EUROPEAN JOURNAL OF PURE AND APPLIED MATHEMATICS
2025, Vol. 18, Issue 3, Article Number 6237
ISSN 1307-5543 – ejpam.com
Published by New York Business Global

Mean-Driven Accuracy: A Contraharmonic and Centroidal Extension of Heun's Method

Syamsudhuha Syamsudhuha[1,*], M Imran[1], Ayunda Putri[1], Rike Marjulisa[1]

[1] *Department of Mathematics, Faculty of Mathematics and Natural Sciences, University of Riau, Pekanbaru 28293, Riau Province, Indonesia*

Abstract. This paper presents a novel modification of Heun's method for solving initial value problems in ordinary differential equations (ODEs) by incorporating contraharmonic and centroidal means into the corrector step. Unlike traditional Heun's method, which relies on the arithmetic means to average the slopes, our approach generalizes this averaging using nonlinear means that better capture the curvature and dynamics of the solution trajectory. Numerical simulations demonstrate that the proposed method offers improved accuracy and enhanced stability, particularly in stiff or rapidly changing systems. Applications include Newton's law of cooling under standard and extreme thermal conditions, where our method consistently maintains accuracy and robustness. The results suggest that contraharmonic and centroidal means provide a viable and efficient alternative to conventional averaging strategies in explicit predictor–corrector methods.

2020 Mathematics Subject Classifications: 65L04, 65L05, 65L20

Key Words and Phrases: Initial value problems, Heun's method, contraharmonic mean, centroidal mean

---

1. Introduction

Differential equations, particularly those describing initial value problems (IVPs),

$$\begin{aligned} y' &= f(x,y) \\ y(x_0) &= y_0, \end{aligned} \tag{1}$$

are fundamental tools in modeling dynamic systems across a wide array of scientific and engineering disciplines. These equations express the relationships between functions and their derivatives, serving as a mathematical framework for analyzing processes that evolve over time. From fluid dynamics and population models in biology to electrical circuits

*Corresponding author.
DOI: https://doi.org/10.29020/nybg.ejpam.v18i3.6237

Email addresses: syamsudhuha@unri.ac.id (S. Syamsudhuha),
mimran@unri.ac.id (M Imran)ayundaputri@lecturer.unri.ac.id (A. Putri)
rikemarjulisa@lecturer.unri.ac.id (R. Marjulisa)

and modeling water storage and hydrology, IVPs are crucial in understanding real-world phenomena.

Solving initial value problems analytically is often impractical due to the complexity of the underlying differential equations. This has driven the development of various numerical methods that approximate solutions by discretizing the problem over a finite set of points. Numerical methods are particularly useful when the differential equation is non-linear, or when the exact solution is not readily available.

Among the classical numerical methods for solving IVPs are Euler's method, Heun's method (also known as the improved Euler method), and the fourth-order Runge-Kutta (RK4) method [1, 2]. Euler's method, while straightforward and easy to implement, is often limited by poor stability and accuracy unless very small time steps are used [2]. To improve upon these limitations, carious modifications have been proposed. For instance, Workie [3] introduced a modified Euler method that integrates Euler's approximation into the correction phase, enhancing its efficiency. Heun's method has also been modified by altering the interior function to improve its order of accuracy [4], and further refinements such as the incorporation of contraharmonic means have been explored to improve its stability and convergence [5].

Runge-Kutta methods, expecially higher-order variants, offer a more favorable trade-off between accuracy and computational cost. Numerous adaptations have been developed using weighted combinations of means - including arithmetic, harmonic, geometric, Lehmer, and centroidal - to improve the performance of such methods [6–10].

Several studies have emphasized the importance of selecting or designing numerical schemes on the trade-offs between local error, convergence rate, and computational stability. Comparative analysis, such as in [11] have demonstrated that among classical methods like Euler, Runge-Kutta of order fourth and sixth (RK4 and RK6), and Adams-Bashforth-Moulton, the latter generally provides superior accuracy. Other approaches, such as those in [12], solve second–order IVPs without restrictive assumptions using RK4 and Euler methods, thereby offering more generalizable frameworks.

In addition to classical methods, recent attention has turned to solving fuzzy differential equations (FDEs), which are critical in modeling uncertain systems in engineering, economics, and environmental sciences. Techniques developed by [13] and [14] introduced enhanced Euler-type methods using contraharmonic, harmonic, cubic, and centroidal means to handle fuzzy initial conditions, achieving improved numerical accuracy.

The stability and consistency of numerical methods are critical in practical applications, particularly in aerospace, climatology, and finance where error can lead to significant negative impact. Accordingly, numerical researchers continue to pursue improvements in convergence rates, local truncation error minimization, and expanded stability regions [15–18].

In classical Heun's method, the correction step uses the arithmetic mean of slopes to refine the predicted value. This averaging acts as a symmetric compromise between the initial and predicted slopes. Recent innovations in Heun's method further demonstrate the benefits of integrating nonlinear averaging mechanisms. For example, [5] enhanced Heun's method using combination of the arithmetic and contraharmonic means. Inspired by this, our current study introduces a new modification of Heun's method that employs both the contraharmonic mean and the centroidal mean in a unified slope approximation. This approach preserves the predictor-corrector structure while improving accuracy and enhancing stability under stiff or rapidly changing conditions. We provide theoretical justifications for the consistency and second-order convergence of the method and assess its stability region using Dahlquist problem [15]. Through comparative numerical experiments with existing methods such as Heun's method, the harmonic–Heun method, and the contraharmonic–Heun method, we evaluate the performance of the proposed method in solving linear, stiff, and thermally driven IVPs.

2. Development and Analysis of the Proposed Method

2.1. Derivation of the Proposed Method

One of the methods to solve equation (1) is Heun's method (HM), given by

$$y_{i+1}^{c} = y_i + \frac{h}{2}\left[f(x_i, y_i) + f(x_{i+1}, y_{i+1}^{p}\right], \tag{2}$$

for $i = 0, 1, 2, \ldots$ and $h > 0$, with

$$y_{i+1}^{p} = y_i + hf(x_i, y_i). \tag{3}$$

Iterative algorithm given by (3) is also known as Euler's method. In Heun's method, the Euler's method is employed as predictor step while the average of the slope is called the corrector step. One of the developments of Heun's method can be found in [6] where the slope in (2) is altered with second order harmonic mean, as the following:

$$y_{i+1} = y_i + h\left(\frac{\gamma_1^2 + \gamma_2^2}{\gamma_1 + \gamma_2}\right), \tag{4}$$

where

$$\gamma_1 = f(x_i, y_i) \tag{5}$$

$$\gamma_2 = f(x_{i+1}, y_{i+1}^{p}) = f\left(x_i + h, y_i + hf(x_i, y_i)\right), \tag{6}$$

and will be called as HHM for the rest of this article. Furthermore, in [5], Heun's method is developed by assuming that the slope is equal the average of the second-order contraharmonic mean and the arithmetic mean, which is given by

$$y_{i+1} = y_i + \frac{h}{2}\left(\frac{\gamma_1 + \gamma_2}{2} + \frac{\gamma_1^2 + \gamma_2^2}{\gamma_1 + \gamma_2}\right). \tag{7}$$

The method described by (7) will be referred to as CAM from this point on.

Inspired by the modification done in [5], we make the assumption that the slope is the average of the contraharmonic mean and centroidal mean. Hence, our proposed method is given by

$$y_{i+1}^c = y_i + \frac{h}{2}\left(\frac{\gamma_1^2+\gamma_2^2}{\gamma_1+\gamma_2} + \frac{2\left(\gamma_1^2+\gamma_1\gamma_2+\gamma_2^2\right)}{3\left(\gamma_1+\gamma_2\right)}\right), \tag{8}$$

where γ_1 and γ_2 are (5) and (6) respectively. The method defined by equation (8) will be referred to as CCH for the rest of the article.

The following subsections provide the stability, consistency analysis and the local truncation error analysis of the suggested method.

2.2. Stability Analysis

To analyze the stability of the CCH, we employ Dahlquist's test problem [15], defined by

$$\frac{dy}{dx} = \lambda y; \quad y(x_0) = 1, \quad \lambda \in \mathbb{C}. \tag{9}$$

The stability of CCH is satisfied when

$$y(x_i) \longrightarrow 0 \quad \text{as} \quad i \longrightarrow \infty, \tag{10}$$

for any stepsize $h > 0$.

We begin the stability analysis of CCH by substituting (9) into (8) and simplifying, which results in

$$\begin{aligned}
y_{i+1}^c &= y_i + \frac{h}{2}\left[\frac{(\lambda y_i)^2 + \left(\lambda y_{i+1}^p\right)^2}{\lambda y_i + \lambda y_{i+1}^p} + \frac{2\left(\lambda y_i^2 + \lambda y_i \lambda y_{i+1}^p + \left(\lambda y_{i+1}^p\right)^2\right)}{3\left(\lambda y_i + \lambda y_{i+1}^p\right)}\right] \\
&= y_i + \frac{h}{2}\Bigg[\frac{(\lambda y_i)^2 + (\lambda\left(y_i + h\lambda y_i\right))^2}{\lambda y_i \lambda\left(y_i + h\lambda y_i\right)} \\
&\quad + \frac{2\left((\lambda y_i)^2 + \lambda y_i\left(\lambda\left(y_i + h\lambda y_i\right)\right) + \left(\lambda\left(y_i + h\lambda y_i\right)\right)^2\right)}{3\left(\lambda y_i + \left(\lambda\left(y_i + h\lambda y_i\right)\right)\right)}\Bigg] \\
y_{i+1}^c &= \left[1 + \frac{5\left(h^3\lambda^3 + 2h^2\lambda^2 + 2h\lambda\right)}{6\left(h\lambda + 2\right)}\right] y_i.
\end{aligned} \tag{11}$$

After running the iteration for $i = 0, 1, 2, \ldots, n$, for any large integer n, we obtain

$$y_n^c = \left[1 + \frac{5\left(h^3\lambda^3 + 2h^2\lambda^2 + 2h\lambda\right)}{6\left(h\lambda + 2\right)}\right]^n. \tag{12}$$

Letting $z = h\lambda$, the stability function becomes

$$y_n^c = \left(1 + \frac{5\left(z^3 + 2z^2 + 2z\right)}{6\left(z + 2\right)}\right)^n, \tag{13}$$

and the stability region is

$$|G(z)| = \left| 1 + \frac{5\left(z^3 + 2z^2 + 2z\right)}{6\left(z+2\right)} \right| \leq 1. \tag{14}$$

The following figure shows the stability region of CCH method, which was determined by evaluating the method over a grid in the complex plane. The horizontal axis represents the real part of the complex variable z, $Re(z)$, ranging from -3 to 2, while the the vertical axis represents the imaginary part, $Im(z)$, ranging from -3 to 3. In this plot, the white (unshaded) region represents the stability region ($G(z)$) of CCH method. This is the set of all complex value $z \in \mathbb{C}$ for which the numerical method remains stable when applied to the test equation (9). The green shaded region corresponds to values of z where the method is unstable.

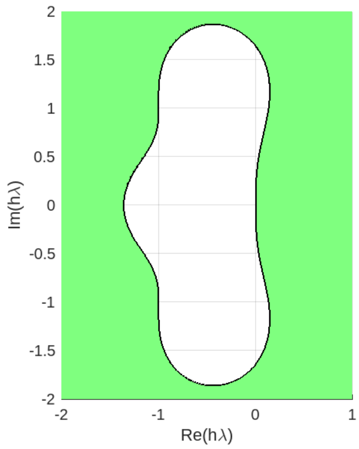

Figure 1: Stability region of CCH method

2.3. Consistency Analysis

According to [15], equation (3) can be written in the form of

$$y_{i+1} = y_i + h\phi(x_i, y_i, h) \tag{15}$$

and the consistency is preserved if

$$\lim_{h\to 0} \phi(x_i, y_i, h) = f(x_i, y_i) \tag{16}$$

Equating (8) with (15), yields

$$\phi(x_i, y_i, h) \quad = \quad \frac{1}{2}\left(\frac{\gamma^2 + \gamma_2^2}{\gamma_1 + \gamma_2} + \frac{2\left(\gamma_1^2 + \gamma_1\gamma_2 + \gamma_2^2\right)}{3\left(\gamma_1 + \gamma_2\right)}\right)$$

$$\phi(x_i, y_i, h) = \frac{1}{2}\left(\frac{f(x_i, y_i)^2 + f(x_{i+1}, y^p_{i+1})^2}{f(x_i, y_i) + f(x_{i+1}, y^p_{i+1})}\right. + \frac{f(x_i, y_i)^2 + f(x_i, y_i)f(x_{i+1}, y^p_{i+1}) + f(x_{i+1}, y^p_{i+1})^2}{3f(x_i, y_i) + 3f(x_{i+1}, y^p_{i+1})} \tag{17}$$

where $y^p_{i+1} = f(x_i + h, y_i + f(x_i, y_i))$.

Expanding $f(x_i, y_i)$ by Taylor expansion and taking the limit of equation (17) for $h \to 0$, yields

$$\begin{aligned}\lim_{h\to 0} \phi(x_i, y_i, h) &= \lim_{h\to 0} \frac{1}{2}\left[2f + h\left(f - xf\right)\right] + \left[f_{xy}f + \frac{2}{3}f_x f_y + \frac{1}{2b}f_{yy}f^2\right. \\ &\quad \left. + \frac{1}{3}ff_y^2 + \frac{1}{3}\frac{f_x^2}{f} + \frac{1}{2}f_{xx}\right]h^2 \\ \lim_{h\to 0} \phi(x_i, y_i, h) &= f, \end{aligned} \tag{18}$$

where $f = f(x_i, y_i)$, $f_x = \dfrac{\partial f}{\partial x}$, $f_y = \dfrac{\partial f}{\partial y}$, $f_{xx} = \dfrac{\partial^2 f}{\partial x^2}$, $f_{yy} = \dfrac{\partial^2 f}{\partial y^2}$, $f_{xy} = \dfrac{\partial^2 f}{\partial x \partial y}$.

We conclude that the proposed method is consistent.

2.4. Local Truncation Error

The local truncation error (LTE) quantifies the error introduced in a single step of a numerical method under the assumption that all prior steps are exact. It serves as a key measure in evaluating the accuracy and convergence order of numerical integration schemes. To analyze the LTE of the proposed CCH method, we compare the exact Taylor expansion of the true solution $y(x_{i+1})$ with the numerical approximation y_{i+1} generated by the method. The difference between these expressions, when expanded in powers of the step size h reveals the leading error term and thereby the method's order of accuracy. Consider

$$y'(x) = f(x, y(x))$$

The Taylor expansion of $y(x_{i+1})$ is

$$\begin{aligned} y(x_{i+1}) &= y(x_i + h) \qquad (19) \\ &= y(x_i) + hy'(x_i) + \frac{h^2}{2}y''(x_i) + \frac{h^3}{3!}y'''(x_i) + O(h^4), \\ y(x_{i+1}) &= y(x_i) + hf + \frac{h^2}{2}\left[f_x + ff_y\right] + \frac{h^3}{3!}\left[f_{xx} + 2ff_{xy} + f^2 f_{yy} + ff_y^2 + f_x f_y\right] \\ &\quad + O(h^4), \qquad (20) \end{aligned}$$

and the Taylor expansions of γ_1 and γ_2 are

$$\gamma_1 = f(x_i, y_i) = f \tag{21}$$

$$\begin{aligned}
\gamma_2 &= f(x_{i+1}, y_{i+1}^p = f(x_{i+1}, y_i + hf(x_i, y_i)) \\
\gamma_2 &= f + h(f_x + ff_y) + \frac{h^2}{2}\left[f_{xx} + 2ff_{xy} + f^2 f_{yy}\right]. \qquad (22)
\end{aligned}$$

By using (21) and (22), the sum of the contraharmonic and centroidal means becomes:

$$\begin{aligned}
\frac{\gamma_1^2 + \gamma_2^2}{\gamma_1 + \gamma_2} + \frac{2\gamma_1\gamma_2}{\gamma_1 + \gamma_2} &= 2f + h(f_x + ff_y) + \frac{h^2}{2}\left(f^2 f_{yy} + 2ff_{xy}f_{xx}\right) \\
&\quad + O(h^3). \qquad (23)
\end{aligned}$$

Substituting (23) to (8) and simplifying, yields

$$\begin{aligned}
y_{i+1} &= y_i + 2f + +h(f_x + ff_y) + h^2\Bigg(f_{xy}f + \frac{2}{3}f_x f_y + \frac{1}{2}f_{yy}f^2 \\
&\quad + \frac{1}{2}f_{xx} + \frac{1}{3}ff_y^2 \frac{1}{3}\frac{f_x^2}{f}\Bigg) + O(h^3). \qquad (24)
\end{aligned}$$

By comparing equations (20) and (24), we obtain the local truncation error as follows:

$$\begin{aligned}
LTE &= y(x_{i+1}) - y_{i+1} \\
LTE &= h^3\left(\frac{-1}{12}f_{yy}f^2 - \frac{1}{6}f_{xy}f - \frac{1}{6}f_x f_y - \frac{1}{12}f_{xx} - \frac{1}{6}\frac{f_x^2}{f}\right) + O(h^3). \qquad (25)
\end{aligned}$$

Consequently, the proposed method is of second order and has a local truncation error of order $O(h^3)$, which is smaller than that of Heun's method ($O(h^2)$) [1].

3. Numerical Results and Discussion

In this section, we implement and compare the performance of the four methods namely HM, HHM, CAM, and the proposed method CCH to several IVPs. The comparison is carried out by reporting the error produced by each method for three different step-sizes: $h = 0.1, h = 0.01$, and $h = 0.001$. Additionally, we include the run time (in second) for each method to evaluate the computational time required during the simulations.

Example 1.

$$\begin{aligned}
\frac{dy}{dx} &= x^3 e^{-2x} - 2y \\
y(0) &= 1
\end{aligned}$$

Example 2.

$$\begin{aligned}
\frac{dy}{dx} &= -100y + e^{-2x} \\
y(0) &= 0
\end{aligned}$$

Example 3. *Consider an insulated box with internal temperature $y(t)$. According to Newton's law of cooling, y satisfies the differential equation*

$$\frac{dy}{dt} = -\alpha(y(t) - T(t)) \tag{26}$$

where $T(t)$ is the external temperature. Suppose that $T(t)$ varies sinusoidally, for example, assume that $T(t) = T_0 + T_1\cos(\omega t)$ [19] with $T(0) = 32°F$. Set $w = \pi/12$, which is corresponding to a period of 24 hour for $T(t)$, and let $T_0 = 60°F$, $T_1 = 15°F$, and $\alpha = 0.2/hour$. We are looking for the behavior of the studied method for different step sizes h, for $T(1)$.

Table 1: Error and run time comparison of HM, HHM, CAM and CCH for Example 1

h	$x(i)$	HM	HHM	CAM	CCH
$h = 0.1$	0.1000000000	1.289715e-03	9.015333e-02	1.737973e-04	1.981754e-04
	0.2000000000	2.146271e-03	1.556291e-01	2.948941e-04	3.216807e-04
	0.3000000000	2.674663e-03	2.014410e-01	3.574880e-04	4.135400e-04
	0.4000000000	2.955967e-03	2.316413e-01	3.678603e-04	4.925873e-04
	0.5000000000	3.053693e-03	2.495483e-01	3.376646e-04	5.645702e-04
	0.6000000000	3.018016e-03	2.578993e-01	2.800325e-04	6.287749e-04
	0.7000000000	2.888664e-03	2.589557e-01	2.073892e-04	6.819157e-04
	0.8000000000	2.696980e-03	2.545802e-01	1.303383e-04	7.203185e-04
	0.9000000000	2.467462e-03	2.462961e-01	5.719880e-05	7.410556e-04
	1.0000000000	2.218966e-03	2.353362e-01	6.062800e-06	7.424430e-04
	run time	0.001	0.001	0.001	0.001
$h = 0.01$	0.1000000000	1.128470e-05	8.654601e-02	2.808500e-06	1.680000e-08
	0.2000000000	1.879290e-05	1.490950e-01	4.739300e-06	5.490000e-08
	0.3000000000	2.342270e-05	1.925875e-01	5.846100e-06	1.240000e-08
	0.4000000000	2.587680e-05	2.210016e-01	6.261000e-06	2.774000e-07
	0.5000000000	2.671070e-05	2.375831e-01	6.142900e-06	7.129000e-07
	0.6000000000	2.636750e-05	2.449999e-01	5.651900e-06	1.253300e-06
	0.7000000000	2.519860e-05	2.454488e-01	4.930700e-06	1.825200e-06
	0.8000000000	2.348370e-05	2.407339e-01	4.100400e-06	2.360500e-06
	0.9000000000	2.143980e-05	2.323266e-01	3.255300e-06	2.806300e-06
	1.0000000000	1.923480e-05	2.214142e-01	2.464500e-06	3.125600e-06
	run time	0.012	0.007	0.007	0.01
$h = 0.001$	0.1000000000	1.115000e-07	8.614185e-02	2.830000e-08	1.600000e-09
	0.2000000000	1.856000e-07	1.483639e-01	4.810000e-08	3.400000e-09
	0.3000000000	2.315000e-07	1.915968e-01	5.980000e-08	3.700000e-09
	0.4000000000	2.559000e-07	2.198097e-01	6.420000e-08	1.600000e-09
	0.5000000000	2.638000e-07	2.362405e-01	6.330000e-08	2.300000e-09
	0.6000000000	2.609000e-07	2.435497e-01	5.830000e-08	8.000000e-09
	0.7000000000	2.492000e-07	2.439277e-01	5.110000e-08	1.440000e-08
	0.8000000000	2.320000e-07	2.391724e-01	4.280000e-08	1.940000e-08
	0.9090000000	2.100000e-07	2.298541e-01	3.290000e-08	2.420000e-08
	1.0000000000	1.903000e-07	2.198422e-01	2.600000e-08	2.730000e-08
	run time	0.077	0.075	0.076	0.081

Table 2: Error and run time comparison of HM, HHM, CAM and CCH for Example 2

h	$x(i)$	HM	HHM	CAM	CCH
	0.1000	4.058258e+01	5.172037e+01	7.151583e+01	8.182692e+01
	0.2000	1.663887e+03	2.702392e+03	5.167022e+03	6.764362e+03
	0.3000	6.821935e+04	1.412000e+05	3.733173e+05	5.591872e+05
	0.4000	2.796993e+06	7.377699e+06	2.697218e+07	4.622614e+07
$h = 0.1$	0.5000	1.146767e+08	3.854848e+08	1.948740e+09	3.821361e+09
	0.6000	4.701746e+09	2.014158e+10	1.407964e+11	3.158992e+11
	0.7000	1.927716e+11	1.052398e+12	1.017254e+13	2.611433e+13
	0.8000	7.903635e+12	5.498777e+13	7.349662e+14	2.158785e+15
	0.9000	3.240490e+14	2.873111e+15	5.310131e+16	1.784596e+17
	1.0000	1.328601e+16	1.501201e+17	3.836570e+18	1.475266e+19
	run time	0.001	0.001	0.001	0.001
	0.1000	9.687756e-04	1.291157e-03	1.372350e-05	5.826586e-06
	0.2000	2.349522e-06	2.029662e-04	1.358166e-06	1.333645e-06
	0.3000	1.151696e-06	1.634442e-04	1.104609e-06	1.089625e-06
	0.5000	7.713880e-07	1.095502e-04	7.404370e-07	7.303960e-07
$h = 0.01$	0.6000	6.315580e-07	8.969216e-05	6.062180e-07	5.979970e-07
	0.7000	5.170770e-07	7.343373e-05	4.963300e-07	4.895990e-07
	0.8000	4.233470e-07	6.012245e-05	4.063610e-07	4.008500e-07
	0.9000	3.466070e-07	4.922410e-05	3.327000e-07	3.281880e-07
	1.0000	2.837770e-07	4.030128e-05	2.723920e-07	2.686970e-07
	run time	0.006	0.005	0.01	0.011
	0.1000	4.622350e-05	7.699141e-03	4.566264e-05	4.547745e-05
	0.2000	9.438000e-09	2.070950e-04	9.328000e-09	9.291000e-09
	0.3000	5.977000e-09	1.232044e-04	5.911000e-09	5.889000e-09
	0.4000	4.894000e-09	1.005181e-04	4.839000e-09	4.821000e-09
$h =$ 0.001	0.5000	4.006000e-09	8.229458e-05	3.961000e-09	3.946000e-09
	0.6000	3.280000e-09	6.737708e-05	3.244000e-09	3.232000e-09
	0.7000	2.686000e-09	5.516369e-05	2.657000e-09	2.647000e-09
	0.8000	2.199000e-09	4.516421e-05	2.174000e-09	2.166000e-09
	0.9000	1.801000e-09	3.697733e-05	1.781000e-09	1.774000e-09
	1.0000	1.475000e-09	3.027447e-05	1.458000e-09	1.453000e-09
	run time	0.047	0.047	0.196	0.181

3.1. Discussion

The numerical results presented in Table 1, Table 2, Table 3, and Table 4 compare the performance of four numerical methods HM, HHM, CAM, CCH, for solving the initial value problems in Example 1, Example 2, Example 3, and Example 3 with parameter changing $\alpha = 3$ respectively. These methods were applied using three different step sizes: $h = 0.1, h = 0.01$, and $h = 0.001$. The analysis focuses on how the accuracy of each method varies with step size and problem complexity as well as the run time of each method.

In general, all methods show improved accuracy as the step size decreases. This is consistent with theoretical expectations since smaller step sizes reduce the local truncation error. The improvement is particularly notable when comparing results at $h = 0.1$ and $h = 0.001$. While HM, CAM, and CCH tend to show this expected behavior clearly, HHM appears to be less stable and often yields larger errors, especially for larger step sizes.

For Example 1, which seems relatively well-behaved, all methods perform reasonably

Table 3: Error and run time comparison of HM, HHM, CAM and CCH for Example 3

h	$x(i)$	HM	HHM	CAM	CCH
$h = 0.1$	0.1000000000	7.435000e-05	4.257047e-01	3.039000e-05	1.574000e-05
	0.2000000000	1.467700e-04	8.386625e-01	5.953000e-05	3.045000e-05
	0.3000000000	2.172800e-04	1.239067e+00	8.741000e-05	4.412000e-05
	0.4000000000	2.859100e-04	1.627108e+00	1.140200e-04	5.673000e-05
	0.5000000000	3.527200e-04	2.002973e+00	1.393900e-04	6.829000e-05
	0.6000000000	4.177200e-04	2.366847e+00	1.635000e-04	7.877000e-05
	0.7000000000	4.809400e-04	2.718911e+00	1.863600e-04	8.818000e-05
	0.8000000000	5.424200e-04	3.059345e+00	2.079700e-04	9.650000e-05
	0.9000000000	6.021800e-04	3.388325e+00	2.283200e-04	1.037100e-04
	1.0000000000	6.602400e-04	3.706024e+00	2.474000e-04	1.098000e-04
	run time	0.001	0.001	0.001	0.004
$h = 0.01$	0.1000000000	7.300000e-07	4.237957e-01	3.000000e-07	1.700000e-07
	0.2000000000	1.460000e-06	8.348833e-01	5.900000e-07	3.200000e-07
	0.3000000000	2.150000e-06	1.233456e+00	8.800000e-07	4.800000e-07
	0.4000000000	2.820000e-06	1.619705e+00	1.140000e-06	6.000000e-07
	0.5000000000	3.470000e-06	1.993816e+00	1.390000e-06	7.200000e-07
	0.6000000000	4.110000e-06	2.355974e+00	1.630000e-06	8.300000e-07
	0.7000000000	4.730000e-06	2.706361e+00	1.880000e-06	9.300000e-07
	0.8000000000	5.360000e-06	3.045156e+00	2.120000e-06	1.010000e-06
	0.9000000000	5.960000e-06	3.372534e+00	2.320000e-06	1.090000e-06
	1.0000000000	6.530000e-06	3.688671e+00	2.500000e-06	1.160000e-06
	run time	0.007	0.007	0.022	0.028
$h = 0.001$	0.1000000000	6.000000e-08	4.236042e-01	5.000000e-08	5.000000e-08
	0.2000000000	1.100000e-07	8.345042e-01	1.000000e-07	1.000000e-07
	0.3000000000	1.000000e-07	1.232893e+00	6.000000e-08	6.000000e-08
	0.4000000000	9.000000e-08	1.618962e+00	5.000000e-08	5.000000e-08
	0.5000000000	1.000000e-07	1.992897e+00	6.000000e-08	6.000000e-08
	0.6000000000	1.100000e-07	2.354883e+00	6.000000e-08	6.000000e-08
	0.7000000000	1.500000e-07	2.705101e+00	9.000000e-08	9.000000e-08
	0.8000000000	1.100000e-07	3.043731e+00	5.000000e-08	5.000000e-08
	0.9000000000	1.100000e-07	3.370949e+00	5.000000e-08	4.000000e-08
	1.0000000000	1.400000e-07	3.686929e+00	6.000000e-08	5.000000e-08
	run time	0.143	0.158	0.286	0.276

well at finer step sizes. At $h = 0.1$, HHM yields significantly larger errors compared to the other methods, indicating potential stability issues or higher local errors. On the other hand, CAM and CCH show better accuracy, with CCH generally outperforming the others at most points. As the step size is reduced to $h = 0.01$, the errors of all methods drop considerably, with CAM and CCH again providing the most accurate approximations. At $h = 0.001$, the differences between methods become less significant, but CCH maintains a slight edge, followed closely by CAM. HM also performs reasonably well, but HHM continues to trail in terms of accuracy.

In contrast, Example 2 appears to be more challenging, as evidenced by the much larger errors observed for $h = 0.1$. In particular, HHM and CAM produce extremely large errors (up to 10^{18}), suggesting that they are unstable or diverge for this problem at large step sizes. HM also shows very large errors, though slightly lower than HHM and CAM.

Table 4: Error and run time comparison of HM, HHM, CAM and CCH for Example 3 with $\alpha = 2$

h	$x(i)$	HM	HHM	CAM	CCH
$h = 0.1$	0.1000000000	6.888063e+00	3.066158e+00	6.935901e+00	6.951847e+00
	0.2000000000	1.239813e+01	5.791583e+00	1.247666e+01	1.250281e+01
	0.3000000000	1.677933e+01	8.207803e+00	1.687604e+01	1.690822e+01
	0.4000000000	2.023592e+01	1.034347e+01	2.034184e+01	2.037705e+01
	0.5000000000	2.293539e+01	1.222460e+01	2.304420e+01	2.308034e+01
	0.6000000000	2.501505e+01	1.387487e+01	2.512243e+01	2.515807e+01
	0.7000000000	2.658749e+01	1.531574e+01	2.669060e+01	2.672478e+01
	0.8000000000	2.774501e+01	1.656676e+01	2.784209e+01	2.787425e+01
	0.9000000000	2.856330e+01	1.764565e+01	2.865339e+01	2.868320e+01
	1.0000000000	2.910440e+01	1.856855e+01	2.918709e+01	2.921444e+01
	run time	0.001	0.001	0.004	0.004
$h = 0.01$	0.1000000000	6.942350e+00	3.221079e+00	6.942713e+00	6.942834e+00
	0.2000000000	1.248718e+01	6.072520e+00	1.248778e+01	1.248798e+01
	0.3000000000	1.688891e+01	8.589842e+00	1.688964e+01	1.688989e+01
	0.4000000000	2.035580e+01	1.080519e+01	2.035660e+01	2.035687e+01
	0.5000000000	2.305836e+01	1.274766e+01	2.305918e+01	2.305946e+01
	0.6000000000	2.513618e+01	1.444357e+01	2.513699e+01	2.513726e+01
	0.7000000000	2.670352e+01	1.591676e+01	2.670430e+01	2.670456e+01
	0.8000000000	2.785392e+01	1.718879e+01	2.785466e+01	2.785490e+01
	0.9000000000	2.866397e+01	1.827918e+01	2.866465e+01	2.866488e+01
	1.0000000000	2.919634e+01	1.920560e+01	2.919697e+01	2.919717e+01
	run time	0.007	0.007	0.022	0.021
$h = 0.001$	0.1000000000	6.942824e+00	3.238457e+00	6.942827e+00	6.942828e+00
	0.2000000000	1.248796e+01	6.103972e+00	1.248797e+01	1.248797e+01
	0.3000000000	1.688986e+01	8.632531e+00	1.688987e+01	1.688987e+01
	0.4000000000	2.035684e+01	1.085669e+01	2.035685e+01	2.035685e+01
	0.5000000000	2.305943e+01	1.280589e+01	2.305944e+01	2.305944e+01
	0.6000000000	2.513723e+01	1.450677e+01	2.513724e+01	2.513724e+01
	0.7000000000	2.670453e+01	1.598344e+01	2.670453e+01	2.670454e+01
	0.8000000000	2.785487e+01	1.725768e+01	2.785488e+01	2.785488e+01
	0.9000000000	2.866484e+01	1.834924e+01	2.866485e+01	2.866485e+01
	1.0000000000	2.919714e+01	1.927593e+01	2.919715e+01	2.919715e+01
	run time	0.062	0.138	0.278	0.284

Interestingly, CCH performs better than the other methods in this case, suggesting that it has superior stability characteristics under these conditions. As the step size is reduced to $h = 0.01$, the errors become much more manageable. CAM and CCH once reduced to $h = 0.01$, the errors become much more manageable. CAM and CCH once again emerge as the most accurate methods, while HHM shows improvement but still lags behind. Finally, at $h = 0.001$, all methods provide highly accurate results. CCH consistently delivers the lowest errors across the board, and CAM closely follows. HM performs adequately, while HHM, although better than at larger step sizes, still does not match the accuracy of the other methods.

Example 3 describes heat transfer within an insulated box governed by Newton's law of cooling where the internal temperature $y(t)$ changes in response to the difference between the internal temperature and a time-dependent external temperature $T(t) =$

$T_0 + T_1 \cos(\omega t)$, simulating a daily cycle with $(\omega = \pi/12)$. Numerical results are obtained for different step sizes $h = 0.1, h = 0.01$, and $h = 0.001$, assuming an initial external temperature $T(0) = 32°F$. The cooling coefficient is set to $\alpha = 0.2$ per hour, introducing moderate stiffness as the step size decreases. The resulting errors from each method indicate their effectiveness in capturing the system's transient thermal dynamics under periodic forcing.

At the largest step size $h = 0.1$, the errors for the HHM method are significantly higher compared to the other methods. For example, at $t = 1$, HHM produces an error of approximately 3.76, while CAM and CCH yield much smaller errors. HM performs better than HHM but worse than CAM and CCH. The results show that HHM struggles to maintain accuracy under larger time steps for this thermally driven problem, likely due to its sensitivity to rapid changes in external forcing. CAM and CCH, which incorporate correction steps, show much better performance and stability under this coarser discretization.

As the step size is reduced to $h = 0.01$, the errors decrease dramatically for all methods. CCH and CA maintain their positions as the most accurate methods, with errors in the order of 10^{-7} to 10^{-6}. HM also improves substantially, with errors dropping into the range of 10^{-6}. However, HHM continues to exhibit much higher errors-on the order of 1-throughout the interval. This, again indicates that HHM not be well-suited for this particular problem setup, especially where the sinusoidal external temperature induces rapid transitions.

As the finest resolution, $h = 0.001$, the results converge and the error differences between methods become less dramatic, except for HHM, which continues to show relatively large errors (up to approximately 3.69 at $t = 1$). Meanwhile, CAM and CCH show consistently low errors around 5×10^{-8}, and HM follows closely behind with slightly higher but still very acceptable errors. The accuracy of CAM and CCH at this fine step size confirms their robustness and reliability for simulating time-dependent thermal processes with periodic behavior. The CCH consistently provides the best performance across all step sizes, followed closely by CAM. These two methods are particularly effective in handling the sinusoidal external temperature and the moderate stiffness introduced by the parameter α. HM performs adequately, especially as the step size becomes smaller, but does not reach the precision of CCH and CAM. The HHM, on the other hand, proves to be unreliable for this example, particularly for coarser step sizes, and should be used cautiously or avoided in similar thermal simulations.

To challenge the proposed method, we change the parameter in Example 3 to be $\alpha = 2$ per hour, which means that the object changes temperature rapidly. The numerical results is presented in the Table 4. For a coarse step size $h = 0.1$, CCH consistently achieves the lowest errors across all evaluated points, with CAM and HHM also outperforming HM. As the step size decreases to $h = 0.01$, the accuracy of all methods improves significantly, and both CCH and CAM yield nearly identical and better results. At the finest resolution, $h = 0.001$, errors become negligible across all methods, but CCH and CAM again maintain better precision. However, this increase accuracy comes at the cost of higher computational time. CCH shows the highest run time of all step sizes, especially at $h = 0.001$, where it requires 0.284 seconds, slightly more than CAM (0.278 seconds), slightly more than HHM

(0.138 seconds) and HM (0.062 seconds). This indicates that CCH method involves more function evaluations or computational steps per iteration reflecting the added complexity of its correction terms.

Overall, the CCH method proves to be the most accurate and stable across varying step sizes, especially for stiff problems, making it well-suited for problems with large cooling coefficients. While it demands more computational effort, its performance justifies the cost when high accuracy is essential. In addition, all the tested methods require two function evaluations per time step. Although the CCH method involves slightly more complex algebraic expressions in its update formula, its cost per step remain comparable to other methods such as HM, HHM, and CAM. Therefore, the increase in computational cost is marginal and justified by the improved accuracy.

4. Conclusion and Outlook

In this study, we have introduced a novel numerical method for solving initial value problems (IVPs) by modifying Heun's method through the integration of contraharmonic and centroidal means. The proposed method, referred to as CCH, was designed to enhance the stability, accuracy, and consistency of Heun's original approach. A comprehensive theoretical analysis confirmed that the method is both consistent and of second-order accuracy, with local truncation error of order $O(h^3)$. The stability region of the CCH method was also established using the Dahlquist test problem, revealing a favorable profile compared to existing methods.

Through extensive numerical experiments on various test problems, including both linear and stiff differential equations, the performance of CCH was bench marked against classical Heun's method (HM), HHM, and CAM. The results consistently demonstrated that the CCH method outperforms the others in terms of accuracy, especially at smaller step sizes. In particular, CCH exhibited excellent stability and precision even for challenging problems, such as those involving rapidly oscillating or stiff systems.

Example 1 showed that while all methods improved with smaller step sizes, CCH and CAM delivered the most accurate results, with HHM performing poorly at larger steps. Example 2, being more sensitive and stiff, further highlighted the robustness of CCH, which maintained stable and accurate results even when other methods diverged. In Example 3, which modeled a physically meaningful scenario based on Newton's law of cooling with sinusoidal external temperature, the CCH method again proved superior in both coarse and fine discretizations, confirming its applicability in real-world simulations involving periodic or time-varying behaviors.

Overall, the proposed CCH method provides a reliable and efficient alternative to existing numerical methods for solving IVPs. Its mean-based formulation not only enhances numerical accuracy but also extends the stability region, making it suitable for a broader class of differential equations. Future work may explore extending the CCH methods to systems of ordinary differential equations (ODEs), applications in partial differential equations (PDEs), or incorporation of adaptive step-size control for further performance optimization in complex dynamic systems.

Acknowledgements

The authors gratefully acknowledge the financial support provided by the Institute for Research and Community Service (LPPM), University of Riau, which has been instrumental in the successful completion of this research.

References

[1] Kendall E Atkinson and Weimin Han. *Elementary numerical analysis.* Wiley New York, 1985.

[2] E Kreyszig. *Advanced Engineering Mathematics.* Wiley, 9 edition.

[3] Abushet Hayalu Workie. Small modification on modified euler method for solving initial value problems. In *Abstract and Applied Analysis*, volume 2021, page 9951815. Wiley Online Library, 2021.

[4] MS Chandio and AG Memon. Improving the efficiency of heun's method. *Sindh University Research Journal-SURJ (Science Series)*, 42(2), 2010.

[5] Abushet Hayalu Workie. New modification on heun's method based on contraharmonic mean for solving initial value problems with high efficiency. *Journal of Mathematics*, 2020(1):6650855, 2020.

[6] Osama Yusuf Ababneh and Rokiah Rozita. New third order runge kutta based on contraharmonic mean for stiff problems. *Applied Mathematical Sciences*, 3(8):365–376, 2009.

[7] Rini Yanti, M Imran, et al. A third runge kutta method based on a linear combination of arithmetic mean, harmonic mean and geometric mean. *Applied and Computational Mathematics*, 10(2):231–234, 2014.

[8] Endah Dwi Jayanti, M Imran, et al. A third order runge-kutta method based on a convex combination of lehmer means. *J. Math. Comput. Sci.*, 8(6):673–682, 2018.

[9] R Gethsi Sharmila, S Suvitha, and M Sarah Sunithy. A third order runge-kutta method based on a linear combination of arithmetic mean, geometric mean and centroidal mean for first order differential equation. In *AIP Conference Proceedings*, volume 2261. AIP Publishing, 2020.

[10] Tulja Ram, Muhammad Anwar Solangi, Wajid Ali Shaikh, and Asif Ali Shaikh. Mean-based efficient hybrid numerical method for solving first-order ordinary differential equations. *Quaid-e-Awam University Research Journal of Engineering Science and Technology*, 20(1):8–12, 2022.

[11] Md Monirul Islam Sumon and Md Nurulhoque. A comparative study of numerical methods for solving initial value problem (ivp) of ordinary differential equations (ode). *American Journal of Applied Mathematics*, 11(6):106–118, 2023.

[12] Yenesew Workineh, Habtamu Mekonnen, and Basaznew Belew. Numerical methods for solving second-order initial value problems of ordinary differential equations with euler and runge-kutta fourth-order methods. *Frontiers in Applied Mathematics and Statistics*, 10:1360628, 2024.

[13] A. Hari Ganesh. A numerical approach for solving fuzzy differential equation us-

ing enhanced euler's methods based on contra harmonic mean and centroidal mean. 32(8):271–284. Number: 8s.

[14] R Balaji, M Saradha, et al. Numerical solutions of fuzzy differential equations by harmonic mean and cubic mean of modified euler's method. *Contemporary Mathematics*, pages 581–591, 2023.

[15] Robert M Corless, C Yalçın Kaya, and Robert HC Moir. Optimal residuals and the dahlquist test problem. *Numerical Algorithms*, 81:1253–1274, 2019.

[16] Noori Y Abdul-Hassan, Zainab J Kadum, and Ali Hasan Ali. An efficient third-order scheme based on runge–kutta and taylor series expansion for solving initial value problems. *Algorithms*, 17(3):123, 2024.

[17] Mohammad Asif Arefin, Biswajit Gain, Rezaul Karim, and Saddam Hossain. A comparative exploration on different numerical methods for solving ordinary differential equations. *J. Mech. Cont. & Math. Sci*, 15(12):1–11, 2020.

[18] Md Kamruzzaman and Mithun Chandra Nath. A comparative study on numerical solution of initial value problem by using euler's method. *Journal of computer and mathematical sciences*, 9(5):493–500, 2018.

[19] William E Boyce, Richard C DiPrima, and Douglas B Meade. *Elementary differential equations and boundary value problems*. John Wiley & Sons, 2017.