1

A TOUR OF CURRENT SATELLITE MISSIONS AND PRODUCTS

There are thousands of datasets containing observations of the Earth. This chapter describes some satellite types, orbits, and missions, which benefit a variety of fields within Earth sciences, including atmospheric science, oceanography, and hydrology. Data are received on the ground through receiver stations and processed for use using retrieval algorithms. But the raw data requires further manipulation to be useful, and Python is a good choice for analysis and visualization of these datasets.

COPYRIGHTED MATERIAL

At present, there are over 13,000 satellite-based Earth observations freely and openly listed on www.data.gov. Not only is the quantity of available data notable, its quality is equally impressive; for example, infrared sounders can estimate brightness temperatures within 0.1 K from surface observations (Tobin et al., 2013), imagers can detect ocean currents with an accuracy of 1.0 km/hr (NOAA, 2020), and satellite-based lidar can measure the ice-sheet elevation change with a 10 cm sensitivity (Garner, 2015). Previously remote parts of our planet are now observable, including the open oceans and sparsely populated areas. Furthermore, many datasets are available in near real time with image latencies ranging from less than an hour down to minutes – the latter being critically important for natural disaster prediction. Having data rapidly available enables science applications and weather prediction as well as to emergency management and disaster relief. Research-grade data take longer to process (hours to months) but has a higher accuracy and precision, making it suitable for long-term consistency. Thus, we live in the "golden age" of satellite Earth observation. While the data are accessible, the tools and skills necessary to display and analyze this information require practice and training.

Earth Observation Using Python: A Practical Programming Guide, Special Publications 75,
First Edition. Rebekah B. Esmaili.
© 2021 American Geophysical Union. Published 2021 by John Wiley & Sons, Inc.
DOI: 10.1002/9781119606925.ch1

Python is a modern programming language that has exploded in popularity, both within and beyond the Earth science community. Part of its appeal is its easy-to-learn syntax and the thousands of available libraries that can be synthesized with the core Python package to do nearly any computing task imaginable. Python is useful for reading Earth-observing satellite datasets, which can be difficult to use due to the volume of information that results from the multitude of sensors, platforms, and spatio-temporal spacing. Python facilitates reading a variety of self-describing binary datasets in which these observations are often encoded. Using the same software, one can complete the entirety of a research project and produce plots. Within a notebook environment, a scientist can document and distribute the code to other users, which can improve efficiency and transparency within the Earth sciences community.

Satellite data often require some pre-processing to make it usable, but which steps to take and why are not always clear. Data users often misinterpret concepts such as data quality, how to perform an atmospheric correction, or how to implement the complex gridding schemes necessary to compare data at different resolutions. Even to a technical user, the nuances can be frustrating and difficult to overcome. This book walks you through some of the considerations a user should make when working with satellite data.

The primary goal of this text is to get the reader up to speed on the Python coding techniques needed to perform research and analysis using satellite datasets. This is done by adopting an example-driven approach. It is light on theory but will briefly cover relevant background in a nontechnical manner. Rather than getting lost in the weeds, this book purposefully uses realistic examples to explain concepts. I encourage you to run the interactive code alongside reading the text. In this chapter, I will discuss a few of the satellites, sensors, and datasets covered in this book and explain why Python is a great tool for visualizing the data.

1.1. History of Computational Scientific Visualization

Scientific data visualizing used to be a very tedious process. Prior to the 1970s, data points were plotted by hand using devices such as slide rules, French curls, and graph paper. During the 1970s, IBM mainframes became increasingly available at universities and facilitated data analysis on the computer. For analysis, IBM mainframes required that a researcher write Fortran-IV code, which was then printed to cards using a keypunch machine (Figure 1.1). The punch cards then were manually fed into a shared university computer to perform calculations. Each card is roughly one line of code. To make plots, the researcher could create a Fortran program to make an ASCII plot, which creates a plot by combining lines, text, and symbols. The plot could then be printed to a line-printer or a teleprinter. Some institutions had computerized graphic devices, such as Calcomp plotters. Rather than create ASCII plots, the researcher could use a Calcomp plotting

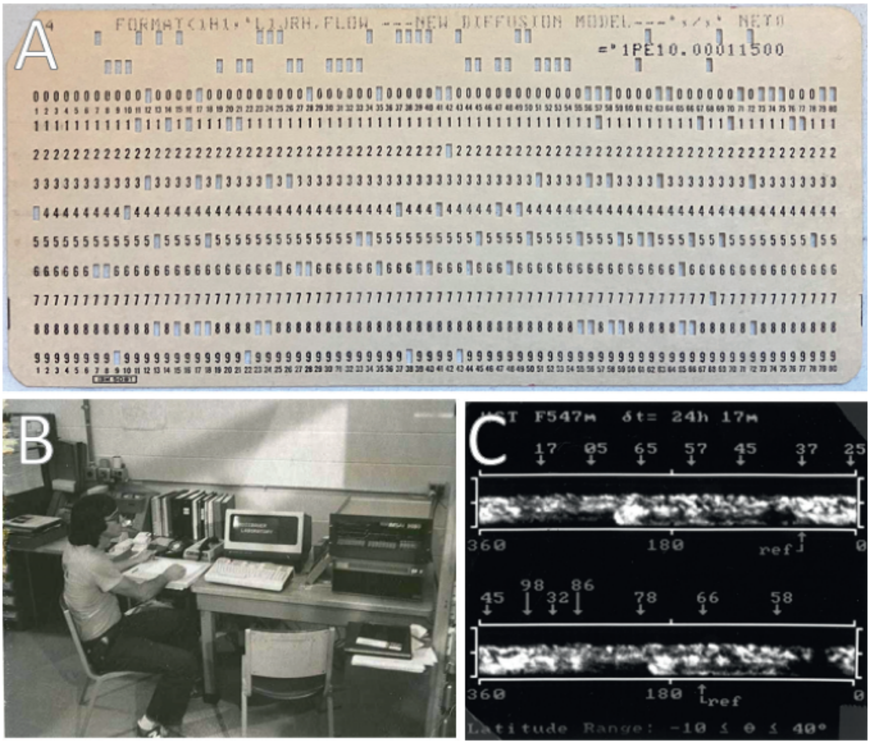

Figure 1.1 (a) An example of a Fortran punch card. Each vertical column represents a character and one card roughly one line of Fortran code. (b) 1979 photo of an IMSAI 8080 computer that could store up to 32 kB of the data, which could then be transferred to a keypunch machine to create punch cards. (c) an image created from the Hubble Space Telescope using a Calcomp printer, which was made from running punch cards and plotting commands through a card reader.

command library to control how data were visualized and store the code on computer tape. The scientist would then take the tape to a plotter, which was not necessarily (or usually) in the same area as the computer or keypunch machine. Any errors – such as bugs in the code, damaged punch cards, or damaged tape – meant the whole process would have to be repeated from scratch.

In the mid-1980s, universities provided remote terminals that would eventually replace the keypunch and card reader machine system. This substantially improved data visualization processes, as scientists no longer had to share limited resources such as keypunch machines, card readers, or terminals. By the late 1980s, personal computers became more affordable for scientists. A typical PC, such as the IBM XT 286, had 640 Kb of random access memory, a 32 MB hard drive, and 5.25 inch floppy disks with 1.2 MB of disk storage (IBM, 1989). At this time, pen plotters became increasingly common for scientific visualization, followed later by the prevalence of ink-jet printers in the 1990s. These technologies allowed researchers to process and visualize data conveniently from their offices. With the proliferation of user-friendly person computers, printers eventually made their way into all homes and offices.

Now with advances in computing and internet access, researchers no longer need to print their visualizations at all, but often keep data in digital form only. Plots can be created in various data formats that easily embed into digital presentations and documents. Scientists often do not ever print visualizations because computers and cloud storage can store many gigabytes of data. Information is created and consumed entirely in digital form. Programming languages, such as Python, can tap into high-level plotting programs and can minimize the axis calculation and labeling requirements within a plot. Thus, the expanded access to computing tools and simplified processes have advanced scientific data visualization opportunities.

1.2. Brief Catalog of Current Satellite Products

In Figure 1.2, you can see that the international community has developed and launched a plethora of Earth-observing satellites, each with several onboard sensors that have a range of capabilities. I am not able to discuss every sensor, dataset, and **mission** (a term coined by NASA to describe projects involving

Figure 1.2 Illustration of current Earth, space weather, and environmental monitoring satellites from the World Meteorological Organization (WMO). Source: U.S. Department of Commerce / NOAA / Public Domain.

spacecraft). However, I will describe some that are relevant to this text, organized by subject area.

1.2.1. Meteorological and Atmospheric Science

Most Earth-observing satellites orbit our planet either in either **geostationary** or **low-Earth orbiting** patterns. These types of satellites tend to be managed and operated by large international government agencies, and the data are often freely accessible online:

- **Geosynchronous equatorial orbit (GEO) satellites.** Geostationary platforms orbit the Earth at 35,700 km above the Earth's surface. GEO satellites are designed to continuously monitor the same region on Earth, and thus can provide many images over a short period of time to monitor change. National Oceanic and Atmospheric Administration (NOAA) operates the Geostationary Environmental Satellite System (GOES) satellites for monitoring North and South America. GOES-16 and -17 have an **advanced baseline instrument** (ABI) onboard to create high-resolution imagery in **visible** and **infrared** (IR) wavelengths. The GOES-16 and -17 satellites are also equipped with the Geostationary Lightning Mapper (GLM) to detect lightning. Instruments designed for space weather include the Solar Ultraviolet Imager (SUVI) and X-ray Irradiance Sensors (EXIS). The European Organization for the Exploitation of Meteorological Satellites (EUMETSAT) operates and maintains the **Meteosat** series GEO satellites that monitor Europe and Africa. The Japan Aerospace Exploration Agency (JAXA) operates and maintains the **Himawari** satellite that monitors Asia and Oceania.
- **Low-Earth orbit (LEO) satellites.** Polar orbiting satellites provide approximately twice daily global observations at the equator (with more observations per day at the poles). Figure 1.3 displays the **equatorial crossing time** for historic and existing LEO satellites, which refers to the local time at the equator when observations are made. Overpasses from some LEO satellites shift during a mission, while others are periodically adjusted back to maintain a consistent overpass time throughout the duration of a mission. Polar orbiting satellites are called low-Earth orbit satellites because they are much closer to the Earth's surface (at 400–900 km) than GEO satellites, which are approximately 40 times further away from the earth or at ~35,000 km. The lower altitude of LEO satellites facilitates their higher spatial resolution relative to GEO, although the temporal resolution tends to be lower for LEO satellites. The **Suomi-NPP** and **NOAA-20** are two satellites that were developed and maintained by NASA and NOAA, respectively. They are each equipped with an **imager**, the Visible Infrared Imaging Radiometer Suite (VIIRS), and infrared and microwave **sounders**, the Cross-track Infrared Sounder (CrIS) and an Advanced Technology Microwave Sounder (ATMS). The **MetOp** series of LEO satellites (named MetOp-A, -B, and -C) were developed by the European Space Agency (ESA) and are operated by EUMETSAT.

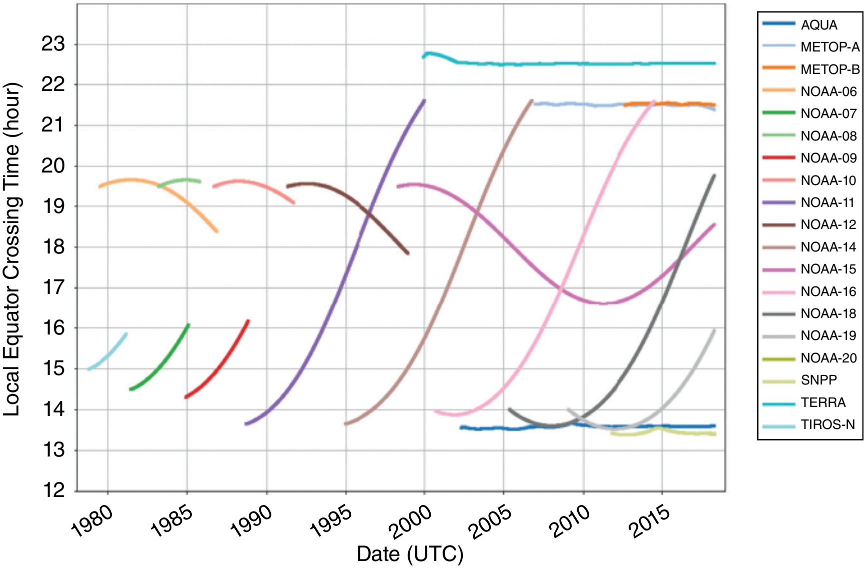

Figure 1.3 Equatorial crossing times for various LEO satellites displayed using Python.

1.2.2. Hydrology

Because water is sensitive to microwave frequencies, microwave instruments and sounders are useful for detecting water vapor, precipitation, and ground moisture. The Global Precipitation Mission (GPM) uses the core GPM satellite along with a constellation of microwave imagers and sounders to estimate global precipitation. The SMAP satellite mission uses active and passive microwave sensors to observe surface soil moisture every two to three days. The GRACE-FO satellite measures gravitational anomalies, that can be used to infer changes in global sea levels and soil moisture. All three hydrology missions were developed and operated by NASA.

1.2.3. Oceanography and Biogeosciences

Both GEO and LEO satellites can provide sea surface temperature (SST) observations. The GOES series of GEO satellites provides continuous sampling of SSTs over the Atlantic and Pacific Ocean basins. The MODIS instrument on the Aqua satellite has been providing daily, global SST observations continuously since the year 2000. Visible wavelengths are useful for detecting ocean color, particularly from LEO satellites, which are often observed at very high resolutions.

Additionally, LEO satellites can detect global sea-surface anomaly parameters. **Jason-3** is a low-Earth satellite developed as a partnership between EUMETSAT, NOAA, NASA, and CNES. The radar altimeter instrument on Jason-3 is sensitive to height changes less than 4 cm and completes a full Earth scan every 10 days (Vaze et al., 2010).

1.2.4. Cryosphere

ICESat-2 (Ice, Cloud, and land Elevation Satellite 2) is a LEO satellite mission designed to measure ice sheet elevation and sea ice thickness. The GRACE-FO satellite mission can also monitor changes in glaciers and ice sheets.

1.3. The Flow of Data from Satellites to Computer

The missions mentioned in the previous section provide open and free data to all users. However, **data delivery**, the process of downloading sensor data from the satellite and converting it into a usable form, is not trivial. Raw sensor data are first acquired on the satellite, then the data must be relayed to the Earth's ground system, often at speeds around 30 Mbits/second. For example, GOES satellite data are acquired by NASA's Wallops Flight Facility in Virginia; data from the Suomi NPP satellite is downloaded to the ground receiving station in Svalbard, Norway (Figure 1.4). Once downloaded, the observations are calibrated and several corrections are applied, such as an atmospheric correction to reduce haze in the image or topographical corrections to adjust changes in pixel brightness on complex terrain. The corrected data are then incorporated into physical products using satellite **retrieval algorithms.** Altogether, the speed of data download and processing can impact the **data latency**, or the difference between the time the physical observation is made and the time it becomes available to the data user.

Data can be accessed in several ways. The timeliest data can be downloaded using a direct broadcast (DB) antenna, which can immediately receive data when the satellite is in range. This equipment is expensive to purchase and maintain, so usually only weather and hazard forecasting offices install them. Most users will access data via the internet. FTP websites post data in near real time, providing the data within a few hours of the observation. Not all data must be timely – research-grade data can take months to calibrate to ensure accuracy. In this case, ordering through an online data portal will grant users access to long records of data.

While data can be easily accessed online, they are rarely analysis ready. Software and web-based tools allow for quick visualization, but to create custom analyses and visualizations, coding is necessary. To combine multiple datasets, each must be gridded to the same resolution for an apples-to-apples comparison. Further, data providers use quality flags to indicate the likelihood of a suitable retrieval. However, the meaning and appropriateness of these flags are not always

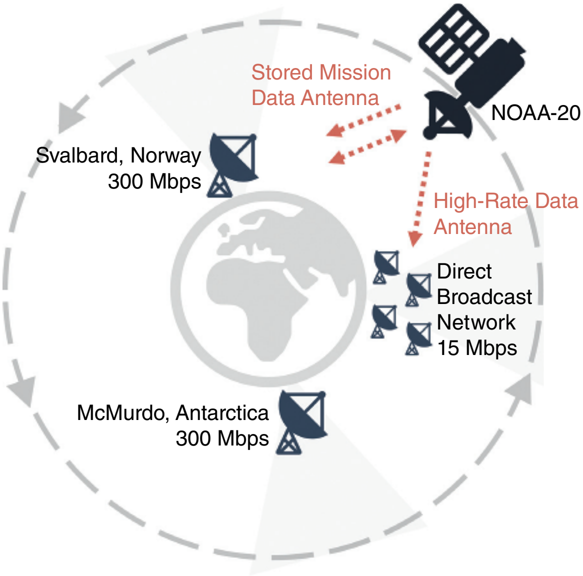

Figure 1.4 NOAA-20 satellite downlink.

well communicated to data users. Moreover, understanding how such datasets are organized can be cumbersome to new users. This text thus aims to identify specific Python routines that enable custom preparation, analysis, and visualization of satellite datasets.

1.4. Learning Using Real Data and Case Studies

I have structured this book so that you can learn Python through a series of examples featuring real phenomena and public datasets. Some of the datasets and visualizations are useful for studying wildfires and smoke, dust plumes, and hurricanes. I will not cover all scenarios encountered in Earth science, but the skills you learn should be transferrable to your field. Some of these case studies include:

- **California Camp Fire (2018).** California Camp Fire was a forest fire that began on November 8, 2018, and burned for 17 days over a 621 km^2 area. It was primarily caused by very low regional humidity due to strong gusting wind events and a very dry surface. The smoke from the fire also affected regional air quality. In this case study, I will examine satellite observations to show the location and intensity as well as the impact that the smoke had on regional CO, ozone, and aerosol optical depth (AOD). Combined satellite channels also provide useful imagery for tracking smoke, such as the dust

RGB product. Land datasets such as the Normalized Difference Vegetation Index (NDVI) are useful for highlighting burn scars from before and after the fire events.

- **Hurricane Michael (2018).** Michael was a major hurricane that affect the Florida Panhandle of the United States. Michael developed as a tropical wave on October 7 in the southwest Caribbean Sea and grew into a Category 5 storm by October 10. Throughout its life cycle, Michael caused extensive flooding, leading to 74 deaths and $25 billion in damage. Several examples in this text utilize visible and infrared imagery of Hurricane Michael.
- **Louisiana Flood Event (2016).** Thousands of homes were flooded in Louisiana when over 20 inches of rain fell between August 12 and August 21, 2016. The event began after a mesoscale convective system stalled over the area near Baton Rouge and Lafayette, Louisiana. I will use the IMERG global rainfall dataset to examine this event.

1.5. Summary

I have provided a brief overview of the many satellite missions and datasets that are available. This book has two main objectives: (1) to make satellite data and analysis accessible to the Earth science community through practical Python examples using real-world datasets; and (2) to promote a reproducible and transparent scientific code philosophy. In the following chapters, I will focus on describing data conventions, common methods, and problem-solving skills required to work with satellite datasets.

References

Tobin, D., Revercomb, H., Knuteson, R., Taylor, J., Best, F., Borg, L., et al. (2013). Suomi-NPP CrIS radiometric calibration uncertainty. *Journal of Geophysical Research: Atmospheres, 118*(18), 10,589–10,600. https://doi.org/10.1002/jgrd.50809

Garner, R. (2015, July 10). ICESat-2 Technical Requirements. www.nasa.gov/content/goddard/icesat-2-technical-requirements

IBM (1989, January). Personal Computer Family Service Information Manual. IBM document SA38-0037-00. http://bitsavers.trailing-edge.com/pdf/ibm/pc/SA38-0037-00_Personal_Computer_Family_Service_Information_Manual_Jul89.pdf

National Oceanic and Atmospheric Administration (2020, June 12). GOES-R Series Level I Requirements (LIRD). www.goes-r.gov/syseng/docs/LIRD.pdf.