Circular Phase Geometry Without Trigonometric Parametrisation: A Recursive Algebraic Construction and a Constant-Curvature Example of Recursive Survival Geometry

Peter M. Austin
Information Physics Institute

Companion note, revised 24 May 2026

Abstract

This note gives a self-contained algebraic construction of circular phase geometry from recursive updates. The construction begins with a simple failure case: if two coordinates are updated identically, the resulting path is diagonal rather than circular. Circular motion instead requires perpendicular coupling, where each coordinate changes according to the other coordinate with a sign reversal. A raw perpendicular update captures this geometry but produces Euler-type radial drift. A norm-preserving update is then obtained by normalising the perpendicular step, yielding an exact map of the circle to itself using only algebraic operations. The usual sine and cosine representation is not taken as primitive; it is recovered afterwards as the analytic description of repeated algebraic rotations. The numerical implementation in this note is strict: it declares the algebraic step coefficient directly and does not call sine, cosine, tangent, or inverse-trigonometric functions. The note also connects the construction to the reduced phase variables (Θ, Π), the action norm $J = \Theta^2 + \ell^2\Pi^2$, closure versus non-closure, and the null transport limit used in Recursive Survival Geometry.

Keywords: Circular phase geometry; Recursive Survival Geometry; Norm preservation; No-trig recursion; Phase support; Closure.

1 Purpose and Scope

The phase-space system used in Recursive Survival Geometry,

$$\frac{\mathrm{d}\Theta}{\mathrm{d}t} = \Pi, \qquad \frac{\mathrm{d}\Pi}{\mathrm{d}t} = -\Omega^2\Theta, \tag{1}$$

has the familiar form of a rotational or oscillator-like flow. It is often represented using sine and cosine functions. Such a representation is convenient, but it is not necessary as a primitive construction. The same circular phase geometry can be generated by recursive algebraic updates based on perpendicular coupling and norm preservation.

The aim is modest. This note does not claim to replace standard trigonometry, Hamiltonian mechanics, or geometric numerical integration. It isolates a structural point: circular phase geometry can be built from a two-component state, perpendicular cross-coupling between components, and exact preservation of a quadratic norm. The result is useful as a small technical foundation for the phase plane (Θ, Π). The circular structure used by the model need not be imported as sine and cosine at the start. Instead, those analytic functions appear later as names for components generated by repeated norm-preserving recursion.

2 Diagonal Failure

Circular motion does not arise when two coordinates are updated in the same way. If both coordinates receive identical increments,

$$x_{n+1} = x_n + \alpha\Delta t, \qquad y_{n+1} = y_n + \alpha\Delta t, \tag{2}$$

then the difference is preserved:

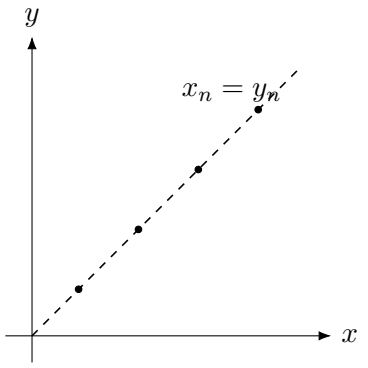
$$x_{n+1} - y_{n+1} = x_n - y_n. \tag{3}$$

In particular, if $x_0 = y_0$, then $x_n = y_n$ for every n. The resulting path lies on a diagonal line rather than on a circle.

Circular motion requires complementary coupling:

$$\Delta x \propto -y, \qquad \Delta y \propto x. \tag{4}$$

Each coordinate changes according to the other coordinate, with a perpendicular sign change. The state does not move by equal increments in the same direction; it is continually turned by the other component.



$$x_{n+1} = x_n + \alpha\Delta t, \qquad y_{n+1} = y_n + \alpha\Delta t$$

Figure 1: Identical coordinate updates produce diagonal motion rather than circular motion. Circular motion requires complementary perpendicular coupling.

3 Raw Perpendicular Recursion and Euler Drift

Let a point in the plane be written as

$$z_n = \begin{pmatrix} x_n \\ y_n \end{pmatrix}, \qquad (x_0, y_0) = (\rho, 0), \tag{5}$$

where $\rho > 0$ is the initial radius. The target support level is

$$J_0 = \rho^2. \tag{6}$$

This notation avoids confusing radius-squared with the statistical coefficient of determination.

For an algebraic recursive step coefficient $q \in \mathbb{R}$, the raw perpendicular update is

$$\tilde{x}_{n+1} = x_n - qy_n, \qquad \tilde{y}_{n+1} = y_n + qx_n. \tag{7}$$

In matrix form,

$$\tilde{z}_{n+1} = A_q z_n, \qquad A_q = \begin{pmatrix} 1 & -q \\ q & 1 \end{pmatrix}. \tag{8}$$

The update correctly encodes perpendicular coupling, but it does not preserve the radius exactly:

$$\tilde{x}_{n+1}^2 + \tilde{y}_{n+1}^2 = (x_n - qy_n)^2 + (y_n + qx_n)^2 \tag{9}$$

$$= (1+q^2)(x_n^2 + y_n^2). \tag{10}$$

After N raw steps, the radius is multiplied by $(1+q^2)^{N/2}$. Thus the raw perpendicular update spirals outward unless $q \to 0$ in the continuous limit or the radius is corrected.

This is the familiar failure of the forward Euler discretisation of a pure rotational vector field. The direction of the update is right, but the step lands slightly outside the circle. The next section corrects this without changing the perpendicular structure.

4 Norm-Preserving Algebraic Recursion

A radius-preserving algebraic update is obtained by normalising the perpendicular step. Declare $q \in \mathbb{R}$ as the finite recursive step coefficient and set

$$x_{n+1} = \frac{x_n - qy_n}{\sqrt{1+q^2}}, \qquad y_{n+1} = \frac{y_n + qx_n}{\sqrt{1+q^2}}. \tag{11}$$

This update uses only algebraic operations once q is specified. It does not introduce sine, cosine, tangent, or inverse-trigonometric functions as primitive definitions of the coordinates.

Theorem 1 (Exact radius preservation)**.** *Let*

$$M_q = \frac{1}{\sqrt{1+q^2}} \begin{pmatrix} 1 & -q \\ q & 1 \end{pmatrix}. \tag{12}$$

Then $M_q^T M_q = I$ *and* $\det M_q = 1$*. Consequently, if* $z_{n+1} = M_q z_n$*, then*

$$x_{n+1}^2 + y_{n+1}^2 = x_n^2 + y_n^2. \tag{13}$$

In particular, if $x_0^2 + y_0^2 = J_0$*, then* $x_n^2 + y_n^2 = J_0$ *for every* n*.*

Proof. Compute

$$M_q^T M_q = \frac{1}{1+q^2} \begin{pmatrix} 1 & q \\ -q & 1 \end{pmatrix} \begin{pmatrix} 1 & -q \\ q & 1 \end{pmatrix} \tag{14}$$

$$= \frac{1}{1+q^2} \begin{pmatrix} 1+q^2 & 0 \\ 0 & 1+q^2 \end{pmatrix} = I. \tag{15}$$

Also

$$\det M_q = \frac{1+q^2}{1+q^2} = 1. \tag{16}$$

Therefore M_q preserves the Euclidean norm and preserves orientation. Equivalently,

$$x_{n+1}^2 + y_{n+1}^2 = \frac{(x_n - qy_n)^2 + (y_n + qx_n)^2}{1+q^2} \tag{17}$$

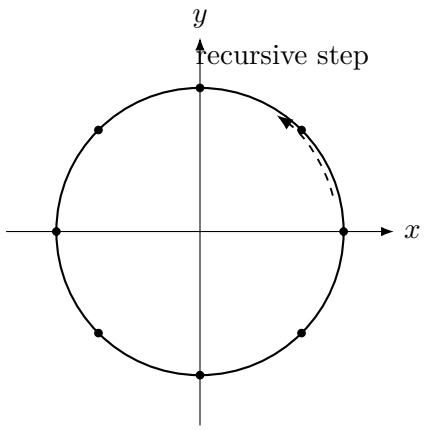
$$= x_n^2 + y_n^2. \tag{18}$$

□

The same construction may be written as the matrix recurrence

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \frac{1}{\sqrt{1+q^2}} \begin{pmatrix} 1 & -q \\ q & 1 \end{pmatrix} \begin{pmatrix} x_n \\ y_n \end{pmatrix}. \tag{19}$$

Thus the circular trajectory is generated by repeated algebraic recursion.



$$x_{n+1} = \frac{x_n - q y_n}{\sqrt{1+q^2}}, \qquad y_{n+1} = \frac{y_n + q x_n}{\sqrt{1+q^2}}$$

Figure 2: Circular phase geometry generated by norm-preserving algebraic recursion. The construction uses perpendicular coupling and normalisation rather than trigonometric parametrisation.

5 Analytic Angle Recovery Is Not Part of the Construction

The recurrence above does not require an angle variable. It requires only the state $z_n = (x_n, y_n)^T$, the algebraic coefficient q, and the norm-preserving update M_q.

After the recurrence has been specified, one may compare M_q with the standard analytic rotation notation. Since M_q is an orientation-preserving orthogonal map of the plane, standard Euclidean geometry allows it to be represented by a rotation angle ϕ_q. In ordinary angular language, that angle satisfies

$$\tan \phi_q = q, \qquad \phi_q = \arctan q. \tag{20}$$

This comparison is diagnostic only. It is not used by the strict construction, and it is not used by the code in Section 9. The phrase "without trigonometric parametrisation" means that the coordinates are generated by the algebraic recurrence first. The standard circular functions are names one may attach later to the resulting iterates.

Starting from $(C_0, S_0) = (1, 0)$, define

$$C_{n+1} = \frac{C_n - q S_n}{\sqrt{1+q^2}}, \qquad S_{n+1} = \frac{S_n + q C_n}{\sqrt{1+q^2}}. \tag{21}$$

By the theorem,

$$C_n^2 + S_n^2 = 1 \tag{22}$$

for all n. If an ordinary angle variable is introduced after the recursion, then these sequences may be recognised as the usual circular components. The recursion itself only needs (21).

Remark 1 (Strict no-trig code path)**.** *The strict numerical implementation declares q directly. It does not set $q = \tan(\omega h)$, and it does not call sine, cosine, tangent, or inverse-trigonometric functions. If a later analytic comparison chooses to match a pre-named angle exactly, that is a separate calibration convention, not the algebraic construction used here.*

6 Connection to the (Θ, Π) Phase Plane

The construction clarifies the role of the phase equations in the main framework. The circular or rotational structure of the (Θ, Π) plane does not require sine and cosine to be fundamental. They may be recovered as a convenient analytic representation, but the recursion itself is sufficient.

Suppose $\Omega > 0$ is locally constant. The RSG action norm is

$$J(\Theta, \Pi) = \Theta^2 + \ell^2 \Pi^2. \tag{23}$$

In the constant-Ω circular cell, the norm-preserving calibration is

$$\ell = \frac{1}{\Omega}. \tag{24}$$

Define calibrated coordinates

$$x = \Theta, \qquad y = \ell \Pi = \frac{\Pi}{\Omega}. \tag{25}$$

Then

$$J(\Theta, \Pi) = x^2 + y^2. \tag{26}$$

The constant-support class

$$J(\Theta, \Pi) = J_0 \tag{27}$$

is therefore the circle

$$x^2 + y^2 = J_0 \tag{28}$$

in calibrated coordinates.

Equation (1) becomes

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \Omega^2 y, \qquad \frac{\mathrm{d}y}{\mathrm{d}t} = -x, \tag{29}$$

under the choice $y = \Pi/\Omega$. If instead one rescales the time parameter locally, or uses the energy-normalised variables $(X, Y) = (\Omega\Theta, \Pi)$, the same perpendicular coupling appears in symmetric oscillator form. The essential point is unchanged: in calibrated coordinates, the lossless phase cell is a norm-preserving two-component recursion.

Multiplying the constant-support relation by Ω^2 gives

$$\Pi^2 + \Omega^2 \Theta^2 = \Omega^2 J_0. \tag{30}$$

This is the same quadratic form used in the lossless oscillator energy diagnostic

$$E = \frac{1}{2}\left(\Pi^2 + \Omega^2 \Theta^2\right). \tag{31}$$

Thus the recursive circle is not an additional physical assumption. It is an algebraic construction of the phase geometry already encoded in the lossless part of the reduced dynamics.

7 Closure, Non-Closure, and Recursive Transport

Recursive Survival Geometry distinguishes closing from non-closing transport. In a two-dimensional circular phase example, norm preservation keeps the state on a constant-support circle, while closure depends on whether the finite recursive step eventually returns to the initial state.

If the optional analytic angle ϕ_q is introduced after the recurrence, one may define the rotation number

$$\rho_q = \frac{\phi_q}{2\pi}. \tag{32}$$

If $\rho_q \in \mathbb{Q}$, the orbit closes after finitely many steps. If $\rho_q \notin \mathbb{Q}$, the orbit does not close and its points are dense on the circle. This angle-based description is a convenient after-the-fact classification. The recurrence itself remains algebraic.

In the null transport limit of Recursive Survival Geometry, one imposes three independent conditions:

$$\Gamma \to 0, \qquad \|\alpha_{n+1}\| = \|\alpha_n\|, \qquad \text{no finite recurrence into a rest-frame-admitting cycle.} \tag{33}$$

The first condition removes dissipative survival loss. The second gives norm-preserving transport. The third separates non-closing transport from recurrent matter-like cycling. The algebraic circle map supplies a constant-curvature model for the second condition and a simple arena for discussing the third.

8 Exposure and Survival Weighting

The circular recursion above describes lossless phase geometry. Recursive Survival Geometry then adds a separate selection layer. For phase state (Θ, Π), define the action norm

$$J = \Theta^2 + \ell^2 \Pi^2, \tag{34}$$

and the exposure factor

$$W = \frac{\Theta^2}{J} = \frac{\Theta^2}{\Theta^2 + \ell^2 \Pi^2}, \qquad 0 \le W \le 1. \tag{35}$$

A non-negative dissipation coefficient Γ defines the survival law

$$\frac{\mathrm{d}S}{\mathrm{d}t} = -\Gamma W S. \tag{36}$$

For a discrete recursive history $\gamma_i = \{\alpha_{i,0}, \alpha_{i,1}, \ldots\}$, the corresponding discrete survival weight is

$$S_i = \exp\left(-\sum_k \Gamma(\alpha_{i,k}) W(\alpha_{i,k}) \Delta t \right). \tag{37}$$

Normalising over histories gives

$$p_i = \frac{S_i}{\sum_j S_j}. \tag{38}$$

The construction therefore separates two roles. The algebraic phase map generates possible circular or rotational transport. The survival functional filters generated histories when $\Gamma W \neq 0$. In the lossless circular case $\Gamma \to 0$, the survival layer does not select between histories by dissipation. In the non-null case $\Gamma > 0$, histories with lower accumulated ΓW remain more strongly represented after normalisation.

9 Numerical Demonstration

The following Python-style snippet implements the norm-preserving recursive circle update. It deliberately updates the state from perpendicular coupling and normalisation rather than from direct sine and cosine parametrisation. The code path is strict: the step coefficient q is declared directly and no trigonometric function is called.

Listing 1: Iterative circular trajectory without direct trigonometric parametrisation

```
import numpy as np
import matplotlib.pyplot as plt

# Parameters
rho = 1.0 # Initial radius
J0 = rho * rho # Target support level
q = 0.00625 # Algebraic recursive step coefficient
steps = 1000 # Number of iterations

scale = np.sqrt(1.0 + q*q)

# Initial conditions
x_vals = [rho]
y_vals = [0.0]

for _ in range(steps):
    x = x_vals[-1]
    y = y_vals[-1]

    x_new = (x - q*y) / scale
    y_new = (y + q*x) / scale

    x_vals.append(x_new)
    y_vals.append(y_new)

support_error = [
    abs(x*x + y*y - J0)
    for x, y in zip(x_vals, y_vals)
]

print("maximum support error:", max(support_error))

plt.figure(figsize=(6, 6))
plt.plot(x_vals, y_vals, label="algebraic recursive path")
plt.axis("equal")
plt.xlabel("x")
plt.ylabel("y")
plt.title("Circular trajectory from norm-preserving recursion")
plt.legend()
plt.grid(True)
plt.show()
```

The radius-preserving step can also be written with explicit normalisation of the raw Euler move:

Listing 2: Equivalent normalise-after-step implementation

```
x_raw = x - q*y
y_raw = y + q*x
norm = np.sqrt(x_raw*x_raw + y_raw*y_raw)
x_new = rho * x_raw / norm
y_new = rho * y_raw / norm
```

When $\rho = 1$, this is equivalent to (11). For general ρ, the same normalisation projects the raw perpendicular step back to the circle $x^2 + y^2 = J_0$.

10 What the Construction Does and Does Not Claim

The construction establishes a limited but useful fact. It shows that a circular phase plane can be generated from perpendicular recursion and norm preservation without taking trigonometric parametrisation as primitive.

It does not claim that sine and cosine are unnecessary in analysis. Once an angle coordinate is introduced, the standard functions are the most compact way to describe the iterates. It also does not claim that all oscillator discretisations are equivalent. The raw Euler update, the normalised perpendicular update, exact analytic flow, and other structure-preserving schemes differ at finite step size.

The specific value of the construction is conceptual and structural: it gives a compact algebraic origin for circular phase geometry, and it connects that geometry to the norm-preserving sector of Recursive Survival Geometry. This allows the phase plane (Θ, Π) to be introduced recursively first and analytically second.

A Minimal Recurrence Summary

The entire construction can be compressed to the following recurrence. Given $q \in \mathbb{R}$ and $J_0 > 0$, set

$$z_0 = \begin{pmatrix} \sqrt{J_0} \\ 0 \end{pmatrix}, \qquad z_{n+1} = M_q z_n, \qquad M_q = \frac{1}{\sqrt{1+q^2}} \begin{pmatrix} 1 & -q \\ q & 1 \end{pmatrix}. \tag{39}$$

Then

$$\|z_n\|^2 = J_0 \qquad \text{for every } n. \tag{40}$$

If an angle coordinate is introduced after the recurrence, then M_q may be represented analytically as a rotation. The recurrence itself does not require that representation.

B Code-Path Audit

The strict implementation in Section 9 uses the following operation classes only:

addition, subtraction, multiplication, division, square root, absolute value, and plotting. (41)

It does not call sine, cosine, tangent, inverse sine, inverse cosine, or inverse tangent. The optional analytic comparison in Section 5 should therefore not be confused with the algorithmic construction.

References

[1] V. I. Arnold, *Mathematical Methods of Classical Mechanics*, 2nd ed., Springer, 1989.

[2] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*, 2nd ed., Springer, 2006.

[3] R. de la Llave, A. González, À. Jorba, and J. Villanueva, "KAM theory without action-angle variables," *Nonlinearity* **18** (2005), 855–895.

[4] P. M. Austin, *Recursive Survival Geometry v1.4*, revised manuscript, Information Physics Institute, 2026.