ChatGPT

Newton 4 – 2D Physics Engine for After Effects

Newton 4 is a powerful plugin for Adobe After Effects that turns 2D layers into bodies in a physics simulation. Rather than a standard effect, Newton is a *composition* command – you launch it from After Effects' Composition menu [1] [2]. When you open Newton, a clean cross-platform interface appears with an OpenGL preview [3] [4]. In this interface you assign layers in your comp as "bodies" and define physics parameters (gravity, friction, joints, etc.). Newton then simulates real-world physics on those 2D bodies and exports the result back to After Effects as keyframes. In fact, the plugin **"only creates keyframes"** [5] – it never bakes frames or renders images. All motion is ultimately delivered as AE transform keyframes you can edit or retime.

Internally, **Newton uses the Box2D physics library** (a popular 2D game physics engine) to perform its calculations [6]. To match AE's pixel units to Box2D's meters, Newton scales compositions so that **100px in AE = 1 meter in Newton** [6]. The solver is tuned for object sizes from 0.1–10 m (i.e. 10–1000 px). You can adjust the simulation precision via parameters like *Time Divider* (slows down time, like high-precision slow motion), *Collision Tolerance* (controls gaps/overlap when collisions occur), and *Substeps* (increases solver iterations) [7]. These allow you to trade off accuracy for performance. For example, a smaller Collision Tolerance or more Substeps yields fewer tunneling artifacts at the cost of compute time [8] [9].

All input to Newton comes from your AE layers' **shapes**. Newton **ignores** audio layers, guide layers, 3D layers, and any layer whose video is disabled [10]. It also does **not** interpret pixel data or layer styles – it only sees the *path outlines* of shapes. (If you supply a precomp or footage layer, Newton treats it as a simple rectangle body.) Complex shapes are automatically tessellated into polygons, with a **Mesh Precision** parameter controlling how finely corners are resolved [11]. If your shapes have many points or curves, performance can suffer; in those cases you can enable the **Convex Hull** option to simplify a shape into its convex envelope [12] [13]. As Newton's FAQ notes, "Make sure each shape has the minimum acceptable value for Mesh Precision... and use the Convex Hull setting to simplify your shape" [14]. In practice, designers often redraw complex art as simpler mask shapes to improve speed. Newton does not support AE mask blending modes or inverted masks – it strictly uses the Path vertices to build geometry [15].

Body Types and Properties

Any layer you add becomes a **body** in the simulation, and you can choose its type and physical properties. Newton 4 supports seven body types [16] [17]:

- **Static** – a non-moving body (immovable wall or anchor).
- **Dynamic** – a fully simulated body; it's driven entirely by physics.
- **Kinematic** – an animated body (with AE keyframes) that moves along its path, but turns dynamic at the end of its animation.
- **AEmatic** – a hybrid of kinematic and dynamic. An AEmatic body follows AE animation (via keyframes or expressions) *while* also feeling physics forces. In other words, you can keyframe it but it will still collide and respond to gravity along the way.

- **Dormant** – initially unmovable (no gravity) until it's hit by another body, at which point it becomes dynamic.
- **Dead** – a placeholder body that participates in the simulation (it collides) but does not react (i.e. receives no forces).
- **Triggermatic** – a special type that stays dormant until triggered by another "sensor" body colliding with it. Once the trigger occurs, the Triggermatic body plays out a pre-set AE animation (see below).

Each body has standard physics properties:

- **Density (mass)** – higher density makes a body "heavier" in collisions (though gravity acceleration is the same for all masses). Two bodies of different density will react differently on impact [18].
- **Friction** – controls sliding resistance when surfaces contact. Ranges from 0 (no friction, very slippery) to 1 (high friction) [19].
- **Restitution (Bounciness)** – how much energy is conserved in a collision. 0 means no bounce (stops dead), 1 means perfect elastic bounce. Intermediate values give softer bounces [20].
- **Color** – purely visual color for the preview (does not affect physics) [21].
- **Mesh Precision** – number of points for rounded corners; higher = more accurate shape, slower simulation [11].
- **Initial Velocity** – you can give any body a starting linear or angular velocity. A velocity arrow tool (shortcut P) lets you draw the direction and speed of initial motion [22]. This is useful for e.g. giving a body a push at the start.
- **Damping** – linear and angular damping slow a body over time (like air resistance). These apply continuous drag to reduce velocity [23].

In the **Advanced Body** panel, you get further controls (not mentioned on the main promotional pages):

- **Collision Group / Collide With** – five groups let you mask collisions. Bodies in Group A can be set to only collide with, say, groups B and C, etc [24]. (Default is "collide with all").
- **Fixed Rotation** – locks a body's rotation (so it can translate but not spin) [25].
- **Gravity Scale** – per-body gravity multiplier. 0 = no gravity, 1 = normal, <0 = anti-gravity [26]. You can even inherit gravity direction from a layer's position – for example, a Null moving in 2D space can drive the gravity vector in real time [27].
- **Hidden** – makes a body invisible in the preview, though it still collides (useful if you just need an invisible stopper or sensor) [28].
- **Use Convex Hull** – approximates a complex shape by its convex hull, greatly increasing speed for concave shapes [12] [13].
- **Export Contacts** – a powerful feature new in Newton 4: this toggles whether Newton records every collision ("contact") event of that body as AE keyframes. When enabled, Newton adds a **Contacts** slider effect to the layer, with a keyframe at each collision [29] [30]. You can use these contact keyframes in expressions or layer triggers (for example, triggering a sound or particle burst on each impact). Motion Boutique provides sample projects showing how to use the contacts slider with expressions [30] [31].
- **Magnetism** – you can turn a body into a magnet. With **Type = Attract or Repulse**, the body will pull in or push away nearby bodies. You also set an **Intensity** and effective **Distance** [32]. (Bodies can also be set to *Accept* magnetism so one attracts another.)
- **Waterlike (Buoyancy)** – enabling this makes a body act like fluid. Other bodies that overlap it feel an upward buoyancy force (opposite to gravity). You can set the water body's **Density** (low density =

easy to penetrate, high density = solid) and **Drag/Lift Factors** which tweak how it pushes on objects [33] [34]. This lets you simulate floating or objects bobbing on an invisible fluid.

- **Grenade (Explosive)** – this mode makes the body explode outward at the start of the sim. You configure an explosion **radius, intensity, number of rays**, etc., in a special Grenade Settings dialog [35] [36]. (Note: grenades only work on simple shapes like circles/rectangles/triangles [37].) When triggered, lines show which bodies will be hit [37]. This is useful for sudden blasts or scattering effects.
- **Teleportation / Portals** – Newton lets you set up warp portals. First, mark a body **Teleportable**. Then assign it a **Teleport Trigger** (on collision begin or end) and a **Destination** body [38] [39]. When a Teleportable body touches a portal body, it will instantly move to the destination body. You can choose whether it retains its original velocity or copies the destination's velocity, and whether its rotation is preserved [40] [41]. In effect, any body can become a portal entrance to any other body. Even the composition edges ("walls") can be turned into loop-around portals [42] [43]. For example, enabling **Teleportation Portal** on a wall teleports a body to the opposite side or a specified body when it leaves the comp [42] [43].

Finally, you can also **hide** bodies, adjust **collision filtering** per body, and more via the Advanced panel. All parameters support keyframing: you can animate any property (even density, magnet strength, etc.) over time. A built-in Keyframe panel lets you set values with curves and loops [44] [45]. This means you can have a body suddenly change mass mid-sim, or toggle its joints on and off during the simulation.

Joints and Constraints

Newton 4 provides a rich set of **joints** to constrain bodies together, akin to how mechanical or soft-body rigs work. You select two (or more) bodies in the Newton interface and add a joint of one of the following types:

- **Distance Joint** – fixes the distance between two anchor points on two bodies. It can be *rigid* or *soft* (with tension/springiness). A high tension = stiff rod; low tension = flexible connector. This is commonly used to make "stick" constraints or soft bodies (several small bodies linked by low-tension distance joints to form an elastic mesh) [46] [47]. You can also turn on "Collide Connected" if you want the linked bodies to still collide with each other.
- **Pivot Joint** – forces two bodies to share a common hinge point. Each body can then rotate relative to that pivot. You can set angle **limits** (min/max rotation) and even a **motor** to spin the joint at a set speed/torque [48] [49]. This is like a hinge or axle – for example, pendulums, door hinges, or wheel axles (though see Wheel Joint below). To create one, you usually set the pivot point in the preview with a special anchor tool.
- **Piston Joint (Prismatic Joint)** – allows one body to slide along a line relative to another (no rotation). Think of a piston or slider. You define the translation axis, and can enable limits and a motor to push/pull at a set speed [50] [51].
- **Weld Joint** – glues two bodies together at their relative positions. By default it's fully rigid (bodies move as one), but you can add a small *tension* to allow a tiny bit of flex [52]. Under the hood, a weld joint is like a stiff distance joint that resists rotation as well as translation. (Damping can also be added.) It's useful for temporarily welding objects or building compound shapes.
- **Spring Joint** – connects two bodies as if by a spring. It has a **resting length**, a **spring constant (springiness)**, and **damping** [53]. The connected bodies will oscillate around the rest length like a physical spring. This can be used for rag-doll limbs, sloshy connectors, or suspension-like effects.

- **Wheel Joint** – a specialized joint combining a pivot and a spring. It is designed for vehicle wheels. You select two bodies (e.g. a car chassis and a wheel) and specify the axle. The wheel joint provides suspension (a spring/damper) and can include a rotational motor [54] [55]. In effect, it's a wheel on an axle with shock absorber. (Motion Boutique's examples mention selecting the vehicle body first, then the wheel [54].)
- **Blob Joint** – this creates a soft-body effect for a group of bodies. It forces the **area** enclosed by multiple bodies to stay constant. Internally it uses a network of *soft distance joints* between them. You specify a group of bodies and the joint keeps them all together like a squishy blob [56]. It's handy for simulating soft clusters or jiggly masses.

All joint parameters (anchor points, limits, motors, etc.) can be keyframed as well. You can copy/paste joint settings between joints, and even randomize values via the Randomizer tool if you want non-uniform behavior. The UI provides a joint table listing all existing joints, their types, and properties for quick editing [57].

Workflow and Interface

When you're ready to simulate, you hit **Start/Play** in the Newton panel. The OpenGL preview animates in real time (fast previews) showing collisions and responses [3]. You can scrub the timeline too. The preview respects gravity, collisions, and all the body/joint settings. You can pan/zoom the view, select and drag bodies, or show different visualization layers (wireframes, contact points, velocity arrows, etc.). Common commands are also available via a right-click context menu on the preview or body list [58]. There are handy tools like grabbing velocity from the preview, or a Gravity tool (G) that lets you drag to change gravity direction [59]. You can also define **walls**: toggling the top/bottom/left/right composition edges to act as static collision walls with their own friction/bounce [42]. (Walls can even be made one-sided or turned into looping portals [43].)

If you need to iterate, Newton supports **Scene Snapshots**: you can save and restore entire scenes (body placements and settings) for comparison [60]. This lets you tweak parameters, try a simulation, then revert to an earlier snapshot if needed.

When you're satisfied, click **Apply** or **Export** in Newton. This writes out all the resulting motion as standard AE keyframes on the original layers' Transform (position/rotation) properties. Any animated body will now have keyframes at every frame (or custom intervals if you set output settings). You'll see the layer names unchanged in your AE timeline, but now with animated position/rotation matching the simulation. The excellent design point is: **after export, you can still tweak everything** – since it's keyframes, you can move, retime, or even override them in AE if you like, and then re-run Newton to update.

Because Newton only adds keyframes (never bakes images), you can also do things like precomposing simulated layers with effects (blur, color) after export, or drive other animations off the simulation. For example, in a Motion Boutique demo, the *"Export Contacts"* slider was tied to a text layer so that on each collision the text changed (using an expression) [31] [61]. In short, Newton's output plays nicely with all of AE's native features.

Limitations and Tips

- **Strictly 2D:** Newton operates only in 2D space (XY plane). Each body has only X/Y position and Z-rotation. If your scene is truly 3D (e.g. layers flying toward/away from camera), Newton cannot simulate that; it will only treat everything as if on a flat table. As one user notes, "You cannot do [3D coin stacking] with Newton... you have position and rotation on the 3 axes" but the camera's Z is ignored [62]. In practice, people use Cinema 4D, Blender, or other 3D tools for fully 3D physics. Newton is ideal for *flat* compositing scenes.
- **Layer Setup:** Remember to use vector shape layers or masks. If you try to use footage, Newton will treat it as a 100% rectangular block. To get arbitrary shapes, convert Illustrator/vector layers via *Create Shapes from Vector Layer* or draw masks.
- **Mask Limitations:** Don't use inverted or blending-mode masks for body shapes – Newton will misinterpret them [15]. Also, if a layer has multiple masks/shapes, Newton will prompt whether to treat them as one compound body or separate bodies [63]. Often separating into multiple bodies is better for realistic behavior.
- **Performance:** Complex scenes (many bodies, high-precision meshes) can run slowly. Tip: lower Mesh Precision, turn on Convex Hull, and reduce polygon counts on shapes. Also, the FAQ suggests watching the "info panel" for polygon counts and using a low Mesh Precision (like 1 or 2) where possible [14]. If things freeze, you can also reduce Newton's default simulation resolution or reduce substeps.
- **Gap Between Bodies:** Newton's collisions are not pixel-perfect. By default you'll often see a 1-pixel gap when bodies touch. The FAQ notes this and suggests lowering Collision Tolerance [64] if you need tighter contact (at the risk of some overlap glitching).
- **Solver Scale:** Because of the 100px=1m scale, keep your scene sizes in a reasonable range. If your composition is huge (tens of thousands of pixels), consider scaling it down before simulating. Conversely, very small objects (few pixels) might be under-precision.
- **Navigation:** In the preview, use Alt+/– (zero) to adjust stroke width of open shapes if they're hard to see [65]. Use right-click and the context menus for quick tasks (like adding joints or resetting parameters) [58]. The Randomizer tool (in the context menu) can auto-generate randomized values for any numeric property over a range [66]. This is handy to give, say, slightly different masses or speeds to multiple bodies with one command.
- **Missing in Effects Menu:** A common newbie mistake is looking under Effects. Newton is found under **Composition > Newton** [2]. If you don't see it, check that you installed it correctly (AE must be restarted after install).

Community Usage and Unofficial Tricks

Beyond the core features above, the user community has devised many creative uses of Newton 4:

- **Kinetic Typography and Reactive Text:** Several tutorials (some by Motion Boutique themselves) show how to make text animations driven by collisions. For example, using the *Export Contacts* slider, people have set up expressions where each collision triggers a new word or sound. In one demo, collisions changed a text layer's content through a list of onomatopoeic words (e.g. "Boom!", "Crash!") in real time [31] [61]. Another AEScripts tutorial uses Newton with Element 3D models, and then bounces physics-driven text over 3D scenery. In general, using expressions with Newton's outputs opens a lot of interactivity.

- **Combining Plugins:** Many users pair Newton with other Motion Boutique or third-party tools. For instance, **Connect Layers Pro** (another Motion Boutique plugin) can link layers with strokes; some tutorials animate those strokes to react to Newton collisions. Others use **Pastiche** (fluid animations) to make objects appear "liquid" and then use Newton to add gravity/bounce. The combination of Newton's physics with generative or text tools is popular.
- **Audio Reactivity:** Though Newton itself doesn't directly use audio, some makers animate the Newton world in sync with sound. For example, one might keyframe gravity or an explosion timing to a bass drop in music, so that at the moment of a beat, a body explodes or gravity reverses. Others have fed audio-derived numbers into body mass or friction via AE expressions, making the motion subtly change with the soundtrack.
- **Games and Simulations:** People have even prototyped simple physics games using Newton in AE (using expressions to detect contacts and count scores, etc.). The *Triggermatic* body type (and the sensor setup) is especially useful for trigger-based effects – e.g. a swinging pendulum that, when hitting a sensor object, causes another object to burst into pieces (all done by pre-animating the bursting object in AE and using Newton just to detect the hit and release it) [17] [67].
- **Limitations Workarounds:** A known limitation is no built-in rotational motors for *distance* or *spring* joints (only pivot/piston have motors). Some users have faked this by keyframing a body's rotation in AE after export to achieve a similar effect. Another clever trick: use a very thin or invisible static body as a "guide rail" for a moving body, thereby mimicking constraints that Newton doesn't directly support. Because you can animate any parameter, folks animate a body's gravity scale or apply wind fields via slight rotations of gravity to simulate forces that Newton doesn't have natively.
- **Performance Hacks:** If the scene is slow, an unofficial tip is to temporarily switch bodies to Static or Kinematic during setup (so they don't move), then switch back to Dynamic just for the final sim. Or break a complex sim into smaller pieces: simulate half the bodies first, export, then add more bodies and simulate again. Since Newton can import previous keyframes, you can build up scenes in stages.
- **Hidden Features:** Motion Boutique often responds on forums with little-known tips. For example, they note that pressing **W** will remove a selected joint [68], and **Alt+W** does the same. Holding Shift while dragging a joint anchor will snap it to geometry points. These UI shortcuts (and others listed in the manual) can speed up workflow but are often only found by reading the fine manual or noticing them in context.
- **User Stories:** On Reddit and FB groups, users post their Newton experiments. Some have built things like Newton-powered clocks, Newton-driven transitions where one shape shatters another, or Newton-based puzzles (blocks that stack, roll, and solve some spatial challenge). Others use Newton for logo animations – e.g. letting letters fall into place and bounce realistically. Because Newton outputs editable keyframes, many animators also polish the result by hand: smoothing keyframe tangents after the simulation, or adding squash-stretch via additional effects.

Finally, note that **Motion Boutique's own support** (on Reddit and Discord) is active. The plugin creator (Val from Motion Boutique) often answers questions about tricky behaviors [62] [69]. So if you really want to push Newton 4 beyond the manual, the user community and developer channels can share hacks (such as how to set up a multi-body trigger system, or how to trigger a precomp start at contacts). In summary, Newton 4 is an extremely flexible tool: it does *all the features above*, plus anything you can dream up by creatively combining its collisions, joints, and AE expressions in unexpected ways.

Sources: Official documentation and feature lists from Motion Boutique and AEScripts [70] [71] [3] [6] [30]; Motion Boutique's FAQ [2] [64] [15]; and community discussions and tutorials (After Effects forums, Reddit) [62] [31] [69].

---

1 3 4 6 7 8 10 11 12 13 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 63 66 67 68 71 Newton 4.0 - User Guide
https://www.motionboutique.com/files/newton4/

2 9 14 15 64 65 Newton frequently asked questions - aescripts.com
https://aescripts.com/knowledgebase/index/view/faq/newton-frequently-asked-questions/?srsltid=AfmBOoplbPHH7nc_mfJa4AdO3mOkqtyYBydinAFlLX9EhwcUrjqTjImL

5 70 Newton 4 | Motion Boutique
https://www.motionboutique.com/products/newton-4

31 61 Make Your Text React to Collisions in After Effects (Newton + Expressions Tutorial) : r/AfterEffects
https://www.reddit.com/r/AfterEffects/comments/1kwsdpw/make_your_text_react_to_collisions_in_after/

62 Can we able to do this money bounce with newton 4? if yes , can you explain the base with the 2d shapes, so that i can match the position of the 2d shapes with element 3d u/motionboutique : r/AfterEffects
https://www.reddit.com/r/AfterEffects/comments/1hcff8s/can_we_able_to_do_this_money_bounce_with_newton_4/

69 Newton4 Question : r/AfterEffects
https://www.reddit.com/r/AfterEffects/comments/1h5r7o0/newton4_question/