Install cantera-magma

in titanv

목차 : local python3 설치, local python2 설치, cuda 설치, magma 설치, sundials 설치, sundials-magma 설치, local scons 설치, cantera-sundials 설치, cantera-magma 설치, cantera ignition test

1. local python3 설치

(참고 : https://danielerikssson.net/2017/02/08/how-to-install-latest-python-on-centos/)

< 주의사항 >

python3 설치 후 python2를 설치해야 한다. 2먼저 설치하면 3가 2를 덮어쓸 수 있다.

< python 설치 전 필요한 모듈 설치 >

```
$ sudo apt install make build-essential libssl-dev zlib1g-dev
$ sudo apt install libbz2-dev libreadline-dev libsqlite3-dev wget curl llvm
$ sudo apt install libncurses5-dev libncursesw5-dev xz-utils tk-dev
```

< python 설치 >

```
$ wget http://python.org/ftp/python/3.6.5/Python-3.6.5tar.xz
```

(원하는 버전으로 바꿀 것)

```
$ tar xvf Python-3.6.5.tar.xz
$ cd Python-3.6.5
$ ./configure --prefix=$HOME/local/
$ make && make altinstall
```

make install을 사용하면 system에 기본 설치된 python3($ sudo apt install python3 로 설치된)을 덮어쓸 수 있다. make altinstall을 하면 python3와 python3.6으로 구별된다.

```
$ export PATH=$HOME/local/bin:$PATH >> ~/.bashrc
```

$ python3.6 (local python3)

```
kimms@titan:~$ python3.6
Python 3.6.5 (default, Jul 25 2018, 10:39:47)
[GCC 5.4.0 20160609] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

(exit:ctrl+D)

cf) $ python3 (system default python3)

```
kimms@titan:~$ python3
Python 3.5.2 (default, Nov 23 2017, 16:37:01)
[GCC 5.4.0 20160609] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

2. local python2 설치

```
$ wget http://python.org/ftp/python/2.7.14/Python-2.7.14.tar.xz
$ tar xf Python-2.7.14.tar.xz
$ cd Python-2.7.14
$ ./configure --prefix=$HOME/local/ --enable-unicode=ucs4
```

memory 낭비를 줄이고 호환성을 높이기 위해 unicode support 옵션을 준다.

```
$ make && make altinstall
$ export PATH=$HOME/local/bin:$PATH >> ~/.bashrc
```

3에서 PATH를 추가해 줬다면 필요 없는 작업이다.

$ python2.7

```
kimms@titan:~$ python2.7
Python 2.7.15 (default, Jul 25 2018, 10:49:19)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

(exit:ctrl+D)

cf) system python

```
kimms@titan:~$ python
Python 2.7.12 (default, Dec  4 2017, 14:50:18)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

< cantera설치에 필요한 python모듈 설치 >

pip command

pip install [packagename]
pip install --upgrade [packagename]
pip uninstall [packagename]
pip install [packagename]== (설치 가능한 버전을 보여준다)
pip install [packagename]==1.2.1 (원하는 버전의 모듈 설치)
pip install --force-reinstall [packagename]==1.2.1 (원하는 버전의 모듈 재설치)
pip install –Iv [packagename]==1.2.1 (다른 버전이 있어도 강제로 설치한다)

```
$ pip3.6 install --upgrade pip
$ pip3.6 install numpy
$ pip3.6 install cython
$ pip3.6 install 3to2
$ wget https://bootstrap.pypa.io/get-pip.py
$ python2.7 get-pip.py
$ pip2.7 install --upgrade pip
$ pip2.7 install numpy
$ pip2.7 install 3to2
$ pip2.7 install cython
```

python3는 pip이 기본으로 설치되고 python2는 기본 설치되지 않는다.

$HOME/local/bin에 python3, pip3.6 등이 설치되고 $HOME/local/lib/python3.6/site-packages에 각종 파이썬 모듈과 pip으로 설치한 패키지 등이 설치된다(setuptools, wheel, numpy 등).

< 설치 확인 >

$ python3.6 –c 'import packagename'

ex)

```
$ python3.6 –c 'import numpy'
```

< pip, pip2, pip2.7 구별 >

pip: $PATH에서 순서대로 python을 찾아 가장 먼저 나오는 python.

예를 들어 $PATH=$HOME/local/bin:/usr/local/bin 두 경로 다 python이 있으면 pip이 $HOME/local/bin의 python 모듈을 설치한다

pip2: $PATH에서 순서대로 python2를 찾아 가장 먼저 나오는 python2.

pip2.7: $PATH에서 순서대로 python2.7을 찾아 가장 먼저 나오는 python2.7.

< 설치 중 발생할 수 있는 에러 >

```
zipimport.ZipImportError: can't decompress data; zlib not available
Makefile:1109: recipe for target 'altinstall' failed
make: *** [altinstall] Error 1
```

zlib이 없거나 버전이 이상한 것

$ sudo apt-get zlib-dev (--update)

```
kimms@titan:~/usr/local/bin$ python2.7 get-pip.py
pip is configured with locations that require TLS/SSL, however the ssl module in Pytho
n is not available.
Collecting pip
  Retrying (Retry(total=4, connect=None, read=None, redirect=None, status=None)) after
 connection broken by 'SSLError("Can't connect to HTTPS URL because the SSL module is
not available.",)': /simple/pip/
```

ssl에 문제가 있는 것이다.

$ sudo apt install libssl-dev (--update)

```
kimms@titan:~/local/python2.7/bin$ pip2.7 install 2to3
Collecting 2to3
  Could not find a version that satisfies the requirement 2to3 (from versions: )
No matching distribution found for 2to3
```

오타거나, 저런 이름의 모듈이 없거나, pip으로 설치 불가한 모듈이다.

3. cuda 설치

< CUDA 설치를 위한 준비단계 >

```
$ sudo apt install linux-headers-$(uname -r)
$ sudo apt purge nvidia*
$ sudo apt-get install freeglut3-dev build-essential libx11-dev libxmu-dev libxi-dev libgl1-mesa-glx
libglu1-mesa libglu1-mesa-dev libglfw3-dev libgles2-mesa-dev
```

< CUDA toolkit & driver 설치 >

```
$ wget
https://developer.nvidia.com/compute/cuda/9.2/Prod/local_installers/cuda_9.2.88_396.26_linux
$ chmod +x cuda_9.2.88_396.26_linux
$ sudo service lightdm stop
$ sudo sh ./cuda_9.2.88_396.26_linux
```

물어보는 것의 default 답이 있으면 그냥 enter 로 하고, 질문에는 y 로 답하면 된다.

(주의: Xconfig 바꿀거냐 default 가 no 이고, 그대로 no 해야 함)

```
$ vi ~/.bashrc
```

(끝에 다음 2 줄을 추가)

export PATH=$PATH:/usr/local/cuda-9.2/bin
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/cuda-9.2/lib64

CUDA 작동테스트:

```
$ ./NVIDIA_CUDA_9.2_Samples/0_Simple/clock/clock
```

4. magma 설치

magma : GPU 버전의 선형대수라이브러리

```
$ sudo apt install libopenblas-dev libopenmpi-dev
$ wget http://icl.cs.utk.edu/projectsfiles/magma/downloads/magma-2.0.2.tar.gz
$ tar xzvf magma-2.0.2.tar.gz
$ cd magma-2.0.2
$ cp make.inc.openblas make.inc
$ export CUDADIR=/usr/local/cuda
$ vi Makefile
```

comment out lines for compute_20 etc

```
#MIN_ARCH ?= 200
#NV_SM       += -gencode arch=compute_20,code=sm_20
#NV_COMP    := -gencode arch=compute_20,code=compute_20
```

```
$ make -j 6
$ cd testing
$ ./testing_dgetrf
$ export LD_LIBRARY_PATH=$HOME/src/magma-2.0.2 >> ~/.bashrc
```

5. sundials 설치

버전마다 설치법이 조금씩 다르므로 유의할 것 (아래는 sundials-2.5.0. install note에 버전에 따른 설치법이 나와있다)

```
$ wget https://launchpad.net/ubuntu/+archive/primary/+sourcefiles/sundials/2.5.0-1/sundials_2.5.0.orig.tar.gz
$ tar xvzf sundials_2.5.0.orig.tar.gz
$ ./configure --prefix=$HOME/local/sundials-2.5.0 --exec-prefix=$HOME/local/sundials-2.5.0 --with-cflags=-fPIC --disable-mpi CC=gcc F77=gfortran
$ make && make install
```

6. Sundials-MAGMA 설치

sundials의 CVODE와 CVODES solver가 GPU기반 LU factorization을 할 수 있도록 개조된 sundials이다(MAGMA 라이브러리를 참조한다). CUDA와 MAGMA가 설치되어 있어야 하며 설치법은 INSTALL_NOTES에 자세히 나와있다.

magma 함수의 역할 등이 궁금하면

```
$ vi magma-2.0.2/docs/documentation.txt
```

에 자세히 나와있다.

```
$ git clone https://github.com/athlonshi/Sundials-MAGMA.git --recursive
$ cd Sundials-MAGMA
$ grep –r magmablas *
```

libmagmablas는 magma 구버전에 있던 library인데 지금은 libmagma로 통합되어서 삭제해야 한다.

```
$ vi src/cvode/cvode_gpu.c
$ vi src/cvodes/cvodes_gpu.c
```

delete –lmagmablas

CVodeInitGPU()에서 MAGMA_CUDA_INIT() 부분을 지우고 magma_init()으로 다음과 같이 바꾼다. (magma를 initialize하는 함수 이름이 바뀌었다)

```
if (type == 1) {
  cv_mem->GPU = TRUE;
  //MAGMA_CUDA_INIT();
  magma_init();
}
```

```
$ grep –r printout_devices
```

printout_devices()도 magma 구버전에 있던 함수인데 지금은 magma_print_environment()로 바뀌였다. 다음 두 방법 중 하나를 택하면 된다.

방법 1. printout_devices()가 있는 부분을 magma_print_environment()로 바꾼다.

```
$ vi include/cvode/cvode_gpu.h
$ vi include/cvodes/cvodes_gpu.h
```

```
#define MAGMA_CUDA_INIT()                                                   \
    if( CUBLAS_STATUS_SUCCESS != cublasInit() ) {                           \
        fprintf(stderr, "ERROR: cublasInit failed\n");                      \
        exit(-1);                                                           \
    }                                                                       \
    magma_print_environment();        \
    //printout_devices();                                                  \
```

방법 2. printout_devices를 define해주면 된다.

```
$ vi src/cvode/cvode_gpu.c
#include <cuda.h>
#include <cuda_runtime_api.h>
#include <cublas.h>
```

```
#include <cuda.h>
#include <cuda_runtime_api.h>
#include <cublas.h>
```

headerfile 목록에 추가

```
/* ////////////////////////////////////////////////////////////////////////////
      -- Print the available GPU devices
 */
 void printout_devices( )
 {
    int ndevices, idevice;
    cuDeviceGetCount( &ndevices );
    for( idevice = 0; idevice < ndevices; idevice++ )
       {
          char name[200];
 #if CUDA_VERSION > 3010
          size_t totalMem;
 #else
          unsigned int totalMem;
 #endif

          int clock;
          CUdevice dev;

          cuDeviceGet( &dev, idevice );
          cuDeviceGetName( name, sizeof(name), dev );
          cuDeviceTotalMem( &totalMem, dev );
          cuDeviceGetAttribute( &clock,
                                CU_DEVICE_ATTRIBUTE_CLOCK_RATE, dev );
          printf( "device %d: %s, %.1f MHz clock, %.1f MB memory\n",
                  idevice, name, clock/1000.f, totalMem/1024.f/1024.f );
       }
 }
```

```
/* ////////////////////////////////////////////////////////////////
   -- Print the available GPU devices
*/
void printout_devices( )
{
  int ndevices, idevice;
  cuDeviceGetCount( &ndevices );
  for( idevice = 0; idevice < ndevices; idevice++ )
    {
      char name[200];
#if CUDA_VERSION > 3010
      size_t totalMem;
#else
      unsigned int totalMem;
#endif

      int clock;
      CUdevice dev;

      cuDeviceGet( &dev, idevice );
      cuDeviceGetName( name, sizeof(name), dev );
      cuDeviceTotalMem( &totalMem, dev );
      cuDeviceGetAttribute( &clock,
                            CU_DEVICE_ATTRIBUTE_CLOCK_RATE, dev );
      printf( "device %d: %s, %.1f MHz clock, %.1f MB memory\n",
              idevice, name, clock/1000.f, totalMem/1024.f/1024.f );
    }
}
```

$ vi src/cvodes/cvodes_gpu.c

```
#include <cuda.h>
#include <cuda_runtime_api.h>
#include <cublas.h>


/* ////////////////////////////////////////////////////////////////////////
      -- Print the available GPU devices
 */
 void printout_devices( )
 {
    int ndevices, idevice;
    cuDeviceGetCount( &ndevices );
    for( idevice = 0; idevice < ndevices; idevice++ )
       {
          char name[200];
 #if CUDA_VERSION > 3010
          size_t totalMem;
 #else
          unsigned int totalMem;
 #endif

          int clock;
          CUdevice dev;

          cuDeviceGet( &dev, idevice );
          cuDeviceGetName( name, sizeof(name), dev );
          cuDeviceTotalMem( &totalMem, dev );
          cuDeviceGetAttribute( &clock,
                                CU_DEVICE_ATTRIBUTE_CLOCK_RATE, dev );
          printf( "device %d: %s, %.1f MHz clock, %.1f MB memory\n",
                  idevice, name, clock/1000.f, totalMem/1024.f/1024.f );
       }
 }
```

앞서 했던 작업과 동일하다

DenseGETRFGPU(DlsMat A, long int *p) 내에서

printf("============> Matrix size: %d %d\n", M, N); //추가 (gpu를 사용하는지 확인하는 용도)

//magma_dgetmatrix(M, N, d_A, ldda, A->data, lda);

(comment out)

return(info); //추가

```
long int DenseGETRFGPU(DlsMat A, long int *p)
{
  int i;
  int M = A->M;
  int N = A->N;
  int lda = M;
  int ldda = ((M+31)/32)*32;
  int info;

  /*Call MAGMA LU factorization solver*/
  printf("============> Matrix size: %d %d\n", M, N);
  magma_dsetmatrix( M, N, A->data, lda, d_A, ldda );
  magma_dgetrf_gpu( M, N, d_A, ldda, p, &info);
  //magma_dgetmatrix( M, N, d_A, ldda, A->data, lda );

  return(info);

}
```

DenseGETRSGPU(DlsMat A, long int *p, realtype *B) 내에서

//magma_dgetrs_gpu('N', M, NRHS, d_A, ldda, p, d_B, lddb, &info); //NO transpose (comment out)

magma_dgetrs_gpu(MagmaNoTrans, M, NRHS, d_A, ldda, p, d_B, lddb, &info); //NO transpose

//추가

return(info); //추가

```
long int DenseGETRSGPU(DlsMat A, long int *p, realtype *B)
{
  int i;
  int M = A->M;
  int N = A->N;
  int ldb = N;
  int ldda = ((M+31)/32)*32;
  int lddb = M;
  int NRHS = 1;
  int info;

  /*Call MAGMA linear equation solver*/
  magma_dsetmatrix( M, NRHS, B, ldb, d_B, lddb );
  //magma_dgetrs_gpu( 'N', M, NRHS, d_A, ldda, p, d_B, lddb, &info );  //NO transpose
  magma_dgetrs_gpu( MagmaNoTrans, M, NRHS, d_A, ldda, p, d_B, lddb, &info );  //NO trans
pose
  magma_dgetmatrix( M, NRHS, d_B, lddb, B, ldb );

return(info);

}
```

```
$ vi include/cvode/cvode_gpu.h
```

```
#define MAGMA_DEVFREE(ptr)                                                  \
        magma_free_internal( ptr, __func__, __FILE__, __LINE__ )
    //magma_free( ptr );
```

```
#define MAGMA_DEVFREE(ptr)                                                  \
        magma_free_internal( ptr, __func__, __FILE__, __LINE__ )
    //magma_free( ptr );
```

```
$ vi include/cvodes/cvodes_gpu.h
```

```
#define magma_free( ptr ) \
        magma_free_internal( ptr, __func__, __FILE__, __LINE__ )

#define magma_free_pinned( ptr ) \
        magma_free_pinned_internal( ptr, __func__, __FILE__, __LINE__ )
//추가
```

```
#define magma_free( ptr ) \
        magma_free_internal( ptr, __func__, __FILE__, __LINE__ )

#define magma_free_pinned( ptr ) \
        magma_free_pinned_internal( ptr, __func__, __FILE__, __LINE__ )
```

```
#define MAGMA_DEVFREE(ptr)                                                  \
        magma_free_internal( ptr, __func__, __FILE__, __LINE__ )
    //magma_free( ptr );
//바꿈
```

```
#define MAGMA_DEVFREE(ptr)                                                  \
        magma_free_internal( ptr, __func__, __FILE__, __LINE__ )
    //magma_free( ptr );
```

```
$ ./configure --cudainclude=/usr/local/cuda-9.2/include --cudalib=/usr/local/cuda-9.2/lib64 --
magmainclude=$HOME/src/magma-2.0.2/include --magmalib=$HOME /src/magma-2.0.2/lib --
prefix=$HOME/local/sundials-2.5.0-magma --exec-prefix=$HOME /local/sundials-2.5.0-magma --
with-cflags=-fPIC --disable-mpi CC=gcc F77=gfortran --enable-examples
$ make
```

< 제대로 작동하는지 확인>

```
$ cd examples/cvodes/serial/
$ ./cvsRoberts_dns
```

```
kimm@titanv:~/src/Sundials-MAGMA/examples/cvodes/serial$ ./cvsRoberts_dns

Call CPU LU Solver.

3-species kinetics problem

At t = 2.6391e-01      y =  9.899653e-01    3.470564e-05    1.000000e-02
    rootsfound[] =   0   1
At t = 4.0000e-01      y =  9.851641e-01    3.386242e-05    1.480205e-02
At t = 4.0000e+00      y =  9.055097e-01    2.240338e-05    9.446793e-02
```

```
$ ./cvsRoberts_dns gpu
```

```
kimm@titanv:~/src/Sundials-MAGMA/examples/cvodes/serial$ ./cvsRoberts_dns gpu

Call GPU LU Solver.

3-species kinetics problem

===========> Matrix size: 3 3
===========> Matrix size: 3 3
===========> Matrix size: 3 3
```

< 주의사항 >

configure option에 --enable-examples를 추가하면 에러가 발생하는데, sundials-magma와 magma버전이 맞지 않아 생기는 문제 같다. (sundials-magma는 magma-1.3.0을 기준으로 만들어졌다) 무시해도 되지만 예제로 테스트 하고 싶으면 약간의 수정이 필요하다.

```
$ cd $HOME/local/sundials-2.5.0-magma/examples/cvodes/serial
$ make
```

```
kimms@titan:~/local/sundials-magma/examples/cvodes/serial$ make
gcc  -fPIC -I/home/kimms/local/sundials-magma/include  -c cvsAdvDiff_ASAi_bnd.c
gcc  -fPIC -I/home/kimms/local/sundials-magma/include  -c cvsAdvDiff_FSA_non.c
gcc  -fPIC -I/home/kimms/local/sundials-magma/include  -c cvsDiurnal_kry_bp.c
gcc  -fPIC -I/home/kimms/local/sundials-magma/include  -c cvsFoodWeb_ASAp_kry.c
gcc  -fPIC -I/home/kimms/local/sundials-magma/include  -c cvsKrylovDemo_prec.c
gcc  -fPIC -I/home/kimms/local/sundials-magma/include  -c cvsAdvDiff_bnd.c
gcc  -fPIC -I/home/kimms/local/sundials-magma/include  -c cvsDirectDemo_ls.c
gcc  -fPIC -I/home/kimms/local/sundials-magma/include  -c cvsDiurnal_kry.c
gcc  -fPIC -I/home/kimms/local/sundials-magma/include  -c cvsHessian_ASA_FSA.c
gcc  -fPIC -I/home/kimms/local/sundials-magma/include  -c cvsRoberts_ASAi_dns.c
gcc  -fPIC -I/home/kimms/local/sundials-magma/include  -c cvsRoberts_dns_uw.c
gcc  -fPIC -I/home/kimms/local/sundials-magma/include  -c cvsDiurnal_FSA_kry.c
gcc  -fPIC -I/home/kimms/local/sundials-magma/include  -c cvsFoodWeb_ASAi_kry.c
gcc  -fPIC -I/home/kimms/local/sundials-magma/include  -c cvsKrylovDemo_ls.c
gcc  -fPIC -I/home/kimms/local/sundials-magma/include  -c cvsRoberts_dns.c
cvsRoberts_dns.c: In function ‘main’:
cvsRoberts_dns.c:169:10: warning: implicit declaration of function ‘CVodesInitGPU’ [-Wimplicit-function-declaration]
   flag = CVodesInitGPU(cvode_mem, type); //set CPU solver
          ^
gcc  -fPIC -I/home/kimms/local/sundials-magma/include  -c cvsRoberts_FSA_dns.c
make: *** No rule to make target 'cvsRoberts_dnsL.o', needed by 'all'.  Stop.
```

위와 같은 error가 발생 수 있는데,

```
$ vi Makefile
```

EXAMPLES = cvsAdvDiff_ASAi_bnd cvsAdvDiff_FSA_non cvsDiurnal_kry_bp cvsFoodWeb_ASAp_kry cvsKrylovDemo_prec cvsAdvDiff_bnd cvsDirectDemo_ls cvsDiurnal_kry cvsHessian_ASA_FSA cvsRoberts_ASAi_dns cvsRoberts_dns_uw cvsDiurnal_FSA_kry cvsFoodWeb_ASAi_kry cvsKrylovDemo_ls cvsRoberts_dns cvsRoberts_FSA_dns cvsRoberts_dnsL cvsAdvDiff_bndL

위 목록에서 cvsRoverts_dnsL 앞에 # 추가하고,

LIBRARIES_BL 맨 앞에 -L/usr/local/cuda-9.2/lib64 -lcuda -lcublas –L/home/username/src/magma-2.0.2/lib -lmagma 를추가한다.

```
$ make clean
$ make
$ ./cvsRoverts_dns (gpu)
```

7. local scons 설치

scons는 보통 시스템에 기본적으로 설치되어 있지만 cantera가 사용하는 python과 scons의 버전이 다르면 에러가 생기는 것을 발견했다. 따라서 scons를 local하게 설치하였다.

또한 scons는 script interpreter로 python2만 사용한다.

(참고 : http://wiki.nmr-relax.com/Multiple_Python_versions)

```
$ wget http://prdownloads.sourceforge.net/scons/scons-3.0.0.tar.gz
$ tar xvzf scons-3.0.0.tar.gz
$ cd scons-3.0.0/script
$ vi scons
```

첫 줄의 /usr/bin/env python을 다음과 같이 수정한다.

```
#! /home/kimm/local/bin/python2.7
#
# SCons - a Software Constructor
```

```
$ python2.7 setup.py install –prefix=/home/kimms/local/scons-3.0.0
$ export PATH=~/local/scons-3.0.0/bin:$PATH >>~/.bashrc
```

아래와 같은 결과가 나오면 잘 설치된 것이다.

```
kimms@titan:~$ which scons
/home/kimms/local/scons/bin/scons
```

```
$ vi ~/local/scons/bin/scons
```

```
#! /home/kimm/local/bin/python2.7
#
# SCons - a Software Constructor
```

8. cantera-sundials 설치

(참고 : https://www.cantera.org/docs/sphinx/html/index.html)

< prerequisites >

g++ python scons libboost-dev

(python2 module) cython python-dev python-numpy python-numpy-dev python-setuptools

(python3 module) cython python3 python3-dev python3-detuptools python3-numpy

앞에서 필요한 모듈은 거의 설치했고, libboost-dev와 gcc가 system에 없다면 설치해야 한다.

python setuptools는 python설치시 기본 설치된다.

$ sudo apt install gcc libboost-dev

```
$ git clone --recursive https://github.com/Cantera/cantera.git
$ mv cantera cantera-sundials
$ cd cantera-sundials
$ scons build prefix=$HOME/local/cantera-2.4.0-sundials
python2_cmd=$HOME/local/bin/python2.7 python3_cmd=$HOME/local/bin/python3.6
python2_package=full python3_package=full sundials_include=$HOME/local/sundials-
2.5.0/include/ sundials_libdir=$HOME/local/sundials-2.5.0/lib/ CC=gcc CXX=g++
FORTRAN=gfortran env_vars=all –j 6
```

cantera는 sundials3.x버전과 호환되지 않으므로 2.x버전으로 빌드해야 한다.

```
$ scons test
$ scons install
```

만약 칸테라 버전을 바꿔서 설치하고 싶으면 다음과 같이 하면 된다.(기본은 2.4.0b1)

```
$ git tag --list
$ git checkout tags/v2.1.2
```

(나머지 동일. test 전에 아래와 같은 수정이 필요하다)

```
$ vi test/data/kineticsfromscratch.cti
```

species = “”” h2o2: H2 H O O2 OH H2O HO2 H2O2 AR ”””
change it to
species = “”” h2o2: AR O H2 H OH O2 H2O H2O2 HO2 ”””

< 설치 후 작업 >

```
$ vi ~/local/cantera-2.4.0-sundials/bin/setup_cantera
```

```
if [ "/home/kimm/local/bin/python3.6" != `which python` ]; then
   alias ctpython=/home/kimm/local/bin/python3.6
```

만약 위와 같은 부분이 있다면 cantera interpreter가 python3.6이기 때문에 python2 example에 문제가 생긴다.

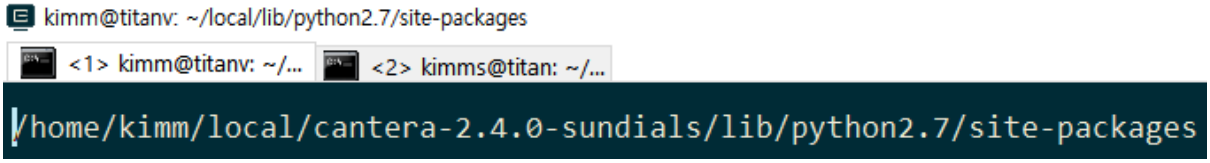
```
#if [ "/home/kimm/local/cantera-2.4.0-sundials/lib/python3.6/site-packages" != "" ]; the
n
#    if [ -z $PYTHONPATH ]; then
#        PYTHONPATH=/home/kimm/local/cantera-2.4.0-sundials/lib/python3.6/site-packages
#    else
#        PYTHONPATH=/home/kimm/local/cantera-2.4.0-sundials/lib/python3.6/site-packages:
$PYTHONPATH
#    fi
#fi
```

위와 같이 PYTHONPATH를 바꿔주는 부분을 comment out 하고 따로 sys.path를 수정해준다.

```
$ cd ~/local/lib/python2.7/site-packages
$ vi cantera-sundials.pth
```

이 디렉토리에 .pth 파일을 추가하면 sys.path에 추가가 된다

~/local/cantera-magma/lib/python2.7/site-packages



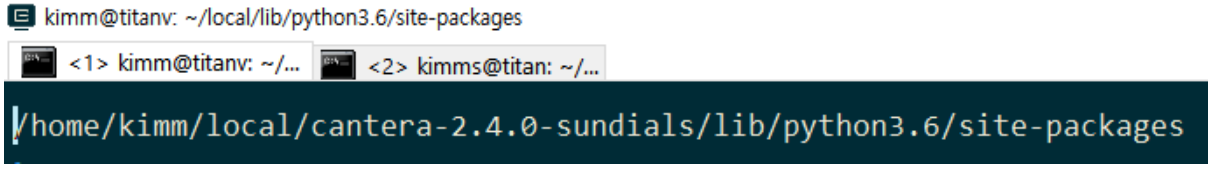
잘 되었는지 확인. import cantera에서 에러가 뜨지 않으면 잘 된 것이다.

```
$ source ~/local/cantera-2.4.0-sundials/bin/setup_cantera
$ python2.7
```

```
kimm@titanv:~/local/cantera-2.4.0-magma$ python2.7
Python 2.7.14 (default, Aug 15 2018, 21:35:22)
[GCC 7.3.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import sys
>>> sys.path
['', '/home/kimm/local/lib/python27.zip', '/home/kimm/local/lib/python2.7', '/home/kimm/
local/lib/python2.7/plat-linux2', '/home/kimm/local/lib/python2.7/lib-tk', '/home/kimm/l
ocal/lib/python2.7/lib-old', '/home/kimm/local/lib/python2.7/lib-dynload', '/home/kimm/l
ocal/lib/python2.7/site-packages', '/home/kimm/local/cantera-2.4.0-magma/lib/python2.7/s
ite-packages', '/home/kimm/local/cantera-2.4.0-sundials/lib/python2.7/site-packages', '/
home/kimm/local/cantera-2.4.0/lib/python2.7/site-packages']
>>> import cantera
>>>
```

python3도 같은 방법으로 수정한다.

```
$ cd ~/local/lib/python3.6/site-packages
$ vi cantera-sundials.pth
```




```
kimm@titanv:~/local/cantera-2.4.0-magma$ python3.6
Python 3.6.5 (default, Aug 15 2018, 21:27:01)
[GCC 7.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import sys
>>> sys.path
['', '/home/kimm/local/lib/python36.zip', '/home/kimm/local/lib/python3.6', '/home/kimm/
local/lib/python3.6/lib-dynload', '/home/kimm/local/lib/python3.6/site-packages', '/home
/kimm/local/cantera-2.4.0-magma/lib/python3.6/site-packages', '/home/kimm/local/cantera-
2.4.0-sundials/lib/python3.6/site-packages', '/home/kimm/local/cantera-2.4.0/lib/python3
.6/site-packages']
>>> import cantera
>>>
```

< example 컴파일 >

```
$ cd ~/local/cantera-2.4.0/sundials/share/cantera/samples
$ cd rankine
$ make
$ ./rankine
```

```
kimms@titan:~/local/cantera-magma/share/cantera/samples/cxx/rankine$ ./rankine
 1            300     101325  -1.58581e+07      3913.17     0
 2s       300.014     800000  -1.58574e+07      3913.17     0
 2        300.126     800000  -1.58569e+07      3914.73     0
 3        443.624     800000  -1.32016e+07      10182.9     1
 4s       373.177     101325   -1.3553e+07      10182.9  0.89
 4        373.177     101325  -1.34827e+07      10371.3  0.92
efficiency = 0.105873
```

```
$ cd ../../f77
$ vi Makefile
```

CPPFLAGS를 다음과 같이 수정

```
CPPFLAGS=$(CANTERA_INCLUDES) -std=c++11
```

```
$ ./isentropic
```

```
kimms@titan:~/local/cantera-magma/share/cantera/samples/f77$ ./isentropic
     0.13047E+02     0.41845E+01     0.27020E+03     0.49751E-02
     0.82256E+01     0.36786E+01     0.32994E+03     0.99502E-02
     0.63008E+01     0.33959E+01     0.37041E+03     0.14925E-01
     0.52263E+01     0.32002E+01     0.40197E+03     0.19900E-01
```

9. cantera-magma 설치

```
$ git clone --recursive https://github.com/Cantera/cantera.git
$ mv cantera cantera-magma
$ cd cantera-magma
```

cantera-sundials와는 다르게 cantera-magma는 GPU를 사용하도록 script를 수정해야 한다.

```
$ vi src/numerics/CVodesIntegrator.cpp
```

#include <cvodes/cvodes_gpu.h>
#include "cvodes/cvodes_dense.h"
#include "cvodes/cvodes_band.h"

위와 같이 헤더파일 추가한다. (if문 안에 넣으면 안 된다)

```
#include "sundials/sundials_types.h"
#include "sundials/sundials_math.h"
#include "sundials/sundials_nvector.h"
#include "nvector/nvector_serial.h"
#include "cvodes/cvodes.h"
#include <cvodes/cvodes_gpu.h>
#include "cvodes/cvodes_dense.h"
#include "cvodes/cvodes_band.h"
```

int flag = CVodeInit(m_cvode_mem, cvodes_rhs, m_to, m_y);

아래에 다음과 같이 추가

//GPU (magma) initialization
// CPU vs GPU switch hard-coded for now
int type = 1;
//int type = 0;
flag = CVodesInitGPU(m_cvode_mem, type);

```
int flag = CVodeInit(m_cvode_mem, cvodes_rhs, m_t0, m_y);
 //GPU (magma) initialization
 // CPU vs GPU switch hard-coded for now
 int type = 1;
 //int type = 0;
 flag = CVodesInitGPU(m_cvode_mem, type);
```

type이 1일 땐 GPU를 쓰고 0일 땐 CPU를 쓴다.

CVLapackBand를 다음과 같이 수정한다.

```
#if CT_SUNDIALS_USE_LAPACK
    //CVLapackBand(m_cvode_mem, N, nu, nl);
    CVBand(m_cvode_mem, N, nu, nl);
#else
    CVBand(m_cvode_mem, N, nu, nl);
#endif
```

CVLapackDense를 다음과 같이 수정한다.

```
#if CT_SUNDIALS_USE_LAPACK
    //CVLapackDense(m_cvode_mem, N);
     CVDense(m_cvode_mem, N);
#else
    CVDense(m_cvode_mem, N);
#endif
```

$ vi SConstruct

defaults.cxxFlags= '-I/usr/local/cuda-9.2/include –I/home/username/src/magma-2.0.2/include'

```
defaults.cxxFlags = '-I/usr/local/cuda-9.2/include -I/home/kimm/src/magma-2.0.2/include'
```

sundials_libs에 다음과 같이 cuda, magma, cublas 추가한다.

```
if env['system_sundials'] == 'y':
    env['sundials_libs'] = ['sundials_cvodes', 'sundials_ida', 'sundials_nvecserial','cuda','magma','cublas']
    if env['use_lapack'] and LooseVersion(env['sundials_version']) >= LooseVersion('3.0'):
        if env.get('has_sundials_lapack'):
            env['sundials_libs'].extend(('sundials_sunlinsollapackdense',
                                         'sundials_sunlinsollapackband'))
        else:
            env['sundials_libs'].extend(('sundials_sunlinsoldense',
                                         'sundials_sunlinsolband'))
```

```
$ scons build prefix=$HOME/local/cantera-magma-test python2_cmd=$HOME/local
/bin/python2.7 python3_package=full python3_cmd=$HOME/local/bin/python3.6 FORTRAN
=gfortran sundials_include=$HOME/local/sundials-2.5.0-magma/include/
sundials_libdir=$HOME/local/sundials-2.5.0-magma/lib/ env_vars=all
extra_inc_dirs=$HOME/src/magma-2.0.2/include/:/usr/local/cuda-9.2/include/
extra_lib_dirs=$HOME/src/magma-2.0.2/lib/:/usr/local/cuda-9.2/lib64/
$ scons install
```

< 설치 후 작업 >

cantera-sundials때와 동일하다.

```
$ vi ~/local/cantera-2.4.0-magma/bin/setup_cantera
```

```
#if [ "/home/kimm/local/cantera-2.4.0-magma/lib/python3.6/site-packages" != "" ]; then
#    if [ -z $PYTHONPATH ]; then
#        PYTHONPATH=/home/kimm/local/cantera-2.4.0-magma/lib/python3.6/site-packages
#    else
#        PYTHONPATH=/home/kimm/local/cantera-2.4.0-magma/lib/python3.6/site-packages:$PY
THONPATH
#    fi
#fi
```

위와 같이 comment out 해준다

```
$ vi ~/local/lib/python2.7/site-packages/cantera-magma.pth
```

type

home/username/local/cantera-2.4.0-magma/lib/python2.7/site-mackages

```
$ vi ~/local/lib/python3.6/site-packages/cantera-magma.pth
```

type

/home/username/local/cantera-2.4.0-magma/lib/python3.6/site-mackages

cantera를 사용하기 전에

```
$ source ~/local/cantera-2.4.0-magma/bin/setup_cantera
```

혹은

$ vi ~/.bashrc

```
#set alias
alias cantera_basic='source /home/kimms/local/cantera/bin/setup_cantera'
alias cantera_magma='source /home/kimms/local/cantera-magma/bin/setup_cantera'
alias cantera_sundials='source /home/kimms/local/cantera-sundials/bin/setup_cantera'
alias cantera_magma_cpu='source /home/kimms/local/cantera-magma-cpu/bin/setup_cantera'
```

위와 같이 별칭을 지정하고

$ cantera_magma

후 사용해도 된다.

10. cantera ignition test

```
$ vi cantera_ingnition_test/gpu/Make.small
```

```
include /home/kimm/local/cantera-2.4.0-magma/include/cantera/Cantera.mak

CC=gcc
CXX=g++
RM=rm -f
CCFLAGS=-g
CPPFLAGS=$(CANTERA_INCLUDES) -std=c++11
LDFLAGS=
GPU_LIBS=-lcuda -L/usr/local/cuda-9.2/lib64 -lmagma -L/home/kimm/src/magma-2.0.2/lib
LDLIBS=$(CANTERA_LIBS) $(GPU_LIBS)

SRCS=kin_small.cpp
OBJS=$(subst .cpp,.o,$(SRCS))

all: kin_small

kin_small: $(OBJS)
	$(CXX) $(LDFLAGS) -o kin_small $(OBJS) $(LDLIBS)

clean:
	$(RM) $(OBJS)

dist-clean: clean
	$(RM) *~
```

위와 같은 Makefile을 만들고

Make.medium, Make.large는 small부분을 medium, large로 바꾸면 된다.

```
$vi setup_cantera
```

```
source ~/local/cantera-2.4.0-magma/bin/setup_cantera
```

```
$ vi run_small
```

```
source ~/src/cantera_ignition_test/gpu/setup_cantera
make -f Make.small
./kin_small > out.small
```

위와 같은 파일을 작성하고

```
$ source run_small
```

결과가 out.small에 저장된다.