A Restless World: When Chaos Looks Calm

Nov 26, 2026 · @Tom

Start here

Three equal masses pulling on each other do not have to be chaotic: special orderly solutions exist, such as the famous figure-eight orbit. The arrangement studied here is chaotic, yet in the Restless World lab it often looks calm for minutes at a time. This case study explains why, and shows how long a computer can follow the exact motion before its own small numerical errors take over: in the run studied here, about 290 time units, roughly two and a half minutes at the lab's default speed.

- **The laboratory:** A Restless World, with its introduction page and how the labs are built.
- **The lab's source:** the lab is a single web page with its JavaScript gravity engine written into it, so View Page Source shows the code that runs. This study examined version 27, which was also run outside the browser during the independent review; a version-fixed copy of its source, checked against the original release, is on the research files page.
- **Every computed result:** the workbook A Restless World — exact-start analysis, and the code and raw output on Restless World research files, described at the end of this case study.

Part 1 is for students and needs no equations. Part 2 is for teachers and advanced readers. In Part 2, an **observation** is something we computed or measured; an **interpretation** is our reading of it.

This work is a human–AI collaboration with additional independent investigation and review by other AI. There are no quizzes and no tests. The invitation is simply to watch something beautiful, get curious, and investigate.

Part 1 — For students

The mystery: chaos that looks calm

Set the lab's C starting distance to 3.5 and the binary separation to 1.8, then press Play. Within seconds the three bodies tangle, swapping partners and flinging each other around. Then, surprisingly often, they settle into something that looks like a tidy little solar system.

Watch for three things:

- **A pair.** Two bodies circle each other in small, fast loops, like a double star.
- **An outsider.** The third body traces a big, slow arc around the pair, sometimes far off the screen.
- **A balance point.** The white crosshair marks the centre of mass of all three, and it never moves. The gold diamond marks the point midway between A and B. Because the three masses are equal, C always sits exactly opposite the diamond and twice as far from the crosshair. When A and B are the pair, the diamond is the pair's centre; when B and C pair up instead, it still marks A and B.

Then, without warning, the outsider dives back in, there is a brief tangle, and a new arrangement begins, sometimes with different partners. Nothing here is scripted or random: the program works out the motion step by step from Newton's law of gravity. So why does it switch between chaos and calm, and why can nobody say exactly what it will be doing an hour from now?

Temporary pairs and long excursions

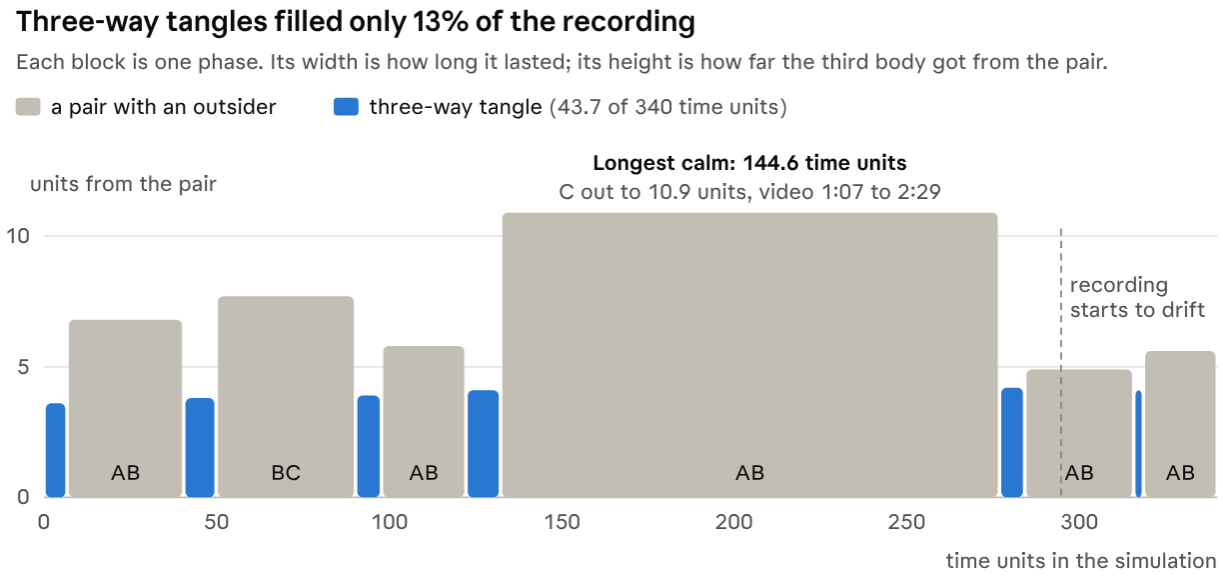
The calm is real but temporary: most of the time the system is a pair plus an outsider, and only briefly a three-way tangle.

Why it looks calm. When the outsider is far away, the two bodies in the pair mostly feel each other, so they orbit like an ordinary double star. The outsider feels the pair almost as one heavy body, so it follows a nearly ordinary orbit too. For a while, one hard three-body problem behaves like two easy two-body problems.

Why it doesn't last. The outsider's orbit brings it back. When it passes close, all three trade energy in a quick encounter. Afterwards the outsider may be flung onto a bigger orbit, swap places with one of the pair, or leave for good.

Why far trips take so long. An orbit twice as large takes almost three times as long to go round (Kepler's third law). So the encounters that throw a body far out create very long calm stretches, and those stretches take up most of the clock.

Here is what happened in the recording of your run, in the stretch where the lab still follows the exact motion:



rec_phases.json in the research files · reference solution, time 0 to 340, video 0:01 to 3:09

The longest calm stretch is also the one in which C swung farthest out, as Kepler's third law leads you to expect. The phases are worked out from the reference solution: the recording stays within 0.02 units of it until about time 295 and within about a tenth of a unit to time 340, far too little to change which bodies are paired.

In simulations run much longer, 20,000 time units each, the pattern holds. Across 11 of them, three-way tangles took 2% of the time on average and never more than 4.4%. The outsider spent about three quarters of the time more than 10 units away, and one trip lasted 9,491 time units, reaching 378 units out before it came back. Most runs eventually end when an encounter gives one body enough energy to leave for good.

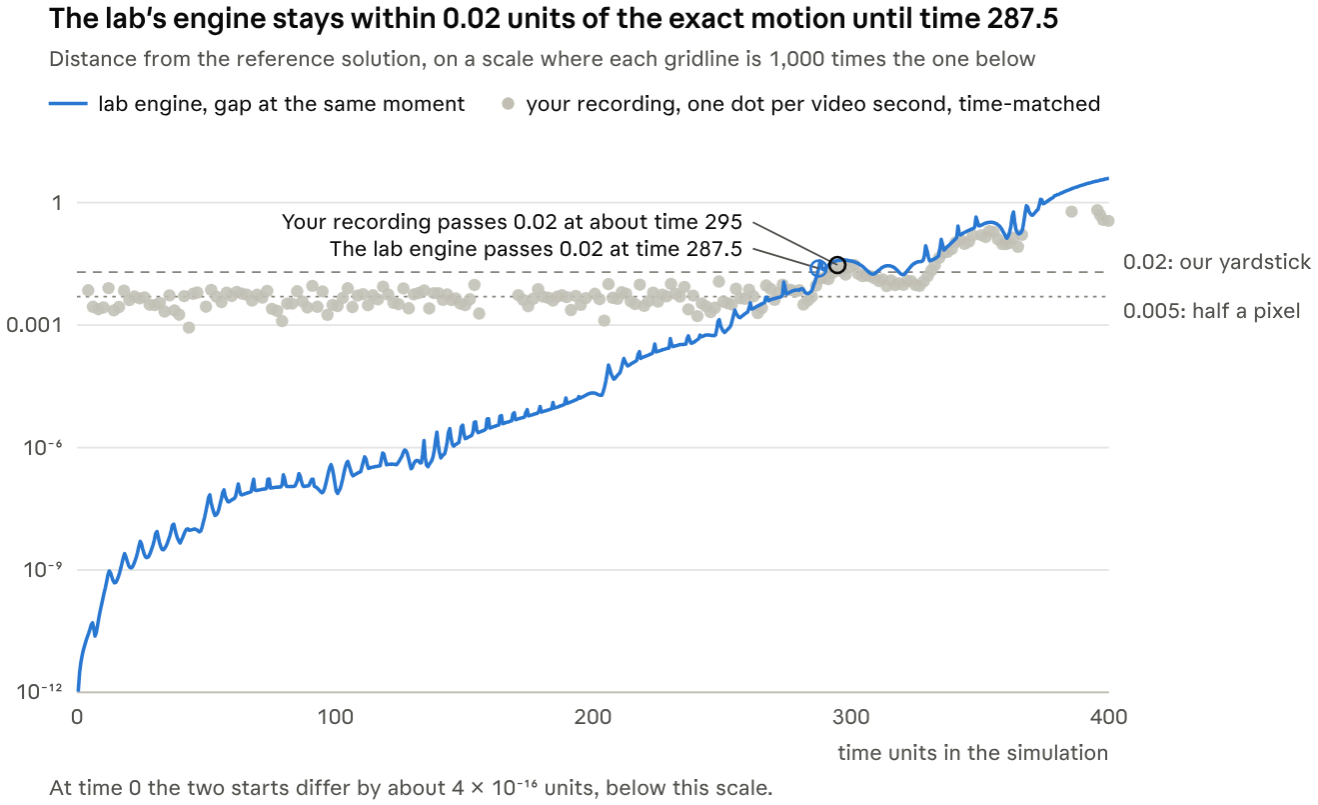
How long can a computer predict it?

In this run, the lab follows the exact motion for about 290 time units, roughly two and a half minutes at the default speed. After that it shows a different, equally lively path.

A computer can only approximate the motion. It moves the bodies forward in small time steps, and each step makes a tiny error: mostly because a step only approximates the curving path (truncation error), and a little because every number is stored to about 16 significant digits (rounding). In this run, the step errors dominate. In a chaotic system tiny differences grow: here, a difference grows about 100-fold every 85 time units. A slip in the tenth decimal place becomes visible on screen in under 300 time units.

To measure this we built a reference calculation with about 19 significant digits, which keeps the total energy constant to about 2.5 parts in 100 million billion. Then we compared two things with it: the lab's own gravity engine, run outside the browser, and your screen recording.

- The lab's engine stays within a millionth of a unit of the reference until time 134.5, and within 0.02 units until time 287.5.
- Your recording stays within 0.02 units of the reference until about time 295, measured from the video frames to about half a pixel.
- Your recording and the lab's engine stay within half a pixel of each other until about time 349: the video faithfully shows what the engine computes.



chart_gap.json in the research files (made by chart_gap.py) · lab engine every 0.5 time units and recording frames, to time 400

On this scale, growing 100-fold every 85 time units looks like a steady climb. Your recording's dots sit near the half-pixel floor while the real gap is too small to see, then rise with the engine's line once it passes that floor. Each dot is matched to the nearest moment of the reference, so the picture shows strong numerical agreement, not frame-by-frame identity.

That "290" belongs to this starting arrangement, this solver, our reference and our 0.02-unit yardstick. Change any of them and the number changes, so it is not a universal limit. Even our best calculation stays within 0.02 units of a second, equally careful run only until about time 650.

Why two computers can give different futures

Two computers solving the same equations from the same start agree only until their tiny differences in rounding and step size grow big enough to see. This study caught that happening four times:

- **Same method, different program.** The lab's engine and our own copy of its method agree to within 0.0000000003 units for the first 100 time units, yet drift 0.02 units apart by time 418.
- **The 16th digit.** The lab works out its starting speeds with slightly different rounding, so its start differs from ours in the 16th significant digit. That alone grows to 0.001 units by about time 476.
- **Two accuracy settings.** We ran the same 24 starting arrangements at two accuracy settings. The two copies of a start ended the same way in only 7 of the 24.
- **The live page and the engine.** Your live lab and the lab's engine run outside the browser split apart after about time 350, because the page advances time in screen-frame-sized chunks.

After the paths split, each computer still shows motion that obeys the same rules in every visible way: pairs, outsiders, encounters, escapes. Weather forecasts are similar: the physics is the same everywhere, but after a week the details depend on whose computer you ask. So the lab stays a good way to see how gravity works. Two things do not follow, though. A later path is not shown to be the exact future of some slightly different start. And obeying the rules does not by itself make the statistics right: how often encounters happen, how long trips last and how runs end have to be checked separately, by running many starts and comparing them, as Part 2 does.

Things to try

These are things to look at, not questions to answer.

- **The balance.** During a wild tangle, keep your eye on the crosshair and the diamond. Does C ever stop sitting exactly opposite the diamond, twice as far from the crosshair, even while C is part of a pair?
- **Calm against chaos.** Using the time readout at the top right, time a few tangles and the calm stretches between them. Which takes up more of the clock?
- **The energy readout.** When the lab's engine was run outside the browser, its "Scaled energy drift" was about 1.55×10^{-9} at time 370. Treat that as a reference result, not a number your screen must show: once the live page's steps differ from that run, its readout can differ too. Compare it with yours. In the reference run the path had already drifted by then, yet the energy had barely moved: the small errors of each step shift the bodies along their paths much more than they change the total energy.

- **A second viewing.** Run the same settings twice and note when the two runs stop looking alike. Your screen's frame rate matters too: in your recording the lab's clock slowed from 1.96 to 1.59 time units per second.
- **A nudge.** Lower C's tangential speed to 95% or 90% and see how soon the first big tangle comes.
- **The pair's view.** Switch the viewpoint to "Inner binary center" and watch the outsider swing round a pair that seems to stand still.
- **The ending.** Leave a run going. In our longer simulations, a small sample from this one starting arrangement, most runs ended with one body escaping, typically after half an hour to an hour of watching at the default speed, but some went on for many hours. What happens in yours?

Part 2 — For teachers and advanced readers

Model and exact initial conditions

The system is the planar Newtonian three-body problem with three point masses of 1, G = 1 and no softening, written in the centre-of-mass frame.

$$\ddot{\mathbf{r}}_i = \sum_{j \neq i} \frac{\mathbf{r}_j - \mathbf{r}_i}{|\mathbf{r}_j - \mathbf{r}_i|^3}, \quad i, j \in \{A, B, C\}$$

Newton's law of gravity for three bodies of mass 1, with G = 1 (vectors in bold)

The state is y = (xA, yA, xB, yB, xC, yC, vxA, vyA, vxB, vyB, vxC, vyC). The lab's settings were C starting distance 3.5, C tangential speed 100%, C radial speed 0 and binary separation 1.8. They correspond to this construction: the suns on the x-axis 1.8 apart with A on the side away from C, relative speed $\sqrt{(2/1.8)} \approx 1.0541$ anticlockwise; C 3.5 from the suns' midpoint along +x with speed $\sqrt{(3/3.5)} \approx 0.9258$ along +y, the circular speed counting all three masses; everything then shifted to the centre-of-mass frame.

Body	x	y	vx	vy
A	-2.066666666666667	0	0	-0.8356529766189138
B	-0.2666666666666667	0	0	0.2184395767705461

Body	x	y	vx	vy
C	2.333333333333333	0	0	0.6172133998483675

These are the float64 values every Python integrator in this study started from. Total energy E0 = -0.603951603951604 (kinetic 0.563492, potential -1.167444); angular momentum L0 = 3.1089302. The lab's engine computes the same start with rounding that differs by at most 4.4×10^{-16}.

Observation: the construction was checked against your recording. At t = 3.40, the first frame, it reproduces all three bodies' positions to within 0.003 units. The alternative reading, C's speed counting only the suns' mass, misses by 0.34 units.

Numerical methods and the reference solution

Every comparison is measured against one reference: a Taylor-series integration in 80-bit extended precision. Double-precision runs and the lab's own engine are judged against it.

Integrator	Method	Error control	How samples are taken
Taylor LD (reference)	Taylor series from automatic-differentiation recurrences, numpy longdouble (machine epsilon 1.08×10^{-19})	order $N = \lceil -\ln \varepsilon / 2 \rceil + 1 = 23$ at $\varepsilon = 10^{-19}$; step $h = \min(\rho_{22}, \rho_{23}) / e^2$ with $\rho_j = (s / \|c_j\|\infty)^{(1/j)}$, $s = \max(1, \|y\|\infty)$ (Jorba & Zou 2005)	the step's own Taylor polynomial evaluated at the sample time
DOP853 (SciPy)	explicit Runge–Kutta 8(5,3)	step accepted when the RMS over all 12 components of err / (atol + rtol · max(|y_old|, |y_new|)) is at most 1	7th-order dense output, or a restart at every sample (no interpolation)
Radau (SciPy)	implicit Radau IIA, order 5	same RMS norm	collocation polynomial, or a restart at every sample
RK45 (SciPy)	Dormand–Prince 5(4)	same RMS norm	4th-order dense output

Integrator	Method	Error control	How samples are taken
Lab engine (WHTS v27)	Dormand–Prince 5(4), rtol 10^{-10}, atol 10^{-12}, maxStep 0.04	the maximum, not the RMS, over components of the scaled error	exported by the independent AI reviewer's headless run every 0.5 units to t = 500
Python copy of the lab engine	same tableau and settings	maximum-component norm; controller h·min(5, max(0.2, 0.9 · err^(−1/5))) assumed	free steps landing on samples, or frame-sized chunks

The gap between two runs is, at each sample, the largest of the three bodies' Euclidean position differences. Video frames are scored differently: each frame is matched to the reference time that minimises the RMS position error.

Observation: two Taylor reference runs with different step sizes stay within 10^{-9} units of each other until t = 348.5, and within 0.02 until t = 649.5. The reference's energy error stays below 2.5×10^{-17} and its angular-momentum error below 1.2×10^{-17} to t = 900.

Interpretation: at the 0.02-unit level the reference is reliable across the whole 0 to 500 comparison window, but at the 10^{-9} level only to about t = 350. Note that numpy's longdouble is 80-bit only on some platforms, such as x86-64 Linux (the independent AI reviewer's environment reports the same epsilon). Where it is plain double precision, the Taylor reference loses its advantage.

Convergence and trajectory comparisons

The lab's engine leaves the reference by 0.02 units at t = 287.5, exactly where an independent copy of its method does, and the screen recording follows the engine to within measuring noise until about t = 349.

Observation, accuracy ladder (sim time at which each run first moves the stated distance from the Taylor reference; continuous runs, dense output):

Run	atol	10^{-6} units	0.001 units	0.02 units	Largest relative energy error to t = 370
DOP853, rtol 10^{-8}	10^{-10}	18.0	139.5	205.5	2.0×10^{-6}
DOP853, rtol 10^{-9}	10^{-11}	50.5	204.5	241.5	2.0×10^{-7}
DOP853, rtol 10^{-10}	10^{-12}	118.5	230.0	274.0	1.7×10^{-8}
DOP853, rtol 10^{-11}	10^{-13}	159.0	273.0	335.0	1.5×10^{-9}

Run	atol	10^{-6} units	0.001 units	0.02 units	Largest relative energy error to t = 370
DOP853, rtol 10^{-12}	10^{-14}	206.0	329.0	373.0	1.4×10^{-10}
DOP853, rtol 10^{-13}	10^{-15}	254.5	373.0	431.0	1.1×10^{-11}
DOP853, rtol 3×10^{-14}	10^{-16}	288.0	431.0	503.0	3.0×10^{-12}
Radau, rtol 10^{-11}	10^{-13}	273.5	411.5	470.5	1.7×10^{-11}
RK45, rtol 10^{-11}	10^{-13}	149.5	261.0	329.0	1.3×10^{-9}
Lab engine (WHTS v27), rtol 10^{-10}, maximum norm	10^{-12}	134.5	248.5	287.5	4.4×10^{-9}
Python copy of the lab engine	10^{-12}	134.5	248.5	287.5	4.4×10^{-9}

Observation, other comparisons:

- Restarting at every sample instead of interpolating shifts these departure times by at most 11 units. DOP853 at 3×10^{-14} and Radau at 10^{-11}, compared directly with each other, part by 0.02 units at t = 470.5.
- Random starting-state differences, 10 seeded directions each, part from the reference by 0.02 units between t = 335 and 390 for size 10^{-11}, 274 and 348.5 for 10^{-10}, and 248.5 and 288 for 10^{-9}. Departure times cluster at the same few moments, because the gap grows in jumps at close passes. Between them, a gap grows about 100-fold every 85 time units (a fitted rate of 0.054 per time unit).
- The lab engine and the Python copy stay within 2.8×10^{-10} units of each other to t = 100, within 10^{-6} until t = 248.5, and within 0.02 until t = 418.
- In the copy, how time is advanced (free steps, fixed 60 fps frames, uneven frames) leaves the 0.02 departure at t = 287.5. Changing the assumed controller constants gives 287.5 to 288.0; using SciPy's RMS norm instead of the maximum gives 274.0.
- Scored with the video's time-matched measure, the lab engine passes 0.02 units at t = 288.5 and 0.1 at t = 340.5; the recording passes them at t = 294.8 and 340.6.
- Frame by frame against the lab engine, the recording's median mismatch is about 0.005 units (half a pixel, the measuring noise) up to $t \approx 349$. Its first frame above 0.02 is at t = 348.7.

Interpretation: the lab's departure from the exact motion at about t = 288 is fully accounted for by its integrator's truncation error at rtol 10^{-10}. The recording is strongly numerically consistent with the engine. Because each frame was time-aligned by optimising positional agreement (playback speed varied), this is consistency to within measurement resolution, not proven frame-by-frame identity.

Conservation checks and timing

Momentum, energy and angular momentum were tracked in every run; none drifted enough to explain any finding.

Observation:

- **Momentum.** In every frame measured by hand, seven across two of your recordings and across calm and chaotic stretches, C sat exactly opposite the gold diamond, which marks the midpoint of A and B, and twice as far from the fixed centre of mass, 2.0 to 1. That is `r_C = −2 r_AB`, with `r_AB` the A–B midpoint: for equal masses it holds exactly in the centre-of-mass frame, whichever pair happens to be bound.
- **Energy, short runs.** The reference's relative error stays below 2.5×10^{-17} to t = 900. The lab engine's is 4.437×10^{-9} at t = 370. The lab displays this as 1.548×10^{-9}, because its "Scaled energy drift" divides by kinetic plus absolute potential energy at the start (1.730936) rather than by the absolute total energy (0.603952).
- **Angular momentum.** The reference's error stays below 1.2×10^{-17}. The DOP853 runs range from 1.7×10^{-7} at rtol 10^{-8} to 4.9×10^{-13} at 3×10^{-14}, up to t = 370.
- **Energy, long runs.** Over 20,000 time units the relative drift stays below 4.6×10^{-7} at rtol 10^{-11} and below 2.7×10^{-9} at 10^{-13}. The two return integrations reach 1.3×10^{-6} (run 4, about 135,000 time units) and 1.0×10^{-5} (run 7, about 465,000).
- **Constrained starts.** The 24 starts built to keep energy and angular momentum fixed match the exact start to within 4.4×10^{-16}, confirmed by the independent AI reviewer.
- **The lab's clock.** In the recording, the lab advanced 1.96 time units per second of video in the first 20 seconds, slowing steadily to 1.59 between 3:00 and 3:25 (average 1.774; setting 2.0). Video times in the workbook come from matching each frame, so they already include the slowdown.

Interpretation: the slowdown most likely reflects the browser's rendering pace. It does not change the physics: in the solver copy, advancing time in uneven frame-sized chunks left the departure time unchanged.

Classification rules: phases, escapes and survivors

The phase shares, escape counts and survivor verdicts in this study all come from the fixed rules below. The thresholds (3 semi-major axes, 10 units, 150 units) are conventions, so shares would shift somewhat under other choices.

Phases, checked at every sample (every 0.1 time units for the recording, every 0.5 in long runs):

1. Take the pair with the lowest two-body energy, $E = |\Delta v|^2/4 - 1/r$ for unit masses.

2. If that pair is bound and the third body is farther than 3a from the pair's barycentre, with a = 1/(2|E|) the pair's semi-major axis, the state is "pair plus outer body" (AB, AC or BC). Otherwise it is "three-way interplay".
3. A **wide excursion** is a pair-plus-outer-body sample with the outer body more than 10 units out. A **pair switch** is a change of pair between consecutive pair-plus-outer-body phases.
4. For the recording only, phases shorter than 1.5 time units are merged into their neighbours.

Escape rule in the runs. Whenever any body crosses 150 units from the centre of mass moving outward, take the closest bound pair and compute the third body's energy relative to it, $E_{out} = \tfrac{1}{2} \cdot (2/3) \cdot |V|^2 - 2/R$. If $E_{out} > 0$ the run ends as an escape; otherwise it continues and the next outward crossing is checked again. Runs stop at t = 20,000.

Osculating verdicts for the end state of each run. Relative to the most tightly bound pair, the outer body's orbital energy per unit reduced mass is $\varepsilon = |V|^2/2 - 3/R$. The margin is $3 \times (\tfrac{1}{2}\, a^2/R^3 + |E(t) - E0| \div (2/3))$: a heuristic for the pair's tidal correction plus the run's whole energy error. If $\varepsilon < -\text{margin}$ the body is "bound now"; if $\varepsilon > +\text{margin}$ and it is moving away, it is "escaping now"; otherwise "unresolved".

Observation (11 futures): 8 runs end "escaping now", with ε between +0.009 and +0.076 at 225 units and margins of 8.3×10^{-7} or less. Three are "bound now":

Run	Outer body, distance at t = 20,000	ε	Margin	Return within 10 units: Kepler estimate	Return: fully integrated
4	B, 1,410 units, moving away	-1.35×10^{-3}	5.2×10^{-7}	114,772.64	114,772.59
7	C, 1,436 units, moving away	-5.9×10^{-4}	1.1×10^{-6}	444,675.73	444,675.71
8	C, 5.6 units	−0.245	0.011	already within 10	—

Interpretation: these verdicts describe the current osculating two-body orbit with a heuristic margin. They are diagnostics, not proofs of indefinite boundedness or of escape. The full integrations confirm that both distant bodies came back. Their timing is uncertain by up to about 0.1% (run 4) and 2% (run 7), because the accumulated energy error cannot be cleanly separated from the outer orbit. Neither return shows that the system stays bound afterwards.

Ensembles and statistics

Long-run outcomes here are statistics of small samples: they show what kind of behaviour this system has, not general probabilities. Individual outcomes are not robust, and medians from groups of 24 runs scatter by a factor of two.

Design:

Ensemble	Runs	Perturbation of the exact start	Solver	What it tests
Original futures	11	C's x position + k × 10^{-9}, k = 0 to 10 (one line)	DOP853 rtol 10^{-11}, atol 10^{-12}	a first look; all 11 share their first 220 to 300 time units
Random	24 starts × 2 tolerances	random 12-D direction (seed 20261127), centre of mass fixed and at rest, length 10^{-6}; energy shifts up to 1.46×10^{-6} relative	DOP853 rtol 10^{-11} / atol 10^{-12} and 10^{-13} / 10^{-15}	tolerance sensitivity, paired by start
Constrained	24 starts × 2 tolerances	the same raw draws, projected to keep energy and angular momentum fixed (to 4.4×10^{-16})	as above	whether those small energy shifts matter; a matched experiment, not an independent replication

Observation, outcomes by t = 20,000:

Group	C thrown out	A sun thrown out	Still together	Median end time	Middle half of end times	Time in three-way interplay	Time on wide excursions
Original 11	4	4	3	5,577	4,652 to 17,675	2.1%	74%
Random, rtol 10^{-11}	8	15	1	2,904	2,174 to 3,694	4.2%	71%
Random, rtol 10^{-13}	7	15	2	4,327	2,773 to 9,409	4.2%	72%
Constrained, rtol 10^{-11}	11	12	1	5,458	4,272 to 7,767	3.7%	74%

Group	C thrown out	A sun thrown out	Still together	Median end time	Middle half of end times	Time in three-way interplay	Time on wide excursions
Constrained, rtol 10^{-13}	13	8	3	5,884	3,131 to 12,101	3.2%	79%

Runs still together count as 20,000 in the medians and quartiles; all of them were "bound now" at the end.

Observation, paired comparisons (same starts; counts of which run ended first, then effect sizes with 95% intervals):

- **Tolerance, random starts:** the rtol 10^{-13} run outlasted its 10^{-11} twin in 18 of 24 pairs (sign test p = 0.023). The pair-stratified hazard ratio is 3.0 (1.1 to 9.2) and the median end-time ratio 1.49 (1.13 to 2.66), but the mean difference, +2,077 time units (−785 to +4,751), includes zero.
- **Tolerance, constrained starts:** 14 of 24 (p = 0.54), hazard ratio 1.4 (0.6 to 3.5). Pooled over both ensembles: 32 of 48 (p = 0.029).
- **Constraint:** at rtol 10^{-13}, 13 against 10 (hazard ratio 1.3, 0.5 to 3.3); at rtol 10^{-11}, the constrained start lasted longer in 17 of 24 (p = 0.064, hazard ratio 2.4, 1.0 to 6.9).
- **Same start, same outcome** at both tolerances: 7 of 24.

The hazard ratios assume a pair-stratified proportional-hazards model; the first-event counts and their exact intervals stand without it.

Interpretation: whether tolerance shifts lifetimes is unresolved, since the random set suggests it and the constrained set does not confirm it. Holding energy and angular momentum fixed shows no detectable effect at rtol 10^{-13}, as expected physically: those shifts (about 10^{-6}) are tiny next to the energy one close encounter exchanges. The original 11 are not representative, because their perturbations are too small to separate them before t ≈ 220.

Consistency with research: bound, non-hierarchical triples of point masses almost always break up into one escaping body and a bound binary (Stone & Leigh 2019). The long tail of their lifetimes is dominated by long, barely bound excursions, called Lévy flights (Manwadkar, Trani & Leigh 2020). Our runs illustrate both; their numbers are not estimates of general probabilities.

What we observed and what we interpret

Each claim in this case study is one of three kinds: an observation we computed or measured, an interpretation of those observations, or a question we leave open.

Claim	Kind	Evidence or reason
The lab integrates Newton's equations for three point masses with Dormand–Prince 5(4), rtol 10^{-10}, atol 10^{-12}, maxStep 0.04 and a maximum-component error norm	Observation	the WHTS v27 source as read by the independent AI reviewer, and its exported states
The lab's engine moves 0.02 units from the reference at t = 287.5	Observation	engine export against the Taylor reference
The recording matches the engine to within measuring noise until about t = 349	Observation, with frames time-aligned by best fit	frame-by-frame comparison
The departure near t = 288 comes from the integrator's truncation error	Interpretation, strongly supported	an independent copy departs at the same time; frame timing and controller details barely change it; it fits the accuracy ladder
"About 290" is the accuracy horizon	Interpretation, conditional	holds only for this start, solver, reference and 0.02-unit criterion
After departing, the screen shows a computed path that illustrates possible gravitational mechanisms	Interpretation	same encounter and excursion patterns; no claim that it is the exact path of any nearby start (no shadowing result), and no claim that its frequencies or lifetimes are right without separate ensemble checks
The system spends most of its time as a pair plus an outsider	Observation, under the stated rules	87% of the recording's first 340 units; 95.6% to 99.5% in the 11 long runs
Calm stretches last long because far orbits take disproportionately long	Interpretation	Kepler's third law; long excursions dominate lifetime tails in published work

Claim	Kind	Evidence or reason
Most runs end with one body escaping	Observation, small samples	8 of 11 futures; 21 to 23 of 24 in each ensemble group
Bound non-hierarchical triples almost always break up eventually	Interpretation from published research	Stone & Leigh 2019; our runs are consistent with it
Two distant survivors were bound at t = 20,000 and their outer bodies came back	Observation	osculating energy far below zero; full integrations to the return
A tighter tolerance lengthens lifetimes	Open question	paired tests disagree between the two ensembles
Holding energy and angular momentum fixed changes lifetimes	Not detected	no difference at rtol 10^{-13}; 24 matched starts

Remaining limitations

None of these changes the conclusions in the table above, but each bounds how far they reach.

- **One starting state.** Every horizon and statistic belongs to C at 3.5 and suns 1.8 apart; other starts need their own runs.
- **No shadowing.** We show where computed paths depart from the reference, not that a later path is the exact future of some nearby start.
- **Video measurement.** Frames were sampled once per second and measured to about 0.005 units (half a pixel at 80 pixels per unit, assumed constant). Each frame was time-aligned by best fit, because the lab's playback rate varied.
- **The lab engine.** The independent AI reviewer read the source and ran the engine headless; our Python copy assumed the step controller, and the version 27 source shows the real one differs slightly (step factors limited to 0.25 to 4 rather than 0.2 to 5, plus an extra step limit during close encounters). The lab's numbers come from the engine's own export, and sensitivity tests show such controller details move the 0.02 departure by at most half a time unit. The live page's own frame-by-frame stepping was not exported, so its split from the headless run near t = 350 is inferred from the recording.
- **Reference precision depends on the platform.** numpy's longdouble is 80-bit on x86-64 Linux, but on platforms where it is plain double the Taylor reference loses its advantage.

- **Small, partly matched ensembles.** Groups of 24 runs with heavy-tailed lifetimes, an arbitrary perturbation length of 10^{-6}, and constrained and unconstrained starts that share random directions. The ensembles also used a faster force function, which rounds differently from the one used elsewhere.
- **Censoring and escape checks.** Runs stop at t = 20,000, escapes are checked only when a body crosses 150 units, and the bound or escaping verdicts are osculating diagnostics with a heuristic margin. The two return times carry the uncertainty given above.
- **Classification conventions.** The phase shares depend on the 3-semi-major-axis and 10-unit thresholds.
- **Physical idealisation.** Point masses with no size, so no collisions, and three equal masses: the lab's "imagined world" C weighs as much as each sun, as the lab itself notes. The lab's gravitational-wave overlay is an illustration and was not part of this verification.

Research files, reproducibility and attribution

Every number in this case study can be traced to code and raw output kept on a companion page, Restless World research files. It lists 54 files with a line on what each one is, including the scripts and rows behind both charts, shows the full README, and offers the whole set as one 434 KB archive with a SHA-256 checksum, so a reviewer can confirm they hold the same copy. The tables behind each section are in the workbook A Restless World — exact-start analysis.

What the files contain

- **The model:** the exact starting state and equations of motion (triple.py), the extended-precision Taylor reference (taylor_ld.py), and the rules for phases, escapes and bound-versus-escaping verdicts (phases.py, lifetime.py, orbits.py).
- **The experiments:** one script per workbook tab. The README maps each tab to its script, solver, tolerances, sampling interval and time span.
- **The raw results:** every script's output, the reference positions every 0.5 time units to t = 900, and the lab engine's own export (WHTS v27, states every 0.5 to t = 500).
- **The video measurement:** the tracked body positions and the frame-to-time matches. The screen recordings themselves are not included.

How to rerun the calculations

1. Use Python 3.13 with NumPy 2.5 and SciPy 1.18 on x86-64 Linux. The Taylor reference relies on NumPy's 80-bit long double; on machines where that type is ordinary double precision, it loses its advantage.
2. Run ref_compare.py first. It writes the reference positions that the convergence, start-difference and lab-engine checks read.
3. Run any other experiment script. Each writes its result file, which can be compared with the archived copy.
4. On the companion page, scripts carry an extra .txt so they open in a browser. The archive keeps the original names.

To check a later version of the lab, export its states as CSV with the header t, xA, yA, xB, yB, xC, yC, vxA, vyA, vxB, vyB, vxC, vyC, at t = 0, 0.5, 1 and so on, with 17 significant digits in the centre-of-mass frame. Then run verify_lab_csv.py and lab_csv_timematched.py on it.

The lab's source code

The live lab, A Restless World, is one self-contained web page with its JavaScript integrator written into it: View Page Source in any browser shows the code that runs. Use it to experiment. The live page can change after version 27, the version studied here, so a version-fixed archive of the version 27 source is kept on the research files page. The independent AI reviewer checked it against the original 41-file release, and the copy received matches both SHA-256 checksums: for the archive, 0862248511548ffcab9555d55c2c39f82361405c02ef65f8d00291633fd8fdd8, and for the lab page inside it, ddb3c72ebeac82ef3e9d0a4869bbfc601872b558c6831b9b719a95db71ef41f3. Beside it, the engine export records what version 27 computed from the exact start.

What is not in the archive, and what the scripts regenerate

- **The screen recording is not included.** Only track.py reads it. Its output, the tracked body positions, is included, so every later step runs without the video.
- **The lab engine export comes from outside the scripts.** It was produced by running version 27 outside the browser; the scripts only read it.
- **The reference positions are regenerated.** ref_compare.py writes them and five later scripts read them, which is why it runs first. The gap chart's script and lab_csv_timematched.py compute a finer reference internally each time and store none.
- **Some results feed later steps,** and two ensemble scripts add to their output files rather than replacing them. The README's dependency table lists each file, the script that writes it and the scripts that read it.

- **Regenerated is not always identical.** The reference depends on how the computer stores extended-precision numbers, so on another machine it should agree within the stated tolerances, not digit for digit.

Attribution

This work is a human–AI collaboration with additional independent investigation and review by other AI. Four rounds of independent review shaped the checks in Part 2: the choice of reference solution, the bound-versus-escaping verdicts, the paired statistics, and the direct comparison with the lab's own engine.