

Article

Not peer-reviewed version

Proposal and Prototype of a GUI-Based Algorithm for R-Wave Correction and Immediate R-R Interval Updating

[Yutaka Yoshida](#)^{*} and Kiyoko Yokoyama

Posted Date: 21 October 2025

doi: [10.20944/preprints202509.0164.v2](https://doi.org/10.20944/preprints202509.0164.v2)

Keywords: ECG viewer; GUI; R-R interval; signal correction; biosignal processing



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Proposal and Prototype of a GUI-Based Algorithm for R-Wave Correction and Immediate R-R Interval Updating

Yutaka Yoshida ^{1,*} and Kiyoko Yokoyama ²

¹ Graduate School of Design & Architecture, Nagoya City University, 2-1-10, Kita Chikusa, Chikusa-ku, Nagoya 464-0083, Japan

² Graduate School of Date Science, Nagoya City University, 1, Yamanohata, Mizuho-cho, Mizuho-ku, Nagoya 467-8501, Japan

* Correspondence: yoshida@sda.nagoya-cu.ac.jp

Abstract

Electrocardiography (ECG) is a representative biosensor that noninvasively records weak bioelectrical signals of the heart. It is widely used to assess cardiac function, autonomic activity, and exercise responses. Accurate detection of R-wave peaks and precise calculation of R-R intervals (RRIs) are essential for heart rate variability (HRV) analysis, which is applied to stress assessment, sleep stage evaluation, and the detection of subtle cardiac changes. Automated R-peak detection algorithms, ranging from the classical Pan–Tompkins method to recent deep learning approaches, enable rapid processing of large datasets but remain susceptible to misdetections caused by noise, baseline fluctuations, or prominent T-waves. Conventional correction methods often rely on adjusting filters or thresholds, but these approaches risk introducing new errors in other regions, highlighting a trade-off problem and emphasizing the need for a user-friendly manual correction function. To overcome these limitations, we proposed a prototype graphical user interface (GUI)-based ECG viewer implemented in Fortran to ensure high computational efficiency. The system enables interactive insertion or deletion of R-wave peaks, followed by immediate recalculation of RRIs and automatic updates of associated analyses, including power spectral density, histograms, Lorenz plots, and polar plots. Validation with synthetic ECG signals at four sampling frequencies (125, 250, 500, and 1000 Hz) and three time scales (2, 5, and 10 s) demonstrated correction errors below 0.7% and stable update times within 20–30 ms. These results demonstrate reliable, immediate feedback that may be applicable in future contexts. Accordingly, the proposed algorithm may further support applications in research, clinical, and educational domains of biosignal processing.

Keywords: ECG viewer; GUI; R-R interval; signal correction; biosignal processing

Electrocardiography (ECG) is a representative biosensor that noninvasively measures weak bioelectrical signals, and improving the accuracy of these signals is essential for reliable applications. ECG is widely used as a measurement tool across various medical and physiological fields, including the diagnosis of cardiovascular diseases, assessment of autonomic nervous system activity, and exercise physiology research [1–5]. Since ECG records the electrical activity of the heart over time, analysis of the rhythm and morphology of the waveform enables the evaluation of both normal and abnormal cardiac events, as well as subtle changes in cardiac function. In particular, the R-R interval (RRI), defined as the interval between consecutive R-peaks, serves as a fundamental metric in heart rate variability (HRV) analysis and is broadly applied in studies assessing autonomic function, monitoring stress load, analyzing sleep stages, and evaluating exercise responses [6–8]. Such applications of ECG highlight its central role in biosignal processing, where cardiac dynamics are analyzed alongside other physiological signals to assess systemic conditions. Furthermore, RRI time-series data, when combined with techniques such as power spectral analysis, time-series analysis,

and nonlinear analysis, have been reported to detect subtle changes in cardiac activity that cannot be captured by mean heart rate alone [9–14]. Therefore, accurate detection of R-peaks and precise calculation of RRIs are critically important in ECG analysis research.

Automated R-peak detection algorithms, originating from the Pan–Tompkins method proposed in 1985 [15] and subsequent improvements, have been widely utilized to support large-scale data analysis in both research and clinical settings. These algorithms combine multiple processing steps, including noise removal via bandpass filtering, R-peak enhancement using differentiation and squaring, peak shaping through moving integration, and peak extraction via thresholding, to achieve rapid and automated R-peak detection [16,17]. More recently, deep learning-based methods for R-peak detection have also been proposed [18–20].

However, baseline variations caused by experimental conditions or subject movement, cases where T-waves exceed R-waves [21], and the presence of noise can result in inaccurate R-peak identification. Such errors reduce the precision of RRI calculation and may introduce inaccuracies in subsequent analyses, such as HRV or stress evaluation. Manual correction of misdetections is burdensome even in short recordings, and the workload increases dramatically in long-term recordings such as 24-hour Holter ECG. Therefore, an efficient and user-friendly function that allows analysts to quickly and intuitively correct R-wave positions is essential. In the present study, the focus is on short-term experimental recordings and partial corrections within long-term or clinical recordings, rather than manual correction of entire long-duration datasets. Traditional approaches typically correct misdetections by readjusting filters or thresholds, but these methods carry the risk of introducing new errors outside the corrected regions, underscoring the need for a manual yet efficient correction function that avoids such trade-offs.

Previously developed software tools include Kubios HRV, widely used for HRV analysis [22]; RR-APET, a Python-based graphical user interface (GUI) tool allowing R-peak editing and batch processing [23]; ECGdeli, a MATLAB-based ECG visualization tool [24]; as well as CEPS [25] and PhysioZoo [26]. These tools provide valuable functionalities for HRV and ECG research, offering extensive metrics and analysis capabilities. However, their support for interactive, immediate correction of single R-peak misdetections remains limited. For example, Kubios HRV provides comprehensive HRV metrics and some manual editing options but does not prioritize immediate propagation of edits across all derived analyses. RR-APET allows GUI-based editing, but corrections are applied in batch rather than supporting fine-grained, single-beat adjustments with immediate updates. ECGdeli is powerful for ECG delineation but does not emphasize interactive correction or immediate recalculation of RRIs, while CEPS and PhysioZoo focus on complexity analysis or mammalian HRV, without a strong emphasis on GUI-based correction. In contrast, the proposed system was specifically designed to address this gap by enabling intuitive, single-event correction. Users can visually identify and directly adjust an R-wave with a single mouse click, after which the RRI time series and related analyses are immediately updated. This unique combination of fine-grained manual correction and immediate feedback distinguishes our system from existing tools.

To address these challenges, this study proposes an ECG viewer equipped with GUI-based correction functionality. The system allows simple manual correction of failed R-peak detection or T-wave misidentification and immediate updating of the RRI time series. By complementing the limitations of automated detection algorithms, the proposed system aims to improve both the efficiency and accuracy of ECG analysis.

2. Design

2.1. Programming Language and Drawing Library

The computational core of the system was implemented in Fortran (FTN95 compiler, Silverfrost Ltd., Manchester, UK), and the DISLIN graphics library was used for drawing. Fortran was chosen because of its high numerical computation performance and compatibility with existing code assets.

DISLIN provides versatile two- and three-dimensional graphics functions as well as bitmap output, making it suitable for waveform visualization and interactive drawing.

For the development environment, the Fortran editor on Windows (Plato IDE, version 4.3.0, Silverfrost Ltd., Manchester, UK) was used to implement both the numerical processing and GUI logic. DISLIN (version 10.0, Dislin Software, Göttingen, Germany) was employed for graphics, and the linkage to DISLIN was established through dynamic linking. In Plato IDE, diss32.lib was added as the link library, enabling calls to DISLIN functions. At runtime, disdll.dll, diss32.dll, and the FTN95 runtime library salflibc.dll were placed in the same directory, allowing the graphics functions to be accessed via dynamic linking. The specifications of the PC and display used in this study are summarized in Table 1.

Table 1. Environmental specifications.

System Specifications	
Operating System	Windows 10, 64-bit
Processor	Intel (R) Core (TM) i5-8500 CPU @ 3.00GHz 3.00 GHz
Installed RAM	16.0 GB
Display Specifications	
Graphics Card	Intel(R) UHD Graphics 630 (128 MB)
Panel Type	27-inch wide TFT LED, non-glare, ADS panel
Maximum Resolution	1920×1080
Pixel Pitch (H × V)	0.3114 mm × 0.3114 mm
Display Area (H × V)	597.888 mm × 336.312 mm
Maximum Colors	16.77 million colors
Response Time	14 ms [GTG] (6.1 ms [GTG] with overdrive enabled)

2.2. Viewer Design and Operation

The viewer was designed using the DISLIN functions CALL CURVE and CALL BARS, as illustrated in Figure 1. First, the CALL METAFL('XWIN') function was used to define a graphical output window, with the window size set to approximately 465 mm (H) × 310 mm (V). The screen layout was organized as follows:

- (a) Top panel: ECG waveform (cyan) and peak flags (magenta)
- (b) Middle panel: RRI time series (cyan)
- (c) Bottom panel: From left to right, the power spectral density function of the RRI time series (yellow), histogram (orange), Lorenz plot (green), and polar plot (peach, circular waveform of RRI)

The four analyses were mathematically defined as follows:

Power spectral density (PSD): estimated using the fast Fourier transform (FFT) of the RRI time series $r(n)$:

$$P(f) = \frac{1}{N} \left| \sum_{n=0}^{N-1} r(n) e^{-2\pi f n / N} \right|^2$$

(1)

Histogram: distribution of RRI values $r(n)$, represented as frequency counts within bins:

$$h(k) = \sum_{n=0}^{N-1} \delta(r(n) \in bin_k)$$

(2)

Lorenz plot: scatter plot of successive RRIs: (r_n, r_{n+1})

Polar plot: circular mapping of RRIs:

$$(r_n, \theta_n), \theta_n = 2\pi(n/N) \quad (3)$$

In addition, a toolbar was placed at the top of the screen. The toolbar contained the following buttons: Insert, Delete, Resolution: up, Resolution: down, Reset, and Quit, which could be operated using the left mouse button. Right clicking was disabled.

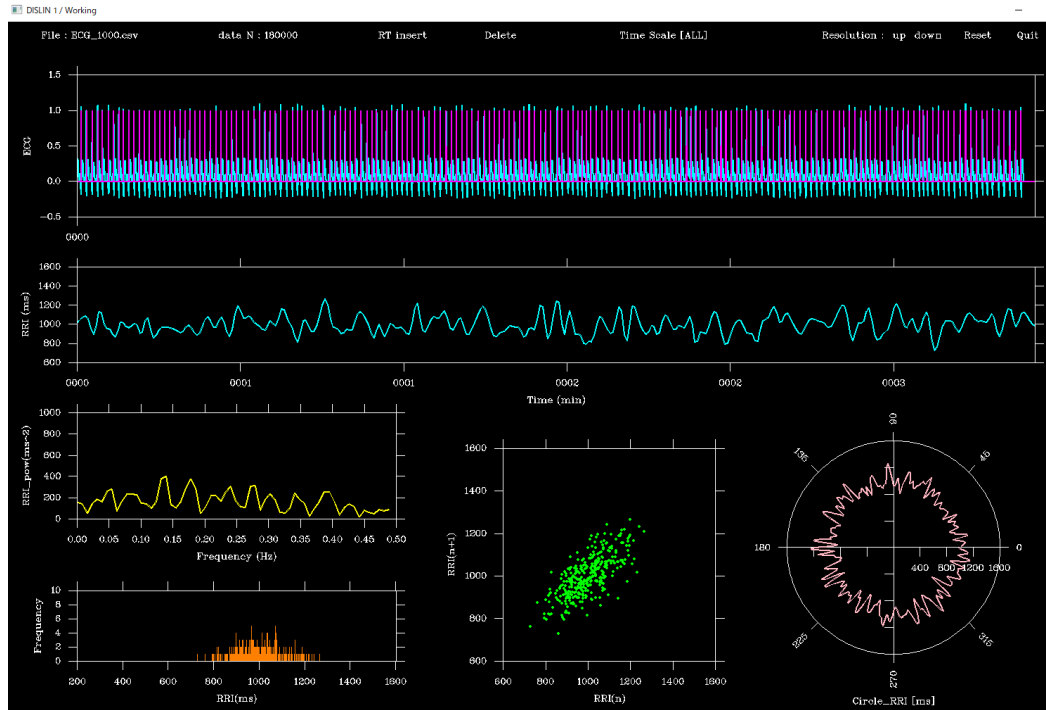


Figure 1. Graphical ECG viewer. ECG sampling frequency:1000Hz, data length:3min, data number:180000.

2.2.1. Time Scale Adjustment

Clicking the Resolution: up button shortened the time scale of the ECG display (e.g., 60 s $\rightarrow$ 30 s $\rightarrow$ 10 s), down to a minimum of 1 s. Conversely, the Resolution: down button extended the scale (e.g., 120 s $\rightarrow$ 180 s $\rightarrow$ 300 s), up to a maximum of approximately 1 hour at a sampling frequency of 1000 Hz.

These operations affected only the time scale of the ECG graph, without influencing other plots. Clicking outside the toolbar advanced the time axis by one time scale unit. Similarly, pressing the numeric keypad "6" or the right arrow key advanced the display by half a time scale unit, while the Backspace key moved back one unit, and the numeric keypad "4" or the left arrow key moved back by half a unit. Clicking Reset restored the full-length ECG display.

2.2.2. Insert Function

Clicking the Insert button activated insertion mode, indicated by the button turning yellow. In this mode, clicking within the ECG graph frame or pressing the Enter key displayed a vertical line (flag), and the corresponding time was recorded. Clicks or key presses outside the frame were ignored. This function allowed the user to add flags at arbitrary time points simply by positioning the mouse cursor and clicking. Clicking Insert again terminated the mode.

2.2.3. Delete Function

The Delete function removed a flag and its associated time, using an operation similar to the Insert function. By clicking or pressing Enter just before the target flag, the flag and its recorded time were deleted. The deletion interval was fixed at 0.2 s, independent of the sampling frequency. This feature was useful for removing misdetections caused by P- or T-waves.

2.3. Operational Flowchart and Algorithm

The processing flow of the system is illustrated in Figure 2. After acquisition, ECG data were converted into a three-column, two-dimensional array, $array(3,)$, storing measurement time, ECG values, and peak information as follows:

- (a) $array(1, i)$: Measurement time (in seconds or sampling index). For example, at a sampling frequency of 1000 Hz, values such as 0.001 s, 0.002 s, 0.003 s ... were stored.
- (b) $array(2, i)$: Sampled ECG waveform (raw data).
- (c) $array(3, i)$: Flag indicating whether an R-wave peak was detected (1 for peak, 0 otherwise).

The RRI is calculated by extracting the measurement times corresponding to $array(3, i) = 1$ and computing the time differences between consecutive peaks. With this structure, when the user interactively inserts or deletes an R-wave peak, only the corresponding flag value in $array(3, i)$ is updated, while the raw ECG data remain unchanged. After editing, the RRI is recalculated from the updated $array(3, i)$, interpolated using a spline function, and other graphs (RRI time series, power spectrum, Lorenz plot, histogram, and polar plot) are updated rapidly. This non-destructive raw-data approach with flag updates preserves data integrity and enables lightweight, immediate updates. Thus, error correction and temporal adjustments can be performed intuitively and safely.

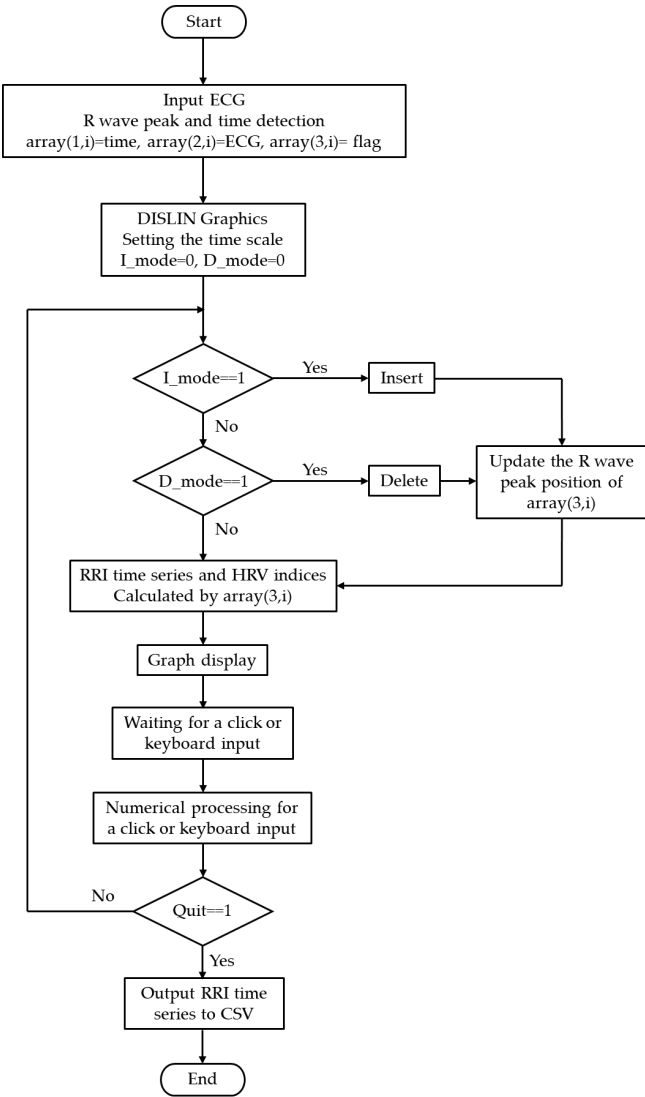


Figure 2. System flowchart.

Figure 3 compares the ECG before and after editing for a 5-s segment around 2 minutes. In Insert mode (Figure 3, middle), the P- and T-wave times for this segment were extracted, whereas in Delete mode (Figure 3, lower), three R-waves at the same time point were removed. Compared with the unedited version, the RRI time series and other graphs changed accordingly.

After displaying the graphs, the system waited for user input, responding interactively to mouse clicks and keyboard commands. To convert mouse click positions on the screen into time information, the following equation (Eq. (4)) was defined:

$$T_{pos} = x_t + \frac{Time\ scale \times (NX - X_{left})}{X_{right} - X_{left}} \tag{4}$$

Here, x_t is the starting time of the currently displayed ECG (s), Time scale is the display length (s), NX is the mouse click x- coordinate (in pixels), Xleft is the x- coordinate of the left ECG axis, and Xright is the x- coordinate of the right ECG axis. This transformation ensured that any mouse click on the screen could be mapped precisely to the corresponding time in the ECG time series, allowing accurate addition or deletion of R-wave peaks.

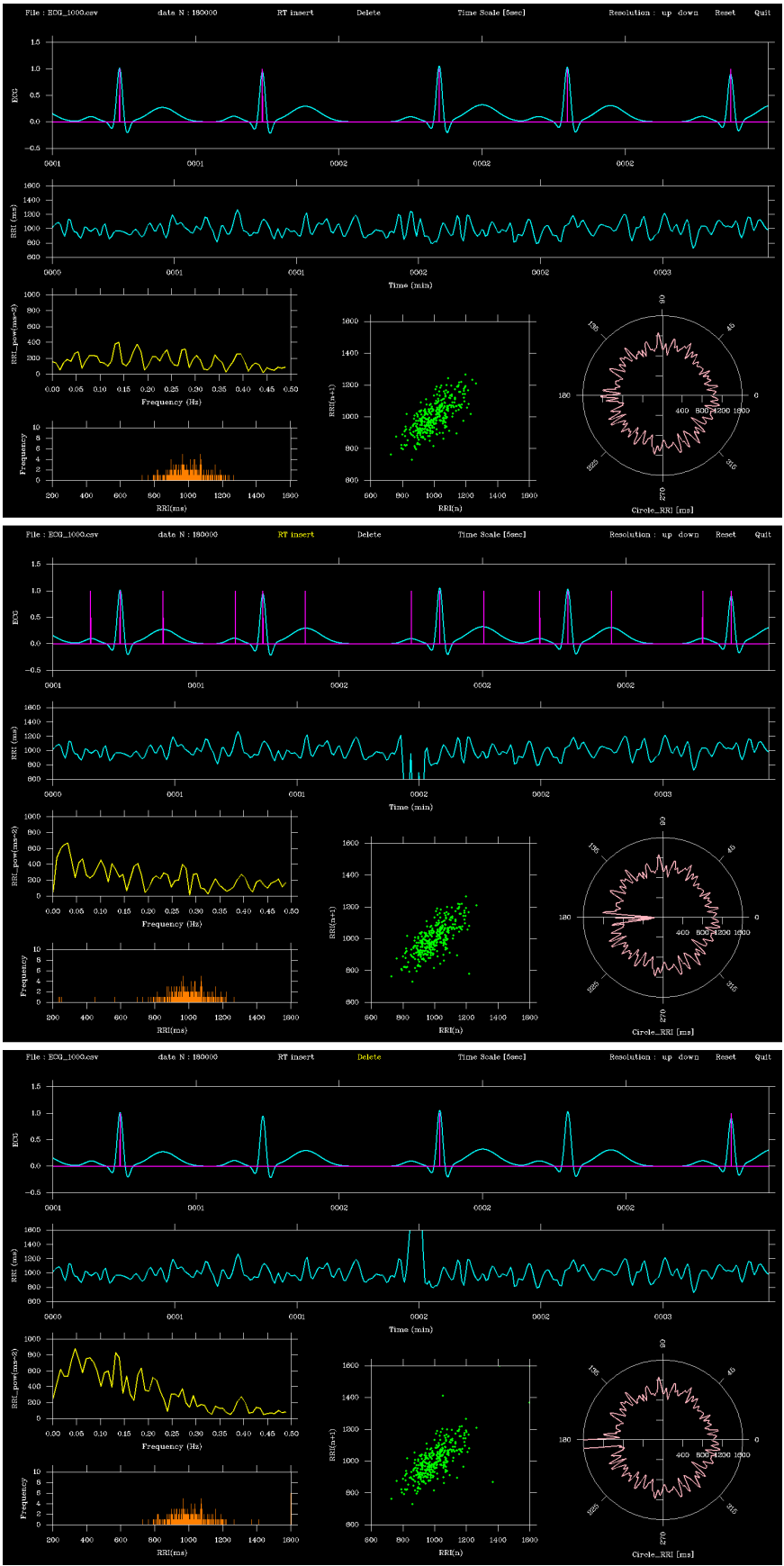


Figure 3. Comparison before and after insertion/deletion of ECG flags via the GUI. Upper: before editing, Middle: insert mode, Lower: delete mode. Editing the flags alters the RRI time series, which propagates to the other graphs.

2.4. ECG Simulation Data for Accuracy Evaluation

Because R-wave correction in the GUI was performed by converting screen coordinates to time, errors could arise depending on the accuracy of this transformation. Such errors were expected to depend on the ECG sampling frequency and the time scale used for display. To evaluate the accuracy of the system, a synthetic ECG dynamic model was employed (Eq. (5)), in which the P-, Q-, R-, S-, and T-waves were approximated using Gaussian functions with specified amplitudes (A_k), time offsets (t_k), and standard deviations (σ_k). Each parameter was randomly varied within $\pm 10\%$ of its reference value to reproduce beat-to-beat morphological variability (Table 2).

The reference values were derived from the widely used model of McSharry et al. [27], which has become a standard for generating physiologically plausible ECG signals under normal sinus rhythm. Using this model ensured that the simulated ECG retained realistic morphology while providing precise gold-standard R-wave timings for accuracy evaluation. Although the core aim of this study was simply to test the accuracy of the coordinate-to-time transformation—something that could theoretically be evaluated with any waveform containing known peaks—a synthetic ECG was selected to align the validation process with the context of ECG analysis.

$$ECG(t) = \sum_{k \in \{P,Q,R,S,T\}} A_k \cdot \exp\left\{-\frac{(t-t_k)^2}{2\sigma_k^2}\right\}$$

(5)

Table 2. Parameters of a synthetic ECG dynamic model.

Wave	t_k [s]	A_k	σ_k [s]
P	0	0.15	0.025
Q	0.15	-0.1	0.012
R	0.2	1	0.015
S	0.25	-0.25	0.015
T	0.5	0.35	0.05

Simulation data were generated at sampling frequencies of 125 Hz, 250 Hz, 500 Hz, and 1000 Hz, with cycle lengths set to approximately 1 s, yielding 180 s of synthetic ECG signals. RRI values were calculated from each signal and defined as the gold-standard reference. Then, across 12 conditions (four sampling frequencies $\times$ three time scales: 2 s, 5 s, and 10 s), the author manually clicked on R-wave peaks using the mouse cursor, and relative errors between the GUI-derived RRI and the gold standard were computed using Eq. (6). In this validation, the 180 s ECG contained approximately 180 R-wave peaks. All peaks were manually clicked to detect RR intervals, and 180 relative errors were calculated for each condition. The mean and standard deviation of these errors were then obtained.

$$\text{Relative error} = \frac{|GUI_{RRI} - simulation_{RRI}|}{simulation_{RRI}} \times 100 \text{ (\%)}$$

(6)

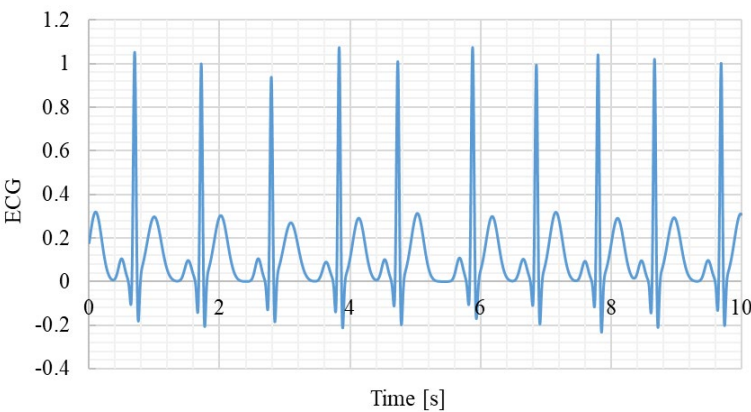


Figure 4. Example of a synthetic ECG signal. Sampling frequency 1000Hz.

Additionally, to evaluate system responsiveness, the processing time between using the Insert/Delete functions and updating the graphical display (see Figure 2 flowchart) was measured with the Fortran CALL CPU_TIME function. For each condition (four sampling frequencies $\times$ two modes), 30 clicks were performed, and the mean response time was calculated.

Furthermore, to evaluate the accuracy of the GUI-derived RRIs, statistical analyses were performed by comparing GUI-derived values with the simulated gold standard. For each condition (four sampling frequencies $\times$ three time scales), the following indices were calculated: the mean $\pm$ standard deviation (SD) and 95% confidence interval (CI) of both GUI-derived and simulated RRIs; the mean $\pm$ SD of relative errors; the Shapiro–Wilk test (W and p values) to assess the normality of error distributions; and the number of outliers ($|z| > 3$). In the Shapiro–Wilk test, the null hypothesis was that the error distributions follow a normal distribution, with a significance level of $\alpha = 0.05$. Thus, $p > 0.05$ indicated no significant deviation from normality.

All statistical analyses were performed using Python (version 3.12.7). Data handling was conducted with *pandas*, descriptive statistics with *numpy*, and normality testing with the Shapiro–Wilk test implemented in *scipy.stats*.

3. Results

Table 3 presents the mean $\pm$ SD and 95% CI of simulated RRIs (gold standard reference) and GUI-derived RRIs at different sampling frequencies and time scales. GUI-derived values were obtained from manual peak selections at 2, 5, and 10 s time scales. Across all conditions, GUI-derived RRIs showed close agreement with the simulated reference values, indicating consistency between manual corrections and the gold standard.

Table 4 summarizes the relative errors of GUI-derived RRIs compared with the simulated RRIs. The smallest relative error was $0.146 \pm 0.264\%$ at 125 Hz with a time scale of 2 s, whereas the largest error was $0.694 \pm 0.601\%$ at 125 Hz with a time scale of 10 s. The smallest standard deviation was $\pm 0.132\%$ at 1000 Hz with a time scale of 10 s. Shapiro–Wilk tests indicated that the error distributions did not significantly deviate from normality (all $W > 0.98$, $p > 0.19$), and the number of outliers ($|z| > 3$) was negligible (≤ 2 cases per condition). These results confirm that GUI-induced errors were small, statistically well- behaved, and robust across different sampling frequencies and time scales.

Table 5 presents the mean and standard deviation of operation response times for Insert and Delete modes across different sampling frequencies. The mean response time for Insert ranged from 25 to 29 ms, while that for Delete ranged from 24 to 30 ms. The standard deviation of response times ranged from 8 to 17 ms for Insert and from 8 to 14 ms for Delete.

Table 3. Mean $\pm$ SD and 95% CI of simulated RRI and GUI-derived RRI at different sampling frequencies and time scales.

Sampling frequency [Hz]	Time scale [s]	GUI_RRI [ms]	Simulation RRI [ms]
125	2	991±103 (976-1007)	992±103 (977-1007)
	5	992±103 (976-1007)	
	10	991±102 (976-1007)	
250	2	1006±98 (992-1021)	1007±98 (992-1021)
	5	1006±98 (992-1021)	
	10	1006±98 (992-1021)	
500	2	995±94 (981-1009)	996±94 (982-1010)
	5	995±94 (981-1009)	
	10	995±94 (981-1009)	
1000	2	992±100 (977-1007)	992±100 (977-1007)
	5	992±100 (977-1007)	
	10	992±100 (977-1007)	

Table 4. Relative errors (Mean ± SD, %) of GUI-derived RRIs compared with simulated RRIs across sampling frequencies and time scales, together with Shapiro–Wilk test results and outlier counts.

Sampling frequency [Hz]	Time scale [s]	Relative error [%]	Shapiro-Wilk test		n_outliers (z >3)
			W	p-value	
125	2	0.146±0.264	0.991	0.360	0
	5	0.362±0.434	0.991	0.352	0
	10	0.694±0.601	0.989	0.197	0
250	2	0.168±0.186	0.993	0.575	2
	5	0.372±0.288	0.992	0.484	2
	10	0.674±0.492	0.993	0.584	2
500	2	0.188±0.142	0.995	0.799	0
	5	0.372±0.271	0.994	0.727	0
	10	0.649±0.491	0.994	0.684	0
1000	2	0.175±0.132	0.990	0.259	0
	5	0.325±0.224	0.991	0.307	0
	10	0.646±0.481	0.990	0.238	0

Table 5. Operation response time in insert and delete for each sampling time.

Sampling frequency [Hz]	Time scale [s]	Insert response time [ms]	Delete response time [ms]
125	2	28±10	28±10
	5	27±17	28±12

	10	27±10	27±12
250	2	25±8	26±10
	5	26±16	26±10
	10	27±13	26±9
500	2	28±9	28±9
	5	29±14	28±14
	10	28±8	30±11
1000	2	28±15	24±8
	5	27±16	30±8
	10	27±17	29±8

4. Discussion

The GUI-based ECG viewer developed in this study is characterized by its ability to intuitively and immediately correct R-wave misdetections, distinguishing it from conventional software. For example, Kubios HRV covers a wide range of indices required for heart rate variability analysis and is widely used in both clinical and research settings; however, its R-peak correction functionality is limited [22]. RR-APET, a Python-based tool, allows GUI-driven R-peak editing and batch processing, but it is not fully optimized for immediate RRI updates [23]. ECGdeli is a powerful MATLAB-based tool for ECG visualization and fiducial point detection [24], and tools such as CEPS [25] and PhysioZoo [26] provide specialized functions for complexity analysis and mammalian HRV analysis, respectively. However, all of these tools primarily focus on analysis capabilities, and their GUI interfaces offer limited functionality for interactive and immediate correction of misdetections. More recently, attempts have been made to benchmark HRV analysis toolboxes [28,29], and LightWAVE [30] allows waveform and annotation editing in a browser environment. Yet, the functionality required in short-term experimental settings—namely simple correction of misdetections with simultaneous immediate updates of the RRI series and analysis results—remains insufficient. The proposed algorithm is unique in this regard, as it allows the correction of R-wave misdetections through the GUI while instantly updating the RRI time series, making it potentially applicable to short-term experimental settings.

From an engineering perspective, the system achieves both high accuracy and immediate responsiveness. The relative error of RRIs calculated after GUI-based correction was found to be less than 0.7%, demonstrating that manual corrections can be incorporated without compromising reliability. As shown in Table 4, relative error is minimally affected by sampling frequency but is sensitive to the time scale setting. This is because a longer time scale compresses the waveform per beat, making it difficult to accurately click the R-peak with a mouse, whereas a shorter time scale enlarges the waveform, facilitating visual peak identification and reducing relative error. Therefore, appropriate time scale settings are important for achieving high correction accuracy via GUI operation. Notably, at 125 Hz with a 2 s time scale, the relative error was smallest on average (0.146%), but the standard deviation was relatively large ($\pm 0.264\%$). This suggests that with coarse sampling, even a slight deviation in mouse clicking leads to substantial variability, since each sample point corresponds to a relatively large time step (8 ms). Conversely, at 1000 Hz with a 2 s time scale, the relative error showed the smallest standard deviation ($\pm 0.132\%$). This is because the temporal resolution is high (1 ms per sample), so even if the mouse click deviates by one sample, the resulting error remains small. Statistical analyses further confirmed that the errors introduced by the GUI followed a normal distribution, as indicated by the Shapiro–Wilk test (all $W > 0.98$, $p > 0.19$). This finding suggests that the errors primarily reflect random fluctuations in mouse positioning rather than systematic bias in the GUI itself. The number of outliers ($|z| > 3$) was negligible (≤ 2 per

condition), supporting the stability of the error distribution. Taken together, these results indicate that the correction accuracy achieved by the system is statistically reliable, with errors being small and well behaved.

Furthermore, as shown in Table 5, the time required to update the RRI after correction was approximately 20–30 ms and remains stable regardless of the sampling frequency or time scale. This update time includes not only RRI recalculation but also updates of four graphs based on the RRI time series: power spectral density, Lorenz plot, histogram, and polar plot. Despite the relatively high computational load of these analyses, the total processing time remains within 20–30 ms, highlighting the system's immediate responsiveness, which offers significant advantages by providing instant feedback.

A key feature of the proposed algorithm is its simplicity: by maintaining a dedicated flag array that records the positions of detected peaks, the system enables immediate recalculation of the RRI time series whenever peaks are added or removed, without altering the raw ECG data. This non-destructive structure ensures that only the flag values need to be updated, which greatly reduces computational cost and allows near-instantaneous updates of the RRI and associated analyses. The implementation of the system in Fortran is also an important strength in terms of processing speed. Fortran is a compiled language specialized for numerical computation, providing faster execution compared to interpreted languages such as Python or MATLAB. It is widely used in fields such as fluid dynamics, high-speed mathematical model development, and aerospace engineering [31–33], and its long-standing track record in high-performance computing (HPC) and numerical analysis is well recognized in academia. For example, its compiled array-based optimization structures and explicit memory access directives can achieve several- to hundred-fold speed improvements compared to Python [34]. Moreover, Fortran supports High Performance Fortran (HPF) for HPC, enabling efficient parallelization using FORALL constructs and PURE procedures [35]. In this system, the GUI and computational modules are closely integrated, and although the entire waveform is recalculated each time, high responsiveness is still achieved. Thus, the Fortran implementation represents a rational design choice emphasizing both performance and responsiveness.

Overall, the results indicate that the proposed system combines low error, low latency, and high efficiency, demonstrating feasibility for immediate updating after R-wave correction in short-term experimental settings. It should also be noted that the present study focused mainly on the development and validation of the underlying algorithm. Broader aspects of usability, such as interface layout, toolbar design, and user experience, were not considered. Future work should therefore include validation with real data, systematic benchmarking against existing tools, and structured user studies to establish practical applicability.

In addition, while the present validation isolated GUI-induced errors using synthetic signals with exact gold standard, broader benchmarking and clinical evaluation remain essential. Synthetic ECG was chosen because it provides a precise gold standard for R-wave timing, allowing GUI-induced errors to be quantified without confounding from annotation uncertainty. While this approach is appropriate for proof-of-concept evaluation, future studies should extend validation to public ECG datasets and direct comparisons with existing tools, and should involve cardiologists to verify reliability under realistic noisy and pathological conditions.

Moreover, although the current validation focused on R-wave correction, the system can be applied to arbitrary waveform landmarks (e.g., P-, Q-, S-, T-waves) and potentially to other biosignals such as respiration or electromyography (EMG) when displayed as time series. This flexibility broadens its potential applications in biosignal processing. However, cardiologist involvement and evaluation on real-world ECG data will be essential to confirm its clinical utility.

To enhance transparency and reproducibility, Supplementary Materials have been provided, including the synthetic ECG signals, simulated RRI values (gold standard), and GUI-derived RRI values at different time scales (2, 5, and 10 s) in CSV format, together with a demonstration video. These resources allow readers to directly reproduce the validation procedure and confirm the results presented in the manuscript.

Finally, the present validation was performed by a single operator, which precludes assessment of inter-rater reliability. Future studies should involve multiple users with varying levels of expertise to evaluate user consistency, operator-related bias, and the robustness of the correction process across different evaluators.

5. Conclusions

In this study, we proposed a prototype GUI-based ECG viewer with an embedded algorithm for intuitive R-wave correction. The system enabled immediate updates of the RRI time series and related analyses, with correction errors below 0.7% and update times of 20–30 ms. These results demonstrate reliable, immediate feedback that may be applicable in future contexts.

Furthermore, because the proposed algorithm allows manual extraction of arbitrary time points on the screen, it may be extended to extract not only RRIs but also P-P and Q-Q intervals. Such an extension may further support applications in research, clinical, and educational domains of biosignal processing.

Supplementary Materials: The following supporting information can be downloaded at the website of this paper posted on Preprints.org, Video S1: Demonstrating the ECG viewer; File S2: synthetic ECG signal used for verification; File S3: RRI_GUI vs gold standard _2s_5s_10s.

Name	Type	Description
S1	Video (.mp4)	Demonstration of the GUI-based ECG viewer in operation (90 s).
S2	File (.csv)	Synthetic ECG signals used for validation at four sampling frequencies (125, 250, 500, and 1000 Hz). The file contains columns for time, ECG amplitude, and an R-wave peak flag.
S3	File (.csv)	This file contains the simulated RRI (gold standard) as well as the RRI generated by the GUI at three-time scales (2, 5, and 10 s).

Author Contributions: Conceptualization, Y.Y. ; methodology, Y.Y.; software, Y.Y.; validation, Y.Y., K.Y. ; formal analysis, Y.Y.; investigation, Y.Y.; resources, K.Y.; data curation, Y.Y.; writing—original draft preparation, Y.Y.; writing—review and editing, Y.Y.; visualization, Y.Y.; supervision, K.Y.; project administration, K.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article and Supplementary Materials. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare that they have no competing interests.

Abbreviations

The following abbreviations are used in this manuscript:

- ECG Electrocardiogram
- GUI Graphical User Interface
- HPC High Performance Computing
- HPF High Performance Fortran
- HRV Heart Rate Variability
- RRI R–R Interval

References

1. Ahmad, M.; Ahmed, A.; Hashim, H.; Farsi, M.; Mahmoud, N. Enhancing Heart Disease Diagnosis Using ECG Signal Reconstruction and Deep Transfer Learning Classification with Optional SVM Integration. *Diagnostics* **2025**, *15*, 1501.
2. Nechita, L.C.; Nechita, A.; Voipan, A.E.; Voipan, D.; Debita, M.; Fulga, A.; Fulga, I.; Musat, C.L. AI-Enhanced ECG Applications in Cardiology: Comprehensive Insights from the Current Literature with a Focus on COVID-19 and Multiple Cardiovascular Conditions. *Diagnostics* **2024**, *14*, 1839.
3. Grégoire, J.M.; Gilon, C.; Carlier, S.; Bersini, H. Autonomic Nervous System Assessment Using Heart Rate Variability. *Acta Cardiol.* **2023**, *78*, 648–662.
4. Brazdzionyte, R.; Velicka, D.; Motiejunaite, K.; Poderiene, K.; Kairiukstiene, Z. Advanced ECG Analysis to Evaluate Multimodal Exercise Effects on Cardiovascular Health. *Medicina* **2025**, *61*, 473.
5. Beltrame, J.F.; La, S.; Marathe, J. Clinical Utility of the Humble Exercise ECG Stress Test. *J. Am. Coll. Cardiol.* **2024**, *83*, 300–302.
6. Peabody, J.E.; Ryznar, R.; Ziesmann, M.T.; Gillman, L. A Systematic Review of Heart Rate Variability as a Measure of Stress in Medical Professionals. *Cureus* **2023**, *15*, e34345.
7. Eddie, D.; Bentley, K.H.; Bernard, R.; Mischoulon, D.; Winkelman, J.W. Aggregating Heart Rate Variability Indices Across Sleep Stage Epochs Ignores Significant Variance Through the Night. *Sleep Med.* **2022**, *90*, 262–266.
8. Hebisz, R.G.; Hebisz, P.; Zatoń, M.W. Heart Rate Variability After Sprint Interval Training in Cyclists and Implications for Assessing Physical Fatigue. *J. Strength Cond. Res.* **2022**, *36*, 558–564.
9. Moura-Tonello, S.C.G.; Carvalho, V.O.; Godoy, M.F.; Porta, A.; Leal, Â.M.O.; Bocchi, E.A.; Catai, A.M. Evaluation of Cardiac Autonomic Modulation Using Symbolic Dynamics After Cardiac Transplantation. *Braz. J. Cardiovasc. Surg.* **2019**, *34*, 572–580.
10. Santos, M.C.S.; Silveira, L.C.L.; Moura-Tonello, S.C.G.; Porta, A.; Catai, A.M.; Souza, G.D.S. Heart Rate Variability in Multibacillar Leprosy: Linear and Nonlinear Analysis. *PLoS One* **2017**, *12*, e0180677.
11. Georgieva-Tsaneva, G.; Gospodinova, E. Heart Rate Variability Analysis of Healthy Individuals and Patients with Ischemia and Arrhythmia. *Diagnostics* **2023**, *13*, 2549.
12. Hargens, T.A.; Chambers, S.; Luden, N.D.; Womack, C.J. Reliability of the Heart Rate Variability Threshold During Treadmill Exercise. *Clin. Physiol. Funct. Imaging* **2022**, *42*, 292–299.
13. Kumar, S.M.; Vaishali, K.; Maiya, G.A.; Shivashankar, K.N.; Shashikiran, U. Analysis of Time-Domain Indices and Frequency-Domain Measures of Heart Rate Variability Derived from ECG Waveform and Pulse-Wave-Related HRV Among Overweight Individuals: An Observational Study. *F1000Res.* **2023**, *12*, 1229.
14. Miyagi, T.; Yamazato, M.; Nakamura, T.; Tokashiki, T.; Namihira, Y.; Kokuba, K.; Ishihara, S.; Sakima, H.; Ohya, Y. Power Spectral Analysis of Heart Rate Variability as a Screening Tool for Detecting Sympathetic and Parasympathetic Nervous Dysfunctions in Parkinson's Disease. *BMC Neurol.* **2022**, *22*, 339.
15. Pan, J.; Tompkins, W.J. A Real-Time QRS Detection Algorithm. *IEEE Trans. Biomed. Eng.* **1985**, *32*, 230–236.
16. Neri, L.; Oberdier, M.T.; Augello, A.; Suzuki, M.; Tumarkin, E.; Jaipalli, S.; Geminiani, G.A.; Halperin, H.R.; Borghi, C. Algorithm for Mobile Platform-Based Real-Time QRS Detection. *Sensors* **2023**, *23*, 1625.
17. Agostinelli, A.; Marcantoni, I.; Moretti, E.; Sbröllini, A.; Fioretti, S.; Di Nardo, F.; Burattini, L. Noninvasive Fetal Electrocardiography Part I: Pan-Tompkins' Algorithm Adaptation to Fetal R-Peak Identification. *Open Biomed. Eng. J.* **2017**, *11*, 17–24.
18. Yun, D.; Lee, H.C.; Jung, C.W.; Kwon, S.; Lee, S.R.; Kim, K.; Kim, Y.S.; Han, S.S. Robust R-Peak Detection in an Electrocardiogram with Stationary Wavelet Transformation and Separable Convolution. *Sci. Rep.* **2022**, *12*, 19638.
19. Chen, H.; Maharatna, K. An Automatic R and T Peak Detection Method Based on the Combination of Hierarchical Clustering and Discrete Wavelet Transform. *IEEE J. Biomed. Health Inform.* **2020**, *24*, 2825–2832.

20. Rizwan, A.; Priyanga, P.; Abualsauod, E.H.; Zafrullah, S.N.; Serbaya, S.H.; Halifa, A. A Machine Learning Approach for the Detection of QRS Complexes in Electrocardiogram Using Discrete Wavelet Transform Algorithm. *Comput. Intell. Neurosci.* **2022**, *2022*, 9023478.
21. Kenny, B.J.; Brown, K.N. ECG T Wave. *StatPearls*, 2022. Available online: <https://www.ncbi.nlm.nih.gov/books/NBK538264/> (accessed on 20 August 2025).
22. Tarvainen, M.P.; Niskanen, J.P.; Lipponen, J.A.; Ranta-Aho, P.O.; Karjalainen, P.A. Kubios HRV: Heart Rate Variability Analysis Software. *Comput. Methods Programs Biomed.* **2014**, *113*, 210–220.
23. McConnell, M.; Schwerin, B.; So, S.; Richards, B. RR-APET: Heart Rate Variability Analysis Software. *Comput. Methods Programs Biomed.* **2020**, *185*, 105127.
24. Pilia, N.; Nagel, C.; Lenis, G.; Becker, S.; Dössel, O.; Loewe, A. ECGdeli: An Open Source ECG Delineation Toolbox for MATLAB. *SoftwareX* **2021**, *13*, 100639.
25. Mayor, D.; Panday, D.; Kandel, H.K.; Steffert, T.; Banks, D. CEPS: An Open Access MATLAB Graphical User Interface for the Analysis of Complexity and Entropy in Physiological Signals. *Entropy* **2021**, *23*, 321.
26. Behar, J.A.; Rosenberg, A.A.; Weiser-Bitoun, I.; Shemla, O.; Alexandrovich, A.; Konyukhov, E.; Yaniv, Y. PhysioZoo: A Novel Open Access Platform for Heart Rate Variability Analysis of Mammalian Electrocardiographic Data. *Front. Physiol.* **2018**, *9*, 1390.
27. McSharry, P.E.; Clifford, G.D.; Tarassenko, L.; Smith, L.A. A Dynamical Model for Generating Synthetic Electrocardiogram Signals. *IEEE Trans. Biomed. Eng.* **2003**, *50*, 289–294.
28. Vest, A.N.; Li, Q.; Liu, C.; Nemati, S.; Shah, A.; Clifford, G.D. Benchmarking Heart Rate Variability Toolboxes. *J. Electrocardiol.* **2017**, *50*, 744–747.
29. Vest, A.N.; Da Poian, G.; Li, Q.; Liu, C.; Nemati, S.; Shah, A.J.; Clifford, G.D. An Open Source Benchmarked Toolbox for Cardiovascular Waveform and Interval Analysis. *Physiol. Meas.* **2018**, *39*, 105004.
30. Moody, G.B. LightWAVE: Waveform and Annotation Viewing and Editing in a Web Browser. *Comput. Cardiol.* **2013**, *40*, 17–20.
31. Kulikov, I.; Chernykh, I.; Karavaev, D.; Prigarin, V.; Sapetina, A.; Ulyanichev, I.; Zavyalov, O. A New Parallel Code Based on a Simple Piecewise Parabolic Method for Numerical Modeling of Colliding Flows in Relativistic Hydrodynamics. *Mathematics* **2022**, *10*, 1865.
32. Kim, B.; Yoon, K.S.; Kim, H.-J. GPU-Accelerated Laplace Equation Model Development Based on CUDA Fortran. *Water* **2021**, *13*, 3435.
33. Kan, Y.; Liu, X.; Liu, H. An Image and State Information-Based PINN with Attention Mechanisms for the Rapid Prediction of Aircraft Aerodynamic Characteristics. *Aerospace* **2025**, *12*, 434.
34. Aruoba, B.; Fernández-Villaverde, J. A Comparison of Programming Languages in Macroeconomics. *J. Econ. Dyn. Control* **2015**, *58*, 265–273.
35. Markus, A.; Mullin, L. Implementing Mathematics of Arrays in Modern Fortran: Efficiency and Efficacy. *Software* **2024**, *3*, 534–548.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.