

Journal and End of Quarter Report - Spring 2011

Vusi M. Magagula

in collaboration with

Dr. Martin J. Mohlenkamp

Department of Mathematics



SPRING 2011

Journal - Spring 2011

28 March - 1 April 2011:

I read a tutorial document on Python. I was interested in the syntax of Python as I have a programming background. Basically, I looked at the following:

- Writing Hello World using Python
- Control Statements, that is if, for e.t.c
- Little bit of Data Structures
- Lists

- Informal Introduction

My colleagues Son and Ranjita showed me how to compile and run Python. This site (<http://docs.python.org/tutorial/>) was also helpful.

4 April - 8 April 2011:

I wrote a program with a function named `check_FIVlist_vs_case` which takes `base`, `index_connections` and `FIVlist` as input and returns `True` or `False`. It checks if a `Function_IV` list is equivalent to a `geminal/index` case.

- Parameter `Base` : gives the geminal connections as produced by `restricted_canonical_base` and the type of `base` is a tuples of integers

- `Index_connections:index` coupling, as produced by `base_index_connections` and the type `index_connections` is tuples of tuples of integers and scalar

It calls a function called `FIVlist_to_case` which takes `FIVlist` as input and returns `base` and `index_connections`. This function converts a list of `Function_IV` to a geminal/index case. It also calls a function called `case_to_FIVlist` which takes `base` and `index_connections` as inputs and returns list of `Function_IV`. This function converts a geminal/index case to a list of `Function_IV`. However, when I tried to run the program it didn't compile just because of some syntax error.

11 April - 15 April 2011:

I wrote a program with functions named `sum_FIVlist` which takes `FIVlist`, `index` and `op_cache` as inputs and returns `FIVlist`. The main

aim of this function is to sum over a given index and it should return a newlist without that index. I also wrote a function named `integrate_FIVlist` which takes `FIVlist`, `variable` and `op_cache` as inputs and returns a new `FIVlist` which does not contain the variable we were integrating with respect to. However, my program compiled very well without producing the desired results. Dr Mohlenkamp helped me with the `integrate_FIVlist` function and my next duty was to implement the `sum_FIVlist` function.

18 April - 22 April 2011:

I modified the `integrate_FIVlist` function and came up with the `sum_FIVlist` function. It compiled well but I had difficulty trying to understand the input and hence I had to see Dr Mohlenkamp to clarify the meaning of the input he used to test the `integrate_FIVlist` function. I met him and he explained it to me.

25 April - 29 April 2011:

I was supposed to write a class called Operation_Cache which record operations already performed for re-use. It is going to be dictionary with a key such that the key can be identified by an operation, function, relative index/variable, index overlaps and variable overlaps. I tried to spend sleepless nights thinking about how I could implement this but I couldn't even figure out how I could do it. I tried several codes but I kept deleting them because the codes were not working at all. I need more time to think about it and hopefully there will be a breakthrough.

2 May - 6 May 2011:

This week I was supposed to continue with my last week's task, that of writing a class called `Operation_Cache`. Since, I had trouble trying to figure out how to solve the problem, Dr Mohlenkamp came to my rescue and he developed the code and gave me a clue on how I could write the remaining code. It helped me a lot and I was then able to write a code which recorded operations already performed for re-use with respect to a variable. He gave me an example on how to code the same function but for indexes and I implemented one for variables. I had a minor bug and Dr Mohlenkamp again came to my rescue and I saw my mistake. I then tested it for different cases and it worked.

9 May - 13 May 2011:

This week I was supposed to test the function `Operation_Cache` for symmetrical cases and see if it works. Fortunately, it did work. The code was able to recognise that these were similar cases. I also tested

if it can integrate loops and it also worked for that case. I was also to modify the program so that when integrating two functions with the same variable, the resultant function should not involve the variable we integrated with respect to. I was also to implement two more functions which write the output in Latex format. I tried and Ranjita emailed me her code which was not that helpful but it was an eye opener on how to implement the functions. I'm still working on the functions.

16 May - 20 May 2011:

I was to write a code for the function `latex_eqn_Sum_Int`. This function takes `FIVlist`, `split_incentive`, `iname`, and `vname` as inputs and it returns a latex representation of the sum or integral function. `FIVlist` is a list of `Function_IV`, `split_incentive` is parameter that can be used to split the equation over multiple lines where 0 means do

not split and higher numbers mean shorter lines and it is of integer type. `iname` is a default name for an index and is of string type, while `vname` is the default name for a variable and is of string type. The pseudocode is as follows:

- extract the function from the list and append it
- name the function to be written in latex representation
- extract the index from from the `Function_IV` list and compare it to the `iname` and if they are the same, sum over the index.
- extract the variable from from the `Function_IV` list and compare it to the `vname` and if they are the same, integrate over the variable.
- Break if the line is too long in between the variables.
- Return a latex file.

The code was implemented but however, the results were not the desired ones.

23 May - 27 May 2011:

This week, I worked on the code of the function `latex_string`. Dr Mohlenkamp suggested that I first implement the function `latex_string` and then if time permits, we work on `latex_eqn_Sum_Int`. The function compiled well and produced the required results.

30 May - 3 June 2011:

This was the last week of the quarter. We were to submit our End of Quarter Report and finalise any pending work. The End of Quarter report is in detail in the next page.

End of Quarter Report - Spring 2011

Spring 2011 Tasks

This quarter, I was supposed to implement the following functions:

- Operation_Cache
- latex_string
- Function_IV
- latex_eqn_Sum_Int
- check_FIVlist_vs_case
- sum_FIVlist
- integrate_FIVlist

Classes / Functions Analysis

- `Function_IV` - This function holds a function-type object with indexes and variables.
- `integrate_FIVlist` - This function integrates the multiplicative terms over a given variable. It returns a new list of `Function_IV`. The list consists of the input that did not include the variable and those that did include it, they are replaced by a single `Function_IV` representing the result.
- `sum_FIVlist` - This function sums the multiplicative terms over a given index. It returns a new list of `Function_IV`. The list consists of the input that did not include the index and those that did include it, they are replaced by a single `Function_IV` representing the result.
- `check_FIVlist_vs_case` - This function checks if a `Function_IV` list is equivalent to a geminal or index case. It returns true or false.

- `Operation_Cache` - this function records operations already performed for re-use.
- `latex_string` - This function returns a LaTeX representation of the given input.
- `latex_eqn_Sum_Int` - This function returns a LaTeX representation of a sum or integral of `Function_IV`.

The above functions were successfully implemented with the extra help by Dr M.J. Mohlenkamp except the `latex_eqn_Sum_Int` function.

Summary

For a long time, I thought MATLAB was the best programming language in the Mathematics field. After being part of this research group this Spring 2011, I changed my mind. My next goal after the research, is to learn all the available mathematics functions in Python. I've learned that Python is not only a good programming language but

it is also available for free as opposed to the expensive MATLAB for individual use. I'm grateful for being part of the programming section of the research group. I've seen implementation of the theories I've learnt in Software Engineering. At this point, I would gladly say I've learnt how to organize a research group and how to involve students and how to motivate them to believe in themselves. It was a fruitful quarter as far as learning something new. I may have not implemented all the programs by myself but trying, failing and seeing the right thing I was suppose to do was the best learning strategy.