Command line for pyomo installation with IPOPT from scratch. Command prompt lines are set within <>. File names are set within {}. When sending command, do not type in <, >, {, or }.

Requirements for simple installation

Python 3.8.5

Pyomo 6.4.0

Pandas 1.2.5

Openpyxl 3.0.7

Cobra 0.26.3

Scipy 1.8.0

Ipopt 3.14.0

Numpy 1.23.1

Requirements for manual compilation of Ipopt with HSL solver

Coinhsl-2021.05.05.tar.gz

Ipopt-releases-3.14.0.tar.gz  
ThirdParty-HSL-stable-2.1

ThirdParty-ASL-stable-2.0

HSL package 2021.05.05  
LAPACK 3.11.0

All anaconda packages are loaded in environment.yml which can be easily loaded by:

<conda create –prefix=$ENV -f environment.yml>

If this step is taken skip to step 8 for the installation of IPOPT

$ENV is the python environment directory of your choice. This is where you create your python environment. For instructions below, replace the $ENV with your absolute directory (for example, my absolute environment directory is /gpfs/group/cdm8/default/jack/pyomo/1/pyomo\_1)

1. Create conda environment
   1. <conda create –prefix=$ENV python=3.8.5>
2. Activate conda environment
   1. <conda activate $ENV>
3. Install pyomo
   1. <conda install -c conda-forge pyomo==6.4.0>
4. Install pandas
   1. <conda install pandas==1.2.5>
5. Install openpyxl
   1. <conda install openpyxl==3.0.7>
6. Install cobra (conda does not install cobra due to python version not meeting requirements)
   1. <pip install cobra==0.26.3>
7. Install scipy package
   1. <conda install -c conda-forge scipy==1.8.0>
8. Install IPOPT 3.14.0
   1. <conda install -c conda-forge ipopt==3.14.0>

The above protocol will install pyomo and related dependencies to run the toy kinetic model script. However, the automatic installation of ipopt through conda-forge will install the MUMPS linear solver. If you wish you use a different solver, such as the Harwell Subroutine Libraries (HSL), instead of executing step 8, do the following:

1. Apply for an academic license for HSL package at <https://www.hsl.rl.ac.uk/>
   1. Download file {coinhsl-2021.05.05.tar.gz}
2. Download IPOPT 3.14.0 at <https://github.com/coin-or/Ipopt/releases/tag/releases%2F3.14.0>
   1. Download source code {Ipopt-releases-3.14.0.tar.gz}
3. Change to $ENV directory
   1. <cd $ENV>
4. Clone and install ThirdParty-HSL, this package helps build and install the HSL routines for IPOPT
   1. <git clone <https://github.com/coin-or-tools/ThirdParty-HSL.git> -b stable/2.1>
   2. <cd ThirdParty-HSL>
   3. Move file {coinhsl-2021.05.05.tar.gz} into this folder
   4. Unpack the HSL files
      1. <gunzip coinhsl-2021.05.05.tar.gz>
      2. <tar xf coinhsl-2021.05.05.tar>
   5. Rename directory and move to directory
      1. <ln -s coinhsl-2021.05.05 coinhsl>
      2. <cd coinhsl>
   6. Ensure lapack is loaded
      1. <module load lapack>
   7. Set configure file
      1. <./configure -prefix=$ENV>
   8. Build libraries, install libraries, and add environment variable
      1. <make>
      2. <make install>
      3. <libtool finish $ENV>
   9. Configure in ThirdParty-HSL as well
      1. <cd ThirdParty-HSL>
      2. <./configure -prefix=$ENV>
      3. <make>
      4. <make install>
5. Clone and install ThirdParty-ASL, this package creates an executable ipopt when compiling. This executable is required for pyomo to run IPOPT.
   1. Go back to $ENV directory
      1. <cd $ENV>
   2. <git clone <https://github.com/coin-or-tools/ThirdParty-ASL> -b stable/2.0>
   3. <cd ThirdParty-ASL>
   4. <./get.ASL>
   5. <./configure -prefix=$ENV>
   6. <make>
   7. <make install>
6. Compile and install IPOPT
   1. Go back to $Env directory
      1. <cd $ENV>
   2. Move file {Ipopt-releases-3.14.0.tar.gz} into current directory
   3. Unpack files
      1. <gunzip {Ipopt-releases-3.14.0.tar.gz}
      2. <tar xvf {Ipopt-releases-3.14.0.tar}
   4. Rename directory
      1. <mv {Ipopt-releases-3.14.0} Ipopt>
   5. Change to Ipopt directory
      1. <cd Ipopt>
   6. Create build directory and move to build directory
      1. <mkdir build>
      2. <cd build>
   7. Configure the build with openmp flags, the OpenMP flag allows for MA86 run multiple-threads  
      1. (if compiler is mkl may need to module load mkl first)  
         <../configure --prefix=$ENV ADD\_CFLAGS=-fopenmp ADD\_FFLAGS=-fopenmp ADD\_CXXFLAGS=-fopenmp>
   8. Make,test, and build
      1. <make>
      2. <make test>
         1. With HSL solver in place, all test should pass
      3. <make install>
7. Add PATH to the ipopt executable
   1. Export PATH=$PATH:$ENV/Ipopt/build/src/Apps/AmplSolver/

PARDISO SOLVER

1. Grab pardiso license and library package from <https://pardiso-project.org/>
   1. Download libpardiso600\_GNU720-X86-64.so
   2. You will be given a license key in the email when applying. Save the license key as {pardiso.lic} and copy the key inside
   3. Place the lic file in the directory
2. Compile and install IPOPT
   1. Go back to $Env directory
      1. <cd $ENV>
   2. Move file {Ipopt-releases-3.14.0.tar.gz} into current directory
   3. Unpack files
      1. <gunzip {Ipopt-releases-3.14.0.tar.gz}
      2. <tar xvf {Ipopt-releases-3.14.0.tar}
   4. Rename directory
      1. <mv {Ipopt-releases-3.14.0} Ipopt>
   5. Change to Ipopt directory
      1. <cd Ipopt>
   6. Create build directory and move to build directory
      1. <mkdir build>
      2. <cd build>
   7. Configure the build with openmp flags, the OpenMP flag allows for MA86 run multiple-threads
      1. < ../configure --prefix=$MYENV ADD\_CFLAGS=-fopenmp ADD\_FFLAGS=-fopenmp ADD\_CXXFLAGS=-fopenmp --with-pardiso="/gpfs/group/cdm8/default/jack/pyomo/4/pyomo\_4/libpardiso600-GNU720-X86-64.so" >
      2. ADD\_CXXFLAGS=-fopenmp>
   8. Make,test, and build
      1. <make>
      2. <make test>
         1. Tests will not pass, few linking to do after
      3. <make install>
3. Link LD\_PRELOAD variable from where libblas and liblapack is installed.
   1. The information should be located in {libipopt.so}
   2. Change directory to find {libipopt.so} usually stored at:
      1. <cd $ENV/Ipopt/build/src/.libs/>
   3. Look at installation directories
      1. <ldd libipopt.so>
   4. Set LD\_PRELOAD path to the {libblas.so.3} and {liblaplack.so.3} files
      1. <export LD\_PRELOAD=”/lib64/libblas.so.3:/lib64/liblapack.so.3”>

Additional notes

If you have already did <make> , <make install> and wish to restart with a clean directory, use the following commands in that directory:

<make uninstall>

<make distclean>

To run with multiple threads, set the environment variable OMP\_NUM\_THREADS before running python

For example, to run with 24 threads type:

<OMP\_NUM\_THREADS=24>

To properly run the ipopt solver under ma86 you need to create a file {ipopt.opt} and write in the file <linear\_solver ma86>

For PARDISO, use

<linear\_solver pardiso>

Pynumero installation use

pyomo download-extensions

pyomo build-extensions