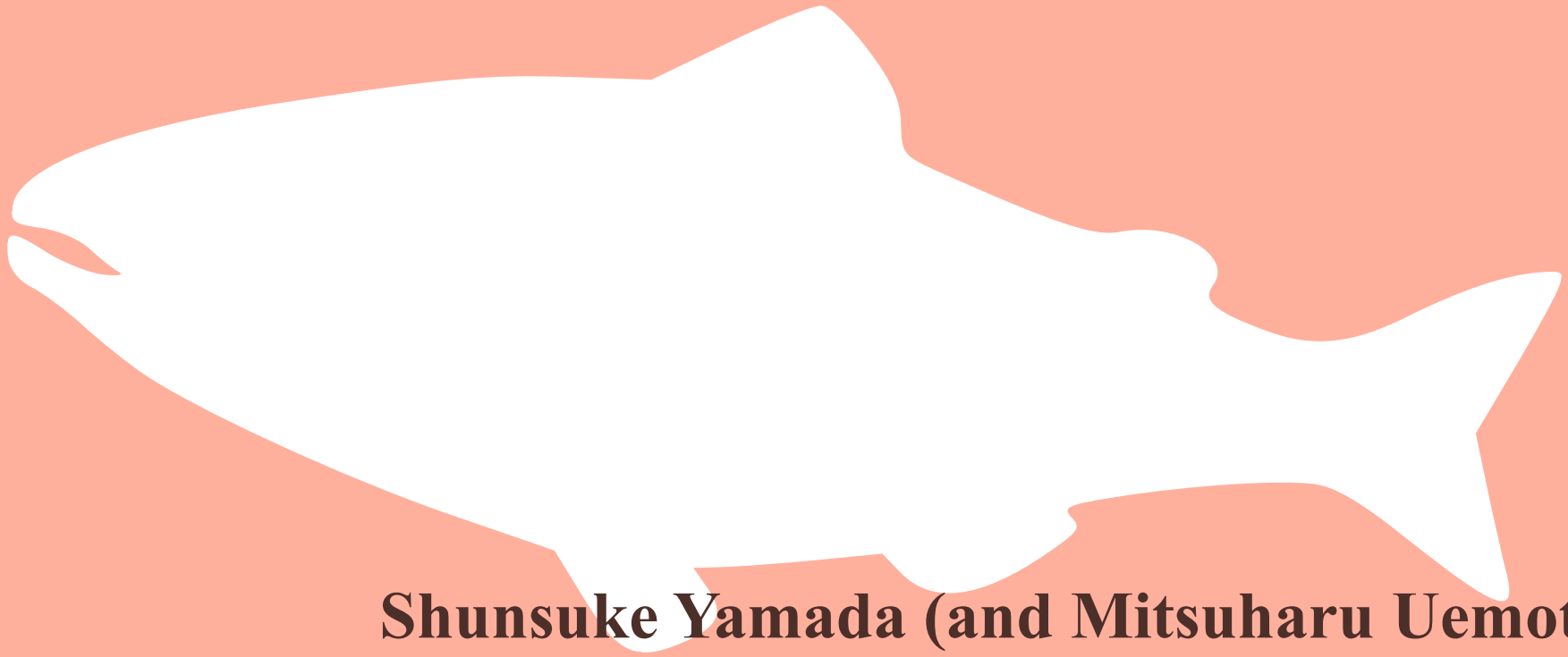
Installation in CMD environment

Quick start of SALMON



Shunsuke Yamada (and Mitsuharu Uemoto)
Center for Computational Sciences, University of Tsukuba

Prerequisites

- Fortran90/C compiler with MPI
  - GCC (Gnu Compiler Collection)
  - Intel Fortran/C Compiler
  - Fujitsu Compiler (at FX100 / K-Computer)
- Library packages for linear algebra
  - BLAS/LAPACK
  - Intel Math Kernel Library (MKL)
  - Fujitsu Scientific Subroutine Library 2 (SSL-II)
- Build tool
  - CMake
  - Gnu Make

Installation plan for CMD environment

```
~stud??/  ← Your home directory in "rl.phys.sci.osaka-u.ac.jp"
└── work
    ├── SALMON-v.1.2.1.tar.gz
    ├── SALMON        ← SALMON installation directory
    │   ├── build     ← Temporally files for build
    │   ├── bin       ←Binary
    │   └── ...
    ├── cmake-3.9.6-Linux-x86_64.tar.gz
    ├── cmake     ← cmake installation directory
    │   ├── bin
    │   └── ...
    └── cmd35     ← Excercises samples
```

- Create your working directory

```
$ cd ~
$ mkdir work
```

CMake: version check & install

Version check

```
$ cmake --version
```

CMake of version 3.0.2 or later is required.

Installation example of CMake version 3.9.6

```
$ cd ~/work
$ wget https://cmake.org/files/v3.9/cmake-3.9.6-Linux-x86_64.tar.gz
$ tar xvzf cmake-3.9.6-Linux-x86_64.tar.gz
$ mv cmake-3.9.6-Linux-x86_64 cmake
$ cmake --version

> cmake version 3.9.6
```

→ success!

Path setting

Modify ~/.bash_profile

* In this example, all programs will be installed in ~/work directory

Add $PATH settings as follows

```
# User specific environment and startup programs

PATH=$HOME/work/cmake/bin:$PATH
PATH=$HOME/work/SALMON-v.1.2.1/bin:$PATH

export PATH
```

Location of executable files

Update your configulation

```
$ source ~/.bashrc
```

Build of SALMON

Installation example for “intel-avx” architecture

```
$ cd ~/work/
$ wget http://salmon-tddft.jp/download/SALMON-v.1.2.1.tar.gz
$ tar xvfz SALMON-v.1.2.1.tar.gz
$ mv SALMON-v.1.2.1 SALMON
$ cd SALMON
$ mkdir build
$ cd build
$ ../configure.py CC=mpicc FC=mpifort --prefix=..
$ make && make install
$ which salmon.cpu
> ~/work/SALMON/bin/salmon.cpu   → success!
```

* Specify the compiler by CC and FC option (or use “arch” option)

Build of SALMON: Compile options

View help

```
$ ../configure.py --help
```

For more information → Install and Run (SALMON web page)

Explicit specification of compile options (example: gnu compilers)

```
$ ../configure.py FC=gfortran CC=gcc FFLAGS=-O3 …
```

Optional: Gnu Make

```
$ cd ~/work/SALMON-v.1.2.0/makefiles
$ make -f Makefile.<PLATFORM>
```

Run SALMON

Copy example input files & job.sh

```
$ cp -r ~teac23/work/cmd35 ~/work
$ cd ~/work/cmd35/test
```

Execute in PC cluster system

```
$ qsub ./job.sh
```

* Ground state calculation of acetylene

Quick check of output

1. Standard output (out.log)

   Syntax error → “error(s) in input”

2. variables.log

   For check of input variables (including default)

   #namelist: <Namelist>, status= 0 or -1 → OK

   otherwise → error