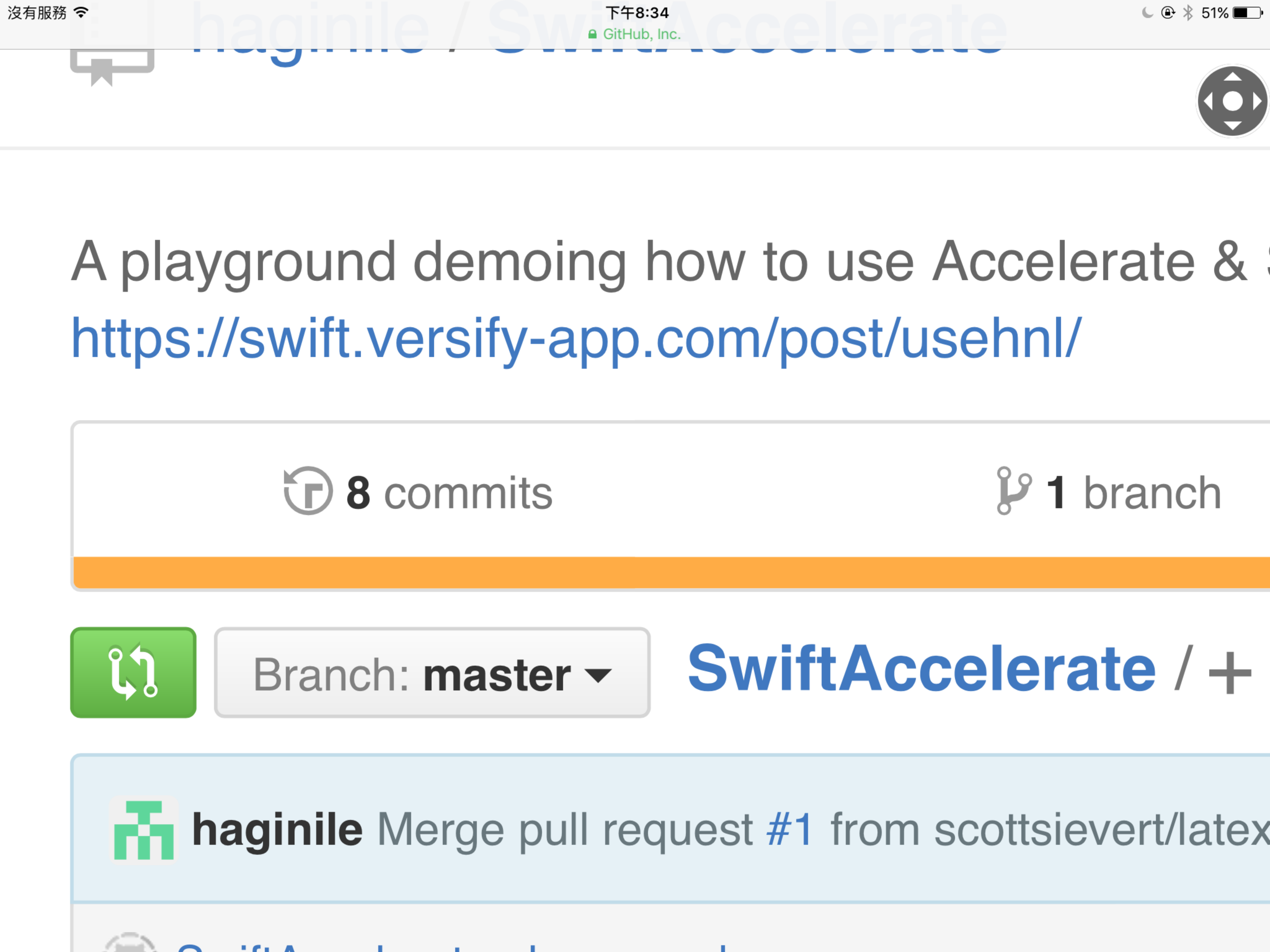
Swift Matrix manipulation



沒有服務 下午8:34 51%

GitHub, Inc.

haginile / SwiftAccelerate

A playground demoing how to use Accelerate &

https://swift.versify-app.com/post/usehnl/

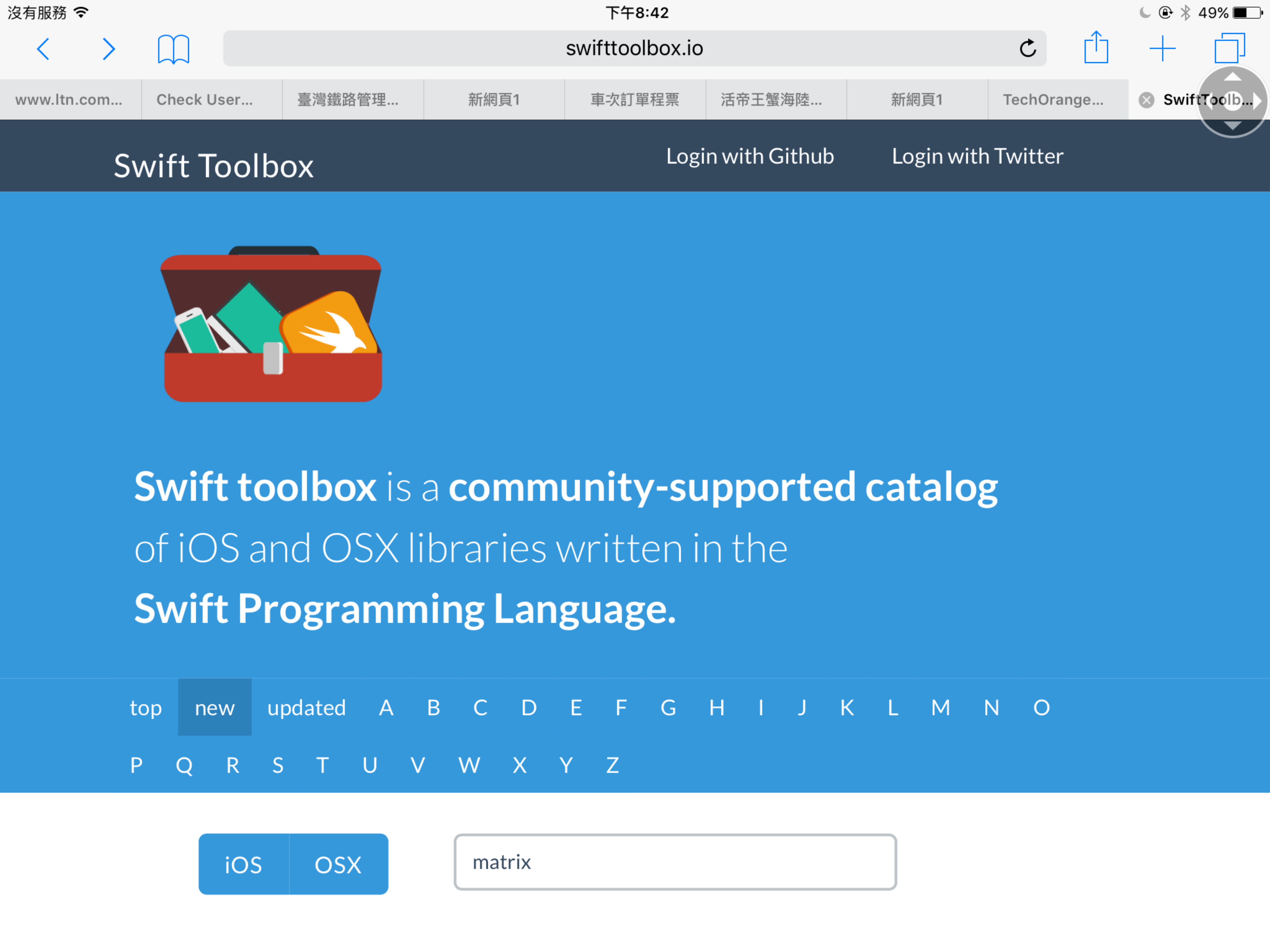
8 commits

1 branch

Branch: **master**

SwiftAccelerate / +

haginile Merge pull request #1 from scottsievert/latex



沒有服務 下午8:42 49%

swifttoolbox.io

www.ltn.com... | Check User... | 臺灣鐵路管理... | 新網頁1 | 車次訂單程票 | 活帝王蟹海陸... | 新網頁1 | TechOrange... | SwiftToolb...

Swift Toolbox

Login with Github Login with Twitter

Swift toolbox is a **community-supported catalog** of iOS and OSX libraries written in the **Swift Programming Language.**

top new updated A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

iOS OSX

matrix

Quick search

Swift Matrix and Machine Learning Library

Apple's Swift is a high level language that's *asking* for some numerical library to perform computation f
the very least *easily*. A way to have iOS run high-level code similar to Python or Matlab is something I'v
waiting for, and am incredibly excited to see the results. This will make porting complex signal processi
algorithms to C much easier. Porting from Python/MATLAB to C was (and is) a pain in the butt, and this
aims to make the conversion between a Python/Matlab algorithm and a mobile app *simple*.

Currently, the **Sw**ift Matr**ix** Library or **swix** gives you

- operators (+, etc) and various functions (sin, etc) that operate on entire arrays
- easy initializers for 1D and 2D arrays
- dot product, matrix inversion, eigenvalues, etc
- machine learning algorithms (SVM, kNN, SVD/PCA, more to come)
- speed optimizations
- one dimensional Fourier transforms

In most cases, this library calls the Accelerate framework or OpenCV. I optimized what I needed to, me
operators and select mathematical functions are fast while the functions I didn't need are slow. If you w
speed up some function or add add another feature in those libraries, feel free to submit a pull request
Github (preferred!) or contact me at @stsievert or sieve121@umn.edu. Oh, and if you use this project I
to hear about it!

When I was crafting this library, I primarily followed the footsteps and example set by NumPy. For the m
complex mathematical functions (e.g., SVD) I tested it against NumPy. Matlab, at least for the SVD, ret
slightly different output.