IMI

Release 0.1.0

TODO

Mar 20, 2024

GETTING STARTED

Important: Contributions (e.g., suggestions, edits, revisions) would be greatly appreciated. See *editing this guide* and our *contributing guidelines*. If you find something hard to understand, let us know!

The Integrated Methane Inversion (IMI) workflow is a cloud-computing tool for quantifying methane emissions by inversion of satellite observations from the TROPOspheric Monitoring Instrument (TROPOMI). It uses GEOS-Chem as forward model for the inversion and infers methane emissions at 25×25 km^2 resolution.

This site provides instructions for using the IMI, including launching an AWS compute instance, configuring and running an inversion, and analyzing the results with a ready-made jupyter notebook.

Some instructions are specific to the Amazon Web Services (AWS) cloud, but the IMI can also be run on a local compute cluster.

QUICK START GUIDE

1.1 1. Register with us

We encourage new users to email the IMI team at **integrated-methane-inversion@g.harvard.edu** with a description of your project and the organization you are affiliated with. Knowing our user base helps us to prioritize new features and updates to the IMI. Additionally, registered users can contact us for support and will receive notifications of any critical bugfixes or new releases/features added to the IMI.

Template introduction email:

```
Hello IMI Team!

My name is <insert name here> and I am affiliated with <insert-organization>.
We work on <research-interests> and are interested in using the IMI to <insert-
↪application>.
Here is the link to our research page: <insert-link>. Please send us updates on any␣
↪future
releases or critical bugfixes to the IMI.
```

1.2 2. Create an Amazon Web Services (AWS) account

If you do not already have an AWS account, you'll need to sign up for one. Go to http://aws.amazon.com and click on "Create an AWS Account" in the upper-right corner:

You'll need to enter some basic personal information and a credit card number.

Running the IMI is relatively inexpensive (usually on the order of USD $10-$100). The cost depends on the length of the inversion period, the size of the inversion domain, how long you retain your compute instance after completing the inversion, and how you store the final results.

For more information on costs, see *Tips for Minimizing AWS costs*.

Note: Students can check out subsidized educational credits at https://aws.amazon.com/education/awseducate/.

1.3 3. Add S3 user permissions

Default input data for the IMI are stored in the Amazon Simple Storage Service (S3). These include TROPOMI methane data, default prior emission estimates, GEOS-Chem meteorological data, and boundary condition data.

The IMI will automatically fetch the data needed for your inversion, but to enable this data transfer you'll need to add S3 user permissions to your AWS account.
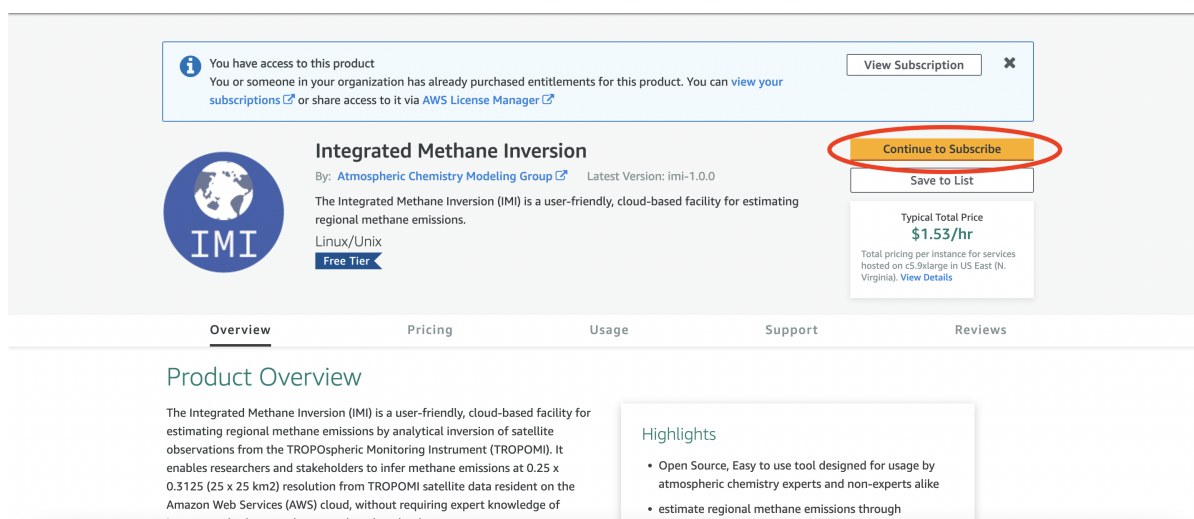
The easiest way to do this is to grant S3 access to an IAM role. Attaching the IAM role to a compute instance on the AWS Elastic Compute Cloud (EC2; Amazon's basic computing service) will give the EC2 instance full access to S3.

Instructions to create an IAM role with full S3 access are provided in the GEOS-Chem Cloud Documentation. For more information on IAM roles, check out the AWS Documentation.

1.4 4. Launch an instance with the IMI

Once you've setup S3 permissions on your AWS account, login to the AWS console and go to the AWS Marketplace IMI listing (listed for free). This image contains the latest version of the IMI including all required software dependencies on an Amazon Machine Image (AMI). An AMI fully specifies the software side of your virtual system, including the operating system, software libraries, and default data files.

On the listing page click "Continue to Subscribe".

On the following page click "Continue to Configuration".

Select desired region and IMI version and click "Continue to Launch". Choosing a region closer to your physical location will improve your network connectivity, but may result in increased costs compared to using the region where GEOS-Chem data are hosted (us-east-1, N.Virginia).

On the launch screen select "Launch through EC2" and then click launch.

Now it's time to specify the hardware for running your system. Hardware choices differ primarily in CPU and RAM counts.

You can select from a large number of instance types in the "Instance Type" section. The IMI will run more quickly with a higher number of CPUs.

Choose the c5.9xlarge instance type, which includes 36 CPU cores and 72GB of RAM. Depending on your use case you may choose a different instance type with more/less cores and memory.

Note: Note: new AWS users may encounter a limit on the number of CPUs they can allocate. To request a limit increase follow the steps outlined in the aws docs on how to calculate a vCPU limit increase.

In the next section you create, or select an existing, ssh key pair. This is equivalent to the password you enter to `ssh` to your local server. Click "Create new key pair". In the dialog box give your key pair a name (eg. imi_testing) and click "Create key pair". In the future, you can simply select your existing keypair from the dropdown menu.

The "Network Settings" section can be left as the defaults. Proceed to "Configure Storage" and select the size of your storage volume.

Note: Your storage needs will depend on the length of the inversion period, size of the inversion domain, and the inversion resolution. 100GB is generally sufficient for a 1-week inversion (such as for the Permian Basin), and 5 TB will likely be enough for a 1-year inversion.

Storage costs typically amount to USD $100 per month per TB of provisioned space. See our advice on selecting storage volume size to help minimize storage fees. And when your inversion is complete, consider *copying output data to S3* and *terminating your EC2 instance* to avoid continued storage fees.

Expand the "Advance Details" section and select the IAM role you created in *step 2* under "IAM Instance Profile". This ensures that your EC2 instance has access to S3 (for downloading TROPOMI data and GEOS-Chem input data). All other config settings in "Advanced Details" can be left as the defaults.

Then, after reviewing the summary, just click on the "Launch Instance" button.. Once launched, you can monitor the instance in the EC2-Instance console as shown below. Within one minute of initialization, "Instance State" should show "running" (refresh the page if the status remains "pending"):

You now have your own system running on the cloud! Note that you will be charged continuously while the instance is running, so make sure to do the *final tutorial step: shutdown the server* if you need to pause your work to avoid unnecessary compute charges.

1.5 5. Login to your instance

Select your instance and click on the "Connect" button (shown in the figure above) near the blue "Launch Instance" button to show this instruction page:

- On Mac or Linux, use the `ssh -i ...` command under "Example" to connect to the server in the terminal. Some minor changes are needed:

  (1) `cd` to the directory where your Key Pair is stored. People often put the key in `~/.ssh/` but any directory will do.

  (2) Use `chmod 400 your-key-name.pem` to change the key pair's permission (also mentioned in the above figure; this only needs to be done once).

  (3) Change the user name in the command from `root` to `ubuntu` so that the full command looks like `ssh -i "your-key-name.pem" ubuntu@ec2-##-###-##-##.compute-1.amazonaws.com`

- On Windows, you can install Git-BASH to emulate a Linux terminal. Simply accept all default options during installation, as the goal here is just to use Bash, not Git. Alternatively, you can use MobaXterm, Putty, Windows Subsystem for Linux (WSL), or PowerShell with OpenSSH. The Git-BASH solution should be the most painless, but these other options can work as well. Note: there is a bug on older versions of WSL that can prevent the `chmod` command from functioning.

Once you've followed the above instructions, you should see a "Welcome to Ubuntu" message indicating you've logged into your new EC2 instance.

Connect To Your Instance   ✕

I would like to connect with   ● A standalone SSH client
                                    ○ A Java SSH Client directly from my browser (Java required)

To access your instance:

1. Open an SSH client. (find out how to connect using PuTTY)
2. Locate your private key file (my-aws-key.pem). The wizard automatically detects the key you used
   to launch the instance.
3. Your key must not be publicly viewable for SSH to work. Use this command if needed:

   ```
   chmod 400 my-aws-key.pem
   ```

4. Connect to your instance using its Public DNS:

   ```
   ec2-54-236-245-121.compute-1.amazonaws.com
   ```

Example:

```
ssh -i "my-aws-key.pem" root@ec2-54-236-245-121.compute-1.amazonaws.com
```

Please note that in most cases the username above will be correct, however please ensure that
you read your AMI usage instructions to ensure that the AMI owner has not changed the default
AMI username.

1.6 6. Configure the IMI

Navigate to the IMI setup directory:

```
$ cd ~/integrated_methane_inversion
```

Open the `config.yml` file with vim (`vi`) or emacs:

```
$ emacs config.yml
```

This configuration file contains many settings that you can modify to suit your needs. See the *IMI configuration file page* for information on the different settings/options. Also see the *common configurations page*.

1.7 7. Run the IMI

After editing the configuration file, you can run the IMI by executing the following command:

```
$ sbatch run_imi.sh
```

The sbatch command runs the IMI and writes to the imi_output.log output file. You can track it's progress by using:

```
$ tail --follow imi_output.log
```

The IMI can take minutes to days to complete, depending on the configuration and EC2 instance type. You can safely disconnect from your instance during this time, but the instance must remain active in the AWS console.

Alternatively, you can *run the IMI with tmux* to obtain a small to moderate speed-up.

Note: We strongly recommend using the *IMI preview feature* before running an inversion.

1.8 8. Visualize results with Python

When your inversion is complete, you can use the visualization notebook provided with the IMI to quickly inspect the results.

First navigate to the inversion directory:

```
$ cd /home/ubuntu/imi_output_dir/{YourRunName}/inversion
```

You can use the `ls` command to view the contents of the directory, which will include several scripts, data directories, and netcdf output files, along with `visualization_notebook.ipynb`. For more information on the contents, see Contents of the inversion directory.

To set up and connect to a jupyter notebook server on AWS follow these short instructions. Once connected to the server, open `visualization_notebook.ipynb` and run its contents to display key inversion results including the state vector, prior and posterior emissions, TROPOMI data for the region/period of interest, averaging kernel sensitivities, and more.

1.9 9. Shut down the instance

When you are ready to end your session, right-click on the instance in the AWS EC2 console to get this menu:

There are two options for ending the session: "Stop" (temporary shutdown) or "Terminate" (permanent deletion):

- "Stop" will make the system inactive. You won't be charged for CPU time, but you will be charged a disk storage fee for the number of GB provisioned on your EC2 instance. You can restart the instance at any time and all files will be preserved. When an instance is stopped, you can also change its hardware type (right click on the instance -> "Instance Settings" -> "Change Instance Type").

- "Terminate" will completely delete the instance so you will incur no further charges. Unless you save the contents of your instance as an AMI or transfer the data to another storage service (like S3), you will lose all your data and software.

1.10 10. Store data on S3

S3 is our preferred cloud storage platform due to cost and ease of access.

You can use the `cp` command to copy your output files to an S3 bucket for long term storage:

```
$ aws s3 cp </path/to/output/files> s3://<bucket-name> --recursive
```

For more information on using `s3` check out our tips for exporting data to S3.

IMI CONFIGURATION FILE

This page documents settings in the IMI configuration file (`config.yml`).

2.1 General

RunName	Name for this inversion; will be used for directory names and prefixes.
isAWS	Boolean for running the IMI on AWS (`true`) or a local cluster (`false`).
UseSlurm	Boolean for running the IMI as a batch job with `sbatch` instead of interactively. Select `true` to run the IMI with `sbatch run_imi.sh`. Select `false` to run the IMI with `./run_imi.sh` (*via tmux*).
SafeMode	Boolean for running in safe mode to prevent overwriting existing files.

2.2 Period of interest

StartDate	Beginning of the inversion period in `YYYYMMDD` format (this date is included in the inversion, 0-24h UTC).
EndDate	End of the inversion period in `YYYYMMDD` format (this date is excluded from the inversion, 0-24h UTC).
SpinupMonths	Number of months for the spinup simulation.

2.3 Region of interest

LonMin	Minimum longitude edge of the region of interest (only used if `CreateAutomaticRectilinearStateVectorFile` is `true`).
LonMax	Maximum longitude edge of the region of interest (only used if `CreateAutomaticRectilinearStateVectorFile` is `true`).
LatMin	Minimum latitude edge of the region of interest (only used if `CreateAutomaticRectilinearStateVectorFile` is `true`).
LatMax	Maximum latitude edge of the region of interest (only used if `CreateAutomaticRectilinearStateVectorFile` is `true`).
NestedGrid	Boolean for using the GEOS-Chem nested grid simulation. Must be `true` for IMI regional inversions.
NestedRegion	Nesting domain for the inversion. Select `AF` for Africa, `AS` for Asia, `EU` for Europe, `ME` for the Middle East, `NA` for North America, `OC` for Oceania, `RU` for Russia, or `SA` for South America. For global met fields set this option to `""` See the GEOS-Chem horizontal grids documentation for details about the available nested-grid domains.

2.4 State vector

CreateAutomaticRectilinearStateVectorFile	Boolean for whether the IMI should automatically create a rectilinear state vector for the inversion. If `false`, a custom/pre-generated state vector netcdf file must be provided below.
nBufferClusters	Number of buffer elements (clusters of GEOS-Chem grid cells lying outside the region of interest) to add to the state vector of emissions being optimized in the inversion. Default value is `8`.
BufferDeg	Width of the buffer elements, in degrees; will not be used if `CreateAutomaticRectilinearStateVectorFile` is `false`. Default is 5 (~500 km).
LandThreshold	Land-cover fraction below which to exclude GEOS-Chem grid cells from the state vector when creating the state vector file. Default value is `0.25`.
OffshoreEmisThreshold	Offshore GEOS-Chem grid cells with oil/gas emissions above this threshold will be included in the state vector. Default value is `0`.

2.4.1 Clustering Options

For more information on using the clustering options take a look at the clustering options page.

ReducedDimensionStateVector	Boolean for whether to reduce the dimension of the statevector from the native resolution version by clustering elements. If `false` the native state vector is used with no dimension reduction.
ClusteringMethod	Clustering method to use for state vector reduction. (eg. "kmeans" or "mini-batch-kmeans")
NumberOfElements	Number of elements in the reduced dimension state vector. This is only used if `ReducedDimensionStateVector` is `true`.
ForcedNativeResolutionElements	xml list of of coordinates that you would like to force as native resolution state vector elements [lat, lon]. This is useful for ensuring hotspot locations are at the highest available resolution.

2.5 Custom/pre-generated state vector

These settings are only used if `CreateAutomaticRectilinearStateVectorFile` is `false`. Use them to *create a custom state vector file* from a shapefile in conjunction with the `statevector_from_shapefile.ipynb` jupyter notebook located at:

```
$ /home/ubuntu/integrated_methane_inversion/src/notebooks/statevector_from_shapefile.
→ipynb
```

StateVectorFile	Path to the custom or pre-generated state vector netcdf file. File will be saved here if generating it from a shapefile.
ShapeFile	Path to the shapefile.

Note: To setup a remote Jupyter notebook check out the quick start guide visualize results with python section.

2.6 Inversion

PriorError	Error in the prior estimates (1-sigma; relative). Default is `0.5` (50%) error.
ObsError	Observational error (1-sigma; absolute; ppb). Default value is 15 ppb error.
Gamma	Regularization parameter; typically between 0 and 1. Default value is `1.0`.
PrecomputedJacobian	Boolean for whether the Jacobian matrix has already been computed (`true`) or not (`false`). Default value is `false`.

2.7 Grid

Res	Resolution for inversion. Options are `"0.25x0.3125"` and `"0.5x0.625"`.
Met	Meteorology to use for the inversion. Options are `"geosfp"` (for `Res:  "0.25x0.3125"`) and `"merra2"` (for `Res:  "0.5x0.625"`).

2.8 Setup modules

These settings turn on/off (`true` / `false`) different steps for setting up the IMI.

SetupTemplateRundir	Boolean to create a GEOS-Chem run directory and modify it with settings from `config.yml`.
SetupSpinupRun	Boolean to set up a run directory for the spinup-simulation by copying the template run directory and modifying the start/end dates, restart file, and diagnostics.
SetupJacobianRuns	Boolean to set up run directories for N+1 simulations (one reference simulation, plus N sensitivity simulations for the N state vector elements) by copying the template run directory and modifying the start/end dates, restart file, and diagnostics. Output from these simulations will be used to construct the Jacobian.
SetupInversion	Boolean to set up the inversion directory containing scripts needed to perform the inverse analysis; inversion results will be saved here.
SetupPosteriorRun	Boolean to set up the run directory for the posterior simulation by copying the template run directory and modifying the start/end dates, restart file, and diagnostics.

2.9 Run modules

These settings turn on/off (`true` / `false`) different steps for running the inversion.

`RunSetup`	Boolean to run the setup script (`setup_imi.sh`), including selected setup modules above.
`DoSpinup`	Boolean to run the spin-up simulation.
`DoJacobian`	Boolean to run the reference and sensitivity simulations.
`DoInversion`	Boolean to run the inverse analysis code.
`DoPosterior`	Boolean to run the posterior simulation.

2.10 SLURM Resource Allocation

These settings are used to allocate resources (CPUs and Memory) to the different simulations needed to run the inversion. Note: some python scripts are also deployed using slurm and default to using the `SimulationCPUs` and `SimulationMemory` settings.

`RequestedTime`	Max amount of time to allocate to each sbatch job (eg. "0-6:00")
`SimulationCPUs`	Number of cores to allocate to each in series simulation.
`SimulationMemory`	Amount of memory to allocate to each in series simulation (in MB).
`JacobianCPUs`	Number of cores to allocate to each jacobian simulation (run in parallel).
`JacobianMemory`	Amount of memory to allocate to each jacobian simulation (in MB).
`SchedulerPartition`	Name of the partition(s) you would like all slurm jobs to run on (eg. "debug,huce_intel,seas_compute,etc").

2.11 IMI preview

`DoPreview`	Boolean to run the *IMI preview* (`true`) or not (`false`).
`DOFSThreshold`	Threshold for estimated DOFS below which the IMI should automatically exit with a warning after performing the preview. Default value `0` prevents exit.

2.12 Advanced settings: GEOS-Chem options

These settings are intended for advanced users who wish to modify additional GEOS-Chem options.

PerturbValue	Value to perturb emissions by in each sensitivity simulation. Default value is `1.5`.
UseEmisSF	Boolean to apply emissions scale factors derived from a previous inversion. This file should be provided as a netCDF file and specified in HEMCO_Config.rc. Default value is `false`.
UseOHSF	Boolean to apply OH scale factors derived from a previous inversion. This file should be provided as a netCDF file and specified in HEMCO_Config.rc. Default value is `false`.
HourlyCH4	Boolean to save out hourly diagnostics from GEOS-Chem. This output is used in satellite operators via post-processing. Default value is `true`.
PLANEFLIGHT	Boolean to save out the planeflight diagnostic in GEOS-Chem. This output may be used to compare GEOS-Chem against planeflight data. The path to those data must be specified in input.geos. See the planeflight diagnostic documentation for details. Default value is `false`.
GOSAT	Boolean to turn on the GOSAT observation operator in GEOS-Chem. This will save out text files comparing GEOS-Chem to observations, but has to be manually incorporated into the IMI. Default value is `false`.
TCCON	Boolean to turn on the TCCON observation operator in GEOS-Chem. This will save out text files comparing GEOS-Chem to observations, but has to be manually incorporated into the IMI. Default value is `false`.
AIRS	Boolean to turn on the AIRS observation operator in GEOS-Chem. This will save out text files comparing GEOS-Chem to observations, but has to be manually incorporated into the IMI. Default value is `false`.

2.13 Advanced settings: Local cluster

These settings are intended for advanced users who wish to (*run the IMI on a local cluster*).

OutputPath	Path for IMI runs and output.
DataPath	Path to GEOS-Chem input data.
DataPathTROPOMI	Path to TROPOMI input data.
CondaFile	Path to file containing Conda environment settings.
CondaEnv	Name of conda environment.
RestartDownload	Boolean for downloading an initial restart file from AWS S3. Default value is `true`.
RestartFilePrefix	Path to initial GEOS-Chem restart file plus file prefix (e.g. `GEOSChem.BoundaryConditions.` or `GEOSChem.Restart.`). The date string and file extension (`YYYYMMDD_0000z.nc4`) will be appended. This file will be used to initialize the spinup simulation.
RestartFilePreviewPrefix	Path to initial GEOS-Chem restart file plus file prefix (e.g. `GEOSChem.BoundaryConditions.` or `GEOSChem.Restart.`). The date string and file extension (`YYYYMMDD_0000z.nc4`) will be appended. This file will be used to initialize the preview simulation.
BCpath	Path to GEOS-Chem boundary condition files (for nested grid simulations).
BCversion	Version of TROPOMI smoothed boundary conditions to use (e.g. `v2023-04`). Note: this will be appended onto BCpath as a subdirectory.
PreviewDryRun	Boolean to download missing GEOS-Chem data for the preview run. Default value is `true`.
SpinupDryRun	Boolean to download missing GEOS-Chem data for the spinup simulation. Default value is `true`.
ProductionDryRun	Boolean to download missing GEOS-Chem data for the production (i.e. Jacobian) simulations. Default value is `true`.
PosteriorDryRun	Boolean to download missing GEOS-Chem data for the posterior simulation. Default value is `true`.
BCDryRun	Boolean to download missing GEOS-Chem data for the preview run. Default value is `true`.
PreviewDryRun	Boolean to download missing GEOS-Chem boundary condition files. Default value is `true`.

Note for *DryRun options: If you are running on AWS, you will be charged if your ec2 instance is not in the us-east-1 region. If running on a local cluster you must have AWS CLI enabled or you can modify the `./download_data.py` commands in `setup_imi.sh` to use `washu` instead of `aws`. See the GEOS-Chem documentation for more details.

IMI PREVIEW

The IMI preview feature allows users to estimate the quality and information content of a proposed inversion without actually performing the inversion.

Under the default configuration, the IMI performs only the preview and then stops. This is to prevent accidental initiation of low-quality but potentially expensive inversions. For more details about the preview (default) configuration, see the Common configurations page.

To run the preview after selecting a region and time period of interest in the *configuration file* (and modifying any other configurable settings), simply run the IMI with the `DoPreview` configuration option set to `true`.

The IMI preview provides the following information for users to assess their proposed inversion (as it is described in the configuration file):

- Map of mean TROPOMI observations for the region and period of interest
- Map of prior emission estimates to be used in the inversion
- Map of observation density for the region and period of interest
- Map of mean SWIR albedo for the region and period of interest
- Total number of observations available in the region of interest during the inversion period
- Rough estimate of degrees of freedom for signal (DOFS) for the inversion
- Rough estimate of USD financial cost of the inversion

This information is generated as a `.txt` file and collection of `.png` files in the preview directory, which is located at:

```
/home/ubuntu/imi_output_dir/{YourRunName}/preview_run/
```

The `.txt` file can be viewed directly in the terminal. To view the `.png` files, first download them from EC2 to your local computer using:

```
$ scp -i /local/path/to/my-key-pair.pem ubuntu@my-instance-public-dns-name:/path/to/my-
→file.png /local/path/to/my-file.png
```

For more informaton on this command, see the AWS Documentation.

Tips for interpreting the IMI preview results:

- Inspect the maps of XCH4 observations and observation density to evaluate TROPOMI coverage for the region and period of interest.
- Compare the maps of XCH4 observations and prior emission estimates to evaluate spatial correspondence between the two datasets.

- Compare the maps of XCH4 observations and SWIR albedo to confirm that there are no obvious albedo-related artifacts in the methane retrieval field, which would be diagnosed by high spatial correlation between XCH4 and albedo.

- DOFS > 1 is a bare minimum to achieve any solid information about emissions.

- DOFS < 2 is marginal for most applications.

- If there is an obvious mismatch between the XCH4 observations and the prior emissions (for example due to severe bias in the prior inventory) OR if the expected DOFS is low, consider: (a) using an improved prior inventory, (b) increasing the inversion period to incorporate more observations, and/or (c) increasing the prior error estimate.

- If there is indication of albedo-related artifacts in the XCH4 field, consider removing the affected observations. This can be done by modifying the TROPOMI data filters via the `filter_tropomi()` function in `/home/ubuntu/integrated_methane_inversion/src/inversion_scripts/utils.py`.

TIPS FOR MINIMIZING AWS COSTS

4.1 Switching instance types

The IMI can be very efficient when run on an EC2 instance with significant compute power. But if you only wish to run the IMI preview or analyze output data, then much of this compute power will be wasted.

Thankfully, it is possible to switch the instance type of an existing instance if you expect to be doing less compute-heavy work. See the AWS Documentation on how to change your instance type for more information.

4.2 Spot instances

Spot instances take advantage of unused compute capacity on the AWS cloud, allowing users to launch instances at a 70-90% reduction in price compared to on-demand instances.

However, this reduced pricing comes with the understanding that AWS can take back this extra capacity at any time, so your instance may be interrupted (with a 2 minute warning), causing the IMI to crash. Interruptions are generally rare (~5% of instances get interrupted) and once interupted your instance will be either Stopped or Terminated depending on the EC2 configuration.

We recommend using spot instances for inversions that take hours (not days) as it can greatly reduce your EC2 costs. For information on how to launch a spot instance see Create a Spot Instance Request. For more information on how to avoid and handle interruptions check out this post on Best Practices.

4.3 Selecting storage volume size

AWS charges continuous fees for the storage volume provisioned to an EC2 instance. These fees can become significant if you retain the volume for long periods of time (weeks/months).

It is best to provision only the amount of storage needed, and to delete your volume once finished with it to minimize costs.

You can always add storage space after launching an EC2 instance, but it is very difficult to retroactively reduce storage space; see the AWS Documentation for details.

Note: When unsure of the storage needs for an inversion, we recommend starting small. A good starting point is ~100 GB.

To determine your true storage needs, first `ssh` into the instance and run a 1-week inversion for your region of interest. When the 1-week inversion is complete, check how much storage has been used. From there, you can scale-up the storage according to your actual period of interest. Consider that the AMI itself takes about 20 GB of storage.

For example, if after the 1-week inversion you find that 75/100 GB are occupied, then you should budget 75 - 20 = 55 GB per inversion week. If you want to perform a 1-year inversion, then increasing the storage to 3.5 TB will leave you with about 500 GB of additional space to work with once the inversion is complete.

4.4 Exporting data to S3

Storing data in EBS volumes is more expensive than storing data in Amazon S3. Additionally, with S3 you are only charged for the amount of space you use, whereas EBS volumes charge you for the amount of space provisioned.

For these reasons, after running the IMI, we recommend pushing your output data to an S3 bucket for long term storage, rather than retaining the entire EBS volume. Resources for creating an S3 bucket and pushing data to it can be found here:

- Creating an S3 Bucket
- Uploading/Downloading files using the cp command

RUNNING THE IMI WITH TMUX

The IMI can be run with tmux as an alternative to sbatch. Like sbatch, tmux allows you to run a program on your EC2 instance, disconnect, and then reconnect later to check progress.

Because of the way the IMI is parallelized, using tmux can grant a small to moderate speed-up.

Note: Before running the IMI with tmux, make sure the `UseSlurm` option in the *configuration file* is set to `false`.

5.1 Using tmux

tmux comes preinstalled on the AMI. To start tmux run the following:

> $ tmux

This enters a tmux shell. From there you can run the inversion script:

> $ ./run_imi.sh > imi_output.log

This will start the workflow. To keep it running in the background, press `ctrl-b`. Then press `d` (without holding `ctrl`) to detach the tmux shell and get back to the original terminal. At this point you can disconnect the from ssh and the IMI will continue to run in the background.

To check back in on the IMI, ssh back onto the EC2 instance and run the following to attach to the active tmux session:

> $ tmux attach-session -t 0

SIX

SETTING UP JUPYTER ON EC2

The IMI relies on Jupyter notebooks to visualize the results of inversions. However, in order to view and run Jupyter notebooks you will need to set up a jupyter server on your EC2 instance and securely access it from your local browser. You can do this using a few different methods:

6.1 Using an SSL certificate (AWS Recommended)

Follow these short instructions to set up and connect to a jupyter notebook server on AWS using a self-signed SSL certificate. Note: If you are using Git-BASH, or similar software with ssh, you can follow the Configure a Linux or macOS Client section, which provides a simpler setup than the Windows instructions.

6.2 Using an automatically generated authentication token

If the above AWS recommended method is causing trouble you can also use the following method to create and connect to a jupyter server using an authentication token. The authentication token is a randomly generated hash code appended to the jupyter server url. The token, similar to a password, verifies you have permission to access the server.

To set up a jupyter notebook server on your ec2 instance, run the following command on your **remote/EC2** terminal:

```
$ jupyter notebook --no-browser --port 8080
```

This will start a jupyter server on port 8080 and will print out a link with an authentication token, eg:

```
$ jupyter notebook --no-browser --port 8080
  ....
  http://localhost:8080/?token=7a7ae708966c68e631bc76ba9eae7b1d287e4747cf7072e7
```

Then in a new **local** terminal (or GIT-Bash) window run the following command:

```
$ ssh -NL 8080:localhost:8080 -i /path/to/private_key
```

This creates an ssh tunnel from your ec2 instance to your local computer over port 8080, which will allow you to view your jupyter notebooks from your browser. Go to the link outputted from your remote serve command above (eg. http://localhost:8080/?token=7a7ae708966c68e631bc76ba9eae7b1d287e4747cf7072e7).

CREATING A CUSTOM STATE VECTOR FILE

By default the IMI uses latitude/longitude bounds to automatically create a gridded state vector file for a rectilinear region of interest with surrounding buffer elements.

The state vector file is located at

```
/home/ubuntu/imi_output_dir/{YourRunName}/StateVector.nc
```

It contains the state variable labels for every grid cell in the inversion domain. For example, if the region of interest contains 200 emission elements and the IMI is configured to use 8 additional buffer elements, then the total number of state variables is 208 and the state vector file will assign a number between 1 and 208 to every grid cell in the inversion domain.

Instead of the default rectilinear region of interest, you may want to use an irregular region as was done for the Permian Basin by Varon et al. (2022; link_TODO). To do so you will need to generate the state vector file yourself.

The easiest way to do this is by using a shapefile for the region of interest in conjunction with the `statevector_from_shapefile.ipynb` jupyter notebook. The notebook is located at

```
/home/ubuntu/integrated_methane_inversion/src/notebooks/statevector_from_shapefile.ipynb
```

First upload a shapefile for the custom region of interest to your EC2 instance:

```
$ scp -i /local/path/to/my-key-pair.pem /local/path/to/my-shapefile ubuntu@my-instance-
→public-dns-name:/path/to/my-shapefile
```

Next, open the configuration file and insert the path to your shapefile in the custom state vector section. Also provide the latitude/longitude bounds for the desired inversion domain, which will include both the irregular region of interest and the additional coarse buffer elements.

Next, follow these short instructions to set up and connect to a jupyter notebook server on AWS. Once connected to the server, open `statevector_from_shapefile.ipynb` and run its contents to generate a state vector file from your shapefile.

If no shapefile is available, you will need to construct the custom state vector file manually. You may want to start from an automatically generated rectilinear state vector file.

USING THE IMI CLUSTERING OPTIONS

8.1 Why use the clustering options?

The main computational cost of the IMI is running the perturbation simulations necessary to construct the jacobian. This requires running a (GEOS-Chem) Jacobian simulation for each state vector element. The default state vector that is generated with the IMI has state vector elements in native resolution, meaning each element corresponds with a GEOS-Chem grid cell (.25 degree or .5 degree resolution). However, if your state vector has a sufficiently large number of elements this can limit the feasibility of running the IMI – either due to prohibitively high AWS costs or compute time. Clustering your state vector elements reduces the number of state vector elements by aggregating elements together.

8.2 Using the IMI clustering config options

To enable the IMI clustering options in the imi config file set `ReducedDimensionStateVector: true`. This enables the clustering component of the IMI. Once enabled the IMI uses your specified `NumberOfElements` to aggregate native resolution state vector elements within your domain of interest using the specified `ClusteringMethod`. eg:

```
ReducedDimensionStateVector: true
ClusteringMethod: "kmeans"
NumberOfElements: 39
```

This automatically generates a state vector with 39 elements (including buffer elements) in the domain of interest. This is done by creating a set of information content informed clustering pairs (eg. [[1, 15], [2, 24]]). *Note:* As you reduce the dimension of your state vector, you should also correspondingly decrease the value of your regularization factor `Gamma`. It can be scaled by the ratio of reduced number of elements over the original number of elements (eg. `len(new_elements)/len(orig_elements)`).

Each clustering pair consists of the the aggregation level and the number of cells you are allocating with the aggregation level. In the above example, the user is requesting 39 total state vector elements and the algortithm determines the information content informed pattern to be 15 native resolution state vector elements and 24 state vector elements to be aggregated with another element. Any additional elements that have not been allocated are then aggregated into a single element. Using the above clustering pairs, if the domain of interest has 63 elements in the original state vector, 15 of the elements would maintain the original resolution and 48 of the elements would be aggregated into 24 2-gridcell elements. If the original state vector has 75 elements in the domain of interest, then the remaining 12 unallocated elements are aggregated into a single element, netting a new state vector with 40 elements in the domain of interest.

The cluster pairings are generated by aggregating elements until they reach a threshold in the estimated DOFS per cluster, which is a measure of information content. We find using the threshold of `total_DOFs / num_state_vector_elements` provides a reasonable result.

The `ClusteringMethod` specifies which clustering method to use for state vector reduction. Currently `kmeans` or `mini-batch-kmeans` are valid options. `mini-batch-kmeans` is very similar to `kmeans`, but can be less accurate. It

is best used for very large state vectors to speed up state vector reduction.

Additionally, the `ForcedNativeResolutionElements` is a configuration option that allows you to specify areas that you would like to maintain high resolution for. Eg:

```
ForcedNativeResolutionElements:
  - [31.5, -104]
```

For instance, if the user suspects a location to be an emission hotspot they can specify the lat/lon coordinates as in the example above and the clustering algorithm will ensure that the native resolution element is preserved during the aggregation. In order for the IMI to preserve the element, you must have enough `NumberOfElements` specified to accomodate the number of gridcells you would like to force to be native resolution.

Note: The IMI preserves the original state vector file as NativeStateVector.nc in your run directory.

8.3 IMI clustering scheme

The IMI clustering algorithm uses a similar k-means based method as described in Nesser et al., 2021 to maintain native resolution in areas with high information content (high prior emissions, high observation density), while aggregating cells with low information content.

8.4 Reducing computational cost while maintaining inversion quality

While clustering is an effective method for alleviating computational constraints for running inversions at high resolution for large regions, it can introduce aggregation error and degrade the quality of your inversion (Turner and Jacob., 2014). Therefore, it is important to weigh the computational benefits of reducing your state vector against the inversion quality loss. This can be done by iteratively tuning the cluster pairings and running the IMI preview.IMI preview to assess the estimated DOFS. Ideally, you should find a middle groud where the estimated DOFS and computation cost is at a acceptable level before proceeding with the inversion.

NINE

MODIFYING PRIOR EMISSION ESTIMATES

To modify the default prior emission inventories, first generate the template run directory following the configuration instructions for modifying emissions.

Once the template run directory is ready, you will need to modify the emission inventories via HEMCO.

Start by transferring your custom emission inventory to EC2:

```
$ scp -i /local/path/to/my-key-pair.pem /local/path/to/my-inventory.nc ubuntu@my-
↪instance-public-dns-name:/path/to/my-inventory.nc
```

The emissions need to be defined in a netcdf file formatted as HEMCO expects. See the HEMCO documentation for preparing data files for details on how to format your custom emission inventory for use with HEMCO.

Once your inventory has been properly formatted, you can include it as an emission field via HEMCO. To do this, navigate to the template run directory and open the HEMCO configuration file with vim (`vi`) or emacs:

```
$ cd /home/ubuntu/imi_output_dir/{YourRunName}/template_run
$ emacs HEMCO_Config.rc
```

Follow instructions in the HEMCO User's Guide to add a new emission field.

You can run the *IMI preview* to quickly check that the updated emissions are working as expected.

USING CUSTOM REGIONS WITH THE IMI

The IMI supports regions within the following domains:

- Africa: 37°S-40°N, 20°W-53°E

- Asia: 11°S-55°N, 60°E-150°E

- Europe: (33°N-61°N, 30°W-70°E

- Middle East: 12°N-44°N, 20°W-70°E

- North America: 10°N-70°N, 140°W-40°W

- Oceania 50°S-5°N, 110°E-180°E

- Russia 41°N-83°N, 19°E-180°E

- South America 59°S-16°N, 88°W-31°W

These are the nested-grid windows used in GEOS-Chem for which pre-cut meteorological files are available. You may apply the IMI to other regions, but this requires either using global meteorological fields which can be computationally expensive (not recommended) or cropping global meteorological fields via a pre-processing step.

To facilite cropping global meteorological fields, a sample script (crop_met.sh) has been included with the IMI. This script utilizes the Climate Data Operators (CDO) . It also includes an option to first download global meteorological fields at 0.25° x 0.3125° resolution. The global files are large (approx. 300G per month), so when using that option it is recommend that you process short periods at a time and delete the global files before processing additional periods.

In a text editor, modify the user settings section in `crop_met.sh`. The region defined in `crop_met.sh` should be the same or larger than the domain defined for your IMI in *config.yml*.

NestedRegion	Two-letter string to identify region (e.g. SA for South America). This should match the value of `NestedRegion` in `config.yml`.
LonMin	Minimum longitude edge of the region of interest.
LonMax	Maximum longitude edge of the region of interest.
LatMin	Minimum latitude edge of the region of interest.
LatMax	Maximum latitude edge of the region of interest.
DownloadGlobalMet	Boolean for downloading global 0.25° x 0.3125° meteorology fields for cropping. Default is `true`. Set to `false` if you already have these files on your system.
RemoveGlobalMet	Boolean for deleting global meteorology files after cropping. Default is `false`. Set to `true` once you are sure the cropped meteorology files are working properly in the IMI and you do not plan to generate cropped files for additional regions.
InDir	Directory containing the global high-resolution meteorology fields.
OutDir	Directory where the cropped meteorology files will be placed. We recommend specifying this as `[YOUR_DATA_PATH]/GEOS_0.25x0.3125_${NestedRegion}/GEOS_FP`. Where you replace `[YOUR_DATA_PATH]` with the same path specified for `DataPath` in the IMI's config.yml.

The cropped meteorology files can be generated by then executing `./crop_met.sh` at the command line or submitting the script to your cluster's scheduler if available. Headers for the SLURM scheduler are included at the top of the script, but you can modify or remove those as needed.

To utilize the cropped meteorology files in the IMI, you will need to create a new IMI directory. Modify `config.yml` so that `NestedRegion` matches the value set in `crop_met.sh`. This will automatically add the region ID string in the appropriate locations in the `HEMCO_Config.rc` files utilized by GEOS-Chem.

If you have regional emissions that you would like to use, please see *modifying prior emission estimates*.

Finally, you can run the *IMI preview* to quickly check that the IMI is working as expected for your custom region.

CONSTRUCTING AN INVERSION ENSEMBLE

After performing an inversion, you can use the IMI to create a low-cost ensemble of sensitivity inversions with different inversion parameters. This is because the Jacobian matrix computed in the first inversion can easily be reused.

See the Common configurations page for instructions on how to re-configure the IMI to use a pre-computed Jacobian. Then modify the values of `PriorError`, `ObsError`, and/or `Gamma` in the configuration file and re-run the inversion.

Note: Make sure to archive the final results of the original inversion (`inversion_result.nc` and `gridded_posterior.nc`) before running the sensitivity inversion. Those files will be overwritten.

If you want to run a sensitivity inversion with updated prior emission inventories, the pre-computed Jacobian needs to be scaled according to the differences between the original and updated inventories. Instructions for this to come.

TWELVE

RUNNING THE IMI ON A LOCAL CLUSTER

The IMI is setup to run on AWS by default. However, if you have a local cluster available to you, you may choose to run the IMI there. This option requires some manual changes and is therefore only recommended for advanced users.

You must first ensure you have the proper hardware and software requirements for running GEOS-Chem.

When logged onto your local cluster, navigate to the path where you want to download the IMI repository and type the following command:

```
$ git clone https://github.com/geoschem/integrated_methane_inversion.git
```

This will clone the IMI code into a local folder named `integrated methane_inversion`.

Tip: If you wish, you can clone the IMI repository into a different local folder by supplying the name of the folder at the end of the **git clone** command. For example:

```
git clone https://github.com/geoschem/integrated_methane_inversion.git imi-v1.0
```

Next, download the GEOS-Chem source code and its submodules within the IMI folder using these commands:

```
$ cd integrated_methane_inversion
$ git clone https://github.com/geoschem/GCClassic.git
$ cd GCClassic
$ git submodule update --init --recursive
```

See Downloading rhe GEOS-Chem source code for more details.

Navigate back to the top-level IMI folder and view the contents:

```
$ cd ..
$ ls
config.yml   envs/        LICENSE.md   resources/    setup_imi.sh*
docs/        GCClassic/   README.md    run_imi.sh*   src/
```

Within the IMI is a subfolder called `envs` that constains files for running the IMI on different systems. By default, files are provided for AWS and Harvard's Cannon cluster.

```
$ ls envs/*
envs/aws:
conda_env.yml   slurm/   spack_env.env

envs/Harvard-Cannon:
```

```
ch4_inv.yml                gcc.gfortran10.2_cannon.env*  gcc.ifort19_cannon.env*
config.harvard-cannon.yml  gcc.ifort17_cannon.env*       README
```

We recommend you add a subfolder within `envs` for your own system to easily access your customized files needed for the IMI. In this directory, we recommend storing any environment files needed to load the libraries for GEOS-Chem (e.g. fortran compiler, netcdf, openmpi, cmake), a conda environment file, and a copy of the IMI configuration file modified for your system. See the files in `envs/Harvard-Cannon` for examples.

Within the copied IMI configuration file, you will need to modify the settings in the section labeled "Settings for running on your local cluster." If you already have the GEOS-Chem input data on your system, you may set the `*DryRun` options to `false`.

It is recommended that you set up and run the IMI in stages when running on a local cluster to ensure that each stage works properly. You can do this by modifying the settings under "Setup modules" and "Run modules" and turning them on one or a few at a time. You may find that you need to manually edit some files. For example, after creating the template run directory, but before creating your spinup, Jacobian, and posterior run directories, you should open `ch4_run.template` in a text editor and modify as needed for your system (by default this script is set up to submit to a SLURM scheduler).

Once your have finished customizing the IMI settings for your cluster, you can run the IMI by executing `run_imi.sh` and passing an argument for the location of your IMI configuration file. For example:

```
$ ./run_imi.sh config.harvard-cannon.yml
```

If you do not pass a configuration file, `config.yml` in the top-level IMI directory will be used. That file is set up for running the IMI on AWS by default.

COMMON IMI CONFIGURATIONS

This page provides examples of how to configure the IMI setup modules and run modules to accomplish some common tasks.

13.1 Default (preview) configuration

By default the IMI will download the TROPOMI data for the period of interest, set up the template GEOS-Chem run directory, run the preview, and then stop.

```
## Setup modules
SetupTemplateRundir: true
SetupSpinupRun: false
SetupJacobianRuns: false
SetupInversion: false
SetupPosteriorRun: false

## Run modules
RunSetup: true
DoSpinup: false
DoJacobian: false
DoInversion: false
DoPosterior: false

## IMI preview
DoPreview: true
```

If the results of the preview are satisfactory, you can try the next configuration on this page to run the inversion. If they are not satisfactory, modify the configuration file (e.g., the region and/or period of interest) and try again.

13.2 Running an inversion after the preview

If the preview is complete and the results are satisfactory, you can proceed with the inversion (without re-running the preview).

```
## Inversion
PrecomputedJacobian: false

## Setup modules
```

```
SetupTemplateRundir: false
SetupSpinupRun: true
SetupJacobianRuns: true
SetupInversion: true
SetupPosteriorRun: true

## Run modules
RunSetup: true
DoSpinup: true
DoJacobian: true
DoInversion: true
DoPosterior: true

## IMI preview
DoPreview: false
```

13.3 Running a sensitivity inversion

You've completed an initial inversion. Use the following configuration to run a new inversion with modified prior error (`PriorError`), observational error (`ObsError`), or regularization parameter (`Gamma`).

```
## Inversion
PrecomputedJacobian: true

## Setup modules
SetupTemplateRundir: false
SetupSpinupRun: false
SetupJacobianRuns: false
SetupInversion: false
SetupPosteriorRun: false

## Run modules
RunSetup: false
DoSpinup: false
DoJacobian: false
DoInversion: true
DoPosterior: false

## IMI preview
DoPreview: false
```

Note that the final results of the original inversion (`inversion_result.nc` and `gridded_posterior.nc`) will be overwritten if not archived before running the sensitivity inversion.

13.4 Running an inversion without the preview

We generally don't recommend doing this, but if you wish to perform an inversion without manually inspecting the results of the IMI preview, use the following configuration to run the IMI from end to end, with a threshold on the expected degrees of freedom for signal (DOFS) to cancel the inversion; if the expected DOFS are below the threshold, the IMI will exit with a warning.

```
## Setup modules
SetupTemplateRundir: true
SetupSpinupRun: true
SetupJacobianRuns: true
SetupInversion: true
SetupPosteriorRun: true

## Run modules
RunSetup: true
DoSpinup: true
DoJacobian: true
DoInversion: true
DoPosterior: true

## IMI preview
DoPreview: true
DOFSThreshold: {insert-threshold-value}
```

13.5 Modifying prior emission estimates

Set up the template run directory

```
## Setup modules
SetupTemplateRundir: true
SetupSpinupRun: false
SetupJacobianRuns: false
SetupInversion: false
SetupPosteriorRun: false

## Run modules
RunSetup: true
DoSpinup: false
DoJacobian: false
DoInversion: false
DoPosterior: false

## IMI preview
DoPreview: false
```

Run the preview

After *modifying the prior emission inventories*, run the preview without setting up the template run directory.

```
## Setup modules
SetupTemplateRundir: false
```

```
SetupSpinupRun: false
SetupJacobianRuns: false
SetupInversion: false
SetupPosteriorRun: false

## Run modules
RunSetup: true
DoSpinup: false
DoJacobian: false
DoInversion: false
DoPosterior: false

## IMI preview
DoPreview: true
```

If satisfied with the preview results, continue with one of the above configurations to run the inversion.

IMI DIRECTORY CONTENTS

This page describes the contents of various file directories generated and populated by the IMI in the course of an inversion.

14.1 Inversion directory

The inversion directory is where the IMI computes the Jacobian, obtains the optimal estimate of emissions, and saves the results.

It is located at `/home/ubuntu/imi_output_dir/{YourRunName}/inversion`.

In addition to a shell script and several Python scripts used in the inversion, you will find the following items in the inversion directory after completing an inversion:

`data_converted/`	Directory of Python `.pkl` files containing • TROPOMI observations • virtual TROPOMI observations of the GEOS-Chem reference simulation • elements of the Jacobian matrix for each TROPOMI orbit relevant to the inversion. All quantities have been "converted" to 1D fields indexed by latitude and longitude.
`data_converted_posterior/`	Directory of Python `.pkl` files containing • TROPOMI observations • virtual TROPOMI observations of the GEOS-Chem posterior simulation for each TROPOMI orbit relevant to the inversion. All quantities have been "converted" to 1D fields indexed by latitude and longitude.
`data_geoschem/`	Directory of `.nc` files containing daily GEOS-Chem `SpeciesConc` output from the reference simulation. These files are used to generate virtual TROPOMI observations for comparison with the true observations.
`data_geoschem_posterior/`	Directory of `.nc` files containing daily GEOS-Chem `SpeciesConc` output from the posterior simulation. These files are used to generate virtual TROPOMI observations for comparison with the true observations.
`data_sensitivities/`	Directory of `.nc` files containing daily 4-D GEOS-Chem sensitivities to perturbations in the state variables of the inversion (i.e., in the emission elements being optimized). The data have dimensions (`element`, `lev`, `lat`, `lon`), where `element` is the emission element id (state variable id) and `lev` is the vertical dimension. These files are used to compute the Jacobian matrix by application of the TROPOMI operator.
`inversion_result.nc`	File containing the raw output of the inversion (`invert.py`) as vectors (posterior emission estimate) and matrices (posterior error covariance matrix, averaging kernel matrix).
`gridded_posterior.nc`	

AMI SPECIFICATIONS

The Amazon Machine Image (AMI) for the IMI is accessible through the aws marketplace as a (free) product listing.

The latest AMI for the IMI Workflow contains the following software libraries:

- GNU Compiler Collection 8.2.0

- NetCDF-Fortran 4.5.3

- Slurm 17.11.2

- Python 3.9.7

- GEOS-Chem Classic 13.3.3

TODO: add additional software dependencies

KNOWN BUGS

This page links to known bugs in the IMI. See the Github issues page for updates on their status.

SUPPORT

For support with the IMI workflow please create an issue on the github repository or email us at **integrated-methane-inversion@g.harvard.edu** detailing the nature of the issue you are facing. Please attach your IMI config.yml file, any relevant log files, and the version of the IMI you are using.

17.1 Example Github Issue Template

```
### What institution are you from?
Please tell us what institution you are from.

### Description of the problem
Describe your problem here.  Describe the steps to reproduce the problem here, if
↪possible.

### Description of troubleshooting performed
Describe any troubleshooting that you have already performed here. Also include any
↪leads or suspicions here.

### IMI version
Enter your IMI version here.

### Description of modifications
Describe any modifications to the IMI here.

### Attach relevant files
- imi_output.log
- imi config.yml file
- any other relevant log files to your issue
```

EIGHTEEN

CONTRIBUTING

Contributions can be made in the form of pull requests to our `github repository` <`https://github.com/ACMG-CH4/CH4_inversion_workflow`>.

EDITING THIS USER GUIDE

This user guide is generated with Sphinx. Sphinx is an open-source Python project designed to make writing software documentation easier. The documentation is written in a reStructuredText (it's similar to markdown), which Sphinx extends for software documentation. The source for the documentation is the `docs/source` directory in top-level of the source code.

19.1 Quick start

To build this user guide on your local machine, you need to install Sphinx. Sphinx is a Python 3 package and it is available via **pip**. This user guide uses the Read The Docs theme, so you will also need to install `sphinx-rtd-theme`. It also uses the sphinxcontrib-bibtex and recommonmark extensions, which you'll need to install.

```
$ pip install sphinx sphinx-rtd-theme sphinxcontrib-bibtex recommonmark
```

To build this user guide locally, navigate to the `docs/` directory and make the `html` target.

```
gcuser:~$ cd gcpy/docs
gcuser:~/gcpy/docs$ make html
```

This will build the user guide in `docs/build/html`, and you can open `index.html` in your web-browser. The source files for the user guide are found in `docs/source`.

Note: You can clean the documentation with `make clean`.

19.2 Learning reST

Writing reST can be tricky at first. Whitespace matters, and some directives can be easily miswritten. Two important things you should know right away are:

- Indents are 3-spaces

- "Things" are separated by 1 blank line. For example, a list or code-block following a paragraph should be separated from the paragraph by 1 blank line.

You should keep these in mind when you're first getting started. Dedicating an hour to learning reST will save you time in the long-run. Below are some good resources for learning reST.

- reStructuredText primer: (single best resource; however, it's better read than skimmed)

- Official reStructuredText reference (there is *a lot* of information here)

- Presentation by Eric Holscher (co-founder of Read The Docs) at DjangoCon US 2015 (the entire presentation is good, but reST is described from 9:03 to 21:04)

- YouTube tutorial by Audrey Tavares's

A good starting point would be Eric Holscher's presentations followed by the reStructuredText primer.

19.3 Style guidelines

Important: This user guide is written in semantic markup. This is important so that the user guide remains maintainable. Before contributing to this documentation, please review our style guidelines (below). When editing the source, please refrain from using elements with the wrong semantic meaning for aesthetic reasons. Aesthetic issues can be addressed by changes to the theme.

For **titles and headers**:

- Section headers should be underlined by `#` characters

- Subsection headers should be underlined by `-` characters

- Subsubsection headers should be underlined by `^` characters

- Subsubsubsection headers should be avoided, but if necessary, they should be underlined by `"` characters

File paths (including directories) occuring in the text should use the `:file:` role.

Program names (e.g. `cmake`) occuring in the text should use the `:program:` role.

OS-level commands (e.g. `rm`) occuring in the text should use the `:command:` role.

Environment variables occuring in the text should use the `:envvar:` role.

Inline code or code variables occuring in the text should use the `:code:` role.

Code snippets should use `.. code-block::  <language>` directive like so

```
.. code-block:: python

   import gcpy
   print("hello world")
```

The language can be "none" to omit syntax highlighting.

For command line instructions, the "console" language should be used. The `$` should be used to denote the console's prompt. If the current working directory is relevant to the instructions, a prompt like `gcuser:~/path1/path2$` should be used.

Inline literals (e.g. the `$` above) should use the `:literal:` role.

TERMINOLOGY (TODO: EDIT THIS PAGE)

absolute path

The full path to a file, e.g., `/example/foo/bar.txt`. An absolute path should always start with `/`. As opposed to a *relative path*.

build

See *compile*.

build directory

A directory where build configuration settings are stored, and where intermediate build files like object files, module files, and libraries are stored.

checkpoint file

See *restart file*.

compile

Generating an executable program from source code (which is in a plain-text format).

gridded component

A formal model component. MAPL organizes model components with a tree structure, and facilitates component interconnections.

HISTORY

The MAPL *gridded component* that handles model output. All GCHP output diagnostics are facilitated by HISTORY.

relative path

The path to a file relative to the current working directory. For example, the relative path to `/example/foo/bar.txt` if your current working directory is `/example` is `foo/bar.txt`. As opposed to an *absolute path*.

restart file

A NetCDF file with initial conditions for a simulation. Also called a *checkpoint file* in GCHP.

run directory

The working directory for a GEOS-Chem simulation. A run directory houses the simulation's configuration files, the output directory (`OutputDir`), and input files/links such as *restart files* or input data directories.

stretched-grid

A cubed-sphere grid that is "stretched" to enhance the grid resolution in a region.

target face

The face of a stretched-grid that is refined. The target face is centered on the target point.

INDEX

A
absolute path,

B
build,
build directory,

C
checkpoint file,
compile,

G
gridded component,

H
HISTORY,

R
relative path,
restart file,
run directory,

S
stretched-grid,

T
target face,