Progress and prospects in AO simulation capabilities

Miska Le Louarn
European Southern Observatory, Karl Schwarzschildstr 2, 85748 Garching, Germany;
lelouarn@eso.org

ABSTRACT

In this paper, we present different methods of simulating AO systems, and show how these simulation tools have evolved in the last years. The evolutions have been driven by new projects (like more complex MCAO systems) and new telescopes (especially ELTs). These developments have been made in several directions: increasing simulation speed, complexity (more effects taken into account) and integrating the latest developments in algorithms.

We will also discuss new directions in simulations, like new applications and new ways of estimating AO system performance.

Keywords: Adaptive Optics, Simulations, Modeling, End to end simulations, Parallel computing

1. INTRODUCTION

Adaptive Optics simulations have usually been used to predict and study the performance of AO system, but other applications exist. We can describe these applications by looking at the users of the simulation tools.

1.1 AO system designer

One of the most important aspects of AO simulations is to guide the AO system designer in hardware choices. Currently, the choices of the most fundamental system parameters (number of actuators, sub-apertures, laser power, read-out noise…) are backed-up by simulation results.

The second role of AO simulations for the AO system designer is tolerancing. Indeed, once the main components have been chosen, one tackles the overall system aspects. For example, one wants to address the impact of shift of the Wavefront sensor (WFS) compared to the deformable mirror (DM) to constrain the stiffness of the opto-mechanical assembly housing those components. Other types of tolerancing can be also analyzed, like sensitivity to jitter (to dimension the stroke and bandwidth of a tip-tilt mount or deformable mirror), or specify the specifications of components to be developed (like maximum acceptable read-out noise, in case a WFS detector is developed especially for an AO system).

Finally, a sensitivity analysis is usually carried out, in order to assess how external parameters (like seeing, τ_0, C_n^2 profile, Na profile) impact the performance of the system. This last step can yield new ways of estimating AO system performance, like availability (see section 7).

1.2 AO system user

An increasing number of AO users (for example astronomers) prepare or analyze their observation results with the outputs of AO simulation tools. Indeed, AO systems (existing or planned) have particularities in terms of Point Spread Function (PSF) behavior (spatial and temporal variability, speckled structure, etc…), which can have an impact on the astronomical science extracted from the observations. Therefore, using AO simulation results can help in the data reduction procedure for example.

For AO systems which are still in the planning phase, simulation tools can also be useful for astronomers. In recent years, new AO system modes have been investigated and implemented (GLAO, LTAO, MCAO, MOAO,…), and simulation outputs can be used to analyze which AO system is the most appropriate to reach a given astronomical goal, and answer questions like “is GLAO better suited for my observation than MCAO ?”. For this, one can use an AO simulation tool to create synthetic observations with various AO systems, in identical atmospheric conditions. In Figure 1, we show the result of such a simulation. The PSFs from an AO simulation for the European Extremely Large Telescope (E-ELT) have been used to simulate an astronomical observation of a star forming region in the LMC. This allowed to assess the suitability of the simulated AO system for that observation. Such simulations can also be used further to define parameters relevant for the instrument used behind the AO system. As an example, these could be the pixel scale of the detector and the spectral range of the instrument, which can depend on the useful level of AO correction at short wavelengths.

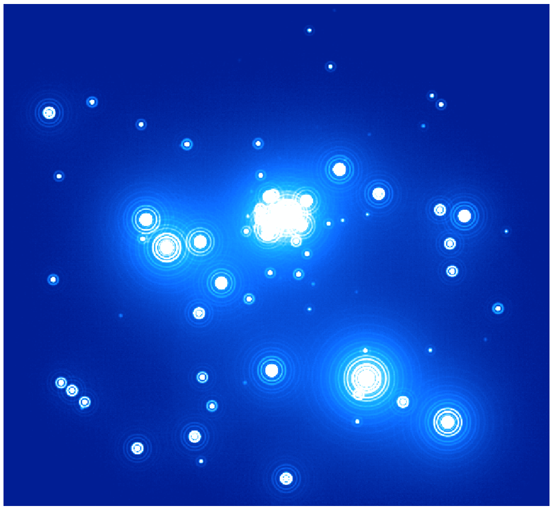


Figure 1: Simulated K band image of a young Star forming region in the Large Magellanic Cloud. Exposure time=25 hour, as seen by the E-ELT. Illustration from Calamida et al., Design Reference Mission report, ESO, 2010. The total field of view is only about ~0.1''.

A further use of AO simulations is to prepare and test a data reduction pipeline. Indeed, once a simulation provides the PSF output of an AO system, and this data can be fed into a simulation that reproduces the instrumental characteristics, one has the ability to simulate realistic observations. These "observations" can then be used to develop data reduction tools / pipelines, which will later be used to process real data, once the instrument is completed.

Finally, such simulation tools can be used to generate Exposure Time Calculators (ETCs). ETCs allow the astronomer to analyze how much time an object should be observed to obtain a given signal to noise ratio. Several parameters (like seeing) influence this results. In the case of AO, the level of AO correction plays a critical role in this, and using a (maybe simplified) simulation tool to estimate the AO performance from a set of predefined inputs can be very useful.

1.3 AO system "Debugger"

Another use for simulation tools is to use them for system debugging. This is of course similar to basic performance estimation and tolerancing, but at yet another added level of complexity. Here the goal is to understand the behavior of an already existing experiment / instrument.

Indeed, once an instrument is in the lab / on the sky, a simulation is used to verify that the instrument is behaving as expected. If it is not, new effects, usually hardware related, like telescope vibrations, or lab-measured hardware limitations (particular CCD characteristics for example) can be introduced in the simulation, to see if they could explain the departure from previous models. This shows that a simulation tool should be flexible enough, that effects can be added into the simulation at any stage in the life of the AO system. Such simulations allow to better understand the instrument, but also to understand the environment (usually due to the telescope) in which the system is run.

2. PRESENTATION OF SIMULATION TOOLS

Different tools can be used to simulate AO performance, and what constitutes a "simulation" is actually quite variable. Here, we will present various levels of "simulations" that are used in AO.

2.1 Error budgets & analytic wavefront error estimates

Error budgets are a convenient way to collect different error sources. The easiest way to populate them is to simply gather sources of error using analytical formulae. For SCAO, this is quite an efficient method, since analytic expressions exist for most of the errors (fitting, aliasing, photon and read-out noise etc).

For errors which don't have an easy analytical expression, one can use a more sophisticate simulation tool (see following sections) and then use the value found by them to add as a term in the budget.

The advantage of error budgets (and more generally analytical formulations yielding the variance of the wavefront associated to this error) is that they are very fast. They allow to quickly get a rough estimation of system performance, and also to clearly identify the major error sources (and therefore the areas where the most performance can be gained in case the performance is not deemed sufficient).

The major drawback is that the corrected Point Spread Function (PSF) is not provided by these formulae. Considering the different applications of AO simulations and especially PSFs produced by them, this is rarely satisfactory. Indeed, deriving more complex metrics (Ensquared energy within a given pixel size, contrast in high contrast imaging, Point Source Sensitivity, etc…) without an accurate PSF model is challenging. Therefore some way of estimating the PSF is usually needed. Another problem with this method (compared to for example end to end simulations – see later) is that correlations between errors are usually neglected, and the method can therefore be pessimistic. Also, this approach not very flexible, since new effects can be added once an analytical expression for it is derived.

Tomography can be broadly tackled by some analytical formulas to see how many DMs and guide stars are required (see [6], [7], [8]).

Currently however, most AO systems are built using an error budget, and this budget is used as a "summary" collecting all the identified sources. The next category of simulations improve on the basic analytic ones by estimating the PSF.

2.2 Analytical / Semi-analytical with PSF estimation

The next level of complexity and accuracy in AO simulations is represented by analytical (but Fourier / power spectral density instead of just wavefront error) and semi-analytical methods estimating the PSF. Examples of such simulation tools are CIBOLA[3], PAOLA[2] and the ONERA Fourier Code[1], and the GLAO modeling of [5]. These methods are based on calculations in Fourier space, usually of the residual Power Spectral Density after AO correction. One can calculate filter functions representing the effect of the AO system on the incoming turbulence, and use these to calculate the corrected PSF. One can also generate many residual phase screens using this filter, in order to estimate the long exposure PSF without having to derive analytically the effect of the filter on atmospheric turbulence. These methods are similar to methods used to calculate the corrected PSF from AO system telemetry data.

Recently, models have been developed to handle tomographic AO systems (like MCAO) and they are quite flexible as to the effects which are simulated. However, they are usually limited in their ability to handle laser guide star specific effects (cone effect and tip-tilt indetermination), since these effects are difficult to express in the Fourier domain. Some work-arounds have been derived for example by ONERA, using an Unseen region correction algorithm ([4]) to account for the cone effect using their Fourier simulation tool.

These approaches are very well suited to the first phases of a project, where a large parameter space needs to be explored (since they are fast), and where different AO systems need to be compared. Once promising candidates are selected, they can be simulated more in depth with the end to end tools (see the following section). The following plot shows a practical application of a Fourier simulation code, in simulating MAORY, a potential MCAO instrument for the E-ELT. The authors used a Fourier simulation Code, with uncompensated region correction (FC+U - to take into account the cone effect) in this MCAO case. This simulation tool allows to predict the performance of the high order loop, including residual anisoplanatism after tomography and correction by multiple DMs, cone effect etc.

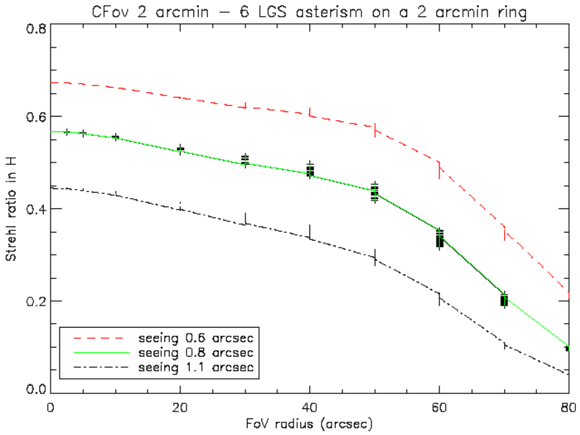


Figure 2: MAORY High Order H band Strehl Ratio performance (no implementation and NGS errors) as a function of the FoV radius (distance of the performance point to the field centre) for the three seeing conditions: good, median, bad (respectively 0.6 ; 0.8 and 1.1 arcsec seeing, see detailed definition in Section 4.1). FC+U results for 42m case. Nominal conditions: median seeing case, 6 LGS on 2 arcmin, 2rd² equivalent uniform slope noise, 500Hz sampling frequency, optimal reconstruction on 9 layers accounting for generalized aliasing, performance optimized on 2 arcmin CFoV., see [9]. Illustration courtesy the MAORY team.

2.3 End to end models

End to end models are used simulate AO systems with the highest level of accuracy. They are based on the generation of phase screens to model the effect of the atmosphere. Light is then propagated to the level of the telescope, taking into account the nature of the guide star (NGS / LGS) and their position. Then, a wave-front sensor module analyzes the phase (and possibly intensity) coming from the guide star(s) and generates (for example via the calculation of slopes from the images in each sub-aperture using an FFT, in the case of a Shack-Hartmann sensor) the measurements. These measurements are fed into a reconstructor, which generates then commands to be sent to the deformable mirror(s). These are also represented by phase screens, and subtracted from the incoming phase. After this, the screens go through temporal evolution (usually shifting) and the cycle is repeated. Usually some thousands of time steps are necessary to simulate realistic AO performance.

Since this procedure is close to what happens in a real AO system, this scheme is relatively accurate. It is also easy to add new effects, since one only needs to find a way to modify the phase in the system to model it. The simulation then shows what the impact of the effect is.

However, these kinds of simulations have many parameters to optimize (which can be time consuming) and are very heavy numerically. Indeed, one needs to model the physical processes (wavefront generation, propagation, evolution), in addition to simulating the AO system itself. Moreover, since the simulation calculates the response of the AO system for each AO loop step, many iterations are needed to reach convergence. This complexity limits the use of these tools to the study of a smaller number of parameters than the analytic models.

A huge amount of work has gone in the last few years in developing and optimizing these codes, like the TMT MAOS [11], yao [10], the ESO Octopus [29], or the ONERA end to end simulation tool [12].

3. PROGRESS IN END TO END SIMULATION TOOLS

The latest developments in end to end simulation tools have revolved around several themes. New modes (like tomographic AO systems) and new telescopes (especially ELT projects) have driven the development of simulations. As an example of new effects taken into account, one can cite spot elongation. Indeed, due to the increase in the size of the telescopes, spot elongation becomes a dominant source of noise in the wavefront sensing process. Therefore, this has been included in most end to end simulations which simulate ELTs (see for example [13]). Another example is shown in Figure 1, where the Rayleigh backscatter from a multi-LGS system is plotted. Several guide stars are used to probe a volume of turbulence. Each LGS produces its own Rayleigh cone and this can introduce noise in the WFS measurement coming from another star. This “fratricide” effect has been modeled and included for example in the yao AO simulation software ([10], see also [14]).

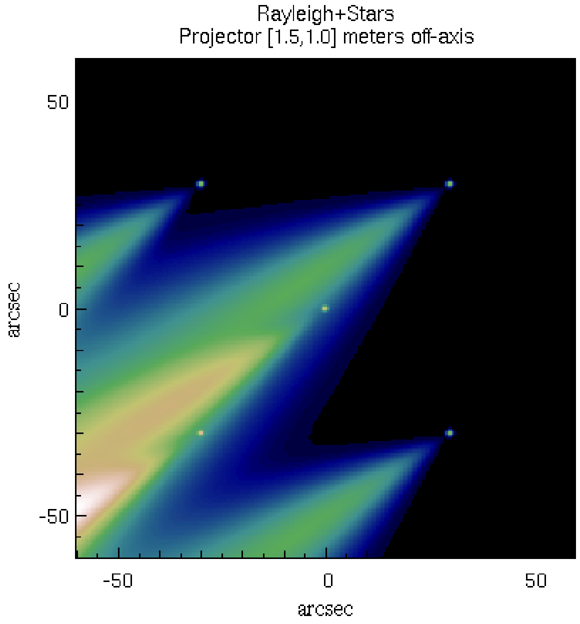


Figure 3: A generated field of 120"x120" as viewed by a subaperture [1.5,1.0] m off-axis, showing the 5 laser guide stars in the MCAO configuration and their associated Rayleigh backscatter (Linear ITT). Courtesy F. Rigaut.

Other aspects, like primary mirror segmentation and the particularities of the modeled telescopes have also pushed major developments. An example is shown in the following figure, coming from the PAOLA software:

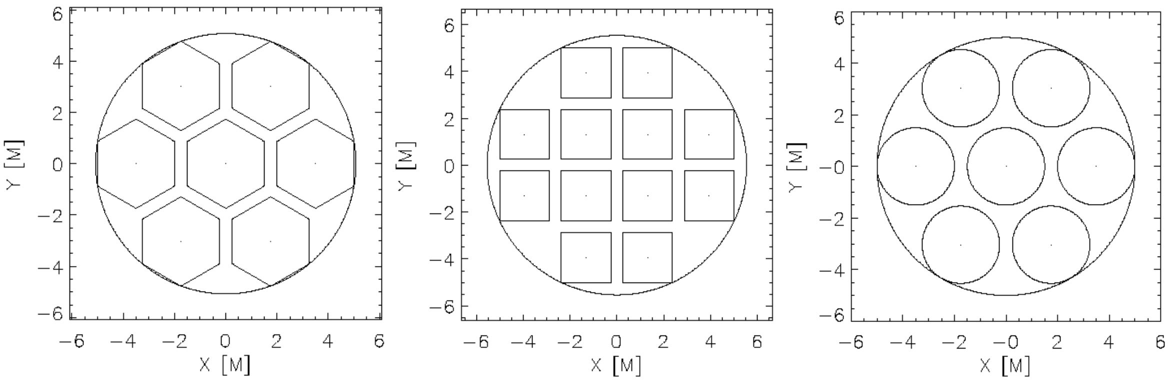
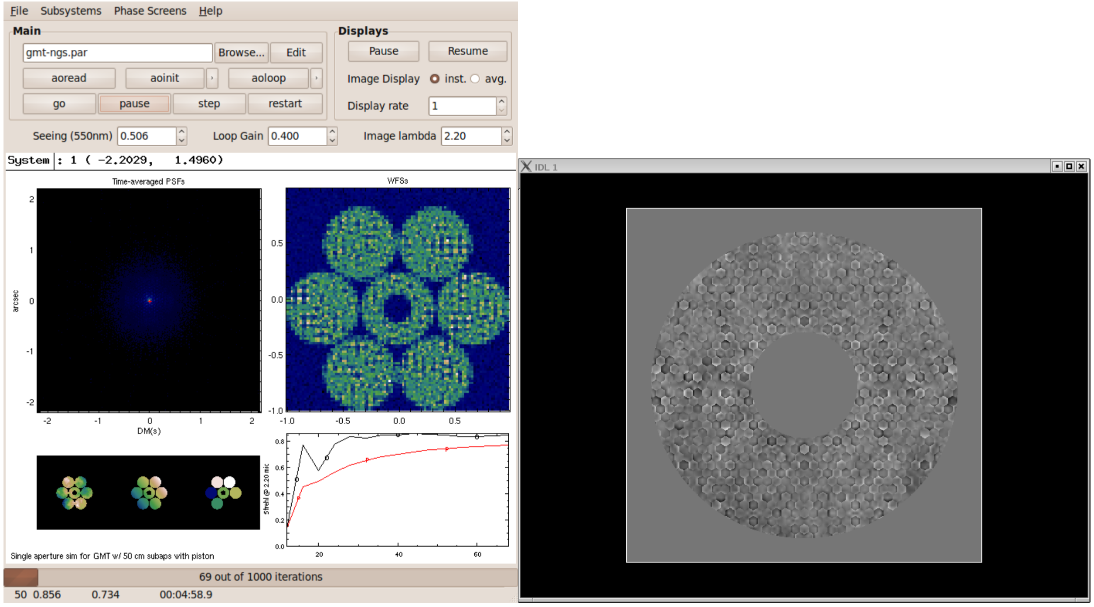


Figure 4: Segmented pupils in the PAOLA AO simulation software – courtesy L. Jolissaint.

PAOLA uses analytical formulae to derive the effect on the AO system and PSF structure of a segmented primary mirror. Different segment shapes (hexagonal, square, circular) and gap sizes between the segments can be considered. This also demonstrate the growth of AO simulations beyond AO, into the domain of telescope design, co-phasing and high contrast imaging. Further examples of new telescope projects driving AO software are shown in the following figure:



gmt-ngs.par

Image Display ● inst. ○ avg.

Display rate 1

Seeing (550nm) 0.506

Loop Gain 0.400

Image lambda 2.20

```
System : 1 ( -2.2029,   1.4960)
```

Single aperture sim for GMT w/ 50 cm subaps with piston

69 out of 1000 iterations

```
50  0.856       0.734        00:04:58.9
```

Figure 5: On the left, modeling of the GMT with yao (see [28]). Illustration courtesy M. van Dam. On the right, we show how an SCAO system corrects a particular mode on the E-ELT segmented telescope. Only high spatial frequency modes remain, see segment imperfection (exaggerated in this case)

It shows an upgrade of yao to handle the primary mirror shape of the Giant Magellan Telescope, which is made of 7 ~8m monolithic mirrors. The opposite direction in segment size is shown on the right, where the ESO Octopus AO tool is shown as it corrects a telescope primary mirror for the aberrations due to wind. In this exaggerated case, we can see that the AO system is very good at removing low spatial frequencies, while some high spatial frequencies still remain.

Most tools can now simulate almost any conceivable AO system, using any number of DMs and WFSs. In addition, new reconstruction algorithms have been implemented, tested and compared including tomographic ones (see for example [15], [16] and the sparse matrix reconstructors in yao, called soy). These algorithms, which have the goal of reducing the requirements on the real time computers of AO systems for ELTs have had the additional benefit of accelerating simulations, since they are much more efficient computationally than the conventional matrix vector multiplications.

Another significant development has been the tighter integration of AO simulation tools with the post-focal instrumentation using AO. For example, the CAOS simulation software has a specific version tailored to the SPHERE instrument, including modules for coronography and the integral field unit and polarimeter. This allows to really get end to end simulations of the whole AO+instrument combination (see [17]). This same software also allows to better verify the consistency of the simulation results by having a unified interface between a full end to end numerical tool (CAOS) and an analytic software (PAOLA).

4. OVERCOMING THE LIMITATIONS OF END-TO-END SIMULATIONS

4.1 Complexity

The computational cost of simulating AO systems is due to different factors. First, a “simple” SCAO command matrix for the E-ELT takes ~400MB of RAM. An MCAO matrix can easily be several Gigabytes, especially in the cases where 5-9 LGSs are used, in addition to 2-3 deformable mirrors. Any manipulation of this kind of array is time consuming. Luckily, new reconstruction algorithms help, and the sparseness of the interaction matrix can reduce the computational and memory requirements – at the cost of an algorithmic complexity.

The second factor taking simulation time is the physics of light propagation that has to be modeled, both in the atmosphere and in the internal optics of the AO. FFTs which can be large are used (in case the PSF needs to be estimated instead of just the wavefront error), and many small ones (to simulate the Shack-Hartmann sensor). One can of course simplify the model to avoid these FFTs, at the cost of accuracy (for example in the centroiding procedure). However, time consuming operations still remain. One needs to shift large phase screens, possibly perform interpolations and other array operations on large matrices (representing for example the telescope aperture). Because array operations are so time consuming, one is tempted to model the telescope with the smallest possible number of pixels. While this may be acceptable in some cases, some others, for example involving LGSs, push the array sizes towards larger ones. Indeed, the size of the array representing the pupil is proportional to the field of view one can simulate. And in the case of LGSs (and spot elongation), the required fields of views can be huge. Just as an example, a 10'' – 20'' field of view (typical Shack-Hartmann WFS sub-aperture size for an LGS WFS with spot elongation on the E-ELT), one needs pupil arrays in the 4000 pixels range. All other arrays (like the atmospheric phase screens) scale accordingly. This implies some huge FFTs, and even the FFTs in the WFS sub-apertures are not that small anymore. And since one is investigating the impact of spot elongation on centroiding algorithms and varying sodium profiles, some accurate centroiding simulation is required.

An additional complexity comes from the fact that the ELT pupils are large, which implies that the time it takes for turbulence to cross the whole aperture is large. This, in principle, means that a long time span needs to be simulated in order to realistically represent the performance of the AO system. Therefore, many iterations of the AO simulation need to be carried out. Luckily, the number of iterations can in some cases be reduced by performing "warm restarts" in the simulation, as implemented in the yao simulation. In that case, the AO loop is closed and run for some time. Then, new phase screens are generated, and the loop reclosed. Performing several of these restarts and adding up the PSFs obtained in each of them allows to accelerate the convergence of the simulation.

Another positive factor (for AO simulators), is the fact that wind shake on the telescope can become a dominant source of tip-tilt. Therefore, its decorrelation is faster than that of the atmosphere (which, thanks to outer scale of turbulence, decorrelates slowly on an ELT in units of necessary iterations of the simulation), reducing the number of simulation steps. This behavior of the tip-tilt points towards the need to sometimes treat tip-tilt (and low orders in general) separately from the high orders. This is not necessarily a bad thing, since this might also happen on the real system (see [18]).

Complexity in AO simulations is not only due to computational complexity. As we have seen, in recent years, the number of different modes for AO has increased tremendously. And of course, the AO simulation tools have grown to be able to cope with these modes. Obviously, tomographic AO systems (MCAO, LTAO, MOAO) with their particularities have driven the AO simulators towards more options in their simulation tools. However, this is not all. In AO there are many possible WFSs (SH (with or without spatial filter), curvature, Pyramid…), DM technologies (Piezo-Stack, curvature, adaptive secondary…), MCAO wavefront sensing schemes (layer oriented and star oriented). And implementing just a subset of these increases the implementation complexity of the simulators considerably. Adding novel reconstruction algorithms (FFT, iterative methods, in addition to the variations of the "classic" matrix vector multiply) does not make the software complexity of the simulation software any easier to manage and optimize.

This increase of complexity has pushed some into developing new tools with reduced capabilities tailored to just one project, while others have upgraded their previously existing tools with new "options". This of course brings the problem of testing and validating new features or new tools. We will discuss this in section 6.

4.2 Hardware requirements

The main hardware limitations of end to end simulation are twofold. First, the computation load in CPU operations increases with system complexity and "size". Second, the need for memory also increases, since large array need to be manipulated. Finally, the software complexity itself increases too, since the simulation tools become more complex and more optimized, which increases the need for complex validation procedures.

Thankfully, hardware has also evolved. It is now common to have 64 bit computers, with 64 bit operating systems allowing the user to manipulate very large arrays in a single block, overcoming the old 32-bit barrier of 4GB, which used to limit the amount of memory which could be allocated by a single process. This in itself is a huge benefit for AO simulations. In addition, hardware architectures allowing tens of GBs of RAM to be installed on a single PC are now "affordable" and help the simulation to scale to the new ELT projects.

Another hardware evolution helping the AO simulators is the development of multi-core PCs. It is now possible to have more than 10 CPU cores on a single machine, allowing in theory to speed simulations up by a factor proportional to the number of cores. However, as we will see (section 4.3), this is not as straightforward as simply allocating more memory when going from a 32 to 64 bit environment. Software adaptations are necessary to benefit from this evolution in number of cores. Some AO simulations are written in a high level "problem solving environment" - PSE, like IDL, Matlab and yorick. Some of the vendors / developers of these environments have been able to upgrade their product to take advantage of the multiple cores present in a PC. This has allowed some simulations to speed up significantly, just by upgrading the version of PSE. This evolution trend (increasing number of cores, PSE upgrades to optimize their use on multi-core machines) is likely to continue. Indeed, the speed of a single core has been more or less stagnating, and CPU vendors seem to increase the number of cores rather than increasing the speed of a single core.

Unfortunately, some PSEs have not been upgraded yet. Also, some simulation tools are written in a lower level language – like C, and it is up to the developer to parallelize the application "by hand".

Other hardware evolutions, like the advent of Graphical Processing Units hold great potential. However, they have not yet been used extensively in AO simulations, although some tests have shown great potential. Some mathematical libraries do have versions benefiting from GPUs, so it is likely that they will make their entrance in the field of AO simulations one way or another (through PSEs or direct programming).

4.3 Paralellization

As we have seen, parallelization (either explicit or through PSEs) seems to be both necessary and desired (to increase the speed of ever more complex simulations and benefit from multi-core CPUs). Here we discuss different ways to achieve this in an AO simulation.

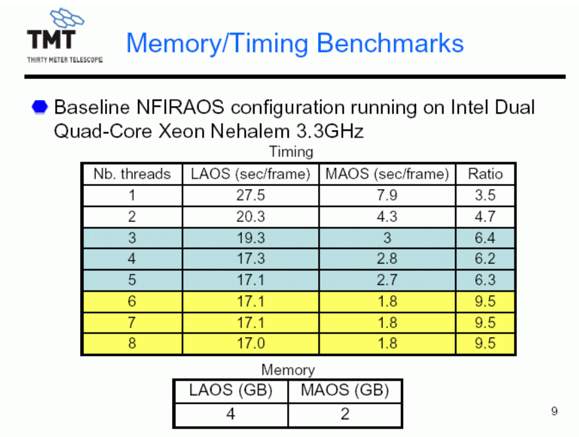
The simplest parallelization scheme is to launch several simulations simultaneously, with different parameters. This is quite efficient, since one doesn't need to update the software at all, and one can reach significant gains in performance. Drawbacks are that a single simulation can take a long time to complete, that one needs enough memory on the simulation machine to hold the many simulations in memory simultaneously. Also keeping track of which simulation did what with which parameter can become challenging if many simulations run simultaneously.

Since the AO simulation is made of time steps, the idea of using "warm restarts" (discussed earlier in section 4.1) can be extended to parallelization. One could run several warm restarts simultaneously, to increase speed (at the cost of memory consumption) and reach convergence more quickly. This also reduces the simulation time and the amount of book-keeping – since a single set of parameters is run once.

If more complex parallelization is required, then one is confronted with a fundamental choice. Does one run the simulation on one machine only (shared memory paradigm) or does one want to use several machines in a cluster connected by a fast network to run a single simulation (distributed memory paradigm). This choice is fundamental. Shared memory approaches are usually easier to program and to manage (no need to get the best out of a network for example, more mainstream programming environments). However, one is limited to the number of cores available on a single machine, and limited to the amount of memory available on that machine. In a distributed memory case, one can add machines to increase the available memory and number of CPUs. Depending on the number of desired machines, this solution can be cheaper than buying a single very powerful computer. Note that a distributed memory code can be also run in a shared memory environment but the reverse is not true.

Different approaches have been taken by AO simulators for parallelization. The ESO AO tool Octopus uses distributed memory in a cluster whereas the TMT MAOS software uses a shared memory scheme and the CAOS IDL code benefits from the IDL multi-threading scheme, and shows significant accelerations thanks to its PSE's evolution.

The following figure shows an example of what gains one can get by parallelizing an AO simulation. In this example, a single threaded application (LAOS) is compared to an optimized multi-threaded version, MAOS. Significant gains in speed can be obtained, using up to 6 cores. Using more cores than this doesn't help, because either the memory bus is saturated so information cannot be exchanged fast enough, or there are not enough independent tasks to take advantage of more cores. However, a gain of a factor of ~10 is reached by parallelizing and optimizing the code, which is quite respectable. The other form of parallelization is shown in the middle and right of that figure. It shows the ESO AO simulation cluster, used to run a distributed memory parallel application, OCTOPUS. One uses many cheap computers instead of one big one, and specialized software written in C, using parallel libraries like MPI.



Memory/Timing Benchmarks

- Baseline NFIRAOS configuration running on Intel Dual Quad-Core Xeon Nehalem 3.3GHz

Timing

Nb. threads	LAOS (sec/frame)	MAOS (sec/frame)	Ratio
1	27.5	7.9	3.5
2	20.3	4.3	4.7
3	19.3	3	6.4
4	17.3	2.8	6.2
5	17.1	2.7	6.3
6	17.1	1.8	9.5
7	17.1	1.8	9.5
8	17.0	1.8	9.5

Memory

LAOS (GB)	MAOS (GB)
4	2

Figure 6: On the left NFIRAOS simulation, 7 atmospheric layers (4kx4k), 6LGS, central launch, physical optics model of pixel intensities for polar coordinate CCD, rms WFE calculated at 9 field points. Figure courtesy of Luc Gilles, see [11]. In the middle and right, ESO AO simulation cluster. This is a distributed memory environment, where many cheap desktop PCs are connected together with a switch (right) to reach the computing power and memory capacity of the sum of the machines. This cluster is used (among other things) to run the OCTOPUS software.

So how difficult is it to parallelize an AO application? In some cases, it is easy. For example, in a Shack-Hartmann WFS, there are many small sub-apertures, which are independent (neglecting diffraction effects between them). Therefore, one can easily parallelize the simulation of lenslets (FFTs, convolution with sodium profile, thresholding, centroiding etc). This is easy and has a big payoff, since ultimately, as many CPUs / cores can be used as there are sub-apertures. A Pyramid sensor can also be parallelized, for example in modulation steps – which are independent.

Mathematical operations can also be parallelized, thanks to specialized libraries. The SCALAPACK library allows to easily parallelize Matrix-vector and Matrix-matrix operations. The FFTw package can be used to parallelize the Fourier transforms – even if this is not necessarily very efficient, due to memory / network bandwidth limitations. So sometimes, already existing and optimized libraries can be used easilly. One should also mention the MKL libraries from Intel, which also help in handling multi-threaded mathematical operations.

Sometimes however, parallelization is harder. Complex algorithms with many steps, iterative methods (needing the result of one iteration before starting the next), sparse algorithms are hard to parallelize because they may not have many independent components. So to summarize, factors of 10 of gain seem in reach with a moderate effort, while higher gains require a lot more though and engineering.

5. NEW INPUTS ARE NEEDED

Because of the advent of new AO modes and ELTs, AO simulations require more accurate input parameters, especially on the atmosphere.

For example, GLAO simulations require better sampled C_n^2 profiles than “conventional” SCAO. Depending on the corrected field of view, the first 100m – 1000m are the most important to estimate correction quality. This also means that statistical understanding of the dome seeing, ground and surface layers of turbulence is now needed, especially to calculate the availability of an instrument – calculation which relies on the statistics of seeing and turbulence profile, rather than just a campaign type set of measurements (see section 7). For example, new measurement campaigns and data re-analysis of old campaigns have been launched in the frame of the ESO Adaptive Optics Facility to better understand the ground layer turbulence at Paranal. Below, we show a high resolution Paranal C_n^2 profile, obtained during the PARSCA Balloon site survey campaign. Note the presence of many extremely thin layers and the complexity of the profile. Trends can be observed (strong layers near the ground). They show the actual complexity of turbulence profiles, and demonstrate that modeling such a complex profiles in AO simulations accurately can be a complex task.

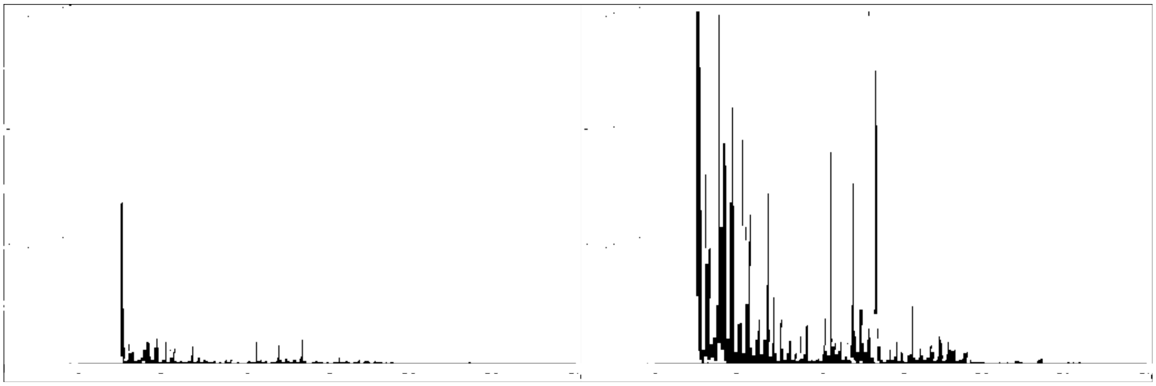


Figure 7: C_n^2 profiles measured during the PARSCA balloon campaign, in Paranal. The plot on the right is a zoom on the plot on the left, to show the fine details of the turbulence profile.

Another example is the sensitivity of tomographic reconstruction algorithms to the C_n^2 profile. On ELTs, this sensitivity can be large ([19]). Using a small number (~5-10 as usual on 8m telescopes) of thin layers might not be accurate enough and more complex profiles are required, especially in tomographic systems (the sensitivity depends on the corrected field of view, and the weight of the priors). New methods (like obtaining these profiles from the WFS measurements - [20]) are being developed, but high precision measurements are likely to be needed – at least to validate the new algorithms. The following figure shows the impact of the input C_n^2 profile on the performance of the ATLAS LTAO instrument concept for the E-ELT. Depending on the position of the LGSs, and the number of simulated layers, the performance of the AO system varies drastically. This shows the impact of using the proper input to model the tomographic AO systems for ELTs to describe its performance.

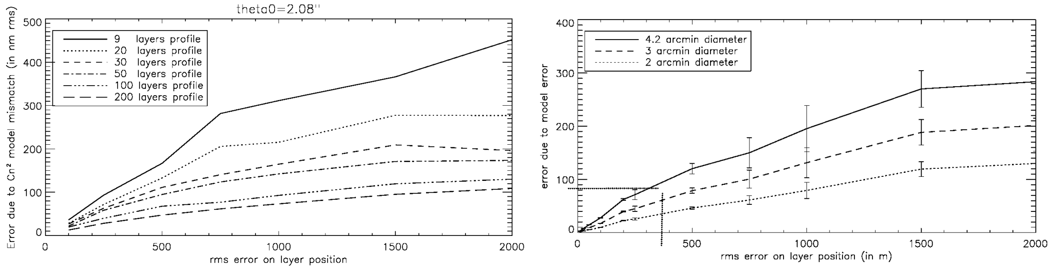


Figure 8: Performance of the ATLAS LTAO system, for different input C_n^2 turbulence profiles, and different positions of the Laser Guide Stars. Courtesy the ATLAS team.

ELTs are also very sensitive to the outer scale of turbulence, L0. Therefore, a better understanding of this parameters is required. Its height distribution (including inside the dome), the turbulence power spectrum itself (von Karman or something else) and seasonal variability are largely unknown, at the scales relevant to 20-40m telescopes. Also, the temporal statistics are very limited, since the L0 measurement campaigns have rarely been more than ~2 weeks long.

Another sensitive parameter for AO on ELTs is the sodium layer. As shown by the results below, high spatial and temporal resolution profiles are needed to make accurate simulations, since the impact of the shape of the sodium layer has a strong impact on the required low order measurements. However, this means a large telescope is required, which is not necessarily the case on the sites selected for the ELTs. This means that profiles from the wrong latitude is likely to be used in the simulations.

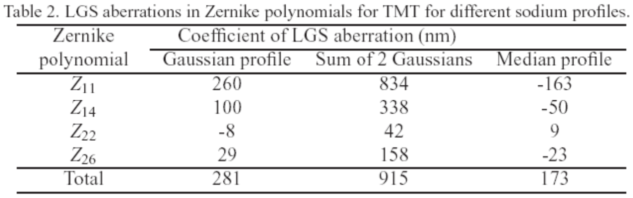


Table 2. LGS aberrations in Zernike polynomials for TMT for different sodium profiles.

Zernike polynomial	Coefficient of LGS aberration (nm)		
	Gaussian profile	Sum of 2 Gaussians	Median profile
Z_{11}	260	834	-163
Z_{14}	100	338	-50
Z_{22}	-8	42	9
Z_{26}	29	158	-23
Total	281	915	173

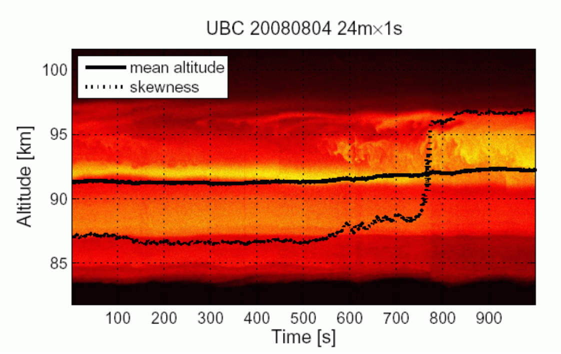


Figure 9: On the left, impact of sodium layer variations on the TMT AO system performance, Clare et al, Optics express, 15, 2007. Illustration courtesy R. Clare. On the right, UBC LIDAR data, Lardiere et al, Appl. Opt, submitted

Finally, the telescope environment is going to be much more complex than in the previous generation of telescopes, especially with adaptive telescopes (like the VLT AO upgrade, LBT, GMT). These telescopes certainly provide a more challenging environment to simulate that the current generation of simulations.

One other assumption which may need verification is the Taylor hypothesis, that most simulations use to simulate the temporal behavior of atmospheric turbulence. Poyneer et al. [21] have shown that ~40% of the turbulence is concentrated in Taylor type flows. This means that most turbulence is not. While this probably has little influence on most AO simulations, those dealing the predictive control should pay attention to this finding.

Most of these parameters are relevant to AO simulations, but also to the performance of the telescope once in operation. Therefore, it makes sense to set up measurement campaigns as soon as possible, since their input will be valuable to better understand the telescope's AO performance.

6. VALIDATING SIMULATIONS

Since simulations are becoming more and more complex with increasing number of modes, the question of how they can be validated arises.

6.1 Against other simulations

The first approach is to compare the simulations against other simulations. This can be very effective, as shown by for example [22]. Obviously, this does not insure that all the proper physical phenomena have been taken into account (all simulations could potentially miss an important effect). However, it is an important part in the development of evry simulation.

6.2 In the lab

The second approach is to set up a lab experiment. In this case, an AO "bench" is made, which simulates as closely as possible the conditions the AO system will encounter in reality. For example, the atmosphere is "simulated" in the lab with moving phase plates. There is a DM, a WFS etc. The advantage is that the results are reproducible (the phase plates stay the same, and one can "replay" them), and the experiment can be simplified compared to a real instrument (simplified hardware can be used, with slower framerates, no need to deal with a telescope environment, no need to automate as many functions etc). Also, this lab experiment can be designed with specific points to verify – and so it doesn't need to be "complete". For example, in the HOMER bench of ONERA, although tomography can be simulated, there is only one WFS, used in sequence to observe different guide stars. This simplifies the setup compared to a true tomographic system, without compromising the meaningfulness of the experiment. In the following figure (on the left), results from the HOMER bench are shown, comparing the lab experiment and simulations. We can see that the agreement is quite good, demonstrating that both the simulation and experiment are meaningful and "compatible".

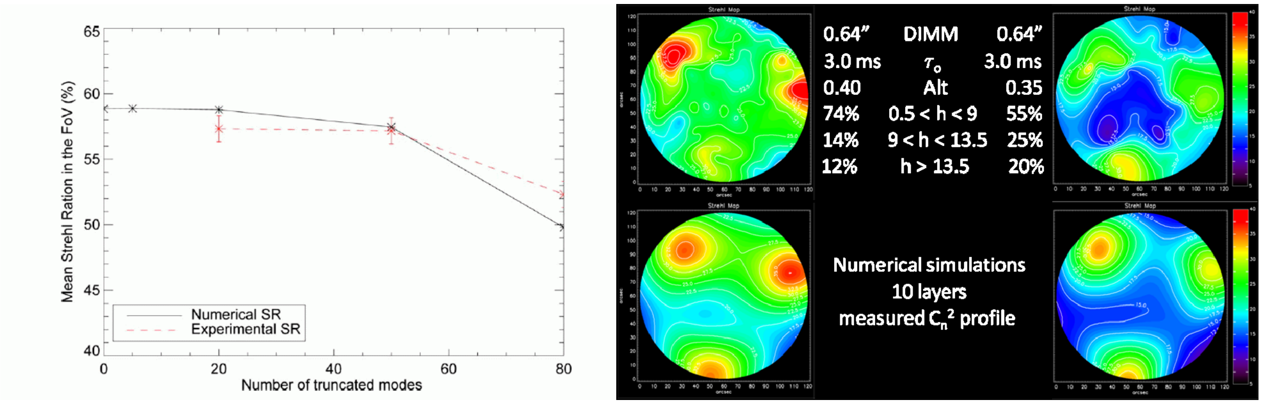


Figure 10: On the left, results from the HOMER bench for tomography, see [23]. On the right, MAD results in the sky and in the lab : Sky, bottom: simulated result, see [24] - Illustration courtesy E. Marchetti.

One drawback of this approach is that some "lab specific" biases are likely to be present – one example among many possibilities is that the phase screen are not exactly Kolmogorov, are too few or have a limited dynamic range, or not the proper chromaticity.

6.3 On the sky

A more complete way to validate simulation results is to build a "demonstrator" and to put it first in the lab in a controlled environment, to understand that it works as it should. Then, after an extensive testing and validation in the lab, it is put on a telescope. This has the advantage to have a validated experiment, and to be able to know that the departures from lab results are either due to the telescope environment or to the atmosphere. This is what was done with the MAD experiment of ESO [24]. The figure above (on the right) shows a comparison between simulation results and a result on the sky. As far as possible, the simulation was made with the same assumptions as the sky results. Although the agreement is good, this figure shows the difficulty to accurately model what happens in a real instrument. In the lab, the agreement between simulation and MAD results was better, since the environment was better controlled and understood.

So what are the difficulties in modeling an on-the-sky AO instrument? First, the atmospheric conditions are complex and highly variable. The seeing changes rapidly (in simulations, usually the r0 is assumed constant), as does the wind and C_n^2 profiles – which is particularly important for an MCAO system like MAD. Even if both parameters are monitored in real time, difficulties remain. These quantities are measured roughly every minute, a time already much longer than a typical AO simulation (a few seconds), but much shorter than the science exposure (several minutes usually). Secondly, the observations are usually not done in the same direction in the sky as the observation, nor are they done from the same location (for example, meteo station vs. main telescope). Both of these can introduce a bias in the simulation input. Also, telescope vibrations, effects of field stabilization can very difficult to identify and model. Of course, some of these effects could be removed by using the AO telemetry data itself to derive the necessary input parameters for the simulations. But this complexifies the demonstrator / instrument.

One way to quantify more finely how the system behaves compared to simulations is a statistical approach, which probably smoothes out some of the effects described above. Instead of modeling an instant in time, one can try to estimate "average" conditions and simulate those, and see how close that simulation is to the "average" observations obtained on sky. This would be a test of the "availability" of the instrument, which we describe in the next section.

7. NEW APPROACH TO PERFORMANCE ESTIMATION

Because of the high variability of C_n^2 profiles and their impact (especially the ground layer) has on GLAO performance, a novel approach was tested on several instruments of ESO's future AOF. Instead of running a few simulations with "good", "median" and "bad" seeing profiles, it was decided to first make a statistical analysis of the site survey instruments in Paranal. These provide a large database of seeing values (measured with a DIMM), and C_n^2 profiles (measured with a MASS). The data was analyzed, to provide a "good" (25% best profiles for GLAO), "median" and "bad" (25% worst for GLAO) C_n^2 profile for 6 bins of seeing (from better than 0.4'' to worse than 1.2''), So for each seeing value (whose occurrence of probability is known), 3 profiles of also known probability was associated. Then, AO simulations were carried out for each profile and each seeing. This allowed to calculate the "availability" of this AO system, i.e to be able to say, what fraction of the time the AO system would perform better than a given criterion (here FWHM, for GRAAL). The result is summarized below:

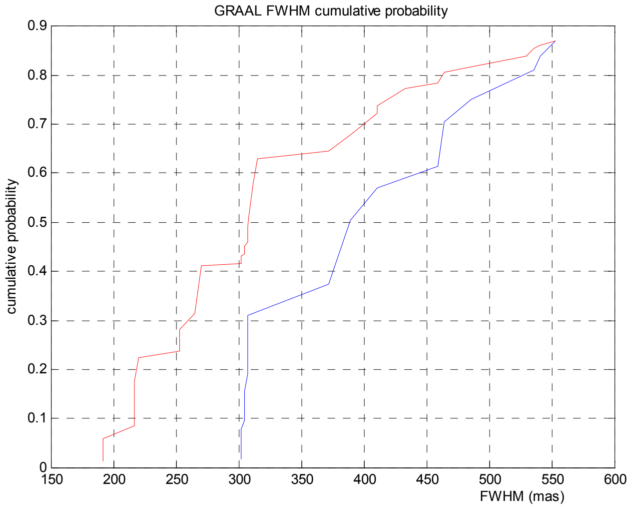


Figure 11: cumulative probability of occurrence of a FWHM at the telescope focus, in K-band. Blue: operation before AOF, red: AOF operation, taking into account 25% unavailability of the AOF (i.e. operating in noAo mode 25% of the time). See [25] for more details. Illustration courtesy of J. Paufique.

We can see that such an approach is a good way to test the agreement between simulation and reality. By comparing the above plot obtained through simulations and an observed curve at the telescope (obtained for example through a quality control approach of the science instrument), discrepancies between the two approaches can be identified. Finding the cause of such discrepancies might however be complex to identify.

8. SIMULATION TRENDS – CONCLUSION

In this paper, we have investigated the current trends in AO simulations. The most salient points are now summarized.

There is a trend of increasing complexity in the simulations, to follow the systems which themselves become more and more complex. This is driven both by more AO modes to address more specialized needs (GLAO, and tomographic systems) and the preparation for the ELTs.

To cope with the increasing complexity, an effort to improve performance of simulation tools has been made. One path has been the creation of new analytical / semi analytical tools to explore large parameter spaces. The other has been the upgrade (or re-writing) of end to end simulation tools. Simulation tools developers want to increase simulation speeds, but want to take better advantage of new computers which tend to have multiple CPUs and multiples cores per CPU. Since the speed of single cores seems to more or less stagnate, this trend is likely to continue. Therefore, parallelization seems like a likely candidate for the development of future end to end AO simulation tools.

The other field where a lot of activity has taken place and is likely to stay active, is the development of new algorithms (especially for wavefront reconstruction). This is driven by the will to reduce the computational load of real time computers, but also the time needed to simulate AO systems.

It is also a trend, which will probably continue, to simulate systems more deeply. Part of this is explained by the complexity of the systems – so more effects need to be investigated. However, another driver is the increased cost of the instruments, so designers need more simulations to back-up their findings – since an increased cost is likely to bring more scrutiny on the design of AO systems. In the future, every bit of the system is likely to be simulated, when previously, for cheaper systems, experience was more used to discard the importance of some error sources.

Some new applications of AO systems are also emerging. In the previous sections, examples of AO simulations used for astronomical observation simulation were shown. In addition, two other novel approaches can be mentioned:

- Testing of RTCs: It has been suggested to use an AO simulation tool to feed slopes to an RTC. The RTC then computes commands, which are then sent back to the simulation to loop. This could even happen in quasi real time for small systems. The advantage is that one can thoroughly test the RTC (and why not other components like the deformable mirror) in "real" conditions, where the values sent to the RTC (and DM) are realistic. See [26]
- Another interesting use of an AO simulation tool is that presented in [27]. The authors use the yao AO simulation tool to control the MCAO AO instrument at the telescope. This provides and easy switch between reality & simulations, allowing to more easily debug the system, should something not function properly. Testing of the whole system is also improved, since should components not be ready, the systems tests can be carried out using a simulation module.

These examples show that in addition to the "normal" performance estimation uses of AO simulations, more creative ideas can extend their usefulness.

9. ACKNOWLEDGEMENTS

The author would like to thank the following persons for helping gathering the results shown here to describe various aspects of AO simulations, sometimes in advance of publications: Alastair Basden, Marcel Carbillet, Jean-Marc Conan, Thierry Fusco, , Luc Gilles, Enrico Marchetti, Jerome Paufique, Marcos van Dam.

REFERENCES

[1] Neichel et al., Tomographic reconstruction for wide-field adaptive optics systems: Fourier domain analysis and fundamental limitations ",JOSA A, Vol. 26 Issue 1, pp.219-235 (2009)
[2] Jolissaint et al., "Analytical modeling of adaptive optics: foundations of the phase spatial power spectrum approach "JOSA A, Vol. 23, Issue 2, pp. 382-394 (2006)
[3] Ellerbroek, "Linear systems modeling of adaptive optics in the spatial-frequency domain ", JOSA A, Vol. 22, Issue 2, pp. 310-322 (2005)
[4] Fusco et al.,"ATLAS: the LTAO system for the E-ELT: design, performance, and sky coverage", these proceedings, 7736-11.
[5] Tokovinin, "Seeing Improvement with Ground-Layer Adaptive Optics", PASP, 116, 2004
[6] Tokovinin et al., "Optimized modal tomography in adaptive optics", A&A, 378, 2001
[7] Tokovinin & Viard, "Limiting precision of tomographic phase estimation", JOSA A, 18, 2001
[8] Tokovinin et al., "Isoplanatism in a multiconjugate adaptive optics system", JOSA, 17, 2000
[9] Conan et al., MAORY study, 2010
[10] Rigaut, http://www.maumae.net/yao/
[11] Gilles et al. , "Modelling update for the TMT LGS MCAO system", these proceedings, 7736-31
[12] Petit et al., "Wide-field AO systems on Extremely Large Telescope: analysis of tomographic reconstruction based on rescaled end-to-end simulation tools" these proceedings, 7736-18
[13] Bechet et al., "Optimal reconstruction for closed-loop ground-layer adaptive optics with elongated spots", JOSA A, Vol. 27, Issue 11, pp. A1-A8 (2010)
[14] Wang et al., "Impact of laser guide fratricide on TMT MCAO system", these proceedings, 7736-16
[15] Ellerbroek and Vogel, "Inverse Problems in Astronomical Adaptive Optics" (Topical Review Article), Inverse Problems, 063001, Vol. 25, No. 6, June 2009
[16] Montilla et al.," Performance comparison of wavefront reconstruction and control algorithms for Extremely Large Telescopes", JOSA A, Vol. 27, Issue 11, pp. A9-A18 (2010)
[17] Carbillet, The CAOS problem-solving environment: recent developments, these proceedings, 7736-152
[18] Gilles et al., "Split atmospheric tomography for multiconjugate and multi-object adaptive Optics", these proceedings, 7736-31
[19] Fusco & Conan, "Impact of C_n^2 profile on ELT wide-field AO performance", these proceedings, 7736-30
[20] Gilles & Ellerbroek, "Real-time turbulence profile estimation from closed-loop laser guide star wavefront sensor measurements", these proceedings, 7736-143
[21] Poyneer et al, "Experimental verification of the frozen flow atmospheric turbulence assumption with use of astronomical adaptive optics telemetry", JOSA A, Vol. 26, Issue 4, pp. 833-846 (2009)
[22] Rigaut, Ellerbroek & Northcott, "Comparison of curvature-based and Shack-Hartmann-based adaptive optics for the Gemini telescope", Appl. Opt, **36**, 1997
[23] Costille et al, "Wide field adaptive optics laboratory demonstration with closed-loop tomographic control", JOSA-A, **27**, 2010

[24] Marchetti et al., “Lessons from MAD: On-Sky Performance Testing”, Proc. MAD and beyond, ESO, 2009.
[25] J. Paufique, “GRAAL: a seeing enhancer for the NIR wide-field imager Hawk-I” these proceedings, 7736-60
[26] Basden et al., “A COTS high-performance real-time control system for adaptive optics”, these proceedings, 7736-172
[27] Rigaut et al., “MYST: a comprehensive high-level control tool for GeMS”, these proceedings,7736-87
[28] van Dam et al., “Modeling the AO systems on the GMT”, these proceedings, 7736-150
[29] Le Louarn et al, “Parallel simulation tools for AO on ELTs”, Proc. SPIE, Vol. 5490, 705, 2004