Analysis tools for numerical simulations of Dynamic Aperture with XSUITE

T. Pugnat*[1], D. Di Croce[2], M. Giovannozzi[1], F. F. Van der Veken[1]

[1]CERN, Geneva, Switzerland, [2]EPFL, Lausanne, Switzerland

Abstract: Recently, several efforts have been made at CERN to develop a new tracking tool, `Xsuite`, which is intended to be the successor to `SixTrack`. In this framework, analysis tools have also been prepared with the goal of providing advanced post-processing techniques for the interpretation of dynamic aperture simulations. The proposed software suite, named `Xdyna`, is meant to be a successor to the existing environment `SixDesk`. It incorporates all recent approaches developed to determine the dynamic aperture for a fixed number of turns. It also enables studying the time evolution of the dynamic aperture and the fitting of rigorous models based on the stability-time estimate provided by the Nekhoroshev theorem. These models make it possible to link the dynamic aperture to beam lifetime, and thus provide very relevant information for the actual performance of particle colliders. These tools have been applied to studies related to the luminosity upgrade of the CERN Large Hadron Collider (HL-LHC), the results of which are presented here.

Brief description of `Xdyna`

1 *Creation of a study:*

For each study, a configuration file is created which contains all the information about the working units such as:

- the study name,
- the normalised emittances,
- the turn number used,
- the number of realisations of the magnetic field errors (seeds),
- the MAD-X scripts if needed.

2 *Generate a distribution of particles:*

`Xdyna` can generate different types of initial distribution of particles:

- Cartesian grid,
- Polar grid (similar to `SixTrack`),
- Random grid.

`Xdyna` also manages pairs of particles. If needed, it is possible to load a costume distribution of initial positions. The post-processing will then be similar to the random grid distribution one.

3 *Tracking and post-processing:*

Tracking can be performed in parallel on `HTCondor`. A number of particles per session needs to be specified, and `Xdyna` automatically manages the seeds. In future development, `Xdyna` will also be able to run simulations on a new version of the volunteer-based computing system BOINC.

Post-processing tools have been implemented to detect the stability border in the (x, y) space. Two borders are estimated. For each of them, their minimum / maximum / average is calculated throughout the angle, as well as for each seed:

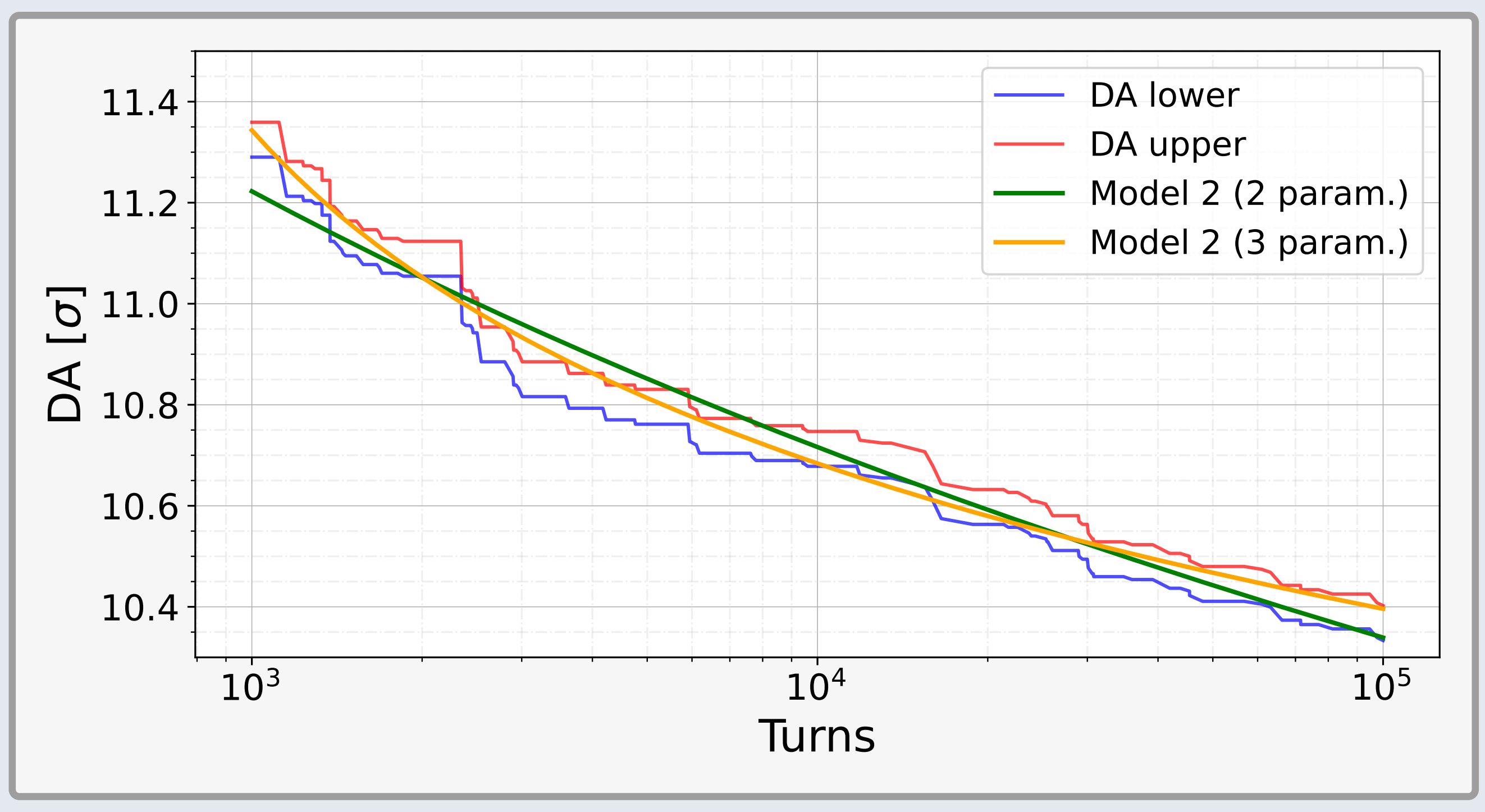
- *Lower estimation* ($\mathrm{DA_l}$): the border of the connected part of the stability domain, corresponding to what is calculated by `SixDesk`,
- *Upper estimation* ($\mathrm{DA_u}$): the lower border of the unstable region.

4 *DA vs turn:*

A new feature added to `Xdyna` is the possibility to calculate the evolution of the DA over time and fit it with different models. Some have already been built-in from the Nekhoroshev theorem.

Example of fitted DA vs Turns

The region delimited by $\mathrm{DA_l}$ and $\mathrm{DA_u}$ is clearly visible, with the DA curves for the 2- or 3-parameter fit of Model $\mathrm{DA_2}$ obtained well adjusted within the acceptable region.



DA versus turn and fit results for Model $\mathrm{DA_2}$ using 2 and 3 parameters, respectively, using a Gaussian random generator.

Tools for DA studies in `Xdyna`

DA vs Turns models:

Two models for the stability time estimate have been implemented from Nekhoroshev theorem:

$$\mathrm{DA}_2 = \rho\left(\frac{\kappa}{2e\ln(N/N_0)}\right)^{\kappa}$$

$$\mathrm{DA}_4 = \frac{\rho}{\left(-e\,\mathcal{W}_{-1}\left(\frac{-1}{e}\;\left(\frac{\rho}{6}\right)^{1/\kappa}\;\left(\frac{8N}{7}\right)^{-2/\kappa}\right)\right)^{\kappa}}$$

with:

- $\mathcal{W}_{-1}$, the negative branch of the Lambert function;
- ρ and N_0, scaling parameters respectively for DA and turns;
- κ, the curve shape.

DA vs Turns fitting procedure:

The fitting procedure is able to fit: $\mathrm{DA_l}$, $\mathrm{DA_u}$, the average, a Uniform or Gaussian distribution of point between both values. A regression with 1, 2, or 3 parameters follows a least-square minimisation. Boundaries have to be provided because of the strongly non-linearity of the models and the parameter interdependence.

Parameter boundaries and default values:

- $\rho > 0$
- $\kappa \in [0.01, 2]$
- $N_0 \in]0, N_{\mathrm{max}}]$

If not used, ρ, κ or $N_0 = 1$.

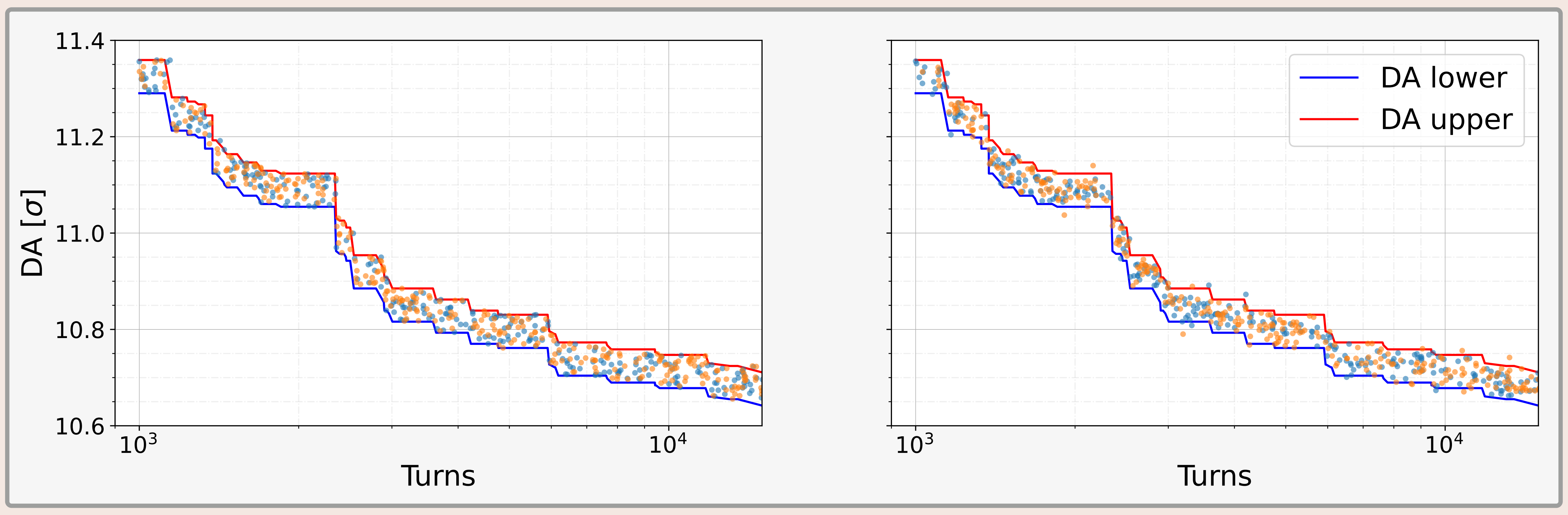
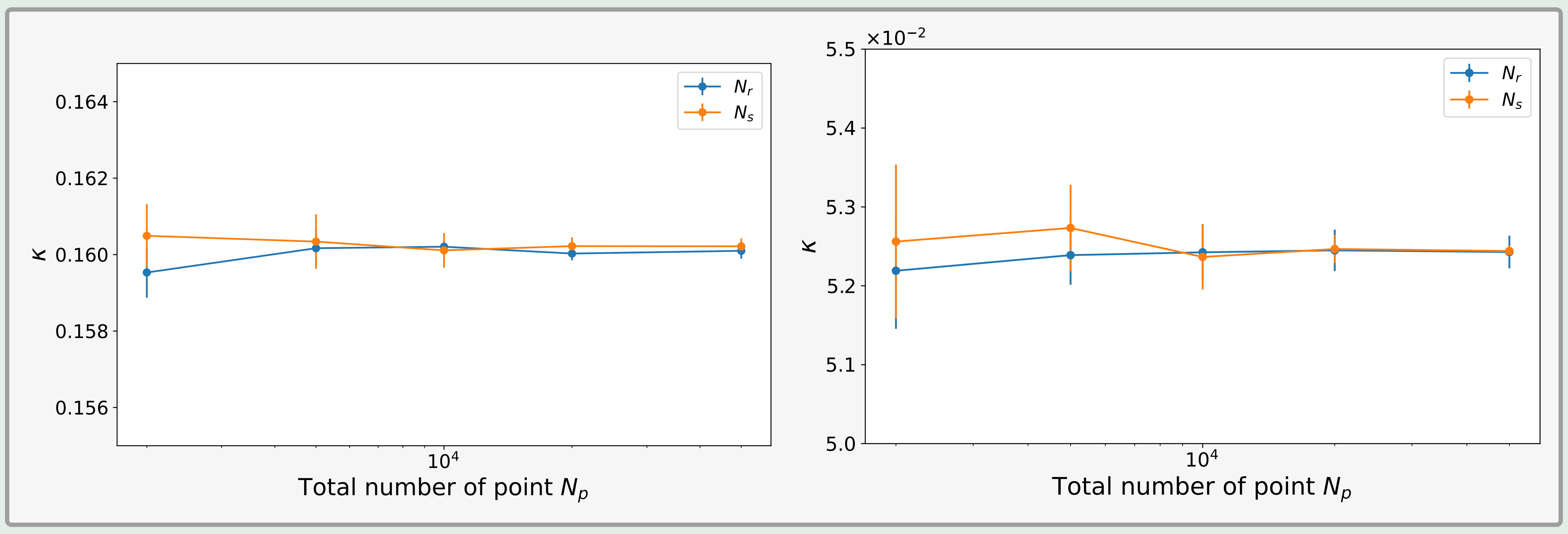


Illustration of the randomly generated set of points used for the fit of the DA vs turn. Continuous curves represent DA_l (blue) and DA_u (red), and uniform and Gaussian random generators are shown in the plots on the left and right, respectively.

Precision of the fitted parameters using a Gaussian distribution of points

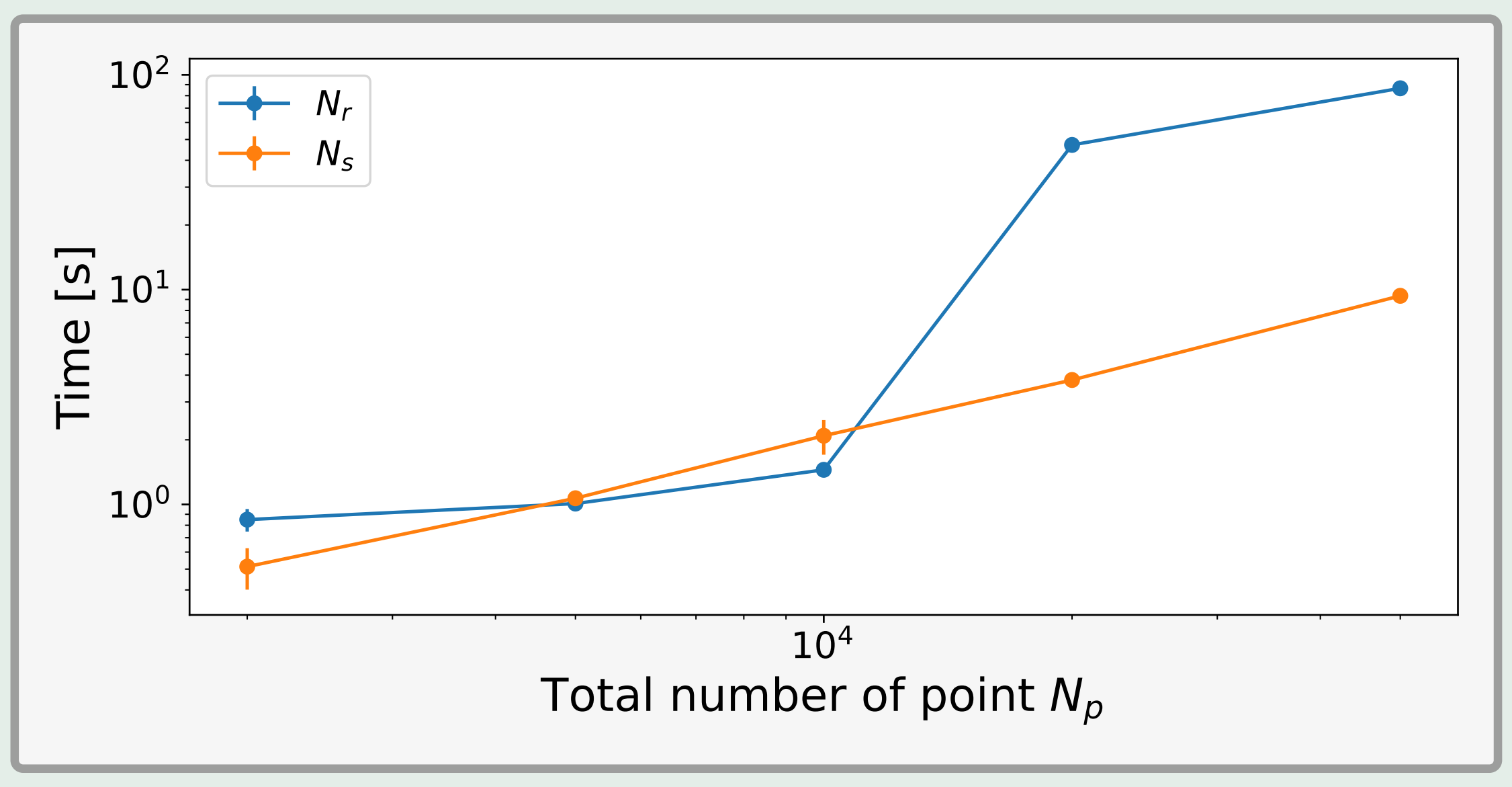


Dependence of κ on N_p when 2 (left) or 3 (right) parameters are used for fitting $\mathrm{DA_2}$. A Gaussian random generator has been used. The two curves represent separate scans of N_s or N_r.

Using the Uniform or Gaussian distribution of points for the DA vs Turns fitting, it is possible to specify the number of samples (N_s) and the number of randomly generated points (N_r). But the most useful quantity to study the accuracy of the fit and the CPU time is the total number of points ($N_\mathrm{p} = N_\mathrm{r} \times N_\mathrm{s}$).

$N_\mathrm{p} \approx 10^3$ is already enough to ensure that κ has settled to the correct value and that the associated error is at the percent level. Note the large difference in the value of κ depending on the number of free parameters used in the model. This feature is well known for this type of fit function and suggests that two free parameters should be used to avoid interplay between the fit parameters.

The dependence of the CPU time on N_p is represented by a power law. The sudden jump when N_r vary could be related to the behaviour of the routine as a function of N_p.



CPU time of the fit procedure as a function of N_p. The two curves represent separate scans of N_s or N_r.

[*] thomas.pugnat@cern.ch