a manageable, readable number of symbols.

The full Windows color palette, new filled symbols, and independent control of line and symbol weight round out the trace settings feature set. These many new options, available through an updated, modern Windows style settings dialog, provide precise control over graph appearance.

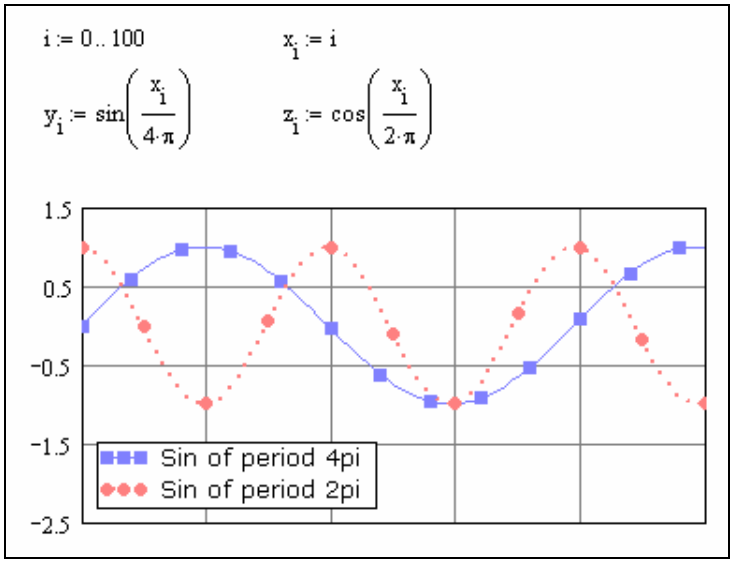

Figure 7: Symbol frequency, new symbols, more line colors, and other trace settings allow precise control over graph appearance, improving the readability of complex information.

Program Debugging and Math Functionality

Never forgetting our primary audience of practicing engineers, Mathcad 13 introduces several important new features for working with programs, parametric fitting, linear algebra, and other solvers.

NEW! Program Debugging

Mathcad programming is a powerful feature for writing multi-step, iterative, or conditional functions in Mathcad worksheets. We've extended this feature in Mathcad 13 with a program debugger. New Mathcad functions 'trace' and 'pause' allow users to report intermediate values in a program to a Trace Window, and optionally step or halt a program if it's not producing the desired results. The 'trace' and 'pause' functions will output to the Trace Window when debug mode is specifically toggled on, so users can leave these functions in their programs but only use them when needed. Results from the Trace Window can be copied for reuse elsewhere.

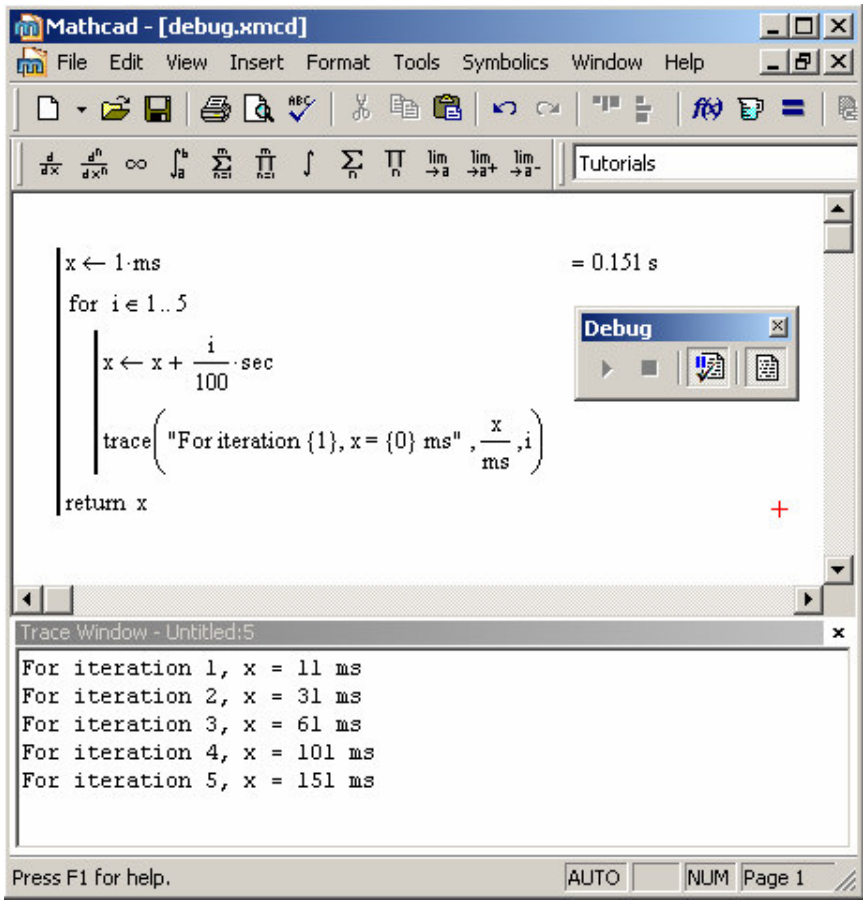

Figure 8: New trace function let users format and report intermediate program values in debug mode.

Enhanced parametric fitting

One of the workhorse Mathcad functions is **genfit**, a function that allows users to find values for the parameters that best fit a model function to a set of measured data. A number of improvements and extensions to this function are added in Mathcad 13. You can choose between the algorithm used in previous versions and a newer one. The new algorithm solves some types of model functions faster and more accurately than the old one. Further, the new algorithm will allow users to supply the fit function only, without generating symbolic derivatives for each fit parameter. This option saves time and complexity in setting up the function, and avoids bad fits resulting from incorrect derivatives, which is a common user error. The function has also been extended to accept any combination of parameter names, rather than insisting that parameters be specified as elements of a vector. This allows users to specify the model function using discipline-correct notation.

Enhanced linear algebra

Large matrix problems are one of the most common time-consuming tasks performed by numerical analysts. Mathcad 13 reduces the time required for all of the linear algebra functions, including matrix inversion, multiplication, linear system

solution, and eigenvalue reduction. The new linear algebra routines are based on the Intel BLAS libraries, which are an industry standard, and provide advances in both speed and functionality for our matrix routines.

Enhanced Unit Error Messages and Function Signatures

Mathcad has always returned error messages when units don't balance or a different type of value is required. Mathcad 13 enhances these messages for cases in which the values supplied have the wrong units. Unit-specific error messages decrease debugging time by showing you what quantities a function expects for its arguments. You can also look at function signatures in Mathcad 13 by evaluating the function name. The signature shows you the number and type of arguments, what their units are, and the type and units of the result.

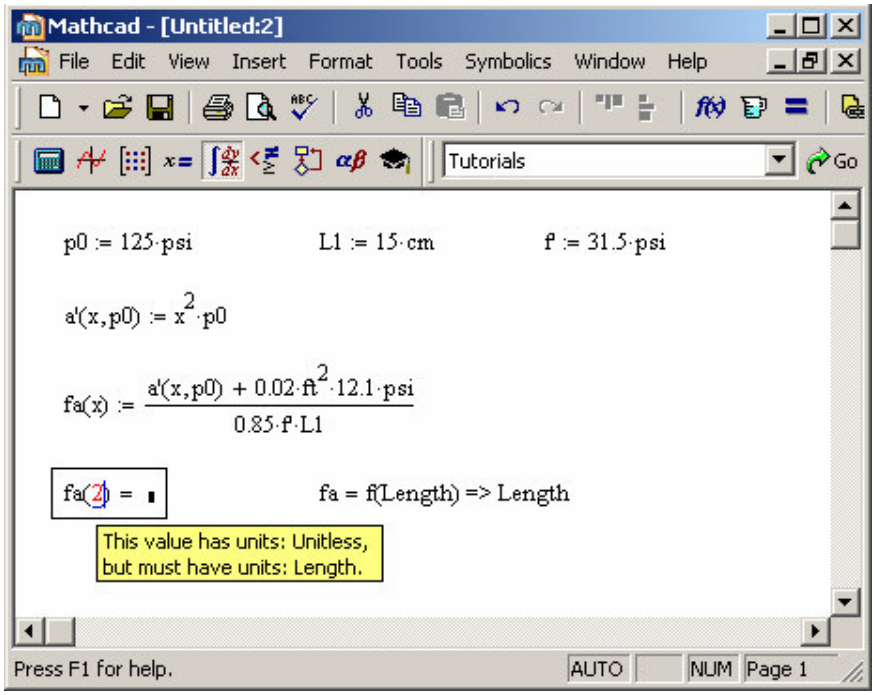

Figure 9: A Long, complex calculations will show the specific quantities supplied and required in the arguments for any function call.

UI and Documentation Additions

NEW! Autosave

Mathcad users can now choose to Autosave their work after a specified time interval, helping to manage loss of work in the event of a crash or hang.

NEW! Compatibility Switches

Over the last few versions, the behavior of certain Mathcad constructs has been changed. Users can now set which way they would like these constructs to behave,

according to version number, using the compatibility switches. The constructs that are switchable include array assignment of multiple variables, local assignment return value, 0/0, and the precision used for truncation functions.

Mathcad Resources

In this guide, we can only begin to convey the richness of Mathcad's capabilities. Fortunately, Mathcad can tell its own story, and the online Resources contain hundreds of worksheets that illustrate the depth of Mathcad's mathematical power and the variety of applications that Mathcad supports. All of these worksheets are live, so as you read the Tutorials or QuickSheets you can watch Mathcad at work, change the inputs, and see the results in the outputs as they update. The Resources window automatically pops up the first time you start Mathcad. From then on, you can access the **Tutorials**, **QuickSheets**, or **Reference Tables** by choosing one of these Resources from the **Help** menu, or by using the Resources toolbar.

A new Tutorial on **Programming** addresses new user unfamiliarity, and gets users up to speed quickly with one of Mathcad's most powerful mathematical features.

New Quicksheets have been added to help users perform common analyses. Most notably, there are sheets showing how to construct **Waterfall Plots**, solve for parameters in an **ODE Data Regression**, and define custom operators such as **Parallel Resistance** and **Polar Notation** for complex numbers.

NEW! Migration Guide

In addition to the traditional Help and Resources, Mathcad 13 documentation offers an important new resource for experienced users. The **Migration Guide** discusses math constructs that have been upgraded in recent versions and any incompatibilities those may create with older versions. The guide gives specific instructions on migrating documents efficiently to take advantage of new functionality and remove errors. It shows best practices to optimize documents, and suggests methods of working with the latest Mathcad technology.