Installation of Ipopt and sIpopt in Windows via Cygwin using GNU compilers

This document provides a procedural description of how to build the AMPL solver executables (**ipopt.exe** and **ipopt_sens.exe**) in Cygwin for general use in Windows Operating System. Illustration on how to hook each solver to solve a given problem ensues. This installation process is illustrated through the following categorized sequence of steps or sections:

1 Installation Steps

There are various ways of installing IPOPT on Windows OS which can be found on the website (https://www.coin-or.org/Ipopt/documentation/documentation.html). This documentation focuses on Cygwin using GNU compilers and a subversion client to download the IPOPT code. The steps/sections are listed as follows:

1.1 Cygwin installation

1.2 Subversion client (SVN) installation

1.3 IPOPT code download

1.4 Third-party or External Code Download

1.5 Compiling and Installing IPOPT

1.6 sIPOPT installation

1.7 Ipopt.exe and ipopt_sens.exe Test

2 Example

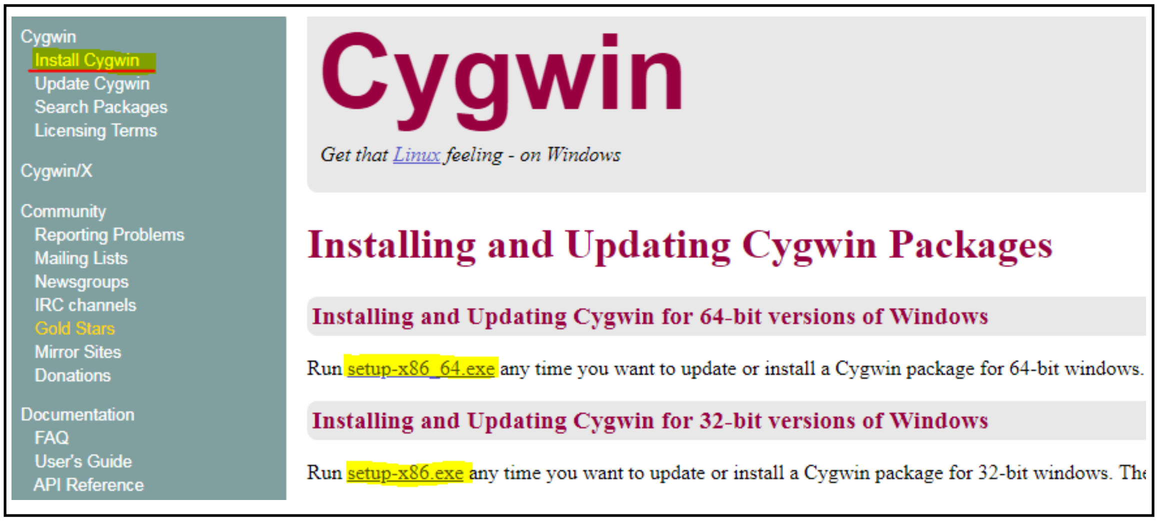
2.1 IPOPT/sIPOPT usage in PYOMO

3 References

Installation Steps

1. Cygwin Installation

- The required setup executable file can be sourced from the website http://www.cygwin.com. On the homepage, click on install Cygwin and select setup-x86_64.exe or setup-x86.exe for 64-bits and 32-bits Windows respectively. The file can be used for **updating or re-installing packages**. Hence it is advisable to select a destination folder where it can be specially preserved (e.g. C:\Cygwin64).



Installing and Updating Cygwin Packages

Installing and Updating Cygwin for 64-bit versions of Windows

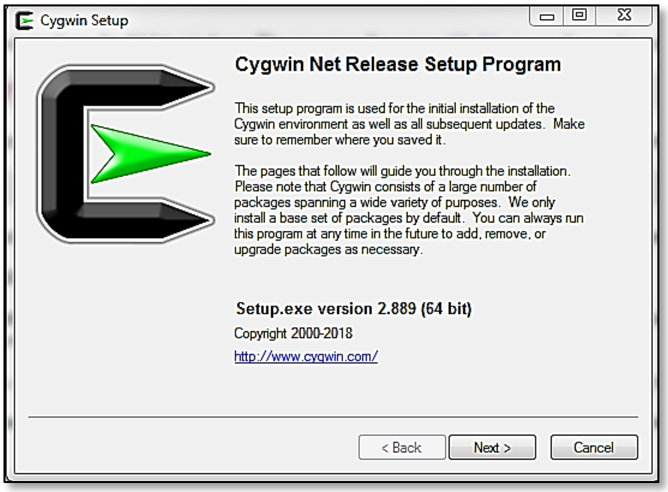
Run setup-x86_64.exe any time you want to update or install a Cygwin package for 64-bit windows.

Installing and Updating Cygwin for 32-bit versions of Windows

Run setup-x86.exe any time you want to update or install a Cygwin package for 32-bit windows. The

Figure 1 : Cygwin installation

- Double-click the setup file to run, the dialog box in Figure 2 will appear. Click on Next and then select 'Install from Internet'.



Cygwin Net Release Setup Program

This setup program is used for the initial installation of the Cygwin environment as well as all subsequent updates. Make sure to remember where you saved it.

The pages that follow will guide you through the installation. Please note that Cygwin consists of a large number of packages spanning a wide variety of purposes. We only install a base set of packages by default. You can always run this program at any time in the future to add, remove, or upgrade packages as necessary.

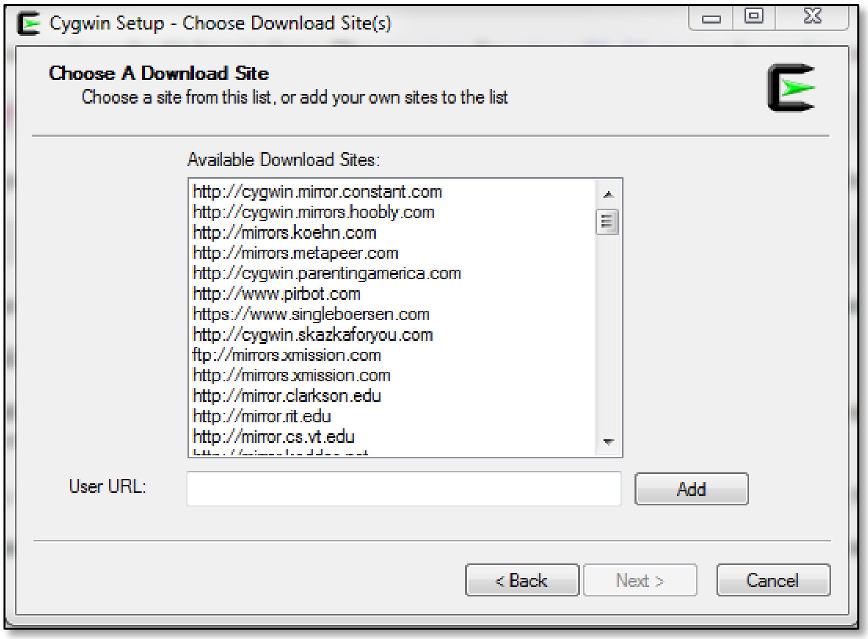
Setup.exe version 2.889 (64 bit)
Copyright 2000-2018
http://www.cygwin.com/

Figure 2 : Cygwin setup pop-up box

i. Select a directory where Cygwin will be installed (C:\Program Files\cywin64)

ii. Select Use System Proxy Settings (default) and click Next.

- Here you have to choose a mirror download site from the list shown in Figure 3 . It is advised to choose a site close to you (e.g. choose http://cygwin.osuosl.org for Oregon and http://cygwin.mirrors.hoobly.com for Pennsylvania). Details on mirror sites can be found on https://www.cygwin.com/mirrors.html.



Cygwin Setup - Choose Download Site(s)

Choose A Download Site

Choose a site from this list, or add your own sites to the list

Available Download Sites:

http://cygwin.mirror.constant.com
http://cygwin.mirrors.hoobly.com
http://mirrors.koehn.com
http://mirrors.metapeer.com
http://cygwin.parentingamerica.com
http://www.pirbot.com
https://www.singleboersen.com
http://cygwin.skazkaforyou.com
ftp://mirrors.xmission.com
http://mirrors.xmission.com
http://mirror.clarkson.edu
http://mirror.rit.edu
http://mirror.cs.vt.edu

User URL:

Figure 3 : Mirror download sites

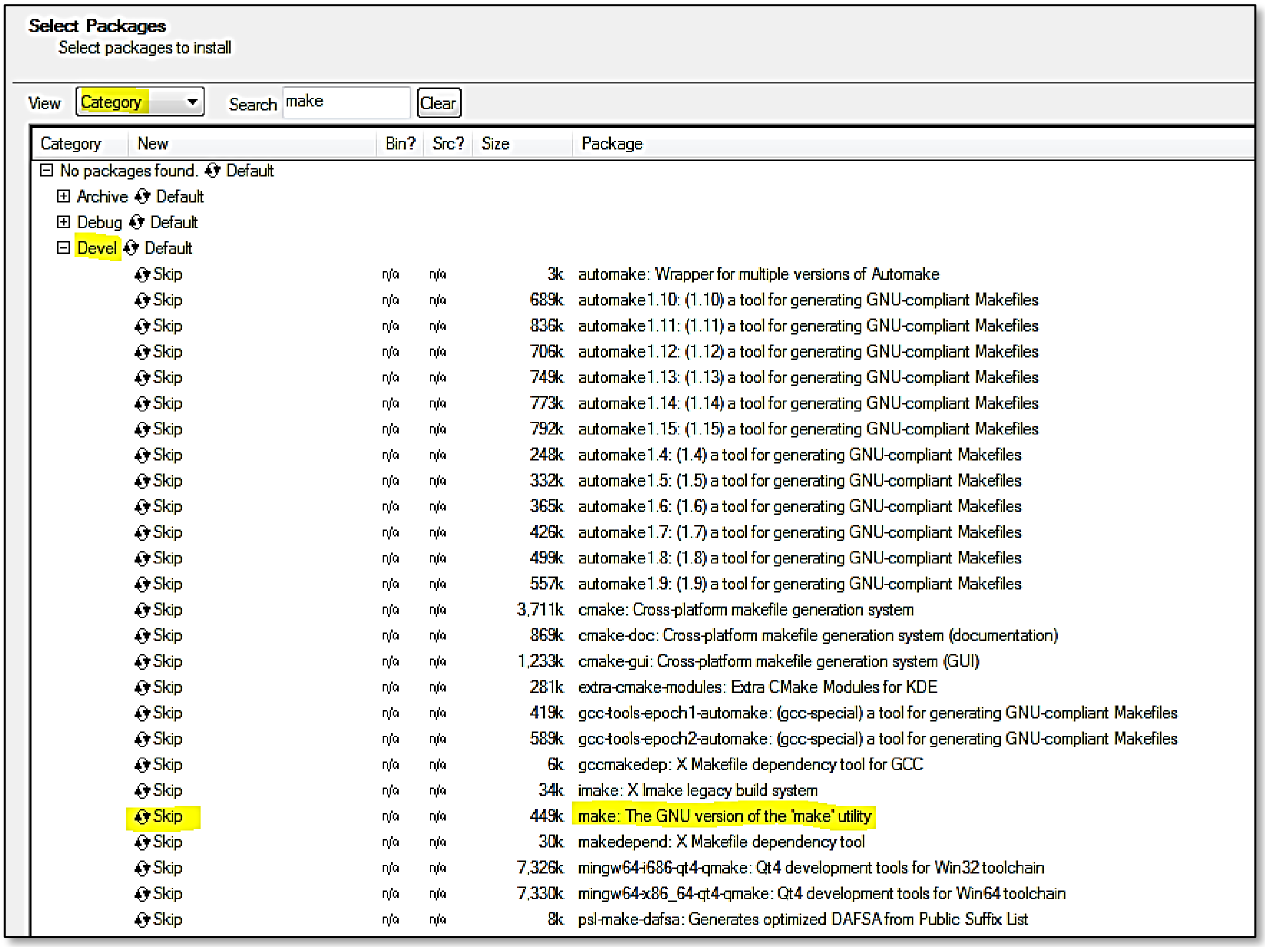
- Now is the time to select the packages carefully (Figure 4). There are basic packages selected by default for installation. However, the compilers need to be chosen. Ensure to select GNU compilers (for Fortran, C, & C++), subversion and additional tools. You can use the search field to select the following packages under the corresponding listed categories:

  i. Devel
    - Make utility (Highly recommended**)**
    - gcc4
    - gcc4-fortran
    - pkg-config
    - subversion (will also be tested)

  ii. Archive –unzip

  iii. Web-wget

  iv. Utils:patch



Select Packages

Select packages to install

View: Category

Search: make

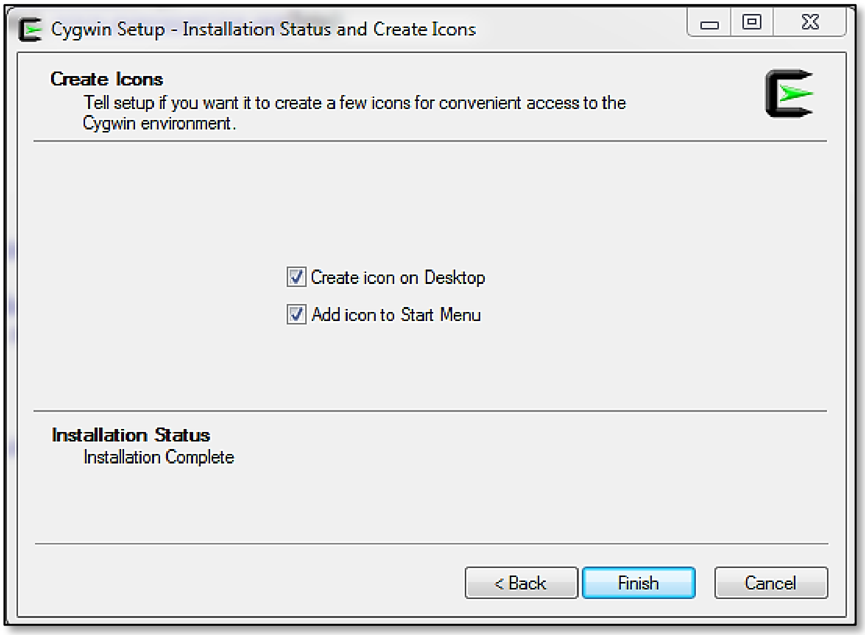
Clear

Category	New	Bin?	Src?	Size	Package
⊟ No packages found. ⟳ Default					
⊞ Archive ⟳ Default					
⊞ Debug ⟳ Default					
⊟ Devel ⟳ Default					
	⟳ Skip	n/a	n/a	3k	automake: Wrapper for multiple versions of Automake
	⟳ Skip	n/a	n/a	689k	automake1.10: (1.10) a tool for generating GNU-compliant Makefiles
	⟳ Skip	n/a	n/a	836k	automake1.11: (1.11) a tool for generating GNU-compliant Makefiles
	⟳ Skip	n/a	n/a	706k	automake1.12: (1.12) a tool for generating GNU-compliant Makefiles
	⟳ Skip	n/a	n/a	749k	automake1.13: (1.13) a tool for generating GNU-compliant Makefiles
	⟳ Skip	n/a	n/a	773k	automake1.14: (1.14) a tool for generating GNU-compliant Makefiles
	⟳ Skip	n/a	n/a	792k	automake1.15: (1.15) a tool for generating GNU-compliant Makefiles
	⟳ Skip	n/a	n/a	248k	automake1.4: (1.4) a tool for generating GNU-compliant Makefiles
	⟳ Skip	n/a	n/a	332k	automake1.5: (1.5) a tool for generating GNU-compliant Makefiles
	⟳ Skip	n/a	n/a	365k	automake1.6: (1.6) a tool for generating GNU-compliant Makefiles
	⟳ Skip	n/a	n/a	426k	automake1.7: (1.7) a tool for generating GNU-compliant Makefiles
	⟳ Skip	n/a	n/a	499k	automake1.8: (1.8) a tool for generating GNU-compliant Makefiles
	⟳ Skip	n/a	n/a	557k	automake1.9: (1.9) a tool for generating GNU-compliant Makefiles
	⟳ Skip	n/a	n/a	3,711k	cmake: Cross-platform makefile generation system
	⟳ Skip	n/a	n/a	869k	cmake-doc: Cross-platform makefile generation system (documentation)
	⟳ Skip	n/a	n/a	1,233k	cmake-gui: Cross-platform makefile generation system (GUI)
	⟳ Skip	n/a	n/a	281k	extra-cmake-modules: Extra CMake Modules for KDE
	⟳ Skip	n/a	n/a	419k	gcc-tools-epoch1-automake: (gcc-special) a tool for generating GNU-compliant Makefiles
	⟳ Skip	n/a	n/a	589k	gcc-tools-epoch2-automake: (gcc-special) a tool for generating GNU-compliant Makefiles
	⟳ Skip	n/a	n/a	6k	gccmakedep: X Makefile dependency tool for GCC
	⟳ Skip	n/a	n/a	34k	imake: X Imake legacy build system
	⟳ Skip	n/a	n/a	449k	make: The GNU version of the 'make' utility
	⟳ Skip	n/a	n/a	30k	makedepend: X Makefile dependency tool
	⟳ Skip	n/a	n/a	7,326k	mingw64-i686-qt4-qmake: Qt4 development tools for Win32 toolchain
	⟳ Skip	n/a	n/a	7,330k	mingw64-x86_64-qt4-qmake: Qt4 development tools for Win64 toolchain
	⟳ Skip	n/a	n/a	8k	psl-make-dafsa: Generates optimized DAFSA from Public Suffix List

Toggle the skip to version number. There is no harm in selecting the auto-make files. Click next when done selecting the packages. Click next again to resolve dependencies. Caveat: Note that some sites have outdated packages that may not work. A test to ensure the packages from a site are functioning will be done at the end of the installation. If the test fails, run the setup file again to install the packages from another mirror site. The site from Pennsylvania is a good first guess.

Figure 4 : Package selection

- Click Finish.
  Congratulations! You are ready to get a feel of Linux in Windows



Cygwin Setup - Installation Status and Create Icons

Create Icons
Tell setup if you want it to create a few icons for convenient access to the Cygwin environment.

☑ Create icon on Desktop
☑ Add icon to Start Menu

Installation Status
Installation Complete

Figure 5 : Cygwin installtion completed

- Test the Installation
  i. Navigate to the Cygwin.bat file (C:\Program Files\Cygwin64\cygwin.bat) and click on it to open the Cygwin Terminal.
  ii. Type the command:

```
$ which make
```

  - If it returns no such file, then click the setup file and choose another mirror site to install the packages. But if it returns a location of the make file (/usr/bin/make) Congratulations!
  - Do same same with svn:

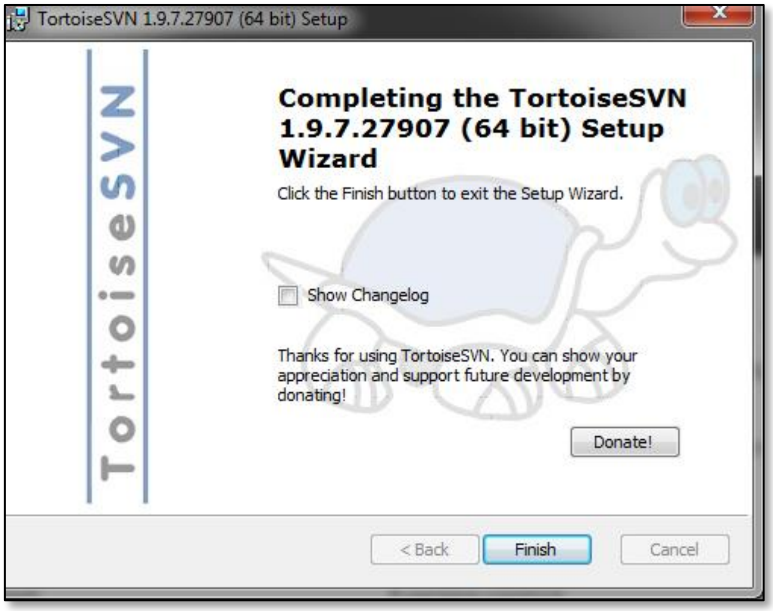
```
$ which svn
```

  iii. You may need to install the listed packages from different mirror sites. The setup-x86_64.exe (in this case for windows 64) is used for updating packages.

2. Subversion Client Installation (SVN)

There are various SVN clients (http://subversion.apache.org.) however TortoiseSVN is recommended.

i. From the website http://tortoisesvn.tigris.org/, click on download and save the file. Follow all instructions, (click Next) and then Finish.



TortoiseSVN 1.9.7.27907 (64 bit) Setup

Completing the TortoiseSVN 1.9.7.27907 (64 bit) Setup Wizard

Click the Finish button to exit the Setup Wizard.

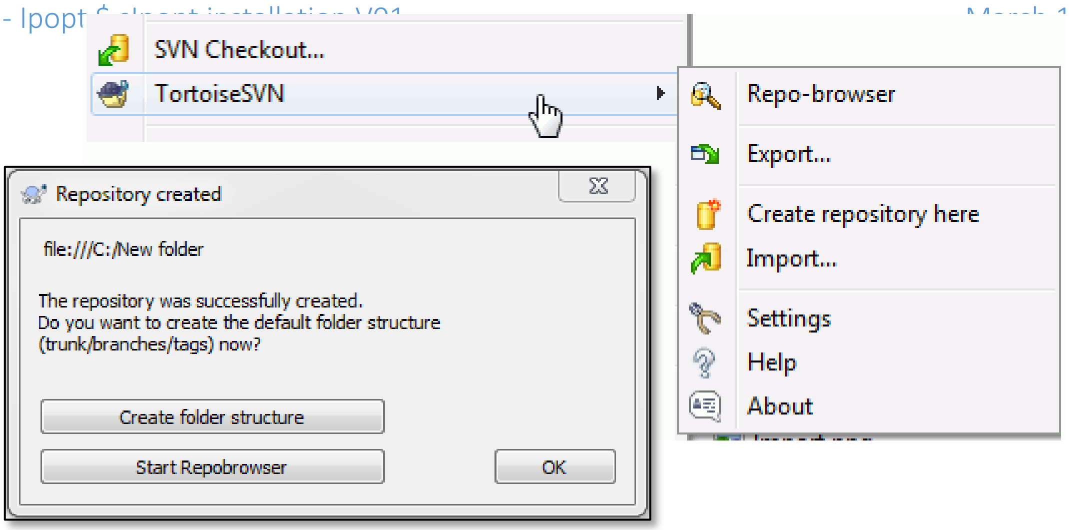
□ Show Changelog

Thanks for using TortoiseSVN. You can show your appreciation and support future development by donating!

Figure 6 :TortoiseSVN installation

ii. Create a new folder in C drive and rename it to SVN (C:\SVN)

iii. Right click on the folder and select tortoiseSVN (Figure 6) and select create repository here. Now click Ok. Congratulations once again, you have installed SVN and initialized a local repository.



file:///C:/New folder

The repository was successfully created.
Do you want to create the default folder structure (trunk/branches/tags) now?

Figure 7 :Creating TortoiseSVN repository and intialization

3. IPOPT Code Download

i. Navigate to the Cygwin.bat file (C:\Program Files\Cygwin64\cygwin.bat) and double-click to open the Cygwin Terminal

ii. Test to see if the installed SVN handles "https" scheme. Type the following in the Cygwin Terminal:

```
$ svn --version
```

iii. Find out the stable version of IPOPt to install by entering the following command in the Cygwin terminal:

```
$ svn list https://projects.coin-or.org/svn/Ipopt/releases
```

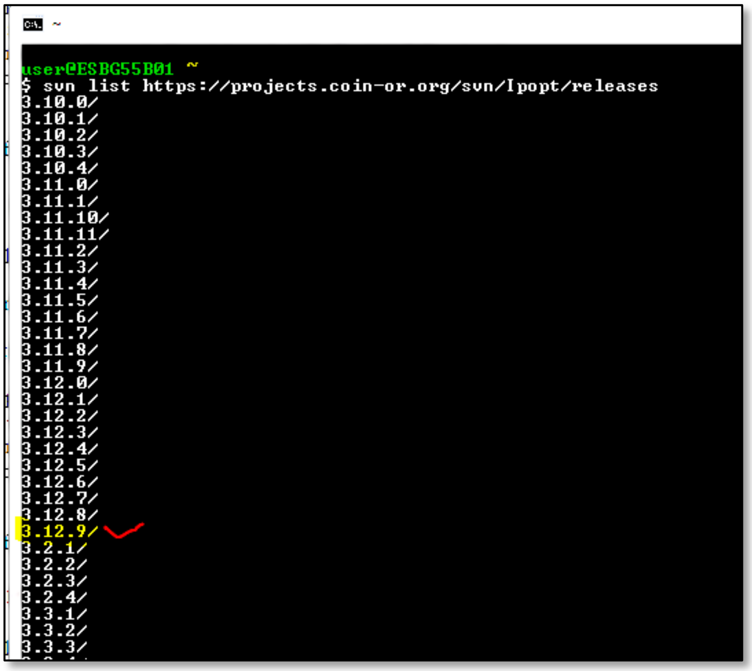
Select the largest number. (as shown in the following picture)

iv. Download the code from the repository into a new folder 'CoinIpopt' with the command:

```
$ svn checkout https://projects.coin-or.org/svn/Ipopt/releases/3.12.9 CoinIpopt
```

v. This can be updated when there is a new release by navigating to this folder and entering the command:

```
$ svn switch https://projects.coin-or.org/svn/Ipopt/releases/3.12.9
```



```
user@ESBG55B01 ~
$ svn list https://projects.coin-or.org/svn/Ipopt/releases
3.10.0/
3.10.1/
3.10.2/
3.10.3/
3.10.4/
3.11.0/
3.11.1/
3.11.10/
3.11.11/
3.11.2/
3.11.3/
3.11.4/
3.11.5/
3.11.6/
3.11.7/
3.11.8/
3.11.9/
3.12.0/
3.12.1/
3.12.2/
3.12.3/
3.12.4/
3.12.5/
3.12.6/
3.12.7/
3.12.8/
3.12.9/
3.2.1/
3.2.2/
3.2.3/
3.2.4/
3.3.1/
3.3.2/
3.3.3/
```

Figure 8 : Ipopt stable releases (time of writing this document)

4. Third-Party or External Code Installation

Blas, Lapack, ASL, Mumps and Metis will be downloaded. Details about them can be found on the website.

i. Navigate to the folder “ThirdParty” with the command:

```
$ cd CoinIpopt/ThirdParty
```

(you can check your current directory [pwd] and check the folders it contains [ls –la])

ii. Get Blas, Lapack and ASL with the command:

```
$ cd Blas; ./get.Blas; cd ./Lapack; ./get.Lapack; cd ../ASL; ./get.ASL
```

iii. Get Mumps and Metis:

```
$ cd../Mumps; ./get.Mumps; cd ../Metis; ./get.Metis; cd ../..
```

Congratulations if all went well!

5. Compiling and Installing IPOP

i. Create a folder called build in CoinIpopt folder (i.e CoinIpopt\build) with the command:

```
$ mkdir build
```

(ensure you are in C:\CoinIpopt)

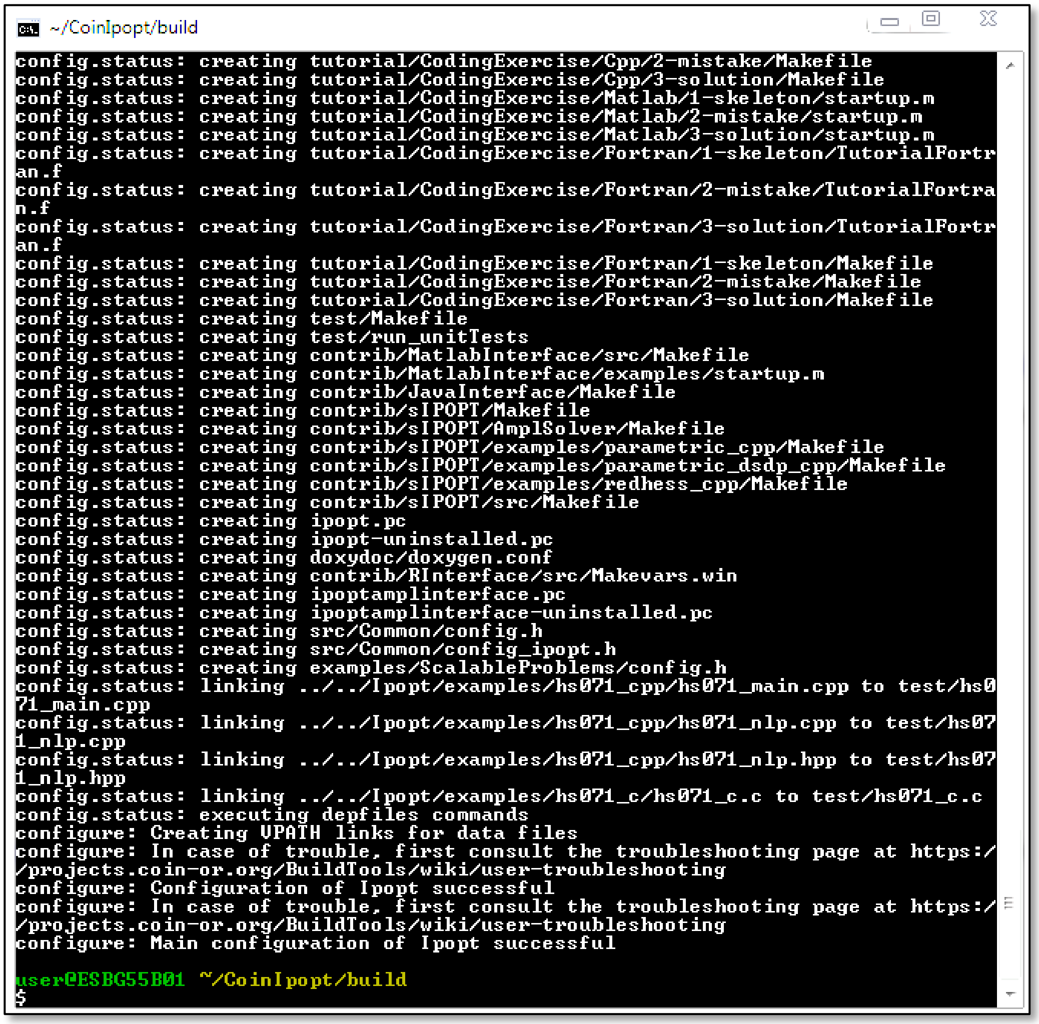
ii. Navigate into the build folder

```
$ cd build
```

iii. Enter the command to compile Ipopt:

```
$ ../configure
```

The configuration takes a couple of minutes . It should end with the message: *"Configure: Main configuration of ipopt successful"*



```
config.status: creating tutorial/CodingExercise/Cpp/2-mistake/Makefile
config.status: creating tutorial/CodingExercise/Cpp/3-solution/Makefile
config.status: creating tutorial/CodingExercise/Matlab/1-skeleton/startup.m
config.status: creating tutorial/CodingExercise/Matlab/2-mistake/startup.m
config.status: creating tutorial/CodingExercise/Matlab/3-solution/startup.m
config.status: creating tutorial/CodingExercise/Fortran/1-skeleton/TutorialFortr
an.f
config.status: creating tutorial/CodingExercise/Fortran/2-mistake/TutorialFortra
n.f
config.status: creating tutorial/CodingExercise/Fortran/3-solution/TutorialFortr
an.f
config.status: creating tutorial/CodingExercise/Fortran/1-skeleton/Makefile
config.status: creating tutorial/CodingExercise/Fortran/2-mistake/Makefile
config.status: creating tutorial/CodingExercise/Fortran/3-solution/Makefile
config.status: creating test/Makefile
config.status: creating test/run_unitTests
config.status: creating contrib/MatlabInterface/src/Makefile
config.status: creating contrib/MatlabInterface/examples/startup.m
config.status: creating contrib/JavaInterface/Makefile
config.status: creating contrib/sIPOPT/Makefile
config.status: creating contrib/sIPOPT/AmplSolver/Makefile
config.status: creating contrib/sIPOPT/examples/parametric_cpp/Makefile
config.status: creating contrib/sIPOPT/examples/parametric_dsdp_cpp/Makefile
config.status: creating contrib/sIPOPT/examples/redhess_cpp/Makefile
config.status: creating contrib/sIPOPT/src/Makefile
config.status: creating ipopt.pc
config.status: creating ipopt-uninstalled.pc
config.status: creating doxydoc/doxygen.conf
config.status: creating contrib/RInterface/src/Makevars.win
config.status: creating ipoptamplinterface.pc
config.status: creating ipoptamplinterface-uninstalled.pc
config.status: creating src/Common/config.h
config.status: creating src/Common/config_ipopt.h
config.status: creating examples/ScalableProblems/config.h
config.status: linking ../../Ipopt/examples/hs071_cpp/hs071_main.cpp to test/hs0
71_main.cpp
config.status: linking ../../Ipopt/examples/hs071_cpp/hs071_nlp.cpp to test/hs07
1_nlp.cpp
config.status: linking ../../Ipopt/examples/hs071_cpp/hs071_nlp.hpp to test/hs07
1_nlp.hpp
config.status: linking ../../Ipopt/examples/hs071_c/hs071_c.c to test/hs071_c.c
config.status: executing depfiles commands
configure: Creating VPATH links for data files
configure: In case of trouble, first consult the troubleshooting page at https:/
/projects.coin-or.org/BuildTools/wiki/user-troubleshooting
configure: Configuration of Ipopt successful
configure: In case of trouble, first consult the troubleshooting page at https:/
/projects.coin-or.org/BuildTools/wiki/user-troubleshooting
configure: Main configuration of Ipopt successful

user@ESBG55B01 ~/CoinIpopt/build
$
```

Figure 9 : Snippet of a successful Ipopt configuration

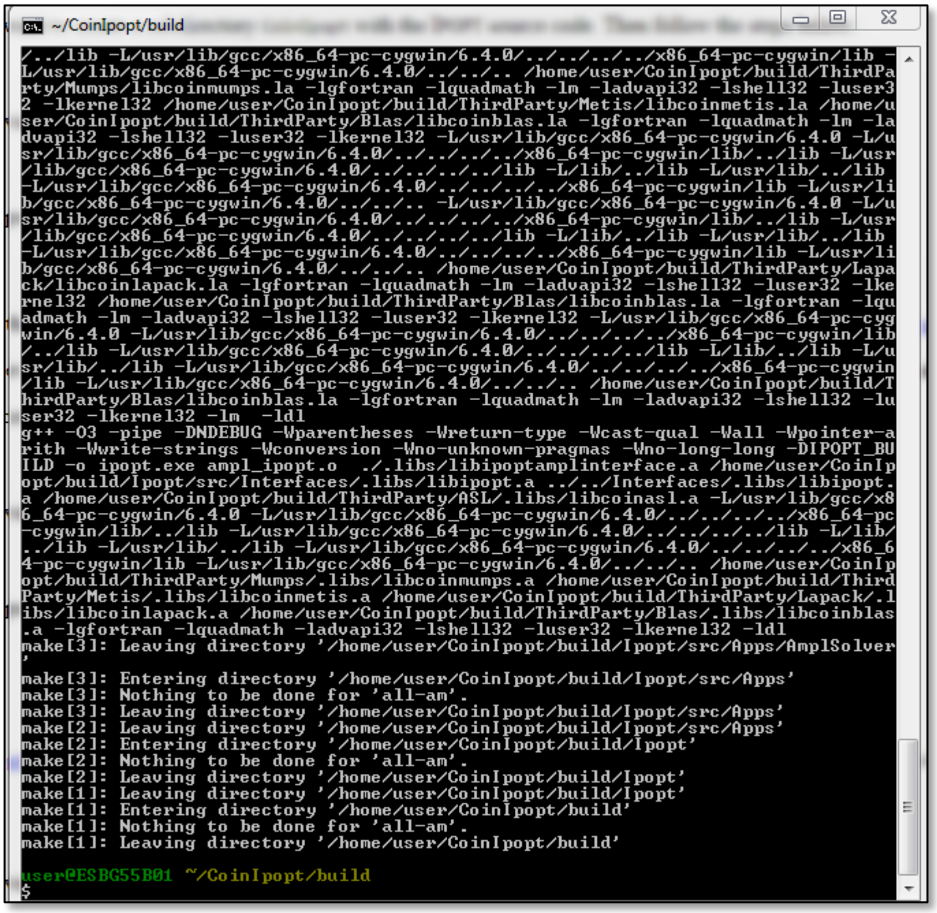
iv. Next enter:

```
$ make
```

The last output should be:

"Make [1]: Leaving directory 'home/...../CoinIpopt/build'"

See below.



```
/../lib -L/usr/lib/gcc/x86_64-pc-cygwin/6.4.0/../../../../x86_64-pc-cygwin/lib -
L/usr/lib/gcc/x86_64-pc-cygwin/6.4.0/../../.. /home/user/CoinIpopt/build/ThirdPa
rty/Mumps/libcoinmumps.la -lgfortran -lquadmath -lm -ladvapi32 -lshell32 -luser3
2 -lkernel32 /home/user/CoinIpopt/build/ThirdParty/Metis/libcoinmetis.la /home/u
ser/CoinIpopt/build/ThirdParty/Blas/libcoinblas.la -lgfortran -lquadmath -lm -la
dvapi32 -lshell32 -luser32 -lkernel32 -L/usr/lib/gcc/x86_64-pc-cygwin/6.4.0 -L/u
sr/lib/gcc/x86_64-pc-cygwin/6.4.0/../../../../x86_64-pc-cygwin/lib/../lib -L/usr
/lib/gcc/x86_64-pc-cygwin/6.4.0/../../../../lib -L/lib/../lib -L/usr/lib/../lib
-L/usr/lib/gcc/x86_64-pc-cygwin/6.4.0/../../../../x86_64-pc-cygwin/lib -L/usr/li
b/gcc/x86_64-pc-cygwin/6.4.0/../../.. -L/usr/lib/gcc/x86_64-pc-cygwin/6.4.0 -L/u
sr/lib/gcc/x86_64-pc-cygwin/6.4.0/../../../../x86_64-pc-cygwin/lib/../lib -L/usr
/lib/gcc/x86_64-pc-cygwin/6.4.0/../../../../lib -L/lib/../lib -L/usr/lib/../lib
-L/usr/lib/gcc/x86_64-pc-cygwin/6.4.0/../../../../x86_64-pc-cygwin/lib -L/usr/li
b/gcc/x86_64-pc-cygwin/6.4.0/../../.. /home/user/CoinIpopt/build/ThirdParty/Lapa
ck/libcoinlapack.la -lgfortran -lquadmath -lm -ladvapi32 -lshell32 -luser32 -lke
rnel32 /home/user/CoinIpopt/build/ThirdParty/Blas/libcoinblas.la -lgfortran -lqu
admath -lm -ladvapi32 -lshell32 -luser32 -lkernel32 -L/usr/lib/gcc/x86_64-pc-cyg
win/6.4.0 -L/usr/lib/gcc/x86_64-pc-cygwin/6.4.0/../../../../x86_64-pc-cygwin/lib
/../lib -L/usr/lib/gcc/x86_64-pc-cygwin/6.4.0/../../../../lib -L/lib/../lib -L/u
sr/lib/../lib -L/usr/lib/gcc/x86_64-pc-cygwin/6.4.0/../../../../x86_64-pc-cygwin
/lib -L/usr/lib/gcc/x86_64-pc-cygwin/6.4.0/../../.. /home/user/CoinIpopt/build/T
hirdParty/Blas/libcoinblas.la -lgfortran -lquadmath -lm -ladvapi32 -lshell32 -lu
ser32 -lkernel32 -lm  -ldl
g++ -O3 -pipe -DNDEBUG -Wparentheses -Wreturn-type -Wcast-qual -Wall -Wpointer-a
rith -Wwrite-strings -Wconversion -Wno-unknown-pragmas -Wno-long-long -DIPOPT_BU
ILD -o ipopt.exe ampl_ipopt.o  ./.libs/libipoptamplinterface.a /home/user/CoinIp
opt/build/Ipopt/src/Interfaces/.libs/libipopt.a ../../Interfaces/.libs/libipopt.
a /home/user/CoinIpopt/build/ThirdParty/ASL/.libs/libcoinasl.a -L/usr/lib/gcc/x8
6_64-pc-cygwin/6.4.0 -L/usr/lib/gcc/x86_64-pc-cygwin/6.4.0/../../../../x86_64-pc
-cygwin/lib/../lib -L/usr/lib/gcc/x86_64-pc-cygwin/6.4.0/../../../../lib -L/lib/
../lib -L/usr/lib/../lib -L/usr/lib/gcc/x86_64-pc-cygwin/6.4.0/../../../../x86_6
4-pc-cygwin/lib -L/usr/lib/gcc/x86_64-pc-cygwin/6.4.0/../../.. /home/user/CoinIp
opt/build/ThirdParty/Mumps/.libs/libcoinmumps.a /home/user/CoinIpopt/build/Third
Party/Metis/.libs/libcoinmetis.a /home/user/CoinIpopt/build/ThirdParty/Lapack/.l
ibs/libcoinlapack.a /home/user/CoinIpopt/build/ThirdParty/Blas/.libs/libcoinblas
.a -lgfortran -lquadmath -ladvapi32 -lshell32 -luser32 -lkernel32 -ldl
make[3]: Leaving directory '/home/user/CoinIpopt/build/Ipopt/src/Apps/AmplSolver
'

make[3]: Entering directory '/home/user/CoinIpopt/build/Ipopt/src/Apps'
make[3]: Nothing to be done for 'all-am'.
make[3]: Leaving directory '/home/user/CoinIpopt/build/Ipopt/src/Apps'
make[2]: Leaving directory '/home/user/CoinIpopt/build/Ipopt/src/Apps'
make[2]: Entering directory '/home/user/CoinIpopt/build/Ipopt'
make[2]: Nothing to be done for 'all-am'.
make[2]: Leaving directory '/home/user/CoinIpopt/build/Ipopt'
make[1]: Leaving directory '/home/user/CoinIpopt/build/Ipopt'
make[1]: Entering directory '/home/user/CoinIpopt/build'
make[1]: Nothing to be done for 'all-am'.
make[1]: Leaving directory '/home/user/CoinIpopt/build'

user@ESBG55B01 ~/CoinIpopt/build
$
```

Figure 10 : Snippet of a successful Ipopt compilation

v. Next enter:

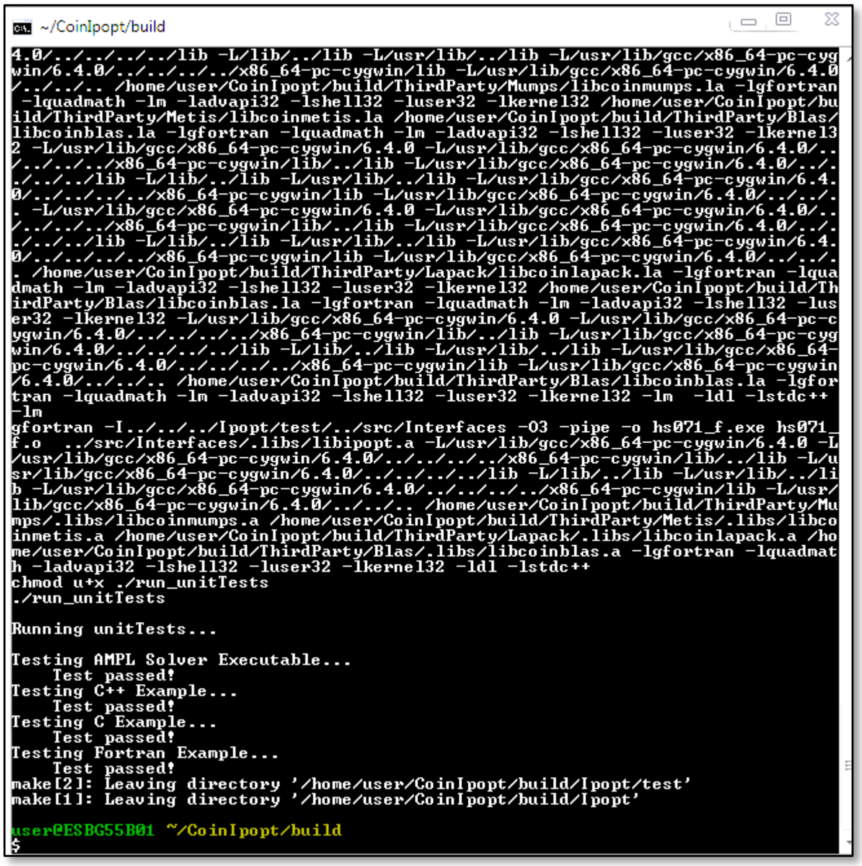
```
$ make test
```

The output screen :

"tests passed for AMPL, C++, C and Fortran example"

vi. Finally, copy the generated libraries and the executable (ipopt.exe) to CoinIpopt/build/bin with the command:

```
$ make install
```



```
4.0/../../../../lib -L/lib/../lib -L/usr/lib/../lib -L/usr/lib/gcc/x86_64-pc-cyg
win/6.4.0/../../../../x86_64-pc-cygwin/lib -L/usr/lib/gcc/x86_64-pc-cygwin/6.4.0
/../../.. /home/user/CoinIpopt/build/ThirdParty/Mumps/libcoinmumps.la -lgfortran
 -lquadmath -lm -ladvapi32 -lshell32 -luser32 -lkernel32 /home/user/CoinIpopt/bu
ild/ThirdParty/Metis/libcoinmetis.la /home/user/CoinIpopt/build/ThirdParty/Blas/
libcoinblas.la -lgfortran -lquadmath -lm -ladvapi32 -lshell32 -luser32 -lkernel3
2 -L/usr/lib/gcc/x86_64-pc-cygwin/6.4.0 -L/usr/lib/gcc/x86_64-pc-cygwin/6.4.0/..
/../../../x86_64-pc-cygwin/lib/../lib -L/usr/lib/gcc/x86_64-pc-cygwin/6.4.0/../.
./../../lib -L/lib/../lib -L/usr/lib/../lib -L/usr/lib/gcc/x86_64-pc-cygwin/6.4.
0/../../../../x86_64-pc-cygwin/lib -L/usr/lib/gcc/x86_64-pc-cygwin/6.4.0/../../.
. -L/usr/lib/gcc/x86_64-pc-cygwin/6.4.0 -L/usr/lib/gcc/x86_64-pc-cygwin/6.4.0/..
/../../../x86_64-pc-cygwin/lib/../lib -L/usr/lib/gcc/x86_64-pc-cygwin/6.4.0/../.
./../../lib -L/lib/../lib -L/usr/lib/../lib -L/usr/lib/gcc/x86_64-pc-cygwin/6.4.
0/../../../../x86_64-pc-cygwin/lib -L/usr/lib/gcc/x86_64-pc-cygwin/6.4.0/../../.
. /home/user/CoinIpopt/build/ThirdParty/Lapack/libcoinlapack.la -lgfortran -lqua
dmath -lm -ladvapi32 -lshell32 -luser32 -lkernel32 /home/user/CoinIpopt/build/Th
irdParty/Blas/libcoinblas.la -lgfortran -lquadmath -lm -ladvapi32 -lshell32 -lus
er32 -lkernel32 -L/usr/lib/gcc/x86_64-pc-cygwin/6.4.0 -L/usr/lib/gcc/x86_64-pc-c
ygwin/6.4.0/../../../../x86_64-pc-cygwin/lib/../lib -L/usr/lib/gcc/x86_64-pc-cyg
win/6.4.0/../../../../lib -L/lib/../lib -L/usr/lib/../lib -L/usr/lib/gcc/x86_64-
pc-cygwin/6.4.0/../../../../x86_64-pc-cygwin/lib -L/usr/lib/gcc/x86_64-pc-cygwin
/6.4.0/../../.. /home/user/CoinIpopt/build/ThirdParty/Blas/libcoinblas.la -lgfor
tran -lquadmath -lm -ladvapi32 -lshell32 -luser32 -lkernel32 -lm  -ldl -lstdc++
-lm
gfortran -I../../../Ipopt/test/../src/Interfaces -O3 -pipe -o hs071_f.exe hs071_
f.o  ../src/Interfaces/.libs/libipopt.a -L/usr/lib/gcc/x86_64-pc-cygwin/6.4.0 -L
/usr/lib/gcc/x86_64-pc-cygwin/6.4.0/../../../../x86_64-pc-cygwin/lib/../lib -L/u
sr/lib/gcc/x86_64-pc-cygwin/6.4.0/../../../../lib -L/lib/../lib -L/usr/lib/../li
b -L/usr/lib/gcc/x86_64-pc-cygwin/6.4.0/../../../../x86_64-pc-cygwin/lib -L/usr/
lib/gcc/x86_64-pc-cygwin/6.4.0/../../.. /home/user/CoinIpopt/build/ThirdParty/Mu
mps/.libs/libcoinmumps.a /home/user/CoinIpopt/build/ThirdParty/Metis/.libs/libco
inmetis.a /home/user/CoinIpopt/build/ThirdParty/Lapack/.libs/libcoinlapack.a /ho
me/user/CoinIpopt/build/ThirdParty/Blas/.libs/libcoinblas.a -lgfortran -lquadmat
h -ladvapi32 -lshell32 -luser32 -lkernel32 -ldl -lstdc++
chmod u+x ./run_unitTests
./run_unitTests

Running unitTests...

Testing AMPL Solver Executable...
    Test passed!
Testing C++ Example...
    Test passed!
Testing C Example...
    Test passed!
Testing Fortran Example...
    Test passed!
make[2]: Leaving directory '/home/user/CoinIpopt/build/Ipopt/test'
make[1]: Leaving directory '/home/user/CoinIpopt/build/Ipopt'

user@ESBG55B01 ~/CoinIpopt/build
$
```

Figure 11 : Snippet of a successful Installation Test

6. sIPOPT Installation

i. Navigate to the folder CoinIpopt/build/Ipopt/contrib/sIpopt. Assuming your current working directory is still build, enter the command:

```
$ cd /Ipopt/contrib/sIpopt
```

ii. Enter the command:

```
$ make
```

iii. Next enter the command:

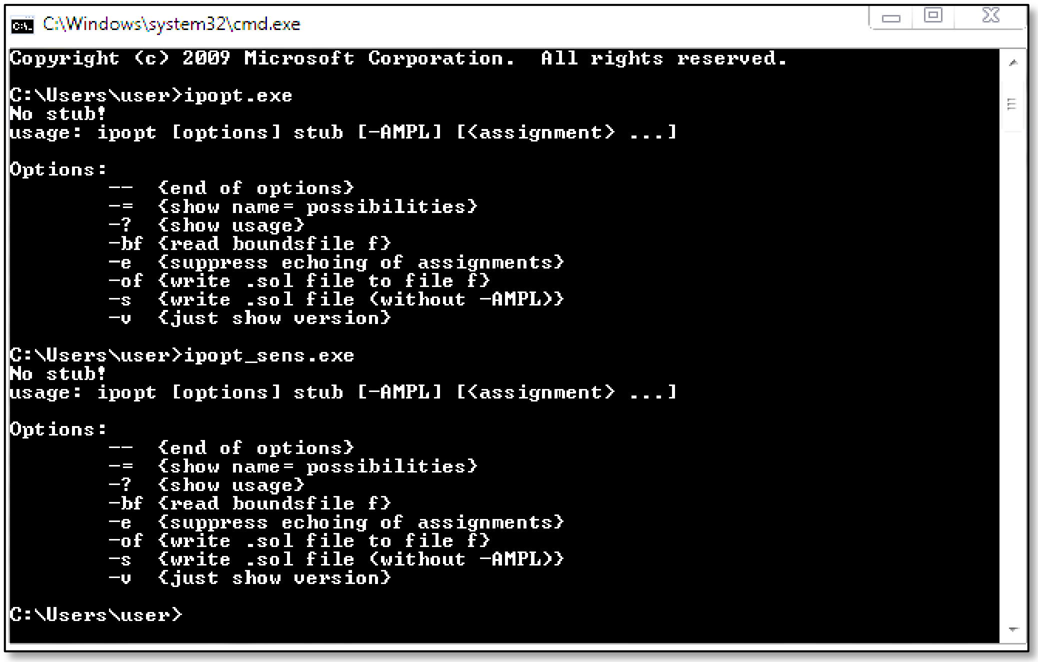
```
$ make install
```

This will generate ipopt_sens.exe and copy it to \CoinIpopt\build\bin.

7. Ipopt.exe and Ipopt_sens.exe test

To test the Ample executables in Windows environment, it is good to run them on the Windows command line and fix any problem that may arise (e.g. path variable settings or missing Cygwin1.dll)

i. Add the location of the Ample executables (C:/Cygwin/CoinIpopt/build/bin) to path.
ii. Open the command line and run ipopt.exe. You should get the output below:
iii. Run also the ipopt_sens.exe and obtain the same output. Now we are ready to hook these solvers to our code!

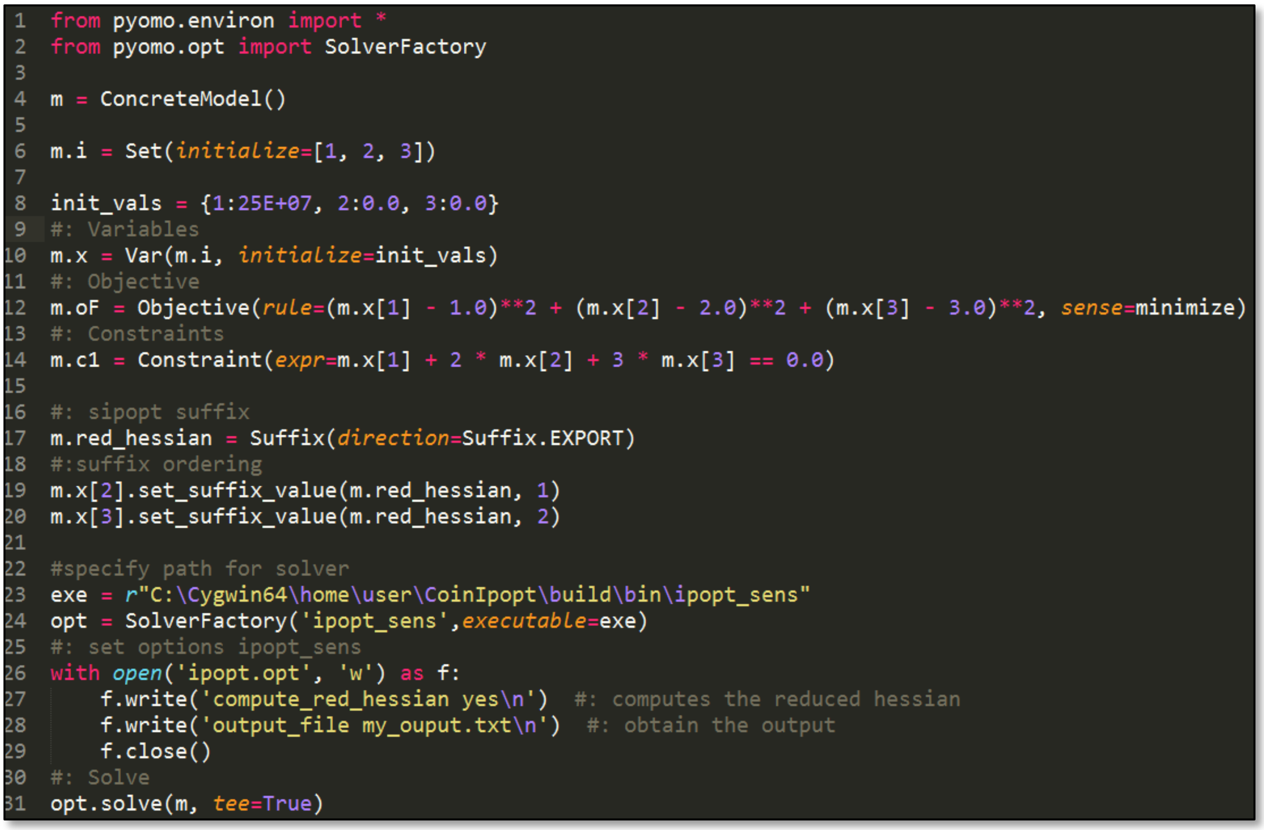


```
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\user>ipopt.exe
No stub!
usage: ipopt [options] stub [-AMPL] [<assignment> ...]

Options:
        --  {end of options}
        -=  {show name= possibilities}
        -?  {show usage}
        -bf {read boundsfile f}
        -e  {suppress echoing of assignments}
        -of {write .sol file to file f}
        -s  {write .sol file (without -AMPL)}
        -v  {just show version}

C:\Users\user>ipopt_sens.exe
No stub!
usage: ipopt [options] stub [-AMPL] [<assignment> ...]

Options:
        --  {end of options}
        -=  {show name= possibilities}
        -?  {show usage}
        -bf {read boundsfile f}
        -e  {suppress echoing of assignments}
        -of {write .sol file to file f}
        -s  {write .sol file (without -AMPL)}
        -v  {just show version}

C:\Users\user>
```

Figure 12 : Test of AMPL executables on windows command line

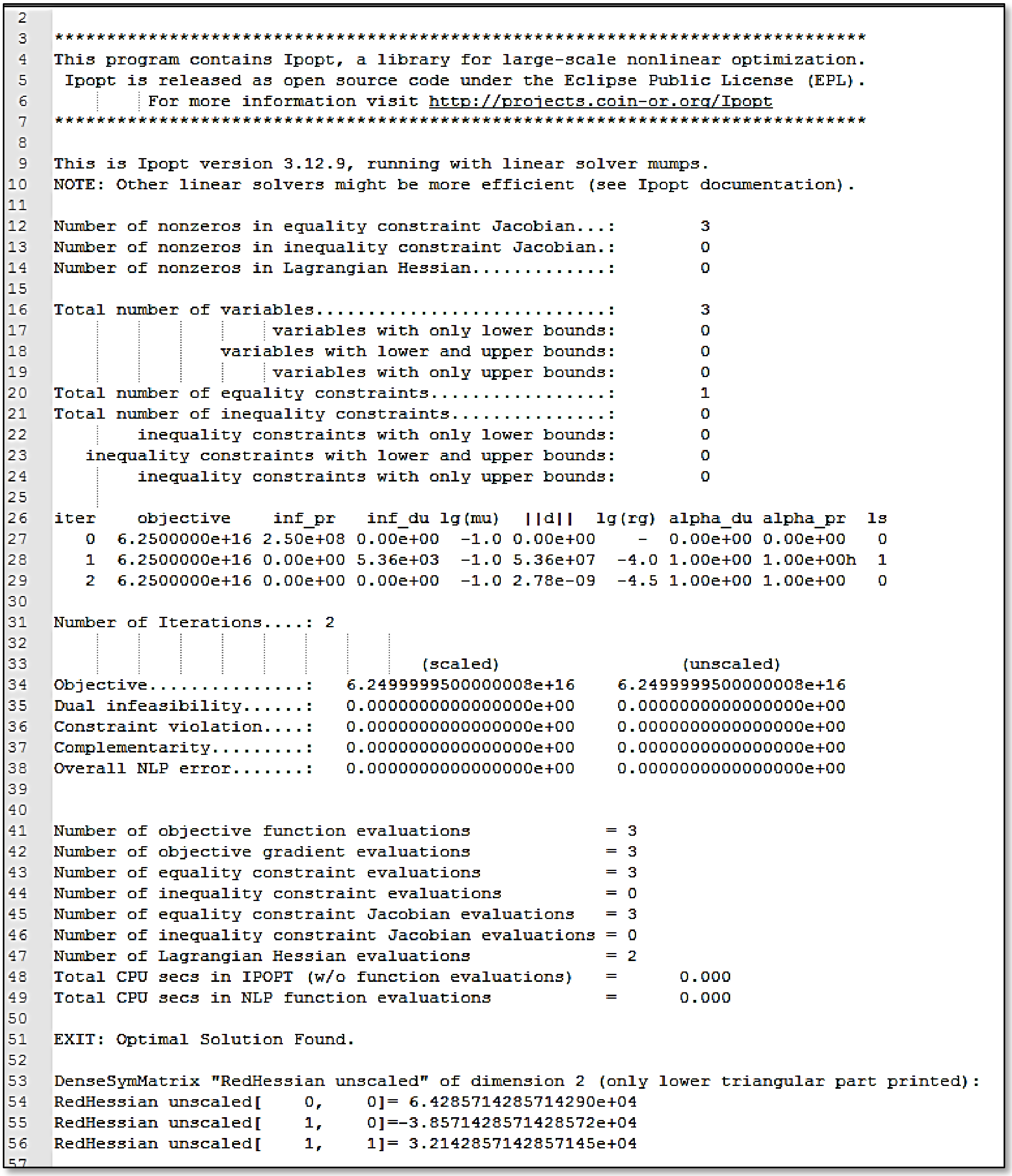
Example

Figure 13 shows the example in the sIpopt Reference Manual while the output file showing the result of the problem is shown in figure 14.



```
from pyomo.environ import *
from pyomo.opt import SolverFactory

m = ConcreteModel()

m.i = Set(initialize=[1, 2, 3])

init_vals = {1:25E+07, 2:0.0, 3:0.0}
#: Variables
m.x = Var(m.i, initialize=init_vals)
#: Objective
m.oF = Objective(rule=(m.x[1] - 1.0)**2 + (m.x[2] - 2.0)**2 + (m.x[3] - 3.0)**2, sense=minimize)
#: Constraints
m.c1 = Constraint(expr=m.x[1] + 2 * m.x[2] + 3 * m.x[3] == 0.0)

#: sipopt suffix
m.red_hessian = Suffix(direction=Suffix.EXPORT)
#:suffix ordering
m.x[2].set_suffix_value(m.red_hessian, 1)
m.x[3].set_suffix_value(m.red_hessian, 2)

#specify path for solver
exe = r"C:\Cygwin64\home\user\CoinIpopt\build\bin\ipopt_sens"
opt = SolverFactory('ipopt_sens',executable=exe)
#: set options ipopt_sens
with open('ipopt.opt', 'w') as f:
    f.write('compute_red_hessian yes\n')  #: computes the reduced hessian
    f.write('output_file my_ouput.txt\n')  #: obtain the output
    f.close()
#: Solve
opt.solve(m, tee=True)
```

Figure 13: Snippet of sIpopt usage in Pyomo



```

******************************************************************************
This program contains Ipopt, a library for large-scale nonlinear optimization.
 Ipopt is released as open source code under the Eclipse Public License (EPL).
         For more information visit http://projects.coin-or.org/Ipopt
******************************************************************************

This is Ipopt version 3.12.9, running with linear solver mumps.
NOTE: Other linear solvers might be more efficient (see Ipopt documentation).

Number of nonzeros in equality constraint Jacobian...:        3
Number of nonzeros in inequality constraint Jacobian.:        0
Number of nonzeros in Lagrangian Hessian.............:        0

Total number of variables............................:        3
                     variables with only lower bounds:        0
                variables with lower and upper bounds:        0
                     variables with only upper bounds:        0
Total number of equality constraints.................:        1
Total number of inequality constraints...............:        0
        inequality constraints with only lower bounds:        0
   inequality constraints with lower and upper bounds:        0
        inequality constraints with only upper bounds:        0

iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du alpha_pr  ls
   0  6.2500000e+16 2.50e+08 0.00e+00  -1.0 0.00e+00    -  0.00e+00 0.00e+00   0
   1  6.2500000e+16 0.00e+00 5.36e+03  -1.0 5.36e+07  -4.0 1.00e+00 1.00e+00h  1
   2  6.2500000e+16 0.00e+00 0.00e+00  -1.0 2.78e-09  -4.5 1.00e+00 1.00e+00   0

Number of Iterations....: 2

                                   (scaled)                 (unscaled)
Objective...............:   6.2499999500000008e+16    6.2499999500000008e+16
Dual infeasibility......:   0.0000000000000000e+00    0.0000000000000000e+00
Constraint violation....:   0.0000000000000000e+00    0.0000000000000000e+00
Complementarity.........:   0.0000000000000000e+00    0.0000000000000000e+00
Overall NLP error.......:   0.0000000000000000e+00    0.0000000000000000e+00


Number of objective function evaluations             = 3
Number of objective gradient evaluations             = 3
Number of equality constraint evaluations            = 3
Number of inequality constraint evaluations          = 0
Number of equality constraint Jacobian evaluations   = 3
Number of inequality constraint Jacobian evaluations = 0
Number of Lagrangian Hessian evaluations             = 2
Total CPU secs in IPOPT (w/o function evaluations)   =      0.000
Total CPU secs in NLP function evaluations           =      0.000

EXIT: Optimal Solution Found.

DenseSymMatrix "RedHessian unscaled" of dimension 2 (only lower triangular part printed):
RedHessian unscaled[    0,     0]= 6.4285714285714290e+04
RedHessian unscaled[    1,     0]=-3.8571428571428572e+04
RedHessian unscaled[    1,     1]= 3.2142857142857145e+04
```

Figure 14: Output file for sIpopt problem in Pyomo

References

1) https://www.coin-or.org/Ipopt/documentation/

2) https://www.cygwin.com/

3) https://tortoisesvn.net/

4) Pirnay, Hans & Opez-Negrete, Rodrigo & Biegler, Lorenz. (2018). sIPOPT Reference Manual.