Spatial data interpolation in the AgDataBox Platform using graphics processing unit parallelism

Vitor Henrique Wagner Malacarne[1], Ricardo Sobjak[1], Claudio Leones Bazzi[1], Eduardo Godoy de Souza[2], Kelyn Schenatto[1], Marlon Rodrigues[1]

[1]Federal University of Technology - Paraná (UTFPR), Brazil
[2]Western Paraná State University (UNIOESTE), Brazil

A paper from the Proceedings of the
17th International Conference on Precision Agriculture and 11th Brazilian Congress on Precision and Digital Agriculture
13-15 July 2026
Porto Alegre, Brazil

Abstract.

The generation of thematic maps in precision agriculture depends on the appropriate selection of spatial interpolation methods and their parameter settings. On the AgDataBox platform, parameter selection for inverse distance weighting (IDW) and ordinary kriging (OK) is performed by routines that evaluate several candidate configurations through cross-validation and statistical criteria. However, this process may become computationally costly as the number of candidate combinations or sampling points increases. This paper presents a GPU-enabled version of the parameter-selection routines used in AgDataBox, implemented in Python and exposed via an application programming interface (API), while preserving the option to execute on a central processing unit (CPU) when graphics processing unit (GPU) resources are unavailable. We evaluated the implementation's compatibility in two case studies. First, 15 georeferenced datasets from three experimental fields in western Parana, Brazil, were processed using the original AgDataBox routine in R, the Python routine using GPU resources, and the Python CPU fallback. The selected parameters were then used to generate thematic maps in AgDataBox-Map with identical boundaries, resolution, class intervals, and color scales. The relative deviation coefficient (CDR) was used to compare the maps generated by the Python routines with the AgDataBox reference maps. Most comparisons yielded CDR values rounded to 0.00%, and the maximum CDR was 0.01%, indicating high compatibility among the generated maps. A second case study used dense datasets of altitude and apparent soil electrical conductivity, with up to 3507 and 3211 sampling points, respectively. The GPU routine remained functional in all evaluated scenarios. Sequential CDR values ranged from 0.06% to 0.15% for altitude and from 2.08% to 12.34% for conductivity, with the larger deviations attributed mainly to the representativeness of reduced subsets. The results indicate that the proposed implementation reproduced the behavior of the original routine and provides a basis for integrating GPU-supported interpolation-parameter selection into a more flexible AgDataBox architecture.

Keywords. *precision agriculture; thematic maps; parallel computing.*

Introduction

Precision agriculture (PA) has been established as an important approach for modernizing agricultural activities, providing tools to understand and manage the spatial variability of soil attributes and crop production (Bazzi et al., 2019). In this context, thematic maps play a fundamental role, as they spatially represent the distribution of these attributes in a cultivated field and make visible patterns that would be difficult to identify solely from sample data. These representations contribute to the definition of management zones, the localized use of inputs, and strategic decision-making in the field (Sobjak et al., 2023).

To create thematic maps in PA, data from different sampling approaches can be used, including sparse sampling grids (regular or irregular), data collected along field trajectories (e.g., by sensors attached to equipment), and data organized in regular grids, such as raster images. When values are unavailable for certain regions of the field, it is necessary to estimate them to produce a continuous surface of the analyzed attribute (Dall'Agnol et al., 2020). This process is carried out using spatial interpolation methods, among which Inverse Distance Weighting (IDW) and Ordinary Kriging (OK) stand out and are widely applied in PA. However, the quality of the generated maps depends not only on the choice of the interpolation method but also on the proper definition of its parameters, such as the power and the number of neighbors in IDW, and the theoretical semivariogram model in OK (Betzek et al., 2019).

Thus, different combinations of methods and parameters can produce maps with distinct characteristics. Computational routines were proposed by Betzek et al. (2019) to automatically evaluate different interpolation configurations, identifying the best parameters for the IDW and OK methods through cross-validation and statistical selection indexes. Subsequently, this process was improved by Sobjak et al. (2023) by incorporating new evaluation criteria for selecting geostatistical models, accounting for effective spatial dependence and the adequacy of the fitted model to the experimental semivariogram. We incorporated these routines into the AgDataBox platform, where the automatic selection process of the interpolator and its parameters is utilized as part of the thematic map generation workflow.

However, since the automatic selection process evaluates multiple combinations of methods, models, and parameters, which can generate hundreds of candidates, and then analyzes each combination using leave-one-out cross-validation to calculate statistical metrics, this algorithm can become computationally expensive as the number of combinations or samples increases.

Given this context, the feasibility of developing a hybrid execution routine capable of parallelizing parameter-combination evaluations while maintaining compatibility with the AgDataBox platform was investigated. Consequently, a Python-based version supporting GPU execution and CPU fallback was developed for the IDW and OK parameter selection routines. The implementation focuses on reproducing the existing workflow, statistical metrics, and reference comparisons, thereby ensuring conceptual and practical compatibility with the current system while leveraging GPU parallelism to enhance computational efficiency.

Materials and Methods

Implementation of the Python routines

We implemented the routines in Python in a Linux Ubuntu 22.04 LTS development environment. NumPy and Pandas were used for numerical processing and data manipulation; SciPy for numerical optimization and specialized mathematical functions required by the Matern model; and CuPy for GPU-accelerated array operations. The routines were exposed via a web API built with FastAPI, with input and output data represented in JSON and validated using Pydantic models. The Uvicorn server executed the application.

The API organizes the route layer separately from the mathematical processing layer. The interpolation service receives samples with coordinates x and y and values, as well as execution parameters such as variogram models, fitting methods, lag configuration, filters, and IDW settings. The interpolator selection endpoint triggers the complete selection workflow. It returns the candidate configurations ordered by the Interpolator Selection Index (ISI), with the first item corresponding to the recommended interpolator and parameter set.

The system divides the selection process into two pipelines: one deterministic and one geostatistical. In both cases, the workflow begins by validating the sample structure, converting the coordinate and attribute arrays to the selected execution backend, and computing a distance matrix among the sampling points. This matrix was reused in the subsequent stages to avoid recalculating pairwise distances for each candidate configuration. When GPU execution is enabled, the system allocates arrays in device memory and expresses operations as vectorized array computations; when the GPU is not available, it uses equivalent NumPy operations as the CPU fallback.

For the IDW routine, the algorithm assembled a grid of candidate points formed by combinations of power exponents and numbers of nearest neighbors. For each candidate, the system performed leave-one-out validation by temporarily omitting the point being predicted and estimating it from the remaining samples selected as nearest neighbors. The same sorted-distance structure was reused across candidates, reducing the need for repeated sorting operations. On the GPU, the weights, predictions, and errors for several candidates were computed through vectorized operations, allowing independent candidate evaluations to run in parallel.

For the OK routine, the geostatistical workflow reproduced the logic used by the AgDataBox R routine. The system constructed the experimental semivariogram from the lag structure, fitted theoretical models using the selected methods, and filtered the resulting candidates using the same validity criteria as the reference process. Each valid candidate was then evaluated by leave-one-out cross-validation, followed by the calculation of selection metrics and ranking of the resulting models.

The system implements the leave-one-out cross-validation stage in OK as a matrix-based procedure inspired by Dubrule (1983). In the conventional approach, one kriging system would be assembled and solved for each sample removed from the dataset. The procedure combines the covariance matrix and the ordinary kriging constraint into an augmented system, which is solved once for the entire set of samples. The resulting matrix contains the information required to obtain the leave-one-out prediction errors for all samples through algebraic operations. Therefore, the most expensive part of cross-validation is concentrated in a single linear-system solution, reducing repeated system assembly and solving while preserving the same validation logic used by the original routine.

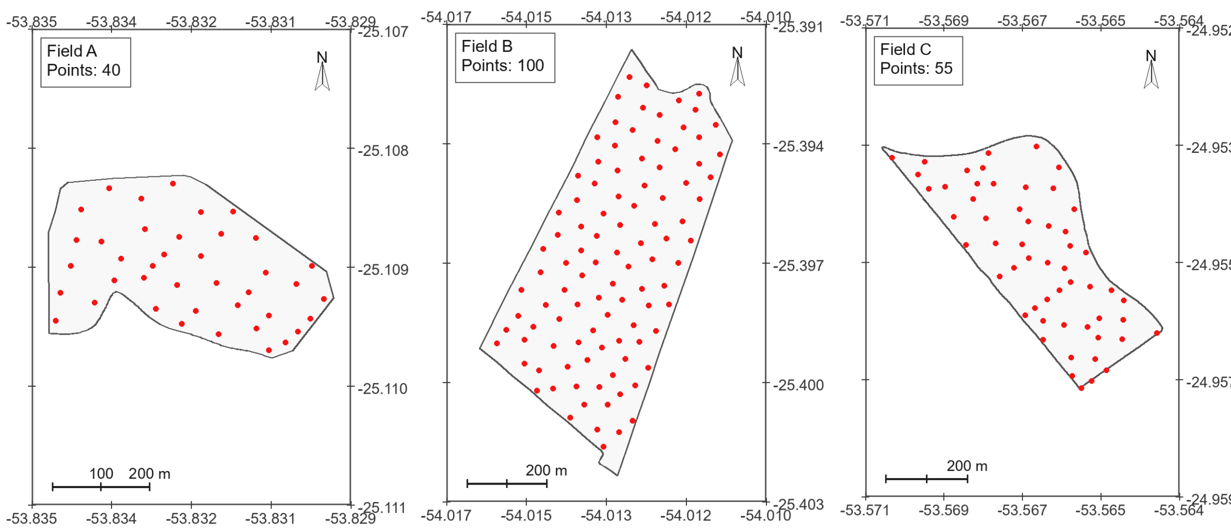
After cross-validation, each candidate was summarized by statistical metrics, including the mean error (EM) and the standard deviation of the mean error (DPEM). Bier and Souza (2017) proposed the interpolation selection index (ISI; Equation 1), which normalizes error tendency and dispersion across all j candidates evaluated in the same selection process.

$$ISI_i = \frac{|EM_i|}{\max\limits_{i=1,\ldots,j} |EM_i|} + \frac{DPEM_i - \min\limits_{i=1,\ldots,j} DPEM_i}{\max\limits_{i=1,\ldots,j} DPEM_i} \tag{1}$$

In which, ISI_i is the selection index of candidate i, EM_i is the mean error obtained by cross-validation, $DPEM_i$ is the standard deviation of the mean error, and j is the number of candidates under comparison. Lower ISI values indicate better candidates because they combine lower absolute bias and lower error dispersion in a single ranking criterion.

Experimental fields and datasets

The case study used data from three experimental fields in western Paraná, Brazil (Fig. 1). The fields and their respective sampling grids are: Field A with 40 points (25°06'32.7" S, 53°49'57.8" W), located in Céu Azul, PR; Field B with 100 points (25°24'28" S, 54°00'17" W), located in Serranópolis do Iguaçu - PR, and Field C with 55 points (24°57'17.03" S; 53°34'1.68" W) located in Cascavel - PR.



a) Field A, Céu Azul - PR **b) Field B, Serranópolis do Iguaçu - PR** **c) Field C, Cascavel - PR**

Fig. 1. Boundaries and sampling grids of the experimental fields: (a) A in Céu Azul-PR, (b) B in Serranópolis do Iguaçu-PR, and (c) C in Cascavel-PR.

We converted the sampling point coordinates to the WGS 84 / UTM zone 22S coordinate reference system (EPSG:32722), enabling the use of metric distances in the interpolation routines. In the first stage of the case study, the study used datasets based on the sampling grids of each field. The datasets included physical soil attributes, such as clay and sand, and chemical attributes, such as carbon (C), calcium (Ca), potassium (K), magnesium (Mg), phosphorus (P), zinc (Zn), copper (Cu), and iron (Fe). Table 1 summarizes the datasets used in this case study.

Table 1. Experimental datasets used in the first compatibility case study

Field	Sampling points	Analyzed attributes
A	40	Clay, Ca, K, Mg, and P
B	100	C, Ca, K, Mg and Zn
C	55	Sand, C, Ca, Cu and Fe

The second stage of the case study verifies whether the GPU implementation remains functional for datasets with higher point density. After preprocessing, the study used two dense datasets: altitude (3,507 points) and apparent soil electrical conductivity (EC_a) (3,211 points). Duplicate, null, negative, and outlier records were removed, and points outside the field boundaries were excluded. Subsets of 40, 100, 300, 500, and 2000 points were generated for each attribute. Additional subsets corresponding to one fourth, one third, and one half of each complete dataset were also generated via stratified random sampling based on k-means clustering of the coordinates. Each subset was submitted to the API three times, and the average response time was recorded as complementary information. The selected parameters were then used to generate thematic maps. Comparisons were performed sequentially, from the lowest to the highest sampling density, using the CDR between consecutive maps.

For each dataset, the original selection routine in R was executed within AgDataBox. In contrast, the new Python routines were executed through the API in two modes: GPU execution and CPU fallback. The same input data and selection criteria were used in all executions. For OK, the exponential, spherical, Gaussian, and Matern variogram models were evaluated using ordinary least squares and weighted least squares. For the Matern model, kappa values of 0.5, 1.0, 1.5, and 2.0 were tested. The Matheron estimator was used for the experimental semivariogram, with at least 30 pairs of points per interval, a 50% cutoff relative to the maximum distance among samples, and automatic lag-number calculation. Filters related to null range, effective spatial dependence, first semivariance, and model slope were applied.

For IDW, the routine evaluated power exponents from 1.0 to 6.0, with increments of 0.5, and numbers of nearest neighbors from 4 to 12. It used a search radius of 0, meaning that the number of neighbors defined in each candidate configuration controlled neighborhood selection. After selection, the routine used the parameters recommended by each implementation to generate thematic maps in AgDataBox-Map.

All maps used the same field boundaries, 5 m × 5 m pixel dimensions, four classes, equal-interval classification, and shared class limits for each map pair. This approach ensured that visual differences were associated with the selected parameters and numerical results, rather than with differences in map rendering or color classification.

Map-comparison metric

The AgDataBox-Map quantified map compatibility using the relative deviation coefficient (CDR; Coelho et al., 2009; Equation 2). This coefficient expresses the mean absolute percentage difference between the interpolated values in a test map and those in a reference map. The maps generated by the R routine in AgDataBox were used as a reference and compared with those generated by the fallback routines on the GPU and CPU in Python.

$$CDR = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{P_i^{ref} - P_i^{test}}{P_i^{ref}}\right| \times 100 \qquad (2)$$

in which n is the number of interpolated raster cells compared, P_i^{ref} is the value of cell i in the reference map generated from the AgDataBox routine, and P_i^{test} is the corresponding value in the map generated from the Python routine. CDR values closer to zero indicate higher similarity between the compared maps.

Results and Discussion

Implementation results

The Python service was successfully integrated into the API and made available through the specific endpoint (POST /interpolation/select). The endpoint received samples and parameters in JSON format, validated the input structure, and executed the interpolation-selection workflow (Fig. 2). The response returned the candidate models ordered by the global selection criterion, allowing the recommended model to be obtained directly from the first returned item.

```json
{
  "samples": [
    { "x": 100.0, "y": 200.0, "value": 12.5 },
    { "x": 110.0, "y": 205.0, "value": 13.1 },
    { "x": 120.0, "y": 215.0, "value": 11.8 }
    ...
  ],
  "params": {
    "models": ["exp", "sph", "gaus", "matern"],
    "methods": ["ols", "wls"],
    "kappas": [0.5, 1, 1.5, 2],
    "estimator": "classical",
    "range_intervals": 5,
    "partialsill_intervals": 5,
    "cutoff": 50,
    "lambda": 1,
    "auto_lags": true,
    "lags": 10,
    "pairs": 30,
    "fix_limits": true,
    "filters": ["range0", "esdi", "y1", "sme"],
    "idw": { "min_exp": 1.0, "max_exp": 6.0, "step_exp": 0.5, "min_neighbors": 4, "max_neighbors": 12, "radius": 0 }
  }
}
```


Fig. 2. Example JSON payload for requesting interpolator selection analysis through the API.

In the geostatistical pipeline, the routine calculates the distance matrix once and reuses it across the candidate models. In GPU execution, it transfers the matrix to device memory and keeps it available for evaluating different combinations. This approach reduced redundant CPU-GPU transfers and enabled the GPU to be used primarily for repeated numerical operations during the validation and candidate-evaluation stages. The matrix-based cross-validation formulation also reduced the need to solve many independent systems for each omitted sample.

In the IDW pipeline, the parameter grid combined different power exponents and neighbor counts. The vectorized GPU implementation evaluated the candidate combinations using the shared distance matrix and partitioning operations to isolate nearest neighbors. This organization avoided repeated full sorting of distances for each candidate configuration and made the deterministic routine compatible with the same API-level workflow used for OK.

Compatibility with the AgDataBox reference routine

The results of the first case study indicated that the Python routines reproduced AgDataBox's selection behavior. When IDW was selected, the power exponent and the number of neighbors were identical across the AgDataBox, Python GPU, and Python CPU results. When the Python implementation selected OK, the variogram model, fitting method, and parameters were identical or very close, with differences limited to small numerical rounding errors.

One particular case occurred in field B for Ca. The reference routine returned two models with identical ISI at high decimal precision: an exponential model and a Matern model. The GPU execution returned the Matern model as the best configuration, whereas the CPU fallback returned the exponential model. The procedure retained both alternatives in the comparison, and the CDR remained 0% in both cases. This result indicates that ties or near-ties in the selection index may change the ordering of candidate models without affecting the generated thematic map in a meaningful way.

Table 2 summarizes the CDR values observed in the first case study. Most values were rounded to 0%. The highest Cu value was observed in field C (0.01%). These values are well below 1%, indicating that the maps generated by the Python GPU and CPU routines were highly consistent with the AgDataBox reference maps.

Table 2. Summary of relative deviation coefficient values for maps generated from the AgDataBox, Python GPU, and Python CPU routines

Field	Compared raster cells	Attributes	Max. CDR (%), AgDataBox vs GPU	Max. CDR (%), AgDataBox vs CPU
A	6740	Clay, Ca, K, Mg, P	0.00	0.00
B	14586	C, Ca, K, Mg, Zn	0.00	0.00
C	7794	Sand, C, Ca, Cu, Fe	0.01	0.01

The visual comparison of maps supported the numerical results (Fig. 3). For each dataset, we generated maps with the same color scale, number of classes, and class intervals. Under these conditions, the thematic maps produced from the Python-selected parameters presented the same spatial interpretation as the reference maps. The small differences observed in some fitted variogram parameters were therefore not relevant for the final map output.

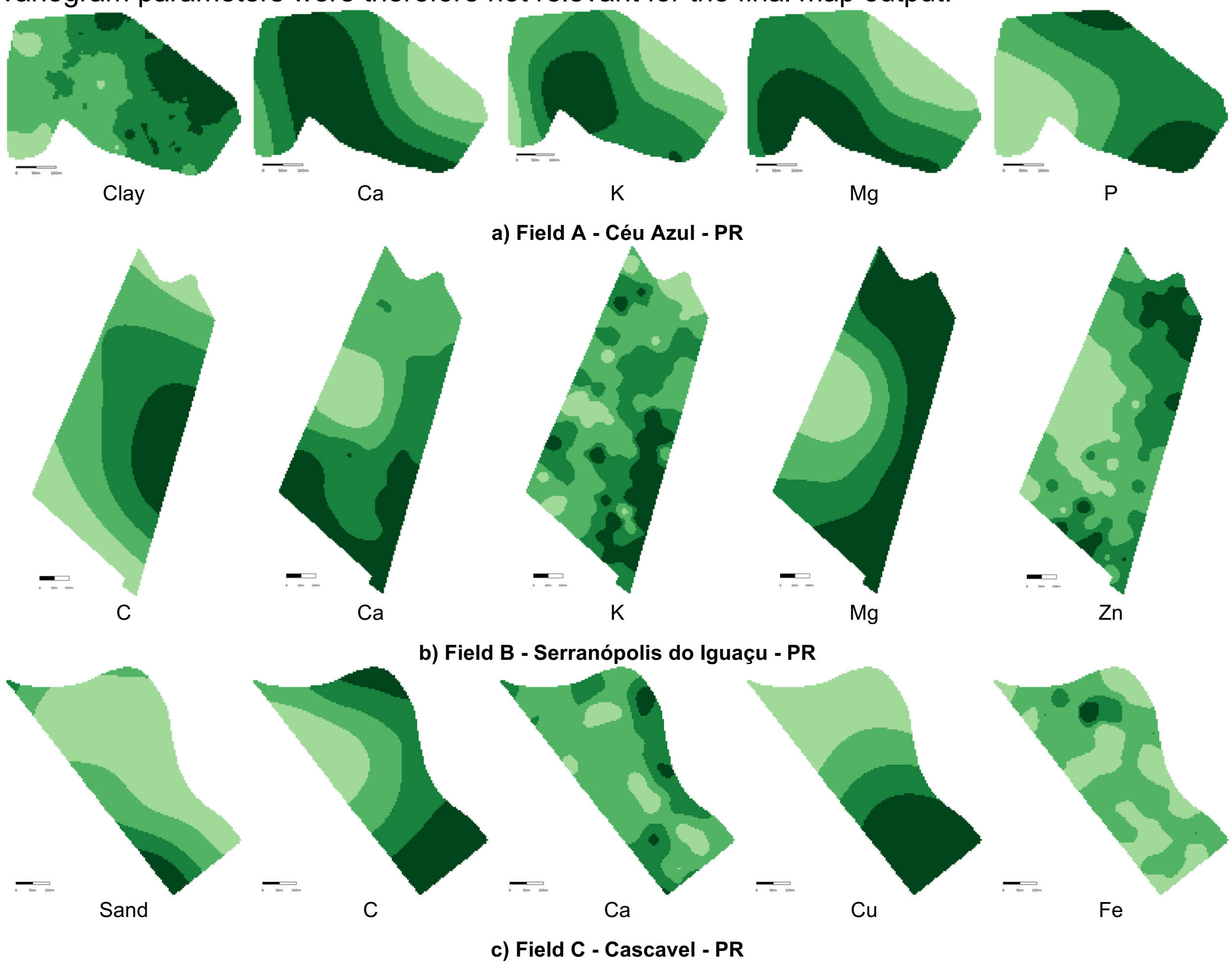


Fig. 3. Thematic maps of the sample data for Fields A, B, and C.

Parameter selection based on dense datasets showed that the GPU routine remained functional for datasets containing thousands of sampling points. The selected models varied across sampling densities, as expected, because the reduced subsets did not necessarily preserve the same spatial structure and attribute variability as the complete datasets. This variability in selected models did not always translate into large changes in the generated maps, particularly in altitude.

For altitude, the complete dataset contained 3507 points. Across the evaluated densities, OK and IDW were both selected in different subsets. The sequential CDR values ranged from 0.06% to 0.15%, indicating that the maps were highly similar as the sampling density increased. The largest sequential deviation occurred between the maps generated with 40 and 100 points, with a CDR of 0.15% (Table 3).

For EC_a, the complete dataset contained 3211 points. The first two subsets selected IDW, while most of the larger subsets selected OK with different variogram models and fitting methods. The sequential CDR values ranged from 2.08% to 12.34%. The highest value occurred between the 100-point and 300-point maps. This greater variability was mainly attributed to the sampling-reduction procedure, which accounted for sample spatial location but did not guarantee preservation of the attribute distribution or the spatial autocorrelation structure (Table 3).

Table 3. Summary of the dense-dataset case study using sequential map comparisons

Attribute	Complete dataset size	Evaluated sample sizes	Selected interpolators across densities	Sequential CDR range (%)	Highest sequential CDR (%)
Altitude	3507	40, 100, 300, 500, 877, 1169, 1754, 2000, 3507	OK in 6 subsets; IDW in 3 subsets	0.06-0.15	0.15 (40 vs 100)
Apparent soil electrical conductivity	3211	40, 100, 300, 500, 803, 1070, 1606, 2000, 3211	IDW in 2 subsets; OK in 7 subsets	2.08-12.34	12.34 (100 vs 300)

Fig. 4 shows the thematic maps generated from datasets with different sampling densities in Field A. The altitude maps remained very similar even when the number of sampling points was substantially reduced, preserving the main spatial patterns observed in the complete dataset. In contrast, the apparent soil electrical conductivity maps showed greater variation across sampling densities, with changes in local patterns becoming more noticeable as the number of points decreased.

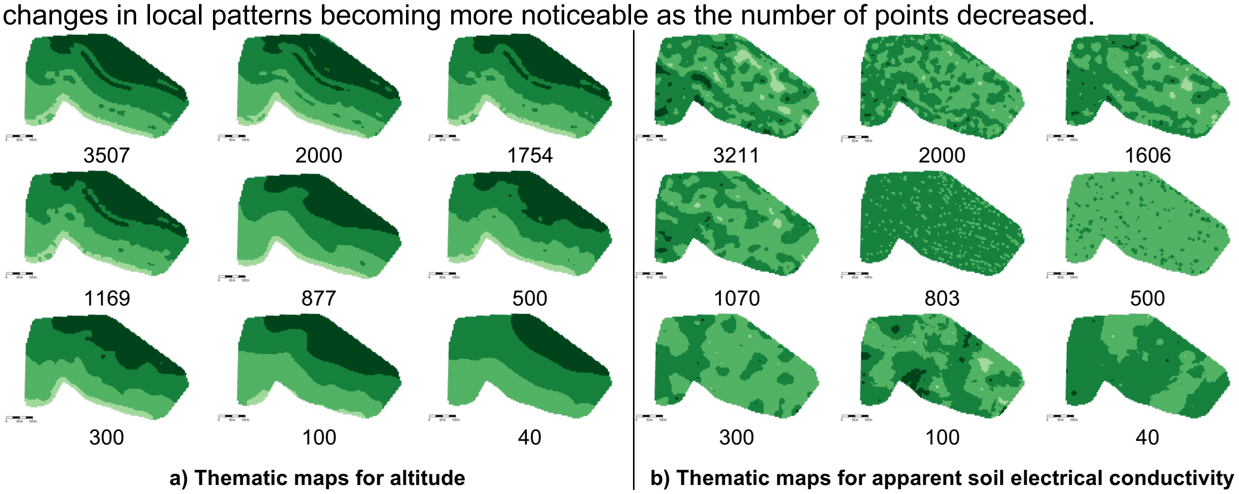


Fig. 4.Thematic maps for the dense datasets, including the complete dataset and stratified subsets reduced to 40 points in Field A.

The API response times increased as the number of sampling points increased. For small subsets, the increase was less pronounced, which may be related to fixed processing costs such as data reading, structure preparation, and initialization. For larger subsets, especially above approximately 1600 to 2000 points, the increase became more evident. This behavior suggests that operations that depend on the number of samples, including distance calculation, candidate evaluation, and cross-validation, dominate execution cost in larger scenarios.

Although a formal performance benchmark was outside the scope of the study, the dense-dataset experiment indicated where future optimization efforts should be concentrated. The current implementation expanded the feasibility of automatic parameter selection beyond the smaller datasets currently used in the platform workflow. However, additional work is still required to improve response time for larger production scenarios.

Conclusions

A GPU-enabled Python implementation of the AgDataBox parameter-selection routines for IDW and OK was developed and integrated into an API. The implementation preserved CPU fallback execution and reproduced the main stages of the original selection workflow, including candidate generation, cross-validation, statistical evaluation, and ranking by a global selection criterion.

In the compatibility case study with 15 experimental datasets, the thematic maps generated by the Python GPU and Python CPU routines, selected using the Python GPU and Python CPU routines, were highly similar to those generated by the AgDataBox reference routine. The CDR values were close to or equal to zero in almost all comparisons, and the maximum CDR observed was 0.01%. These results indicate that the implementation reproduced the practical output of the original routine for the evaluated datasets.

In the dense-dataset case study, the GPU routine remained functional for datasets with 3507 altitude points and 3211 conductivity points. The altitude maps showed low sequential variation, whereas the conductivity maps showed greater differences in some comparisons, mainly because the reduced subsets did not fully preserve the representativeness of the original spatial structure.

The proposed implementation contributes to the evolution of the AgDataBox platform by extending automatic interpolation-parameter selection to a Python API with GPU support. Future work should investigate additional performance improvements for datasets with more sampling points, extend GPU execution to the final interpolation step for IDW and OK, integrate the routine into the complete production workflow of AgDataBox, and validate the system with users in operational environments.

Acknowledgments

The authors thank the Federal University of Technology - Parana (UTFPR), the Multi-user Agricultural Technology Laboratory (AGRILAB-MD), the Soil Laboratory (LabSolos-PB), the Araucaria Foundation - Paraná, and the National Council for Scientific and Technological Development (CNPq).

References

Bazzi, C. L., et al. (2019). AgDataBox API - integration of data and software in precision agriculture. *Computers and Electronics in Agriculture*, 10, 100327.

Betzek, N. M., et al. (2019). Computational routines for the automatic selection of the best parameters used by interpolation methods to create thematic maps. *Computers and Electronics in Agriculture*, 157, 49-62.

Bier, V. A., Souza, E. G. (2017). Interpolation selection index for delineation of thematic maps. *Computers and Electronics in Agriculture*, 136, 202-209.

Coelho, E. C., et al. (2009). Influencia da densidade amostral e do tipo de interpolador na elaboracao de mapas tematicos. *Acta Scientiarum. Agronomy*, 31, 165-174.

Dall'Agnol, R. W., et al. (2020). Web applications for spatial analyses and thematic map generation. *Computers and Electronics in Agriculture*, 172, 105374.

Dubrule, O. (1983). Cross validation of kriging in a unique neighborhood. *Journal of the International Association for Mathematical Geology*, 15(6), 687-699.

Sobjak, R., et al. (2023). Process improvement of selecting the best interpolator and its parameters to create thematic maps. *Precision Agriculture*, 24(4), 1461-1496.