A Singularity-Aware Symbolic Algebra Framework for Physical Equations

Draft v0.2 — WheelPhysics Project Last updated: 2026-05-20 Status: DRAFT — wszystkie sekcje uzupełnione

---

Abstract

Singular points in theoretical physics — black hole singularities, the Big Bang at t=0, infrared and ultraviolet divergences in quantum field theory — are routinely treated as computational breakdowns rather than mathematical objects. We present a symbolic algebra framework based on Wheel Algebra (Carlström 2004) in which division by zero produces a well-defined absorbing element $\perp$ ("bottom") instead of an error. Applied to 46 equations across six physical domains (general relativity, cosmology, quantum field theory, classical mechanics, thermodynamics, and pure mathematics), the framework correctly identifies singular points, classifies them into four structural categories, and — for pole singularities — computes residues via the Cauchy theorem. We demonstrate a formal isomorphism between classical resonance and the QFT on-shell condition, and show that the Kretschmann scalar allows the framework to distinguish coordinate artefacts from genuine physical singularities. The complete implementation is available as verifiable Python code.

---

1. Introduction

1.1 The problem: singularities as computational failures

Physical theories break down at singular points in two distinct ways. Some singularities are artefacts of the coordinate system — the Schwarzschild horizon at $r = r_s$ is a well-known example, regular in Kruskal coordinates but singular in Schwarzschild coordinates. Others are genuine physical singularities: the curvature invariant $K = 12r_s^2/r^6$ diverges at $r = 0$ regardless of coordinate choice.

Classical analysis handles this distinction by external argument (change of coordinates, check invariants). The algebra itself does not encode it.

A second class of problems arises in quantum field theory. The Feynman prescription ($i\varepsilon$ in the propagator denominator) is an explicit workaround: the physical pole $1/(p^2 - m^2)$ is displaced off the real axis to avoid division by zero during momentum integration. The prescription works, but its justification is procedural rather than algebraic.

In both cases, division by zero appears as an obstacle to be circumvented. We ask a different question: what if division by zero were admitted as a first-class algebraic operation?

1.2 Wheel Algebra

Wheel Algebra, introduced by Carlström (2004), extends a commutative ring with two operations: inversion /x (not reciprocal 1/x) and an absorbing element $\perp$ ("bottom") that captures all undefined forms. The axioms include:

- $/0 = \perp$
- $\perp + x = \perp$ for all x (absorption under addition)
- $\perp \cdot x = \perp$ for all x (absorption under multiplication)
- $0/0 = \perp$, $1/0 = \perp$, $\infty/\infty = \perp$ — all indeterminate forms map to the same element

Crucially, $\perp$ is not an error — it is an element of the algebra. Expressions containing $\perp$ remain algebraically manipulable.

Carlström's algebra differs from IEEE 754 floating-point NaN in two essential respects. NaN is not an algebraic element — it propagates through arithmetic by exception rules, not by algebraic absorption. In a Wheel, $\perp$ satisfies full ring-compatible axioms. Furthermore, NaN does not satisfy $\perp \cdot 0 = \perp$; IEEE 754 mandates $0 \cdot \text{NaN} = \text{NaN}$ but the result is not algebraically derived — it is a convention imposed externally.

Meyenburg (2025) extends Wheel to a semi-ring with two directed infinities $+\infty$ and $-\infty$ as first-class elements, enabling $1/0^+ \neq 1/0^-$. This is mathematically richer but collapses the clean interface between algebra and analysis. In Wheel-Physics, directional pole structure is recovered by the wheel_calculus layer via Laurent expansion — without modifying the base algebra. Section 2.4 details this choice.

We deliberately use Carlström's formulation rather than Meyenburg's (2025) semi-ring extension, which introduces distinct $+\infty$ and $-\infty$ elements. The directional structure of poles (which way does the function blow up?) is recovered by the analytic layer (wheel_calculus) rather than built into the base algebra. This separation is an intentional architectural decision; we discuss it in Section 2.

1.3 Scope and contributions

This paper makes the following contributions:

1. A working implementation of Wheel Algebra as a SymPy extension, with 11/11 ring axioms passing automated tests.
2. An analytic extension layer (wheel_calculus) that stratifies singularities into four categories: regular points, removable singularities, poles (with residue and Laurent hint), and irreducible $\perp$.

3. A formal singularity type system (SingularityType enum, 12 types) distinguishing coordinate artefacts, physical singularities, logarithmic divergences, and complex poles.
4. Application to 46 equations across GR, cosmology, QFT, classical mechanics, thermodynamics, and pure mathematics — with 60 known singular points catalogued.
5. A formal isomorphism between classical resonance and QFT on-shell condition, documented via residue analysis.

---

2. Architecture

2.1 Design principles

The framework is organised around a clean separation:

```
wheel_number.py      — WheelNumber type: values, ⊥, arithmetic
wheel_algebra.py     — Wheel Algebra rules, evaluate_at, rewriting
sympy_extension.py   — SymPy integration: wheel_subs, singularity_map
wheel_calculus.py    — Analytic extension: classification, residues, Laurent
```

Wheel Algebra ≠ limit theory. This separation is fundamental. wheel_algebra.py implements the axiomatic algebra: at a singular point it returns ⊥, unconditionally. wheel_calculus.py is a separate diagnostic layer: when Wheel returns ⊥, it asks why and what kind.

2.2 The four-way split (wheel_calculus)

Wheel result at point x_0:

```
Finite(v)          → regular point
⊥ ──┬──→ 0/0 form ──→ Taylor expansion ──→  RemovableSingularity(lim=v)
    │                                └──→ WheelBottom (no structure)
    └──→ true pole ──→ residue analysis ──→  PoleSingularity(order, res, Laurent)
```

For a true pole (numerator ≠ 0, denominator → 0), the analytic layer computes: - pole order n: via lim $(x-x_0)^n \cdot f(x)$ for n = 1, 2, … - residue (order-1 poles): res = lim $(x-x_0) \cdot f(x)$ — Cauchy's theorem - Laurent hint: human-readable leading term of the Laurent expansion

2.3 The key technical fix: _has_division_by_zero_at()

SymPy's subs() can silently simplify expressions before substitution, losing information about division by zero. For example, 1/(1 − r_s/r) with r_s → r is first simplified by SymPy to 1/(1 − 1) = 1/0, but if the simplification proceeds differently the zero in the denominator may be masked.

We solve this by recursively inspecting all denominators in the AST before any substitution. If any denominator evaluates to zero at the target point, wheel_algebra.py returns ⊥ directly, bypassing SymPy's substitution.

2.4 Carlström vs Meyenburg: architectural rationale

The fundamental question is: should directional information (does f blow up to $+\infty$ or $-\infty$?) be encoded in the algebra or in the analysis layer?

Carlström's choice — a single ⊥ for all indeterminate forms — yields a uniform interface. Every singular point is ⊥. The wheel_calculus layer can then apply different diagnostic tools depending on what it finds: Taylor expansion for 0/0 forms, Cauchy residue for true poles, Poincaré asymptotic series for logarithmic poles. The algebra does not need to know which tool will be used.

Meyenburg's extension embeds the directional answer in the algebra itself. This is consistent and mathematically interesting, but it couples the base algebra to a particular analytic technique (one-sided limits). A framework for physical equations benefits from the separation of concerns: the algebra reports what happened (undefined), the analysis reports why and what kind.

In practice: g_rr at r = r_s gives 1/0 with a definite sign — but the sign is coordinate-dependent. Encoding it in the algebra would give a false precision. The Kretschmann invariant approach (Section 3.2) provides coordinate-independent information at a higher level — the correct place for physically meaningful distinctions.

---

3. Results

3.1 Summary statistics

Domain	Equations	Singular points	⊥	Finite	Mixed
General Relativity	12	~18	✓	—	4
Cosmology	5	5	✓	—	—
QFT	13	~20	✓	3	2
Classical	4	4	✓	—	—
Thermodynamics	6	7	✓	1	—
Mathematics	6	6	✓	—	—
Total	46	60	38	4	4

3.2 Kretschmann invariant: distinguishing singularity types

The metric component g_rr of the Schwarzschild solution diverges at both r = r_s (the horizon) and r = 0 (the geometric centre): both return ⊥. This is correct but insufficient — g_rr depends on the choice of coordinates.

The Kretschmann scalar $K = 12r_s^2/r^6$ is a coordinate-independent curvature invariant. Evaluating both:

$K(r = r_s) = 12 / r_s^4$ (finite — the horizon is a coordinate artefact)
$K(r = 0) = \perp$ (genuine physical singularity)

This is our strongest result in GR: Wheel Algebra, applied to a tensorial invariant, correctly distinguishes the two types of singularity without any external argument. The distinction is algebraic, not geometric.

3.3 Feynman propagators: on-shell = ⊥

Every Feynman propagator in the database becomes ⊥ exactly at the on-shell condition:

Propagator	Singular point	Wheel result
$1/(p^2-m^2)$	$p = \pm m$ (on-shell)	⊥
$1/k^2$	$k = 0$ (massless on-shell)	⊥
$(p+m)/(p^2-m^2)$	$p = m$ (electron), $p = -m$ (positron)	⊥
$1/p$	$p = 0$ (massless fermion)	⊥
$1/(s-m^2)$	$s = m^2$ (Compton)	⊥

Off-shell (virtual particles): all propagators return finite, computable values.

Hypothesis: ⊥ on-shell is an algebraic definition of observability in QFT. Real (observable) particles correspond to algebraically undefined points; virtual particles are algebraically well-defined. The $i\varepsilon$ prescription is a classical workaround for the absence of Wheel Algebra in standard analysis.

This framing is consistent with the LSZ reduction formalism: the reduction formula extracts S-matrix elements precisely at the on-shell poles of the full propagator. In the Wheel framework, these poles are the points where the algebra returns ⊥.

3.4 Residue analysis: the resonance ↔ on-shell isomorphism

Residue analysis for pole singularities yields:

Expression	Point	Order	Residue	Physical meaning
$1/(p^2-m^2)$	$p = m$	1	$1/(2m)$	On-shell amplitude (QFT)
$1/(\omega^2-\omega_0^2)$	$\omega = \omega_0$	1	$1/(2\omega_0)$	Classical resonance
g_{rr} at horizon	$r = r_s$	1	r_s	Coordinate pole (GR)
$1/k^2$	$k = 0$	2	N/A	IR divergence

The key observation:

$$\mathrm{res}(1/(p^2-m^2),\ p = m) = 1/(2m)$$
$$\mathrm{res}(1/(\omega^2-\omega_0^2),\ \omega = \omega_0) = 1/(2\omega_0)$$

Both expressions have identical algebraic structure. A classical harmonic oscillator at resonance and a relativistic particle propagating on-shell are described by the same residue formula. This isomorphism — between classical resonance and quantum mechanical on-shell condition — is here documented formally for the first time via residue analysis.

3.5 Removable singularities: Wheel ≠ limit theory

Wheel Algebra returns $\perp$ at every 0/0 form, including removable singularities. This is not a deficiency — it is the correct algebraic behaviour. The analytic layer recovers the limit when it exists:

Expression	Point	Wheel	Limit
$\sin(x)/x$	$x = 0$	$\perp$	1
$\sin^2(x)/x^2$	$x = 0$	$\perp$	1
$(1-\cos x)/x^2$	$x = 0$	$\perp$	1/2
$x/(e^x-1)$	$x = 0$	$\perp$	1

The sinc function, used extensively in signal processing, is a removable singularity. Wheel + calculus handles it correctly: the algebra flags the undefined form; the analytic layer recovers the value.

3.6 Friedmann equation: continuity through a = 0

The Friedmann equation $H^2 = (\dot{a}/a)^2 = 8\pi G\rho/3 - kc^2/a^2$ contains a singularity at $a = 0$. Wheel returns $\perp$ at this point for both the kinetic term $\dot{a}^2/a^2$ (when $a \to 0$) and the curvature term k/a^2.

A less obvious observation: the equation is well-defined for $a < 0$ in the Wheel framework — $\perp$ occurs at $a = 0$, but negative values of the scale factor are not algebraically forbidden. This is consistent with pre-big-bang cosmological models (e.g. Gasperini-Veneziano) in which the universe contracts through a bounce. The Wheel framework does not mandate a physical interpretation; it records that $a = 0$ is the singular point and that the algebra is symmetric under $a \to -a$.

This symmetry is not imposed — it emerges from the algebraic structure. The Friedmann equation treats the scale factor as a variable; Wheel treats $a = 0$ as an absorbing point and makes no distinction between the contracting and expanding branches.

3.7 Thermodynamic singularities

The Boltzmann factor e^{−E/kT} is well-defined for all $T > 0$. At $T = 0$, the exponent diverges: $E/kT \to \infty$, and the factor $\to 0$. The Wheel framework captures this as $\perp$ at the division by zero in $1/T$, consistent with the third law of thermodynamics, which states that $T = 0$ is unattainable. The algebraic signal and the physical law coincide.

The Bekenstein-Hawking black hole entropy S = A/4G_N propagates the $\perp$ from the area singularity through inversion: if $K(r{=}0) = \perp$, then $S \to \perp$ by the absorption rule $\perp \cdot x = \perp$. This is not a new physical result, but it demonstrates that $\perp$ propagates consistently through thermodynamic relations — the framework does not lose track of singularity information across derived quantities.

3.8 Open cases

Logarithmic poles (QCD): The gluon propagator with one-loop correction contains a logarithmic divergence (Landau pole). Wheel correctly returns $\perp$; Taylor expansion fails for logarithms. The appropriate analytic tool is an asymptotic (Poincaré) series. This is classified as LOGARITHMIC in SingularityType and remains open.

Complex poles: The damped harmonic oscillator Green's function has poles at $\omega = \pm\sqrt{(\omega_0^2 - \gamma^2/4)} \pm i\gamma/2$, which lie off the real axis for $\gamma > 0$. Wheel, operating over $\mathbb{R}$, does not encounter them at real substitution points. Classified as COMPLEX_POLE; extension to $\mathbb{C}$ is open.

---

4. Discussion

4.1 What does $\perp$ mean physically?

The element $\perp$ arises in three distinct physical contexts in our analysis, and each suggests a different interpretation.

At $K(r{=}0)$: $\perp$ marks a genuine curvature singularity — a point where the spacetime manifold structure breaks down. Here $\perp$ seems to say: "this point is not part of the physical domain."

At $p = 0$ for a massless fermion: $\perp$ is algebraically correct and physically meaningful — a massless particle cannot be at rest. The algebra enforces a physical constraint.

At $T = 0$: $\perp$ reflects the third law — the point exists geometrically but is dynamically unreachable.

We do not unify these interpretations. $\perp$ is not a single physical concept — it is an algebraic signal that the expression is undefined at this point. What that means physically depends on the equation. This is, we argue, the correct level of abstraction: the algebra should report the signal; the physical interpretation belongs to the theory.

4.2 Relationship to regularisation

Dimensional regularisation (DR) and ζ-regularisation assign finite values to expressions that classically diverge — the same expressions that Wheel maps to $\perp$.

$\Gamma(\varepsilon) = \perp$ (Gamma function at non-positive integers)
$\zeta(1) = \perp$ (Riemann zeta at $s = 1$)

The formal correspondence suggests that Wheel Algebra and classical regularisation techniques share a common diagnostic signal: both mark the same expressions as undefined in the same places. However, regularisation does not stop at $\perp$ — it assigns a finite value via analytic continuation or renormalisation group methods.

Whether the wheel_calculus layer could, in principle, replicate this assignment (via Laurent coefficients at the pole) is an open question. The POLE_SIMPLE case already computes residues, which correspond to the leading Laurent coefficient — the same object that dimensional regularisation extracts as the divergent part. We note this formal correspondence; the claim that renormalisation becomes unnecessary under Wheel Algebra would require a separate, detailed argument and is beyond the scope of this paper.

4.3 Towards Wheel on Riemannian manifolds

The current framework operates on scalar expressions over $\mathbb{R}$. Extension to tensor fields on Riemannian manifolds would require a Wheel-valued tensor calculus — a structure in which metric components, Christoffel symbols, and curvature tensors can take the value $\perp$ without breaking covariance. The Kretschmann result (Section 3.2) already points in this direction: a tensorial invariant evaluated in Wheel gives physically correct and coordinate-independent information.

A natural conjecture is that Wheel-valued differential geometry would allow the metric to be extended through singular points rather than truncated at them. We consider this a natural next step toward applications in quantum gravity, but leave it for future work.

4.4 Limitations

- Wheel operates pointwise. UV divergences in QFT live in integrals $\int d^4k$, not at individual points. Wheel cannot directly address UV renormalisation.

- Coordinate-dependent objects require invariants. Christoffel symbols, metric components in fixed coordinates — their $\perp$ may be an artefact. The framework requires pairing with tensorial invariants (Kretschmann, Ricci scalar) for physical interpretation.
- Complex poles. Wheel over $\mathbb{R}$ misses poles with non-zero imaginary part.
- Wheel ≠ limit theory — this is a feature, not a bug, but it means the framework returns $\perp$ for removable singularities. The analytic layer (wheel_calculus) is required to recover limits.

---

5. Related Work

Carlström (2004) provides the foundational axiomatics. The paper demonstrates that any commutative ring can be embedded in a Wheel, and that the resulting structure is consistent. Our implementation follows his axioms directly, with WheelNumber as the concrete type and wheel_algebra.py implementing the rewriting rules.

Meyenburg (2025) extends Wheel to a semi-ring with directed infinities ($+\infty$ and $-\infty$ as first-class elements). As discussed in Section 2.4, we deliberately use Carlström's formulation, recovering directional information at the analytic layer rather than the algebraic one.

Bergstra (2019) provides a systematic survey of options for handling division by zero, covering approaches from partial algebras and meadows to wheel theory and transreal arithmetic. The survey contextualises Wheel Algebra as one of several consistent extensions of standard arithmetic — the one with the strongest purely equational algebraic foundation. Bergstra & Ponse (2015) develop "common meadows" as an alternative total algebra with a distinct absorbing error element; like $\perp$ in Wheel, the error propagates through all operations.

IEEE 754 NaN is the engineering standard for undefined floating-point results. Unlike $\perp$, NaN is not algebraically absorbing in a ring-theoretic sense: $0 \cdot \text{NaN} = \text{NaN}$ is a convention imposed by the standard, not a theorem derived from axioms. NaN does not form an algebraic structure; $\perp$ does.

- Carlström, J. (2004). Wheels — On Division by Zero. Journal of Logic and Algebraic Programming, 1–24.
- Meyenburg, T. (2025). Meyenburg Algebra and the Mass Gap. International Journal of Mathematics Trends and Technology, 71(10), 29–35. DOI: 10.14445/22315373/IJMTT-V71I10P105
- Bergstra, J.A. & Ponse, A. (2015). Division by Zero in Common Meadows. In: Software, Services, and Systems. Lecture Notes in Computer Science, vol. 8950. Springer. DOI: 10.1007/978-3-319-15545-6_6

- Bergstra, J.A. (2019). Division by Zero: a Survey of Options. Transmathematica. DOI: 10.36285/tm.v0i0.17

---

6. Conclusion

We have presented a symbolic algebra framework in which division by zero is a well-defined algebraic operation. Applied to canonical equations of theoretical physics, the framework identifies and classifies singularities consistently with known physical results. Several non-trivial observations emerge: the algebraic isomorphism between resonance and on-shell conditions, the role of curvature invariants in distinguishing physical from coordinate singularities, and a formal connection between the ⊥ element and standard regularisation procedures.

The framework is implemented in Python (SymPy), tested, and verifiable by running python main.py.

The broader significance is methodological. Singular points in physical theories are not computational failures to be regularised away — they carry information. Wheel Algebra provides a language in which that information is preserved algebraically, classified structurally, and made available to higher-level analysis. Whether this language can be extended to cover integral divergences, complex poles, and tensor fields on curved manifolds remains open. The present framework establishes that the approach is viable at the level of symbolic expressions.

Appendix A: Singularity Type System

SingularityType enum (wheel_calculus.py)

```
REGULAR       → ✓   regular point
REMOVABLE     → ⊥→v  removable (Taylor recovers limit)
POLE_SIMPLE   → ⊥   simple pole (order 1), residue defined
POLE_HIGHER   → ⊥   higher-order pole, residue N/A
POLE          → ⊥   generic pole (order unknown)
ESSENTIAL     → ⊥   essential singularity (Picard)
LOGARITHMIC   → ⊥   logarithmic divergence (QCD)
BRANCH_POINT  → ⊥   branch point
COORDINATE    → ⊥*  coordinate artefact (needs invariant)
PHYSICAL      → ⊥   confirmed physical singularity
COMPLEX_POLE  → ⊥?  complex pole (off real axis)
UNKNOWN       → ?   fallback
```

Properties: - .is_genuine_singularity — not removable - .has_residue — Cauchy residue defined (POLE_SIMPLE only) - .short — label for tables and logs

---

Appendix B: Equation Database Summary

Full catalogue available by running python main.py --db. Summary: 46 equations across 6 domains, 60 singular points, 38 returning ⊥, 4 finite, 4 mixed. All results reproducible without modification.

Appendix C: Code Availability

All results in this paper are reproducible by running:

```
git clone https://github.com/Mariusz-Rossa/WheelPhysics.git
cd WheelPhysics
pip install sympy numpy
python main.py                    # full analysis
python main.py --quick            # project summary
python main.py --module gr        # general relativity only
python main.py --module qft       # quantum field theory only
python main.py --module calculus  # singularity classification + JSON output
python main.py --module viz       # ASCII plots with ⊥ markers
python main.py --regen-log        # recompute wheel_results.json
python main.py --calculus-log     # show calculus_results.json
python main.py --db               # equation catalogue
```

DOI: https://doi.org/10.5281/zenodo.20305458

— End of draft v0.2 —