Python

numbapro, PyCuda, scikits

NumbaPro

- Automatically handle your function via a decorator

- Example provided in folder, from https://developer.nvidia.com/how-to-cuda-python

- Comes from ContinuumAnalytics (anaconda python distribution)

- Part of package Anaconda Accelerate

- Install with
  conda update conda
  conda install accelerate

- 30 days trial licence…

PyCuda

Getting started tutorial : http://documen.tician.de/pycuda/tutorial.html#transferring-data

- JIT compilation
- Write your own kernels
- Or just use gpuarray
- PO : installation not easy

scikits.cuda : higher level

http://scikit-cuda.readthedocs.org/en/latest/index.html

- pip install scikits.cuda
- Library wrapper functions
  - scikits.cuda.cublas.cublasSgemv(handle, trans, m, n, alpha, A, lda, x, incx, beta, y, incy)
  - Also cufft, cusolver …
- Higher level routines, using pycuda.gpuarray
  - scikits.cuda.linalg.dot(x_gpu, y_gpu, transa='N', transb='N', handle=None, out=None)
  - scikits.cuda.misc.sum, std, cumsum, diff, mean, var…

Matlab

gpuarray

Class gpuArray

- You can create a gpuArray explicitly with a host vector as argument in constructor

- With a standard creation and ‘gpuArray’ argument example : x = rand(M, N, ‘gpuArray’);

Operations on gpuArray

- 3 ways :
  - arrayfun with a custom matlab function
  - use built-in matlab functions that accept gpuArrays (see help)
  - Build a custom kernel (see example in Codes/MATLAB)

Exercise

- Build a fast convolution algorithm with both cpus and gpus arrays and compare time.