2021-02-11

Parallel processing and matrix solver interface: mtxp

Contents

Notice

- Version number of the software in the following will change owing to updates of the software.

1 Parallel matrix solver library: PETSc

1.1 Download by git

1. Make a directory (if already /opt directory exists, skip first two lines)

```
sudo mkdir /opt
cd /opt
sudo mkdir PETSc
sudo chown username:username PETSc ("username" should be replaced by your user name
cd PETSC
```

2. Download by git

```
git clone -b release https://gitlab.com/petsc/petsc.git petsc
cd petsc
```

1.2 Configure

1. Create a configure script "gfortran.py" by your favarite editor
2. Compiler names, "gcc-mp-10", "g++-mp-10", "gfortran-mp-10", should be modified according your configuration. The suffix shown as an example "-mp-10" is for gnu compiler version 10 installed by MacPorts
3. If you already installed MPI libraries, such as MPICH or OpenMP, use " '–with-mpi-dir=/usr/local/bin'," for the installed directory instead of " '–download-mpich=1',".

gfortran.py

```
#!/usr/bin/env python

# Build PETSc with gfortran

configure_options = [
  '--with-cc=gcc-mp-10',
  '--with-cxx=g++-mp-10',
  '--with-fc=gfortran-mp-10',
  '--with-shared-libraries=0',
  '--with-cxx-dialect=C++11',
  '--download-mpich=1',
  '--download-hypre=0',
  '--download-fblaslapack=1',
  '--download-spooles=1',
  '--download-superlu=1',
  '--download-metis=1',
  '--download-parmetis=1',
  '--download-superlu_dist=1',
  '--download-blacs=1',
  '--download-scalapack=1',
  '--download-mumps=1'
  ]

if __name__ == '__main__':
  import sys,os
  sys.path.insert(0,os.path.abspath('config'))
  import configure
  configure.petsc_configure(configure_options)
```

4. Make "gfortran.py" executable

```
chmod 755 gfortran.py
```

5. Setup environment variables for PETSc

```
export PETSC_DIR=/opt/PETSc/petsc
export PETSC_ARCH=gfortran
```

6. Configure

```
./gfortran.py
```

1.3 Compile

1. Compile

```
make all
```

2 Compile of task/mtxp module

2.1 Setup make.mtxp file

1. Goto mtxp directory

```
cd task/mtxp
```

2. Create a setup file

```
cp make.mtxp.org make.mtxp
```

3. Edit the setup file "make.mtxp"

   - If MPI is not available, remove comment mark "#" on lines 4–9
   - If MPI is available but PETs not, remove comment mark "#" on lines 12–17
   - If MPI and PETSc are available, remove comment mark "#" on lines 20–27

2.2 Modify make.header file

1. Go to task directory

```
cd ..
```

2. Edit make.header to use lapack and blas libraries for fortran

   - Near the beginning of the file, remove comment and ajust the path

```
LAPACK = lapack.f
LIBLA=-L /opt/PETSc/petsc/gfortran/lib -lflapack -lfblas
```

   - In the following, add comment marks

```
#LAPACK = nolapack.f
#LIBLA =
MODLA95 =
```

     Keep MODLA95 as it is, since lapack95 is not used.

   - Go back to mtxp directory

```
cd mtxp
```

2.3 Compile

1. Compile

```
make clean
make
```

2.4 Test mtxp

1. Test programs solving 1d, 2D and 3D Poisson equation are generated
   - `testbnd`: Direct band matrix solver (non-parallel)
   - `testpcg`: Iterative band matrix solver (non-parallel)
   - `testdmumps`: Direct band matrix solver (parallel using MUMPS)
   - `testkdsp`: Iterative band matrix solver (parallel using PETSc)
2. Input parameters
   - `idimen` : number of dimension (i or 2 or 3), 0 for quit
   - `isiz` : number of mesh point in one dimension
   - `isource` : source position is all dimensions
   - `itype` : tyoe of initial guess for PETSc 0..5 (default=0)
   - `m1` : type of solver (methodKSP) of PETSc 0..13 (default=4)
   - `m2` : type of preconditioner (methodPC) 0..12 (default=5)
   - `tolerance` : tolerance in iterative method
3. Example input For parallel processing

```
mpirun -np 4 ./testdmumps
# INPUT: idimen,isiz,isource,itype,m1,m2,tolerance,idebug=
1,11,6,0,4,5,1.D-7/
3/
0/
```

3 Compile of task/fp and related modules

3.1 Update Makefile

1. Change directory

```
cd ../fp
```

2. Edit Makefile
   - To use serial band matrix solver, remove comment mark "#" on lines 4-5.
   - To use serial iterative solver, remove comment mark "#" on lines 6-7.
   - To use parallel direct solver MUMPS, remove comment mark "#" on lines 8-9.
   - To use parallel direct solver library PETSc, remove comment mark "#" on lines 10-11. .

3.2 Compile

1. Compile related modules and task/fp files

```
make
```

4 Install on Ubuntu

4.1 Install of required modules

```
sudo apt-get install gfortran-8
sudo apt-get install gcc-8
sudo apt-get install g++-8

sudo apt-get install emacs
sudo apt-get install git
sudo apt-get install xorg-dev
sudo apt-get install valgrind
sudo apt-get install cmake
sudo apt-get install python
```

4.2 Install of MPICH

1. Download of mpich-3.3.1.tar.gz (*See 1.1*)
2. Expand at /soft/mpich (*See 1.1*)
3. Configure by executing "./run" (*See 1.2*)

run:

```
CC=gcc-8 CFLAGS=''-m64'' CXX=g++-8 CXXFLAGS=''-m64'' FC=gfortran-8
FFLAGS=''-m64'' ./configure --prefix=/usr/local/mpich331-gfortran-gcc8
--enable-cxx --enable-fast --enable-romio --disable-shared
```

4. Compile and install (*See 1.3*)

4.3 Install of PETSc

1. Download of petsc-3.11.3.tar.gz (*See 2.1*)
2. Expand at /opt/PETSc/ (*See 2.1*)
3. Setup environment variables (*See 2.2*)

```
export PETSC_DIR=/opt/PETSc/petsc-3.11.3
export PETSC_ARCH=gfortran
```

4. Configure by gfortran.py (*See 2.2*)
5. Execute fortran.py

gortran.py:

```
#!/usr/bin/env python

# Build PETSc, with gfortran

configure_options = [
  '--with-mpi=1',
  '--with-mpi-dir=/usr/local/mpich331-gfortran-gcc8',
  '--with-shared-libraries=0',
```

```
    '--with-cxx-dialect=C++11',
    '--download-mpich=0',
    '--download-hypre=0',
    '--download-fblaslapack=1',
    '--download-spooles=1',
    '--download-superlu=1',
    '--download-metis=1',
    '--download-parmetis=1',
    '--download-superlu_dist=1',
    '--download-blacs=1',
    '--download-scalapack=1',
    '--download-mumps=1'
    ]

if __name__ == '__main__':
    import sys,os
    sys.path.insert(0,os.path.abspath('config'))
    import configure
    configure.petsc_configure(configure_options)
```