$Id: README.debian,v 1.2 2024/06/29 22:01:42 pm Exp $

QUICK START:
1. Install packages that BSU depends on.
   BLAS (libblas-dev libblas3)
   LAPACK (liblapack-dev liblbapack3)
   GSL (gsl-bin libgsl-dev libgslcblas0)
   GNUPLOT
        gnuplot
        gnuplot-data
        gnuplot-qt
   LIBMSEED
   libmseed2 and libmseed-dev
   PLPLOT
libplplot-data
libplplot-dev
libplplot-java
libplplot-lua:amd64
libplplot-ocaml
libplplot-ocaml-dev
libplplot17:amd64
libplplotada4.1:amd64
libplplotada5-dev
libplplotcxx15:amd64
libplplotfortran0:amd64
libplplotqt2:amd64
libplplotwxwidgets1:amd64
octave-plplot:amd64
plplot-doc
plplot-driver-cairo:amd64
plplot-driver-qt:amd64
plplot-driver-wxwidgets:amd64
plplot-driver-xwin:amd64
plplot-examples
plplot-tcl
plplot-tcl-bin
plplot-tcl-dev
python3-plplot
python3-plplot-qt

NOTE: The default configure uses GNUPLOT rather than PLPLOT,
so if you have trouble with plplot.

(see details below if a problem here)

2. Install BSU Binary Packages:
For 64 Bit machines:
sudo dpkg -i bsu_3.0.3-1_amd64.deb
  OR
sudo gdebi bsu_3.0.3-1_amd64.deb

For 32 Bit machines:
sudo dpkg -i bsu_3.0.3-1_i386.deb

====================DETAILS==================================
Tools for downloading and/or installation of packages include:

dpkg
apt-get
aptitude (like yum on CentOs or other Redhat)
synaptic (like yumex on CentOs or other Redhat, GUI Tool)

IMPORTANT NAMING ISSUE:
Exact package names will change with each new release of a distribution.
The following examples will give you an idea of what to look for.
In general, Debian tends toward many sub-packages (compared to Redhat).
To compile code, you will need development packages. These typically
end in *-dev for Debian (compared to *-devel in Redhat).
RECOMMENDATION: Use the synaptic tool and search button to look for all
the plplot packages, or all the lapack packages, etc., then install all
of them.

==========SPECIFIC SOURCE PACKAGING DETAILS:
Debian (and Debian based distributions, like Ubuntu)

WHAT IS A SOURCE PACKAGE?
In Debian distributions, this would typically be several files:
  bsu_3.0.3.orig.tar.gz
  bsu_3.0.3-1.dsc
  bsu_3.0.3-1.diff.gz

TIP: If you need a source package from a repository, you can do it with
the apt-get command. Here is an example to get the blas package:

  apt-get source blas

Assuming a working internet connection, apt-get would download the
following files (or ones like these) into your current directory.
  blas_1.2.orig.tar.gz
  blas_1.2-1.3ubuntu4.dsc
  blas_1.2-1.3ubuntu4.diff.gz

With the BSU package, you can download the files from
http://cgiss.boisestate.edu/~pm/downloads.php

BUILDING A SOURCE PACKAGE:
You would unpack the source with a command like this:

  dpkg-source -x bsu_3.0.3-1.dsc

This will untar the archive file into a directory, here named bsu-3.0.3
Change into that directory, then into the debian directory:

  cd bsu-3.0.3/debian

You then edit what files you need (for example, at least edit the
changelog file, use your email and make the release number name specific
to yourself).

Then, change back one directory,

  cd ../

and issue the command,
Builds both binary and source:

```
dpkg-buildpackage -rfakeroot -uc -us
```

Builds binary:

```
dpkg-buildpackage -rfakeroot -b -uc
```

And the package should start building. If all goes well, it will build a new package for your architecture, and place it just above the bsu-3.0.0 directory.

Of course, a lot of things can go wrong. If you are missing dependencies on your system, sometimes a good command to issue is

```
apt-get build-dep bsu
```

(where bsu would be the root name of whatever package you are building). This may seek out and install whatever you need.

TO BUILD A PACKAGE FROM SCRATCH
You will want to learn about dh_make and related programs. Too big a topic for this readme.