

# Macaulay2 in Debian

Doug Torrance

June 26, 2020



# What is Debian?

Debian is a free operating system with support for a wide variety of architectures.

Most Debian systems run on the Linux kernel, but there is also some support for the FreeBSD and Hurd kernels.

At the heart of Debian is the APT package manager, which allows users to install software easily.

```
sudo apt install foo
```

# Downstream distributions

According to **distrowatch.com**, 7 out of the top 10 most popular Linux distributions are based on Debian:

- ▶ MX Linux
- ▶ Mint
- ▶ Ubuntu
- ▶ Debian
- ▶ elementary
- ▶ Zorin
- ▶ Pop!\_OS

# Goal

Get a Macaulay2 package in Debian!

## Big hurdle – Debian policy 4.13, Embedded code copies

“Some software packages include in their distribution convenience copies of code from other software packages, generally so that users compiling from source don’t have to download multiple packages. Debian packages should not make use of these convenience copies unless the included package is explicitly intended to be used in this way. If the included code is already in the Debian archive in the form of a library, the Debian packaging should ensure that binary packages reference the libraries already in Debian and the convenience copy is not used. If the included code is not already in Debian, it should be packaged separately as a prerequisite if possible.”

# Packaging dependencies

Dependencies I have packaged from scratch:

- ▶ cohomcalg
- ▶ frobby
- ▶ mathic
- ▶ mathicgb
- ▶ memtailor
- ▶ topcom
- ▶ mpsolve (in NEW queue)
- ▶ bootsidemenu.js (waiting for sponsor)
- ▶ node-nouislider (waiting for sponsor)

Dependencies I have adopted:

- ▶ coinor-csdp
- ▶ fflas-ffpack
- ▶ gfan
- ▶ givaro

## mpir v. gmp

Until recently (when the `eigen` branch was merged), Macaulay2 needed to use a patched version of `mpir`, and building using the Debian `gmp` package resulted in segfaults.

But with the release of 1.16, a Macaulay2 Debian package is finally possible!

## .deb binary packages

When you run `apt install foo`, a `foo.deb` file is downloaded, which contain two tarballs:

- ▶ `control.tar.xz`, containing metadata, scripts, etc.
- ▶ `data.tar.xz`, containing the actual files to install



# Source packages

In order to upload a package into Debian, you must provide a *source* package, which consists of three files:

- ▶ `foo_0.1-1.dsc`, containing metadata
- ▶ `foo_0.1.orig.tar.xz`, containing the upstream source
- ▶ `foo_0.1-1.debian.tar.xz`, containing the contents of the `debian` directory

Binary packages are built from source packages using `dpkg-buildpackage`.

# debian directory

## Contains:

- ▶ `control`: metadata
- ▶ `copyright`: license information
- ▶ `rules`: Makefile for building the package. `debhelper` does most of the heavy lifting
- ▶ `watch`: where to download orig tarball
- ▶ `patches`: directory containing any patches (uses quilt)
- ▶ Several other files to tell `debhelper` where to install things, etc.