PICNIC user manual

Justin Angus

Outline

- Obtaining PICNIC
- Obtaining Chombo
- Obtaining/installing PETSc
- Building PICNIC
- Chombo basics
- Particle containers
- Time solvers
- Scattering

Obtaining PICNIC from GitLab

The PICNIC code is git controlled via LC's GitLab interface. In order to get the code, follow these steps:

Step 1: Get an LC and LC-GitLab account (see https://lc.llnl.gov/gitlab/)

Step 2: Configure SSH on non-LC machine (see https://dev.llnl.gov/securityaccess/ssh/)
If not working on an LC machine, then you want to add a 'config' file in your .ssh/ directory. Since I'm a CZ only user, I copy the file found here (https://dev.llnl.gov/securityaccess/ssh/cz_user/). Make sure to replace the <YOUR_LC_USERNAME_HERE> line at the end of the file with your lc username. For me, having this config file was necessary for being able to use 'git clone ssh…'. A bonus is that after I log into the LC one time, I no longer need the password for ssh or scp in any other terminal.

Step 3: Generate an SSH key and add it to GitLab profile (see the next page for details)

Step 4: Clone PICNIC to your machine. Go to the directory where you want the root PICNIC folder to live and type 'git clone ssh://git@czgitlab.llnl.gov:7999/angus1/picnic.git'. If all works, then you should have a new directory called 'picnic' that is git controlled

Obtaining PICNIC from GitLab

Step 3: Generate an SSH key

If you follow all the steps in the official git page, you might encounter the following error once you want to pull repo code: "git@czgitlab.llnl.gov: Permission denied (publickey)". This issue appeared after major LC updates in May 2023. As it explained in the solution, *As LC transitions its systems to the TOSS 4 operating system, support for older SSH key types is being phased out*. Here is the proper way to generate an SSH key:

- Run keygen: **ssh-keygen -b 4096 -t rsa**
- Do NOT use any passphrase
- Make sure the public and private keys were generated by checking **~/.ssh/id_rsa** and **~/.ssh/id_rsa.pub**
- Add the created key to .ssh/authorized_keys by running **ssh-copy-id <user>@oslic.llnl.gov**
- Log into your gitlab profile, go to *Preferences*, then to *SSH Keys* and add the key there by copy-pasting all the contents of **~/.ssh/id_rsa.pub** to the *Key* field. Make sure you copy the key correctly, as it can be broken down to multiple lines with a new line separator "\". Remove all such separators.
- Remove expiration date and click *Add Key*.

Obtaining Chombo

PICNIC is written in C++/FORTRAN uses the Chombo library for data containers with efficient memory and message passing interface (MPI) handling.

If you have your own Chombo library, then use it.

If not, then use Debo's git-controlled Chombo that also lives on LC's GitLab:
'git clone ssh://git@czgitlab.llnl.gov:7999/ghosh5/chombo.git'

You should never have to touch anything in Chombo. When new updates are available, Debo will manage that and all anyone else must do is use 'git pull'

No need to configure or install Chombo. That will be done at an initial compilation of PICNIC

Obtaining PETSc*

- PICNIC has its own native JFNK solver, but it is also setup to use linear and nonlinear solvers in PETSc (https://gitlab.com/petsc/petsc)

- A quick set of instructions for using PETSc is available here: https://debog.github.io/codes/petsc.html

- To get the release branch of PETSc,
  'git clone -b release https://gitlab.com/petsc/petsc.git petsc'

- I typically put Chombo and PETSc in a *libraries* directory. In other words, the *libraries* folder has sub-directories named chombo/ and petsc/.

* You can compile/run PICNIC without PETSc if you plan to run explicit time integrator

Installing PETSc*

- A quick set of instructions for using PETSc is available here: https://debog.github.io/codes/petsc.html
- Step 1: Set environment variables PETSC_DIR and PETSC_ARCH (I define them in my .bashrc file which is read on login to LC)**
  - E.g., export PETSC_DIR = "/user/workspace/angus1/libraries/petsc"
  - E.g., export PETSC_ARCH = "arch-opt"
- Step 2: Go to PETSC_DIR and configure. E.g, on quartz LC I do

./configure --with-batch --with-cc=mpicc --with-fc=mpif90 --with-cxx=mpicxx COPTFLAGS="-O2 -std=c99" FOPTFLAGS="-O2" CXXOPTFLAGS="-O2" --with-shared-libraries --with-debugging=0 --download-make --download-cmake --download-hypre --download-superlu --download-superlu_dist --download-parmetis --download-metis --with-cxx-dialect=C++11

- If configure works fine, then do '*make all*' to finish installing PETSc

* You can compile/run PICNIC without PETSc if you plan to run explicit time integrator

** See the next slide for more details of .bashrc (or .bashrc.ext)

Setup environment variables before compiling PICNIC

- Define environment variables for cpp and fortran compilers in startup scripts (e.g., .bashrc)
- My .bashrc file looks like this ➔
- CH_TIMER flag is needed for profiling during simulations
- CHOMBO_DIR is used at compile time
- PETSC_DIR is used at compile time
- Writing is done in parallel using HDF5
- .git-prompt.sh is useful for seeing your repo status on the command line

```
echo "Setting up environment for Quartz…"
module --force purge
module load StdEnv
module load gcc/10.3.1-magic
module load mvapich2/2.3.7
module load mkl/2022.1.0

export CH_TIMER=TRUE
export CC=$(which gcc)
export CXX=$(which g++)
export CPP=$(which cpp)
export F77=$(which gfortran)
export FC=$(which gfortran)
export MPICC=$(which mpicc)
export MPICXX=$(which mpicxx)
export MPIF77=$(which mpif77)
export MPIF90=$(which mpif90)
export MACHINE="quartz"

#export DARSHAN_DISABLE=1
export CHOMBO_DIR="~/PICNIC/chombo"
export PETSC_DIR="/usr/workspace/angus1/libraries/petsc"
export PETSC_ARCH="arch-opt"

module load hdf5-parallel/1.14.0
H5DIFF_SUFFIX=/bin/h5diff
H5DIFF_PATH=$(which h5diff)
HDF5_DIR=${H5DIFF_PATH%$H5DIFF_SUFFIX}
export HDF5_DIR_SERIAL="${HDF5_DIR}"
H5DIFF_PATH=$(which h5diff)
HDF5_DIR=${H5DIFF_PATH%$H5DIFF_SUFFIX}
export HDF5_DIR_PARALLEL="${HDF5_DIR}"

source ~/.git-prompt.sh
PS1='[\d \t \u@\h:\W$(__git_ps1 " (%s)")]\$ '
```

Building PICNIC

- Go to your **picnic/exec** directory
- If this is your first time building here, copy the *Make.defs.local* file from this directory to **Chombo/lib/mk**
- The *Make.defs.local* file that exist here is specific for quartz. It may need to be modified for other systems. I will get those setups (one day…)
- If you open the *GNUmakefile*, you will see basic directory and library linking commands used on compile time
- Simple build instructions are in README.txt. To build:
  1D: '**make -j all MPI=TRUE DEBUG=FALSE OPT=TRUE DIM=1**'
  2D: '**make -j all MPI=TRUE DEBUG=FALSE OPT=TRUE DIM=2**'

Species moments (mesh_data files)

By default, the following moments are written for each species:

$$\text{(mass density)} \qquad \rho \equiv m \int_{\mathbf{v}} f d\mathbf{v},$$

$$\text{(momentum density)} \quad \rho \mathbf{U} \equiv m \int_{\mathbf{v}} \mathbf{v} f d\mathbf{v},$$

$$\text{(diagonal elements of energy density tensor)} \quad E_{ij} \equiv \frac{m}{2} \int_{\mathbf{v}} v_i v_j f d\mathbf{v}, \text{ for } i = j$$

The energy density tensor is the sum of the pressure tensor and a mean energy tensor:

$$E_{ij} = \frac{m}{2} \int_{\mathbf{v}} v_i v_j f d\mathbf{v} = \frac{1}{2} P_{ij} + \frac{\rho}{2} U_i U_j, \text{ with } P_{ij} \equiv m \int_{\mathbf{v}} v_i' v_j' f d\mathbf{v}$$

The total energy density is the trace of this tensor (i.e., it is the sum of the diagonal elements)

Optional species moments

- pic_species.write_nppc = true ➔ grid file with number of macro particles per cell
- pic_species.write_energy_off_diagonal = true ➔ grid file with the three off-diagonal elements of the energy density tensor (stored as XY, XZ, and YZ)
- pic_species.write_energy_flux = true ➔ grid file with the energy density flux vector

$$\begin{aligned}\Gamma_E \equiv \int_{\mathbf{v}} \frac{1}{2} m v^2 \mathbf{v} f d\mathbf{v} &= \frac{1}{2} m \int_{\mathbf{v}} \left[|\mathbf{v}'|^2 + 2\mathbf{v}' \cdot \mathbf{U} + |\mathbf{U}|^2\right] (\mathbf{v}' + \mathbf{U}) f d\mathbf{v}, \\ &= \mathbf{q} + \frac{3}{2} P \mathbf{U} + m \int_{\mathbf{v}} (\mathbf{v}' \cdot \mathbf{U}) \mathbf{v}' f d\mathbf{v} + \frac{\rho}{2} U^2 \mathbf{U}, \\ &= \mathbf{q} + \frac{3}{2} P \mathbf{U} + \mathbf{P} \cdot \mathbf{U} + \frac{\rho}{2} U^2 \mathbf{U}.\end{aligned}$$

$$\text{where } \mathbf{q} \equiv \frac{1}{2} m \int_{\mathbf{v}} |\mathbf{v}'|^2 \mathbf{v}' f d\mathbf{v} \text{ is the heat flux,}$$

$$\text{and } P \equiv \frac{P_{xx} + P_{yy} + P_{zz}}{3} \text{ is the scalar pressure}$$