

Groundwater Monitoring with GRACE (A2.1)

Authors

A.J. Purdy, J.S. Famiglietti

Overview

The following tutorial details how to use observations from the Gravity Recovery and Climate Experiment (GRACE) to evaluate changes in groundwater storage for a large river basin. Here, you will learn how to apply remote sensing estimates of total water storage anomalies, land surface model output, and in situ observations to resolve groundwater storage changes in California's Central Valley. The following method has been applied to study water storage changes around the world, and can be ported to quantify groundwater storage change for major river basins.

Learning Outcomes

- Plotting changes in total water storage using GRACE.
- Mapping trends in water storage.
- Resolving changes in groundwater storage for a river basin.
- Import image collections and create image collections from assets.
- Create charts by reducing an `ImageCollection` with a feature geometry.

Helps if you know how to:

- Use expressions to perform calculations on image bands (Chap. F3.1).
- Write a function and `map` it over an `ImageCollection` (Chap. F4.0).
- Fit linear and nonlinear functions with regression in an `ImageCollection` time series (Chap. F4.6).
- Use `ee.Join` to join one `ImageCollection` to another to compute differences (Chap. F4.9).
- Filter a `FeatureCollection` to obtain a subset (Chap. F5.0, Chap. F5.1).

Introduction to Theory

Since 2002, GRACE and the follow-on mission, GRACE-FO, have provided a new vantage to track changes in water resources (Tapley et al. 2004). GRACE holds the unique ability to directly track changes in total water storage anomalies (TWSa), according to the following equation:

$$\text{TWSa} = \text{CANa} + \text{SWa} + \text{SMa} + \text{SWEa} + \text{GWa} \quad (\text{A2.1.1})$$

where CANa is canopy water storage anomaly, SWa is the surface water anomaly, Sma is the soil moisture anomaly, SWEa is the snow water equivalent anomaly, and Gwa is the groundwater storage anomaly.

By utilizing supplemental observations from other remote sensing platforms and land surface models and rearranging Eq. A2.1.1, scientists have been able to resolve changes in groundwater storage within major river basins around the planet (Famiglietti et al. 2014). From Bangladesh (Purdy et al. 2019) and India (Rodell et al. 2009) to the Middle East (Voss et al. 2013) and the American Southwest (Castle et al. 2014), the problem of declining groundwater storage has emerged with varying levels of severity (Richey et al. 2015). Along with many other regions around the world, California shares an overreliance on groundwater (Famiglietti et al. 2011). This tutorial demonstrates the analytical steps to resolve groundwater storage changes using GRACE for California's Central Valley.

Practicum

Section 1. Exploring the Study Area

If you have not already done so, you can add the book's code repository to the Code Editor by entering https://code.earthengine.google.com/?accept_repo=projects/gee-edu/book (or the short URL bit.ly/EEFA-repo) into your browser. The book's scripts will then be available in the script manager panel to view, run, or modify. If you have trouble finding the repo, you can visit bit.ly/EEFA-repo-help for help.

Evaluating changes in hydrologic storage requires examining change within a hydrologically connected system. Watersheds and basins represent areas of land where precipitation drains to a common point. We will use already-generated basins from the Watershed Boundary Dataset (WBD) to delineate the drainage area for California's Central Valley. The WBD includes hydrologic unit codes (HUCs) to identify connected basins within the United States.

In the following sections of code, we will load three basins by their unique four-digit HUCs and merge the basins together. To accomplish this task, we will use the `ee.Filter.inList` function to filter the `basins` variable by the `'huc4'` property, extracting three to a variable `basin`.

```
// Import Basins.
var basins = ee.FeatureCollection('USGS/WBD/2017/HUC04');

// Extract the 3 HUC 04 basins for the Central Valley.
```

```
var codes = ['1802', '1803', '1804'];
var basin = basins.filter(ee.Filter.inList('huc4', codes));

// Add the basin to the map to show the extent of our analysis.
Map.centerObject(basin, 6);
Map.addLayer(basin, {
  color: 'green'
}, 'Central Valley Basins', true, 0.5);
```

Section 1.1. Map the Extent of Agriculture in the Region

To get a sense for the extent of agriculture in California, we can visualize all cultivated land in the Central Valley. This will map where the greatest need for water occurs.

```
var landcover = ee.ImageCollection('USDA/NASS/CDL')
  .filter(ee.Filter.date('2019-01-01', '2019-12-31'))
  .select('cultivated');

Map.addLayer(landcover.first().clip(basin), {}, 'Cropland', true,
  0.5);
```

The extent of cultivated lands shows up as a translucent purple (Fig. A2.1.1).



Fig. A2.1.1 California's Central Valley Basin, including agricultural lands

Section 1.2. Load Reservoir Locations

California has over 150 reservoirs distributed across the state. These reservoirs vary in size and capacity and the regions of the state that they support. For the Central Valley, water conveyance infrastructure allows the transport of water from north to south. We will use our basin boundary to select the reservoirs within our basin to quantify changes in surface water storage. The list of reservoirs was gathered from the California Department of Water Resources' Data Exchange Center (CDEC). For an application in another study region, acquiring in situ surface water storage would be required to resolve that region's groundwater storage changes.

```
// This table was generated using the index from the CDEC website
var res = ee.FeatureCollection(
  'projects/gee-book/assets/A2-1/ca_reservoirs_index');
// Filter reservoir locations by the Central Valley geometry
var res_cv = res.filterBounds(basin);
Map.addLayer(res_cv, {
  'color': 'blue'
}, 'Reservoirs');
```

The blue dots that now appear on the map represent the distribution of water storage across the Central Valley. Water conveyance infrastructure and natural rivers deliver water to farms across the valley. Despite all these reservoirs, many water users continue to rely on groundwater to meet their needs. A 2011 study detailed the magnitude of this reliance using gravity-sensing satellites (Famiglietti et al. 2011). The next sections of this chapter reveal how to resolve groundwater storage changes using these methods.

Code Checkpoint A21a. The book's repository contains a script that shows what your code should look like at this point.

Section 2. Tracking Total Water Storage Changes in California with GRACE

GRACE can directly track changes in TWSa. Changes in TWSa indicate which regions are gaining or losing water.

Section 2.1. Import GRACE Data and Plot Changes in Total Water Storage in California

First, we will import the image collection and select the proper band to chart.

```
var GRACE = ee.ImageCollection('NASA/GRACE/MASS_GRIDS/MASCON_CRI');
```

```
// Subset GRACE for liquid water equivalent dataset
var basinTWSa = GRACE.select('lwe_thickness');
```

The GRACE data imported here have already been processed to provide units of TWSa. The data contained in this dataset are units of “equivalent water thickness” anomalies. GRACE hydrologic data are presented as anomalies because GRACE does not directly observe the gravitational pull of only water. The gravity observed also includes Earth's surface (e.g., mountains). To disentangle the signal of water we can look at changes relative to a longer term mean gravity signal. The anomalies represent the difference between a given month's observation and a multi-year mean. We will now plot TWSa for basins in a large part of California (Fig. A2.1.2).

```
// Make plot of TWSa for Basin Boundary
var TWSaChart = ui.Chart.image.series({
  imageCollection: basinTWSa.filter(ee.Filter.date(
    '2003-01-01', '2016-12-31')),
  region: basin,
  reducer: ee.Reducer.mean(),
})
.setOptions({
  title: 'TWSa',
  hAxis: {
    format: 'MM-yyyy'
  },
  vAxis: {
    title: 'TWSa (cm)'
  },
  lineWidth: 1,
});
print(TWSaChart);
```

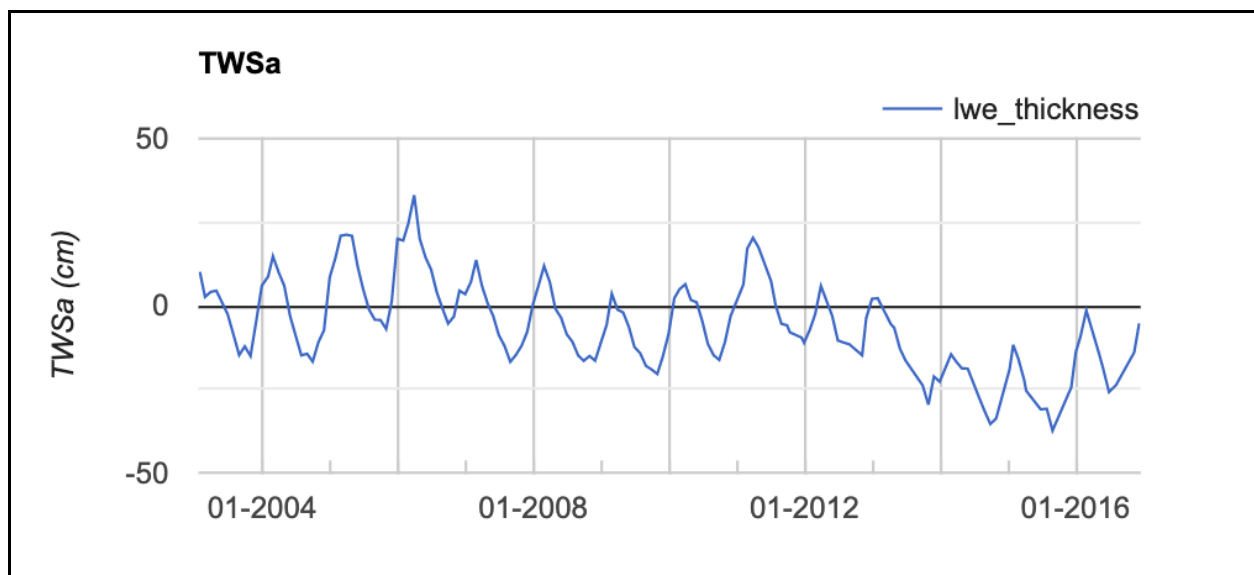


Fig. A2.1.2 TWSa for the Sacramento-San Joaquin Basin

In the **Console**, you will see a plot of TWSa. Notice the seasonality and interannual variations in TWSa. Winter months reveal periods of maximum water storage due to snowpack, full reservoirs, and wet soil. Summer and early fall reveal less TWSa, as the snow has melted, reservoir water has been used, and soil is drying out. Additionally, summer months are periods when groundwater is extracted and used to supplement a limited surface water supply. Evidence of drought emerged through declining TWSa between 2006–2009 and 2012–2017.

Next, we will look at the trend in TWSa for the entire period of record.

Section 2.2. Estimate the Linear Trend in TWSa Over Time

As presented in Chap. F4.6, Earth Engine can fit linear models to time series data, with unique linear fits for each pixel based on the values through time. Consider the following linear model, where et is a random error:

$$pt = \beta_0 + \beta_1 t + et \quad (1) \quad (A2.1.2)$$

This is the model behind the trendline added to the chart we just created. This model is useful for detrending data and reducing non-stationarity in the time series (Shumway and Stoffer 2017). The goal of the regression is to discover the values of the β 's in each pixel.

To fit this trend model to the GRACE-based TWSa series using ordinary least squares, we can use the `linearRegression` reducer.

```

// Compute Trend for each pixel to map regions of most change
var addVariables = function(image) {
  // Compute time in fractional years since the epoch.
  var date = ee.Date(image.get('system:time_start'));
  var years = date.difference(ee.Date('2003-01-01'), 'year');
  // Return the image with the added bands.
  return image
    // Add a time band.
    .addBands(ee.Image(years).rename('t').float())
    // Add a constant band.
    .addBands(ee.Image.constant(1));
};
var cvTWSa = basinTWSa.filterBounds(basin).map(addVariables);
print(cvTWSa);
// List of the independent variable names
var independents = ee.List(['constant', 't']);

// Name of the dependent variable.
var dependent = ee.String('lwe_thickness');
// Compute a linear trend. This will have two bands: 'residuals' and
// a 2x1 band called coefficients (columns are for dependent
// variables).
var trend = cvTWSa.select(independents.add(dependent))
  .reduce(ee.Reducer.linearRegression(independents.length(), 1));

```

The image of coefficients, computed below, is a two-band image in which each pixel contains values for β_0 and β_1 . The β_1 value will represent the temporal slope for the GRACE mascon.

```

// Flatten the coefficients into a 2-band image
var coefficients = trend.select('coefficients')
  .arrayProject([0])
  .arrayFlatten([independents]);

```

Next, we can visualize the GRACE trends to capture the spatial scales on which GRACE can resolve TWSa. GRACE is adept at capturing these changes only for larger basins.

```

// Create a layer of the TWSa slope to add to the map
var slope = coefficients.select('t');

```

```
// Set visualization parameters to represent positive (blue) &
// negative (red) trends
var slopeParams = {
  min: -3.5,
  max: 3.5,
  palette: ['red', 'white', 'blue']
};
Map.addLayer(slope.clip(basin), slopeParams, 'TWSa Trend', true,
  0.75);
```

The slope layer reveals that the Tulare Basin (the southernmost basin in the Central Valley) experienced the largest negative changes in TWSa over the time period (Fig. A2.1.3). Darker reds indicate greater negative change and blue represents positive change. This is a result of the region not receiving winter rain or snow and having the most junior surface water rights in the Central Valley.

The next steps in this chapter will review how to unpack the TWSa signal to resolve changes in groundwater storage anomalies for the basin.

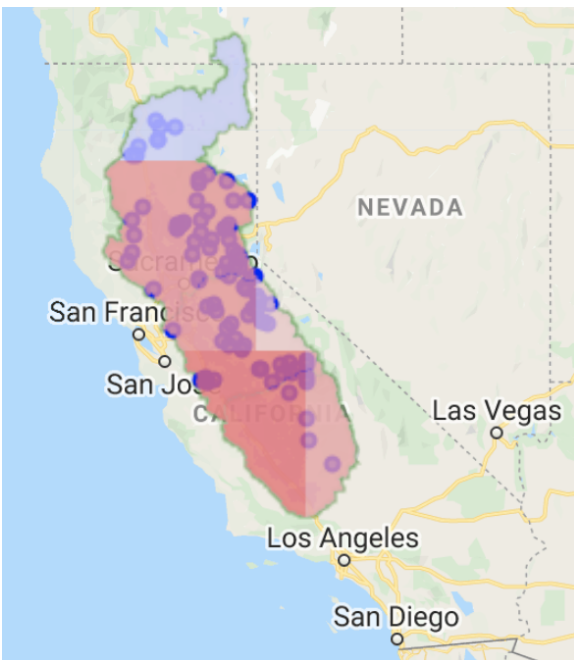


Fig. A2.1.3 Slope in TWSa for California. Darker reds indicate greater declines in total water storage. Blue represents increases in water storage. White represents no change in water storage.

Code Checkpoint A21b. The book's repository contains a script that shows what your code should look like at this point.

Section 3. Tracking Changes in Soil Water Storage and Snow Water Equivalent in California

The Global Land Data Assimilation System (GLDAS) utilizes multiple land surface models to globally resolve fluxes in storage of water (like soil moisture and snow) and energy at a three-hour frequency (Rodell et al. 2004). An example of how to convert three-hourly GLDAS snow water equivalent to annual SWEa for 2003 can be found in script **A21s1** in the book's repository. Running the supplemental script is an optional part of this lab: it is added to provide clarity on how the image assets were created for each GLDAS variable in this chapter.

For the next analysis, you will be starting with a script that imports the GLDAS SMA and SWEa processed by the methods above. GLDAS estimates of soil moisture and snow water equivalent are resolved at a three-hour temporal frequency. Therefore, we have taken the time to reduce the GLDAS data to annual means from the three-hour estimates.

Additionally, we aggregated monthly GRACE observations to annual average estimates to improve the efficiency of running this analysis. More experienced users can adapt these methods to resolve monthly changes. However, it should be noted that to replicate the same methods at a monthly cadence would require the interpolation of missing months of GRACE observations. Please use the code starting point below, as the script imports the necessary assets to complete the final analysis.

Section 3.1. Load GLDAS Soil Moisture Images from an Asset to an Image Collection

Code Checkpoint A21c. The book's repository contains a script to use to begin this section. You will need to start with that script and paste code below into it.

When you run the script, you will see a number of assets being imported and an annual time series of GRACE. Additionally, the script is set to convert the list of annual mean soil moisture images to an `ImageCollection`.

```
var gldas_sm_list = ee.List([sm2003, sm2004, sm2005, sm2006, sm2007,
                             sm2008, sm2009, sm2010, sm2011, sm2012, sm2013, sm2014,
                             sm2015, sm2016])
```

```
]);
var sm_ic = ee.ImageCollection.fromImages(gldas_sm_list);
```

Before we compute groundwater storage anomalies from GRACE and GLDAS data following Eq. A2.1.1, we should inspect the units to ensure that our math is sound. In the **search bar** of Earth Engine, search for “GLDAS” and click on **GLDAS-2.1: Global Land Data Assimilation System**, then navigate to **Bands** to see what the units are for `'RootMoist_inst'` and `'SWE_inst'`.

The units for GLDAS are currently showing as kg/m². We need to convert the soil moisture and snow values to equivalent water depth units of centimeters. Define the following conversion variable and map this over the `ImageCollection`. As described in Chap. F4.0 and Chap. F4.1, mapping over an `ImageCollection` is similar to running a loop: You apply the same function to each image and return the value back to the `ImageCollection`.

```
var kgm2_to_cm = 0.10;
var sm_ic_ts = sm_ic.map(function(img) {
  var date = ee.Date.fromYMD(img.get('year'), 1, 1);
  return img.select('RootMoist_inst').multiply(kgm2_to_cm)
    .rename('SMa').set('system:time_start', date);
});
```

In addition to converting the units, the code renames the variable and sets properties such as `'system:time_start'`, which is necessary in Earth Engine to plot data and compare it with other image collections. Note that you might print out the variables `sm_ic` and `sm_ic_ts` to explore the differences between them. You should notice the new band name and properties (e.g., `'system:time_start'`).

Next, plot the data to evaluate soil moisture anomalies during the study period.

```
// Make plot of SMa for Basin Boundary
var SMaChart = ui.Chart.image.series({
  imageCollection: sm_ic_ts.filter(ee.Filter.date(
    '2003-01-01', '2016-12-31')),
  region: basin,
  reducer: ee.Reducer.mean(),
  scale: 25000
});
```

```

.setChartType('ScatterChart')
.setOptions({
  title: 'Soil Moisture anomalies',
  trendlines: {
    0: {
      color: 'CC0000'
    }
  },
  hAxis: {
    format: 'MM-yyyy'
  },
  vAxis: {
    title: 'SMa (cm)'
  },
  lineWidth: 2,
  pointSize: 2
});
print(SMaChart);

```

You may notice that SMa is of a similar magnitude to TWSa, but is slightly out of phase with TWSa.

Section 3.2. Load GLDAS Snow Water Equivalent Images from an Asset to an Image Collection

Use similar code to load the snow water equivalent data to Earth Engine.

```

var gldas_swe_list = ee.List([swe2003, swe2004, swe2005, swe2006,
  swe2007, swe2008, swe2009, swe2010, swe2011, swe2012,
  swe2013, swe2014, swe2015, swe2016
]);
var swe_ic = ee.ImageCollection.fromImages(gldas_swe_list);

```

Next, convert the snow values to equivalent water depth units of centimeters.

```

var swe_ic_ts = swe_ic.map(function(img) {
  var date = ee.Date.fromYMD(img.get('year'), 1, 1);
  return img.select('SWE_inst').multiply(kgm2_to_cm).rename(
    'SWEa').set('system:time_start', date);
});

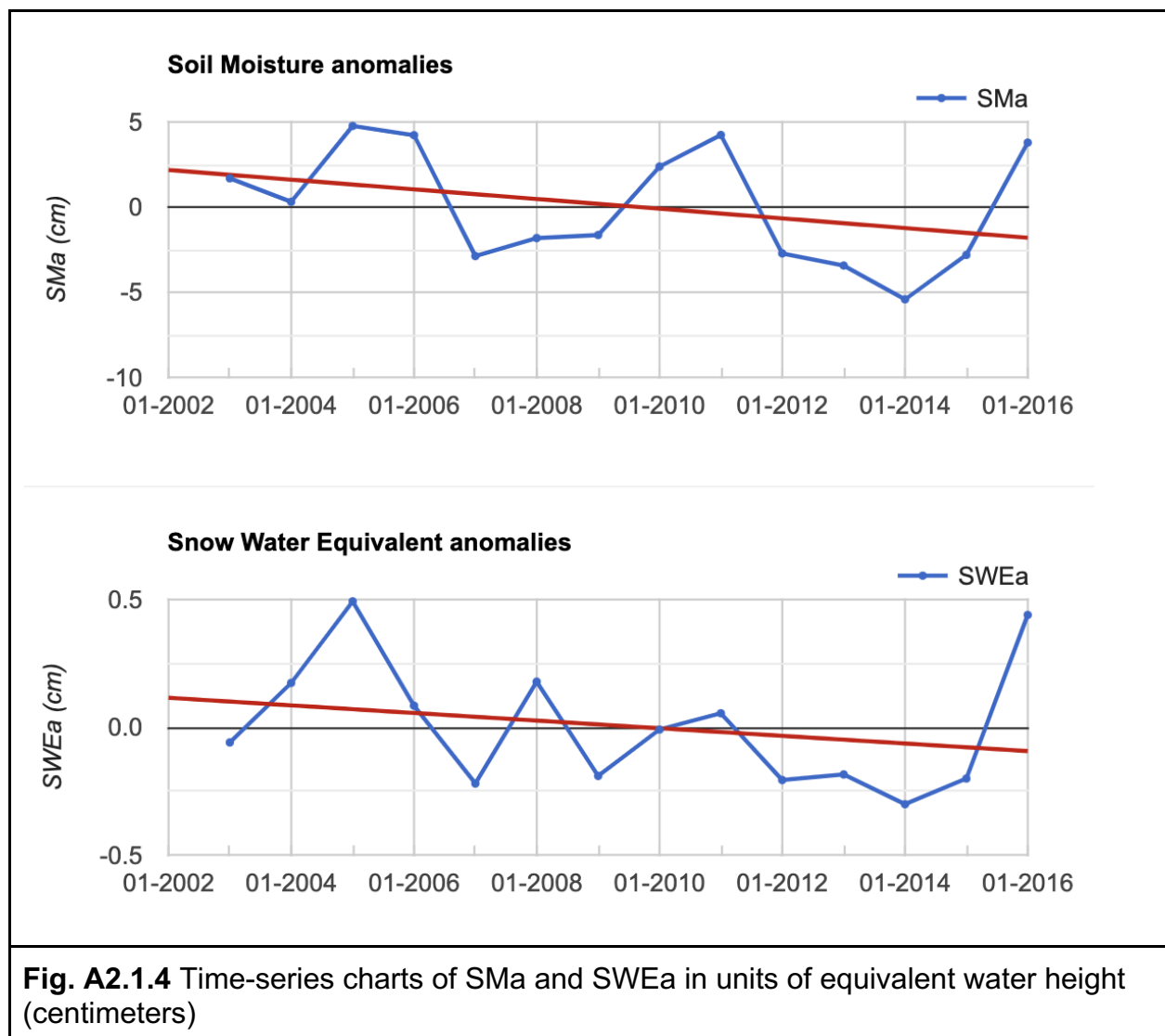
```

Next, we will visualize the new `ImageCollection`. If you did not do the previous step, your code here will not run.

```
// Make plot of SWEa for Basin Boundary
var SWEaChart = ui.Chart.image.series({
  imageCollection: swe_ic_ts.filter(ee.Filter.date(
    '2003-01-01', '2016-12-31')),
  region: basin,
  reducer: ee.Reducer.mean(),
  scale: 25000
})
.setChartType('ScatterChart')
.setOptions({
  title: 'Snow Water Equivalent anomalies',
  trendlines: {
    0: {
      color: 'CC0000'
    }
  },
  hAxis: {
    format: 'MM-yyyy'
  },
  vAxis: {
    title: 'SWEa (cm)'
  },
  lineWidth: 2,
  pointSize: 2
});
print(SWEaChart);
```

You successfully plotted soil moisture and snow water equivalent (Fig. A2.1.4). You may notice that SWEa is much smaller in magnitude than the other two variables.

Code Checkpoint A21d. The book's repository contains a script that shows what your code should look like at this point.



Section 4. Importing a Table of Surface Water Storage

Reservoir storage data from the California Data Exchange Center (CDEC) facilitated computing Surface Water storage anomalies (SWa) for the Sacramento-San Joaquin Basin. Surface water storage, unlike the other components of water storage, is not represented in land surface models. Instead, SWa is sourced from in situ observations. Here, the reservoir storage observations were summed for the Central Valley and then total converted to annual anomalies to directly compare with SWEa and SMa. Prior to reading the table of reservoir storage, we compute the area from the combined HUC8 basins.

```
// Extract geometry to convert time series of anomalies in km3 to cm
```

```
var area_km2 = basin.geometry().area().divide(1000 * 1000);
var km_2_cm = 100000;
```

Next, the imported table `res_table` is converted to an `ImageCollection` of constant values to facilitate combining the data with GRACE and GLDAS-resolved water storage anomalies.

```
// Convert csv to image collection
var res_list = res_table.toList(res_table.size());
var yrs = res_list.map(function(ft) {
  return ee.Date.fromYMD(ee.Feature(ft).get('YEAR'), 1, 1);
});
var SWanoms = res_list.map(function(ft) {
  return ee.Image.constant(ee.Feature(ft).get('Anom_km3'));
});
var sw_ic_ts = ee.ImageCollection.fromImages(
  res_list.map(
    function(ft) {
      var date = ee.Date.fromYMD(ee.Feature(ft).get('YEAR'),
        1, 1);
      return ee.Image.constant(ee.Feature(ft).get(
        'Anom_km3')).divide(area_km2).multiply(
        km_2_cm).rename('SWa').set(
        'system:time_start', date);
    }
  )
);
```

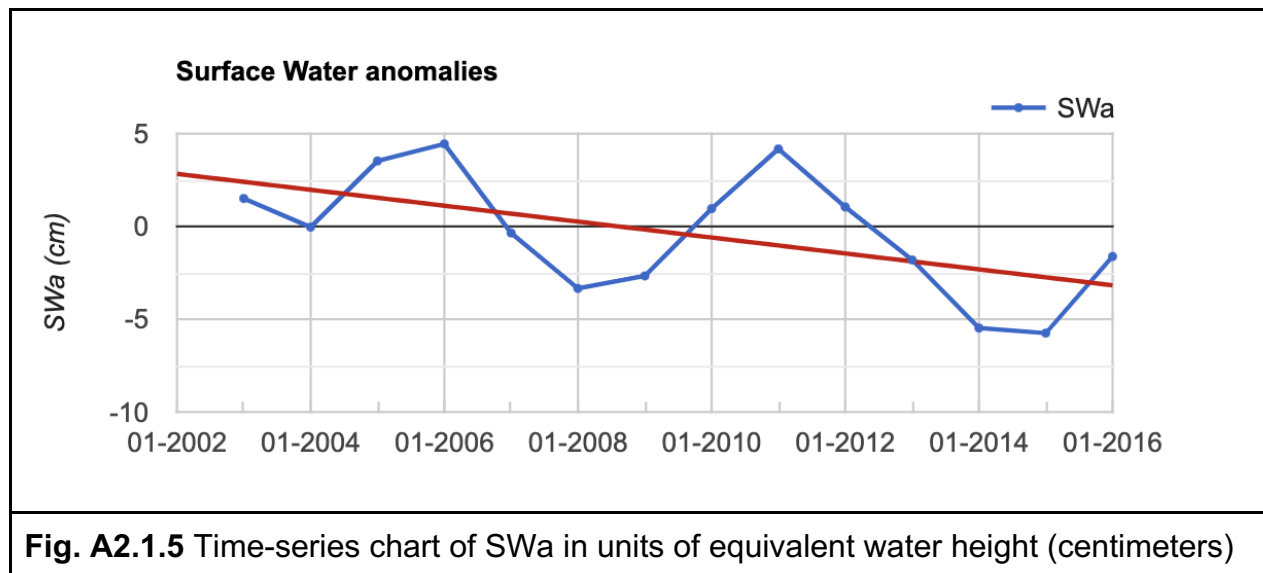
Plot SWa in equivalent units of centimeters per year (Fig. A2.1.5).

```
// Create a time series of Surface Water Anomalies
var SWaChart = ui.Chart.image.series({
  imageCollection: sw_ic_ts.filter(ee.Filter.date(
    '2003-01-01', '2016-12-31')),
  region: basin,
  reducer: ee.Reducer.mean(),
  scale: 25000
})
.setChartType('ScatterChart')
```

```

.setOptions({
  title: 'Surface Water anomalies',
  trendlines: {
    0: {
      color: 'CC0000'
    }
  },
  hAxis: {
    format: 'MM-yyyy'
  },
  vAxis: {
    title: 'SWa (cm)'
  },
  lineWidth: 2,
  pointSize: 2
});
print(SWaChart);

```



The chart shows that surface water anomalies are of a similar magnitude to soil moisture anomalies. As expected, SWa decreases during each drought period in California (2006–2008 and 2012–2016). These reservoir storage declines show use is greater than inputs during each period.

Now, we will combine the previous datasets to resolve changes in groundwater during the period of record. Unfortunately, it's still hard to quantify change without having all the

variables on one plot. It might be best to compute the differences via Eq. A2.1.1 from the introductory paragraph at the top of the document.

Code Checkpoint A21e. The book's repository contains a script that shows what your code should look like at this point.

Section 5. Combining Image Collections

Here, you will see how to combine multiple image collections and compute differences via an expression. We will start by joining the GLDAS image collections together. This is accomplished with the `ee.Join.inner` function.

```
// Combine GLDAS & GRACE Data to compute change in human accessible
water
var filter = ee.Filter.equals({
  leftField: 'system:time_start',
  rightField: 'system:time_start'
});
// Create the join.
var joindata = ee.Join.inner();
// Join GLDAS data
var firstJoin = ee.ImageCollection(joindata.apply(swe_ic_ts, sm_ic_ts,
  filter));
var join_1 = firstJoin.map(function(feature) {
  return ee.Image.cat(feature.get('primary'), feature.get(
    'secondary'));
});
print('Joined', join_1);
```

Next, we join the reservoir data.

```
// Repeat to append Reservoir Data now
var secondJoin = ee.ImageCollection(joindata.apply(join_1, sw_ic_ts,
  filter));
var res_GLDAS = secondJoin.map(function(feature) {
  return ee.Image.cat(feature.get('primary'), feature.get(
    'secondary'));
});
```

Lastly, we need to repeat this step by joining GRACE to the output from the last join.


```
// Repeat to append GRACE now
var thirdJoin = ee.ImageCollection(joindata.apply(res_GLDAS, GRACE_yr,
  filter));
var GRACE_res_GLDAS = thirdJoin.map(function(feature) {
  return ee.Image.cat(feature.get('primary'), feature.get(
    'secondary'));
});
```

Take a moment to print out the `ImageCollection` `GRACE_res_GLDAS`.

To resolve groundwater storage changes in the basin, one can rearrange Eq. A2.1.1 to solve for GWa . Here we assume canopy storage anomalies are very small relative to other storage components and ignore them in the equation below.

$$GWa = TWSa - SWa - SMa - SWEa \quad (A2.1.2)$$

To execute this step we map an expression across an `ImageCollection` to produce a new variable named `GWa`.

```
// Compute groundwater storage anomalies
var GWa = ee.ImageCollection(GRACE_res_GLDAS.map(function(img) {
  var date = ee.Date.fromYMD(img.get('year'), 1, 1);
  return img.expression(
    'TWSa - SWa - SMa - SWEa', {
      'TWSa': img.select('TWSa'),
      'SMa': img.select('SMa'),
      'SWa': img.select('SWa'),
      'SWEa': img.select('SWEa')
    }).rename('GWa').copyProperties(img, [
      'system:time_start'
    ]);
}));
print('GWa', GWa);
```

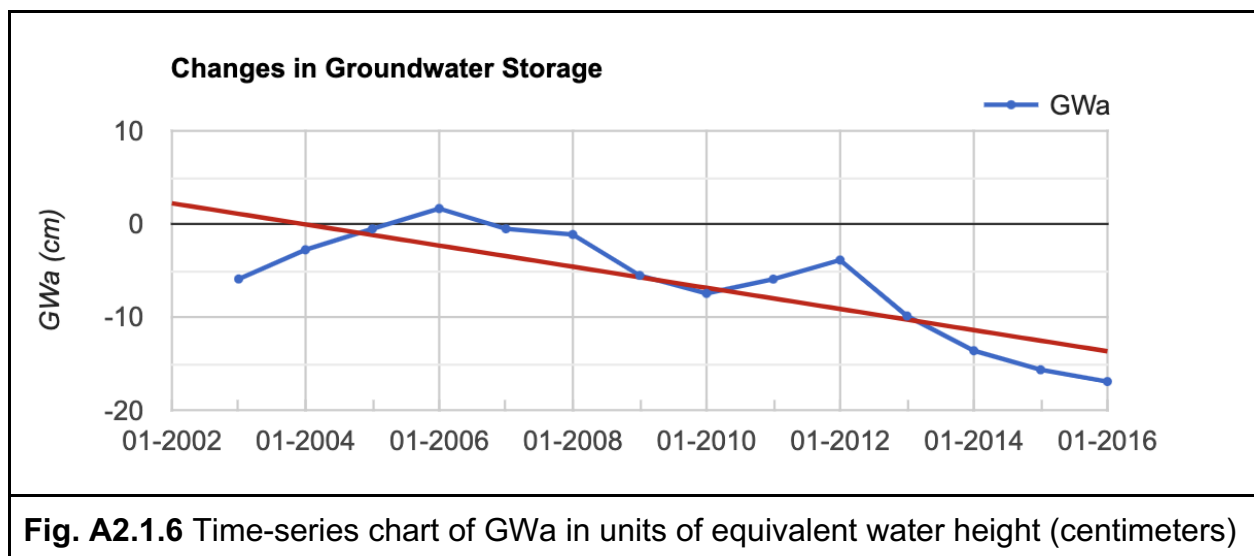
You can see how the variable `img` is used to extract bands from the combined `ImageCollection` and create a new one with just one band.

We'll plot this to see how groundwater storage is changing (Fig. A2.1.6).

```

// Chart Results
var GWaChart = ui.Chart.image.series({
  imageCollection: GWa.filter(ee.Filter.date('2003-01-01',
    '2016-12-31')),
  region: basin,
  reducer: ee.Reducer.mean(),
  scale: 25000
})
.setChartType('ScatterChart')
.setOptions({
  title: 'Changes in Groundwater Storage',
  trendlines: {
    0: {
      color: 'CC0000'
    }
  },
  hAxis: {
    format: 'MM-yyyy'
  },
  vAxis: {
    title: 'Gwa (cm)'
  },
  lineWidth: 2,
  pointSize: 2
});
print(GWaChart);

```



You can see how reliant California is on groundwater. The chart shows large declines during recent drought periods. Using the chart, you can estimate how much groundwater was used during the 2012–2016 drought period. In the **Console**, hover your mouse over the chart and jot down the value for GWa in 2012 and in 2016. You will use this information, in addition to the area and unit conversion, to estimate groundwater usage during this period in cubic kilometers.

```
// Now look at the values from the start of 2012 to the end of 2016
// drought.
// 2012 -3.874 cm --> 2016 -16.95 cm
// This is a ~13 cm / 100000 (cm/km) * Area 155407 km2 =
var loss_km3 = ee.Number(-3.874).subtract(-16.95).divide(km_2_cm)
    .multiply(area_km2);
print('During the 2012-2016 drought, CA lost ', loss_km3,
    'km3 in groundwater');
```

Code Checkpoint A21f. The book’s repository contains a script that shows what your code should look like at this point.

Synthesis

Assignment 1. This chapter provides a roadmap to monitor changes in groundwater storage at a basin scale using observations of TWSa from the GRACE satellites and hydrologic data from GLDAS. Now you can apply these methods to another river basin anywhere in the world.

Conclusion

In this chapter, we reviewed how GRACE observations can be used to estimate changes in water storage for a region of interest like California's Central Valley. Specifically, this chapter demonstrated how to combine equivalent water thickness observations from GRACE with model simulations of soil moisture, snow water equivalent, and in situ reservoir storage observations to quantify groundwater storage declines. Along the way, some advanced Earth Engine skills were explored, including creating an `ImageCollection` from a table and joining multiple image collections. Earth Engine users now have the skills and the knowledge of GRACE observations to apply this methodology to other regions around the world.

Feedback

To review this chapter and make suggestions or note any problems, please go now to bit.ly/EEFA-review. You can find summary statistics from past reviews at bit.ly/EEFA-reviews-stats.

References

- Castle SL, Thomas BF, Reager JT, et al (2014) Groundwater depletion during drought threatens future water security of the Colorado River Basin. *Geophys Res Lett* 41:5904–5911. <https://doi.org/10.1002/2014GL061055>
- Famiglietti JS (2014) The global groundwater crisis. *Nat Clim Chang* 4:945–948. <https://doi.org/10.1038/nclimate2425>
- Famiglietti JS, Lo M, Ho SL, et al (2011) Satellites measure recent rates of groundwater depletion in California's Central Valley. *Geophys Res Lett* 38 <https://doi.org/10.1029/2010GL046442>
- Purdy AJ, David CH, Sikder MS, et al (2019) An open-source tool to facilitate the processing of GRACE observations and GLDAS outputs: An evaluation in Bangladesh. *Front Environ Sci* 7 <https://doi.org/10.3389/fenvs.2019.00155>
- Richey AS, Thomas BF, Lo MH, et al (2015) Quantifying renewable groundwater stress with GRACE. *Water Resour Res* 51:5217–5237. <https://doi.org/10.1002/2015WR017349>
- Rodell M, Houser PR, Jambor U, et al (2004) The global land data assimilation system. *Bull Am Meteorol Soc* 85:381–394. <https://doi.org/10.1175/BAMS-85-3-381>

Rodell M, Velicogna I, Famiglietti JS (2009) Satellite-based estimates of groundwater depletion in India. *Nature* 460:999–1002. <https://doi.org/10.1038/nature08238>

Voss KA, Famiglietti JS, Lo M, et al (2013) Groundwater depletion in the Middle East from GRACE with implications for transboundary water management in the Tigris-Euphrates-Western Iran region. *Water Resour Res* 49:904–914. <https://doi.org/10.1002/wrcr.20078>