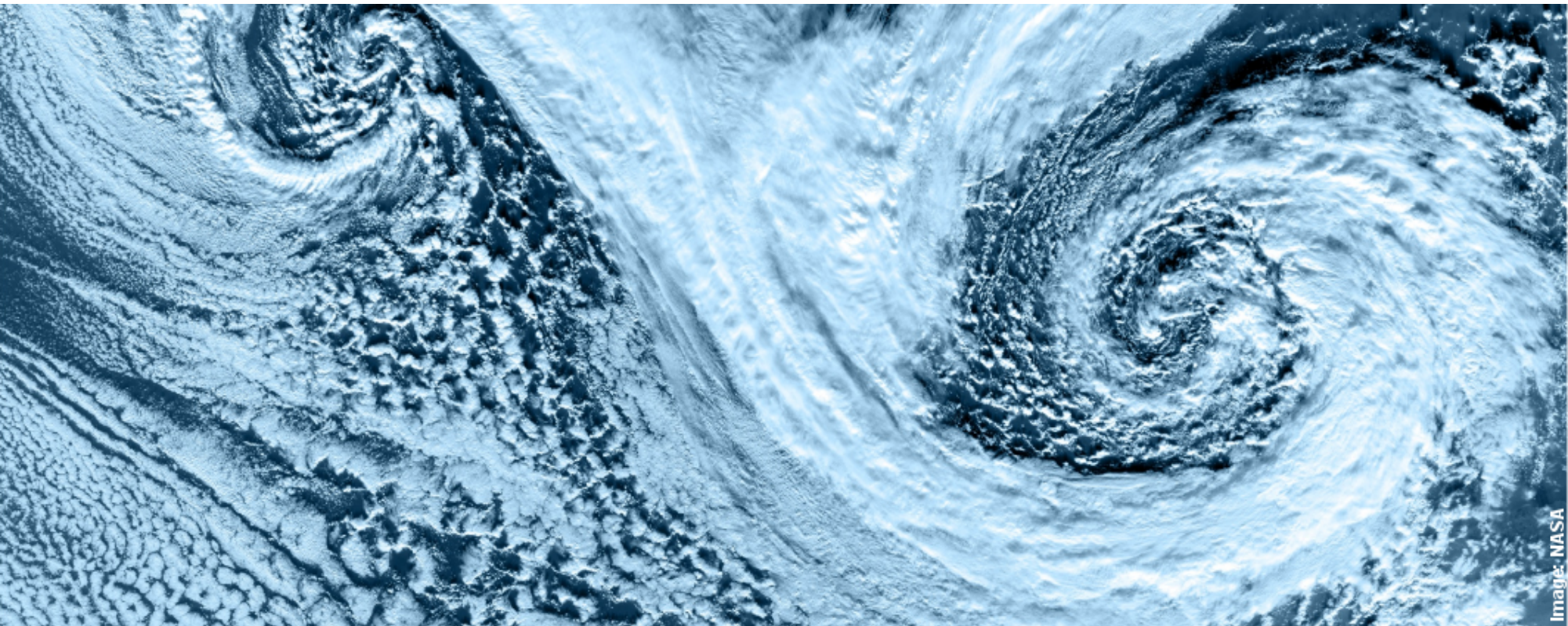
C2SM workshop
Scientific Programming in Python

Harald von Waldow (C2SM)
Bas Crezee (IAC/Atm.Dyn.)
Nicolas Piaget (IAC/Atm.Dyn.)
Marina Dütsch (IAC/Atm.Dyn.)

2014–09–11



Other language features

try - except

- catch exceptions
- handle exceptions
- throw your own exceptions

Other language features

try - except

- catch exceptions
- handle exceptions
- throw your own exceptions

Other language features

try - except

- catch exceptions
- handle exceptions
- throw your own exceptions

Standard Library Modules

random

- generate random numbers
- various distributions
- sampling

os.path

- OS - independent pathname manipulation
- prefer to constructing paths with string methods
- many convenience methods, e.g.
  `os.walk`: traverses directory-tree
- file information, e.g.
  `os.path.exists`,
  `os.path.mtime` ...

Standard Library Modules

random

- generate random numbers
- various distributions
- sampling

os.path

- OS - independent pathname manipulation
- prefer to constructing paths with string methods
- many convenience methods, e.g.
  `os.walk`: traverses directory-tree
- file information, e.g.
  `os.path.exists`,
  `os.path.mtime` ...

Standard Library Modules

random

- generate random numbers
- various distributions
- sampling

os.path

- OS - independent pathname manipulation
- prefer to constructing paths with string methods
- many convenience methods, e.g.
  `os.walk`: traverses directory-tree
- file information, e.g.
  `os.path.exists`,
  `os.path.mtime` ...

Standard Library Modules

random

- generate random numbers
- various distributions
- sampling

os.path

- OS - independent pathname manipulation
- prefer to constructing paths with string methods
- many convenience methods, e.g.
  `os.walk`: traverses directory-tree
- file information, e.g.
  `os.path.exists`,
  `os.path.mtime` ...

Standard Library Modules

random

- generate random numbers
- various distributions
- sampling

os.path

- OS - independent pathname manipulation
- prefer to constructing paths with string methods
- many convenience methods, e.g.
  `os.walk`: traverses directory-tree
- file information, e.g.
  `os.path.exists`,
  `os.path.mtime` ...

Standard Library Modules

random

- generate random numbers
- various distributions
- sampling

os.path

- OS - independent pathname manipulation
- prefer to constructing paths with string methods
- many convenience methods, e.g.
  `os.walk`: traverses directory-tree
- file information, e.g.
  `os.path.exists`,
  `os.path.mtime` ...

Standard Library Modules

random

- generate random numbers
- various distributions
- sampling

os.path

- OS - independent pathname manipulation
- prefer to constructing paths with string methods
- many convenience methods, e.g.
  `os.walk`: traverses directory-tree
- file information, e.g.
  `os.path.exists`,
  `os.path.mtime` ...

Standard Library Modules

glob

- glob.glob: finds pathnames matching Unix shell patterns

os

operating system functionality

- `os.environ`: environment variables
- `os.chdir, os.getcwd, ...`
- `os.getpid, os.getuid, ...`

sys

- `sys.exit`: terminates program
- `sys.argv`: gets command line arguments for python script

Standard Library Modules

glob

- glob.glob: finds pathnames matching Unix shell patterns

os

operating system functionality

- `os.environ`: environment variables
- `os.chdir`, `os.getcwd`, ...
- `os.getpid`, `os.getuid`, ...

sys

- `sys.exit`: terminates program
- `sys.argv`: gets command line arguments for python script

Standard Library Modules

glob

- glob.glob: finds pathnames matching Unix shell patterns

os

operating system functionality

- `os.environ`: environment variables
- `os.chdir, os.getcwd, ...`
- `os.getpid, os.getuid, ...`

sys

- `sys.exit`: terminates program
- `sys.argv`: gets command line arguments for python script

Standard Library Modules

glob

- glob.glob: finds pathnames matching Unix shell patterns

os

operating system functionality

- `os.environ`: environment variables
- `os.chdir, os.getcwd, ...`
- `os.getpid, os.getuid, ...`

sys

- `sys.exit`: terminates program
- `sys.argv`: gets command line arguments for python script

Standard Library Modules

glob

- glob.glob: finds pathnames matching Unix shell patterns

os

operating system functionality

- `os.environ`: environment variables
- `os.chdir, os.getcwd, ...`
- `os.getpid, os.getuid, ...`

sys

- `sys.exit`: terminates program
- `sys.argv`: gets command line arguments for python script

Standard Library Modules

glob

- glob.glob: finds pathnames matching Unix shell patterns

os

operating system functionality

- `os.environ`: environment variables
- `os.chdir, os.getcwd, ...`
- `os.getpid, os.getuid, ...`

sys

- `sys.exit`: terminates program
- `sys.argv`: gets command line arguments for python script

Standard Library Modules

subprocess

spawn new processes, connect to their input/output/error pipes, and obtain their return codes.

- `subprocess.call`: launch subprocess
- `subprocess.Popen`: lower level process handling

multiprocessing

- good to use multiple cores
- interprocess communication (`Queues` and `Pipes`
- use a "pool of workers"

unittest

- supports "test-driven" programming
- very useful for larger projects

Standard Library Modules

subprocess

spawn new processes, connect to their input/output/error pipes, and obtain their return codes.

- `subprocess.call`: launch subprocess
- `subprocess.Popen`: lower level process handling

multiprocessing

- good to use multiple cores
- interprocess communication (`Queues` and `Pipes`
- use a "pool of workers"

unittest

- supports "test-driven" programming
- very useful for larger projects

Standard Library Modules

subprocess

spawn new processes, connect to their input/output/error pipes, and obtain their return codes.

- `subprocess.call`: launch subprocess
- `subprocess.Popen`: lower level process handling

multiprocessing

- good to use multiple cores
- interprocess communication (`Queues` and `Pipes`
- use a "pool of workers"

unittest

- supports "test-driven" programming
- very useful for larger projects

Standard Library Modules

subprocess

spawn new processes, connect to their input/output/error pipes, and obtain their return codes.

- `subprocess.call`: launch subprocess
- `subprocess.Popen`: lower level process handling

multiprocessing

- good to use multiple cores
- interprocess communication (`Queues` and `Pipes`
- use a "pool of workers"

unittest

- supports "test-driven" programming
- very useful for larger projects

Standard Library Modules

subprocess

spawn new processes, connect to their input/output/error pipes, and obtain their return codes.

- `subprocess.call`: launch subprocess
- `subprocess.Popen`: lower level process handling

multiprocessing

- good to use multiple cores
- interprocess communication (`Queues` and `Pipes`
- use a "pool of workers"

unittest

- supports "test-driven" programming
- very useful for larger projects

Standard Library Modules

subprocess

spawn new processes, connect to their input/output/error pipes, and obtain their return codes.

- `subprocess.call`: launch subprocess
- `subprocess.Popen`: lower level process handling

multiprocessing

- good to use multiple cores
- interprocess communication (`Queues` and `Pipes`
- use a "pool of workers"

unittest

- supports "test-driven" programming
- very useful for larger projects

Standard Library Modules

subprocess

spawn new processes, connect to their input/output/error pipes, and obtain their return codes.

- `subprocess.call`: launch subprocess
- `subprocess.Popen`: lower level process handling

multiprocessing

- good to use multiple cores
- interprocess communication (`Queues` and `Pipes`
- use a "pool of workers"

unittest

- supports "test-driven" programming
- very useful for larger projects

Other modules

pandas

- data-structures for data analysis
- a "Dataframe", like in R, but better.
- a timeseries-object
- huge number of utility functions
- integrated with matplotlib and statsmodels

Statsmodels

- statistical models, e.g.
  GLMs
  robust linear models
  nonparametric estimators
  time-series analysis
  many tests
- very fast developing

Other modules

pandas

- data-structures for data analysis
- a "Dataframe", like in R, but better.
- a timeseries-object
- huge number of utility functions
- integrated with matplotlib and statsmodels

Statsmodels

- statistical models, e.g.
  GLMs
  robust linear models
  nonparametric estimators
  time-series analysis
  many tests
- very fast developing

Other modules

pandas

- data-structures for data analysis
- a “Dataframe”, like in R, but better.
- a timeseries-object
- huge number of utility functions
- integrated with matplotlib and statsmodels

Statsmodels

- statistical models, e.g.
  GLMs
  robust linear models
  nonparametric estimators
  time-series analysis
  many tests
- very fast developing

Other modules

pandas

- data-structures for data analysis
- a "Dataframe", like in R, but better.
- a timeseries-object
- huge number of utility functions
- integrated with matplotlib and statsmodels

Statsmodels

- statistical models, e.g.
  GLMs
  robust linear models
  nonparametric estimators
  time-series analysis
  many tests
- very fast developing

Other modules

pandas

- data-structures for data analysis
- a “Dataframe”, like in R, but better.
- a timeseries-object
- huge number of utility functions
- integrated with matplotlib and statsmodels

Statsmodels

- statistical models, e.g.
  GLMs
  robust linear models
  nonparametric estimators
  time-series analysis
  many tests
- very fast developing

Other modules

pandas

- data-structures for data analysis
- a “Dataframe”, like in R, but better.
- a timeseries-object
- huge number of utility functions
- integrated with matplotlib and statsmodels

Statsmodels

- statistical models, e.g.
  GLMs
  robust linear models
  nonparametric estimators
  time-series analysis
  many tests
- very fast developing

Other modules

pandas

- data-structures for data analysis
- a “Dataframe”, like in R, but better.
- a timeseries-object
- huge number of utility functions
- integrated with matplotlib and statsmodels

Statsmodels

- statistical models, e.g.
  GLMs
  robust linear models
  nonparametric estimators
  time-series analysis
  many tests
- very fast developing

Other modules

Sympy

Symbolic math. Like Matlab's symbolic toolbox.

scikit-learn

- machine learning
- "big-data" analysis
- data-mining

Other modules

Sympy

Symbolic math. Like Matlab's symbolic toolbox.

scikit-learn

- machine learning
- "big-data" analysis
- data-mining

Other modules

Sympy

Symbolic math. Like Matlab's symbolic toolbox.

scikit-learn

- machine learning
- "big-data" analysis
- data-mining