THE OPEN DATABASE OF REGIONAL MODELS OF THE INTERNATIONAL SERVICE FOR THE GEOID

J. F. Toro [1, *], D. Carrion [1], L. Rossi [1], M. Reguzzoni [1]

[1] Politecnico di Milano, Dept. of Civil and Environmental Engineering, Politecnico di Milano, Piazza Leonardo da Vinci, 32, 20133 Milan, Italy (juanfernando.toro, daniela.carrion, lorenzo.rossi, mirko.reguzzoni)@polimi.it

Commission V, WG V/4

KEY WORDS: ISG, Geoid, PostgreSQL, PostGIS, Django, OpenLayers, WebGIS

ABSTRACT:

This paper describes the development of a web application to facilitate the management and processing of geoid models in the ISG collection as well as of additional information available in the ISG website (mainly projects, services and publications). The ISG service pays significant attention to metadata and data interoperability from published geoid models. As a result, the possibility of creating a database with all necessary information for describing geoid models has been considered. PostgreSQL was chosen as the database for the implementation to take advantage of PostGIS geographic processing functions for developing a Database Management System for the resources available inside the ISG service catalogue (including geoid products, software, publications, etc.). The project's web development environment is Django, a web framework which is supported by the PostgreSQL database back-end and which makes it easier to integrate new applications or processing services into the website. A model solution for raster data integration using a PostGIS database back-end named Django-raster has also been investigated as part of the implementation, provided also that it is possible to take advantage of the vast Django community. The Django-raster model allows for the storage of datasets as tiles in the database, their exposure as a Tiled Map Services (TMS), and allows as well raster map algebra operations. TMS services will be utilized to share the geoid models and metadata included in the open access archive using a WebGIS, created with OpenLayers.

1. INTRODUCTION

Open accessibility to geoid models can encourage their use in multiple research fields such as geodynamics, geophysics, oceanography, and other climate-related studies. The International Service for the Geoid (ISG, https://www.isgeoid.polimi.it/), which is part of the International Association of Geodesy (IAG), has collected a large archive of local and regional geoid models worldwide from the Geodesy researchers' community (Reguzzoni et al., 2021).

Large attention is dedicated internationally to the open sharing of geodetic data with specific attention to metadata and documentation (Homburg et al., 2019; Izdebski et al., 2021). In particular, other services of IAG share details on how they store data and metadata, such as ICGEM (International Centre for Global Earth Models, Ince et al., 2019) or IGFS (The International Gravity Field Service, Vergos et al., 2019). This works presents the design and implementation of a geodatabase and a web application for the dissemination of the geoid models for ISG, exploiting Free and Open-Source Software (FOSS) for an effective storage, management and publication of the data. The Object-Relational Database System PostgreSQL (Group PostgreSQL Global Development, 2022) has been selected to serve as the database back-end, to allow the management of geospatial components with the PostGIS (*PostGIS 3.0.6dev Manual*, 2021) extension. For an effective integration of the ISG website and considering the web services which have already been implemented within the ISG website using a python environment, Django (Django Software Foundation, 2021) has been selected as the web-framework for generating the pages and to provide a web-dashboard to update and manage the website.

The database has been designed to be integrated as well with a WebGIS and with web-based applications for data extraction and statistics computation to enrich and improve the interoperability of the ISG Website (https://www.isgeoid.polimi.it/). Aligned with the interests of the ISG, this work aims at identifying the current matters and possible implementations for the management of data, metadata, computations from ISG web services as well as additional information, by means of a database management system.

2. GEOID OPEN DATA ACCESS

2.1 Local geoid models

Geoid models provide information on the geoid undulation, i.e. the separation between the ellipsoidal height, corresponding to a specific datum (e.g. WGS84), and the orthometric height, which is referred to the mean sea level in ideal conditions (which can be materialised with tide gauges) and is physically related to the Earth gravity field (Heiskanen and Moritz, 1967). As it is well known, the ellipsoidal height is provided by GPS (Global Positioning System) based tools, however, for engineering design it is necessary to refer to the orthometric heights in order to take into account the physical behaviour of the objects on the Earth (e.g. to know the direction of water flow). The availability of the geoid undulation allows to measure ellipsoidal heights with a GPS and to compute from it the orthometric height. This represents one of the main applications of the use of geoid models, which can be much wider, including applications in geophysics and oceanography, e.g. to estimate the behaviour of geostrophic currents. Geoid models can be estimated at global level, starting from global databases of gravity measures, with limited resolution ranging approximately from 100 km to 10 km

* Corresponding author

at medium latitude (Heiskanen and Moritz, 1967; Ince et al., 2019). There is a specific service of IAG, ICGEM (Ince et al., 2019), which is devoted to collection, storage and sharing of global geoid models. To increase the level of detail of the knowledge of the geoid undulation, local geoid models are computed, based on the availability of local databases of gravity data, usually coupled with GPS/levelling data (networks of points measured with GPS as well as with spirit levelling, to obtain both ellipsoidal and orthometric height). ISG service of IAG is devoted to the collection and sharing of local geoid models, providing support to researchers and engineers who need access to detailed information about the geoid undulation.

2.2 ISG geoid collection

The ISG archive currently stores 255 geoid models. The model acquisition is continuous work in progress: data are collected after the spontaneous offer by the authors of the computation of the local solution, or after directly contacting the authors to offer them the possibility to store their data at ISG. The geoid models can be archived at ISG with different policies: they can be publicly available, in this case they can be directly downloaded from the website (https://www.isgeoid.polimi.it/), or they can be available upon a request which is addressed to the geoid author, or they can be private, so not accessible to users. Since 2020, the geoid models are being progressively provided with a DOI for permanent and unique indexing and reference and to allow also the direct citation of the products (Reguzzoni et al., 2021). Geoid models indexed with DOIs are hosted at GFZ (German Research Centre for Geosciences https://www.gfz-potsdam.de/en/home/), with which a cooperation agreement has been established. At the time of the production of this paper, more than 45 models have already been assigned with a DOI. The archived data are provided as well with a Web of Science ID, Clarivate's Data Citation Index™ (https://clarivate.com/webofsciencegroup/solutions/webofscience-data-citation-index/).

Through the ISG website, the metadata of the geoid models are exposed: Figure 1 shows an example.

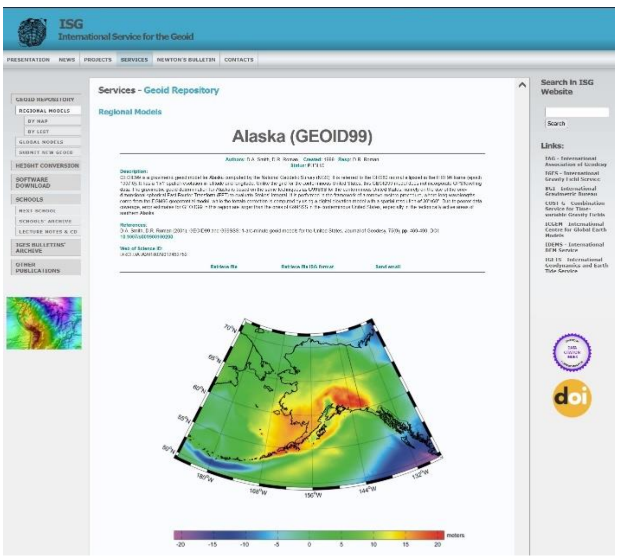

Figure 1. Overview of the products metadata for Alaska

Metadata include: the geoid name, its code, the list of authors, the type of access (public, on-demand, or private), a short description of the model characteristics (such as the area of interest, the resolution, the technique used for its computation,…), a reference paper, if available, the DOI, if available, the Web of Science ID, the links to access the data, if available, and the contact of the author and a static figure showing a map of the local geoid.

The models are stored in the ISG format (with extension .isg, Reguzzoni et al., 2021), as well as in the original format provided by the authors. The ISG format, coded with ASCII characters, includes a header to archive metadata.

2.3 ISG website

The ISG website is devoted not only to the exposure and sharing of geoid models, but also to show pieces of information related to ISG activities and publications. In the following, the main sections of the website are listed, because the database structure discussed in this paper includes those components as well.

The main goal of ISG is to collect and share local geoid models, therefore, one section is dedicated to show the list and map of available models, as well as thier metadata, as introduced in the previous section.

A website section is dedicated to projects in which ISG is involved or which share their results through ISG, like the Colorado experiment (Wang et al., 2021).

ISG has an educational role, with respect to geoid computation. Since 1994, Geoid Schools have been organised to teach the theory and the processing chain, with different techniques, to obtain geoid undulation. Pieces of information regarding the Schools, past and future, are shared through the website.

In addition, open software related to geoid computation as well as ISG publications are shared.

Recently, a height conversion tool has been implemented to allow to obtain ellipsoidal heights from orthometric ones or vice versa on given coordinates, after interpolation of the geoid models available at ISG.

Currently, the ISG front-end components are based on HTML and JavaScript files. Data are stored as single files and are accessed directly from the HTML pages.

The height conversion web service is built as a back-end component, which allows the front-end user to execute computations from the server side. For this purpose, Django, a high-level Python web framework was used for the development of a REST API (Representational State Transfer Application Programming Interface), enabling calculation on a python environment. All requests, from front-end to back-end, are submitted using the HTTP POST method, and send a response as a JSON file with the processing outputs. Considering the current developments from the web services, this work proposes the use of the existing Django application to create the needed models, views, and templates, to manage the website from a dashboard interacting with a database.

3. ISG WEBSITE DESIGN

This section introduces the design for updating the ISG website, presenting the structure and technologies focused on facilitating the management and update of the website contents, as well as the storage of the resources archived within the website, including the ISG's catalogue of geoid models. The technologies allow the integration of the existing (and under development) processing services, taking into account potential developments regarding the implementation of new processing services, as well as the incorporation of a WebGIS for the exposure of the geoid models.

3.1 Database design

The database design considers both the georeferenced data corresponding to the geoid models as well as other information and services available on the ISG website. Here, the design of the Database Managements System (DBMS) to support the ISG website is presented.

On the one hand, we have the storage of data regarding the templates designed for the website, each of these templates are devoted to build its corresponding component page within the website structure presented in Figure 2. In this case, the different pieces of information to be found inside of the website will be stored as independent tables in the database matching its component (for example, section name, description, products, files, etc.).

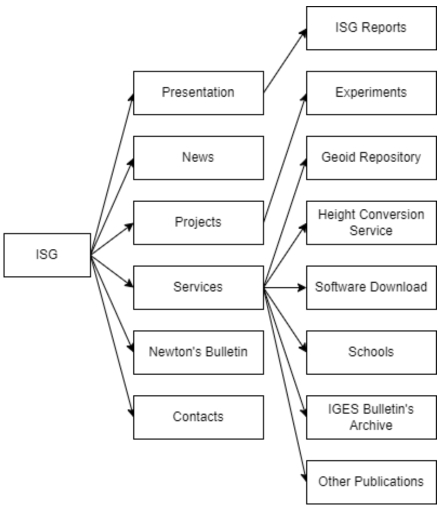

Figure 2. Website structure of the ISG.

There are two main benefits of this implementation within the database. The first, is the possibility of using the data stored to build the website taking advantage of a web framework, and possibly using an interactive dashboard for the update of its components, avoiding the need of editing HTML static pages. The database will allow to map all resources (including geoid models, metadata, archived software and archived documents) inside the file system, with direct access to their path.

On the other hand, the database design highly concerns about the efficient management of the geoid models. For this reason, this work presents the use of a database with extended capabilities on the support of spatial data types to store both geoid models and metadata. In addition, considering that the geoid products are made available through different formats, all of them will be accessible through the database inside the file system. The task of mapping the paths into the file system is relevant for the possibility of keeping track of versioning of the different products and available formats.

The management of the spatial data is possible thanks to the Object-Relational Database Management System (ODBMS) of PostgreSQL (PostgreSQL License)(Group PostgreSQL Global Development, 2022) using the PostGIS (General Public license) (*PostGIS 3.0.6dev Manual*, 2021) spatial extension. PostgreSQL and PostGIS, are an open source spatial database implementation which has been exploited in different earth science applications (Swain et al., 2015). The spatial extension adds spatial data types to the ODBMS such as raster, geometry, and geography, along with functions for the processing to the GIS objects and coordinates transformation. Also, it provides extensive implementation of OGC standards (Swain et al., 2016).

PostGIS will enable the input of a geoid product provided with the following data and ancillary data into the database:

- Name of the model
- Year of computation
- Type of the model: gravimetric, geometric, or hybrid
- Classification: geoid and quasi-geoid
- Data representation: sparse or gridded, and if gridded the ordering of the data
- Reference ellipsoid and datum, the reference frame, and the tidal system
- Coordinate Reference System
- Units of the undulation data and of the coordinates.
- Geoid Product
- Version

Profiting from the use of PostgreSQL and PostGIS, it is possible to store the georeferenced product using the raster data type implemented within PostGIS and enable the development of new processing services using the raster functions available for that data type.

3.2 Web framework implementation

In order to exploit the use of a database for the management of the contents and resources made available by ISG, a Django back-end implementation has been chosen to build the website.

Django is a popular, Python-based, web framework that simplifies the web applications development through a structured software architectural pattern (Django Software Foundation, 2021). The three components of a Django app can be described as: Model, to manage the components for the application; Views; to structure and style the user interface and web client pages; Controllers, to setup the logic and behaviour of the application. In this case, the ingestion of the contents of the website and different resources is achieved by the implementation of different models. Following the website structure presented in Figure 2, it is considered to implement different models for each website section and subsection. Then, each of the dedicated database tables, which have been produced for storing the information for each section of the website, will be used for rendering the templates design for recreating the website.

Django covers many of the web framework best practices such as, Internationalization, Forms Validations, AJAX support, and Object Relational Mapping, HTML5 Support, between many others (Salas-Zárate et al., 2015), which makes it a robust tool for web development. That’s why it is broadly used in geospatial data management applications (Baranovskiy & Zharikova, 2014; Evans et al., 2020; Qiao et al., 2019; Tarboton et al., 2014).

Figure 3 shows the architecture of the designed website, which represents an update with respect to current static HTML version: the Django framework allows integrating the website contents and the geoid model repository as a part of the DBMS. Currently, the website is built upon static HTML pages, and the updating of the site is time a demanding task. The implementation of Django will allow to manage the different components through a web-

https://doi.org/10.5194/isprs-archives-XLIII-B5-2022-29-2022

client application using the dedicated database of the ISG service. Figure 4 shows an example for the input form of the web-client application that will allow the ingestion of data into the database.

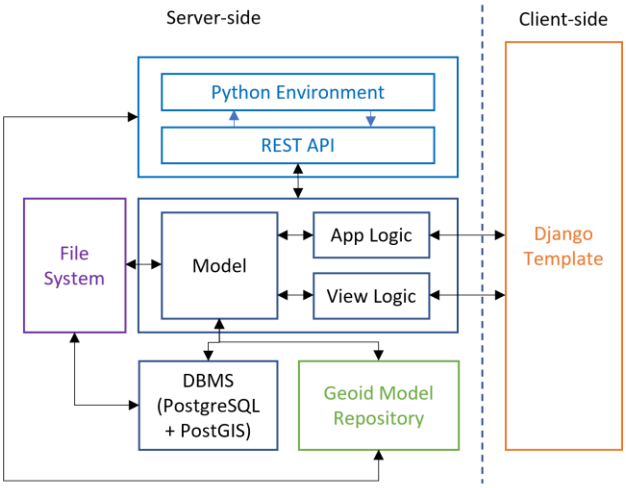

Figure 3. Website designed architecture

The implementation benefits from the Django-Raster library. This library extends the capabilities of the objects handled by Django with a model that enables raster data functionalities for projects taking advantage of the PostGIS database back-end. The model uses Django's raster data type RasterField and GDAL binding through GDALRaster (Wiesmann & Flaxman, 2021). The model permits the ingestion of raster data, for GDAL supported raster data formats into the database using the client-based dashboard as seen in Figure 4. The raster model not only allows the storage of the datasets as tiles into the database, but also the use of Tiled Map Services (TMS) and raster map algebra operations. One of the benefits from the storage of the raster datasets into the database is the possibility to exploit the raster functions implemented by PostGIS. In addition, the creation of the TMS for each of the raster products ingested into the database allows to visualize the products as georeferenced images using several web mapping libraries, such as Leaflet or OpenLayers, to integrate WebGIS components into the website. One limitation of the Django raster module is that it focuses on single band rasters, which is the case of geoid models, thus this is not posing issues for our case. The raster tiles projection is based on the Web Mercator (EPSG:3857), that is the projection commonly used for web mapping.

The interactive visualization of the geoid models is based on the JavaScript web-mapping library of OpenLayers (2-Clause BSD License). The WebGIS will make use of the TMS served by the PostGIS back-end implementation for the visualization of the georeferenced images of the geoids. The implementation aims at providing the users the possibility to explore the set of products made available openly by the ISG. OpenLayers permits rendering maps that allow users to pan, zoom, and select features from a web page. Following the creation and parsing of a RasterLayer, the tiles can be accessed by means of a url using the following structure:

/raster/tiles/layer_id/{z}/{x}/{y}.png

Where, the layer_id corresponds to the primary key for the raster layer. Such structure permits the use of different web-mapping applications, like OpenLayers or Leaflet.

Furthermore, the Django-raster model has been customised considering that for ISG it is relevant the provision of the data through standardized data formats compliant to the processing of geoid products (e.g. ISG data format – .isg), which is why a single geoid product can be uploaded using different raster data formats enabling the storage within the database and/or file system.

In the following set of figures, it is possible to explore the custom Django model for the ingestion of a geoid product into the implemented database. Figures 4-6 present the Django dashboard for the customized raster model implementation to upload the geoid products metadata and raster datasets into the DBMS and the Geoid Model Repository.

Figure 4 presents the form in which the metadata for a specific geoid product, namely the one presented in section 3.1 (Database Design), requesting information such as the Geoid's name, an abstract describing the type of the product and computation method, the datatype (if provided as sparse or grided data), the corresponding raster file, a source URL (or reference site to the product, URL or DOI), the coordinate reference system of the product, reference ellipsoid and datum, the reference frame, and the tidal system, coordinate Reference System, units of the undulation data and of the coordinates. The versioning of the product will be automatically managed by the database accounting for the existence of a product with the same name of the geoid. The next step regards the partitioning of the geoid model into the database as tiles, as seen in Figure 5, the dashboard will present a log with the status of the partitioning as well as the number of tiles which have been produced for the ingested products. Once the raster layer tiling is complete, the "Finished parsing" message status will be shown and it will be possible to observe the subsequent piece of information about the layer as presented in Figure 6. The last section for a raster layer upload will allow to verify if the layer has been correctly ingested into the database and to check the provided additional metadata about the layer concerning the coordinate reference system, number of bands and additional georeferencing data of the image, metadata about the contents of the layer including statistics and a sample of values in the layer, and at last the location of the file within the file system (in the Geoid Model Repository). Finally, when confirming that all the data have been filled and correctly read, it is possible to proceed to save the geoid model.

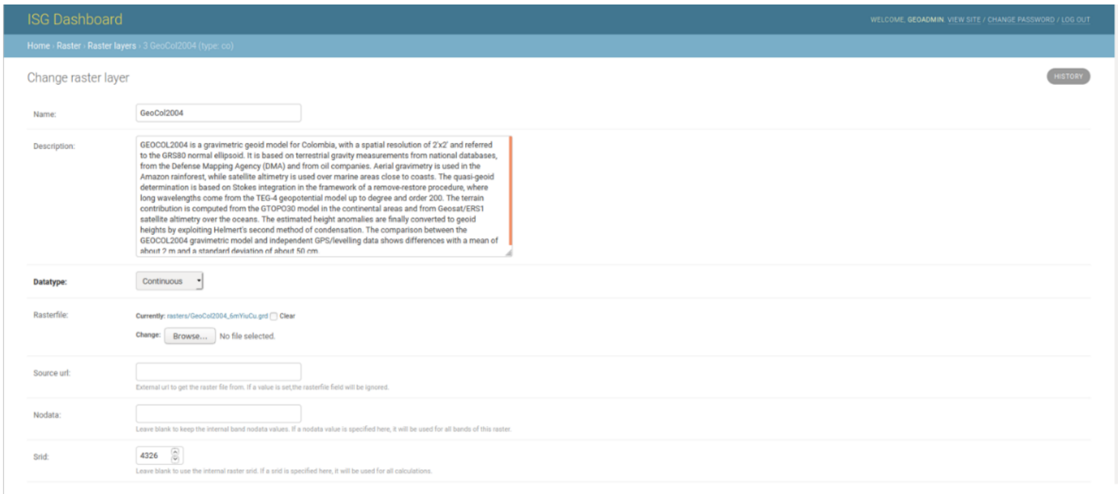

Figure 4. Geoid Model – Product Metadata – Add product (1/3)

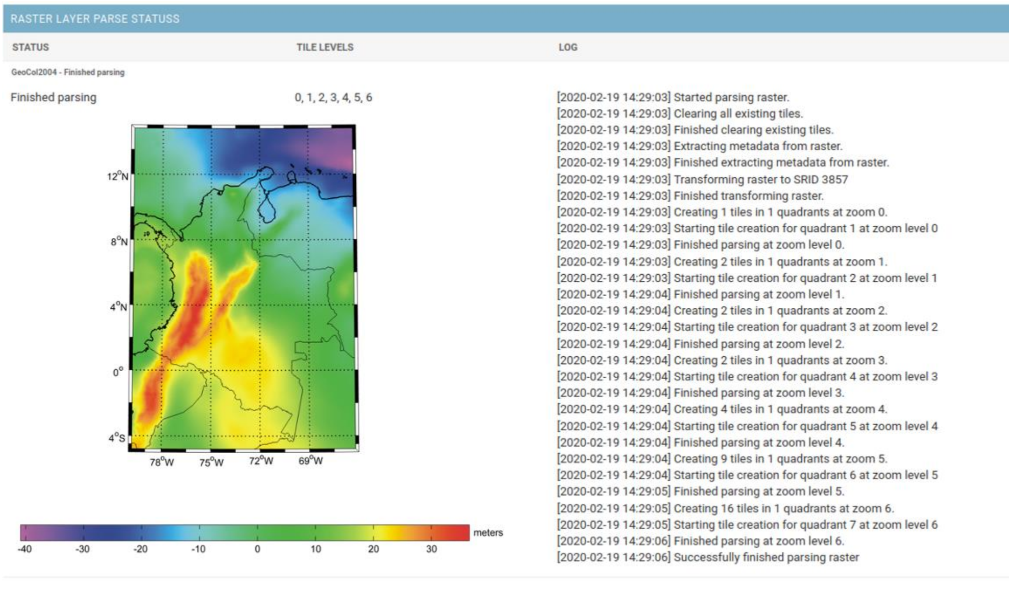

Figure 5. Geoid Model – Model tiling – Add product (2/3)

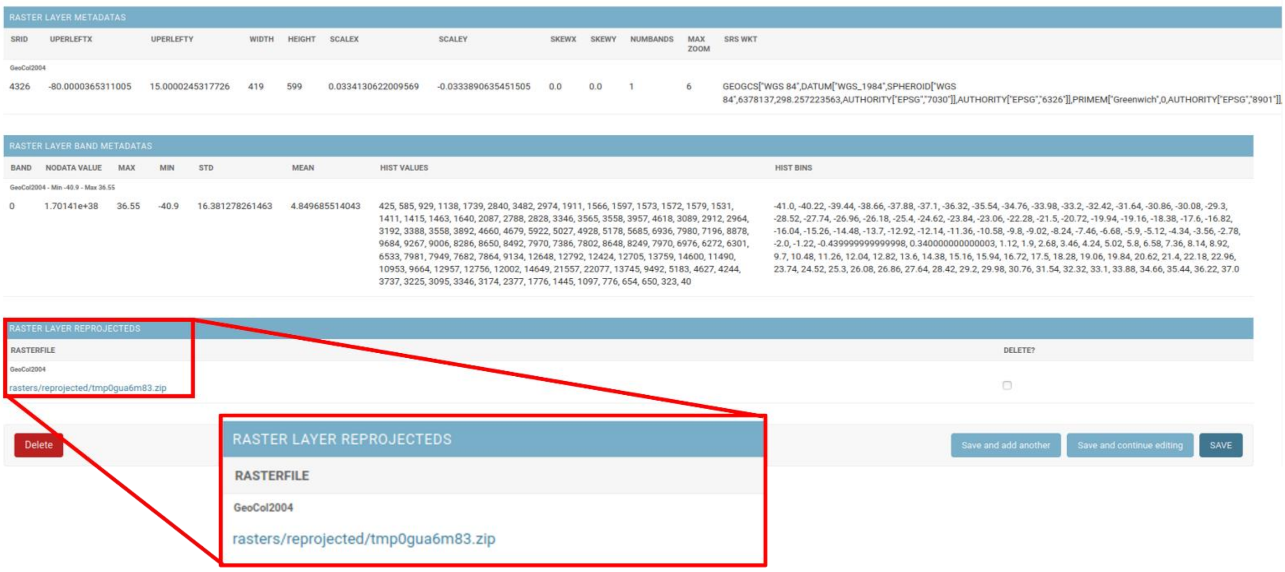

Figure 6. Geoid Model – Save product into DBMS – Add product (3/3)

4. DISCUSSION

This work presents the implementation of a web application to ease the management and processing of geoid models available within the ISG collection as well as the data shared through the ISG website. The metadata and data interoperability from published geoid models are given special consideration by the ISG service. As a result, the creation of a database containing the necessary data for describing geoid models has been explored. PostgreSQL was chosen as the database for the implementation to take advantage of PostGIS geographic processing functions for designing a Database Management System for the resources available within the ISG service catalogue (including geoid products, software, publications, etc.). The PostgreSQL database back-end supports the Django web framework. The use of Django as a development environment for the project facilitates the integration of additional applications, or processing services into the website. Furthermore, because of the large Django community, a model implementation for raster data integration using a PostGIS database back-end called Django-raster has been considered within the implementation. The raster model allows not only for the storage of datasets as tiles in the database, but also for the creation of Tiled Map Services (TMS) and raster map algebra operations for online maps. WebGIS, designed with OpenLayers, will be used to visualize the many items contained in the open access archive using TMS services.

While partitioning the datasets can be a solution for the management of large raster datasets, the implementation efficiency can vary depending on the partitioning methods. This techniques can vary, and in case of PostGIS Raster, there is the possibility to manage 2D tiles x/y and sets as default a size for the tiling of 100 x 100 pixels (Baumann et al., 2021). It can be considered the use of a multi-dimensional array solution for the management of raster data, especially when the possibility of presenting the data as an n-Dimensional array (accounting for multiple versions of the geoids) is relevant for the community. For this, further studies can explore the use of full-stack Array Databases implemented from scratch (e.g., rasdaman, SciDB) for managing data as n-Dimensional arrays, which implement OGC standards.

REFERENCES

Baranovskiy, N., & Zharikova, M. (2014). A web-oriented geoinformation system application for forest fire danger prediction in typical forests of the Ukraine. *Lecture Notes in Geoinformation and Cartography*, *0*(199669), 13–22. Scopus. https://doi.org/10.1007/978-3-319-08180-9_2

Baumann, P., Misev, D., Merticariu, V., & Huu, B. P. (2021). Array databases: Concepts, standards, implementations. *Journal of Big Data*, *8*(1), 28. https://doi.org/10.1186/s40537-020-00399-2

Django Software Foundation. (2021). *The web framework for perfectionists with deadlines | Django*. https://www.djangoproject.com/

Evans, S. W., Jones, N. L., Williams, G. P., Ames, D. P., & Nelson, E. J. (2020). Groundwater Level Mapping Tool: An open source web application for assessing groundwater sustainability. *Environmental Modelling & Software*, *131*, 104782. https://doi.org/10.1016/j.envsoft.2020.104782

Gkatzoflias, D., Mellios, G., & Samaras, Z. (2013). Development of a web GIS application for emissions inventory spatial allocation based on open source software tools. *Computers and Geosciences*, *52*, 21–33. Scopus. https://doi.org/10.1016/j.cageo.2012.10.011

Group PostgreSQL Global Development. (2022, marzo 30). *PostgreSQL*. PostgreSQL. https://www.postgresql.org/

Heiskanen,W. A. and Moritz, H.: *Physical Geodesy*, W.H. Freeman and Co., USA, 364 pp., 1967.

Homburg, T., Steppan, S., & Würriehausen, F. (2019). Semantic Data Integration and Quality Assurance of Thematic Maps in the German Federal Agency for Cartography and Geodesy. In W. Abramowicz & R. Corchuelo (A c. Di), *Business Information Systems Workshops* (pagg. 543–555). Springer International Publishing. https://doi.org/10.1007/978-3-030-36691-9_46

Ince, E. S., Barthelmes, F., Reißland, S., Elger, K., Förste, C., Flechtner, F., & Schuh, H. (2019). ICGEM – 15 years of successful collection and distribution of global gravitational models, associated services, and future plans. *Earth System Science Data*, *11*(2), 647–674. https://doi.org/10.5194/essd-11-647-2019

Izdebski, W., Zwirowicz-Rutkowska, A., & Nowak da Costa, J. (2021). Open data in spatial data infrastructure: The practices and experiences of Poland. *International Journal of Digital Earth*, *14*(11), 1547–1560. https://doi.org/10.1080/17538947.2021.1952323

PostGIS. *PostGIS — Spatial and Geographic Objects for PostgreSQL*. Recuperato 30 marzo 2022, da https://postgis.net/
PostGIS 3.0.6dev Manual. (2021). 871.

Qiao, X., Li, Z., Ames, D. P., Nelson, E. J., & Swain, N. R. (2019). Simplifying the deployment of OGC web processing services (WPS) for environmental modelling – Introducing Tethys WPS Server. *Environmental Modelling & Software*, *115*, 38–50. https://doi.org/10.1016/j.envsoft.2019.01.021

Reguzzoni, M., Carrion, D., De Gaetani, C. I., Albertella, A., Rossi, L., Sona, G., Batsukh, K., Toro Herrera, J. F., Elger, K., Barzaghi, R., & Sansó, F. (2021). Open access to regional geoid models: The International Service for the Geoid. *Earth System Science Data*, *13*(4), 1653–1666. https://doi.org/10.5194/essd-13-1653-2021

Salas-Zárate, M. D. P., Alor-Hernández, G., Valencia-García, R., Rodríguez-Mazahua, L., Rodríguez-González, A., & López Cuadrado, J. L. (2015). Analyzing best practices on Web development frameworks: The lift approach. *Science of Computer Programming*, *102*, 1–19. Scopus. https://doi.org/10.1016/j.scico.2014.12.004

Swain, N. R., Christensen, S. D., Snow, A. D., Dolder, H., Espinoza-Dávalos, G., Goharian, E., Jones, N. L., Nelson, E. J., Ames, D. P., & Burian, S. J. (2016). A new open source platform for lowering the barrier for environmental web app development. *Environmental Modelling & Software*, *85*, 11–26. https://doi.org/10.1016/j.envsoft.2016.08.003

Swain, N. R., Latu, K., Christensen, S. D., Jones, N. L., Nelson, E. J., Ames, D. P., & Williams, G. P. (2015). A review of open source software solutions for developing water resources web applications. *Environmental Modelling & Software*, *67*, 108–117. https://doi.org/10.1016/j.envsoft.2015.01.014

Tarboton, D. G., Idaszak, R., Horsburgh, J. S., Heard, J., Ames, D., Goodall, J. L., Band, L., Merwade, V., Couch, A., Arrigo, J., Hooper, R., Valentine, D., & Maidment, D. (2014). *Hydro share: Advancing collaboration through hydrologic data and model sharing*. *1*, 23–29. Scopus.

Vergos, G. S., Grigoriadis, V. N., Barzaghi, R., & Carrion, D. (2019). IGFS Metadata for Gravity. Structure, Build-up and Application Module. In J. T. Freymueller & L. Sánchez (A c. Di), *International Symposium on Advancing Geodesy in a Changing World* (pagg. 165–171). Springer International Publishing. https://doi.org/10.1007/1345_2018_38

Wang, Y.M., Sánchez, L., Ågren, J. *et al.* Colorado geoid computation experiment: overview and summary. *J Geod* **95,** 127 (2021). https://doi.org/10.1007/s00190-021-01567-9

Wiesmann, D., & Flaxman, M. (2021). *Django-raster Documentation*. https://django-raster.readthedocs.io/_/downloads/en/stable/pdf/