

OpenLB technical report: Configuring OpenLB on MacOS

Clara Schragmann

May 2022

The here described installation procedure has been tested with OpenLB 1.5 and MacOS 11.6.

1 Configuration using Homebrew

You can setup parallelization by using a package manager like Homebrew. This allows you to install and uninstall all necessary packages manually. Homebrew is easy to use and therefore recommended for beginners.

1. Download OpenLB from <http://www.openlb.net/> and unzip it to a folder (e.g. ~/Documents/openlb).
2. Open a new Terminal
3. Enter:

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

to download the package manager Homebrew
4. Enter:

```
brew install gcc open-mpi tinyxml zlib
```

to install all necessary packages
5. Enter:

```
brew info gcc
```

This shows the currently installed version of gcc (Should be 11 in May 2022), use this for the next steps
6. Open the config.mk-file in your OpenLB-folder
7. Set

```
CXX := mpic++  
CC := gcc-11 #Change this number to your installed gcc-version  
PARALLEL_MODE := MPI  
USE_EMBEDDED_DEPENDENCIES := OFF
```
8. Open a new Terminal
9. Enter:

```
cd Documents/openlb
```

to navigate to your OpenLB-directory (change this path according to your safe directory)

10. Enter:
cd examples/laminar/cavity2d
to navigate to the cavity2d-example
11. Enter:
make
to compile cavity2d
12. Enter:
mpirun -np 2 cavity2d
to run cavity2d on two processors

2 Configuration using Nix

Another way to configure parallelization on MacOS is by using a Nix-Shell. NixOS is also a package manager, but by opening a Nix-Shell all dependencies are being installed automatically. For the configuration of NixOS and its later uninstallation advanced programming skills are needed. This method is therefore only recommended for users already familiar with NixOS.

1. Download OpenLB from <http://www.openlb.net/> and unzip it to a folder (e.g. ~/Documents/openlb).
2. Open a new Terminal
3. Enter:
sh <(curl -L https://nixos.org/nix/install) --daemon
to download the package manager NixOS
4. Open the shell.nix-file in your OpenLB-folder with TextEdit
5. Adapt this file to look like this:

```
{ pkgs ? import <nixpkgs> { }, openlb_modes ? import ./nixexprs/build_modes.nix
  pkgs, ... }:
let
  mode = openlb_modes.gcc.openmpi;
in pkgs.stdenvNoCC.mkDerivation rec {
  name = "openlb-env";

  env = pkgs.buildEnv {
    name = name;
    paths = buildInputs;
  };

  buildInputs = with pkgs; [
    gnumake gnuplot openssl
  ] ++ mode.buildInputs;

  shellHook = let
    config_file = pkgs.writeTextFile {
      name = "openlb_makefile";
      text = import ./nixexprs/config.mk.nix mode;
    };
  in ''
    export NIX_SHELL_NAME="${name}"
    export OPENLB_CONFIG="${config_file}"
    export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/run/opengl-driver/lib
  ''
}
```

6. Open the config.mk-file in your OpenLB-folder
7. Set:
CXX := mpic++
CC := gcc
PARALLEL_MODE := MPI
USE_EMBEDDED_DEPENDENCIES := ON
8. Open a new Terminal
9. Enter:
cd Documents/openlb
to navigate to your OpenLB-directory (change this path according to your safe directory)

10. Enter:
nix-shell --pure
to open your Nix-environment
11. Enter:
make
to compile dependencies
12. Enter:
cd examples/laminar/cavity2d
to navigate to the cavity2d-example
13. Enter:
make
to compile cavity2d
14. Enter:
mpirun -np 2 cavity2d
to run cavity2d on two processors
15. Press:
control + d
to leave the Nix-shell