

The METIS Simulator

What is it?

- An instrument simulator?
- An observation simulator?
- A data simulator?

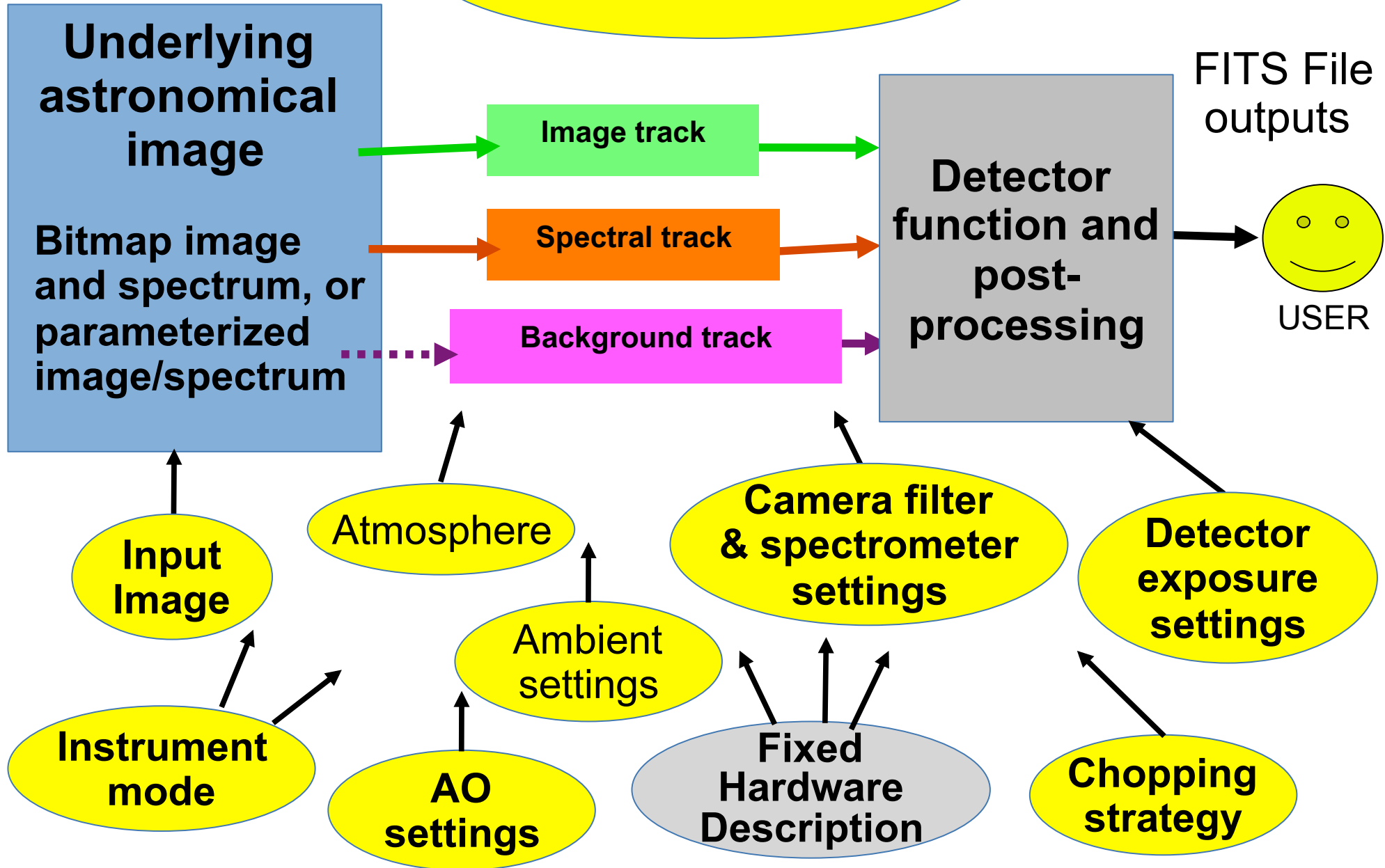
All of the above!

Outline

- 1) Overview of the METIS simulator
- 2) Job preparation and simulator execution
- 3) The simulator program and internal data flow
- 4) Image formats. Flexibility in spatial resolution, spectral resolution, "color resolution", and internal storage requirements
- 5) Output files
- 6) Simulator use in assessing METIS performance and hardware choices
- 7) Evaluation and planning of scientific observations using the METIS simulator
- 8) Generation of (fake) raw data for testing pipeline and data reduction systems
- 9) Limitations of the METIS simulator

METIS simulator

User inputs and settings



METIS simulator

Job flow

Job file
ASCII File
Accepts IDL syntax

```
[MAIN]
simulationtype = 1 ; IM camera
wavelengthprecision = .05

mayexpandplanes=1 ; default is 1
may2pass=0
DoDiffRefr = 0 ; default is not
doatmosphrput=1 ; default is to include the atmospheric
trans.

#### PSF generation. If 1, will generate a diffraction-
limited PSF using the aperture/mask files.
#### If 3, will import a PSF based on an AO simulation.
#### Specification of the actual files involved still to
be implemented, currently uses defaults.
applyPSF=1 ;
groundtemperature= 280.

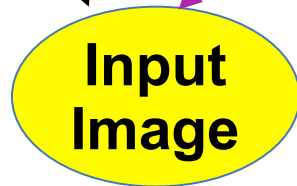
[INPUT]

wavelengthsetup = CreateWavelengthSetup(3.0, 5.6, 65000)
#### INPUT FILE
#### The name of the input FITS file(s) is specified here.
filename = "high_galaxies/
2planeHighLens_METIS_px10mas.fits"

#### BLANK IMAGE
createstar= 0 ; to create an artificial star (default is 0)

[IM DETECTOR]
; ACCEPT ALL DEFAULTS
; ron = 20 ; This is the rms readout noise per pixel in
e-, current default = 20. for HAWAII detector
; fullwell = 130000. ; This is the fullwell charge in e-,
current default = 130000. for HAWAII detector
; darkcurrent= 0.5 ; dark current in e-/sec/px, default = .
5/sec thus practically zero
; xnoise=1. ; Excess noise factor (unlikely for this sort
of detector) if entered. This is a SINGLE SETTING.
```

USER



FITS Files

Log file

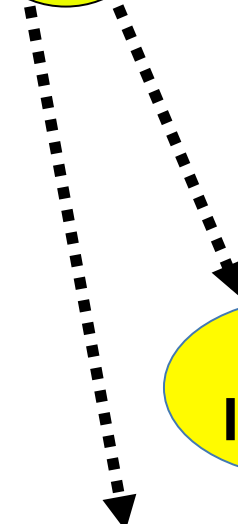
Report file or
output of
custom user
routines

Input
Image

All other
settings

Support
data files

custom user
IDL code



User control of the METIS simulator: the Job File

**The user-readable
(and writable) job file,
incorporating IDL
syntax, controls all
adjustable parameters
of the simulator and
points to any source
input (image+spectra)
files.**

Typical job file excerpt:

```
[MAIN]
simulationtype = 1 ; LM cameras

#### For PSF generation (esp from aperture/mask)
#### and differential refraction, need to specify
#### a wavelength precision, and to tell it to expandplanes
#### and/or do a 2-pass to get lambda_eff right

wavelengthprecision = .05

mayexpandplanes=1 ; default is 1

may2pass=0

DoDiffRefr = 0 ; default is not

doatmosthrupt=1 ; default is to include the atmospheric
createstar= 0 ; to create an artificial star (default is 0)

[LM_DETECTOR]

; ACCEPT ALL DEFAULTS
; ron = 20. ; This is the rms readout noise per pixel in e-,
; fullwell = 130000. ; This is the fullwell charge in e-,
; current default = 130000. for HAWAII detector
; darkcurrent= 0.5 ; dark current in e-/sec/px, default
; xsnoise=1. ; Excess noise factor
; seed = 55 ; Specific random seed
#### These are the actual exposures used during obs:

numexposures = 0 ; will compute if <=0
exposuretime = -1. ; If <=0 will compute based on background
```

User control of the METIS simulator: "Multisimulation" capability

“Multisimulations” are achieved by setting one or more controlling parameters to an **array** of values. If different parameters' arrays use different array dimensions, then a simulation will be carried out for **each combination** of parameter values spanned by the specified arrays.

Job file excerpt:

```
[INPUT]
```

```
##### Nesting level 2 with the 3 files:
```

```
filename = REFORM(FileList("/home/metissim/METIS/InputData/IFU/*.fits"), 1,3) ;
```

```
##### Nesting level 1 with the 5 brightnesses:
```

```
brightnessmultiplier = [1., 2., 4., 7., 10., 20.]
```

Main simulator code:

```
FOR msi7 = 0, multisimulationdimensions[7]-1 DO BEGIN $  
  FOR msi6 = 0, multisimulationdimensions[6]-1 DO BEGIN $  
    FOR msi5 = 0, multisimulationdimensions[5]-1 DO BEGIN $  
      FOR msi4 = 0, multisimulationdimensions[4]-1 DO BEGIN $  
        FOR msi3 = 0, multisimulationdimensions[3]-1 DO BEGIN $  
          FOR msi2 = 0, multisimulationdimensions[2]-1 DO BEGIN $  
            FOR msi1 = 0, multisimulationdimensions[1]-1 DO BEGIN $  
              FOR msi0 = 0, multisimulationdimensions[0]-1 DO BEGIN $  
; Store the indices in this array:  
multisimindices = [msi0, msi1, msi2, msi3, msi4, msi5, msi6, msi7]
```

METIS simulator

User code

Simulator Program

FITS Files

Log file

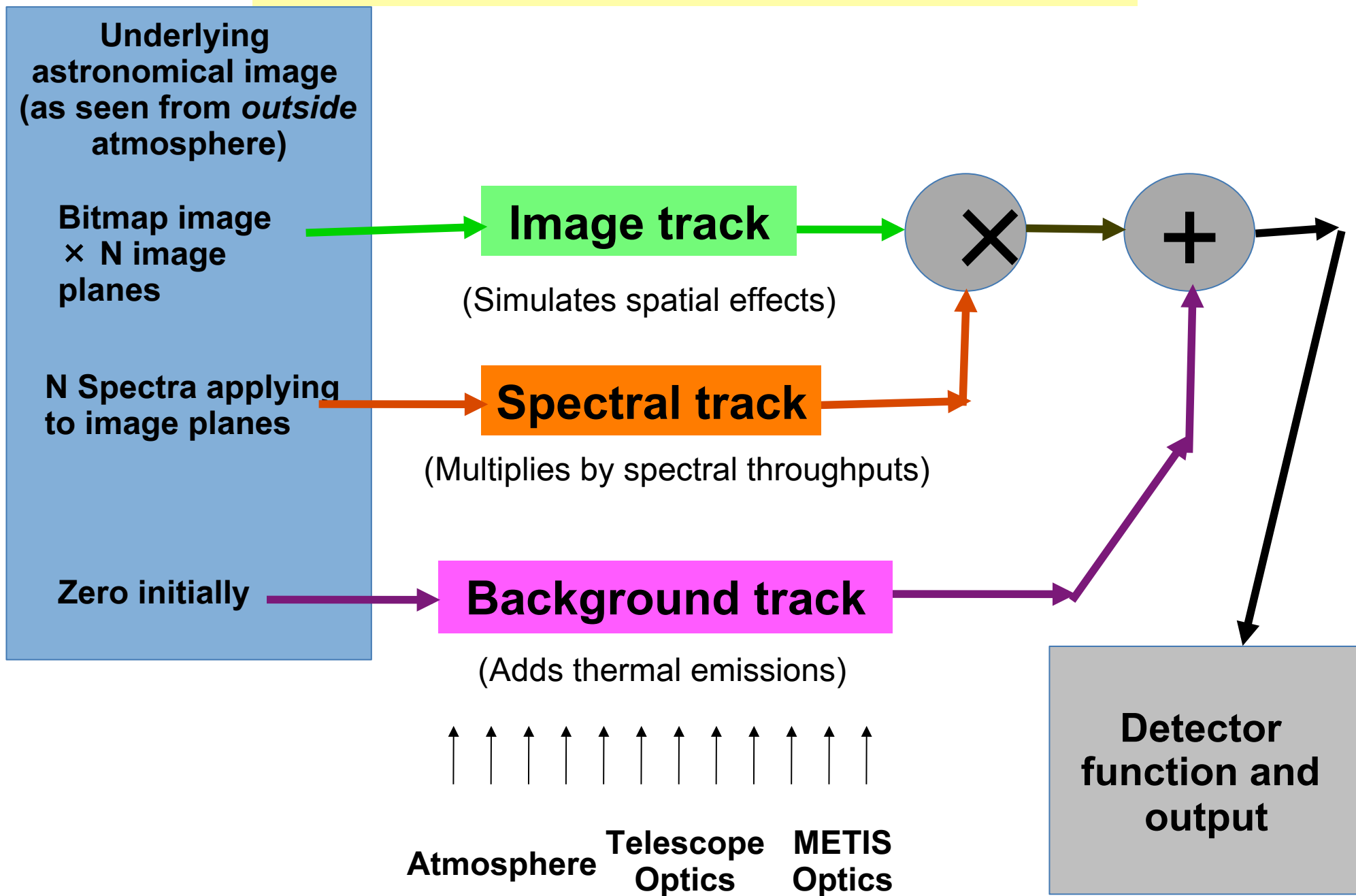
custom user IDL code

custom user IDL code

Custom functions written by user in IDL intercept execution through a number of hooks in the code, for examining and/or altering internal simulator variables.

METIS simulator conceptual diagram

Overview



METIS simulator conceptual diagram

page 1 / 4

Input

Hour angle

Atmosphere

Telescope and AO

Image track

Input Image

Rotation on sky

Differential Refraction

Apply PSF

Rotation (Altitude axis)

Spectral track

Input Spectra

Atmospheric transmission

Telescope mirrors and window transmission

Background track

Zero

and emission

and emission

Model from ESO Sky Calc

Corrections for:

Airmass

Ground temperature

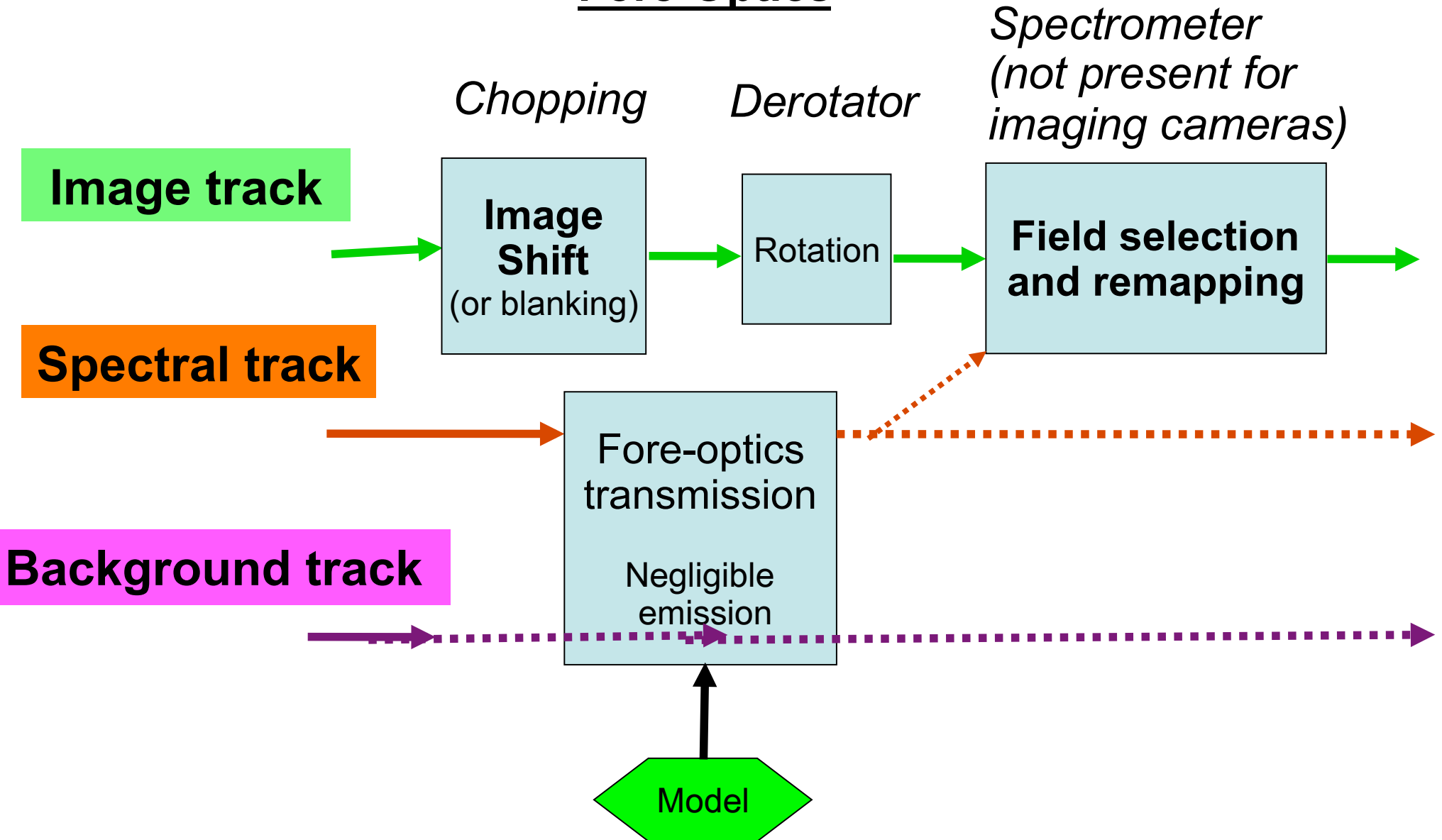
Atmospheric Model

Telescope mirror model

Ground temperature

METIS simulator conceptual diagram
page 2 / 4

Fore-Optics



METIS simulator conceptual diagram
page 3 / 4

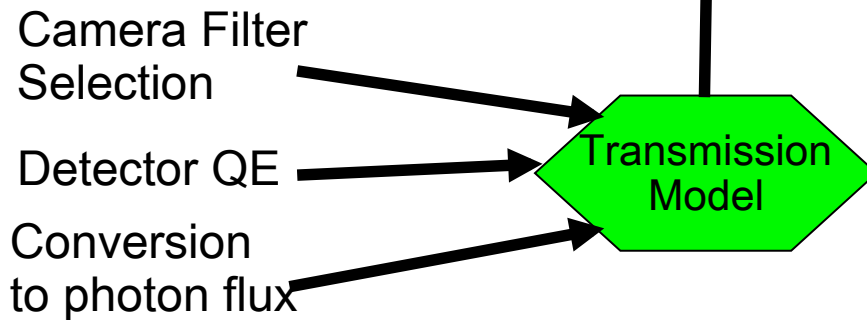
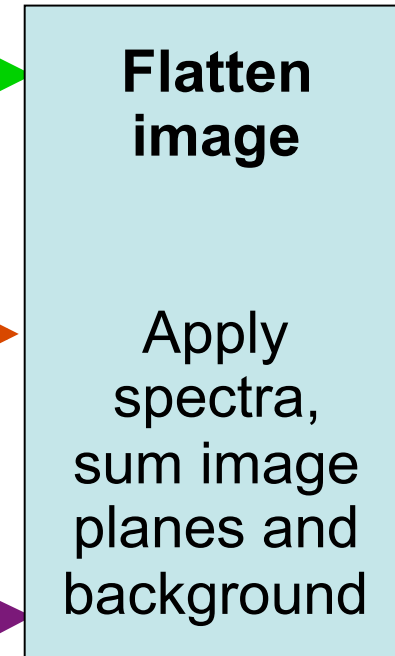
Camera

Generate photocurrent image

Image track

Spectral track

Background track

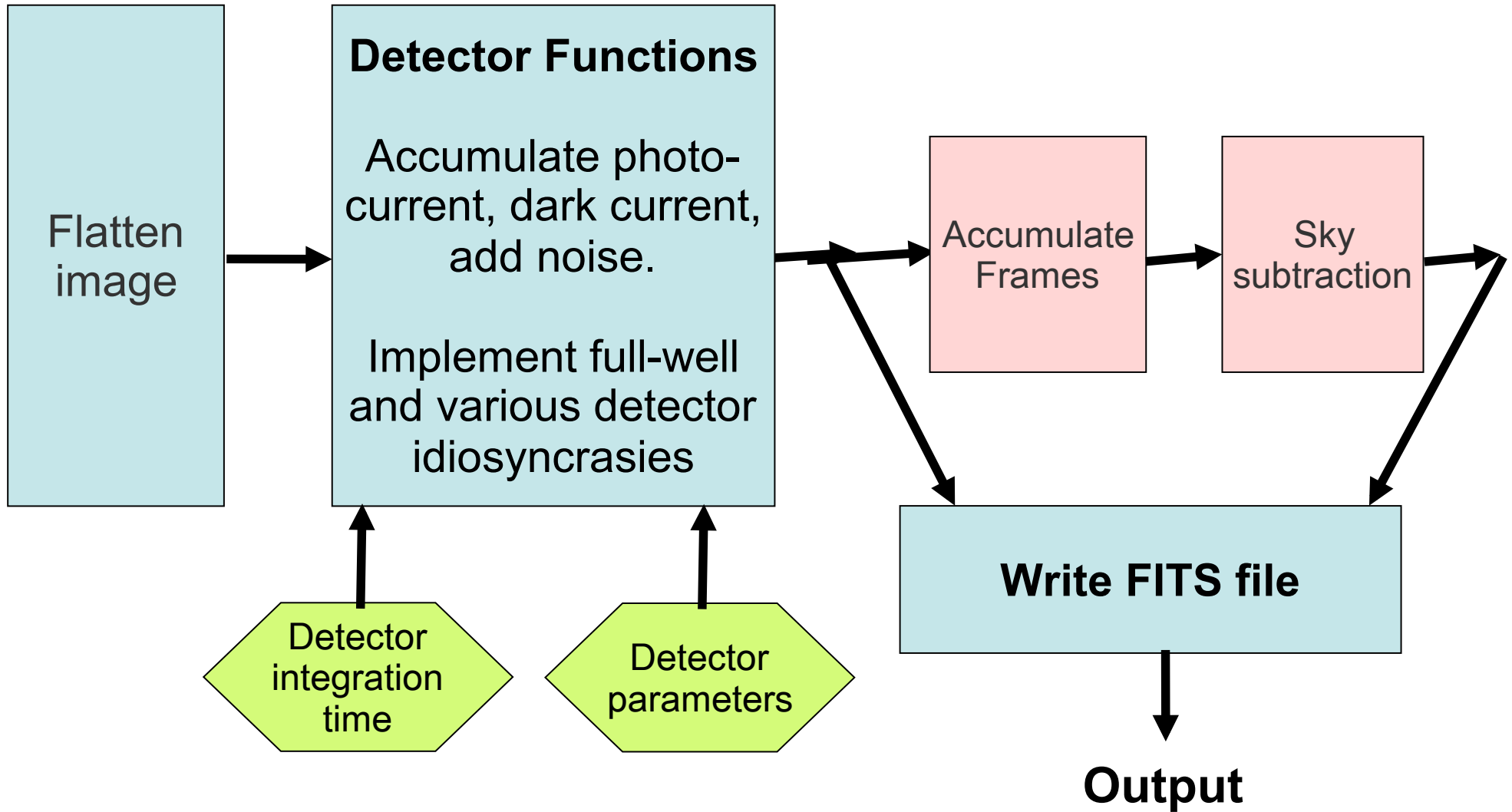


METIS simulator conceptual diagram

page 4 / 4

Detector

*Post-processing
(optional)*



Order of operations as actually performed by simulator program

ATMOSPHERE, TELESCOPE, FORE-OPTICS:

- Input: Source image on the celestial sphere
- Rotation of image according to the telescope orientation
- Image affected by differential refraction (atmospheric dispersion)
- Image convolved by the PSF
- Image rotated by the telescope geometry
- Image shifted by the chopping mirror
- Image rotated by the derotator
- Spectral brightness of image modified by inclusion of all throughput elements
- Inclusion of background radiation produced by those elements

CAMERAS:

- Spectral brightness further modified by spectral filter, conversion to photon flux, detector quantum efficiency, etc.
- Image formed in terms of photon flux, integrating the image raster, spectra, and background
- Image resampled onto detector pixel geometry
- Image data created as read out from one detector exposure, adding random noise and applying other detector characteristics
- (Optional) Summation of multiple detector frames to obtain long exposure

Order of operations as actually performed by simulator program

SPECTROGRAPHS:

Spectral brightness further modified by spectral efficiency function, conversion to photon flux, detector quantum efficiency, etc

IFU Spectrograph: Spatial reconfiguration of image (image slicing and stacking)

Slit Spectrographs: Masking of image by specified slit

Expansion of image and spectral information to an image structure in "imagecube" format

Inclusion of thermal background into the imagecube

Application of spectrographic dispersion

Convolution with the spectral resolution function

Binning (downsampling) to match the detector geometry if oversampling used

Image data created as read out from one detector exposure, adding random noise and applying other detector characteristics

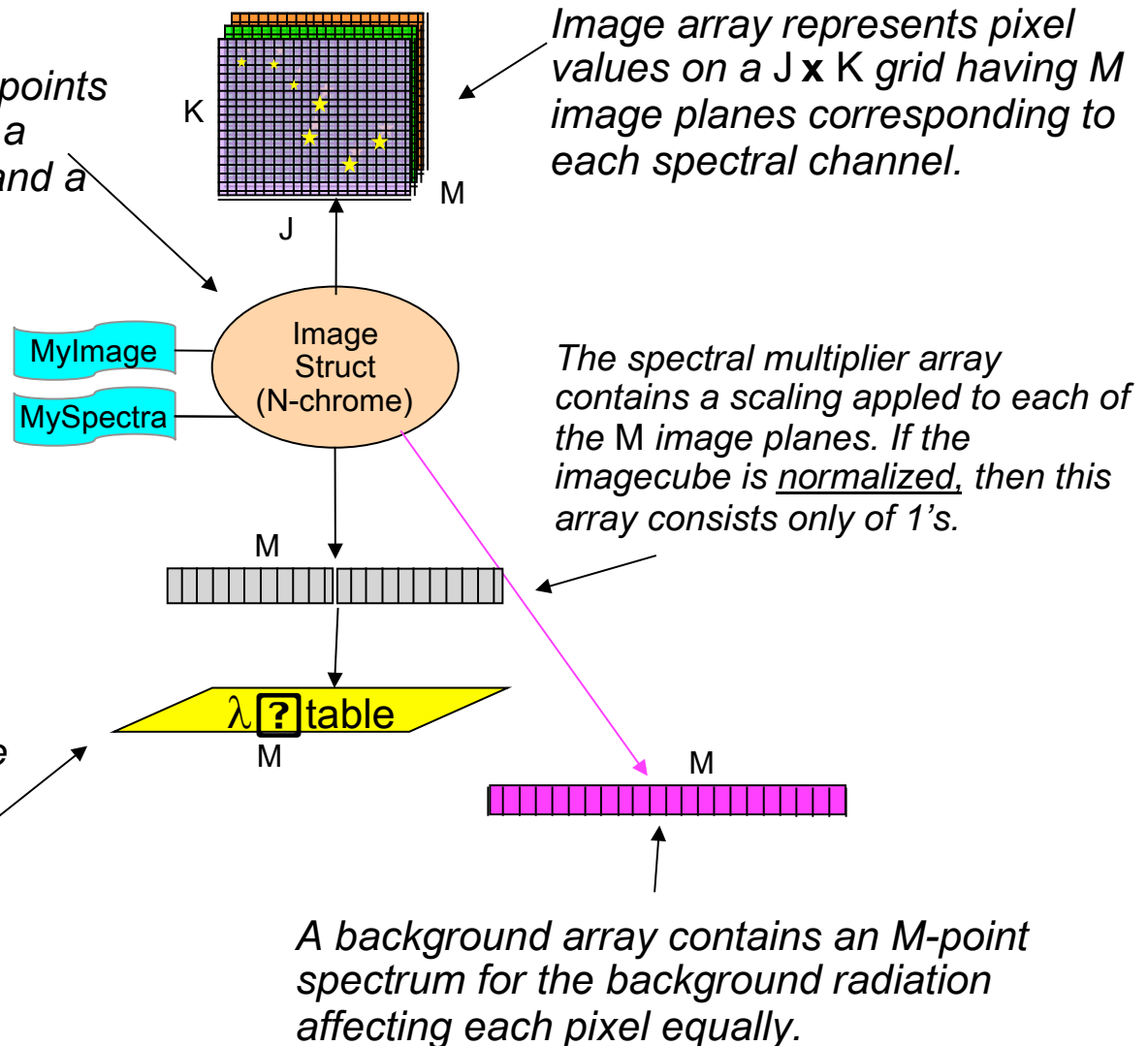
(Optional) Summation of multiple detector frames to obtain long exposure

Image representation in Imagecube mode

Image struct. in Imagecube mode points to an image array ($J \times K \times M$), and a spectral scaling array (length M), and a background array (length M)

*The two “ownership” flags are set when the image array or the spectra and background arrays “belong” to this particular image struct. If **not** set, then the corresponding array will **not** be deallocated by a request to destroy this image struct.*

A wavelength setup defines the center wavelengths applying to the M columns of the spectral and background arrays of (usually multiple) images. It also specifies the spectral widths of each wavelength bin. It is pointed to by “image.WavelengthSetup” but no image struct ever “owns” it.



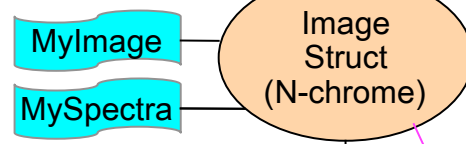
Note that Imagecube mode has very large storage requirements! Its use is generally avoided where possible.

Image representation in N-chrome mode

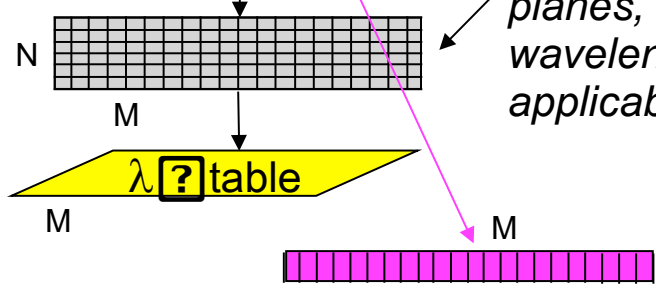
Image struct. in N-chrome mode (including monochrome mode $N=1$) points to an image array ($J \times K \times N$), and optionally a spectral array ($M \times N$), and a background array (length M)

Image array represents pixel values on a $J \times K$ grid having N image planes corresponding to the N "colors", or just one image plane in monochrome mode ($N=1$) or when no spectral information present.

The two "ownership" flags are set when the image array or the spectra and background arrays "belong" to this particular image struct. If **not** set, then the corresponding array will **not** be deallocated by a request to destroy this image struct.



The $M \times N$ spectral array contains an M -point spectrum for each of the N image planes, corresponding to wavelengths defined by the applicable wavelength setup.

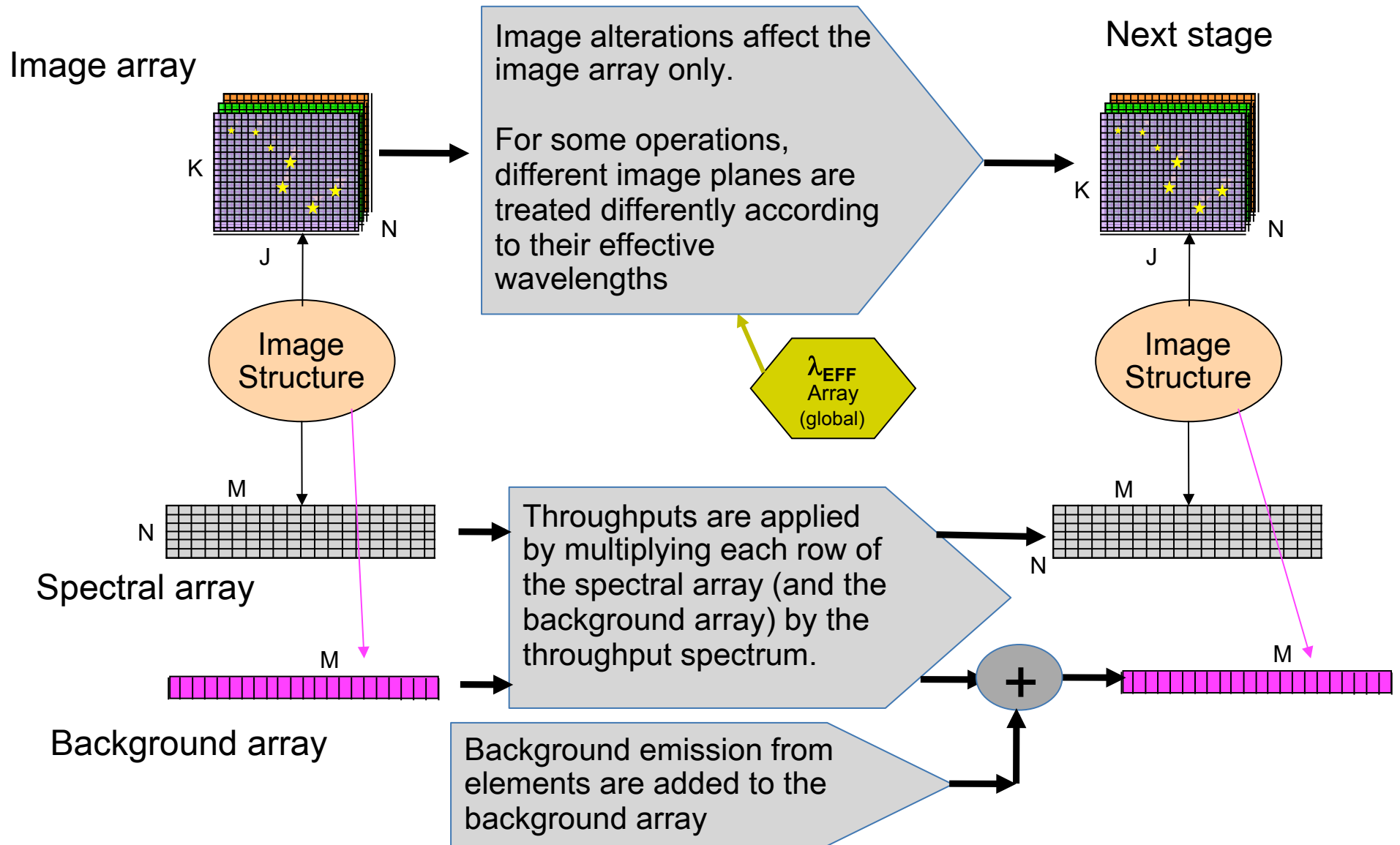


A wavelength setup defines the center wavelengths applying to the M columns of the spectral and background arrays of (usually multiple) images. It also specifies the spectral widths of each wavelength bin. It is pointed to by "image.WavelengthSetup" but no image struct ever "owns" it.

A background array contains an M -point spectrum for the background radiation affecting an entire image (each pixel equally).

N-chrome mode has greatly reduced storage requirements. High spatial resolution is possible even with high spectral resolution as long as the "color resolution" (N) isn't too large.

Processing image structures from stage to stage



Up/down-grading of spatial or spectral resolution for improving accuracy or computational efficiency

- Input image at high resolution (better than detector resolution) when imaging resolution is at issue
- Define wavelength setup (M points) with high resolution for careful accounting of atmospheric lines or for use of IFU spectrograph: little cost to storage or execution speed *except* in imagecube mode.
- Input image only in N colors ($N \ll M$) to describe most models.
- Promotion of color resolution (N) to accurately process wavelength-dependent imaging processes (e.g. PSF, differential refraction) according to the λ_{EFF} array
- Reduction of image extent to that of the detector, or especially to the IFU field of view in spectrographic mode. (Or to spectrographic slit)
- Alter pixel resolution to match detector format from previous stages
- Promote (cropped) image to imagecube for processing through high resolution spectrograph, while limiting spectral range to that of spectrograph.
- Interpolation or smoothing of spectral functions (including atmosphere) to the M-point wavelength setup before applying

And Finally:

- Flattening of N-chrome image or imagecube to compute net photocurrent (no spectrum!) seen by detector.

Output from simulator

- Accumulated frames (for each chopping position)
- Accumulated frames, background subtracted
- Individual frames (not currently supported)
- Raw IFU spectrometer accumulated image (for each chopping position, or background-subtracted)
- Reconstructed IFU spectrometer imagecube

Outputs are FITS images (normally 32 bit floating point)
thus with file sizes:

- L/M band images: 16 MB
- N/Q band images: 4 MB
- IFU spectrometer, raw or reconstructed: 64 MB

(*Internal* representations are generally 64-bit double precision at stages where image values include background pedestals)

Output from simulator: volume of data

- Outputs are normally 32 bit floating point, full detector frames.
- Should be an option for outputting double-precision arrays (especially for external subtraction of sky from on-source frames)

Possible means to reduce data storage requirements:

- Would be possible to represent in 16 bit “block floating point,” also with common offset for images including background pedestals (as supported using standard FITS header lines).
- Could also just present cropped detector frames to reduce data size.
- Simulator–pipeline testing possibly using streams between processes (or machines) to avoid large storage requirements.

Technical simulations and sensitivity determinations

Simulator parameters – 1/2

Input image as:

- Point source (simulating a star); *or*:
- Extended source (over large area, usually full FOV)
- + Assumed spectrum, usually white or blackbody (except for spectrometer)

Detector/exposure settings:

Total observation time: Often assume 1 hour (can then scale SNR)

Environmental parameters:

- Atmospheric transmission/emission model (from ESO sky calculator)
- Ground temperature (affects emission from warm optics & atmosphere)
- Atmospheric water vapor column (pwv)
- Telescope altitude and barometric pressure (affect airmass)
- Ambient humidity (affects differential refraction in N band)
- Seeing (for AO model selection)

Simulator parameters – 2/2

AO parameters (when using AO models):

Guide star magnitude and offset angle, seeing and altitude, (+ wavelength)

Telescope parameters:

- **Masked aperture** (for diffraction-limited PSF generation)
- Telescope area (within mask)
- Net telescope emissivity within mask (specification = 15%)

METIS optics parameters:

- **Transmission curves for:**
 - Transmissive optics (entrance window, dichroics)
 - Reflective optics (T ~ 99%)
 - Camera filter used
 - Detector quantum efficiency
- **Optics' wavefront errors** (currently assumed perfect)

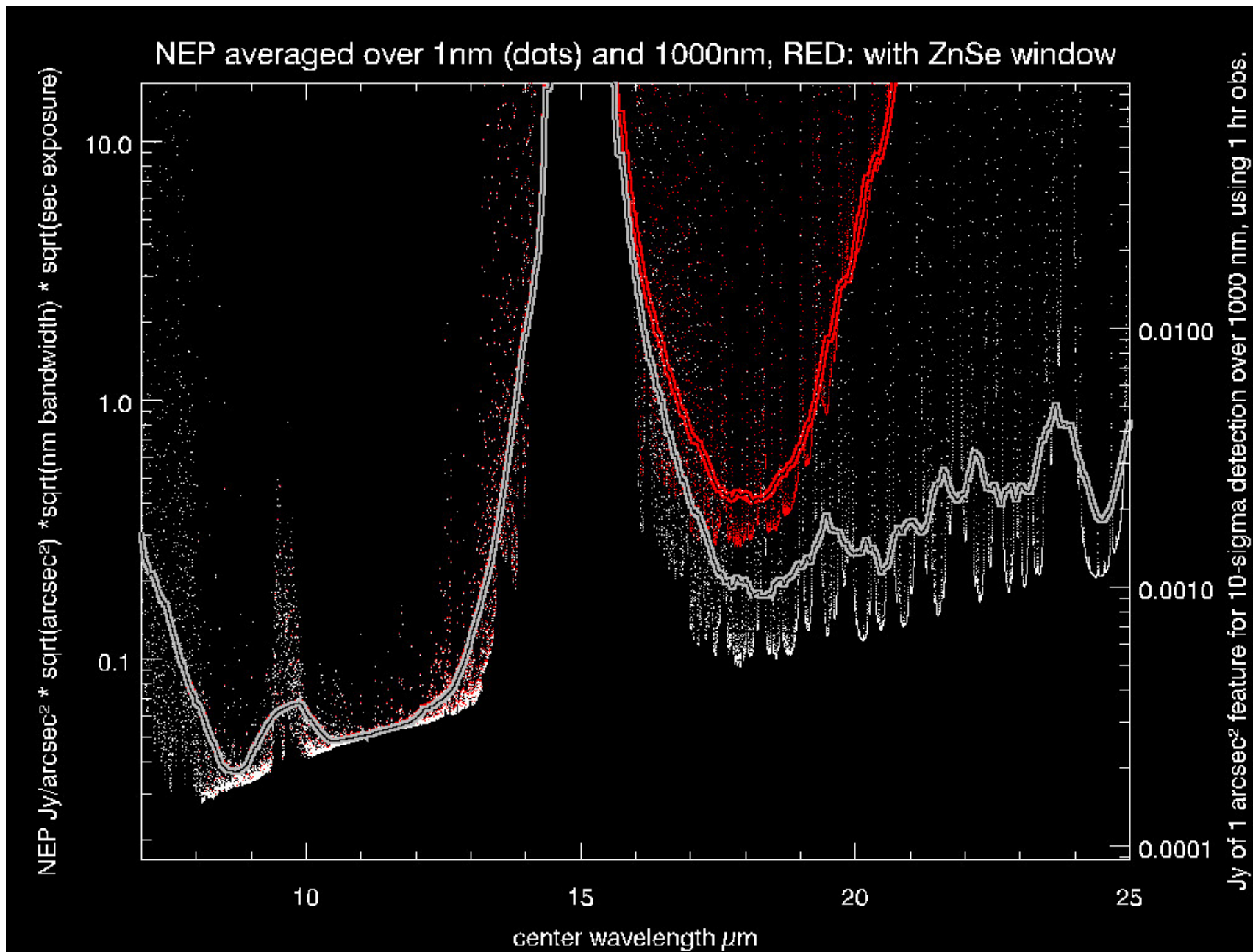
Simulator parameters – 3/3

Detector characteristics:

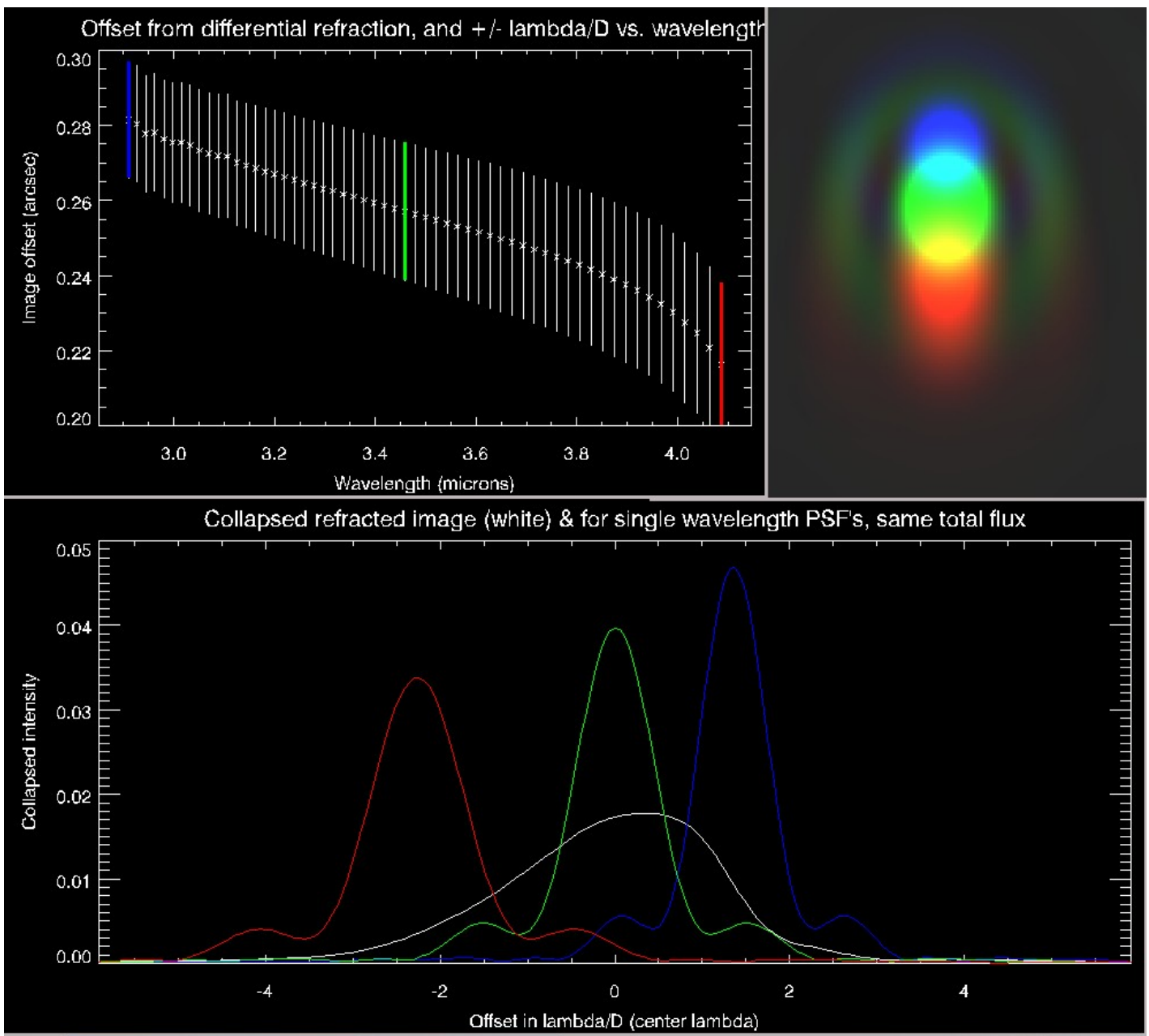
- Readout noise
 - Dark current
 - Full well saturation (affects maximum exposure time)
 - Non-linearity, stochastic gain fluctuations, excess noise factor, **blooming, other row/column/group artifacts** (not currently considered)
-

- Many of these are fixed and completely beyond our control. BUT:
- Some are beyond our control but are variable (weather, zenith angle) so we can evaluate sensitivity as a function of anticipated variations.
- Some are selected by users for reasons dictated by their specific observational programs (wavelength range, observing time allotment)
- Some vary according to which specific object within an observing program is selected (available AO guide stars, range of zenith angle)
- Some are subject to design decisions not yet finalized (a shrinking list)
- In *all* of the latter cases, knowledge of performance variations can help inform design engineers, science users, and data processing programmers

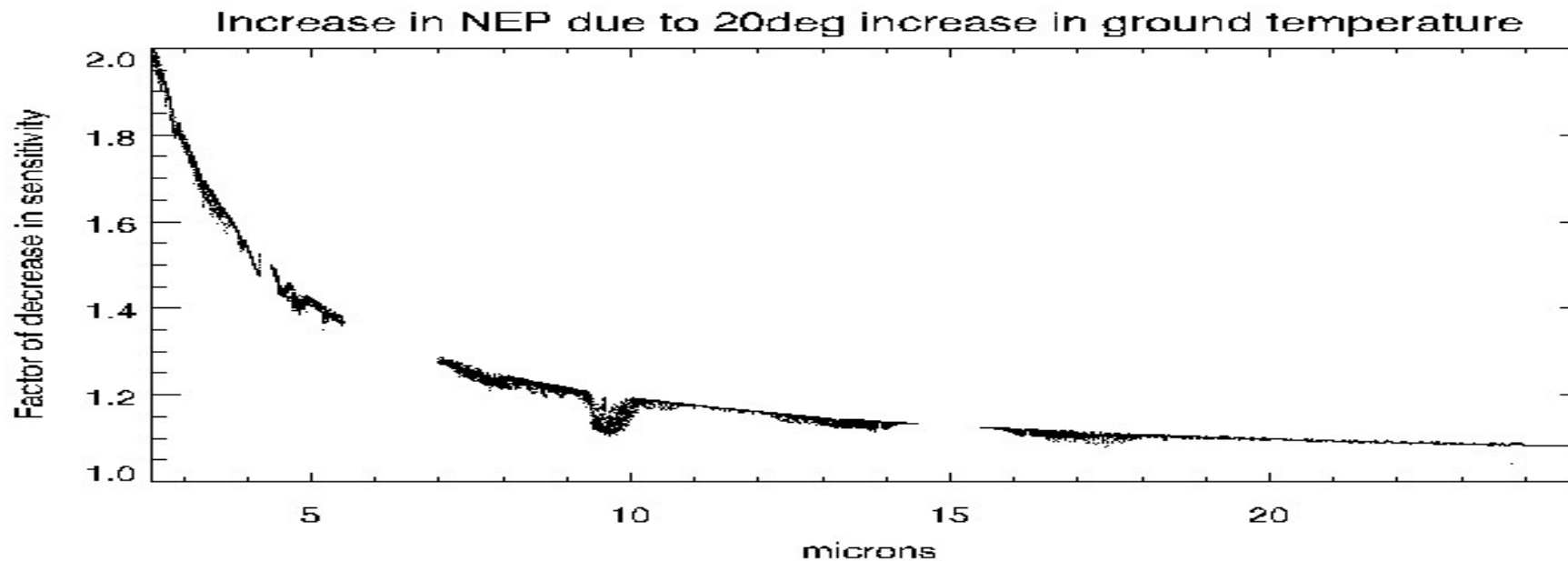
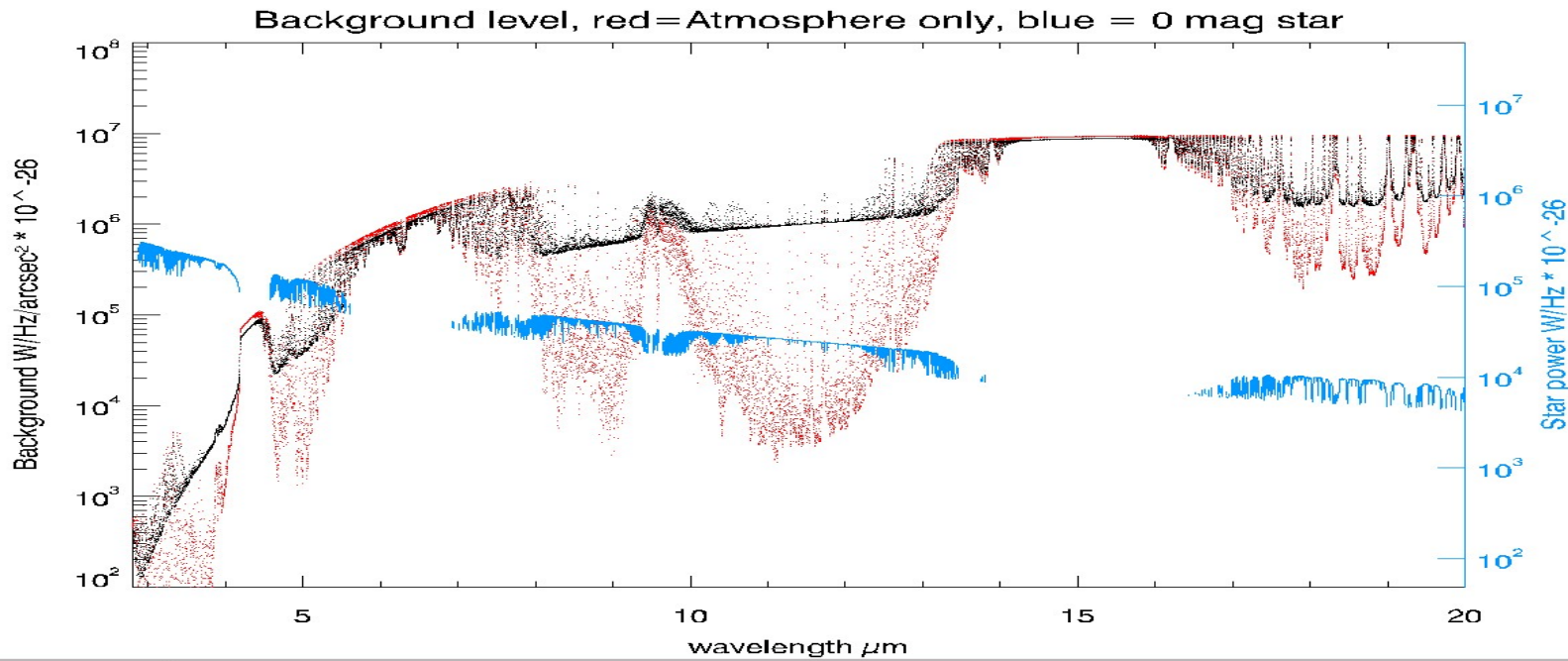
Example: Sensitivity reduction in Q band due to inclusion of ZnSe window



Example: Using the simulator to compute differential refraction (due to atmosphere) in L band



Example: calculation of NEP from L to Q bands, and increase factor due to 20 deg. increase in ground temperature



Evaluation and planning of scientific observations

Inputs to simulator – 1/2

Input image as:

- Expected image on sky (monochrome) + spectrum
- Image on sky in N colors + N spectra (“N-chrome mode”)
- Image on sky, spectra for each pixel (“Imagecube mode”)
- Parametrically defined point source or extended source + spectrum
Spectrum can also be specified as white or a black-body temperature

Environmental parameters:

- Ground temperature (emission from warm optics & atmosphere)
- Atmospheric water vapor column (pwv) and pressure (minor)
- Altitude (affects airmass of atmosphere)
- Seeing (for AO model selection)

AO parameters:

Guide star magnitude and offset angle, seeing and altitude, (+ wavelength)

Evaluation and planning of scientific observations

Inputs to simulator – 2/2

Instrumental parameters:

(Usually defaults used to reflect expected hardware)

Chopping mode, background subtraction strategy

Instrument mode and filter selection:

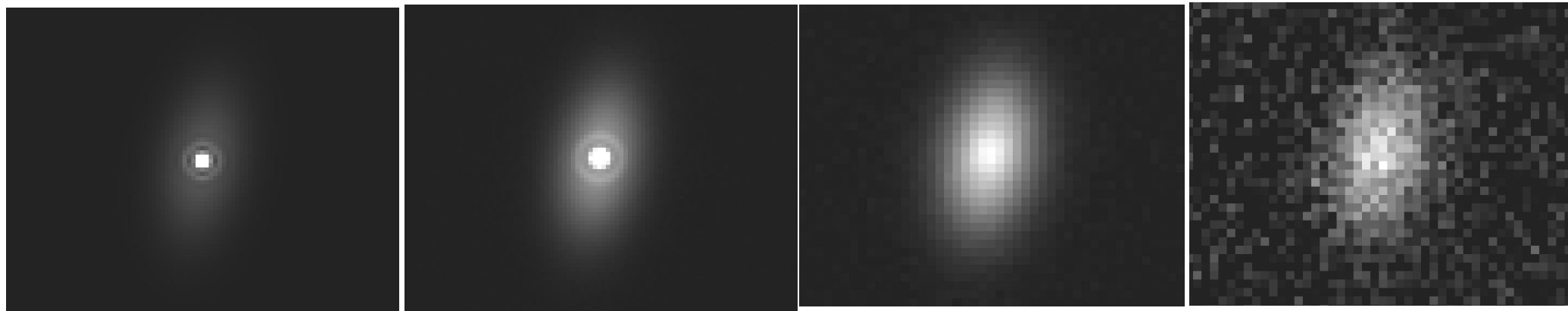
- L/M band camera or slit spectrometer
- N/Q band camera or slit spectrometer
- IFU (high resolution) spectrometer
- Select pre-defined filter bandpass for each camera

Detector/exposure settings:

- Total observation time (most important, 1 hour is reasonable)
- Detector integration time
- Number of frames
- Detector chip characteristics (Usually defaults used)

Example: Input image with two color components (N=2) in N-chrome mode:
1) An unresolved star, very detectable at short wavelengths; and
2) A cool nebulous component with most of its emission at longer wavelengths

Here is what the simulator outputted, observing the same image at L, M, N, and Q bands:



Producing fake data for exercising the METIS pipeline and data reduction software

- This would especially involve the backend of the simulator
- Verifying acceptance of output detector data frames as generated by the simulator
- Testing routines involved with background subtraction from simulated chopping/nodding (+ drift scanning, dithering, etc.)
- Evaluate robustness or sensitivity (depending on purpose) of data reduction system to random or arbitrary effects simulated at *any* point in the observing chain (atmosphere → detector, inclusive)

This all remains for the future, inasmuch as the pipeline development has not (or has barely?) begun, and no data interfaces from the instrument (or simulator) have been established. So.....

Limitations of the METIS simulator:

- The simulator does *not* implement physical (wave) optics. It is based solely on incoherent optical concepts (such as considered by ray-tracing and radiometry) connecting one image plane (or usually a *virtual* image plane) with the next one using a rule defined in terms of *intensity* only (not wave amplitude).
- One *apparent* exception is the simulator's implementation of diffraction by the (masked) ELT aperture. However this again is accomplished through convolution of the preceding image intensity by an intensity PSF. That PSF in intensity is obtained in a side calculation (which *is* based on physical optics), but the wave nature of the light in the image is never directly considered.
- For that reason, coherent effects involving interaction between two physical optics effects in the optical chain *cannot* be properly solved. This is not an actual limitation in normal cases because all optical elements (besides the telescope aperture) are accurately described using incoherent optics as long as the imaging stages are in focus.
- However it does mean that coronagraphic masks in the pupil plane (or phase masks in the image plane, I believe) cannot be integrated into the algorithm unless that element incorporates telescope diffraction, removing the diffraction calculation from the simulator.