Research Primer

A Presentation

Introduction

- Every new member of the research group must work through the research primer upon joining.
- It is designed to help to acquire the skills needed to succeed in the group.

There are 9 tasks in the program:

- Make a time sheet
- Learn about HPC and get and FSL account
- Learn Unix and Choose a local machine
- Learn Bash Slurm and choose a text editor
- Learn Python
- Learn C++
- Learn Git and get a Bitbucket account
- **Solve a ODE using the Explicit Euler Method**
- Talk about it with Dr. Tree

Methods...

- Most of what I did throughout the training was reading things and watching videos.
- The ODE project itself was split into 6 steps that each had some overlap but were intended to get you to use all of the various things learned in the training.

Step 1

Create a directory called training/ in fsl

...boring...

Step 2

Write a C++ code that solves the ODE using the Explicit Euler method. This should be able to read in a parameters file and output to an output file.

- This took by far the longest to figure out how to do well.
- A lot of the work I had to do was learning how to use command line arguments and getting the pathing to work out.

Step 3

Create a Git repository to track the changes as you write the C++ code. Make commits and push to Bitbucket a lot.

This probably should have been step 2.

Step 4

After you are sure it works compile it and put it the compiled code into another directory.

Again not very exciting except for the fact that you can’t be sure it works until you compile it and run tests.

So a bit of a Catch-22 here.

Step 5

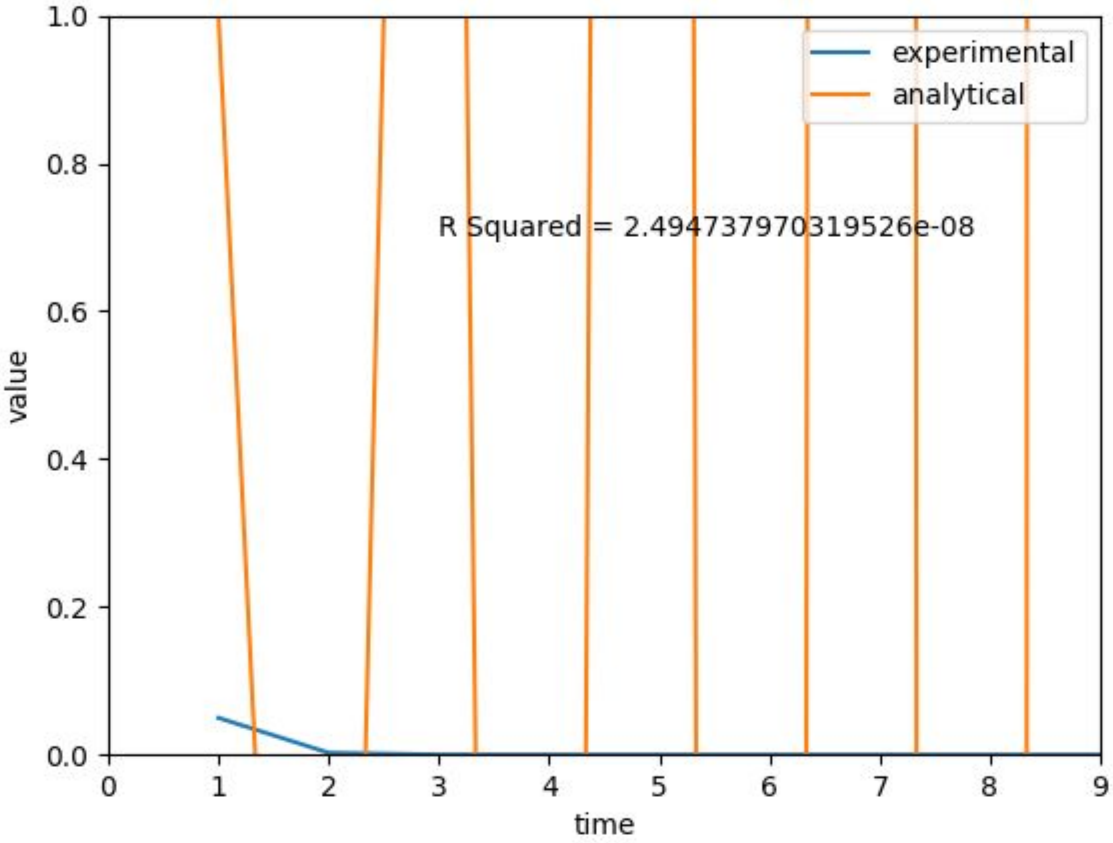
Write a Slurm script to run it for several different parameters.

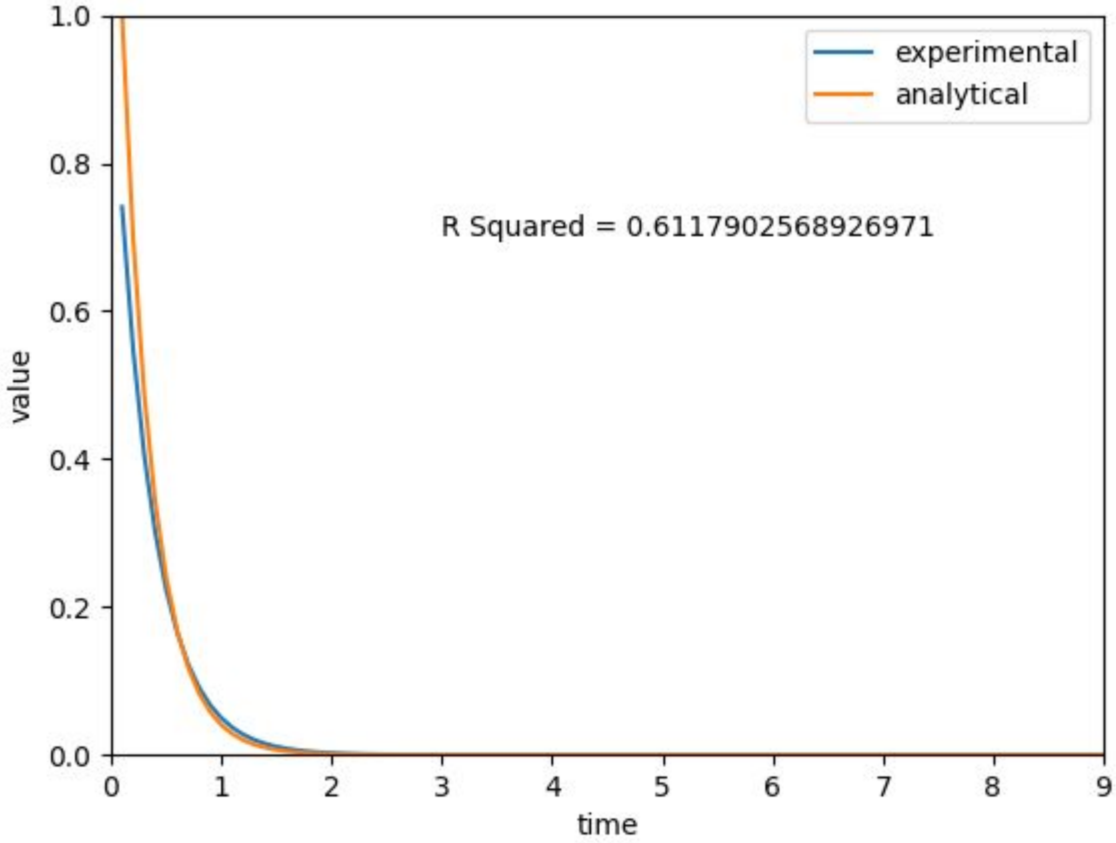
- This doesn’t seem that hard but at this point I had to actually run it several times to catch issues that I couldn’t catch when compiling. Runtime errors and such.
- Also because I wanted to be lazy I went and rewrote my C++ code at this point so I could do everything in as short of a slurm script as possible.

Step 6

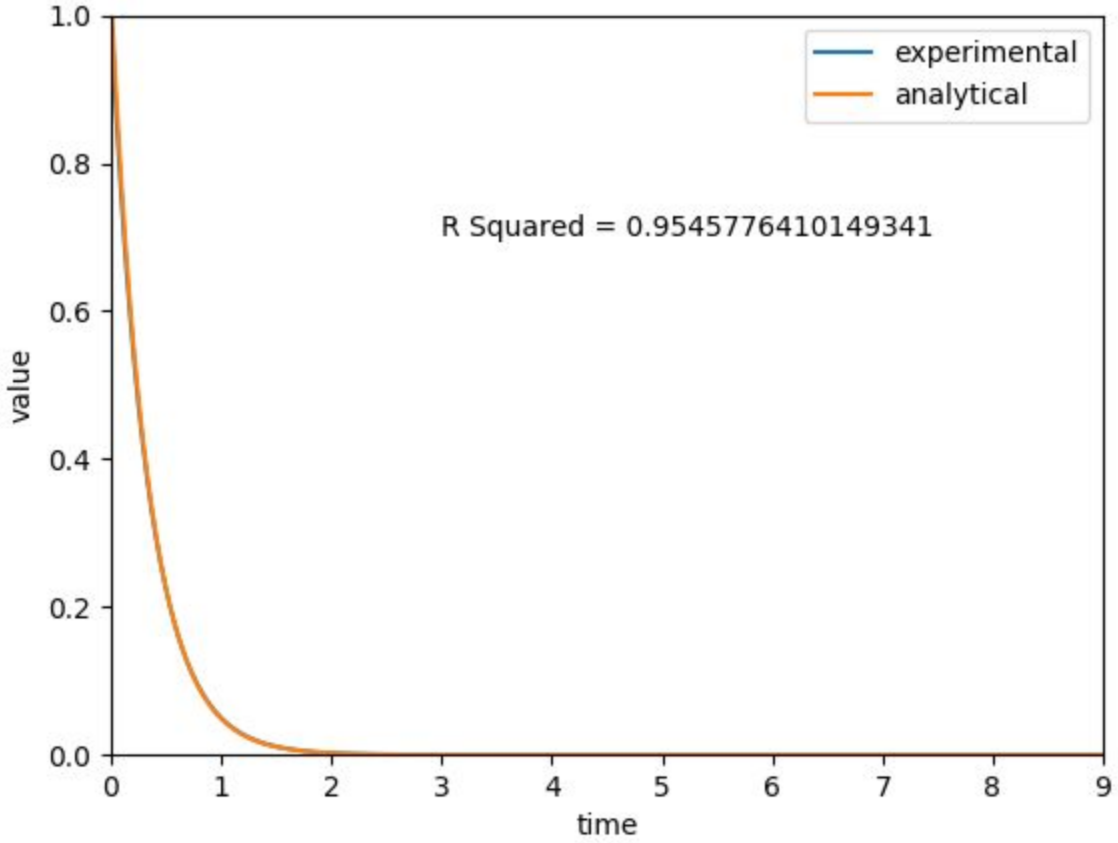
Write a python Script that takes the raw data and graphs it and gives some sort of error feedback.

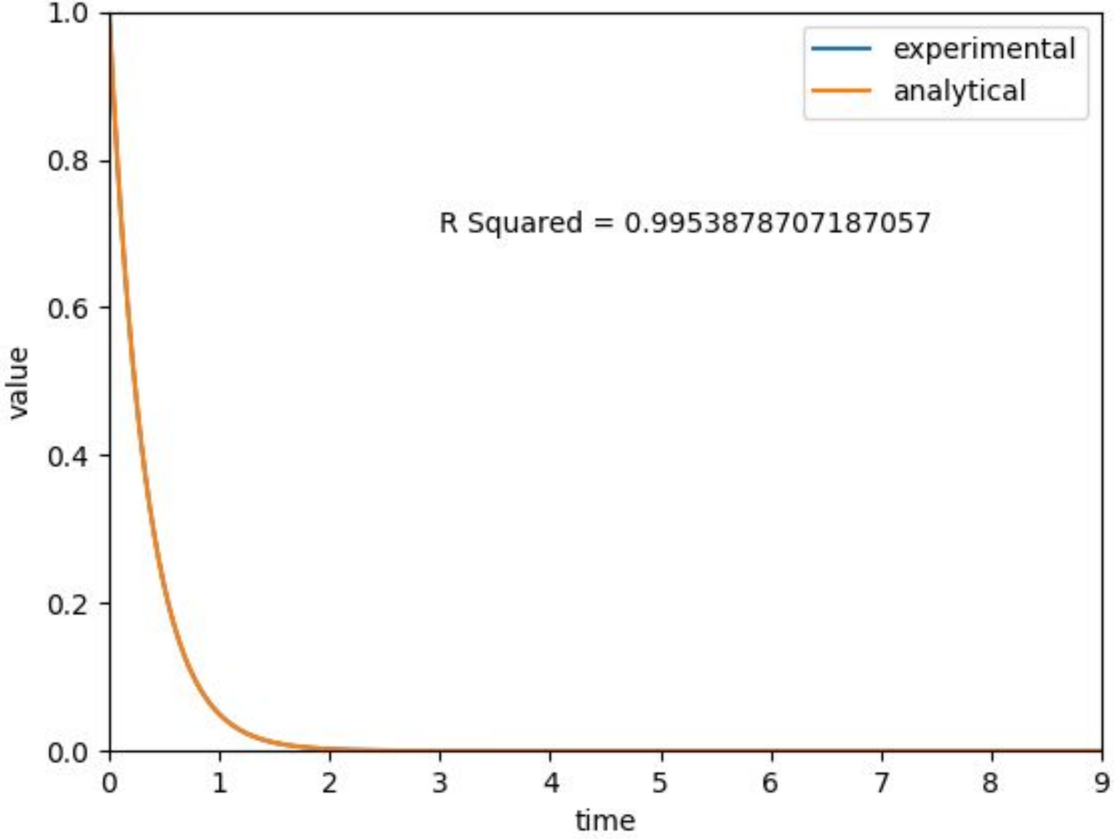
- This was pretty simple to create.
- The hardest and most annoying part was figuring out how to effectively communicate the error in the results.
  - I decided to go with using a coefficient of determination which sucked...

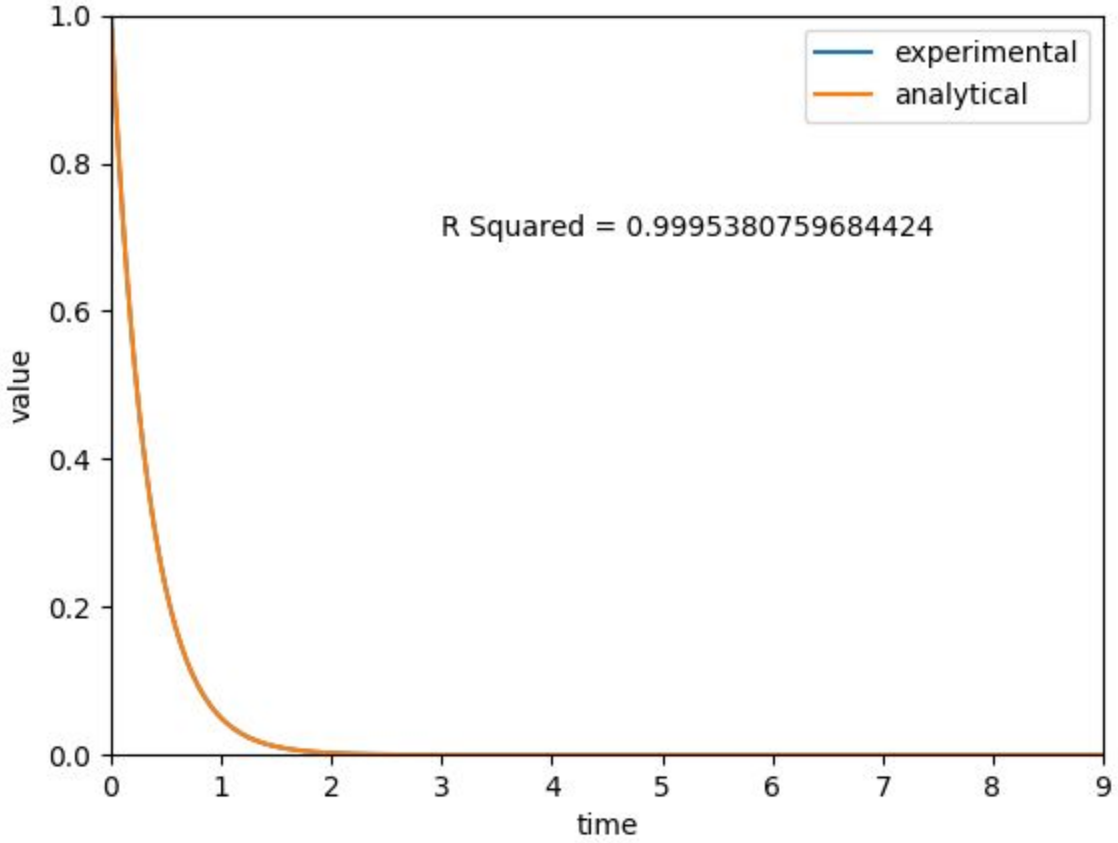

experimental
analytical
R Squared = 2.494737970319526e-08
value
1.0
0.8
0.6
0.4
0.2
0.0
0
1
2
3
4
5
6
7
8
9
time

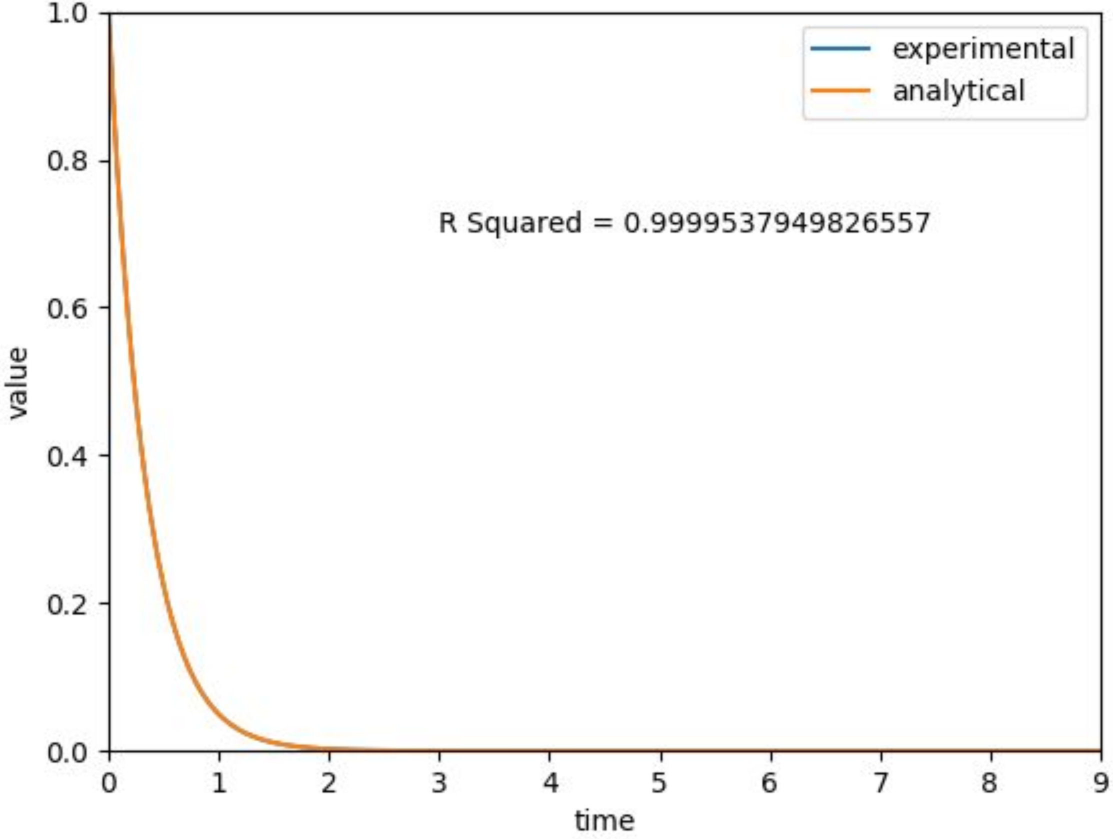


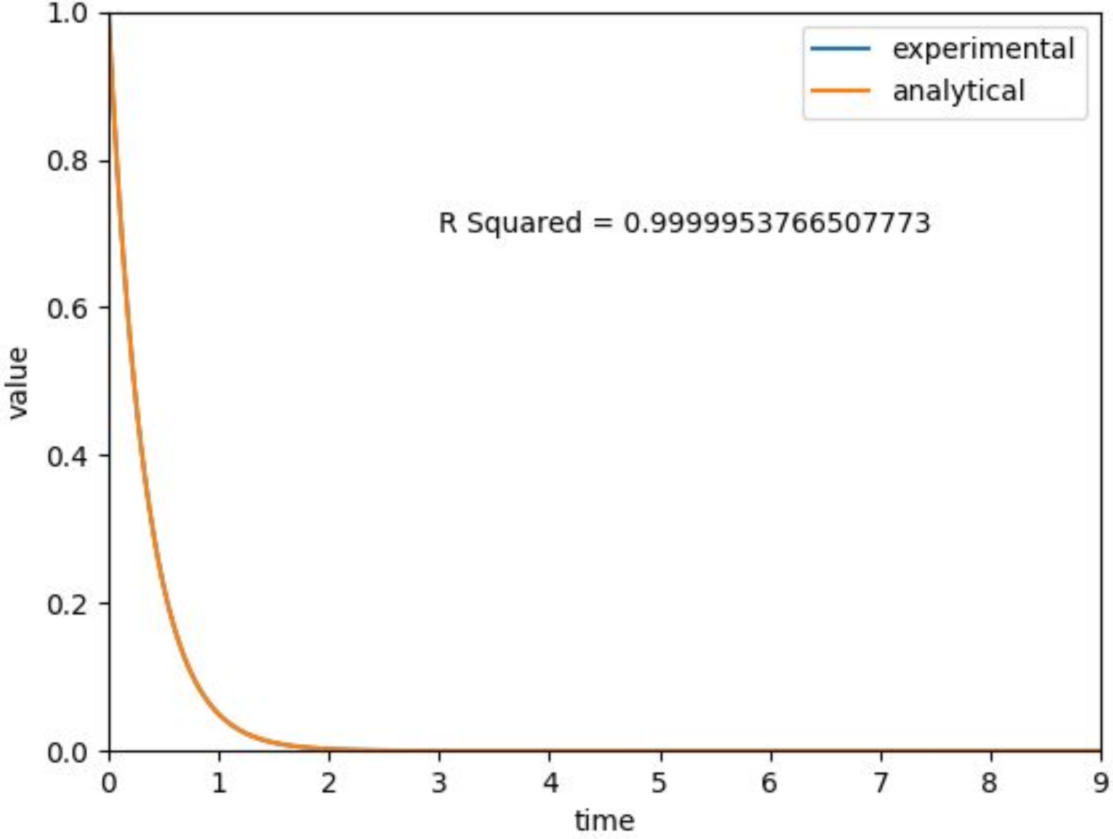
1.0
0.8
0.6
0.4
0.2
0.0
value
0
1
2
3
4
5
6
7
8
9
time
experimental
analytical
R Squared = 0.6117902568926971

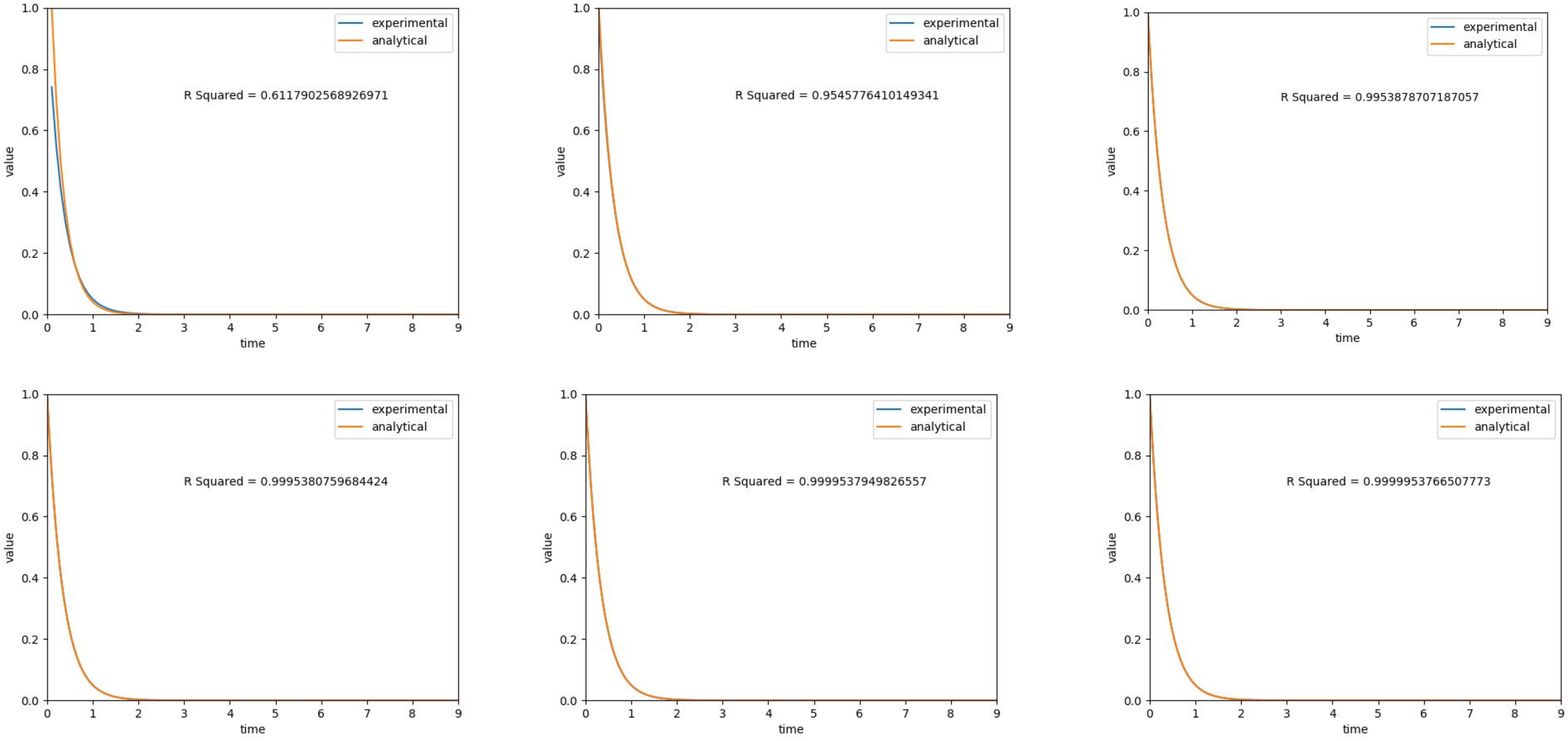

1.0
0.8
0.6
0.4
0.2
0.0
value
0
1
2
3
4
5
6
7
8
9
time
experimental
analytical
R Squared = 0.9545776410149341


experimental
analytical
R Squared = 0.9953878707187057
value
1.0
0.8
0.6
0.4
0.2
0.0
0
1
2
3
4
5
6
7
8
9
time


1.0
0.8
0.6
0.4
0.2
0.0
value
0
1
2
3
4
5
6
7
8
9
time
experimental
analytical
R Squared = 0.9995380759684424


experimental
analytical
R Squared = 0.9999537949826557
value
1.0
0.8
0.6
0.4
0.2
0.0
0
1
2
3
4
5
6
7
8
9
time


1.0
0.8
0.6
0.4
0.2
0.0
value
0
1
2
3
4
5
6
7
8
9
time
experimental
analytical
R Squared = 0.9999953766507773


experimental
analytical
R Squared = 0.6117902568926971
R Squared = 0.9545776410149341
R Squared = 0.9953878707187057
R Squared = 0.9995380759684424
R Squared = 0.9999537949826557
R Squared = 0.9999953766507773
value
time
0.0
0.2
0.4
0.6
0.8
1.0
0
1
2
3
4
5
6
7
8
9

Suggestions

- Some sort of tutorial or information on modules and which ones would be needed would be nice.
- A cheat sheet of some of the more common commands in linux and bash and what they do/mean.
- Something about file transfer between remote locations using scp or sftp kind of things if you are not using one of the file transfer programs.
- SSH Keys help such as this site.
- Maybe a vim tutorial link as well since that is the easiest text editor to work with in Linux (or at least the most convenient.)

Questions?
Comments?
Concerns?
Compliments?