User Manual and Technical Report

Heterogeneous Energy Data Analysis System

Version 1.2

February 6, 2026

Environment: MATLAB R2020a or higher

Contents

1 Introduction

1.1 Software Objective

This MATLAB-based software addresses a critical problem faced by energy data analysts: the exploitation of Excel reports (such as the *BP Statistical Review of World Energy*) whose structure is **heterogeneous, unpredictable, and non-standardized**.

These files typically present:

- Headers offset vertically (row 2, 3, or even 6)
- Dispersed units of measurement (cell A1, A3, or previous row)
- Mixed formats: *Wide* data (countries in rows, years in columns) or *Long* (years in rows, series in columns)
- Merged cells, parasitic text, and typographic inconsistencies

1.2 Functional Scope

The software offers three main capabilities:

1. **Robust ETL**: Automated extraction, transformation, and loading with intelligent structure detection
2. **Interactive Interface**: Menu-based navigation with contextual search
3. **Visual Analysis**: Chart generation with statistical tools (regression, prediction, derivation)

Target Audience

This software is intended for researchers, engineers, and analysts working with energy time series data from documentary sources (Excel reports, statistical databases).

2 Technical Architecture

2.1 System Overview

The software is based on a modular three-layer architecture:

Layer	Description
Backend	Import pipeline with raw parsing and normalization
Storage	Persistent database structured in `containers.Map`
Frontend	Interactive user interface with visualization engine

2.2 Database Structure

Data is stored in a `Base_Complete_Energy.mat` file following a three-level hierarchy:

```
BaseDeDonnees : Map
            "2023" : Map
                  "Oil Production" : Table (100 45 )
                  "Coal Consumption" : Table (75 50 )
                  ...
            "2024" : Map
                  ...
```

Listing 1: Logical database structure

Advantages of this architecture:

- $O(1)$ access by year and sheet (no repetitive parsing)
- Native support for heterogeneous structures (each table retains its metadata)
- Immediate persistence via MATLAB `save`

2.3 The Cache System

Critical Optimization

Each processed file is saved as `Cache_2023.mat`. On the next launch, if the cache exists, loading takes **0.3 seconds instead of 45 seconds**.

```
fichierCache = fullfile(dossierExcel, ['Cache_' anneeReport '.mat']);

if exist(fichierCache, 'file')
    fprintf('[CACHE] Fast loading...\n');
    charge = load(fichierCache);
    DonneesAnnee = charge.DonneesAnnee;
else
    DonneesAnnee = TraiterUnFichier(cheminFichier, fid);
    save(fichierCache, 'DonneesAnnee');
end
```

Listing 2: Cache logic in Importation.m

To regenerate data: Manually delete the `Cache_*.mat` files.

3 User Guide

3.1 Installation and Preparation

3.1.1 System Requirements

- MATLAB R2020a or higher

- Toolboxes: *Statistics and Machine Learning* (optional for `movmean`)
- Operating system: Windows, macOS, or Linux

3.1.2 File Installation

1. Create a working directory (e.g., `C:\EnergyProject`)
2. Place all `.m` files in this directory:
   - `Importation.m`
   - `Explorateur.m`
   - `boucle_analyse_pays.m`
   - `tracer_tableau.m`
   - `analyse_statistique.m`
   - `extraire_colonnes_temps.m`
   - `demander_choix.m`
3. Create a subdirectory for your Excel files (or place them directly)

3.1.3 Excel File Naming Conventions

CRITICAL RULE

Excel files **must** contain a 4-digit year in their name:

- `BP_Statistical_Review_2023.xlsx`
- `Energy_Data_2024.xls`
- `Annual_Report.xlsx` (no detectable year)

This year serves as a key in the database. If no year is detected, the file will be indexed as `Unknown_1`.

3.2 Step 1: Data Import

1. Open MATLAB and navigate to your working directory:

```
cd 'C:\EnergyProject'
```

2. Launch the import:

```
Importation()
```

3. The process displays:

```
Beginning processing of 3 files...
[PROCESSING] Reading BP_2023.xlsx...
[CACHE] Fast loading of BP_2024.xlsx...
Processing complete. See Rapport_Importation.txt
```

Generated files:

- `Base_Complete_Energy.mat`: Complete database
- `Cache_*.mat`: Cache files by year
- `Rapport_Importation.txt`: Error log

3.3 Step 2: Interactive Exploration

Launch the interface:

```
Explorateur()
```

3.3.1 Menu Navigation

Main Menu: Year Selection

```
--- MAIN MENU ---
>> Choose report year (or 'q'=quit, 'r'=return): 2023
```

Secondary Menu: Sheet Selection The system offers intelligent text search:

```
--- REPORT 2023 ---
>> Search data sheet (e.g., "Solar") or 'r'=return: oil
```

If multiple matches exist:

```
Multiple results:
 1. Oil Production
 2. Oil Consumption
 3. Oil Prices
>> Number: 1
```

3.4 Step 3: Chart Generation

3.4.1 Entity Selection

Once the sheet is loaded, enter the countries/entities to plot (comma-separated):

```
[Analysis: Oil Production | Unit: Million tonnes]
>> Your choice: France, Germany, United Kingdom
```

Partial matching enabled: Typing `Unit` will find `United States` and `United Kingdom`.

3.4.2 Period Selection

```
Range: 1965 - 2023
>> Start Year (Enter=Min): 2000
>> End Year   (Enter=Max): 2023
```

Simply press `Enter` to use the entire available range.

3.4.3 Automatic Display

The software automatically generates:

1. A table of descriptive statistics (mean, median, standard deviation)
2. A temporal (line) or categorical (bar) chart depending on context
3. An automatic legend (if fewer than 20 series)

4 Advanced Analysis Features

4.1 Command Overview

After generating a chart, you can apply analyses directly by typing a command:

Command	Category	Description
`reg`	Graphical	Displays linear trend (regression)
`pred`	Graphical	+5 year prediction (extrapolation)
`smooth`	Graphical	Smoothing by moving average (5 years)
`poly2`	Graphical	Quadratic polynomial fit
`deriv`	Calculation	Calculates derivative (rate of change)
`cumul`	Calculation	Cumulative integral (total over period)
`export`	System	Exports data to Excel
`list`	System	Lists all available entities
`r`	System	Return to previous menu

4.2 Graphical Analyses

4.2.1 Linear Regression (`reg`)

Objective: Identify the average growth/decline trend.

Mathematical formula:

$$y = ax + b \quad \text{where } a = \text{slope (annual growth)} \tag{1}$$

Console interpretation:

```
> France: Average growth = 2.34 units/year
```

A positive slope ($a > 0$) indicates an increase over time.

4.2.2 Prediction (`pred`)

Extrapolates the linear trend over the next 5 years.

Limitation

This prediction assumes the historical trend continues. It is unreliable for volatile series or those with structural breaks.

Console output:

```
> Germany: +5 year prediction = 145.67
```

4.2.3 Smoothing (`smooth`)

Applies a 5-year centered moving average to reduce noise.

Formula:

$$y_{\text{smoothed}}(t) = \frac{1}{5} \sum_{i=-2}^{+2} y(t+i) \tag{2}$$

Useful for identifying long-term trends by removing annual fluctuations.

4.2.4 Polynomial Fit (`poly2`)

Fits a degree-2 polynomial: $y = ax^2 + bx + c$

Allows capturing acceleration or deceleration dynamics (parabolic curves).

4.2.5 Base 100 Normalization (`base100`)

Objective: Compare the **relative dynamics** of multiple countries with very different orders of magnitude (e.g., China vs. Belgium).

Formula: Sets the value of the first year (t_0) to 100 for each series:

$$y_{\text{indexed}}(t) = \frac{y(t)}{y(t_0)} \times 100 \tag{3}$$

Visual interpretation: All curves start from the same point ($y = 100$).

- Curve > 100: Growth relative to the starting year.
- Curve < 100: Decline.

4.3 Computational Analyses

4.3.1 Derivation (`deriv`)

Calculates the **rate of change** using finite differences:

$$\frac{dy}{dx} \approx \frac{y_{t+1} - y_t}{x_{t+1} - x_t} \tag{4}$$

Use cases:

- Identify years of maximum growth
- Detect ruptures (abrupt policy changes)

Console output:

```
> China: Peak variation = 12.5 in 2007
```

4.3.2 Cumulative Integration (`cumul`)

Calculates the **cumulative total** using the trapezoidal method:

$$\text{Cumul} = \sum_{i=1}^{n-1} \frac{y_i + y_{i+1}}{2} \times (x_{i+1} - x_i) \tag{5}$$

Use cases:

- Total energy production over a decade
- Cumulative CO_2 emissions

4.3.3 Compound Annual Growth Rate (`cagr`)

Calculates the theoretical constant growth rate that would transition from the initial to the final value (geometric smoothing).

Formula:

$$\text{CAGR} = \left(\frac{y_{\text{end}}}{y_{\text{start}}} \right)^{\frac{1}{n}} - 1 \tag{6}$$

Unlike linear regression (`reg`) which is additive, CAGR is multiplicative. It is standard for financial and energy analyses.

Console output:

```
> India: CAGR = 5.42%
```

4.4 Advanced Statistical Analyses

4.4.1 Volatility / Risk (`vol`)

Measures the **instability** of the time series by calculating the rolling standard deviation over a sliding window (5 years).

Formula:

$$\sigma_t = \sqrt{\frac{1}{N-1} \sum_{i=0}^{N-1} |y_{t-i} - \mu|^2} \tag{7}$$

Use cases:

- Assess energy security (stable production is less risky).
- Analyze price volatility (Spot vs. Long Term).

Console output:

```
> Crude Oil: Average Volatility = 14.20
```

4.4.2 Anomaly Detection (`anom`)

Uses a statistical algorithm to identify outliers that deviate significantly from the local median.

Method: Detection based on median absolute deviation (MAD):

$$|y_t - \text{MovingAverage}| > 3 \times \sigma_{\text{local}} \tag{8}$$

Utility

Automatically identifies **exogenous shocks** (wars, economic crises, data errors) without manual inspection.

Console output:

```
> Venezuela: 3 anomalies detected
```

4.5 System Commands

4.5.1 Excel Export (`export`)

Exports the last plotted sub-table to `Resultats_Export.xlsx`.

Output Format

The Excel file contains:

- Column 1: Entity names
- Columns 2..N: Selected years
- Metadata (unit) in file properties

4.5.2 Listing (`list`)

Displays all entities available in the current sheet. Useful for knowing the exact spelling of countries.

5 Error Handling and Troubleshooting

5.1 Common Problems

5.1.1 Empty Chart or Incorrect Axes

Symptom: The chart displays but without data, or the years on the X-axis are incorrect.

Possible causes:

1. **Corrupted Cache:** A `Cache_*.mat` file contains an obsolete version.
2. **Year Regex:** Years are not detected (e.g., format "Year 2023" instead of "2023").

Solutions:

1. Delete all `Cache_*.mat` files
2. Relaunch `Importation()`
3. Check `Rapport_Importation.txt` for `[SKIP]` messages

5.1.2 Incorrect Structure Detection (Wide/Long)

Symptom: The software performs an "Auto-Pivot" when the data is already in the correct format.

Diagnosis: Open the `Rapport_Importation.txt` file and look for the line:

```
Detected structure: Long
```

Solution: Modify the detection threshold in `Importation.m` (line 138):

```
if countYearsHeader >= 3  % Instead of >= 2
    structureType = 'Wide';
```

5.1.3 Missing or Incorrect Unit

Symptom: The chart displays "Unit: Unknown" or an incorrect value.

Cause: The upward scan algorithm did not find any text above the header.

Temporary solution: Manually add the unit after loading:

```
Tableau.Properties.UserData.Unit = 'Million␣barrels␣per␣day';
```

5.2 Specific Error Messages

Message	Meaning
`No Excel file found`	The specified directory contains no `.xls` or `.xlsx` files
`Unreadable structure`	The sheet contains no detectable dense row (probably empty or poorly formatted)
`No temporal data`	No year column was detected (check header format)
`Out of range`	The entered years are outside the available period

5.3 Result Validation

To verify the integrity of imported data:

1. Manually load the database:

```
load('Base_Complete_Energy.mat');
keys(BaseDeDonnees)  % Lists the years
```

2. Inspect a specific table:

```
T = BaseDeDonnees('2023')('Oil Production');
head(T)  % Displays the first 8 rows
T.Properties.UserData  % Displays metadata
```

3. Verify year columns:

```
[annees, ~] = extraire_colonnes_temps(T);
disp(annees)  % Should display [1965, 1966, ..., 2023]
```

6 Practical Use Case Examples

6.1 Use Case 1: Multi-Country Comparison

Objective: Compare oil production of 5 countries over 20 years with trends.

```
>> Explorateur()
>> Choose Year: 2023
>> Search Sheet: oil production
>> Your choice: USA, Russia, Saudi Arabia, China, Canada
>> Start Year: 2000
>> End Year: 2023

% Chart displays with 5 curves

>> reg  % Adds linear trends
> USA: Average growth = -0.15 Mt/year
> Russia: Average growth = 0.45 Mt/year
> Saudi Arabia: Average growth = 0.12 Mt/year

>> export  % Saves to Excel
```

Listing 3: Complete session

6.2 Use Case 2: Volatility Analysis

Objective: Identify periods of high variation in oil prices.

```
>> Search Sheet: oil prices
>> Your choice: Brent, WTI
>> Start Year: 1980
>> End Year: 2023

>> deriv  % Calculates derivative
> Brent: Peak variation = 45.2 $/year in 2008
> WTI: Peak variation = 42.8 $/year in 2008

% Interpretation: 2008 financial crisis
```

6.3 Use Case 3: Energy Forecasting

Objective: Estimate electricity consumption in 2030.

```
>> Search Sheet: electricity consumption
>> Your choice: World
>> Start Year: 1990
>> End Year: 2023

>> smooth  % Smooths the data
>> pred    % +5 year extrapolation
> World: +5 year prediction = 28,450 TWh (2028)

% Note: For 2030, rerun with +7 year prediction
% or manually adjust in analyse_statistique.m
```

7 Appendices

7.1 Appendix A: Displayed Statistical Metrics

For each chart, the software automatically calculates:

Metric	MATLAB Formula
Mean	`mean(data, 'omitnan')`
Median	`median(data, 'omitnan')`
Standard Deviation	`std(data, 'omitnan')`
Minimum	`min(data)`
Maximum	`max(data)`

7.2 Appendix B: Useful MATLAB Commands

```
% List all available years
load('Base_Complete_Energy.mat');
disp(keys(BaseDeDonnees))

% Inspect a specific sheet
T = BaseDeDonnees('2023')('Oil␣Production');
summary(T)

% Check metadata
disp(T.Properties.UserData)

% Force cache regeneration
delete('Cache_*.mat')
Importation()

% Export a table manually
writetable(T, 'manual_export.xlsx')
```

Listing 4: Advanced debugging commands

8 Glossary

ETL Extract, Transform, Load - Data processing workflow

Wide Format Structure where years are in columns (rows = entities)

Long Format Structure where years are in rows (columns = series)

Auto-Pivot Automatic Long → Wide transformation

Cache `.mat` file containing preprocessed data

Metadata Ancillary information (unit, sheet name, structure)

Heuristic Algorithm based on practical rules (not mathematically optimal)