ISSN 1831-9424

European Commission

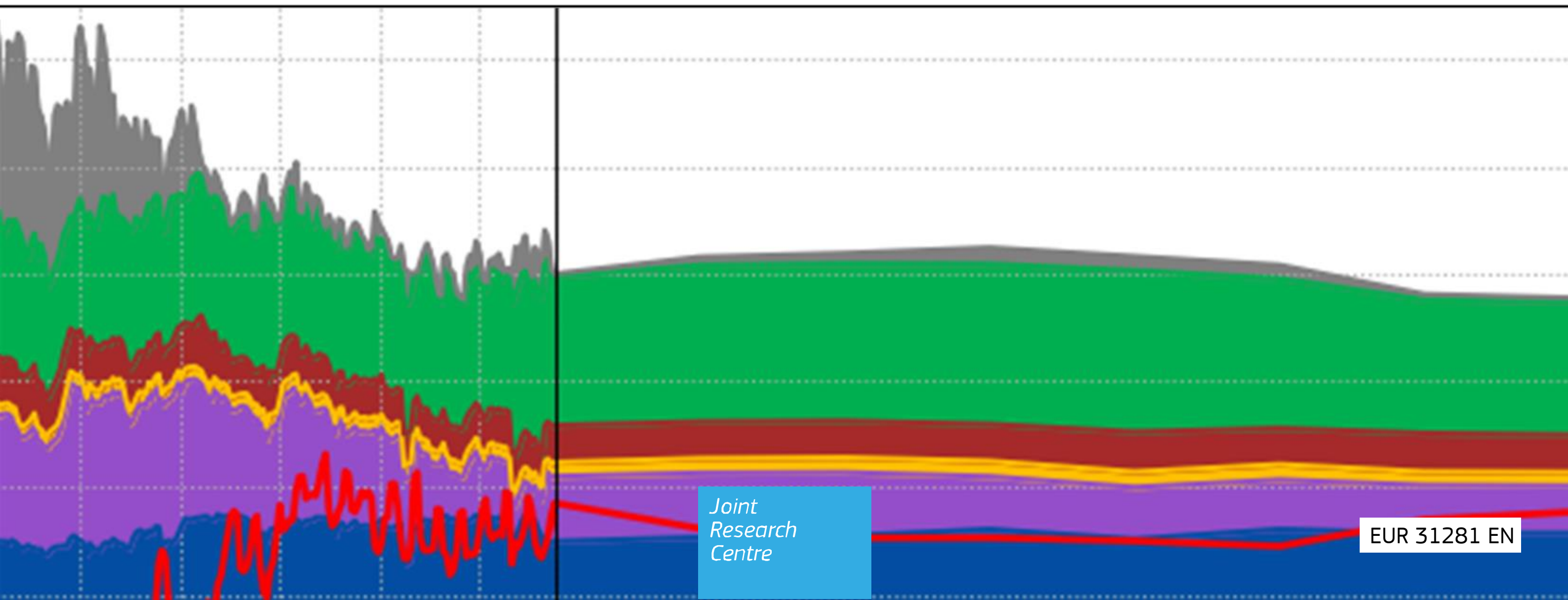
JRC TECHNICAL REPORT

Tools for analysing the European natural gas system with public data – The Python package `eurogastp`

eurogastp v0.1.0
Technical Manual

Jung, D., Vuillaume, J.-F., Fernández-Blanco Carramolino, R., Calisto, H., Rodríguez-Gómez, N., Bolado Lavín, R.

2022



Joint Research Centre

EUR 31281 EN

Contact information
Name: Ricardo Bolado Lavin
Address: Westerduinweg 3, 1755 LE Petten, Netherlands
Email: Ricardo.BOLADO-LAVIN@ec.europa.eu
Tel.: (+31) 0224-56-5349

EU Science Hub
https://joint-research-centre.ec.europa.eu

JRC130771

EUR 31281 EN

PDF ISBN 978-92-76-58764-4 ISSN 1831-9424 doi:10.2760/153669 KJ-NA-31-281-EN-N

Luxembourg: Publications Office of the European Union, 2022

How to cite this report: Jung, D., Vuillaume, J-F., Fernández-Blanco Carramolino, R., Calisto, H., Rodríguez-Gómez, N., Bolado Lavín, R., *Analysing the European natural gas system with public data – The Python package eurogastp*, Publications Office of the European Union, Luxembourg, 2022, doi:10.2760/153669, JRC130771.

Contents

Abstract

The European Network of Transmission System Operators for Gas (ENTSOG) publishes key transport indicators on the ENTSOG Transparency Platform. Gas Infrastructure Europe (GIE) maintains similar transparency initiatives, namely AGSI+ and ALSI, where operators provide data on the use of liquefied natural gas (LNG) terminals and underground gas storages. These platforms are a proven and useful source of information for researchers, practitioners and policy advisors, which is publicly available without registration or cost. In both cases, an application programming interface is provided, allowing for automated access to the data by user applications. In this report, we briefly describe the usefulness of these transparency initiatives, show how these data can conveniently be downloaded, processed and analysed, and provide an introduction into an extension package for the programming language Python that has been developed to perform most of the steps involved.

Acknowledgements

The authors thank very much Erica Guerini (JRC.I.3) from the JRC Data Support, Anna Wojcik and Adeline-Raluca Toader (both JRC.I.4) from the JRC's Central Intellectual Property Rights (IPR) team, Gijs Hillenius (DIGIT.B.3.002, GitLab administrator) and Alessio Spadaro (JRC.I.3) from the JRC Github Support for their tremendous help in publishing both data and software resulting from this project.

Authors

Jung, Daniel (JRC.C.3)

Vuillaume, Jean-Francois (JRC.C.3)

Fernández-Blanco Carramolino, Ricardo (JRC.C.3)

Calisto, Hugo (JRC.C.3)

Rodríguez-Gómez, Nuria (JRC.C.3)

Bolado Lavín, Ricardo (JRC.C.3)

1 Introduction

The European Network of Transmission System Operators for Gas (ENTSOG) was founded on 1 December 2009. Its principal mission, as stated in Reg. (EU) No 715/2009[1], is to enhance cooperation between its members, among which are companies from European Union (EU) as well as non-EU countries[2], and develop a comprehensive gas transmission system spanning the European continent, led by the intention of EU policy makers to create a single internal gas market[3].

Through its winter and summer supply outlooks and reviews, ENTSOG also directly helps monitoring the Union's security of gas supply.

As part of ENTSOG's primary goals, gas transmission operators are obliged to publish key transport indicators, in particular for cross-border interconnecting pipelines as well as certain other network points, on the ENTSOG Transparency Platform (ENTSOG TP).

Similar transparency platforms (TPs) are run by Gas Infrastructure Europe (GIE) which is an association whose members are operators of gas infrastructure across Europe (EU Member States, but also Norway, Switzerland, Ukraine and the United Kingdom) including pipelines, underground gas storage (UGS) facilities and liquefied natural gas (LNG) regasification terminals.

Arguably the main advantages of these databases when compared to other data sources lie in the up-to-dateness and time resolution of the data. While other data sources often offer only a monthly or even yearly resolution, the time series provided on the ENTSOG and GIE platforms possess either a daily or, for some indicators on the ENTSOG TP, even an hourly resolution. Furthermore, the data is reported rapidly, in some cases with a delay as little as 3 hours, although in other cases it can take 24 hours or more for a first estimate of the data to appear on the platform. Even then, with a time delay as little as this, it is possible to follow developments in the system almost in "real time", and the daily or hourly resolution makes it possible to investigate and understand the system's reaction to many kinds of events, incidents and disruptions.

Providing such a comprehensive database of transport indicators (physical flow, firm capacity, among others), these platforms are essentially an archive of the European transmission system. Its data can be used to analyse the evolution of the EU natural gas system and draw some important conclusions about the EU's state of security of gas supply, among other uses.

To provide the best support to policy-making Directorates General of the European Commission (EC), the Commission's Joint Research Centre (JRC) has developed methods to download, process and analyse data provided from the ENTSOG and GIE TPs in an automated manner, and an extension package for the popular programming language Python was implemented. This toolset, which is available as open source software to the generic public[4], is briefly described by this report, while also providing a general overview of the transparency platforms themselves.

Besides the programming code itself, an important part of the developed toolset is a mapping from the heterogeneous raw data provided by ENTSOG to a "target topology" that can be more conveniently be used for further processing. This mapping is also be used to decide if and how each time series of the raw data is being used, thereby homogenizing and cleaning the data to a certain extent. The topology mapping is fully customizable by the user, and will need to be updated regularly to reflect any future changes taking place in the European gas network.

This report is structured as follows: After this introduction, Section 2 takes a closer look at some public data sources available for studying the European natural gas system. Section 3 explains in detail the steps necessary to use in particular data from the ENTSOG TP, given its higher degree of complexity compared to GIE AGSI+ and ALSI TP. Section 4 will then introduce the developed Python package implementing the steps necessary to download and post-process the data from the ENTSOG and GIE TPs. Section 5 draws some conclusions.

[1] See article 5 of the Regulation (https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32009R0715&from=EN).
[2] ENTSOG's members currently comprise 45 TSO members, 2 associated partners and 9 observers.
[3] https://www.europarl.europa.eu/factsheets/en/sheet/45/internal-energy-market
[4] https://code.europa.eu/jrc-energy-security/eurogastp, https://github.com/ec-jrc/eurogastp

2 Public data sources for the European gas transmission system

Table 1 lists some useful public data sources to analyse the European gas transmission system. We focus mainly on the transparency platforms of ENTSOG and GIE, only mentioning other data sources for comparison or validation purposes. Although similar data is also available on websites of the operators, the transparency platforms are clearly preferable as they provide a one-stop-shop solution in most cases that contain most needed data under one roof. Nevertheless, operators' data portals can certainly be of use for crosschecking. Only for EU-external borders, a list of data sources is therefore provided in Table 35 in Annex 1 where direct web links per cross-border interconnector pipeline are provided.

Table 1. Public data sources for the European gas transmission system.

Acronyms	Full name	URL
ENTSOG	European Network of Transmission System Operators for Gas	https://www.entsog.eu/
GIE ALSI	Gas Infrastructure operators of Europe: Aggregated LNG Storage Inventory	https://alsi.gie.eu/
GIE AGSI+	Gas Infrastructure operators of Europe: Aggregated Gas Storage Inventory	https://agsi.gie.eu/
Eurostat	Key European statistics, including production and consumption of natural gas	https://ec.europa.eu/eurostat

Source: JRC, 2022.

What follows is a brief discussion of the main indicators of interest for gas flow studies and what data is available on the mentioned transparency platforms of GIE and ENTSOG to that respect.

2.1 Transport indicators

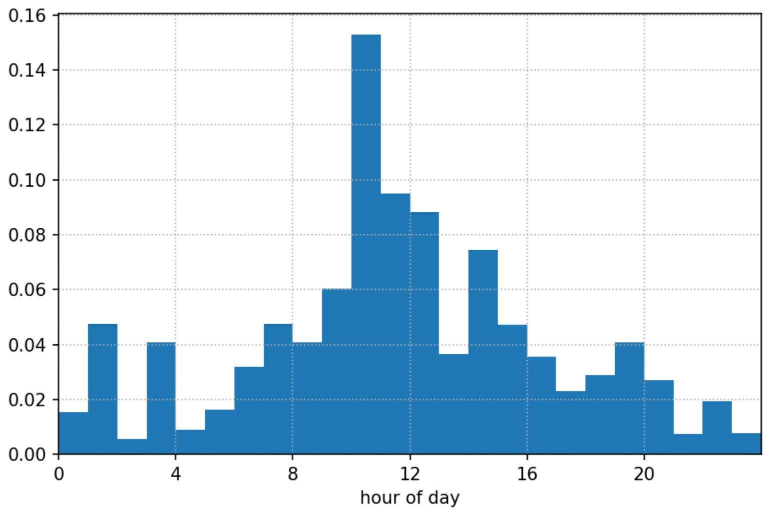
The most important source for transport data is arguably the ENTSOG TP[5]. It is kept up-to-date by the European transmission system operators themselves on a daily basis, on mandate of Regulation (EU) No 715/2009. The platform was launched on 1 October 2014, which is also the date of the earliest coherent data records available on the platform.

Naturally, there is a certain delay with which Transmission System Operators (TSOs) share their data on the TP. While the data of some TSOs are updated quite fast (only a few hours of delay), others can take up to 24 hours, exceptionally also more. Many of the TSOs seem to have a specific time of day for dealing with uploading updated information to the platform, thus providing updates exactly once a day. Others seem to provide updates on a more continuous basis, in some cases with as little of three-hour delay. Further delay is caused by ENTSOG processing the data which is necessary before it can be presented on ENTSOG's transparency web portal and be available for download through the application programming interface (API).

The distribution of the field *lastUpdateDateTime* (based on data between October 2014 and January 2022, covering only interconnector and production points) shows that most data is updated between 10:00 and 11:00 (Central-European Time) (see Figure 1). This is however not identical with the time and date operators are uploading this data, nor when it is becoming available on the platform. It just reflects the time the data was processed internally as indicated by the TSOs[6].

[5] https://transparency.entsog.eu/
[6] Information received bilaterally from ENTSOG.

Figure 1. Distribution of data update times on the ENTSOG TP.



Source: JRC based on ENTSOG TP, 2022.

In most countries, a so-called "gas day" is beginning and ending at 6 am (in some countries it begins and ends at 7 am or 8 am). Daily values for the previous day can therefore not be provided before 6 am. Considering a certain delay, in the best case values can only become available at 9 am or later. As mentioned above, for some TSOs this delay can be much higher. For these reasons, it is usually not possible to consistently analyse a time span that is stretching beyond the end of the gas day of download minus two days ("D-2") when aggregating over a large number of network points, as otherwise data for some interconnectors could be missing in the aggregation.

In any case, processing should be done in such a way that missing data is not silently ignored in an aggregation process. On the other hand, there is the problem of ambiguity in the case of non-existing values: Can those values be considered zero, or are they just missing due to a fault? Often it is necessary to use common sense, and also external information, such as planned maintenance periods and other operational status information of the various interconnectors to decide how missing values must be interpreted.

The ENTSOG TP offers multiple APIs. One of them is dedicated to operational data, which we focus on here. The operational data API contains a multitude of transport indicators: *Physical flow*, *gross-calorific value* (GCV), *Wobbe index*, *firm capacity*, *firm booked capacity*, *interruptible capacity*, *interruptible booked capacity*, *nomination*, *renomination*, *allocation*, *interruptions*, etc. We focus here on some of the most interesting indicators from our point of view for studies related to security of supply:

— Physical flow, i.e. flows actually taken place, as measured by TSOs on one or both sides of a network point (unit: kWh/d);

— Firm capacity, i.e. the part of a network point's capacity that is offered as "firm" at a network point (unit: kWh/d);

— GCV, for being able to express results in volumetric units instead of energy units, if so desired (unit: kWh/Nm^3).

The TP offers datasets with hourly as well as daily granularity. For most studies daily resolution should be enough, which is also keeping file sizes manageable. Furthermore, the TP offers two options regarding the "periodization" of the downloaded data: "compact" or "periodized". In periodized form, the downloaded file will contain exactly one entry per day (or per hour), even if the value is not changing between days. While this is convenient for filtering and aggregating the data, it increases file sizes considerably. We therefore opt for the compact format in which one entry can also span a time frame longer than a day (sometimes, in the case of firm capacity, even years). This way, file sizes are kept as small as possible. On the other hand, this makes it necessary to periodize the dataset after downloading, before we can process it further, as detailed in Section 3.

The raw data of ENTSOG comes in form of large tables. Each row contains one indicator value (e.g., physical flow) for a given operator, point, direction and time period, plus some metadata. In our view, the most interesting metadata are the unit of measure (commonly flow rates are measured in kWh/d), the last modification date that specifies when data was processed by the TSO (in case the TSO makes a correction later on), and the column in which the TSO can leave some remarks about the data (what's the legal status about a point, how is the counting done, etc.). Table 2 provides an overview over the columns deemed most relevant (ENTSOG, 2021).

Table 2. Selected ENTSOG Transparency Platform data fields.

Field	Meaning	Remarks
indicator	Name of the indicator	E.g. "Physical flow"
periodFrom	Start of the period for which the information is valid	Often the start of the gas day
periodTo	End of the period for which the information is valid	Often the end of the gas day (start of the next gas day)
operatorKey	Unique identifier (ID) of the TSO	
operatorLabel	TSO name	
pointKey	Unique ID of the point	
pointLabel	Name of the point	
directionKey	Flow direction (Entry or Exit)	
unit	Units in which the indicator value is presented	For flow rates, usually kWh/d
itemRemarks	Item specific optional remark	
generalRemarks	Generic optional remark over a group of data items published by the TSO at the same time	
value	Value of the indicator	
lastUpdateDateTime	The date of the upload of the information on the TP, or in some cases, the date at which the last change in value was observed	

Source: JRC based on ENTSOG TP, 2022.

We consider that the fields *operatorKey*, *pointKey*, *directionKey* form a unique identifier to the specific network element of interest. Adding the dimension of time, *periodFrom* and *periodTo* have to be considered as well to select a specific data record. This will become important for filtering and aggregating data (see Section 3).

2.2 UGS & LNG indicators

Concerning LNG terminals and UGS facilities, there are two transparency platforms operated by GIE: ALSI[7] and AGSI+[8]. Similar to the ENTSOG TP, operators of LNG and storage facilities are uploading key indicators such as *LNG inventory*, *LNG send-out flow*, *LNG send-out capacity*, among others.

AGSI+ stands for Aggregated Gas Storage Inventory. The platform provides data about UGSs of EU countries and of the United Kingdom, Ukraine, and Serbia. The members of GIE publish UGS data on a voluntary basis. The data is provided per Member State, per storage operator and per single storage site. The earliest data available is from 2011, however several important updates have been performed throughout the time. On 9 August 2016, the website was relaunched and since then the key indicators are provided in energy units rather than volumetric units. The conversion to energy units is done by the corresponding storage system operators (SSO), thus no conversion is performed by GIE. The only data modified by GIE is the Ukrainian data prior to 2017, which is converted to energy units by using a fixed conversion factor of 11.4 kWh/Nm^3, as indicated in the data definition section of AGSI+ website[9]. As acknowledged by GIE, this initiative helps to

[7] https://alsi.gie.eu/
[8] https://agsi.gie.eu/
[9] https://agsi.gie.eu/data-definition

provide information that market participants need to efficiently and effectively operate the system whilst protecting commercially sensitive information.

There are 19 EU Member States (MS) providing information on the platform on a daily basis, namely Austria (AT), Belgium (BE), Bulgaria (BG), Croatia (HR), Czech Republic (CZ), Denmark (DK), France (FR), Germany (DE), Hungary (HU), Ireland (IE), Italy (IT), Latvia (LV), Netherlands (NL), Poland (PL), Portugal (PT), Romania (RO), Slovakia (SK), Spain (ES), and Sweden (SE). Furthermore, two non-EU countries send UGS data to GIE, namely the United Kingdom (UK) and Ukraine (UA). Serbia (RS) is also included in the list but there is currently no data available on the platform. Several remarks should be made:

— IE UGS Kinsale Southwest ceased operation in April 2017, however no data are shown earlier than that date either.

— All UK datasets were migrated to the non-EU countries datasets on 17 February 2022. Therefore, two time series can be found for UK UGSs, one reporting prior to BREXIT spanning from 1 January 2011 until 1 January 2021, which remains under the EU aggregated dataset, and another reporting post-BREXIT under a dedicated non-EU category from 1 January 2021 onwards.

— The Ukrainian dataset was available since 29 April 2014. Any data prior to 2017 was currently based on historical data in volumetric units, which is converted using a fixed conversion factor of 11.4 kWh/Nm^3. One should be careful when looking at this dataset because the long-term gas storage was not accounted for as gas in storage since 17 September 2020. This long-term gas storage amounted to 4662 mcm at 20 degrees Celsius, as reported by the Ukrainian TSO (approximately 49.52 TWh). However, at the beginning of 2022, the Ukrainian dataset was removed from the TP and the reporting was resumed by Ukrtransgaz (Ukrainian TSO) from 16 April 2022 in an aggregated way.

As with the ENTSOG TP, the data can be accessed through an API. The API provides quick, continuous and direct access to the database and allows for filtering, creating queries and exporting the data as required. As with the ENTSOG TP, all data on the GIE TPs is subject to changes, as the operators frequently provide corrections to early estimates or misprocessed values. These changes can be closely monitored in the news section of AGSI+ webpage[10].

The data provided on the AGSI+ TP for each individual UGS facility, group of facilities, company and country are summarized in Table 3. Regarding the publication date and timing, as pointed out by AGSI+ TP, the key indicators show the situation at the end of the previous gas day. Data is updated every day at 19:30 CET and a second time at 23:00. Since some SSOs do not provide data in the first round, it is always better to look at the datasets after midnight.

Similar to AGSI+, GIE also runs a transparency platform for LNG data, called ALSI[11]. Its structure is quite similar to AGSI+ and it provides information about four key indicators, namely the LNG inventory, send-out flow, and the corresponding storage and regasification capacities Its data fields are summarized in Table 4. It should be noted that the LNG inventory is typically given in volumetric units at 0 degrees Celsius and 1013 mbar while the LNG send-out flow is reported in energy units.

There are 10 EU MSs providing information about the LNG regasification terminals: BE, HR, FR, Greece (GR), IT, Lithuania (LT), NL, PL, PT, and ES. UK LNG terminals are split into two datasets, namely UK pre- and post-BREXIT datasets. However, LNG terminals stopped providing information at some point and currently there are no up-to-date data for UK LNG terminals.

The publication date and timing in ALSI TP follows the same criteria as the outlined for the storage-related data, i.e. the data represents LNG information at the end of the previous gas day and there are two rounds LNG system operators (LSOs) reports data to the platform (a first time at 19:30 CET and a second time at 23:00).

[10] https://agsi.gie.eu/news
[11] https://alsi.gie.eu/

Table 3. Data fields of the AGSI+ transparency platform.

Field	Description	Unit of measure / remarks
gasInStorage	Total amount of gas in storage	TWh
full	Ratio of gas in storage and working gas volume	%
consumption	Annual consumption	TWh
consumptionFull	Filling level compared to annual consumption	%
trend	Daily increase or decrease of gas in storage	% / Difference between injection and withdrawal, divided by working gas volume
injection	Gas injected during gas day	GWh/d
withdrawal	Gas withdrawn during gas day	GWh/d
workingGasVolume	Maximum amount of gas that can be stored	TWh
injectionCapacity	Declared total maximum technical injection	GWh/d
withdrawalCapacity	Declared total maximum technical withdrawal	GWh/d
status	Data quality status	Can be either "estimated", "confirmed" or "no data"

Source: JRC based on AGSI+ TP, 2022.

Table 4. Data fields of the ALSI transparency platform.

Field	Meaning	Unit of measure / Remarks
lngInventory	Aggregated amount of LNG in LNG tanks at the end of gas day	10^3 m^3 of LNG (volume of the liquid)
sendOut	Gas flow out of the LNG facility (send-out during gas day)	GWh/d
dtmi	Declared total maximum inventory (LNG storage capacity)	10^3 m^3 of LNG (volume of the liquid)
dtrs	Declared total reference send-out (send-out capacity)	GWh/d
status	Data quality status	Can be either "estimated", "confirmed" or "no data"

Source: JRC based on ALSI TP, 2022.

2.3 Production indicators

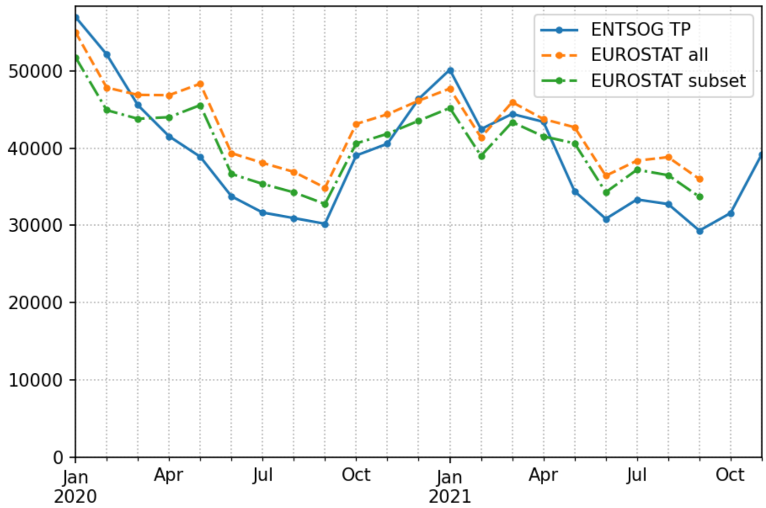
On the ENTSOG TP, TSOs also report daily flows received from production facilities, usually aggregated country-wise. Such a "production point" is offered for ten countries: BG, DE, Estonia (EE), ES, HR, HU, IT, NL, PL, and RO. It is important to note that the ENTSOG TP reflects the TSOs' point of view, so they likely only include production facilities directly connected to the transmission grid. For DK, there is no production node defined, but we can safely assume that flows crossing the inter-zonal interconnection point Nybro reflects Denmark's production, as that is the point where gas produced in the Danish offshore wells is entering the Danish mainland. All other countries might not possess large production facilities connected to the transmission grid, which is probably why they do not occur on the platform.

There is a second source of production data that can be helpful, although it provides data with monthly resolution only, and with a certain delay: Eurostat[12]. In Figure 2 we compare the total gas production in the EU27 as reported on the ENTSOG TP and by Eurostat. As the ENTSOG TP data only covers 11 countries (BG,

[12] https://ec.europa.eu/eurostat

DE, DK, EE, ES, HR, HU, IT, NL, PL, RO), we show the Eurostat value aggregated over the whole EU as well as aggregating only those countries that are included in the ENTSOG TP dataset, with the hope to make it more comparable. But even when considering only the subset of countries reported on the ENTSOG TP, it does not fully agree with Eurostat, even though the deviation is moderate for the most part.

Figure 2. EU27 natural gas production: Comparison of ENTSOG TP and Eurostat (GWh/month).



Source: JRC based on ENTSOG TP and Eurostat, 2022.

As production data is not (yet) available for all EU countries, its usefulness is somewhat limited, as many research questions would require a comprehensive dataset. Here we can only appeal to the remaining EU MSs to fill this gap, which would make the ENTSOG TP an even greater asset. The data already provided can however be already useful for country-level analyses.

2.4 Consumption indicators

For some time now the ENTSOG TP is also reporting on some consumption data. So far, the data covers only 16 EU MSs[13]. Some even distinguish consumption in certain sectors, such as consumption of natural gas in industry and for power production. Other just provide the total consumption. The indicators (partially) provided are *Final Consumption* (FNC) and *Distribution* (DIS). The use of either of the indicators to report on consumption varies between the countries. It is concluded that both FNC and DIS must be considered and aggregated to obtain a country's total consumption (or consumption per sector).

For validation purposes, the consumption data from the ENTSOG TP can for example be compared to the European Natural Gas Demand Database[14] (ENaGaD) (Zaccarelli et al., 2021; 2022). It also comes in daily resolution, in some cases with data disaggregated by sector. In the currently available version 1.1, the timeframe included is 2015-2021. It is based partly on the ENTSOG TP and partially on operators' data portals and covers all EU MSs and some other European countries. The data was carefully selected and cleaned of outliers using standard statistical methods (Z score filtering). The downside is that it is only updated with a considerable delay (about once a year) and cannot deliver up-to-date information with a delay as small as 1-2 days. For this, the transparency platforms of ENTSOG and the data platforms of the operators themselves have to be considered, which is not as convenient as working with a single database. The ENaGaD can however conveniently be used for studying daily consumption of recent years, modelling simulations[15], as well as applying consumption forecasting techniques.

[13] BE, BG, DE, EE, ES, FR, EL, HR, HU, IT, LU, NL, PL, PT, RO, SI.

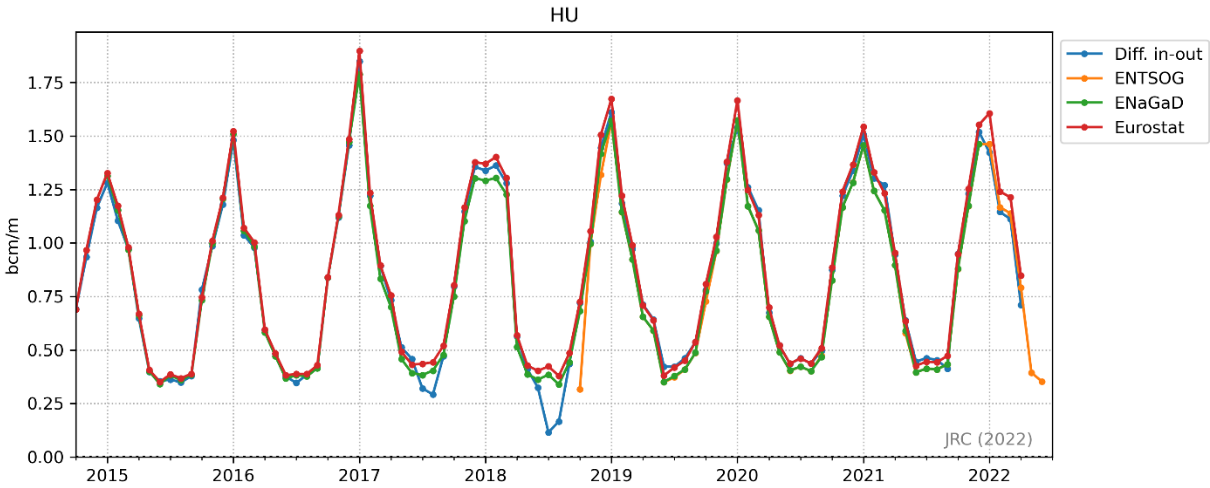
[14] https://doi.org/10.5281/zenodo.6364875

[15] Fernandez Blanco Carramolino, R., Giaccaria, S., Costescu, A. and Bolado Lavin, R., Impact of storage obligations on the EU gas market, EUR 30994 EN, Publications Office of the European Union, Luxembourg, 2022, ISBN 978-92-76-47665-8, doi:10.2760/088735, JRC128147.

When resampling ENTSOG and ENaGaD data to a monthly resolution, also a comparison to Eurostat data is possible. Figure 3 shows the respective data for Hungary. It also includes a proxy for demand which is obtained by taking the difference of the total inflow and the total outflow to/from the Hungarian transmission system (considering cross-border flows and storage injection and withdrawal).

For Hungary, the data from all the different sources are quite well aligned, with only minor deviations. It is shown representative for data of other countries as "good example". However, for some countries the alignment is worse. We generally observe that Eurostat consumption data is usually a bit higher than ENaGaD and ENTSOG data. For some countries, ENaGaD data is identical to ENTSOG data, simply because ENaGaD authors decided to take it from there, while for others, they differ, as ENaGaD authors took consumption data from other sources (e.g. TSO websites).

Figure 3. Consumption of Hungary: Comparison between data from ENTSOG, ENaGaD, Eurostat and the difference between total inflow and total outflow.



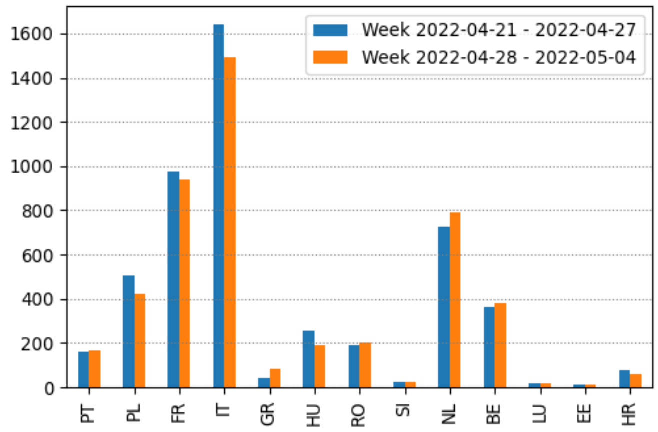
Source: JRC, 2022.

The proxy based on the difference of inflow and outflow can only be meaningful if all other data, i.e. cross-border transmission, production, LNG send-out, storage withdrawal and storage injection are accurate for a given country. For some countries this is the case, for others not, especially for production data which is not always (or only incompletely) reported, as we saw in Section 2.3.

When comparing the data to other sources, such as the monthly-aggregated data of Eurostat, one must bear in mind that the ENTSOG database is populated with data by transmission operators, so the consumption data is likely to reflect a "TSO's point of view", perhaps only including large consumers connected directly to the transmission grid, as well as what is delivered to distribution system operators (DSOs). Gas produced and directly consumed at distribution grid level might not be accounted for. This could explain part of the differences with Eurostat.

As limited as the information so far provided on the ENTSOG TP is (in terms of countries covered), the clear advantage over other sources (e.g. Eurostat) is the daily resolution and the up-to-dateness of the data, which is, like all the other transport indicators, provided with a delay of only 1-2 days, so trends concerning the consumption in certain countries can be followed almost in real-time. The downside is that not all countries (i.e. the respective TSOs) are (so far) providing this service. Figure 4 shows examples for weekly aggregated total consumption of those countries already available.

Figure 4. Average weekly consumption in certain EU countries (GWh/d).



Source: JRC based on ENTSOG TP, 2022.

2.5 Gas quality indicators

Natural gas is far from being pure methane (CH_4). It consists of a multitude of substances, influencing its *calorific value* (or *heating value*), i.e. the amount of heat released when the gas is combusted. As this is supposedly the most interesting property of natural gas when used as an energy carrier, the calorific value is commonly simply referred to as its *quality*. The *gross calorific value* (GCV; also *higher heating value*) is the upper limit of thermal energy that could be released in case of a complete combustion of a certain amount of gas.

TSOs report GCVs on the ENTSOG TP for most network points along with the other transport indicators. They are reported in terms of kilowatt-hours per normal cubic meter (kWh/Nm³). This offers the opportunity to convert flows and capacities from the standard energetic units (e.g. Gigawatt-hours, GWh) to volumetric units (e.g. million cubic meters, mcm), using the simple relation

$$\text{Indicator [mcm/d]} = \text{Indicator [GWh/d]} / \text{GCV [kWh/m}^3\text{]} \quad .$$

The GCV is usually reported only if there is a finite flow, otherwise it is either zero or Not-a-Number (NaN). At a few network points, it is reported even in the absence of flow. In a few cases, it is not reported at all, also not when gas is flowing. As for all data on the TP, some GCV data can include gaps and outliers. In such cases, an estimate must be used, based on other nearby network points or external information, or simply taken from an earlier period of time at the same point.

Table 34 in the annex lists the average GCV in January 2022 at EU-external-border interconnectors (except those with Switzerland), grouped by major pipeline import routes. It is also indicated if the route was actually carrying a finite flow during that period, as we do not expect a GCV value to be reported without any flow. The table shows the general geographic distribution of GCVs, depending on the source of gas. Natural gas from North African countries usually has a GCV of around 11.5 kWh/Nm³, but gas received from Tunisia reaches almost 11.8 kWh/Nm³. Gas reaching the EU mainland from the UK currently has a GCV of around 11.53 kWh/Nm³, while for gas sent to IE is as high as 11.64 kWh/Nm³. Norwegian gas (route “North Sea”) usually has a quite high GCV of over 11.6 kWh/Nm³ when it reaches the EU mainland at the German or Dutch coast, but in pipelines going to France and Belgium, the value is somewhat lower, as gas quality differs per gas field/extraction site. Gas originating in Russia can have a GCV as low as 11.2 kWh/Nm³ (Nord Stream, Yamal, also towards Baltic States and Finland), but gas transiting Ukraine can have a GCV of almost 11.4 kWh/Nm³. Gas received by Bulgaria or Greece from Turkey currently has a GCV generally ranging between 11.24 and 11.34 kWh/Nm³. The relatively new Caspian route, bringing gas from Azerbaijan, transiting Turkey, has a high GCV of over 11.7 kWh/Nm³.

It is sometimes desirable to display results in volumetric (e.g. million cubic meter, mcm/d) instead of energetic units (e.g. GWh/d). Sometimes, GCV data might be missing for the selected network point and time frame. To be able to reliably convert to energy units, we must therefore decide on an alternative conversion factor. The first idea would be to go further into the past and consider the all-time average GCV at that network point, but for some network points there was never any GCV data reported, and furthermore we have seen that quality can change over time, so historic GCVs should be considered with caution.

Another idea is to consider another nearby network point which should normally carry gas of similar quality. This approach is pursued here.

The suggested steps to be taken for computing the mean GCV per network point over a certain amount of time (e.g. the last month or the last year) are:

1. Replace any unphysical values, for example values outside the range [8, 13] kWh/Nm^3, by NaN (not to be considered for computing the mean);
2. Further remove outliers using standard methods (values with absolute Z-score of 3 or larger are removed);
3. For network points not having any GCV data for the chosen period, specify a nearby alternative network point to take the data from instead.

For example., one could keep average GCVs of a certain period (e.g. the last 12 months) and the name of an alternative network point in a spreadsheet file that could be used to convert flows and capacities from energetic to volumetric units whenever up-to-date GCV data are not available.

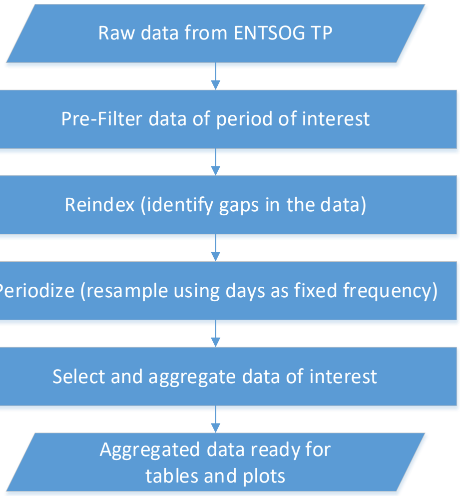
3 Processing data from the ENTSOG Transparency Platform

TSOs who are members of ENTSOG publish certain data on the ENTSOG TP and keep it up to date[16]. This section explains all the steps to download and process the data that we have identified as necessary to make best use of it.

We found that some operators upload data with a very short delay (about three hours), while others with a delay of 24 hours or more. Therefore, in many cases comparative analysis can take place only for periods ending approximately two days before the date when the data is downloaded. Also, all data can be subject to subsequent changes. At times, uploaded data constitute only a preliminary estimate of a TSO, which is replaced by a "confirmed" value about two weeks later, but often corrections are made weeks or even months later. This means that when analysing the same period with two datasets downloaded at different times, the results could differ. This must be kept in mind when working with operational data published on the ENTSOG TP and similar to other platforms of this kind such as AGSI+ TP and ALSI TP operated by GIE.

Using the provided API, we composed a set of Python scripts that download all data of interconnection points (IPs) for the period of interest in the form of Excel spreadsheet files. The data is then periodized, filtered, and aggregated in the manner required for the analysis (see Figure 5). The data represent nodes and edges of a complex network (the European natural gas system). In the following, we summarise the most important steps taken to download, process and visualise data from the ENTSOG TP. We also describe how raw data is appropriately selected, cleaned and combined and thereby translated to a *target topology*. This mapping is defined in the *topology file*.

Figure 5. Work flow for downloading and analysing data from the ENTSOG Transparency Platform.



Source: JRC, 2022.

3.1 Download

The ENTSOG TP is offering an API, enabling automated access to its data. For using the API, it is not necessary to have a user account, nor to provide an API key. Accessing the API is facilitated by calling a certain URL that contains all information regarding what data is requested. The call initiates the file download, provided that the request is valid and no error occurs.

The server then returns an HTML status code. A positive status code would be 200 "OK", which means the data was downloaded successfully. Also, receiving a status code 404 "Not found" is usually nothing to worry about. It just means that there is simply no data available for the interconnector for the chosen time interval. This can easily happen when requesting data for a year in which the interconnector did not yet exist, or the

[16] It is important to note that ENTSOG is only the entity that is operating the TP and is in principle not responsible for the quality and timeliness of the data, which are the sole responsibility of the users, i.e., the TSOs.

TSO not providing data on other grounds. Any other status code however points to a problem with the connection to the server or with the server itself.

There are some caveats when downloading a large quantity of data from the ENTOSG TP. First, there is a time limit per HTML request of 60 seconds. Thus, it might be impossible to download all required data in one go. We generally do a separate download per edge in our target topology, which comes down to one download per interconnector and per direction. If that is not enough, one also has to do a separate download per year of data. Sometimes it might be possible to download all interconnectors of a certain pair of countries, to limit the number of requests.

On the other hand, there is also a limit regarding the number of requests per user within a certain time interval. So, if submitting too many HTML requests within a certain time frame, there is the risk of receiving the HTML status code 429 "Too Many Requests". Also, when the server is overloaded (e.g. too many users issuing requests at the same time), the request might fail with some status code such as 502 "Bad gateway" or similar, or simply take a long time to complete. These requests must be repeated later on. The most busy time seems to be (not surprisingly) weekdays morning between 7:00 am and 11:00 am, so we recommend to avoid those periods if possible to guarantee a quick and successful download.

Once the download is complete, it is wise to load all downloaded files and pre-filter them, removing columns that are not necessary for later analysis (e.g. unwanted metadata), so to save some valuable disk space. Also, all files are then merged into a single table and saved to a single HDF5 file (*.h5), so that the data resides in a compact format and can be loaded and processed much faster and easier in a later stage. HDF5 is the file format of our choice, as it allows faster access than comma-separated value (CSV) files while leading to smaller file sizes. Spreadsheet files (i.e. Excel format) are limited to around one million rows per table, which can be too small if the dataset must cover many network points, different indicators and span multiple years.

The result of the download phase is a single HDF5 file containing a single table, containing what we call the *raw data*.

3.2 Reindex

As mentioned above, the raw data ("compact format") from the ENTSOG TP is indexed time-wise by a pair of date columns, standing for the beginning and the end of the value's *period of validity*. This is much better for performance and saving memory, as often values (especially firm capacities) do not change much over time and would otherwise result in repeating the same value each day. But for visualising and further processing the data, this format is not particularly practical. This is why, in a first step, we re-index the dataset, thereby turning the two time-related columns into only one column. Essentially, we are choosing to mark only the beginning[17] of each period of validity, and define that the value is valid until the beginning of the next period. That implies that we must add periods for which no data exists, indicated by NaN ("not a number"), as otherwise the previous entry would be valid for too long. Also, one final NaN entry must be added at the end of the time series, simply to mark the end of the last period.

In the following, we indicate the value of a data record i as v_i (the unit is typically kWh/h) and its period of validity as ranging from t_i^{from} to t_i^{to} (both values correspond to a specific date and time). Note that the data records in the raw data are ordered by time period, and these do not overlap.

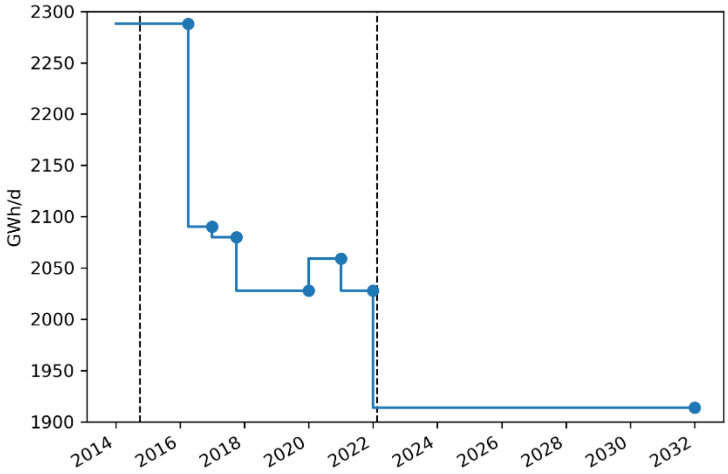
The first step is to identify gaps in the raw dataset, so periods for which in fact no value exists. In other words, we look for cases where the period end time t_i^{to} corresponding to a data record i with value v_i is not identical to the period beginning of the next data record, t_{i+1}^{from}. In such cases, we insert a new data record with $v_{i+1} = NaN$, shifting all upcoming data records such that $i + 1$ becomes $i + 2$ etc. In the end, we have a consecutive data series with $t_{i+1}^{from} = t_i^{to}$ for all data records $i = 1, \dots, N$. Between the first and last record, there are no time intervals missing anymore, just that for some time periods, the value might be *NaN*.

Now we have no longer use for t_i^{to}, as all periods can be assumed to last until t_{i+1}^{from}, the beginning of the period of validity of the subsequent record. But we must add one additional value in the end of the series, $v_{N+1} = NaN$, to mark the length of the final period of the time series. So we set $t_{N+1}^{from} = t_N^{to}$.

[17] This choice is deliberate. We could also choose the end of each period. Both approaches are exchangeable.

The result is a dataset that only has a single field (t_i^{from}) for indexing the data time-wise, marking only the beginning of validity of each value. It can already conveniently be plotted now using a step plot, similar to what the web interface of the transparency platform is doing. An example is shown in Figure 6. Note that the data downloaded can stretch considerably beyond the requested period (vertical dashed lines), as each data value comes with a specific period of validity. In the example shown in Figure 6, the period of validity of the last data record stretches all the way until 2032.

Figure 6. Step plot of the firm technical capacity at Velke Kapusany (UA-SK) (GWh/d). Period end times are indicated by a dot. The vertical dashed lines indicate beginning and end of the period of interest.



Source: JRC, 2022.

3.3 Periodize

We download data from the ENTSOG TP in "compact format" instead of "periodized format". That keeps download and file sizes small and could already be plotted in a step plot, but is unpractical for later processing, in particular when having to aggregate over several network points which will almost certainly be necessary during the analysis. We therefore have to perform the periodization ourselves, after downloading. In essence, this means we have to resample the data with (gas) days as (fixed) frequency, so that

$$t_i^{from} - t_{i-1}^{from} = 1d \quad ,$$

where t_i^{from} is the beginning of the period of validity of the value with index i. We also have to forward-fill the new rows with the last value existing in the original dataset. If the last value was NaN, consequently the NaN value will be copied for all days within that period.

The result is a time series for each edge of the target topology that contains exactly one value per gas day, even if there is no value available for that particular day. It can also be plotted as a step plot, but now that the time axis has a fixed frequency of one day, it also makes sense to use a simple line plot. Note that this time series is now bounded by the exact start and end dates of the specified period of interest, in this example 1 October 2014 until 15 February 2022, as illustrated by the vertical dashed lines in Figure 6. This makes the dataset more convenient for further processing and plotting.

3.4 Topology mapping

While the GIE TPs are relatively simple in structure and size and straightforward to download and analyse, processing data from the ENTOSG TP is slightly more complex. The first reason is that the number of "facilities", i.e. cross-border points and other network points (which include also LNG and UGS facilities), and also the number of published indicators are much larger than those of the GIE TPs. The second reason is that we are dealing with networked data.

Understanding the principal concepts of graph theory definitely helps to process the data. A graph consists of a set of nodes and a set of edges connecting those nodes. As real-world interconnectors can be either unidirectional or bidirectional and can have different transmission capacity in either direction, the target topology is given by a *directed graph*. Nodes are usually corresponding to *balancing zones* (and sometimes to LNG or UGS facilities, production or consumption nodes etc.), connected by interconnectors to each other. Each interconnector is split into one or more *directions* and each of those directions is further split into multiple *entries* and *exits* belonging to different *operators*.

It proves difficult to understand the topology just from the downloaded data itself. We found it is necessary to refer to the web interface of the TP, checking which balancing zones a point connects, and also connecting the data to geographic information. Even then some points containing multiple exits or entries per direction are hard to interpret correctly. Often, their correspondence could only be checked by comparing their respective time series for physical flow: if they are identical or at least very similar, one could assume that those entries and exits are in fact connected to each other, or that the entry flow is split across several exits etc.

In some cases, multiple entries or multiple exits are specified as “pipe-in-pipe” (P-in-P), meaning that each operator reports the aggregated physical flow. Otherwise, each operator reports only their own flow, so they need to be summed to obtain the overall flow at that point, as it is generally the case for other indicators such as firm capacity.

For many purposes it is better to have one node per country (not counting points representing sources and sinks such as production, consumption, LNG and UGS), as most other data such as production and consumption also comes on a per-country basis, and because for many studies tackling security of supply questions this leads to a good compromise between network complexity, numerical effort and accuracy of the results. Thus, it is necessary to map the ENTOSG TP topology to the desired target topology. This mapping is done using what we call the *topology file*.

The *topology file* has two purposes:

— Provide a mapping from the TP topology to the target topology;

— Define what data to take into account from the TP for a particular edge of the target topology.

The topology file is a simple (Excel) spreadsheet file, containing multiple tables. Different spreadsheets in the file correspond to the different types of interconnectors: Interconnectors (ITP), production (PRD), LNG, UGSs, distribution (DIS), final consumption (FNC), virtual trading points (VTP).

Each table is following the same template. In the first set of columns, the complete ENTSOG TP topology is represented by the fields *pointKey*, *operatorKey*, *directionKey*. The second set of columns defines the target topology with the fields *edge_name*, *from_node*, *to_node*. Node names are usually the country codes following ISO 3166. Deviations from this naming scheme occur in case we want to represent a country by two nodes. This can be worthwhile for countries with a distinct transit pipeline system (e.g. Poland, Romania, Greece), but is only possible if the database describes also the interconnection between national and transit system. Also North Ireland (NI) can be represented by its own node, as interconnection points with the Republic of Ireland and Great Britain are reported on the platform.

Larger countries could be split into multiple nodes as well (e.g. Germany) as long as enough intra-zone points are published. For some studies, this would however require the knowledge of disaggregated production and consumption per node, and to which of the nodes LNG and UGS facilities are connected. A split into multiple nodes is thus usually impossible, except for the cases mentioned above where a dedicated transit system exists (which does not connect directly to any consumption or facilities).

Table 5 shows an excerpt of the topology file, showing only the first 40 rows and the most relevant columns. The first five columns (white) identify the network point in the ENTSOG TP raw data. The next three columns (orange) define edges and nodes of the target topology, provided that the respective raw data is to be included in the new topology (otherwise it is left empty). The remaining six columns shown in Table 5 (green, orange and yellow) decide if and how the raw data is used to form the edge in the new topology. For each indicator individually, it is possible to use a single time series from the raw dataset (value 1), to exclude it (value 0), or to combine multiple raw time series by using a certain operation: take the minimum per day (min), take the maximum per day (max), take the average (av).

Table 5. Excerpt from the topology file.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	pointKey	pointLabel	operatorKey	operatorLabel	directionKey	edge_name	from_node	to_node	firm	firmbooked	gcv	flow	nom	renom
2	ITP-00001	PIR MIDI	FR-TSO-0002	TIGF	entry									
3	ITP-00001	PIR MIDI	FR-TSO-0002	TIGF	exit									
4	ITP-00005	Bacton (IUK)	UK-TSO-0003	Interconnector	exit	ZeebruggeBEUK	BE	UK	0	0	0	0		
5	ITP-00005	Bacton (IUK)	UK-TSO-0001	National Grid Gas	exit	ZeebruggeUKBE	UK	BE	0	0	0	0		
6	ITP-00005	Bacton (IUK)	UK-TSO-0003	Interconnector	entry	ZeebruggeUKBE	UK	BE	0	0	0	0		
7	ITP-00006	Oberkappel (OGE)	DE-TSO-0009	Open Grid Europe	entry	OberkappelATDE	AT	DE	sum	sum	max	0		
8	ITP-00006	Oberkappel (OGE)	DE-TSO-0009	Open Grid Europe	exit	OberkappelDEAT	DE	AT	sum	sum	max	av		
9	ITP-00007	Überackern SUDAL (AT) / Überackern 2 (DE	AT-TSO-0001	Gas Connect Austria	exit	Ueberackern2ATDE	AT	DE	min	min	max	av		
10	ITP-00007	Überackern SUDAL (AT) / Überackern 2 (DE	DE-TSO-0010	bayernets	entry	Ueberackern2ATDE	AT	DE	min	min	max	av		
11	ITP-00007	Überackern SUDAL (AT) / Überackern 2 (DE	AT-TSO-0001	Gas Connect Austria	entry	Ueberackern2DEAT	DE	AT	min	min	max	av		
12	ITP-00007	Überackern SUDAL (AT) / Überackern 2 (DE	DE-TSO-0010	bayernets	exit	Ueberackern2DEAT	DE	AT	min	min	max	av		
13	ITP-00008	Melendugno - IT / TAP	AL-TSO-0001	TAP	entry	MelendugnoITTAP	IT	TAP	min	min	max	av		
14	ITP-00008	Melendugno - IT / TAP	IT-TSO-0001	Snam Rete Gas	exit	MelendugnoITTAP	IT	TAP	min	min	max	av		
15	ITP-00008	Melendugno - IT / TAP	AL-TSO-0001	TAP	exit	MelendugnoTAPIT	TAP	IT	min	min	max	av		
16	ITP-00008	Melendugno - IT / TAP	IT-TSO-0001	Snam Rete Gas	entry	MelendugnoTAPIT	TAP	IT	min	min	max	av		
17	ITP-00010	Brandov-OPAL (DE)	CZ-TSO-0001	NET4GAS	exit	BrandovOpalCZDE	CZ	DE	1	1	max	1		
18	ITP-00010	Brandov-OPAL (DE)	CZ-TSO-0001	NET4GAS	entry	BrandovOpalDECZ	DE	CZ	0	0	0	0		
19	ITP-00010	Brandov-OPAL (DE)	DE-TSO-0016	OPAL Gastransport	exit	BrandovOpalDECZ	DE	CZ	0	0	0	0		
20	ITP-00011	Dravaszerdahely	HR-TSO-0001	Plinacro Ltd	exit	DravasHRHU	HR	HU	min	min	max	av		
21	ITP-00011	Dravaszerdahely	HU-TSO-0001	FGSZ	entry	DravasHRHU	HR	HU	min	min	max	av		
22	ITP-00011	Dravaszerdahely	HR-TSO-0001	Plinacro Ltd	entry	DravasHUHR	HU	HR	min	min	max	av		
23	ITP-00011	Dravaszerdahely	HU-TSO-0001	FGSZ	exit	DravasHUHR	HU	HR	min	min	max	av		
24	ITP-00013	Lasów	DE-TSO-0003	ONTRAS	exit	LasowDEPL	DE	PL	min	min	max	av		
25	ITP-00013	Lasów	PL-TSO-0002	GAZ-SYSTEM	entry	LasowDEPL	DE	PL	min	min	max	av		
26	ITP-00013	Lasów	DE-TSO-0003	ONTRAS	entry	LasowPLDE	PL	DE	min	min	max	av		
27	ITP-00013	Lasów	PL-TSO-0002	GAZ-SYSTEM	exit	LasowPLDE	PL	DE	min	min	max	av		
28	ITP-00015	Hora Svaté Kateřiny (CZ) / Deutschneudorf (S	CZ-TSO-0001	NET4GAS	exit	HoraSaydaCZDE	CZ	DE	min	min	max	av		
29	ITP-00015	Hora Svaté Kateřiny (CZ) / Deutschneudorf (S	DE-TSO-0003	ONTRAS	entry	HoraSaydaCZDE	CZ	DE	min	min	max	av		
30	ITP-00015	Hora Svaté Kateřiny (CZ) / Deutschneudorf (S	CZ-TSO-0001	NET4GAS	entry	HoraSaydaDECZ	DE	CZ	min	min	max	av		
31	ITP-00015	Hora Svaté Kateřiny (CZ) / Deutschneudorf (S	DE-TSO-0003	ONTRAS	exit	HoraSaydaDECZ	DE	CZ	min	min	max	av		
32	ITP-00018	Larrau	ES-TSO-0006	Enagas	exit	LarrauESFR	ES	FR	min	min	max	av		
33	ITP-00018	Larrau	FR-TSO-0002	TERÉGA	entry	LarrauESFR	ES	FR	min	min	max	av		
34	ITP-00018	Larrau	ES-TSO-0006	Enagas	entry	LarrauFRES	FR	ES	min	min	max	av		
35	ITP-00018	Larrau	FR-TSO-0002	TERÉGA	exit	LarrauFRES	FR	ES	min	min	max	av		
36	ITP-00019	Überackern ABG (AT) / Überackern (DE)	AT-TSO-0001	Gas Connect Austria	exit	UeberackernATDE	AT	DE	1	1	max	1		
37	ITP-00019	Überackern ABG (AT) / Überackern (DE)	DE-TSO-0009	Open Grid Europe	entry	UeberackernATDE	AT	DE	0	0	max	0		
38	ITP-00019	Überackern ABG (AT) / Überackern (DE)	DE-TSO-0010	bayernets	entry	UeberackernATDE	AT	DE	0	0	max	0		
39	ITP-00019	Überackern ABG (AT) / Überackern (DE)	AT-TSO-0001	Gas Connect Austria	entry	UeberackernDEAT	DE	AT	1	1	max	0		
40	ITP-00019	Überackern ABG (AT) / Überackern (DE)	DE-TSO-0009	Open Grid Europe	exit	UeberackernDEAT	DE	AT	0	0	max	av		
41	ITP-00019	Überackern ABG (AT) / Überackern (DE)	DE-TSO-0010	bayernets	exit	UeberackernDEAT	DE	AT	0	0	max	av		

Source: JRC, 2022.

3.5 Filter

The third set of columns in the *topology file* defines what exact data series (or what combination of data series) to take into account for a particular edge of the target topology. A decision must be made based on the availability and quality of the data provided in the respective exit and entry points at each interconnector (or bunched interconnectors) in the ENTSOG TP. This is the step where also any needed topological interpretation must take place. Especially points with more than two directions as well as conglomerations of multiple points that exist in a small area are sometimes hard to interpret just from the data and the web interface of the TP alone and require either interpretation of reported physical flows and – where that is not enough – some form of external knowledge.

At the moment, we include the following columns, representing certain indicators: *firm* (firm technical capacity), *firmbooked* (firm booked capacity), *gcv* (GCV), *flow* (physical flow). Each of these columns can contain specific values. Each of these values represents a specific aggregation strategy to form the time series for this particular edge of the target topology:

- 0 (do not consider this data);
- 1 (consider only this data);
- *sum* (for each gas day, take the sum of the specified data series);
- *min* (for each gas day, take the minimum of the specified data series);
- *max* (for each gas day, take the maximum of the specified data series);
- *av* (for each gas day, take the arithmetic mean of the specified data series).

If two sets of data, for example exit and entry at two sides of an interconnector, apparently represent the same flow, we often average them. In case of GCV, we often use the maximum reported, as it is sometimes zero on one side of the interconnector while the other side seems to contain the correct value. For capacities the lesser-rule should be applied, taking the minimum of the offered capacities at either side of the interconnector.

Even with this set of strategies it is not always straightforward to make the good decision on how to process the existing data. In some cases, a particular indicator is reported under one *pointKey* up to a certain date, and after that, the TSO continues reporting the same data under another *pointKey*. This is often the case whenever *virtual interconnection points* (VIPs) are created, or if changes to the topology occur. Also, outliers and data gaps still remain. Outliers can be treated using standard techniques, such as removing all points in the time series that have a *Z-score* of three or larger, but there is arguably no one-fits-all approach to dealing with outliers.

After selecting and aggregating the points of interest, we obtain a new dataset that is no longer indexed by the TP's nomenclature (*pointKey*, *operatorKey*, *directionKey*, …), but instead by a certain *edge name* that we gave it, and the corresponding *from node* and *to node* that exist in our desired target topology.

3.6 Aggregate

Having periodized (resampled) the data and mapped it to the target topology, the data can now easily be aggregated. It is now fairly simple to aggregate all flows and capacities that exist between a certain pair of countries, or between one set of countries (e.g. the whole EU) and another set of countries (e.g. all non-EU countries or a specific set of third-party countries). As the developed Python package utilises *Pandas dataframes*, Pandas[18] functionality can be used to conveniently aggregate the so prepared ENTSOG TP data.

3.7 Visualise

There are a number of ways to visualise the filtered and aggregated data. In principle, any tool can be used for visualisation: Line charts, bar charts, stacked area charts, pie charts etc. We have included a range of

[18] https://pandas.pydata.org/

plotting functions into the Python module for the user's convenience which use the well-known Python plotting package Matplotlib[19].

The following section explains the basic usage of the developed Python package `eurogastp` which is implementing most of the above steps.

[19] https://matplotlib.org/

4 Basic usage of the Python package eurogastp

This section introduces the Python package `eurogastp` and the API of its main functions and attributes. It also contains short application examples.

4.1 General structure

The package is currently divided into two submodules: `eurogastp.entsog` and `eurogastp.gie`, dealing with downloading and processing of data of the ENTSOG Transparency Platform (TP) and the GIE AGSI+ and ALSI TPs, respectively. Processing of ENTSOG TP data generally needs more steps, which is why it gains more weight in the description that follows.

4.2 Submodule entsog

The submodule `eurogastp.entsog` deals with interfacing with the ENTSOG Transparency Platform API and processing the data retrieved.

Next to API functions, several lists and dictionaries (Python's associative arrays) are defined at module level that are either needed directly by certain functions or help the user with using the API.

4.2.1 The function download_entsog_tp()

This function `entsog.download_entsog_tp()` downloads data from the ENTSOG TP. Its call signature is

```
error = download_entsog_tp(start_date, end_date, topo, edges=None,
                           dir_name=None,
                           indicators=['Physical Flow', 'Firm Technical',
                                       'Firm Booked', 'GCV', 'Nomination',
                                       'Renomination'],
                           proxy=None, delay=0, overwrite=False,
                           max_points_per_request=3, show_api_call=False)
```

The function downloads raw data from the ENTSOG TP, making use of its API. Execution of the function leads to the creation of a folder named ENTSOG_TP_data_YYYY-MM-DD in the current working directory, where YYYY-MM-DD is the current date (if it does not yet exist). In that folder, a separate folder for every year of data is created. In each subfolder, a separate file is created for each HTML request that contains the downloaded raw data.

Usually, one file for each edge of the target topology is created. However, if the number of network points included in an edge is larger than the parameter `max_points_per_request`, multiple HTML requests will be done for this edge. In that case, the name of the files will end with ascending numbers.

Table 6 summarises the different function arguments.

Table 6. Arguments of the function download_entsog_tp().

Function argument	Description
`start_date, end_date`	datetime.date objects specifying the time frame for the downloaded data
`topo`	DataFrame holding the topology data, as retrieved by the `load_topo()` function
`edges`	List of edges to download, as specified in the topology file; if None, downloading all edges defined in the topology file
`dir_name`	Name of the folder to store the downloaded files. By default, using the name "ENTSOG_TP_data_YYYY-MM-DD", where YYYY-MM-DD is the current date
`indicators`	list of strings, specifying the indicators to download (see the `ind_col_map` dictionary for a list of allowed indicator names/short forms)
`proxy`	if specified, set proxy to the given address; could include username, password and port number (example: "http://user:password@proxy-address:1234")
`delay`	Time in seconds to wait between API calls
`overwrite`	if True, overwrite existing files; otherwise, skip download for already existing files (default)
`max_points_per_request`	maximum number of network points put into one HTML request; if an edge has more network points defined, data will be split into multiple raw files. Default: 3
`show_api_call`	if true, dump each API call into the console; for debugging purposes

Source: JRC, 2022.

The function returns an error status of 1 if at least one of the downloads failed (HTML request resulted in a HTML status code other than 200 or 404); otherwise, 0 is returned.

4.2.2 The function raw_to_file()

This function `entsog.raw_to_file()` loads raw files and combines them into a single HDF5 file. Its call signature is

```
raw_to_file(dir_name=None, dir_name2='ENTSOG_TP_data_previous_years',
            out_name=None)
```

The function loads raw data from the directories `dir_name` and `dir_name2` and saves the data to a single file (HDF5 format) for easier and faster access later on. By default, the resulting file will have the name `[dir_name].h5`.

The HDF5 file format is deemed necessary as Excel files can only contain ~1e6 rows, and csv files would be unnecessarily large and slow to access. On the other hand, csv files would be far better for exchanging data across platforms and different Pandas versions, as the data is put into HDF5 files in pickled form and might thus only be used with the same or a similar Pandas version.

Table 7 summarises the different function arguments.

Table 7. Arguments of the function raw_to_file().

Function argument	Description
dir_name	Path to the directory that contains the raw data downloaded from the ENTSOG Transparency Platform. If not specified, it will look for a directory called ENTSOG_TP_data_YYYY-MM-DD, where YYYY-MM-DD is the current date.
dir_name2	Secondary directory with data that should be merged with the data from the primary one; it is thought to be holding data from previous years that do not to be updated as often as data from the current year.
out_name	Name of the HDF5 to save the data to. If None, it will be set to [dir_name].h5.

Source: JRC, 2022.

4.2.3 The function load_raw_file()

The function `entsog.load_raw_file()` loads ENTSOG TP raw data from a file. Its call signature is

```
raw = load_raw_file(filename)
```

The file can be either an Excel® spreadsheet file (*.xls, *.xlsx), a file containing comma-separated values (*.csv), or a HDF5 file (*.h5). The file type is recognised by the corresponding filename ending.

The only parameter is `filename` which holds the path to the file containing the raw data.

The function returns a Pandas DataFrame containing the raw data.

4.2.4 The function load_topo()

The function `entsog.load_topo()` reads in the topology file. Its call signature is

```
topo = load_topo(topo_file='topo/ENTSOG_TP_Network_v2.xlsx',
                 sheets=['ITP', 'PRD', 'LNG', 'UGS', 'DIS', 'FNC', 'VTP'])
```

The topology file contains essentially a mapping from the data provided on the ENTSOG Transparency Platform onto a "target topology". This mapping is mainly needed as input for the function select_and_aggregate() (see Section 4.2.6). It is defined in form of a spreadsheet file (Excel, or in principle any other spreadsheet format supported by the Pandas read_excel() function). `eurogastp` comes with a default topology file, located in the subdirectory “topo”, but the user is free to choose a different topology file (or a modified version of the existing one).

Table 8 summarises the different function arguments.

Table 8. Arguments of the function load_topo().

Function argument	Description
topo_file	Path to the topology file. Default: topo/ENTSOG_TP_Network_v2.xlsx
sheets	Sheets to load from the spreadsheet file, corresponding to the network point type. By default, loading all types of network points (ITP, PRD, LNG, UGS, DIS, FNC, VTP).

Source: JRC, 2022.

The function returns the topology mapping from the file, with all sheets merged into a single Pandas DataFrame.

4.2.5 The function reindex_and_periodize()

The function `entsog.reindex_and_periodize()` is a convenience function that calls both `entsog.reindex_by_period_endtime()` and `entsog.periodize()` in succession on a given raw dataset downloaded from the ENTSOG TP. Its call signature is

```
perd = reindex_and_periodize(raw, start_date, end_date)
```

Table 9 summarises the different function arguments.

Table 9. Arguments of the function reindex_and_periodize().

Function argument	Description
`raw`	`pandas.DataFrame` containing the raw data from the ENTSOG TP
`start_date, end_date`	`datetime.date` objects indicating period of interest

Source: JRC, 2022.

The function returns a `pandas.DataFrame` containing the re-indexed and periodized data.

4.2.6 The function select_and_aggregate()

The function `entsog.select_and_aggregate()` is used to filter and aggregate a periodized dataset. Its call signature is

```
ind = select_and_aggregate(edges, topo, df, indicator, quiet=False)
```

The data inside a periodized dataset is filtered by selecting one (string) or more edges (list of strings). The possible edge names the target topology contains can previously be identified with the functions `filter_data()` or `filter_nodes()` or be looked up in the topology file.

Table 10 summarises the different function arguments.

Table 10. Arguments of the function select_and_aggregate().

Function argument	Description
`edges`	List of edge names to aggregate (either str or list of str)
`topo`	Topology mapping
`df`	Input data from ENTSOG Transparency Platform in re-indexed and periodized format
`indicator`	Indicator of interest (e.g. Firm Booked, Firm Technical, Physical Flow, …); see the dictionary attribute ind_col_map for possible values (both keys and values of this dictionary can be used to refer to a specific indicator)

Source: JRC, 2022.

The user has to choose an indicator. If the user is interested in multiple indicators (e.g. both physical flow and firm capacity), two separate calls to `select_and_aggregate()` have to be performed. To identify a particular indicator, both the long name (e.g. “Physical Flow”) and the short-hand (e.g. “flow”) can be used. See Section 4.2.11 for possible indicator names and short-hands. Note however that only a fraction of the indicators are so far covered by the topology file shipped with `eurogastp`. Further indicators can be accessed by the user by customising the topology file.

The function does not aggregate over the specified edges. It only aggregates those network points making up one edge, hence the name.

The function returns a Pandas DataFrame with the data of the different edges of the selected indicator in columns (one column per specified edge). The unit is generally Gigawatt-hours per day (GWh/d).

4.2.7 The function filter_data()

The function `entsog.filter_data()` is a convenience function for filtering several pieces of data that are of interest when working with `eurogastp.entsog`. Its call signature is

```
df2 = filter_data(df, indicator=None, operatorKey=None, pointKey=None,
                  directionKey=None, edge_name=None, from_node=None,
                  to_node=None)
```

The DataFrame `df` is filtered by the values one or more field names (columns). One or more values can be specified per field (`str` or list of `str`). A new DataFrame `df2` is returned which is the filtered version of `df`.

Table 11 summarises the different function arguments.

Table 11. Arguments of the function filter_data().

Function argument	Description
`df`	A DataFrame as retrieved from certain functions (load_raw(), reindex_by_period_endtime(), periodize(), reindex_and_periodize(), load_topo())
all other function arguments	Column headers typically found in abovementioned DataFrames; can be passed a single string with an existing column name or a list of column names. Values can be single values or lists of values.

Source: JRC, 2022.

4.2.8 The function filter_nodes()

The function `entsog.filter_nodes()` provides a convenient way to filter the topology mapping (`topo`) as retrieved from the function `load_topo()` by nodes (`from_node` and `to_node`). The call signature is

```
df2 = filter_nodes(topo, from_node=None, to_node=None)
```

Single nodes as well as lists of nodes can be specified. The function returns a list of edges, connecting `from_node` to `to_node` in the topology. Note that the topology forms a directed graph, so the result would be different when swapping the contents of the arguments `from_node` and `to_node`. Both arguments are optional.

Table 12 summarises the different function arguments.

Table 12. Arguments of the function filter_nodes().

Function argument	Description
`topo`	`A DataFrame as retrieved from the function load_topo()` containing a topology mapping.
`from_node`, `to_node`	Strings or list of strings with node names for filtering the topology mapping.

Source: JRC, 2022.

4.2.9 The function get_corridors()

The function `entsog.get_corridors()` filters the target topology (`topo`) for the major inflow corridors to the EU and returns them as an (ordered) dictionary. The call signature is

```
corridors = get_corridors(topo)
```

The corridors are:

— North Africa (Morocco, Algeria, Tunisia, Libya).

— UK.

— North Sea (Norway).

— East (gas of mainly Russian origin: Russia, Belarus, Ukraine, Türkiye).

— Caspian (Azerbaijan).

Table 13 summarises the different function arguments.

Table 13. Arguments of the functions get_corridors() and get_routes().

Function argument	Description
`topo`	A DataFrame as retrieved from the function `load_topo()` containing a topology mapping.

Source: JRC, 2022.

The function returns a (ordered) dictionary with the names of the corridors as keys and arrays with the edge names that form the corridors as values.

4.2.10 The function get_routes()

The function `entsog.get_routes()` filters the target topology (`topo`) for the different routes along all major inflow corridors to the EU and returns them as an (ordered) dictionary. The call signature is

```
routes = get_routes(topo)
```

Some corridors (see `get_corridors()`) are further split into different routes, leading to this set of routes.

The routes are:

— North Africa -> ES

— North Africa -> IT

— UK

— North Sea

— East -> Nord Stream

— East -> Baltic+Finland

— East -> Yamal

— East -> Ukraine

— East -> Türkiye

— Caspian

Note that some corridors are not further split into routes, hence they are identical to the routes.

Table 13 (above) summarises the different function arguments (they are the same as `get_corridors()`).

The function returns a (ordered) dictionary with the names of the routes as keys and arrays with the edge names that form the routes as values.

4.2.11 The attribute ind_col_map

The dictionary `entsog.ind_col_map` contains a mapping between the ENTSOG TP's indicator names and `eurogastp`'s own shortcuts that allow the user easier interaction with the API. The contents of `ind_col_map` are listed in Table 14.

So far, `eurogastp` uses only a fraction of indicators presented in Table 14: Nomination, renomination, physical flow, gross-calorific value (GCV) and firm technical capacity. The user can in principle customise the topology file (see Section 3.4) to access more indicators, and future version of `eurogastp` might actually include them.

Table 14. Mapping between ENTSOG TP's indicator names and eurogastp shortcuts.

ENTSOG TP indicator name	**eurogastp shortcut**
Nomination	`nom`
Renomination	`renom`
Allocation	`alloc`
Physical Flow	`flow`
GCV	`gcv`
Wobbe Index	`wob`
Firm Technical	`firm`
Firm Booked	`firmbooked`
Firm Available	`firmavail`
Interruptible Total	`int`
Interruptible Booked	`intbooked`
Interruptible Available	`intavail`
Planned interruption of firm capacity	`planinterruptfirm`
Unplanned interruption of firm capacity	`unplaninterruptfirm`
Planned interruption of interruptible capacity	`planinterruptint`
Actual interruption of interruptible capacity	`unplaninterruptint`

Source: JRC, 2022.

4.2.12 The attributes eu_nodes, non_eu_nodes, north_african_nodes and russian_origin_nodes

The lists `eu_nodes`, `non_eu_nodes`, `north_african_nodes` and `russian_origin_nodes` (defined inside the `entsog` submodule) exist merely to make it more convenient for the user to filter data by specific countries and regions of interest. For example, the user might want to aggregate all flow entering or leaving the EU, or the combined flow coming from North Africa or along Eastern routes (gas of primarily Russian origin).

The list `eu_nodes` contains the names of all nodes representing EU countries inside the target topology. The list `non_eu_nodes` contains exactly the opposite: All nodes of the target topology that represent non-EU countries.

The dictionary `north_african_nodes` contains the names of all nodes representing North African countries inside the target topology. Likewise, the `russian_origin_nodes` contains the names of all nodes representing EU-neighbouring countries from which gas of primarily Russian origin is imported into the EU.

The nodes contained in each of the abovementioned lists is presented in Table 15. Note that the node PLYAM represents the Polish part of the Yamal pipeline, TBP the Trans-Balkan Pipeline, TAP the Trans-Adriatic Pipeline, and RUKAL the Russian enclave Kaliningrad. All other nodes resemble ISO 3166 codes.

Table 15. Contents of the entsog attributes eu_nodes, non_eu_nodes, north_african_nodes and russian_origin_nodes.

Attribute	**Contents**
`eu_nodes`	AT, BE, BG, CZ, DE, DK, EE, ES, FR, GR, HR, HU, IE, SK, PT, IT, PL, LU, MT, CY, LV, LT, FI, SI, RO, SE, NL, PLYAM, TBP, TAP
`non_eu_nodes`	RU, BY, UA, TR, MA, DZ, TN, LY, UK, NO, BA, MK, MD, RS, CH, AZ, GE, ME, AL, RUKAL
`north_african_nodes`	MA, DZ, TN, LY
`russian_origin_nodes`	RU, BY, UA, TR

Source: JRC, 2022.

4.3 Submodule gie

The submodule `eurogastp.gie` deals with interfacing with the API of the GIE transparency platforms AGSI+ (European natural gas storage) and ALSI (European LNG terminals).

4.3.1 The function download_gie_alsi()

The function `gie.download_gie_alsi()` downloads data from the GIE ALSI transparency platform (data of European LNG terminals), aggregated per country. Its call signature is

```
df = download_gie_alsi(start_date, end_date, api_key, proxy=None,
                       timeout=60, delay=0,
                       countries=['be', 'hr', 'fr', 'gr', 'it', 'lt', 'nl',
                                  'pl', 'pt', 'es', 'gb*'])
```

The data downloads LNG data per country from the GIE ALSI transparency platform. The GIE ALSI database contains data starting from January 1, 2012.

Table 16 summarises the different function arguments.

Table 16. Arguments of the function download_gie_alsi().

Function argument	Description
`start_date, end_date`	`datetime.date` objects specifying the time frame for the downloaded data
`api_key`	Personal API key; must be requested from GIE
`proxy`	If specified, set proxy to the given address; could include username, password and port number (example: "http://user:password@proxy-address:1234")
`timeout`	Set timeout for the API call, seconds; default: 60 seconds.
`delay`	Time in seconds to wait between API calls
`countries`	list of countries for which LNG data shall be downloaded; by default, downloading all countries (be, hr, fr, gr, it, lt, nl, pl, pt, es, gb*); gr stands for Greece, gb* stands for pre-Brexit Great Britain.

Source: JRC, 2022.

The function returns a DataFrame with the columns described in Table 17.

Table 17. Columns of the pandas.DataFrame returned by the function download_gie_alsi().

Column header	Description
`lngInventory`	Aggregated amount of LNG in the LNG tanks status at end of gas day [1000 m^3]
`sendOut`	Aggregated gas flow out of the LNG facility, send-out during gas day [GWh/d]
`dtmi`	Declared total maximum inventory (lng storage capacity) [1000 m^3]
`dtrs`	Declared total reference send-out (send-out capacity) [GWh/d]

Source: JRC, 2022.

4.3.2 The function download_gie_alsi_per_terminal()

The function `gie.download_gie_alsi_per_terminal()` downloads data from the GIE ALSI transparency platform (data of European LNG terminals) per facility (individual LNG terminals). Its call signature is

```
df = download_gie_alsi_per_terminal(start_date, end_date, api_key,
                                    eics_file='topo/Dataproviders.csv',
                                    proxy=None, timeout=60, delay=0)
```

The function downloads data from the GIE ALSI (LNG) transparency platform, disaggregated per single LNG terminal. The GIE ALSI database contains data starting from January 1, 2012.

Table 19 summarises the different function arguments.

Table 18. Arguments of the function download_gie_alsi_per_terminal().

Function argument	**Description**
`start_date, end_date`	`datetime.date` objects specifying the time frame for the downloaded data
`api_key`	Personal API key; must be requested from GIE
`eics_file`	A CSV file containing the list of EIC codes of each facility; default: `topo/Dataproviders.csv`
`proxy`	If specified, set proxy to the given address; could include username, password and port number (example: `http://user:password@proxy-address:1234`)
`timeout`	Set timeout for the API call, seconds; default: 60 seconds.
`delay`	Time in seconds to wait between API calls

Source: JRC, 2022.

The function requires specification of an "EICS file", containing the list of Energy Identification Codes[20] (EICs) of the different facilities. This file can be retrieved from the GIE ALSI transparency platform's website once API access is requested.

The EICS file is provided in different formats on the GIE ALSI transparency platform. However, the function `gie.download_gie_alsi_per_terminal()` is expecting the EICS file in CSV format. The user is recommended to regularly look for updates of the EICS file on the ALSI platform, especially when new LNG terminals are coming operational.

The returned Pandas DataFrame has the columns described in Table 17 (the same as the function `download_gie_alsi()`).

4.3.3 The function download_gie_agsi()

The function `gie.download_gie_agsi()` downloads data from the GIE AGSI+ transparency platform (data of European natural gas storages), aggregated per country. Its call signature is

```
df = download_gie_agsi(start_date, end_date, api_key,
                       proxy=None, timeout=60, delay=0,
                       countries=['at', 'be', 'bg', 'hr', 'cz', 'dk', 'fr',
                                  'de', 'hu', 'it', 'lv', 'nl', 'pl', 'pt',
                                  'ro', 'se', 'sk', 'es'])
```

The function downloads data from the GIE AGSI+ transparency platform per country. The units are GWh or GWh/d, depending on the column.

Table 19 summarises the different function arguments.

[20] Used in Europe to uniquely identify market participants and energy resources.

Table 19. Arguments of the function download_gie_agsi().

Function argument	**Description**
`start_date, end_date`	`datetime.date` objects specifying the time frame for the downloaded data
`api_key`	Personal API key; must be requested from GIE
`proxy`	If specified, set proxy to the given address; could include username, password and port number (example: `http://user:password@proxy-address:1234`)
`timeout`	Set timeout for the API call, seconds; default: 60 seconds.
`delay`	Time in seconds to wait between API calls
`countries`	List of countries for which UGS data shall be downloaded; by default, downloading all countries (at, be, bg, hr, cz, dk, fr, de, hu, it, lv, nl, pl, pt, ro, se, sk, es)

Source: JRC, 2022.

The function returns a DataFrame with the columns described in Table 20.

Table 20. Columns of the pandas.DataFrame returned by the function download_gie_agsi().

Column header	**Description**
`gasInStorage`	Volume of gas in storage at end of gas day [GWh]
`consumption`	Annual consumption [GWh]
`consumptionFull`	Filling level compared to annual consumption [%]
`injection`	Amount of gas injected during gas day [GWh/d]
`withdrawal`	Amount of gas withdrawn during gas day [GWh/d]
`workingGasVolume`	Technical capacity – maximum amount that can be stored [GWh]
`injectionCapacity`	Declared total maximum technical injection (DTMTI) [GWh/d]
`withdrawalCapacity`	Declared total maximum technical withdrawal (DTMTW) [GWh/d]
`status`	either "confirmed" or "estimated"
`trend`	Daily increase or decrease of gas in storage [%]
`full`	percentage of working gas volume in storage [%]

Source: JRC, 2022.

4.3.4 The function update_gie_alsi_archive()

The function `gie.update_gie_alsi_archive()` updates an existing GIE ALSI archive file (data aggregated per country), or creates a new one if it does not yet exist. Its call signature is

```
update_gie_alsi_archive(start_date, end_date, api_key,
        archive_file='GIE_TPs_Archive/GIE_ALSI_archive_GWh_d.xlsx',
        proxy=None, timeout=60, delay=0)
```

The function does not have a return value.

If the specified archive file does not yet exist, it is created. Any specified subdirectory that does not yet exist is created as well.

When the function is updating an existing archive file, we mean the following: It overwrites any data for dates already existing in the file, and simply adding data for new dates not yet in the file. For example, if we download data for the last 30 days, but data for all but the last 5 days is already in the archive, it is adding those 5 days and updating the data for the 25 days before that date with the new (more up-to-date) data from the transparency platform.

Table 21 summarises the different function arguments.

Table 21. Arguments of the function update_gie_alsi_archive().

Function argument	**Description**
`start_date, end_date`	`datetime.date` objects specifying the time frame for the downloaded data
`api_key`	Personal API key; must be requested from GIE
`archive_file`	Path to where the archive file is kept (or where it shall be created); default: `GIE_TPs_Archive/GIE_ALSI_archive_GWh_d.xlsx`
`proxy`	If specified, set proxy to the given address; could include username, password and port number (example: `http://user:password@proxy-address:1234`)
`timeout`	Set timeout for the API call, seconds; default: 60 seconds.
`delay`	Time in seconds to wait between API calls

Source: JRC, 2022.

The archive file is a simple spreadsheet (Excel) file containing a single table.

4.3.5 The function update_gie_alsi_archive_per_terminal()

The function `gie.update_gie_alsi_archive_per_terminal()` updates an existing GIE ALSI archive file (disaggregated per LNG terminal), or creates a new one if it does not yet exist. Its call signature is

```
update_gie_alsi_archive_per_terminal(start_date, end_date, api_key,
  archive_file='GIE_TPs_Archive/GIE_ALSI_archive_per_terminal_GWh_d.xlsx',
  proxy=None, timeout=60, delay=0, eics_file='topo/Dataproviders.csv')
```

The function does not have a return value.

If the specified archive file does not yet exist, it is created. Any specified subdirectory that does not yet exist is created as well.

When the function is updating an existing archive file, we mean the following: It overwrites any data for dates already existing in the file, and simply adding data for new dates not yet in the file. For example, if we download data for the last 30 days, but data for all but the last 5 days is already in the archive, it is adding those 5 days and updating the data for the 25 days before that date with the new (more up-to-date) data from the transparency platform.

Table 22 summarises the different function arguments.

Table 22. Arguments of the function update_gie_alsi_archive_per_terminal().

Function argument	Description
`start_date, end_date`	`datetime.date` objects specifying the time frame for the downloaded data
`api_key`	Personal API key; must be requested from GIE
`eics_file`	A CSV file containing the list of EIC codes of each facility; default: `topo/Dataproviders.csv`
`archive_file`	Path to where the archive file is kept (or where it shall be created); default: `GIE_TPs_Archive/GIE_ALSI_archive_per_terminal_GWh_d.xlsx`
`proxy`	If specified, set proxy to the given address; could include username, password and port number (example: `http://user:password@proxy-address:1234`)
`timeout`	Set timeout for the API call, seconds; default: 60 seconds.
`delay`	Time in seconds to wait between API calls

Source: JRC, 2022.

The archive file is a simple spreadsheet (Excel) file containing a single table.

4.3.6 The function update_gie_agsi_archive()

The function `gie.update_gie_agsi_archive()` updates an existing GIE AGSI+ archive file (data aggregated per country), or creates a new one if it does not yet exist. Its call signature is

```
update_gie_agsi_archive(start_date, end_date, api_key,
        archive_file='GIE_TPs_Archive/GIE_AGSI_archive_GWh_d.xlsx',
        proxy=None, timeout=60, delay=0)
```

The function does not have a return value.

Table 23. Arguments of the function update_gie_agsi_archive().

Function argument	Description
`start_date, end_date`	`datetime.date` objects specifying the time frame for the downloaded data
`api_key`	Personal API key; must be requested from GIE
`archive_file`	Path to where the archive file is kept (or where it shall be created); default: `GIE_TPs_Archive/GIE_AGSI_archive_GWh_d.xlsx`
`proxy`	If specified, set proxy to the given address; could include username, password and port number (example: `http://user:password@proxy-address:1234`)
`timeout`	Set timeout for the API call, seconds; default: 60 seconds.
`delay`	Time in seconds to wait between API calls

Source: JRC, 2022.

If the specified archive file does not yet exist, it is created. Any specified subdirectory that does not yet exist is created as well.

When the function is updating an existing archive file, we mean the following: It overwrites any data for dates already existing in the file, and simply adding data for new dates not yet in the file. For example, if we download data for the last 30 days, but data for all but the last 5 days is already in the archive, it is adding those 5 days and updating the data for the 25 days before that date with the new (more up-to-date) data from the transparency platform.

Table 23 summarises the different function arguments.

The archive file is a simple spreadsheet (Excel) file containing a single table.

4.3.7 The function load_lng()

The function `gie.load_lng()` loads LNG data per country from the LNG archive file. Its call signature is

```
df_lng = load_lng(lng_file='GIE_TPs_Archive/GIE_ALSI_archive_GWh_d.xlsx')
```

The function returns a Pandas data frame with the LNG data per country (the whole contents of the archive file).

Table 24 summarizes the function arguments.

Table 24. Arguments of the function load_lng().

Function argument	Description
`lng_file`	Path to where the archive file is kept; default: `GIE_TPs_Archive/GIE_ALSI_archive_GWh_d.xlsx`

Source: JRC, 2022.

4.3.8 The function load_lng_per_terminal()

The function `gie.load_lng_per_terminal()` loads LNG data per terminal from the LNG per terminal archive file. Its call signature is

```
df_lng_term = load_lng_per_terminal(lng_term_file='GIE_TPs_Archive/
                                    GIE_ALSI_archive_per_terminal_GWh_d.xlsx')
```

The function returns a Pandas data frame with the LNG data per terminal (the whole contents of the archive file).

Table 25 summarizes the function arguments.

Table 25. Arguments of the function load_lng_per_terminal().

Function argument	Description
`lng_term_file`	Path to where the archive file is kept; default: `GIE_TPs_Archive/GIE_ALSI_archive _per_terminal_GWh_d.xlsx`

Source: JRC, 2022.

4.3.9 The function load_ugs()

The function `gie.load_ugs()` loads UGS data per country from the UGS archive file. Its call signature is

```
df_ugs = load_ugs(ugs_file='GIE_TPs_Archive/GIE_AGSI_archive_GWh_d.xlsx')
```

The function returns a Pandas data frame with the UGS data per country (the whole contents of the archive file).

Table 26 summarizes the function arguments.

Table 26. Arguments of the function load_ugs().

Function argument	Description
`ugs_file`	Path to where the archive file is kept; default: `GIE_TPs_Archive/GIE_AGSI_archive_GWh_d.xlsx`

Source: JRC, 2022.

4.4 Other functions

Most of times, the user will only need the main API functions listed in Sections 4.2 and 4.3. In the following we discuss some additional functions that the user might want to use, depending on the use case.

Furthermore, there are functions in the Python package that start with an underscore ("_"). Those are helping functions that are not officially part of the API and hence not discussed within this guide.

4.4.1 The function load_raw()

The function `entsog.load_raw()` loads ENTSOG TP raw data from the folder `dir_name`, potentially merging it with the raw data from the folder `dir_name2`. The call signature is

```
raw = load_raw(dir_name=None, year=None,
               dir_name2='ENTSOG_TP_data_previous_years')
```

If `year` is specified, the function loads data only from a certain year or from a certain set of years, using common globbing rules on the subdirectories of `dir_name` and `dir_name2` (`glob.glob`).

Table 27 summarises the different function arguments.

Table 27. Arguments of the function load_raw().

Function argument	Description
`dir_name`	`Path to the directory that contains the raw data downloaded from the ENTSOG Transparency Platform. If not specfied, it will look for a directory called ENTSOG_TP_data_YYYY-MM-DD, where YYYY-MM-DD is the current date.`
`year`	Specify to load a specific year / specific years (the subfolders containing the respective data). Multiple years can be specified using common globbing rules (see documentation of `glob.glob` for more information[21]). Default: Load all years (all subdirectories).
`dir_name2`	Secondary directory with data that should be merged with the data from the primary one; it is thought to be holding data from previous years that do not to be updated as often as data from the current year

Source: JRC, 2022.

The function returns a Pandas DataFrame containing raw ENTSOG TP data, converted to GWh/d (in case of GCV: GWh/MNm^3).

This function is primarily being used by `raw_to_file()` (see Section 4.2.2), but can also be used by the user for using and analysing raw data without converting it to HDF5 format first. The raw data can conveniently be filtered using `filter_data()` (see Section 4.2.7).

[21] https://docs.python.org/3/library/glob.html

4.4.2 The function reindex_by_period_endtime()

The function `entsog.reindex_by_period_endtime()` pre-filters data downloaded from the ENTSOG TP around a certain period of interest, and re-indexes the data by period end time. The call signature is

```
rexd = reindex_by_period_endtime(raw, start_date, end_date)
```

The function pre-filters data downloaded from the ENTSOG TP around the desired time period and re-indexes it by period end times. This essentially removes the column `periodFrom` and makes `periodTo` the index of the DataFrame. It also inserts additional rows with NaN values for periods with missing data (important for showing gaps when plotting). It will also add a row with a NaN value at the top to mark the beginning of the first period.

Table 28 summarises the different function arguments.

Table 28. Arguments of the function reindex_by_period_endtime().

Function argument	Description
`raw`	`ENTSOG TP raw data, such as returned by load_raw() or load_raw_file().`
`year`	Specify to load a specific year / specific years (the subfolders containing the respective data). Multiple years can be specified using common globbing rules (see documentation of `glob.glob` for more information). Default: Load all years (all subdirectories).
`dir_name2`	Optional secondary directory with data that should be merged with the data from the primary one; it is thought to be holding data from previous years that do not to be updated as often as data from the current year.

Source: JRC, 2022.

The function returns a Pandas DataFrame `rexd` containing the pre-filtered and re-indexed data.

This function is automatically called by `reindex_and_periodize()` (see Section 4.2.5). Normally there is little use for the user to execute this function alone, without passing the data subsequently to periodize() (see Section 4.4.3).

For performance reasons in certain applications, it could be of interest to plot the re-indexed data directly, without periodizing it first, using a step plot[22]. To do this, the results can for example be plotted using the Pandas expression

```
rexd.value.plot(drawstyle='steps')
```

For further processing the data, in particular when aggregating over several data or applying mathematical expressions, it is generally much easier to pass the re-indexed data to `periodize()` first. It also allows for easier plotting methods, for example plotting simple line charts (other than step plots), bar charts etc.

4.4.3 The function periodize()

The function `entsog.periodize()` periodizes a previously re-indexed dataset. Its call signature is

```
perd = periodize(rexd, start_date, end_date)
```

The function periodizes a previously re-indexed dataset. The result will be a Pandas DataFrame that contains exactly one value per day. Unlike `rexd`, the first entry does not necessarily contain a NaN value anymore.

Table 29 summarises the different function arguments.

[22] These step plots are similar to what the ENTSOG Transparency Platform seems to be using inside its web interface.

Table 29. Arguments of the function periodize().

Function argument	Description
`rexd`	`pandas.DataFrame containing the re-indexed data from the ENTSOG TP, generated using the function reindex_by_period_endtime().`
`start_date, end_date`	`datetime.date` objects indicating period of interest.

Source: JRC, 2022.

The function returns a Pandas DataFrame containing the periodized data.

The results can already be plotted the usual way using line plots or other types of plots, i.e. using the Pandas expression

```
perd.value.plot()
```

The returned data can also more easily be post-processed using mathematical expressions or aggregated over data of different network points. As in most cases the user is likely to work with periodized data, there is the convenience function `reindex_and_periodize()` (see Section 4.2.5) which calls both `reindex_by_period_endtime()` and `periodize()` in succession.

In most cases, the user might want to pass the periodized data to the function `select_and_aggregate()` (see Section 4.2.6) to select data by a particular edge of the target topology and by a particular indicator of interest.

4.4.4 The function remove_outliers()

The function `entsog.remove_outliers()` removes outliers using standard statistical methods (Z-score screening). Its call signature is

```
df2 = remove_outliers(df, threshold=3, inplace=False)
                      remove_outliers(df, threshold=3, inplace=True)
```

The function removes outliers of a given time series by replacing values with a Z-score of `threshold` or larger by NaN.

Table 30 summarises the different function arguments.

Table 30. Arguments of the function remove_outliers().

Function argument	Description
`df`	`pandas.DataFrame containing the data`
`threshold`	Threshold for Z-score values; data with a Z-score larger than or equal to `threshold` are being replaced by `NaN` (not-a-number).
`ddof`	Delta degrees of freedom. This value is passed to the function `pandas.DataFrame.std()` (standard deviation). Default value is 0 (data is normalised by N instead of N-1).
`inplace`	if `True`, replace outliers by `NaN` in-place, modifying the original dataset `df`; if False, leave the original dataset the way it was and return a copy of it (`df2`) with the respective data replaced by `NaN`.

Source: JRC, 2022.

If an expression evaluating to True is passed to `inplace`, the function does not return anything. Otherwise, it returns a copy of the original DataFrame `df` with the identified outliers replaced by NaN.

4.4.5 The function get_display_names()

The function `entsog.get_display_names()` returns the edge display names, as defined in the topology `topo` (field `edge_display_name`), of the given list of edges. Its call signature is

```
names = get_display_names(edge_names, topo)
```

Edges for which no display name has been defined in the topology file are silently ignored, passing through the original edge name.

Table 31 summarises the different function arguments.

Table 31. Arguments of the function get_display_names().

Function argument	Description
`edge_names`	iterable containing the edge names
`topo`	DataFrame holding the topology data, as retrieved by the `load_topo()` function

Source: JRC, 2022.

4.4.6 The function display()

The function `entsog.display()` returns a version of the given data frame `df` with the edge names in the column headers replaced by the corresponding edge display name, as defined in the topology `topo` (field `edge_display_name`). Its call signature is

```
df2 = display(df, topo)
```

Column headers for which no display name is defined in the topology are silently passed through.

Table 32 summarises the different function arguments.

Table 32. Arguments of the function display().

Function argument	Description
`df`	DataFrame holding the data, with columns representing different edges and edge names as column headers, as for example received from the function `select_and_aggregate()`
`topo`	DataFrame holding the topology data, as retrieved by the `load_topo()` function

Source: JRC, 2022.

4.5 Application example

In the following, we show how the Python package `eurogastp` could be used to download and analyse data from the ENTSOG and GIE ALSI transparency platforms. In this example, we want to compare the evolution of physical flow of natural gas into the EU along the five major inflow corridors mentioned in Section 4.2.9 as well as the EU LNG send-out between January 2022 and now (July 2022). The full code is also included as a Jupyter notebook with the `eurogastp` package.

4.5.1 Step 1: Download data and save to file

First, we have to do some initial settings. We import the required Python modules, set our GIE ALSI API key and, if necessary, set a proxy. We then find out what date is today and define the period of interest. ENTSOG TP data is usually quite reliable until the day before yesterday, so we choose the current day minus two days (D-2) as the end of our period of interest.

```
import os
import datetime as dt
import pandas as pd
import eurogastp.entsog as etp
import eurogastp.gie as gtp
import matplotlib.pyplot as plt
import itertools as it

api_key_alsi = 'XXX'  # fill your API key received from GIE
proxy = None

today = dt.date.today()
download_start = dt.date(today.year, 1, 1)
download_end = today - dt.timedelta(2)
download_start, download_end
```

Then, we load the topology file.

```
topo = etp.load_topo('../topo/ENTSOG_TP_Network_v2.xlsx')
```

We are going to download the ENTSOG data using the function `entsog.download_entsog_tp()`. As we are interested only in data of network points at EU borders, we can filter the topology file for the respective interconnectors (edges of the target topology). In this special case, we can use the function `entsog.get_corridors()` to retrieve the list of edges that each corridor is consisting of.

```
corridors = etp.get_corridors(topo)
edges_eu_border = list(it.chain(*corridors.values()))
```

Now we can start the download. We use a delay of one second between API calls to prevent overloading the ENTSOG server and specify to download only data of the edges at EU borders. If necessary, the user can set a proxy.

```
etp.download_entsog_tp(download_start, download_end, topo, delay=1,
                       edges=edges_eu_border, proxy=proxy)
```

After execution of the function above, a new subdirectory should have been created under the current working directory, containing the raw files. Its name should be `ENTSOG_TP_data_YYYY-MM-DD`, where `YYYY-MM-DD` is the current date.

We still have to collect all the raw files and merge the information into a single HDF5 file for easier post-processing.

```
dir_name = f'ENTSOG_TP_data_{dt.date.today()}'
etp.raw_to_file(dir_name)
```

Now there should be a new HDF5 file in the current working directory with the name `ENTSOG_TP_data_YYYY-MM-DD.h5`, where `YYYY-MM-DD` is the current date.

We also want to show LNG send-out flow in our plot, so we need also to download data from the GIE ALSI transparency platform. To access the transparency platforms of GIE, you must set an API key (to be requested from GIE). If necessary, the user can also set a proxy.

```
gtp.update_gie_alsi_archive(download_start, download_end, proxy=proxy,

                            api_key=api_key_alsi)
```

After execution of the function above, a new subdirectory should have been created under the current working directory, containing an Excel spreadsheet file with the LNG data. By default its path is `GIE_TPs_Archive/GIE_ALSI_archive_GWh_d.xlsx`.

4.5.2 Step 2: Load data from file

The data that we have downloaded above shall now be loaded from the files that were created.

Typically, Step 1 and Step 2 would occur in different scripts and at different times, but here we show both in a single script.

First, we load the data from the ENTSOG Transparency Platform.

```
raw_file = f'ENTSOG_TP_data_{today}.h5'
raw = etp.load_raw_file(raw_file)
df_lng = pd.read_excel('GIE_TPs_Archive/GIE_ALSI_archive_GWh_d.xlsx',
                       index_col=[0, 1])
```

4.5.3 Step 3: Process data

In this case, we set the time under analysis equal to the period for which data was downloaded.

```
analyse_end = download_end
analyse_start = download_start
```

The data is now re-indexed and periodized.

```
df = etp.reindex_and_periodize(raw, analyse_start, analyse_end)
```

For each corridor, we now select and aggregate data of the edges it consists of. To this end, we iterate over the dictionary `corridors` created earlier.

```
flow = {}
for corr, edges in corridors.items():
    flow[corr] = etp.select_and_aggregate(edges, topo, df, 'flow',
                                          quiet=True)
```

The keyword argument `quiet=True` suppresses certain warning messages in case the ENTSOG TP data does not contain any data for a certain edge within the chosen period.

It is time to aggregate over the edges of each corridor. Also, we create a new `pandas.DataFrame` from the dictionary, as it is more convenient to work with. We do this using the Pandas function `concat()`.

```
flow2 = {}
for corr, edges in corridors.items():
    flow2[corr] = flow[corr].sum(axis=1)
flow3 = pd.concat(flow2.values(), keys=flow2.keys(), axis=1)
```

Now we can also add the LNG data to the table. As with the ENTSOG data, the standard unit of measure is GWh/d.

```
total_lng = df_lng.groupby(level=1).sum()
flow3['LNG'] = total_lng.sendOut
```

Now we have all data collected in a neat `pandas.DataFrame`, as shown in Table 33.

Table 33. Example of the Pandas DataFrame flow3.

	North Africa	UK	North Sea	East	Caspian	LNG
date						
2022-01-01	1045.096815	809.413887	2721.002172	2874.025006	314.715318	2652.1
2022-01-02	1100.189346	813.269950	2727.356532	2712.597102	314.903299	2885.9
2022-01-03	1112.887012	854.052104	2724.963894	2596.721511	314.724888	3226.0
2022-01-04	1092.480178	642.121968	2722.616074	2526.575072	292.676872	3483.2
2022-01-05	1109.952828	226.044013	2554.094241	2736.340816	307.055776	3738.5
...	...	...	...	...	...	...
2022-07-27	890.855759	886.150629	2734.608586	1381.789352	400.980734	4135.6
2022-07-28	904.186579	712.508021	2642.287070	1354.420628	401.312844	4203.8
2022-07-29	875.642759	935.678508	2701.673047	1295.103263	401.047777	3887.3
2022-07-30	890.895537	887.765494	2627.176664	1225.749289	352.730631	3473.8
2022-07-31	915.417705	876.496070	2617.462963	1188.821157	352.735684	3471.3

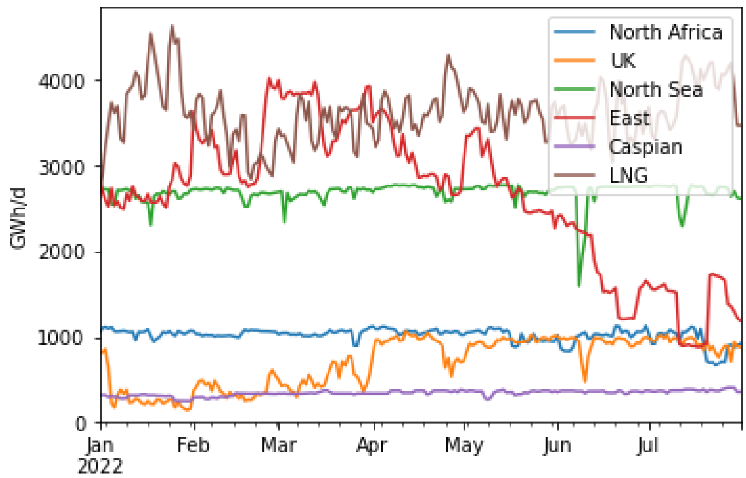
Source: JRC, 2022.

4.5.4 Step 4: Visualise data

Now the data can be plotted using standard methods offered by the Pandas package or any means the user prefers. A code example is shown below and the resulting diagram in Figure 7.

```
flow3.plot()
plt.ylim(0)
plt.xlabel(None)
plt.ylabel('GWh/d')
```

Figure 7. Physical flow along major inflow corridors to the EU.

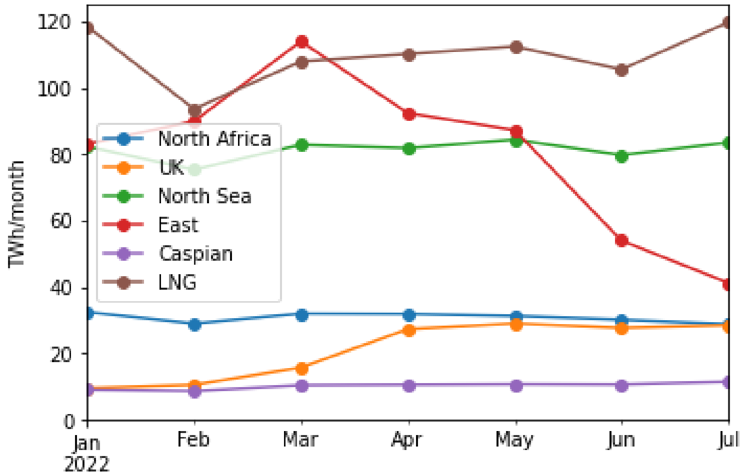


Source: JRC based on ENTSOG TP and ALSI TP, 2022.

Of course, the daily data could also be resampled, for example using monthly resolution, as shown in Figure 8. The code is given below. In this example, we choose the unit TWh/month.

```
(flow3 / 1000).resample('M').sum().plot(marker='o')
plt.ylim(0)
plt.title('Physical flow along major inflow corridors to the EU')
plt.xlabel(None)
plt.ylabel('TWh/month')
```

Figure 8. Physical flow along major inflow corridors to the EU, aggregated with monthly resolution.



Source: JRC based on ENTSOG TP and ALSI TP, 2022.

5 Conclusions

In this report we have described in detail three transparency platforms (TPs), one operated by the European Network of Transmission System Operators of Gas (ENTSOG) and two by Gas Infrastructure Europe (GIE), all of which publish up-to-date data of high temporal resolution (daily or even hourly in the case of the ENTSOG TP). Our conclusion is that these TPs already represent an established and highly useful source of information for many purposes, be it in research, policy making, or in the business world. Since additional indicators such as production and (for some countries) consumption data were added as of lately, it has become an even better asset. Needless to say, if consumption data could cover all countries in the future, its usefulness would increase even further.

The ENTSOG TP is arguably the largest and most complex database of the three. Making best use of the provided data is not trivial and usually requires several post-processing steps. For this reason, we have developed the Python package `eurogastp` implementing the steps necessary to download and post-process data from the ENTSOG TP, while also supporting the user to download data from the GIE TPs, ALSI and AGSI+ and combining all three data sources.

Most data on the ENTSOG TP relates to flows through a complex network. Correct interpretation of the provided data requires sufficient knowledge of the European gas network topology. Linking each time series to the correct facility is not always trivial and partially requires external knowledge. A careful selection and combination of raw time series is necessary for its usage, which includes case-by-case decision making in case of redundant, flawed or missing data, so that the resulting dataset can easily be further processed, filtered and aggregated by time and space (e.g. by country).

One important element of the presented method to process data from the ENTSOG Transparency Platform is therefore the *topology file* which defines a mapping from ENTSOG's source topology to a user-defined target topology. It is also responsible for choosing how to select and aggregate (and clean) data from the ENTSOG TP, carefully deciding which data series to take to avoid – among other issues – possible double-counting of cross-border flows and to reach the best possible coverage over time and space in the resulting dataset. It is however crucial for this topology file being updated regularly, as the source topology – and sometimes the reporting behaviour of the infrastructure operators – is constantly evolving (new infrastructure being built, data being reported under a different name, new indicators being reported by a participating TSO, etc.).

It must also be mentioned that the topology file currently provided with the software does not (yet) cover all indicators available on the ENTSOG platform, nor does it cover all network points. While all existing cross-border interconnection points and production and consumption data are accounted for (at the time of writing), there are still gaps in other network point categories, especially those regarding the inflow and outflow from/to LNG terminals and underground storage facilities. These gaps can however be compensated by using the respective data from the GIE platforms instead. Nevertheless, the completion of the topology file as well as regular updates thereof remain important future tasks.

It should also be mentioned that the provided topology file can serve as basis for user-specific customizations. Not only could the user complete and update/correct the existing topology file, but she is also encouraged to create a fully custom target topology that may be fit for her needs.

Future releases of the software package could also include additional functions for visualising the data of the ENTSOG and GIE TPs. This has however intentionally been left out so far as basically any of the usual means of visualising the resulting data can readily be used (including the plotting facilities incorporated in the Pandas package) and every user might have very specific needs to this regard (corporate design requirements etc.). Another possible extension would be to enable the software to work with hourly data, as right now it allows only the use gas days as output frequency.

`eurogastp` is an open source project[23]. The original authors of the software warmly welcome not only feedback, but any community-inspired additions and alterations, and hope the package could eventually serve as an element in many third-party software solutions.

[23] https://code.europa.eu/jrc-energy-security/eurogastp, https://github.com/ec-jrc/eurogastp

References

ENTSOG, *Professional Data Warehouse System Documentation – Transparency Platform User manual*, version 4.5.00, 2021; https://transparency.entsog.eu/#/helps/tp-documentation.

ENTSOG, *ENTSOG Transparency Platform – API User Manual*, version 2.1, 2018; https://transparency.entsog.eu/#/helps/tp-documentation.

Zaccarelli, N., Giaccaria, S., Feofilovs, M., Bolado-Lavìn, R., The European Natural Gas Demand database (ENaGaD) – An archive of daily time series from 2015 to 2020, Publications Office of the European Union, Luxembourg, 2021, ISBN 978-92-76-40208-4, doi:10.2760/497677, JRC125730.

Zaccarelli, Nicola; Giaccaria, Sergio; Feofilovs, Maksims; Bolado--Lavin, Ricardo; The European Natural Gas Demand database (ENaGaD), version 1.1 from 15 March 2022; doi:10.5281/zenodo.6364875.

List of abbreviations and definitions

AGSI+	Aggregated Gas Storage Inventory (one of GIE's transparency platforms)
ALSI	Aggregated LNG Storage Inventory (one of GIE's transparency platforms)
API	Application programming interface
CSV	Comma-separated values (a file format)
DIS	Distribution point
DTMI	Declared Total Maximum Inventory (of an LNG terminal)
DTRS	Declared Total Reference Send-out (send-out capacity of an LNG terminal)
EIC	Energy Identification Code
ENaGaD	European Natural Gas Demand database
ENTSOG	European Network of Transmission System Operators of Gas
EU	European Union
FNC	Final consumption point
kWh/Nm^3	Kilowatt-hours per normal cubic meter
GCV	Gross-calorific value
GIE	Gas Infrastructure Europe
GWh	Gigawatt-hours
GWh/d	Gigawatt-hours per day
IP	Interconnection point
JRC	Joint Research Centre (of the European Commission)
LNG	Liquefied natural gas
LSO	LNG system operator
mcm	Million cubic meters
mcm/d	Million cubic meters per day
MS	Member State (of the European Union)
NaN	Not-a-number (used in computing systems to indicate an invalid or missing number)
NG	Natural gas
SSO	Storage system operator
TP	Transparency platform
TSO	Transmission system operator
UGS	Underground gas storage
VIP	Virtual interconnection point
VTP	Virtual trading point

List of figures

List of tables

Annexes

Annex 1. Additional tables

Table 34. Average gross-calorific values at EU-external-border interconnectors from January 2022.

Import route	Interconnection point	Mean GCV (kWh/Nm^3)	Finite flow
North Africa -> ES	**Almeria (DZ-ES)**	11.55	X
	Tarifa (MA-ES)	11.44	
North Africa -> IT	**Gela (LY-IT)**	11.56	X
	Mazara (TN-IT)	11.79	X
UK	**Bacton (UK-NL)**	11.54	X
	Moffat (UK-IE)	11.64	X
	South-North (UK-IE)	---	
	Zeebrugge (UK-BE)	11.53	X
North Sea (Norway)	**Dornum (NO-DE)**	11.62	X
	Dunkerque (NO-FR)	11.42	X
	Emden EPT1 (NO-DE)	11.65	X
	Zeebrugge (NO-BE)	11.52	X
East -> Nord Stream	**Greifswald (RU-DE)**	11.26	X
East -> Baltic+Finland	**Imatra (RU-FI)**	11.22	X
	Kotlovka (BY-LT)	11.21	X
	Luhamaa (RU-EE)	---	
	Narva (RU-EE)	---	
	Varska (RU-EE)	---	
East -> YAMAL	**Kondratki (BY-PL/Yamal)**	11.19	X
	Tieterowka (BY-PL)	11.21	X
	Wysokoje (BY-PL)	11.22	X
East -> Ukraine	**Aurit (UA-RO)**	---	
	Beregdaróc VIP (UA-HU)	11.37	X
	Budince (UA-SK)	11.38	
	Drozdow (UA-PL)	11.29	X
	Isaccea1 (UA-RO)	11.37	X
	Isaccea2 (UA-RO/TBP)	---	
	Isaccea3 (UA-RO/TBP)	---	
	Velke Kapusany (UA-SK)	11.38	X
East -> Turkey	**Kipi (TR-EL)**	11.24	
	Strandzha 2 (TR-BG/BTG)	11.34	X
Caspian	**Kipoi (TR/TANAP-EL/TAP)**	11.74	X

Source: JRC based on ENTSOG TP, 2022.

Table 35. List of EU-external-border interconnection points, partially with direct links to the TSOs' data portals.

	IP	URL
NORD STREAM	Greifswald (RU->DE)	https://www.nord-stream.info/
BALTIC-FINLAND	Imatra (RU->FI)	https://gasgrid.fi/en/gas-market/market-information/
	Kotlovka (BY->LT)	https://www.ambergrid.lt/en/services/capacity-information/capacity-and-gas-flow-data
	Korneti (RU->LV)	https://capacity.conexus.lv/?id=158&lang=eng&fp_punkti%5B%5D=11&fp_pa_dienam=1&fp_datums_no=2022-02-23&fp_datums_lidz=2022-02-24&fp_submit=Generate+report
	Varska (RU->EE)	https://dashboard.elering.ee/en/gas-transmission/volume?interval=minute&period=days&start=2022-02-23T06:00:00.000Z&end=2022-02-24T05:59:59.999Z&show=table
	Narva (RU->EE)	https://dashboard.elering.ee/en/gas-transmission/volume?interval=minute&period=days&start=2022-02-23T06:00:00.000Z&end=2022-02-24T05:59:59.999Z&show=table
YAMAL (BY-PL)	Wysokoje (BY->PL)	https://swi.gaz-system.pl/swi/public/#!/ksp/actualQuantity?lang=en
	Tieterowka (BY->PL)	https://swi.gaz-system.pl/swi/public/#!/ksp/actualQuantity?lang=en
	Kondratki (BY->PL-YAM)	https://swi.gaz-system.pl/swi/public/#!/sgt/actualQuantity?lang=en
UKRAINE	Aurit (UA->RO)	https://www.transgaz.ro/en/clients/operational-data/physical-flows
	Isaccea1 (UA->RO)	ENTSOG
	Drozdow (UA->PL)	ENTSOG
	Bereg14 (UA->HU)	https://ipnew.rbp.eu/Fgsz.Tso.Data.Web/?locale=en#main
	Velke (UA->SK)	https://tis.eustream.sk/TisWeb/#/?nav=rd.flw
	Isaccea2 (UA->TBP)	ENTSOG
	Isaccea3 (UA->TBP)	ENTSOG
	Budince (UA->SK)	https://tis.eustream.sk/TisWeb/#/?nav=rd.flw
	Bereg VIP (UA->HU)	https://ipnew.rbp.eu/Fgsz.Tso.Data.Web/?locale=en#main
TURKEY	Kipi (TR->GR)	http://www.δεσφα.gr/scada/quantity-hourly.php?la=EN&T=1645717068227
	Strandzha2 (TR->BTG)	https://www.bulgartransgaz.bg/en/pages/operational-data-185.html

Source: JRC.

GETTING IN TOUCH WITH THE EU

In person

All over the European Union there are hundreds of Europe Direct centres. You can find the address of the centre nearest you online (european-union.europa.eu/contact-eu/meet-us_en).

On the phone or in writing

Europe Direct is a service that answers your questions about the European Union. You can contact this service:

— by freephone: 00 800 6 7 8 9 10 11 (certain operators may charge for these calls),

— at the following standard number: +32 22999696,

— via the following form: european-union.europa.eu/contact-eu/write-us_en.

FINDING INFORMATION ABOUT THE EU

Online

Information about the European Union in all the official languages of the EU is available on the Europa website (european-union.europa.eu).

EU publications

You can view or order EU publications at op.europa.eu/en/publications. Multiple copies of free publications can be obtained by contacting Europe Direct or your local documentation centre (european-union.europa.eu/contact-eu/meet-us_en).

EU law and related documents

For access to legal information from the EU, including all EU law since 1951 in all the official language versions, go to EUR-Lex (eur-lex.europa.eu).

Open data from the EU

The portal data.europa.eu provides access to open datasets from the EU institutions, bodies and agencies. These can be downloaded and reused for free, for both commercial and non-commercial purposes. The portal also provides access to a wealth of datasets from European countries.

The European Commission's science and knowledge service

Joint Research Centre

JRC Mission

As the science and knowledge service of the European Commission, the Joint Research Centre's mission is to support EU policies with independent evidence throughout the whole policy cycle.

EU Science Hub
joint-research-centre.ec.europa.eu

@EU_ScienceHub

EU Science Hub - Joint Research Centre

EU Science, Research and Innovation

EU Science Hub

EU Science

Publications Office
of the European Union