

# Supercomputing in Plain English, Spring 2009

## Exercise #3: Arithmetic Operations

In this exercise, we'll use the same conventions and commands as in Exercises #1 and #2. You should refer back to the Exercise #1 and #2 descriptions for details on various Unix commands.

In the exercise, you'll benchmark various arithmetic operations, using various compilers and levels of compiler optimization. Benchmark means to run timing tests.

Specifically, you'll benchmark using the following compilers:

- the GNU Fortran compiler, `gfortran`, for various optimization levels;
- the Intel Fortran compiler, `ifort`, for various optimization levels;
- the Portland Group Fortran compiler, `pgf90`, for various optimization levels.

1. Log in to Sooner.
2. Copy the `ArithmeticOperations` directory:

```
cp -r ~hneeman/SIPE2009_exercises/ArithmeticOperations/  
~/SIPE2009_exercises/
```

3. Go (`cd`) into the appropriate directory:

```
cd ~/SIPE2009_exercises/ArithmeticOperations/
```

Note that this exercise has Fortran90 only. (I couldn't get the C version to work properly.)

4. Edit the batch script `arithmetic_operations.bsub` so that it contains your username and your e-mail address.
5. Compile, using the `make_cmd` shell script (a shell script is a file of Unix commands), which in turn invokes `make`:

```
make_cmd
```

6. Submit the batch job:

```
bsub < arithmetic_operations.bsub
```

7. Once the batch job completes, examine the various output files to see the timings for your runs with executables created by the various compilers under the various levels of optimization.
8. Use your favorite graphing program (for example, Microsoft Excel) to create graphs of your various runs, so that you can compare the various methods visually.