Summer 2020 Kiel University
Prof. Dr. Thomas Slawig, Niko Schmidt Dep. of Computer Science

Computer Science in Ocean and Climate Research

Exercise 3

Tutorial: (Exercise April 28th, 2020)

1. Install the GNU Fortran compiler `gfortran` on your computer.
2. Write a simple "Hello world" program in Fortran. A typical Fortran write-statement looks like this:

   ```
   write(*,*) 'Hello world!'
   ```

   The remaining necessary things you find in the lecture notes.
3. Write a Fortran program that computes the matrix product of an arbitrary real matrix and an arbitrary real vector.
   - Define the matrix and the vector als allocatable arrays with random numbers:

     ```
     real(8), allocatable :: A(:,:), x(:)
     integer :: n
     n = ...
     allocate(A(n,n))
     call random_number(A)
     ...
     ```

     What about deallocation?
   - Try to access the $(n+1)$st element of the vector. What does the compiler say?
   - Is there a compiler option that detects the error?
   - Measure the time that you need for the multiplication if you choose $n = 10, 100, 1000$. A simple time measurement can be done by

     ```
     call cpu_time(t1)
     ! ...
     call cpu_time(t2)
     write(*,*) 'cputime = ', t2-t1
     ```
   - Try different optimization compiler options (`-O, -O2, -O3`)
   - Compare the necessary time with the built-in Fortran function

     ```
     RESULT = MATMUL(MATRIX_A, MATRIX_B)
     ```

Home: (until May 5th, 2020)
Write a Fortran program that performs a simulation with the predator-prey model and a one-step time integrator (see last exercise).

- Store the results in two vectors, each of which contains a value for every time-step.
- Write the results of your predator-prey simulation to a file using

  ```
  open(unit = 20, file = 'outfile.txt', action = 'write')
  write(20,*) x
  ...
  close(20)
  ```

  Here the number 20 is a file identifier.
- Visualize the result using python etc.