DEVELOPMENT OF COMPUTATIONAL TOOLS FOR NOISE STUDIES IN THE LHC

S. Kostoglou*, CERN, Geneva, Switzerland and NTUA, Athens, Greece
N. Karastathis, Y. Papaphilippou, D. Pellegrini, CERN, Geneva, Switzerland
P. Zisopoulos, CERN, Geneva, Switzerland and Uppsala University, Sweden

Abstract

Noise can have a significant impact on the beam dynamics in the LHC, enhancing diffusion processes and leading to emittance blowup. In order to study the details of such effects with computer simulations, a new set of tools is being developed. In particular, a demonstrator GPU-based particle tracker has been built profiting from the technology provided by the NVRTC Cuda library. The performance of the tracker for short beam dynamic simulations, in the presence of many macroparticles, is highly promising. In addition, refined Fourier analysis has been performed on the tracking data by using the Numerical Analysis of Fundamental Frequencies (NAFF) algorithm. After thorough inspection, several alternatives to its fundamental steps have been tested in a modern C++ implementation. The method was also used to produce frequency maps and benchmark these tools with other simulations.

INTRODUCTION

Several sources of external noise in particle colliders such as ground motion, power supply ripple, Crab Cavities RF ripples and transverse damper may have detrimental effects on the beam dynamics, leading to transverse emittance growth, halo population and losses by enhancing diffusion mechanisms, especially in presence of strong non-linearities and beam-beam effects [1–3]. Such mechanisms, which are not yet fully understood, result in a reduction of the machine performance in terms of delivered luminosity.

We aim to study the impact of noise with beam-beam effect in the LHC, adopting the weak-strong approximation in which only a single beam is tracked while the beam-beam lenses (both for the head-on and long-range interactions) are static. This approximation is well suited for particles with an action of a few sigma, whose dynamics determines both the tail evolution and the losses, without being highly influenced by the coherent motion of the beam core. Furthermore, such treatment of the beam-beam effect enables the independent tracking of each macroparticle in a distribution and therefore, the simulations can be performed in parallel.

To this end, we choose to explore the capabilities of Graphic Processing Units (GPU) for particle tracking purposes. Nowadays, these devices present thousands of computing cores, which can be exploited for parallelisation. A technology demonstrator has been built in C++ profiting from the CUDA NVRTC library which allows just-in-time compilation of the machine lattice into GPU code [4]. This approach aims at maximizing the execution speed and the results are very promising.

For the data analysis in presence of noise, the NAFF method of Laskar has been reviewed [5, 6]. A new implementation of the algorithm has been written in modern C++ and interfaced to Python. In this way, the steps of the algorithm have been inspected and several alternatives have been identified and tested. An overview of these tools is given in the next sections.

ALGORITHM DESCRIPTION

The NAFF algorithm provides a quasi-periodic approximation of a complex signal $\psi(t)$ over a time interval $[0, T]$. Compared to the Fast Fourier Transform (FFT) it allows for a more accurate determination of both the frequencies and the Fourier coefficients of the signal with a much faster convergence [7].

The first step of the NAFF algorithm consists of an FFT, which provides a rough indication of the locations of the main spectral components; for this we rely on the FFTW library [8]. In order to refine the determination of the frequency along with its complex amplitude, the Fourier integral is introduced:

$$\phi(f) = \langle \psi(t)\,,\ e^{i2\pi f} \rangle = \frac{1}{T}\int_0^T \psi(t)\, e^{-i2\pi f t}\ x(t)\, dt \quad (1)$$

where $x(t)$ denotes a window function.

Windows are used to reduce the leakage effect which occurs as a result of processing finite-duration signals [9]. Figure 1 illustrates the Fourier integral with a rectangular window ($x(t) = 1$) for two signals composed by one and two frequencies respectively. The appearance of side lobes is caused by the boundary discontinuities of the truncated signal multiplied with the rectangular window, which is assumed to be periodic from the Fourier integral. This effect leads to frequency and amplitude displacement.

Three types of window functions were tested: Taylor, Dolph-Chebyshev and Hann window. All the windows exhibited similar results in terms of accuracy and convergence. In this paper, we focus on the Hann window which can be defined as [10]:

$$x_h(t) = \frac{2^h (h!)^2 (1 + \cos \pi t)^h}{(2h)!} \quad (2)$$

where h is the power of the window, allowing to tune the ratio between side lobe attenuation and main-lobe width. By increasing h the side lobe level is significantly reduced at

* sofia.kostoglou@cern.ch

ISBN 978-3-95450-182-3

Copyright © 2017 CC-BY-3.0 and by the respective authors

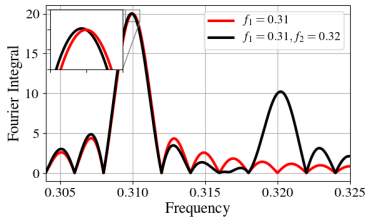

Figure 1: A signal with one (red) and two (black) frequencies. The main component f_1 is equal in both cases, however in the second one a shift in amplitude and frequency is observed due the interaction with the side lobes of the second frequency f_2.

the expense of a wider main lobe. By using this window, the shift in amplitude or/and frequency is suppressed.

The NAFF algorithm proceeds to the determination of the refined spectral component by maximising the Fourier integral. Other merit functions were also examined.

The first alternative method is based on the minimization of the residual of the Fourier integral. The refined frequency is the one which, when subtracted, results in the minimum area of the residual frequency-domain signal, computed in an interval of width 2δ around the FFT peak. This merit function can be defined as:

$$\phi(f) = \int_{f_{FFT}-\delta}^{f_{FFT}+\delta} \langle \psi_f(t),\ e^{i2\pi f'} \rangle df' \tag{3}$$

with

$$\psi_f(t) = \psi(t) - \mathrm{proj}_{u_f}(\psi(t)) \tag{4}$$

where $u_f = e^{i2\pi f}$ for the first frequency component and its amplitude is calculated by the projection on the signal:

$$\mathrm{proj}_{u_f}(\psi(t)) = \frac{\langle \psi(t), u_f \rangle}{\langle u_f, u_f \rangle} u_f \tag{5}$$

The integration requires several evaluations of the Fourier integral in the $f_{FFT} \pm \delta$ interval and therefore more computational power is required. However, this method proved to be more numerically accurate when tested with simulation data.

The last merit function concerns the minimization of the residual energy in time domain. The removal of the accurate frequency component results in a minimum residual signal in time domain. In this case, the merit functions is defined as:

$$\phi(f) = \sum_{t=1}^{N} (\psi_f(t))^2 \tag{6}$$

It should be noted that this merit function does not enable the use of the window, as it does not involve computations of the Fourier integral where the window function is included, thus it is not well suited for signals with strong frequency crosstalk.

The maximization of the merit functions is based on the Brent algorithm and we use the implementation available in Boost [11,12]. The refined frequency is then subtracted from the signal and the search is iterated for additional spectral components.

From the second frequency onwards, the new components u_i must be orthogonalised to the previous ones before the subtraction. We implemented the modified Gram-Schmidt method, in order to construct orthonormal basis functions:

$$u_i^{\perp} = u_i - \sum_{j=1}^{i-1} \mathrm{proj}_{u_j^{\perp}}(u_i) \tag{7}$$

The procedure is repeated until no relevant spectral components are present in the signal.

The numerical integrations employ Hardy's 7-point integration rule [5]. Furthermore, we foresee the possibility of up-sampling the initial signal either with a linear interpolation or with a cubic spline interpolation. The following results were obtained with an up-sampling by a factor of 10 with the cubic spline interpolation.

SIMULATION RESULTS

Ad-hoc built signals were used in order to investigate the behavior of the Hann window in two cases: In the case of frequency crosstalk, when the lobes of two components are overlapping and in the case of low SNR (Signal to Noise Ratio).

In the first case, we use a cosine signal with two frequency components $\psi(t) = \alpha\cos(2\pi 0.3t) + \beta\cos(2\pi 0.4t)$ with $t \in [0, N]$ where N is the number of samples and α, β real constants with $\beta > \alpha$. Figure 2 shows the evolution of the determination of the frequency for an increasing number of points and with different orders of the Hann window. The results are also compared with the case where only a rectangular window is applied. Figure 2 indicates that, for an incrementing number of turns, $h = 5$ shows the best convergence. This outcome is in agreement with the theoretical result obtained by Laskar for the NAFF algorithm implicating that the convergence asymptoticly scales with $1/T^{2h+2}$ [7]. Figure 2 also highlights a significant difference in accuracy with the rectangular widow. It should be noted that with less separation between the frequencies, higher order windows show less accurate results due their broader main lobe which increase the crosstalk.

The effect of noise on the efficiency of the windows was investigated by introducing white noise in the signal with an SNR of 9.7 dB. Based on Fig. 3, it can be seen that the best convergence is achieved with windows of lower powers. A possible explanation can be attributed to the fact that these types of windows have a more narrow main-lobe and therefore, they prove to be less susceptible to noise effects. As shown in Fig. 4, due to their wider main lobe, higher powers are more sensitive to noise, causing a displacement of the Fourier integral, which has an impact both in frequency and amplitude determination.

Copyright © 2017 CC-BY-3.0 and by the respective authors

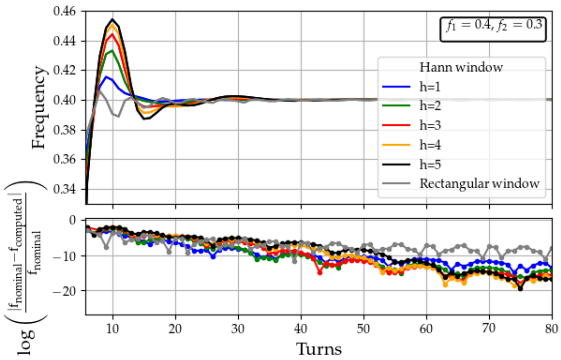

Figure 2: The convergence of different orders of the Hann window. $f_{nominal}$ is equal to the main frequency component of the simulation data $f_1 = 0.4$ and $f_{computed}$ is the frequency calculated for the specified number of turns.

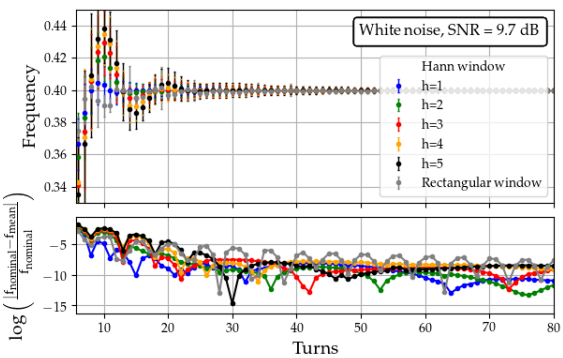

Figure 3: Convergence of different orders of the Hann window in the presence of white noise. $f_{nominal}$ is equal to $f_1 = 0.4$, f_{mean} is the mean value of the computed frequencies for 100 different seeds of the white noise random generator and the error bars represent one standard deviation.

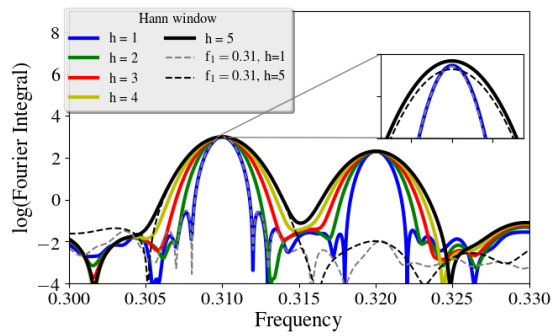

Figure 4: The impact of white noise (SNR=9.7 dB) in the efficiency of the different powers of the Hann window. The solid lines represent signals with two components, while dashed lines include only $f_1 = 0.31$.

LHC RESULTS

We tested the proposed enhancements using real data from a beam position monitor of the LHC with a 2 mm peak-to-peak amplitude. We compared the original NAFF consisting of the maximization of the Fourier integral and a Hann window $h = 1$ to the different merit functions, including up-sampling. The results shown in Fig. 5 indicate that up-sampling appears to be particularly relevant for less than ~ 40 turns. In all cases, good accuracy in the frequency determination is achieved after a few turns.

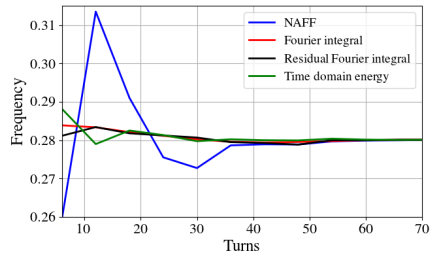

Figure 5: Comparison of the NAFF algorithm (blue), which does not include up-sampling, with the proposed improvements.

GPU TRACKING DATA

The complete LHC lattice in collision, without beam-beam, was imported into the GPU tracker demonstrator. 4×10^4 particles with initial coordinates covering a grid in the (x, y) configuration space up to 6σ, were tracked for 10^4 turns. The turn-by-turn data have been post-processed with the NAFF implementation. Diffusive frequency maps as shown in Fig. 6 have been computed by determining the tunes from the tracking data divided into two equal and consecutive time intervals [13]. The two tune determinations for each particle are compared, the color code representing their variation. The resolution of the frequency map provides detailed resonant lines both in the frequency and in the configuration space with a clear correlation to the diffusive processes. Thus, it shows the proof of concept for both the tracker and the NAFF implementation.

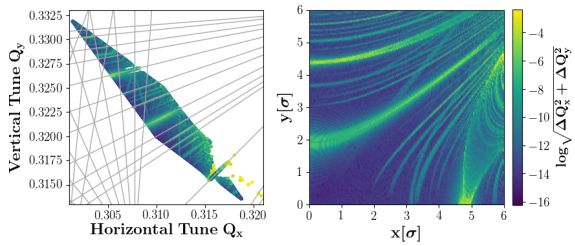

Figure 6: Frequency and amplitude maps from the tracking data of the GPU for the LHC lattice with h=3.

CONCLUSIONS

A new C++ implementation of the NAFF algorithm interfaced to Python is available. We examined the impact of frequency crosstalk and the presence of white noise in the signal on the response of several orders of the Hann window. We inspected potential improvements which proved to be promising in terms of convergence and accuracy for frequency determination. The code was used in combination

Copyright © 2017 CC-BY-3.0 and by the respective authors

ISBN 978-3-95450-182-3

with a GPU-based tracker demonstrator for the examination of the betatron motion in the LHC, proving the solidity of these tools which are being prepared for noise studies.

ACKNOWLEDGEMENTS

The author would like to acknowledge the European Physical Society for granting the EPS-AG/JUAS sponsorship in order to attend IPAC'17.

REFERENCES

[1] V. Lebedev *et al.*, "Emittance growth due to noise and its suppression with the feedback system in large hadron colliders" *Part. Accel.*, vol. 44, pp 147-164, 1992.

[2] P. Baudrenghien,T. Mastoridis "Transverse emittance growth due to RF noise in the high-luminosity LHC crab cavities" *Phys. Rev. ST Accel. Beams*, vol. 18, pp 101001, 2015.

[3] R.E Meller *et al.*, "Decoherence of kicked beams." *SSCN-360*, 1987.

[4] CUDA NVRTC Library, `http://docs.nvidia.com/cuda/nvrtc/`.

[5] J. Laskar, C. Froeschle and C. Celletti, "The measure of chaos by the numerical analysis of the fundamental frequencies. Application to the standard mapping", *Physica D*, vol. 56, pp 253-269, 1992.

[6] Y. Papaphilippou, "Detecting chaos in particle accelerators through the frequency map analysis method", *Chaos. An Interdisciplinary Journal of Nonlinear Science. Amer. Inst. Phys.*, vol. 24, pp 024412, 2014.

[7] J. Laskar "Introduction to Frequency Map Analysis", in *Hamiltonian Systems with Three or More Degrees of Freedom*, 1999, pp. 134–150.

[8] M. Frigo and S.G.. Johnson, "FFTW: An adaptive software architecture for the FFT", in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Seattle, Oct. 1994, vol. 3, pp. 1381–1384.

[9] A.V. Oppenheim, W.S. Ronald and R.B John, *Discrete-time signal processing.*, Printice Hall Inc, 1989.

[10] F.J. Harris "On the use of windows for harmonic analysis with the discrete Fourier transform", in *Proc. of the IEEE*, Jan. 1978, vol. 66, pp. 51-83.

[11] Boost C++ Libraries, `http://www.boost.org/`.

[12] R.P. Brent, *Algorithms for minimization without derivatives*, Courier Corporation, 2013.

[13] Y. Papaphilippou "Frequency maps of LHC models", in *Proc. Particle Accelerator Conf. (PAC'99)*, 1999, vol. 3, pp. 1554–1556.

Copyright © 2017 CC-BY-3.0 and by the respective authors