3D Drop Shock — Numerical Method Description

1. Overview

This document describes the numerical method used in the 3D Drop Shock module. The module models a rigid rectangular solid dropped from a specified height onto a rigid ground plane. The box is supported by four linear spring–damper elements located at the bottom corners. The solver integrates the full 3D rigid-body equations of motion, including translation and rotation, with unilateral contact, using an adaptive Runge–Kutta–Fehlberg (RKF45) method (Cash–Karp coefficients).

2. Geometry and Coordinate Systems

World coordinates. The world frame is defined with axes (x, y, z). The ground plane coincides with $z = 0$. Gravity acts in the negative z-direction as a constant acceleration G.

Body coordinates. The box is defined in its own body-fixed frame with dimensions Lx (width), Ly (depth), and Lz (height). The body origin is located at the box center of gravity (CG). In body coordinates, the eight corners of the box are given by ±Lx/2, ±Ly/2, ±Lz/2.

The simulation tracks the rigid-body translation of the CG in world coordinates and the orientation of the box using a unit quaternion that maps body coordinates to the world frame.

3. Mass and Inertia Properties

The user specifies the box weight W and gravitational constant G. The corresponding mass m is computed internally as:

- $m = W / G$

Assuming a uniform rectangular solid, the principal mass moments of inertia about the CG, expressed in the body frame, are:

- $Ixx = (m / 12) * (Ly^2 + Lz^2)$
- $Iyy = (m / 12) * (Lx^2 + Lz^2)$
- $Izz = (m / 12) * (Lx^2 + Ly^2)$

The inertia matrix in body coordinates is therefore diagonal: I_body = diag(Ixx, Iyy, Izz).

4. Corner Springs, Damping, and Contact Model

Four identical linear spring–damper elements are located at the bottom corners of the box. Each spring has stiffness k and damping c, specified per corner. The ground plane is assumed rigid and coincides with $z = 0$ in world coordinates.

For each time step, the corner positions and velocities are evaluated as follows:

- Transform each corner from body to world using the current rotation matrix.
- Add the current CG position r to obtain the world-space corner position p.
- Compute corner velocity as $v_{corner} = v_{CG} + \omega \times r_{corner}$, where ω is the angular velocity in world coordinates and r_{corner} is the corner offset from the CG in world coordinates.

Penetration is measured as the amount the corner lies below the ground:

- $\text{penetration} = \max(0, -p.z)$

If the penetration is positive, a compressive normal force is computed using a unilateral (no-tension) linear spring–damper law:

- $Fz = k * \text{penetration} - c * v_n$

where v_n is the normal component of corner velocity in the vertical (z) direction. If the computed Fz is negative, it is clipped to zero so that the springs do not pull the box downward.

The total force and moment contributions are accumulated over all eight corners:

- Total force F is the sum of gravity and all contact forces.
- Total moment T about the CG is the sum of $r_{corner} \times F_{corner}$ over all corners.

Gravity is modeled as a constant body force acting at the CG:

- $F_{gravity} = [0, 0, -m * G]$

5. Degrees of Freedom and State Vector

The rigid-body has six mechanical degrees of freedom (three translations and three rotations). The implementation uses a 13-component state vector that includes position, velocity, quaternion, and angular velocity:

- $y = [\ r_x, r_y, r_z, v_x, v_y, v_z, q_w, q_x, q_y, q_z, \omega_x, \omega_y, \omega_z\ ]$

Here r is the CG position in world coordinates, v is the translational velocity, q is the unit quaternion representing the box orientation (body to world), and ω is the angular velocity vector expressed in the world frame. Quaternion normalization is enforced after each step to control numerical drift.

6. Equations of Motion

The translational and rotational equations of motion are written in the world frame.

- Translational motion:
  - $m * dv/dt = F$
  - $dr/dt = v$

- Rotational motion:
    - $Iw * d\omega/dt + \omega \times (Iw * \omega) = T$

where Iw is the inertia tensor expressed in world coordinates:

- $Iw = R * I_{body} * R^T$

and R is the 3×3 rotation matrix corresponding to the current quaternion q (body → world).

- Quaternion kinematics:

Let $q = [q_w, q_x, q_y, q_z]$ and define the pure-rotation quaternion $q_\omega = [0, \omega_x, \omega_y, \omega_z]$. The quaternion time derivative is:

- $dq/dt = 0.5 * (q_\omega \otimes q)$

where ⊗ denotes quaternion multiplication. This formulation avoids the singularities associated with Euler angles and provides a smooth representation of orientation.

7. Contact Evaluation per Time Step

For a given state (r, v, q, ω), the contact and load computation proceeds as:

1. 1. Form the world rotation matrix R from the current quaternion q.
2. 2. For each box corner in body coordinates, compute its world position $p = r + R * c_{body}$.
3. 3. Compute corner velocity $v_{corner} = v + \omega \times (R * c_{body})$.
4. 4. Compute penetration = max(0, −p.z).
5. 5. If penetration > 0, evaluate the vertical contact force $Fz = k * penetration - c * v_n$, clip to $Fz \geq 0$, and form the force vector [0, 0, Fz].
6. 6. Add this force to the global force balance and accumulate the corresponding torque contribution $r_{corner} \times F$.
7. 7. Add gravity to obtain the final total force vector F.

The same procedure is used to compute individual spring loads at the four bottom corners, which are tracked for plotting and export.

8. Numerical Integration: RKF45 (Cash–Karp)

The ordinary differential equations for the state y(t) are integrated in time using an embedded Runge–Kutta–Fehlberg 4(5) scheme with Cash–Karp coefficients. For each attempted time step of size h:

• The solver evaluates the right-hand side f(t, y) at six intermediate stages.
• A 5th-order solution y_{5th} and a 4th-order solution y_{4th} are both formed as linear combinations of these stage derivatives.
• The local truncation error estimate is computed from the norm of $(y_{5th} - y_{4th})$.

If the estimated error is below the user tolerance (tol), the step is accepted and the state is advanced to y_5th. If the error is larger than tol, the step is rejected, the time step is reduced, and the step is retried. A safety factor and exponent are used to scale the time step between iterations:

- h_new = safety * h * (tol / err)^(1/5)

The step size is constrained by minimum and maximum limits (h_min, h_max) to avoid overly small or overly large steps. Quaternion normalization is applied after each accepted step.

9. Time Step Limits and Reference Times

To maintain stability and accuracy, the maximum allowable time step is tied to an estimate of the system's natural frequencies. The code computes approximate modal frequencies for heave, pitch, roll, and a diagonal tilt mode of the box–spring system using rigid-body approximations:

- Heave (vertical translation): fz ≈ sqrt(4k / m) / (2π)
- Pitch about x-axis: fx ≈ sqrt(k * Ly^2 / Ixx) / (2π)
- Roll about y-axis: fy ≈ sqrt(k * Lx^2 / Iyy) / (2π)

- Diagonal tilt about an axis in the x–y plane: f_diag ≈ sqrt(K_diag / I_avg) / (2π), with K_diag and I_avg formed from the box dimensions and inertia.

The highest estimated natural frequency f_max is used to set the maximum solver time step as:

- Δt_max = (1 / f_max) / 20

This corresponds to at least 20 integration steps per period of the highest mode.

A reference drop time is also computed from the free-fall relation:

- t_drop = sqrt(2h / G)

Together with the lowest significant natural period, these quantities are used to define a reference time T_ref and an overall simulation end time t_end. The user's 'speed' slider scales the target simulation time step relative to this reference.

10. Output Quantities and Plots

The solver records time histories of several key response quantities for visualization and export.

- CG acceleration:

The resultant acceleration of the box CG is computed from the net force F and mass m, and converted to units of G for plotting:

  - $a_G = ||F|| / (m * G)$
  - This is plotted as "CG acceleration (G) vs time."

- Corner spring loads:

  - For each of the four bottom corners, the vertical spring force at each accepted time step is stored. These four time histories are plotted together as "Corner spring loads vs time," with a distinct color for each corner.

- HUD metrics:

The heads-up display in the viewport shows several derived metrics, including:

  - Current time t
  - Minimum corner height z(min corner)
  - An approximate damping ratio ζ based on mass m, stiffness k, and damping c
  - Maximum estimated natural frequency f_{max}
  - Estimated drop time t_{drop}
  - Estimated end time t_{end}

- CSV and JSON export:

The module can export the recorded response data as either CSV or JSON. Each export contains a header and columns for:

  - t, aG, corner0, corner1, corner2, corner3

The JSON export also includes metadata describing the units and simulation parameters.

11. 3D Visualization

The box and ground plane are rendered using an orthographic projection driven by user-controlled azimuth, elevation, and zoom. A rotation matrix R_{view} is built from the azimuth and elevation angles. World coordinates are mapped to screen coordinates via:

- $p_{view} = R_{view} * p_{world}$
- $x_{screen} = cx + zoom * p_{view}.x$
- $y_{screen} = cy - zoom * p_{view}.y$

The ground is drawn as a shaded quadrilateral and a grid in the x–y plane. The box is drawn by projecting its eight corners and connecting edges. Bottom corner springs are visualized as stylized bar segments between corner contact points and the box body, with color indicating the corner index. A mini-axis widget shows the current view orientation.

An optional auto-fit mode adjusts the effective zoom when the box becomes too large in the current view to keep the entire body visible while preserving the user's base zoom setting as much as possible.

12. Initial Conditions

The initial CG position, orientation, and velocities are constructed from user inputs:

- The user specifies a drop height h and initial tilt angles θx and θy.
- A quaternion q is formed from the specified tilt angles.
- The box is positioned such that the lowest corner in the tilted configuration is located at height h above the ground. This is done by scanning all corners in world coordinates and shifting the CG so that the minimum corner z equals h.
- Initial translational and rotational velocities are set to zero.

13. Assumptions and Limitations

Key modeling assumptions include:

- The box is perfectly rigid; no internal flexibility is modeled.
- Springs and dampers are linear and act only in the vertical direction.
- The contact surface is a rigid, perfectly flat plane at z = 0.
- Contact is frictionless; tangential forces are not modeled.
- No plasticity, damage, or non-linear spring behavior is included.
- Air resistance and other external forces are neglected.

Within these assumptions, the model provides a detailed 3D rigid-body response of a tilted box impacting a ground-supported four-corner spring system, including translation, rotation, CG acceleration, and corner spring loads.

14. References (General)

The formulation used in this module is consistent with standard treatments of rigid-body dynamics and structural dynamics. Any of the following references may be consulted for background on the underlying theory:

- Chopra, A. K., “Dynamics of Structures: Theory and Applications to Earthquake Engineering.”
- Clough, R. W., and Penzien, J., “Dynamics of Structures.”
- Kane, T. R., and Levinson, D. A., “Dynamics: Theory and Applications.”