PASSATA — Object oriented numerical simulation software for adaptive optics

G. Agapito[a], A. Puglisi[a], S. Esposito[a]
[a]Osservatorio Astrofisico di Arcetri, Largo E. Fermi 5, Firenze italy

INAF - Arcetri

LABORATORIO NAZIONALE ADONI OTTICA ADATTIVA

ISTITUTO NAZIONALE DI ASTROFISICA INAF

Abstract We present the last version of the PyrAmid Simulator Software for Adaptive opTics Arcetri (PASSATA), an IDL and CUDA based object oriented software developed in the Adaptive Optics group of the Arcetri observatory for Monte-Carlo end-to-end adaptive optics simulations. The original aim of this software was to evaluate the performance of a single conjugate adaptive optics system for ground based telescope with a pyramid wavefront sensor. After some years of development, the current version of PASSATA is able to simulate several adaptive optics systems: single conjugate, multi conjugate and ground layer, with Shack Hartmann and Pyramid wavefront sensors. It can simulate from 8m to 40m class telescopes, with diffraction limited and resolved sources at finite or infinite distance from the pupil. The main advantages of this software are the versatility given by the object oriented approach and the speed given by the CUDA implementation of the most computational demanding routines. We describe the software with its last developments and present some examples of application.

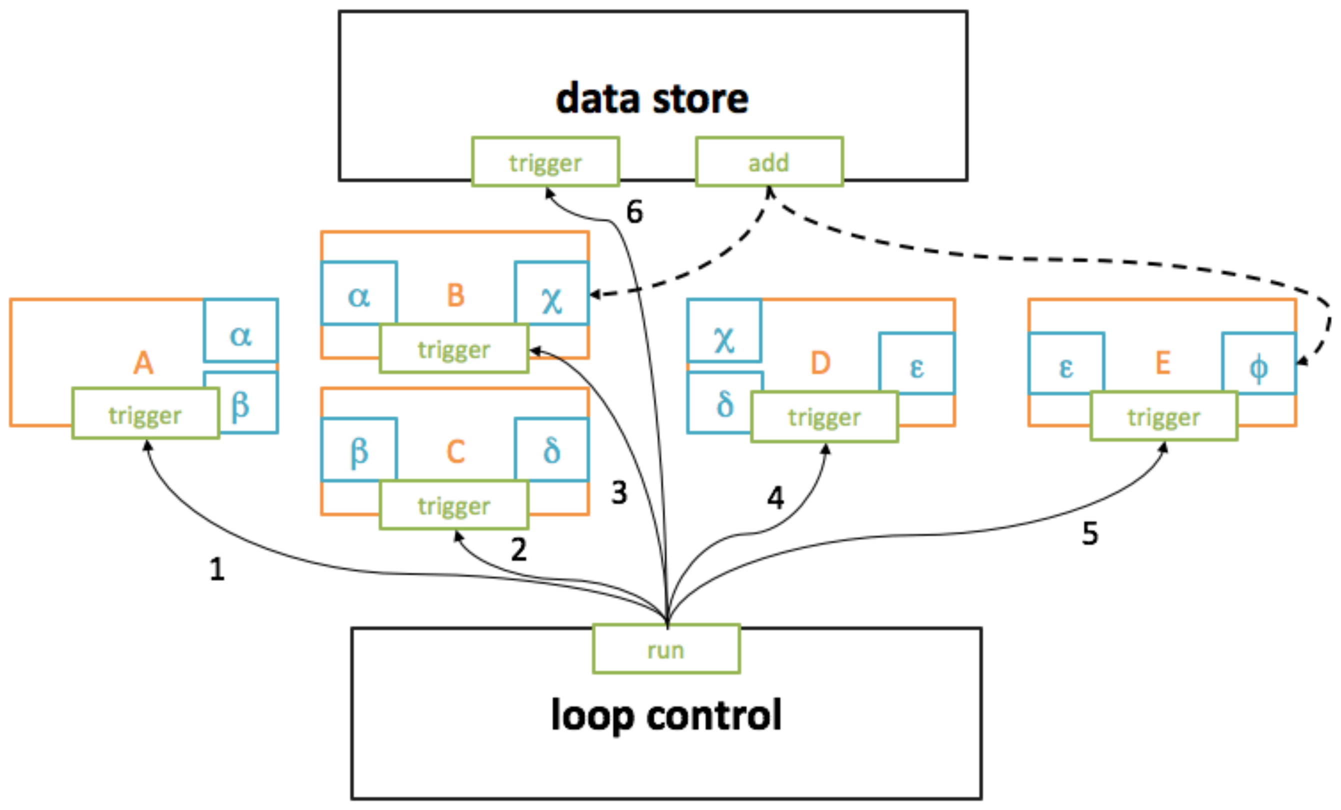
Structure

A PASSATA simulation is composed of loosely coupled **processing objects** each dedicated to a single task. Processing objects exchange data through **data objects**, chosen from a list of pre-defined datatypes.
In addition to processing and data object, a number of **housekeeping objects** help in coordinating the simulation.

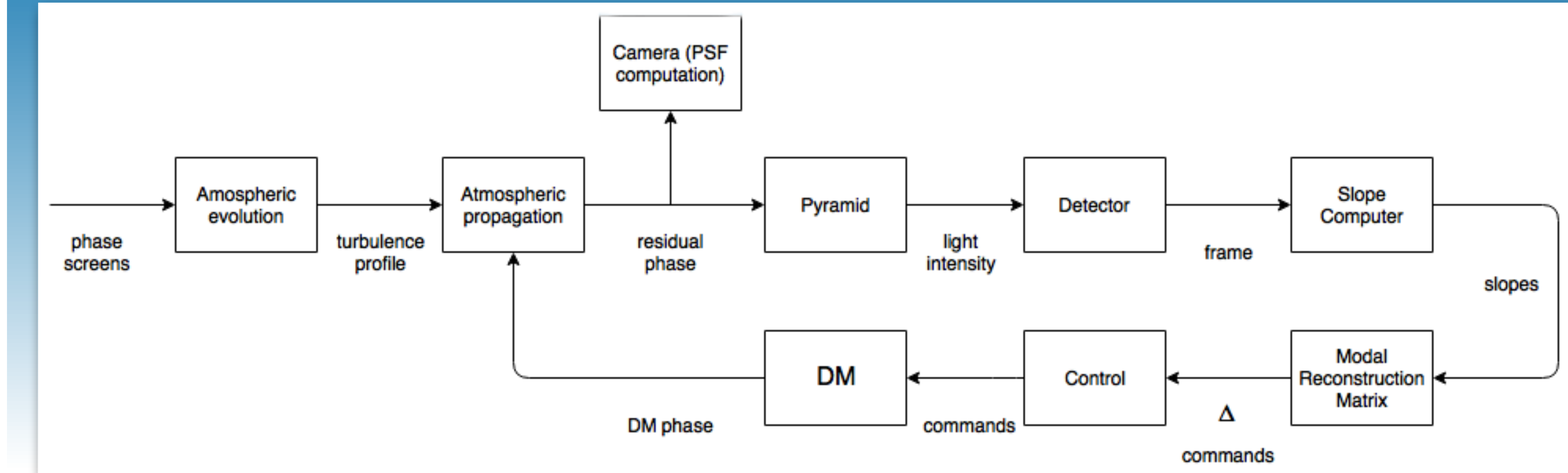
The most important house keeping object is the **loop control** object, that keeps track of simulated time and triggers all processing objects once per simulation step.
Once triggered a processing object is not forced to produce data: typically, it will check whether its input data has been refreshed, and decides whether to process it immediately, or wait for more input at a later time (example: a detector might integrate its input for several simulation steps).

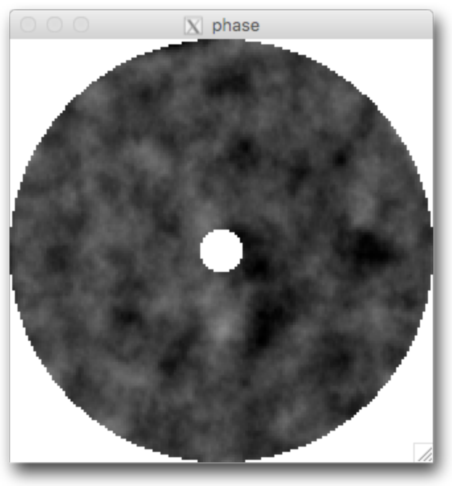
Build and run a simulation loop

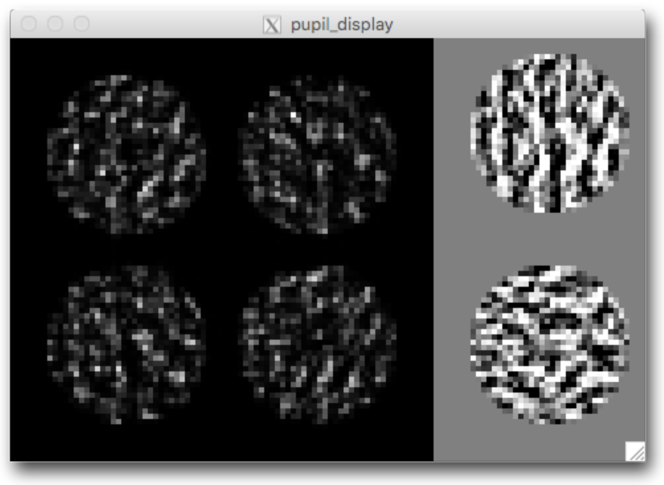
- write the parameters dictionary for each object;
- make a new **factory** object;
- call the factory methods to build all the processing and data objects needed from the parameters dictionary;
- set the input data objects of the processing objects;
- add in the correct order the processing objects to the **loop control** object;
- add to the **data store** the desired output objects;
- call the run method of the **loop control** object.

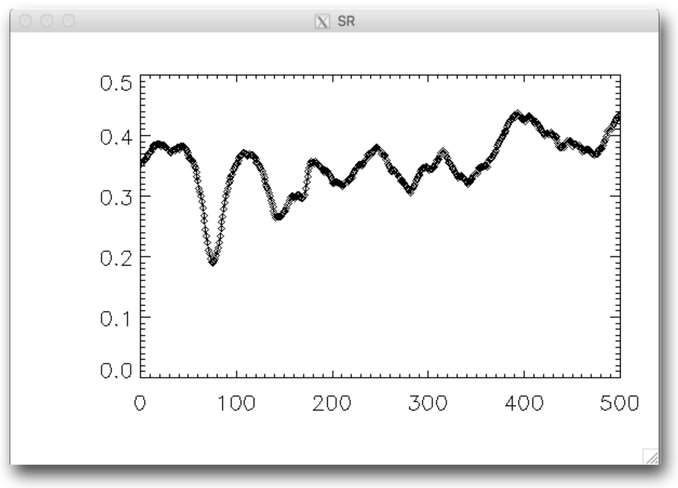


Example: SCAO loop









Hystory of simulated systems...

- LBT: FLAO [1] and SOUL [7]
- GMT: NGSAO [2]
- VLT: ERIS [4]
- E-ELT: MAORY SCAO [8]
- Visible MCAO [5]
- LGS system for WFS sensing study [9]

LARGE BINOCULAR TELESCOPE

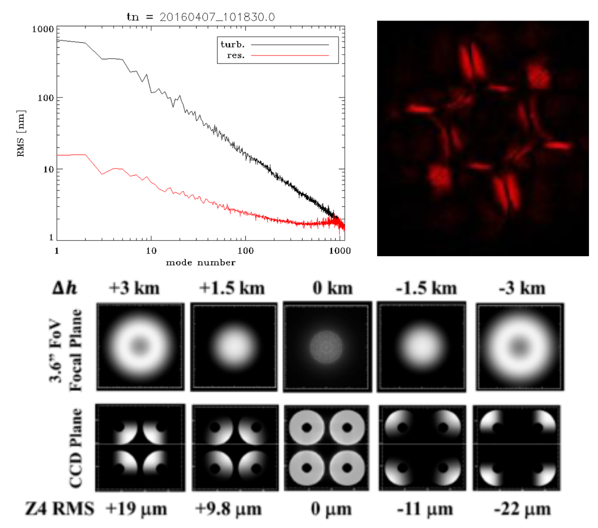
ERIS

MAORY

GIANT MAGELLAN TELESCOPE

... and what can be done with PASSATA

- SCAO, GLAO and MCAO with SH and Pyramid WFSs
- On-sky modal reconstruction matrix calibration
- Off-line PSFs from saved simulations
- Modal decompositions of residual/turbulence phase
- Guide star 3D extension with both SH and Pyramid WFS
- Disturbance management (vibrations, NCPA, ...)
- NCPA compensation
- Segmented DM phasing



and much more!

GPU acceleration

PASSATA offers the possibility (optional) of accelerating the code using a GPU. We selected the CUDA API developed by NVIDIA.
The system automatically detects the accelerated DLM and builds GPU-accelerated versions of the processing objects whenever available.
In order to minimize data transfers, these objects will allocate GPU-aware data objects, that keep the data on the GPU and just provide a **handle** (an opaque identifier) to the GPU-allocated data that later processing objects can understand. The data is retrieved automatically from the GPU and made available to IDL for each object which does not have a GPU implementation.

Task (on 25m telescope)	CPU time (6 cores)	GPU time
Pyramid module (64 FFTs)	5.0 s	0.1 sec.
PSF generation (2 big FFTs)	1.5 s	0.1 sec.
Total iteration time	10 sec.	0.5 sec.
Practical result	1 simulation run /day	20 simulation runs/day

REFERENCES

[1] F. Quirós-Pacheco, L. Busoni, G. Agapito, S. Esposito, E. Pinna, A. Puglisi, and A. Riccardi, "First light AO (FLAO) system for LBT: performance analysis and optimization," in *Adaptive Optics Systems II*, *Proc. SPIE* **7736**, July 2010.

[2] E. Pinna, G. Agapito, F. Quirs-Pacheco, J. Antichi, L. Carbonaro, R. Briguglio, M. Bonaglia, A. Riccardi, A. Puglisi, V. Biliotti, C. Arcidiacono, M. Xompero, G. Di Rico, A. Valentini, A. Bouchez, F. Santoro, G. Trancho, and S. Esposito, "Design and numerical simulations of the GMT Natural Guide star WFS," *Proc. SPIE* **9148**, pp. 91482M–91482M–15, 2014.

[3] S. Rabien, N. Ageorges, L. Barl, U. Beckmann, T. Blümchen, M. Bonaglia, J. L. Borelli, J. Brynnel, L. Busoni, L. Carbonaro, R. Davies, M. Deysenroth, O. Durney, M. Elberich, S. Esposito, V. Gasho, W. Gässler, H. Gemperlein, R. Genzel, R. Green, M. Haug, M. L. Hart, P. Hubbard, S. Kanneganti, E. Masciadri, J. Noenickx, G. Orban de Xivry, D. Peter, A. Quirrenbach, M. Rademacher, H. W. Rix, P. Salinari, C. Schwab, J. Storm, L. Strüder, M. Thiel, G. Weigelt, and J. Ziegleder, "Argos: the laser guide star system for the lbt," *Proc. SPIE* **7736**, pp. 77360E–77360E–12, 2010.

[4] A. Riccardi, "The ERIS adaptive optics system," *Proc. SPIE* **9909**, p. in these proceedings, 2016.

[5] S. Esposito, "A preliminary study for a visible MCAO system at VLT-AOF," *Proc. SPIE* **9909**, p. in these proceedings, 2016.

[6] C. Arcidiacono, "Effects of the outer scale on the adaptive optics simulations," *Proc. SPIE* **9909**, p. in these proceedings, 2016.

[7] E. Pinna, "SOUL: the single conjugated adaptive optics upgrade for LBT," *Proc. SPIE* **9909**, p. in these proceedings, 2016.

[8] P. Span, "NGS WFSs module for MAORY at E-ELT," *Proc. SPIE* **9909**, p. in these proceedings, 2016.

[9] S. Esposito, "LGS wavefront sensing: pyramid and SH sensor comparison," *Proc. SPIE* **9909**, p. in these proceedings, 2016.