

Mathematica Link for LabVIEW

From Design to Deployment:

Leveraging the combined power of LabVIEW and *Mathematica*
throughout the development cycle

Introduction

As research and production cycles accelerate, researchers and design engineers are under increasing pressure to get results faster than ever before. To meet these challenges, many technical professionals turn to LabVIEW or *Mathematica* as their technical computing tool of choice. And while both of these powerful programs can be used successfully without the aid of the other, dramatic productivity gains can be realized by using both programs together in an integrated, hybrid workflow.

The LabVIEW Perspective

Countless technical users worldwide have standardized on LabVIEW. With its rapid application development capabilities, integrated GUI (graphical user interface) panels, robust I/O connectivity, and run-time productivity advantages, it is not hard to understand why. Still, imagine how flexible LabVIEW could become with the addition of *Mathematica*'s powerful, integrated "Notebook" interface. *Mathematica*—complete with its advanced function libraries and symbolic mathematical capabilities, is the perfect complement to LabVIEW's graphical VI ("Virtual Instrument") metaphor. Consider how convenient a hybrid LabVIEW/*Mathematica* system might be—particularly during the early stages of the design cycle, before the parameters of the design problem have been completely defined. Rather than committing to several wire-test-then-rewire cycles, you could use *Mathematica*'s rich command-style interface from within LabVIEW, developing, testing, and verifying your mathematical models in a single, integrated workflow. Imagine the productivity advantages of a single environment that transparently accommodates both advanced mathematical notation and real-world, LabVIEW-acquired data. This is the *Mathematica Link for LabVIEW* advantage.

The *Mathematica* Perspective

While LabVIEW leads the software pack for data acquisition and test & measurement applications, many engineers, physicists, researchers and applied scientists turn to *Mathematica* when they face a difficult application challenge. And for good reason: *Mathematica*'s flexible Notebook interface and rich programming language enable mathematically-savvy users to interactively explore a complex problem in many different ways. *Mathematica*'s underlying architecture is based on transformation rules and pattern matching, which in turn provides powerful symbolic capabilities. Indeed, if you require sophisticated mathematical modeling and symbolic manipulation, *Mathematica*'s extensive function libraries and specialized Add-ons are unmatched in the technical software industry. But while the standard *Mathematica* distribution is unrivaled for theoretical problem-solving, wouldn't it be convenient if you could acquire experimental data directly into your *Mathematica* Notebook using National Instruments data acquisition hardware? Or perhaps you could benefit from quickly and easily embedding your *Mathematica* commands into a full-featured LabVIEW GUI application for convenience, experimental repeatability, enhanced instrument connectivity, and improved usability (as illustrated in Figure 1).

A Synergistic Workflow

These and other productivity advantages come into reach when “*Mathematica Link for LabVIEW*” becomes part of your technical tool arsenal. The “*Mathematica Link for LabVIEW*” toolkit includes over 150 LabVIEW and *Mathematica* components to facilitate seamless bi-directional data transfer between LabVIEW and *Mathematica*. With this robust data exchange mechanism in place, you can realize countless synergistic workflow advantages by leveraging the combined power of two of the world’s most popular technical applications: LabVIEW and *Mathematica*.

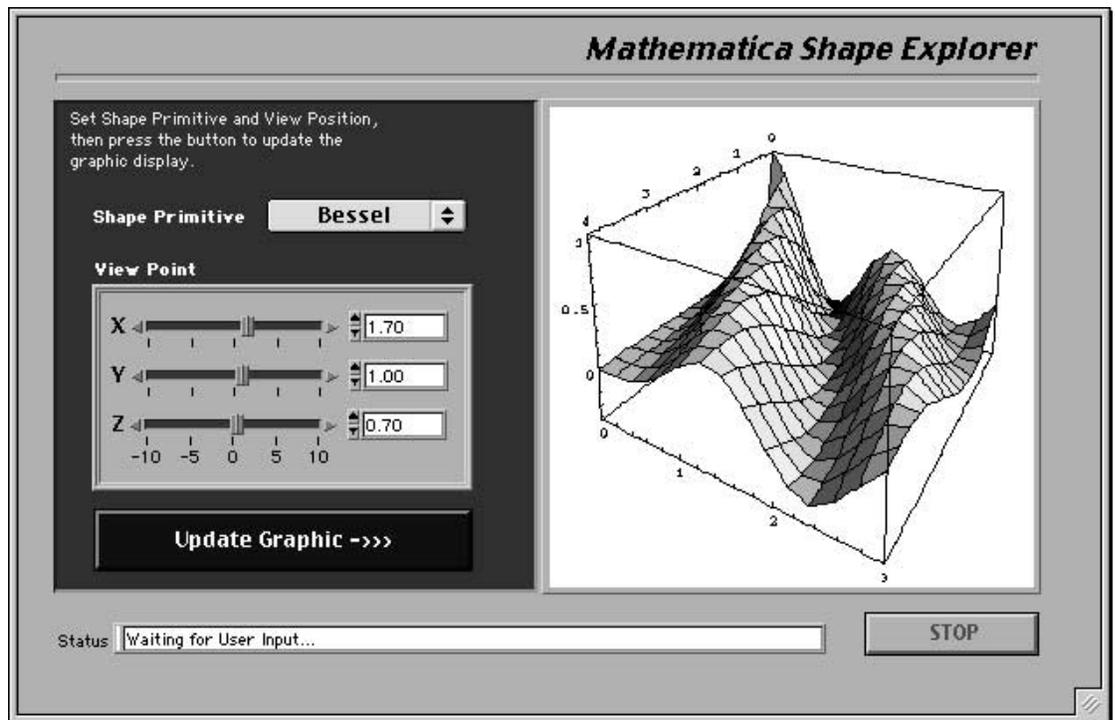


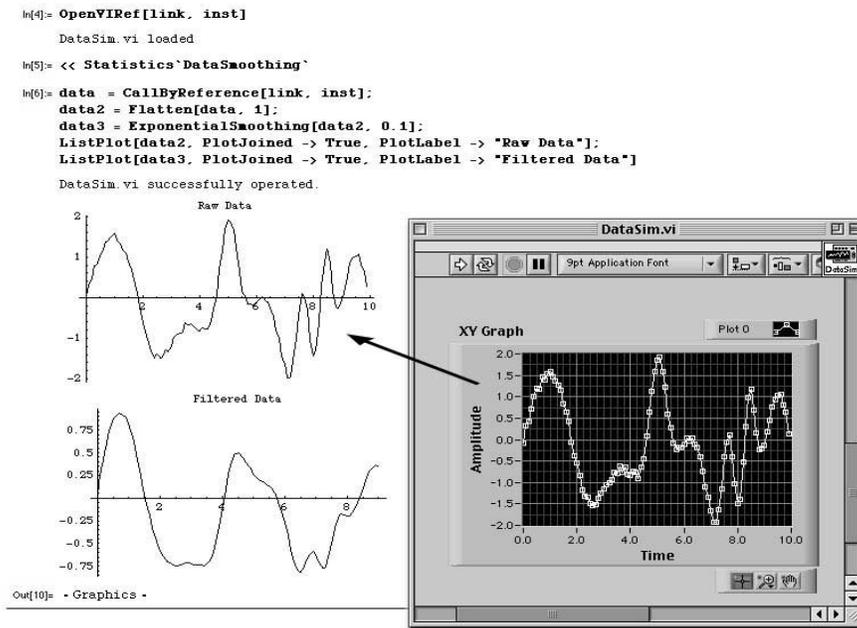
Figure 1. The Shape Explorer Demo VI – a simple example of a hybrid Mathematica/LabVIEW application.

Mathematica Link for LabVIEW: Overview

At the most basic level, *Mathematica Link for LabVIEW* is a bi-directional communications link between LabVIEW and *Mathematica*. The package enables you to:

- call a LabVIEW VI as a subprocess of a *Mathematica* Notebook, and
- call *Mathematica*'s *MathKernel* as a subprocess of a LabVIEW VI.

These capabilities are highlighted in Figures 2 and 3 respectively:



**LabVIEW VI called from a
Mathematica Notebook**
- VI data acquired directly into
the notebook session.

Figure 2. LabVIEW functioning as a subprocess of Mathematica.

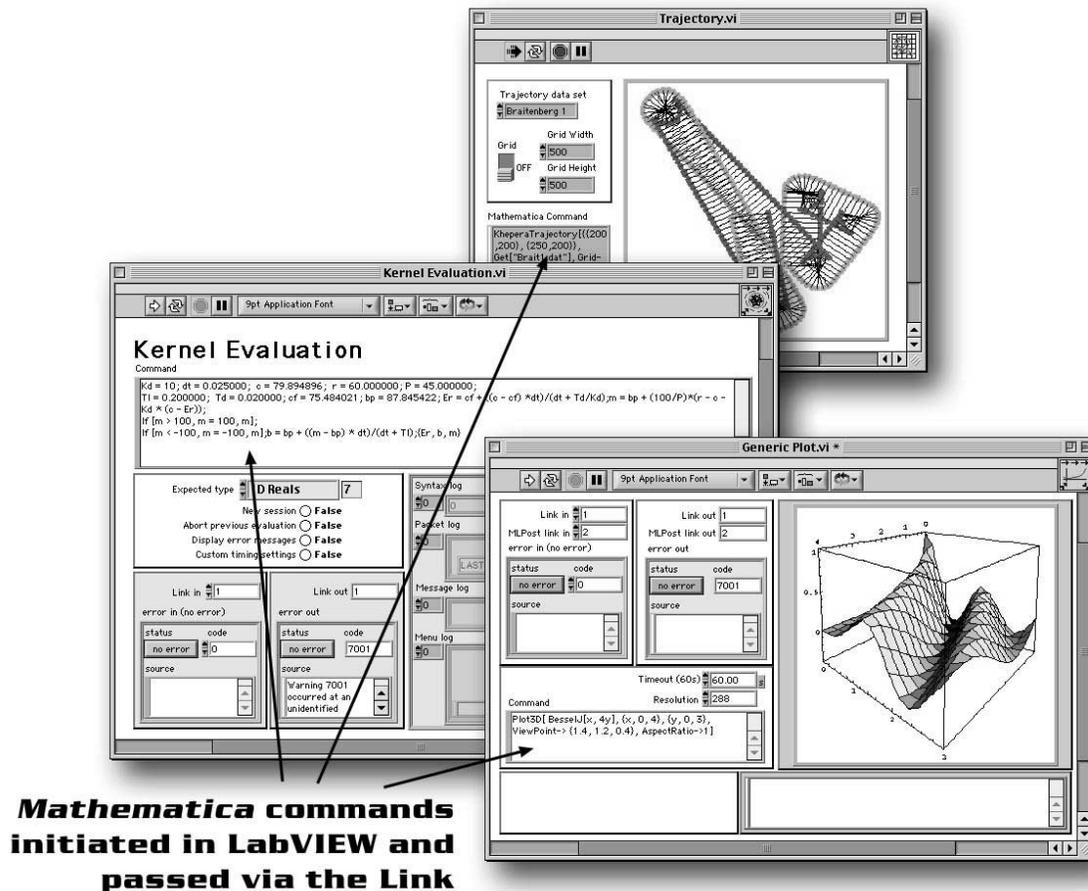


Figure 3. Mathematica, functioning as a subprocess of LabVIEW.

Several Tools in One Package

While *Mathematica Link for LabVIEW* is fundamentally a communications tool, a large number of examples and extra components are included in the package. These components give you immediate access to a wide range of useful functions without the need for additional programming. Using only the supplied components, *Mathematica Link for LabVIEW* is:

1. A powerful, text-based symbolic interpreter for your LabVIEW VIs. Never again will you need to launch MathCad, MatLab or *Mathematica*, and swap back and forth between applications when working with involved symbolic calculations. Simply send the calculations from a LabVIEW VI to the *MathKernel* for evaluation. Or build the symbolic calculations right into your VI and return results dynamically at run-time!

2. A dynamic, interactive visualization tool. Sure, you can wire a VI to display a complex mathematical function in a Picture Control or X-Y graph. But why spend the time when you can get the same results in a fraction of the time simply by passing a function request to *Mathematica*? The standard installation includes VIs to pass text, graphic content, and standard data types across the Link for processing, plotting and visualization. PostScript image data can be generated and rasterized by the *MathKernel*, then returned for display directly on your LabVIEW front panels.

3. An archiving and reporting tool. When you run a series of experiments to test different parameters, you typically need to associate the input parameters with the resulting output. In most cases, you also need an efficient way to document this information for further analysis, future reference, and to ensure repeatability of your experiments. Users familiar with the *Mathematica* Notebook know that *Mathematica* allows you to save your entire session in a single file. With *Mathematica Link for LabVIEW*, you can extend this capability into the LabVIEW domain. Simply call your test VI from *Mathematica*, return the LabVIEW data to the *Mathematica* Notebook, then process the data in the Notebook as necessary. When you save the Notebook file, all the steps of your experiment are automatically recorded—including the VIs you called, the parameters passed, the data return, the subsequent post-processing steps, and so on. Should you need to review the experiment at a later date, all of the information has been meticulously documented - automatically!

4. A graphic file import/export utility. LabVIEW's graphics support is adequate for some applications. However, with the formats limited to .JPG, .PNG, and .BMP, it is difficult to produce high quality printed output directly from LabVIEW data. If your reporting requirements call for publication-quality Postscript files, .PDF, .TIF, or .DXF formats, LabVIEW alone won't provide the solution. On the other hand, *Mathematica* supports a wide range of graphics formats including those listed above, and several others. Simply pass your data or function requests to the *MathKernel*, then instruct the *MathKernel* to export the file in the graphics format of your choice. *Mathematica*'s built-in graphics functions take care of the rest!

5. A way to access powerful *Mathematica* Add-ons from within LabVIEW. The standard *Mathematica* installation offers an extensive library of built-in functions. Still, you may benefit the advanced features and productivity advantages of *Mathematica*'s Add-on packages. Several toolkits are available including Control System Professional, the Digital Image Processing Toolkit, Signals and Systems, Wavelet Explorer, the Fuzzy Logic Toolkit, to name a few. *Mathematica Link for LabVIEW* permits you to access the domain-specific functions in these powerful toolkits from within your LabVIEW VIs. After using the *Mathematica* Notebook to interactively design your solution, you can develop a stand-alone GUI VI and continue to use these same advanced tools from within the LabVIEW environment!

6. A means to acquire data directly into a *Mathematica* Notebook. Build new data acquisition VIs in LabVIEW, or use your existing VI libraries. Do as little or as much of the preprocessing in LabVIEW as you like, then pass the results to *Mathematica* for

further processing and analysis. The *Mathematica Link for LabVIEW* tools allow you to call any VI from within *Mathematica*, and you can specify which parameters you want passed. Specify front panel controls values, and even change control values on the fly - all from within a *Mathematica* Notebook. And when you are finished, save the entire session in a Notebook file to create a permanent record of your entire experiment!

7. A robust tool for quickly building GUI applications with advanced mathematical sophistication. *Mathematica* is a powerful tool in the hands of mathematically-savvy professionals. However, if you are developing an application that will be used by end users or lab assistants with varying levels of mathematical capability, you may prefer to hide the mathematical complexity below the surface. *Mathematica Link for LabVIEW* enables you to build an easy-to-use GUI interface using LabVIEW, while still offering all of the mathematical sophistication of the entire *Mathematica* function library. From the end user's point of view, they can access all of the power and flexibility without the implied complexity.

These are just a few of the advantages you can expect to enjoy with the standard *Mathematica Link for LabVIEW* distribution. And because it is an open system with full access to all VI diagrams and *Mathematica* components, you are free to extend the toolkit as necessary for your own applications.

Structure of the Link

The low-level component that provides the communication link between the *Mathematica*'s *MathKernel* and the outside world is Wolfram's own *MathLink*. *MathLink* is a library of C functions that manage program-to-program communication. The *MathLink* protocol is responsible for all communication with the kernel, including communication with the standard *Mathematica* front end. This protocol can also be used to establish a connection between the kernel and an external *MathLink*-compatible program. (Interested parties are directed to the *Mathematica* documentation for more information about *MathLink*.)

Software Architecture

Mathematica Link for LabVIEW is a straightforward LabVIEW translation of the C functions contained in the *MathLink* library. A structural depiction of the software hierarchy is illustrated in Figure 4.

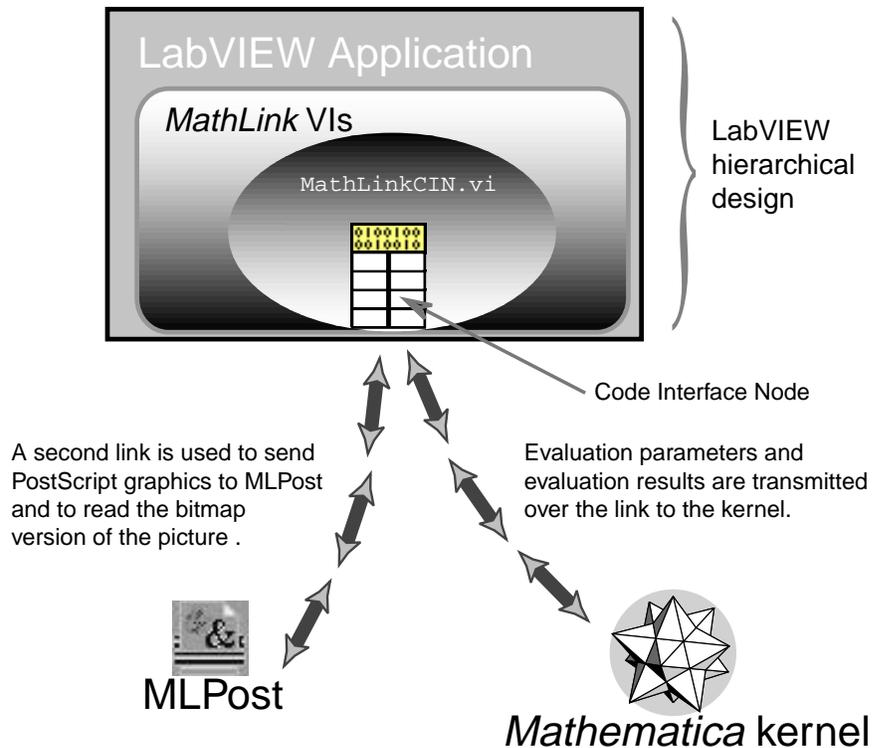


Figure 4. A structural overview of Mathematica Link for LabVIEW.

Using the mechanism illustrated here, a LabVIEW application can invoke *MathLink* functions simply by calling VIs that implement the desired *MathLink* functionality. Each *MathLink* VI implements a single *MathLink* function. So, for instance, you will find functions such as **MLOpen.vi**, **MLPutInteger.vi**, and so on in the standard *Mathematica Link for LabVIEW* distribution. However, not all of the VIs included in the distribution are *MathLink* VIs. Several libraries provide utilities or high-level VIs, which have no direct *MathLink* counterparts. Some specific examples of these VIs will be discussed next.

High-Level Functions

The high-level VIs provide direct access to the most common Link functions, and are designed to be used immediately by users with only a minimal familiarity with *MathLink*. They provide ready-made solutions for a variety of standard applications.

MathLink VI Server.vi

This VI is a bridge between *MathLink* and various LabVIEW VI Server functions. It interfaces with a custom *Mathematica* script containing strict *Mathematica* equivalents to several native LabVIEW VI Server functions. Thus, in a *Mathematica* Notebook you can call functions such as `OpenVIRef`, `OpenPanel`, `RunVI`, and others to control your LabVIEW applications directly. Furthermore, LabVIEW/*MathLink* sessions initiated and

administered by these functions can run either on the same computer or on different computers linked over a network.

Kernel Evaluation.vi

This VI passes a string of commands in standard *Mathematica* syntax to the kernel for evaluation. When the evaluation completes, the results are returned to LabVIEW. When using this particular VI, you need to specify the expected data type for the result from among a finite number of data types: Boolean, Integers, Real, Complex, Strings, and lists and matrices of all these types. The results of the evaluation are returned in the requested data format to front panel indicators and the VI's connector pane. **Kernel Evaluation.vi** has high-level functions to seamlessly manage all aspects of a *Mathematica* session, such as *MathKernel* messages, error messages, and so on. It is designed to function as a general-purpose interface to *Mathematica* functions from within a LabVIEW diagram.

Mathematica Graphics Generators

Four VIs are provided to make *Mathematica* graphical functions available to LabVIEW. **Generic ListPlot.vi** displays a graphical representation of sets of data. **Generic Plot.vi** is used to visualize, from within LabVIEW, graphics returned by *Mathematica* commands. **Display Bitmap.vi** is similar in function to **Generic Plot.vi**, but uses *Mathematica* graphic functions to generate a .BMP file. **Generate Graphic File.vi** illustrates how images generated in *Mathematica* can be exported to one of several different image file formats including .PS, .PDF, .TIF, and .DXF.

Development Tools

A number of VIs included are designed to help you build your own *MathLink* applications. They combine the action of several *MathLink* functions in ways similar to those frequently encountered when developing your own *MathLink* applications.

Data Transfer VIs

Thirty VIs are provided to transfer the most commonly used data structures including lists and matrices of all kinds. The VIs are documented and their source code is accessible, so they can be the basis for building new VIs to transfer other data structures of your own specification.

Error Handling VIs

Mathematica Link for LabVIEW follows the LabVIEW error cluster scheme for managing *MathLink* errors. All the VIs have "Error in" and "Error out" clusters. A **MathLink Error Manager.vi**, based on **General Error Handler.vi**, is provided to identify *MathLink* errors and take the appropriate corrective action when possible.

Dedicated MathLink VIs

Forty-three of the most commonly used *MathLink* functions are implemented directly in *Mathematica Link for LabVIEW*.

Other Features of the Development Environment

A number of example VIs and *Mathematica* .nb files are provided to illustrate how the *MathLink* components can be used in various combinations to construct hybrid applications. Some of the examples are based on the standard VIs contained in the LabVIEW distribution that have been modified slightly to take advantage of *MathLink* functions more efficiently. Others —such as the PID Control Demo and the *Mathematica* Shape Explorer —are entirely new VIs, developed for the package using Link components as subVIs. Comments, guidelines, and recommendations are provided in several of the VIs to accelerate the development of your own *MathLink* applications, and three hands-on tutorial Notebooks are provided to get you up to speed quickly with the steps required for calling LabVIEW VIs from *Mathematica* .

A Flexible Toolkit with Wide Ranging Applications

Mathematica Link for LabVIEW is designed for flexibility. Therefore, it has been adopted by professionals in several different fields of endeavor, including:

- Bioscience
- Pharmaceuticals
- Automotive R & D
- Aerospace
- Applied Physics
- Mechanical, Chemical, and Electrical Engineering
- Academia

Pick up your own copy of *Mathematica Link for LabVIEW*, and begin realizing the productivity benefits of a combined LabVIEW and *Mathematica* Workflow!

Mathematica Link for LabVIEW is distributed worldwide by Wolfram Research Inc. It is also available directly from BetterVIEW Consulting. For more information contact:

Wolfram Research Inc. - Worldwide Distributor
<http://www.wolfram.com/products/applications/labview/>

BetterVIEW Consulting
<http://www.bettervi.com/mlink/>
email: solutions@bettervi.com