Install OPERA DISP-S1 data access and preparation environment

Instructions derived and modified from https://github.com/nisar-solid/ATBD/blob/main/docs/installation.md and https://github.com/OPERA-Cal-Val/calval-DISP/blob/main/docs/installation.md/

Prepared by: Bryan Raimbault, Alexander Handwerger, Grace Bato Simran Sangha, Jin Woo Kim

1. Install Miniforge - Conda/Mamba

```
mkdir -p /path/to/folder/tools; cd /path/to/folder/tools

# download, install and setup (mini)conda/mamba
# for Linux:
wget https://github.com/conda-forge/miniforge/releases/latest/download/Miniforge3-Linux-x86_64.sh
# for macOS with Apple Silicon:
curl https://github.com/conda-forge/miniforge/releases/latest/download/Miniforge3-MacOSX-arm64.sh -o Miniforge3-MacOSX-arm64.sh
# for macOS with Intel:
curl https://github.com/conda-forge/miniforge/releases/latest/download/Miniforge3-MacOSX-x86_64.sh -o Miniforge3-MacOSX-x86_64.sh
# Install Miniforge (adjust filename and installation path as needed)
bash Miniforge3-{Version}.sh -b -p /path/to/folder/tools/miniconda3
# Initialize conda for your shell:
/path/to/folder/tools/miniconda3/bin/conda init bash
```

Close and restart the shell for changes to take effect.

```
conda config --add channels conda-forge
conda install git mamba --yes
```

2. Install OPERA DISP-S1 tools to `opera_disp` environment

Download source code

```
cd /path/to/folder/tools
git clone --depth 1 --no-checkout --branch main --no-checkout https://github.com/OPERA-Cal-Val/OPERA_Applications.git && cd OPERA_Applications
git sparse-checkout set DISP/Timeseries && git checkout && cd DISP/Timeseries
#This enables you to only clone the folder of interest opera_disp within the entire repository
```

Create `opera_disp` environment and install pre-requisites

```
cd /path/to/folder/tools/OPERA_Applications/DISP/Timeseries/opera_disp
# create new environment
# install dependencies with mamba by using `environment.yml`
mamba env create -f environment.yml
# load the environnement disp
conda activate opera_disp
```

Source your installation

Create a file (*e.g.*: config.rc) for easy activation and loading of the paths to your files:

```
# creation of a empty file
touch /path/to/folder/tools/OPERA_Applications/DISP/Timeseries/opera_disp/config.rc
```

Add the following paths within the config.rc file:

```
##----------------------- OPERA DISP -----------------------##
# add repo tools to your path
export TOOL_DIR=/path/to/folder/tools
export PATH=${PATH}:${TOOL_DIR}/OPERA_Applications/DISP/Timeseries/opera_disp
export DISP_HOME=${TOOL_DIR}/OPERA_Applications/DISP/Timeseries/opera_disp
export PYTHONPATH=${PYTHONPATH}:${DISP_HOME}
```

Create an alias `load_disp` in `~/.bash_profile` file for easy activation, that call the config.rc file *e.g.*:

```
alias load_disp='conda activate opera_disp; source /path/to/folder/tools/OPERA_Applications/DISP/Timeseries/opera_disp/config.rc'
#Close and restart the terminal for changes to take effect
```

3. Update the `opera_disp` environment MintPy packages

Install MintPy from source

```
# Load your environnement and paths
load_disp
cd /path/to/folder/tools/OPERA_Applications/DISP/Timeseries/opera_disp
```

```
git clone https://github.com/insarlab/MintPy.git
python -m pip install -e MintPy
```

4. Prepare credentials or register for NASA Earthdata access

1. Register for an account with NASA Earthdata at https://urs.earthdata.nasa.gov/users/new
2. After creating the username and confirming your email, store your username/password in a `~/.netrc` file with the hostname `urs.earthdata.nasa.gov`:

```
machine urs.earthdata.nasa.gov
  login MYUSERNAME
  password MYPASSWORD
```

Troubleshooting Advice

If you encounter errors during usage, the most effective solution is to **"quit, re-open the terminal, and relaunch the Conda environment"**. This approach has successfully resolved the issue in all cases we've tested.

Test the installation

Run the following to test the installation:

```
# Load OPERA displacement module
load_disp

# Display help for the download script (try using 'python' if issues occur)
run1_download_DISP_S1_Static.py --h

# Display help for MintPy
smallbaselineApp.py -h
```

5. Available frames on OPERA AWS S3 bucket (OPERA DISP-S1 datasets are from 20160101 to 20241231):

```
#FrameID,   location,       reference_lalo,
08882,     Houston,        29.692 -095.635,
11115,     Central CA,     37.104 -121.651,
11116,     Central CA,     36.612 -121.064,
12640,     Florida,        29.056 -081.263,
18903,     Rosamond,       35.039 -118.006,
28486,     Oklahoma,       35.483 -098.971,
33039,     Hawaii,         19.450 -155.525,
33065,     Unimak Isl.,    54.831 -163.781,
36542,     Central CA,     36.516 -120.853,
42779,     Alaska,         61.550 -149.327,
25018,     Alaska,         65.117 -147.433,
08622,     New York,       40.703 -073.979,
09156,     South SF,       36.293 -121.403,
```

6. Run the OPERA data downloading script:

For example, here is a sample run for the Central Valley, California case study for descending Sentinel-1 track 042. The lastest preliminary version is v0.9.

For the Frame 11116, the size of the entire dataset of 300 interferograms is ~102Gb, ~340Mb for a file. By default, the script processes all available dates, which may require substantial storage and processing time. To reduce the dataset size, you can select a specific date range using the --startDate and --endDate arguments.

```
# Args:
# --frameID    OPERA frame number
# --version    OPERA dataset version
# --staticDir  Folder for static layers/metadata
# --geomDir    Folder for geometry files
# --dispDir    Folder for data
# --startDate  Start date (optional)
# --endDate    End date (optional)

run1_download_DISP_S1_Static.py \
      --frameID 11116 \
      --version 0.9 \
      --staticDir /path/to/work/folder/static_lyrs \
      --geomDir /path/to/work/folder/geometry \
      --dispDir /path/to/work/folder/data #\
     #--startDate 20170101
     #--endDate 20190101
```

7. Run the MintPy output script

For example, here is a sample run for the Central Valley, California case study for descending Sentinel-1 track 042.

```
## Example Command to Run `run2_prep_mintpy_opera.py`
# Args:
# -m   Folder for static layers/metadata
# -u   Folder with data (*.nc for all files)
# -g   Folder for geometry files
# -o   Folder for timeseries output
# --water-mask-file  Water mask file (auto-generated)
# --dem-file         DEM file (auto-generated)
# --ref-lalo         Spatial reference for timeseries
# --apply-mask       Apply mask (optional)
```

```
run2_prep_mintpy_opera.py \
        -m "/path/to/work/folder/static_lyrs" \
        -u "/path/to/work/folder/data/*.nc" \
        -g "/path/to/work/folder/geometry" \
        -o /path/to/work/folder/mintpy_output \
        --water-mask-file esa_world_cover_2021 \
        --dem-file glo_30 \
        --ref-lalo '36.612 -121.064' \
        --apply-mask
```

Note: `--apply-mask` applies the `recommended_mask` layer that is embedded within each of the DISP-S1 nominal product (i.e. *.nc) **on an epoch based**. The `recommended_mask` is the suggested mask to remove low quality pixels, where 0 indicates a bad pixel, 1 is a good pixel.

8. How to view the data?

In a terminal, you can visualize the timeseries.h5 newly created with the MintPy tools.

```
## Need help with the arguments: tsview.py -h
tsview.py /path/to/work/folder/mintpy_output/timeseries.h5 \
        -m
/path/to/work/folder/mintpy_output/recommended_mask90threshold.h5 \
```

Note: `recommended_mask90threshold.h5` is based on the time-series of `recommended_mask` layers (i.e. `recommended_mask.h5`). We picked the top 90% representing the "most reliable pixels in time" after normalizing the `recommended_mask` against the total number of epoch/dataset.