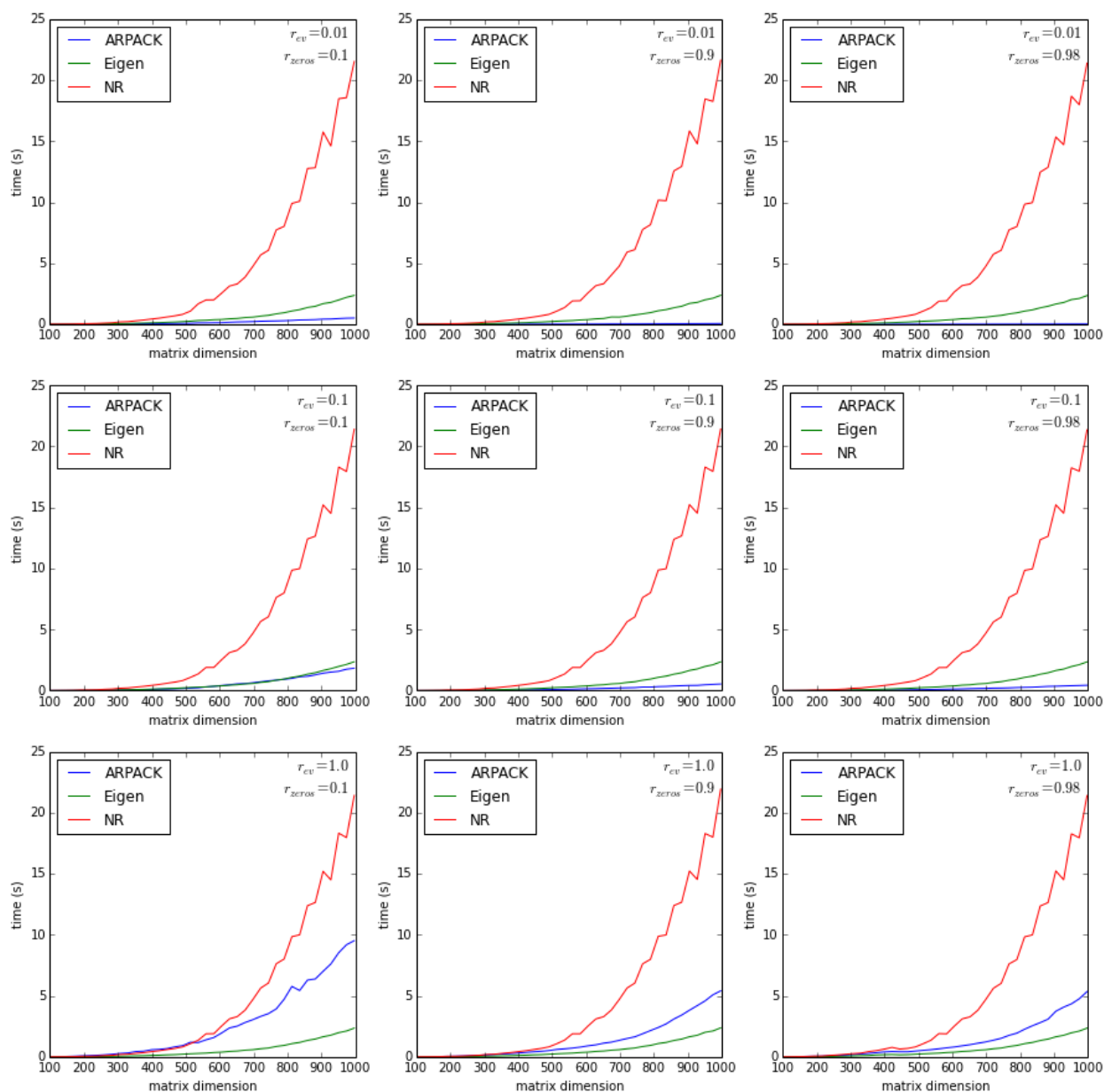


# 1 Performance of ARPACK, Eigen, and Numerical Recipes

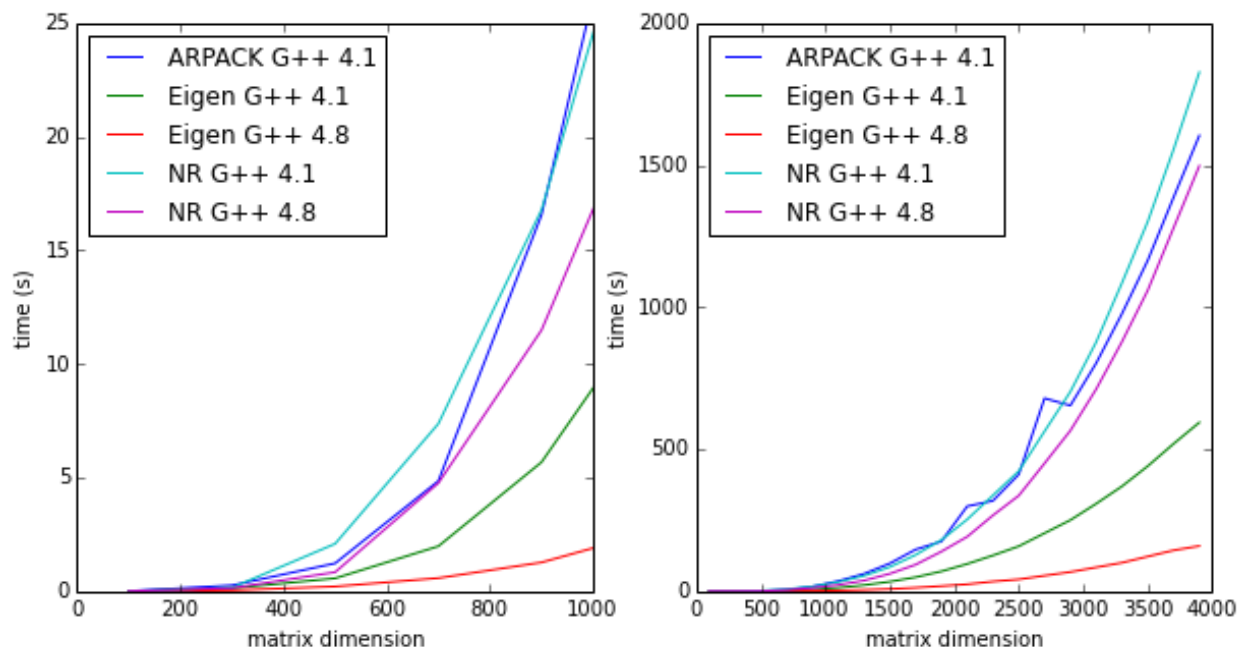
Burkhard Ritter, Christopher Polachic, February 2014

We compare the performance of the Numerical Recipes (C++ version, 3rd edition), ARPACK and Eigen eigensolvers for different real symmetric matrices with varying sparseness. Both Numerical Recipes and Eigen are dense eigensolvers and always compute the full eigenvalue spectrum, whereas ARPACK uses sparse matrices and is typically used to calculate only a few eigenvalues and not the full spectrum.



The time the eigensolvers take to diagonalize a random matrix of a given size. Here,  $r_{ev}$  is the “ratio” of eigenvalues. Hence  $r_{ev} = 0.1$  means calculating 10% of the eigenvalues, for ARPACK only. Eigen and Numerical Recipes always calculate all eigenvalues.  $r_{zeros}$  is the “ratio” of zeros. Hence  $r_{zeros} = 0.25$  means 25% sparse. Both Eigen and Numerical Recipes are not affected by  $r_{ev}$  and  $r_{zeros}$  and thus the corresponding curves should be the same for all plots above.

This test was run on OS X 10.9.1, the compiler was GNU G++ 4.8.2 (installed from MacPorts). Arpack is 3.1.4 (it's Arpack-ng), installed from MacPorts. Arpack was used as the Arpack wrapper.



A very similar test, this time run on WestGrid's Jasper (Linux). Uses latest Arpack from Rice university and Arpack++ as the wrapper. The matrix is the Hamiltonian from a real physical problem, it's about 1% to 2% sparse. Numerical Recipes and Eigen are compiled with two different versions of the GNU G++ compiler, as indicated. The full spectrum of eigenvalues is computed. This scenario corresponds to the  $r_{ev} = 1.0$  and  $r_{zeros} = 0.98$  (the bottom right-most) plot in the last graph.

While the general trends agree, there are some notable differences: Arpack is much slower here than above. Reversely, Numerical Recipes is faster in this test, compared to the last.

It should be noted that ARPACK was used in a black box like fashion (via the wrappers) and there might be considerable room for improving its performance, for example by compromising on precision (we use machine precision) or just using it in a cleverer way. To compute the full spectrum we actually run ARPACK twice, once to get all but one eigenvalue, starting from the smallest, and once the compute the missing largest eigenvalue. (ARPACK does not seem to allow to compute the whole eigenvalue spectrum directly, at least not in the simple direct manner we are using via the wrappers.)

We summarize the results of these tests:

1. Never ever use Numerical Recipes, it's extremely slow.
2. Use new compilers. They are significantly faster than the old ones, especially for sophisticated C++ libraries like Eigen.
3. To calculate only a couple of eigenvalues, use Arpack; and it's increasingly faster for increasingly sparse matrices. No surprises here.
4. To calculate all eigenvalues, use Eigen.