Eigenvalue and Nonlinear P–Δ Buckling Method

This document describes the numerical formulation implemented in the onlinear buckling module. The module computes the Euler buckling load by solving an eigenvalue problem on a refined beam-column mesh and then performs a geometrically nonlinear P–Δ analysis using an initial imperfection based on the first buckling mode.

1. Structural Model and Degrees of Freedom

The structure is a prismatic or stepped vertical column, discretized by user-defined primary segments. Each primary segment i has length L_i, elastic modulus E_i and second moment of area I_i. Primary nodes exist at the ends of these segments, and each primary node carries two Euler–Bernoulli degrees of freedom (DOF): lateral deflection v and cross-section rotation θ about the out-of-plane axis.

At each primary node, boundary conditions (free, simply supported/pinned, or fixed) can be specified. In addition, two types of nodal springs may be assigned:

- • kLat: a lateral translational spring in the v DOF.
- • kTor: a rotational (torsional) spring in the θ DOF.

2. Refined Beam–Column Mesh

For improved accuracy of the buckling and nonlinear solutions, each user-defined primary segment is internally subdivided into a fixed number of equal sub-elements. In the current implementation each primary segment is split into 6 sub-elements, creating 5 internal refined nodes in addition to the two primary end nodes.

If there are N_p primary nodes, the number of refined nodes is:

- $N_r = (N_p - 1) \times \text{splits} + 1$, with splits = 6.

Each refined node also has two DOFs [v, θ], so the total number of refined DOFs is $2N_r$. The refined nodes and elements are used only for analysis; they are not exposed to the user in the graphical interface.

3. Element Stiffness and Geometric Stiffness

The column is modeled as an Euler–Bernoulli beam-column. For each sub-element of length L, elastic modulus E and second moment of area I, the conventional 4×4 local bending stiffness matrix in the $[v_i, \theta_i, v_j, \theta_j]$ order is used:

$K^e = (EI / L^3) \times [[\ 12,\ \ 6L,\ -12,\ \ 6L\ ],\ [\ 6L, 4L^2,\ -6L, 2L^2\ ],\ [-12, -6L,\ \ 12, -6L\ ],\ [\ 6L, 2L^2,$ $-6L, 4L^2\ ]].$

Geometric stiffness due to an axial compressive load P is represented using a standard beam-column geometric stiffness matrix. In the implementation, the geometric stiffness for unit axial load (P = 1) is precomputed as:

$K^{ge}(\text{unit}) = (1 / (30L)) \times [[\ 36,\ \ 3L,\ -36,\ \ 3L\ ],\ [\ 3L,\ 4L^2,\ -3L, -L^2\ ],\ [-36,\ -3L,\ 36,\ -3L\ ],\ [\ 3L, -L^2,\ -3L, 4L^2\ ]]$.

For a given axial load level P, the element geometric stiffness is simply $P \times K^{ge}(\text{unit})$.

4. Global Assembly on the Refined Mesh

Global elastic stiffness K and unit-load geometric stiffness $K^g(\text{unit})$ are assembled over the refined mesh. For each sub-element, the local matrices K^e and $K^{ge}(\text{unit})$ are added into the corresponding rows and columns of K and $K^g(\text{unit})$ according to the refined DOF indices. The procedure is:

1. 1. Initialize K and $K^g(\text{unit})$ as zero matrices of size $(2N_r \times 2N_r)$.
2. 2. Loop over primary segments; for each segment, compute Lseg, E, I and its sub-element length Lsub.
3. 3. For each of the 6 sub-elements, build $K^e(E, I, Lsub)$ and $K^{ge}(\text{unit})(Lsub)$ and add them into K and $K^g(\text{unit})$.

Nodal springs are included after assembly by adding their stiffness directly to the appropriate diagonal entries of K at the refined DOF corresponding to the primary node:

- • kLat is added to the v DOF stiffness term.
- • kTor is added to the θ DOF stiffness term.

5. Boundary Conditions and DOF Reduction

Boundary conditions are specified only at primary nodes and then mapped to the refined mesh. A primary node p corresponds to refined node r = p × splits. The logic is:

- • Fixed node: both v and θ at refined node r are constrained and omitted from the free-DOF list.
- • Simply supported / pinned node: v is constrained, θ at r is free and kept in the free-DOF list.
- • Free node: both v and θ at r are treated as free DOFs.

Interior refined nodes (those not coincident with primary nodes) are considered free (both v and θ) because no boundary conditions are imposed there.

After determining the list of free DOF indices, reduced matrices K_r and $K^g{}_r(\text{unit})$ are formed by extracting the submatrices corresponding to the free DOFs. These reduced matrices are used in the eigenvalue and nonlinear solutions.

6. Linear Eigenvalue Buckling Analysis

The classical Euler-type buckling problem is formulated as a generalized eigenvalue problem based on the balance of elastic and geometric stiffness:

$$(K_r - P\, K^g{}_r(\text{unit}))\ \varphi = 0.$$

This can be rearranged into the numerical form:

$K_r^{-1} K^g{}_r(\text{unit})\ \varphi = (1 / P)\ \varphi$,

where φ is the buckling mode shape on the reduced DOF set and P is the critical axial load.

The implementation solves this problem by computing the matrix $M = K_r^{-1} K^g{}_r(\text{unit})$ using LU factorization and multiplying K_r^{-1} by each column of $K^g{}_r(\text{unit})$. A standard eigenvalue decomposition is then performed on M, returning eigenvalues λ and eigenvectors φ. For each positive finite λ, a candidate critical load is computed as P = 1 / λ.

The candidate eigenpairs are sorted by increasing P, and the first physically meaningful mode is selected by checking that its lateral displacement field has a non-trivial maximum magnitude. If all modes are nearly zero, the lowest P mode is used as a fallback.

The resulting reduced-mode vector φ (on free DOFs) is expanded back to the full refined DOFs using the free-DOF map, producing modeRef (length $2N_r$). A corresponding primary-node mode vector is also formed for post-processing by sampling the refined mode at the refined nodes that coincide with primary nodes.

7. Initial Geometric Imperfection

The nonlinear analysis uses an initial crookedness based on the first buckling eigenmode. The lateral displacement component of the refined eigenmode is scaled to achieve a target maximum amplitude proportional to the total column length L_{tot}:

- $a_0 = 0.0001 \times L_{tot}$ (imperfection amplitude ≈ L / 10,000).

If $\hat{v}$ is the lateral displacement field extracted from the eigenmode (at all refined nodes), and $\hat{v}_{max}$ is its maximum absolute value, the initial imperfection vector u_0 is defined as:

$u_0 = (a_0 / \hat{v}_{max}) \times \text{modeRef}$.

This vector u_0 (containing both v and θ DOFs) represents the initial deformed shape, defined on the full refined DOF set. A reduced version u_0^r is obtained by selecting the free DOFs, and this is used as the starting point for the nonlinear P–Δ iterations.

8. Incremental Nonlinear P–Δ Analysis

The nonlinear response is approximated by an incremental P–Δ procedure that updates the lateral equilibrium configuration under increasing axial load. The formulation is based on the tangent stiffness matrix:

$K_t(P) = K_r - P\ K^g{}_r(\text{unit})$.

Rather than solving a fully consistent nonlinear problem (with iterative updates within each load step), the algorithm applies a sequence of load increments and updates the displacement field in a pseudo-time marching fashion. The steps for each load increment are:

4. 1. Start with the current base shape u_0^r on the reduced DOFs (initially the imperfection shape).
5. 2. Select the current axial load P and compute the tangent stiffness $A(P) = K_r - P\, K^g{}_r(\text{unit})$.
6. 3. Form the right-hand side as $R = P\, K^g{}_r(\text{unit})\, u_0^r$. This approximates the geometric P–Δ forcing arising from the existing crookedness u_0^r.
7. 4. Solve $A(P)\, \Delta u^r = R$ using LU factorization to obtain the incremental displacement Δu^r.
8. 5. Expand Δu^r back to the full refined DOFs and add it to the base shape: $u_0 \leftarrow u_0 + \Delta u$.
9. 6. Update u_0^r by reducing u_0 to the free DOFs; this becomes the base shape for the next load step.
10. 7. Measure the maximum lateral deflection v_{max} over the refined nodes and store the pair (P, v_{max}) in a trace array used for plotting.

The global load level P is increased monotonically in fixed increments, starting from a fraction of the Euler load:

- • $P_0 = 0.05$ Pcr (start at 5% of the critical load).
- • $\Delta P = 0.01$ Pcr (use 1% of Pcr as a constant load step).

At each step, the code checks for numerical instability or divergence (e.g., failure in LU factorization or non-finite displacements). It also stops if either:

- • $P \geq$ Pcr (the nominal Euler load has been reached), or
- • v_{max} exceeds 0.5 L_{tot} (large-deflection cutoff).

The final nonlinear state stores the completed P–v_{max} trace, the last converged load level and the final refined displacement shape.

9. Visualization and Output

The graphical interface displays the current column configuration in the left canvas and a load–deflection chart in the right canvas.

• Column view: The column is drawn as a stack of segments with thickness scaled to √(EI) to indicate stiffness. Boundary conditions and springs are shown at the primary nodes. During eigen or nonlinear analyses, the lateral mode shape or total displacement shape is overlaid as a red polyline. The shape is normalized or scaled using a fixed visual gain so that deflections remain visible and within the canvas bounds. A small badge shows the current load P, the computed Euler critical load Pcr, and, when applicable, a theoretical Pcr estimate for uniform columns.

• Load–deflection chart: The right-hand canvas plots the trace of maximum lateral deflection versus axial load for the nonlinear P–Δ analysis. The horizontal axis is $|v_{max}|$, and the vertical axis is P. Axes, ticks and labels are drawn manually for clarity.

10. Summary

In summary, ColumnX combines a refined finite element discretization, a classical eigenvalue buckling solution and a simplified incremental geometric-nonlinear P–Δ scheme. The refined mesh improves mode shape resolution, while the eigenmode-based imperfection and tangent-stiffness updates provide a computationally efficient way to illustrate post-buckling trends and sensitivity to initial crookedness for prismatic or stepped beam-columns with arbitrary boundary conditions and springs.