La Sapienza University of Rome

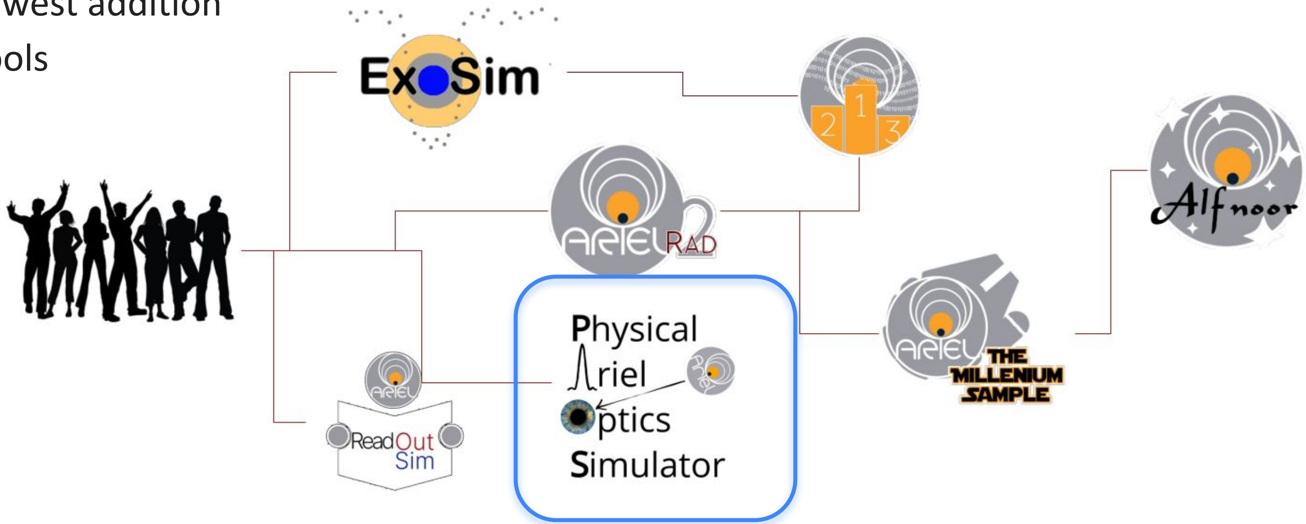
Predicting the optical performance of the Ariel Telescope using PAOS

Andrea Bocchieri

Sapienza Università di Roma
(andrea.bocchieri@uniroma1.it)

A new Ariel simulator

PAOS, the *Physical Ariel Optics Simulator*, is the newest addition to Ariel's suite of tools



The idea behind PAOS

- Ariel measurements do not require high imaging quality
- Photons must be collected in a sufficiently compact region of the focal plane

Need to ensure that

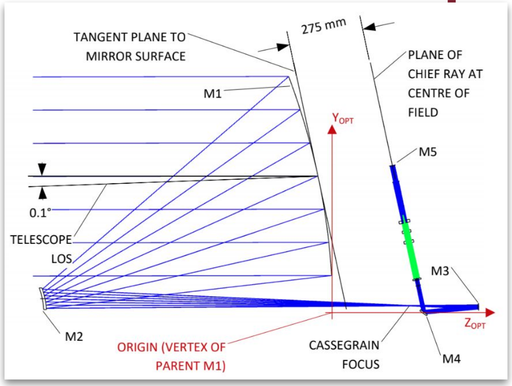
1. the instrument design is not above specification
2. Ariel still delivers high-quality data for scientific analysis even at wl where it is not diffraction limited

→ Need a tool to assess the impact of diffraction & aberrations on the Ariel optical performance: **PAOS**

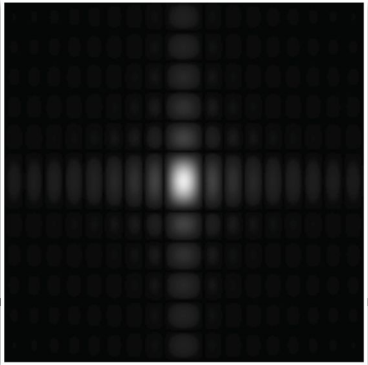
Physical
Ariel
Optics
Simulator

Optical propagation

- **Ray-tracing** estimates the path of the individual, imaginary lines which represent normals to the wavefront (the surfaces of constant phase)
  - design phase
  - determine e.g. magnification, aberrations, and vignetting



- **Physical optics propagation (POP)** estimates the changes in the wfo as it travels through the optical components.
  - predict effects of diffraction
  - aberrations must be input separately



Propagation codes

- Code V, Zemax OpticStudio
  - ray-tracing programs, also offer POP calculations
  - not easily customizable (Monte Carlo simulations)
  - not free, publicly available

- LightPipes, POPPY, PROPER
  - technical limitations
    - POPPY only supports image and pupil planes
    - PROPER not suitable for off-axis propagation
    - None supports refractive elements

PAOS – An overview

- PAOS, the *Physical Ariel Optics Simulator*, is an end-to-end Physical Optics Propagation (POP) model of the Ariel telescope and subsystems.

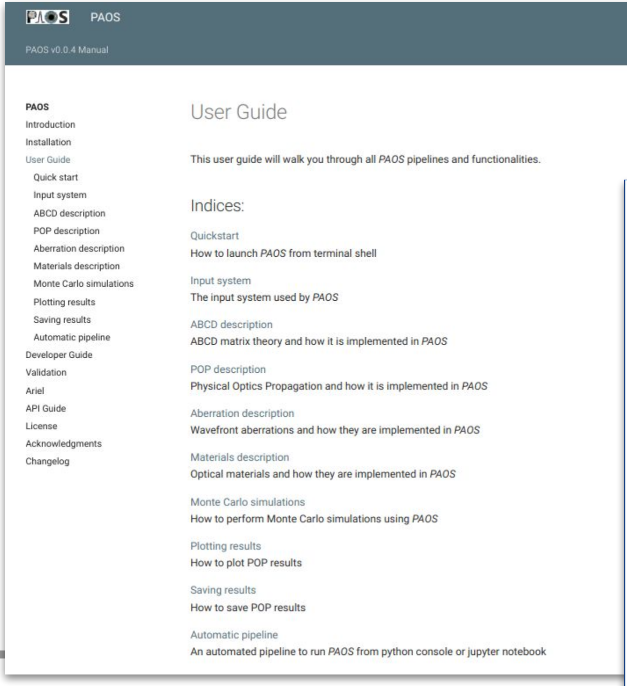
  - ✓ Implements Fresnel diffraction in the near and far fields
  - ✓ Simulates the propagation of the wfo through the Ariel optical chain
  - ✓ Delivers the PSFs vs. lambda at the intermediate and focal planes
  - ✓ Supports refractive elements
  - ✓ Implements geometric aberrations
  - ✓ Has a generic input system and built-in GUI

PAOS – Code & docs

- PAOS is written with a full Python 3 stack and comes with
    - an installer
    - documented examples
    - an exhaustive guide

```
$ cd path/to/PAOS
$ pip install .
$ paos -h

# see all available
options
```

html docs


PAOS
Search
Git repo
PAOS v0.0.4 Manual

PAOS
Introduction
Installation
User Guide
Quick start
Input system
ABCD description
POP description
Aberration description
Materials description
Monte Carlo simulations
Plotting results
Saving results
Automatic pipeline
Developer Guide
Validation
Ariel
API Guide
License
Acknowledgments
Changelog

User Guide

This user guide will walk you through all *PAOS* pipelines and functionalities.

Indices:

Quickstart
How to launch *PAOS* from terminal shell

Input system
The input system used by *PAOS*

ABCD description
ABCD matrix theory and how it is implemented in *PAOS*

POP description
Physical Optics Propagation and how it is implemented in *PAOS*

Aberration description
Wavefront aberrations and how they are implemented in *PAOS*

Materials description
Optical materials and how they are implemented in *PAOS*

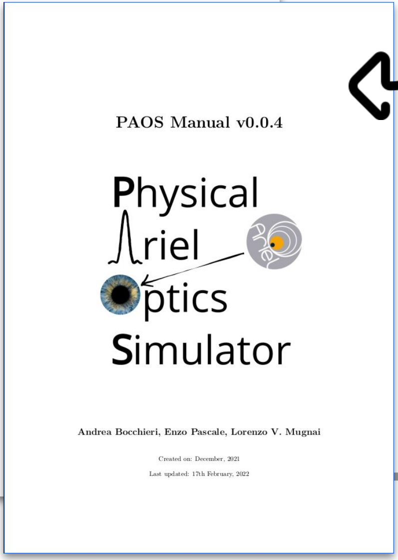
Monte Carlo simulations
How to perform Monte Carlo simulations using *PAOS*

Plotting results
How to plot POP results

Saving results
How to save POP results

Automatic pipeline
An automated pipeline to run *PAOS* from python console or jupyter notebook

pdf docs


PAOS Manual v0.0.4

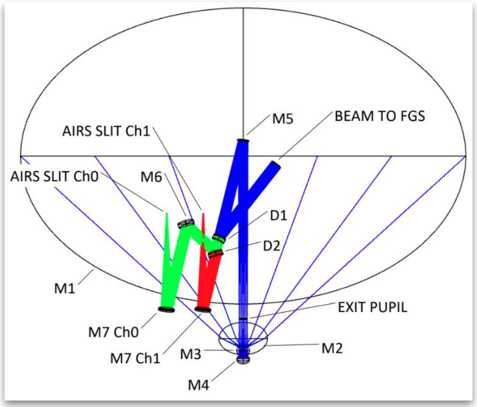
Andrea Bocchieri, Enzo Pascale, Lorenzo V. Mugnai

Created on: December, 2021
Last updated: 17th February, 2022

PAOS – Implementation (1)

PAOS is an ETE POP model of the Ariel telescope and subsystems…

- Fresnel diffraction in the near and far fields
  - the Paraxial theory described by G. N. Lawrence in *“Applied Optics and Optical Engineering Vol XI”*
- Paraxial ray-tracing to estimate the projections of the physical apertures to propagate the wf for an off-axis optical system



PAOS – Code example (1)

- Paraxial theory handled by the Python class ABCD
    - Supports thin lenses, diopters, medium changes, thick lenses, magnifications, and prisms

Diopter 1

Thin lens

Diopter 2

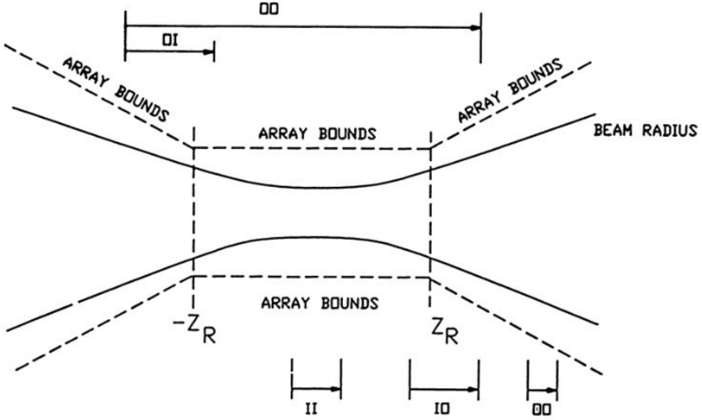
```
import numpy as np
from paos.classes.abcd import ABCD

radius1, radius2 = np.inf, -20.0  # mm
n_os, n_l, n_is = 1.0, 1.25, 1.0
center_thickness = 5.0
abcd = ABCD(curvature = 1.0/radius1, n1 = n_os, n2 = n_l)
abcd = ABCD(thickness = center_thickness) * abcd
abcd = ABCD(curvature = 1.0/radius2, n1 = n_l, n2 = n_is) * abcd
print(abcd.ABCD)
```

```
[[ 1.       4.    ]
 [-0.0125  0.95   ]]
```

Thick lens

PAOS – Implementation (2)

- Pilot beam to automatize the choice of algorithm to propagate the wfo in near-field and far-field conditions



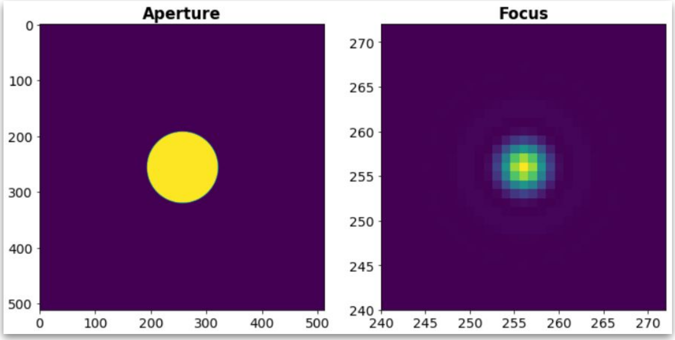
pilot beam: an analytically-traced on-axis Gaussian beam

- Three primitive operators to move from any point in space to any other (II, IO, OI, OO):
  - plane-to-plane
  - waist-to-spherical
  - spherical-to-waist

PAOS – Code example (2)

- POP handled by the Python class WFO
  - Implements Gaussian beams, wfo propagation, paraxial phase correction, apertures, and stops

```
from paos.classes.wfo import WFO
beam_diameter, wavelength, grid_size, zoom = 1.0, 3.0e-6, 512, 4
fl, thickness, xc, yc = 1.0, 1.0, 0.0, 0.0

wfo = paos.WFO(beam_diameter, wavelength, grid_size, zoom)
wfo.aperture(xc=xc, yc=yc, r=beam_diameter/2, shape='circular')
wfo.make_stop()
wfo.lens(lens_fl=fl)
wfo.propagate(dz=thickness)
```

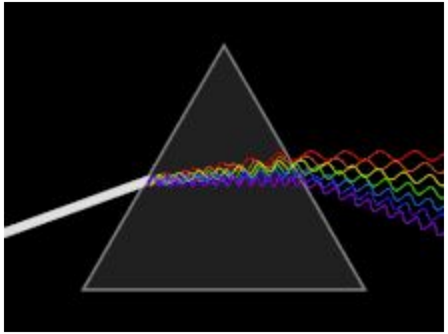


PAOS – Implementation (3)

- Propagation through refractive media using the Sellmeier 1 equation
  - provides index of refraction relative to air for a glass at ref. T, P
- Can estimate of refractive index of an optical material at any given P, T

- Supported materials:

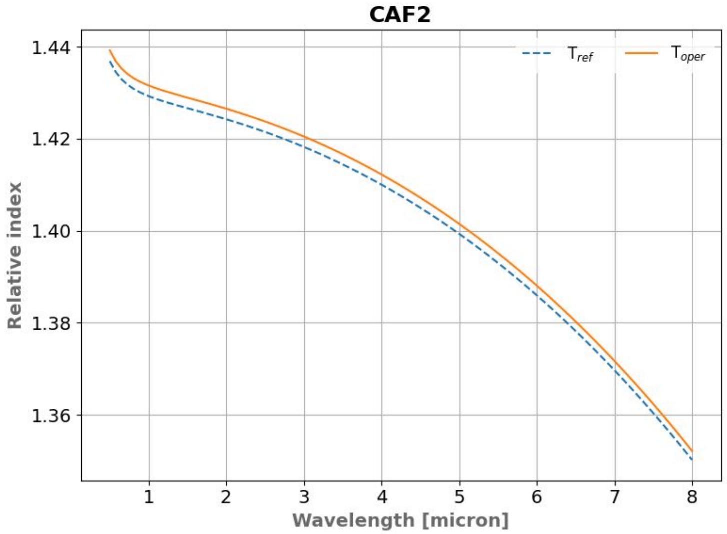
  CAF2, SAPPHIRE, ZNSE, BK7, SF6, SF11, BAF2 (list can grow as needed)



PAOS – Code example (3)

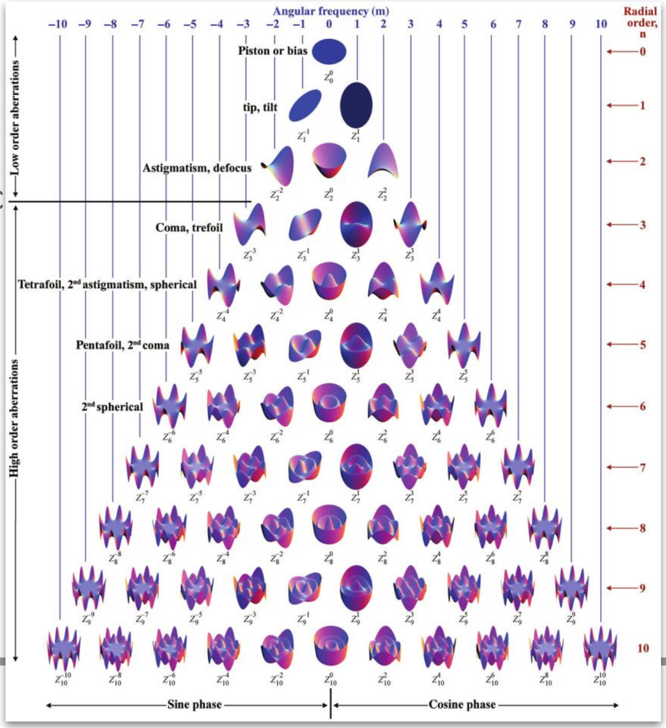
- Refractive media handled by class Material
  - Implements Sellmeier 1 equation, P and T-dependent refractive index, and plotting routines

```python
import numpy as np
from paos.util.material import Material

wl = np.linspace(0.5, 8.0, 100)
mat = Material(wl=wl)
mat.plot_relative_index(material_list=[‘BK7’])
```



PAOS – Implementation (4)

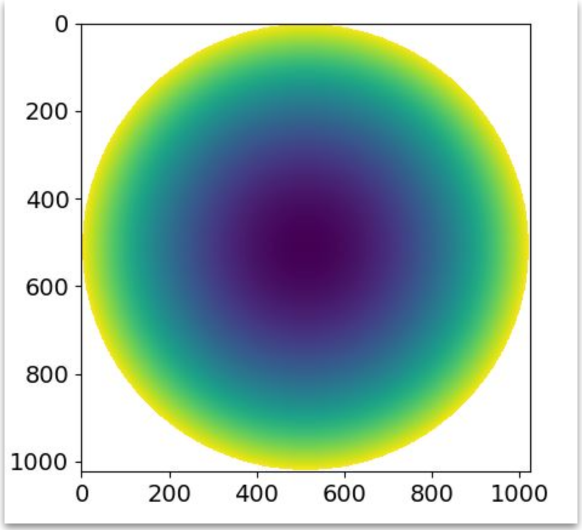
- Optical aberration using a series of Zernike polynomials, up to a specified radial order
  - Laksminarayan & Fleck, Journal of Modern Optic (2011)
- Ordering can be ANSI, Noll, Fringe, or Standard
  - Born and Wolf, Principles of Optics, (1999)
- Both ortho-normal and orthogonal polynomials
  - Supports circular and elliptical pupil geometry



PAOS – Code example (4)

- (Geometric) aberrations handled by class Zernike
    - Implements Zernike polynomials, orthogonal or orthonormal

```
import numpy as np
from paos.classes.zernike import Zernike

x = np.linspace(-1.0, 1.0, 1024)
xx, yy = np.meshgrid(x, x)
rho = np.sqrt(xx**2 + yy**2)
phi = np.arctan2(yy, xx)
zernike = Zernike(36, rho, phi, ordering='noll', normalize=True)
defocus = zernike(3)
```



PAOS – Input system

- Generic input system to be used by anyone expert in optical CAD

- Primary inputs are in .ini configuration files describing the simulation parameters and the optical surfaces

```
[general]
project = Template
grid_size = 512
zoom = 4
lens_unit = m
tambient = 20.0
pambient = 1.0

[wavelengths]
w1 = 3.0

[fields]
f1 = 0.0,0.0
```

General simulation parameters

```
[lens_01]
surfacetype = INIT
comment = input beam init
save = False
ignore = False
stop = False
aperture = elliptical aperture,0.25,0.25,0.0,0.0

[lens_02]
surfacetype = Standard
comment = M1
radius = 2.0
thickness = 1.0
material = MIRROR
save = True
ignore = False
stop = True
aperture = elliptical aperture,0.25,0.25,0.0,0.0

[lens_03]
surfacetype = Standard
comment = IMAGE_PLANE
save = True
ignore = False
stop = False
```

Sequential optics chain

PAOS – GUI

- Graphical User Interface (GUI) built using the Python package PySimpleGui
  - bridge the “gap” between software developers and end-users
- User can
  - write/edit configuration files
  - launch POP simulations
  - launch MC simulations

```
$ paosgui -h

$ paosgui -c /path/to/input/file.ini
```

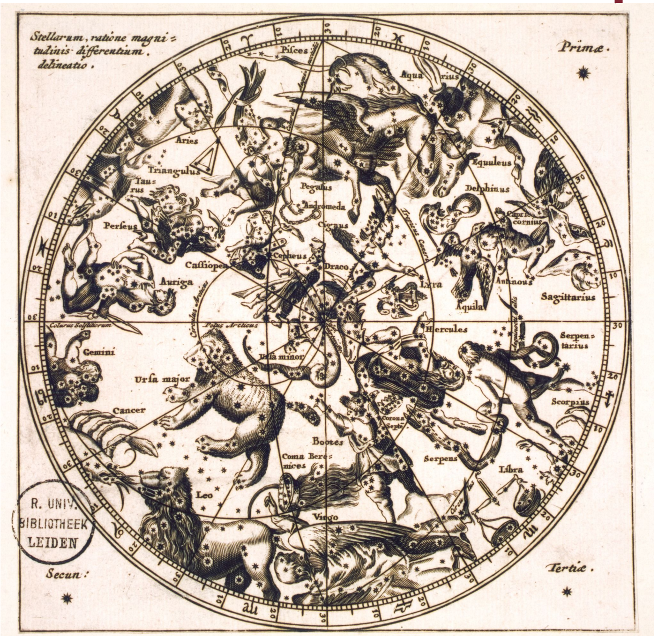
PAOS – Mapping the key points

General design and motivation:

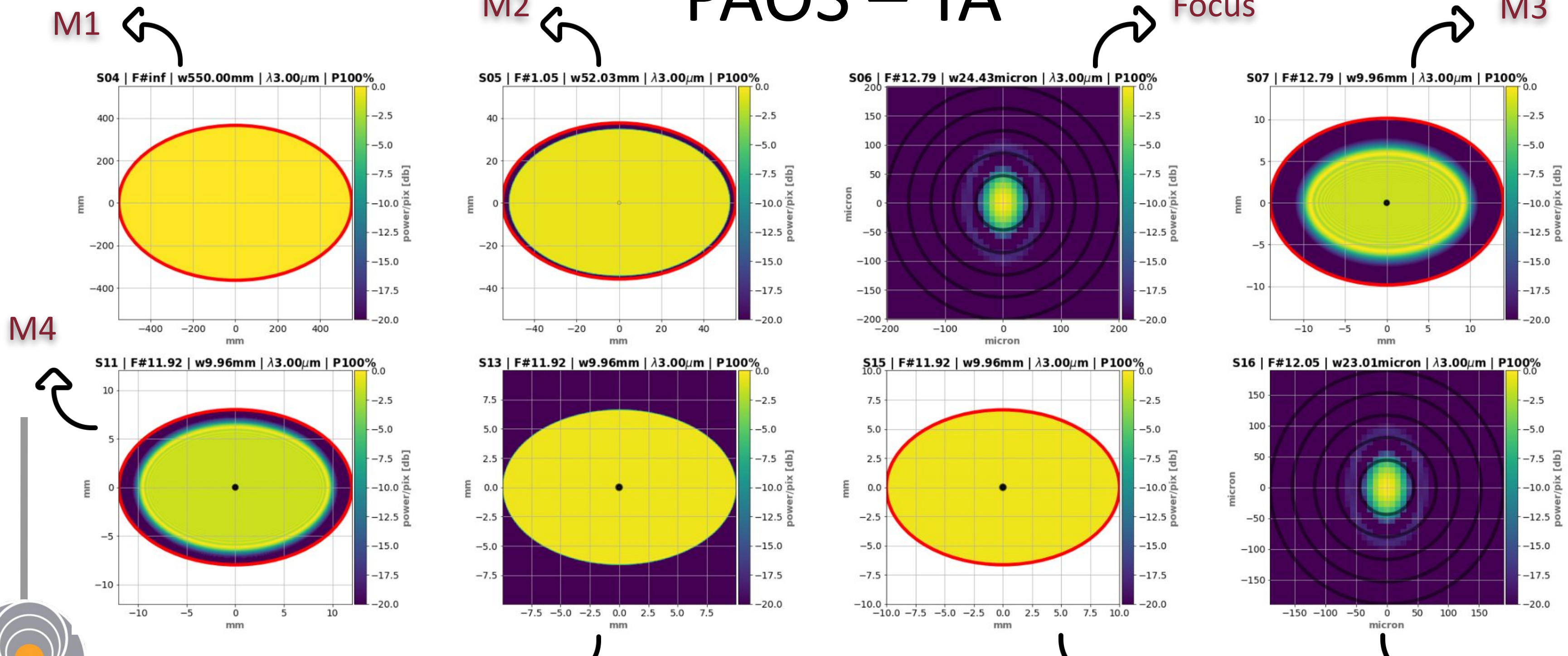
- Study the effects of ***diffraction and aberrations*** impacting the Ariel optical performance and related ***systematics***

General aims and objectives:

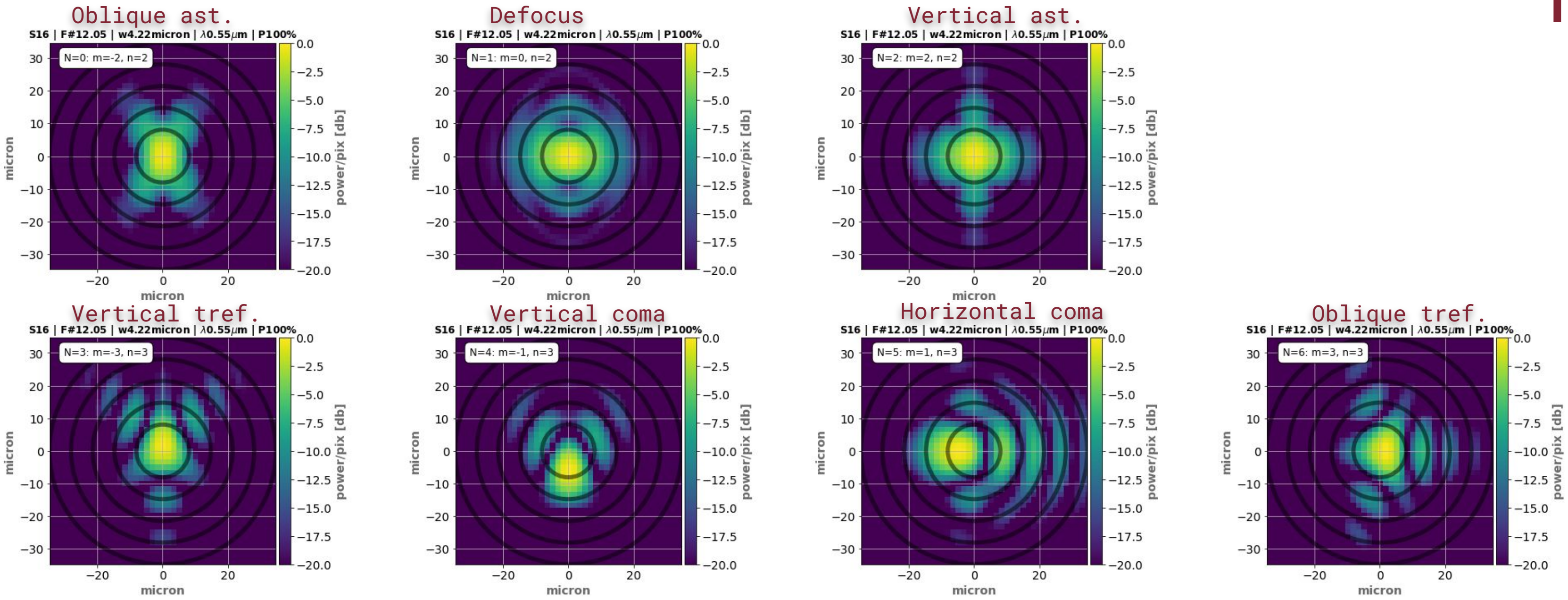
- optimize the ***instrument design***
- develop ground and in-flight ***calibration strategies***
- optimize the ***scientific return***



PAOS – TA


M1
M2
Focus
M3
M4
ExP
ParL
ImgP
S04 | F#inf | w550.00mm | λ3.00μm | P100%
S05 | F#1.05 | w52.03mm | λ3.00μm | P100%
S06 | F#12.79 | w24.43micron | λ3.00μm | P100%
S07 | F#12.79 | w9.96mm | λ3.00μm | P100%
S11 | F#11.92 | w9.96mm | λ3.00μm | P100%
S13 | F#11.92 | w9.96mm | λ3.00μm | P100%
S15 | F#11.92 | w9.96mm | λ3.00μm | P100%
S16 | F#12.05 | w23.01micron | λ3.00μm | P100%
mm
micron
power/pix [db]

PAOS – TA – 50 nm SFE



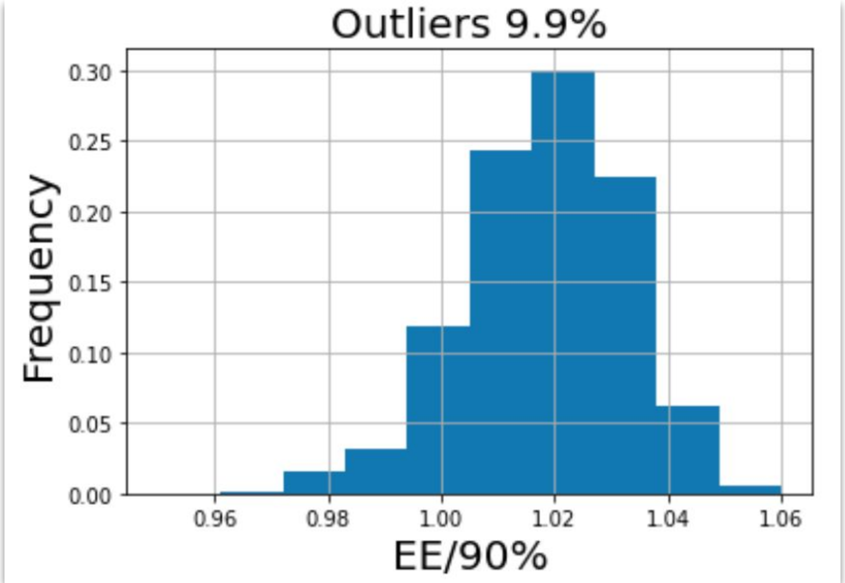
PAOS – Simulations (1)

- Support the maximum amplitude of the aberrations compatible with the scientific requirement for the manufacturing of Ariel's M1

  Initially presented at ACM 2020-02-12, later confirmed with PAOS and inserted in tech. note

Radius of Encircled Energy - to - WFE

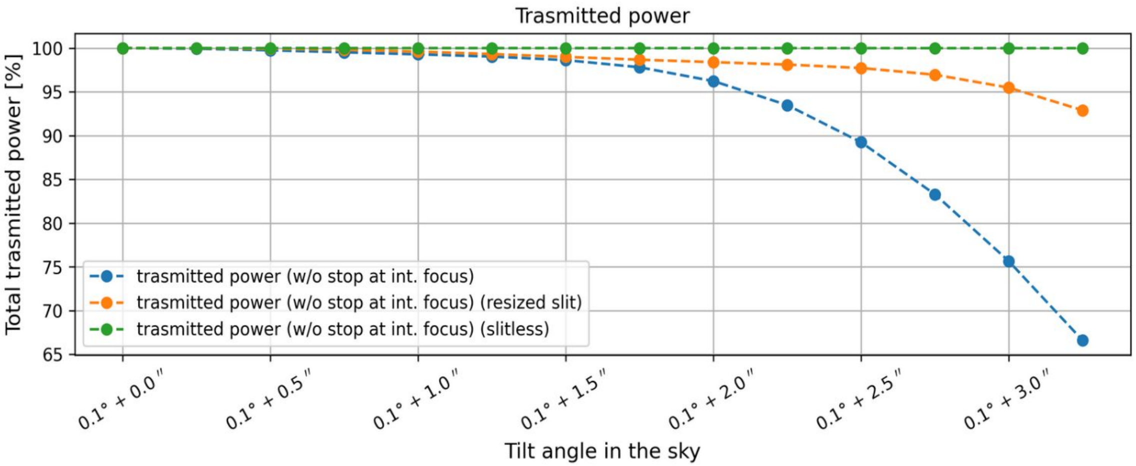
ARIEL-SAP-INST-TN-001



PAOS – Simulations (2)

- ✅ Support the resizing of the Ariel AIRS slits to avoid additional gain noise in the presence of pointing errors

Presented at ACM 2021-06-18



Worked out translation into photometric error at the AIRS focal planes

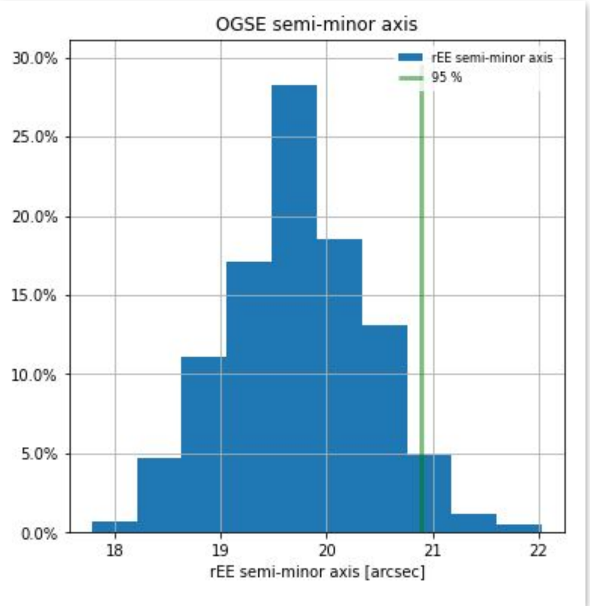
PAOS – Simulations (3)

- ✅ Translate the Surface Form Error allocation on the Telescope Assembly into an rEE requirement for the Ariel OGSE

Presented at ACM 2021-10-15

REQUIREMENT ID R-GSE-2700

Histogram of aperture sizes that give EE ~ 90% at OGSE exit pupil

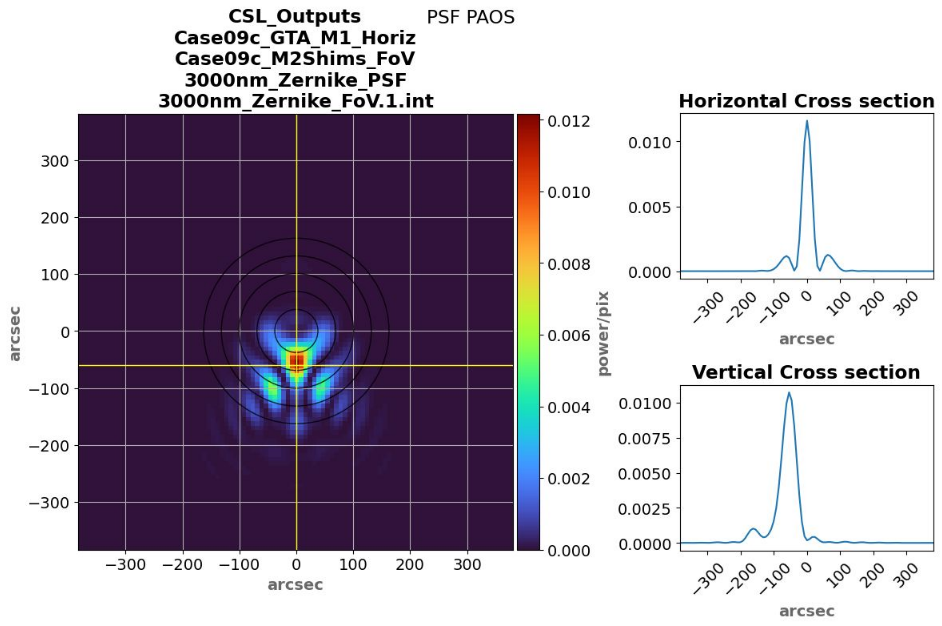


PAOS – Simulations (4)

- ✅ Derive the expected representative PSFs for the Telescope Assembly using inputs from the Ariel STOP analysis Cycle B3-2

Presented at ACM 2022-03-03

Case 9c: GTA (M1 Horizontal) w. M2 shims and FoV



PAOS – Future simulations

Simulate the impact on optical efficiency by thermoelastic variations during the flight

Verify that the realistic PSFs of the mission are compatible with the scientific requirements

Develop strategies for ground and in-flight calibration

PAOS and other Ariel simulators

- Compatible with the other Ariel mission simulators
  - ArielRad (Mugnai et al, 2020a)
  - ExoSim2.0 (Mugnai et al, in prep)
- Can provide ExoSim2.0 with aberrated PSFs to create realistic images on the focal planes of the instruments

Physical
Ariel
Optics
Simulator

Take home messages

1. PAOS is an ***ETE POP model*** of the Ariel telescope and subsystems
2. PAOS propagates the wfo and delivers the ***realistic PSFs*** vs. lambda
3. PAOS can assess the impact of ***diffraction, aberrations***, and related ***systematics***
4. PAOS is written in ***Python*** and has an installer, examples, and an exhaustive guide
5. PAOS is easy to use, generic and versatile, thanks to its ***input system and GUI***