Report for Summer Job at PetroStreamz

Rafid Hoda

2012-08-07

Summary

My main task at PetroStreamz was to develop a prototype of a differentiation utility for the Optimizer. We agreed that the best option would be to use open source libraries to handle the differentiation and that Python would be a sensible language to use because of it's simplicity. The utility was to work in such a way that the user could choose a txt or ppo file as input with functions and variables listed. The utility would differentiate all the functions with respect to all the variables and output the result to a txt or ppo file. This is especially useful for use with the Optimizer because it automates the differentiation process saving the user a lot of time and is much less error-prone.

SymPy, a Python based library seemed to be the best choice. CasADI was also considered and tested, but did not work as smoothly as SymPy. After simpler prototypes, the final one was coded and tested with typical equations.

WEEK 1

The first week I used most of my time getting to know Pipe-It and the Optimizer. I played with the Optimizer and used to it solve simple optimization problems in math. I was then given other example files for further testing. Based on my testing I evaluated the Optimizer's use for math and science students.

At the end of the week we got an idea of how the problem could be solved. I also used much of the week researching different symbolic differentiation libraries and picking out worthy candidates. The conclusion after the first week was that SymPy(Python based) and CasADI(Python/C++ based) could be the best options. Sage, Scilab and Octave were also considered. I also read about licensing for the different options. SymPy is licensed under the modified BSD license. This basically means it can be used for commercial purposes either in it's original state or modified, but a copyright note must be included with the product.

WEEK 2

Started seriously testing SymPy to see if it could work effectively with Pipe-It's Optimizer. I was given some example files to work with and tested them with SymPy. The results were positive and SymPy seemed to be the best choice. I coded a prototype in Python that would consume a txt file and output another txt file with derivatives.

See separate file for SymPy testing*

WEEK 3

Continued work and testing on the prototype. Tested with common equations in Petroleum. Got started on testing CasADI. Conversed with the developer of CasADI to see if it could be a good option. Although CasADI seemed to work for most cases, it had awkward notation and did not work as smoothly as SymPy. Had problems making a txt prototype for CasADI and ended up concluding that it was not a good choice.

WEEK 4, 5 and 6

The final weeks I coded the final utility. A lot of time went away to test with different types of files and filetypes. The three main projects were to handle txt to txt, txt to ppo, and ppo to ppo. I coded these separately and finally combined them into one. Some research and testing had to be done in order to compile the final product. PyInstaller was used for compiling.