Observables codes: written by Richard, Jesper, and Sebastien
written in C
2 codes: line-of-sight observables, and IQUV-averaging

**Lev0**

remove overscans
apply flat-field
run limb finder
detect cosmic-ray hits and bad pixels

**Lev1**

Apply line-of-sight observables code
Apply IQUV-averaging observables code

**Lev1.5**

Data series:

l.o.s: hmi.V_45s, hmi.M_45s, hmi.Ic_45s
+ hmi.Lw_45s, and hmi.Ld_45s (?)

IQUV: hmi.S_720s

## Line-of-sight observables code:

**User provides:**

- time range
- target wavelength
- camera to use
- cadence of the observables sequence

The code opens all lev1 records within the time range + extra records outside of the range

it calculates the lev1.5 slotted time (target time) closest to the beginning of the time range requested

It locates the lev1 filtergram with the proper wavelength and closest to the target time: used to determine which observables sequence was run (number of frames, order of the wavelengths and polarizations, need or not to combine cameras, cadence, type of observables that can be produced...)

## Line-of-sight observables code:

it reads 72 lev1 images: 6 wavelengths x 2 polarizations x 6 times (3 before target time, 3 after target time)

It gap fills each image and returns an error array (Richard)

for each polarization and wavelength it sends 6 images to a routine (Richard) that: removes distortion, de-rotates the images, center them to the same (CRPIX1,CRPIX2) pixel and resize them to the same R_SUN
NB: CRPIX1,CRPIX2,R_SUN are chosen as median values from 72 filtergrams, and vary every 45s (=> R_SUN has a 24 hour periodicity).
NB2: solar radius is different for each wavelength, but all are resized to same R_SUN

Using the same 6 images, another routine (Jesper) temporally interpolates them to produce a filtergram at the target time

the 12 temporally interpolated filtergrams are then sent to a polarization calibration routine (Jesper) to produce true LCP and RCP (or I,Q,U, and V)

the 6 LCP and 6 RCP images are sent to a MDI-like algorithm routine (Sebastien)

the output l.o.s. observables are saved as records in different series

the code increases the target time by the cadence (45s) and starts all over again until the end of the user-provided time range

Most of the code is about error handling (keywords are missing, images are missing, images have too many bad pixels, some keywords seem wrong, image centers or radii vary too much...)

## MDI-like algorithm:

is applied separately for the 6 LCP and 6 RCP images:

computes a discrete approximation of 1st and 2nd Fourier coefficients a1,a2,b1, and b2 of the Fe I line profile based on the 6 wavelengths

assumption: solar line has Gaussian profile $I(\lambda)=I_0-I_d \exp(-(\lambda-\lambda_0)^2/\sigma^2)$

velocity = $dv/d\lambda \; T/(2\pi) \; \mathrm{atan}(b1/a1)$ T=338 mA (corresponding to +/- 8.2 km/s or 3800 G)

|B| = (vLCP-vRCP)*constant (constant based on Lande factor 2.5)

$\sigma=T/\pi \sqrt{(\log( (a1^2+b1^2)/(a2^2+b2^2) )/6)}$

$I_d = T/(2\sigma\sqrt{\pi}) \; (a1^2+b1^2)^{1/2} \; \exp(\sigma^2\pi^2/T^2)$

Because HMI filters are not Dirac functions, because only 6 filters are used to estimate the Fourier coefficients, and because the line is not a Gaussian, needs to correct velocities=> use of look-up tables (produced by another code; Sebastien)

look-up tables are 256x256; have front-window fringe pattern; are for Calmode (phases are different in Obsmode)

## IQUV-averaging observables code:

is similar to the l.o.s. code but order of loops is different (outer loop is over wavelength, not time)

instead of doing temporal interpolation, does temporal averaging (Jesper) over 12 minutes.

Produces I,Q,U, and V polarization states at the 6 wavelengths (are used by vfisv for Stokes vector inversion).
Will be modified to also produce l.o.s. magnetogram and continuum intensity at same time.