CSE 280: Assignment 2

Evan Ott
UT EID: eao466

November 7, 2016

2.0

```
tar -xf gsl-1.16.tar.gz
cd gsl-1.16
mkdir builds
cd builds
mkdir gcc0 gcc3 icc0 icc3

cd gcc0
module unload intel
module load gcc
../../configure CC=gcc CFLAGS=-O0
(time make -j1) 2> serial.txt
make clean
../../configure CC=gcc CFLAGS=-O0
(time make -j16) 2> parallel.txt
make check -k 2>&1 | tee check_results.txt

cd ../gcc3
../../configure CC=gcc CFLAGS=-O3
(time make -j1) 2> serial.txt
make clean
(time make -j16) 2> parallel.txt
make check -k 2>&1 | tee check_results.txt

cd ../icc0
module unload gcc
module load intel
../../configure CC=icc CFLAGS=-O0
(time make -j1) 2> serial.txt
make clean
(time make -j16) 2> parallel.txt
make check -k 2>&1 | tee check_results.txt

cd ../icc3
../../configure CC=icc CFLAGS=-O3
(time make -j1) 2> serial.txt
make clean
(time make -j16) 2> parallel.txt
make check -k 2>&1 | tee check_results.txt
```

I found the results for the table by using `cat` on the `serial.txt` and `parallel.txt` files and using a one-liner for pass/fail:

```
grep "# PASS: " check_results.txt | \
awk '{print substr($0, 10)}' | awk '{ sum += $1 } END { print sum }'
```

Configuration	Compilation Times		Regression Tests	
	Serial (secs)	Parallel (secs)	Passes	Failures
gcc/noOpts	145.767	35.637	48	0
gcc/O3	221.200	54.408	48	0
icc/noOpt	334.042	60.797	48	0
icc/O3	450.186	90.841	44	4

It was faster to build optimized GNU rather than optimized Intel, by around a factor of 2: (2.04 for serial, 1.67 for parallel).

Non-optimized GNU was 4.09x faster to compile in parallel than in serial.

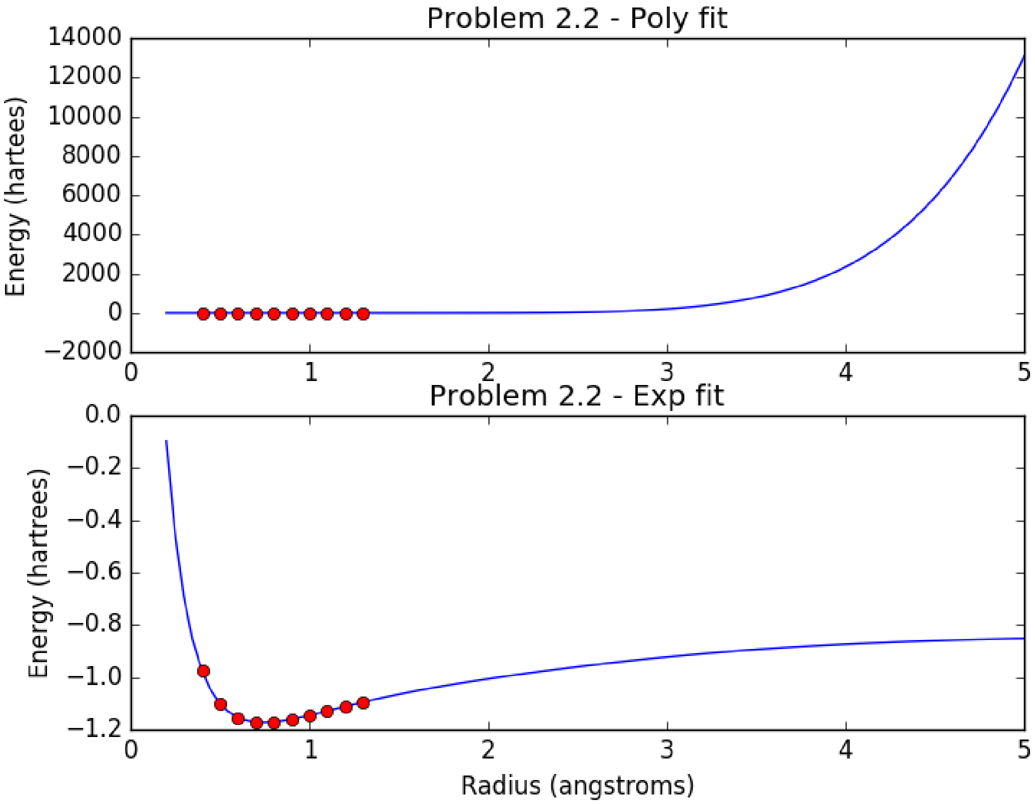
The only regression failures occured using the Intel compiler using `-O3`, which isn't surprising given the "magic" that Intel does at that level of optimization. I ran

```
grep -A4 "# FAIL:  1" icc3.check_results.txt | grep "See " | awk '{print substr($0, 5)}'
```

to identify the directories where failures occured (in this case, I got lucky because each time there were any failures in a directory, there was exactly one failure, so I used `grep` for exactly that condition then looked a few lines down to find the directory. In this case, the failing directories were (including a note from the GSL documentation):

- `linalg` – "functions for solving linear systems"
- `specfunc` – "GSL special function library" for families of functions like Bessel, elliptic integrals, etc.
- `poly` – "functions for evaluating and solving polynomials"
- `ode-initval2` – "functions for solving ordinary differential equation (ODE) initial value problems"

2.2


Problem 2.2 - Poly fit
Energy (hartees)
Problem 2.2 - Exp fit
Energy (hartrees)
Radius (angstroms)