Kapok Windows Installation Guide

A brief description of the installation procedure and package dependencies is given in the main Kapok user's manual. Here we provide a more in depth installation guide, particularly aimed at those unfamiliar with Python, or who do not have Python (or a Python package manager) installed. This guide is for Windows operating systems. See "install_guide_mac.pdf" in this folder for the Mac OS X installation guide.

1 Installing Python

The first step is to install Python. In this guide, we use the miniconda Python distribution, available at: http://conda.pydata.org/miniconda.html.

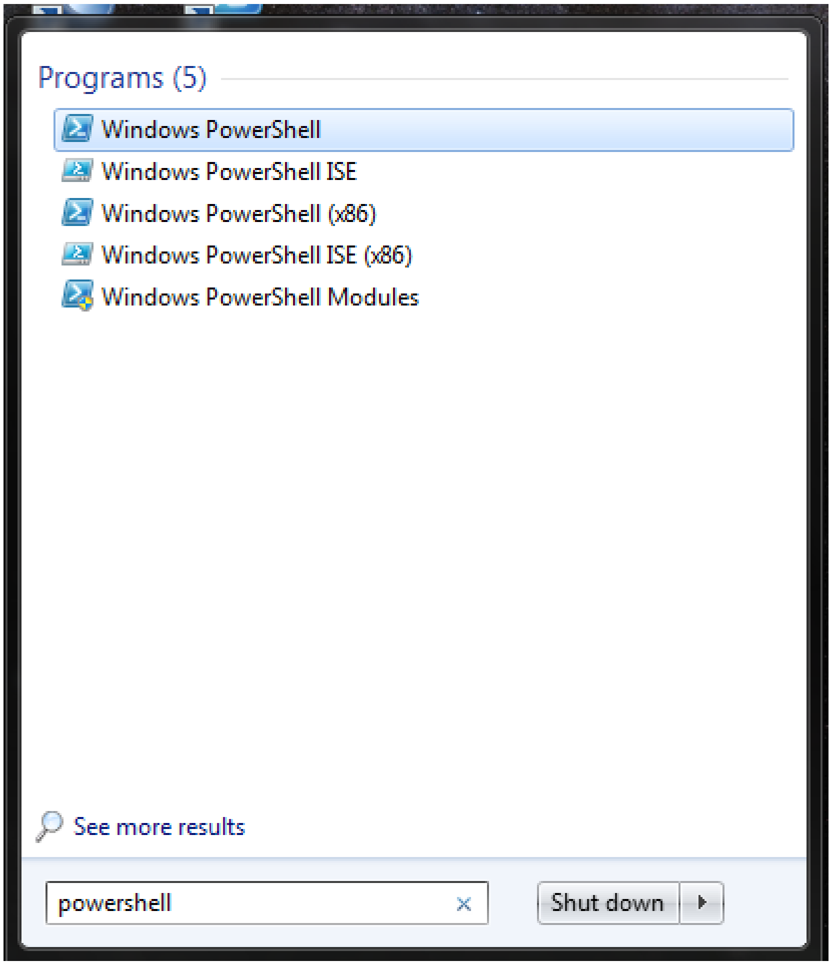
Choose the Python 3.5 version, not the Python 2.7 version. Choose the Windows version appropriate to your system (either 32-bit or 64-bit).

When installing miniconda, it is recommended that you choose the default option to install the program for the current user, rather than for "All Users". Note that if you do install the software for all users, you will need administrator rights in order to install any of the required Python packages in the following steps.

2 Opening a Terminal

Open the Windows PowerShell, if available on your system. To find it, go to the Start Menu, and type PowerShell in the box, then click Windows PowerShell, as shown in Fig. 1.

If Windows PowerShell is not available, you can use the Command Prompt instead.

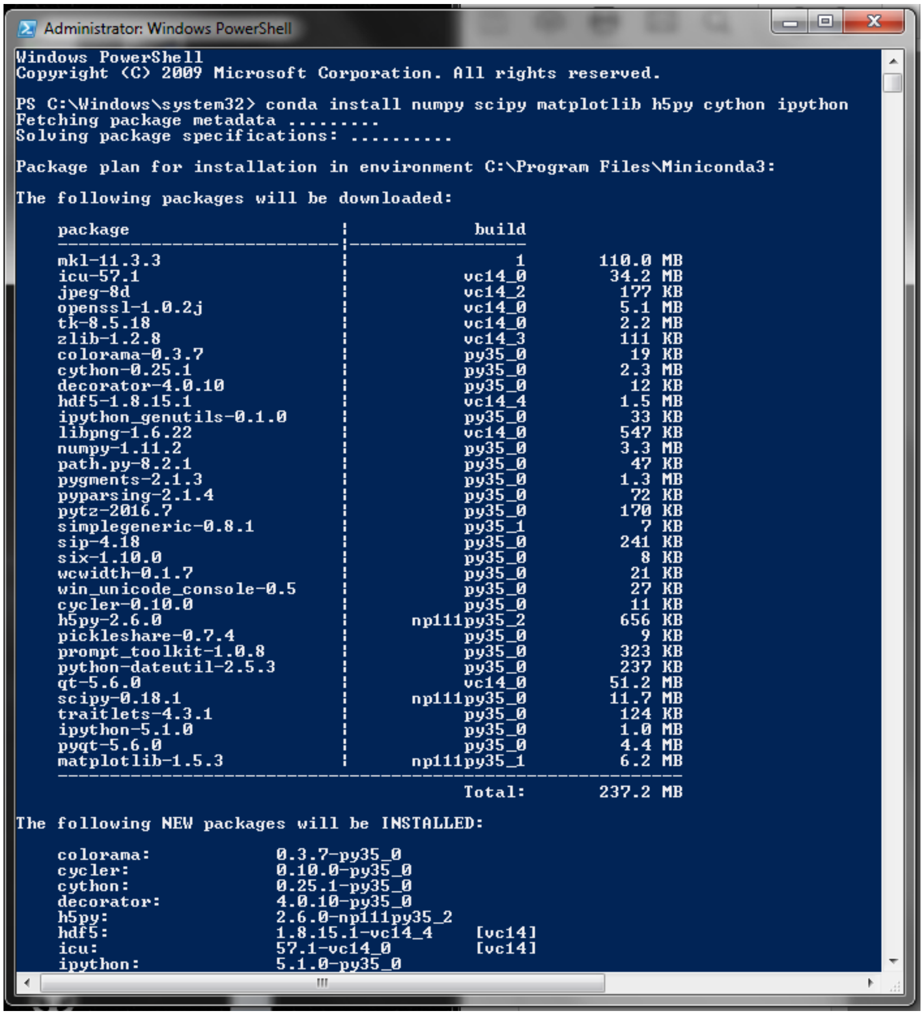


powershell

Figure 1: Opening Windows PowerShell from the Start Menu.

3 Install Python Packages

In Windows PowerShell or the Command Prompt, type the following: “conda install numpy scipy matplotlib cython h5py ipython” (minus the quotes). You should see something similar to Fig. 2. You will have to type “y” to confirm the installation. It may take some time to download and install the packages. You may have to hit the Enter key periodically to keep the installation progressing.



```
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> conda install numpy scipy matplotlib h5py cython ipython
Fetching package metadata .........
Solving package specifications: ..........

Package plan for installation in environment C:\Program Files\Miniconda3:

The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    mkl-11.3.3                 |                1       110.0 MB
    icu-57.1                   |           vc14_0        34.2 MB
    jpeg-8d                    |           vc14_2         177 KB
    openssl-1.0.2j             |           vc14_0         5.1 MB
    tk-8.5.18                  |           vc14_0         2.2 MB
    zlib-1.2.8                 |           vc14_3         111 KB
    colorama-0.3.7             |           py35_0          19 KB
    cython-0.25.1              |           py35_0         2.3 MB
    decorator-4.0.10           |           py35_0          12 KB
    hdf5-1.8.15.1              |           vc14_4         1.5 MB
    ipython_genutils-0.1.0     |           py35_0          33 KB
    libpng-1.6.22              |           vc14_0         547 KB
    numpy-1.11.2               |           py35_0         3.3 MB
    path.py-8.2.1              |           py35_0          47 KB
    pygments-2.1.3             |           py35_0         1.3 MB
    pyparsing-2.1.4            |           py35_0          72 KB
    pytz-2016.7                |           py35_0         170 KB
    simplegeneric-0.8.1        |           py35_1           7 KB
    sip-4.18                   |           py35_0         241 KB
    six-1.10.0                 |           py35_0           8 KB
    wcwidth-0.1.7              |           py35_0          21 KB
    win_unicode_console-0.5    |           py35_0          27 KB
    cycler-0.10.0              |           py35_0          11 KB
    h5py-2.6.0                 |      np111py35_2         656 KB
    pickleshare-0.7.4          |           py35_0           9 KB
    prompt_toolkit-1.0.8       |           py35_0         323 KB
    python-dateutil-2.5.3      |           py35_0         237 KB
    qt-5.6.0                   |           vc14_0        51.2 MB
    scipy-0.18.1               |      np111py35_0        11.7 MB
    traitlets-4.3.1            |           py35_0         124 KB
    ipython-5.1.0              |           py35_0         1.0 MB
    pyqt-5.6.0                 |           py35_0         4.4 MB
    matplotlib-1.5.3           |      np111py35_1         6.2 MB
    ------------------------------------------------------------
                                           Total:       237.2 MB

The following NEW packages will be INSTALLED:

    colorama:            0.3.7-py35_0
    cycler:              0.10.0-py35_0
    cython:              0.25.1-py35_0
    decorator:           4.0.10-py35_0
    h5py:                2.6.0-np111py35_2
    hdf5:                1.8.15.1-vc14_4     [vc14]
    icu:                 57.1-vc14_0         [vc14]
    ipython:             5.1.0-py35_0
```

Figure 2: Installing Python packages using conda.

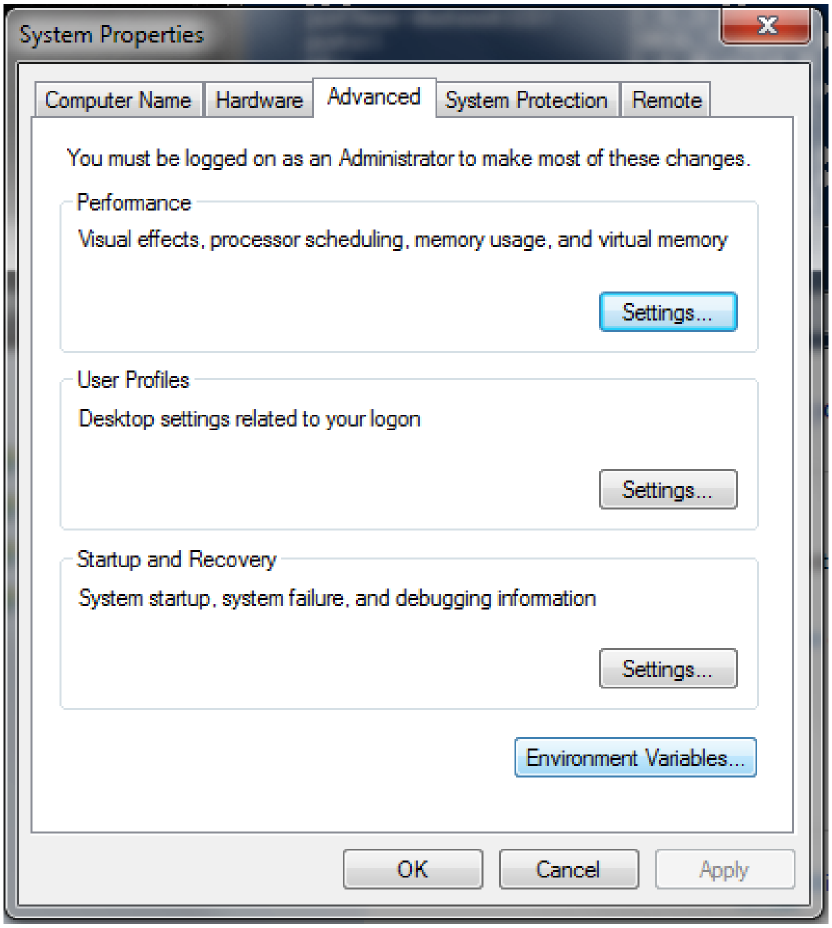
4 Install GDAL

The next step is to install the Geospatial Data Abstraction Library (GDAL). This software is used by Kapok to output geocoded rasters of the PolInSAR-derived parameters. Windows binaries of GDAL can be downloaded as part of the open source OSGeo4W package, available here: https://trac.osgeo.org/osgeo4w/.

When installing, it is recommended to choose the option for "Express Install". During the installation, the software will ask which packages you wish to install. Make sure that the checkbox next to GDAL is checked. Take note of the installation location (the default is "C:\OSGeo4W64\", for the 64-bit version), as we will need to add it to the system PATH in order to use GDAL from the command line.

5 Add GDAL to the System Path

To edit environment variables in Windows, right click on My Computer and go to Properties. Go to Advanced System Properties. You should see a Window similar to that shown in Fig. 3. Click the Environment Variables button in the bottom right corner.



You must be logged on as an Administrator to make most of these changes.

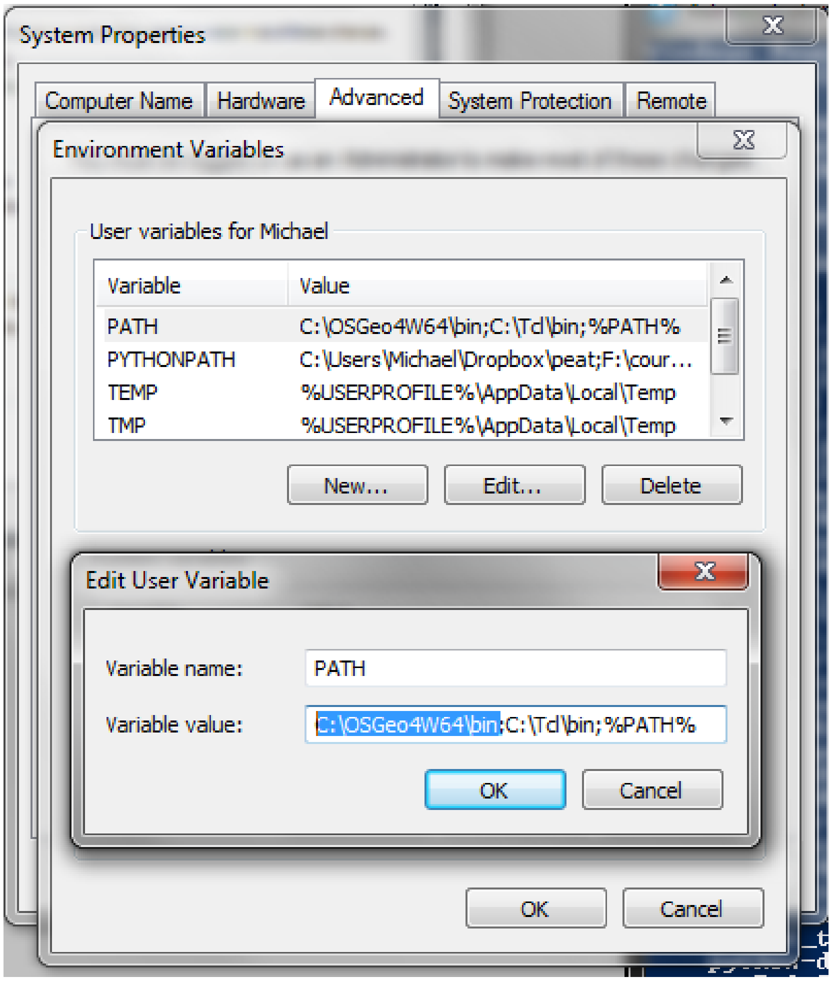
Visual effects, processor scheduling, memory usage, and virtual memory

Desktop settings related to your logon

System startup, system failure, and debugging information

Figure 3: The Environment Variables can be accessed by clicking the button in Advanced System Properties.

Highlight the PATH environment variable, then click the Edit button. The GDAL binaries are located in the \bin \subfolder of the location where GDAL was installed. For the default installation location, this will be in “C:\OSGeo4W64\bin”. Add the path to the front of the PATH environment variable, separated from any other pre-existing paths by a semicolon, as shown in Fig. 4.



User variables for Michael

Variable	Value
PATH	C:\OSGeo4W64\bin;C:\Tcl\bin;%PATH%
PYTHONPATH	C:\Users\Michael\Dropbox\peat;F:\cour...
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

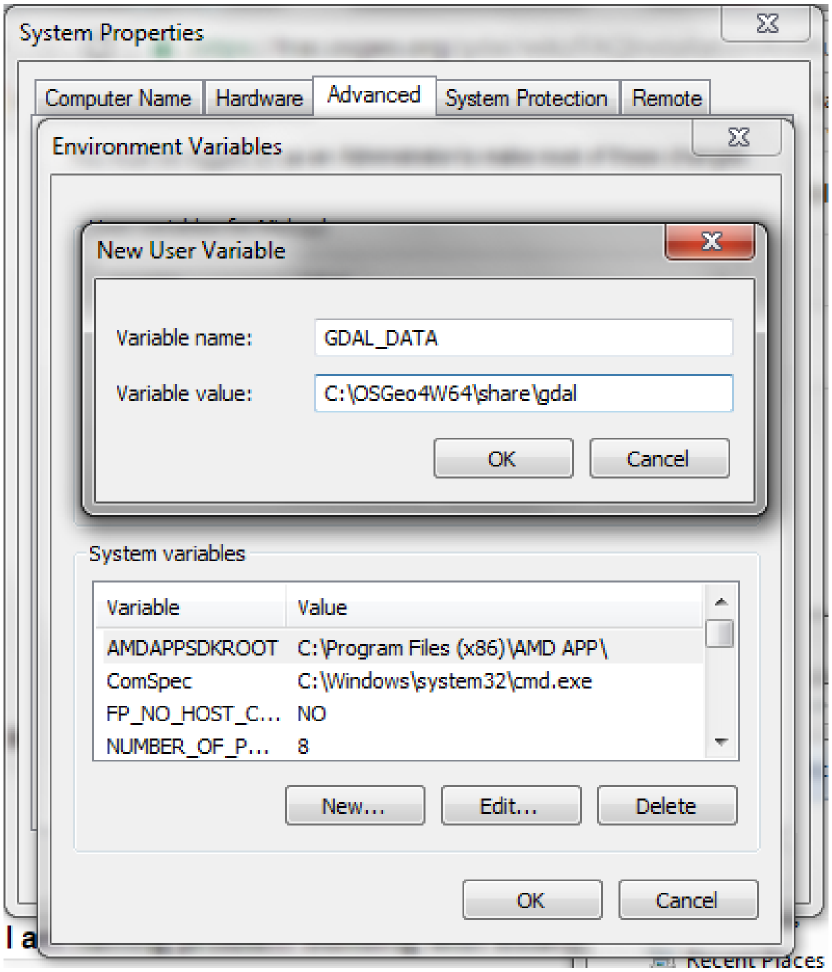
Variable name: PATH

Variable value: C:\OSGeo4W64\bin;C:\Tcl\bin;%PATH%

Figure 4: Adding GDAL to the system path. The newly added text is highlighted.

6 Set GDAL_DATA

We now need to add a new environment variable called “GDAL_DATA” which points to the \share\gdal\subfolder of the location where GDAL was installed. By default, this will be in “C:\OSGeo4W64\share\gdal”. Click the New button under the Environment Variables box, then fill in the variable name as “GDAL_DATA” and the variable value as the path, as shown in Fig. 5.



Variable name: GDAL_DATA

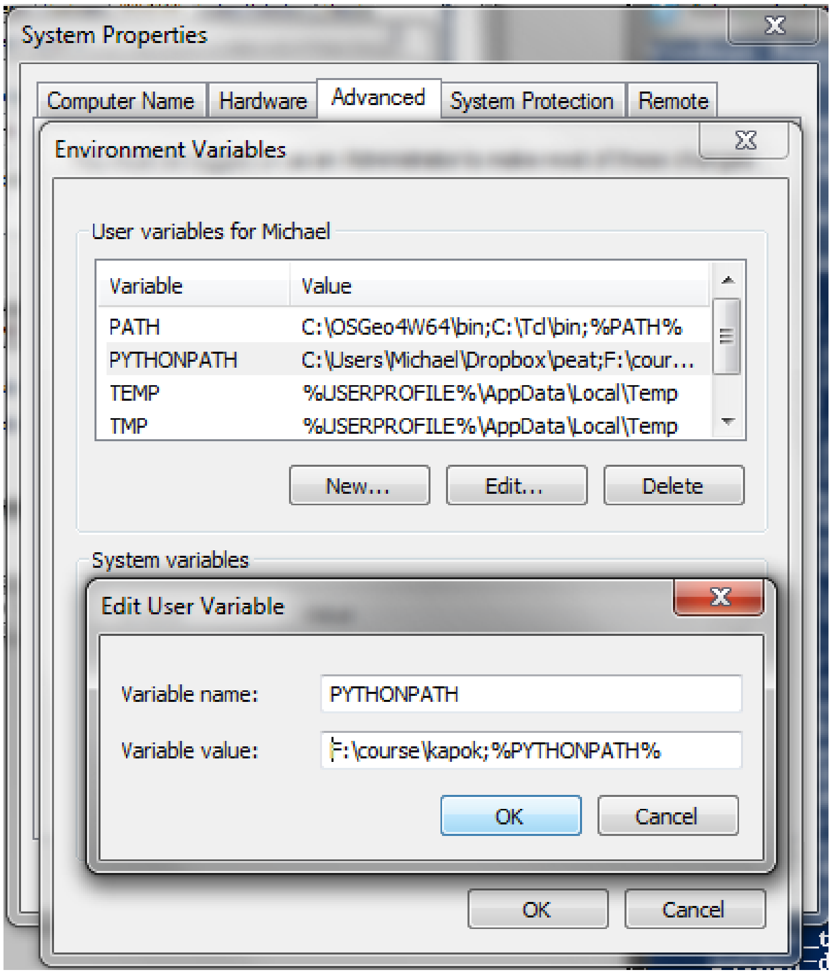
Variable value: C:\OSGeo4W64\share\gdal

Variable	Value
AMDAPPSDKROOT	C:\Program Files (x86)\AMD APP\
ComSpec	C:\Windows\system32\cmd.exe
FP_NO_HOST_C...	NO
NUMBER_OF_P...	8

Figure 5: Adding a new GDAL_DATA environment variable.

7 Add Kapok to the Python Path

We now add the path where Kapok is located to the PYTHONPATH environment variable, so that the Python console will know where the software is installed. The PYTHONPATH environment variable must point to the main \kapok\directory, not the kapok subfolder containing the actual Python code files (.py and .pyx). Note that I extracted the Kapok software to the folder “F:\course\kapok\”. Adding this folder to the PYTHONPATH is shown in Fig. 6.



User variables for Michael

Variable	Value
PATH	C:\OSGeo4W64\bin;C:\Tcl\bin;%PATH%
PYTHONPATH	C:\Users\Michael\Dropbox\peat;F:\cour...
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

Variable name: PYTHONPATH

Variable value: F:\course\kapok;%PYTHONPATH%

Figure 6: Adding Kapok to the Python path environment variable.

8 Install a C Compiler for Cython

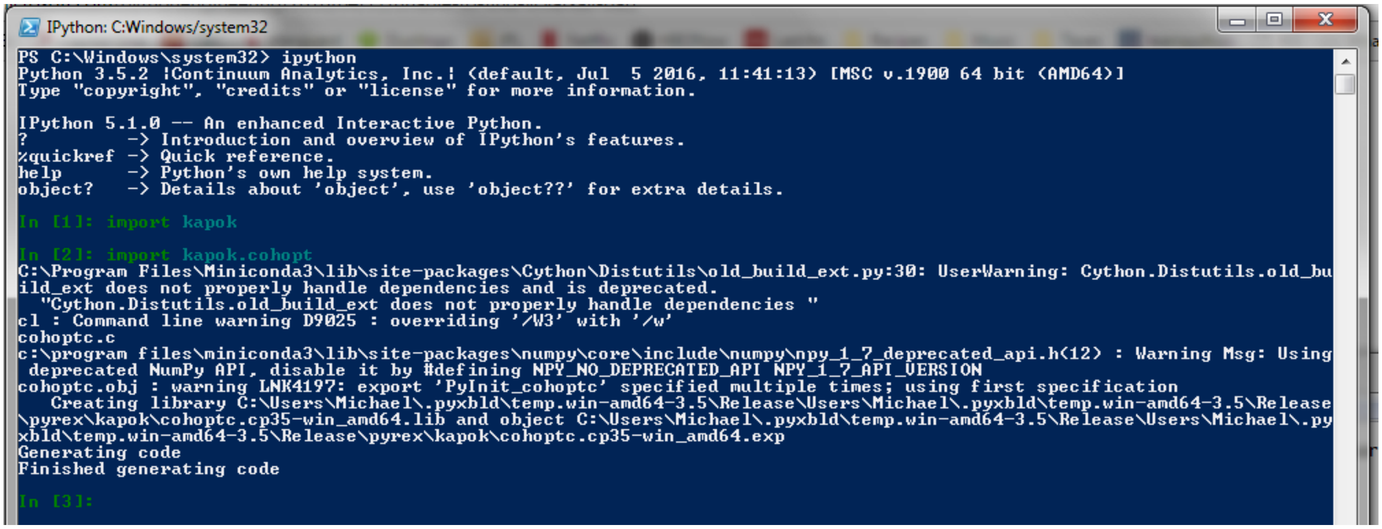
The Cython Python package which we installed in step 3 requires a C compiler to run. Assuming you are running the Python 3.5 version recommended in step 1, you can use the Visual C++ Build Tools, available for download from: http://go.microsoft.com/fwlink/?LinkId=691126.

9 Test the Installation

Open up Windows PowerShell or a Command Prompt again as in step 2. Enter "ipython" into the terminal. This should open the Interactive Python console.

Enter "import kapok" into the console in order to import the Kapok software.

Enter "import kapok.cohopt" in order to import the kapok.cohopt module, which will test that the Cython package is installed and configured correctly. If the Cython version of the module cannot be imported, a warning message will be displayed, and Kapok will import a native Python implementation instead. The functionality will be the same, but the code will run much slower in this case. Note that even if the software is working correctly, there may be some compiler warnings displayed to the screen, such as those shown in Fig. 7. As long as the console does not display an "ImportError", this is fine.



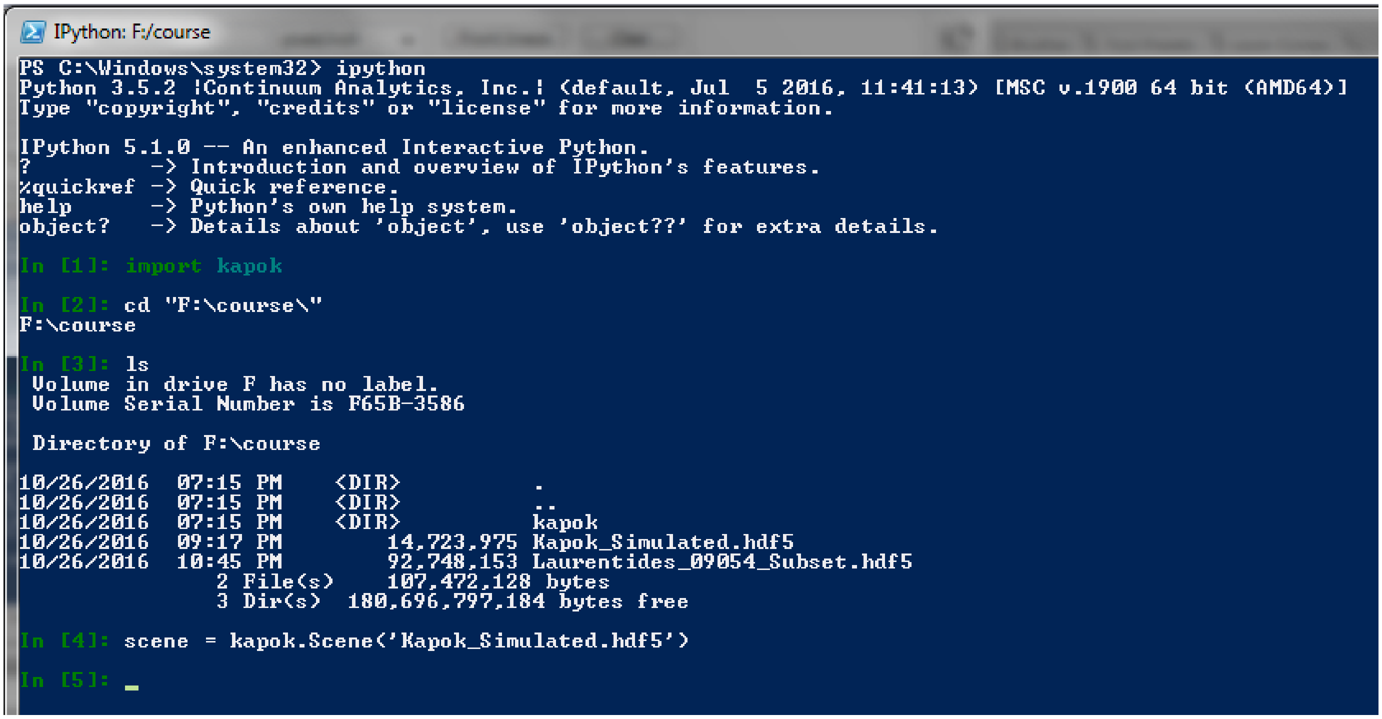
```
PS C:\Windows\system32> ipython
Python 3.5.2 |Continuum Analytics, Inc.| (default, Jul  5 2016, 11:41:13) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 5.1.0 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

In [1]: import kapok

In [2]: import kapok.cohopt
C:\Program Files\Miniconda3\lib\site-packages\Cython\Distutils\old_build_ext.py:30: UserWarning: Cython.Distutils.old_bu
ild_ext does not properly handle dependencies and is deprecated.
  "Cython.Distutils.old_build_ext does not properly handle dependencies "
cl : Command line warning D9025 : overriding '/W3' with '/w'
cohoptc.c
c:\program files\miniconda3\lib\site-packages\numpy\core\include\numpy\npy_1_7_deprecated_api.h(12) : Warning Msg: Using
 deprecated NumPy API, disable it by #defining NPY_NO_DEPRECATED_API NPY_1_7_API_VERSION
cohoptc.obj : warning LNK4197: export 'PyInit_cohoptc' specified multiple times; using first specification
   Creating library C:\Users\Michael\.pyxbld\temp.win-amd64-3.5\Release\Users\Michael\.pyxbld\temp.win-amd64-3.5\Release
\pyrex\kapok\cohoptc.cp35-win_amd64.lib and object C:\Users\Michael\.pyxbld\temp.win-amd64-3.5\Release\Users\Michael\.py
xbld\temp.win-amd64-3.5\Release\pyrex\kapok\cohoptc.cp35-win_amd64.exp
Generating code
Finished generating code

In [3]:
```

Figure 7: Importing Kapok and the kapok.cohopt module.

10 Load Sample Data

While in IPython, you can navigate between folders using the commands “cd” as in a normal command prompt. Navigate to the folder containing the sample data. One of the sample data files should be called “Kapok_Simulated.hdf5”. We are going to load this file to ensure that things are working correctly. Execute the command “import kapok” to import the software.

Then execute the command “scene = kapok.Scene(’Kapok_Simulated.hdf5’)”. This should load the Kapok simulated sample data. An example is shown in Fig. 8. In the figure, I import Kapok, navigate to the folder containing the sample data, then load the sample data into a Kapok Scene object.



```
PS C:\Windows\system32> ipython
Python 3.5.2 |Continuum Analytics, Inc.| (default, Jul  5 2016, 11:41:13) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 5.1.0 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

In [1]: import kapok

In [2]: cd "F:\course\"
F:\course

In [3]: ls
 Volume in drive F has no label.
 Volume Serial Number is F65B-3586

 Directory of F:\course

10/26/2016  07:15 PM    <DIR>          .
10/26/2016  07:15 PM    <DIR>          ..
10/26/2016  07:15 PM    <DIR>          kapok
10/26/2016  09:17 PM        14,723,975 Kapok_Simulated.hdf5
10/26/2016  10:45 PM        92,748,153 Laurentides_09054_Subset.hdf5
               2 File(s)    107,472,128 bytes
               3 Dir(s)  180,696,797,184 bytes free

In [4]: scene = kapok.Scene('Kapok_Simulated.hdf5')

In [5]: _
```

Figure 8: Loading sample data into Kapok.

To display a Pauli RGB image of the loaded data, we can use the command “scene.show()”. An example of the output is shown in Fig. 9.

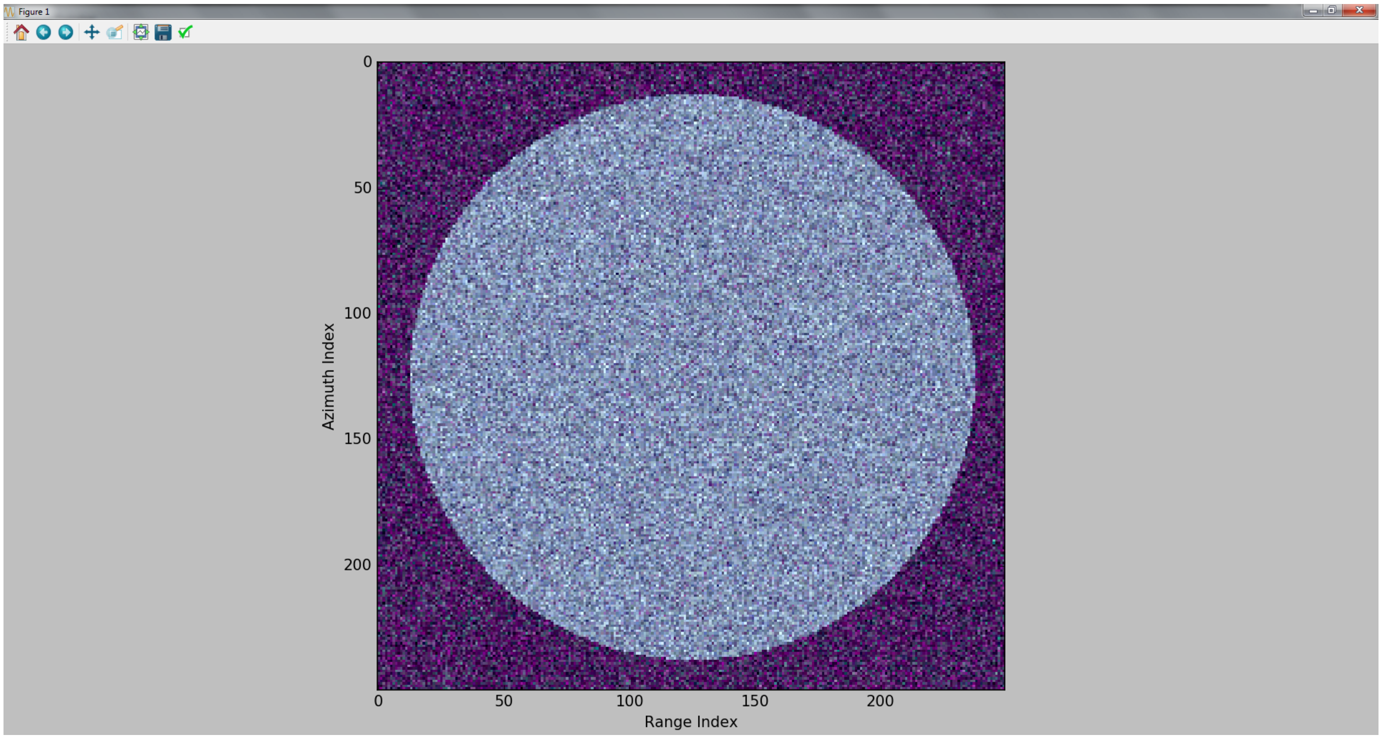


Figure 9: Displaying sample data.