Tutorial on CVM Tools

Manochehr Bahavar

CVM Topical Workshop
May 13-14, 2024

GAGE
SAGE

Operated by EarthScope Consortium

Presentation Overview

- Fundamentals
- Current State
- Future Plan
- Hands-on installation

Presentation Overview

- Fundamentals

Formats

Format	Type	Description
GeoCSV	ASCII	An extension of the *human readable*, CSV (Comma-Separated Values) format with extensive metadata (more)
netCDF	Network Common Data Form (netCDF) , Version 4 Classic	Provides the netCDF-4 performance benefits while using the classic data model. Supports compression to reduce the file size (more)
Zarr or HDF5	TBD	

Metadata

- Every file, irrespective of its format, will be equipped with comprehensive metadata, adhering to the CF Metadata Conventions.
- This standardization guarantees seamless data management and interoperability across various platforms and tools.
- With uniform metadata, users can efficiently search, share, and analyze data, enhancing collaboration and decision-making processes.

Enhanced Coordinate Support

The proposed GeoCSV and netCDF file formats support both projected coordinate variables (x and y) and the geographic latitude and longitude variables. This would facilitate model query and visualization based on (x,y) or (longitude, latitude) coordinates.

GeoCSV Structure - Example

Stephenson's
casc1.6_velmdl model

-10800	4467300	60000	8438	4801
-10300	4467300	60000	8438	4801
-9800	4467300	60000	8438	4801
-9300	4467300	60000	8438	4801
-8800	4467300	60000	8438	4801
-8300	4467300	60000	8438	4801
-7800	4467300	60000	8438	4801
-7300	4467300	60000	8438	4801
-6800	4467300	60000	8438	4801
-6300	4467300	60000	8438	4801

Data

```
# dataset: GeoCSV2.0
# created: 2023-12-24 05:27:59 UTC (netCDF_2_GeoCSV.py)
# netCDF_file: casc1.6_velmdl_no-trans.nc
# delimiter: |
# global_title: P- and S-wave Seismic Velocity Models Incorporating the Cascadia Subduction Zone
# global_id: casc1.6_velmdl
# global_summary: P- and S-wave Seismic Velocity Models Incorporating the Cascadia Subduction Zone for 3D Earthquake Ground Motion simulations- Upd
# global_reference: Stephenson, 2017
# global_references: https://www.sciencebase.gov/catalog/item/59f1e68be4b0220bbd9dd4b4
# global_keywords: seismic, tomography, shear wave, s wave, elastic waveform, Cascadia Subduction Zone
# global_Conventions: CF-1.0
# global_Metadata_Conventions: Unidata Dataset Discovery v1.0
# global_creator_name: IRIS EMC
# global_creator_url: http://ds.iris.edu/ds/products/emc/
# global_creator_email: product@iris.edu
# global_institution: IRIS DMC
# global_acknowledgment: Model from USGS Open-File Report 2007-1348
# global_history: 2023-12-24 05:27:59 UTC Converted to GeoCSV by netCDF_2_GeoCSV.py ,v2022.237 from casc1.6_velmdl_no-trans.nc; Created by GeoCSV_2
# global_comment: model converted to netCDF by IRIS EMC
# global_geospatial_northing_min: 4467300
# global_geospatial_northing_max: 5540300
# global_geospatial_northing_units: m
# global_geospatial_northing_resolution: 500
# global_geospatial_easting_min: -10800
# global_geospatial_easting_max: 643200
# global_geospatial_easting_units: m
# global_geospatial_easting_resolution: 500
# global_geospatial_vertical_min: 0
# global_geospatial_vertical_max: 60000
# global_geospatial_vertical_units: m
# global_geospatial_vertical_positive: down
# global_source: Converted from casc1.6_velmdl.cdl
# y_column: northing
# y_axis: Y
# y_variable: northing
# y_dimensions: 1
# y_global_geospatial_min: 4467300
# y_global_geospatial_max: 5540300
# y_global_geospatial_units: m
# y_global_geospatial_resolution: 500
# x_column: easting
# x_axis: X
# x_variable: easting
# x_dimensions: 1
# x_global_geospatial_min: -10800
# x_global_geospatial_max: 643200
# x_global_geospatial_units: m
# x_global_geospatial_resolution: 500
# z_column: depth
# z_variable: depth
# z_dimensions: 1
# latitude_column: latitude
# latitude_variable: latitude
# latitude_dimensions: 2
# latitude_long_name: latitude
# latitude_units: degrees_north
# latitude_min: 40.0
# latitude_max: 50.0
# longitude_column: longitude
# longitude_variable: longitude
# longitude_dimensions: 2
# longitude_long_name: Longitude
# longitude_units: degrees_east
# longitude_min: -122.00
# longitude_max: -117.00
# vs_column: vs
# vs_variable: vs
# vs_dimensions: 3
# vs_FillValue: 99999.0
# vs_long_name: Shear Velocity
# vs_display_name: S Velocity (m/s)
# vs_units: m.s-1
# vs_missing_value: 99999.0
# vs_coordinates: longitude latitude
# vp_column: vp
# vp_variable: vp
# vp_dimensions: 3
# vp_FillValue: 99999.0
# vp_long_name: P Velocity
# vp_display_name: P Velocity (m/s)
# vp_units: m.s-1
# vp_missing_value: 99999.0
# vp_coordinates: longitude latitude
easting|northing|longitude|latitude|depth|vp|vs
-10800|4467300|-128.9998|40.2004|60000|8438|4801
-10300|4467300|-128.9940|40.2007|60000|8438|4801
-9800|4467300|-128.988|40.2010|60000|8438|4801
-9300|4467300|-128.9823|40.2013|60000|8438|4801
-8800|4467300|-128.9765|40.2016|60000|8438|4801
-8300|4467300|-128.9706|40.2019|60000|8438|4801
-7800|4467300|-128.9648|40.2022|60000|8438|4801
```

Metadata

Converted Files

Sample files are available under the [Google Drive Folder]

Model	Author(s)	Description	Files	Coordinates
casc1.6_velmdl	William Stephenson	P- and S-wave velocity models incorporating the Cascadia subduction zone for 3D earthquake ground motion simulations, version 1.6	Original: casc1.6_velmdl.txt (.txt 16G, .txt.gz 1.3G)	UTM
			GeoCSV: casc1.6_velmdl_r0.1.csv (.csv 15G, .csv.gz 3.1G)	UTM/Geographic
			Compressed netCDF: casc1.6_velmdl_r0.1.nc (1.3G)	UTM/Geographic
Cascadia-ANT+RF-Delph2018	Jonathan Delph, Alan Levander, and Fenglin Niu	3D vertical shear-wave velocity model of the Cascadian forearc from the joint inversion of ambient noise dispersion and receiver functions	Original: Cascadia_ANT+RF_Delph2018.txt (.txt 1.6M, .txt.gz 401K)	Geographic
			GeoCSV: Cascadia-ANT+RF-Delph2018.r0.1.csv (.csv 3.5M, .csv.gz 1.0M)	Geographic/UTM
			Compressed netCDF: Cascadia-ANT+RF-Delph2018_r0.1.nc (264K)	Geographic/UTM

Exploring the netCDF Files Using Xarray

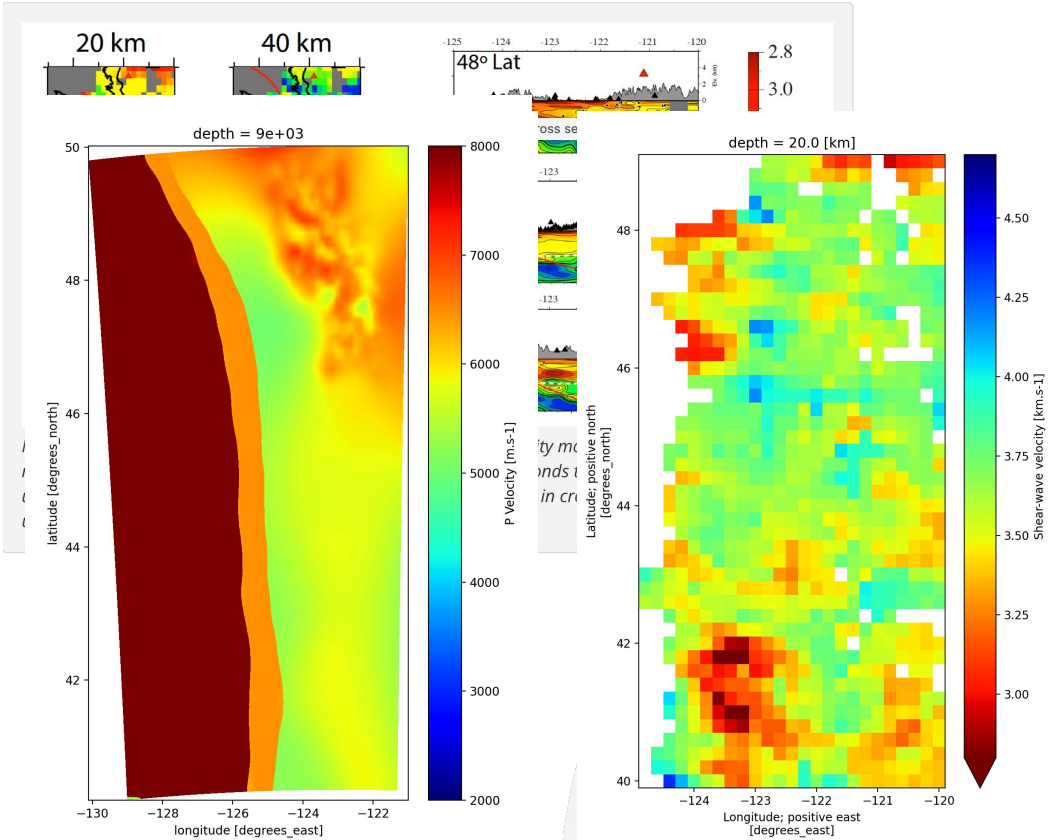
```
filename = "Cascadia-ANT+RF-Delph2018_r0.1.nc"
depth = 20
variable = "vs"
divider = 80 * "="
vmin = 2.8
vmax = 4.7
figure_size = (6, 10)
cmap = "jet_r"
x = "longitude"
y = "latitude"
auxiliary_x = "easting"
auxiliary_y = "northing"
y_min = 46
x_min = -122

# Read the netCDF content to an xarray dataset and display its content.
ds = xr.open_dataset(filename)
print(f"\n\n\n{filename} dataset content:\n{divider}\n{ds}")

# Extract the designated variable's dataset and display its content.
ds_var = ds[variable]
print(f"\n\n{variable} variable's dataset content:\n{divider}\n{ds_var}")

# Extract the horizontal (depth) slice for the give depth.
ds_var_depth = ds_var.where(ds_var.depth == depth, drop=True)
ds_var_depth.plot(figsize=figure_size, cmap=cmap, vmin=vmin, vmax=vmax)
plt.show()

# Plot the same slice in the auxiliary coordinates.
ds_var_depth.plot(
    figsize=figure_size, cmap=cmap, vmin=vmin, vmax=vmax, x=auxiliary_x, y=auxiliary_y
)
plt.show()
```



Presentation Overview

- Fundamentals
- Current State

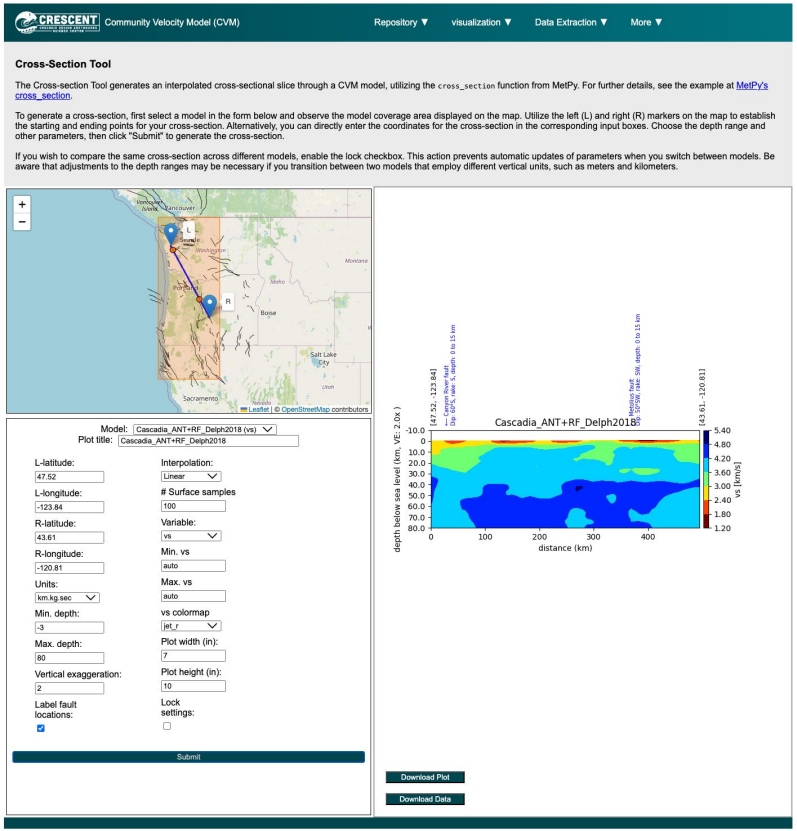
Tools Developed

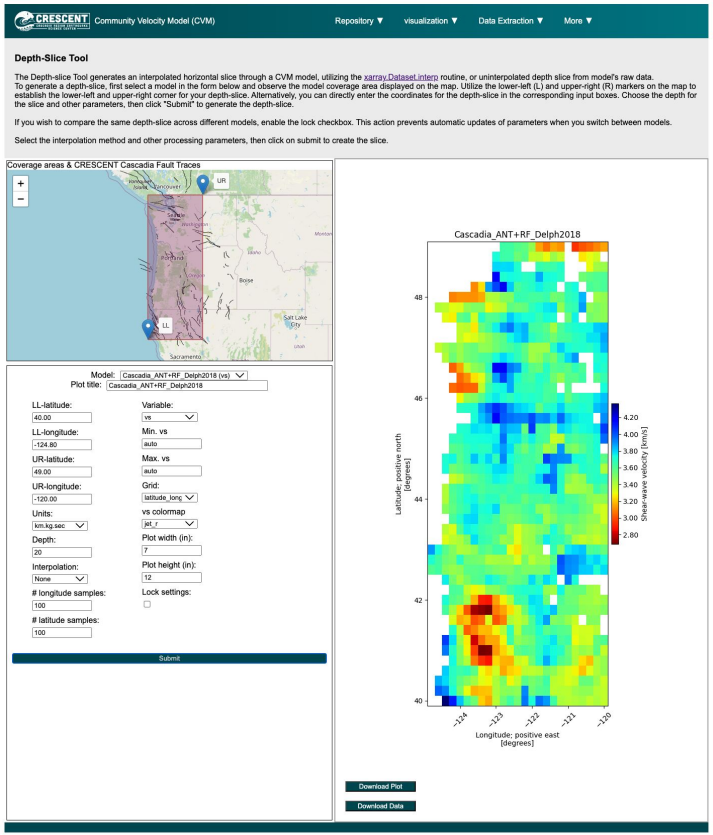
GAGE SAGE Operated by EarthScope Consortium

[GitHub Repository]
[Installation notes]

Tool	Application	Description
cvm_writer	Metadata and data storage	Reads a text model metadata file and outputs the corresponding model metadata in GeoCSV and JSON formats. If a CSV model data file is also provided, then the output could also include the model in GeoCSV and/or netCDF formats.
netcdf_to_geocsv	Format conversion	Convert a CVM netCDF file to GeoCSV and/or output its metadata in JSON
cvm_slicer	Data extraction and plotting	To extract data from a CVM netCDF file. Users can interactively inspect the metadata, slice the data, plot, and save the sliced data along the existing coordinate planes and for vertical cross-sections in arbitrary direction.
simple_plotter.py	Plotting	A simple Python code to extract and plot depth slices from the CVM files in netCDF format. This tool is mainly intended for checking the generated netCDF files.

CVM Tools Online





Presentation Overview

- Fundamentals
- Current State
- Future Plan

Future Plan

- **Current tools**
- **Support for HDF5**
- **More format conversion tools:**
  - NonLinLoc
  - GeoModelGrids
  - SeisSol

Presentation Overview

- Fundamentals
- Current State
- Future Plan
- Hands-on installation

GitHub repository folder via Google Drive link

https://drive.google.com/drive/folders/1GcEtjm-T9qiSy5VRTCxN-STjtuYrC155?usp=share_link

Install the required packages

- Python 3
- Python modules listed in `requirements.txt`
    - Install these modules with `pip install -r requirements.txt`

This package has been tested under Python 3.12.0 on macOS 14.2.1. It may work with older Python 3 versions.

cvm_writer - get started

Input: text model metadata file ⇨ **output**: model metadata in GeoCSV and JSON formats

NOTE: Templates for the metadata files are available under the *template* folder. You can use either *metadata_template_detailed.txt* or *metadata_template.txt* for model's metadata. The only difference is that the *metadata_template_detailed.txt* has full description for the model parameters.

```
# Rules:
# Blank lines and lines starting with # are ignored
# Each parameter should be defined on a new line with key=value format (quotes are not required)
# Lines starting with "-" indicate a new parameter definition
# Lines starting with ">" indicate a new parameter grouping.
#    - Each parameter group should contain one or more parameter definition (key=value) on a new line
#    - The leading and trailing blank spaces are ignored
# Lines starting with ">>" indicate a new parameter sub-grouping.
#    - Each parameter sub-group should be nested under a parameter group
#    - Each parameter sub-group should contain one or more parameter definition (key=value) on a new line
#    - The leading and trailing blank spaces are ignored

> delimiter
   data =
   geocsv = |

> model
    model = Cascadia_ANT+RF_Delph2018
    title = 3D vertical shear-wave velocity model of the Cascadian forearc from the joint inversion of ambient noise dispersion and receiver functions
    summary = Cascadia_ANT+RF_Delph2018 was created from the joint inversion of ambient noise Rayleigh waves dispersion measurements (8-50
seconds)....
    reference = Delph, Levander, and Niu (2018)
    reference_pid = doi:10.1029/2018gl079518
    data_revision = r0.1
    version = v0.0
```

cvm_writer - metadata output example

Input: text model metadata file ⇨ **output**: model metadata in GeoCSV and JSON formats

NOTE: Templates for the metadata files are available under the *template* folder. You can use either *metadata_template_detailed.txt* or *metadata_template.txt* for model's metadata. The only difference is that the *metadata_template_detailed.txt* has full description for the model parameters.

1. cd to the model directory *sample-files/Cascadia-ANT+RF-Delph2018*

2. *Cascadia-ANT+RF_meta.txt* is the metadata file

3. *./../src/**cvm_writer.py -m** Cascadia-ANT+RF_meta.txt **-o** Cascadia-ANT+RF-test **-t** metadata*
   -m Cascadia-ANT+RF_meta.txt is *the metadata file*
   -o Cascadia-ANT+RF-test is the output filename (a postfix of *_metada* will be added to the filename)
   -t metadata tells the code that we only want the metadata files.

4. After running the above command, you should see two new files in your model directory:
   Metadata JSON file: Cascadia-ANT+RF-test_metadata.json
   Metadata GeoCSV file: Cascadia-ANT+RF-test_metadata.csv

cvm_writer - metadata output example

Metadata GeoCSV file: Cascadia-ANT+RF-test_metadata.csv

```
# dataset: GeoCSV 2.0
# delimiter: |
# global_model: Cascadia_ANT+RF_Delph2018
# global_id: Cascadia_ANT+RF_Delph2018
# global_title: 3D vertical shear-wave velocity model of the Cascad
# global_summary: Cascadia_ANT+RF_Delph2018 was created from the jo
adaptive CCP-derived receiver functions (see Delph et al., 2015, 20
# global_reference: Delph, Levander, and Niu (2018)
# global_reference_pid: doi:10.1029/2018gl079518
# global_data_revision: r0.1
# global_version: v0.0
# global_Conventions: CF-1.0
# global_Metadata_Conventions: Unidata Dataset Discovery v1.0
# global_author_name: Jonathan R. Delph
# global_author_email: jdelph@purdue.edu
# global_author_institution: Purdue University
# global_author_url: https://www.eaps.purdue.edu/delph/
# global_geospatial_lon_min: -124.8
# global_geospatial_lon_max: -120.0
# global_geospatial_lon_units: degrees_east
# global_geospatial_lon_resolution: 0.2
# global_geospatial_lat_min: 40.0
# global_geospatial_lat_max: 49.0
# global_geospatial_lat_units: degrees_north
# global_geospatial_lat_resolution: 1.0
# global_geospatial_vertical_min: -3.0
# global_geospatial_vertical_max: 80.0
# global_geospatial_vertical_units: km
# global_geospatial_vertical_positive: down
```

Metadata JSON file: Cascadia-ANT+RF-test_metadata.json

```
{
    "model": "Cascadia_ANT+RF_Delph2018",
    "id": "Cascadia_ANT+RF_Delph2018",
    "title": "3D vertical shear-wave velocity model of the Cascadian forearc f
    "summary": "Cascadia_ANT+RF_Delph2018 was created from the joint inversion
aptive CCP-derived receiver functions (see Delph et al., 2015, 2017 for detail
    "reference": "Delph, Levander, and Niu (2018)",
    "reference_pid": "doi:10.1029/2018gl079518",
    "data_revision": "r0.1",
    "version": "v0.0",
    "Conventions": "CF-1.0",
    "Metadata_Conventions": "Unidata Dataset Discovery v1.0",
    "author_name": "Jonathan R. Delph",
    "author_email": "jdelph@purdue.edu",
    "author_institution": "Purdue University",
    "author_url": "https://www.eaps.purdue.edu/delph/",
    "geospatial_lon_min": -124.8,
    "geospatial_lon_max": -120.0,
    "geospatial_lon_units": "degrees_east",
    "geospatial_lon_resolution": 0.2,
    "geospatial_lat_min": 40.0,
    "geospatial_lat_max": 49.0,
    "geospatial_lat_units": "degrees_north",
    "geospatial_lat_resolution": 1.0,
    "geospatial_vertical_min": -3.0,
    "geospatial_vertical_max": 80.0,
    "geospatial_vertical_units": "km",
    "geospatial_vertical_positive": "down",
```

cvm_writer - model output example

- **Input**: text model metadata + CSV model data files ⇨ **output**: model in GeoCSV and/or netCDF formats

1. The *Cascadia_ANT+RF_Delph2018.txt.gz* file is the raw data file for the model.

2. Unzip it and copy it to a file (for example *Cascadia-ANT+RF_data.txt*) for testing. **All model data files need to have a header as their first line that identifies the columns in the file using the same delimiter as data between column names:**
   longitude latitude depth vs

   NOTE: the delimiter parameter should have been defined in the metadata file

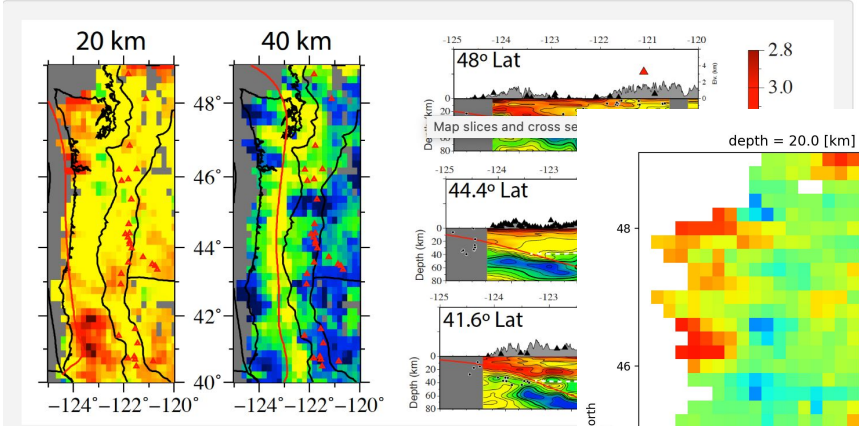
3. Create the netCDF model files by running:
   ../../src/**cvm_writer.py -m** Cascadia-ANT+RF_meta.txt **-o** Cascadia-ANT+RF-test **-d** Cascadia-ANT+RF_data.txt **-t** netcdf
   -d Cascadia-ANT+RF_data.txt is the model data
   -t netcdf we want the output to be in netCDF
   NOTE: we could have used *-t geocsv* to have the output as GeoCSV file or we could have combined them all and have *-t metadata,netcdf,geocsv* to get all the outputs

   Model netCDF file: Cascadia-ANT+RF-test.nc

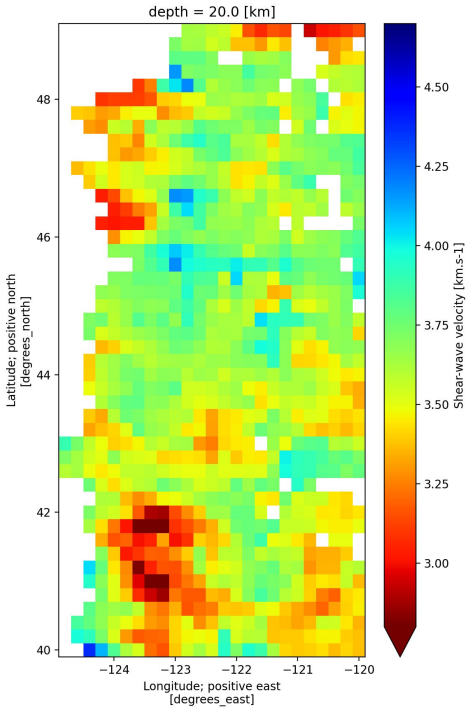
Working with netCDF Files Using Xarray

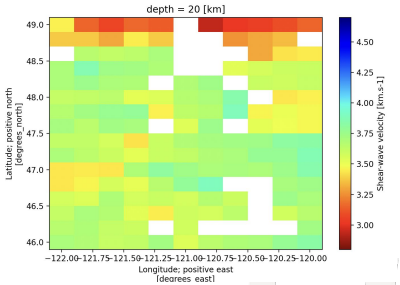
Needs a configuration file (provided):
simple_plotter_prop.py

```
filename = "Cascadia-ANT+RF-Delph2018.r0.1.nc"
depth = 20
variable = "vs"
vmin = 2.8
vmax = 4.7
figure_size = (6, 10)
figure_size_s = (4, 4)
cmap = "jet_r"
x = "longitude"
y = "latitude"
auxiliary_x = "easting"
auxiliary_y = "northing"
y_min = 46
x_min = -122
```



Map slices (left) and cross sections (right) through shear wave velocity mo model of (McCrory et al., 2012). While line in cross sections corresponds t uncertain). Red triangles: Holocene volcanic centers; Black triangles in cr used in receiver function analysis.





*../../src/**simple_plotter.py -i** Cascadia-ANT+RF-test.nc*
Slice at the depth of 20km in moel coordinates
Slice at the depth of 20km in auxiliary coordinates
Slice at the depth of 20km of a subsection

netcdf_to_geocsv

Convert a netCDF CVM file to GeoCSV format and optionally output its metadata as JSON

Input: netCDF model file ⇨ **output**: model in GeoCSV and metadata JSON formats

NOTE: The output will have the same name as the input file

1. cd to the model directory *sample-files/Cascadia-ANT+RF-Delph2018*

2. Convert the netCDF file from the previous slides to GeoCSV:
   *../../src/**netcdf_to_geocsv.py -i** Cascadia-ANT+RF-test.nc **-g** true **-m** true*
   -i Cascadia-ANT+RF-test.nc name of the input file to convert
   -g true sets the GeoCSV conversion flag to True
   -m true sets the metadata output flag to True
   -o conversion_test is the output file name

3. After running the above command, you should see two new files in your model directory:
   GeoCSV file: Cascadia-ANT+RF-test.csv
   JSON metadata file: Cascadia-ANT+RF-test.json

Netcdf_to_geocsv - output

GeoCSV file:
Cascadia-ANT+RF-test.csv

```
# dataset: GeoCSV 2.0
# delimiter: |
# created: 2024-03-21T12:28:30 UTC (cvm_convert.py)
# netcdf_file: Cascadia-ANT+RF-test.nc
# global_model: Cascadia_ANT+RF_Delph2018
# global_id: Cascadia_ANT+RF_Delph2018
# global_title: 3D vertical shear-wave velocity model of the Cascadian forearc from the joint inversion of ambient noise dispersion ar
# global_summary: Cascadia_ANT+RF_Delph2018 was created from the joint inversion of ambient noise Rayleigh waves dispersion measuremen
Delph et al., 2015, 2017 for details of methodology; Delph et al., 2018 for details of this model).
# global_reference: Delph, Levander, and Niu (2018)
# global_reference_pid: doi:10.1029/2018gl079518
# global_data_revision: r0.1
# global_version: v0.0
# global_Conventions: CF-1.0
# global_Metadata_Conventions: Unidata Dataset Discovery v1.0
# global_author_name: Jonathan R. Delph
# global_author_email: jdelph@purdue.edu
# global_author_institution: Purdue University
# global_author_url: https://www.eaps.purdue.edu/delph/
# global_repository_name: EMC
# global_repository_institution: EarthScope
# global_repository_pid: doi:10.17611/dp/cascadiaantrfd2018
# global_geospatial_lon_min: -124.8
# global_geospatial_lon_max: -120.0
# global_geospatial_lon_units: degrees_east
# global_geospatial_lon_resolution: 0.2
# global_geospatial_lat_min: 40.0
# global_geospatial_lat_max: 49.0
# global_geospatial_lat_units: degrees_north
# global_geospatial_lat_resolution: 1.0
# global_geospatial_vertical_min: -3.0
# global_geospatial_vertical_max: 80.0
# global_geospatial_vertical_units: km
# global_geospatial_vertical_positive: down
# global_keywords: seismic,Rayleigh waves dispersion,shear wave,s wave,Cascadian forearc,ambient noise dispersion,receiver functions
# global_acknowledgment: Model was provided by Jonathan R. Delph of Department of Earth, Environmental and Planetary Sciences, Rice Un
# global_history: [2024-01-03] Converted to netCDF via CRESCENT data tools.
# global_comments: CRESCENT CVM tools development project.
# global_grid_mapping_name: latitude_longitude
# global_utm_zone: 10
# global_ellipsoid: WGS84
# depth_variable: depth
# depth_dimensions: 1
# depth_column: depth
# depth_positive: down
# depth_standard_name: depth
# depth_long_name: depth below sea level
# depth_units: km
# latitude_variable: latitude
# latitude_dimensions: 1
# latitude_column: latitude
# latitude_long_name: Latitude; positive north
# latitude_units: degrees_north
# latitude_standard_name: latitude
# longitude_variable: longitude
# longitude_dimensions: 1
# longitude_column: longitude
# longitude_standard_name: longitude
# longitude_long_name: Longitude; positive east
# longitude_units: degrees_east
# easting_variable: easting
# easting_dimensions: 2
# easting_column: easting
# easting_standard_name: easting
# easting_long_name: easting; UTM
# easting_units: m
# northing_variable: northing
# northing_dimensions: 2
# northing_column: northing
# northing_long_name: northing; UTM
# northing_units: m
# northing_standard_name: northing
# vs_variable: vs
# vs_dimensions: 3
# vs_column: vs
# vs_long_name: Shear-wave velocity
# vs_display_name: S Velocity (km/s)
# vs_units: km.s-1
# vs_grid_mapping: latitude_longitude
depth|latitude|longitude|easting|northing|vs
-3|40.0|-124.8|346348.0967416621|4429308.903691186|nan
-3|40.0|-124.6|363421.3590923465|4428983.187625435|nan
-3|40.0|-124.4|380494.3291546299|4428695.814829797|nan
-3|40.0|-124.2|397567.0433531772|4428446.776403481|nan
-3|40.0|-124.0|414639.5381572299|4428236.06463309|nan
-3|40.0|-123.8|431711.8500742161|4428063.672992349|nan
-3|40.0|-123.6|448784.0156431892|4427929.596141886|nan
-3|40.0|-123.4|465856.07142943953|4427833.829929045|nan
```

cvm_slicer.py - get started

Interact with the netCDF file, visualize and extract data

- **Input**: netCDF model file ⇨ **output**: model metadata, subset of model data, plots

1. cd to the model directory *sample-files/Cascadia-ANT+RF-Delph2018*
2. Start interacting with the netCDF file from previous slide:
   *../../src/**cvm_slicer.py -v -i** Cascadia-ANT+RF-test.nc*

   -v to run in the verbose mode and provide more information on the required parameters.
   NOTE: You can remove this tag to run the tool in a less verbose mode.
   -i Cascadia-ANT+RF-test.nc the netCDF input file

```
cvm_slicer
----------
Tool for interactively extracting data from a CVM netCDF file.
User can inspect the metadata, slice the data, plot, and save
the sliced data. Slicing can be performed along the existing
coordinate planes or as an interpolated cross-sectional slice through gridded data.

Loaded Cascadia-ANT+RF-test.nc and it is a netCDF file


The available options are:
        meta - to view file's metadata
        ranges - to display value ranges for variables
        subset - to subset the data
        ----------
        help
        exit
select option [meta, ranges, subset, help, exit]?
```

cvm_slicer.py - ranges

Interact with the netCDF file, visualize and extract data

*../../src/**cvm_slicer.py -v -i** Cascadia-ANT+RF-test.nc*

select option [meta, ranges, subset, help, exit]? meta

```
[Metadata] Global attributes:

                geospatial_lon_min: -124.8
                geospatial_lon_max: -120.0
                geospatial_lon_units: degrees_east
                geospatial_lon_resolution: 0.2
                geospatial_lat_min: 40.0
                geospatial_lat_max: 49.0
                geospatial_lat_units: degrees_north
                geospatial_lat_resolution: 1.0
                geospatial_vertical_min: -3.0
                geospatial_vertical_max: 80.0
                geospatial_vertical_units: km
                geospatial_vertical_positive: down

[Metadata] Coordinate Variables:
                depth:
        positive: down
        standard_name: depth
        long_name: depth below sea level
        units: km
        Values:
 -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
        ...
 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80

                latitude:
        long_name: Latitude; positive north
        units: degrees_north
        standard_name: latitude
.
.
.
```

select option [meta, ranges, subset, help, exit]?

cvm_slicer.py - ranges

Interact with the netCDF file, visualize and extract data

*../../src/**cvm_slicer.py -v -i** Cascadia-ANT+RF-test.nc*

select option [meta, ranges, subset, help, exit]? ranges

Ranges:

depth: -3.00 to 80.00 km
latitude: 40.00 to 49.00 degrees_north
longitude: -124.80 to -120.00 degrees_east
easting: 346348.10 to 756099.65 m
northing: 4427757.22 to 5431792.86 m
vs: 1.02 to 5.86 km.s-1

select option [meta, ranges, subset, help, exit]?

cvm_slicer.py - subset

Interact with the netCDF file, visualize and extract data

*../../src/**cvm_slicer.py -v -i** Cascadia-ANT+RF-test.nc*

select option [meta, ranges, subset, help, exit]? subset

You can subset the data as a:
- volume - a subvolume of data
- slice - a slice along a coordinate axis
- xsection - a vertical slice in an arbitrary direction

- back - takes you to the previous step
- exit

select [volume, slice, xsection, back, exit]?

cvm_slicer.py - depth slice

select [volume, slice, xsection, back, exit]? slice

A slice cuts the model along one of the coordinate axis.

direction - direction of the slice, the coordinate to cut the model along

---

back - takes you to the previous step

exit

direction [depth, latitude, longitude, back, exit]?depth

slice depth [-3 to 80, back, exit]?20

Select slice limits in the latitude direction.

limits - provide minimum,maximum or press enter to accept the full range default

---

back - takes you to the previous step

exit

latitude limits [default values (40.0, 49.0), back, exit]:
longitude limits [default values (-124.8, -120.0), back, exit]:

```
The sliced data summary:
<xarray.Dataset> Size: 28kB
Dimensions:    (latitude: 46, longitude: 25)
Coordinates:
    depth      int64 8B 20
  * latitude   (latitude) float64 368B 40.0 40.2 40.4 40.6 ... 48.6 48.8 49.0
  * longitude  (longitude) float64 200B -124.8 -124.6 -124.4 ... -120.2 -120.0
    easting    (latitude, longitude) float64 9kB 3.463e+05 ... 7.194e+05
    northing   (latitude, longitude) float64 9kB 4.429e+06 ... 5.432e+06
Data variables:
    vs         (latitude, longitude) float64 9kB nan nan 4.367 ... 3.097 3.152
```

cvm_slicer.py - depth slice

What to do with the slice.

plot2d - a 2D plot of ['latitude', 'longitude']
plot3d - a 3D plot of ['latitude', 'longitude'] and the model variable on the 3rd axis.
The plot is interactive and can be rotated.
gmap - a 2D plot of ['latitude', 'longitude'] in geographical coordinate system
cmap - change the color map for the plots
save - save the slice data
---
back - takes you to the previous step
exit

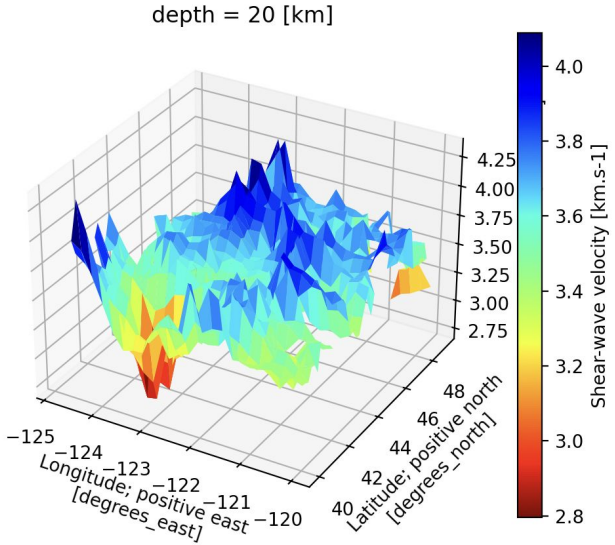
Action [plot2d, plot3d, gmap, cmap, save, back, exit]:

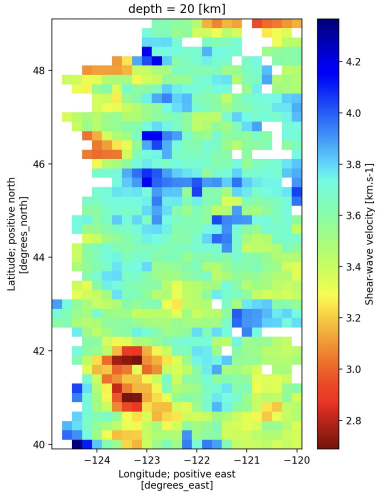
cvm_slicer.py - gmap

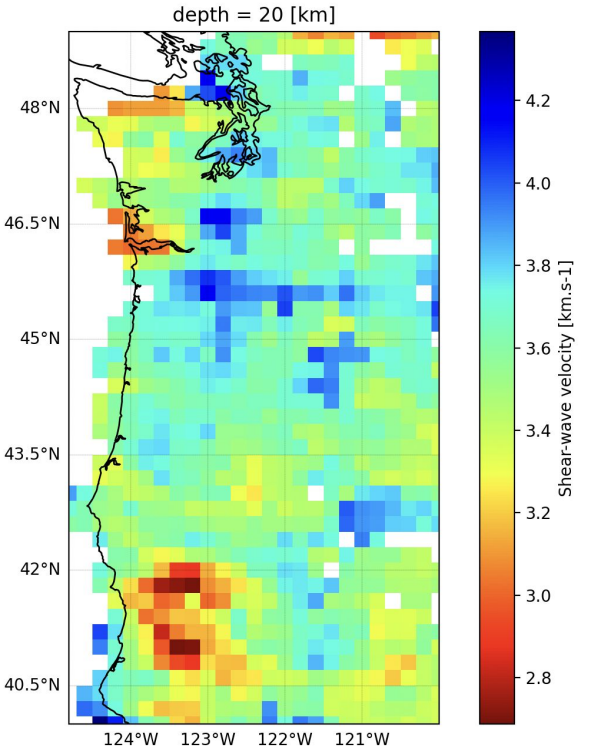
Action [plot2d, plot3d, gmap, cmap, save, back, exit]:gmap

Action [plot2d, plot3d, gmap, cmap, save, back, exit]:plot2d

Action [plot2d, plot3d, gmap, cmap, save, back, exit]:plot3d







cvm_slicer.py

GAGE SAGE Operated by EarthScope Consortium

Many options to explore

Questions?

Your feedback is highly appreciated. For questions, comments and suggestions please contact Manoch:

manochehr.bahavar@earthscope.org