Beam Dropped Weight Response
User Guide and Solution Method Notes

HTML/JavaScript single-page simulation tool

Document date: January 26, 2026
Scope: user interface, solution approach, and references

Summary

This document describes the “Beam Dropped Weight Response” module. The tool lets a user define a segmented beam with distributed mass and stiffness, specify supports and springs at nodes, define a dropped weight with a drop height, and simulate the resulting transient response. Outputs include a time-history plot at the drop node and an animated deflected-shape visualization.

Quick start workflow

1. Set the number of segments (1–10), then click Build Beam.
2. Click on a segment to edit its properties (E, I, L, ρ, A).
3. Click on a node to edit boundary condition and node properties (BC, W, Kv, Kr, Drop H).
4. Ensure exactly one node has a positive Drop H and a positive W (this is the dropped weight location).
5. Set Gravity, Δt, total duration T, Impact mode, coefficient of restitution e (if Bounce), and damping ratio ζ. The Δt and total duration T are automatically calculated based on the 1st mode frequency but the user can input their own values.
6. Click Simulate Drop to run the transient analysis.
7. Use Play/Pause and Speed to view the animated response. Export CSV/JSON as needed.

User interface

The interface is organized into three toolbars followed by a primary beam canvas, a time-history plot, and an animation panel. Segment and node data are edited by clicking directly on the beam or node graphics; a modal dialog (SweetAlert2) is used for entry.

Toolbar controls

Control	Purpose	Notes / typical use
Segments (max 10)	Sets number of beam segments.	Click Build Beam to apply. Default segment properties are assigned on build.

Build Beam	Creates the segment and node arrays and redraws the beam.	Also triggers automatic Δt and T update based on the first mode estimate.
Gravity (G)	Gravitational acceleration used for unit conversion and drop kinematics.	Default 386 in/s² (US customary). The dropped mass is m = W/G.
Δt (s)	Time step for integration.	Auto-updated to approximately $T_1/500$ based on the first mode period T_1. User can override.
T (s)	Total simulation duration.	Auto-updated to approximately $4 \cdot T_1$. User can override.
Impact mode	Selects Stick (e=0) or Bounce (e>0) behavior.	Stick adds the dropped mass to the beam DOF after impact and applies static weight W.
e	Coefficient of restitution for Bounce.	Enabled only in Bounce mode. $0 \le e \le 1$.
ζ (%)	Modal damping ratio (applied equally to retained modes).	Entered as percent; internally converted to fraction.
Simulate Drop	Runs the analysis and refreshes plots/animation.	Requires one node with Drop H > 0 and W > 0, and that node must be vertically unconstrained.
Import JSON	Loads a previously exported model definition.	Updates toolbar inputs, segments, and nodes, then redraws the beam.
Export JSON	Saves the current model definition.	Useful for archiving or sharing scenarios.
Export CSV	Exports response history from the last run.	CSV contains time, drop-node displacement, and displacement at each original node.
Play / Pause	Controls the animation playback.	Animation includes an optional pre-drop “fall” sequence then the beam response.
Speed	Animation playback rate.	Adjusts frame increment per animation tick.

Beam canvas interaction

The primary canvas shows the beam as gray rectangular segments. Segment height is scaled by relative EI (E·I) to give a visual cue of stiffness distribution. Nodes are shown as blue circles; support and spring icons may be drawn at the nodes depending on settings. Clicking behavior is:

- Click a segment body to open the Segment Properties dialog (E, I, L, ρ, A).
- Click a node circle (large hit radius) to open the Node Properties dialog (BC, W, Kv, Kr, Drop H).
- Nodes are hit-tested first to avoid ambiguity when a node overlaps a segment.

Node graphics: pinned support is shown as a triangular symbol, fixed support as a thick vertical line with hatching. Vertical springs (Kv) and rotational springs (Kr) have stylized spring icons. If a node has a positive weight W, the node circle is drawn larger, and if Drop H > 0 the circle is lifted upward to depict an elevated drop mass prior to impact. Selected nonzero node parameters are also labeled near the node (W, H, Kv, Kr).

Plots and animation

Time plot: a Chart.js line plot displays the vertical deflection history at the drop node (w_drop). Animation: the second canvas shows the deflected shape (blue polyline) across a refined nodal grid, and the dropped mass is drawn as a black circle. In Bounce mode, the mass trajectory is shown relative to the beam motion using a simple ballistic model after impact.

Model definition and assumptions

The beam is modeled using an Euler–Bernoulli bending formulation with two degrees of freedom per (refined) node: vertical displacement w and rotation θ. Each user-defined segment is subdivided into multiple internal beam elements for solution fidelity (INTERNAL_NODES_PER_SEG = 10, i.e., 11 sub-elements per segment).

Segment properties

Parameter	Meaning	Used as
E	Young's modulus	Flexural rigidity EI
I	Second moment of area	Flexural rigidity EI
L	Segment length	Sub-element length Lsub = L/(m+1)
ρ	Material weight density (force/volume)	Mass density ρ/G (for m = ρ·A·L/G)
A	Cross-sectional area	Distributed mass per length μ = (ρ/G)·A

Node properties

Parameter	Meaning	Notes
BC	Boundary condition	Free, pinned (w=0), fixed (w=0 and θ=0). Constraints are enforced by DOF elimination.

W	Dropped weight (lbf)	Converted to mass m = W/G. Used only at the node with Drop H > 0.
Kv	Vertical spring stiffness	Added to the global stiffness at the vertical DOF of that node.
Kr	Rotational spring stiffness	Added to the global stiffness at the rotational DOF of that node.
Drop H	Drop height (in)	Positive value defines the drop node and sets impact speed v = sqrt(2·G·H).

Solution method

The solver uses a finite-element (FE) assembly for mass and stiffness, applies support constraints by removing constrained degrees of freedom, performs a modal reduction using a generalized eigenvalue solution, and then integrates the retained modal equations in time using a fixed-step fourth-order Runge–Kutta (RK4) method.

1) FE assembly of M and K

Each refined sub-element uses the standard 2-node Euler–Bernoulli beam element matrices (4×4) in local coordinates with DOFs [w1, θ1, w2, θ2]. The stiffness matrix is proportional to EI/L^3 and the consistent mass matrix is proportional to μ·L, where μ is the mass per unit length. Sub-element contributions are assembled into global matrices M and K. Node springs are incorporated by adding Kv to K(v,v) and Kr to K(θ,θ) at the corresponding DOFs.

Key implementation notes (as implemented in the code):

- Each user segment is subdivided into m+1 equal sub-elements (m = INTERNAL_NODES_PER_SEG).
- Distributed mass uses μ = (ρ/G)·A with consistent mass integration (the 420 denominator form).
- The drop mass is included only for Stick mode and only at the vertical DOF of the drop node.

2) Boundary conditions

Supports are enforced by DOF elimination. For a pinned node, the vertical displacement DOF is constrained (w=0) while rotation remains free. For a fixed node, both w and θ are constrained. After constraints are identified, reduced matrices M_red and K_red are formed by extracting rows/columns associated with free DOFs.

3) Modal decomposition and reduction

The generalized eigenproblem $K_{red}\,\phi = \omega^2 M_{red}\,\phi$ is solved by forming $A = M_{red}^{-1} K_{red}$ and computing its eigenpairs. Real positive eigenvalues are retained and converted to natural frequencies ω.

Each retained mode is mass-normalized so that $\phi^T M_{red} \phi = 1$. The solver keeps up to the lowest eight modes for time response calculations.

4) Impact modeling (Stick vs Bounce)

The dropped weight is represented as a lumped mass impacting the beam at the selected node. The impact speed is computed from free-fall kinematics: v_impact = sqrt(2·G·H). An impulse J is applied to the beam at the drop-node vertical DOF. An effective modal/structural mass m_eff at the impact DOF is estimated from the reduced mass matrix inverse via $m_{eff} = 1 / (e_i^T M_{red}^{-1} e_i)$, where e_i selects the impact DOF.

Impulse magnitude:

- Stick (e = 0): $J = (m_{eff} \cdot m_{drop} / (m_{eff} + m_{drop})) \cdot v_{impact}$.
- Bounce (e > 0): $J = (1 + e) \cdot (m_{eff} \cdot m_{drop} / (m_{eff} + m_{drop})) \cdot v_{impact}$.

In Stick mode, after applying the impulse, the dropped mass is added to the system mass matrix at the impact DOF and the constant weight W is applied as a static force at that DOF. In Bounce mode, the mass is not added to the beam; instead, the post-impact mass velocity is estimated using restitution-based two-body impact relations and the mass is animated with a simple ballistic trajectory under gravity.

5) Time integration (modal RK4)

Let y_j be the generalized coordinate for the j-th retained mass-normalized mode. The code integrates independent SDOF equations of the form:

$$\ddot{y}_j + 2\zeta\omega_j \dot{y}_j + \omega_j^2 y_j = f_j$$

where f_j is the modal force (constant for the Stick case due to applied weight) and ζ is the user-specified damping ratio applied uniformly to all retained modes. Initial conditions are generated by mapping the impact impulse to initial modal velocities ($\dot{y}_j(0)$) using the reduced-system velocity vector and $\phi^T M_{red} v$.

A fixed time step Δt is used. For convenience, the tool estimates the first natural frequency after each edit/build and sets $\Delta t \approx T_1/500$ and $T \approx 4 \cdot T_1$, where $T_1 = 1/f_1$.

Practical verification checks

The module is intended for engineering insight and rapid what-if studies. The following checks are recommended when setting up a new case:

- Units consistency: if using inches and pounds-force, set $G = 386\ in/s^2$ and use ρ as weight density (lbf/in^3).
- Constrained drop node: the drop node must be vertically free (not pinned or fixed) or the solver will stop with an error.
- Time step adequacy: reduce Δt if the time plot appears noisy or if peak response changes materially with Δt refinement.
- Mode truncation: for stiff or highly segmented models, consider reducing the number of segments or increasing damping to avoid numerical issues.

References

- Cook, R. D., Malkus, D. S., Plesha, M. E., and Witt, R. J., Concepts and Applications of Finite Element Analysis, 4th ed., Wiley, 2001.
- Bathe, K.-J., Finite Element Procedures, Prentice Hall, 1996.
- Timoshenko, S. P., Young, D. H., and Weaver, W., Vibration Problems in Engineering, Wiley, 5th ed., 1990.
- Chopra, A. K., Dynamics of Structures: Theory and Applications to Earthquake Engineering, Prentice Hall, 4th ed., 2012.
- Clough, R. W. and Penzien, J., Dynamics of Structures, McGraw-Hill, 2nd ed., 1993.
- Rao, S. S., Mechanical Vibrations, Pearson, 6th ed., 2017.
- Goldsmith, W., Impact: The Theory and Physical Behaviour of Colliding Solids, Dover, 2001 (impact and restitution modeling).
- Butcher, J. C., Numerical Methods for Ordinary Differential Equations, Wiley, 3rd ed., 2016 (Runge–Kutta methods).
- Chart.js project documentation (time-history plotting).
- SweetAlert2 project documentation (input dialogs).
- numeric.js library documentation (matrix operations and eigenvalue solution).

Appendix A: Exported file formats

JSON

Export JSON captures: toolbar inputs (segments count, gravity, Δt, T, impact settings), the segments array, and the nodes array. A timestamp and local date string are included for traceability.

CSV

Export CSV writes one header line with: time, w_drop, and w_node0 … w_nodeN. Each subsequent row contains the simulation time sample and the corresponding displacements. CSV is available only after a simulation has been run.