Software Impacts 28 (2026) 100839

Contents lists available at ScienceDirect

Software Impacts

journal homepage: www.journals.elsevier.com/software-impacts

Original Software Publication

GEBCO API: A Web-based and RESTful interface for accessing global bathymetric data

Kumars Mahmoodi [a,*], Jari Böling [b]

[a] Faculty of Natural Sciences and Engineering, Åbo Akademi University, Turku, Finland
[b] Automation, Mechanical and Materials Engineering, University of Turku, Finland

ARTICLE INFO

Keywords:
Bathymetry
GEBCO 2024
FastAPI
Oceanography
Geospatial data

ABSTRACT

This paper presents GEBCO API, a web-based and RESTful platform for accessing global bathymetric data from the GEBCO 2024 dataset. The application provides an interactive browser-based interface alongside programmatic API endpoints for retrieving water depth at single locations, batches of points, or user-defined geographic areas. Outputs can be downloaded in NetCDF format, enabling seamless integration into scientific workflows. Developed using FastAPI and HTML and deployed as a cloud-accessible service, the GEBCO API lowers technical barriers to high-resolution bathymetric data, supporting research and engineering applications in oceanography, marine energy, and geospatial analysis.

Code metadata

Current code version	v1.0.0
Permanent link to code/repository used for this code version	https://github.com/SoftwareImpacts/SIMPAC-2026-55
Permanent link to Reproducible Capsule	Not available at the time of publication
Legal Code License	MIT License
Code versioning system used	Git
Software code languages, tools, and services used	Python, pandas, numpy, FastAPI, Uvicorn, xarray, NetCDF4, Docker, HTML, JavaScript
Compilation requirements, operating environments & dependencies	Python 3.8+, Linux, macOS, Windows; see `requirements` file
If available, link to developer documentation/manual	https://github.com/kumarsmahmoodi/gebcoapi/blob/main/static/example.html
Support email for questions	kumars.mahmoodi@abo.fi

1. Introduction

Bathymetric data are fundamental to a wide range of scientific, engineering, and operational applications, including oceanography [1], marine geophysics [2], coastal and offshore engineering [3,4], climate studies [5], marine renewable energy assessment [6,7], ship routing [8], ship fuel consumption [9], and environmental impact analysis [10]. Among the available global bathymetric products, the General Bathymetric Chart of the Oceans (GEBCO) dataset [11] represents one of the most comprehensive and widely used sources of global elevation and seafloor depth information. This dataset provides near-global coverage at high spatial resolution and is distributed in NetCDF format, making it suitable for advanced numerical and data-driven analyses.

Despite its wide adoption, direct use of the GEBCO dataset often presents practical challenges for users, particularly those who require rapid access to depth values for specific locations or regions without downloading, handling, and post-processing very large NetCDF files. The full GEBCO dataset is several gigabytes in size and typically requires specialized software, programming expertise, and computational resources to extract subsets or query individual points. These barriers can hinder efficient integration of bathymetric data into lightweight workflows, web-based applications, or external tools such as numerical models, optimization frameworks, and decision-support systems.

To address these limitations, this paper introduces the *GEBCO API*, a web-based and RESTful software application designed to provide flexible, programmatic, and user-friendly access to global bathymetric

* Corresponding author.
E-mail addresses: kumars.mahmoodi@abo.fi (K. Mahmoodi), jari.boling@utu.fi (J. Böling).

https://doi.org/10.1016/j.simpa.2026.100839
Received 9 March 2026; Accepted 8 April 2026

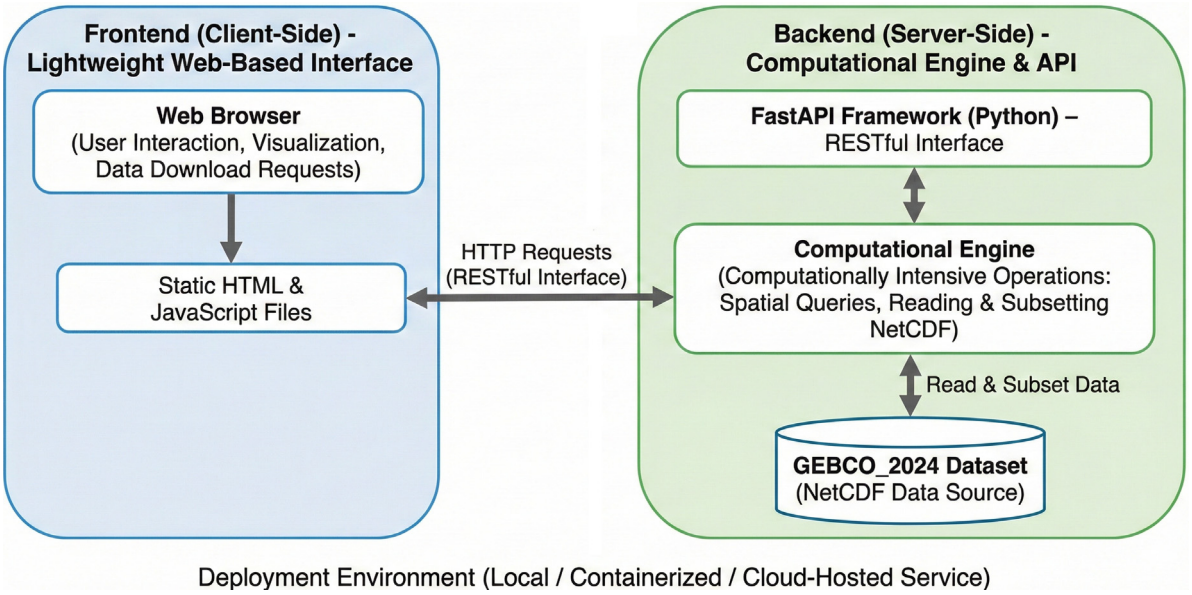

Fig. 1. The overall architecture diagram of the GEBCO API App.

data derived from the GEBCO dataset. The application is implemented using the FastAPI framework and exposes a set of well-defined API endpoints that enable users to retrieve water depth values for single locations, batches of coordinates, or spatially defined areas. In addition, the application offers an interactive web interface that allows users to visually explore bathymetric data, query depths directly from an online map, and download spatial subsets in NetCDF format for further offline analysis.

The GEBCO API is designed with interoperability and reproducibility in mind. It supports integration with external software environments through standard HTTP requests and returns machine-readable outputs in JSON and NetCDF formats. Example client implementations, including MATLAB-based workflows, are provided to demonstrate practical usage and facilitate adoption by the scientific and engineering communities. The software is fully open-source, containerized using Docker, and deployed on a cloud-based infrastructure, allowing both local installation and scalable online access.

By lowering the technical barrier to accessing high-quality global bathymetric data, the GEBCO API aims to support rapid prototyping, reproducible research, and cross-disciplinary applications that rely on accurate seafloor information. This paper describes the design, functionality, and deployment of the software, highlighting its capabilities and potential impact as a reusable research and engineering tool.

2. Background and related software

Despite broad usage, direct interaction with GEBCO data typically requires users to download multi-gigabyte NetCDF files and possess domain-specific expertise in data handling tools such as MATLAB, Python, or GIS software. This workflow can present a barrier for rapid data access, automation, and integration into operational pipelines, particularly for users who require repeated point-based queries, batch extractions, or on-demand regional subsets. As a result, there is a growing need for lightweight, programmatic, and web-accessible interfaces that simplify access to high-resolution bathymetric data without compromising data fidelity.

Several existing technologies partially address this challenge. Data distribution services such as OPeNDAP [12] and ERDDAP [13] enable remote access to NetCDF-based datasets and support standardized query protocols. These services are powerful and flexible but often require familiarity with complex query syntax, metadata structures, and server-side configurations. In addition, their interfaces are generally designed for expert users and are not optimized for rapid integration into custom scripts, lightweight applications, or web-based workflows.

In parallel, various GIS platforms and desktop applications provide graphical interfaces for visualizing and extracting bathymetric data. While effective for exploratory analysis, these tools are typically not well-suited for automation, large-scale batch processing, or seamless integration into numerical modeling and decision-support systems. Furthermore, desktop-based solutions often impose platform-specific dependencies and licensing constraints.

The GEBCO API presented in this work is designed to complement, rather than replace, existing bathymetric data access mechanisms by providing a focused, lightweight, and developer-friendly alternative. By exposing commonly required bathymetric queries through a RESTful interface, the software lowers the technical barrier to GEBCO data usage and facilitates integration into a broad range of Earth science workflows. The application emphasizes simplicity, reproducibility, and interoperability, enabling users to retrieve point-based depths, batch queries, and regional subsets directly through standard HTTP requests or via web-based interfaces, without the need for manual data handling or specialized software environments.

Table 1 summarizes a qualitative comparison between the proposed GEBCO API and commonly used bathymetric data access approaches. It highlights the unique software-oriented contributions of this work.

3. Software overview and architecture

3.1. Overall architecture

The overall architecture (client–server model) diagram of the GEBCO API App is presented in Fig. 1. The GEBCO API is designed as a modular, service-oriented software system that combines a backend computational engine with a lightweight web-based frontend. The backend is implemented using the FastAPI framework in Python and provides a RESTful interface for accessing bathymetric data derived from the GEBCO 2024 dataset. The frontend consists of static HTML and JavaScript files that communicate with the backend via HTTP requests. The software can be deployed locally, within containerized environments, or as a cloud-hosted service.

The core architecture follows a client–server model. Computationally intensive operations, such as reading and subsetting the NetCDF dataset and performing spatial queries, are handled exclusively on the

Table 1
Comparison of the GEBCO API with existing bathymetric data access approaches.

Feature	GEBCO API	Direct NetCDF access	OPeNDAP/ERDDAP	GIS desktop tools
RESTful API access	✓	×	✓	×
Point-based queries	✓	×	✓	✓
Batch query support	✓	×	✓	Limited
On-demand regional subsets	✓	✓	✓	✓
No full dataset download required	✓	×	✓	×
Ease of automation (scripts, workflows)	High	Low	Medium	Low
Web-based interactive interface	✓	×	Limited	✓
Cloud-ready deployment	✓	×	Medium	×
Lightweight installation	✓	✓	Medium	Low
Open-source implementation	✓	✓	✓	Varies

server side, while the client side focuses on user interaction, visualization, and data download requests. This separation provides scalability, reproducibility, and efficient resource utilization.

3.2. Backend structure and packages

The backend is organized as a Python package with clearly separated responsibilities (Fig. 2). The main application entry point initializes the FastAPI instance, configures middleware, loads the GEBCO dataset, and registers all API endpoints. The GEBCO 2024 NetCDF file is loaded once at application startup using the `xarray` library, allowing subsequent queries to reuse the in-memory dataset and avoid repeated disk access.

Key Python libraries used in the backend include:

- **FastAPI**: for defining RESTful endpoints, request validation, and automatic API documentation.
- **Uvicorn**: as the ASGI server for running the application in both development and production environments.
- **xarray**: for efficient handling, slicing, and subsetting of multidimensional NetCDF data.
- **NumPy**: for numerical operations and grid generation.
- **Pandas**: for processing tabular data, particularly for CSV-based batch queries.
- **NetCDF4**: for exporting spatial subsets of bathymetric data in NetCDF format.

Cross-Origin Resource Sharing (CORS) is enabled through FastAPI middleware to allow the frontend and external clients to access the API without browser security restrictions.

3.3. Computational core and data access

At the core of the application lies a depth retrieval function that performs nearest-neighbor interpolation on the GEBCO dataset (see Fig. 3). Given a latitude and longitude pair, the function identifies the closest grid point in the GEBCO dataset and returns the corresponding elevation value. This approach provides fast response times and avoids the computational overhead associated with higher-order interpolation, which is often unnecessary for many engineering and exploratory applications.

Batch queries are supported by accepting a list of geographic coordinates in a single request. Each coordinate pair is processed independently using the same nearest-neighbor lookup procedure, and results are aggregated into a structured JSON response. This design allows efficient integration into automated workflows, where repeated bathymetric queries are required.

For area-based queries, the application generates a structured latitude–longitude grid based on user-defined spatial bounds and step size. Depth values are retrieved for each grid point and assembled into a two-dimensional array. The resulting data are encapsulated in an `xarray.Dataset` object, which preserves spatial coordinates, metadata, and dimensional consistency. The dataset is then written to a temporary NetCDF file and served to the user as a downloadable response.

3.4. API endpoints and functionality

According to Fig. 4, the GEBCO API exposes multiple endpoints to support different usage scenarios:

- **Single-point queries**: return the water depth at a specified latitude and longitude.
- **Batch queries**: accept a JSON array of coordinate pairs and return depth values for all requested locations in a single request.
- **CSV upload**: allow users to upload a file containing multiple coordinates and receive a processed CSV file augmented with depth information.
- **Area-based queries**: enable users to define a geographic bounding box and retrieve bathymetric data for the entire region in NetCDF format.

All endpoints include basic input validation to ensure that latitude and longitude values fall within valid geographic ranges. Errors are returned in a structured and machine-readable format.

3.5. Frontend interface

The frontend is implemented using standard HTML, CSS, and JavaScript, with no external framework dependencies (as seen in Fig. 5). It provides an interactive map-based interface that allows users to select points, visualize queried locations, and initiate downloads. Communication with the backend is handled via asynchronous HTTP requests using the Fetch API. The frontend is served as static content by the FastAPI application.

3.6. Deployment and reproducibility

To support reproducible deployment, the application includes a Docker configuration that encapsulates all software dependencies and runtime settings. This enables consistent behavior across different operating systems and computing environments. The application can be deployed locally for development, on institutional servers, or on cloud platforms such as Google Cloud Run.

The source code is managed using the `git` version control system and hosted in a public repository. Due to its large size, the GEBCO 2024 NetCDF file is not included in the repository and must be downloaded separately from the official GEBCO distribution site. Clear installation instructions are provided to ensure correct configuration and usage.

3.7. Extensibility

The modular design of the GEBCO API allows for straightforward extension. Additional interpolation methods, alternative bathymetric datasets, authentication mechanisms, or usage monitoring features can be integrated without major restructuring. This design choice makes the software suitable as both a standalone application and a reusable component within larger data-processing pipelines or decision-support systems.

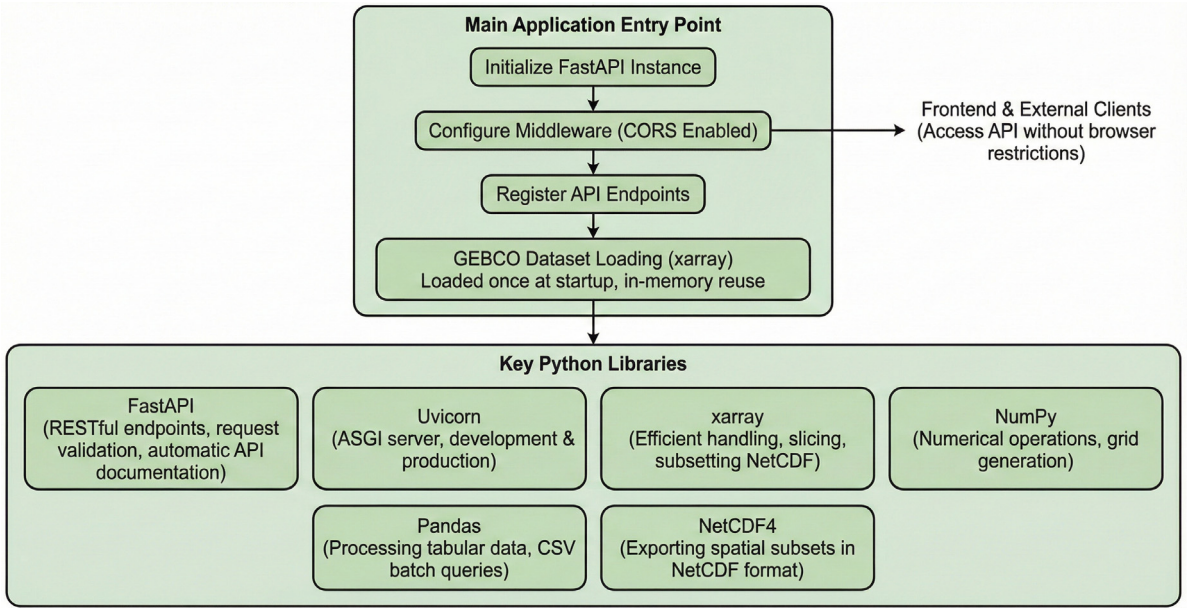

Fig. 2. The backend structure and packages of the GEBCO API App.

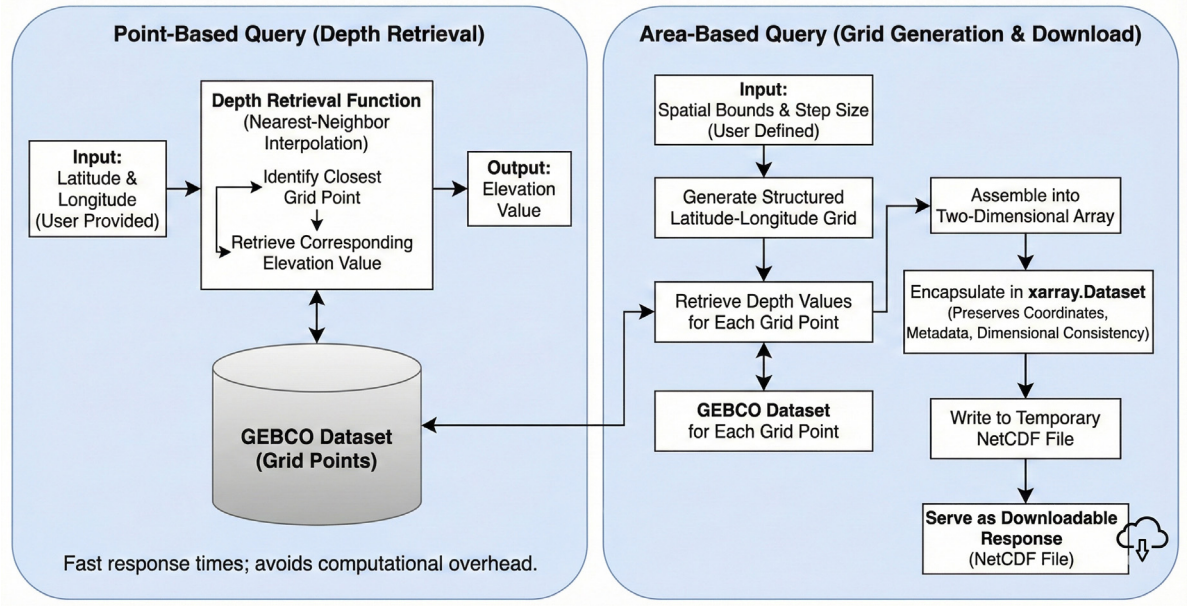

Fig. 3. The computational core and data access of the GEBCO API App.

4. Illustrative examples

As mentioned, the utility of the GEBCO API is demonstrated through two primary modes of interaction: a graphical user interface (GUI) for immediate visual feedback and a programmatic interface for developer integration. An explanatory video of the App environment is submitted as a supplementary file.

4.1. Interactive web interface

For researchers and students who require quick access to bathymetric data without writing code, the application provides a user-friendly web dashboard. As shown in Fig. 6, users can click on the map pr manually input latitude and longitude coordinates to retrieve depth values instantly. The interface includes an interactive map for visual context and a "History" sidebar that logs recent queries. A key feature of this interface is the ability to export these session logs directly as a CSV file, enabling users to build simple datasets interactively.

4.2. API documentation and testing

For developers and automated workflows, the application exposes a fully documented REST API. Fig. 7 illustrates the Swagger UI (automatically generated via FastAPI), which serves as both documentation and a testing sandbox. This interface lists all available endpoints, including:

- `/depth`: For single-point queries.
- `/batch_depths`: For processing lists of coordinates in a single request.
- `/upload_csv`: For bulk processing of external datasets.
- `/area_depths_netcdf`: For extracting custom NetCDF grids.

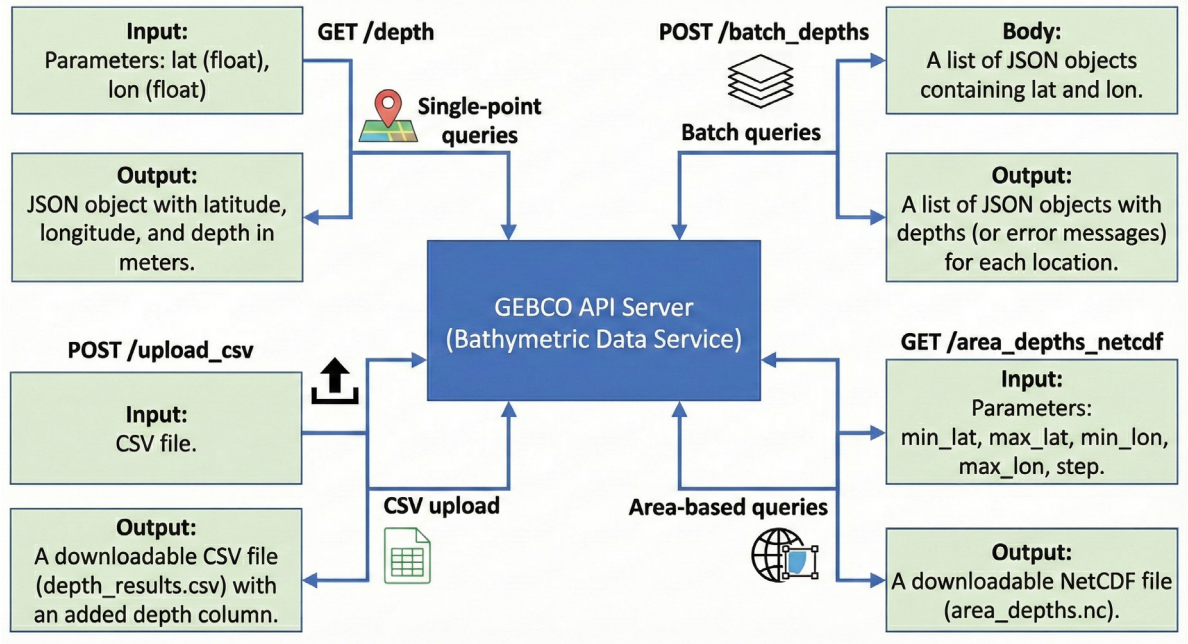

Fig. 4. The GEBCO API endpoints.

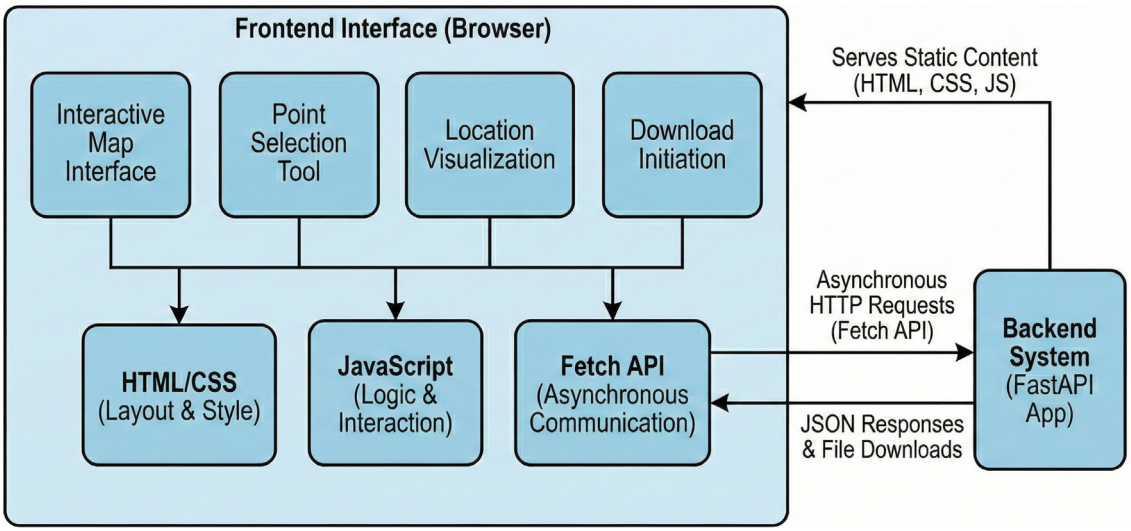

Fig. 5. The frontend structure of the GEBCO API App.

Users can execute requests directly from this page to verify the data structure before integrating the API into external tools like MATLAB or Python scripts.

4.3. Integration with scientific workflows

To facilitate seamless adoption by oceanographers and engineers, the GEBCO API supports standard HTTP protocols compatible with major scientific computing environments, including MATLAB, Python, and R. The App distribution includes a comprehensive set of MATLAB examples that demonstrate three core integration patterns:

1. **Single Point Retrieval:** Fetching depth data for individual coordinates using standard GET requests.
2. **Batch Processing:** Utilizing the `/batch_depths` endpoint to query multiple locations simultaneously, significantly reducing network overhead compared to iterative requests.
3. **Sub-grid Extraction:** Automating the download of NetCDF subsets for specific bounding boxes, allowing researchers to obtain manageable file sizes for regional analysis.

Listing 1 demonstrates how a researcher can retrieve batch elevation data programmatically using MATLAB's native web functions.

```
% Define the API endpoint and headers
apiUrl = 'https://gebcoapi-91108147194.us-central1.run.app/batch_depths';
options = weboptions('MediaType', 'application/json', ...
                     'HeaderFields', {'accept', 'application/json'; ...
                                      'Content-Type', 'application/json'});

% Define coordinates (Lat, Lon)
data = [
```

```
    struct('lat', 35.5, 'lon', 23.1), ...
    struct('lat', -12.3, 'lon', 45.8)
];

% Send POST request and display response
response = webwrite(apiUrl, jsonencode(data),
    options);
disp(response);
```

Listing 1 MATLAB script demonstrating batch depth retrieval via the REST API.

5. Impact overview

The GEBCO API is developed to facilitate efficient, programmatic access to global bathymetric data and to remove practical barriers associated with handling large-scale NetCDF datasets. By exposing the GEBCO dataset through a web-based interface and a RESTful API, the software enables rapid retrieval of water depth information for individual locations, large batches of coordinates, and user-defined geographic areas. This approach supports new research workflows in which bathymetric data can be dynamically queried and integrated into data-driven models, optimization frameworks, and real-time decision-support systems.

The availability of lightweight and automated access to bathymetry enables new research questions that rely on repeated spatial queries, ensemble-based analyses, and uncertainty-aware modeling. In particular, studies involving probabilistic assessment of marine systems, sensitivity analysis, and multi-objective optimization benefit from the ability to retrieve bathymetric information on demand without the overhead of manual data preprocessing. The software also improves the pursuit of existing research questions by significantly reducing setup time and computational overhead, allowing researchers to focus on model development and analysis rather than data handling.

In daily research and engineering practice, the GEBCO API shifts bathymetric data usage from file-centric workflows to service-oriented access. Users can directly integrate depth queries into numerical simulations, optimization algorithms, and scripting environments such as MATLAB, enabling seamless coupling with wave, wind, and operational datasets. This has proven particularly useful in marine renewable energy assessment, ship propulsion analysis, and weather-aware route optimization, where accurate bathymetry is a key environmental constraint.

The software has relevance both within and beyond its initially intended academic audience. While it was developed primarily for researchers and engineers in ocean and marine engineering disciplines, its open-source availability and browser-based interface make it accessible to students, educators, and developers in related fields such as geospatial analysis and marine informatics. Its cloud-based deployment further supports usage in distributed and collaborative research environments.

The GEBCO API directly supports and complements several peer-reviewed studies authored or co-authored by the developer, including work on wave energy converter geometry optimization [14], adaptive control of wave energy systems [15], and investigations into ship propulsion power, fuel consumption, and route optimization under uncertain marine conditions [16,17]. In these studies, efficient access to bathymetric data is essential for defining environmental constraints and improving the robustness and realism of numerical analyses. Beyond academic research, the software architecture and deployment model provide a foundation for future commercial applications in maritime route planning, offshore energy assessment, and digital marine services, although no dedicated spin-off has been established at the time of writing.

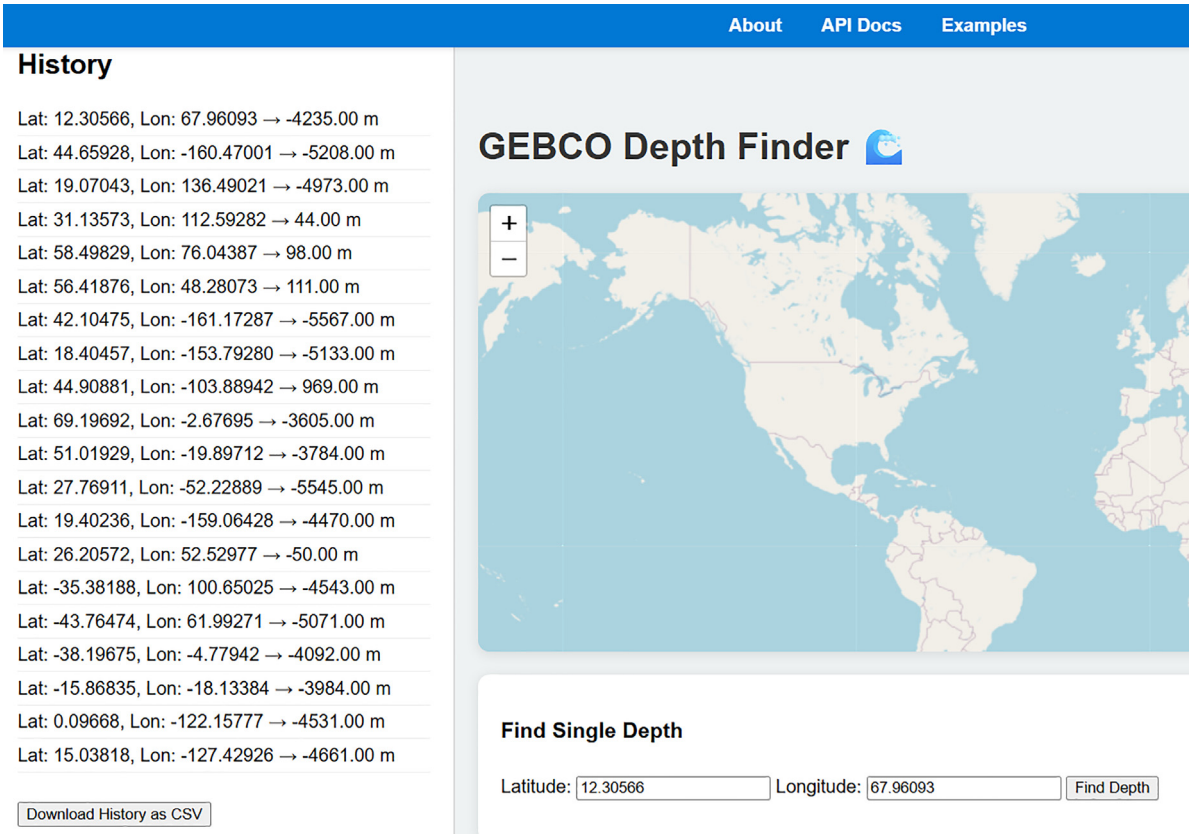

Fig. 6. The GEBCO API web interface.

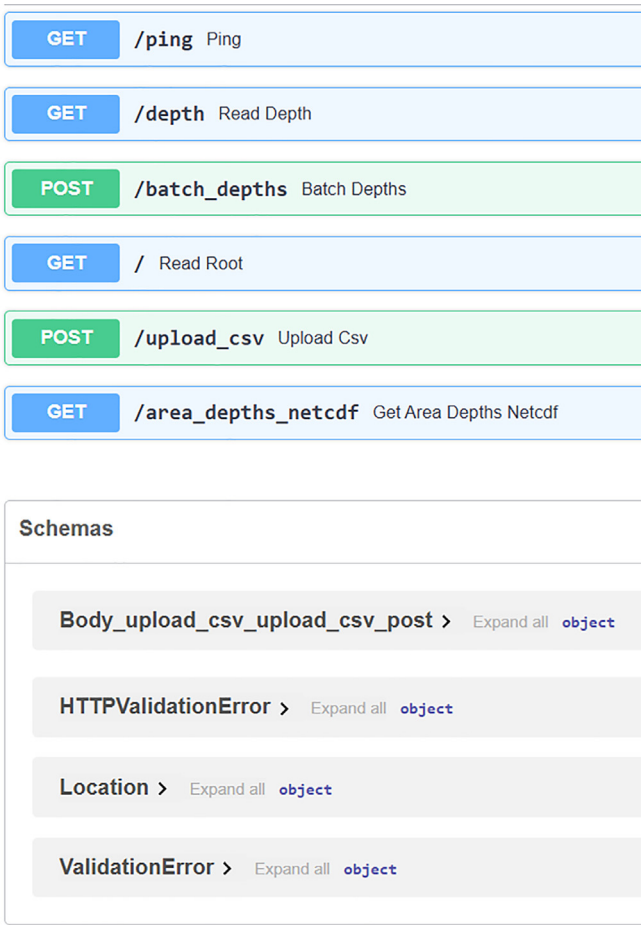

Fig. 7. The automated GEBCO API documentation interface.

6. Conclusions

This paper presented the GEBCO API, a web-based and RESTful software platform designed to provide efficient, programmatic access to global bathymetric data from the GEBCO 2024 dataset. By combining a lightweight HTML interface with a FastAPI-based backend, the software enables interactive exploration, automated querying, and seamless integration of bathymetric information into scientific and engineering workflows. The open-source design, modular architecture, and cloud-ready deployment make the tool accessible to a broad range of users, from academic researchers to developers of marine applications.

Despite its demonstrated utility, several limitations currently exist. The software relies on a locally stored GEBCO NetCDF file, which must be manually downloaded due to its large size, potentially limiting ease of deployment for some users. In addition, the current implementation focuses on depth extraction only and does not yet include derived bathymetric products such as slope, curvature, or uncertainty metrics.

Future development will address these limitations through support for tiled or cloud-optimized data formats, optional integration with object storage services, and query-level caching to improve scalability. Planned extensions include the computation of secondary bathymetric features, multi-resolution data access, and enhanced support for uncertainty-aware analyses. Additional API endpoints for coupling bathymetry with wave, wind, and oceanographic datasets are also envisaged.

CRediT authorship contribution statement

Kumars Mahmoodi: Resources, Software, Data curation, Investigation, Methodology, Visualization, Formal analysis, Validation, Conceptualization, Programming, Writing – original draft. **Jari Böling:** Methodology, Funding acquisition, Writing – review & editing.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used ChatGPT and Microsoft Copilot to refine the language and improve clarity. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Ethical approval

This paper is the author's original work and has not been previously published elsewhere.

Funding

This research was supported by Business Finland through the project INDECS – Integration of Design and Operation of Cruise Ship Energy Systems (Grant No. 7682/31/2022).

Declaration of competing interest

The authors declare that they have no conflicts of interest regarding the publication of this research article. The research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Acknowledgments

The authors wish to thank Business Finland for their financial support through the project INDECS.

Data availability

The study's supporting data can be accessed by contacting the corresponding author upon request.

References

[1] L. Zhang, J. Wen, Z. Huo, Z. Li, M. Xi, J. Yang, From land to ocean: bathymetric terrain reconstruction via conditional generative adversarial network, Earth Sci. Inform. 17 (5) (2024) 4361–4375, http://dx.doi.org/10.1007/s12145-024-01381-9.

[2] M.A. Mohammad, I. Jazireeyan, M. Pirooznia, Improving the modeling of bathymetry in the Persian gulf and the oman sea using data assimilation of geodetic observation data, Earth Sci. Inform. 18 (1) (2025) 16, http://dx.doi.org/10.1007/s12145-024-01502-4.

[3] K. Kabiri, M.H. Kazeminezhad, Depth estimation from multispectral satellite imagery: a comparison of conventional and machine learning-based approaches-case study: Kish island, Persian gulf, Earth Sci. Inform. 18 (1) (2025) 68, http://dx.doi.org/10.1007/s12145-024-01542-w.

[4] H. Mamidisetti, R. Vijay, Bathymetric estimation of mumbai coast using landsat OLI imagery and machine learning models, Earth Sci. Inform. 18 (1) (2025) 147, http://dx.doi.org/10.1007/s12145-024-01619-6.

[5] K. Mahmoodi, H.R. Fard, J. Böling, Long-term climate change effects on power performance of wave energy converters: A case study, Energy 326 (2025) 136101, http://dx.doi.org/10.1016/j.energy.2025.136101, URL https://www.sciencedirect.com/science/article/pii/S0360544225017438.

[6] K. Mahmoodi, H. Ghassemi, A. Razminia, Performance assessment of a two-body wave energy converter based on the Persian gulf wave climate, Renew. Energy 159 (2020) 519–537.

[7] K. Mahmoodi, H. Ghassemi, A. Razminia, Performance assessment of a two-body wave energy converter based on the Persian gulf wave climate, Renew. Energy 159 (2020) 519–537, http://dx.doi.org/10.1016/j.renene.2020.06.071, URL https://www.sciencedirect.com/science/article/pii/S0960148120309812.

[8] D. Ma, S. Zhou, Y. Han, W. Ma, H. Huang, Multi-objective ship weather routing method based on the improved NSGA-III algorithm, J. Ind. Inf. Integr. 38 (2024) 100570, http://dx.doi.org/10.1016/j.jii.2024.100570, URL https://www.sciencedirect.com/science/article/pii/S2452414X24000141.

[9] K. Mahmoodi, J. Böling, R. Vettor, Evaluating the influence of marine weather parameters uncertainties on the ship fuel consumption with Monte Carlo analysis, Ocean Eng. 341 (2025) 122531, http://dx.doi.org/10.1016/j.oceaneng.2025.122531, URL https://www.sciencedirect.com/science/article/pii/S0029801825022140.

[10] K. Mokhtar, L.F. Chuah, M.A. Abdullah, O. Oloruntobi, S.M.M. Ruslan, G. Albasher, A. Ali, M.S. Akhtar, Assessing coastal bathymetry and climate change impacts on coastal ecosystems using landsat 8 and sentinel-2 satellite imagery, Environ. Res. 239 (2023) 117314, http://dx.doi.org/10.1016/j.envres.2023.117314, URL https://www.sciencedirect.com/science/article/pii/S0013935123021187.

[11] GEBCO-Compilation-Group, GEBCO 2024 grid, 2024, Global terrain model for ocean and land, providing elevation data at a 15 arc-second interval grid. URL https://www.gebco.net/data_and_products/gridded_bathymetry_data/.

[12] OPeNDAP, Opendap: Open-source project for a network data access protocol, 2026, https://www.opendap.org/. (Accessed 2026).

[13] ERDDAP, ERDDAP: Environmental research division data access program, 2026, https://coastwatch.pfeg.noaa.gov/erddap/. (Accessed 2026).

[14] H.R. Fard, K. Mahmoodi, M. Vaghefi, Geometry optimization of a two-body heaving point absorber wave energy converter based on the long-term oman gulf wave climate, Renew. Energy 253 (2025) 123498, http://dx.doi.org/10.1016/j.renene.2025.123498, URL https://www.sciencedirect.com/science/article/pii/S0960148125011607.

[15] K. Mahmoodi, A. Razminia, J. Böling, Adaptive optimal disturbance rejection for wave energy converters, Energy Convers. Manag.: X 28 (2025) 101225, http://dx.doi.org/10.1016/j.ecmx.2025.101225, URL https://www.sciencedirect.com/science/article/pii/S2590174525003575.

[16] K. Mahmoodi, J. Böling, A. Razminia, R. Vettor, Investigating the effects of weather condition uncertainties on the required propulsion power of a cruise ship, Ships Offshore Struct. (2025) 1–21, http://dx.doi.org/10.1080/17445302.2025.2545909.

[17] K. Mahmoodi, J. Böling, R. Vettor, Ensemble-based ship weather multi-objective route optimization, J. Ind. Inf. Integr. 50 (2026) 101075, http://dx.doi.org/10.1016/j.jii.2026.101075, URL https://www.sciencedirect.com/science/article/pii/S2452414X26000166.