Nested Angular Progress, Galactic Phase-Space Addressing, and Earth-Language Chronometric Labels

A Gravity-Independent Phase-Geometric Framework from Local Rotating Bodies to a Milky Way Atlas

S. M. H. Emamifar

Independent Research Collaboration on
Black Hole and Cosmology Concepts (IRCBHC)

ORCID: 0009-0007-6257-0163

July 2026

Abstract

This paper develops a gravity-independent kinematic framework for reconstructing the local time accumulated by a point embedded in nested celestial rotations and orbits. The central object is a signed, unwrapped angular-progress ledger

$$\boldsymbol{N}_P(t) = (N_1(t), N_2(t), \ldots, N_n(t)),$$

whose entries record spin, orbital, parent-system, and higher phase progression. Angular progress is not identified directly with time and its components are not added as bare scalars. Instead, the position of the point is treated as a geometric embedding

$$\boldsymbol{x}_P = \boldsymbol{x}_P(q^1, \ldots, q^n), \qquad q^a = 2\pi N_a,$$

which induces the point-dependent phase metric

$$\mathsf{G}^{(P)}_{ab} = \frac{\partial \boldsymbol{x}_P}{\partial q^a} \cdot \frac{\partial \boldsymbol{x}_P}{\partial q^b}.$$

The exact flat-spacetime kinematic speed is then

$$v_P^2 = \mathsf{G}^{(P)}_{ab} \dot{q}^a \dot{q}^b,$$

and the corresponding local proper-time factor is

$$\frac{\mathrm{d}\tau_P}{\mathrm{d}t} = \sqrt{1 - \frac{\mathsf{G}^{(P)}_{ab} \dot{q}^a \dot{q}^b}{c^2}}.$$

Thus, the relevant "sum" of angular progressions is a quadratic geometric composition containing both individual contributions and cross terms that encode their

instantaneous relative directions.

For a point on or above a rotating planet whose centre has velocity $\boldsymbol{V}_C$ and whose angular velocity is $\boldsymbol{\Omega}$, the general formula reduces to

$$\boldsymbol{v}_P = \boldsymbol{V}_C + \boldsymbol{\Omega} \times \boldsymbol{r}_P,$$

$$\frac{\mathrm{d}\tau_P}{\mathrm{d}t} = \sqrt{1 - \frac{|\boldsymbol{V}_C + \boldsymbol{\Omega} \times \boldsymbol{r}_P|^2}{c^2}}.$$

This yields an explicit latitude-, altitude-, longitude-, and epoch-dependent kinematic time field without introducing gravitational potential or gravitational redshift. A full rotation average separates the common centre-of-mass motion from the local spin contribution, while the instantaneous expression retains the orbital–spin interference term. Solar–sidereal and synodic relations remain as signed phase-difference identities within the same ledger.

The same kinematic state is assigned the dimensionless chronometric label

$$\mathscr{K}_P \equiv \gamma_P \frac{v_P^2}{c^2} = \gamma_P - \frac{1}{\gamma_P} = \frac{1}{\alpha_P} - \alpha_P, \qquad \gamma_P = (1 - v_P^2/c^2)^{-1/2}.$$

It uniquely reconstructs the clock factor and, for a uniformly rotating ring, equals the magnitude of the one-winding time holonomy divided by the rotation period. The canonical terrestrial equatorial ring gives

$$\mathscr{K}_\oplus \simeq 2.4068743 \times 10^{-12}, \qquad \Theta_\oplus \equiv \mathscr{K}_\oplus T_{\mathrm{sid},\oplus} \simeq 207.386\,\mathrm{ns}.$$

Here $\mathscr{K}_\oplus$ is the dimensionless terrestrial label, whereas $\Theta_\oplus$ is its Earth-cycle translation. For any point or body X, an Earth-language comparison value is defined by

$$\Theta_X^{(\oplus)} \equiv \mathscr{K}_X T_{\mathrm{sid},\oplus} = \frac{\mathscr{K}_X}{\mathscr{K}_\oplus} \Theta_\oplus.$$

This is an explicitly declared calibration map, not a claim that every translated value is the physical holonomy of the compared body.

The construction is extended from local rotating bodies to a declared Galactocentric phase-space atlas. Each object is assigned a frame- and epoch-specific address

$$\mathfrak{A}_X(t_0 \mid \mathcal{F}_\mathrm{G}) = (t_0; \boldsymbol{x}_X, \boldsymbol{v}_X; \boldsymbol{\Sigma}_X, \mathcal{Q}_X),$$

where position and velocity are kept distinct. In cylindrical coordinates the measured speed obeys

$$v_X^2 = \dot{R}_X^2 + R_X^2 \dot{\phi}_X^2 + \dot{z}_X^2 = v_{R,X}^2 + v_{\phi,X}^2 + v_{z,X}^2.$$

The corresponding atlas record stores the address, the dimensionless label, the clock factor, the Earth-language translation, uncertainties, and a quality class distinguishing measured six-dimensional states from position-only or model-completed entries. A position-to-velocity map is allowed only as an explicitly declared conditional closure; position alone is not treated as sufficient to determine speed.

A conditional rod–flywheel closure is also stated. If an independent mechanical model supplies a capacity law for the angular rates, those rates can be inserted into the chronometric equations; the chronometric construction itself does not require gravity and does not claim to derive the rates from geometry alone. The result is a general

operational map from nested angular progression to local *kinematic* proper time at any specified point of a natural rotating body. It is not a replacement for metric-specific gravitational clock calculations when total observed proper time is required.

Keywords: angular progress; nested phase ledger; Galactic phase space; chronometric atlas; kinematic proper time; dimensionless chronometric label; Earth-language translation; Gaia; phase metric; orbital–spin composition.

Epistemic status. The proper-time relation used in this paper is standard special-relativistic kinematics in a specified quasi-inertial reference frame. The proposed contribution is the explicit organisation of nested celestial phase rates into a point-dependent phase metric, a dimensionless local chronometric label, an operational Earth-language comparison map, and a frame-declared Galactic phase-space atlas architecture. No gravitational field, gravitational potential, or gravitational redshift is used in the derivation. Consequently, the framework returns the kinematic component of proper time, not the complete clock rate in a gravitating spacetime. The value 207.386 ns is used only as the terrestrial translation of the canonical dimensionless Earth label; it is not proposed as a universal time quantum or a replacement for the SI second. The optional rod–flywheel and Galactic velocity-field closures are conditional modelling layers and are not presented as established universal laws.

Contents

1 Introduction **6**

2 Scope and Reference Frames **7**
2.1 Chosen reference time 7
2.2 Meaning of “gravity-independent 7

3 The Nested Angular-Progress Ledger **7**
3.1 Immediate-parent chain 7
3.2 Signed recurrence relations 8

4 Geometric Embedding of the Ledger **8**
4.1 Position as a function of angular phases 8
4.2 The phase metric 9

5 Local Kinematic Chronometry **9**
5.1 Angular-progress norm 9
5.2 Weak-speed approximation 10

6 Dimensionless Chronometric Labels and Earth-Language Translation **10**
6.1 The local dimensionless label 10
6.2 Kinematic equivalence classes 11
6.3 Uniform rotating ring and one-winding translation 11
6.4 Canonical Earth calibration 12
6.5 Earth-normalised comparison and translation 12

7 Galactic Phase-Space Addressing and the Chronometric Atlas 13
7.1 Declared Galactocentric frame . . . 13
7.2 Six-dimensional Galactic chronometric address . . . 13
7.3 Why position alone does not determine speed . . . 14
7.4 Nested local refinement inside a Galactic address . . . 14
7.5 Atlas label, clock factor, and Earth-language coordinate . . . 14
7.6 Catalogue quality classes . . . 15
7.7 Observational input and Gaia-compatible ingestion . . . 15
7.8 Uncertainty propagation . . . 16
7.9 Demonstration of address and velocity non-uniqueness . . . 16
7.10 Atlas construction algorithm . . . 17

8 A Planetary Specialisation 17
8.1 Centre motion plus body rotation . . . 17
8.2 Planar circular approximation . . . 18

9 Any Point on or Above a Rotating Planet 18
9.1 Latitude and altitude . . . 18
9.2 Full-rotation average . . . 19
9.3 Difference between two altitudes . . . 19

10 Multiple Nested Motions 20
10.1 General two-phase form . . . 20
10.2 Moon, planet, star, and galactic layers . . . 20
10.3 Non-commuting axes . . . 20

11 Conditional Rod–Flywheel Closure 20
11.1 Why a separate closure is needed . . . 20
11.2 A capacity relation for orbital angular rate . . . 21
11.3 A general spin-transfer law . . . 21

12 Earth as a Numerical Demonstration 22
12.1 Reference values . . . 22
12.2 Orbital centre motion . . . 22
12.3 Sea level and a higher co-rotating point . . . 23
12.4 Latitude dependence . . . 23
12.5 Illustrative rescaled-Earth translation . . . 23

13 Natural Celestial Examples in Ledger Form 24
13.1 Mars . . . 24
13.2 Venus . . . 24
13.3 The Moon . . . 24
13.4 Differentially rotating bodies . . . 24

14 Operational Calculation Protocol 25

15 What Is Established and What Remains Conditional **26**
15.1 Established within the stated kinematic framework 26
15.2 Not established by the chronometric construction 26

16 Discussion **27**

17 Conclusion **28**

A Derivation in Turn-Count Coordinates **29**

B Exact Two-Phase Example **30**

C Relation to a Rotating-Ring Holonomy **30**

D Stable Numerical Formulae **31**

E Reference Python Implementations **31**

1 Introduction

Natural timekeeping is read from recurrent motion. A sidereal day counts the recurrence of a body's orientation relative to distant directions. A solar day counts the recurrence of a star direction at a local meridian. A year counts orbital phase, a synodic period counts the recurrence of a relative alignment, and a spin–orbit resonance expresses a rational relation between two directed phase rates.

These familiar periods are usually kept in separate formulae. The present paper asks a more structural question:

> Given all angular progressions in which a point participates, can its local kinematic time be reconstructed from the geometry of those progressions without introducing gravity?

The answer is affirmative, provided that one distinction is maintained. Angular phase is not itself proper time. Angular progression determines a spatial world line only after the radii, axes, orientations, and nesting geometry have been specified. Proper time is then reconstructed from the speed of that world line.

The central chain is therefore

$$\boxed{\begin{array}{l}\text{nested angular phases and declared phase-space address} \longrightarrow \text{geometric embedding} \\ \quad \longrightarrow \text{point velocity} \longrightarrow \begin{cases}\text{kinematic proper time,} \\ \text{dimensionless label and calibrated atlas coordinate.}\end{cases}\end{array}} \tag{1}$$

The framework has six objectives:

(1) to preserve signed celestial phase bookkeeping in an immediate-parent hierarchy;

(2) to replace a naive scalar sum of angles by an exact geometric composition;

(3) to define a local time-rate field for any point on or above a rotating body;

(4) to keep the determination of angular rates logically separate from the chronometric reconstruction that uses them;

(5) to define a dimensionless chronometric label and a transparent Earth-language calibration for comparing different points and bodies;

(6) to define a frame-, epoch-, and quality-declared six-dimensional Galactic chronometric address;

(7) to provide an executable catalogue protocol that separates measured velocities from conditional position-to-velocity closures.

The final distinction is essential. The present framework can accept measured angular rates, ephemeris-derived rates, or rates produced by a separate mechanical model. It does not require a gravitational potential to convert those rates into kinematic proper time. However, geometry alone does not determine the angular rates unless an additional dynamical closure is supplied.

2 Scope and Reference Frames

2.1 Chosen reference time

Let t denote the coordinate time of a specified quasi-inertial frame over the domain of the calculation. For Solar-System applications this may be an idealised barycentric inertial frame or a sufficiently accurate local inertial approximation. The speed of a point P in that frame is

$$\boldsymbol{v}_P(t) = \frac{\mathrm{d}\boldsymbol{x}_P}{\mathrm{d}t}. \tag{2}$$

In the flat-spacetime kinematic sector,

$$\boxed{\mathrm{d}\tau_P = \mathrm{d}t\sqrt{1 - \frac{v_P^2}{c^2}}.} \tag{3}$$

Equation (3) is the only clock law assumed in the main derivation.

For Galactic applications, the term "Galactocentric frame" does not denote one universally fixed modern standard. The origin, axis orientation, Solar position, Solar velocity, epoch, and transformation parameters must therefore be stated explicitly. The atlas developed below treats this parameter set as part of the address rather than hiding it in software defaults.

2.2 Meaning of "gravity-independent

No term of the form U/c^2, $GM/(rc^2)$, or any metric-specific gravitational potential is introduced. The word "gravity-independent" therefore means that the construction is a self-contained kinematic sector. It does not mean that gravitational effects are physically absent from actual astronomical clocks.

When two real clocks occupy different gravitational potentials, their total observed rate difference generally contains both kinematic and gravitational components. The present paper isolates the former so that the rotational and orbital geometry can be studied without reversing, reinterpreting, or absorbing it into a gravitational term.

3 The Nested Angular-Progress Ledger

3.1 Immediate-parent chain

Let P be a material point, surface marker, atmospheric parcel, or ideal clock associated with a natural celestial body. Introduce an immediate-parent chain

$$P \longrightarrow B_0 \longrightarrow B_1 \longrightarrow \cdots \longrightarrow B_m \longrightarrow \mathcal{F}_{\mathrm{ref}}, \tag{4}$$

where B_0 is the host body and $\mathcal{F}_{\mathrm{ref}}$ is an operational orientation frame, such as the ICRF when appropriate [3, 4].

Each link has an oriented phase q^a and an unwrapped turn count N_a:

$$q^a(t) = q^a(t_0) + 2\pi N_a(t), \qquad N_a(t) \in \mathbb{R}. \tag{5}$$

The sign of $\dot{N}_a$ records prograde or retrograde progression.

Definition 1 (Nested angular-progress ledger)**.** *The ledger of point P is*

$$\boxed{\boldsymbol{N}_P(t) = \Big(N_1(t), N_2(t), \ldots, N_n(t)\Big).} \tag{6}$$

Its rate vector is

$$\boxed{\dot{\boldsymbol{N}}_P(t) = \Big(\nu_1(t), \ldots, \nu_n(t)\Big), \qquad \nu_a = \dot{N}_a = \frac{\omega_a}{2\pi}.} \tag{7}$$

The ledger is an address in nested cycle space. It is not a clock reading and its entries need not be commensurate.



Figure 1: Immediate-parent phase chain. The phase at each level is defined relative to the next parent rather than being referred ambiguously to one common origin.

3.2 Signed recurrence relations

In a planar common-axis approximation, unwrapped angles add and subtract exactly. If a body has signed sidereal spin rate ω_{sid} and orbital phase rate ω_{orb}, the apparent solar rate is

$$\omega_{\text{sol}} = \omega_{\text{sid}} - \omega_{\text{orb}}. \tag{8}$$

Using signed periods P^*,

$$\boxed{\frac{1}{P^*_{\text{sol}}} = \frac{1}{P^*_{\text{sid}}} - \frac{1}{P_{\text{orb}}}.} \tag{9}$$

Synodic recurrence and spin–orbit locking are analogous differences between selected ledger components. In three dimensions, however, finite rotations must be composed as ordered products in SO(3) rather than as scalar angle sums.

4 Geometric Embedding of the Ledger

4.1 Position as a function of angular phases

Let the spatial position of point P be a differentiable function of the angular coordinates:

$$\boxed{\boldsymbol{x}_P = \boldsymbol{x}_P(q^1, q^2, \ldots, q^n; t).} \tag{10}$$

An explicit t dependence is allowed for translations, slowly changing axes, deformation, precession parameters, or other prescribed non-angular inputs.

Differentiating gives

$$\boldsymbol{v}_P = \frac{\partial \boldsymbol{x}_P}{\partial t} + \sum_{a=1}^{n} \frac{\partial \boldsymbol{x}_P}{\partial q^a} \dot{q}^a. \tag{11}$$

Define

$$\boldsymbol{v}_0^{(P)} \equiv \frac{\partial \boldsymbol{x}_P}{\partial t}, \qquad \boldsymbol{e}_a^{(P)} \equiv \frac{\partial \boldsymbol{x}_P}{\partial q^a}. \tag{12}$$

Then

$$\boldsymbol{v}_P = \boldsymbol{v}_0^{(P)} + \boldsymbol{e}_a^{(P)} \dot{q}^a, \tag{13}$$

where repeated phase indices are summed.

4.2 The phase metric

Definition 2 (Point-dependent phase metric)**.** *The Gram matrix of angular displacement vectors is*

$$\boxed{\mathsf{G}_{ab}^{(P)} \equiv \boldsymbol{e}_a^{(P)} \cdot \boldsymbol{e}_b^{(P)} = \frac{\partial \boldsymbol{x}_P}{\partial q^a} \cdot \frac{\partial \boldsymbol{x}_P}{\partial q^b}.} \tag{14}$$

The diagonal entries encode the squared effective radii associated with each phase. The off-diagonal entries encode the instantaneous geometric alignment between different phase-generated velocities.

Proposition 1 (Angular composition of speed)**.** *If $\partial \boldsymbol{x}_P/\partial t = 0$ in the chosen embedding, then*

$$\boxed{v_P^2 = \mathsf{G}_{ab}^{(P)} \dot{q}^a \dot{q}^b.} \tag{15}$$

For a general embedding,

$$\boxed{v_P^2 = |\boldsymbol{v}_0^{(P)}|^2 + 2\boldsymbol{v}_0^{(P)} \cdot \boldsymbol{e}_a^{(P)} \dot{q}^a + \mathsf{G}_{ab}^{(P)} \dot{q}^a \dot{q}^b.} \tag{16}$$

Proof. Take the Euclidean inner product of equation (13) with itself and use the definition (14). □

This is the precise meaning of "adding angular progressions." The correct composition is not

$$\omega_1 + \omega_2 + \cdots, \tag{17}$$

but a point-dependent quadratic form whose cross terms retain directional information.

5 Local Kinematic Chronometry

5.1 Angular-progress norm

Define the dimensionless angular-progress norm

$$\boxed{\mathcal{A}_P(t) \equiv \frac{v_P^2(t)}{c^2}.} \tag{18}$$

When the position is generated entirely by the phase coordinates,

$$\mathcal{A}_P = \frac{1}{c^2} \mathsf{G}_{ab}^{(P)} \dot{q}^a \dot{q}^b = \frac{4\pi^2}{c^2} \mathsf{G}_{ab}^{(P)} \nu_a \nu_b. \tag{19}$$

Definition 3 (Local kinematic clock factor)**.**

$$\boxed{\alpha_P(t) \equiv \frac{\mathrm{d}\tau_P}{\mathrm{d}t} = \sqrt{1 - \mathcal{A}_P(t)}.} \tag{20}$$

The accumulated kinematic proper time between t_0 and t_1 is

$$\boxed{\tau_P(t_1) - \tau_P(t_0) = \int_{t_0}^{t_1} \sqrt{1 - \mathcal{A}_P(t)}\, \mathrm{d}t.} \tag{21}$$

Master relation. Given the nested angular phases, their rates, and the geometric embedding of a point, the gravity-independent kinematic time of that point is

$$\boxed{\mathrm{d}\tau_P = \mathrm{d}t\sqrt{1 - \frac{1}{c^2}\mathsf{G}^{(P)}_{ab}\dot{q}^a\dot{q}^b}} \tag{22}$$

when the embedding has no additional explicit translation, and is obtained from equations (3) and (16) in the general case.

5.2 Weak-speed approximation

For $v_P \ll c$,

$$\alpha_P = 1 - \frac{\mathcal{A}_P}{2} + \mathcal{O}(\mathcal{A}_P^2), \tag{23}$$

so that

$$\Delta\tau_P \simeq \Delta t - \frac{1}{2c^2}\int v_P^2(t)\, \mathrm{d}t. \tag{24}$$

The clock deficit is therefore governed by the time integral of the squared resultant velocity, not by the arithmetic total of traversed angle.

6 Dimensionless Chronometric Labels and Earth-Language Translation

6.1 The local dimensionless label

The angular-progress norm $\mathcal{A}_P = v_P^2/c^2$ already supplies a dimensionless measure of motion. For translation and comparison it is useful to introduce a second, exactly invertible label that is also the period-normalised rotating-ring holonomy.

Definition 4 (Dimensionless chronometric label)**.** *For a point P with $0 \le \mathcal{A}_P < 1$, define*

$$\boxed{\mathscr{K}_P \equiv \gamma_P\mathcal{A}_P = \gamma_P\frac{v_P^2}{c^2}, \qquad \gamma_P = (1 - \mathcal{A}_P)^{-1/2}.} \tag{25}$$

Using $\alpha_P = 1/\gamma_P$, the same label has the exact identities

$$\boxed{\mathscr{K}_P = \gamma_P - \frac{1}{\gamma_P} = \frac{1}{\alpha_P} - \alpha_P.} \tag{26}$$

Proposition 2 (Exact label–rate correspondence)**.** *For $\mathscr{K}_P \geq 0$, the local kinematic clock factor is uniquely recovered by*

$$\boxed{\alpha_P = \frac{\sqrt{\mathscr{K}_P^2 + 4} - \mathscr{K}_P}{2} = \frac{2}{\sqrt{\mathscr{K}_P^2 + 4} + \mathscr{K}_P}.} \tag{27}$$

Proof. Equation (26) gives $\alpha_P^2 + \mathscr{K}_P \alpha_P - 1 = 0$. Only the positive root can represent a future-directed clock-rate factor. The second expression in equation (27) is algebraically equivalent and is numerically stable for small labels. □

In the weak-speed regime,

$$\mathscr{K}_P = \mathcal{A}_P + \mathcal{O}(\mathcal{A}_P^2), \qquad \alpha_P = 1 - \frac{\mathscr{K}_P}{2} + \mathcal{O}(\mathscr{K}_P^2). \tag{28}$$

6.2 Kinematic equivalence classes

Two specified points A and B, evaluated in the same reference frame and under the same convention, are *instantaneously kinematically chronometric-equivalent* when

$$\boxed{\mathscr{K}_A(t) = \mathscr{K}_B(t).} \tag{29}$$

Then $\alpha_A(t) = \alpha_B(t)$ follows exactly from equation (27). Equality of one averaged label is not, by itself, sufficient to guarantee equality of accumulated proper time when the labels vary, because the inverse map is nonlinear. Equality of the complete functions $\mathscr{K}_A(t)$ and $\mathscr{K}_B(t)$ over a common interval does guarantee equal accumulated kinematic proper time over that interval.

The label is not an intrinsic name detached from measurement convention. A reproducible label must be accompanied by the selected reference frame, point or canonical ring, latitude, altitude, epoch, and any centre-motion contribution. These metadata define what is being compared.

6.3 Uniform rotating ring and one-winding translation

For a uniformly rotating ring,

$$\beta = \frac{|\Omega| R}{c}, \qquad \mathscr{K}_{\mathrm{ring}} = \gamma \beta^2. \tag{30}$$

The one-direction slow-transport holonomy satisfies

$$\boxed{|\mathcal{H}^*| = \mathscr{K}_{\mathrm{ring}} T_{\mathrm{rot}}, \qquad T_{\mathrm{rot}} = \frac{2\pi}{|\Omega|}.} \tag{31}$$

Thus $\mathscr{K}$ is the unit-independent dimensionless content of this particular ring observable, while multiplication by a selected period translates it into a time interval.

6.4 Canonical Earth calibration

Adopt the WGS 84 equatorial radius and the terrestrial sidereal rotation period as the canonical Earth ring. Its dimensionless label is

$$\boxed{\mathscr{K}_\oplus \equiv \gamma_\oplus \frac{\Omega_\oplus^2 R_\oplus^2}{c^2} \simeq 2.4068743 \times 10^{-12}.} \tag{32}$$

Define the Earth-cycle translation constant

$$\boxed{\Theta_\oplus \equiv \mathscr{K}_\oplus T_{\mathrm{sid},\oplus} \simeq 207.386\,\mathrm{ns}.} \tag{33}$$

The number $\Theta_\oplus$ is therefore the terrestrial time-language rendering of $\mathscr{K}_\oplus$ under the canonical equatorial sidereal-cycle convention. It is not the fundamental label itself.

6.5 Earth-normalised comparison and translation

For any point or canonical body state X, define the Earth-normalised ratio

$$\boxed{\mathscr{R}_{X/\oplus} \equiv \frac{\mathscr{K}_X}{\mathscr{K}_\oplus}.} \tag{34}$$

Its Earth-language translated interval is

$$\boxed{\Theta_X^{(\oplus)} \equiv \mathscr{K}_X T_{\mathrm{sid},\oplus} = \mathscr{R}_{X/\oplus}\Theta_\oplus.} \tag{35}$$

When expressed in terrestrial nanoseconds,

$$\boxed{\Theta_X^{(\oplus)}[\mathrm{ns}_\oplus] = \frac{\mathscr{K}_X}{\mathscr{K}_\oplus} \times 207.386\ldots} \tag{36}$$

This is a comparison coordinate in the Earth calibration language. It must not be silently identified with the physical one-winding holonomy of X. That identification is valid only when X is itself represented by a uniform ring and its own native period is used.

For a uniform ring X, define the native-cycle translation

$$\boxed{\Theta_X^{(X)} \equiv \mathscr{K}_X T_X = |\mathcal{H}_X^*|.} \tag{37}$$

The coefficient that converts the native-cycle representation to the Earth-cycle representation is

$$\boxed{C_{X\to\oplus} \equiv \frac{T_{\mathrm{sid},\oplus}}{T_X}, \qquad \Theta_X^{(\oplus)} = C_{X\to\oplus}\Theta_X^{(X)}.} \tag{38}$$

Remark 1. *The translation map changes the numerical language, not the underlying kinematic label. Equal Earth-language translations imply equal labels because the same positive calibration period is used. Equal native-cycle intervals do not necessarily imply equal labels, since the native periods may differ.*

7 Galactic Phase-Space Addressing and the Chronometric Atlas

7.1 Declared Galactocentric frame

A Galactic atlas requires a declared reference construction. Let

$$\boxed{\mathcal{F}_{\mathrm{G}} = \left(O_{\mathrm{G}}, \hat{\boldsymbol{e}}_x, \hat{\boldsymbol{e}}_y, \hat{\boldsymbol{e}}_z, t_{\mathrm{G}}; \boldsymbol{\Pi}_{\mathrm{G}}\right)} \tag{39}$$

contain the adopted Galactic centre, oriented axes, coordinate time, and the parameter set $\boldsymbol{\Pi}_{\mathrm{G}}$ used to transform observations into that frame. The latter may include the Solar distance from the Galactic centre, the Solar height above the adopted midplane, and the Solar velocity vector. There is no single modern standard Galactocentric frame with universally fixed numerical parameters; consequently, reproducible science code must record them explicitly rather than relying on a changing default [15].

This Galactocentric frame is distinct from the conventional *Galactic* sky-coordinate system. The latter has its three-dimensional origin at the Solar-System barycentre, whereas the former is constructed with the Galactic centre as origin. Confusing these origins changes both the position and the velocity attached to an object.

7.2 Six-dimensional Galactic chronometric address

Definition 5 (Galactic phase-space address)**.** *For an object X at epoch t_0, define*

$$\boxed{\mathfrak{A}_X(t_0 \mid \mathcal{F}_{\mathrm{G}}) = (t_0; \boldsymbol{x}_X, \boldsymbol{v}_X; \boldsymbol{\Sigma}_X, \mathcal{Q}_X)\,,} \tag{40}$$

where $\boldsymbol{x}_X$ and $\boldsymbol{v}_X$ are its Galactocentric position and velocity, $\boldsymbol{\Sigma}_X$ is the adopted covariance or uncertainty description, and $\mathcal{Q}_X$ records whether the state is measured, transformed, incomplete, or model-completed.

In Cartesian coordinates,

$$\boldsymbol{x}_X = (x_X, y_X, z_X), \qquad \boldsymbol{v}_X = (v_{x,X}, v_{y,X}, v_{z,X}). \tag{41}$$

The corresponding cylindrical quantities are

$$R_X = \sqrt{x_X^2 + y_X^2}, \qquad \phi_X = \operatorname{atan2}(y_X, x_X), \tag{42}$$

$$v_{R,X} = \frac{x_X v_{x,X} + y_X v_{y,X}}{R_X}, \qquad v_{\phi,X} = \frac{x_X v_{y,X} - y_X v_{x,X}}{R_X}, \tag{43}$$

for $R_X > 0$. Therefore

$$\boxed{v_X^2 = v_{x,X}^2 + v_{y,X}^2 + v_{z,X}^2 = v_{R,X}^2 + v_{\phi,X}^2 + v_{z,X}^2.} \tag{44}$$

Equivalently, when the path is differentiable,

$$\boxed{v_X^2 = \dot{R}_X^2 + R_X^2 \dot{\phi}_X^2 + \dot{z}_X^2.} \tag{45}$$

The Galactic azimuthal phase ϕ_X is thus one ledger component, with rate $\dot{\phi}_X = v_{\phi,X}/R_X$, but the radial and vertical components must also be retained.

7.3 Why position alone does not determine speed

A single position $\boldsymbol{x}_X(t_0)$ does not determine $\boldsymbol{v}_X(t_0)$. Distinct world lines can intersect the same event in configuration space with different tangent vectors. Hence no unique inference

$$\boldsymbol{x}_X \longmapsto \boldsymbol{v}_X \tag{46}$$

exists without additional information.

There are only two legitimate routes from position to speed in this framework:

(1) observe the position at multiple epochs or provide the full measured phase-space state;

(2) declare a conditional velocity-field closure

$$\boxed{\boldsymbol{v}_X = \boldsymbol{\mathcal{V}}_{\mathrm{MW}}(\boldsymbol{x}_X, t; \boldsymbol{\eta}),} \tag{47}$$

where $\boldsymbol{\eta}$ records all model parameters.

Examples of conditional closures include a constant tangential-speed field, a solid-rotation field, a tabulated rotation curve, or a fitted non-axisymmetric velocity map. A velocity supplied this way is not labelled "measured" in the atlas.

7.4 Nested local refinement inside a Galactic address

Let X denote the centre of a star, planet, moon, cluster, or other host object, and let P be a point attached to or moving within it. Write

$$\boldsymbol{x}_P^{\mathrm{G}} = \boldsymbol{x}_X^{\mathrm{G}} + \mathsf{R}_X(t)\boldsymbol{r}_P, \tag{48}$$

where $\mathsf{R}_X(t) \in \mathrm{SO}(3)$ embeds the local coordinates into $\mathcal{F}_{\mathrm{G}}$. Differentiation gives

$$\boldsymbol{v}_P^{\mathrm{G}} = \boldsymbol{v}_X^{\mathrm{G}} + \dot{\mathsf{R}}_X\boldsymbol{r}_P + \mathsf{R}_X\dot{\boldsymbol{r}}_P. \tag{49}$$

For a fixed point co-rotating rigidly with the host,

$$\boxed{\boldsymbol{v}_P^{\mathrm{G}} = \boldsymbol{v}_X^{\mathrm{G}} + \boldsymbol{\Omega}_X^{\mathrm{G}} \times \boldsymbol{r}_P^{\mathrm{G}}.} \tag{50}$$

Additional moon–planet, planet–star, binary, cluster, or stream layers are inserted through the same ordered phase embedding used in the earlier sections. The centre address is therefore not a replacement for the nested ledger; it is its outer Galactic layer.

7.5 Atlas label, clock factor, and Earth-language coordinate

For every complete phase-space address, define

$$\beta_X^{\mathrm{G}} = \frac{|\boldsymbol{v}_X^{\mathrm{G}}|}{c}, \qquad \gamma_X^{\mathrm{G}} = \left(1 - (\beta_X^{\mathrm{G}})^2\right)^{-1/2}, \tag{51}$$

$$\boxed{\mathscr{K}_X^{\mathrm{G}} = \gamma_X^{\mathrm{G}}(\beta_X^{\mathrm{G}})^2, \qquad \alpha_X^{\mathrm{G}} = \frac{1}{\gamma_X^{\mathrm{G}}}.} \tag{52}$$

The Earth-language atlas coordinate is

$$\boxed{\Theta_{X,\mathrm{G}}^{(\oplus)} = \mathscr{K}_X^{\mathrm{G}} T_{\mathrm{sid},\oplus} = \frac{\mathscr{K}_X^{\mathrm{G}}}{\mathscr{K}_\oplus}\Theta_\oplus.} \tag{53}$$

The full catalogue record is

$$\boxed{\mathfrak{C}_X = \left(\mathrm{ID}_X, t_0, \mathcal{F}_{\mathrm{G}}, \boldsymbol{x}_X, \boldsymbol{v}_X, \mathscr{K}_X, \alpha_X, \Theta_X^{(\oplus)}, \boldsymbol{\Sigma}_X, \mathcal{Q}_X\right).} \tag{54}$$

The translated interval is a coordinate in the declared terrestrial calibration language. It is not a claim that the object physically performs an Earth-length cycle or possesses a closed-loop holonomy equal to that interval.

7.6 Catalogue quality classes

Table 1: Minimum provenance classes for a Galactic chronometric atlas.

Code	**State**	**Interpretation**
M6	Measured six-dimensional	Position and all three velocity components are observational inputs in a declared frame.
T6	Transformed six-dimensional	A complete observed sky state has been transformed into the declared Galactocentric frame.
P5	Position plus transverse motion	Radial velocity is absent; no unique total-speed label is issued.
P3	Position only	The atlas stores the address but leaves the chronometric label undefined.
C6	Conditional six-dimensional	Missing velocity components are supplied by a named model closure and are never presented as measurements.

This classification prevents a position-only map from being silently promoted into a measured time map. The catalogue may contain all classes, but only complete M6, T6, or explicitly conditional C6 rows receive $\mathscr{K}$.

7.7 Observational input and Gaia-compatible ingestion

A complete stellar input can be formed from sky position, distance, two proper-motion components, radial velocity, epoch, and their covariance. Gaia DR3 provides astrometry for a very large source set and combined radial velocities for more than 3.3×10^7 stars, making a large six-dimensional implementation feasible for the measured subset [12, 13]. The Gaia archive documents the fields `ra`, `dec`, `parallax`, `pmra`, `pmdec`, and `radial_velocity`, including their reference-frame and uncertainty conventions [14].

For a source with distance d in kiloparsecs and total proper motion μ in milliarcseconds

per year, the transverse-speed magnitude is

$$\boxed{v_\perp[\mathrm{km\,s^{-1}}] = 4.74047\,\mu[\mathrm{mas\,yr^{-1}}]\,d[\mathrm{kpc}].} \tag{55}$$

The complete vector transformation must nevertheless preserve the directions of both proper-motion components and add the line-of-sight velocity before transforming to $\mathcal{F}_{\mathrm{G}}$. The accompanying code offers an optional Astropy path for this transformation, with the Galactocentric parameters supplied explicitly rather than inherited silently from software defaults [15].

7.8 Uncertainty propagation

Let $\boldsymbol{y}$ denote the measured astrometric and velocity variables with covariance $\boldsymbol{\Sigma}_y$. To first order,

$$\boxed{\sigma^2_{\mathscr{K}} \simeq \nabla_{\boldsymbol{y}}\mathscr{K}^{\mathsf{T}}\boldsymbol{\Sigma}_y\nabla_{\boldsymbol{y}}\mathscr{K}\,.} \tag{56}$$

For a directly measured velocity vector, with $v = |\boldsymbol{v}|$,

$$\frac{\mathrm{d}\mathscr{K}}{\mathrm{d}v} = \frac{1}{c}\left(2\beta\gamma + \beta^3\gamma^3\right), \qquad \nabla_{\boldsymbol{v}}\mathscr{K} = \frac{\mathrm{d}\mathscr{K}}{\mathrm{d}v}\frac{\boldsymbol{v}}{v}. \tag{57}$$

The Earth-language uncertainty is simply

$$\boxed{\sigma_{\Theta^{(\oplus)}} = T_{\mathrm{sid},\oplus}\sigma_{\mathscr{K}}.} \tag{58}$$

For nonlinear distance transformations, asymmetric distance posteriors, or correlated astrometric errors, Monte Carlo propagation from the full input covariance is preferable to a diagonal first-order approximation.

7.9 Demonstration of address and velocity non-uniqueness

Table 2 uses an explicit illustrative Galactocentric convention with a Solar-frame velocity vector $(12.9, 245.6, 7.78)$ km/s. These values demonstrate the pipeline; they do not define a unique Galactic standard. The second row occupies the same position as the first but carries an additional velocity component, whereas the third is placed elsewhere with the same velocity magnitude as the first.

Table 2: Illustrative Galactic atlas records. $\Theta^{(\oplus)}$ is the Earth-language comparison coordinate.

Record	v (km/s)	$\mathscr{K}$	$\mathscr{K}/\mathscr{K}_\oplus$	$\Theta^{(\oplus)}$ (ms)
Illustrative Solar-frame state	246.062	6.73669×10^{-7}	2.79894×10^5	58.046
Same position, extra component	270.419	8.13640×10^{-7}	3.38049×10^5	70.107
Different position, equal speed	246.062	6.73669×10^{-7}	2.79894×10^5	58.046
Position only	–	–	–	–

The first and third rows belong to the same instantaneous kinematic chronometric equivalence class despite different positions. The first and second do not, despite identical

positions. This is the operational reason that the atlas is fundamentally a phase-space atlas rather than a position-only map.

7.10 Atlas construction algorithm

A catalogue implementation follows this sequence:

A1: Freeze the Galactocentric frame parameters, axis convention, and epoch.

A2: Ingest either Cartesian phase-space rows or complete sky-coordinate rows.

A3: Transform every complete state to the same $\mathcal{F}_{\mathrm{G}}$; retain the full covariance when available.

A4: Assign the provenance class M6, T6, P5, P3, or C6.

A5: Compute cylindrical coordinates and verify $v^2 = v_R^2 + v_\phi^2 + v_z^2$.

A6: Calculate $\mathscr{K}$, α, $\mathscr{K}/\mathscr{K}_\oplus$, and $\Theta^{(\oplus)}$ only for complete states.

A7: Propagate uncertainties and attach quality flags rather than dropping them from the map.

A8: Store the result as a versioned table and visualise position separately from colour- or contour-coded chronometric labels.

A9: Never mix labels calculated in different frames, epochs, or Solar-parameter conventions without an explicit transformation.

8 A Planetary Specialisation

8.1 Centre motion plus body rotation

Let C be the centre of a rotating planet. Let $\boldsymbol{V}_C(t)$ be its velocity in the chosen quasi-inertial frame, and let $\boldsymbol{r}_P(t)$ be the position of point P relative to the centre. For rigid rotation with angular velocity $\boldsymbol{\Omega}(t)$,

$$\frac{\mathrm{d}\boldsymbol{r}_P}{\mathrm{d}t} = \boldsymbol{\Omega} \times \boldsymbol{r}_P. \tag{59}$$

Therefore

$$\boxed{\boldsymbol{v}_P = \boldsymbol{V}_C + \boldsymbol{\Omega} \times \boldsymbol{r}_P.} \tag{60}$$

The exact local kinematic clock factor is

$$\boxed{\alpha_P = \sqrt{1 - \frac{|\boldsymbol{V}_C + \boldsymbol{\Omega} \times \boldsymbol{r}_P|^2}{c^2}}.} \tag{61}$$

Expanding,

$$\boxed{|\boldsymbol{v}_P|^2 = V_C^2 + |\boldsymbol{\Omega} \times \boldsymbol{r}_P|^2 + 2\boldsymbol{V}_C \cdot (\boldsymbol{\Omega} \times \boldsymbol{r}_P).} \tag{62}$$

The three terms have distinct meanings:

(1) motion of the planetary centre through the selected frame;

(2) local spin motion around the planetary axis;

(3) instantaneous orbital–spin alignment or opposition.

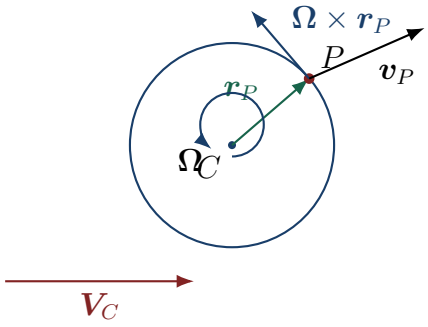

Figure 2: Velocity composition for a point on a rotating planet. The total velocity is the vector sum of centre motion and local spin motion.

8.2 Planar circular approximation

For a circular planetary orbit of radius a, orbital angular rate ω_{orb}, and local cylindrical spin radius ρ, let

$$V_{\mathrm{orb}} = a|\omega_{\mathrm{orb}}|, \qquad v_{\mathrm{spin}} = \rho|\Omega_{\mathrm{spin}}|. \tag{63}$$

If $\psi(t)$ is the instantaneous angle between the orbital and spin velocity vectors,

$$\boxed{v_P^2 = V_{\mathrm{orb}}^2 + v_{\mathrm{spin}}^2 + 2V_{\mathrm{orb}}v_{\mathrm{spin}}\cos\psi(t).} \tag{64}$$

Thus

$$\boxed{\alpha_P(t) = \sqrt{1 - \frac{V_{\mathrm{orb}}^2 + v_{\mathrm{spin}}^2 + 2V_{\mathrm{orb}}v_{\mathrm{spin}}\cos\psi(t)}{c^2}}.} \tag{65}$$

The cross term is required for instantaneous chronometry. It cannot be recovered from the magnitudes of the two angular rates alone; their geometric orientation must also be known.

9 Any Point on or Above a Rotating Planet

9.1 Latitude and altitude

For a spherical reference body of radius R, a point at altitude h and latitude λ has cylindrical distance from the spin axis

$$\boxed{\rho(h,\lambda) = (R+h)\cos\lambda.} \tag{66}$$

The local spin speed is

$$\boxed{v_{\mathrm{spin}}(h,\lambda) = |\Omega_{\mathrm{spin}}|(R+h)\cos\lambda.} \tag{67}$$

Substitution into equation (65) gives a direct kinematic time field over the body and its co-rotating atmosphere:

$$\alpha(h,\lambda,t) = \left[1 - \frac{1}{c^2}\Big(V_{\rm orb}^2 + \Omega_{\rm spin}^2 (R+h)^2\cos^2\lambda + 2V_{\rm orb}|\Omega_{\rm spin}|(R+h)\cos\lambda\cos\psi(t)\Big)\right]^{1/2}. \tag{68}$$

This expression contains no gravitational variable. Its inputs are the selected reference-frame motion of the centre, the spin rate, the point's position relative to the axis, and the instantaneous orientation between the nested motions.

9.2 Full-rotation average

If $\boldsymbol{V}_C$ changes negligibly over one body rotation and the spin velocity explores a complete symmetric cycle, then

$$\langle \boldsymbol{V}_C \cdot (\boldsymbol{\Omega} \times \boldsymbol{r}_P)\rangle_{\rm spin} = 0. \tag{69}$$

Hence

$$\boxed{\left\langle v_P^2 \right\rangle_{\rm spin} = V_C^2 + \Omega_{\rm spin}^2 \rho^2.} \tag{70}$$

In the weak-speed limit,

$$\boxed{\langle \alpha_P \rangle_{\rm spin} \simeq 1 - \frac{V_C^2}{2c^2} - \frac{\Omega_{\rm spin}^2 \rho^2}{2c^2}.} \tag{71}$$

The first deficit is common to points sharing the same centre motion. The second distinguishes latitude and altitude.

9.3 Difference between two altitudes

At the same latitude, compare h_1 and h_2. The spin-only difference in squared speed is

$$\Delta v_{\rm spin}^2 = \Omega_{\rm spin}^2 \left[(R+h_2)^2 - (R+h_1)^2\right]\cos^2\lambda. \tag{72}$$

For $|h_2 - h_1| \ll R$,

$$\Delta v_{\rm spin}^2 \simeq 2\Omega_{\rm spin}^2 R \Delta h \cos^2\lambda. \tag{73}$$

The full-spin-cycle averaged kinematic clock-rate difference is therefore

$$\boxed{\Delta\left\langle \frac{{\rm d}\tau}{{\rm d}t}\right\rangle_{\rm kin} \simeq -\frac{\Omega_{\rm spin}^2 R \Delta h \cos^2\lambda}{c^2}.} \tag{74}$$

Within this gravity-independent sector, the higher co-rotating point is farther from the axis, has greater spin speed, and accumulates slightly less proper time over the same coordinate interval.

10 Multiple Nested Motions

10.1 General two-phase form

For two angular phases q^1 and q^2,

$$\boxed{v_P^2 = \mathsf{G}_{11}(\dot{q}^1)^2 + \mathsf{G}_{22}(\dot{q}^2)^2 + 2\mathsf{G}_{12}\dot{q}^1\dot{q}^2.} \tag{75}$$

Writing

$$\mathsf{G}_{11} = r_1^2, \qquad \mathsf{G}_{22} = r_2^2, \qquad \mathsf{G}_{12} = r_1 r_2 \cos\psi, \tag{76}$$

recovers

$$\boxed{v_P^2 = r_1^2\omega_1^2 + r_2^2\omega_2^2 + 2r_1 r_2 \omega_1\omega_2 \cos\psi.} \tag{77}$$

The sign of each ω_a and the geometry contained in $\cos\psi$ jointly determine whether two phase-generated velocities reinforce or oppose each other.

10.2 Moon, planet, star, and galactic layers

For a lunar surface point, the position may contain at least the following motions:

(1) lunar spin about the lunar axis;

(2) lunar orbital motion about the Earth–Moon barycentre;

(3) barycentric motion of the Earth–Moon system about the Solar-System barycentre;

(4) optional higher motion relative to a Galactic or extragalactic reference construction.

The ledger stores these phase rates separately. The embedding $\boldsymbol{x}_P(\boldsymbol{q})$ determines their exact vector composition. Adding more layers changes the dimension of $\mathsf{G}_{ab}^{(P)}$ but not the chronometric law (22).

10.3 Non-commuting axes

Finite orientations in three dimensions satisfy

$$R_{P/\mathrm{ref}}(t) = R_n(q^n)R_{n-1}(q^{n-1})\cdots R_1(q^1), \qquad R_a \in \mathrm{SO}(3). \tag{78}$$

The order is physical. The position embedding is built from the ordered product, after which the phase metric is computed by differentiation. The quadratic chronometric relation remains valid even though the finite angles themselves do not commute.

11 Conditional Rod–Flywheel Closure

11.1 Why a separate closure is needed

The chronometric equations answer the following question:

> If the angular rates and the geometry are known, how much kinematic proper time does the point accumulate?

They do not, by themselves, answer:

Why does a particular planet possess those angular rates?

That second question requires an independent mechanical or dynamical law. A rod–flywheel model can provide one conditional closure without altering the chronometric construction.

11.2 A capacity relation for orbital angular rate

Suppose a model assigns a fixed radial transport capacity $F_\star$ to a flywheel–rod channel carrying a body of inertial mass m at radius a. Imposing a uniform angular rate ω_{orb} gives the capacity equation

$$\boxed{F_\star = m\omega_{\mathrm{orb}}^2 a.} \tag{79}$$

Then

$$\boxed{\omega_{\mathrm{orb}} = \sqrt{\frac{F_\star}{ma}}.} \tag{80}$$

If the body is approximated by uniform density ϱ and radius R,

$$m = \frac{4\pi}{3}\varrho R^3, \tag{81}$$

so

$$\boxed{\omega_{\mathrm{orb}} = \sqrt{\frac{3F_\star}{4\pi\varrho R^3 a}}.} \tag{82}$$

At fixed $F_\star$, ϱ, and ω_{orb},

$$R \propto a^{-1/3}. \tag{83}$$

This is the scaling used in the illustrative Earth–Neptune rod experiment: increasing the rod length by a factor of approximately 30 requires a same-density terminal body's radius to fall by approximately $30^{-1/3} \simeq 0.322$ if the angular rate and capacity are retained.

11.3 A general spin-transfer law

A separate coupling is needed to relate body spin to orbital transport. The most general dimensionally consistent form based on a dimensionless geometry ratio is

$$\boxed{\Omega_{\mathrm{spin}} = \omega_{\mathrm{orb}}\,\Phi\left(\frac{a}{R};\boldsymbol{\eta}\right),} \tag{84}$$

where $\boldsymbol{\eta}$ denotes any additional dimensionless structural parameters, such as density ratios, moment-of-inertia coefficients, coupling fractions, or internal phase-lock indices.

The function Φ must be derived from a mechanical model or inferred and tested against data. Assuming $\Phi(x) = x$ without such a derivation would be an ansatz, not a consequence of the phase ledger.

Once equations (80) and (84) supply ω_{orb} and Ω_{spin}, they can be inserted directly into equations (22) and (68). Thus the dynamical closure and the chronometric map remain modular.

12 Earth as a Numerical Demonstration

12.1 Reference values

For illustration, adopt the WGS 84 equatorial radius

$$R_\oplus = 6\,378\,137\,\mathrm{m}, \tag{85}$$

and the mean sidereal day

$$T_{\mathrm{sid}} = 86\,164.090\,54\,\mathrm{s}, \tag{86}$$

so that [9, 6]

$$\Omega_\oplus = \frac{2\pi}{T_{\mathrm{sid}}} \simeq 7.292\,115 \times 10^{-5}\,\mathrm{rad/s}. \tag{87}$$

The equatorial spin speed is

$$\boxed{v_{\mathrm{eq}} = \Omega_\oplus R_\oplus \simeq 465.101\,\mathrm{m/s}.} \tag{88}$$

The spin-only dimensionless progress norm at the equator is

$$\mathcal{A}_{\mathrm{spin,eq}} = \frac{v_{\mathrm{eq}}^2}{c^2} \simeq 2.4068743 \times 10^{-12}. \tag{89}$$

The exact canonical Earth label is

$$\boxed{\mathscr{K}_\oplus = \gamma_\oplus \mathcal{A}_{\mathrm{spin,eq}} \simeq 2.4068743 \times 10^{-12}.} \tag{90}$$

The difference between $\mathscr{K}_\oplus$ and $\mathcal{A}_{\mathrm{spin,eq}}$ begins only beyond the displayed precision. Multiplication by the sidereal period gives the Earth-language calibration

$$\boxed{\Theta_\oplus = \mathscr{K}_\oplus T_{\mathrm{sid}} \simeq 207.386\,\mathrm{ns}.} \tag{91}$$

The corresponding proper-time deficit of the co-rotating equatorial clock over one sidereal rotation is a different quantity,

$$\delta\tau_{\mathrm{spin,eq}} = T_{\mathrm{sid}} \left(1 - \frac{1}{\gamma_\oplus}\right) \simeq 103.7\,\mathrm{ns}. \tag{92}$$

At weak speed, this deficit is approximately one half of the one-direction holonomy translation $\Theta_\oplus$.

12.2 Orbital centre motion

Using the exact astronomical unit and the sidereal year listed by JPL [6], a circular mean-speed estimate is

$$V_{\mathrm{orb},\oplus} \simeq \frac{2\pi\,\mathrm{au}}{T_{\mathrm{year}}} \simeq 29.785\,\mathrm{km/s}. \tag{93}$$

This common centre-motion contribution produces a weak-speed kinematic deficit of approximately

$$\frac{V_{\mathrm{orb},\oplus}^2}{2c^2} \simeq 4.94 \times 10^{-9}, \tag{94}$$

or approximately 425 µs over one sidereal day in the selected barycentric flat-spacetime approximation. It is common to nearby terrestrial points and therefore largely cancels in comparisons between them.

The instantaneous cross term in equation (62) can be much larger than the spin-squared term, reaching an equatorial magnitude of order

$$\frac{V_{\mathrm{orb},\oplus} v_{\mathrm{eq}}}{c^2} \simeq 1.54 \times 10^{-10}, \tag{95}$$

but it changes sign through the daily cycle and averages to zero under the symmetric approximation (69).

12.3 Sea level and a higher co-rotating point

At the equator, compare sea level with a point 1000 m higher. Equation (74) gives

$$\Delta \left\langle \frac{\mathrm{d}\tau}{\mathrm{d}t} \right\rangle_{\mathrm{kin}} \simeq -3.77 \times 10^{-16}. \tag{96}$$

Over one sidereal day,

$$\boxed{\Delta\tau_{\mathrm{kin}} \simeq -32.5\,\mathrm{ps}.} \tag{97}$$

The sign is negative because the higher co-rotating point has the greater tangential spin speed. This number is intentionally only the rotational kinematic contribution. It is not the complete observed clock difference between the two altitudes.

12.4 Latitude dependence

At latitude λ, the spin-only contribution scales as

$$\mathcal{A}_{\mathrm{spin}}(\lambda, h) = \frac{\Omega_\oplus^2 (R_\oplus + h)^2 \cos^2 \lambda}{c^2}. \tag{98}$$

It is maximal at the equator and vanishes on the rotation axis. The orbital centre-motion term does not vanish at the poles; only the local spin component does.

12.5 Illustrative rescaled-Earth translation

Consider the conditional scaling discussed in the rod–flywheel model,

$$a' = 27 a_\oplus, \qquad R' = \frac{R_\oplus}{3}, \qquad \Omega' = 9\Omega_\oplus. \tag{99}$$

At leading weak-speed order, the native one-winding interval remains equal to the terrestrial value because

$$\Omega' R'^2 = 9\Omega_\oplus \frac{R_\oplus^2}{9} = \Omega_\oplus R_\oplus^2. \tag{100}$$

However, the equatorial speed becomes $v' = 3v_\oplus$, so the dimensionless label becomes, to leading order,

$$\mathscr{K}' \simeq 9\mathscr{K}_\oplus. \tag{101}$$

Its Earth-language translation is therefore

$$\Theta^{(\oplus)}_{\text{scaled}} \simeq 9\Theta_\oplus \simeq 1866.48\,\text{ns}_\oplus, \tag{102}$$

while its native-cycle translation remains approximately 207.386 ns because its native period is $T' = T_\oplus/9$. This example demonstrates why a native interval and an Earth-language comparison value must be kept distinct.

13 Natural Celestial Examples in Ledger Form

13.1 Mars

For a point on Mars, the same equations apply after replacing the terrestrial inputs by the Martian centre trajectory, rotation axis, angular rate, reference radius, latitude, altitude, and local phase. The mean Martian solar day follows from the signed phase difference

$$\omega_{\text{sol,Mars}} = \omega_{\text{sid,Mars}} - \omega_{\text{orb,Mars}}. \tag{103}$$

The local kinematic time field is not tied to the solar day definition; it is computed from the resultant velocity in the chosen reference frame.

13.2 Venus

Venus has a retrograde signed sidereal spin period. In the ledger, $\Omega_{\text{spin}} < 0$. The local spin velocity $\boldsymbol{\Omega} \times \boldsymbol{r}_P$ therefore reverses direction, changing the instantaneous cross term with orbital motion. The proper-time factor remains positive because it depends on the squared magnitude of the resultant velocity.

13.3 The Moon

For a lunar surface point, synchronous rotation is expressed by near equality between the lunar spin rate and the Moon's orbital phase rate about Earth. These two rates are distinct ledger entries even when their difference is nearly zero. The local velocity still contains the Moon-centre motion plus the surface spin term; phase locking does not erase either physical motion.

13.4 Differentially rotating bodies

For gas giants and stars, Ω_{spin} may depend on latitude, depth, or pattern. Then

$$\Omega_{\text{spin}} = \Omega_{\text{spin}}(\lambda, h, t), \tag{104}$$

and equation (68) remains valid pointwise after the appropriate local angular rate is supplied. A unique body-level spin number requires an explicit convention, whereas the local chronometric field does not.

14 Operational Calculation Protocol

A reproducible calculation for one point, one object, or an atlas catalogue proceeds as follows:

Step 1: Choose the quasi-inertial or Galactocentric reference frame, freeze all frame parameters, and state the coordinate time and epoch.

Step 2: Construct the immediate-parent phase chain and define signed unwrapped phases.

Step 3: For an atlas row, store the position, velocity, covariance, and provenance class separately; do not infer a unique velocity from position alone.

Step 4: Specify the position embedding $\boldsymbol{x}_P(\boldsymbol{q}; t)$, including all relevant radii, axes, inclinations, translations, and ordered rotations.

Step 5: Differentiate the embedding to obtain $\boldsymbol{v}_P$, or compute the phase metric $\mathsf{G}^{(P)}_{ab}$.

Step 6: Evaluate the angular-progress norm $\mathcal{A}_P = v_P^2/c^2$.

Step 7: Compute the dimensionless chronometric label $\mathscr{K}_P = \gamma_P \mathcal{A}_P$ and recover the local clock factor from equation (27).

Step 8: If an Earth-language comparison is required, compute $\mathscr{R}_{P/\oplus}$ and $\Theta_P^{(\oplus)} = \mathscr{K}_P T_{\mathrm{sid},\oplus}$.

Step 9: Integrate α_P over the desired coordinate-time interval when accumulated proper time, rather than an instantaneous label, is required.

Step 10: For comparisons of altitude, latitude, or nested motions, retain instantaneous vector cross terms unless a complete symmetric-cycle average is explicitly intended.

Step 11: If velocity or angular rates are not measured inputs, name and parameterise the separate closure law and mark the output as conditional.

Step 12: Propagate uncertainty from the input covariance or by Monte Carlo sampling.

Step 13: Record the frame, epoch, point convention, latitude, altitude, closure status, and whether the label is spin-only, centre-motion-only, or fully nested.

Step 14: Do not interpret the result as a complete gravitational clock prediction unless a metric-specific gravitational sector is added separately.

15 What Is Established and What Remains Conditional

15.1 Established within the stated kinematic framework

The following statements follow from the geometry and standard relativistic kinematics used here:

(1) Nested celestial cycles can be represented by a signed, unwrapped phase ledger.

(2) A point's resultant velocity is obtained from the geometric embedding of those phases.

(3) The correct combination of angular rates is a point-dependent quadratic form with cross terms.

(4) The local flat-spacetime kinematic proper time follows exactly from the resultant speed.

(5) Latitude, altitude, longitude, epoch, spin direction, and parent-system motion enter through explicit geometric variables.

(6) A complete spin-cycle average removes the centre–spin cross term under the stated symmetry assumptions.

(7) No gravitational potential is required to calculate this isolated kinematic sector.

(8) The dimensionless label $\mathscr{K}_P$ uniquely reconstructs the local kinematic clock factor in the selected frame.

(9) For a uniform rotating ring, $\mathscr{K} = |\mathcal{H}^*|/T_{\mathrm{rot}}$ exactly.

(10) The Earth-language map is an algebraically reproducible calibration once the canonical terrestrial ring is fixed.

(11) A complete Galactocentric six-dimensional state uniquely determines the isolated kinematic label in the declared frame.

(12) Cartesian and cylindrical representations give the same speed and therefore the same label when transformed consistently.

(13) A versioned catalogue can retain measured, incomplete, and conditionally completed rows without conflating their provenance.

15.2 Not established by the chronometric construction

The present framework does not establish that

(1) a scalar arithmetic sum of angles is identical to physical time;

(2) planetary spin and orbital rates are determined by distance and diameter alone;

(3) the rod–flywheel capacity law is an observed universal law of planetary systems;

(4) one universal function $\Phi(a/R)$ explains every planetary day;

(5) gravitational clock effects are absent from real measurements;

(6) strong-field compact objects can be treated by the flat-spacetime formula without metric-specific corrections;

(7) the selected quasi-inertial frame is an absolute state of rest;

(8) 207.386 ns is a universal time quantum or an observer-independent number;

(9) the Earth-language translation of a general point is automatically its physical one-winding holonomy;

(10) equality of native-cycle intervals alone guarantees equality of dimensionless labels or clock rates;

(11) one instantaneous Galactic position uniquely determines an object's velocity;

(12) a conditional Galactic rotation field is observationally measured merely because it supplies a velocity for a position-only row;

(13) a Galactocentric kinematic label is frame-independent or a complete replacement for Galactic gravitational dynamics.

16 Discussion

The main gain of the phase-metric formulation is conceptual economy. Spin, orbit, parent motion, and higher cycles are not forced into one arbitrary scalar. They remain separate in the ledger, while their physical effect on a selected point is combined by the geometry of the point's actual path.

This resolves three recurrent ambiguities.

First, equal angular velocities do not imply equal linear velocities. The local contribution of a phase is weighted by the appropriate effective radius contained in G_{aa}.

Second, the relative direction of two motions matters. The off-diagonal entries G_{ab} retain reinforcement and opposition through the cross terms.

Third, a body-level period does not by itself define the time rate at every location. The local radius from the rotation axis, latitude, altitude, and phase orientation must be supplied.

The framework is therefore naturally local while remaining nested. It can describe a surface clock, a mountain clock, a co-rotating atmospheric clock, a lunar marker, or a point in a multi-body chain with the same master equation.

The optional rod–flywheel closure addresses a different problem: how angular rates might be related to mass, size, and lever length under a specified transport capacity. Keeping this closure separate prevents a tentative mechanical hypothesis from being confused with the exact kinematic map from motion to proper time.

The dimensionless label introduces a second kind of economy. Instead of attaching a dimensional interval directly to every body, the framework first records $\mathscr{K}_X$. The same label

can then be displayed in its native-cycle language, in the canonical Earth-cycle language, or in another declared calibration. The translated number changes with the chosen calibration period; the underlying label does not. In this precise sense, 207.386 ns is a terrestrial translation of the canonical Earth label rather than the label itself.

This translation architecture also defines equivalence classes. Bodies or points with the same $\mathscr{K}$, evaluated under the same frame and point convention, have the same kinematic clock factor even when their radii, angular rates, and native periods differ. Conversely, equal one-cycle dimensional intervals can conceal different clock factors if the native cycle durations are different.

The Galactic extension changes the scale but not the logic. A catalogue row is first a phase-space address, not a scalar time tag. Only after a velocity has been measured or conditionally supplied does the row acquire $\mathscr{K}$. This prevents a colour map of position from being mistaken for a measured map of time rates. The resulting product is best understood as a *kinematic chronometric atlas*: a versioned map of frame-dependent phase-space labels, Earth-language translations, and uncertainties.

The atlas also provides a practical test of the framework. Objects at different positions but equal resultant speeds must occupy the same instantaneous $\mathscr{K}$ class in the selected frame. Objects at the same position but different velocities must not. Large catalogues such as Gaia DR3 allow these classifications to be generated for measured six-dimensional subsets, while incomplete rows remain visibly incomplete or are processed under explicitly named model closures.

17 Conclusion

A nested celestial phase ledger becomes chronometric only after its angular progressions are embedded in space. The embedding supplies a point-dependent phase metric,

$$\mathsf{G}^{(P)}_{ab} = \frac{\partial \boldsymbol{x}_P}{\partial q^a} \cdot \frac{\partial \boldsymbol{x}_P}{\partial q^b}, \tag{105}$$

and the resultant speed is

$$v_P^2 = \mathsf{G}^{(P)}_{ab} \dot{q}^a \dot{q}^b \tag{106}$$

when the motion is fully phase generated. The local gravity-independent kinematic proper time is therefore

$$\boxed{\mathrm{d}\tau_P = \mathrm{d}t \sqrt{1 - \frac{\mathsf{G}^{(P)}_{ab} \dot{q}^a \dot{q}^b}{c^2}}.} \tag{107}$$

For a point on or above a rotating planet,

$$\boxed{\boldsymbol{v}_P = \boldsymbol{V}_C + \boldsymbol{\Omega} \times \boldsymbol{r}_P, \qquad \frac{\mathrm{d}\tau_P}{\mathrm{d}t} = \sqrt{1 - \frac{|\boldsymbol{V}_C + \boldsymbol{\Omega} \times \boldsymbol{r}_P|^2}{c^2}}.} \tag{108}$$

This formula computes the kinematic time field at any specified latitude, altitude, longitude, and epoch once the centre motion and spin state are known.

The associated dimensionless chronometric label is

$$\boxed{\mathscr{K}_P = \gamma_P \frac{v_P^2}{c^2} = \frac{1}{\alpha_P} - \alpha_P,} \tag{109}$$

and its Earth-language translation is

$$\boxed{\Theta_P^{(\oplus)} = \mathscr{K}_P T_{\mathrm{sid},\oplus} = \frac{\mathscr{K}_P}{\mathscr{K}_\oplus}\Theta_\oplus, \qquad \Theta_\oplus \simeq 207.386\,\mathrm{ns}.} \tag{110}$$

The first quantity is dimensionless; the second is a declared terrestrial rendering of it.

At Galactic scale, the canonical data object is

$$\boxed{\mathfrak{A}_X(t_0 \mid \mathcal{F}_{\mathrm{G}}) = (t_0; \boldsymbol{x}_X, \boldsymbol{v}_X; \boldsymbol{\Sigma}_X, \mathcal{Q}_X)\,,} \tag{111}$$

with

$$\boxed{v_X^2 = v_R^2 + v_\phi^2 + v_z^2, \qquad \mathscr{K}_X^{\mathrm{G}} = \gamma_X^{\mathrm{G}} \frac{v_X^2}{c^2}.} \tag{112}$$

A position-only row does not receive a unique speed or label unless a separate, declared velocity-field closure is supplied. The catalogue therefore preserves the distinction among measured six-dimensional, transformed, incomplete, and model-completed addresses.

The disciplined chain is

$$\boxed{\begin{aligned}&\text{signed nested phase ledger and declared phase-space address} \longrightarrow \text{geometric phase metric}\\ &\quad\longrightarrow \text{resultant point velocity} \longrightarrow \begin{cases}\text{local kinematic proper time,}\\ \text{dimensionless label and declared translation.}\end{cases}\end{aligned}} \tag{113}$$

No gravitational variable is required for this chain. If a separate rod–flywheel model supplies the angular rates from mass, size, and lever length, those rates enter as inputs to the same chronometric map. The resulting architecture is therefore modular: one layer determines or measures angular progression and Galactic phase space, a second exact geometric layer translates that state into the kinematic component of time, and a third catalogue layer records calibration, provenance, and uncertainty without merging them into one unqualified number.

A Derivation in Turn-Count Coordinates

Since $q^a = 2\pi N_a$,

$$\frac{\partial \boldsymbol{x}_P}{\partial N_a} = 2\pi \frac{\partial \boldsymbol{x}_P}{\partial q^a}. \tag{114}$$

Define the turn-count metric

$$\widetilde{\mathsf{G}}_{ab}^{(P)} \equiv \frac{\partial \boldsymbol{x}_P}{\partial N_a} \cdot \frac{\partial \boldsymbol{x}_P}{\partial N_b} = 4\pi^2 \mathsf{G}_{ab}^{(P)}. \tag{115}$$

Then

$$\boxed{v_P^2 = \widetilde{\mathsf{G}}^{(P)}_{ab}\nu_a\nu_b, \qquad \nu_a = \dot{N}_a.} \tag{116}$$

The proper-time law may be written directly in ledger rates:

$$\boxed{\mathrm{d}\tau_P = \mathrm{d}t\sqrt{1 - \frac{\widetilde{\mathsf{G}}^{(P)}_{ab}\nu_a\nu_b}{c^2}}.} \tag{117}$$

B Exact Two-Phase Example

Let

$$\boldsymbol{x}(q^1, q^2) = r_1 \begin{pmatrix}\cos q^1 \\ \sin q^1\end{pmatrix} + r_2 \begin{pmatrix}\cos q^2 \\ \sin q^2\end{pmatrix}. \tag{118}$$

Then

$$\frac{\partial \boldsymbol{x}}{\partial q^1} = r_1 \begin{pmatrix}-\sin q^1 \\ \cos q^1\end{pmatrix}, \tag{119}$$

$$\frac{\partial \boldsymbol{x}}{\partial q^2} = r_2 \begin{pmatrix}-\sin q^2 \\ \cos q^2\end{pmatrix}. \tag{120}$$

Therefore

$$\mathsf{G} = \begin{pmatrix} r_1^2 & r_1 r_2 \cos(q^1 - q^2) \\ r_1 r_2 \cos(q^1 - q^2) & r_2^2 \end{pmatrix}, \tag{121}$$

and

$$v^2 = r_1^2\omega_1^2 + r_2^2\omega_2^2 + 2r_1 r_2 \omega_1\omega_2 \cos(q^1 - q^2). \tag{122}$$

This explicit example shows that the cross term is generated by the geometry of the phase embedding and not added by hand.

C Relation to a Rotating-Ring Holonomy

For one uniformly rotating ring of radius R and angular velocity Ω, the local speed is $v = |\Omega|R$ and

$$\frac{\mathrm{d}\tau}{\mathrm{d}t} = \sqrt{1 - \frac{\Omega^2 R^2}{c^2}}. \tag{123}$$

A slow directed transport around the same ring yields the standard one-winding time holonomy

$$\mathcal{H}^* = -\gamma \frac{2\pi\Omega R^2}{c^2}, \qquad \gamma = \left(1 - \frac{\Omega^2 R^2}{c^2}\right)^{-1/2}. \tag{124}$$

Since $T_{\mathrm{rot}} = 2\pi/|\Omega|$, the magnitude obeys

$$\boxed{\frac{|\mathcal{H}^*|}{T_{\mathrm{rot}}} = \gamma \frac{\Omega^2 R^2}{c^2} = \mathscr{K}_{\mathrm{ring}}.} \tag{125}$$

Thus the dimensionless chronometric label is exactly the period-normalised one-direction holonomy for a uniform ring. The phase-metric framework is more general because it applies to open intervals, arbitrary points, multiple nested phases, and non-closed trajectories. For those more general cases, $\mathscr{K}_P$ remains a local rate label, whereas multiplying it by a calibration period is a declared translation rather than an automatically realised loop observable.

D Stable Numerical Formulae

For a known speed v with $\beta = v/c \ll 1$,

$$1 - \sqrt{1-\beta^2} = \frac{\beta^2}{1+\sqrt{1-\beta^2}}, \tag{126}$$

which avoids subtractive loss of precision. For a time series sampled at t_i, the accumulated proper time may be approximated by

$$\tau_P \simeq \sum_i \Delta t_i \sqrt{1 - \frac{v_P^2(t_i)}{c^2}}, \tag{127}$$

or by a higher-order quadrature when ephemeris accuracy requires it.

For a label $\mathscr{K}$, the stable inverse clock factor is

$$\alpha = \frac{2}{\sqrt{\mathscr{K}^2+4}+\mathscr{K}}. \tag{128}$$

The Earth-language translation in nanoseconds is

$$\Theta_X^{(\oplus)}[\mathrm{ns}_\oplus] = 10^9\, T_{\mathrm{sid},\oplus}\mathscr{K}_X. \tag{129}$$

E Reference Python Implementations

The accompanying source package includes two executable files. The first, `chronometric_label_calculator.py`, computes canonical spin-only and local nested labels. The second, `galactic_chronometric_atlas.py`, ingests single phase-space states or catalogue CSV files, converts Cartesian addresses to cylindrical components, distinguishes measured from conditionally completed rows, computes $\mathscr{K}$, and writes Earth-language atlas tables. Its optional ICRS mode uses Astropy with explicit Galactocentric parameters. The package also includes Cartesian and Gaia-style CSV templates plus example measured and conditional outputs. Neither implementation includes a gravitational sector.

Listing 1: Local dimensionless chronometric-label and Earth-language translation calculator.

```
#!/usr/bin/env python3
"""
Dimensionless chronometric-label and Earth-language translation calculator.

The calculator implements the gravity-independent kinematic sector developed in
"Nested Angular Progress and Local Kinematic Time".

```

```
Core definitions
----------------
    beta = v / c
    gamma = 1 / sqrt(1 - beta**2)
    K = gamma * beta**2 = gamma - 1/gamma
    alpha = d_tau/dt = 1/gamma

For Earth's canonical equatorial rotating ring,
    K_E = gamma_E * (Omega_E R_E / c)**2
and the Earth-language calibration interval is
    Theta_E = K_E * T_E ~= 207.386 ns.

For another object or point X, its Earth-language translation is
    Theta_X^(E) = K_X * T_E
                = (K_X / K_E) * Theta_E.

This translated interval is a comparison convention. It is not automatically
the physical one-winding holonomy of X. For a uniformly rotating canonical
ring, the native one-winding holonomy magnitude is K_X * T_X.

No gravitational potential or gravitational redshift is included.
"""

from __future__ import annotations

import argparse
import math
from dataclasses import dataclass, asdict
from typing import Any

C = 299_792_458.0  # m/s, exact SI value

# Canonical Earth calibration adopted in the article.
EARTH_RADIUS_M = 6_378_137.0
EARTH_SIDEREAL_PERIOD_S = 86_164.09054


@dataclass(frozen=True)
class ChronometricRecord:
    """Computed dimensionless label and translation quantities."""

    name: str
    speed_m_s: float
    beta: float
    gamma: float
    clock_factor_alpha: float
    label_K: float
    earth_normalized_ratio: float
    earth_language_seconds: float
    earth_language_nanoseconds: float
    native_period_s: float | None
    native_cycle_equivalent_seconds: float | None
    native_cycle_equivalent_nanoseconds: float | None
    native_to_earth_conversion: float | None


def angular_velocity(rotation_period_s: float) -> float:
    """Return signed angular velocity in rad/s."""
    if rotation_period_s == 0.0:
        raise ValueError("rotation_period_s cannot be zero")
    return 2.0 * math.pi / rotation_period_s


def cylindrical_radius(
    body_radius_m: float,
    altitude_m: float = 0.0,
    latitude_deg: float = 0.0,
) -> float:
    """Distance from the rotation axis: rho = (R + h) cos(latitude)."""
```

```
    if body_radius_m <= 0.0:
        raise ValueError("body_radius_m must be positive")
    if body_radius_m + altitude_m <= 0.0:
        raise ValueError("body_radius_m + altitude_m must be positive")
    if not -90.0 <= latitude_deg <= 90.0:
        raise ValueError("latitude_deg must lie between -90 and +90")
    return (body_radius_m + altitude_m) * math.cos(math.radians(latitude_deg))


def spin_speed(
    body_radius_m: float,
    rotation_period_s: float,
    altitude_m: float = 0.0,
    latitude_deg: float = 0.0,
) -> float:
    """Return signed local spin speed Omega*rho in m/s."""
    omega = angular_velocity(rotation_period_s)
    rho = cylindrical_radius(body_radius_m, altitude_m, latitude_deg)
    return omega * rho


def resultant_speed(
    center_speed_m_s: float,
    local_spin_speed_m_s: float,
    alignment_angle_deg: float,
) -> float:
    """
    Magnitude of V_center + v_spin using their included angle psi.

    v^2 = V^2 + v_spin^2 + 2 V v_spin cos(psi)

    The sign of local_spin_speed_m_s is retained, so retrograde rotation can be
    represented either by a negative rotation period or by the chosen angle.
    """
    if center_speed_m_s < 0.0:
        raise ValueError("center_speed_m_s must be non-negative")
    psi = math.radians(alignment_angle_deg)
    v2 = (
        center_speed_m_s**2
        + local_spin_speed_m_s**2
        + 2.0
        * center_speed_m_s
        * local_spin_speed_m_s
        * math.cos(psi)
    )
    # Protect against tiny negative roundoff in an exact cancellation case.
    return math.sqrt(max(v2, 0.0))


def dimensionless_label_from_speed(speed_m_s: float) -> tuple[float, float, float, float]:
    """Return beta, gamma, alpha, and K for a speed magnitude."""
    if speed_m_s < 0.0:
        raise ValueError("speed_m_s must be non-negative")
    beta = speed_m_s / C
    if beta >= 1.0:
        raise ValueError("speed must be smaller than c")
    gamma = 1.0 / math.sqrt(1.0 - beta * beta)
    alpha = 1.0 / gamma
    label_k = gamma * beta * beta
    return beta, gamma, alpha, label_k


def clock_factor_from_label(label_k: float) -> float:
    """Exact inverse map alpha(K) for K >= 0."""
    if label_k < 0.0:
        raise ValueError("label_k must be non-negative")
    # Stable form of (sqrt(K^2 + 4) - K)/2.
    return 2.0 / (math.sqrt(label_k * label_k + 4.0) + label_k)

```

```
146
147 def earth_calibration() -> dict[str, float]:
148     """Return Earth's canonical equatorial calibration constants."""
149     v_earth = abs(
150         spin_speed(
151             body_radius_m=EARTH_RADIUS_M,
152             rotation_period_s=EARTH_SIDEREAL_PERIOD_S,
153         )
154     )
155     beta, gamma, alpha, label_k = dimensionless_label_from_speed(v_earth)
156     theta_s = label_k * EARTH_SIDEREAL_PERIOD_S
157     return {
158         "speed_m_s": v_earth,
159         "beta": beta,
160         "gamma": gamma,
161         "alpha": alpha,
162         "label_K": label_k,
163         "translation_seconds": theta_s,
164         "translation_nanoseconds": theta_s * 1.0e9,
165     }
166
167
168 EARTH = earth_calibration()
169 EARTH_LABEL_K = EARTH["label_K"]
170 EARTH_TRANSLATION_S = EARTH["translation_seconds"]
171 EARTH_TRANSLATION_NS = EARTH["translation_nanoseconds"]
172
173
174 def make_record(
175     *,
176     name: str,
177     speed_m_s: float,
178     native_period_s: float | None = None,
179 ) -> ChronometricRecord:
180     """Create a complete label and Earth-translation record from speed."""
181     beta, gamma, alpha, label_k = dimensionless_label_from_speed(speed_m_s)
182     ratio = label_k / EARTH_LABEL_K
183     earth_s = label_k * EARTH_SIDEREAL_PERIOD_S
184
185     if native_period_s is None:
186         native_s = None
187         native_ns = None
188         conversion = None
189     else:
190         period = abs(native_period_s)
191         if period == 0.0:
192             raise ValueError("native_period_s cannot be zero")
193         native_s = label_k * period
194         native_ns = native_s * 1.0e9
195         conversion = EARTH_SIDEREAL_PERIOD_S / period
196
197     return ChronometricRecord(
198         name=name,
199         speed_m_s=speed_m_s,
200         beta=beta,
201         gamma=gamma,
202         clock_factor_alpha=alpha,
203         label_K=label_k,
204         earth_normalized_ratio=ratio,
205         earth_language_seconds=earth_s,
206         earth_language_nanoseconds=earth_s * 1.0e9,
207         native_period_s=abs(native_period_s) if native_period_s is not None else None,
208         native_cycle_equivalent_seconds=native_s,
209         native_cycle_equivalent_nanoseconds=native_ns,
210         native_to_earth_conversion=conversion,
211     )
212
213
214 def canonical_rotating_body_record(
```

```
    *,
    name: str,
    radius_m: float,
    rotation_period_s: float,
    latitude_deg: float = 0.0,
    altitude_m: float = 0.0,
) -> ChronometricRecord:
    """
    Canonical spin-only record for a point fixed to a rotating body.

    For the standard body label, use latitude_deg=0 and altitude_m=0.
    """
    v_spin = abs(
        spin_speed(
            body_radius_m=radius_m,
            rotation_period_s=rotation_period_s,
            altitude_m=altitude_m,
            latitude_deg=latitude_deg,
        )
    )
    return make_record(
        name=name,
        speed_m_s=v_spin,
        native_period_s=rotation_period_s,
    )


def local_resultant_record(
    *,
    name: str,
    radius_m: float,
    rotation_period_s: float,
    center_speed_m_s: float,
    alignment_angle_deg: float,
    latitude_deg: float = 0.0,
    altitude_m: float = 0.0,
) -> ChronometricRecord:
    """Instantaneous local label including centre motion and local spin."""
    v_spin = spin_speed(
        body_radius_m=radius_m,
        rotation_period_s=rotation_period_s,
        altitude_m=altitude_m,
        latitude_deg=latitude_deg,
    )
    v_total = resultant_speed(
        center_speed_m_s=center_speed_m_s,
        local_spin_speed_m_s=v_spin,
        alignment_angle_deg=alignment_angle_deg,
    )
    # The native-cycle field here is only a translation convention. If the
    # alignment changes during a cycle, it is not an accumulated physical loop
    # observable; integrate alpha(t) for that calculation.
    return make_record(
        name=name,
        speed_m_s=v_total,
        native_period_s=rotation_period_s,
    )


def print_record(record: ChronometricRecord) -> None:
    """Print one record in a compact scientific format."""
    print("=" * 72)
    print(record.name)
    print("=" * 72)
    print(f"speed                          = {record.speed_m_s:.12g} m/s")
    print(f"beta                           = {record.beta:.12e}")
    print(f"gamma                          = {record.gamma:.16g}")
    print(f"clock factor alpha             = {record.clock_factor_alpha:.16g}")
    print(f"dimensionless label K          = {record.label_K:.12e}")
```

```
    print(f"Earth-normalized ratio        = {record.earth_normalized_ratio:.12e}")
    print(
        "Earth-language translation     = "
        f"{record.earth_language_nanoseconds:.12g} ns_E"
    )
    if record.native_period_s is not None:
        print(f"native period                  = {record.native_period_s:.12g} s")
        print(
            "native-cycle equivalent       = "
            f"{record.native_cycle_equivalent_nanoseconds:.12g} ns"
        )
        print(
            "native -> Earth coefficient   = "
            f"{record.native_to_earth_conversion:.12e}"
        )
    print()


def record_as_dict(record: ChronometricRecord) -> dict[str, Any]:
    """Return a serialization-friendly dictionary."""
    return asdict(record)


def build_argument_parser() -> argparse.ArgumentParser:
    parser = argparse.ArgumentParser(
        description=(
            "Calculate a dimensionless kinematic chronometric label and its "
            "Earth-language translation."
        )
    )
    parser.add_argument("--name", default="Custom body")
    parser.add_argument("--radius-km", type=float, required=True)
    parser.add_argument(
        "--rotation-period-hours",
        type=float,
        required=True,
        help="Use a negative value for retrograde rotation.",
    )
    parser.add_argument("--latitude-deg", type=float, default=0.0)
    parser.add_argument("--altitude-km", type=float, default=0.0)
    parser.add_argument(
        "--center-speed-km-s",
        type=float,
        default=0.0,
        help="Optional centre-of-body speed in the selected reference frame.",
    )
    parser.add_argument(
        "--alignment-angle-deg",
        type=float,
        default=90.0,
        help="Angle between centre velocity and signed local spin velocity.",
    )
    return parser


def main() -> None:
    parser = build_argument_parser()
    args = parser.parse_args()

    radius_m = args.radius_km * 1_000.0
    period_s = args.rotation_period_hours * 3_600.0
    altitude_m = args.altitude_km * 1_000.0
    center_speed = args.center_speed_km_s * 1_000.0

    print("Earth calibration")
    print(f"K_E       = {EARTH_LABEL_K:.12e}")
    print(f"Theta_E   = {EARTH_TRANSLATION_NS:.12f} ns_E")
    print()

```

```
    if center_speed == 0.0:
        record = canonical_rotating_body_record(
            name=args.name,
            radius_m=radius_m,
            rotation_period_s=period_s,
            latitude_deg=args.latitude_deg,
            altitude_m=altitude_m,
        )
    else:
        record = local_resultant_record(
            name=args.name,
            radius_m=radius_m,
            rotation_period_s=period_s,
            center_speed_m_s=center_speed,
            alignment_angle_deg=args.alignment_angle_deg,
            latitude_deg=args.latitude_deg,
            altitude_m=altitude_m,
        )

    print_record(record)


if __name__ == "__main__":
    main()
```

Listing 2: Galactic phase-space chronometric atlas generator.

```
#!/usr/bin/env python3
"""
Galactic phase-space chronometric atlas generator.

This script implements the gravity-independent kinematic sector developed in
"Nested Angular Progress, Galactic Phase-Space Addressing, and Earth-Language
Chronometric Labels".

The fundamental catalogue input is a declared Galactocentric phase-space state
at one epoch:

    (x, y, z, vx, vy, vz)

Position alone does not determine velocity.  Rows without measured velocities
are therefore either marked INCOMPLETE or completed by an explicitly selected
conditional velocity-field closure.

For each complete row the script calculates:

    R, phi, z, v_R, v_phi, v_z, |v|,
    beta, gamma, alpha=d_tau/dt,
    K=gamma*beta^2,
    K/K_Earth,
    Theta_Earth=K*T_sidereal_Earth.

The Earth-language interval is a comparison coordinate.  It is not claimed to
be a native physical loop holonomy unless a separate closed-loop construction
applies to the object being represented.

No gravitational potential or gravitational redshift is included.

Optional dependency:
    astropy -- only required for ICRS/Gaia-style sky-coordinate transformation.

Author: S. M. H. Emamifar
Affiliation: IRCBHC
"""

from __future__ import annotations

import argparse
```

```
import csv
import json
import math
import random
import statistics
from dataclasses import asdict, dataclass
from pathlib import Path
from typing import Any, Iterable

C_M_S = 299_792_458.0
KPC_M = 3.085_677_581_491_367e19
KM_S_TO_M_S = 1_000.0

EARTH_RADIUS_M = 6_378_137.0
EARTH_SIDEREAL_PERIOD_S = 86_164.09054
EARTH_OMEGA_RAD_S = 2.0 * math.pi / EARTH_SIDEREAL_PERIOD_S
EARTH_EQUATORIAL_SPEED_M_S = EARTH_OMEGA_RAD_S * EARTH_RADIUS_M

# Explicit illustrative Galactocentric frame parameters.  They match the
# Astropy v4.0 example convention and are not asserted to be a unique standard.
DEFAULT_GALCEN_DISTANCE_KPC = 8.122
DEFAULT_Z_SUN_PC = 20.8
DEFAULT_V_SUN_KM_S = (12.9, 245.6, 7.78)


@dataclass(frozen=True)
class Vector3:
    x: float
    y: float
    z: float

    def norm(self) -> float:
        return math.sqrt(self.x * self.x + self.y * self.y + self.z * self.z)

    def __add__(self, other: "Vector3") -> "Vector3":
        return Vector3(self.x + other.x, self.y + other.y, self.z + other.z)

    def scale(self, factor: float) -> "Vector3":
        return Vector3(self.x * factor, self.y * factor, self.z * factor)


@dataclass(frozen=True)
class LabelState:
    speed_m_s: float
    beta: float
    gamma: float
    alpha: float
    label_K: float
    earth_ratio: float
    earth_translation_s: float
    earth_translation_ns: float


@dataclass(frozen=True)
class GalacticChronometricRecord:
    name: str
    epoch: str
    frame_name: str
    quality_class: str
    closure_name: str
    x_kpc: float
    y_kpc: float
    z_kpc: float
    R_kpc: float
    phi_deg: float
    vx_km_s: float | None
    vy_km_s: float | None
    vz_km_s: float | None
    vR_km_s: float | None
```

```
    vphi_km_s: float | None
    speed_km_s: float | None
    beta: float | None
    gamma: float | None
    alpha: float | None
    label_K: float | None
    earth_ratio: float | None
    earth_translation_s: float | None
    earth_translation_ms: float | None
    label_K_sigma: float | None
    earth_translation_ms_sigma: float | None
    notes: str


def label_from_speed_m_s(speed_m_s: float) -> LabelState:
    """Calculate the exact dimensionless label and Earth translation."""
    if speed_m_s < 0.0:
        raise ValueError("speed_m_s must be non-negative")
    beta = speed_m_s / C_M_S
    if beta >= 1.0:
        raise ValueError("speed must be smaller than c")
    gamma = 1.0 / math.sqrt(1.0 - beta * beta)
    alpha = 1.0 / gamma
    label_k = gamma * beta * beta
    return LabelState(
        speed_m_s=speed_m_s,
        beta=beta,
        gamma=gamma,
        alpha=alpha,
        label_K=label_k,
        earth_ratio=label_k / EARTH_LABEL.label_K,
        earth_translation_s=label_k * EARTH_SIDEREAL_PERIOD_S,
        earth_translation_ns=label_k * EARTH_SIDEREAL_PERIOD_S * 1.0e9,
    )


# Bootstrap Earth after the function exists.  The temporary calculation is
# written explicitly to avoid recursive access to EARTH_LABEL.
def _earth_label() -> LabelState:
    beta = EARTH_EQUATORIAL_SPEED_M_S / C_M_S
    gamma = 1.0 / math.sqrt(1.0 - beta * beta)
    alpha = 1.0 / gamma
    label_k = gamma * beta * beta
    return LabelState(
        speed_m_s=EARTH_EQUATORIAL_SPEED_M_S,
        beta=beta,
        gamma=gamma,
        alpha=alpha,
        label_K=label_k,
        earth_ratio=1.0,
        earth_translation_s=label_k * EARTH_SIDEREAL_PERIOD_S,
        earth_translation_ns=label_k * EARTH_SIDEREAL_PERIOD_S * 1.0e9,
    )


EARTH_LABEL = _earth_label()


def cartesian_to_cylindrical(
    position_kpc: Vector3,
    velocity_km_s: Vector3 | None,
) -> tuple[float, float, float | None, float | None]:
    """Return R, phi(deg), v_R and v_phi in a right-handed frame."""
    x, y = position_kpc.x, position_kpc.y
    radius = math.hypot(x, y)
    phi_deg = math.degrees(math.atan2(y, x))
    if velocity_km_s is None:
        return radius, phi_deg, None, None
    if radius == 0.0:
```

```
        return radius, phi_deg, None, None
    v_r = (x * velocity_km_s.x + y * velocity_km_s.y) / radius
    v_phi = (x * velocity_km_s.y - y * velocity_km_s.x) / radius
    return radius, phi_deg, v_r, v_phi


def flat_rotation_closure(position_kpc: Vector3, speed_km_s: float) -> Vector3:
    """Conditional circular field with constant signed tangential speed."""
    radius = math.hypot(position_kpc.x, position_kpc.y)
    if radius == 0.0:
        raise ValueError("flat rotation closure is undefined at R=0")
    return Vector3(
        -speed_km_s * position_kpc.y / radius,
        speed_km_s * position_kpc.x / radius,
        0.0,
    )


def solid_rotation_closure(
    position_kpc: Vector3,
    omega_km_s_kpc: float,
) -> Vector3:
    """Conditional solid-rotation field v_phi=Omega*R."""
    return Vector3(
        -omega_km_s_kpc * position_kpc.y,
        omega_km_s_kpc * position_kpc.x,
        0.0,
    )


def apply_velocity_closure(
    position_kpc: Vector3,
    closure: str,
    flat_speed_km_s: float,
    solid_omega_km_s_kpc: float,
) -> tuple[Vector3 | None, str, str]:
    if closure == "none":
        return None, "P3_POSITION_ONLY", "none"
    if closure == "flat":
        return (
            flat_rotation_closure(position_kpc, flat_speed_km_s),
            "C6_MODEL_CLOSED",
            f"flat_rotation({flat_speed_km_s:g} km/s)",
        )
    if closure == "solid":
        return (
            solid_rotation_closure(position_kpc, solid_omega_km_s_kpc),
            "C6_MODEL_CLOSED",
            f"solid_rotation({solid_omega_km_s_kpc:g} km/s/kpc)",
        )
    raise ValueError(f"unsupported closure: {closure}")


def monte_carlo_label_sigma(
    velocity_km_s: Vector3,
    sigma_km_s: Vector3,
    samples: int,
    seed: int,
) -> tuple[float | None, float | None]:
    """Diagonal-error Monte Carlo. Full covariance should be used when available."""
    if samples <= 1:
        return None, None
    if sigma_km_s.x < 0 or sigma_km_s.y < 0 or sigma_km_s.z < 0:
        raise ValueError("velocity uncertainties must be non-negative")
    if sigma_km_s.norm() == 0.0:
        return 0.0, 0.0
    rng = random.Random(seed)
    labels: list[float] = []
    translations_ms: list[float] = []
```

```
    for _ in range(samples):
        draw = Vector3(
            rng.gauss(velocity_km_s.x, sigma_km_s.x),
            rng.gauss(velocity_km_s.y, sigma_km_s.y),
            rng.gauss(velocity_km_s.z, sigma_km_s.z),
        )
        state = label_from_speed_m_s(draw.norm() * KM_S_TO_M_S)
        labels.append(state.label_K)
        translations_ms.append(state.earth_translation_s * 1.0e3)
    return statistics.stdev(labels), statistics.stdev(translations_ms)


def build_record(
    *,
    name: str,
    epoch: str,
    frame_name: str,
    position_kpc: Vector3,
    velocity_km_s: Vector3 | None,
    quality_class: str,
    closure_name: str,
    velocity_sigma_km_s: Vector3 | None = None,
    mc_samples: int = 0,
    seed: int = 20260722,
    notes: str = "",
) -> GalacticChronometricRecord:
    R_kpc, phi_deg, v_r, v_phi = cartesian_to_cylindrical(
        position_kpc, velocity_km_s
    )

    if velocity_km_s is None:
        return GalacticChronometricRecord(
            name=name,
            epoch=epoch,
            frame_name=frame_name,
            quality_class=quality_class,
            closure_name=closure_name,
            x_kpc=position_kpc.x,
            y_kpc=position_kpc.y,
            z_kpc=position_kpc.z,
            R_kpc=R_kpc,
            phi_deg=phi_deg,
            vx_km_s=None,
            vy_km_s=None,
            vz_km_s=None,
            vR_km_s=None,
            vphi_km_s=None,
            speed_km_s=None,
            beta=None,
            gamma=None,
            alpha=None,
            label_K=None,
            earth_ratio=None,
            earth_translation_s=None,
            earth_translation_ms=None,
            label_K_sigma=None,
            earth_translation_ms_sigma=None,
            notes=notes,
        )

    speed_km_s = velocity_km_s.norm()
    state = label_from_speed_m_s(speed_km_s * KM_S_TO_M_S)
    if velocity_sigma_km_s is None:
        k_sigma, theta_sigma = None, None
    else:
        k_sigma, theta_sigma = monte_carlo_label_sigma(
            velocity_km_s,
            velocity_sigma_km_s,
            mc_samples,
```

```
            seed,
        )

    return GalacticChronometricRecord(
        name=name,
        epoch=epoch,
        frame_name=frame_name,
        quality_class=quality_class,
        closure_name=closure_name,
        x_kpc=position_kpc.x,
        y_kpc=position_kpc.y,
        z_kpc=position_kpc.z,
        R_kpc=R_kpc,
        phi_deg=phi_deg,
        vx_km_s=velocity_km_s.x,
        vy_km_s=velocity_km_s.y,
        vz_km_s=velocity_km_s.z,
        vR_km_s=v_r,
        vphi_km_s=v_phi,
        speed_km_s=speed_km_s,
        beta=state.beta,
        gamma=state.gamma,
        alpha=state.alpha,
        label_K=state.label_K,
        earth_ratio=state.earth_ratio,
        earth_translation_s=state.earth_translation_s,
        earth_translation_ms=state.earth_translation_s * 1.0e3,
        label_K_sigma=k_sigma,
        earth_translation_ms_sigma=theta_sigma,
        notes=notes,
    )


def parse_optional_float(row: dict[str, str], key: str) -> float | None:
    value = (row.get(key) or "").strip()
    if value == "":
        return None
    return float(value)


def records_from_cartesian_csv(
    input_path: Path,
    *,
    closure: str,
    flat_speed_km_s: float,
    solid_omega_km_s_kpc: float,
    frame_name: str,
    default_epoch: str,
    mc_samples: int,
) -> list[GalacticChronometricRecord]:
    records: list[GalacticChronometricRecord] = []
    with input_path.open("r", encoding="utf-8-sig", newline="") as handle:
        reader = csv.DictReader(handle)
        required = {"name", "x_kpc", "y_kpc", "z_kpc"}
        missing = required.difference(reader.fieldnames or [])
        if missing:
            raise ValueError(f"missing CSV columns: {sorted(missing)}")

        for index, row in enumerate(reader, start=1):
            position = Vector3(
                float(row["x_kpc"]),
                float(row["y_kpc"]),
                float(row["z_kpc"]),
            )
            vx = parse_optional_float(row, "vx_km_s")
            vy = parse_optional_float(row, "vy_km_s")
            vz = parse_optional_float(row, "vz_km_s")
            velocity_values = (vx, vy, vz)

```

```
            if all(value is not None for value in velocity_values):
                velocity = Vector3(float(vx), float(vy), float(vz))
                quality = "M6_MEASURED"
                closure_name = "none"
            elif all(value is None for value in velocity_values):
                velocity, quality, closure_name = apply_velocity_closure(
                    position,
                    closure,
                    flat_speed_km_s,
                    solid_omega_km_s_kpc,
                )
            else:
                velocity = None
                quality = "P5_PARTIAL_VELOCITY"
                closure_name = "none"

            svx = parse_optional_float(row, "vx_error_km_s") or 0.0
            svy = parse_optional_float(row, "vy_error_km_s") or 0.0
            svz = parse_optional_float(row, "vz_error_km_s") or 0.0
            sigma = Vector3(svx, svy, svz) if velocity is not None else None

            records.append(
                build_record(
                    name=(row.get("name") or f"row-{index}").strip(),
                    epoch=(row.get("epoch") or default_epoch).strip(),
                    frame_name=(row.get("frame_name") or frame_name).strip(),
                    position_kpc=position,
                    velocity_km_s=velocity,
                    quality_class=quality,
                    closure_name=closure_name,
                    velocity_sigma_km_s=sigma,
                    mc_samples=mc_samples,
                    seed=20260722 + index,
                    notes=(row.get("notes") or "").strip(),
                )
            )
    return records


def icrs_to_galactocentric(
    *,
    ra_deg: float,
    dec_deg: float,
    distance_pc: float,
    pmra_mas_yr: float,
    pmdec_mas_yr: float,
    radial_velocity_km_s: float,
    galcen_distance_kpc: float = DEFAULT_GALCEN_DISTANCE_KPC,
    z_sun_pc: float = DEFAULT_Z_SUN_PC,
    v_sun_km_s: tuple[float, float, float] = DEFAULT_V_SUN_KM_S,
) -> tuple[Vector3, Vector3]:
    """Transform one full ICRS phase-space state using explicit Astropy settings."""
    try:
        import astropy.coordinates as coord
        import astropy.units as u
    except ImportError as exc:
        raise RuntimeError(
            "Astropy is required for ICRS transformation. Install with: "
            "pip install astropy"
        ) from exc

    sky = coord.SkyCoord(
        ra=ra_deg * u.deg,
        dec=dec_deg * u.deg,
        distance=distance_pc * u.pc,
        pm_ra_cosdec=pmra_mas_yr * u.mas / u.yr,
        pm_dec=pmdec_mas_yr * u.mas / u.yr,
        radial_velocity=radial_velocity_km_s * u.km / u.s,
        frame="icrs",
```

```
    )
    velocity_components = tuple(
        component * u.km / u.s for component in v_sun_km_s
    )
    # Astropy versions before 8 use CartesianDifferential here, whereas
    # Astropy 8+ accepts CartesianRepresentation.  Keep both paths explicit.
    try:
        solar_velocity = coord.CartesianRepresentation(*velocity_components)
        frame = coord.Galactocentric(
            galcen_distance=galcen_distance_kpc * u.kpc,
            z_sun=z_sun_pc * u.pc,
            galcen_v_sun=solar_velocity,
        )
    except (TypeError, ValueError):
        solar_velocity = coord.CartesianDifferential(*velocity_components)
        frame = coord.Galactocentric(
            galcen_distance=galcen_distance_kpc * u.kpc,
            z_sun=z_sun_pc * u.pc,
            galcen_v_sun=solar_velocity,
        )
    gc = sky.transform_to(frame)
    position = Vector3(
        gc.x.to_value(u.kpc),
        gc.y.to_value(u.kpc),
        gc.z.to_value(u.kpc),
    )
    velocity = Vector3(
        gc.v_x.to_value(u.km / u.s),
        gc.v_y.to_value(u.km / u.s),
        gc.v_z.to_value(u.km / u.s),
    )
    return position, velocity


def records_from_icrs_csv(
    input_path: Path,
    *,
    frame_name: str,
    default_epoch: str,
) -> list[GalacticChronometricRecord]:
    records: list[GalacticChronometricRecord] = []
    with input_path.open("r", encoding="utf-8-sig", newline="") as handle:
        reader = csv.DictReader(handle)
        required = {
            "name",
            "ra_deg",
            "dec_deg",
            "distance_pc",
            "pmra_mas_yr",
            "pmdec_mas_yr",
            "radial_velocity_km_s",
        }
        missing = required.difference(reader.fieldnames or [])
        if missing:
            raise ValueError(f"missing ICRS CSV columns: {sorted(missing)}")
        for row in reader:
            position, velocity = icrs_to_galactocentric(
                ra_deg=float(row["ra_deg"]),
                dec_deg=float(row["dec_deg"]),
                distance_pc=float(row["distance_pc"]),
                pmra_mas_yr=float(row["pmra_mas_yr"]),
                pmdec_mas_yr=float(row["pmdec_mas_yr"]),
                radial_velocity_km_s=float(row["radial_velocity_km_s"]),
            )
            records.append(
                build_record(
                    name=row["name"].strip(),
                    epoch=(row.get("epoch") or default_epoch).strip(),
                    frame_name=frame_name,
```

```
                    position_kpc=position,
                    velocity_km_s=velocity,
                    quality_class="T6_TRANSFORMED",
                    closure_name="none",
                    notes=(row.get("notes") or "").strip(),
                )
            )
    return records


def write_records_csv(records: Iterable[GalacticChronometricRecord], path: Path) -> None:
    rows = [asdict(record) for record in records]
    if not rows:
        raise ValueError("no records to write")
    with path.open("w", encoding="utf-8", newline="") as handle:
        writer = csv.DictWriter(handle, fieldnames=list(rows[0].keys()))
        writer.writeheader()
        writer.writerows(rows)


def write_records_json(records: Iterable[GalacticChronometricRecord], path: Path) -> None:
    with path.open("w", encoding="utf-8") as handle:
        json.dump([asdict(record) for record in records], handle, indent=2)


def print_summary(records: list[GalacticChronometricRecord]) -> None:
    print("Earth canonical calibration")
    print(f"  K_Earth    = {EARTH_LABEL.label_K:.12e}")
    print(f"  Theta_Earth= {EARTH_LABEL.earth_translation_ns:.12f} ns_E")
    print()
    header = (
        f"{'name':24s} {'quality':25s} {'speed km/s':>12s} "
        f"{'K':>13s} {'K/K_E':>12s} {'Theta_E ms':>12s}"
    )
    print(header)
    print("-" * len(header))
    for record in records:
        if record.speed_km_s is None:
            speed = k_value = ratio = theta = "--"
        else:
            speed = f"{record.speed_km_s:.6g}"
            k_value = f"{record.label_K:.6e}"
            ratio = f"{record.earth_ratio:.6e}"
            theta = f"{record.earth_translation_ms:.6g}"
        print(
            f"{record.name[:24]:24s} {record.quality_class[:25]:25s} "
            f"{speed:>12s} {k_value:>13s} {ratio:>12s} {theta:>12s}"
        )


def build_parser() -> argparse.ArgumentParser:
    parser = argparse.ArgumentParser(
        description="Build a gravity-independent Galactic chronometric atlas."
    )
    subparsers = parser.add_subparsers(dest="command", required=True)

    cart = subparsers.add_parser("cartesian-csv", help="Process Galactocentric Cartesian CSV")
    cart.add_argument("input_csv", type=Path)
    cart.add_argument("--output-csv", type=Path, default=Path("atlas_output.csv"))
    cart.add_argument("--output-json", type=Path)
    cart.add_argument("--frame-name", default="Declared Galactocentric frame")
    cart.add_argument("--epoch", default="J2016.0")
    cart.add_argument("--closure", choices=("none", "flat", "solid"), default="none")
    cart.add_argument("--flat-speed-km-s", type=float, default=220.0)
    cart.add_argument("--solid-omega-km-s-kpc", type=float, default=27.0)
    cart.add_argument("--mc-samples", type=int, default=0)

    icrs = subparsers.add_parser("icrs-csv", help="Transform full ICRS CSV with Astropy")
    icrs.add_argument("input_csv", type=Path)
```

```
    icrs.add_argument("--output-csv", type=Path, default=Path("atlas_output.csv"))
    icrs.add_argument("--output-json", type=Path)
    icrs.add_argument("--frame-name", default="Astropy explicit Galactocentric frame")
    icrs.add_argument("--epoch", default="J2016.0")

    single = subparsers.add_parser("single", help="Calculate one Cartesian phase-space state")
    single.add_argument("--name", default="Custom object")
    single.add_argument("--epoch", default="J2016.0")
    single.add_argument("--frame-name", default="Declared Galactocentric frame")
    single.add_argument("--x-kpc", type=float, required=True)
    single.add_argument("--y-kpc", type=float, required=True)
    single.add_argument("--z-kpc", type=float, required=True)
    single.add_argument("--vx-km-s", type=float, required=True)
    single.add_argument("--vy-km-s", type=float, required=True)
    single.add_argument("--vz-km-s", type=float, required=True)
    return parser


def main() -> None:
    args = build_parser().parse_args()
    if args.command == "cartesian-csv":
        records = records_from_cartesian_csv(
            args.input_csv,
            closure=args.closure,
            flat_speed_km_s=args.flat_speed_km_s,
            solid_omega_km_s_kpc=args.solid_omega_km_s_kpc,
            frame_name=args.frame_name,
            default_epoch=args.epoch,
            mc_samples=args.mc_samples,
        )
        write_records_csv(records, args.output_csv)
        if args.output_json:
            write_records_json(records, args.output_json)
        print_summary(records)
        return

    if args.command == "icrs-csv":
        records = records_from_icrs_csv(
            args.input_csv,
            frame_name=args.frame_name,
            default_epoch=args.epoch,
        )
        write_records_csv(records, args.output_csv)
        if args.output_json:
            write_records_json(records, args.output_json)
        print_summary(records)
        return

    if args.command == "single":
        record = build_record(
            name=args.name,
            epoch=args.epoch,
            frame_name=args.frame_name,
            position_kpc=Vector3(args.x_kpc, args.y_kpc, args.z_kpc),
            velocity_km_s=Vector3(args.vx_km_s, args.vy_km_s, args.vz_km_s),
            quality_class="M6_MEASURED",
            closure_name="none",
        )
        print_summary([record])
        print(json.dumps(asdict(record), indent=2))
        return

    raise RuntimeError("unreachable command")


if __name__ == "__main__":
    main()
```

Author Declaration

The author declares that the central claim of this paper is the phase-geometric organisation of standard kinematic proper time together with a dimensionless chronometric label, an explicitly conventional Earth-language translation, and a frame- and provenance-declared Galactic phase-space atlas architecture. The gravity-independent qualification refers to the isolated kinematic sector of the calculation. The terrestrial value 207.386 ns is a calibration output of the canonical Earth ring and is not asserted to be a universal time quantum. The conditional rod–flywheel equations and any position-to-velocity Galactic closure are modelling assumptions to be tested separately and are not represented as measured universal dynamics.

S. M. H. Emamifar
Independent Research Collaboration on Black Hole and Cosmology Concepts (IRCBHC)
ORCID: 0009-0007-6257-0163

References

[1] A. Einstein, "Zur Elektrodynamik bewegter Koerper," *Annalen der Physik* **322**, 891–921 (1905), doi:10.1002/andp.19053221004.

[2] W. Rindler, *Relativity: Special, General, and Cosmological*, 2nd ed., Oxford University Press (2006).

[3] G. Petit and B. Luzum (eds.), *IERS Conventions (2010)*, IERS Technical Note No. 36, Verlag des Bundesamts fuer Kartographie und Geodaesie, Frankfurt am Main (2010), https://iers-conventions.obspm.fr/.

[4] P. Charlot et al., "The Third Realization of the International Celestial Reference Frame by Very Long Baseline Interferometry," *Astronomy & Astrophysics* **644**, A159 (2020), doi:10.1051/0004-6361/202038368.

[5] NASA Jet Propulsion Laboratory, Solar System Dynamics, "Planetary Physical Parameters," https://ssd.jpl.nasa.gov/planets/phys_par.html.

[6] NASA Jet Propulsion Laboratory, Solar System Dynamics, "Astrodynamic Parameters," https://ssd.jpl.nasa.gov/astro_par.html.

[7] NASA Jet Propulsion Laboratory, Solar System Dynamics, "Orbits and Ephemerides," https://ssd.jpl.nasa.gov/orbits.html.

[8] B. A. Archinal, C. H. Acton, M. F. A'Hearn, et al., "Report of the IAU Working Group on Cartographic Coordinates and Rotational Elements: 2015," *Celestial Mechanics and Dynamical Astronomy* **130**, 22 (2018), doi:10.1007/s10569-017-9805-5.

[9] National Geospatial-Intelligence Agency, *World Geodetic System 1984: Its Definition and Relationships with Local Geodetic Systems*, NGA.STND.0036_1.0.0 (2014).

[10] N. Ashby, "Relativity in the Global Positioning System," *Living Reviews in Relativity* **6**, 1 (2003), doi:10.12942/lrr-2003-1.

[11] G. Sagnac, "L'ether lumineux demontre par l'effet du vent relatif d'ether dans un interferometre en rotation uniforme," *Comptes Rendus de l'Academie des Sciences* **157**, 708–710 (1913).

[12] Gaia Collaboration, A. Vallenari, A. G. A. Brown, et al., "Gaia Data Release 3: Summary of the Content and Survey Properties," *Astronomy & Astrophysics* **674**, A1 (2023), doi:10.1051/0004-6361/202243940.

[13] D. Katz, P. Sartoretti, A. Guerrier, et al., "Gaia Data Release 3: Properties and Validation of the Radial Velocities," *Astronomy & Astrophysics* **674**, A5 (2023), doi:10.1051/0004-6361/202244220.

[14] European Space Agency, Gaia Archive, "Gaia DR3 Source Catalogue Data Model," https://gea.esac.esa.int/archive/documentation/GDR3/Gaia_archive/chap_datamodel/sec_dm_main_source_catalogue/ssec_dm_gaia_source.html.

[15] Astropy Collaboration, "Description of the Galactocentric Coordinate Frame," *Astropy Coordinates Documentation*, https://docs.astropy.org/en/stable/coordinates/galactocentric.html.

[16] Gaia Collaboration, R. Drimmel, M. Romero-Gomez, et al., "Gaia Data Release 3: Mapping the Asymmetric Disc of the Milky Way," *Astronomy & Astrophysics* **674**, A37 (2023), doi:10.1051/0004-6361/202243797.