ACT Project Number:
271498

Project acronym:
ELEGANCY

Project full title:
Enabling a Low-Carbon Economy via Hydrogen and CCS

ERA-Net ACT project

Starting date: 2017-08-31
Duration: 36 months

D4.5.2
User Documentation – Design Mode

Issue 1

Actual DELIVERY date: 2020-10-19

Organization name of lead participant for this deliverable:
Imperial College London

ACT ELEGANCY, Project No 271498, has received funding from DETEC (CH), BMWi (DE), RVO (NL), Gassnova (NO), BEIS (UK), Gassco, Equinor and Total, and is cofunded by the European Commission under the Horizon 2020 programme, ACT Grant Agreement No 691712.		
Dissemination Level		
PU	Public	X
CO	Confidential, only for members of the consortium (including the Commission Services)	

Deliverable number:	D4.5.2
Deliverable title:	User Documentation – Design Mode
Work package:	WP 4
Lead participant:	Imperial College London (ICL)

Authors		
Name	**Organisation**	**E-mail**
Nixon Sunny*	ICL	nixon@imperial.ac.uk
Niall Mac Dowell	ICL	niall@imperial.ac.uk
Nilay Shah	ICL	n.shah@imperial.ac.uk

* Corresponding author

Executive Summary

The purpose of this document is to provide a potential user with a reference guide to use in conjunction with the design mode as part of work package (WP) 4 in the ERA-Net ACT ELEGANCY project. Along with this document, a model tool-kit is released. The model tool-kit release is implemented in Pyomo, which is a Python-based algebraic modelling environment. This software is open-source and it allows for ease in interfacing with open-source/ commercial solvers. Issue 1 is principally concerned with the provision of user documentation to support the release in Pyomo. The aim of this user documentation is not to describe the algebraic model in detail. A detailed model description will be provided as part of a scientific article for clarifying the model equations, with procedures for implementing new parameters/ equations, etc. The design mode can be used in liaison with the GIS tools to ensure that the users avoid any potential issues in pre/post processing of the necessary data. If there are any comments/ concerns or general enquiries, please forward them to the following email address: nixon@imperial.ac.uk.

ACT ELEGANCY, Project No 271498, has received funding from DETEC (CH), BMWi (DE), RVO (NL), Gassnova (NO), BEIS (UK), Gassco, Equinor and Total, and is cofunded by the European Commission under the Horizon 2020 programme, ACT Grant Agreement No 691712.

TABLE OF CONTENTS

Page No.

ACT ELEGANCY, Project No 271498, has received funding from DETEC (CH), BMWi (DE), RVO (NL), Gassnova (NO), BEIS (UK), Gassco, Equinor and Total, and is cofunded by the European Commission under the Horizon 2020 programme, ACT Grant Agreement No 691712.

1 PYOMO MODEL

The aim of this document is to provide users with a reference guide to assist in the use of the developed tools and software in WP4 of this project. The function of the design mode in WP4 is to enable users to design H_2-CCS infrastructure in an optimal manner given regional/ technological constraints within the area of interest. This section contains a detailed overview of the interaction between a user and the model release in Pyomo along with important instructions on tailoring the models for specific application to any specific case study. In Pyomo, models are usually constructed in an abstract manner where the model data is separated from the model to enhance reusability of the model equations without the need for extensive modification with each use. This style was adopted for the development of the design mode in this project and as such, a user is not required to modify the independent model equations in an attempt to study multiple scenarios.

The Pyomo modelling environment consists of four key model entities: Sets, Parameters, Variables and Equations. Sets typically contain an ordered/ unordered list of items. Parameters contain known values of metrics that are often indexed by components in the sets. Variables are used to describe often unknown/ varying entities in the physical systems during the course of a simulation. Equations are used to describe both equality and inequality constraints within physical environment. Oftentimes, the equality constraints (model equations), in general, are assumed to be valid irrespective of the scenario that is investigated. However, the inequality constraints may differ amongst scenarios. For example, suppose that there are two scenarios – scenario A and B. Scenarios A and B may differ by limiting the number of H_2 process facilities that can be built in any given location to 5 and 10 respectively. This may be the only distinction between both scenarios, or this may be one of many differences which distinguishes each scenario. Thus, a user may have to interact with such inequality constraints in the model more often than the equality constraints when studying distinct scenarios. In addition, users may also have to interact with the Sets and Parameters entities to modify the options under consideration and alter parameter values for a new simulation if this is of interest.

It was planned in the “User Requirements Specification” that users are expected to be able to interact with the modelling environment closely in order to be able to run their individual case studies with additional support offered by the lead developers. This documentation is targeted at both “beginner” and “expert” users alike with the aim of clarifying some of the key model components to provide a potential user with model insights.

1.1 Software installation and use

1.1.1 How do I use these tools?

If you don’t have Python installed, please visit the following link: https://www.anaconda.com/distribution/. You can install the Anaconda distribution which contains a wealth of open-source packages. Depending on your operating system, you can download the necessary version of Anaconda. Please use Python 3 if you’re a new user and as the legacy version will not be supported beyond 2020.

Installation instructions for Pyomo can be found via the following link: http://www.pyomo.org/installation. Once after you’ve installed Anaconda, you may load up “Anaconda Prompt” on your PC through the search function. When a terminal window opens, type the following “conda install –c conda-forge pyomo pyomo.extras”. Then press “Enter”. Some

ACT ELEGANCY, Project No 271498, has received funding from DETEC (CH), BMWi (DE), RVO (NL), Gassnova (NO), BEIS (UK), Gassco, Equinor and Total, and is cofunded by the European Commission under the Horizon 2020 programme, ACT Grant Agreement No 691712.

packages will be downloaded at this point and when you are prompted to confirm installation in the terminal window, type "y" and hit the "Enter" key once again. If you're having installation issues, please cross-check with their FAQ.

INSTALLING PYOMO WITH CONDA

Some scientific Python distributions also include the **conda** package, which can also be used to download and install the latest Pyomo release. You can install Pyomo in your system Python installation by executing the following in a shell:

```
conda install -c conda-forge pyomo
```

Pyomo also has conditional dependencies on a variety of third-party Python packages. These can also be installed with **conda**:

```
conda install -c conda-forge pyomo.extras
```

1.1.2 How do I install relevant solvers?

Installers

Info: This package contains files in non-standard labels.

conda install

linux-64 v4.65
win-32 v4.65
win-64 v4.65
osx-64 v4.65

To install this package with conda run one of the following:

```
conda install -c conda-forge glpk
conda install -c conda-forge/label/gcc7 glpk
conda install -c conda-forge/label/broken glpk
conda install -c conda-forge/label/cf201901 glpk
```

GLPK is an open-source/ free solver which will be sufficient for most case studies but its performance is a limiting factor in larger models. To install GLPK, please use conda and execute the top line in the Anaconda Prompt once again. You may also install commercial solvers such as CPLEX/ Gurobi, albeit with license charges.

1.1.3 Did I install the solver correctly?

Open up `Anaconda Prompt' and enter `glpsol --help'. You should obtain a similar looking screen by the end. This means that the installation was done correctly.

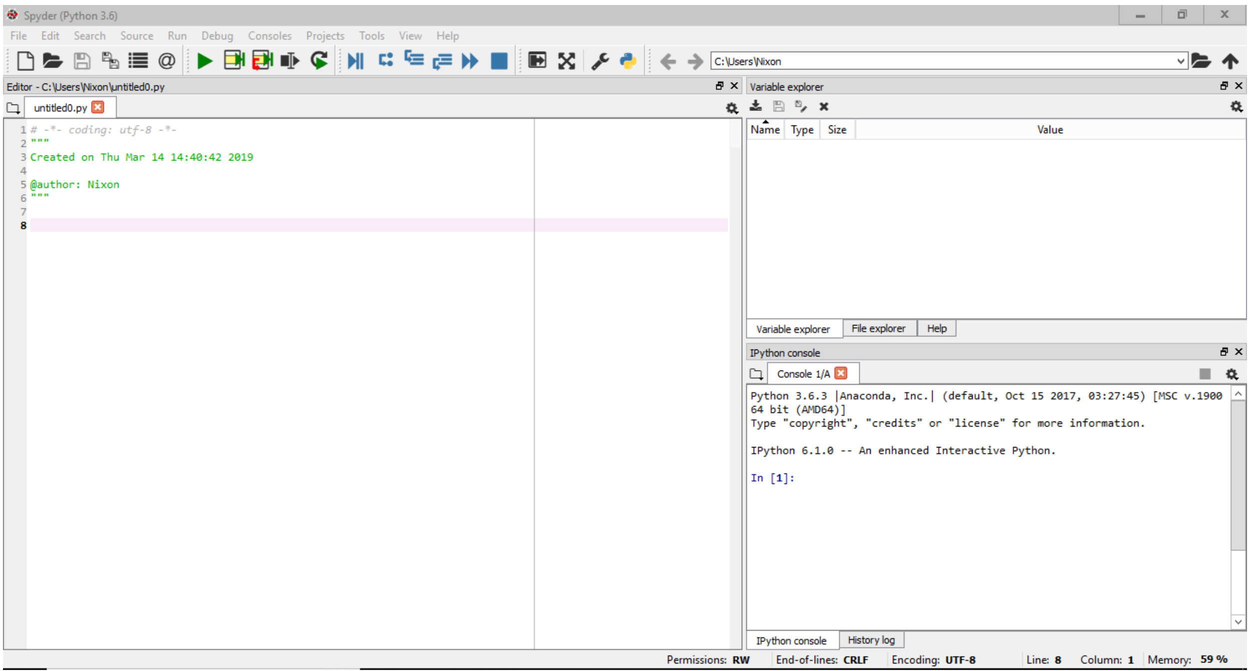
ACT ELEGANCY, Project No 271498, has received funding from DETEC (CH), BMWi (DE), RVO (NL), Gassnova (NO), BEIS (UK), Gassco, Equinor and Total, and is cofunded by the European Commission under the Horizon 2020 programme, ACT Grant Agreement No 691712.

```
Anaconda Prompt - conda  install -c conda-forge pyomo

Options specific to MIP solver:
   --nomip            consider all integer variables as continuous
                      (allows solving MIP as pure LP)
   --first            branch on first integer variable
   --last             branch on last integer variable
   --mostf            branch on most fractional variable
   --drtom            branch using heuristic by Driebeck and Tomlin
                      (default)
   --pcost            branch using hybrid pseudocost heuristic (may be
                      useful for hard instances)
   --dfs              backtrack using depth first search
   --bfs              backtrack using breadth first search
   --bestp            backtrack using the best projection heuristic
   --bestb            backtrack using node with best local bound
                      (default)
   --intopt           use MIP presolver (default)
   --nointopt         do not use MIP presolver
   --binarize         replace general integer variables by binary ones
                      (assumes --intopt)
```

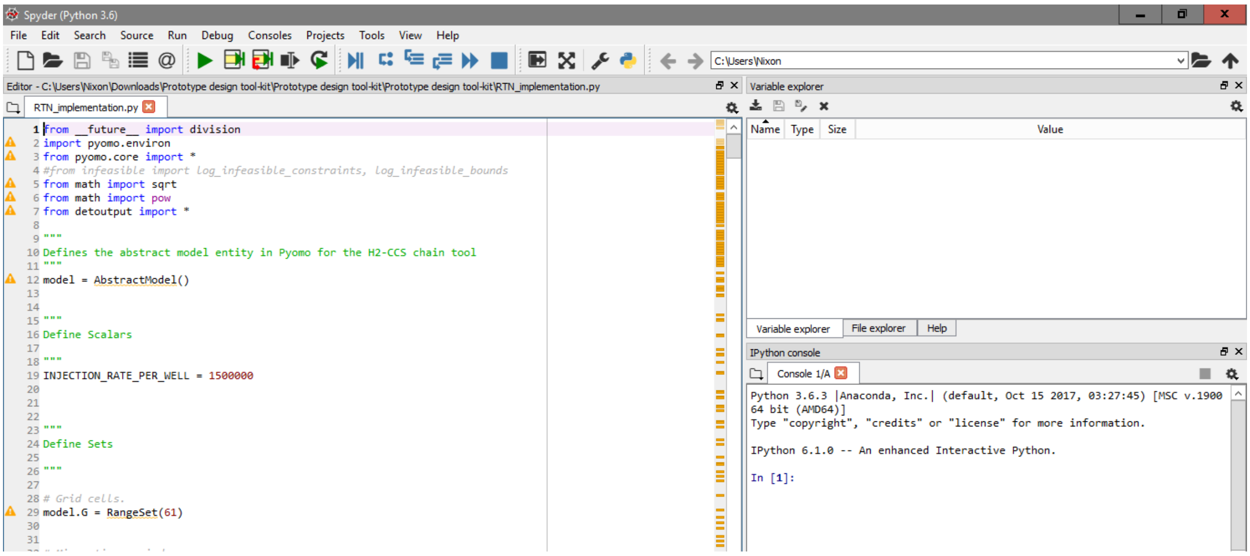
1.1.4 How do I interact with the model?

It should be noted here that the design tool is not released with a Graphical User Interface. Nevertheless, users are expected to be able to interact with the model equations in Python. If you're a regular Python user, open your favourite text editor and start editing. If you're a new user and would enjoy a scientific Python IDE, you can now open `Spyder', which is an editor embedded in the Anaconda installation. Upon opening Spyder, you may see a similar looking screen depending on your settings.



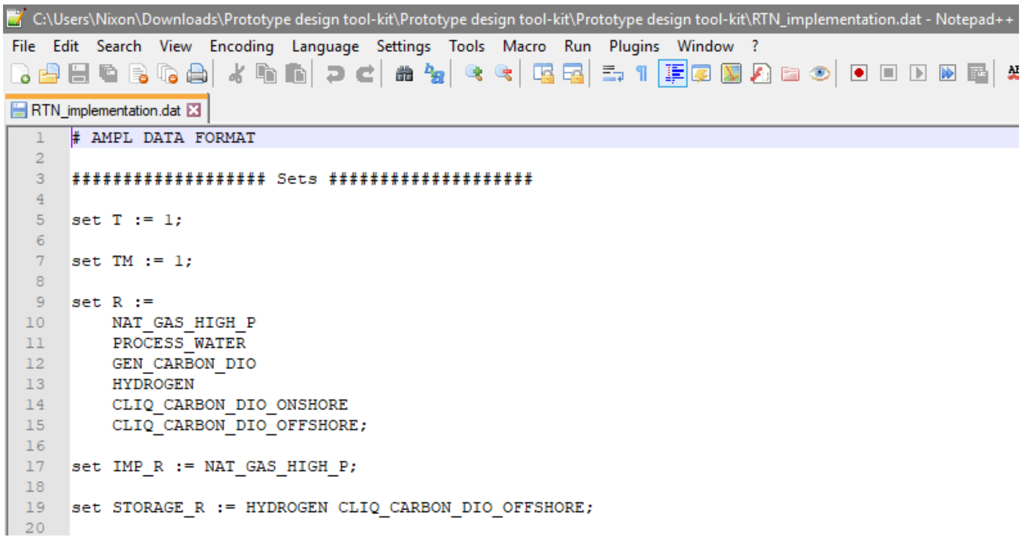
ACT ELEGANCY, Project No 271498, has received funding from DETEC (CH), BMWi (DE), RVO (NL), Gassnova (NO), BEIS (UK), Gassco, Equinor and Total, and is cofunded by the European Commission under the Horizon 2020 programme, ACT Grant Agreement No 691712.

You may open the file titled `RTN_implementation.py' and view/ edit its contents. You will be editing this file mainly for making any modifications to model equations or parameters. Once opened, your screen should look similar to the screenshot below. For further details of the modelling components, please see the next section along with the tool-kit specifications.



1.1.5 How do I edit the model parameter data?

You may open the file, titled `RTN_implementation.dat' for editing the data. The image below should resemble the contents of the data file. Please note that you could alternatively use Python's inherent data structures for `holding' the data.

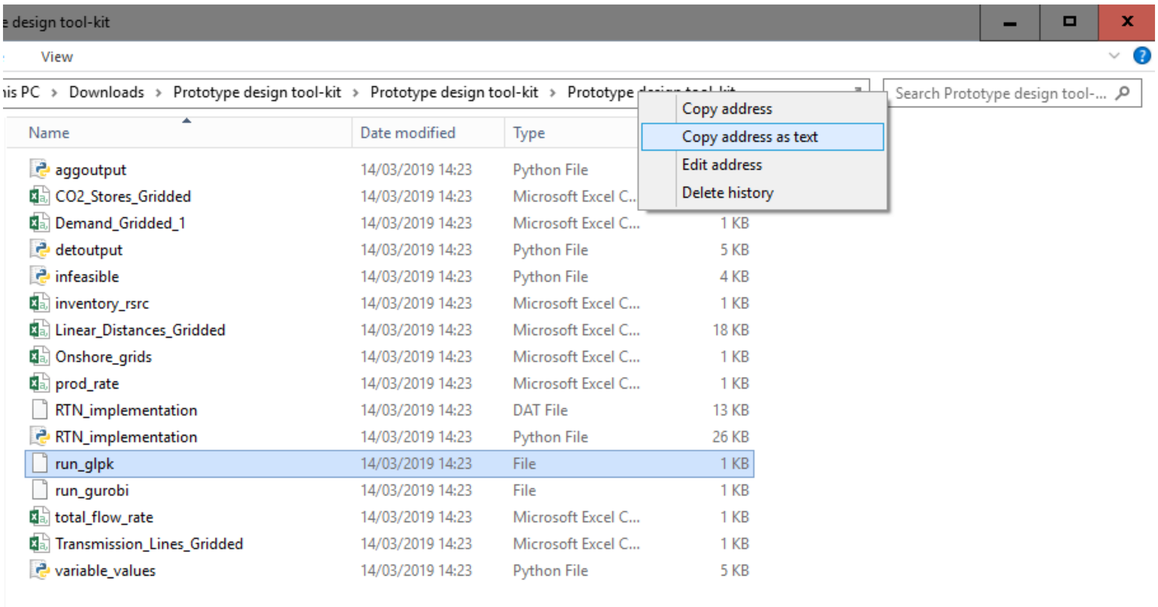


1.1.6 How do I run the model?

For general information on running and solving Pyomo models, you may visit the following link: https://pyomo.readthedocs.io/en/latest/solving_pyomo_models.html. For this particular model, copy the directory path using file explorer. Then, you should open up the command prompt/ Anaconda prompt. Type "*cd /pathname*". An example of how your screen might look is shown below. Following which, type `*pyomo solve RTN_implementation.py RTN_implementation.dat* --

ACT ELEGANCY, Project No 271498, has received funding from DETEC (CH), BMWi (DE), RVO (NL), Gassnova (NO), BEIS (UK), Gassco, Equinor and Total, and is cofunded by the European Commission under the Horizon 2020 programme, ACT Grant Agreement No 691712.

solver glpk --stream-solver --solver-options="mipgap=0.05" '. Then hit the "Enter" key. You should see the following screen as the model solves. The option "mipgap" can be used to determine the stopping criterion of the model.




```
Anaconda Prompt - conda install -c conda-forge pyomo
(base) C:\> cd C:\Users\Nixon\Downloads\Prototype design tool-kit\Prototype design tool-kit\Prototype design tool-kit
```

As a user, you do not need to do anything at this point. The model will continue to run until it has reached the stopping criterion/ found a satsifactory solution. Upon which, output files will be automatically written and subsequently, they may be used to scrutinise the model outputs and to ensure that you achieve relevant results. You can terminate a run using CTRL + C.

```
Anaconda Prompt - conda install -c conda-forge pyomo - pyomo solve RTN_implementation.py RTN_implementation.dat --solver glpk --stream-solver --solver-options="mipgap=0.05"
[   10.83] Applying solver
GLPSOL: GLPK LP/MIP Solver, v4.65
Parameter(s) specified in the command line:
 --mipgap 0.05 --write C:\Users\Nixon\AppData\Local\Temp\tmpvikmd2hx.glpk.raw
 --wglp C:\Users\Nixon\AppData\Local\Temp\tmpj3pnp05q.glpk.glp --cpxlp C:\Users\Nixon\AppData\Local\Temp\tmp_v5spkpd.pyomo.lp
Reading problem data from 'C:\Users\Nixon\AppData\Local\Temp\tmp_v5spkpd.pyomo.lp'...
97299 rows, 122430 columns, 334887 non-zeros
61244 integer variables, none of which are binary
811267 lines were read
Writing problem data to 'C:\Users\Nixon\AppData\Local\Temp\tmpj3pnp05q.glpk.glp'...
711954 lines were written
GLPK Integer Optimizer, v4.65
97299 rows, 122430 columns, 334887 non-zeros
61244 integer variables, none of which are binary
Preprocessing...
62892 rows, 88825 columns, 206400 non-zeros
59426 integer variables, 29280 of which are binary
Scaling...
 A: min|aij| =  3.700e-05  max|aij| =  2.634e+06  ratio =  7.119e+10
GM: min|aij| =  3.269e-01  max|aij| =  3.059e+00  ratio =  9.358e+00
EQ: min|aij| =  1.083e-01  max|aij| =  1.000e+00  ratio =  9.236e+00
2N: min|aij| =  1.169e-01  max|aij| =  1.717e+00  ratio =  1.468e+01
Constructing initial basis...
Size of triangular part is 62891
Solving LP relaxation...
GLPK Simplex Optimizer, v4.65
62892 rows, 88825 columns, 206400 non-zeros
      0: obj =   3.947073900e+06 inf =   1.588e+05 (12)
Perturbing LP to avoid stalling [200]...
    307: obj =   8.598558740e+06 inf =   3.469e-12 (0) 1
*  2314: obj =   7.006715825e+06 inf =   0.000e+00 (4553) 18
*  4533: obj =   6.563787503e+06 inf =   0.000e+00 (4073) 19
*  6524: obj =   6.114768411e+06 inf =   3.255e-09 (6696) 18
```

ACT ELEGANCY, Project No 271498, has received funding from DETEC (CH), BMWi (DE), RVO (NL), Gassnova (NO), BEIS (UK), Gassco, Equinor and Total, and is cofunded by the European Commission under the Horizon 2020 programme, ACT Grant Agreement No 691712.

2 MODEL COMPONENTS

This section details the key aspects to consider and modify when using the design tool for analysing specific case studies. For more information on the Pyomo modelling environment and its inherent functions, please refer to their extensive documentation at the following link: https://pyomo.readthedocs.io/en/latest/. Brief descriptions of the model components are provided in the subsequent sections.

2.1 Model Sets

2.1.1 Grid cells

This is a mathematical set in which the elements represent all the discretised locations in the spatial region of interest. If you've used the GIS input tool, you will note that you had to enter a "Desired number of cells", which was used to create a spatial mesh of grids. Ultimately, the number of grid cells in the spatial mesh may have been identical to your choice of grids or ± 5%. Nevertheless, you will be able to note the number of grids from the CSVs that were created using the tool. Thus, you should replace the default value of the number of grids based on your spatial mesh.

If you have not used the GIS input tool and you have a spatial mesh/ spatial distribution of zones, then you may simply type the total number of grids/ zones that were created using your method.

```
# Grid cells.
model.G = RangeSet(61)
```

2.1.2 Resource sets

There are three distinct resource sets that are used in this model. They are as follows:

- A set of all resources which contains all the potential resources within the system.
- A subset which contains only the importable resources in the system.
- A subset which contains only the resources that are stored in the system.

It is advisable to use the default resource sets unless you have any additional resources which are not currently present in the system for which you can and wish to explicitly define resource conversion coefficients, distribution parameters and performance metrics. Although many additional resources are utilised in the technologies within this model, only the key resources for which, the technological parameters can be explicitly provided are used in the model. A key point to note is that the difficulty in solving the problem increases with increases in the size of the resource set.

ACT ELEGANCY, Project No 271498, has received funding from DETEC (CH), BMWi (DE), RVO (NL), Gassnova (NO), BEIS (UK), Gassco, Equinor and Total, and is cofunded by the European Commission under the Horizon 2020 programme, ACT Grant Agreement No 691712.

```
# All resources.
model.R = Set(ordered=True)


# Importable resources.
model.IMP_R = Set(within=model.R)


# All resources that can be stored in this formulation.
model.STORAGE_R = Set(within=model.R)
```

Obviously, if your respective case study does not involve heating network design, then its superfluous to include “DOMESTIC_HEAT” and “INDUSTRIAL_HEAT” as resource elements. Thus, it is important to ensure that you add all the resources that are necessary for your analysis and remove those that are not.

2.1.3 Major and minor time periods

It should be noted that all the temporal aspects within the model will operate with a base unit. This must be consistent across all the different operations in the network for it to be systematic and avoid any unit inconsistencies. For example, if the production rates of H_2 and capacities are described in terms of MW, then other rates such as flow rates and storage/ retrieval rates must be described in terms of MW for H_2 as they are coupled by the material balance equation. The base unit of time in this instance is seconds as model entities are described in terms of their values per second. Otherwise, you may use different units to report capacities and rates provided that you modify the material balance equation to ensure consistency across all units. However, as abstract modelling is preferred to ensure minimal edits to the equality constraints, it is desirable to use a consistent time unit across the different operations. This has a distinct effect on the choice of the minor and major time periods within the problem instance.

Suppose that the rates were reported per day instead of seconds as in the previous example. If you intend to study the evolution of the network over a 5-year time period, you may choose to make the investment related decisions once every year. This indicates that the number of major time periods is 5. This is due to the fact that major time periods are used to describe the discrete variables (often investment decisions) in the system. Since you intend to make that decision every year, it logically follows that you can only decide 5 times within a 5-year time horizon. This information can be used to update the value of the major time period. Once the major time period has been fixed, this influences the choice of the number of the minor time periods. The minor time period is used to describe the operational decisions, such as time-dependent production, flow, storage rates, etc. You may find that there is no significant variation in resource demand across different days in the same season. Thus, you may choose to split the yearly interval into 3 or 4 seasons depending on the data. Therefore, the number of minor time periods may equal 3 or 4 and this can be amended in the sets accordingly. If instead, there are significant variations within the days in the same season but not within each month, then you may choose to use 12 minor time periods. Thus, the choice of minor time periods must be dependent on the variability in the demand for a particular resource within the system. The minor time periods and the base unit must be chosen to capture the nuances of the system. You may edit both major and minor time periods after appropriate discretisation has been selected. The mathematical model and the GIS output tool

ACT ELEGANCY, Project No 271498, has received funding from DETEC (CH), BMWi (DE), RVO (NL), Gassnova (NO), BEIS (UK), Gassco, Equinor and Total, and is cofunded by the European Commission under the Horizon 2020 programme, ACT Grant Agreement No 691712.

relies on a description of both the major and minor time periods via numerical types as opposed to strings so you must adhere to this format if you intend to utilise these tools.

```
# Minor time periods.
model.T = Set(ordered=True)


# Major time periods.
model.TM = Set(ordered=True)
```

2.1.4 Performance metrics

The set of all performance metrics can contain various metrics that are defined based on the network operations. These metrics may be used in the computation of the objective function or as part of constraints. For example, these performance metrics may constitute of total costs in the form of CAPEX and OPEX as well as other metrics such as primary energy consumption. Such a metric may be valuable as it is possible to optimise for a network in which the primary energy consumption is minimised. Following which, the solution can be used with an associated margin as a constraint to obtain cost-effective network designs which also minimises the primary energy consumption by minimising the total costs. These metrics may also be relevant for multi-objective optimisation problems when considering the development of robust designs. As a user, you are encouraged to scrutinise the default options and edit them as you deem fit for the purposes of your case study. The default performance metrics are shown in the figure below.

```
set M := CAPEX
         OPEX
         CLIMATE_CHANGE_TOT
         FRESHWATER_AND_TERRESTRIAL_ACIDIFICATION
         FRESHWATER_ECOTOXICITY
         MARINE_EUTROPHICATION
         TERRESTRIAL_EUTROPHICATION
         CARCINOGENIC_EFFECTS
         IONISING_RADIATION
         NON_CARCINOGENIC_EFFECTS
         OZONE_LAYER_DEPLETION
         PHOTOCHEMICAL_OZONE_CREATION
         RESPIRATORY_EFFECTS
         FOSSIL_RESOURCE_USE
         LAND_USE
         MINERALS_AND_METALS
         FOSSIL_ENERGY_DEMAND
         NUCLEAR_ENERGY_DEMAND
         PRIMARY_FOREST_ENERGY_DEMAND;
# Set M contains performance metrics to compute.
```

2.1.5 Technology sets

Technology sets are used to describe all the process and storage technologies within the system. Altogether, there are 4 technology sets. The sets are summarised here for you to quickly gauge its relevance for your case study.

ACT ELEGANCY, Project No 271498, has received funding from DETEC (CH), BMWi (DE), RVO (NL), Gassnova (NO), BEIS (UK), Gassco, Equinor and Total, and is cofunded by the European Commission under the Horizon 2020 programme, ACT Grant Agreement No 691712.

- A set which contains all the process (production, compression, purification) and storage technologies.
- A subset which contains only the storage technologies.
- A subset which contains only the process technologies.
- A subset of the storage technologies which indicate geological storage technologies.

You will require all of these sets albeit with different elements depending on your case study.

```
# All technologies.
model.J = Set(ordered=True)


# Process technologies.
model.PROCESS_TECH = Set(within=model.J)


# Storage technologies.
model.STRG_TECH = Set(within=model.J)


# Geological storage technologies.
model.GEOLOGICAL_TECH = Set(within=model.STRG_TECH)
```

2.1.6 Distribution technologies

In a similar manner to the technology sets that were described above, distribution technology sets are described separately although this is not necessary. The distinction is suggested for a few reasons that are summarised later in the Parameter section. The default set elements contain natural gas, H_2 and CO_2 pipelines which are described based on their diameters. In the case of the CO_2 pipelines, a distinction is made between pipes of the same diameters depending on whether they are onshore or offshore pipelines. H_2 pipelines are assumed to be capable of transporting H_2 at various pressure tiers at assumed pressure drops. The estimates for these pipe flow rates should be computed based on analytical pressure drop calculations for the specific case study given the regional cell sizes. This allows for an explicit description of compression along the pipeline segments albeit at the expense of increased computational complexity. The maximum allowable inlet pressure of H_2 in the pipe depends on the pipe schedule and other dimensions. Other distribution technologies such as trucks and ships may also be incorporated based on specific user needs.

ACT ELEGANCY, Project No 271498, has received funding from DETEC (CH), BMWi (DE), RVO (NL), Gassnova (NO), BEIS (UK), Gassco, Equinor and Total, and is cofunded by the European Commission under the Horizon 2020 programme, ACT Grant Agreement No 691712.

```
# Distribution technologies.
model.D = Set(ordered=True)


# Hydrogen pipes.
model.HYDROGEN_PIPES = Set(within=model.D)


# CO2 pipes.
model.CO2_PIPES = Set(within=model.D)


# Onshore CO2 pipes.
model.ONSHORE_PIPES = Set(within=model.D)


# Offshore CO2 pipes.
model.OFFSHORE_PIPES = Set(within=model.D)


# Discrete operating modes for distribution tech.
model.DIST_MODE = Set(ordered=True)
```

2.1.7 Scenarios

The scenarios set is used to define the various scenarios that may be studied using a stochastic model. In deterministic applications (i.e., model parameters are all known exogenously), the size of this set reduces to a single element reflecting a single, absolute scenario under investigation.

```
# Scenarios.
model.S = Set(ordered=True)
```

ACT ELEGANCY, Project No 271498, has received funding from DETEC (CH), BMWi (DE), RVO (NL), Gassnova (NO), BEIS (UK), Gassco, Equinor and Total, and is cofunded by the European Commission under the Horizon 2020 programme, ACT Grant Agreement No 691712.

2.2 Model Parameters

Parameters are often indexed by set elements to note the individual contribution of each element to the evaluation of the overall objective. Indexing by set elements are not necessary as you may also have scalar entities which have fixed values during the simulation. A detailed overview of the key parameters within the modelling framework is presented here. The scripting notation is as follows: capitalisation is used for all sets and parameters that are not represented by one/two characters to easily distinguish them from variables in equality and inequality constraints. For further information on the creation of new model parameters, please refer to the documentation at: https://pyomo.readthedocs.io/en/latest/pyomo_modeling_components/Parameters.html

2.2.1 BETA

Parameter BETA is indexed by the set of major time periods. It represents the total proportion of generated CO_2 emissions that must be stored by that major time period. Naturally, the proportion fraction is bound to be between 0 and 1. This parameter is used in order to constrain the total amount of CO_2 released into the environment. Lower values of BETA allow for the usage of technologies with higher carbon intensities to achieve the demand whereas higher values of BETA enforce the adoption of technologies with lower carbon intensities. The parameter itself is indexed by major time periods to enable varying values of BETA to be enforced with time to enable the evolution of a “low-carbon H_2-CCS chain network”. The parameter values can be edited by a potential user and modified to reflect their individual case needs. The choice of BETA will also be dependent on national and EU related emissions targets. You may wish to omit the usage of BETA altogether to implement another form of emissions constraint such as those which are dependent on emissions intensity.

```
# Capture fraction of generated emissions.
model.BETA = Param(model.TM)
```

2.2.2 Demand parameter

The next key parameter to note is the demand parameter which describes the demand of a resource in any spatial location at any given time. Therefore, this parameter is indexed by the set of resources (r), set of spatial grids (g), set of minor and major time periods (t, tm) and the set of scenarios (s). This is a parameter which you must specify before use.

```
# Spatio-temporal discrete representation of demand for a given resource.
model.DEMAND = Param(model.R, model.G, model.T, model.TM, model.S, default=0, mutable=True)
```

2.2.3 Onshore grids

It is important to make regional classifications to distinguish regions in design problems which consider both onshore and offshore locations. This could be due to reasons such as costing differences for technologies between offshore and onshore regions alongside with technological limitations in deployment in offshore regions as opposed to onshore regions. This can usually be noticed using the GIS software and entered in the parameter as zeros and ones with 1 indicating that it is an onshore grid and 0 indicating that it is an offshore grid. It should be stressed here that

ACT ELEGANCY, Project No 271498, has received funding from DETEC (CH), BMWi (DE), RVO (NL), Gassnova (NO), BEIS (UK), Gassco, Equinor and Total, and is cofunded by the European Commission under the Horizon 2020 programme, ACT Grant Agreement No 691712.

the classification is dependent on whether a grid is offshore or at an intersecting boundary. All grids that are either offshore or at intersecting boundaries are considered to be offshore and given a value of 0 whereas all the other elements in the spatial grids set are given a value of 1, indicating that they are onshore grids. It is assumed that all grids are either offshore or onshore.

```
# Parameter to indicate if a region, g is onshore or offshore.
model.ONSHORE_GRIDS = Param(model.G, within=NonNegativeReals)
```

2.2.4 Nameplate capacity

The nameplate capacity of a technology is used to impose capacity constraints as part of the design problem. For process technologies, the nameplate capacity can be discussed in terms of their processing/ production capacity per base unit of time. For example, in MW for autothermal reforming of natural gas with CCS ("ATR_CCS"). For storage technologies, you normally state the capacity as independent of time (i.e., a fixed volume).

```
# Nameplate capacity of technologies.
model.NAME_PLATE_CAP = Param(model.J)
```

2.2.5 Operational times

The parameter titled "OPER_TIME" is used to describe the number of base units there are in each minor time period. If the number of minor time periods is equal to 12 (reflecting the number of months in a year) using the earlier example concocted, "OPER_TIME" will be indexed by the set of minor time periods with the value of the parameter being equated with number of days in each month. This is the operational time in this instance as the base unit considered for the analysis is in days. If the base unit was reported in hours with 12 minor time periods, constituting distinct months in the year, then the value of the parameter "OPER_TIME" must also be multiplied by the number of hours in each day. This parameter can also read from a CSV file but it can be entered in by hand if that is preferred.

```
# number of base time units within each minor time period.
model.OPER_TIME = Param(model.T)
```

The above figure reads the values of the parameters whilst using half an hour as the base unit.

2.2.6 Distance

The distance parameter denotes a table containing the distances between each grid cell where only the distances between contiguous grid cells are quantified. This is done via the GIS input tool automatically from the spatial layers and the coordinate reference system used. However, you may input this information yourself if this is preferred. The distance parameter is important as it allows for the transport links between grid cells in the spatial mesh to be effectively defined. The format by which a table can be loaded into Pyomo is shown in the data file RTN_implementation.dat.

```
# Distances between grid cells in km.
model.DISTANCE = Param(model.G,model.G)
```

ACT ELEGANCY, Project No 271498, has received funding from DETEC (CH), BMWi (DE), RVO (NL), Gassnova (NO), BEIS (UK), Gassco, Equinor and Total, and is cofunded by the European Commission under the Horizon 2020 programme, ACT Grant Agreement No 691712.

2.2.7 Flow capacity of transport technologies

The maximum flow capacity of a resource in a pipeline is dependent on the physical properties of the fluids, the physical dimensions of the pipeline, routing, pressure drop per unit length, etc. This parameter contains the values of maximum potential flow capacity for a resource per base unit of time. The parameter is indexed by the distribution technology and this value is determined using pressure drop calculations using assumed physical conditions and lengths. A flow modifier is used to mimic the tortuous nature of pipelines in comparison to the straight pipes that may be used in the analytical work.

2.2.8 Resource conversion coefficients

The parameter "RESOURCE_CONV_RATE" contains important information regarding the interconversion of a set of resources into another by a distinct process technology. This is particularly useful in characterising the technical performance of various technologies in terms of the resource requirements and production coefficients. The technological efficiency of each technology can be integrated directly into the conversion of resources rather than via external description in an alternative parameter. Based on the technologies that you may wish to use, these coefficients can be modified and altered to show differences in the technology standards across various spatial regions in addition to the evaluation of new technologies for analysis. An important point to note is that a negative sign is used to signify the consumption of a resource via a technology whereas a positive numerical is used to represent the production of a resource.

2.2.9 LCA score

The parameter "LCA_SCORE", "NETWORK_LCA", "STRG_LCA" and "RSRC_LCA" describes the impact of installing a technology or resource on the overall life-cycle performance metrics. "LCA_SCORE" characterises the various production and conversion technologies, although it is important to note that the upstream emissions from fuel supply are not included for the water electrolysis option in contrast with the other technology options. The "NETWORK_LCA" term describes the impact of installing transport and distribution modes on the life-cycle performance metrics. Similarly, the impact of investment and operation of injection wells and caverns are described through the "STRG_LCA" term. The impact of using electricity is described separately in the "RSRC_LCA" term, which allows a user to choose between the different sources of electricity depending on the analysed region.

2.2.10 Flow consumption and production coefficients

These parameters hold paradoxical names as there is no consumption or production occurrence during a flow of resource from one spatial location to another via a pipeline. Yet, these coefficients are used to describe the conversion of resources from one set to another as they highlight a change in the pressure tier of each resource. Pipelines may be assumed to have a pressure drop of *x* bars across its length and this assumption is used to compute its maximum potential flowrate across the various pipeline distribution modes. Pipelines are also assumed to be capable of operating in different modes, implying that it is possible for a pipeline of a given size to transport a H_2 stream at various discrete pressure levels. Therefore, at any point in time, the distribution technologies can transport H_2 from certain pressure tiers and this choice is a degree of freedom that can be incorporated into the model. Both flow consumption and production coefficients are indexed by the set of distribution technologies (d), the set of distribution modes (DIST_MODE) and the set of all resources (r). For example, a conversion of the resource ("HYDROGEN_60BAR") to the resource ("HYDROGEN_40BAR") via a distribution mode ("MODE_60_40") would record 1 for

ACT ELEGANCY, Project No 271498, has received funding from DETEC (CH), BMWi (DE), RVO (NL), Gassnova (NO), BEIS (UK), Gassco, Equinor and Total, and is cofunded by the European Commission under the Horizon 2020 programme, ACT Grant Agreement No 691712.

the flow consumption coefficient as opposed to -1 in the case of the parameter, “RESOURCE_CONV_RATE”. Similarly, the production coefficient for resource (“HYDROGEN_40BAR”) would also record 1 in the “FLOW_PRODUCTION_COEFF”.

2.2.11 Distribution network design effects

“NETWORK_COEFF” is a parameter used to describe the effects of installing a particular transportation technology on the total “CAPEX” and “OPEX” metrics as well as any other performance metrics analysed in the case study. The parameter itself contains the contribution per km of pipeline installed. Thus, the parameter is indexed by the set of all distribution technologies and the set of the performance metrics.

2.2.12 Investment effects

The parameter “INV_COEFF_GRID” is used to describe the contributions from investing in a particular technology on the overall value of the performance metric. The parameter is indexed by the set of all technologies (j), the set of performance metrics (m), set of major time periods (tm), set of all grid cells (g).

2.2.13 Process coefficients

“PROCESS_COEFF” is a parameter which is used to denote the effects of operating using process technologies on the performance metrics at varying time. It denotes the variable operating effects of using a technology as opposed to the fixed investment charges.

2.2.14 Import effects

The effects of importing any resource from the set of importable resources on the performance metrics are quantified by the “IMPORT_COEFF” parameter.

2.2.15 Import locations

The spatial grids in which certain resources can be imported are quantified using the import location parameter. This is only relevant for those resources that are not assumed to be available everywhere in the spatial region and require substantial infrastructural changes to ensure that they are available everywhere. For example, natural gas is considered only to be available in locations where there are import terminals/ existing natural gas transmission lines. The parameter can also be evaluated using GIS layers and the CSV file generated provides the information by default to the model instance.

2.2.16 Objective weightings

The parameter “OBJ_WEIGHT” describes the relative weightings attached to each metric in the overall objective function. For example, if CAPEX and OPEX are the only two performance metrics that are involved in the evaluation of the objective, calculating the objective would be a

ACT ELEGANCY, Project No 271498, has received funding from DETEC (CH), BMWi (DE), RVO (NL), Gassnova (NO), BEIS (UK), Gassco, Equinor and Total, and is cofunded by the European Commission under the Horizon 2020 programme, ACT Grant Agreement No 691712.

simple sum of the individual CAPEX and OPEX elements across all the time periods. Thus, the objective weightings will equal 1 across the time periods (i.e., sum the product of CAPEX and 1 with the product of OPEX and 1, across the time horizon). This would give an estimate of the total costs which can be used as the objective function. It is also possible to use other objectives by using a relative weighting that can be used to normalise the value of a metric into another. To illustrate this, suppose that there is now an additional performance metric – CO_2 emissions. Since the emissions metric is quantified in metric tonnes as opposed to monetary value, it is not possible to simply sum the metrics in order to evaluate a potential objective function. However, attaching a CO_2 price enables the metric to be quantified in monetary terms and hence, the "OBJ_WEIGHT" parameter value for this particular value would equal the price.

2.2.17 Total storage capacity for CO_2

Similar to the H_2 geological storage parameters, geological storage of CO_2 is described using a CSV file which details the total geological storage capacity in Mt. The total static storage capacity is computed in the GIS input tool using aggregation techniques to provide an informed estimate. The parameter is simply indexed by the set of all spatial cells and the parameter holds a value of 0 if there are no geological storage sites in a grid cell.

ACT ELEGANCY, Project No 271498, has received funding from DETEC (CH), BMWi (DE), RVO (NL), Gassnova (NO), BEIS (UK), Gassco, Equinor and Total, and is cofunded by the European Commission under the Horizon 2020 programme, ACT Grant Agreement No 691712.

2.3 Model Variables

The model contains both discrete and continuous variables which are used to describe various types of decisions. Usually, discrete variables are used to describe the investment decisions in the model such as the number of pipelines of a given size to be built between two grid cells or the number of process facilities to install in any given location. Continuous variables, on the other hand, are used to describe operational decisions within the physical system. In this particular case, they are mainly used to describe the material balance terms which include production, transportation, storage and retrieval rates of a resource. However, continuous variables are not limited to operational decisions as they are also used to compute other key state variables in the system. Discrete variables are also used to describe conditional constraints in this model in addition to investment decisions. In general, it is a good idea to limit the number of integer/ binary variables that are present in the system to minimise the computational difficulties that may arise when solving large problems. It should be noted that every model variable in this model formulation is constrained to be a positive variable. A brief overview of the model variables is provided to enhance user understanding in this section. For more details on the formulation of new variables in the model, please refer to further documentation at: https://pyomo.readthedocs.io/en/latest/pyomo_modeling_components/Variables.html.

2.3.1 Material balance variables

The material balance constraint (discussed later) contains six variables to note. The constraint is enforced across every time period in the time horizon and the variables themselves are described in terms of the quantity of a resource per base unit of time. The six variables are as follows:

- Production rate using a technology.
- Rate of retrieval from a storage technology.
- Rate of storage into a storage technology.
- Emission rate of a resource.
- Flow rate of a resource from current location to another.
- Import rate of a resource.

The variables are indexed slightly differently depending on the nature of the variable, this will be clear once the equations are presented. You are recommended to use these distinctions or use any alternative variables to describe additional terms in the material balance. You are also encouraged to remove any terms that you deem unfit in order to improve its applicability to your own case study.

2.3.2 Inventory variables

Two continuous variables are used to denote the initial inventory along with inventory of a resource in a storage technology at any given time in any location. These variables are used to describe the amount of resource stored in a storage technology at various points in time, thus showing the overall retrieval potential.

ACT ELEGANCY, Project No 271498, has received funding from DETEC (CH), BMWi (DE), RVO (NL), Gassnova (NO), BEIS (UK), Gassco, Equinor and Total, and is cofunded by the European Commission under the Horizon 2020 programme, ACT Grant Agreement No 691712.

2.3.3 Number of storage technologies

Typically, variables which are used to quantify a certain quantity of items in discrete terms are described using discrete variables. Although that is the most accurate approach, great improvements in solution times can be achieved by treating some of these variables as continuous. This is typically referred to as the relaxation of the integrality condition. When the value of a discrete variable is anticipated to be large, the differences between a continuous relaxation of the solution and the discrete value itself are comparatively small. Thus, it may be acceptable to enforce this as a "soft" constraint via the relaxation of the integrality condition. However, you may choose not to do this and describe these terms as discrete and this is perfectly acceptable.

2.3.4 Total metrics and objective variables

The total metrics variable is used to compute the objective function in the problem using some parameters that were mentioned earlier. This particular variable sums the corresponding effects of installing process, transport and storage technologies on the value of the performance metrics across the entire time horizon and the entirety of the spatial region. When combined with the relative weightings attached to each performance metric in the overall objective, this information can be utilised to compute the objective function of the problem instance. This variable also allows a user to implement multi-objective optimisation strategies using methods such as the epsilon-constraint method to generate a set of Pareto-efficient solutions.

2.3.5 Number of process and distribution technologies

Both of these variable blocks quantify the investment decisions associated with installing process and transportation technologies. Unlike storage technologies, these variables are assumed to hold comparatively smaller values. Therefore, it is not advisable to define these entities as continuous variables. Accordingly, they are described as discrete integer variables in this model formulation.

2.4 Variable Bound Constraints

This section provides a very brief overview of the key bound constraints that are enforced on the variables in the system.

2.4.1 Import rate variable bounds

It is immediately possible to see that import rates might be non-zero for importable resources, so this is a statement that is mathematically enforced. The fact that import rates must equal zero for non-importable resources is also embedded within the model. The second bound constraint provides an upper bound constraint which limits the maximum import rate of an importable resource to that described using the "IMPORT_RSRC_MAX" parameter.

2.4.2 Emission rate bounds

It is important to limit the emission variable only to those resources that are emitted, which in this model, solely refers to the CO_2 emitted in the flue gas.

ACT ELEGANCY, Project No 271498, has received funding from DETEC (CH), BMWi (DE), RVO (NL), Gassnova (NO), BEIS (UK), Gassco, Equinor and Total, and is cofunded by the European Commission under the Horizon 2020 programme, ACT Grant Agreement No 691712.

2.4.3 Inventory and storage related bounds

Inventory for resources that are not stored in the system must equal zero at all times in all locations. This is enforced via a series of bound constraints. The first bound constraint enforces that none of the CO_2 sequestration sites have a non-zero inventory in the beginning time period. The subsequent constraints are used to limit the initial inventory, storage and retrieval rates to zero for resources that are not stored in this formulation.

2.4.4 Discrete variable bounds

It is highly beneficial to explicitly define lower and upper bounds for the investment decisions that are undertaken in the model. This allows for the reduction of the search space and the shrinkage of the feasible region. Therefore, the process by which the solution is found is generally quicker than if the default bounds were imposed on the investment variables. However, it should be noted that the bounds imposed on these discrete process and transportation variables must be practical and reflect the nature of the physical system. They may not be concrete bounds and can be varied in different scenarios.

2.4.5 Specific transportation bound constraints

Some specific variable constraints must be enforced depending on the technologies that can be deployed between specific regions. For example, users must ensure that it is not possible to have onshore pipelines in offshore grids for rather obvious reasons in defining costs. Although offshore pipelines are not substantially different to those onshore, the distinction is made in the model formulation due to the non-trivial costing differences in establishing an offshore pipe as opposed to an onshore one.

ACT ELEGANCY, Project No 271498, has received funding from DETEC (CH), BMWi (DE), RVO (NL), Gassnova (NO), BEIS (UK), Gassco, Equinor and Total, and is cofunded by the European Commission under the Horizon 2020 programme, ACT Grant Agreement No 691712.

2.5 Model Equations and Constraints

There are many constraints which acts beyond the variable bounds in order to reflect the physical system more accurately such as the material balance constraint. The broader distinction is that every equality constraint is considered as a model equation, which means that it is possible to use it to compute other variables such as state variables. The number of free variables or optimisation variables in the system can be determined through a degree of freedom (DOF) analysis. Equality constraints are considered along with the list of all variables to analyse the number of free variables. Inequality constraints are used to define a feasible region or a set of feasible solutions that may be valid as potential solutions to the problem. As mentioned earlier, a user needs to be careful when strictly enforcing equality constraints as they are adding model equations. If this addition isn't accompanied with an increase in the number of variables, the number of free variables in the system decreases with each unique addition of an equality constraint. In the model formulation, you may choose to modify the existing equality constraints to edit the existing terms that are already present in the equations or to modify and add/ remove any new terms. You may also/instead choose to interact primarily with the inequality constraints within the system rather than the equality constraints. The purpose of this section is not to describe every equation in detail but rather to highlight the format of the equations and describe the key constraints that are enforced in the model for illustration purposes. It is the authors' impression that this will allow potential users to familiarise themselves with the model equations and make modifications when applying it to other case studies. In Pyomo, functions are used to generate "rules" which can subsequently be used for the generation of constraints. It might be a useful exercise to find suitable names for model equations. For more information on the formulation of new equations, please refer to https://pyomo.readthedocs.io/en/latest/pyomo_modeling_components/Constraints.html.

2.5.1 The objective function

The equation used to describe the objective function in this model is shown by the following.

```
def obj_rule(model):
    """ Defines the objective function for the optimisation process."""
    return sum(sum(sum(model.OBJ_WEIGHT[m,tm]*model.total_metrics[m,tm,s]*model.PROBABILITY[s]
    for tm in model.TM) for m in model.M) for s in model.S)

model.obj = Objective(rule=obj_rule,sense=minimize)
```

As mentioned before, all variables are listed in small letters and all subsets and parameters are typed in capital letters. This allows you to deduce that the term "OBJ_WEIGHT" in the above image is a parameter particularly as it is indexed by two sets. The term "PROBABILITY" is also a parameter due to the fact that it is capitalised. However, the term "total_metrics" is not capitalised and hence, must be a variable. This same logic can be applied to every equation within the model to enable quick analysis of the equations and the corresponding parameters and variables within it. The equation itself equates the objective variable value to the summation of the product of total metrics and objective weightings. In a stochastic simulation, there is more than one probable scenario and hence the "PROBABILITY" term will not necessarily equal 1 and will be part of the product. It should also be noted that the objective term is a scalar entity and as it must be scalar by necessity, any equation which computes the objective must involve a summation across the entire set of performance metrics, major time periods and scenarios.

ACT ELEGANCY, Project No 271498, has received funding from DETEC (CH), BMWi (DE), RVO (NL), Gassnova (NO), BEIS (UK), Gassco, Equinor and Total, and is cofunded by the European Commission under the Horizon 2020 programme, ACT Grant Agreement No 691712.

2.5.2 Technology balance

These equations are repeated in different forms to describe the technology balance for transportation and storage technologies.

```
def eqn2_rule(model,process_tech,g,tm,s):
   """ Process technology capacity balance in each grid cell."""
   if tm > 1 :
      return model.num_process[process_tech,g,tm,s] == model.num_process[process_tech,g,tm-1,s] \
        + model.num_process_invest[process_tech,g,tm,s]
   else :
      return model.num_process[process_tech,g,tm,s] == model.num_process_invest[process_tech,g,tm,s] \
        + model.INIT_TECH[process_tech,g,tm]
model.eqn2 = Constraint(model.PROCESS_TECH, model.G, model.TM, model.S, rule=eqn2_rule)
```

The indices are used in equations to denote the validity of the equation across multiple sets. In these equations, it is clear that as the variable which describes the number of process technologies is indexed by the set of process technologies, grid cells, major time and scenarios, the actual equation which describes the computation of the variable is also indexed by the same sets. In the above snippet, "eqn2" computes the number of process technologies in each major time period. The parameter "INIT_TECH" may be used to input existing numbers of process technologies into the equation. By default, it is assumed that none exists. Therefore, the number of process technologies that are available at the first major time period must equal the number of process technologies that are built in the first major time period. The second equation in the snippet is used to show how the equation evolves at major time periods greater than 1. It can also incorporates lifetime of process technologies to describe the decommissioning of facilities provided that it is incorporated as a model parameter.

2.5.3 Material balance equation

The material balance equation is enforced at every time point in the time horizon and it combines several operational variables that were described in the variables section. The material balance equation is applied on a fictional control volume which encompasses each and every individual grid cell but treats the storage facilities as being distinct from the control volume. Since the equation enforces conservation of material, the positive and negative terms must be balanced. The positive terms are comprised of import, production, inflow, retrieval rates of a resource whereas the negative terms are comprised of the demand, emission rate, outflow, storage rates.

2.5.4 Flow capacity equation

The flow, production and storage capacity limits are analogous to one another. Only the flow capacity equation is discussed here. This equation ensures that the sum of the total flowrate across all the various operating modes of a distribution technology in any given time must be less than the total flow capacity between those locations. A similar equation describes production rate limits and limits on the inventory of a resource in a storage technology.

2.5.5 Total metrics equation

This equation combines the effects of investment in process, transport and storage technologies along with the effects due to the operational decisions. This should provide you with an overview of the level of interaction with the modelling platform. You may choose to adopt it entirely and simulate with the default parameters that reflect UK characteristics or you may alter it to reflect other geographical regions.

ACT ELEGANCY, Project No 271498, has received funding from DETEC (CH), BMWi (DE), RVO (NL), Gassnova (NO), BEIS (UK), Gassco, Equinor and Total, and is cofunded by the European Commission under the Horizon 2020 programme, ACT Grant Agreement No 691712.

3 CASE STUDY PREPARATION

Before running a case study, it is important to look through the model components detailed in the previous section and consider whether they are required for the specific application of the tools. Many of the equations and constraints are only valid for specific instances of the problem. It is important that you isolate the functions that are of need and discard those which are redundant. You are encouraged to contact the developers if you are uncertain about the relevance of certain model components for your use. The remainder of this section gives brief instructions on how to proceed when using these tools to analyse new case studies.

3.1 Planning the case study

This step is the extremely crucial for the exposition of a carefully crafted design problem that can be solved within the limitations of computational power and resources. The planning stage must consider the length of the case study (weeks, months, years, decades, etc.), the design features and technological options, spatial resolution, objectives, etc. The planning phase must consider the goals of the project – design a H_2-CCS network to deliver a certain quantity of H_2? Design a network for the provision of heat/ transportation fuel? Understand the trade-off between different technological options/ scales, etc. The objective of the problem must be clearly stated and all the decision variables clearly outlined with well defined constraints. All modifications should be clearly stated with justifiable assumptions and reported alongside the results.

3.2 Editing the model components

At this stage, it is hoped that you will edit the model components and data to account for new options or to remove certain items from the sets that are not relevant to your own case studies. For example, the inclusion of new resources/ technologies. You are also expected to advance through the parameters section, modifying and editing the values to reflect the differences in case study parameters. You should scrutinise the variables and equations sections in the model for its relevance. It is assumed that you will have the relevant spatial datasets at your disposal for interaction with the GIS input tool or by any other means. Please check the Pyomo user's guide for details on making edits, adding new parameters, etc.

3.3 Using the GIS input tool

Generate location-dependent aggregate data of geological features as well as parameters such as demand through the GIS tool or otherwise. If the GIS tool is used to generate the files, please ensure that the column headers are modified in the CSV files according to the sample headings in the appendix before using Pyomo. These headers are not removed by default so that you can cross-check that the correct columns have been aggregated and reported using the tool. Once the input tool is used, the number of grid cells that were generated can be easily seen via simple observation of any of the CSV files that were generated. This information can be used to update the set of all spatial grids (g). Some of the CSV files such as the file containing linear distances can be used for pressure drop calculations which may then inform estimates. This might not be relevant at all if different pressure tiers are not considered and pressure drop is not of a major concern.

ACT ELEGANCY, Project No 271498, has received funding from DETEC (CH), BMWi (DE), RVO (NL), Gassnova (NO), BEIS (UK), Gassco, Equinor and Total, and is cofunded by the European Commission under the Horizon 2020 programme, ACT Grant Agreement No 691712.

A.1 GIS data curation for Pyomo

Please note how the column headers have been modified to enable delimited files to be read the data quickly. The figure below demonstrates how the .CSV file contents appear before coupling with AMPL data files. This process must be replicated for each .CSV file that you wish to incorporate into the model.

```
Grid_id,HYD_DEMAND
1,0
2,129
3,0
4,0
5,0
6,408
7,304
```

The contents in the .dat file must also indicate the column headers to use for identifying the parameter sets as shown by the "Grid_id" column below. For more information on this, please refer to https://pyomo.readthedocs.io/en/latest/working_abstractmodels/data/datfiles.html.

```
load Demand_Gridded_1.csv using=csv : [Grid_id] HYD_DEMAND;
```

You are also encouraged to use the DataPortal class for storing data, however this might involve a deeper understanding of the data structures within Python and so it has been omitted for the benefit of the broadest range of users.

ACT ELEGANCY, Project No 271498, has received funding from DETEC (CH), BMWi (DE), RVO (NL), Gassnova (NO), BEIS (UK), Gassco, Equinor and Total, and is cofunded by the European Commission under the Horizon 2020 programme, ACT Grant Agreement No 691712.