JLAB: Matlab freeware for data analysis

Jonathan M. Lilly

August 27, 2004

1 Introduction

JLAB is a set of Matlab functions I have written or co-written over the past ten years for the purpose of analyzing data. It consists of over two hundred and fifty m-files spanning fifteen thousand lines of code. JLAB includes functions ranging in complexity from one-line aliases to high-level algorithms for certain specialized tasks. These have been collected together and made publicly available for you to use, modify, and — subject to certain very reasonable constraints — to redistribute.

Some of the highlights are: a suite of functions for the rapid manipulation of multi-component, potentially multi-dimensional datasets; a systematic way of dealing with datasets having components of non-uniform length; tools for fine-tuning figures using compact, straightforward statements; and specialized functions for spectral and time / frequency analysis, including wavelet algorithms developed by the author.

This collection of functions emerged in the course of my research, as I sought more natural solutions for data processing than were provided by standard Matlab. My philosophy is to focus on the creation relatively low-level functions which perform simple common tasks. As a result, perhaps ninety percent of the functions I use on a regular basis are from JLAB, and overall I would say I am able to work perhaps five to ten times faster using this collection than I would be able to otherwise.

This is the first public release of JLAB and hence should be considered a "beta" version. While I have endeavored to make sure all functions are stable and well-tested, other users operating on other platforms will no doubt encounter idiosyncrasies or bugs. If you have any problems, or indeed successes, using JLAB, I would very much appreciate hearing from you at `software@jmlilly.net`.

It is my intention to release updated versions of JLAB periodically, and to include, to the extent that it is feasible, the functions and figure-generating scripts used in future papers. If you would like to be notified of such releases, please contact me at the above email address.

The next section explains in more detail the motivation for this project. Installation instructions are given in Section 3.

2 Motivation

Despite its many advantages as an environment for data processing and figure creation, I have found Matlab to be inadequate on a number of counts. Many common matrix or array operations have no built-in representation, leading to the frequent occurrence of rather awkward constructions. No framework exists for manipulating multi-component datasets, in which one may wish to perform the identical operation on a number of different variables. The "handle graphics" interface is particularly clumsy, and leads to inordinately large amounts of time spent tweaking low-level properties in order to generate high-quality figures.

The preparation of a collection such as this one has been a major endeavor, but seemed to me to be a necessity for my own work. In the course of my everyday research, I regularly wrote new functions which abstracted the essence of a task, so that I could use these in the future. However, under the constant pressure to produce results, I did not take the last step to assimilate my programs into a tested, stable, and organized collection.

After working on perhaps ten projects in as many years, Matlab m-files were scattered over many directories, with no systematic index or organization. Functions written years earlier were lost, forgotten about, re-written, and eventually rediscovered at some inopportune moment. Changes in one function would inadvertently break another function, thus I was not able to reproduce earlier computations without substantial mucking around. Clearly my effort was not adding up.

It seemed to me that I was faced with two alternatives: either to spend my entire life hacking amidst chaos in order to maximize short-term production, or invest the time, now, to create a stable base on which future work could easily be built. If others find this collection useful in their own research, so much the better.

3 Installation

The latest version of JLAB is available at `www.jmlilly.net`. It is available in two formats, a `gzipped` `tar` archive for Unix / Linux users, and an uncompressed version available via `ftp`.

For the tar archive, put jlab.tar into the directory you would like the `jlab` directory to reside in, and run

```
tar -xf jlab.tar
```

which will unpack all files into a new sub-directory `jlab`. Then put the appropriate path statement into your `startup.m` file, e.g.

```
addpath /usr/home/lilly/matlab/jlab
```

with the obvious modifications.

Alternatively you may `ftp` to `www.jmlilly.net` with the following settings

```
username:  guestftp
password:  openses@me
```

and then change to the `jlab` directory and download everything (including a few `.mat` and `.pdf` files) into a local directory called `jlab`. Finally, set the path as above.