GeoSAMS Startup

Table of Contents

[1 GeoSAMS Setup 1](#_Toc186778486)

[1.1 Git 1](#_Toc186778487)

[1.2 Python 1](#_Toc186778488)

[1.2.1 Python packages 1](#_Toc186778489)

[1.3 R 2](#_Toc186778490)

[1.4 Math Support 2](#_Toc186778491)

[1.4.1 Matlab 2](#_Toc186778492)

[1.4.2 Octave 2](#_Toc186778493)

[1.5 Get software and support files 2](#_Toc186778494)

[1.6 Windows 10 Terminal Color 2](#_Toc186778495)

[2 Using the GUI 4](#_Toc186778496)

[2.1 Math Setup 4](#_Toc186778497)

[2.2 GUI Help 5](#_Toc186778498)

[2.3 Starting the Simulation 5](#_Toc186778499)

[2.3.1 Early Termination 6](#_Toc186778500)

[2.4 Upon completion 6](#_Toc186778501)

[3 Manually Running GeoSAMS 7](#_Toc186778502)

[3.1 Unpack.sh/.bat 7](#_Toc186778503)

[3.2 Run GeoSAMS 8](#_Toc186778504)

[3.3 Process Results, DEPRECATED – not kept up to date, now handled by GUI 8](#_Toc186778505)

[3.4 Rscript and Python Kriging 9](#_Toc186778506)

[3.4.1 GAMS Residual computation 9](#_Toc186778507)

[3.4.2 Python Ordinary Kriging 9](#_Toc186778508)

[3.4.3 From within Matlab or Octave 9](#_Toc186778509)

# GeoSAMS Setup

Starting GeoSAMS can be done entirely via a GUI or manually via scripts, however, before it can be run there exists applications that need to be configured.

* Git
* Python
* Microsoft C++ Build Tools
* R
* Math support, either Matlab or Octave

## Git

Git needs to be installed to pull all files from a repository that supports GeoSAMS. For information on how to use git and other tutorials visit <https://www.atlassian.com/git/tutorials/install-git>

See the NEFS-IT Service Desk to actually have git installed on your GFE, <https://apps-st.fisheries.noaa.gov/jirasm/servicedesk/customer/portal/2>

## Python

Python is considered a high-level, general-purpose language, meaning it can be used to build a wide variety of software applications, from websites and data analysis tools to complex scientific simulations, due to its user-friendly syntax and extensive libraries; it's often praised for being beginner-friendly and highly versatile.

See the NEFS-IT Service Desk to actually have python installed on your GFE.

The user is not expected to have a working knowledge of Python. The GeoSAMS GUI calls Python-scripts that do the algorithmic processing.

### Microsoft C++ Build Tools

Before pykrige can be installed, the user may need to get Microsoft C++ Build Tools installed so that the python installer can build any needed libraries. Request the tool from ITD. See the following link:

<https://visualstudio.microsoft.com/visual-cpp-build-tools/>

### Python packages

There are several python packages that GeoSAMS utilizes. These are installed via the command line. “--user" places the package in your personal library[[1]](#footnote-1) as due to ITD restricitions the system libraries are not writeable.

> python -m pip install --user <pkg>

or

> pip install --user <pkg>

Needed packages

* utm
* numpy
* pandas
* pykrige (See Section 1.2.1)

Other packages which missed install can be installed as instructed above.

## R

R is a free, open-source programming language and software environment. R is optimized for statistical computing and data visualization.

See the NEFS-IT Service Desk to actually have R installed on your GFE.

The user is not expected to have a working knowledge of R. The GeoSAMS GUI calls R-scripts that do the algorithmic processing.

## Math Support

To run this simulation you will need to have Matlab or Octave installed. Octave is a free software package distributed under a GNU License and can be installed by visiting: <https://octave.org/download>

The user is not expected to have a working knowledge of Matlab/Octave. The GeoSAMS GUI calls m-scripts that do the algorithmic processing.

NOTE: Only Matlab approved for install on GFE, Octave is not.

### Matlab

See the NEFS-IT Service Desk to actually have Matlab installed on your GFE. Only Matlab is necessary. GeoSAMS is not using any of the Matlab toolboxes. It does use free software under the terms of the GNU General Public License.

### Octave

Octave will also need additional packages from <https://gnu-octave.github.io/packages/>

* io
* geometry
* mapping
* matgeom
* statistics

As you install these packages take note of where they are installed. Although you can find out from inside of Octave by executing, ***pkg list***.

## Get software and support files

The next step will be to get the current GeoSAMS release. This is easily done via git.

Navigate to the directory where you want to install the files. Then enter the command.

* git clone <https://github.com/NEFSC/READ-PDP-GeoSAMS.git> <GeoRoot>

Note: If you intend to make changes managed under git then you will need to pull the git controlled files with

* git clone [git@github.com:NEFSC/READ-PDP-GeoSAMS.git](mailto:git@github.com:NEFSC/READ-PDP-GeoSAMS.git) <GeoRoot>

## Windows 10 Terminal Color

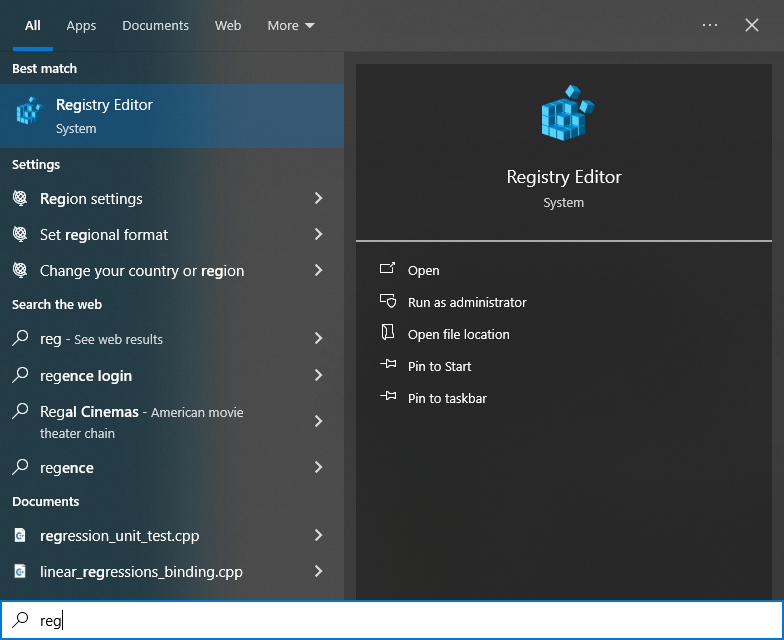
By default, Win10 does not support ANSI Escape Sequences to set font color. To use ANSI colours in the Windows terminal requires setting VirtualTerminalLevel.

In Windows 10 versions 1511 through to 1903 this had to be enabled in the registry at:

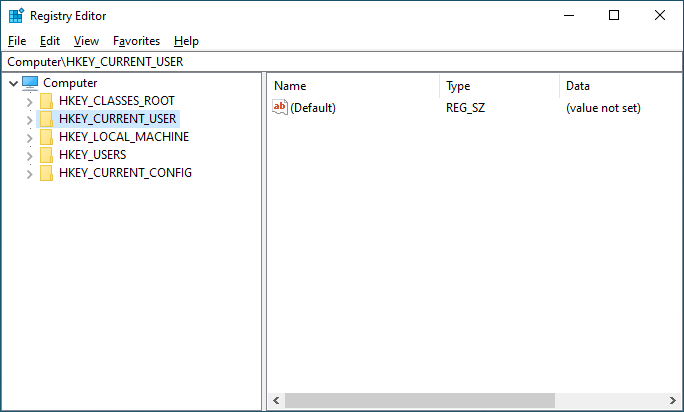
[HKEY\_CURRENT\_USER\Console]

"VirtualTerminalLevel"=dword:00000001

1. Open Registry Editer by click on windows icon and start typing reg.. then select Registry Editor.



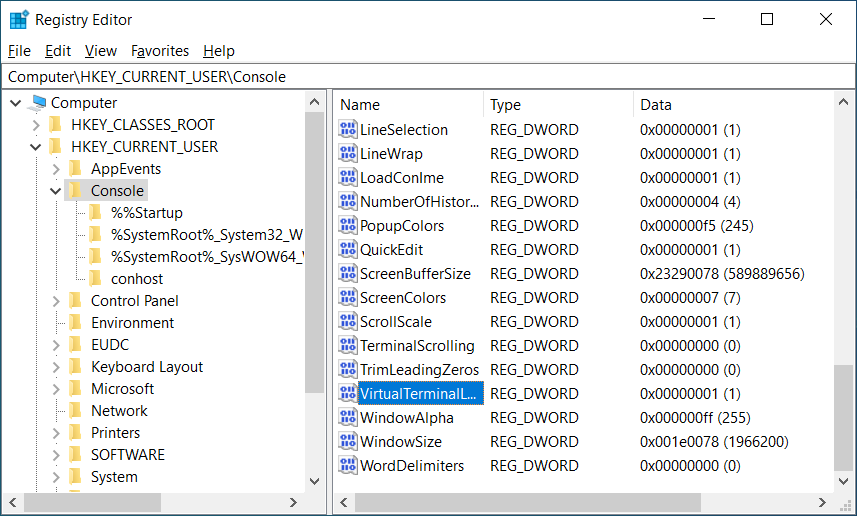
You should see a window similar to:



1. Navigate to

Computer -> HKEY\_CURRENT\_USER -> Console

1. Then right click in right pane and select New -> Dword
2. Enter VirtualTerminalLevel
3. Right Click on new entry, select Modify, and set value to 1



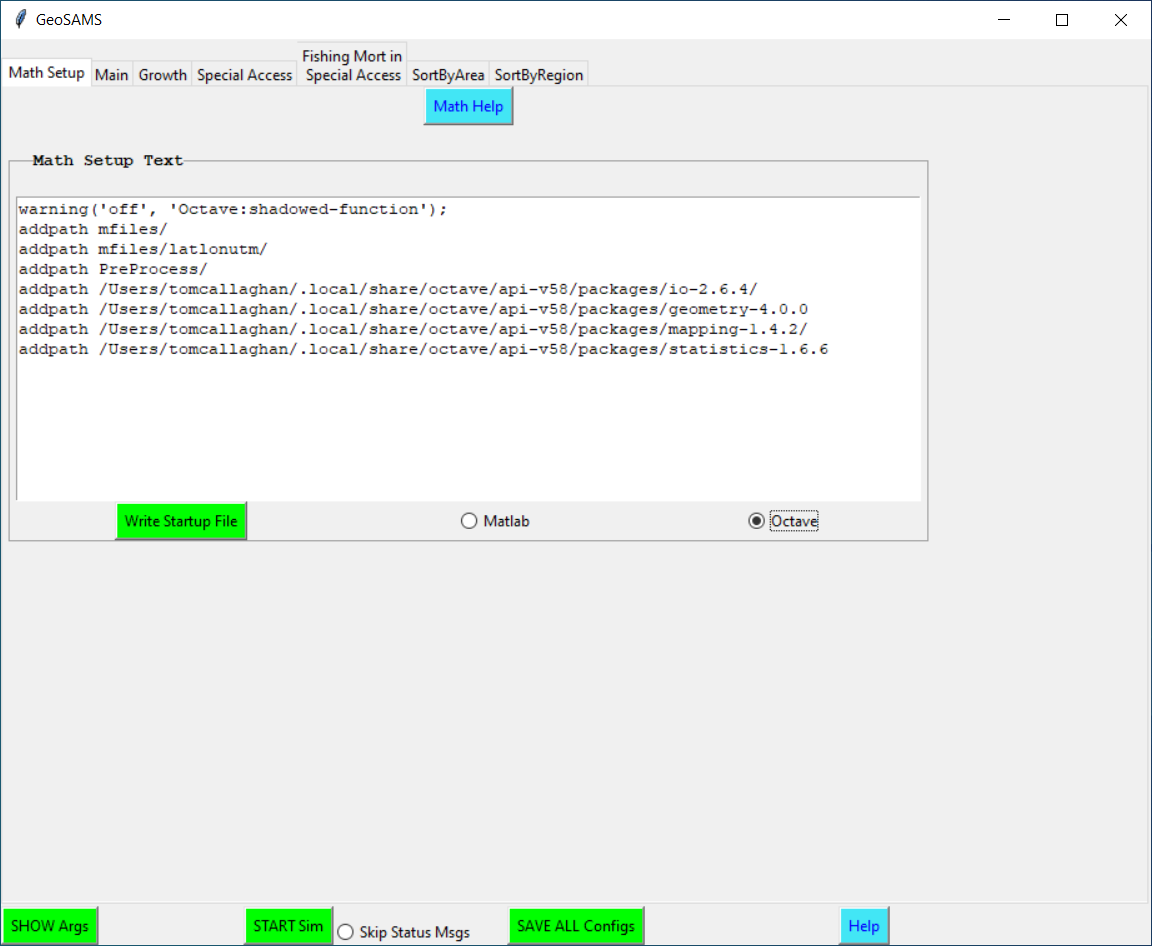
# Using the GUI

The remaining setup is all done via the GUI. This is opened via the command line at the <GeoRoot> directory.

<GeoRoot> > python PythonScripts\GUI\GeoSAM\GeoSams.py

## Math Setup

Upon opening the GUI starts in the Math Setup Tab. This allows the user to select the Math Tool they wish to use, either Matlab or Octave



Here you can edit the .octaverc file to let Octave know where packages have been installed. Click the Octave radio button and you will be presented with the current setting. Make the changes and click Write Startup File to save the file.

On the off chance you will use Matlab, it defaults to that radio button. You should not need to make any changes as the information presented has already been installed when downloaded from git.

## GUI Help

Each tab within the GUI has a help button noted by a blue background on WIN machines or blue outline otherwise. Clicking this button will open another scrollable window that the user can review for more information on what is present on that tab.

## Starting the Simulation

Peruse the other tabs in the GUI. For now the defaults should be adequate. When ready click the ***START Sim*** at the bottom of the GUI. Note: Next to the this button is a radio button, *Skip Status Msgs*. When this button is enabled the GUI will skip showing status messages as each step is completed and will run throught to the end, i.e. data plotting is complete.

START Sim button caused several steps to occur.

A. Saves all of the configuration files, overwriting if no names were changed.

B. When this button is clicked the GUI checks to see if the needed data CSV files have been created. If not it will run the Update.bat/.sh script. This script performs the following:

1. Unzip Dredge Data and HabCam data files

2. Builds GeoSams and UK executables

3. Runs the pull out survey data m-file scripts

**a. TrawlData5mmbin.m** takes the trawl data from Dredge Survey Data file, defaults to *OriginalData/dredgetowbysize7917.csv* and writes it to *Data/bin5mmYYYY[AL|MA|GB].csv*. The output has the dredge observations in 5mm bins for the startup year and domain.

**b. HabCamData5mmbin.m** takes the trawl data from HabCam Survey Data file, defaults to *OriginalData/Habcam\_BySegment\_2000\_2011-2023\_v2.csv* and appends it to, *Data/bin5mmYYYY[AL|MA|GB].csv*". The output has the HabCam observations in 5mm bins for the startup year and domain requested.

4. Runs the PullOutProcessRecruitDatam-file script

**PullOutProcessRecruitData.m** looks at the both trawl data and HabCam data from *OriginalData/dredgetowbysize7917.csv* and *OriginalData/Habcam\_BySegment\_2000\_2011-2023\_v2.csv*, sums the shell lengths from 3 cm to 6 cm.

5. Runs the NearestNeighborRecInterp m-file script

C. Starts the GeoSAMS growth simulation, e.g.  
 > \SRC\ScallopPopDensity.exe Scallop.cfg 2022 2025 AL

D. Starts Rscript GAM residuals, Python Ordinary Kriging iterpolation that takes the GeoSAMS output position on survey locations and interpolates them to regional locations

E. Plots the results and saves the plots as PDF files in <GeoRoot>\Results

As different applications are run and completed, the GUI will display message boxes informing the user of its progress. Or if the user has enabled *Skip Status Msgs*.then no additional messages until process completes

### Early Termination

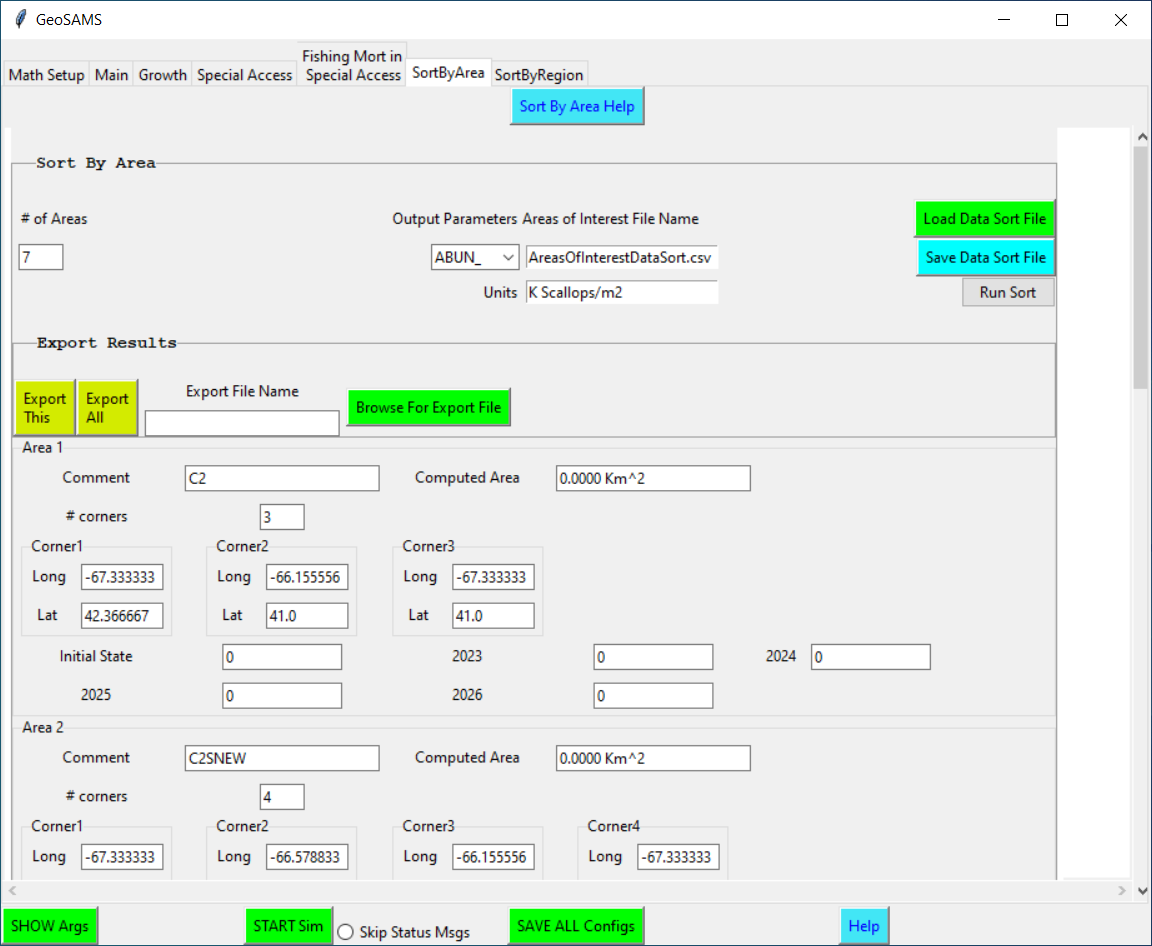
If for any reason the user wishes to terminate the program before completion, type Ctrl-C in the window where you started the GUI. This will terminate the currently executing program but leave the GUI in tact. Typing Ctrl-C twice will stop the current executable and the GUI.

## Upon completion

The user now has a couple of options.

1) Under the Main Tab you will see a button to open and view the PDF plot files.

2) Under the SortByArea Tab



Here the user can click Run Sort to accumulate the identified output parameters by the areas identified in the frames. It will do so by the years given on the Main Tab.

The user may then export this data either for a single output identified in the Output Parameter pull down, ***Export This***, or for all of the identified outputs, ***Export All***.

# Manually Running GeoSAMS

The user does not necessarily need to run the GUI. Executing scripts manually or starting executables are available to walk the simulation and plotting through the command line

## Unpack.sh/.bat

This only needs to be run once to obtain the desired years data. See section 2.3, B.1-6 explains all that it does. Running it without arguments informs the user what is expected:

* <GeoRoot> .\Unpack.bat

Missing arguments

"Expecting: Unpack.bat ReferenceYr RecrYrStrt RecrYrStop Domain [M|O]"

"Domain"

" MA"

" GB"

" ALL, both MA and GB"

"[M|O]"

" M: Use Matlab for numerical processing"

" O: Use Octave for numerical processing"

## Run GeoSAMS

Entering the following command will start the GeoSAMS growth simulation.

* <GeoRoot> .\SRC\ScallopPopDensity.exe Scallop.cfg 2015 2017 AL

Without arguments the user will see this response:

* <GeoRoot> .\SRC\ScallopPopDensity.exe

Missing parameters

Typical use: $ .\SRC\ScallopPopDensity.exe Scallop.cfg StartYear StopYear Domain

## Process Results, DEPRECATED – not kept up to date, now handled by GUI

Entering the following command will complete the universal kriging, UK, interpolation and results plots.

* <GeoRoot> python .\PythonScripts\ProcessResults.py AL 2015 2017 Scallop.cfg UK.cfg

Without arguments the user will get this response:

* <GeoRoot> python .\PythonScripts\ProcessResults.py

Missing command line arguments. Expecting:

$ ProcessResults.py Domain StartYear EndYear Scallop.cfg UK.cfg

The simulation configuration file, Scallop.cfg, is needed so that the interpolation algorithm knows which output parameters need to be interpolated.

1. <GeoSamsRoot>\Results

These are the files output by ScallopPopDensity.exe into the Results directory. There are two sets of files:

1. Lat\_Lon\_Surv\_PARAM\_DN.csv: Survey Growth Data organized by #time steps across by #grids down
2. Lat\_Lon\_Grid\_PARAM\_DN.csv: Interpolated Survey Data onto Region Grids organized by years across and #grids down.   
   Note: the first year column is the intial state for that data.

|  |  |
| --- | --- |
| **Param** | **Equation** |
| ABUN | Abundance\_psqm x grid\_area |
| BMMT | (state(grid,:) • weight(grid,:) / grams\_mt) |
| EBMS | ({selectivity(grid, :) x state(grid,:)} • weight(grid,:)) x grid\_area / grams\_mt |
| FEFF | (BMS x catch / abun) / (sum(BMS2 / abun)) |
| FMOR | C x dot\_product(selectivity(grid, :) x state(grid, :),  weight\_grams(grid,:)) |
| LAND | sum[ (1.\_dp - exp(-FMOR(grid) x delta\_t))  x selectivity(grid, :) x state(grid, :) x grid\_area ] |
| LNDW | LAND • weight(grid,:))/grams\_mt |
| LPUE | computed |
| RECR | file lookup |

mt: metric tons

## Rscript and Python Kriging

Whereas post processing after growth simulation is now handled by the GUI. The user can still run some steps by hand to aid analysis

### GAMS Residual computation

Rscript .\GAM\R\_GAM\_GeoSAMS.r X\_Y\_<param>\_<DN><YYYY>\_>MA|GB>\_BUFFER <MA|GB>RegionGrid.csv, e.g.  
  
> Rscript .\GAM\R\_GAM\_GeoSAMS.r X\_Y\_RECR\_AL2022\_GB\_BUFFER GBRegionGrid.csv

### Python Ordinary Kriging

python .\PythonScripts\OK\pykrige\_geosams.py X\_Y\_<param>\_<DN><YYYY>\_<MA|GB>, e.g.  
  
> python .\PythonScripts\OK\pykrige\_geosams.py X\_Y\_RECR\_AL2022\_GB

### From within Matlab or Octave

View results within math tool for further analysis

**PlotLatLonGridSurvey**('Results/Lat\_Lon\_Surv\_BIOM\_AL', 'Results/Lat\_Lon\_Grid\_BIOM\_AL\_2022\_2026', 2022, 13, 'AL')

Or just the grid data without survey results

**PlotLatLonGrid**('Results/Lat\_Lon\_Grid\_RECR\_AL\_2022\_2026', 2022, 'AL', 1)

1. Location: C:\Users\<user name>\AppData\Roaming\Python\Python313\site-packages [↑](#footnote-ref-1)