

IST Vector



Resolution-independent postprocessing

Reference Manual — Issued June 2026

What this manual is

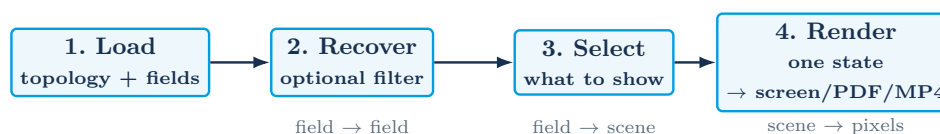
| | |
|-----------------------|--|
| Subject | The IST Vector Postprocessor: a desktop tool that turns finite-element solver output into figures, animations, and reproducible scenes. |
| Audience | Engineers and researchers who already know what nodal vs. element fields, stress recovery, and iso-surfaces are, and want a reference that respects that. |
| Organisation | §1 frames the pipeline; §2 covers data (file formats, fields, mesh tiers). §3 gives the recovery mathematics. §4 documents every representation card, a complete per-control reference (§4.13), animation and playback, symmetry, and the parts and field selectors. §5 covers the camera, tours, saved state, export, and the toolbars, projects, figure recipes, and viewport overlays. §6 is the element catalogue. §7 is the quick reference: shortcuts, mouse, performance, limitations. §8 lists references. |
| What it is not | A tutorial. Each section assumes the previous ones; recipes are dense, not narrated. |

1. The pipeline

Finite-element solver output is rarely visualisation-ready. Stresses are piecewise-constant on cells; displacements live on nodes; tensors arrive as nine loose scalars; every field carries discretisation noise. IST Vector exists to turn that output into figures and animations under three stances that the rest of the manual elaborates:

- **Recovery is upstream of every representation.** Colour, iso-surface, threshold, and deformation all read the active filtered field (§3 catalogues the filters; §4.6 is where the user picks one).
- **One scene, four outputs.** Screen, PDF, MP4, and the saved-view JSON consume the same scene object: what you see is what exports.
- **Resolution-independent throughout.** PDFs are vector-true Gouraud surfaces, not rasterised screenshots; MP4s capture the viewport's physical pixel grid; HiDPI is honoured end-to-end.

Everything in IST Vector is one pipeline. Read this section and the rest follows.



Why this matters. Recovery is upstream of selection: when you switch on **SPR**, the colour bar, the iso-surface, the deformation, *and* the threshold mask all consume the recovered field. You are not painting smoothed colours over raw geometry; you are visualising the recovered field consistently.

One state, four outputs. Screen rendering and the PDF / MP4 / saved-view exporters consume the same scene object. Saved views are JSON snapshots of that scene; a figure is reproducible from one file. Figure 1 shows the live application driving a representative scene.

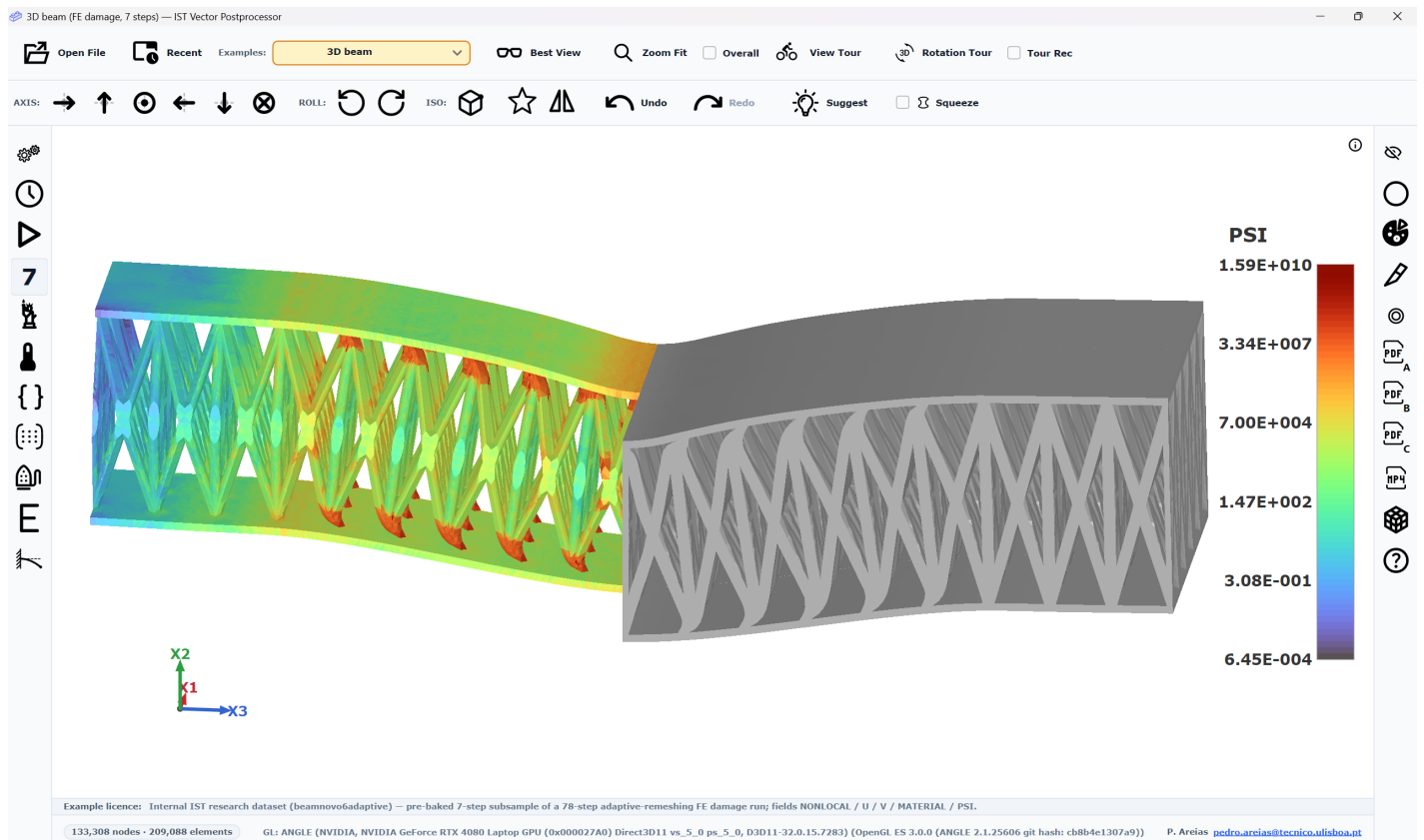


Figure 1. IST Vector Postprocessor on an adaptive-remeshing 3D beam (133,308 nodes, 209,088 elements, 5 fields, step 7 of 7). Left half: PSI scalar field in Turbo. Right half: symmetry-expanded mirror in neutral grey. Top toolbars carry file / view / tour / preset actions; left rail holds field selection (display, animation, geometry, scalar, vector, tensor, filter, error, deformation); right rail holds visibility and export (parts, probes, threshold, sections, iso-surface, PDFs, MP4, mesh quality, help). Colour bar, X1/X2/X3 triad, and bottom status pin to viewport positions that PDF / MP4 exports reproduce identically.

IST Vector



Loading data

File formats, fields, mesh tiers

2. Loading data

2.1 File formats

| Extension | Family | Carries |
|---|---------------------------------|---|
| .case + .geo / .scl / .vec / .ens | EnSight Gold | Topology, parts, transient steps, scalar / vector / tensor fields. ASCII and binary, multi-part, wildcard step files, BEGIN/END TIME STEP wrappers. |
| .vtu | VTK XML UnstructuredGrid | Topology + per-point / per-cell fields. Inline ASCII, base64 binary, and vtkZLibDataCompressor payloads. |
| .vtp | VTK XML PolyData | Verts, lines, polys, strips. Mapped to point / bar2 / tria3 / quad4 / fan-triangulated polygons. |
| .vtk | VTK legacy | ASCII or binary; reads UnstructuredGrid, PolyData, StructuredGrid, RectilinearGrid, StructuredPoints. |
| .pvd | VTK ParaView collection | Time-series index of .vtu / .vtp files; the loader honours timestep attributes. |
| .obj | Wavefront | Reference geometry only — no fields. Triangles and quads loaded directly; longer faces fan-triangulated. |

Open via drag-and-drop, `Ctrl+0`, or the **Tab Recent** flyout. The format is dispatched by extension. Loaded files are logged to the **Tab Recent** flyout (LRU of 20, unreadable entries auto-purged).

2.2 Fields and where they live

| Kind | Indexed by | Notes |
|--------------------------------|------------|---|
| Nodal scalar | node | Smooth shading; iso-lines and iso-surfaces work directly. |
| Element scalar | element | Flat shading (Per-face render is forced); iso-lines disabled — contour lines on piecewise-constant cells are meaningless. |
| Nodal vector | node | Drives arrows, streamlines, deformation, vector-norm derived scalars. |
| Nodal tensor | node | Symmetric 3×3 stored as the nine component scalars; drives ellipsoid glyphs. |
| Element vector / tensor | element | Same arithmetic as nodal; arrows are anchored at element centroids. |

Derived at load. IST Vector auto-builds, for every loaded vector field \mathbf{V} , the three components \mathbf{V}_1 , \mathbf{V}_2 , \mathbf{V}_3 , the Euclidean norm, and the uniform ℓ^∞ norm. For every tensor, the Frobenius and von Mises invariants, the hydrostatic pressure, the three column vectors, and (when a matching nodal/element scalar pair exists, found by longest common prefix) an **Absolute_E** error estimator are derived. This is what populates the field drop-down beyond the solver's literal output.

Force-derive. At the **Large** tier and above, derivation is skipped to keep memory bounded. Click the orange chevron next to the field combo to re-derive on demand.

2.3 Mesh size tiers

Memory and precomputation are gated by the larger of the node and element count:

| Tier | Limit (nodes/ elements) | Behaviour |
|-----------|-------------------------------|--|
| Standard | $\leq 500,000$ | Every derived field eagerly computed at load. |
| Optimized | $\leq 2,000,000$ | Derived fields still built eagerly at load (as in Standard); only the centroid cache is lazy. |
| Large | $\leq 10,000,000$ | Vector components, norms, tensor invariants <i>not</i> built; Force-derive chevron exposed. |
| VeryLarge | $\leq 50,000,000$ | Large-tier savings plus avoided jagged-node materialisation; status-bar warning chip “VERY LARGE”. |
| Huge | $> 50,000,000$ | Rejected at load with the user-readable reason. |

3. Field recovery

The recovery layer (**Field Filter** card, iron icon) replaces the active field with a recovered field *before* representation. When active, the colour bar’s range, iso-surface positions, threshold visibility, and deformation magnitudes all change in lockstep. The kernel radius defaults to $h = L/200$, where $L = \max(\Delta x, \Delta y, \Delta z)$ is the longest side of the mesh bounding box, and is lower-clamped at 10^{-9} .

Let $\Omega \subset \mathbb{R}^3$ be the mesh, $\{x_i\}_{i=1}^N$ its nodes, $f_i = f(x_i)$ the solver-supplied scalar samples. Let $\mathcal{N}_h(x) = \{i : \|x - x_i\| < h\}$ be the support set at x . We seek a recovered $\tilde{f}(x)$.

3.1 SPR — Zienkiewicz–Zhu superconvergent patch recovery

For each evaluation point x , fit a linear polynomial $p(\xi) = a_0 + a_1\xi_1 + a_2\xi_2 + a_3\xi_3$ in the local frame $\xi = \xi(x_i) - x$ by uniform-weight least squares over $\mathcal{N}_h(x)$:

$$\mathbf{a}^* = \arg \min_{\mathbf{a} \in \mathbb{R}^4} \sum_{i \in \mathcal{N}_h(x)} (\mathbf{P}(\xi_i)^\top \mathbf{a} - f_i)^2, \quad \tilde{f}(x) = a_0^*.$$

Normal equations: $(\mathbf{P}^\top \mathbf{P})\mathbf{a} = \mathbf{P}^\top \mathbf{f}$. Use SPR [1, 2] when stress / strain output is piecewise-constant per element and the eye reads the resulting checkerboard as noise; recovery returns the smooth field the solver was approximating.

3.2 Quadratic LS (Z^2)

Same recipe [1, 2] with the ten-term quadratic basis $\mathbf{P}(\xi) = (1, \xi_1, \xi_2, \xi_3, \xi_1^2, \xi_2^2, \xi_3^2, \xi_1\xi_2, \xi_2\xi_3, \xi_3\xi_1)^\top$. Reproduces a quadratic field *exactly*, so it is the recommended recovery for second-order fields (stress in linear-elasticity, plastic strain in J_2 flow). Costs $4\times$ the matrix solve of SPR.

3.3 MLS — Moving Least Squares with Wendland- C^2 kernel

The same least-squares fit, weighted [4]:

$$\mathbf{a}^*(x) = \arg \min_{\mathbf{a}} \sum_{i \in \mathcal{N}_h(x)} w\left(\frac{\|x - x_i\|}{h}\right) (\mathbf{P}(\xi_i)^\top \mathbf{a} - f_i)^2, \quad w(r) = (1 - r)_+^4 (4r + 1).$$

The Wendland- C^2 kernel [3] is compactly supported and twice continuously differentiable. Use MLS when SPR’s uniform weighting picks up too much from the patch boundary and the recovered field shows ringing.

3.4 Taubin $\lambda | \mu$ smoothing

Discrete graph-Laplacian [5] operating on node values \mathbf{f} :

$$\mathbf{f}^{(k+1/2)} = (\mathbf{I} + \lambda \mathbf{L})\mathbf{f}^{(k)}, \quad \mathbf{f}^{(k+1)} = (\mathbf{I} - \mu \mathbf{L})\mathbf{f}^{(k+1/2)}, \quad 0 < \lambda < \mu,$$

with defaults $\lambda = 0.50$ and $\mu = 0.53$. The alternating positive / negative pass cancels low-frequency shrinkage so a long smoothing run does not collapse the field’s mean. Use for noisy displacement-like fields where SPR’s least-squares basis would force a polynomial fit that the data doesn’t support.

3.5 Perona–Malik anisotropic diffusion

The Perona–Malik flow [6],

$$\partial_t f = \operatorname{div}(g(\|\nabla f\|^2) \nabla f), \quad g(s) = \exp(-s/K^2),$$

is discretised explicitly on the mesh's node graph over ten passes at a fixed sub-unit step ($\Delta t = 0.15$), the conductance scale K re-estimated each pass from the local gradient contrast. The flux limiter g attenuates diffusion where the gradient is large, so material boundaries and shock fronts are preserved. Use when the field has sharp transitions you want to keep (phase boundaries, fracture lines, material IDs).

3.6 Choosing a recovery

| Solver output | Recommended | Why |
|--|--------------|--|
| Per-element stress / strain | Z^2 | Reproduces quadratic fields exactly. |
| Noisy nodal scalar (rough mesh) | SPR | Cheap and smooths cleanly when no sharp feature is present. |
| Smoothed-out scalar over-blurred by SPR | MLS | Wendland weighting respects the kernel rather than the bounding patch. |
| Phase / damage / multi-material with sharp jumps | Perona–Malik | Diffusion limiter preserves the edges. |
| Noisy displacement / nodal vector | Taubin | No mean drift. |

Activate any of these via the **Field Filter** card (§4.6); the chosen filter then governs every downstream representation that consumes the field.

Implementation notes. All filters share one uniform spatial grid (cell size $\max(h, L/64)$ with L as above) so neighbour lookup is per-point $O(K)$, not $O(N)$. Vector fields are filtered component-wise through the same traversal. Per-element fields are sampled at cached centroids. NaN and $\pm\infty$ are skipped throughout so a single bad sample cannot poison a colour range or a visibility mask.

IST Vector



Representations

What each card does, and when

4. Representations

The left rail offers field selection cards; the right rail offers visibility and export cards. Pick a card to open its flyout. The scalar, vector, and tensor overlays are mutually exclusive: selecting a field in any one card resets the other two to **None**. The bare-mesh button (statue icon) clears all three at once to show the unadorned geometry; the active deformation is preserved so animation playback can continue.

4.1 Display options (left rail, gears)

Global rendering modifiers that apply across the representation cards, in three groups — **Surface**, **Annotations**, **PDF export**. Lighting is always on; the iso-surface renderer is the lone exception, forcing it off so its steel-blue tint reads against the white page.

| Toggle | Effect |
|---------------------------|--|
| Mesh lines | Every element edge as a thin wireframe over the surface. Mode-aware (Plain / Contours); disabled in Iso-lines mode, which draws no element edges. |
| Edges | Feature edges only (sharp creases, part boundaries). In Plain mode mutually exclusive with mesh lines; in Contours / Iso-lines it overlays the scalar independently. |
| Per-face | Flat per-facet shading: one averaged colour per boundary face (no Gouraud gradient), arrows anchored at centroids, lighting from the face normal. The surface reads as a faceted polyhedron. |
| Polished gray | Render the bare mesh and the symmetry silhouette with chrome / polished-steel shading (neutral grey, tight specular, near-pure metalness) instead of the default matte head-light grey. |
| Dimensions | Bounding-box length / width / height callouts and a scale bar. Plain mode only; shares the mesh-lines mutex. |
| See nodes | A small circle (0.2% of screen width) at every visible node. |
| See node / element labels | Index labels at the nodes / at element centroids. |

Ultra fine pdf. An export-only quality switch: it leaves the live viewport and the video recorder untouched and upgrades only the still PDF. When set, the PDF exporters route through a high-fidelity glossy path — a boosted, broad specular sheen plus a per-sub-vertex re-shaded subdivision of the lit contour / iso faces, so highlights stay tight and colour gradients stay smooth at any zoom (§5.4 gives the shading detail). Always defaults off and is never persisted: it is a per-session choice that will not silently return on a saved view or project.

4.2 Scalar field (left rail, thermometer)

Colour-shade the surface by a chosen scalar. In **Iso-lines** mode, draw N evenly spaced level lines ($3 \leq N \leq 30$) instead, with optional in-figure labels and a **B/W** toggle for grayscale print figures.

Label placement. A greedy multi-criteria pass. Candidate anchors are emitted at five arc-length fractions per polyline, scored on four criteria:

- *Projected depth* — front-most first.
- *Tangent length* — avoid edge-on labels.
- *Local curvature* — prefer straight stretches.
- *Polyline length* — long lines beat fragments.

An axis-aligned bounding-box overlap test then suppresses collisions between accepted labels.

Number format. Iso-line labels, colour-bar ticks, and the live probe readout use scientific notation with three mantissa decimals (e.g. $1.250E1$); magnitudes in $[1, 10)$ show no exponent.

Inline labels and B/W. In the PDF and video export, every inline value label now reads in a clean gap broken into the contour line — topographic-map style, aligned with the line — rather than over a drawn white box: the scene builder clips each contour segment against the rotated footprint of every placed label, and the exporters stamp the rotated text into the gap. (The live viewport, whose iso-lines are GL ribbons painted independently of the 2-D label pass, is unchanged.) Selecting **B/W** draws every line in pure black, forces the labels on, and drops the colour bar; with the bar gone its title would take the quantity name with it, so the plotted field's name is instead shown as a caption centred below the part — on screen *and* in the export — and the colour-free figure still states what it plots.

Logarithmic colour mapping is available for strictly positive fields: $f \mapsto \log_{10} f$ is applied before the colour-bar normalisation so a field spanning several decades does not collapse into the colour map's brightest band. Non-positive samples are clamped to the lower decade ($t = 0$).

| | |
|---------------------------------|--|
| Recipe. Colour by scalar | Click thermometer → pick field → optional Min/Max (blank = auto, re-scans each step). In Iso-lines mode adjust the count slider (3–30) and the Labels / B/W toggles. |
|---------------------------------|--|

4.3 Vector field (left rail, curly braces)

Arrows, streamlines, pathlines, or streaklines. Streamlines integrate the steady field at the current step; pathlines integrate forward and backward across time; streaklines connect particles released continuously over a time window. Integration uses the Dormand–Prince DOP853 [10] step controller with adaptive 8(5,3) error estimation; sub-step sampling along each accepted segment is by linear (chord) interpolation between the segment endpoints, with arc-length and iteration caps and a configurable convergence tolerance.

| | |
|---------------------------------------|--|
| Recipe. Stream / path / streak | Vector card → pick field → choose representation (Arrows / Streamlines / Pathlines / Streaklines). Adjust Line length , Sparsity , and Proportional length . Enable Color by speed for Turbo-shaded ribbons. |
|---------------------------------------|--|

4.4 Tensor field (left rail, matrix)

Principal-axis ellipsoid glyphs. The nine tensor components are grouped into a symmetric 3×3 matrix at each node, symmetrised, and decomposed by Jacobi. Scale controls glyph size as a fraction of mesh diameter; sparsity controls glyph density on a uniform grid.

4.5 Deformed shape (left rail, cantilever beam)

Displace nodes by a chosen vector field multiplied by a magnification factor. Mesh quality is recomputable on the deformed configuration so the user can see how aspect ratios degrade as the displacement grows. The **Undeformed outline** toggle overlays the outer / feature edges of the original (undeformed) geometry as thin grey lines fixed at the reference shape — emitted at the reference positions with no magnification term, so they stay put while the deformed fill and wireframe move with the slider — making the displacement legible against where the mesh started.

4.6 Field filter (left rail, iron)

The recovery card. Pick a method (SPR, MLS, Taubin, Perona–Malik, Z^2 — §3 catalogues the mathematics), set the kernel radius h if the default is not appropriate, and the active field is replaced by its recovered version before any representation reads it. All downstream cards — scalar colouring, iso-surface, threshold, deformation, error estimator — consume the recovered field, so a single switch on this card reshapes the whole scene consistently.

4.7 Error estimator (left rail, letter E)

Surfaces the nodal / element pairs IST Vector detected at load (paired by longest common prefix of the field names). Pick one and a projection direction (per-node or per-element) to materialise an **Absolute_E** scalar that captures $|f_{\text{nodal}} - \text{proj}_{\text{elem} \rightarrow \text{node}}(f_{\text{elem}})|$. Reads as the FE practitioner's a-posteriori error indicator on the live mesh.

4.8 Threshold — element erosion (right rail, pizza)

Hide elements whose scalar value satisfies a relation. Per-node scalars are first reduced to a per-element value by mean / max / min, then compared with $>$, $<$, $=$, or \neq . Equality uses a $\pm 0.5\%$ -of-range tolerance for continuous fields and exact matching for integer-valued fields (material IDs).

Filter interaction. Like every representation in §4, the threshold consumes the *filtered* field selected on the Field Filter card (§4.6). Switching from raw output to SPR or Z^2 can move element-mean values across the cut-off and re-populate the visible set; this is the intended behaviour.

Recipe. Show only highly damaged elements

Threshold → pick FRACN → Reduction **max** → Hide where value < 0.6 → slider scrubs through the field's range live.

4.9 Section planes (right rail, slice)

X / Y / Z cuts at normalised positions on the mesh bounding box, each with a flip toggle. Composed multiplicatively with the part visibility mask and the threshold mask in one element-visibility pass.

4.10 Iso-surface (right rail, concentric)

Extracts a level-set surface from a nodal scalar. Volume cells are decomposed to tetrahedra and marched in 3-D; surface cells use marching triangles in 2-D. The extractor's elementwise early-exit (*range brackets iso*) skips most cells before the marching kernel runs. A slider scrubs the iso-value through the field's [min, max] and the surface re-extracts live; with the count set to $N > 1$ an **End value** gives N equispaced surfaces in one pass. **Keep part outline** overlays the original mesh silhouette so you can see where the surface sits inside the part. **B/W mode** renders the surface in grey and drops the colour bar; as for B/W iso-lines (§4.2) the scalar's name then captions the figure below the part. The card is hidden for 2-D problems (iso-surfaces need a volumetric mesh) and disabled until a nodal scalar is loaded.

Filter interaction. Iso-surfaces consume the *filtered* field. A smoother SPR can turn a broken iso into a coherent level set; Perona–Malik preserves sharp fronts; switching filters mid-stream causes the iso to migrate.

4.11 Probes (right rail, circle / quad)

Node probe: Ctrl+click reports the picked node's coordinates plus every active field value at that node. **Element probe:** Ctrl+Shift+click reports the picked element's index, type and node count, centroid, and the active scalar (per-element value, or the node-average for a nodal field) plus the active vector's average magnitude over the element. Esc clears the probe.

4.12 Mesh quality (right rail, cube)

Per-element-type counts, edge-length statistics, aspect-ratio extrema, signed area / volume distributions, and inverted-element counts. Computed against the as-loaded mesh, or against the deformed configuration when a deformation field is active.

4.13 Card control reference

Every card's flyout exposes the controls below. Slider ranges are the raw control limits; sliders that scrub a physical quantity (threshold, section, iso) map $[0, 1]$ onto the active field's or the bounding box's range and re-scan it each step. Blank text fields mean *auto*.

| Card / control | Type | Range / default | Effect |
|-------------------------------------|---|---------------------|---|
| Scalar Field | combo | -- | Pick the scalar to colour by; resets vector / tensor to None. |
| Min / Max | text ×2 | blank = auto | Fix the colour range; blank re-scans the field each step. |
| Mode | Plain / Contours / Iso-lines | Plain | Solid shade, banded contours, or level lines. |
| Iso-line count | slider | 3-30 / 10 | Number of evenly spaced level lines (Iso-lines mode). |
| Labels, B/W | toggle ×2 | off, off | In-figure value labels; pure-black print with field-name caption. |
| Invert scale | toggle | off | Reverse the Turbo LUT (low = red, high = blue). |
| Log scale | toggle | off | \log_{10} mapping; enabled only for strictly-positive fields. |
| Vector Field | combo | -- | Pick the vector; resets scalar / tensor. |
| Representation | Arrows / Stream / Path / Streak | Arrows | Glyphs, steady streamlines, or time-integrated path / streaklines. |
| Line length | slider | 0-1 / 0.40 | Glyph / integral-curve length as a fraction of model size. |
| Sparsity | slider | 0-1 / 0.50 | Seed density on the uniform grid (higher = sparser). |
| Proportional length | slider | -1-1 / 0 | Scale length by magnitude; 0 = uniform, \pm biases short / long. |
| Colour by speed | toggle | off | Turbo-shade each segment by local $\ v\ $. |
| Tensor Field | combo | -- | Pick the tensor; resets scalar / vector. |
| Scale | slider | 0-5 / 1 | Ellipsoid glyph size × mesh diameter. |
| Sparsity | slider | 0-1 / 0.50 | Glyph density. |
| Trajectory length | slider | 0.02-1 / 0.60 | Principal-line length (trajectory sub-mode). |
| Even spacing | slider | 0.02-0.5 / 0.10 | Target line separation when Evenly spaced is on. |
| Tube radius | slider | 0.004-0.05 / 0.012 | Line-tube radius when Tubes (3D) is on. |
| Deformed Field | combo | -- | Displacement vector to warp by; coexists with a colour field. |
| Magnification | slider | 0-20 / 1 | Displacement multiplier; scrub is $O(1)$ (shader-side). |
| Undeformed outline | toggle | off | Overlay the reference-shape outer / feature edges as thin grey lines (only when magnification $\neq 0$). |
| Field ter Method | fil- SPR / MLS / Taubin / PM / Z^2 | off | Recovery applied before every representation (§3). |
| Kernel radius h | slider | 0.0005-0.05 / 0.005 | Support radius; default $L/200$. |

Right-rail visibility cards:

| Card / control | Type | Range / default | Effect |
|-------------------------------|------------------|-----------------|---|
| Threshold Field | combo | -- | Field driving element erosion. |
| Reduction | mean / max / min | mean | Node→element collapse for nodal fields. |
| Relation | > / < / = / ≠ | > | Comparison against the cut-off. |
| Value | slider | 0-1 / 0.50 | Cut-off, mapped onto the field range; scrubs live. |
| Sections X / Y / Z | slider ×3 | 0-1 / 0.50 | Cut position on the bounding box, per axis. |
| Flip X / Y / Z | toggle ×3 | off | Keep the other half-space. |
| Iso-surface Value | slider | 0-1 / 0.50 | Level, mapped onto [min, max]; re-extracts live. |
| Count | slider | 1-10 / 1 | Number of nested surfaces. |
| End value | slider | 0-1 / 0.75 | Upper level when count > 1 (equispaced set). |
| Keep part outline, B/W | toggle ×2 | off, off | Overlay the part silhouette; grey surface + name caption. |

4.14 Animation and playback (time card)

The time card (clock icon, left rail) is enabled once a transient with two or more steps loads; on a single-step still it and the step badge stay hidden. On load the range is the full span and playback parks on the last step.

| Control | Range / default | Effect |
|-------------------------------|--------------------------------------|--|
| Step slider | 0-(N-1) | Scrubs the displayed step; hand-driven in Manual mode, read-only (timer-driven) in One-cycle / Cyclic . |
| From / To | combos, full span | Restrict playback and MP4 export to a sub-range; each clamps the other so From ≤ To. |
| Play mode | Manual / One-cycle / Cyclic (Manual) | Manual = hand scrub; One-cycle plays From→To once, lands exactly on To, then any postlude; Cyclic loops, wrapping at To. |
| Play / Pause | Space | Toggle playback of the current mode. Glyphs use Geometric-Shapes codepoints, not the media-control block (which renders as tofu on the bundled Linux fonts). |
| Play once (helix icon) | — | One From→To sweep with the card's settings; the red triangle becomes a spinning helix while running; click again aborts. |
| Duration | combo, 10 s | Target seconds for a full sweep — also the MP4 length. |
| Step badge | 0-based frame | The declared source-file step number of the displayed frame (falls back to the frame index); the digit auto-scales to the box. |
| Clock readout | — | Elapsed / total playback seconds. |
| Prelude / Postlude | toggle ×2 | Cinematic ease-in / ease-out envelopes bracketing an exported sweep. |
| Cinematic | toggle | Route the MP4 through the five-stage post pipeline (§5.4). |
| Tour Rec | toggle, off | Record the next View / Rotation Tour as a 1:1 MP4 beside the application. |

Live playback is not a step-snap: it interpolates between integer steps at 60 Hz — Catmull–Rom for same-topology transients, a two-FBO cross-fade for adaptive-remesh transients (§7.2 details both).

4.15 Symmetry, parts, and the field selector

Symmetry / reflections (toolbar, mirror icon). Mirror the loaded part across the X, Y, and/or Z planes and/or generate N rotational copies about an axis, to reconstruct a full body from a modelled sector. Rotational count runs 1–16 (default 1, i.e. none); a custom-axis panel accepts an arbitrary rotation axis. Expansion is capped at 100 million total expanded nodes (§7.4); above the cap it is refused with a status-bar message. Colour fields are carried onto the reflected copies; the mirror silhouette renders in neutral grey (or polished steel, §4.1).

Parts and materials (right rail). **Parts** lists every geometric part with a show/hide checkbox and all/none actions; **Materials** does the same by material ID. The part mask is composed multiplicatively with the section-plane and threshold masks in one element-visibility pass, and is captured in a saved view / project so a figure re-opens with the same parts hidden. Both cards are disabled until a mesh with named parts loads.

Field selector. Every field card shares one drop-down populated with the solver’s fields plus everything derived at load (§2.2): for a vector \mathbf{V} , the components $\mathbf{V}_{1/2/3}$ and the ℓ^2 / ℓ^∞ norms; for a tensor, the Frobenius and von Mises invariants, the hydrostatic pressure, the nine components, and the three column vectors. At the **Large** tier and above these are not pre-built — the orange **Force-derive** chevron beside the combo populates them on demand.

IST Vector



Camera and reproducibility

Best View, tours, saved state

5. Camera and reproducibility

5.1 Best View — viewpoint entropy

Best View picks the camera direction that maximises Vázquez viewpoint entropy [7] over a Fibonacci-spiral sphere sample of 192 candidate directions:

$$H(\mathbf{v}) = - \sum_{i \in F(\mathbf{v})} \frac{A_i(\mathbf{v})}{A_{\text{tot}}(\mathbf{v})} \ln \frac{A_i(\mathbf{v})}{A_{\text{tot}}(\mathbf{v})},$$

where $F(\mathbf{v})$ is the set of mesh faces front-facing and unoccluded from \mathbf{v} , $A_i(\mathbf{v})$ is the projected area of face $i \in F(\mathbf{v})$, and $A_{\text{tot}}(\mathbf{v}) = \sum_{i \in F(\mathbf{v})} A_i(\mathbf{v})$. Faces are area-weighted, so a view that exposes many distinct faces of similar projected area scores higher than one looking straight at a single dominant face. The score is augmented by three small tie-breakers — total projected area, projected-area spread, and projected aspect ratio (weights 0.07, 0.30, 0.40) — to separate near-ties cleanly.

Overall framing (the checkbox next to **Best View**) does *not* change how the direction is scored: the direction is still chosen on the current step's mesh. What it changes is the zoom-to-fit bounding box, which is widened to the union of mesh bounding boxes across all playback steps so the same framing reads cleanly throughout an adaptive-remesh playback.

5.2 View Tour and Rotation Tour

View Tour interpolates a waypoint sequence built from the current camera and the Suggest-card Best-View results: origin \rightarrow wide A \rightarrow detail A \rightarrow (optional scalar-peak stop, when a hot spot is detectable) \rightarrow detail B \rightarrow wide B \rightarrow origin, so the tour both starts and ends at the live camera.

- Rotation: Shoemake *squad* [8] — spherical-cubic, C^1 -continuous across waypoints, with intermediate quaternions chosen so the angular velocity is smooth at every stop. (Earlier versions used piecewise SLERP; *squad* replaced it to eliminate the visible jerk at each stop.)
- Translation and ortho-zoom: C^1 -continuous Catmull–Rom splines [9].
- Segment time runs at constant *visual* speed: the path is arc-length reparameterised over a perceptual metric (pan in screen-heights, orbit in radians, zoom in log-octaves) and traversed at a fixed cruise rate, so the total duration is the perceptual arc length over that rate, clamped to 8–22s. Waypoint importance no longer stretches segment time; it survives only to break ties when near-coincident stops are merged.

Rotation Tour is a one-shot 360° spin about the current up axis under an underdamped-spring profile with prelude and outro envelopes.

5.3 Saved views and persistence

A saved view is a JSON snapshot of: window geometry, display mode, every active field, range overrides, deformation, filter mode + length, iso state, symmetry config, hidden parts, probes, time state, and camera (3×3 rotation + ortho half-height + 3-D pan). Re-loading reproduces the figure.

| State | File / location |
|-------------------|---|
| Saved views | %AppData%\IST-VectorPostprocessor\saved_views.json |
| Per-part sessions | %AppData%\IST-VectorPostprocessor\part_sessions.json (single keyed store covering every loaded part) |
| Recent files | %AppData%\IST-VectorPostprocessor\recent_files.json |
| View preferences | Per-part Best-View correction store, kept in %AppData%\IST-VectorPostprocessor\view_preferences.json. |
| Export folders | Last folder used per export type (PDF, MP4, geometry), kept in %AppData%\IST-VectorPostprocessor\export_folders.json. |
| Diagnostic log | %LocalAppData%\IST-VectorPostprocessor\logs\ |

Note on Windows roots. Saved views, sessions, and the recent-files index live under the roaming profile (%AppData%) so they follow the user across machines. The diagnostic log lives under the local profile (%LocalAppData%) because it is per-machine and may be large.

5.4 Export

PDF. Vector output. Surfaces emit as PDF Type-4 Gouraud-shaded triangles; contour and feature lines emit as ordinary PDF stroked polylines (anti-aliasing is delegated to the PDF viewer’s rasteriser, which is resolution-independent by construction); the colour bar, triad, contour labels, and dimensions print at their on-screen relative positions. Text is set in base-14 Helvetica-Bold (referenced by name, not embedded; WinAnsiEncoding), with a ToUnicode CMap for copy/paste and Latin-1 transliteration of scientific symbols. **Squeeze** mode produces a narrower right-anchored page that mirrors the live overlay layout pixel-for-pixel. When the loaded mesh carries an upstream licence notice (Stanford 3D Scanning Repository, Berlin Egyptian Museum scan, OCCT reference STL, etc.), the same string that appears in the live message bar is emitted as a small grey credit line below the figure — the PDF preserves the attribution the upstream dataset requires.

Ultra-fine PDF. The **Ultra fine pdf** toggle (§4.1) upgrades the still export to a gallery-grade glossy figure; two changes stack. *Gloss boost* — the lit shading runs at a stronger, broader specular (2.5× strength, GGX roughness ≈ 0.22) under a high metalness (0.93) so the sheen inherits the surface’s contour colour (a red region gleams brighter red) instead of bleaching toward white. *Re-shaded subdivision* — each lit contour / iso fan-triangle is barycentrically subdivided into N^2 sub-triangles (N adaptive: 6 on small meshes down to 2 above $\sim 400k$ faces, to keep the Type-4 mesh’s triangle budget bounded) and the full Lambert + GGX/Fresnel + metalness response is re-evaluated at every sub-vertex, with the Turbo LUT re-sampled per node, so tight hotspots and colour gradients survive at sub-element resolution instead of washing out under one-triangle-per-face Gouraud fill. Sub-triangle corners stay byte-identical to the standard bake, so refined and coarse faces meet without a seam. Ultra-fine forces per-face lumping off regardless of mesh size; the document is larger and the write takes longer. The live viewport and the MP4 recorder are unaffected — both keep the calibrated, screen-matching matte.

MP4. FFmpeg (libx264) preferred; Windows Media Foundation is the silent fallback. Frames captured at the live viewport’s physical pixel size (HiDPI-aware), trimmed to even dimensions for H.264 compliance, piped through an ArrayPool-backed producer. The same **camForT** closure that drives the live View / Rotation Tour drives the offline recorder, so the MP4 reproduces the live motion frame-identically. Both encoders tag the output stream as BT.709 (primaries, transfer, colour matrix, limited tv range on the encoded YUV) so QuickTime, Windows Media Player, and hardware decoders interpret the colour space deterministically; without this tag, sub-720p material is silently re-decoded as BT.601 and the rendered gamut shifts.

Cinematic post-pipeline. An optional five-stage per-frame pass: FXAA edge anti-aliasing, soft-knee bloom, Reinhard tone-mapping, radial vignette, film grain. Pure-white background pixels are preserved so the colour bar, triad, and labels stay crisp. Per-frame cost on 1920×1080 : 30–80 ms, dominated by the bloom Gaussian.

5.5 The toolbars, projects, and figure recipes

Two chrome rows sit above the viewport. All controls grey out in **Squeeze** mode except the squeeze toggle itself.

| Button | Action | Key |
|---------------------------|---|--------------------|
| Open File | Open a mesh or <code>.spp</code> project; the format is dispatched by extension. | Ctrl+O |
| Recent | Re-open one of the 20 most-recently loaded files; unreadable entries self-purge. | -- |
| Save Project | Write the mesh reference plus the full view state to a portable <code>.spp</code> . | -- |
| Examples | Load a bundled example dataset; (none) clears the viewport. | -- |
| Best View | Frame from the viewpoint-entropy-optimal angle (§5.1). | F |
| Zoom Fit | Re-fit zoom and pan; orientation unchanged. | -- |
| Overall | Frame every step at once so scrubbing needs no re-fit (default <i>on</i>). | -- |
| View Tour / Rotation Tour | Turntable spin / single 360° orbit (§5.2); click again to cancel. | -- |
| Tour Rec | Record the next tour as a 1:1 MP4. | -- |
| Axis / Roll / Iso | Preset camera orientations (front / side / top, roll steps, isometric). | -- |
| Saved Views | Save, name, apply, and manage named view snapshots. | Ctrl+S |
| Symmetry | Mirror / rotational reflections (§4.15). | -- |
| Undo / Redo | Step through the <i>display-state</i> history (not the file load). | Ctrl+Z / Ctrl+Y |
| Suggest Views | Best-View candidate thumbnails to seed a tour. | -- |

Three tiers of reproducibility. A **Saved view** is a named entry in the in-app library (§5.3); it lives in the roaming profile and applies to whatever mesh is loaded. A **project** (`.spp`) additionally binds the mesh reference, so re-opening one file restores figure *and* data. A **figure recipe** (`.json`, **Export recipe** / **Import recipe** on the Saved Views card) is the portable middle ground: camera plus every display setting, applied to the currently-loaded data with any absent field selection skipped — the format for shipping a figure specification next to a paper.

5.6 Overlays, colour bar, and status

The viewport paints several always-on or conditional overlays; all of them reproduce in the PDF / MP4 export at their on-screen relative positions.

| | Content and when shown |
|------------------------------|---|
| Overlay | |
| Colour bar | Right-edge vertical Turbo gradient with scientific-format ticks and the field name as title; hot at top (flipped by Invert scale). Suppressed for geometry-only views and in B/W iso-lines mode; pinned to the field's global range when the range is locked across frames. |
| Axis triad | Bottom-left X1/X2/X3 arrows (red / green / blue) rotating with the camera. |
| Status bar | Three cells: an amber <i>mesh-tier</i> chip (only above the Standard threshold, tooltip naming the skipped derivations); the centre <i>message</i> line (newest-first, two lines); and the author mailto on the right. |
| Example-licence strip | A grey credit line above the status bar for datasets that carry an upstream licence (Stanford, OCCT, museum scans); the same string prints under the exported figure. |
| Step badge | Square box below Play once showing the displayed frame's declared step number; hidden on single-step data. |
| Video-export ETA | Bottom-right panel shown only while an MP4 renders: frame counter, progress bar, predicted finish, and time remaining, with a Cancel button when the encoder supports it. |
| Busy modal | Full-window dim overlay with a progress ring during long operations (large loads, PDF / collage / video, tour recording); swallows input to block parallel work. |
| Startup splash | Ten-second launch card (auto-dismiss, or close early) carrying the title, author, and mailto; an amber advisory appears when the machine reports under 15 GB of available RAM. |
| Empty-state hint | Centred prompt shown with no mesh loaded, naming the accepted formats and the Ctrl+0 / F1 entry points. |

Discoverable help. A small **i** button at the viewport's top-right reveals a mouse-controls cheat sheet on hover (120 ms); **F1** opens the modal keyboard-shortcut list (§7.1). The two toolbars and the icon rails flush Hasselblad orange on hover and reveal their tooltip only after the pointer settles for 2 s, so the colour is the instant feedback and the text never flickers under a moving pointer.

IST Vector



Element types

The eight corner-only internal types

6. Element types

IST Vector reads the EnSight Gold / Case element vocabulary plus the matching VTK family. Internally, every parsed cell collapses to one of eight *corner-only* types — one per topological-dimension pair. Quadratic mid-edge and mid-face nodes are read past but discarded; rendering and field interpolation use the corner connectivity throughout.

6.1 Catalogue

| EnSight name | Nodes | Dim | Geometry | Quadratic | Notes |
|--------------|-------|-----|----------------------|----------------|---|
| point | 1 | 0 | Vertex | -- | Anti-aliased sprite. VTK / VTP loaders accept it; Case parser drops it. |
| bar2 | 2 | 1 | Line segment | bar3 | Wu-antialiased line. Same parser asymmetry as <code>point</code> . |
| tria3 | 3 | 2 | Linear triangle | tria6 | – |
| quad4 | 4 | 2 | Linear quadrilateral | quad8 / quad9r | – |
| tetra4 | 4 | 3 | Tetrahedron | tetra10 | – |
| pyramid5 | 5 | 3 | Square-based pyramid | pyramid13 | – |
| penta6 | 6 | 3 | Triangular prism | penta15 | Aliases <code>prism6</code> , <code>wedge6</code> . |
| hexa8 | 8 | 3 | Hexahedron / brick | hexa20 | – |
| nsided | var | 2 | General polygon | -- | Read, not rendered. |
| nfaced | var | 3 | General polyhedron | -- | Read, not rendered. |
| block | I·J·K | 3 | Structured grid | -- | Unrolled to <code>hexa8</code> (or <code>quad4</code> for slabs). |

Keyword matching is case-insensitive. Ghost cells (`g_`-prefix) are read past and skipped.

6.2 Reference geometry

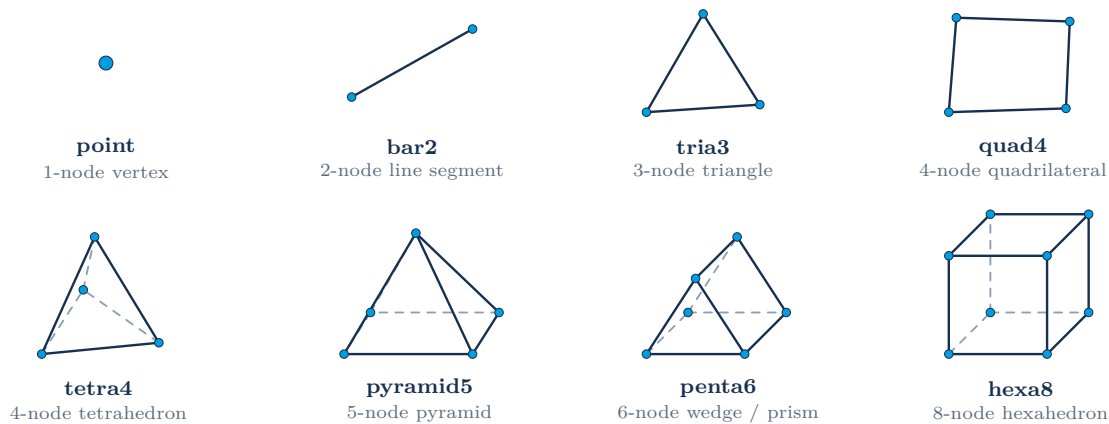


Figure 2. The eight renderable element types — one per topological dimension pair: a 0-D point, a 1-D line, two 2-D surface cells (triangle, quadrilateral), and four 3-D volume cells (tetrahedron, pyramid, triangular prism, hexahedron). Front edges in navy, back edges dashed. Quadratic variants (`bar3`, `tria6`, `quad8`, `tetra10`, `pyramid13`, `penta15`, `hexa20`) parse extra indices but only the corner subset shown here drives rendering.

6.3 VTK ↔ EnSight mapping

| VTK type | EnSight | Notes |
|-------------------------------------|---------------|---|
| VTK_VERTEX, VTK_POLY_VERTEX | point | One point per index |
| VTK_LINE, VTK_POLY_LINE | bar2 | Strip becomes one bar2 per consecutive pair |
| VTK_TRIANGLE, VTK_TRIANGLE_STRIP | tria3 | Strip unrolled to triangles |
| VTK_POLYGON | tria3 / quad4 | Length 3 / 4 direct; ≥ 5 fan-triangulated to tria3 |
| VTK_QUAD, VTK_PIXEL | quad4 | Pixel applies EnSight corner-order swap |
| VTK_TETRA | tetra4 | Direct |
| VTK_PYRAMID | pyramid5 | Direct |
| VTK_WEDGE | penta6 | Direct |
| VTK_HEXAHEDRON, VTK_VOXEL | hexa8 | Voxel applies the corner-order swap |
| VTK_QUADRATIC_* | corner type | Quadratic variants reduced to their corner subset |

IST Vector



Reference

Shortcuts, paths, performance

7. Quick reference

7.1 Keyboard

| Keys | Action |
|-------------------|--|
| Ctrl+O | Open mesh file (file picker). |
| Ctrl+S | Save current view to the saved-views library. |
| Ctrl+P | Export the live scene as a vector PDF. |
| F | Best View at the current step (entropy-scored direction; see §5.1). |
| Space | Toggle playback (play / pause) of the animation timeline. |
| Ctrl+Z | Undo display-state change. Ctrl+Y or Ctrl+Shift+Z redoes. Undo applies to display state, not to the file load. |
| Ctrl+click | Node probe. Reports coords + every active field value. |
| Ctrl+Shift+click | Element probe. Reports index, type and node count, centroid, active scalar (per-element or node-average), and the active vector's average magnitude. |
| Esc | Clear active probe. |
| F1 | Show keyboard-shortcuts cheat sheet (modal). |
| Left mouse drag | Trackball rotate. Roll engages when the drag starts near the arcball pole. |
| Left double-click | Best View at the current step (same as F). |
| Right mouse drag | Pan. |
| Middle mouse drag | Pan (alias for right-drag). |
| Wheel | Ortho zoom. |

The viewport carries a small *i* affordance in its upper-right corner; hovering it surfaces a mouse-controls summary, complementing the F1 modal which lists keyboard shortcuts. The always-visible chrome — the two top toolbars and the left / right icon rails — fills with Hasselblad orange (#FF6A00) on hover and reveals its tooltip only after the pointer has settled for 2 s, so the colour change is the instant feedback and the help text never flickers under a moving pointer; the *i* button keeps its own 120 ms reveal.

7.2 Performance

What scales with what, and where the time goes:

| Operation | Typical cost | Notes |
|--------------------------|--------------|--|
| Open file (case binary) | 1-3 s/100 MB | Parallel parser, mmap-style buffered read. |
| Step swap (transient) | 5-20 ms | Scratch buffers persistent; face-normal cache hits when geometry doesn't change between steps. Per-tick scalar / vector lerp uses a two-slot ping-pong scratch on every FieldData so playback is allocation-free. |
| Field switch | 10-50 ms | Min/max scan memoised per array reference; second visit is $O(1)$. |
| Range edit (slider) | 5-10 ms | Color-only fast path skips face normals, feature edges, contours. |
| Displace-nodes (slider) | uniform-only | Per-vertex unscaled displacement uploaded once at geometry-rebuild time; the slider drives a single <code>u_def_scale_now</code> uniform. Scrub is $O(1)$ regardless of mesh size. On release a one-shot rebuild re-bakes face normals at the new still-frame scale. |
| Iso / contour ribbons | uniform-only | Edge endpoints uploaded once; the vertex shader expands each into a screen-aligned ribbon from <code>u_view_dir</code> per frame. Rotation no longer rebuilds the ribbon VBO. |
| Rotate / zoom (interact) | - | Half-resolution FBO + bilinear upscale during motion. Full res on idle. |
| Contour regeneration | 15-40 ms | Cached on (field, step, N, range, BW, invert, log, labels); next view from the same key is free. |
| PDF export | 0.5-3 s | Vector output; size proportional to visible-face count. Ultra fine pdf raises both (re-shaded sub-triangles, N^2 per face). |
| MP4 export (1080p, 8 s) | 5-15 s | libx264 preset slow, CRF 18 (Cinematic); FFmpeg or Media Foundation. |

Eager vs lazy load. Case bundles are read fully into memory when the total file size is $\leq 85\%$ of the machine's available RAM (15% slack for the OS). Larger bundles fall back to lazy streaming: step0 is read up front, the rest stream in on slider / playback advance, with a ± 2 -step background prefetch absorbing the next integer-step crossing. On a 64 GB host the 3.9 GB 3D-beam transient loads eagerly, so live playback issues zero file reads and the integer-step boundaries are smoothed by the playback animation pipeline below.

Playback animation. Live playback at 60 Hz interpolates between adjacent integer steps:

- **Same-topology transients.** Catmull-Rom [9] cubic across four steps (P / A / B / N) gives C^1 -continuous position and field motion — the velocity is continuous at every integer boundary, eliminating the per-step “kink” a pure linear lerp produces. At the playback range endpoints the spline degenerates to a no-overshoot boundary form.
- **Adaptive-remesh transients.** When adjacent steps differ in node count there is no per-vertex correspondence, so the renderer captures the outgoing step's full scene to an offscreen FBO at the moment of topology change, renders the incoming step to a second FBO each frame of a brief fade window, and composites them on the screen FBO via a fullscreen quad as $\text{prev} \cdot (1 - \alpha) + \text{curr} \cdot \alpha$. Every visible layer (surface, edges, segments, contour ribbons, bars, points) fades uniformly — no overlay snap.
- The fade duration self-tunes to the actual interval between topology changes (clamped [32, 250] ms), so at steady-state playback α reaches 1 exactly as the next fade starts at 0 — one continuous smoothstep dissolve across every boundary.
- Per-tick `SetCurrentStep` is gated on the integer step actually changing, so a 60 Hz tick stream against a slower step rate doesn't re-invoke the lazy-loader (or re-trigger a same-step rebuild) on every frame.

7.3 Numerical guards

NaN and $\pm\infty$ are skipped wherever a reduction is taken (min/max scans, smoothing accumulators, threshold comparisons, mesh-quality statistics) so a single bad sample cannot collapse a colour bar, force the surface to grey, or hide every element.

7.4 Limitations

- Iso-lines on per-element scalars are drawn off an internally recovered nodal field (inverse-area-weighted projection from elements to nodes) — the recovery is implicit, not user-configurable. If the per-element field is genuinely piecewise-constant with sharp jumps, prefer the threshold card or an explicit Field-Filter recovery.
- Tensor glyphs are real-symmetric only; the renderer symmetrises asymmetric input before Jacobi.

- Pathlines and streaklines require a transient vector field; for steady fields the loader synthesises a single step and the integration falls back to streamline equivalents.
- Symmetry expansion is capped at 100 million total expanded nodes (across the union of reflected and rotated copies). Above this the expansion is refused with a status-bar message; reduce symmetry copies or thin the base mesh.
- Meshes above 50 M nodes are rejected at load. Below that, the **Force-derive** chevron is the user's opt-in to populate derived fields.
- **Displace-nodes slider** with symmetry on falls back to a full per-tick rebuild — the shader-side path skips when symmetry is active because the expander does not currently transform displacement vectors alongside positions. Scrubbing remains responsive on small/medium meshes; on very large meshes the symmetry case is the slow path.
- Adaptive-remesh cross-fade requires the GL context to support render-to-texture (FBO + sampler-2D + `glActiveTexture`); on stripped-down GLES 2 hosts it falls back to a snap at every topology-change boundary.
- Below 15 GB of installed RAM the application warns at startup; large transient case bundles will stream from disk (lazy mode) and live playback may stutter at step boundaries when the prefetcher misses.
- Supported display range is 1280×768 to 7680×4320 DIP (covers 1366×768 widescreen, 1280×1024 5:4, and 8K UHD). Outside these bounds a warning dialog flags that chrome scaling may clip toolbar controls (below) or render under-sized relative to the viewport (above); the application still runs but layout is no longer guaranteed.

8. References

- [1] O. C. Zienkiewicz and J. Z. Zhu, “The superconvergent patch recovery and a posteriori error estimates, part 1,” *International Journal for Numerical Methods in Engineering*, vol. 33, no. 7, pp. 1331–1364, 1992.
- [2] O. C. Zienkiewicz and J. Z. Zhu, “The superconvergent patch recovery and a posteriori error estimates, part 2,” *International Journal for Numerical Methods in Engineering*, vol. 33, no. 7, pp. 1365–1382, 1992.
- [3] H. Wendland, “Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree,” *Advances in Computational Mathematics*, vol. 4, no. 1, pp. 389–396, 1995.
- [4] P. Lancaster and K. Salkauskas, “Surfaces generated by moving least squares methods,” *Mathematics of Computation*, vol. 37, no. 155, pp. 141–158, 1981.
- [5] G. Taubin, “A signal processing approach to fair surface design,” in *Proceedings of SIGGRAPH '95*, ACM, 1995, pp. 351–358.
- [6] P. Perona and J. Malik, “Scale-space and edge detection using anisotropic diffusion,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–639, 1990.
- [7] P.-P. Vázquez, M. Feixas, M. Sbert, and W. Heidrich, “Viewpoint selection using viewpoint entropy,” in *Vision, Modeling, and Visualisation 2001*, Aka GmbH, 2001, pp. 273–280.
- [8] K. Shoemake, “Animating rotation with quaternion curves,” in *Proceedings of SIGGRAPH '85*, ACM, 1985, pp. 245–254.
- [9] E. Catmull and R. Rom, “A class of local interpolating splines,” in *Computer Aided Geometric Design*, R. E. Barnhill and R. F. Riesenfeld, Eds. Academic Press, 1974, pp. 317–326.
- [10] J. R. Dormand and P. J. Prince, “A family of embedded Runge–Kutta formulae,” *Journal of Computational and Applied Mathematics*, vol. 6, no. 1, pp. 19–26, 1980.

IST Vector Postprocessor — P. Areias — pedro.areias@tecnico.ulisboa.pt — Instituto Superior Técnico, Universidade de Lisboa.