**Benchmarks (solid effect, laptop)**

**Calculating 1E6 matrix products:**

Matlab (s): 1.34

Python + NumPy, Enthought (s): 2.23

Python + Numpy, Intel (s): 2.27

Python + Numpy, Anaconda (s): 2.38

Python + F2PY (s): 1.90

Fortran, gfortran (s): 1.88

Fortran, Intel (s):

*Try Fortran with Intel MKL then on Ubuntu machine. If still slower than Matlab, post on Stackoverflow for help. This does however explain why Matlab program is much faster than Python. It is curious that Fortran is slower than Matlab, presumably due to high optimisation of Matlab matrix multiplication? Seems odd that overhead of Matlab code would not exceed any benefits however.*

*This would mean there is no point optimising Python code, performance will never exceed Matlab?*

**~~Calculating 1E5 Kronecker products:~~**

~~Matlab (s): 4.18~~

~~Python + NumPy (s): 7.33, 7.34, 7.41~~

**~~Calculating 1E4 matrix exponentials:~~**

~~Matlab (s): 2.02~~

~~Python + NumPy (s): 6.77, 6.70, 6.73~~

**Calculating Hamiltonian:**

Matlab (s): 0.037593

Python + NumPy (s): 0.892, 0.885, 0.887

Python + F2PY (s): 0.070, 0.070, 0.070

Python + F2PY + OpenMP (s): 0.023, 0.025, 0.022

**Calculating Liouville space propagator:**

Matlab (s):

Python + NumPy (s): 10.88

Python + F2PY (s):