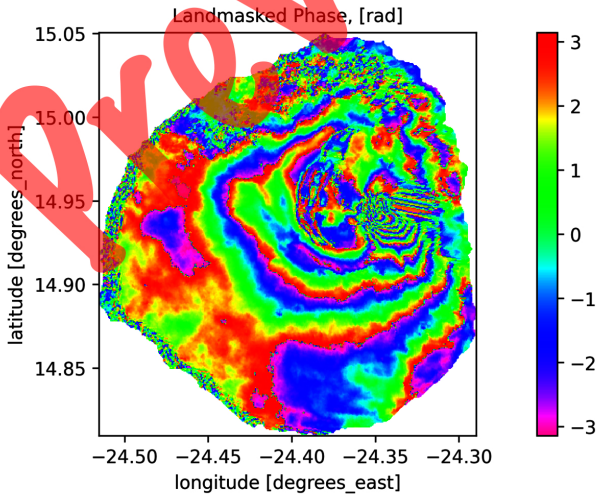
ALEXEY PECHNIKOV

PyGMTSAR: Sentinel-1 Python InSAR

An Introduction



Written by Alexey Pechnikov.

Annotation

The "PyGMTSAR: Sentinel-1 Python InSAR" book series serves as your gateway to mastering the innovative world of Sentinel-1 satellite interferometry using the open-source Python InSAR library, PyGMTSAR. Authored by the developer himself, these books act as hands-on guides for working with PyGMTSAR, whether through Jupyter notebooks or console Python scripts.

The book "PyGMTSAR: Sentinel-1 Python InSAR. An Introduction" employs Google Colab, a free-to-use cloud service, as an ideal platform for beginners. Readers can explore the applications of PyGMTSAR, from seismic activity tracking to infrastructure health assessment, through a series of interactive notebooks. Each notebook comes complete with adaptable instructions to facilitate personalized learning.

The guide also introduces Docker Desktop, an advanced open-source platform for containerization. The PyGMTSAR Docker image sets up a workspace similar to a traditional one, enabling more intense computations on your local computer and on cloud hosts. All the Google Colab examples are available.

This tutorial sheds light on the principles of lazy and delayed computations. It explains how Dask, an advanced task scheduler, intelligently partitions and schedules tasks. These insights enhance your ability to handle Big Data processing with PyGMTSAR efficiently, whether on your local machine or cloud-based systems.

Whether you're a student, a researcher, or an industry professional with an interest in remote sensing and earth observation, the "PyGMTSAR: Sentinel-1 Python InSAR. An Introduction" book equips you with the necessary skills and knowledge to navigate Python-based satellite interferometry.

Table of Contents

1. Overview

In this chapter, we will explore two crucial elements: the fundamental concepts underlying satellite interferometry and an introduction to PyGMTSAR, a powerful tool for processing and interpreting InSAR data.

Section 1.1 provides a roadmap to understanding the basics of satellite interferometry, specifically Interferometric Synthetic Aperture Radar (InSAR). This advanced remote sensing technique involves the use of two or more synthetic aperture radar (SAR) images to generate highly accurate maps of surface deformation or digital elevation models of terrain. This process relies on the detailed calculation of the phase difference between return signals from two nearly identical images, known as interferograms. Furthermore, it highlights how PyGMTSAR is utilized in processing these images, performing operations like Doppler correction, topography correction, and applying Gaussian and Werner/Goldstein filters for noise reduction. The section also elaborates on the SBAS technique for displacement calculation and how the resulting time-series data can be analyzed to extract trends and seasonal movements.

Section 1.2 introduces PyGMTSAR, a powerful software designed to encapsulate the complexities of InSAR processing. The utility of PyGMTSAR lies in its ability to automate advanced algorithms for InSAR data processing, making the extraction of valuable insights from the data accessible regardless of your technical background. As an analogy, the principles of InSAR are compared to those of common Wi-Fi or 4G/5G cellular connectivity, illustrating PyGMTSAR's intention to make the technology as user-friendly and accessible as possible. A notable feature of PyGMTSAR is its capability to run directly on Google Colab, thereby eliminating the need for local software installations. Finally, this section notes the potential of PyGMTSAR to handle large datasets efficiently, further contributing to its ease of use and power.

1.1. The Basics of Satellite Interferometry

Satellite Interferometry, specifically Interferometric Synthetic Aperture Radar (InSAR), is a remote sensing technique that utilizes two or more synthetic aperture radar (SAR) images to generate maps of surface deformation or digital elevation models of the terrain. A satellite emits a radar signal towards Earth; this signal interacts with the surface and then reflects back.

InSAR includes several specialized techniques, including Differential InSAR (DInSAR) and Time series DInSAR, which involve comparing more than two images over time to analyze surface changes more precisely or to track changes over time.

The "D" in DInSAR stands for "Differential", indicating that it is used to observe phase changes between two images over a given time period. DInSAR is typically employed to monitor subsidence/uplift or lateral deformation. To separate the deformation signal from topographic contributions, the topographic phase is simulated using a reference Digital Elevation Model (DEM) and then subtracted from the interferogram.

Time series InSAR, also known as multi-temporal InSAR, is an extension of DInSAR. It involves the use of multiple SAR images collected over the same area at different time intervals to monitor and measure changes that occur over time, such as land displacement due to seismic, volcanic, or anthropogenic activities.

Instead of using just two images as in traditional DInSAR, time series InSAR involves the generation and analysis of multiple differential interferograms, each created using a different pair of images from the available time series. The main advantage of this approach is that it allows for the separation of the deformation signal from other unwanted effects (like atmospheric disturbances), leading to more accurate displacement maps.

There are several processing techniques used for time series analysis in InSAR, including the Persistent Scatterer Interferometry (PSI) and the Small BAseline Subset (SBAS) methods. These techniques have proven to be highly effective in areas with a high density of stable scatterers and moderate to high temporal decorrelation.

In summary, InSAR is related to distance changes from the satellite and is suitable for DEM generation. For surface changes monitoring, we need to estimate distance changes and this technique is called Differential InSAR (DInSAR). More generic analysis based on multiple DInSAR pairs is called Time series DInSAR.

In this book, we use the most common term InSAR while technically we are discussing Time series Differential InSAR. For the time series analysis, we apply the Small BAseline Subset (SBAS) method. PyGMTSAR allows for some inclusion of Persistent Scatterer Interferometry (PSI) in the SBAS analysis, and this direction looks promising. There is a lot of research being done to merge the best features of SBAS and PSI, and we are in the process of enhancing PyGMTSAR in this regard.

The term 'synthetic aperture' refers to a computational technique that involves continuously transmitting and receiving signals and integrating the waves received from various points on Earth's surface. This approach, although increasing the complexity of image processing compared to instant snapshots, significantly enhances image resolution. A radar image is computational in its essence as it's not captured in a single moment but post-processed from a long-duration measured signal. Available for download, Sentinel-1 SLC scenes are focused synthetic radar images and form the data source for PyGMTSAR computations.

An interferogram, entirely computational, is created from the phase difference of the return signals from two separate yet nearly identical images. To achieve nearly identical images, alignment is critical. The well-documented Sentinel-1 satellite orbit is used to calculate the geomet-

ric correction for a minor spatial difference known as the perpendicular baseline. PyGMTSAR can automatically download the orbit files for selected scenes.

Because of the high number of image pair combinations, it's technically unfeasible to provide all pre-calculated interferograms for download, akin to the Sentinel-1 scenes themselves. PyGMTSAR computes interferograms for selected image pairs using downloaded Sentinel-1 scenes and orbit files. This operation is straightforward, hinging on simple math operations. However, for practical reasons, additional processing, such as Doppler correction using orbit information and topography correction using the Earth's area covered in the Sentinel-1 images, is performed. PyGMTSAR automatically downloads and converts SRTM (Shuttle Radar Topography Mission) topography heights to the WGS84 ellipsoid from standalone EGM96 geoid for SRTM and other topography models.

Gaussian and Werner/Goldstein filters are applied to reduce radiometric noise and highlight changes, enhancing the visibility of interferogram fringes at the expense of spatial resolution. As of now, Gaussian and Werner/Goldstein filters are mandatory, but work is underway to make them optional and allow for persistent scatterers interferometry features. The well-known interferogram fringes result from phase measurement in the interval 2π, often referred to as the wrapped phase.

The SNAPHU (Statistical-Cost, Network-Flow Algorithm for Phase Unwrapping) program is used for 2D unwrapping to produce a continuous phase. PyGMTSAR supports both single-raster and tiled SNAPHU unwrapping using predefined or custom unwrapping configurations. Interestingly, we can visually estimate the changes without unwrapping by counting the so-called fringes (a phase change of 2π mapped as a full colormap range) and locating their centers, which can indicate deformation epicenters. By design, a single fringe signifies a 2π phase change, corresponding to a round-trip (to the ground and back to the satellite)

change in distance equivalent to a single radar signal wavelength, or equivalently, a one-way change of half of this wavelength. In other words, while SNAPHU is a highly useful tool for satellite interferometry, it isn't essential.

By using the continuous unwrapped phase, we can gauge the distance change between the satellite and each point on the ground. This change is partially related to surface deformation, but it also depends on atmospheric conditions and other factors. Fundamentally, InSAR measures Earth's surface movements with remarkable precision, often down to a few millimeters or even better over a large series of interferograms.

Knowing the unwrapped phase in radians, it's easy to calculate LOS (Line-of-sight) displacement in millimeters and its vertical and east-west projections using the scenes' geometry. For better accuracy, PyGMTSAR projection transformations are based on a complete grid of the incidence angle computed for every ground pixel. The sum of the projections obtained for two orbits (ascending and descending) are the real displacement components. The displacements are related to the Sentinel-1 image capture date intervals, not to specific dates.

Small BAseline Subset (SBAS) displacements calculation applies a correlation-weighted least-squares algorithm to the interferogram displacements to produce continuous displacement timeseries for every pixel. More interferograms mean more displacements on intersected intervals and potentially better result accuracy. The SBAS technique simplifies atmospheric phase exclusion as atmospheric phase delays are included with different signs in different interferograms with common cloudy images.

The SBAS displacement often presents substantial noise, making it difficult to discern real-world changes. To overcome this, signal processing can be used for pixel-wise analysis to extract trends and seasonal movements from the time series. PyGMTSAR uses Seasonal-Trend

Decomposition using LOESS (STL) for this purpose. Trend extraction applies the power of mathematical statistics for noise reduction. A seasonal trend tends to be vertical and repetitive by definition; thus, we can estimate it more accurately than merely relying on SBAS time series. The non-seasonal trend, averaged over the duration of the time series, yields the average velocity. These extracted features reflect real-world characteristics that can be validated by ground measurements, providing valuable insights for surface monitoring.

And still, that's not the end of the interferometric processing journey. The produced results need to be geocoded from scene radar coordinates to geographic coordinates and exported in a common format. PyGMTSAR processing is based on NetCDF (Network Common Data Form) data files which can be opened directly in open-source QGIS, GDAL, and other GIS (geographic information system) software. Also, PyGMTSAR allows exporting 3D rasters in VTK format for interactive 3D visualization and analysis in open-source ParaView software.

1.2. Introduction to PyGMTSAR

Numerous InSAR software applications have been developed by various experts, from seasoned geophysicists to specialized software development companies, and even driven amateurs. However, PyGMTSAR carves out a unique position among them. It's the creation of a fundamental radio physics scientist with extensive experience in computer science and applied programming. PyGMTSAR represents a significant investment of time and effort into the development of mathematical models of interferometry and holography, the construction of hardware to test these models in lab environments, and the education of students, postgraduates, and researchers in these concepts. This deep-rooted understanding and proficiency in the core principles and numerical calculation algorithms distinctively set PyGMTSAR apart.

The developer of PyGMTSAR isn't just a programmer or a geophysicist, but a well-rounded researcher who has worked on every aspect of interferometry, from the initial theoretical modeling to the processing of collected data. This comprehensive perspective, coupled with a commitment to usability, has culminated in a tool that combines cutting-edge scientific accuracy with user-friendly design.

PyGMTSAR simplifies the complex realm of satellite interferometry, making it accessible to everyone, regardless of technical background or expertise. It stands on the belief that advanced scientific tools should be universally available to anyone interested in understanding and harnessing the power of satellite data. The mission of PyGMTSAR is clear — democratize access to advanced remote sensing technologies and inspire a new generation of users and innovators.

The handling of data in the ever-evolving field of satellite remote sensing can become a complicated task. PyGMTSAR rises to the occasion by streamlining the process, making it more manageable and accessible. It

simplifies the most advanced algorithms for InSAR data processing for all users, regardless of their technical expertise.

While understanding InSAR can be challenging due to its computational intensity and complex principles, such as synthetic aperture and interferogram, these principles share similarities with those used in everyday technologies like Wi-Fi or 4G/5G cellular connectivity. The key is to encapsulate the complexities of InSAR into a user-friendly tool that doesn't require the user to fully comprehend the technical specifics. This is the ethos behind PyGMTSAR, aiming to be as accessible and straightforward to use as your cellphone or wireless modem.

One of PyGMTSAR's key features is its ability to run directly on Google Colab with a single click. This eliminates the need for local software installations or complicated procedures to generate results. Users can access live, annotated examples directly in their web browsers, allowing them to interactively change source datasets and processing parameters to observe different outcomes - an effective learning tool for understanding InSAR data processing.

The subsequent chapters will provide more specifics on effectively utilizing PyGMTSAR, offering a curated set of examples illustrating how to extract valuable insights from InSAR data. These examples are not only available on Google Colab but also in a Docker image, making them easily accessible to anyone keen on learning.

For those who prefer a more traditional approach, PyGMTSAR can be installed directly on a computer or used on cloud hosts, like Amazon or Google Cloud computing instances. Although this book doesn't cover this procedure, it's well-documented in the PyGMTSAR documentation.

Despite its simplicity and ease of use, PyGMTSAR is incredibly powerful and can handle large datasets with ease. Whether processing thousands of interferograms for multiple Sentinel-1 scenes stitched together at high resolution, or producing hundreds of multi-gigabyte displacement

rasters on common hardware like an Apple Air laptop, PyGMTSAR maintains an impressive performance level.

PyGMTSAR bridges the gap between complex satellite data and user accessibility in the realm of satellite remote sensing and InSAR data processing. With a development approach centered around simplifying technical complexities, PyGMTSAR emerges as an invaluable resource for anyone seeking to navigate the world of InSAR.

2. Getting Started

In this chapter, you'll discover the first steps to using PyGMTSAR - a software package for satellite interferometry processing. The chapter is divided into two sections, focusing on how to set up and start working with PyGMTSAR in two environments: Google Colab and Docker Desktop.

Section 2.1 introduces Google Colab, a free cloud service, as a convenient way to run PyGMTSAR examples directly in your web browser. Google Colab, although not suitable for all workloads, is ideal for lighter tasks and provides an excellent starting point for beginners. Here, you'll find various example notebooks highlighting the application of PyGMTSAR in different areas like seismology, volcanology, hydrology, and infrastructure monitoring. The instructions provided in these notebooks will guide you on how to adjust the radar scenes and processing parameters to match your specific needs.

Section 2.2 shifts the focus to Docker Desktop, an open-source platform that allows you to run applications in isolated containers. This approach is more suited for those wanting to work offline or locally, or those whose workloads demand more computational power than what Google Colab can provide. Here, you will learn how to download and install Docker Desktop, create a DockerHub account, configure Docker Desktop, download the PyGMTSAR Docker image, and finally run the image using Docker Desktop. This provides an interactive workspace where you can access and use the PyGMTSAR examples in an environment closer to a traditional development setup.

2.1. Launching Online with Google Colab

Google Colaboratory, or Google Colab, is a free cloud service that allows you to develop and execute code in Python, directly in your browser. It is similar to Jupyter Notebook, and it offers the added benefits of leveraging Google's cloud computing infrastructure.

Google Colab provides a host of advantages:

- **No setup required**: Google Colab is ready to use immediately. PyGMTSAR example notebooks include all the commands to initialize the environment and perform the processing.
- **Free access to GPUs and TPUs**: Google Colab provides free access to powerful graphics processing units (GPUs) and tensor processing units (TPUs), which are useful for training machine learning models. For now, PyGMTSAR does not use GPUs and TPUs, but these might be helpful to speed up your own code.
- **Shareability**: Notebooks on Google Colab can be easily shared, allowing for seamless collaboration among teams. This feature can be beneficial for researchers and developers working in groups. Open the example PyGMTSAR notebooks and make your changes and share the updated notebook with everyone or just for selected users.
- **Integration with Google Drive and GitHub**: Google Colab integrates smoothly with Google Drive and GitHub, making it easy to load data, notebooks, and store your work. You can load Sentinel-1 scenes stored on your Google Drive and save the results.
- **Interactive tutorials and documentation**: Google Colab notebooks can contain live code, equations, visualizations, and narrative text, making it a great tool for creating interactive tutorials and documentation. PyGMTSAR uses all the features to provide the example notebooks as the complete interactive learning tutorials.

Given these advantages, Google Colab is an excellent platform to run the PyGMTSAR examples, which provide a comprehensive, self-explained InSAR pipeline right in your web browser, producing detailed graphs and maps. You can use these examples as templates for your tasks, simply by replacing the used radar scenes and altering the processing parameters to meet your specific needs.

To access these notebooks directly in your browser, follow the provided links below. The links are sorted into thematic groups, and the complete list of examples is also available on the PyGMTSAR GitHub page.

Remember, while Google Colab is an excellent resource, it's crucial to note that it's not suitable for all use cases because of its limitations in terms of available RAM and processing time. For heavier workloads, you might need to consider more powerful hardware or cloud solutions.

Seismology

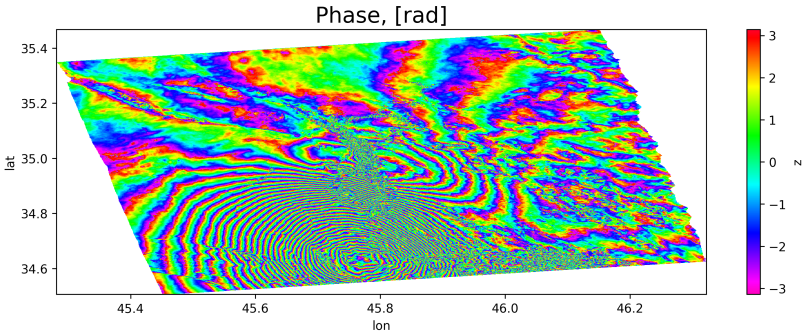
InSAR plays a vital role in monitoring and studying seismic activities. It can detect ground deformation before, during, and after earthquakes. This technology provides valuable data on fault systems and magma movements, enhancing our understanding of these natural phenomena. The following examples show the application of InSAR data in studying and interpreting seismic events.

This notebook analyses the 2017 Iran–Iraq Earthquake, with a comparison to results from other software tools such as GMTSAR, GAMMA, and SNAP.

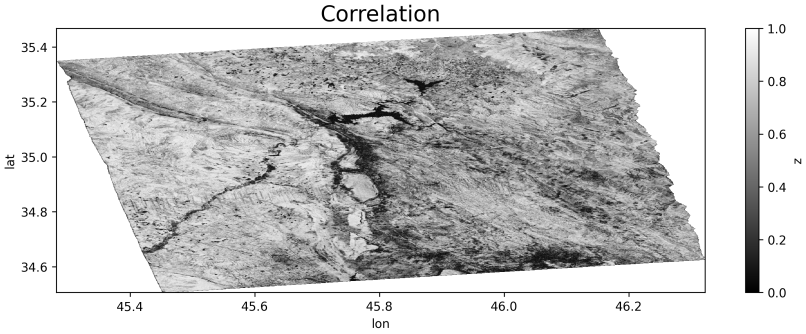
In this analysis, the notebook fetches Sentinel-1 scenes from the Alaska Satellite Facility (ASF). This facilitates the generation of an interferogram and coherence map for a single cropped subswath. It exports NetCDF rasters, which are compatible with QGIS, GDAL, and other GIS software. A comparative review of the results with GMTSAR, SNAP, and GAMMA software is also provided. Note: To generate an

interferogram and Line-Of-Sight (LOS) displacement for your area of interest, simply replace the scene names.

The first image illustrates an interferogram of the 2017 Iran–Iraq Earthquake, a seismic event captured using InSAR data. The fringes visible in the image represent different amounts of ground movement that occurred during the event. Each cycle from one color back to the same color represents the half wavelength of displacement in Line-Of-Sight (LOS) direction. Well-defined fringes are indicative of significant ground deformation associated with the earthquake.



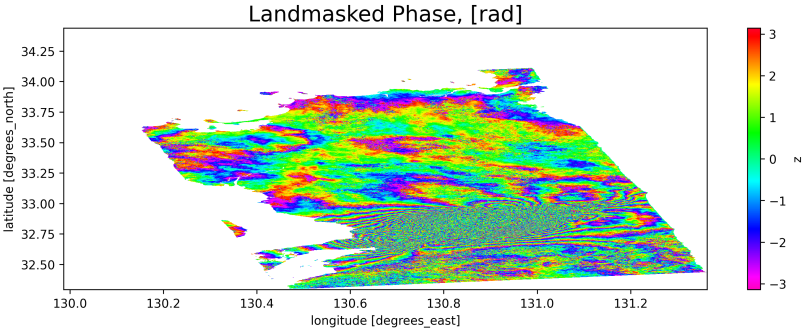
The second image depicts a coherence map, a measure of the quality of the interferometric phase data. High coherence values indicate areas where the phase difference is reliable, and this typically aligns with regions of stable scatterers, like rocks or bare ground. Areas of low coherence, on the other hand, might represent regions where the radar signal was disrupted, perhaps due to vegetation, surface changes, or atmospheric effects.



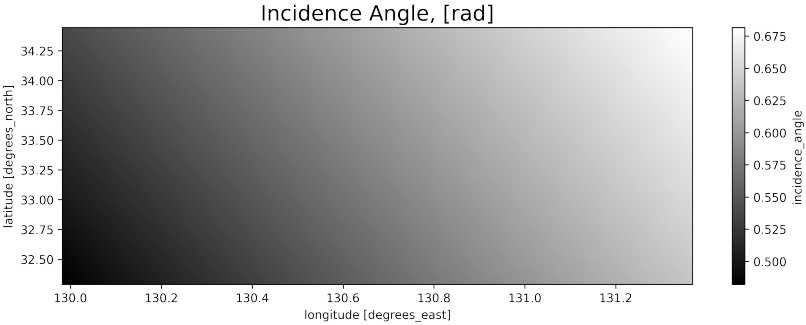
Open in Colab This notebook focuses on the 2016 Kumamoto Earthquake. It demonstrates the processing of InSAR data to create a co-seismic interferogram, which highlights earth's deformation because of the seismic event. The findings are contrasted with results from the ESA Sentinel 1 Toolbox on the Alaska Satellite Facility.

The example shows the processing of a single subswath with a land mask applied to the interferogram, unwrapped phase, and Line-Of-Sight (LOS), as well as east-west and vertical displacement results.

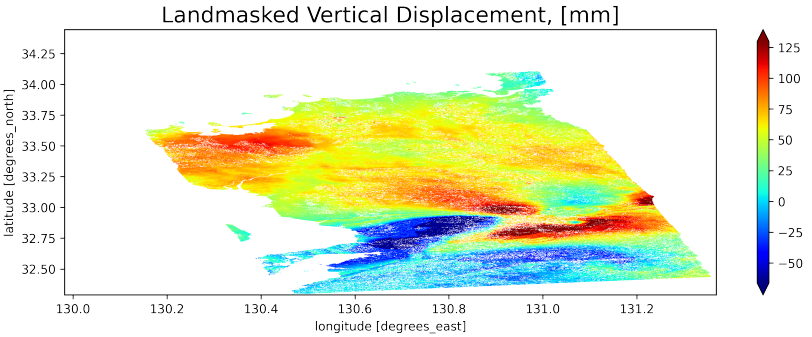
The first image shows the interferogram for the 2016 Kumamoto Earthquake, with a land mask applied. Water bodies are removed from the analysis because they produce a random, or "noisy," phase result in the interferogram due to their dynamic nature. The visible fringes in the image, especially those that are challenging to interpret, highlight the complex ground movements that occurred during the seismic event.



The second image displays an incidence angle map which corresponds to the angle at which the satellite's radar signal hits the ground. This information is crucial for converting the Line-Of-Sight (LOS) displacement to horizontal and vertical displacement components.



The third image illustrates the estimated vertical displacement for the 2016 Kumamoto Earthquake. The applied land mask removes the noisy offshore values.



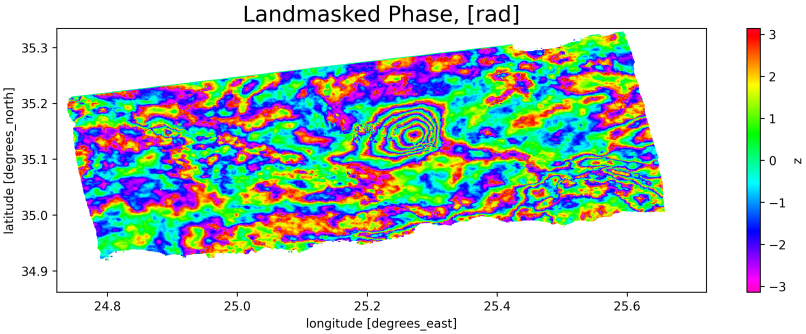
Open in Colab The 2021 Crete Earthquake Co-Seismic Interferogram notebook showcases the application of InSAR in processing data from a recent seismic event to understand the associated ground deformations. The results are contrasted with a report from the Centre of EO Research & Satellite Remote Sensing in Greece.

This example illustrates the processing of a single, cropped subswath with a land mask applied to the interferogram, unwrapped phase, and Line-Of-Sight (LOS), east-west, and vertical displacement results.

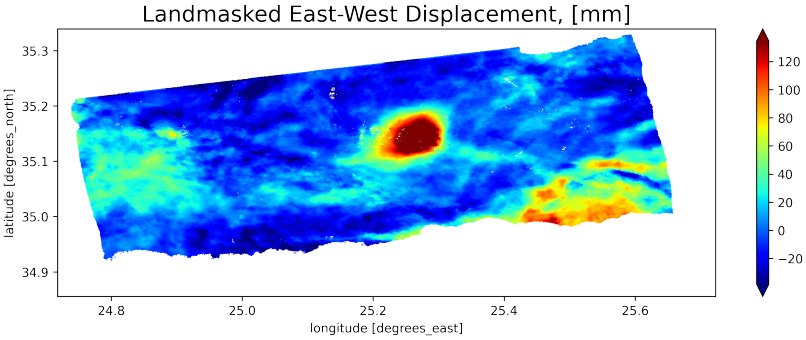
The first image showcases an interferogram of the 2021 Crete Earthquake. The fringes represent phase changes between the before and after images of the event, which correspond to the ground deformations caused by the earthquake. Each fringe cycle corresponds to a half-wavelength change in the distance between the satellite and the ground surface, translating to a relative ground movement.

The land mask applied to this image enhances the visibility of these fringes by eliminating the noise produced by the ocean. Offshore areas tend to exhibit more noise due to water motion and low radar reflectivity.

The epicenter of the earthquake can be visually pinpointed from the concentric fringes originating from a common center. The varying fringe patterns surrounding the epicenter indicate the diverse ground displacements caused by the seismic activity.



The second image presents the east-west component of the displacement caused by the earthquake, computed from the incidence angle map and the unwrapped phase of the interferogram. Again, a land mask is applied to remove offshore noise.



Open in Colab The 2023-02-06 Türkiye Earthquake Co-Seismic Interferogram notebook demonstrates the application of InSAR to process data from one of the most catastrophic seismic events in recent history. On 6 February 2023, a magnitude 7.8 earthquake impacted southern and central Turkey and northern and western Syria, causing widespread destruction.

The example makes use of InSAR techniques to process Sentinel-1 Scenes, obtained from the Alaska Satellite Facility (ASF). This processing involves stitching 3 scenes, merging subswaths, detrending phase, and

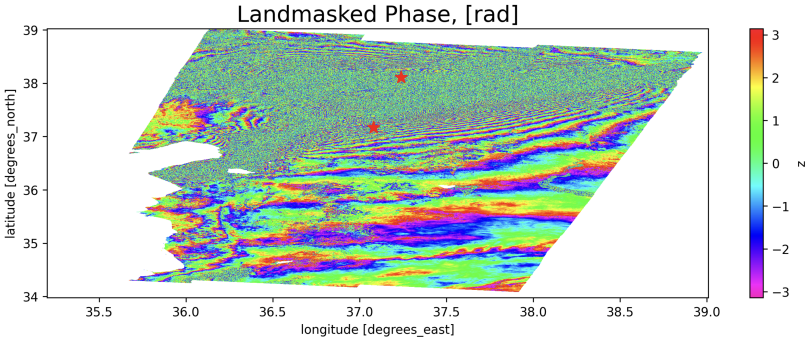
performing lazy exporting of NetCDF rasters (compatible with QGIS, GDAL, and other GIS software).

This case illustrates some tricks to process a large volume of data on Google Colab effectively. Note that you can replace the scene names in this notebook to produce an interferogram for your specific area of interest.

The first image displays an interferogram derived from the Sentinel-1 scenes of the 2023-02-06 Türkiye Earthquake. The colorful fringes represent phase changes between the pre- and post-seismic event, illustrating the ground deformations caused by the earthquake.

This interferogram covers a large area, which corresponds to the substantial extent of the earthquake's impact. The heterogeneous pattern of the fringes might suggest various types of ground movements such as landslides and surface ruptures associated with the seismic event. These different movements can be further studied for a comprehensive understanding of the event.

A land mask is applied to this interferogram to eliminate noise, particularly in the lower left corner where a water body is located.

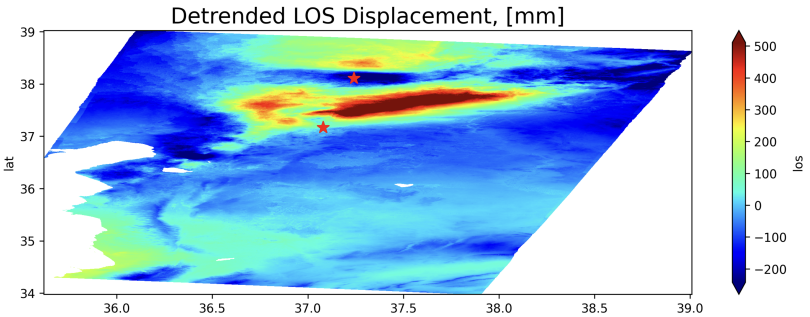


The second image showcases the Line-Of-Sight (LOS) displacement map of the same event, after detrending the phase. The detrending process

aims to remove the linear ramp present in the original phase map, which can distort the final displacement results.

Despite the noisiness of the original interferogram, the phase unwrapping tool SNAPHU manages to unwrap the phase, providing a continuous displacement pattern. The result may have some inaccuracies due to unrecognized fringes, but the overall displacement pattern gives a reasonable approximation of the ground movements during the earthquake.

The colors in the map represent the magnitude and direction of displacement along the satellite's line of sight, providing a clearer visualization of the ground deformations caused by the earthquake.



Volcanology

InSAR is a useful tool in volcanology. It uses the phase differences in radar waves bounced back from the Earth's surface to construct detailed maps of surface deformation with high spatial resolution and precision. This technology is especially valuable for monitoring volcanoes, as it can cover sizeable areas and work in all weather conditions, day or night.

The ability of InSAR to detect minute changes in the Earth's surface (down to a fraction of the wavelength of the radar wave, typically a few centimeters) makes it particularly effective for monitoring volcanic activity. Before an eruption, magma rising towards the surface can

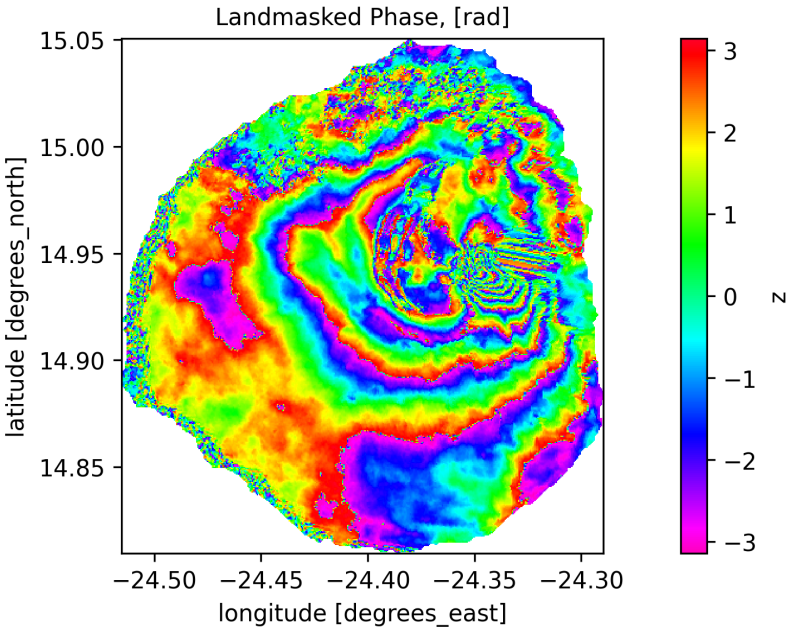
cause the ground to swell. This inflation can be picked up by InSAR, providing a warning sign of a possible impending eruption.

Similarly, after an eruption, the ground can deflate as magma chambers are empty. Again, InSAR can detect these changes, providing valuable data on the volume of magma involved and the mechanics of the eruption. This kind of information can help scientists better understand a volcano's behavior and improve eruption forecasting.

This notebook explores the eruption of the Pico do Fogo volcano on Fogo Island in Cape Verde, which occurred on November 23, 2014. The study uses InSAR data to investigate the geophysical changes related to this volcanic event, offering insights into the eruption dynamics and the subsequent impacts on the local environment.

In this example, a single cropped subswath is processed with a land mask applied to the interferogram. The output includes the interferogram, unwrapped phase, and Line-Of-Sight (LOS), east-west, and vertical displacement maps. These results give an overview of the earth's deformation due to the eruption.

The image is a phase map showing the deformation of the Pico do Fogo volcano due to an eruption. The land mask filters out the water bodies. The fringes indicate surface deformation due to the volcanic activity and the eruption. In this case, it seems the whole island experienced some level of deformation due to the eruption, which would have been caused by the movement of magma beneath the surface.



Hydrology

InSAR is a powerful tool that can monitor changes in water levels and subsurface movements. This technology has been successfully used in various hydrological studies, including tracking the rate of groundwater extraction, mapping wetland water levels, and observing glacier movements.

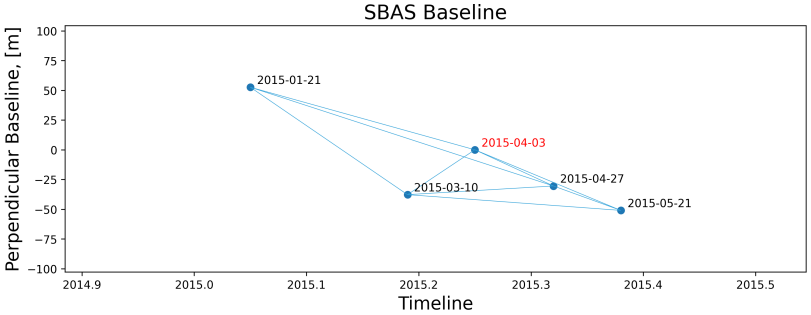
Open in Colab This notebook guides you through the entire process of using PyGMTSAR to analyze changes in water levels in the Imperial Valley using Small BAseline Subset (SBAS).

In this example, you will learn how to implement the SBAS approach, a powerful InSAR technique specifically designed to detect slow ground deformation over time. The SBAS approach involves selecting a subset of interferograms with small temporal and spatial baselines, which

effectively reduces the impact of decorrelation and atmospheric delays on the results.

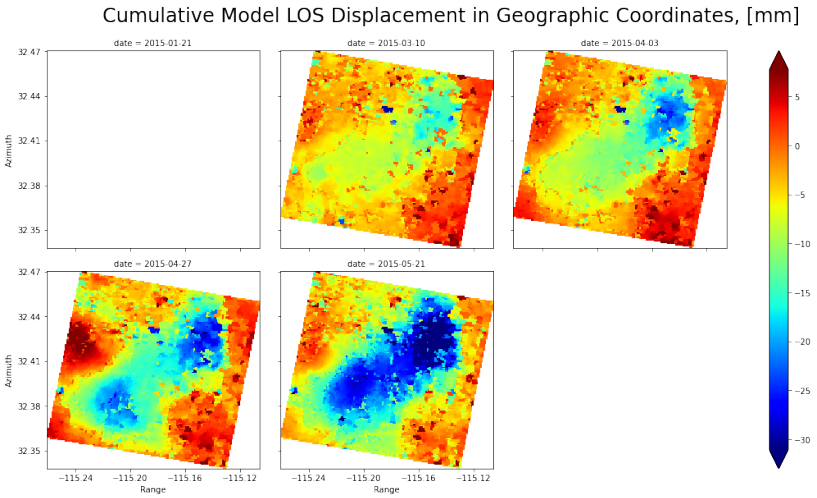
And the example introduces you to the detrending technique used in PyGMTSAR to remove atmospheric noise, thus significantly improving the quality of the resulting interferograms and making the observed water level changes more accurate and reliable.

The first image is a baseline chart for the SBAS InSAR technique. It shows all the interferograms that were used in the analysis, arranged by their acquisition dates (X-axis) and the perpendicular baseline (Y-axis). Each point on the plot represents an individual Sentinel-1 scene, and each edge represents an interferogram.



The second image illustrates the cumulative Line of Sight (LOS) displacements over time in the Imperial Valley. This map of displacement provides a clear visualization of the relative movement of the ground surface due to changes in water levels.

In regions where groundwater is extracted, we often observe a seasonal pattern in subsidence and uplift - subsidence typically occurs during the dry months when more water is pumped out, and uplift occurs during the wet months when aquifers are replenished. Hence, the map not only provides valuable information on the spatial distribution of subsidence/uplift but also potentially reveals insights about the region's groundwater dynamics.



Infrastructure Monitoring

InSAR is used to monitor infrastructure like dams, bridges, and buildings. It can detect subtle movements and deformations that might suggest potential structural problems or failures.

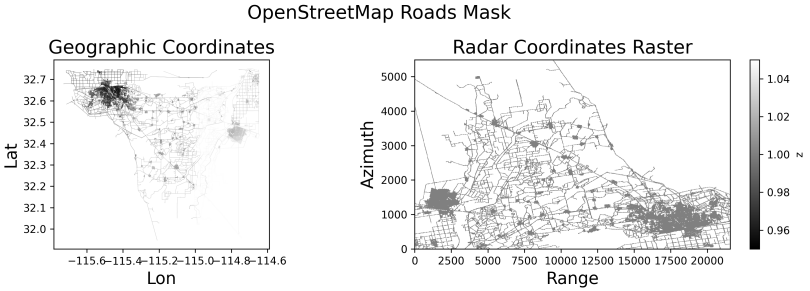
For now, PyGMTSAR applies Goldstein and Gaussian filters to the processed interferograms, making the results more smooth and highlighting fridges. That makes the outputs more suitable for the kinds of analysis explained above. Be careful with spatial accuracy for small infrastructure objects monitoring. When monitoring infrastructure, accurate identification and assessment of subtle changes can mean the difference between timely intervention and significant structural failure.

Open in Colab This notebook presents an analysis of OpenStreetMap's road infrastructure to monitor road subsidence. That's just an example and the most subsidence are related to seasonal water level changes for the area.

The first image portrays a vector and a rasterized representations of road infrastructure derived from OpenStreetMap (OSM) data. These roads

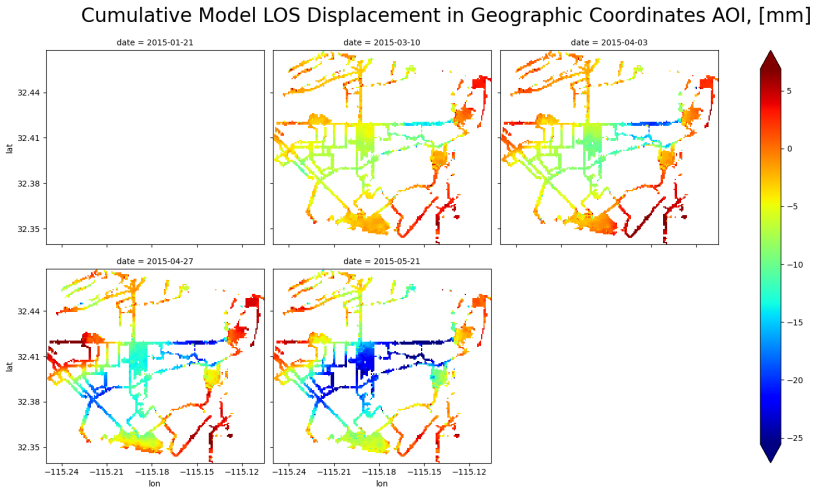
are converted from their original geographic coordinates into radar coordinates, to allow their alignment with the radar-based interferometric data.

Such a road mask can be crucial in studies looking to identify ground movement affecting infrastructure, such as roads. It enables the extraction of InSAR data specific to these regions, allowing for detailed analysis of any detected ground deformation.



The second image presents a cumulative Line-Of-Sight (LOS) displacement map. Here, the displacements observed across multiple radar acquisitions are summed to illustrate the total changes over time. These shifts could be due to a variety of factors, with seasonal fluctuations in water levels being a common cause in this region.

The roads, previously identified and masked from the OSM data, are included on this map. This enables an investigation into how changes in ground movement, possibly due to fluctuating water levels, impact the stability and integrity of road infrastructure. With a long enough time series of data, this analysis can detect not only common seasonal changes but also trends in road movement that may indicate potential issues or the overall health state of the infrastructure.


Cumulative Model LOS Displacement in Geographic Coordinates AOI, [mm]
date = 2015-01-21
date = 2015-03-10
date = 2015-04-03
date = 2015-04-27
date = 2015-05-21
lat
lon
32.44
32.41
32.38
32.35
−115.24
−115.21
−115.18
−115.15
−115.12
5
0
−5
−10
−15
−20
−25

2.2. Running Locally with Docker Desktop

Docker is an open source platform that uses OS-level virtualization to deliver software in packages called containers. A Docker container is a standalone, executable package that includes everything needed to run an application: the code, a runtime, libraries, environment variables, and config files.

If you prefer to work offline or locally, or if your project necessitates a more powerful setup than what's available on Google Colab, you can make use of Docker. The author of PyGMTSAR has prepared a Docker image with the same examples as those on Google Colab. This can be found on the DockerHub page.

Here's why consider Docker:

- **Isolation**: You can avoid the common software issue where an application works on one machine but not on another because of differences in their environment. Containers with different PyGMTSAR versions can be run on the same host even simultaneously and can be transferred between different computers and operation systems. Define exactly available processor cores, memory, and disk space.
- **Consistency**: Docker ensures consistency across multiple computers and operation systems. The ready-to-use PyGMTSAR images include reproducible examples which work equally on any host. Use the stable and well tested PyGMTSAR images and containers or make your own ones and do not worry about portability.
- **Control**: Docker provides control over your environments and allows configurations to be versioned and reused. Operate the stable and well tested PyGMTSAR images, which can be rebuilt and reused locally or from DockerHub.

To get PyGMTSAR running on your computer, you will need to download the PyGMTSAR Docker image and run it using Docker Desktop. Docker Desktop is a user-friendly and open source interface for Docker, providing a GUI and additional features for managing containers. If you do not have Docker Desktop already installed, the following steps will guide you through the installation process. Note, however, that you could use the lower-level Docker tool as an alternative to Docker Desktop, though it may be more challenging to navigate.

Installing Docker Desktop

First, you need to download and install Docker Desktop on your operating system. Just follow these simple guides:

- For Windows users: Install Docker Desktop on Windows,
- For Linux users: Install Docker Desktop on Linux,
- For Mac users: Install Docker Desktop on Mac.

How to Register on DockerHub

Before you can explore the PyGMTSAR collection of container images on DockerHub, you'll need to set up an account. DockerHub is a cloud-based platform where Docker images can be found, shared, and managed. It acts as a connection point between your local Docker installation and Docker's public repository.

Docker Desktop offers various methods to download prepackaged software images. You can use console commands provided on DockerHub software pages if you're an advanced user, or use the Docker Desktop's user-friendly interface.

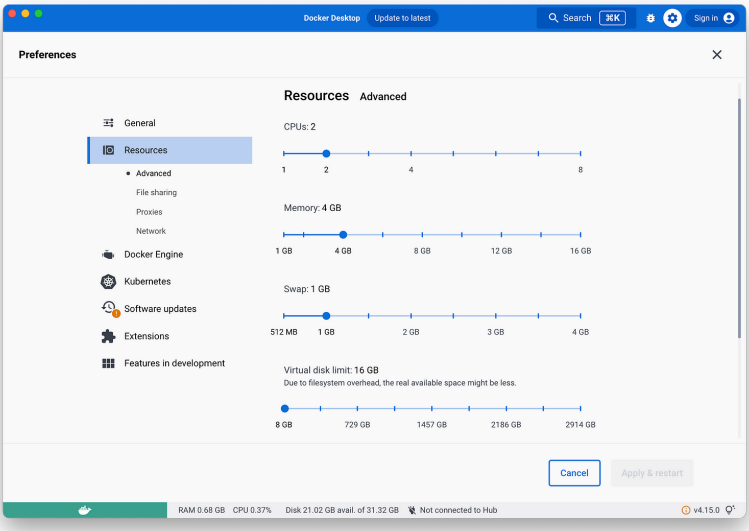
To register on DockerHub through Docker Desktop, just click "Sign in" in the top right corner of the application window. This action will direct you to a sign-in page where you can create a new account.

If you'd rather register directly on a web browser, follow this link: Create a Docker Account. Keep your Docker ID and password handy; you'll need them to download and manage Docker images.

Having a DockerHub account enables you to download and use any public images, create and manage your own Docker images, and even share them with others.

How to Configure Docker Desktop

To access Docker Desktop settings, click on the gear icon at the top right corner and navigate to the "Resources" section:

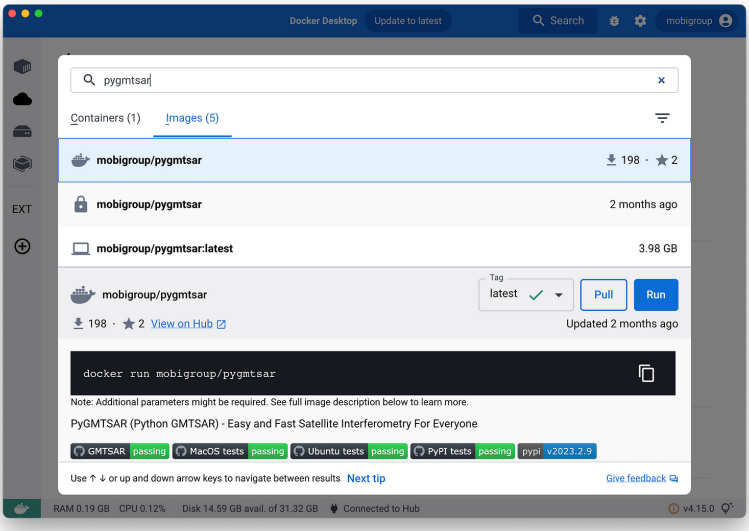


If you plan to run examples from the mobigroup/pygmtsar image, you'll need to allocate 2 CPU cores, 4 GB RAM, 1 GB swap, and a virtual disk of 16+ GB. If you intend to execute all the notebooks sequentially, set the virtual disk limit to 200+ GB. Remember, satellite interferometry processing can take up significant storage!

If you aim to launch the larger mobigroup/pygmtsar-large image, your virtual system should have 4 CPU cores, 16 GB RAM, 1 GB swap, and a 500 GB virtual disk. This configuration allows you to produce 34 interferograms, each of 3 GB size, along with their corresponding correlation files. It also enables detrending and SBAS analysis. Even a task this heavy can be performed on an Apple Silicon iMac or Air with 16 GB RAM, as Docker Desktop can utilize all available RAM.

How to Download the PyGMTSAR Docker Image

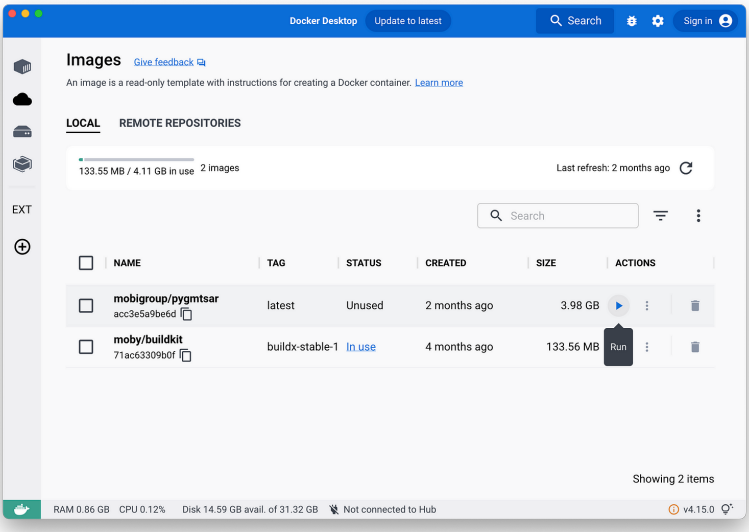
Once you've registered and logged in, look for the "pygmtsar" in the search box located in the "Images" tab:



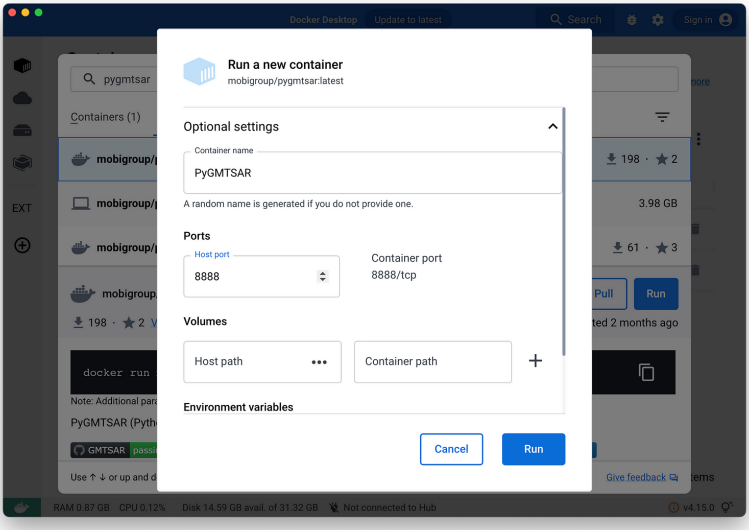
Two options will appear: mobigroup/pygmtsar is a standard 1 GB PyGMTSAR image that meets the requirements of most users, and mobigroup/pygmtsar-large is a more advanced 50 GB image designed for expert users. Choose the one that fits your needs and click on the "Pull" button to download it onto your computer.

Steps to Run the PyGMTSAR Docker Image

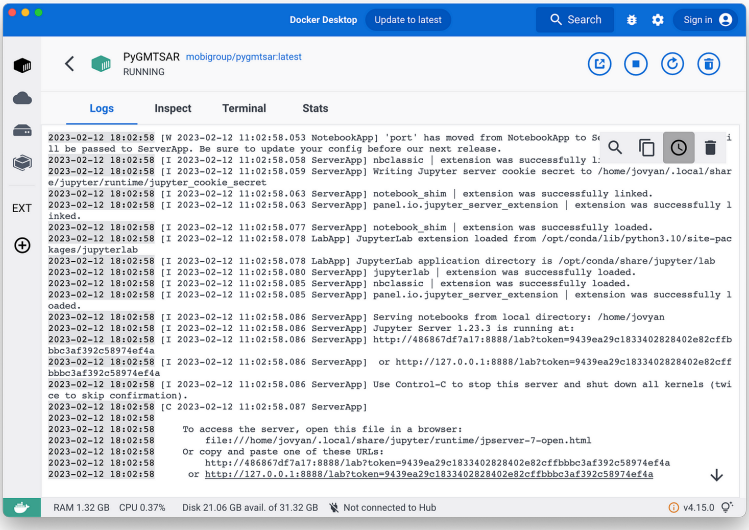
In Docker Desktop, the “Images” section contains all the software images you’ve downloaded. You can manage these images from here. To run the PyGMTSAR image you downloaded, click the triangular button, as shown in the image below.



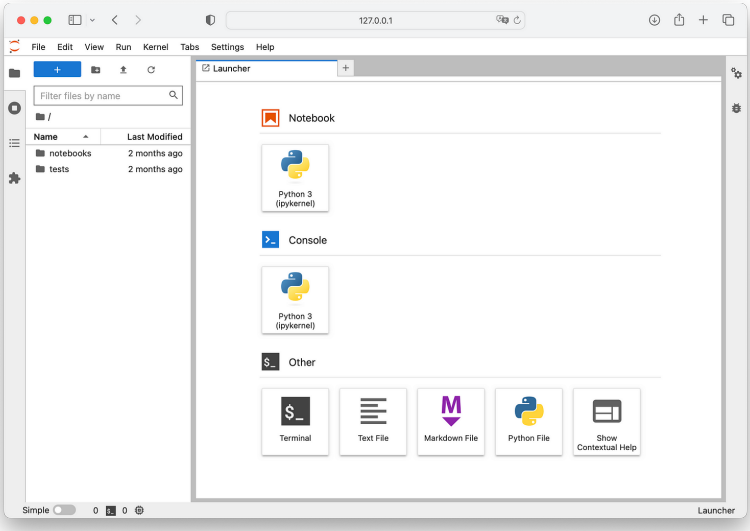
PyGMTSAR will connect with you through a webpage in your browser, utilizing network port 8888. Set it as shown in the image below and, if you want, give it a name, such as “PyGMTSAR”:



After setting it up, click the “Run” button in the form above to start the software:



Next, click on the bottom link in the “Logs” page to open a new webpage and begin working:

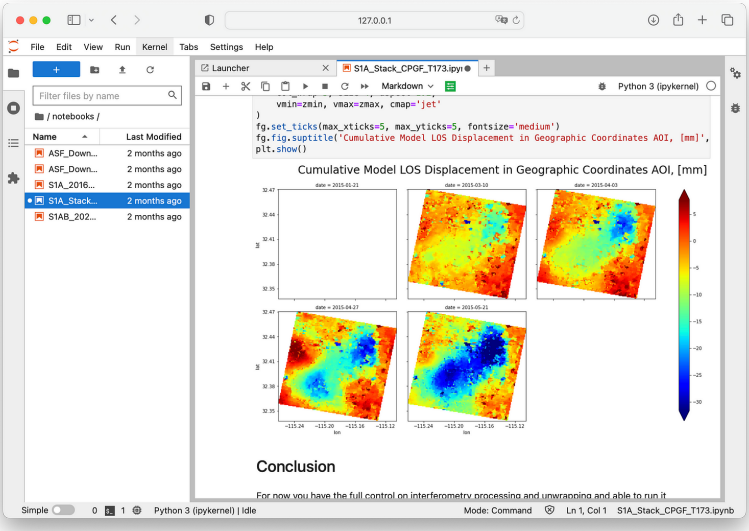


The interface presents an interactive workspace: the left panel shows all the files and directories in your Docker image, while the right panel acts as an interactive editor, viewer, and console, among other functions. There are two key directories, “notebooks” and “tests”.

“notebooks” contains interactive Jupyter notebook examples that guide you through various PyGMTSAR processes, allowing you to interact with the data and results in real time.

In contrast, “tests” contains Python script examples for those who prefer using console scripts. These scripts work in batch mode and produce the same visual outputs as the interactive ones. Importantly, these scripts are used for continuous integration testing in the PyGMTSAR GitHub repository. You can access the outputs of these tests on the PyGMTSAR GitHub Actions page.

To get started, click on an example in the left panel. You can navigate through the example by scrolling, reading the instructions, and viewing the maps:



To run the whole example, select “Kernel” -> “Restart Kernel and Run All Cells…” from the toolbar menu (Note: the exact menu item may vary depending on the image version).

This interface allows you to monitor the execution steps, pause and resume processing as needed, alter parameters, and view the resulting outputs. It provides the same functionality as the Google Colab live online examples, but without third-party timeouts and processing limits. This makes Docker Desktop a robust tool for comprehensive analysis and research.

3. Exploring PyGMTSAR

Chapter 3 unveils the principles and functionality of PyGMTSAR, a powerful tool designed for efficient processing of interferometric synthetic aperture radar (InSAR) data. This section serves as a comprehensive guide for using this software, shedding light on its operational aspects without delving too deep into the intricacies of its core algorithms.

In Section 3.1, we present the operational facets and the architecture of PyGMTSAR. This Python library is versatile, designed to operate across a variety of hardware, from standard laptops to high-performance workstations. We explore how PyGMTSAR streamlines processing tasks, making execution as simple as a single click, even in the absence of local software installation. This feature is powered by its compatibility with multiple environments, including interactive Jupyter notebooks on Google Colab, Docker containers, and GitHub Action runners. Underpinning this scalability is Dask, a distributed process manager, which optimizes the execution of PyGMTSAR operations. Dask segments large operations into manageable tasks and executes them in an optimal order, efficiently leveraging all available resources.

In Section 3.2, we delve into the principle of lazy and delayed numeric computations that PyGMTSAR implements. This strategy allows operations to be deferred, enabling quick access to input and output rasters. We provide a detailed explanation of how to initialize the Dask parallel and distributed scheduler, a critical component for managing parallel processing tasks. Additionally, we cover on-the-fly transformations, such as geocoding, and discuss the trade-off between memory consumption and processing speed. To monitor and understand the software's internal parallel processing, we introduce the Dask Dashboard and the PyGMTSAR progress indicator. These tools offer valuable insights into your code's performance, helping identify potential bottlenecks or areas for improvement and ensuring that computations are efficient and resource-effective.

In Section 3.3, the critical role of the core SBAS object that PyGMTSAR manipulates during processing is highlighted. This object is central to PyGMTSAR's functionality, acting as the primary data structure with which the software interacts. The SBAS object serves a dual purpose: it stores essential data and enables effective management and processing of this information.

In Section 3.4, we explore the internal operations on NetCDF grids, focusing on PyGMTSAR's capabilities to read and write in NetCDF and GeoTIFF, and to export data into VTK files. Given that InSAR time-series analysis deals with large datasets and generates equally large outputs, a deep understanding of PyGMTSAR's data operation features becomes crucial for efficient data processing.

Section 3.5 focuses on the essential steps of InSAR processing that PyGMTSAR executes. From data acquisition to reframing, image co-registration, interferogram formation, geocoding, phase unwrapping, phase detrending, displacement map creation, displacement projection, time-series analysis, trend analysis, and data exporting, each step is thoroughly explored. This section also introduces the machine learning algorithms integrated into PyGMTSAR that simplify these processes for users. Leveraging advanced techniques, PyGMTSAR provides not just an efficient, but also a robust and high-quality InSAR processing pipeline.

The principles and functioning of PyGMTSAR covered in this chapter form a robust foundation for understanding the examples provided in Google Colab and in Docker images. By following these principles and keeping the operation of delayed computation in mind, you can effectively use all the InSAR processing steps.

3.1. Understanding PyGMTSAR

The complete e-book “PyGMTSAR: Sentinel-1 Python InSAR: An Introduction” is available worldwide on Amazon and as a PDF file on Patreon for sponsors.

3.2. Lazy and Delayed Computations

The complete e-book “PyGMTSAR: Sentinel-1 Python InSAR: An Introduction” is available worldwide on Amazon and as a PDF file on Patreon for sponsors.

3.3. The Primary SBAS Object

The complete e-book “PyGMTSAR: Sentinel-1 Python InSAR: An Introduction” is available worldwide on Amazon and as a PDF file on Patreon for sponsors.

3.4. Data Reading and Writing

The complete e-book “PyGMTSAR: Sentinel-1 Python InSAR: An Introduction” is available worldwide on Amazon and as a PDF file on Patreon for sponsors.

3.5. InSAR Workflow Steps

The complete e-book “PyGMTSAR: Sentinel-1 Python InSAR: An Introduction” is available worldwide on Amazon and as a PDF file on Patreon for sponsors.

Books in the PyGMTSAR Tutorial Series

1. **An Introduction**: The book provides an in-depth introduction to the interactive Jupyter notebook examples of PyGMTSAR, available on both Google Colab and Docker images. It covers software architecture, the concept of lazy and delayed computations, outlines the InSAR processing and the related workflow steps available in PyGMTSAR. [**Published**]

2. **Functions Reference**: This reference presents a catalog of all user-level PyGMTSAR functions. Each function is organized by functionality and comes with arguments, explanations, and usage examples for better understanding. [**In Progress**]

3. **An Interferogram**: The handbook guides readers through the steps to produce a single interferogram, its continuous phase and displacement maps. It shows how to compute and plot various types of maps and how to export them in different formats for further analysis and visualization. [**Planned**]

4. **Time-Series Analysis**: The manual covers Persistent Scatterers (PS), Small Baseline Subset (SBAS) time-series, and Seasonal-Trend Decomposition using LOESS (STL) analyses within PyGMTSAR. It outlines constructing and analyzing a series of interferograms, generating displacement time series, and offers guidance on exporting the output velocities and trends in various formats. [**Planned**]

5. **Mastering**: This guide details PyGMTSAR internals, offering theoretical knowledge and practical tips for mastering InSAR principles. It uncovers the underlying patterns of core InSAR algorithms and their implementations, aiding you in extracting valuable information even from the most challenging projects. It serves as your bridge between academic sciences, high-performance computing, and real-world InSAR cases. [**Planned**]

About the Author

My name is Aleksei (Alexey) Pechnikov, a data scientist and software engineer with a Master's degree in Radio Physics and Electronics. I have specialized in forward and inverse modeling for non-linear optics, interferometry, and holography. My work in these areas earned me the first prize in the All-Russian Physics competition in 2004.

With over 20 years of experience, I have engaged in various projects for government agencies, universities, and multinational corporations like LG Corp and Google Inc. I also have teaching experience at the university level, including postgraduate students.

For many years now, I have been living in Thailand with my family, enjoying the country while pursuing my professional endeavors.

I publish articles and posts on LinkedIn, Medium, and Patreon. These platforms allow me to connect with readers and fellow professionals for discussions on concepts, findings, and collaboration opportunities.

For over 9 years, I have been successfully completing projects on Upwork, a renowned freelance platform, working on a variety of challenging scientific and industrial projects. I have also provided long-term support for some of them. My work is characterized by its complexity, efficiency, excellent communication, and effective time management.

Connect with me on the following platforms:

1. YouTube
2. GitHub
3. DockerHub
4. LinkedIn
5. Medium
6. Patreon
7. Upwork