Modern compilers implement many automatic program optimizations. E.g., when compiling a program with gcc and the -O3 flag, 98 optimizations are applied\(^1\). Each of these optimizations can be applied individually by setting the respective flag. Program optimizations interact with each other, some complement and some interfere with each other and also the order in which they are applied makes a difference. Therefore, in order to achieve the best performance improvements using a given set \(N = \{o_1, \ldots, o_n\}\) of optimizations, the best subset of optimizations together with the respective best order of application \((o_{i_1}, \ldots, o_{i_\ell})\), \(i_j \neq i_k\) for \(j \neq k\) for \(j, k \in \{1, \ldots, \ell\}\) and \(i_j, i_k \in \{1, \ldots, n\}\), needs to be found.

Of course, in general for a sufficiently large set of optimizations \(N\) and arbitrary interactions it is infeasible to compute the best subset and corresponding best order of optimizations — there are \(2^n\) subsets \(A \subseteq N\) and for each subset \(A\) there are \(|A|!\) possible application orders. We recently introduced the mathematical foundation of Fourier analysis for data indexed by powersets \(^1\), meet/join lattices \(^2\), and posets (under review). Our novel Fourier analysis provides the mathematical tools to pinpoint which interactions between program optimizations are important for the performance improvements achieved on a certain program.

**Your contribution** The goal of this project is to compose a PyPI\(^2\) package providing all of our algorithms for the Fourier analysis of set functions and general poset data and to demonstrate its utility on the example of program optimizations outlined above.

Your library should provide:

1. Pybind\(^3\) wrappers of our fast Fourier transform algorithms for set functions and for data indexed by posets,

2. our recent sparse set function Fourier transform algorithms \(^3\).

---

\(^1\)See [https://gcc.gnu.org/onlinedocs/gcc/Optimize-Options.html](https://gcc.gnu.org/onlinedocs/gcc/Optimize-Options.html)

\(^2\)See [https://pypi.org/](https://pypi.org/)

\(^3\)See [https://github.com/pybind/pybind11](https://github.com/pybind/pybind11)
3. the robust sparse Walsh-Hadamard transform from [4], and

4. plotting utilities.

**Deliverables**

*Final report:* The final report may be written in English or German. It must contain an abstract written in both English and German. It should include an introduction, an analysis of related work, and a complete documentation of all used software tools and mathematical derivations. Three copies of the final report must be delivered to the supervisor.

*PyPI package:* A PyPI package providing the functionality described above together with implementations, configuration scripts and instructions to reproduce the results reported in the thesis must be delivered in electronic form.

*Presentation:* The results of the thesis must be presented to our research group. The presentation is capped to 30 minutes and should give an overview as well as the most important details of the work.

**Contact** If you are interested in pursuing this master thesis, please contact wendlerc@ethz.ch or pueschel@ethz.ch.

**References**


