CALIFORNIA
TRANSPORTATION
SUPPLY
MODEL (CATS)

The California Transportation Supply (CATS) Model v0.2

About the Model

The CATS model was created as an exploratory tool, and is provided without any warranties. The model is a substantial simplification of the overall California transportation and fuel market, and therefore results should be interpreted with caution.

CATS can be used to explore how different assumptions relating to the cost, supply, demand, and carbon intensities of various fuel may impact the transportation market, and how Low Carbon Fuel Standard credit prices may respond to changes in market conditions and program stringency.

Getting Started

1. Download the code base
2. Setup the Python environment
3. Configure the scenario inputs
4. Running the model
5. Save and analyze the results

Setting up the Python environment

To setup the environment, download and install Anaconda. Once Anaconda is installed, open up the Anaconda command prompt, and create a new environment.

```
conda create --name CATS python=3.9
conda activate CATS
```

If you intend to edit the scripts, or write your own methods or procedures, the Spyder Development Environment is recommended. This can be installed from the Anaconda command prompt.

```
conda activate CATS
conda install spyder
```

Once the Spyder development environment is installed, you will be able to open Spyder from your desktop, and create specific methods to interact with CATS data.

From the Anaconda command prompt, navigate to the directory that you extracted CATS into. An example directory is below.

```
cd "c:/CATS Model/"
```

Before running the model, you must first install model dependencies. The model makes use of the or-tools optimization framework from Google.

```
pip install -r requirements.txt
```

Configuring the scenario inputs

Data for each scenario must be input into the model using a `scenario_inputs.xlsx` workbook located in the scenario directory.

> Note that the `scenario_inputs` file name may contain additional wildcard inputs so long as the naming structure is as follows: `scenario_inputs*.xlsx`. Ensure that only one such file is contained in a given scenario directory. Example file name: `scenario_inputs_BAUScenario.xlsx`

To generate a new scenario template for a scenario named `testing`, run the following:

```
cd "c:\CATS model"
conda activate CATS
python -m cats -t testing
```

This indicates that you will be running the CATS module (`-m` flag) from the `c:\CATS model` directory. You have specified the `-t` flag to create a new template named `testing`. A `scenario_inputs.xlsx` file will be created for you with the correct headings and worksheets in the `testing` folder located in the scenario directory that is specified in the `config.ini` configuration.

To run this scenario, make sure it is correctly specified in the `config.ini` file before you run the model.

> **Important Information About Inputs:** For some of the scenario inputs, the model will use the closest date value provided, while for other inputs the value will be `None` unless specified. The `FuelProduction` pathways do not need to be defined for each year, the closest value will be used in the model. If yields or CI change, however, then a new production pathway should be provided for that year. Each Energy Demand `FuelPool` will also use the value provided from the closest year. For instance, if a value for 2040 is provided, and no `FuelPool` is provided in 2020, then the energy demand in 2020 for that `FuelPool` will be equal to the 2040 energy demand value. `LCFS` benchmarks will use the closest year entered. All other worksheets will only use the values entered for a given year. As such, things like blend requirements must be entered annually.

The fundamental structure and description of worksheets for inputting data and assumptions into the model is provided in the table below. The Python classes used to represent different `datatypes` in the model are also indicated. Ensure that a `scenario_inputs.xlsx` file has been generated and modified for each scenario that you want to run. Each scenario should be located in its own directory (`/scenario/Default/scenario_inputs.xlsx`)

Worksheet Name	Description
Energy Demand	The minimum amount (MJ) of energy that must be provided for a given `FuelPool` in a specified year. A value of `0` may be used in the 'Energy' column to represent possible production pathways to create a product that does not have an energy requirement, such as Direct Air Capture. Additionally, the 'Exceed?' column acts as a `True` or `False` flag to indicate whether or not the value can be exeeded with fuel production. If `False` is specified, the fuel pool will be constrained to the exact value. If `True` is specified, additional fuel may be consumed in this fuel pool beyond the level specified as part of meeting `LCFS` constraints
Defined Supply	This worksheet may be left blank. Anything added to this worksheet will create a lower bound fuel energy (MJ) requirement/constraint for the model in a given year. The model will require a quantity of `Fuel` to be generated in a given year. 'Policy Attribution' is the name or effect is driving that the user would like to attribute the constraint.
Fuel Production	The `ProductionPathway` and costs ($/ton) needed to convert (MJ/ton) `Feedstock` (ton) to `Fuel` (MJ). This sheet also specifies the `FuelPool` the fuel will satisfy demand for, the `LCFS` benchmark that the carbon intensity (gCO2e/MJ) will be evaluated against, and a type identifier for `Credit` or Deficit generation associated with using this pathway. The defined `Credit` type should have a name that's different from any values entered under the 'Results Name' column, or some results will be overwritten. Exogenous subsidies ($/MJ) that can increase or decrease fuel costs may also be specified. An energy economy ratio ('EER') should be specified for calculating the amount of fuel that is displaced by a specific pathway for a given `FuelPool`. The EER factors into the Credit and Deficit generation functionality of the model, where: Credits/Deficits = ($CI_{standard}$ - CI_{Fuel}/EER) x Energy x EER. The 'Blend Requirement' column is used to specify if a fuel has a specific Blend Requirement or blend limitation. The in the `BlendRequirement` is specified by name in the 'Blend Requirements' worksheet.
Fuel Production (Results)	In addition to specify the fuel production information, the Fuel Production worksheet also contains information pertaining to how results are aggregated and displayed. Final model run results will be aggregated based on the 'Results Name' that is specified. For instance, if 'Ethanol (CCS)' and 'Ethanol' are two different `Fuels`, they may be aggregated as 'Ethanol' in the results file. The 'Results Units' column is where the units you would like to convert the finished fuel into for the purpose of presenting results is specified (e.g. MM Gallons, rather than MJ). Finally, the 'Results Multiplier' column is the conversion factor needed to move from the model units (e.g. MJ) to the final results units (e.g. MM Gallons).
Coproducts	This worksheet allows coproducts to be defined. The specified `Fuel` in the worksheet will be constrained such that production of the defined fuel will be produced at a ratio equivalent to the 'Production Multiplier' multiplied by the amount of 'Base Fuel' `Fuel` that is produced. For instance, if Alternative Jet Fuel is specified to have a multiplier of 0.05 (5%) relative to Renewable Diesel, then Alternative Jet Fuel will be produced until the energy content is 5% the value of Renewable Diesel, and will not exceed the value for the pathways specified unless the `Exceed` column is marked `TRUE`.
Production Limits	This worksheet allows for 'Fuel' limits to be imposed. This will prevent fuel production from exceeding an allowed amount in a given year, or a limit to growth relative to the past year, whichever is less. The 'Maximum Volume' in MJ may be specified, or 'Maximum YoY Percent Change' as a decimal value. A zero value for the `Maximum Value` will remove the constraint, so a value of `1` may instead be used to limit fuel volumes to negligible quantities.

Worksheet Name	Description
Additional Credits	This allows a specific `Credit` quantity to be added to (or subtracted from) the model in a given year without needing to produce fuel to generate those credits. Credits added in this way will effectively reduce (or increase) demand for credits under the `LCFS` constraint. The Credit Type specified here must have a valid `ProductionPathway` specified in the 'Fuel Production' worksheet. Dummy pathways can be created that will not result in fuel production (very high production prices) to create new Credit Type categories.
LCFS Benchmark	This worksheet is where the `LCFS` policy benchmarks are defined. A 'Year', 'Benchmark' (e.g gasoline, diesel, alt. jet), and carbon intensity 'Standard' must be provided. The model will use the closest standard to a given benchmark year that is defined for each model run.
Feedstock	This worksheet is where each `Feedstock` supply curve is defined. Feedstock costs should be provided in units consistent with the 'Fuel Production' worksheet inputs for yield and conversion costs Feedstock price points and supply (tons) will only be parameterized as whole integers.
Credit Type Limits	Similar to the 'Production Limits' worksheet, a constraint is added to the model to ensure that a minimum or maximum `Credit` quantity of a specific type is generated in a specific year. Numerical values from `-inf` through `inf` may be specified.
Blend Requirements	Specify the minimum and maximum blend requirement for a defined `Fuel` as a percentage of the energy content for a defined `FuelPool`. For instance, a 10% ethanol blend by volume is a 7.04% blend by energy content for the gasoline `FuelPool`. The 'Requirement Name' corresponds to the 'Blend Requirement' value provided in the 'Fuel Production' worksheet.

Running the Model

The `config.ini` file should be modified to ensure that the model will run the specific scenarios you are interested in running. Ensure that your results folder will be in the same location that you have extracted the model to.

Once your scenarios and configuration files are configured, the model can be run from the command prompt. The example provided below assumes you have extracted the model to the `c:\CATS model` folder, and installed an Anaconda instance as described previously.

```
cd "c:\CATS model"
conda activate CATS
python -m cats
```

Save and Analyze Results

The model will run, and store results files in the specified results directory. Results will be stored, by default, as an excel document for each year that the model run converged. To store higher resolution outputs by year that are not aggregated into a final results file, the verbose flag (`-v`) can be specified:

```
python -m cats -v
```

Results will be aggregated up using parameters specified on the `Fuel Production` tab. The `Results Name` parameter defines the name to aggregate a given production pathway by. The `Results Units` will be added to a column in the results file, and the `Results Multiplier` coefficient is the value used to convert from fuel production (likely in MJ) to the Results Units that have been specified. The model will not ensure unit consistency, that is up to the user that parameterizes the scenario inputs.