The GWTBoost: Modernizing Unmodeled Gravitational-Wave Analysis from cWB to PycWB in the High-Performance Computing Era

D. Di Piero[1,2], G. Principe[1,2], and E. Milotti[1,2]

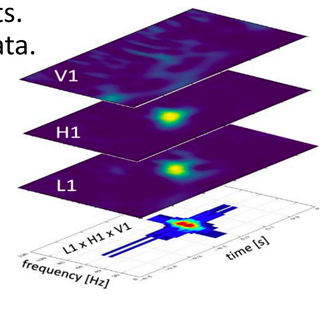
[1] Dipartimento di Fisica, Università di Trieste, I-34127 Trieste, Italy,
[2] Istituto Nazionale di Fisica Nucleare, Sezione di Trieste, I-34127 Trieste
On Behalf of the PycWB Developing Team

Abstract: Unmodeled search methods play a crucial role in detecting generic gravitational-wave transients (GWTs), especially in the case of signals without precise theoretical predictions. Coherent WaveBurst (cWB) is one of the primary data-analysis pipelines used for the detection and coherent reconstruction of such signals. Within the GWTBoost non-flagship use case, we focus on developing and optimizing algorithms for unmodeled gravitational-wave analyses on the LEONARDO high-performance computing (HPC) platform. To this end, we are developing PycWB, a modern Python-based reimplementation of the cWB algorithm. By exploiting HPC infrastructures, PycWB aims to enhance scalability, flexibility, and computational efficiency—key requirements for handling the ever-increasing data volumes produced by current and future gravitational-wave observatories. The framework is designed to integrate seamlessly with contemporary scientific software ecosystems, fostering extensibility, reproducibility, and collaborative development. Here, we present the motivation, design, and implementation strategies behind PycWB, as well as the broader perspectives it opens for next-generation gravitational-wave signal analysis.

Detecting Unmodelled Transient Signals

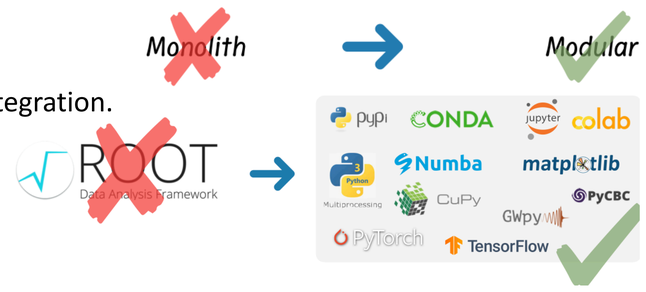
Coherent WaveBurst (cWB) [S. Klimenko et al, 2008][S. Klimenko et al, 2021]:
- is an open source software (written in C/C++ and ROOT) for GW data analysis.
- detects the excess of signal power that is coherent in the network of detectors.
- is weakly-modelled algorithm used to search for GW bursts.
- uses time-frequency analysis to identify signals in noisy data.
- Successfully detected major events like GW150914.

Pipeline workflow:
1. *Time-frequency representation and whitening*
2. *Selection of most energetic pixels*
3. *Coherent analysis (maximum likelihood)*
4. *Compute ranking statistic of each trigger*



PycWB – A Python Implementation of cWB

PycWB [Yumeng Xu et al, 2024]
- Why Python? Widely adopted in GW data analysis → easy, modular, collaborative.
- Limitations of cWB (C++): Hard to customize, steep learning curve, complex dependencies.
- PycWB advantages:
  - Python-based → accessible & extensible.
  - Modular structure → easier customization & integration.
- Computational gains:
  - Designed for HPC & parallelization,
  - scalable to large datasets.

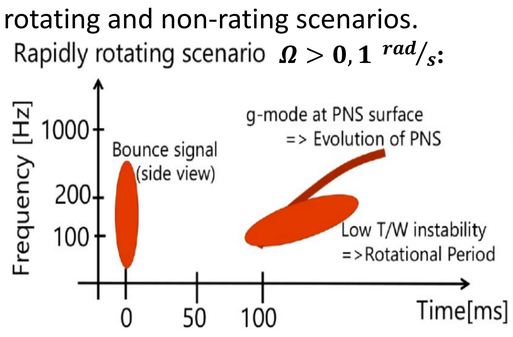


Supernovae explosion and GW emission

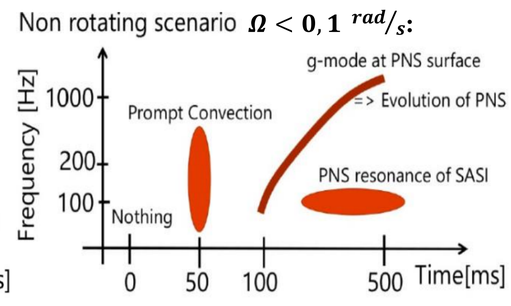
As test for the PycWB pipeline we aim to investigate the GW emission of supernovae explosion for both rotating and non-rating scenarios.

Supernovae explosion phases:
1. The star runs out of fuel
2. The core is stabilized against gravity by the pressure of degenerate e^-
3. Collapse begin when $M_{core} > M_{ch}$ electron capture processes become relevant
4. Core bounces when $\varrho \approx 10^{14}\ g/cm^3$
5. After being launched the shock-wave propagates outwards and dissipates energy.
6. Neutrino heating and Standing Accretion Shock Instability (SASI) revive the shock
7. The explosion is launched

Rapidly rotating scenario $\boldsymbol{\Omega > 0,1\ ^{rad}/_{s}}$:

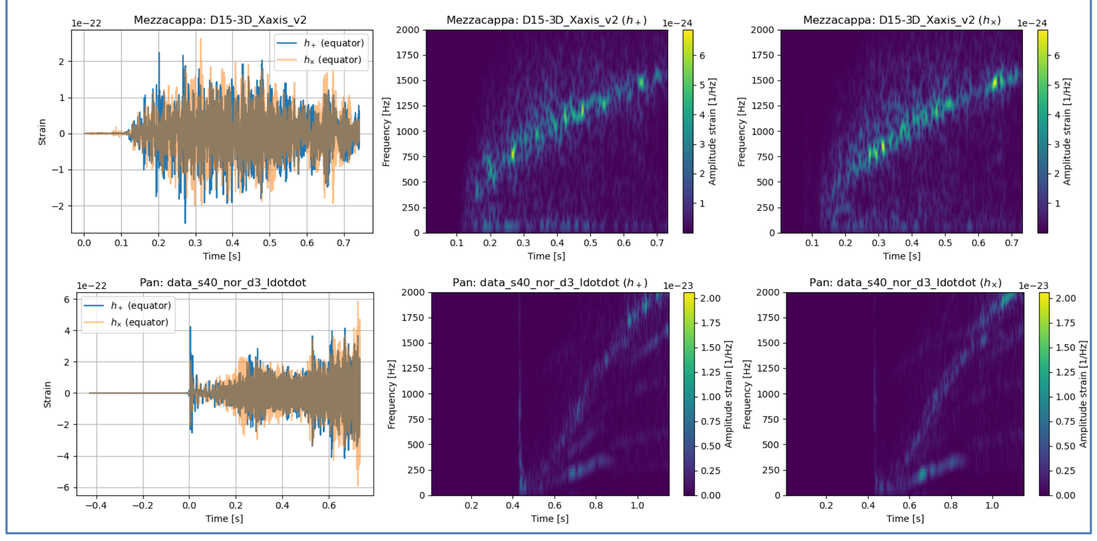


Non rotating scenario $\boldsymbol{\Omega < 0,1\ ^{rad}/_{s}}$:



Supernovae waveforms injected

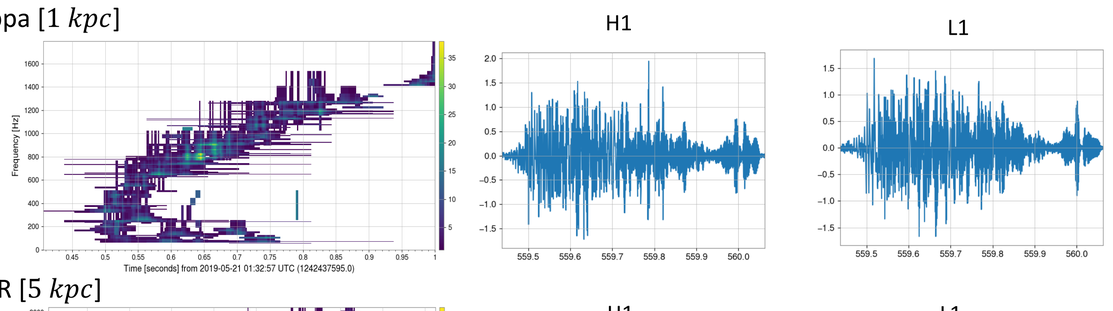
To evaluate PycWB perfomances, we injected SN waveforms into background data. We tested roughly a dozen distinct waveform models; for brevity, we present results for two representative examples. Mezzacappa et al., 2023 Pan et al., 2021



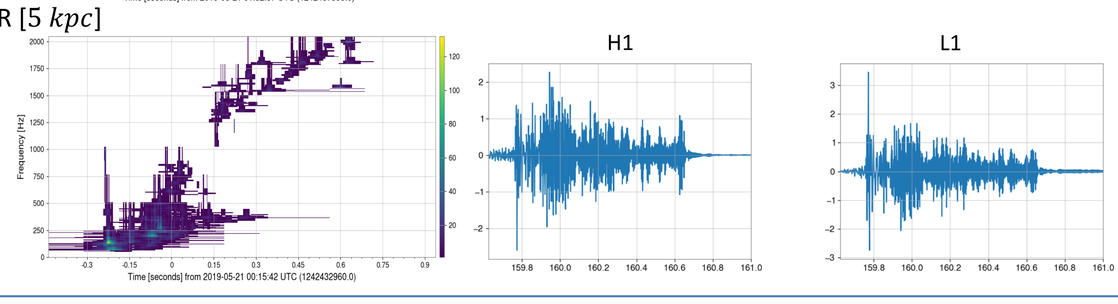
PycWB injected events reconstruction

We report here the detection and reconstruction of SN waveforms with PycWB, demonstrating that the pipeline successfully identifies and reconstructs the injected signals.

Mezzacappa [1 kpc]



Pan s40 SR [5 kpc]



Conclusions

While GW have opened a new window on the Universe, PyCWB brings cWB to Python, making the pipeline more accessible, modular, and extensible, and enabling its use in future searches for GWTs. PycWB advantages:
- Astrophysical impact → easily enables searches for unmodelled signals beyond compact binary mergers.
- Computational scalability → optimized for HPC & large GW datasets.
- Scientific impact → enables unmodelled searches, waveform reconstruction, sky localization.
- Future directions → GPU acceleration, Numba, optimized parallelization for next-gen data volume, support for next-gen observatories (i.e. Einstein Telescope).

References

- S. Klimenko et al. “A coherent method for detection of gravitational wave bursts”. In: Classical and Quantum Gravity 25.11 (May 2008), p. 114029.
- S. Klimenko et al. cWB pipeline library: 6.4.1. Version cWB-6.4.1. Dec. 2021.
- Yumeng Xu, Shubhanshu Tiwari, and Marco Drago. “PycWB: A user-friendly, Modular, and python-based framework for gravitational wave unmodelled search”. In: SoftwareX 26 (2024),p. 101639. issn: 2352-7110
- Anthony Mezzacappa et al. “Core collapse supernova gravitational wave emission for progenitors of 9.6, 15, and 25MJ”. In: Phys. Rev. D 107 (4 Feb. 2023), p. 043008.
- Kuo-Chuan Pan et al. “Stellar Mass Black Hole Formation and Multimessenger Signals from Three-dimensional Rotating Core-collapse Supernova Simulations”. In: 914.2, 140 (June 2021), p. 140.

Finanziato dall'Unione europea NextGenerationE
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