Finanziato dall'Unione europea NextGenerationEU

Ministero dell'Università e della Ricerca

Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA

ICSC Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

Search for orbital modulated periodic signals in radio timeseries: evaluating CPU vs GPU codes

Andrea Possenti & Rouhin Nag

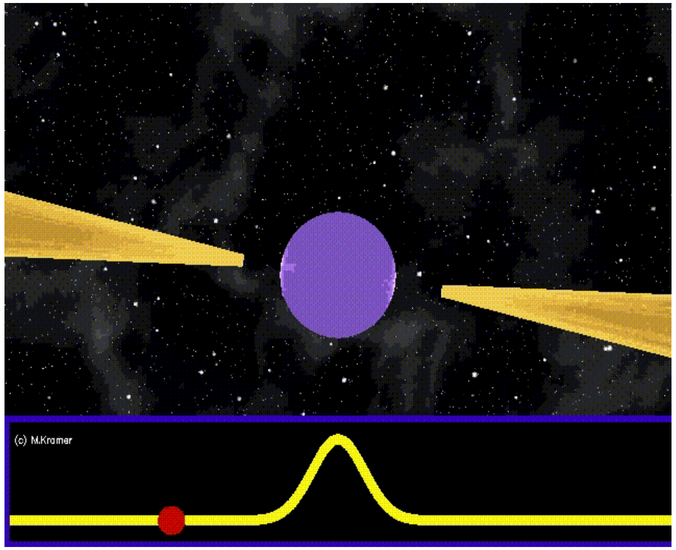
(INAF-Cagliari & Univ of Cagliari)

Spoke 3 II Technical Workshop, Bologna Dec 17 -19, 2024

Scientific Context

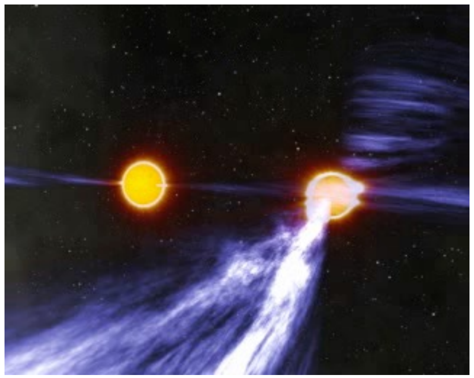
Pulsars are great tools to investigate:

Neutron star magnetosphere
Neutron star birth and population
Supernova explosion
Ionized Interstellar medium
Galactic magnetic field
Globular cluster potential well
Globular cluster internal dynamics
Ultralong period Gravitational Waves

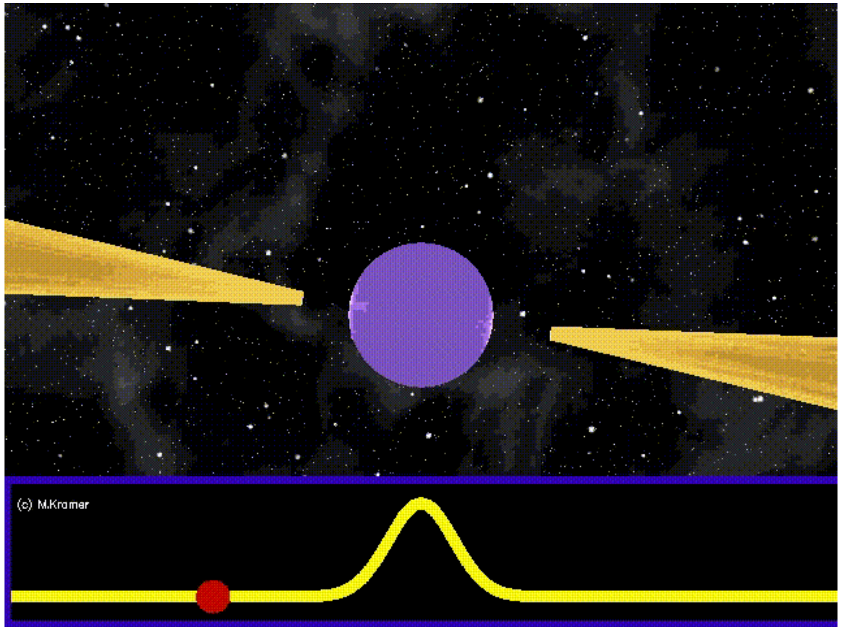


Binary pulsars in close orbital systems are magnificent tools to perform unique experiments in:

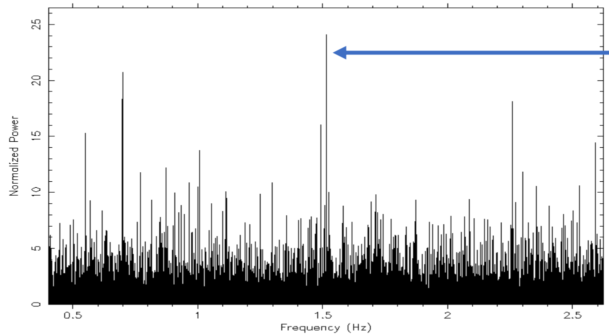
Tests of General Relativity
Study of alternate Gravity Theories
Measuring the mass of a Neutron Star
Constraining nuclear Physics
Study of Plasma Physics



How to search for new isolated pulsars

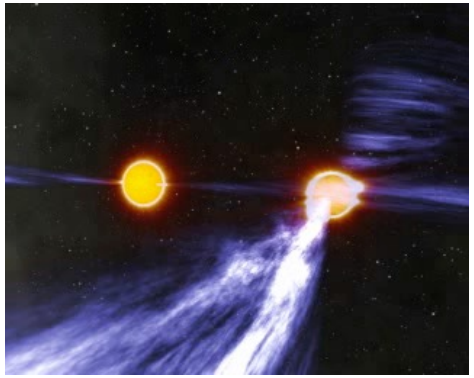


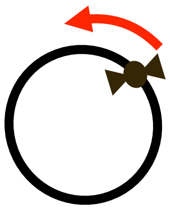
To be searched applying FFTs or Fast Folding Algorithms



Fourier power is very Peaked

How to search for new interesting objects: binary pulsars

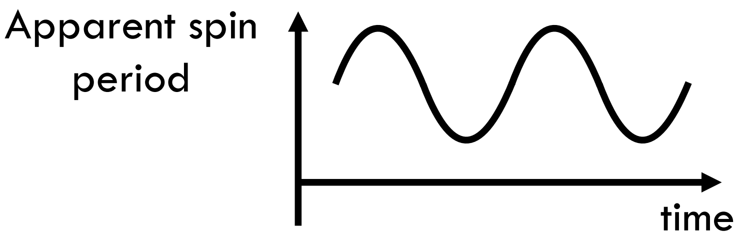


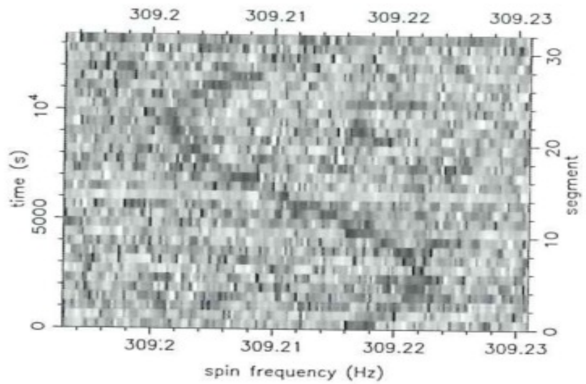


Equation for Doppler shift:

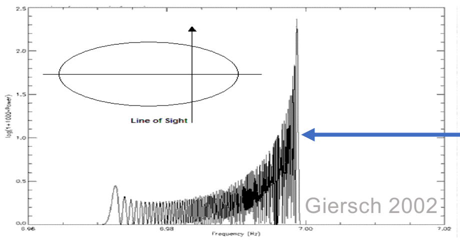
$$P'(t) = P_0(1 + v(t)/c)$$

Evolution of the period in circular orbit:





Direct FFTs is not possible due to the changing apparent spin period caused by the Doppler effect



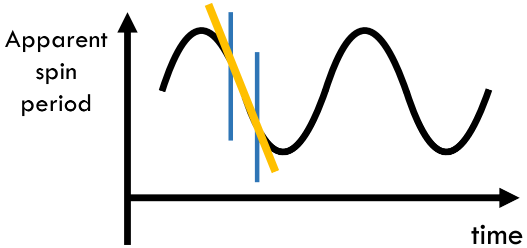
Fourier power is spread across many bins

How to search for new interesting objects: binary pulsars

Several proposed Algorithms so far: e.g.

Linear acceleration code

$T_{obs} \leq P_{orb}/10$



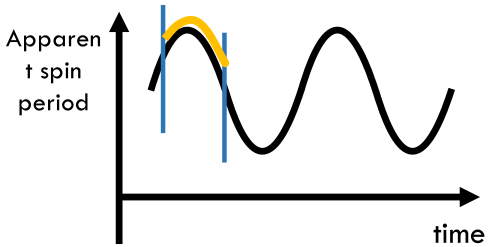
Assumes constant acceleration(a) during an observation.

v(t)=at hence $p' = p_0(1 + \frac{at}{c})$

where p_0 is the initial spin period and p' is the first time derivative

Jerk code

$T_{obs} \sim 0.05 - 0.15\, P_{orb}$

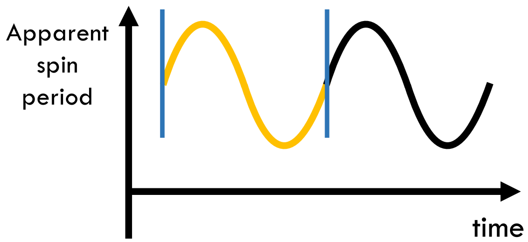


Assume a constant jerk(j)

$$\mathbf{a}(t) = jt + a_0 \qquad P'(t) = P_0(1 + a_0 t/c + jt^2/c)$$

Jerk search approximates a constant jerk with linearly varying acceleration.

Template Banking code



It aims to search over the Keplerian parameters of a circular orbit: orbital period, projected radius of the orbit, longitude of the ascending node, masses of the two bodies. That is achieved by matching filtering of the data convolved with a multidimensional space of waveforms

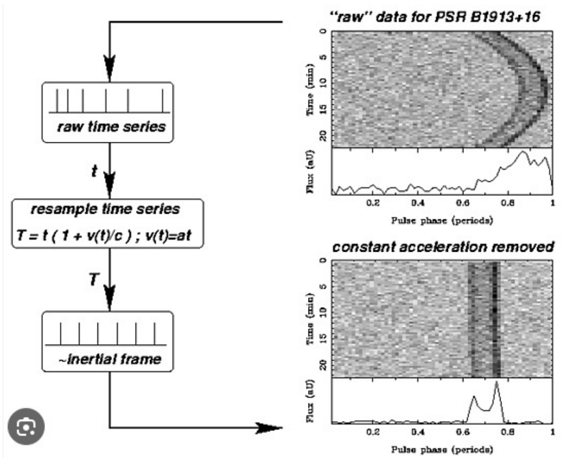
Technical Objectives, Methodologies and Solutions

Radio frequency Interference mitigation → De-dispersion → Periodicity searches using FFT → Preparation of pulsar candidate list and diagnostic plots → Inspection of the candidates by an Astronomer

Nag 2024

Several alternative approaches

Linear acceleration search



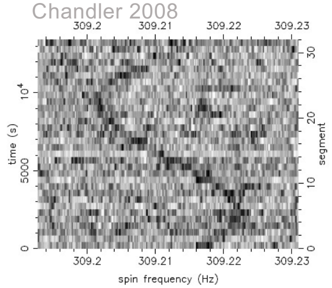
Not very sensitive to tight systems

Template banking search

Potentially good, no discoveries so far

Jerk search

Chandler 2008



Very good for tight binaries
Computationally Very Expensive

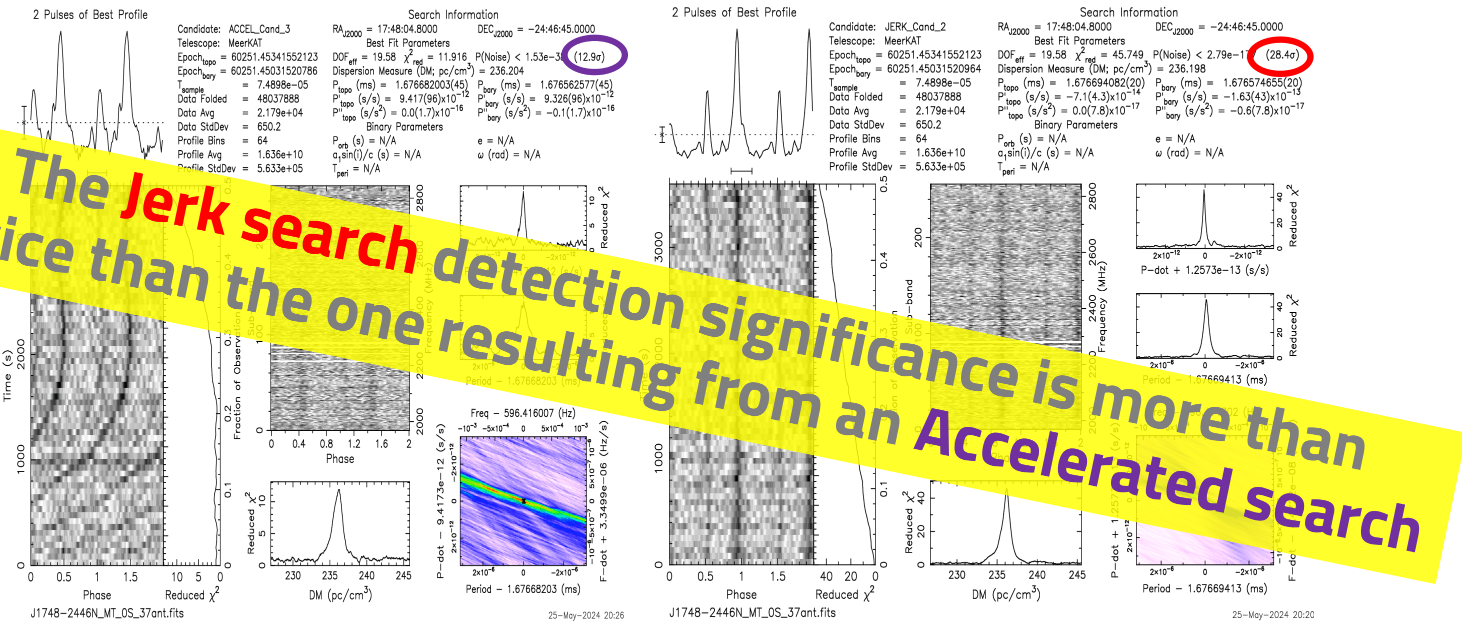
Needed accelerators ... GPUs

Results 1

14 computing nodes

8 CPU nodes with 64 AMD Milan 7513 (16 core) each, 512 GB DDR4 3200 RAM ,960GB SSD and 4 x 1.92 TB SSD

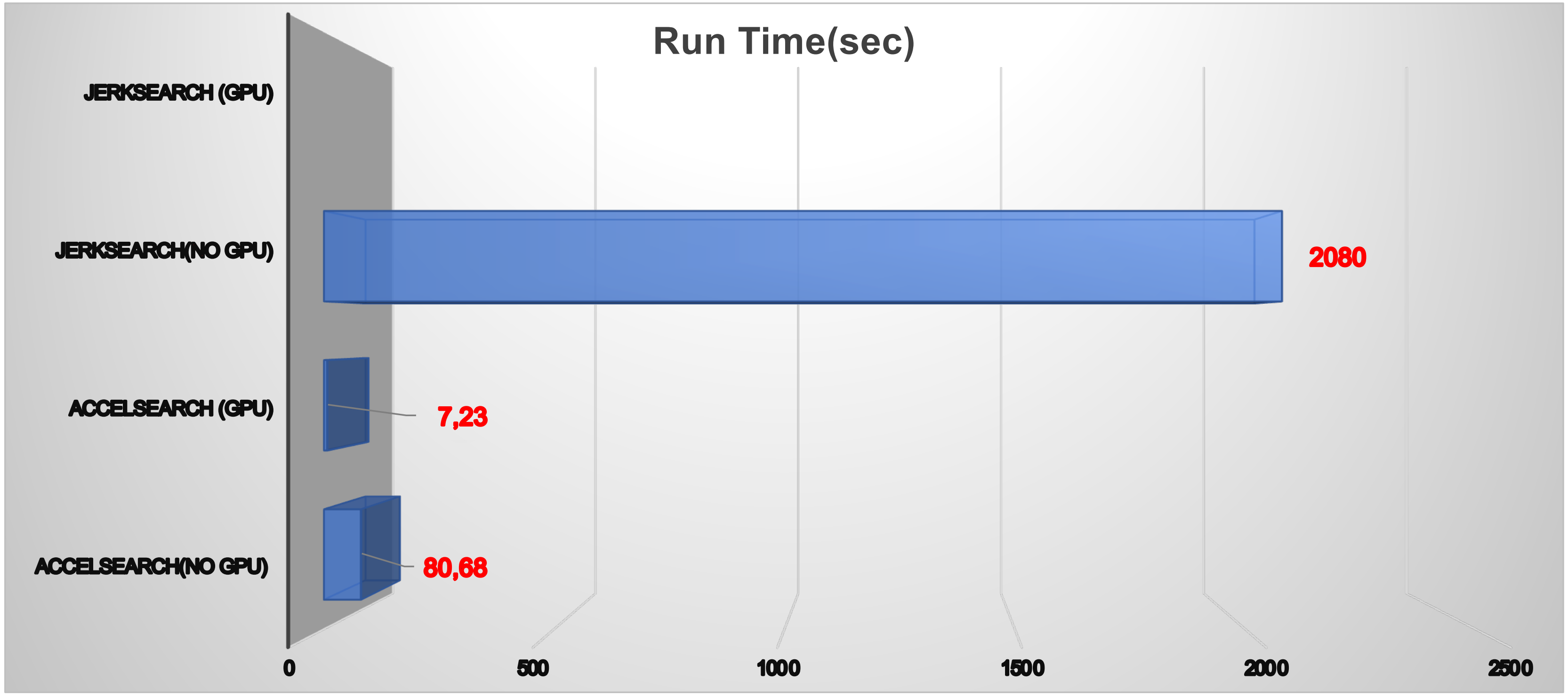
6 GPU nodes with the same configuration and the addition of 2 GPU's (NVIDIA A40 48GB RAM DDR6) per node



Accelerated search (GPU implementation) | Jerk search (CPU implementation)

Results 2

We investigated the runtime performance of the existing codes (PRESTO in this plot) in performing the «accelerated search» (with and w/o GPU) and the «jerk search» (w/o GPU)



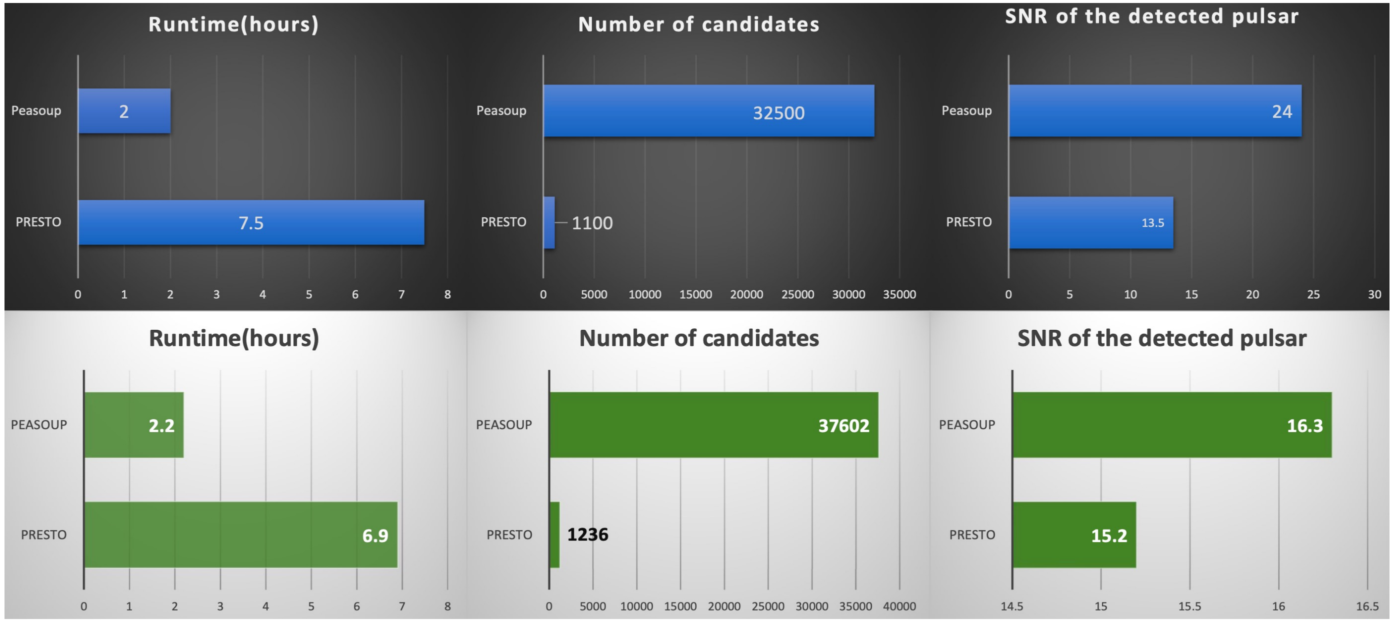
Results 3

We investigated the existing codes (Peasoup and PRESTO) for performing searches for «accelerated pulsars», in order to choose the one which is more suitable for performing «jerk searches» with our incoming GPUs implementation

Both codes already use GPUs for the de-dispersion step and the accelerated search step

Peasoup is quicker and finds higher signal-to-noise ratio (SNR) for the «discovered» pulsars.

However it produces many more pulsar candidates (i.e. fake candidates, due to radio interferences)



Final Steps

- **Add a GPU implementation for the Peasoup pipeline**
- **(optional) Combine Jerk search and an optimized Template Banking search in a new module of the Peasoup pipeline**
- **(optional) Test the ML capabilities developed for the «accelerated search» to reduce the huge number of candidates expected from the new combined module of Peasoup pipeline**

Timescale, Milestones and KPIs (WP1, 1.10)

Timeline	Description	Target	KPI
M7 (Sept 2023 – Feb 2024)	Acquisition of the basic knowledge of the most used existing codes to search for periodic signal in timeseries, as well as their usage in the astrophysical context. Implementation and test of these codes on CPU machines	Conceptual summary of the acquired knowledges	Presentation at doctorate first year review at University of Cagliari: https://www.openaccessrepository.it/record/143685 (also doi: 10.15161/oar.it/143685)
M8 (Mar 2024 – June 2024)	Review and implementation of the latest CPU codes for searching for strongly modulated periodic signals in timeseries and investigation of their performance with respect to the codes examined in the previous milestone. In particular, this class of codes, usually dubbed as codes for "jerk" searches, have been developed in order to detect periodically emitting cosmic sources included in a very tight binary system.	Runnable version of the CPU based codes for performing "jerk" searches and application to some real data case. Comparison of the performance with respect to the code simply running "accelerated" searches	Presentation at a talk given at INAF
M9 (Jul 2024 – Oct 2024)	In-depth examination of the available CPU-based code for performing "jerk" searches and initial tests for replacing some of the tasks, by using GPUs	Exploration of the modifications of the CPU code in order to use GPUs for performing "jerk" searches	Runs data output with specific plots.Identifying the parameter space where the Jerk search has maximum and minimum effectiveness. Open of a gitlab repository to host the code: link https://github.com/Rouhin1997/presto_Jerk_modLink to a gitlab repository
M10a (Nov 2024 – Mar 2025)	Development and implementation of a GPU-based code for running a "jerk" search of periodic signals generated by radio sources located in tight binary system	Tests of application of a first beta version of the GPU-based code for performing a "jerk" search	Public presentation of the features and the performances of the beta version of the code
M10 final (Apr 2025 – Aug 2025)	Development and implementation of a GPU-based code for running a "jerk" search of periodic signals generated by radio sources located in tight binary system	Runnable version of the GPU-based code for running a "jerk" search	Link to the gitlab repository. Publication about the description and first use of the GPU jerk code

We are basically in line with all the milestones and KPIs until M9.
As to M10 (first part) we are about 50% , while we are about 25% of the final M10