Going from Matlab to Python
A generic workflow

October 9th, 2017 | Sandra Diaz
Where to start

• Identify the type of computation you are performing
• Identify computational steps in your code
• Do basic profiling on your original Matlab code
• Identify input / output
• If you need parallel, think parallel from the beginning
What to do next

• Take one step at the time and translate the functionality
• Build a unit test which allows you to compare to known results => your Matlab results
• Document your code as you write it
• Commit frequently
• Check performance
• Try to adhere to programming standards (pep8)
How to implement your steps

• Each computation step has an input and an output

• Identify the interfaces among steps
How to implement your steps

• Identify the correct data structures in your computation
• Python provides a larger diversity of data structures (lists, dictionaries, arrays, matrices, etc...)
How to implement your steps

• Remember Python has a large community of users. Look for modules which can make your computation easier.

• If you use Matlab commands from a specific toolbox, look for equivalent Python modules.
How to implement your steps

• Break down computation into functions, classes or even modules.
• Always think about clean / reusable / maintainable code
Debugging

• PDB
• Print and test