Progress Report: Sprint 2

CS.028 Optimizing Research Software Codes - 26 October 2025

Kathryn Butler, Michael McAllister, Joseph Schaab

Executive Summary

- Our repository has been verified and we've analyzed the initial runtime using the sample GNSS datasets. We've discovered bottlenecks in init and load_config.
- We've implemented logging tools for future testing, and are planning to modulate the main script and further automate testing.
- Our team encountered environment setup delays but achieved code execution and are researching ways to improve init and load_config.

Progress vs Plan

Planned Item	Status	Links	Notes/Blockers	Plan Change
Clone repo and verify initial run	Done	GitHub	Initial run verified on test data	None
Examine main functions for runtime	Done	GitHub	See timing summary	None
Identify top problem functions	Done	GitHub	init and load_config = 95% of runtime	Focus on specifically
Implement function testing	Partial	N/A	Still researching how to optimize the functions	Extend to next sprint
Documentation draft	Partial	N/A	Outlines started, substantial progress delayed	Extend one week

Evidence of Working Software

Commit: GitHub

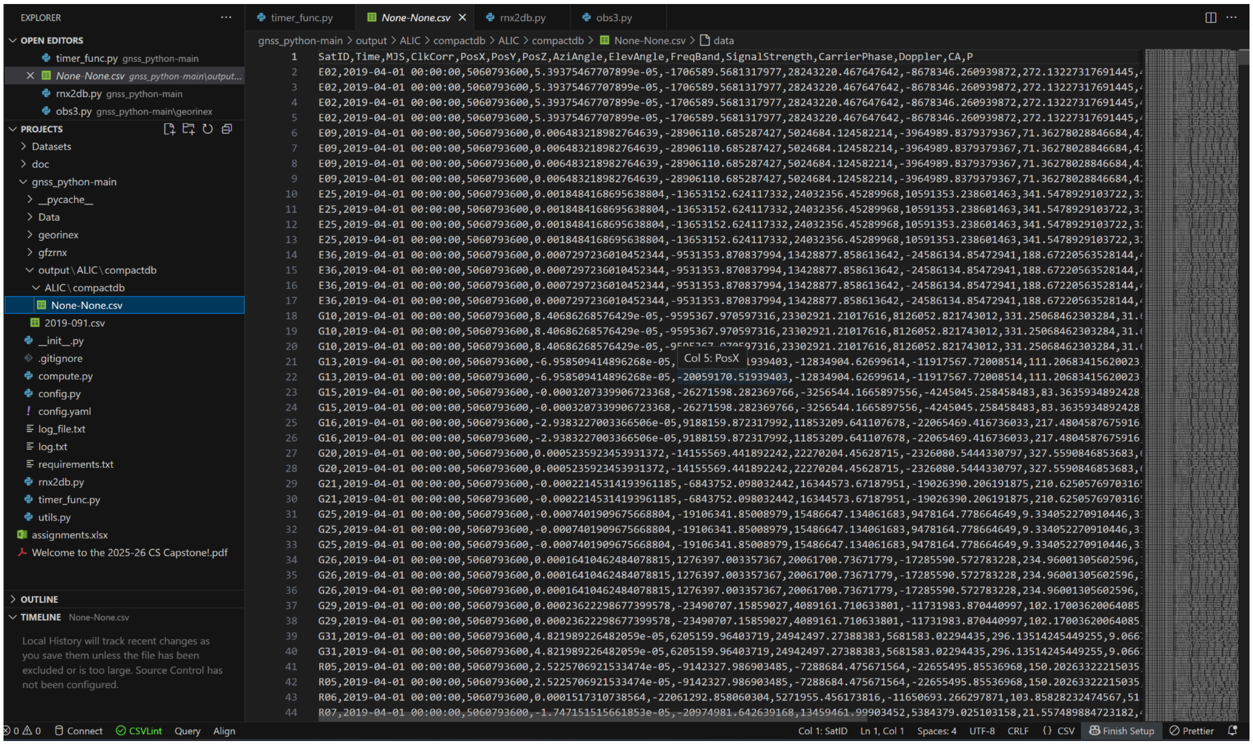
Run Output: The run output is a very large CSV file, included in screenshots.

Test Summary:

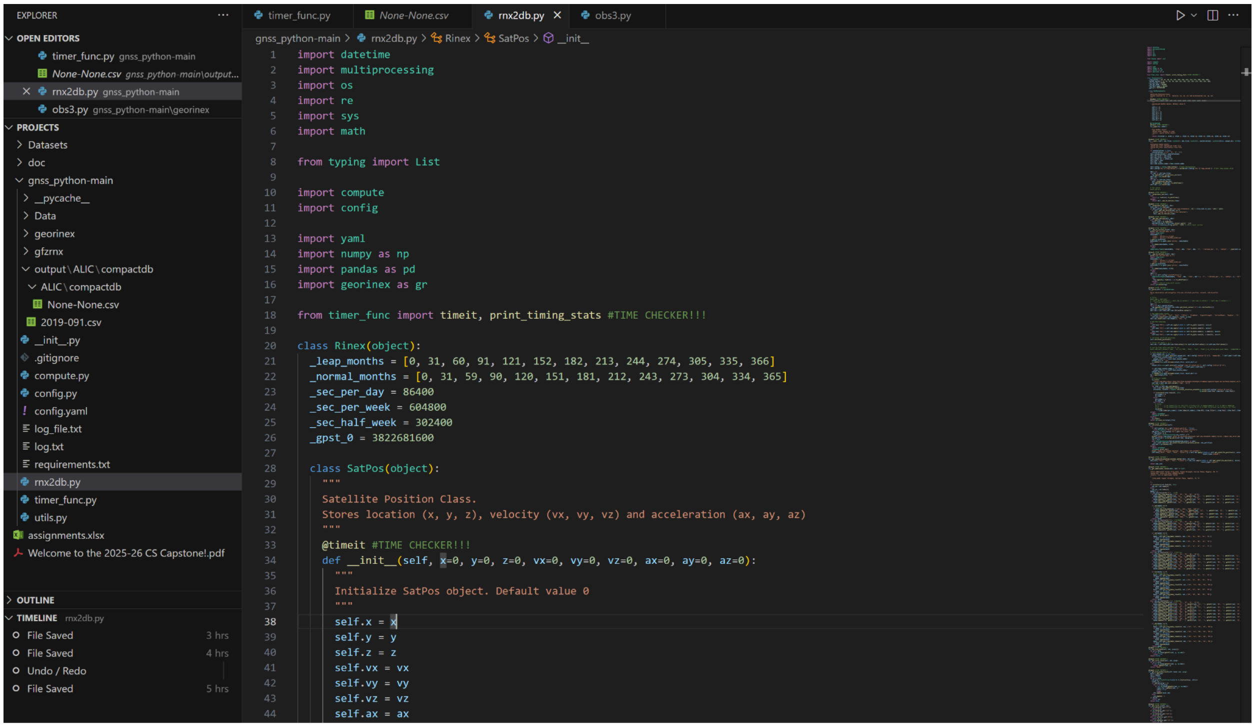
So far we have gotten the inherited code running on an Anaconda environment, and implemented 1 test to determine potential bottlenecks in our code. We have created a python tool titled timer_func.py that keeps track of all accumulated time spent inside each function inside rnx2db.py and prints this information after the process is complete. This will be very useful in all future testing as it will be a tool we can use to check the speed of any function in any of the files inside /gnss_python-main/

CI/Build: https://github.com/KathrynJButler/optimizing-research-software-codes

Screenshots:



```
SatID,Time,MJS,ClkCorr,PosX,PosY,PosZ,AziAngle,ElevAngle,FreqBand,SignalStrength,CarrierPhase,Doppler,CA,P
E02,2019-04-01 00:00:00,5060793600,5.39375467707899e-05,-1706589.5681317977,28243220.467647642,-8678346.260939872,272.13227317691445,
E02,2019-04-01 00:00:00,5060793600,5.39375467707899e-05,-1706589.5681317977,28243220.467647642,-8678346.260939872,272.13227317691445,
E02,2019-04-01 00:00:00,5060793600,5.39375467707899e-05,-1706589.5681317977,28243220.467647642,-8678346.260939872,272.13227317691445,
E02,2019-04-01 00:00:00,5060793600,5.39375467707899e-05,-1706589.5681317977,28243220.467647642,-8678346.260939872,272.13227317691445,
E09,2019-04-01 00:00:00,5060793600,0.006483218982764639,-28906110.685287427,5024684.124582214,-3964989.8379379367,71.36278028846684,4
E09,2019-04-01 00:00:00,5060793600,0.006483218982764639,-28906110.685287427,5024684.124582214,-3964989.8379379367,71.36278028846684,4
E09,2019-04-01 00:00:00,5060793600,0.006483218982764639,-28906110.685287427,5024684.124582214,-3964989.8379379367,71.36278028846684,4
E09,2019-04-01 00:00:00,5060793600,0.006483218982764639,-28906110.685287427,5024684.124582214,-3964989.8379379367,71.36278028846684,4
E25,2019-04-01 00:00:00,5060793600,0.0018484168695638804,-13653152.624117332,24032356.45289968,10591353.238601463,341.5478929103722,3
E25,2019-04-01 00:00:00,5060793600,0.0018484168695638804,-13653152.624117332,24032356.45289968,10591353.238601463,341.5478929103722,3
E25,2019-04-01 00:00:00,5060793600,0.0018484168695638804,-13653152.624117332,24032356.45289968,10591353.238601463,341.5478929103722,3
E25,2019-04-01 00:00:00,5060793600,0.0018484168695638804,-13653152.624117332,24032356.45289968,10591353.238601463,341.5478929103722,3
E36,2019-04-01 00:00:00,5060793600,0.0007297236010452344,-9531353.870837994,13428877.858613642,-24586134.85472941,188.67220563528144,
E36,2019-04-01 00:00:00,5060793600,0.0007297236010452344,-9531353.870837994,13428877.858613642,-24586134.85472941,188.67220563528144,
E36,2019-04-01 00:00:00,5060793600,0.0007297236010452344,-9531353.870837994,13428877.858613642,-24586134.85472941,188.67220563528144,
E36,2019-04-01 00:00:00,5060793600,0.0007297236010452344,-9531353.870837994,13428877.858613642,-24586134.85472941,188.67220563528144,
G10,2019-04-01 00:00:00,5060793600,8.40686268576429e-05,-9595367.970597316,23302921.21017616,8126052.821743012,331.25068462303284,31.
G10,2019-04-01 00:00:00,5060793600,8.40686268576429e-05,-9595367.970597316,23302921.21017616,8126052.821743012,331.25068462303284,31.
G10,2019-04-01 00:00:00,5060793600,8.40686268576429e-05,-9595367.970597316,23302921.21017616,8126052.821743012,331.25068462303284,31.
G13,2019-04-01 00:00:00,5060793600,-6.958509414896268e-05,               1939403,-12834904.62699614,-11917567.72008514,111.20683415620023
G13,2019-04-01 00:00:00,5060793600,-6.958509414896268e-05,-20059170.51939403,-12834904.62699614,-11917567.72008514,111.20683415620023
G15,2019-04-01 00:00:00,5060793600,-0.0003207339906723368,-26271598.282369766,-3256544.1665897556,-4245045.258458483,83.3635934892428
G15,2019-04-01 00:00:00,5060793600,-0.0003207339906723368,-26271598.282369766,-3256544.1665897556,-4245045.258458483,83.3635934892428
G16,2019-04-01 00:00:00,5060793600,-2.9383227003366506e-05,9188159.872317992,11853209.641107678,-22065469.416736033,217.4804587675916
G16,2019-04-01 00:00:00,5060793600,-2.9383227003366506e-05,9188159.872317992,11853209.641107678,-22065469.416736033,217.4804587675916
G20,2019-04-01 00:00:00,5060793600,0.0005235923453931372,-14155569.441892242,22270204.45628715,-2326080.5444330797,327.5590846853683,
G20,2019-04-01 00:00:00,5060793600,0.0005235923453931372,-14155569.441892242,22270204.45628715,-2326080.5444330797,327.5590846853683,
G21,2019-04-01 00:00:00,5060793600,-0.00022145314193961185,-6843752.098032442,16344573.67187951,-19026390.206191875,210.6250576970316
G21,2019-04-01 00:00:00,5060793600,-0.00022145314193961185,-6843752.098032442,16344573.67187951,-19026390.206191875,210.6250576970316
G25,2019-04-01 00:00:00,5060793600,-0.0007401909675668804,-19106341.85008979,15486647.134061683,9478164.778664649,9.334052270910446,3
G25,2019-04-01 00:00:00,5060793600,-0.0007401909675668804,-19106341.85008979,15486647.134061683,9478164.778664649,9.334052270910446,3
G25,2019-04-01 00:00:00,5060793600,-0.0007401909675668804,-19106341.85008979,15486647.134061683,9478164.778664649,9.334052270910446,3
G26,2019-04-01 00:00:00,5060793600,0.00016410462484078815,1276397.003357367,20061700.73671779,-17285590.572783228,234.96001305602596,
G26,2019-04-01 00:00:00,5060793600,0.00016410462484078815,1276397.003357367,20061700.73671779,-17285590.572783228,234.96001305602596,
G26,2019-04-01 00:00:00,5060793600,0.00016410462484078815,1276397.003357367,20061700.73671779,-17285590.572783228,234.96001305602596,
G29,2019-04-01 00:00:00,5060793600,0.00023622298677399578,-23490707.15859027,4089161.710633801,-11731983.870440997,102.17003620064085
G29,2019-04-01 00:00:00,5060793600,0.00023622298677399578,-23490707.15859027,4089161.710633801,-11731983.870440997,102.17003620064085
G31,2019-04-01 00:00:00,5060793600,4.821989226482059e-05,6205159.96403719,24942497.27388383,5681583.02294435,296.13514245449255,9.066
G31,2019-04-01 00:00:00,5060793600,4.821989226482059e-05,6205159.96403719,24942497.27388383,5681583.02294435,296.13514245449255,9.066
R05,2019-04-01 00:00:00,5060793600,2.5225706921533474e-05,-9142327.986903485,-7288684.475671564,-22655495.85536968,150.20263322215035
R05,2019-04-01 00:00:00,5060793600,2.5225706921533474e-05,-9142327.986903485,-7288684.475671564,-22655495.85536968,150.20263322215035
R06,2019-04-01 00:00:00,5060793600,0.000151731073856 4,-22061292.858060304,5271955.456173816,-11650693.266297871,103.85828232474567,51
R07,2019-04-01 00:00:00,5060793600,-1.747151515661853e-05,-20974981.642639168,13459461.99903452,5384379.025103158,21.557489884723182,
```



```python
import datetime
import multiprocessing
import os
import re
import sys
import math

from typing import List

import compute
import config

import yaml
import numpy as np
import pandas as pd
import georinex as gr

from timer_func import timeit, print_timing_stats #TIME CHECKER!!!

class Rinex(object):
  _leap_months = [0, 31, 60, 91, 121, 152, 182, 213, 244, 274, 305, 335, 366]
  _normal_months = [0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334, 365]
  _sec_per_day = 86400
  _sec_per_week = 604800
  _sec_half_week = 302400
  _gpst_0 = 3822681600

  class SatPos(object):
    """
    Satellite Position Class.
    Stores location (x, y, z), velocity (vx, vy, vz) and acceleration (ax, ay, az)
    """
    @timeit #TIME CHECKER!!!
    def __init__(self, x=0, y=0, z=0, vx=0, vy=0, vz=0, ax=0, ay=0, az=0):
      """
      Initialize SatPos object. Default value 0
      """
      self.x = x
      self.y = y
      self.z = z
      self.vx = vx
      self.vy = vy
      self.vz = vz
      self.ax = ax
```

Risk & Quality

Risk	Owner	Status	Concrete Actions
Code complexity	Team	Active	Begin modulating code sections, add inline comments
Setup inconsistencies	Team	Resolved	Documented environment dependencies
Limited testing time	Team	Active	Prioritize optimizing the slowest functions

Bug count: 0 known errors so far, the only issue is runtime efficiency.

Next Goals

- REQ-006: Add validation tests to confirm an initial runtime.
- REQ-008: Log the runtime tests to an output file and store them on GitHub.
- REQ-011: Begin documentation to log performance and reproducibility.
- REQ-014: Create additional GitHub branches for testing and experimentation.

Team Process Reflection

During the last sprint, our primary concern was establishing a stable work environment for our code base, which contains many unique requirements. We achieved this through the use of Anaconda which was recommended to us by our project partner. With the primary problem facing our team out of the way, we moved on to planning our next steps. We landed on implementing a testing tool so that we could check the total runtime of all functions in the main file. Now that that has been completed, we are well positioned to move forward with our primary aim of decreasing the overall runtime of the program through optimization.

Individual Contributions

Kathryn Butler

- Created Progress Report and Requirements documentation
- Verified program runtime results and created the runtime visualizations
- Maintained the README and related documentation

Michael Mcallister

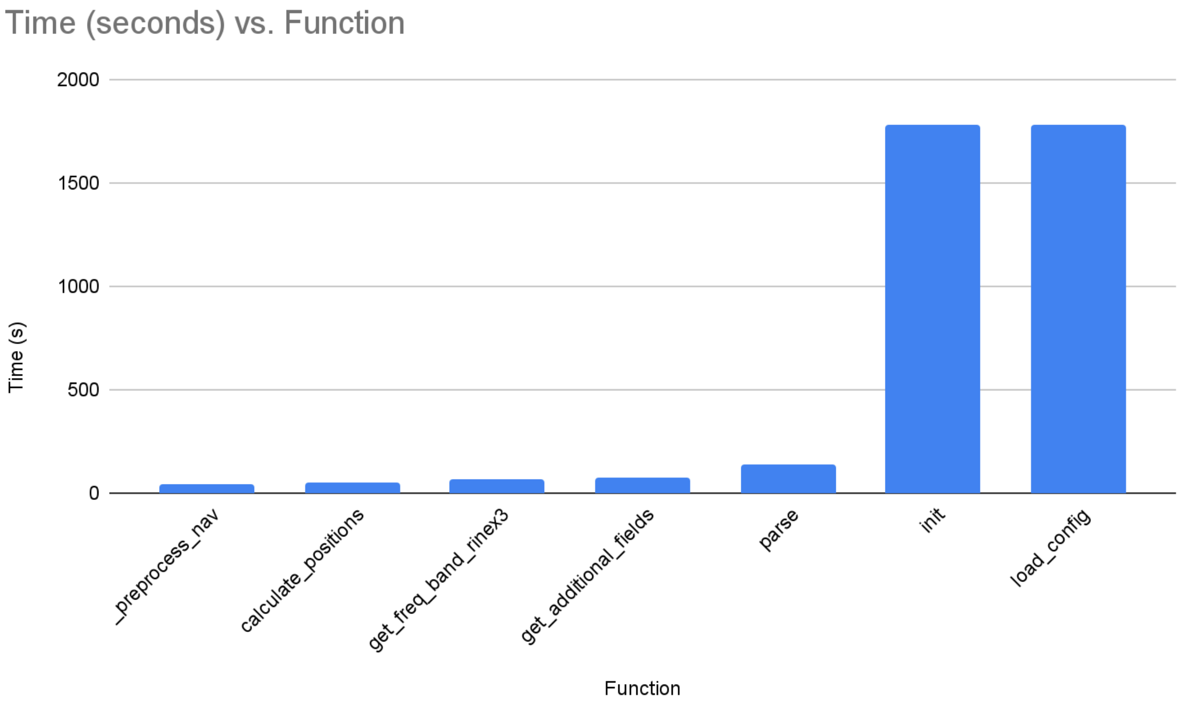
- Created separate branch for testing
- Created timer_func.py and implemented bottleneck tests in rnx2db.py
- Gathered runtime results by running the program locally
- Make PR to merge main and testing branch to make testing easier

Joseph Schaab

- Tested efficiency benefits and constraints of various software environments
- Verified Anaconda dependencies (if we decide to change how we run the program

- Researched memory mapping strategies with the goal of reducing exponential runtimes. https://realpython.com/python-mmap/

Most time-consuming functions:


Time (seconds) vs. Function
Time (s)
_preprocess_nav
calculate_positions
get_freq_band_rinex3
get_additional_fields
parse
init
load_config
Function