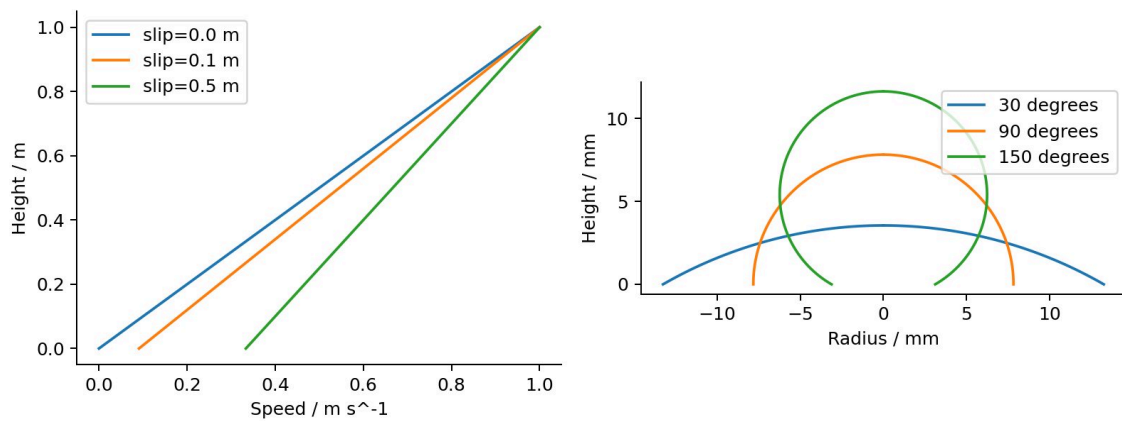


Rheon

Discrete Fluid Simulation

Solids, liquids and compressible waves



Mathematics · Checked contracts · Numerical references · 3D teaching laboratories

14 October 2026

Contents

1. [1 The numerical contract](#)
2. [2 From conservation to a tractable model](#)
3. [3 Staggered geometry and discrete operators](#)
4. [4 Pressure as a constrained correction](#)
5. [5 Solving the pressure system](#)
6. [6 Transport and the price of stability](#)
7. [7 Higher order transport with bounded failure](#)
8. [8 Boundaries and obstacles](#)
9. [9 Liquid interfaces and volume](#)
10. [10 Forces viscosity and material behavior](#)
11. [11 Particle alternatives and hybrid transfers](#)
12. [12 Time stepping and error budgets](#)
13. [13 Memory layout and parallel execution](#)
14. [14 Guidance as a geometric measurement](#)
15. [15 Exact mathematics and finite arithmetic](#)
16. [16 The checked Lean companion](#)
17. [17 A staged Rust implementation](#)
18. [18 Validation and remaining research](#)
19. [19 Forces and moving-boundary work](#)
20. [20 From collision meshes to fluid boundaries](#)
21. [21 Density and real liquid interfaces](#)
22. [22 Viscosity, stress and dissipation](#)
23. [23 Wetting, adhesion and the contact line](#)
24. [24 Slip and wall friction](#)
25. [25 Surface tension and integrated qualification](#)
26. [26 Compressible state, thermodynamics and waves](#)
27. [27 Conservative compressible discretization](#)
28. [28 Compressible boundaries and a verification ladder](#)
29. [A Algorithm reference](#)
30. [B Reproduction and evidence](#)
31. [C Primary sources](#)
32. [D Notation and provenance](#)
33. [E A worked matrix laboratory](#)
34. [F Expansion contracts, sources and evidence](#)
35. [One progression: wall traction, exact trace and external force](#)
36. [Native finite wall friction in a fixed flat slab](#)
37. [Exact compatible endpoint no-slip in a fixed slab](#)
38. [Uniform tangential forcing and Poiseuille startup](#)
39. [One static obstacle geometry for collision and flux](#)
40. [Stationary obstacle pressure and reduced wall shear](#)
41. [Requirements map and the next solid-fluid geometry contract](#)

- 42. [Implementation guide: the bounded material preview](#)
- 43. [CPU scheduling contracts and the Pantograph comparison](#)
- 44. [CAD and engineering visualization contracts](#)

1 The numerical contract

Rheon needs to place a moving three-dimensional fluid shape deliberately while leaving memory available for image-generation models. The repository describes a low-resolution, interactive simulation framework and a standalone demonstration application. This book develops the mathematical and engineering foundation for that framework. The recommended first implementation is a bounded Cartesian staggered-grid solver with explicit diagnostics, followed by a deliberately separate free-surface branch. Neither a rendered fluid image nor a successful pressure solve alone establishes the intended result.

The repository at revision 0cb48429789699578b640d30d9d5dc99c6921ce4 contains a README, license and ignore file. It has no existing solver whose behavior can be assumed. Accordingly, algorithms here are specifications and research recommendations until implemented and qualified. The numerical companion executes small reference experiments; the Lean companion checks selected exact discrete contracts. Proposed architecture must not quietly become a claim about software that does not exist.

Spatial accuracy has several meanings

A fluid can be physically plausible while occupying the wrong silhouette, or match a silhouette while violating mass conservation. We distinguish geometric error, numerical error, model error and downstream conditioning error. Geometric error measures interface position against a defined target. Numerical error compares the discretization against the selected mathematical model. Model error compares that model with the real phenomenon. Conditioning error measures how a generated image responds to a guidance representation. These quantities can move in different directions.

Let the simulated state be s , a camera be C , and the deterministic guidance renderer be R_C . An image-generation system applies a map F to that representation and additional inputs z :

$$s \mapsto R_C(s) \mapsto F(R_C(s), z).$$

A bound on the first map says nothing automatic about the sensitivity of the second. ControlNet demonstrates spatial conditioning with several image types, including depth and edges [S14]; it supplies no guarantee that any particular fluid boundary will be preserved exactly. Rheon’s measurable responsibility should end at a documented guidance image and metadata unless the integrated diffusion model is separately evaluated.

For an orthographic camera with world width L_x and image width W , a transverse position error δx produces approximately $W\delta x/L_x$ pixels of displacement. At $L_x = 2$ metres and $W = 1024$, a one-centimetre error is 5.12 pixels. A 64-cell domain of the same width has 31.25 mm cells, or 16 pixels per cell. These arithmetic facts constrain what positionally accurate can mean. Interpolated surfaces vary continuously inside a cell but cannot recover arbitrary unresolved topology.

A staged scope

The first stage should solve incompressible velocity in a stationary box with smoke-like scalar transport. This isolates pressure, advection, boundaries and memory ownership. The second adds solid obstacles and measured geometric error. The third adds liquid interfaces, volume diagnostics and liquid-specific pressure boundaries. Non-Newtonian stress and particle-grid methods are extensions whose cost must be justified by a concrete failure of the baseline.

A source mask, obstacle mask and fluid mask are different fields. A source injects state according to a schedule. An obstacle restricts motion. A liquid mask identifies pressure unknowns and free-surface neighbors. Treating all three as one occupancy array creates contradictory boundary behavior. Visual density is an advected tracer unless a model explicitly equates it with mass density; smoke brightness cannot be used directly as the coefficient in an incompressibility equation.

The first acceptance suite should establish finite state, controlled divergence, stated transport behavior, stable boundary semantics and reproducible guidance. A frame-time target remains a design target until a workload, hardware class, competing model-memory load and variability policy are chosen. Real time is not a theorem and a short reference script is not its benchmark.

Reading the evidence

Each chapter derives discrete contracts before discussing implementation. Worked examples use small systems so sign, scale and conservation failures can be inspected by hand. The experiments extend these checks to a three-dimensional graph and periodic transport. Chapter 16 maps every formal theorem to its assumptions. Chapter 17 defines a Rust implementation sequence; later language ports should inherit verified behavior rather than independently reinterpret the equations.

Gating a pressure proposal

Start with two cells, unit volumes and conductance, and source $b = (1, -1)$. The candidate $p = (s, s-1+e)$ produces shared flux $1-e$ and full residual $(e, -e)$. Set $e = 1/4$ and tolerance $1/8$: the proposal is rejected and the exact fallback $p = (0, -1)$ supplies flux 1. The displayed used field is the fallback, so its residual is zero. This is a synthetic acceptance example; no AI model runs in the browser. A real provider must additionally bind model/schema, physical units, source identity, component compatibility and resource limits.

A gauge shift is not an error

Set $e = 0$ and vary s . Flux and residual remain unchanged because only the pressure jump enters this two-cell operator. Set tolerance $1/8$ and $e = 1/16$: acceptance now permits a nonzero residual. A passed tolerance therefore means an explicitly bounded error, rather than an exact solve. Inspect the used residual and the candidate contract together. A nonfinite or wrong-unit proposal fails before numerical acceptance. Camera and reading controls leave the candidate and used state unchanged.

► Practical study proof scope

2 From conservation to a tractable model

Consider a fluid region Ω , velocity $\mathbf{u}$, mass density ρ , pressure p , and body acceleration $\mathbf{f}$. Local mass conservation is

$$\partial_t \rho + \nabla \cdot (\rho \mathbf{u}) = 0.$$

If density is constant along trajectories, then $D\rho/Dt = 0$, and the product rule gives $\nabla \cdot \mathbf{u} = 0$. This is a model assumption about volume preservation, not a statement that pressure is zero. Pressure is the constraint force that makes velocity compatible with incompressibility and boundaries.

For a Newtonian fluid of constant density and viscosity, momentum becomes

$$\partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\rho^{-1} \nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}, \quad \nabla \cdot \mathbf{u} = 0.$$

Here $\nu = \mu/\rho$ is kinematic viscosity, with units square metres per second. Pressure has units kilograms per metre per second squared. Confusing dynamic viscosity μ with ν introduces a factor of density. Chorin's original projection work provides a foundational velocity-pressure splitting construction [S3]. Rheon's finite operators are specified independently in later chapters.

Nondimensionalization

Choose a length L , speed U , time L/U , and pressure scale ρU^2 . Substitution gives

$$\partial_t \hat{\mathbf{u}} + (\hat{\mathbf{u}} \cdot \hat{\nabla}) \hat{\mathbf{u}} = -\hat{\nabla} \hat{p} + \text{Re}^{-1} \hat{\nabla}^2 \hat{\mathbf{u}} + \hat{\mathbf{f}}, \quad \text{Re} = UL/\nu.$$

Changing scene scale while retaining numerical settings changes the apparent material. A two-metre scene moving at one metre per second with $\nu = 0.01$ has $\text{Re} = 200$. Reducing length by ten at the same speed and viscosity gives 20. Configuration should retain world units or explicitly declare nondimensional parameters, never silently interpret one value as both.

Gravity introduces $\text{Fr} = U/\sqrt{gL}$; surface tension introduces $\text{We} = \rho U^2 L/\sigma$, where σ is force per unit length.

Changing surface tension to obtain a desired shape can be intentional but should not be called calibrated water.

Features with radii comparable to grid spacing have poorly resolved curvature and pressure jumps.

Energy and boundaries

Multiply momentum by $\rho \mathbf{u}$, integrate, and assume sufficient smoothness, incompressibility and vanishing boundary terms, such as periodic boundaries:

$$\frac{d}{dt} \frac{\rho}{2} \int_{\Omega} |\mathbf{u}|^2 dV = -\mu \int_{\Omega} |\nabla \mathbf{u}|^2 dV + \rho \int_{\Omega} \mathbf{f} \cdot \mathbf{u} dV.$$

Advection becomes a boundary flux; pressure vanishes against divergence-free velocity; integration by parts makes viscosity dissipative. Moving walls, inlets and outlets add work and transport terms. Energy gain is not automatically wrong, but it needs an identified source. Vorticity confinement is an added force and belongs in this accounting.

What the model excludes

A constant-density incompressible solver omits acoustic waves, compressibility-driven shocks and phase-transition density changes. A buoyant smoke tracer under a Boussinesq approximation is not combustion. A single-phase liquid with prescribed atmospheric pressure does not model trapped compressible bubbles. An inviscid baseline cannot predict calibrated viscous folding. No finite collection of discrete lemmas proves global regularity of three-dimensional Navier-Stokes flow.

These exclusions guide the API. A material enum should describe implemented constitutive behavior. If the only stress is constant Newtonian viscosity, expose that. If foam means a thresholded visual tracer, say so rather than imply a multiphase physical model.

Splitting errors

Let A denote transport and B another evolution operator. One step $e^{\Delta t B} e^{\Delta t A}$ differs from $e^{\Delta t (A+B)}$ by commutator terms of order Δt^2 locally under suitable smoothness. Over fixed duration, the usual global order is one. Symmetric splitting can improve order under compatible assumptions, but changing fluid regions and boundary handling can invalidate that improvement. Test accuracy on smooth manufactured problems before advertising it for obstacles or free surfaces.

Wall forces and body work

Consider constant viscosity $\mu > 0$ in a one-metre plate gap with unit area. The lower wall is stationary, the upper wall moves at U , and a uniform body force G acts along x . The steady solution is

$$u(y) = Uy + \frac{G}{2\mu}y(1-y), \quad \tau(y) = \mu U + \frac{G}{2}(1-2y).$$

Traction on the fluid uses its outward normal: bottom traction is $-\mu U - G/2$ and top traction is $\mu U - G/2$. Their sum plus the body force is zero. The browser recomputes nine velocity/shear samples, wall traction and work. With $\mu = U = 1$ and $G = 2$, the top traction is zero and the bottom is -2 Pa. Body work supplies the dissipation, even though the upper wall moves.

The volume-average speed is $U/2 + G/(12\mu)$. Wall power is $U(\mu U - G/2)$, body power is G times that average, and dissipation is $\mu U^2 + G^2/(12\mu)$. Their exact balance follows by integrating the polynomial profile and its squared derivative. Density does not enter this already specified steady unidirectional balance. Arbitrary geometry, unsteady inertia, thermal work and solid response require additional equations and boundary data.

► Practical study proof scope

3 Staggered geometry and discrete operators

A Cartesian staggered grid stores scalars at cell centers and velocity components on faces normal to their axes. Harlow and Welch's marker-and-cell formulation is the historical foundation [S2]. Pressure has $n_x n_y n_z$ entries; x-velocity has $(n_x + 1)n_y n_z$, y-velocity has $n_x(n_y + 1)n_z$, and z-velocity has $n_x n_y(n_z + 1)$. These shapes differ even in a cubic domain.

With origin $\mathbf{o}$ and spacing h , a cell center is $\mathbf{o} + h(i + 1/2, j + 1/2, k + 1/2)$. An x-face is at $\mathbf{o} + h(i, j + 1/2, k + 1/2)$. Each component therefore requires its own sampling offset. Using cell-centered offsets for all components creates a half-cell error and can destroy an otherwise correct projection.

Divergence from flux balance

Integrate divergence over a cubic cell, approximate each face flux by its stored normal velocity, and divide by volume:

$$(D\mathbf{u})_{ijk} = \frac{u_{i+1,j,k} - u_{i,j,k}}{h} + \frac{v_{i,j+1,k} - v_{i,j,k}}{h} + \frac{w_{i,j,k+1} - w_{i,j,k}}{h}.$$

The pressure gradient at an interior x-face is $(p_{i,j,k} - p_{i-1,j,k})/h$. The same face appears in two neighboring divergences with opposite signs. This shared-face requirement gives cancellation of internal fluxes.

For the formal model, define a cells-by-faces matrix B . Each oriented internal face column contains -1 at its tail and $+1$ at its head. At unit spacing, physical divergence is $-B\mathbf{u}$, while pressure difference is $B^T p$. The proofs and experiments use this convention. A uniform physical grid inserts factors $1/h$. Cut cells require cell-volume and face-area metrics, not an informal reuse of the unweighted identity.

The adjoint identity

Exchange finite sums:

$$\sum_i p_i \sum_e B_{ie} u_e = \sum_e \left(\sum_i B_{ie} p_i \right) u_e.$$

This exact discrete integration-by-parts identity says $p^T B u = (B^T p)^T u$. Physical divergence and gradient are negative adjoints. The Lean theorem named `adjoint_identity` proves the finite-sum statement for every real matrix of the declared shape. It does not prove that array traversal assembled the right matrix.

Two cells connected by one face give

$$B = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \quad B^T p = p_1 - p_0, \quad BB^T = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}.$$

Face velocity 3 gives physical divergence $(3, -3)$, whose sum vanishes. Reversing both orientation and stored velocity sign preserves the physical field; reversing only one is a bug.

Layout and indexing

Use x-fastest flattening:

$$\text{index}(i, j, k) = i + n_x(j + n_y k).$$

The Lean theorem `flatten_in_bounds` proves this lies below $n_x n_y n_z$ for in-range natural coordinates. Machine integers add a separate obligation: shape products and additions must fit before allocation or indexing. Validate shape arithmetic with checked operations when constructing the grid. Retain distinct validated shape types for cells and each face orientation. Repeated handwritten dimension arithmetic spreads the contract across every kernel.

Separate interior stencil loops from boundary updates. Ghost layers simplify sampling but enlarge storage and need one refresh owner. Stale ghost velocities can make transport and pressure use different wall conditions. Tests should include a constant field, linear pressure ramp, one nonzero face and manufactured divergence-free vortex, checking boundaries as well as interiors.

Rectangular cells and physical units

Let cell widths be h_x, h_y, h_z , volume $V = h_x h_y h_z$, and x-face area $S_x = h_y h_z$. Integrating mass flux first, then dividing by volume, gives $S_x/V = 1/h_x$. Thus the rectangular-grid divergence has different denominators per direction. Replacing all three by one average spacing changes both the PDE and the pressure spectrum.

Pressure in pascals divided by x-center separation h_x gives a pressure gradient in pascals per metre. Multiplying by inverse density and time step gives metres per second, as needed for a velocity correction. The x-face correction is

$$u_{i,j,k}^+ = u_{i,j,k}^* - \frac{\Delta t}{\rho_{i,j,k}^x h_x} (p_{i,j,k} - p_{i-1,j,k}).$$

Substituting all three face corrections into divergence gives the familiar seven-point operator. With constant density, an interior row has diagonal $2/(\rho h_x^2) + 2/(\rho h_y^2) + 2/(\rho h_z^2)$, and opposite-direction neighbors receive the corresponding negatives. Every coefficient has units inverse density per square metre. The right-hand side is negative physical divergence divided by time step.

This row-divided form is symmetric when all volumes are equal. For unequal cell volumes, dividing each row by its own volume usually destroys Euclidean symmetry. CG must then use a symmetrized integrated system or a correctly weighted formulation. Calling the nonsymmetric row-divided matrix a Laplacian does not make ordinary CG valid.

Integrated metric construction

Separate topology from geometry. Let B be the unscaled incidence matrix, S the diagonal open-face area matrix, L the diagonal pressure-center distance matrix, and V the diagonal cell-volume matrix. Unknown face speed is u . Let Q be the vector of known outward volume flux through prescribed-velocity boundaries. Then

$$Du = V^{-1}(Q - BSu), \quad Gp = L^{-1}B^T p.$$

The first expression is affine because of known boundary flux. The two-point gradient assumes pressure sample separation is aligned with the face normal and L contains that normal separation. This holds for the selected Cartesian MAC pressure sample locations. Positive areas, distances and volumes alone do not establish consistency for arbitrary cut-cell centroids. Nonorthogonal centroid corrections require a separately derived gradient and matching adjoint operator; they are outside this baseline. An affine-pressure fixture independently checks the normal-gradient consistency and demonstrates failure on a skew connection. With diagonal face density R , correction is $u^+ = u^* - \Delta t R^{-1} G p$. The symmetric integrated pressure system is

$$Kp = b, \quad K = BSR^{-1}L^{-1}B^T, \quad b = (BSu^* - Q)/\Delta t.$$

For positive geometry and density, face coefficient $S_e/(\rho_e \ell_e)$ is nonnegative. K has the same graph energy structure as the Lean model. It is symmetric in the ordinary Euclidean inner product because cell-volume division was left outside the solve. The physical residual relation is now

$$Du^+ = -\Delta t V^{-1}(b - Kp).$$

Consequently a small integrated residual can still produce large divergence in tiny cells. A stopping test should inspect volume-scaled divergence as well as solver norms. The existing Lean theorem is the unweighted algebraic core; this physical metric interpretation is a derived specification.

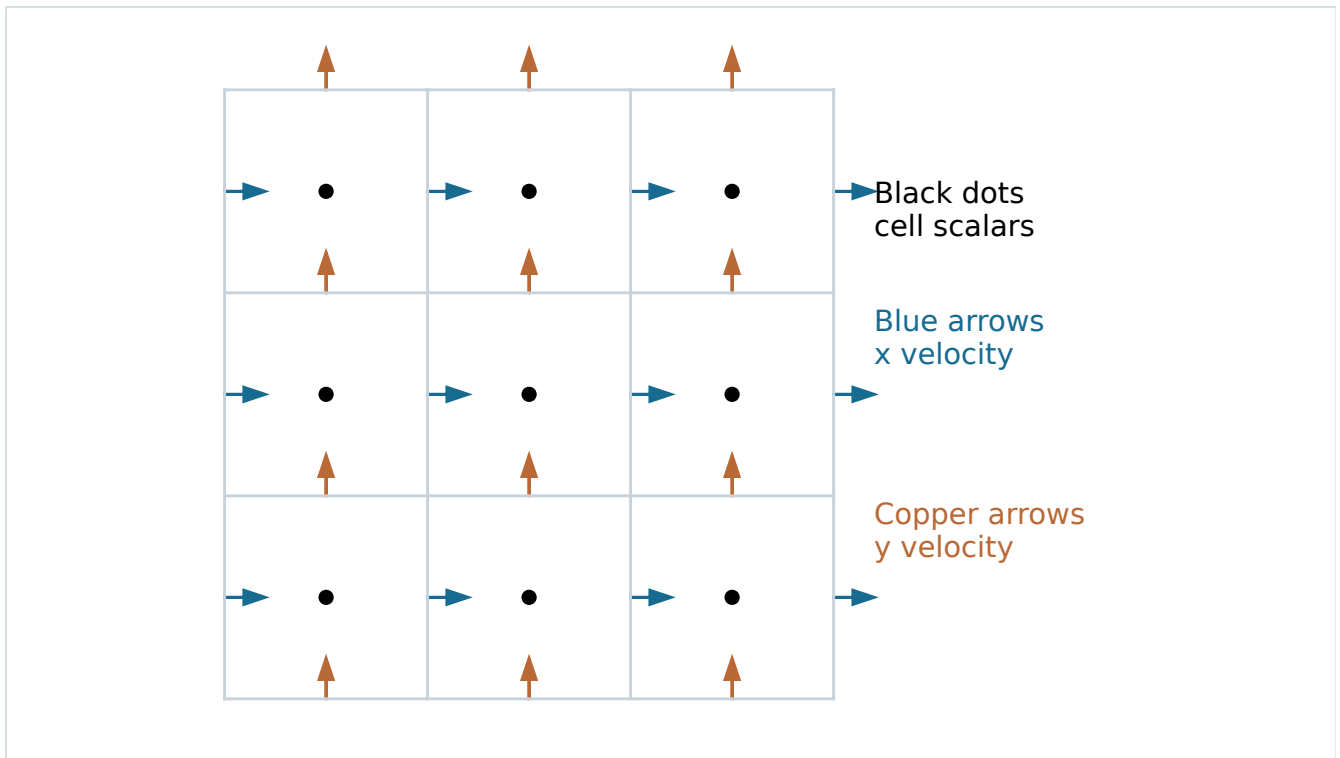
For dimensional checking, $S/(\rho \ell)$ has units metres to the fourth per kilogram. Multiplying by pressure gives cubic metres per second squared, matching volume flux divided by time. V then converts the corrected flux mismatch to inverse seconds. This check catches missing density, distance and volume factors before a simulation is run.

Coefficient storage

For a regular rectangular grid, store dimensions and three reciprocal spacings once. For irregular geometry, each active face needs open area, pressure distance and density information, or a precomputed transmissibility with provenance. One transmissibility value must be shared by both adjacent pressure rows. Independently computing it twice can introduce asymmetry through different clipping or density choices.

At a density discontinuity, derive the coefficient from the selected pressure-gradient model. For two segments with lengths ℓ_a, ℓ_b and densities ρ_a, ρ_b , a series-resistance argument gives transmissibility $S/(\rho_a \ell_a + \rho_b \ell_b)$. This differs from blindly averaging inverse density at the face. The correct choice depends on how pressure and acceleration are integrated; retain the derivation and test a one-dimensional interface fixture.

Cell-volume, face-area and center-distance validity are separate checks. A positive volume does not imply a valid connecting face. A zero area removes a connection; a zero distance is invalid. These distinctions determine graph connectivity and pressure gauges.



Original reference illustration for 3 Staggered geometry and discrete operators

4 Pressure as a constrained correction

Let $\mathbf{u}^*$ be velocity after non-pressure updates and W a diagonal matrix of nonnegative face weights. The graph correction is

$$\mathbf{u}^+ = \mathbf{u}^* - \Delta t W B^T p, \quad A = B W B^T, \quad b = B \mathbf{u}^* / \Delta t.$$

Solving $A p = b$ makes $B \mathbf{u}^+ = 0$, hence physical divergence $D = -B$ vanishes. Density and grid-spacing factors can be absorbed into W and B , but the implementation must state how. A dimensionless graph formula must not be copied into a metre-based solver without scaling.

The pressure quadratic form

$$p^T A p = (B^T p)^T W (B^T p) = \sum_e w_e (p_{\text{head}(e)} - p_{\text{tail}(e)})^2.$$

This is nonnegative when weights are nonnegative; the matrix is symmetric. Theorems `pressure_energy`, `pressure_nonnegative` and `pressure_symmetric` establish these facts. Strictly positive weights make zero energy imply equal pressure across every edge. On a connected closed graph, pressure is constant; on a disconnected graph, each component has its own constant. This connectivity-to-kernel argument is not currently in the Lean inventory.

A closed domain therefore gives a positive semidefinite operator. Pin one pressure per component by eliminating the row and column, or solve in the mean-zero subspace. Replacing only a row with an identity row breaks symmetry unless the column is handled consistently. Prescribed pressure is eliminated and its known contribution moved to the right-hand side.

Compatibility

Each closed component needs a zero-sum pressure right-hand side. Balanced incidence columns imply $\mathbf{1}^T B = 0$. A net inlet into a sealed domain violates compatibility. Subtracting the right-hand-side mean changes the requested problem. It may be an explicitly chosen correction policy but cannot be silent recovery.

The `constant_pressure_nullspace` theorem proves the constant nullspace under balanced columns.

`Internal_flux_conservation` proves matching cancellation. A nonzero total divergence in a fully closed graph, beyond expected rounding, should trigger assembly, boundary timing or unit checks before increasing solver iterations.

Residual as a diagnostic

For approximate pressure p_k , let $r_k = b - A p_k$. Substitution gives, without a convergence assumption,

$$B \mathbf{u}^+ = \Delta t r_k, \quad D \mathbf{u}^+ = -\Delta t r_k.$$

Projection_residual is the central checked theorem. It links stopping to a velocity constraint. A residual norm means different things if the matrix has been multiplied by h^2 , density or time step. Recompute divergence through the production operator and report units. Relative residual alone is insufficient near zero right-hand side; use a documented absolute floor.

Two cells worked completely

Take time step 1, weight 1 and face velocity 3. Then $b = (-3, 3)$. Fix $p_0 = 0$; the reduced equation gives $p_1 = 3$. Corrected velocity is $3 - (3 - 0) = 0$. At $p_1 = 2.9$, residual is $(-0.1, 0.1)$, velocity is 0.1 and physical divergence is $(0.1, -0.1)$. Residual and divergence agree exactly, while acceptable magnitude remains a scene-dependent engineering decision.

For positive weights, minimize velocity change in the W^{-1} metric subject to incompressibility. A Lagrange-multiplier calculation yields the same normal equations. This explains nonincrease of the corresponding kinetic norm for compatible exact projection without boundary work. The current Lean package proves operator identities, not the complete constrained-minimization theorem or its finite-precision version.

The matrix can be algebraically correct and geometrically wrong. The theorem accepts arbitrary B ; it cannot distinguish a neighboring cell from an incorrectly indexed distant one. Independent geometry and stencil tests are essential.

Assembly by boundary case

Every candidate face should be classified before assembling a pressure row. A fluid-fluid face contributes coefficient w to both diagonals and $-w$ to both off-diagonals. A fluid-solid face with prescribed normal speed supplies known boundary flux; it does not add a fictitious pressure neighbor. A fluid-air face with prescribed pressure contributes w to the fluid diagonal and $w p_{\text{air}}$ to its right-hand side. An inactive air-air face contributes no liquid pressure equation.

For example, one liquid cell connected to a prescribed zero-pressure surface at half a cell distance has coefficient $S/(\rho h/2) = 2S/(\rho h)$. Treating the surface as a whole-cell distance halves the coefficient and changes the correction. Conversely, treating a solid as zero-pressure air creates a leaking wall. These two common mistakes can both produce plausible-looking motion while violating the intended boundary.

An isolated liquid cell surrounded entirely by fixed walls has no pressure correction degree of freedom through internal faces. If its prescribed net flux is zero, its pressure is undetermined and velocity is already constrained by boundary data. If net flux is nonzero, the problem is incompatible. It is not repaired by adding a tiny diagonal unless the model explicitly permits compressibility or leakage.

Connected components and gauges

Build a graph using pressure unknowns and strictly positive fluid-fluid transmissibilities. Traverse it with a queue or union-find and record component IDs. A component touching a prescribed-pressure face is anchored by that boundary. A closed component needs a compatibility test and one gauge.

Compute compatibility from integrated flux, using a numerically careful reduction. Its tolerance should account for the magnitudes accumulated and the accepted flux budget. A component with a large inlet and equally large outlet can have cancellation error much larger than an otherwise identical quiescent component. A fixed unscaled epsilon cannot distinguish these cases reliably.

For a closed compatible component, choose a deterministic gauge cell, such as the smallest stable cell index. Remove its row and column; set its known pressure to zero. If using a mean-zero projection instead, apply the projection consistently to residuals, directions and preconditioning. A preconditioner that introduces constant components can corrupt a nominally mean-zero solve.

The topology snapshot should retain the mapping between full cell indices and reduced unknown indices. Do not rebuild it nondeterministically inside every matrix application. When geometry changes, invalidate the operator, component map, preconditioner and any pressure warm start whose gauge mapping is no longer meaningful.

Acceptance in physical space

After solving, apply the pressure gradient once using the same transmissibilities or metric factors as assembly. Then recompute physical divergence independently from the corrected face velocities and prescribed boundary data. Compare this with the predicted residual-derived divergence.

Their difference has two sources: floating arithmetic and inconsistent implementation. The companion quantifies a small arithmetic discrepancy on its reference graph. A much larger mismatch is evidence of wrong coefficients, stale geometry, inconsistent density or a boundary overwrite after correction. This paired diagnostic localizes failures better than either pressure residual or divergence alone.

The worked matrix appendix supplies exact chain and cycle fixtures. Add production tests for one closed component, two disconnected components, one prescribed-pressure opening, an isolated cell and incompatible moving-wall flux. These cases test the domain logic that the arbitrary-B theorem intentionally leaves open.

A collocated checkerboard

The earlier staggered graph argument is not automatically a collocated method. Take four periodic cells at spacing 1 m and kinematic pressures $(1, -1, 1, -1) \text{ m}^2/\text{s}^2$. Each central cell gradient uses its two neighbours, which have equal pressure, so every gradient vanishes. Cell velocities computed as minus r times this gradient vanish too; naive face averaging cannot detect this pressure mode.

In the restricted zero-source, constant-response model, form the face response by adding r times the averaged cell gradient and subtracting r times the direct neighbour pressure difference divided by spacing. At $r=1 \text{ s}$, face velocities are $(2, -2, 2, -2) \text{ m/s}$ and cell divergences $(4, -4, 4, -4) / \text{s}$. Global divergence still sums to zero. Inspecting only a global sum would miss the local response. The extruded strip is one-dimensional and periodic; end faces are identified.

Constant pressure control

Switch to constant pressure. Both central and compact responses now vanish. The [research dossier](#) derives the formula and its periodic quadratic form. The face construction illustrates why pressure and momentum interpolation must be designed together, following the developer-authored [coupling derivation](#). It is not a complete Rhie–Chow/SIMPLE

implementation: variable coefficients, relaxation, transient consistency, boundaries and nonorthogonal reconstruction are absent. No pressure solve or timestep is performed by these controls.

► Checkerboard proof scope

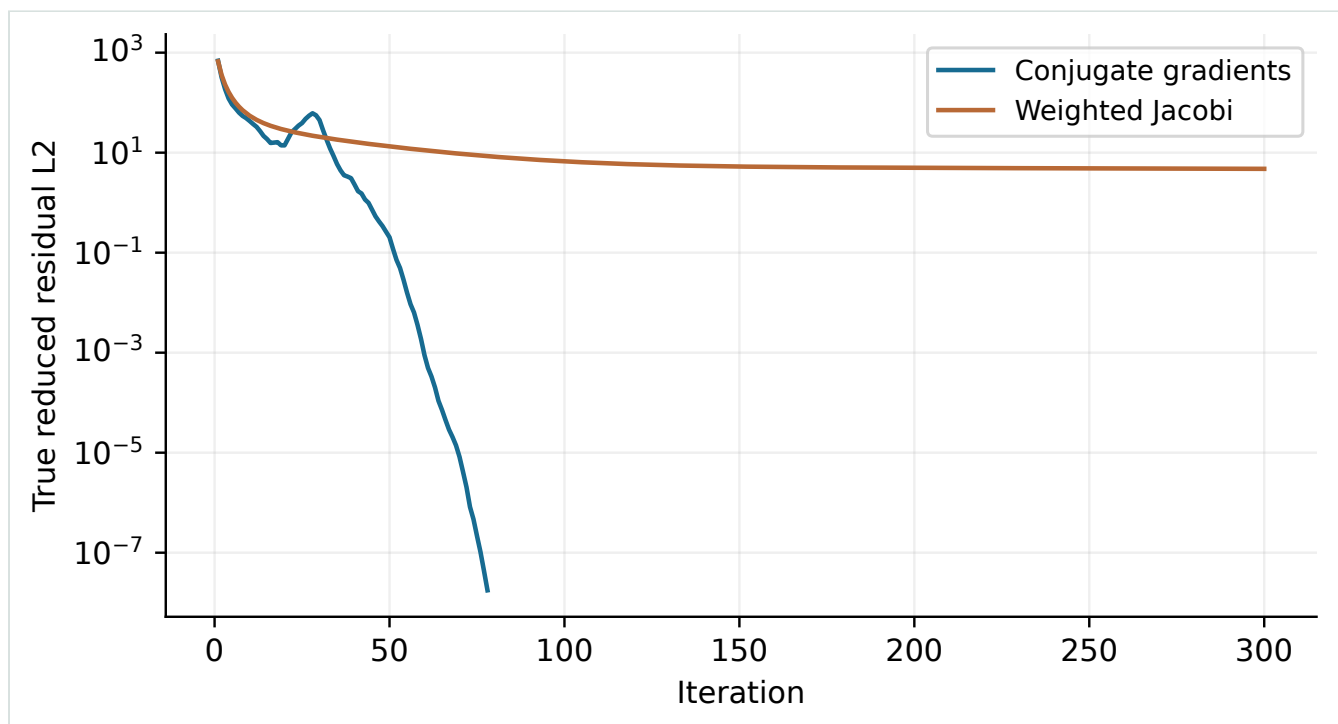
Continuity is not momentum convergence

The embedded two-path network executes a segregated recurrence for $H = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$, $H u + (1,1)p = (1,0)$, $u_1 + u_2 = 0$. Its exact solution is $u = (1/6, -1/6)$, $p = 1/2$. After four iterations at relaxation 1, $u = (5/32, -5/32)$ and $p = 1/2$. Continuity is zero, yet momentum residuals are $(1/32, -1/32)$. The arrows are nondimensional path velocities; this is a reduced linear study, not a collocated CFD backend or a general SIMPLE implementation.

An overrelaxed two-path counterexample

Increase relaxation to $3/2$. The velocity error multiplies by $-5/4$ each iteration, so its squared magnitude grows by $25/16$; pressure error still shrinks by $-1/2$. A good pressure history or zero continuity residual alone cannot qualify the coupled state. Compare actual iteration counts at fixed relaxation.

► Reconstruction and coupling proof scope



Original reference illustration for 4 Pressure as a constrained correction

5 Solving the pressure system

Pressure often determines both frame cost and volume quality. The solver receives an SPD reduced system or an explicitly handled semidefinite system. It returns pressure, iterations, true residual, termination reason and breakdown diagnostics. A vector alone loses information needed to accept or reject a frame.

Jacobi as a reference

With diagonal D_A , weighted Jacobi is

$$p^{k+1} = p^k + \omega D_A^{-1}(b - Ap^k).$$

Convergence requires spectral radius below one for $I - \omega D_A^{-1}A$ on the solved subspace. Parallel local access is simple, but smooth error can decay slowly as grids grow. A fixed iteration count is not fixed divergence quality. Isolated zero-diagonal cells must be classified, never divided by zero.

Jacobi is useful as an independent reference and multigrid smoother. Compare time to an accepted residual, including reductions and synchronization, not one iteration against an unrelated solver iteration.

Conjugate gradients

For an SPD matrix and SPD preconditioner M , initialize $r_0 = b - Ap_0$, $z_0 = M^{-1}r_0$, $d_0 = z_0$. Then

$$\alpha_k = \frac{r_k^T z_k}{d_k^T A d_k}, \quad p_{k+1} = p_k + \alpha_k d_k, \quad r_{k+1} = r_k - \alpha_k A d_k,$$

$$z_{k+1} = M^{-1}r_{k+1}, \quad \beta_k = \frac{r_{k+1}^T z_{k+1}}{r_k^T z_k}, \quad d_{k+1} = z_{k+1} + \beta_k d_k.$$

Shewchuk develops Krylov methods and conditioning in detail [S5]. Rheon's operational contract additionally detects nonfinite values, nonpositive curvature, zero denominators and exhausted budgets. Check the initial residual before forming any quotient. Zero right-hand side and zero guess should return immediately.

The exact-arithmetic energy-error bound for SPD systems is

$$\|e_k\|_A \leq 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k \|e_0\|_A.$$

It is not a runtime prediction. Spectrum, preconditioner cost, precision and memory traffic matter; a singular system requires a specified subspace. The current Lean package does not prove CG convergence.

Preconditioner selection

Diagonal scaling is cheap and parallel. Incomplete Cholesky can reduce iterations but adds factorization, ordering dependence and triangular solves. Modified incomplete Cholesky is discussed in Bridson's notes [S4]. Geometric multigrid attacks smooth error, but coarse levels must retain obstacle and free-surface semantics. A coarse operator that reconnects sealed regions changes the nullspace.

A matrix-free seven-point stencil is attractive for a known regular coefficient structure. Reconstruct neighbor contributions from face weights and diagonals. An explicit sparse matrix remains valuable in tests as an inspectable independent oracle. Comparing two wrappers around one kernel cannot catch shared assembly errors.

Residual and warm starts

Recursive residuals drift from $b - Ap$ in floating point. Recompute the true residual at termination and periodically where needed. Document direction restarts after replacement. Warm starts help only when operator, pressure gauge and topology remain compatible. New disconnected components or grid rescaling require remapping or reset.

The companion uses an 8-cubed closed grid, removes one row and column, and projects deterministic random velocity. It records CG and weighted-Jacobi histories. This compares mechanisms on that small system, not production frame time. Failure leaves the last accepted state intact or publishes a declared degraded preview according to an explicit lifecycle policy.

Residual scales and breakdown policy

Let the integrated solve be $Kp = b$. A relative criterion $\|r\|_2 \leq \tau_{\text{rel}} \|b\|_2$ can be combined with an absolute criterion, but physical acceptance is additionally $\|\Delta t V^{-1} r\|_\infty \leq \tau_{\text{div}}$. The first measures linear-system progress; the second limits maximum cell divergence. They can disagree in cut cells. Physical acceptance must reconstruct the full ungauged residual on every active cell, including eliminated gauge cells. A reduced-system residual alone can hide accumulated imbalance in the removed row. Independently recompute divergence on every active cell after correction. The companion includes a compatible chain where all retained residual entries pass a threshold but the eliminated row fails it.

For a diagonal preconditioner, require positive diagonal entries for every reduced unknown. If the matrix has a zero row, classify the topology before invoking CG. If $r^T z$ or $d^T K d$ is nonpositive in an expected SPD solve, return a breakdown with the iteration and scalar values. Do not take an absolute value or clamp the denominator; that changes the method and conceals the failed assumption.

The source of a breakdown can be an invalid operator, wrong gauge, loss of precision or an inappropriate preconditioner. Retrying in higher precision is useful only after the operator checks pass. A retry should start from a coherent state, recompute true residual and retain the original failure diagnostic.

Why multigrid helps

Jacobi damps different error frequencies at different rates. For a one-dimensional uniform Dirichlet Laplacian, the diagonal-scaled Fourier symbol is $1 - \cos \theta$. Weighted Jacobi's error multiplier is $1 - \omega(1 - \cos \theta)$. Small θ gives a multiplier near one, so smooth error decays slowly. After transferring to a coarser grid, that same physical error becomes less smooth in grid coordinates and easier to correct.

This argument motivates a two-grid method: smooth the current estimate, restrict its residual, solve for a coarse correction, prolong it, and smooth again. Saad provides a primary textbook treatment of multigrid and sparse iterative methods [S22]. The construction below states Rheon's proposed contract rather than asserting an implemented hierarchy.

Transfer operators

Let P map coarse pressure corrections to fine pressure corrections. For uniform cell-centered grids, construct P by trilinear interpolation at fine cell centers expressed in coarse-cell coordinates. The offset is important: a fine center is not generally a coarse node. Near boundaries, interpolation must respect prescribed pressure and component membership.

Choose restriction R consistently with the inner product. In an integrated symmetric system, $R = P^T$ yields the Galerkin operator $K_H = P^T K_h P$. If K_h is SPD and P has full column rank on the solved subspace, then for nonzero coarse x ,

$$x^T K_H x = (Px)^T K_h (Px) > 0.$$

Thus coarse SPD follows from construction, with explicit rank and subspace assumptions. For volume-normalized residuals, volume-weighted restriction is more natural; convert consistently instead of mixing formulas.

On a closed graph, preserving constants requires $P\mathbf{1}_H = \mathbf{1}_h$. Coarse connectivity must not join separate fine components. A coarse cell spanning a thin solid barrier needs multiple component degrees of freedom or a hierarchy policy that preserves separation. Otherwise the coarse correction allows nonphysical communication.

V cycle specification

```
v_cycle(level, rhs, estimate):
    if level is coarsest:
        solve the gauge-fixed coarse system accurately
        return estimate
    apply fixed pre-smoothing sweeps
    residual = rhs - K[level](estimate)
    coarse_rhs = transpose(P[level]) * residual
    coarse_error = zero
    coarse_error = v_cycle(level + 1, coarse_rhs, coarse_error)
    estimate += P[level] * coarse_error
    apply compatible post-smoothing sweeps
    return estimate
```

As a standalone solver, monitor the finest-grid true residual after cycles. As a CG preconditioner, the V cycle must act as a fixed linear SPD map on the relevant subspace. Matching adjoint pre/post smoothing and an SPD coarse solve are important. Adaptive iteration counts, nonsymmetric Gauss-Seidel order or changing hierarchy can violate the usual PCG assumptions. A flexible Krylov method is a different algorithmic choice, not a label to add after the fact.

The expanded companion constructs a small Dirichlet one-dimensional hierarchy with interpolation P, Galerkin coarse matrix and weighted Jacobi smoothing. It checks symmetry, positive eigenvalues and measured residual reduction. This is an executable mechanism fixture, not a three-dimensional obstacle-aware multigrid implementation.

Storage and work

A regular three-dimensional geometric hierarchy has cell counts $N, N/8, N/64, \dots$; their sum approaches $8N/7$. This favorable count excludes per-level masks, transfer metadata, coefficients and temporary vectors. An irregular hierarchy can have much more overhead. Count actual simultaneous buffers, especially when used inside PCG.

For low-resolution Rheon scenes, a simpler diagonal-PCG solve may outperform a complex hierarchy once setup and changing geometry are included. Measure complete solve time to a physical tolerance. Multigrid is a promising mechanism, not a required optimization before evidence.

Draft checkpoint: qualify a model's pressure before using it

Unmerged [PR49](#), inspected at `b8ca87cab7f74f64c1686714a33e843fbb178c3d`, introduces an opt-in [pressure-provider workspace](#). The native modal model evaluates a frozen learned transform/gain checkpoint in one fixed 32^3 sealed unit box, with density 1.25 kg/m^3 and $dt = 1/60 \text{ s}$. The [draft's model provenance and qualification contract](#) links the exact upstream source/checkpoint and separates its historical rollout measurements from this native pressure-stage experiment.

Predict the response when density or grid shape changes: the model declines and the matching numerical reference is used. For an admitted request, every output must be finite and gauge-consistent, and the workspace recomputes the full ungauged residual, including the gauge cell. A declined or rejected candidate uses a fresh zero-start numerical reference; cancellation aborts without fallback. The caller must still check divergence of its actual corrected $f32$ velocity before publication. Pressure acceptance alone is not a complete step contract.

The SHA256-verified export and trusted checkpoint supply artifact provenance; the loader's FNV corruption check is not cryptographic authentication. Timing columns separate inference, qualification and fallback, while excluding model construction and caller velocity correction. Neither an accepted prediction nor a historical speed ratio establishes faster full trajectories, general physical accuracy, GPU support or arbitrary-geometry learning. Defaults and PR48's transport facade are not automatically switched to this draft provider.

Two disconnected pressure regions

Start with two regions that cannot exchange fluid, each containing two cells. The embedded example recomputes an all-Neumann pressure correction with unit cross-section, density 1 kg/m^3 and $dt=1 \text{ s}$. Cell widths encode actual volumes. For volumes $(1,3,2,2) \text{ m}^3$, both centre distances are 2 m and both face weights are $1/2$. Compatible integrated data $b=(1,-1,2,-2)$ gives fixed-left pressures $(0,-2,0,-4) \text{ Pa}$ and internal velocity corrections $(1,2) \text{ m/s}$.

Predict what adding the same pressure to both cells of region A will change. The scalar pressure bars move, but that region's pressure difference and face correction do not. Bars use the display depth as a scalar axis; they are not force vectors. The separated boxes represent disconnected regions, not a fluid simulation with hidden exchange across the gap.

Global balance can hide component errors

Now select $b=(1,0,-1,0)$. Its global sum is zero, but its two component sums are 1 and -1. The original equations cannot both be satisfied in either region. Fixing each left cell and solving only the right rows returns zero pressures. The retained-row residual is zero, yet the full residual is $(1,0,-1,0)$ and actual divergence is $(-1,0,1/2,0)$ /s. Read the rejected status before trusting a small reduced residual. A gauge cannot repair incompatible boundary data.

Fixed-cell and mean-zero gauges

For compatible data, fixing one pressure per region or subtracting each region's arithmetic pressure mean changes only its representative. The pressures become $(1,-1,2,-2)$ Pa with arithmetic means zero; the face corrections stay $(1,2)$ m/s. Fixing one cell in the entire disconnected system is insufficient. These are algebraic gauges, not prescribed physical pressure boundaries.

Volume-weighted pressure gauges

A volume mean uses V-weighted pressure. For unequal cells it gives $(3/2,-1/2,2,-2)$ Pa, which differs from the arithmetic mean while preserving all within-region pressure jumps. Minimum-norm language needs its inner product: Euclidean and volume-weighted norms choose different representatives. The offset controls deliberately shift away from the selected zero constraint for comparison. Pressure differences across independent regions remain undetermined here.

Projecting a source changes the problem

For incompatible b , the explicit projection comparison changes it to $(1/2,-1/2,-1/2,1/2)$. That modified system has a solution. Its original-system residual is still $(1/2,1/2,-1/2,-1/2)$, so the original boundary data remains rejected. The source-change readout is part of the result: projection is not a silent gauge choice or a physically justified repair.

Gauge shifts and boundary work

The outward-flux pairing includes external work:

$$p_L(q + Q_L) + p_R(-q + Q_R) = -(p_R - p_L)q + p_L Q_L + p_R Q_R.$$

A common shift c changes boundary work by $c(Q_L + Q_R)$, while the internal term is unchanged. It vanishes for each balanced region. With incompatible data, region offsets $(2,-2)$ Pa produce -4 W of boundary pairing even though the global flux sum is zero. This exposes inconsistent data, not an absolute-pressure prediction for a valid closed flow.

A pressure error is not a gauge

Perturb only the right pressure in region A by 1/4 Pa. Compatible unequal cells then have residual (1/8,-1/8,0,0), divergence (-1/8,1/24,0,0) /s and internal velocity correction 7/8 m/s in A. The physical pressure difference changed; zero total flux does not make this locally valid. This is an intentional finite perturbation, not a floating-point model.

► Exact gauge contracts and their limits

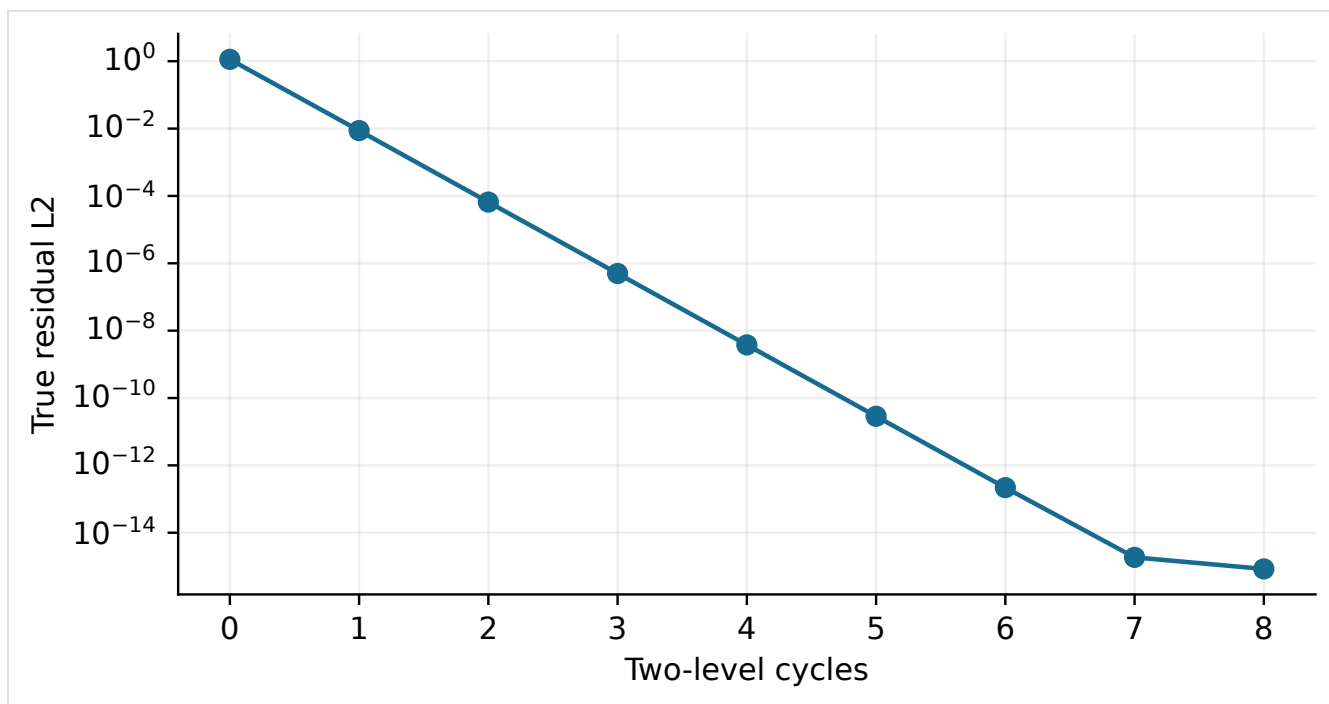
Two unknowns and a real iteration

Use $K = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$, corresponding to a two-cell Dirichlet stencil. From $x = 0$, each control-selected iteration performs $x \leftarrow x + \alpha(b - Kx)/2$ using the complete old vector. This is simultaneous weighted Jacobi, rather than a saved-field blend. For $b = (1, 0)$, the exact solution is $(2/3, 1/3)$. At $\alpha = 1$ after four iterations, $x = (5/8, 5/16)$, the true residual is $(1/16, 0)$, and error energy $e^T K e$ is $1/384$. The readout recomputes $b - Kx$. At $\alpha = 1/2$ with alternating b after eight iterations, each residual magnitude is $1/65536$. Its fixed-scale arrow is nearly invisible; the numerical readout remains available. A minimum visible shaft would exaggerate that residual and violate the scale.

A positive matrix with an unstable iteration

The matrix energy is $(x-y)^2 + x^2 + y^2$ and is positive away from zero. Smooth and alternating error modes nevertheless have iteration factors $1-\alpha/2$ and $1-3\alpha/2$. Both contract only when $0 < \alpha < 4/3$. Select $\alpha = 3/2$ and alternating b : the alternating error factor is $-5/4$ and its energy grows by $25/16$ each iteration. Matrix positivity alone does not choose a stable Jacobi relaxation. [Shewchuk, sections 5.2–5.3](#) explains the splitting and spectral-radius criterion; this matrix and its closed-form error modes are independently derived here. Production Krylov methods additionally need suitable operator/preconditioner properties and fresh full residuals; no general CG or multigrid result is inferred.

► Practical study proof scope



Original reference illustration for 5 Solving the pressure system

6 Transport and the price of stability

A passive scalar satisfies $\partial_t q + \mathbf{u} \cdot \nabla q = 0$ for incompressible source-free flow. Along an exact characteristic it is constant. Semi-Lagrangian transport traces backward from an arrival sample and interpolates the old field. Stam's stable-fluids construction popularized this in interactive graphics [S1]. Stability must name a norm and interpolation rule.

Backtracing

First-order tracing uses $\mathbf{x}_d = \mathbf{x} - \Delta t \mathbf{u}(\mathbf{x})$. A midpoint estimate first samples velocity halfway along that estimate, then traces with the midpoint velocity. Each staggered component uses its own lattice offset; the scalar has another offset.

Trilinear interpolation is a convex combination of eight values when fractional coordinates lie in $[0, 1]$. Nonnegative weights summing to one give

$$\min_i q_i \leq \sum_i w_i q_i \leq \max_i q_i.$$

Convex_sum_bounds proves this finite real statement. Repeated bounded interpolation cannot leave the original global range if boundaries and sources obey it. This does not imply conservation of all sample values.

Row sums and column sums

For $q^+ = Tq$, convex interpolation means nonnegative entries and unit row sums. Conservation of an unweighted total requires unit column sums. For example,

$$T = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 1/2 & 0 \\ 1/2 & 1/2 & 0 \end{bmatrix}$$

maps $(1, 0, 0)$ to $(1/2, 1/2, 1/2)$. Outputs are bounded but total rises from 1 to 1.5. The exact rational arithmetic is checked in Lean. This is a matrix counterexample to an implication, not a claim that this matrix comes from smooth incompressible periodic flow.

Constant translation on a periodic uniform one-dimensional grid happens to give a doubly stochastic interpolation matrix. The companion's box test verifies that special case, not general semi-Lagrangian conservation. Variable velocity and irregular boundaries need separate evidence.

Numerical diffusion

For positive constant speed and Courant number $c = u\Delta t/h$ in $[0, 1]$,

$$q_j^+ = (1 - c)q_j + cq_{j-1}.$$

A Fourier mode has amplification $g = (1 - c) + ce^{-i\theta}$, hence

$$|g|^2 = 1 - 2c(1 - c)(1 - \cos \theta).$$

Amplitude generally decays. Taylor expansion gives leading artificial diffusion $uh(1 - c)/2$ under fixed-Courant refinement. A bounded solver can therefore erase thin jets or smear silhouettes. Arbitrarily large stable steps also degrade trajectory accuracy and obstacle interaction.

Conservative flux updates

Finite volume computes one shared face flux and applies opposite signs:

$$q_j^+ = q_j - \frac{\Delta t}{h}(F_{j+1/2} - F_{j-1/2}).$$

Summation over periodic cells cancels all internal fluxes. Positive-speed upwind flux $F_{j+1/2} = uq_j$ preserves positivity for $0 \leq c \leq 1$, proved by `upwind_positive`. At $c = 3/2$, $q_j = 1$, $q_{j-1} = 0$, the new value is $-1/2$, also formally checked. Conservation does not imply positivity.

The field determines the contract. A smoke appearance tracer may tolerate measured dissipation. A liquid fraction needs bounded conservative transport and interface geometry. A signed-distance field needs surface-position analysis. One generic quality parameter cannot hide these differences.

Component aware trilinear sampling

For an x-face field, convert world position to lattice coordinates $s = ((x - o_x)/h_x, (y - o_y)/h_y - 1/2, (z - o_z)/h_z - 1/2)$. For a y-face, subtract half from x and z; for a z-face, subtract half from x and y. For cell-centered scalars subtract half from all coordinates.

Let $a = \lfloor s \rfloor$ and $t = s - a$. The eight weights are products of either t_d or $1 - t_d$. For corner $\epsilon \in \{0, 1\}^3$,

$$w_\epsilon = \prod_{d=1}^3 (\epsilon_d t_d + (1 - \epsilon_d)(1 - t_d)).$$

Their sum factors into $\prod_d (t_d + 1 - t_d) = 1$. Nonnegativity follows when each fractional coordinate lies in $[0, 1]$. This derivation explains the exact hypotheses of the interpolation proof. Boundary remapping must preserve or deliberately replace the stencil; simply dropping invalid corners breaks partition of unity.

An affine scalar $q(x, y, z) = a + bx + cy + dz$ is reproduced exactly by trilinear interpolation on a complete stencil in real arithmetic. The expanded companion tests this on all four lattice offsets using independently evaluated world positions. It then deliberately uses the wrong offset for an x-face field to show the expected half-cell bias. This fixture catches a defect that constant-preservation tests miss.

Sampling boundaries

Periodic sampling wraps integer indices modulo the component's periodic extent, accounting for duplicated boundary faces if the storage convention has them. A closed-wall sampler uses its declared ghost extension. A clamped texture sampler repeats edge values; it is not automatically a no-slip boundary condition. An out-of-domain trace at an open boundary needs a prescribed exterior value or a validity failure.

Return a sample together with validity information when the method needs it. Corrected advection needs to know whether forward and reverse stencils are valid and reversible. A scalar value alone cannot communicate that a trace hit a wall or crossed a source discontinuity.

Trace integrators

For an autonomous frozen velocity field, midpoint backtracing is

$$k_1 = \mathbf{u}(\mathbf{x}), \quad k_2 = \mathbf{u}(\mathbf{x} - \tfrac{1}{2}\Delta t k_1), \quad \mathbf{x}_d = \mathbf{x} - \Delta t k_2.$$

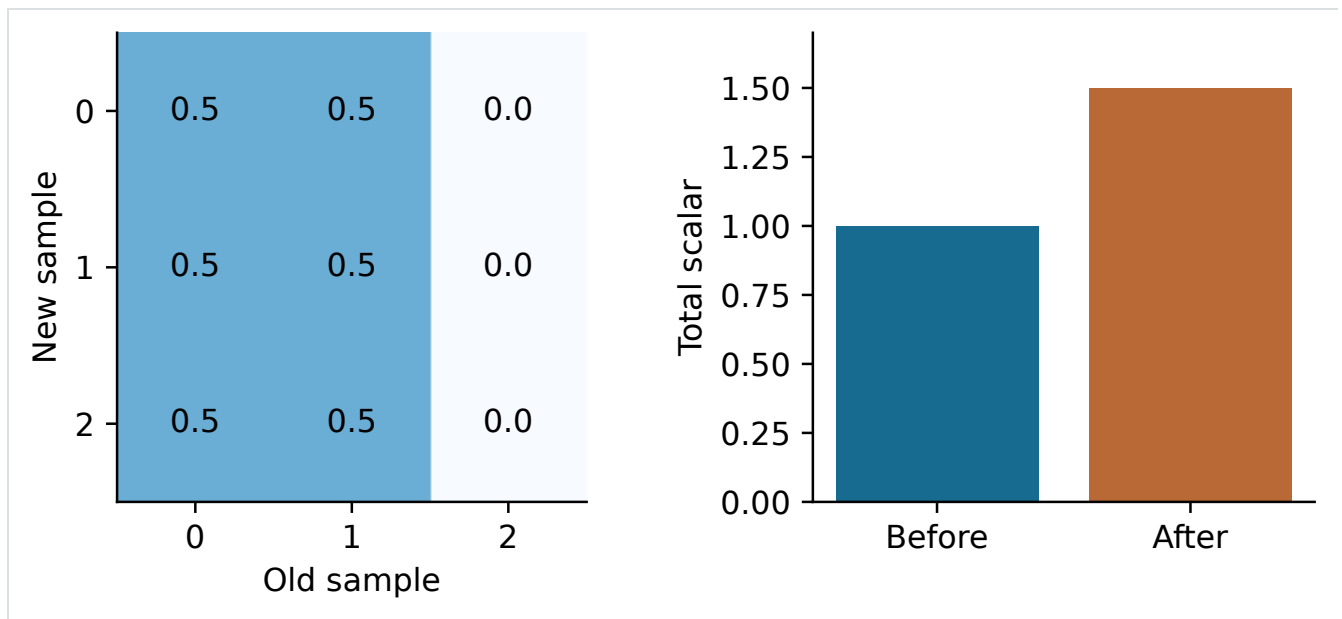
A higher-order trace reduces ODE integration error but cannot improve interpolation order or unresolved velocity detail. For time-varying velocity, temporal interpolation between known levels is another approximation. Name which field drives tracing: old velocity, provisional velocity or projected velocity. These choices change the splitting.

A solid-rotation fixture with $\mathbf{u} = (-\omega y, \omega x, 0)$ has an analytic rotation trajectory. Compare departure points against the exact rotation over one step and over one period. A translation fixture cannot distinguish Euler from midpoint tracing because both are exact for constant velocity. The companion includes the rotation test specifically to expose that difference.

Failure fixtures

Use a constant field to test partition of unity, an affine field to test offsets, a sinusoid to measure dissipation, a top-hat to reveal overshoot and limiter behavior, and a segment crossing a thin wall to test collision handling. Add a trace ending exactly on a lattice boundary, where floor and integer conversion policies matter.

For corrected transport, record the number of reverted samples and the maximum unbounded overshoot before reversion. A zero overshoot on a smooth wave is not evidence for discontinuities. For conservative transport, compare integrated amount separately from min/max bounds. These fixtures should remain distinct because each isolates a different contract.



Original reference illustration for 6 Transport and the price of stability

7 Higher order transport with bounded failure

Linear interpolation is robust but dissipative. Let $S_{\Delta t}$ denote a basic semi-Lagrangian step. A forward-backward correction computes

$$q_f = S_{\Delta t} q, \quad q_b = S_{-\Delta t} q_f, \quad q_c = q_f + \frac{1}{2}(q - q_b).$$

This is the teaching companion's precise formula. Selle and collaborators analyze an unconditionally stable MacCormack construction [S6]. Several related formulas share that informal label; specify the exact correction and limiter rather than relying on the name.

Overshoot and limiting

Forward and reverse interpolation average values. Their composition loses detail; adding half the estimated loss sharpens smooth fields. Subtraction can make effective weights negative, so the convex maximum principle no longer applies. Sharp interfaces can acquire negative concentration or excessive values.

The companion reverts an out-of-range corrected value to the forward value. Bounds come from the old departure interpolation stencil. Clamping is another policy with different truncation behavior. Neither automatically conserves scalar total. The limiter belongs in the method specification and saved configuration.

The inverse trace here reuses one velocity field. It does not invert a time-dependent physical trajectory. Near collisions, emitters and open boundaries, the paths may not represent reversible transport. Mark invalid traces and revert instead of treating arbitrary boundary values as error estimates.

Executed refinement

The companion advects $q(x) = \sin(2\pi x)$ once around a periodic unit interval at Courant number 0.5. At 32, 64, 128 and 256 cells, linear RMS errors are 0.187922, 0.101090, 0.052478 and 0.026743. Limited corrected errors are 0.037630, 0.009022, 0.003211 and 0.001022. These are executed measurements, not assumed order labels.

The corrected scheme retains more amplitude. Refinement ratios vary because the limiter activates near extrema. Do not claim uniform second-order convergence from the unbounded formula alone. Compute $\log_2(E_h/E_{h/2})$, inspect limiter activity and include nonsmooth data. A different step-size sequence asks a different convergence question.

Velocity and conservation

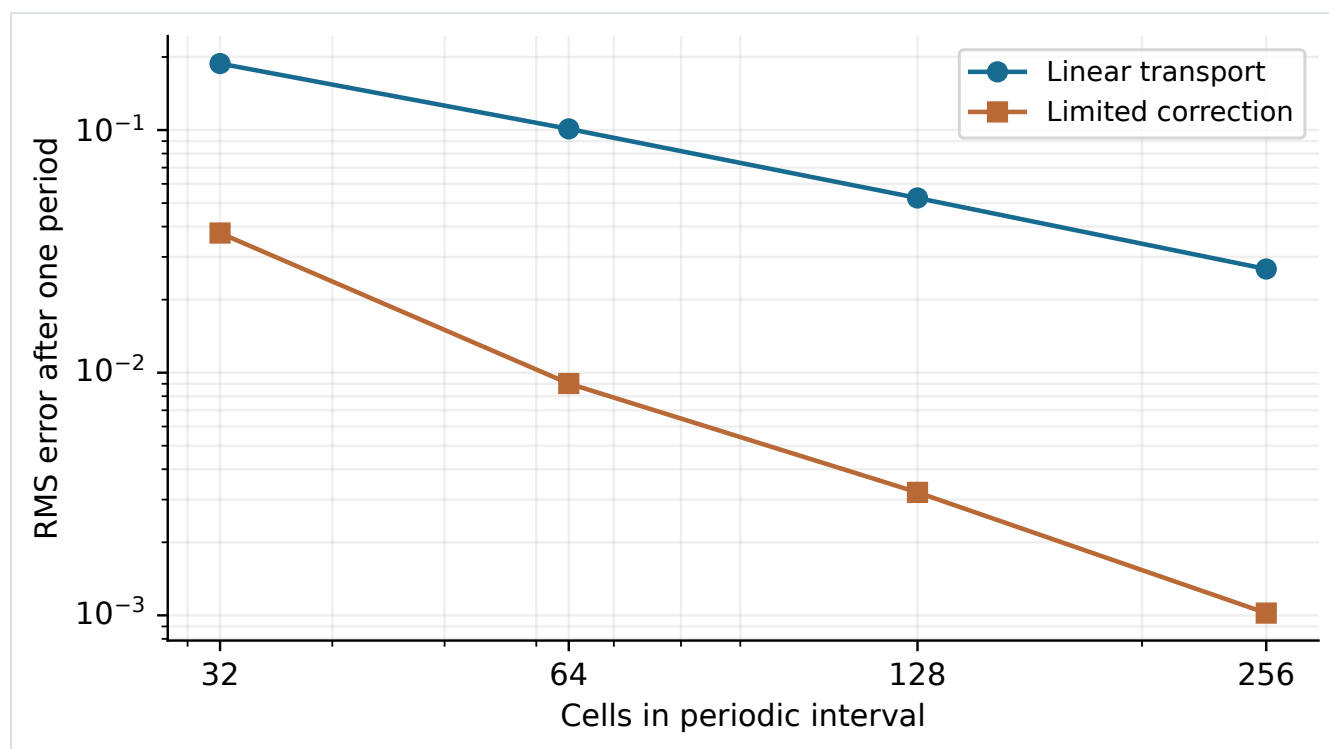
For material volume, use a bounded conservative finite-volume or geometric volume-of-fluid construction. A corrected sampler is an appearance tool unless additional conservation structure is established. Sharp plume images are not mass or momentum evidence.

Velocity components live on different grids and have no scalar positivity requirement. Componentwise limiting may suppress meaningful extrema. Projection removes divergence but cannot undo all momentum or energy errors from interpolation. Measure circulation, energy and vortex displacement as well as divergence. Prescribed-velocity tracer accuracy does not prove self-advected vortex accuracy.

Buffer dependencies

Old data must survive both forward and backward passes; forward data survives correction. Overwriting it early creates an order-dependent in-place scheme. Per-component processing lowers peak memory only if velocity reconstruction does not read an already overwritten component. Immutable input views and separate output slices expose this dependency directly.

The implementation should count fallback samples and invalid backtraces. A method that reverts over most of the domain is functionally close to the first-order baseline, even if its configuration says high order. Report observed behavior rather than only the selected option.



Original reference illustration for 7 Higher order transport with bounded failure

8 Boundaries and obstacles

On a wall moving at $\mathbf{u}_s$, impermeability requires $(\mathbf{u} - \mathbf{u}_s) \cdot \mathbf{n} = 0$. No-slip also matches tangential velocity; free-slip does not. These are distinct models. Prescribed normal wall velocity enters the divergence right-hand side.

Atmospheric-pressure openings instead fix pressure and allow flow. Combining both conditions on one face can overconstrain the system.

Binary geometry

A first obstacle model can classify cells and block faces. The resulting staircase approximates a curved boundary but changes effective volume and creates grid-aligned features. Refinement can move the represented wall discontinuously as classifications change. An accurate linear solve cannot remove geometric error.

A boundary snapshot should have one version consumed by advection, pressure and rendering for a step. If geometry changes during pressure solution, discard or restart the stale candidate rather than publish velocity corrected for a different domain.

Fractional faces

An open area fraction a_e and fluid cell-volume fraction θ_i produce divergence terms proportional to $a_e h^2 / (\theta_i h^3)$. Tiny θ_i creates small-cell stiffness. Clamping coefficients changes the model; removing cells changes topology. Either can be an explicit approximation, not a free fix.

Batty, Bertails and Bridson develop variational solid-fluid coupling on coarse Cartesian grids [S7]. The structural lesson is to choose gradient, divergence and energy together. Rheon's arbitrary-matrix theorem does not establish cut-cell quadrature.

For the linear homogeneous operator D_0 , with cell metric M_c and face metric M_f , require

$$\langle p, D_0 \mathbf{u} \rangle_{M_c} = -\langle Gp, \mathbf{u} \rangle_{M_f}, \quad D_0 = -M_c^{-1} G^T M_f.$$

An attractive divergence formula paired with an unrelated gradient can yield an inconsistent pressure operator. Test this linear weighted adjoint identity with random vectors on every supported geometry class after prescribed boundary flux is separated. For the metric construction in Chapter 3, $M_c = V$, $M_f = SL$, and $D_0 = -V^{-1}BS$. The full physical divergence is affine: $D(u) = D_0 u + V^{-1}Q$. Therefore its identity includes boundary work:

$$\langle p, D(u) \rangle_V = -\langle Gp, u \rangle_{SL} + p^T Q.$$

Omitting the final term is valid only when its value is zero, for example homogeneous prescribed flux. The companion includes nonzero Q to reject that omission. The existing Lean arbitrary-B theorem proves the linear algebraic identity, not moving-boundary assembly.

Trajectory collision

A departure point outside solid material does not prove the path avoided it. A long trace can cross a thin wall and emerge on the far side. Intersect straight traces with geometry and apply the declared rule at the earliest hit. Multi-stage curved traces need compatible approximation. Limiting travel helps but is not collision proof.

Boundary interpolation must define its extension. Constant extrapolation, mirrored normal velocity and no-slip ghost values differ. Do not bury them in a texture sampler mode. Check samples on both sides of walls and at corners.

Diagnostic sequence

Test a sealed box with uniform translation, then stationary fluid under gravity with hydrostatic pressure, then moving obstacles with displaced-volume and wall-work accounting. A wall thinner than one cell must be resolved or explicitly unsupported rather than silently leak.

Report flux by boundary category and closed component. Long-term leaks often originate in one inconsistent face update. More pressure iterations help only if pressure residual is responsible; topology, geometry and buffer timing remain separate suspects.

Two cells and boundary work

Predict the power before changing the controls. The embedded example uses two unit cells, a shared opening of area A , prescribed face speed u and balanced external outward fluxes $(q, -q)$. The shared flux is $\phi = A u$. With unit volumes, the divergences are $(\phi + q, -\phi - q)$. They sum to zero, even if neither is zero. The pressure test values are prescribed probes, not a solved pressure field.

At $A = 1/2 \text{ m}^2$, $u = 1 \text{ m/s}$, $q = 1/2 \text{ m}^3/\text{s}$ and $p = (1, 3) \text{ Pa}$, the pressure-divergence pairing is -2 W . The interior term $-(p_R - p_L)\phi$ is only -1 W . The remaining -1 W is $(p_L - p_R)q$, the prescribed boundary work. Set $q = 0$ to recover the homogeneous identity; do not drop the term for general data.

Balanced flow is not zero boundary work

Now $q = -1/2 \text{ m}^3/\text{s}$ cancels each cell's divergence. Interior work is -1 W and boundary work $+1 \text{ W}$. The vanishing total does not imply vanishing contributions. The same shared transfer must be used with opposite signs in both cell ledgers. Area changes flux; it does not change the velocity arrow's fixed scale.

The [research dossier](#) derives units, signs, source applicability and the 81-state domain. This is instantaneous finite algebra, not arbitrary mesh assembly, a moving wall or an accepted CFD step.

► **Exact finite proof and limits**

Skew samples and normal gradients

At samples $L=(-1/2,0)$, $R=(1/2,1/2)$, $T=(-1/2,1)$ m, take $p=x+y+2$ Pa. The pressures are $(3/2,3,5/2)$. The two-point contribution along the centre vector predicts a face-normal gradient of $6/5$ Pa/m although the exact x gradient is 1. The rendered patch has normal x ; its arrows encode pressure gradient, not velocity. Moving the skew control changes the actual sample geometry.

An affine correction with a third sample

Use the third sample to reconstruct $b=(p_T-p_L)/t$ and then $a=p_R-p_L-s b$. Adding the missing normal component corrects $6/5$ by $-1/5$ to give 1 Pa/m. The linearly interpolated face value $9/4$ Pa also needs a $-1/4$ Pa skew correction. Two samples alone do not determine both gradient components. This finite affine result is not a general nonlinear mesh reconstruction or a collision-mesh reader.

► Reconstruction and coupling proof scope

9 Liquid interfaces and volume

A liquid needs an interface representation beyond velocity. A level set stores ϕ , negative inside, with zero contour at the surface. A signed-distance field has $|\nabla\phi| = 1$ nearby. Osher and Sethian introduced a Hamilton-Jacobi framework for evolving fronts [S8]. Merging and splitting are natural, but interpolation and reinitialization can move the contour.

Reinitialization

Transport satisfies $\partial_t\phi + \mathbf{u} \cdot \nabla\phi = 0$. A common distance-restoring evolution is

$$\partial_\tau\phi + \text{sgn}(\phi_0)(|\nabla\phi| - 1) = 0.$$

Artificial time τ seeks distance-like values with fixed reference sign. Discretization can shift the zero set, especially in thin sheets. Reinitialization frequency, stencil, sign smoothing and termination belong to the method specification. Smooth appearance does not imply correct volume.

A narrow band must cover interpolation support and travel before rebuilding. Outside values need valid classification and reconstruction when activated. Invalid far-field sentinels must not be interpolated into the surface.

Pressure at the free surface

Neglecting air dynamics, surface tension and liquid normal viscous stress, set surface pressure to atmospheric zero. If normal viscous stress is retained, this is only a declared split approximation; a coupled traction condition is required as discussed in Chapter 10 and Batty-Bridson Section 3 [S11]. A pressure connection with interface distance αh uses that shorter distance, increasing its coefficient as α shrinks. A minimum-distance policy changes pressure and interface error and must be evaluated.

Velocity needs extension into a limited air band for transport. The pressure-unknown mask, velocity-validity mask and distance-band mask are distinct. Extending velocity does not mean the whole air region solves the liquid equations.

Volume fractions

A fraction $f_i \in [0, 1]$ gives represented volume

$$V_h = \sum_i f_i V_i.$$

Shared liquid-volume fluxes with opposite signs preserve global volume except for sources and boundaries. Hirt and Nichols established the volume-of-fluid approach [S9]. Geometric reconstruction estimates a local interface to determine swept liquid volume; it is substantially more work than advecting a color scalar.

A cell fraction 0.3 and Courant number 0.5 illustrate the risk: an outgoing flux selected without interface geometry can remove more liquid than exists. Bounding available volume must coordinate faces in multiple dimensions. Clamping final fractions can restore range while destroying total volume. Preserve both by construction or disclose the correction.

Particle correction and extraction

Particle level-set methods use particles carrying side-of-interface information to correct local errors [S10]. Reseeding, radii and collision handling add state and memory. They do not make level-set volume exact.

Select a liquid method from acceptance metrics. Level sets simplify smooth geometry but need drift budgets. Volume fractions expose conservation but need reconstruction. Hybrids increase verification burden. Finish the baseline grid contracts before committing to a liquid branch.

Mesh extraction adds its own error. Edge crossings and ambiguous cases need consistent rules across cells. Normals use physical spacing and safe treatment of small gradients. Validate spheres at several subcell offsets, thin sheets and merging components. Compare volume, surface distance and rasterized silhouette separately. A guidance mesh need not be manufacturing-grade, but its promised topology must be explicit.

Piecewise linear interface reconstruction

A volume-fraction method needs geometry inside partially filled cells. A piecewise linear interface reconstruction chooses a unit normal $\mathbf{n}$ and plane offset d so the liquid region inside a cell is $\{x : \mathbf{n} \cdot x \leq d\}$ and its volume matches $f_i V_i$. For this liquid-side inequality, outward $\mathbf{n}$ is minus the normalized fraction gradient, since f decreases toward air. For a negative-inside level set, outward $\mathbf{n}$ is plus the normalized phi gradient. Use a resolved neighborhood estimate; when its norm is too small, return an unresolved normal or use a documented geometry-derived fallback, never normalize zero. The offset fraction-step fixture checks both signs. Then solve for d by a monotone volume function.

On a unit cube with $\mathbf{n}$ along positive x , the solution is simply $d = f_i$ for fractions in $[0, 1]$. For a general normal, clip the cube polyhedron against the plane and compute its volume, or use a verified analytic formula. Bisection is robust because volume increases monotonically with d . Bracket d using the minimum and maximum plane projections of cube vertices. Stop according to volume and geometric tolerances, not an arbitrary iteration count alone.

Transport then clips the reconstructed liquid region against the swept face volume over the step. In a simple axis-aligned constant-speed case, this is a slab of width $u\Delta t$. In multiple dimensions, directional splitting and unsplit flux constructions handle corner transport differently. A raw sum of independently computed face outflows can exceed available liquid. A production implementation must select one coherent bounded-conservative method and test its multidimensional flux accounting.

The expanded companion only checks the axis-aligned slab case: a cell filled to fraction 0.3 with a positive swept width 0.5 sends the geometrically intersected amount under an explicitly positioned interface. It does not claim a complete arbitrary-normal PLIC implementation. General polyhedral clipping, orientation robustness and multidimensional boundedness remain implementation work.

Curvature from a level set

With outward normal $\mathbf{n} = \nabla\phi/|\nabla\phi|$, mean-curvature sum is $\kappa = \nabla \cdot \mathbf{n}$. In three dimensions a sphere of radius R has $\kappa = 2/R$ under this convention. When air viscous stress is negligible, normal traction balance is $p_{\text{liquid}} - p_{\text{air}} = \sigma\kappa + \mathbf{n}^T \tau_{\text{liquid}} \mathbf{n}$. The simpler Laplace jump $\sigma\kappa$ additionally neglects normal liquid viscous stress. Tangential traction

still requires its own condition. See Chapter 10 and [S11] Section 3; a pressure-only split must declare its approximation. Reversing the sign convention for phi reverses normal and curvature, so the pressure-jump sign must change consistently.

A direct finite-difference curvature formula is

$$\kappa = \frac{|\nabla\phi|^2 \text{tr}(H_\phi) - (\nabla\phi)^T H_\phi \nabla\phi}{|\nabla\phi|^3}.$$

It requires first and second derivatives and becomes ill-conditioned when gradient magnitude is small. Regularizing the denominator changes the curvature estimate. Near underresolved sheets or merged surfaces, a smooth signed-distance assumption may fail even if every value is finite.

The companion evaluates a sphere distance field and central-difference curvature at a known surface point over decreasing h. It compares against $2/R$. This is a derivative-stencil fixture, not a moving-droplet surface-tension validation. Production tests must additionally measure parasitic currents in a stationary droplet, pressure jump and oscillation behavior.

Surface tension time scales

Balance capillary pressure σ/h against inertial acceleration over length h. Acceleration scale is $\sigma/(\rho h^2)$; moving distance h under that acceleration gives time scale $\sqrt{\rho h^3/\sigma}$. This dimensional derivation explains severe restrictions as resolution increases. The stability constant depends on the actual discretization and integration method.

For a 1 cm feature scale, density 1000 kg per cubic metre and surface tension 0.072 newtons per metre, the dimensional time scale is approximately 0.118 seconds. At a 1 mm scale it is about 0.00373 seconds. These are scale estimates, not accepted time-step limits for an unimplemented solver.

Implicit tension may relax a stability restriction but introduces coupled geometry or linearization errors. It does not eliminate the need to resolve curvature or validate the interface. Likewise, increasing surface tension to suppress noise changes the material and can alter the desired guidance shape.

Volume correction as a declared operation

A level-set volume correction sometimes shifts phi by a constant until a target volume is restored. This moves the interface everywhere, including regions already accurate. It can repair a scalar volume metric while damaging local positional accuracy. If used, record shift magnitude, affected volume and resulting silhouette displacement.

A more informative acceptance report retains both volume drift and a spatial surface metric. Two shapes can have equal volume and completely different placement. Rheon's guidance purpose makes that distinction especially important.

A shared-face experiment

Start with one unit-metre cube and an empty receiving cube. Prescribe positive x transport over one interval, with swept width $c = u\Delta t/h$ between zero and one. The outgoing region is $[1 - c, 1] \times [0, 1]^2$. The interface position, not just the liquid fraction, determines how much liquid intersects this slab.

The embedded example places the same fraction f in either $[0, f] \times [0, 1]^2$ or $[1 - f, 1] \times [0, 1]^2$. The two transferred volumes, in cubic metres for these unit cubes, are

$$Q_L = \max(0, f + c - 1), \quad Q_R = \min(f, c).$$

Predict before switching **Liquid location**: with $f = 0.3$ and $c = 0.5$, the left-filled cell transfers zero, while the right-filled cell transfers 0.3 m^3 . A uniform-mixture estimate $cf = 0.15$ cannot distinguish these interfaces. It can preserve a shared ledger while giving a different interface transport. Teal marks the initial liquid, orange the swept region, and the dark intersection the outgoing liquid. Orbit the cube to inspect their spatial relationship. Lower cubes encode the resulting amounts with fill heights; they do not claim reconstructed post-step geometry.

The receiver starts empty. Subtract the same Q from the donor and add it to the receiver: $(f - Q) + Q = f$. Now inspect the deliberately incorrect full-slab transfer $Q = c$: the amounts become -0.2 and 0.5 m^3 . Independently clamping them gives zero and 0.5 m^3 , creating 0.2 m^3 . Conservation, bounded fractions and geometrically correct transport are three separate questions.

This is a one-face geometric/ledger experiment, not a closed-box divergence-free velocity solution. Two liquid-containing cells, several outgoing faces, inflow, non-unit cell volumes and arbitrary normals require additional accounting. The receiver's capacity must also be respected. Slider width up to one is an intersection domain, not a stability recommendation for a multidimensional scheme.

Research connection and native scope

The official [Basilisk VOF implementation](#) reconstructs a plane before evaluating liquid in a swept face region. Its one-dimensional sweep includes a compression contribution; the multidimensional construction follows Weymouth and Yue's conservation treatment and states an exactly non-divergent velocity condition. Its documented CFL restriction is 0.5. These details explain why isolated face cancellation does not establish bounded, conservative multidimensional VOF. See its `sweep_x` and `vof_advection` implementations (consulted 2026-10-13), alongside Hirt and Nichols [S9]. This lesson implements neither complete scheme.

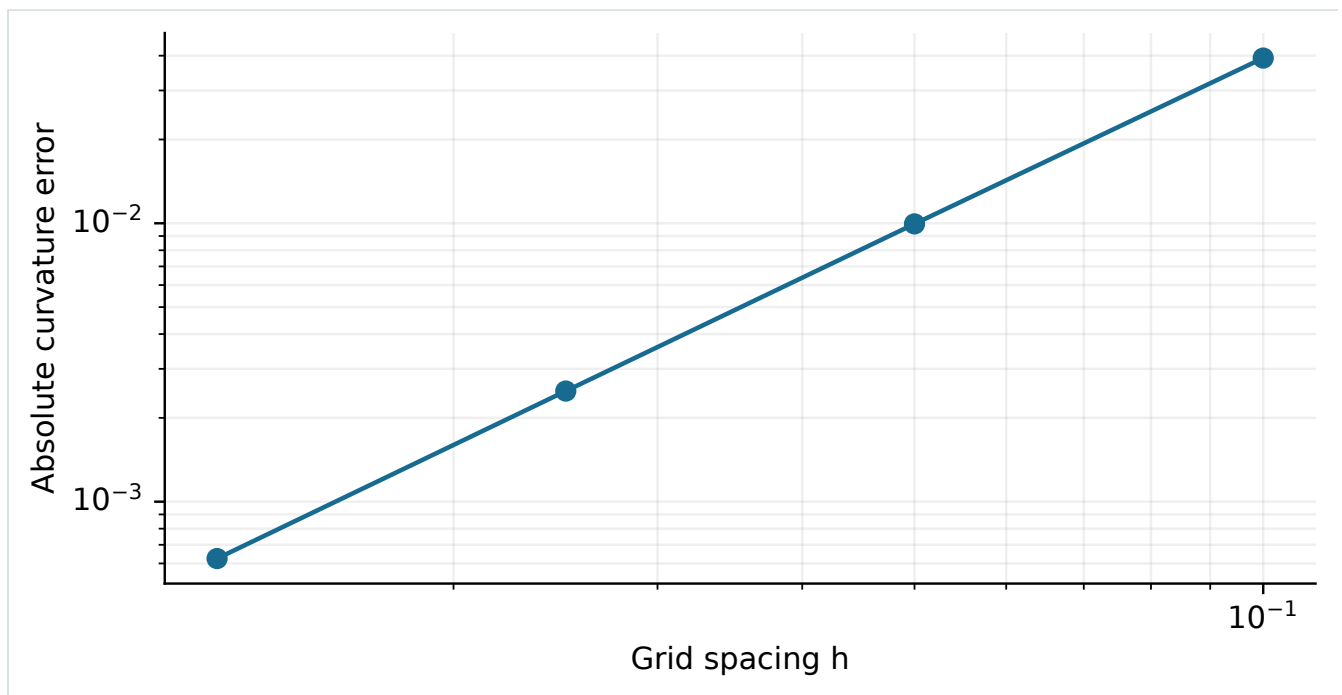
Rheon's existing [liquid-volume owner](#) and [liquid step](#) remain separate native code. This browser lesson does not invoke them, change their admission rules, or qualify geometric reconstruction in Rust. Its independent equations are visible in [volume_reference.js](#).

Checked volume-flux proof bridge

[VolumeFlux.lean](#) states exact-real properties with explicit hypotheses. For $0 \leq f, c \leq 1$, the left slab flux is nonnegative and no greater than either f or c . The right slab has the same bounds under nonnegative f and c . A shared transfer cancels from the combined ledger for any real donor, receiver and transfer amount. Bounded post-transfer

amounts additionally require a nonnegative transfer, sufficient donor availability, donor amount at most one, nonnegative receiver amount and receiver capacity. Exact rational witnesses distinguish the two placements and show clamping's 0.2 m^3 volume creation.

These statements do not prove that JavaScript implements the geometry, that IEEE arithmetic is exact, or that the native liquid algorithm converges. The source inventory and kernel qualification status are recorded separately in the [coverage ledger](#); numerical comparisons and browser control checks supply different evidence. Proof text is collapsed by default in the interactive chapter.



Original reference illustration for 9 Liquid interfaces and volume

10 Forces viscosity and material behavior

Gravity is acceleration: $\mathbf{u}^* = \mathbf{u} + \Delta t \mathbf{g}$ needs no density multiplier. Pressure and stress use density according to the selected formulation. Configuration and types should distinguish world lengths, grid coordinates and dimensionless coefficients.

Hydrostatic balance

With upward z and gravity $-\mathbf{g}\mathbf{e}_z$, rest requires $\partial_z p = -\rho g$. A linear pressure field with face differences $-\rho g h$ cancels gravity in ideal discrete arithmetic when boundaries agree. Measure speed, divergence and surface drift over many steps instead of judging whether a tank looks still.

Diffusion

One-dimensional explicit diffusion is

$$q_j^+ = q_j + r(q_{j-1} - 2q_j + q_{j+1}), \quad r = \nu \Delta t / h^2.$$

Nonnegative coefficients require $0 \leq r \leq 1/2$, established by `explicit_diffusion_positive`. In isotropic three-dimensional seven-point diffusion, the center coefficient is $1 - 6r$, giving the analogous restriction $r \leq 1/6$. That extension is derived here, not in the current Lean inventory.

Implicit Euler solves

$$(I + \nu \Delta t L) q^+ = q$$

for a positive semidefinite negative Laplacian L . An eigenmode of eigenvalue $\lambda \geq 0$ is multiplied by $1/(1 + \nu \Delta t \lambda)$, between zero and one. Large steps still change decay accuracy; boundaries remain essential.

The companion's 64-cell periodic sine mode uses $\nu = 0.01$, time step 0.001 and 100 steps. Final amplitude 0.9613286500 differs from continuum 0.9612907007 by about 3.79×10^{-5} . This combines spatial and temporal errors in a scalar oracle, not a complete viscous-fluid implementation.

Stress and constitutive choices

Newtonian stress is

$$\tau = 2\mu D(\mathbf{u}), \quad D(\mathbf{u}) = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T).$$

Replacing its divergence by $\mu \nabla^2 \mathbf{u}$ uses constant viscosity and incompressibility with compatible boundary terms. Variable viscosity and free surfaces can need coupled stresses. Batty and Bridson's variable-coefficient Stokes work is a primary reference [S11]. State the validity of a simpler componentwise approximation.

Smoke buoyancy may depend on temperature deviation and a density-like tracer. This is not compressible thermodynamics. Vorticity $\boldsymbol{\omega} = \nabla \times \mathbf{u}$ can drive confinement using a normalized gradient of its magnitude crossed with vorticity. Fedkiw, Stam and Jensen use this to compensate for visually lost rotation [S12]. It injects energy; record work and guard near-zero normalization. Resolution changes its effective behavior.

A shear-dependent viscosity may use $\dot{\gamma} = \sqrt{2D : D}$ and regularized power law $\mu = K(\dot{\gamma}^2 + \epsilon^2)^{(n-1)/2}$. Regularization changes zero-shear behavior. Lagged coefficients, Picard iteration and Newton methods have different convergence requirements. Viscoelasticity additionally needs stress history and cannot be modeled by viscosity alone. Delay extensions until their constitutive tests and memory costs are justified.

A discrete variable viscosity energy

Use a discrete strain operator E mapping face velocities to symmetric strain components at appropriate sample locations. Let M contain quadrature volumes and positive viscosity coefficients, with shear factors chosen consistently with tensor contraction. A viscous dissipation quadratic has the form $u^T E^T M E u$. Its nonnegativity follows from positive coefficients, independent of a particular stencil layout.

A backward-Euler viscous update with face mass matrix M_u is

$$(M_u + \Delta t E^T M E) u^* = M_u u^n + \Delta t f.$$

This construction exposes symmetry and dissipation. It also shows why independently smoothing each component can miss cross derivatives when viscosity varies or free-surface traction matters. The pressure constraint can be coupled through a saddle-point system or handled by a specifically justified splitting; those alternatives have different boundary errors.

At an impermeable no-slip solid, prescribed velocity contributes known terms when eliminating boundary degrees of freedom. At a stress-free liquid surface, traction conditions belong to the variational boundary treatment. Reusing a closed-box Laplacian at the free surface silently imposes a different stress model.

The expanded companion assembles a small weighted difference energy $E^T M E$ and checks symmetry, nonnegative spectrum and implicit decay. This validates the algebraic construction only. It does not implement the complete three-dimensional symmetric-gradient stencil or free-surface traction quadrature.

Nonlinear viscosity iteration

For generalized Newtonian viscosity, coefficients depend on strain. A lagged-coefficient iteration takes the current velocity guess, evaluates shear rate and viscosity, builds the linear viscous system, solves it, and repeats until both velocity change and nonlinear residual meet their criteria.

```
u_guess = old_velocity
require max_iterations is a finite nonnegative integer
for iteration in range(max_iterations):
    strain = E(u_guess)
    viscosity = material_law(strain)
    require positive finite viscosity
    solve the frozen-coefficient system for u_candidate
```

```

    evaluate the original nonlinear residual at u_candidate
    if accepted: return u_candidate
    apply declared damping if needed
    u_guess = updated guess
    return NonlinearIterationLimit

```

The declared iteration budget bounds frozen-coefficient solves. Exhausting it (including a zero budget) returns failure without publishing a candidate velocity.

A small iterate change alone can mean stagnation, not convergence. Viscosity clipping can make the linear system manageable but changes the material law. Record the range and active clipping fraction. A yield-stress model needs a stated regularization or complementarity treatment; a high-viscosity threshold is only an approximation.

A simple shear test is the first constitutive oracle. Prescribe a linear velocity profile, calculate constant shear rate analytically, and compare stress against the selected law. Then use a channel-flow case with known or independently computed profile. A visually thick fluid is not sufficient calibration.

Prescribed and dynamic solid coupling

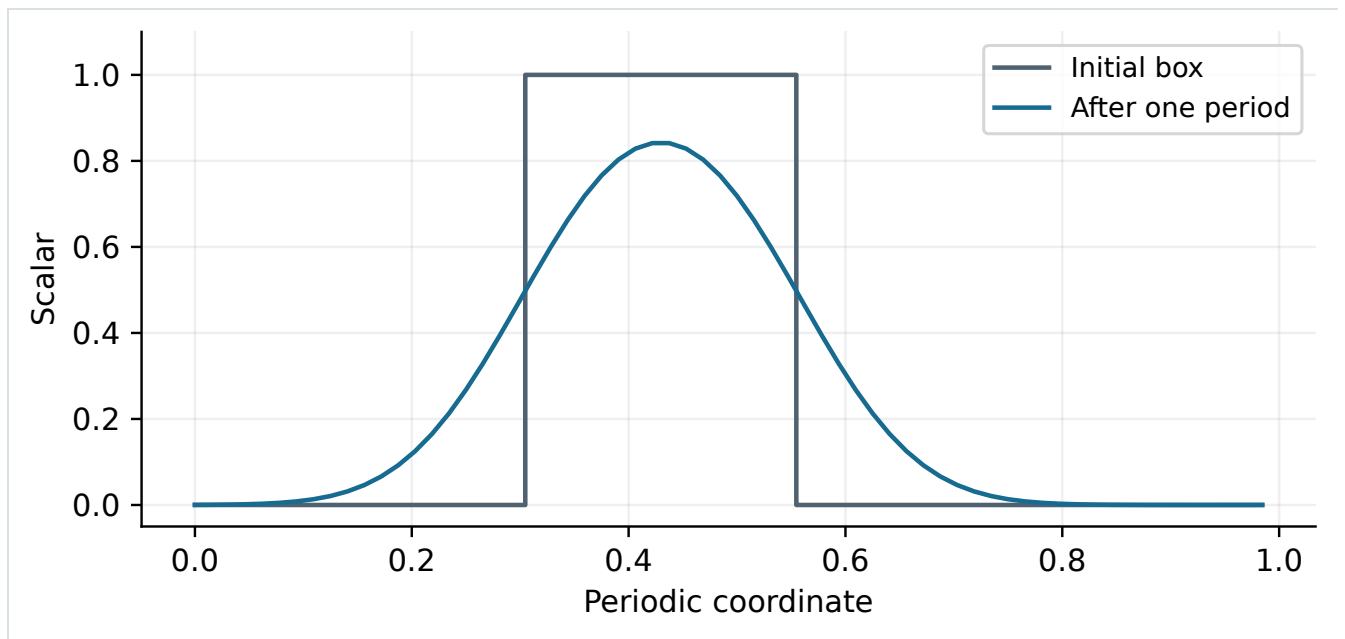
A prescribed obstacle has externally given motion. It can inject work into the fluid without receiving a matching reaction in the simulation. This is a valid one-way model if stated. A dynamic rigid body needs pressure forces and torques integrated over the same geometric interface used by the fluid operator.

For pressure traction $-p\mathbf{n}$, force and torque are

$$\mathbf{F} = - \int_{\partial S} p \mathbf{n} dA, \quad \boldsymbol{\tau} = - \int_{\partial S} (\mathbf{x} - \mathbf{x}_c) \times p \mathbf{n} dA.$$

Discretize these with the same face areas and normals used for flux coupling. Unrelated force quadrature can violate momentum exchange even if fluid divergence is small. The rigid body update also changes wall velocity, so explicit one-way sequencing can have added-mass instability for light bodies. A coupled solve or carefully qualified iteration is a separate capability.

Rheon's initial scope should support prescribed obstacles with explicit wall-work diagnostics. Dynamic two-way coupling should wait for momentum, rest-state and light-body tests. The variational coupling literature [S7] motivates that extension, but the current proof package does not verify it.



Original reference illustration for 10 Forces viscosity and material behavior

11 Particle alternatives and hybrid transfers

Dense grids pay for empty space; particles pay for occupied material and neighborhood organization. Include neighbors, sorting, scratch and surface reconstruction when comparing memory. Particle count alone is not a budget.

SPH

A smoothing-kernel density estimate is

$$\rho_i = \sum_j m_j W_h(\mathbf{x}_i - \mathbf{x}_j).$$

Support defines neighborhoods, often found through uniform bins or sorting. Pairwise force formulations can preserve antisymmetry when consistently evaluated, but free-surface neighbor deficiency and boundary quadrature remain. Müller, Charypar and Gross present an interactive SPH method [S13]; its published performance is not inherited by Rheon.

Weakly compressible pressure laws turn density deviation into force. Higher stiffness reduces compression but introduces fast waves and smaller steps. Visual softness may reflect tolerated compression. Measure density and volume errors before claiming incompressibility.

Position based fluids

Position based fluids iteratively corrects predicted positions using density constraints [S15]. For $C_i = \rho_i/\rho_0 - 1$, a local multiplier has the form

$$\lambda_i = -\frac{C_i}{\sum_k |\nabla_{\mathbf{x}_k} C_i|^2 + \epsilon}.$$

Regularization prevents singular corrections but changes behavior. Lists may become stale as positions change; rebuilding trades work for accuracy. Reconstructed velocity includes position corrections. Finite iterations yield approximate density constraints, not exact incompressibility or calibrated viscosity.

Compare complete frame cost at fixed occupied volume, obstacles, density tolerance, surface metric and memory cap. One density iteration and one pressure iteration are incomparable work units.

PIC and FLIP

PIC transfers particle momentum to a grid, applies grid forces and pressure, then samples velocity back. Repeated averaging dissipates detail. FLIP instead adds grid velocity changes:

$$\mathbf{v}_p^+ = \mathbf{v}_p + \sum_i w_{ip}(\mathbf{u}_i^+ - \mathbf{u}_i^{old}).$$

Blending PIC and FLIP trades damping against retained particle noise. Zero or tiny grid mass needs explicit handling. Parallel deposits need race-free accumulation.

APIC

APIC adds a local affine matrix C_p to particle velocity. A transfer includes

$$m_i \mathbf{u}_i = \sum_p w_{ip} m_p (\mathbf{v}_p + C_p (\mathbf{x}_i - \mathbf{x}_p)).$$

Jiang and collaborators derive conservation properties for specified transfer rules [S16, S17]. They depend on weights, moment matrices and updates. Merely adding nine floats does not inherit angular-momentum conservation. Boundary-truncated support also needs analysis.

Position, velocity, affine matrix and one scalar use 16 single-precision values, or 64 bytes before IDs, padding and scratch. Eight particles per cell in a filled 64-cubed region require 128 MiB for those fields alone; at 128 cubed, 1 GiB. These calculated costs matter in a model-memory-constrained host.

Choose dense grids for bounded low-resolution domains, sparse grids when empty space dominates, and particles when material motion justifies their overhead. Compare identical initial geometry, camera and metrics. No candidate wins every scene.

Complete PIC and FLIP update sequence

At particle positions x_p , evaluate weights w_{ip} on the selected grid lattice. Require partition of unity and, when claiming linear reproduction, $\sum_i w_{ip} x_i = x_p$. Deposit mass and momentum:

$$m_i = \sum_p m_p w_{ip}, \quad g_i = \sum_p m_p w_{ip} v_p, \quad u_i = g_i / m_i \quad (m_i > 0).$$

Store this normalized velocity as the old grid velocity before forces, pressure and collision changes. Apply the grid update to obtain $\tilde{u}_i$. Then

$$v_p^{\text{PIC}} = \sum_i w_{ip} \tilde{u}_i, \quad v_p^{\text{FLIP}} = v_p + \sum_i w_{ip} (\tilde{u}_i - u_i).$$

A blend with parameter $\beta \in [0, 1]$ is $(1 - \beta) v_p^{\text{PIC}} + \beta v_p^{\text{FLIP}}$. Advect positions using a declared grid-velocity interpolant and integrator, rather than assuming noisy FLIP particle velocity is always the best trajectory field. Collisions and reseedling follow a defined stage order.

A face-centered grid performs these deposits separately for each component lattice. Face mass is a transfer weight, not automatically the physical control-volume mass used in a pressure derivation. Keep the transfer normalization and fluid density model distinct. Near empty faces, a threshold must distinguish unavailable velocity from a valid zero.

Constant particle velocity is preserved by PIC transfer when all required mass and weight conditions hold. FLIP can preserve particle modes invisible to the grid because only grid changes are added. A pair of opposite particle velocities depositing to the same grid sample gives zero grid velocity; PIC removes that mode while FLIP retains it under a zero

grid update. This is a concrete noise/dissipation tradeoff, not simply an implementation defect.

Affine transfer moments

For frozen particle positions define offsets $r_{ip} = x_i - x_p$ and moment matrix

$$D_p = \sum_i w_{ip} r_{ip} r_{ip}^T.$$

A local affine particle field is $v_p + C_p r_{ip}$. Deposit

$$g_i = \sum_p m_p w_{ip} (v_p + C_p r_{ip}).$$

After grid evolution, a frozen-position least-squares affine reconstruction is

$$v_p^+ = \sum_i w_{ip} \tilde{u}_i, \quad H_p = \sum_i w_{ip} \tilde{u}_i r_{ip}^T, \quad C_p^+ = H_p D_p^{-1}.$$

These equations assume partition of unity, zero first moment and invertible D. They describe a transfer subproblem. Full time-integrated APIC variants require matching position and affine updates; conservation results from [S16, S17] cannot be transferred to an arbitrary integrator.

To see affine reproduction, let grid velocity be $a + Cx_i$. The particle average becomes $a + Cx_p$ because the weighted offset sum is zero. The cross moment is CD_p , so multiplying by D_p^{-1} returns C. The companion tests this with an independently constructed affine field. It does not test a full APIC liquid simulation.

Moment degeneracy and kernels

Linear hat weights in one dimension give $D = h^2 t(1 - t)$, which vanishes when a particle lies exactly on a grid point. An inverse then becomes undefined. Ignoring the case can generate huge affine coefficients near nodes. A pseudo-inverse, a limiting formula or a higher-order kernel are different choices that require their own consistency tests.

A tensor-product quadratic B-spline uses one-dimensional kernel

$$N(r) = \begin{cases} 3/4 - r^2, & |r| < 1/2, \\ \frac{1}{2}(3/2 - |r|)^2, & 1/2 \leq |r| < 3/2, \\ 0, & \text{otherwise.} \end{cases}$$

With complete regular-grid support, its moment is $D = (h^2/4)I$, giving a simple inverse and 27 collocated nodes per particle in three dimensions. Staggered component lattices require their own shifted supports; three components can mean 81 scalar deposits. The cost is not equivalent to eight-corner trilinear sampling.

Truncating support at a wall breaks moment identities. Renormalizing surviving weights restores their sum but generally not the zero first moment. Recentring changes the effective sample location. Ghost support, constrained reconstruction or boundary-specific transfer can address this, but each must be specified. The companion deliberately

removes one support node and measures the resulting moment defect instead of silently claiming affine preservation.

Momentum bookkeeping

Partition of unity gives total grid mass equal to total particle mass. Zero first moment removes the affine contribution to total linear momentum. Angular momentum has an internal affine contribution:

$$L_p = m_p x_p \times v_p + m_p \sum_i w_{ip} r_{ip} \times (C_p r_{ip}).$$

Comparing only $m_p x_p \times v_p$ would omit represented subparticle rotation. A transfer test should compare the full particle representation with grid angular momentum using the same support. Boundaries can exchange momentum; those impulses must be included rather than blamed on transfer error.

SPH kernels and gradients

For an explicit three-dimensional density fixture, use the compact poly6 kernel

$$W(r) = \frac{315}{64\pi h^9} (h^2 - |r|^2)^3 \quad \text{for } |r| < h,$$

and zero outside. Its gradient is

$$\nabla W(r) = -\frac{945}{32\pi h^9} (h^2 - |r|^2)^2 r.$$

The gradient is zero at coincident positions and at the support boundary under this formula. Other kernels may improve pressure behavior, but replacing the gradient with that of a different kernel changes the constraint derivative. The companion finite-differences the actual density constraint to test this exact gradient relationship.

For equal particle mass m and $C_i = \sum_j m W(x_i - x_j) / \rho_0 - 1$,

$$\nabla_{x_i} C_i = \frac{m}{\rho_0} \sum_{j \neq i} \nabla W(x_i - x_j), \quad \nabla_{x_j} C_i = -\frac{m}{\rho_0} \nabla W(x_i - x_j).$$

These derivatives sum to zero, expressing invariance under uniform translation. Include self density $mW(0)$ while excluding a spurious self-force. A finite-difference test that accidentally omits self density can still have the right gradient but the wrong density target.

PBF correction and neighbor validity

With equal solver weights, set

$$\lambda_i = -\frac{C_i}{\sum_k |\nabla_{x_k} C_i|^2 + \epsilon}.$$

Accumulating all constraint contributions gives

$$\Delta x_i = \frac{m}{\rho_0} \sum_{j \neq i} (\lambda_i + \lambda_j) \nabla W(x_i - x_j).$$

Use Jacobi-style separate correction buffers if all multipliers were computed from the same predicted positions. In-place updates change the iteration into an order-dependent method. Unequal masses require the corresponding inverse-mass-weighted constraint formulation; do not reuse the equal-weight equation unchanged.

A full step predicts positions from forces, builds neighborhoods, iterates density/collision corrections, reconstructs velocity from corrected displacement, then applies any declared viscosity or vorticity postprocess. Recompute density error after the final correction. A fixed iteration count defines a quality budget, not an incompressibility guarantee.

Uniform bins of width h need the 27 neighboring bins in three dimensions for support radius h . Hash collisions must be resolved by cell keys, and every candidate still needs an exact distance test. A cached neighbor list needs a skin radius and a displacement bound before reuse; otherwise particles entering support are missed. Sorting, prefix sums, bin ranges and particle permutations must preserve particle IDs and all state fields consistently.

Complete particle memory accounting

For P particles and K stored directed neighbor entries, a possible PBF layout uses old and predicted positions, velocity and correction vectors: 12 floats, or 48P bytes. Density and multiplier add 8P bytes. Two 32-bit sort/index arrays add 8P bytes. CSR neighbor offsets add approximately $4(P+1)$ bytes and neighbor IDs add 4K bytes. This totals roughly $68P + 4K$ bytes before boundary data, capacity padding, bin tables, renderer fields and alternative sort scratch.

At $P=100,000$ and mean directed neighbor count 60, these named arrays occupy about 30.8 MB in decimal units. A dense cluster can increase K sharply unless a cap changes the method; dropping neighbors silently changes density and forces. A checked capacity failure is preferable to an unreported truncated neighborhood.

For APIC, the earlier 64P-byte state estimate excludes old/new grid velocity, grid mass, sorting, particle output buffers and interface reconstruction. Count peak concurrent state for the actual update schedule. Particle methods save empty space only when these complete costs beat the selected grid alternative for the target scene.

The material preview now implemented in Rheon

The alternatives above describe a broader design space. The current Rust [ParticleFluidSimulation](#) implements a bounded SPH material preview. It does not implement the PBF, PIC, FLIP or APIC transfer sequences described in this chapter. Its particles carry physical position, velocity, mass, material identity and temperature. A particle represents a parcel of material, rather than a point drawn merely to make an image move.

This distinction matters when choosing a scene. The original fixed-box carrier solver has fluid throughout its pressure domain, even where its appearance tracer is zero. The material preview can instead begin with no parcels in a vacuum domain and add mass through a prescribed source. The existing browser 24³ source-pulse lesson still runs the carrier model; its volume and velocity curves are not a browser implementation of the material preview.

For unlike materials, counting a heavy neighbor's entire mass as the local material would distort the density estimate. Rheon weights each neighbor by its reference volume. Writing $\rho_{0,i}$ for material i 's entered reference density, its implemented estimate is

$$f_i = \sum_j \frac{m_j}{\rho_{0,j}} W_h(x_i - x_j), \quad \rho_i = \rho_{0,i} \max(f_i, 0.05).$$

The dimensionless sum f_i measures local reference-volume coverage; 0.05 is a numerical regularization floor. For example, two neighbors with the same reference volume contribute the same amount to f_i when their kernel weights match, even if one has ten times the mass. This does not prove that a material interface has the correct continuum pressure or surface shape.

The liquid pressure law is $p_i = \max(c^2(\rho_i - \rho_{0,i}), 0)$, with a deliberately reduced acoustic stiffness. Gas pressure uses $p_i = \rho_i R_i T_i$. Neither formula turns this preview into an exact incompressible liquid or a qualified shock/vacuum-gas solver. Pressure uses the poly6 density kernel's own gradient, while symmetric viscous pair forces damp relative motion. The [material implementation guide](#) spells out these coefficients and the limitations of cohesion and wall contact.

A two-parcel prediction before a full scene

Consider an isolated one-dimensional viscous pair with masses 1 kg and velocities +1 and −1 m/s. Let the pair drag coefficient be 1 kg/s, so the forces are −2 and +2 N. With an explicit interval of 1/4 s, the new velocities are +1/2 and −1/2 m/s. Total momentum stays zero, and kinetic energy falls from 1 J to 1/4 J. This is an exact arithmetic teaching fixture, not a claim that a complete source-driven scene must lose energy every step.

Now use 5/4 s in the same frozen-pair calculation. The velocities become −3/2 and +3/2 m/s and energy grows to 9/4 J. The continuous pair power was nonpositive, yet the explicit update was too large. Rheon's timestep checks and bounded retries are therefore separate from the force-sign argument. Native pair and full-step tests must exercise the implemented update as well.

For a visual experiment, keep the physical scene and particle sampling fixed and increase only the desktop preview resolution. The expected change is a finer voxel display, with the same numerical parcel spacing. Then start a new run with more physical samples: the numerical problem and its cost change. Record the actual stored preview size, accepted time, parcel count, total mass and source mass before comparing images. Compare at a common physical time; an equal number of adaptive steps need not reach the same time.

Checked material-preview proof bridge

[MaterialPreview.lean](#) states pressure-pair action/reaction, viscous-pair momentum balance and a conditional finite-step energy bound. The latter requires positive masses and $\Delta t b(1/m_i + 1/m_j) \leq 2$, where $b \geq 0$ is the frozen pair drag coefficient. It covers the isolated pair above, not all forces in an evolving neighbor graph. The density floor, finite precision, neighbor enumeration and continuum accuracy require separate evidence. Qualification status and the implementation-to-statement mapping belong to the [material contract guide](#).

Draft checkpoint: inspect the stage that produced a value

The unmerged [PR47 inspection checkpoint](#), reviewed at 1923d5bd3442a5369878a1a317b52f4a89026d5b, adds [particle_diagnostics\(\)](#). Its committed density, pressure and acceleration describe the **post-emission, pre-integration** force evaluation. The returned evaluation time is distinct from the accepted positions' and velocities' time.

Before the first accepted step the accessor returns no evaluated diagnostics; a rejected attempt does not publish overwritten force scratch. This draft is not part of the current book checkout's material API.

A practical exercise is to compare reported force with mass times the recorded acceleration, then inspect a wall-contact step. Gravity and implemented pair forces enter the evaluation; subsequent wall clipping/drag does not. Therefore mass times the total accepted velocity change divided by dt need not equal the reported evaluation force. The draft's wall-stage and rollback tests exercise this distinction. A pressure number without its stage can suggest a consistency failure that is actually a comparison of different operations.

The [draft inspector](#) retains one latest numerical copy and at most 256 selected-particle samples. Selecting another particle starts its trace from the current sample; it cannot recover that particle's unrecorded past from eight-bit visual history. IDs are append-only indices scoped to one simulation instance, not global identities. CSV uses binary64 round-trip values and leaves unavailable initial fields blank. Its raw mass, momentum and energy totals are not closed-system residuals without emission, gravity, dissipation and wall exchanges. The native window, clipboard and user-device performance remain outside the draft's headless validation.

One grid sample, two particles

Give two fixed particles masses $(1, m)$, velocities $(v+d, v-d)$, and one grid sample with transfer weights one. Its mass-weighted velocity is $u = [(v+d)+m(v-d)]/(1+m)$. After a prescribed grid increment Δu , PIC returns $u+\Delta u$ to both particles. With equal masses, $v = 0$ and $d = 1$, PIC removes the $(+1, -1)$ counterflow although the total momentum was already zero. The lost quantity is unresolved kinetic variation, not momentum.

Preserving an unresolved transfer mode

FLIP instead returns each old velocity plus Δu . The displayed blend uses a FLIP fraction a in $[0,1]$: $v'_i = u+\Delta u+a(v_i-u)$. It preserves the expected momentum and scales kinetic energy about the mass-weighted mean by a^2 . Select $a = 1$ to retain the counterflow, or $a = 1/2$ to retain one-quarter of its variation energy. Unequal masses change the mean and must enter every momentum and variance calculation. Positions stay fixed; this lesson does not advect particles or advance SPH/PBF. [The APIC authors' publication summary](#) identifies information loss during transfer and motivates retaining affine velocity information. This one-sample experiment is PIC/FLIP only; it does not establish APIC angular momentum or infer that retaining unresolved modes makes a complete simulation stable.

► Practical study proof scope

12 Time stepping and error budgets

Physical time and display frames are separate clocks. A frame can contain substeps; a slow frame must not silently enlarge the physical step beyond its accepted regime. Decide whether overload slows simulation, drops presentation frames or changes a declared quality tier. Record the policy in replay metadata.

Restrictions

Directional Courant numbers are $c_x = |u|\Delta t/h_x$ and analogues. Multidimensional positivity depends on the update and can involve their sum. Characteristic transport may remain bounded without that restriction, but travel limits still help trajectory accuracy, collision detection and narrow-band validity.

Explicit diffusion depends on $\nu\Delta t \sum_d h_d^{-2}$. Surface tension introduces a capillary scale proportional to $\sqrt{\rho h^3/\sigma}$, with method-dependent constant. Acceleration and moving walls add constraints. Return a step and its limiting reason, not an unexplained number. If required substeps exceed a cap, return a typed outcome rather than label a partial interval complete.

Error accounting

A conceptual budget is

$$E_{\text{total}} \lesssim E_{\text{model}} + E_{\text{space}} + E_{\text{time}} + E_{\text{solve}} + E_{\text{round}} + E_{\text{surface}} + E_{\text{render}}.$$

This is an accounting guide, not a universal inequality across incompatible norms. Residual controls one constraint error, not advection displacement or mesh extraction. Machine-precision pressure with a coarse staircase obstacle may overspend on a subdominant error.

Refine space at fixed Courant number for smooth transport. Refine time with sufficiently small spatial error for temporal order. Simultaneously changing both mixes effects. For interfaces measure volume, surface distance and silhouette displacement; small average error can hide catastrophic filament loss.

Manufactured solutions

A periodic stream function $\psi = \sin x \sin y$ gives velocity $(\partial_y \psi, -\partial_x \psi)$ with zero continuous divergence. Sampling it does not automatically give zero discrete divergence; a discrete curl construction can ensure cancellation. Distinguish those tests.

Choosing discrete pressure p , computing $b = Ap$ and solving tests the solver for that operator, not the operator's physical accuracy. Add independent analytic and stencil checks. Expected values should not reuse the same indexing routine whose correctness is in question.

Long horizons and outcomes

Track volume or tracer amount, energy, divergence, surface position and rejection count after sources stop. Include rest, translation, rotation and eventually liquid and obstacle cases. Sensitive flows may not match pointwise indefinitely; use bounded time windows and defined geometric or statistical observables. Exact replay remains a separate implementation property.

Outcomes should distinguish converged, iteration limit, invalid input, nonfinite state, incompatible flux and resource limit. A nonconverged preview may be permitted explicitly but must not masquerade as accepted scientific state. Candidate and accepted buffers need clear ownership so cancellation cannot corrupt future steps.

One mode, three time updates

A clean test removes spatial error before asking about timestep error. In a periodic unit-metre cube with density 1 kg/m³, prescribe

$$\mathbf{u}(x, y, z, t) = (A(t) \sin(2\pi y), 0, 0), \quad \nu = \frac{1}{4\pi^2} \text{ m}^2/\text{s}.$$

There is no x dependence, so divergence and self-advection vanish. With no forcing and constant pressure, viscosity gives $A' = -\lambda A$, with $\lambda = \nu(2\pi)^2 = 1 \text{ s}^{-1}$. Initially $A = 1 \text{ m/s}$. The exact amplitude is $e^{-t} \text{ m/s}$ and the unit-cube kinetic energy is $K = A^2/4 \text{ J}$: the integral of the squared sine over one period is one half. The cube is periodic; its edges in the picture are not stationary walls.

For $a = \lambda\Delta t$, explicit Euler multiplies amplitude by $1 - a$, whereas implicit Euler multiplies it by $1/(1 + a)$. The browser actually repeats these scalar updates. It samples their resulting three-dimensional velocity field at fixed locations, without solving a spatial stencil or pressure system. The field varies only in y; repeated x/z samples show that symmetry. All arrows share a fixed scale, and dark arrows show exact decay at the same positions. The energy readout uses the integral above, not the number of visible arrows.

Start with target time **2 s**, **one step**, **Explicit Euler**. Predict the sign before inspecting the arrows: A becomes -1 m/s while K remains 0.25 J. The exact flow still points in its original direction, with $A \approx 0.135335 \text{ m/s}$ and $K \approx 0.00457891 \text{ J}$. Energy nonincrease alone did not detect the reversed flow. The explicit factor is nonnegative for $0 \leq a \leq 1$, reverses sign for $a > 1$, and has magnitude at most one only for $0 \leq a \leq 2$. At $a = 2$ there is no decay at all, despite the energy bound.

Implicit decay and fixed-time refinement

Now choose **Implicit Euler** without changing time or step count. It yields $A = 1/3 \text{ m/s}$ and $K = 1/36 \text{ J}$, neither the explicit result nor the exact answer. Its energy decays for every nonnegative a , but a single large step still has substantial temporal error. Finally keep target time fixed at 2 s and change step count to 2, 4, 8 and 16. Compare both amplitude-error columns at the same physical time. Changing final time while refining would answer a different question. **Displayed step** inspects intermediate states; it does not animate or advance a native simulation.

An energy-growing step

The 3 s, one-step explicit preset gives $A = -2$ m/s and $K = 1$ J: viscosity has numerically increased energy above its initial 0.25 J. That is a rejected method choice for this mode, not a physical instability of the specified unforced flow. The lesson's controls bound amplitudes to ± 2 m/s, so arrows need no hidden normalization or clipping. Choosing Exact changes only which field is displayed.

Research connection and what the test leaves out

[Stam's original *Stable Fluids* paper](#) (1999), section 2.2, replaces an explicit viscous update with an implicit diffusion solve; section 2.3 explains the simpler Fourier-space form for a periodic domain. Its introduction also separates graphical utility from engineering accuracy. Those observations motivate inspecting amplification rather than treating "stable" as an accuracy certificate. The scalar comparison here isolates the time update; it does not reproduce the paper's whole split algorithm or its total dissipation. In particular, this single mode's implicit amplitude at coarse timesteps exceeds the exact amplitude.

The derivation above assumes constant coefficients, a periodic smooth shear, no force and no changing geometry. It contains no spatial discretization error; the display samples are not a resolution control. A general diffusion grid has many eigenmodes, and its most restrictive mode determines an explicit timestep bound. Advection, moving boundaries, capillarity, nonlinear materials, approximate linear solves and finite arithmetic add separate error and acceptance conditions. Neither this lesson nor a passing amplitude comparison establishes a stable whole Rheon step, physical calibration, or runtime performance.

Checked modal amplification proof bridge

[ModalDecay.lean](#) proves the exact-real explicit energy-change identity $K(A^+) - K(A) = a(a - 2)K(A)$ and derives energy nonincrease for $0 \leq a \leq 2$. Sign preservation has its separate hypothesis $a \leq 1$ with nonnegative incoming amplitude; strictly positive amplitude reverses for $a > 1$. The $a = 2$ witness formally preserves energy while reversing unit amplitude.

For the implicit relation $(1 + a)A^+ = A$, the file proves $K(A^+) - K(A) = -a(2 + a)K(A^+)$ and energy nonincrease for nonnegative a . It also checks that division by $1 + a$ satisfies that update under this hypothesis. These are scalar amplification theorems. The PDE ansatz, sine integral, exponential solution, convergence order, JavaScript and Rust execution, IEEE arithmetic and rendering are outside the Lean statements. The current kernel status is source-bound in the [coverage ledger](#), separate from numerical oracle and browser evidence.

13 Memory layout and parallel execution

A solver leaving no room for diffusion weights fails Rheon's intended role even if fast alone. The relevant quantity is peak simultaneous memory: persistent state, scratch, render resources, readback buffers and the host application. Persistent simulation arrays alone underestimate it.

An explicit layout

For a cubic grid, face count is $F = 3n^2(n + 1)$. Current and next velocity, eight cell-centered single-precision arrays and a one-byte mask require

$$M(n) = 4(2F + 8n^3) + n^3 \text{ bytes.}$$

This calculated layout excludes allocator overhead, alignment, ghost layers, coefficients, meshes, particles, driver allocations and model weights. The eight scalars might be pressure, right-hand side, residual, search direction, matrix product, preconditioned residual and two tracer buffers. Assign actual lifetimes before reuse.

The formula gives 1.805 MiB at 32 cubed, 14.344 MiB at 64 cubed, 48.305 MiB at 96 cubed and 114.375 MiB at 128 cubed. Doubling every axis multiplies leading volume cost by eight. At 256 cubed this incomplete layout is about 913.5 MiB. Show estimates before allocation.

Ownership and access

Separate velocity arrays naturally match staggered shapes. Particle structures of arrays aid per-field traversal, while tiled layouts may suit transfers reading several fields. Benchmark actual access patterns. Avoid abstractions that conceal shape or introduce unmeasured per-sample dispatch.

A shape owns counts, strides, spacing, origin and checked capacity. State owns persistent fields. Workspace owns scratch and alias rules. A solver borrows them during a step. This exposes reuse without a global pool whose lifecycle is unclear.

For a face joining a and b, the pressure operator contributes $w(p_a - p_b)$ and its negative. Cell gathering writes one output without scatter races. Face scattering emphasizes paired conservation but requires atomics, coloring or local accumulation. Compare reproducibility and bandwidth.

Parallel reductions

CG dot products are global reductions. Floating addition is nonassociative, so tree shape changes results. Fixed trees support repeatability on a pinned configuration; arbitrary atomics generally do not. Pairwise or compensated accumulation reduces some error at cost. Single-precision fields with double-precision reductions are a candidate to qualify, not a universally available GPU feature.

State whether replay is bitwise on one backend, tolerance-based across backends, or only configuration-reproducible. Save precision, solver tolerance, step policy and algorithm version when they affect the promise.

Sparse grids

Sparse tiles save empty-space cost but add keys, occupancy maps, halos and fragmented work. Benefits depend on active fraction and stencil expansion. A moving plume may activate many tiles at once; reserve or reject before mutating accepted state. Disconnected sparse pressure regions need consistent component gauges.

Begin with dense bounded grids and honest peak reporting. Adopt sparsity when representative scenes show empty-space cost is the limiting resource and topology machinery has a clear owner.

Three sizes that the desktop keeps separate

The current material scene has three independently meaningful sizes: its physical domain in metres, its parcel sampling, and its stored preview grid. Doubling the side of a filled cube at the same reference density multiplies initial mass by eight. Raising preview resolution changes how accepted parcels are displayed; it does not add numerical parcels or improve the fluid update.

The desktop stores an eight-bit preview and metadata for step zero and every accepted step. A requested resolution may be reduced before the run so that the whole planned history fits its budget. For orientation, one 128^3 scalar byte array is 2 MiB; 1,025 such arrays already occupy 2,050 MiB before metadata. These calculated payload sizes exclude numerical state, meshes, thread stacks and allocator overhead. The current sidebar reports the actual stored size and planned bytes. Its history budget is not a process-memory cap.

The particle owner has a different budget. It reserves accepted and candidate particles, per-particle density/pressure and force scratch, neighbor links and bin heads. Capacity accounting uses the compiled element sizes and allocated capacities, including padding. A full particle array must cause an explicit failure rather than silently drop parcels or neighbors. See the [material contract guide](#) and the [desktop ownership contract](#).

What an adaptive CPU batch changes

Imagine 1,000 independent output entries divided into batches of 256. The four jobs own intervals [0,256), [256,512), [512,768) and [768,1000). Every entry has one writer; the final job simply has fewer entries. Four jobs do not mean four workers will remain busy for the entire pass. Their costs may differ, and a join waits for the slowest unfinished job.

Rheon's run-owned [CpuScheduler](#) chooses bounded batch sizes above Rayon's existing work-stealing scheduler. It times complete joined passes and periodically compares a small set of candidates. It changes work boundaries, while each item's arithmetic and the ordered reductions retain their established order. Smaller jobs expose more parallel work but also add scheduling overhead. The fastest observed candidate is evidence for that workload and timing sample, not a universal best grain size.

The exploration counter also has a scope: for a stable workload with one outstanding invocation and exactly one success record per pass, the first eight successful calls visit four candidates twice, and the cycle repeats after 128 calls. That conditional accounting does not guarantee exploration under arbitrary overlapping calls, failures or changing workloads.

SGS sweeps also have dependencies. A tile may read completed predecessor tiles, so only a ready wave can run together. In a $2 \times 2 \times 2$ tile box, grouping by $b_x + b_y + b_z$ gives waves containing 1, 3, 3 and 1 tiles. Eight total jobs therefore do not provide eight independent jobs at once. Rheon separately calibrates its supported SGS tile sizes on scratch and keeps ordered wave joins. The [scheduling contract guide](#) separates interval coverage, dependency order and timing feedback.

To read the CPU inspector, compare complete-pass wall time before interpreting worker boxes. Orange markers sample actual visits at the repaint rate, and idle workers may retain their last position. Kernel shares include recorded passes only; validation, preview capture and other unrecorded work are outside that denominator. A busier display need not mean a faster accepted step. The [scheduling evidence](#) retains exact workloads, build identities and the limits of its historical measurements.

Checked scheduling proof bridge

Finite-index and dependency statements can establish that a declared partition covers its entries and that a predecessor lies in an earlier wave. They do not prove that the OS runs every worker, that a timing choice is optimal, or that floating reductions may be reordered safely. The current [CpuScheduling.lean](#) source and its [implementation mapping](#) must be read alongside actual cross-thread replay and scope-cleanup tests. A source listing alone is not a fresh kernel qualification.

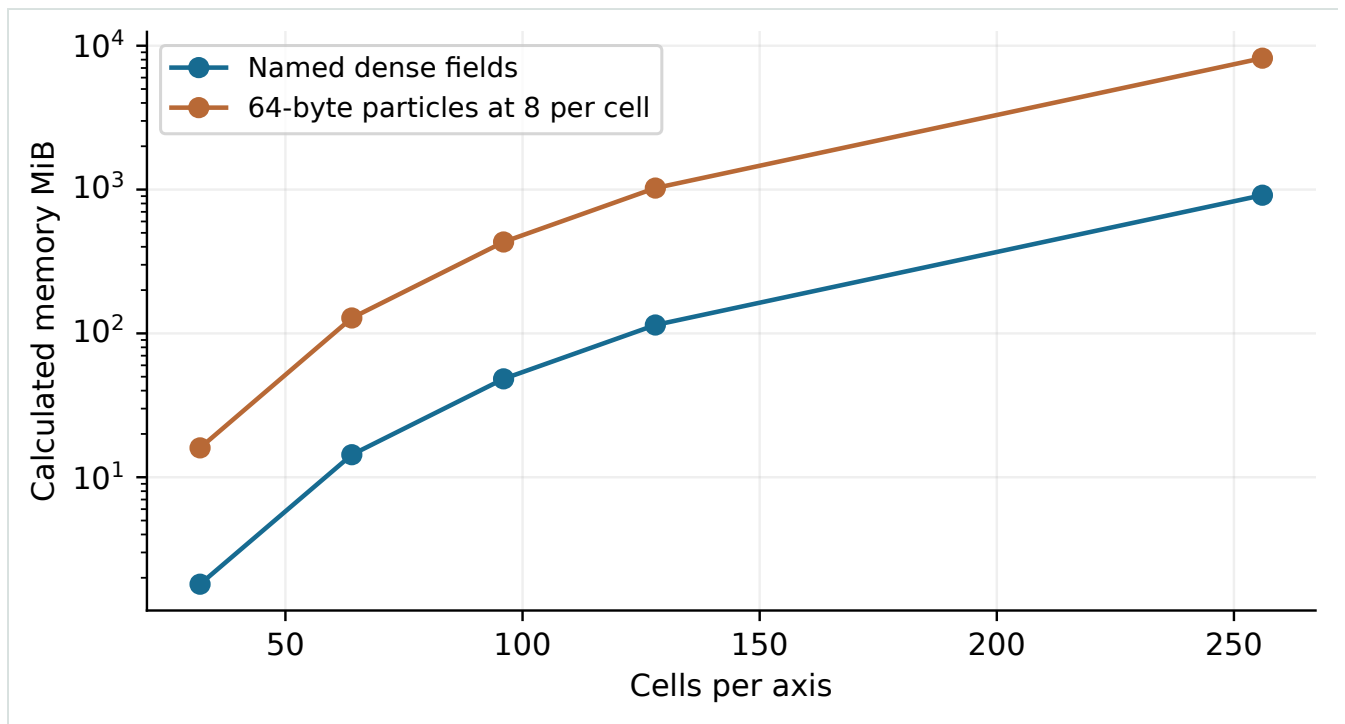
Counting faces and sparse arrays

For n^3 cells and c cell-centred scalar fields, all three face families contain $3n^2(n+1)$ scalars. With w bytes per scalar, fields require $w[cn^3+3n^2(n+1)]$ bytes. The displayed cube and face grids expose the indexing shape; they are an allocation diagram, not physical cell dimensions. A seven-point Dirichlet stencil has $n^3+6n^2(n-1)$ nonzeros, including the diagonal. CSR uses one scalar and one 32-bit column index per nonzero, plus n^3+1 32-bit row offsets. Five scalar solver vectors are counted separately.

Doubling resolution and scalar width

Choose $n = 16$, $c = 4$ and $w = 4$. Fields use 117,760 bytes, CSR 233,476 bytes and the five vectors 81,920 bytes: included total 433,156 bytes. Doubling n and moving to eight-byte scalars changes both the field count and the value/index mix; an eightfold or sixteenfold estimate is only a leading-order intuition. The actual readout evaluates every term. No allocator padding, ghosts, particles, rendering buffers or GPU copies are included. Measured peak memory and CPU time require executing the specified runtime; this byte model cannot establish a performance claim.

► Practical study proof scope



Original reference illustration for 13 Memory layout and parallel execution

14 Guidance as a geometric measurement

Guidance rendering is a measurement layer with a camera model, clipping, sampling and units. It must preserve the spatial information actually simulated. A downstream model may change appearance; Rheon should export reproducible geometric evidence.

Camera coordinates

For camera coordinates (X, Y, Z) with positive Z forward,

$$u = f_x X / Z + c_x, \quad v = f_y Y / Z + c_y.$$

Axial depth Z differs from radial range $\sqrt{X^2 + Y^2 + Z^2}$ off axis. Name the representation, handedness, image origin, clipping and pixel-center convention. Mirroring or shifts may be camera-contract defects rather than fluid errors.

Linearized projection gives

$$\delta u \approx (f_x / Z) \delta X - (f_x X / Z^2) \delta Z.$$

The same world-space error moves more pixels near the camera, and depth errors shift off-axis silhouettes. Evaluate actual camera distributions, not just a world-space norm.

Surface and volume outputs

Liquid can export silhouette, depth and normals in a declared frame. Smoke may integrate extinction. Piecewise constant extinction along a ray gives

$$T = \exp \left(- \sum_i \sigma_i \Delta s_i \right), \quad \alpha = 1 - T.$$

Increasing samples without adjusting segment length changes opacity incorrectly. Concentration-to-extinction is a rendering model, not conservation. A diffuse volume has no unique surface; first-hit depth needs a named threshold or accumulated-opacity rule.

Graphics depth buffers may be nonlinear in distance. Exporting raw values as metre depth is incorrect without conversion. Validate planes at known distances and off-axis spheres. Include validity masks rather than ambiguous zero depth.

Metrics and integration

Silhouette IoU is $|S \cap T| / |S \cup T|$, with an explicit both-empty case. It can hide a displaced thin filament in a large body. Add boundary-distance percentiles and maximum displacement. Compare depth on mutually valid pixels and report missing surfaces separately. Normals use angular error.

Rest scenes under a fixed camera expose flicker from grid crossings, mesh ambiguity and thresholds. Filtering can reduce flicker while shifting edges, which must remain in the error budget.

Export camera calibration, world scale, dimensions, channel semantics, masks, physical time, state ID and renderer version. Resizing requires updated intrinsics or a stated resampling transform. A depth image without its mapping is ambiguous even when attractive.

The first renderer should be simple and deterministic. Its acceptance is geometric agreement with known primitives and state, not photorealism. Integrated diffusion experiments need separately pinned models and seeds.

Read the field before interpreting the picture

The sticky example beside this chapter is the existing **24³ carrier-flow lesson**: it displays a computed tracer and instantaneous velocity from that bounded grid model. It is not the native material/SPH preview, a CAD flow solution, or a pressure-load measurement. The additions below explain how to ask better questions of each model; they add no new live material or CAD solver.

Start with a field's meaning. A passive tracer marks where transported material has been represented; opacity is a display rule applied to that marker. Neither is automatically mass density. A velocity arrow represents a vector, while its length may be fixed or proportional to speed. An instantaneous streamline is tangent to one frozen velocity field; it does not record a parcel's past motion. The [engineering-visualization guide](#) develops sampling, sections, pressure, loads and export from these distinctions.

Rheon has two separate data paths. The native material preview exposes particle position, velocity, mass, temperature and material ID. Its per-particle density and pressure are private, and the reported maximum pressure is evaluated before the position and wall-temperature update; accepted speed describes a later stage. A pressure maximum cannot become a synchronized pressure slice merely by coloring accepted particle positions. The carrier path instead has staggered face velocities and a pressure correction. Its correction is not particle pressure, and its aligned face flux is not SPH mass crossing. See the [material contract](#) and [source-anchored inventory](#).

For every probe, ask: **what quantity, where, in which units, at what physical time and evaluation stage, and from which data?** Mark it native, derived, prescribed or unavailable. Show the sampling rule, validity and pressure reference when applicable. An unavailable value must remain distinguishable from zero. A color range belongs to the measurement: lock it across compared runs, and show numerical values alongside color.

Predict a measurement before adding a view

Consider a unit box with constant density. Prescribe three fields: rest, uniform velocity $(1, 0, 0)$ m/s, and shear $(y\text{ s}^{-1}, 0, 0)$ with y measured in metres. A density-only image is identical in all three cases. Through the unit face normal to positive x ,

$$Q = \int_A \mathbf{u} \cdot \mathbf{n} dA$$

is respectively 0, 1 and 0.5 m³/s. Reversing the normal reverses the signs. These are prescribed analytic fields, not three native simulations satisfying the same box-wall conditions. This small example teaches why a slice, arrows and an oriented section answer questions that opacity cannot.

The seven [lesson specifications](#) are a proposed progression, **not seven implemented browser controls**. The [analytic companion](#) checks selected mathematical reference calculations; it does not call Rheon or validate its physical models. Detailed controls, assumptions and failure probes are in [the worked lessons](#).

Question	Prediction to check	Limit to keep visible
Can constant density hide motion?	Unit-section flux is 0, 1 or 0.5 m ³ /s.	A prescribed field need not meet the solver's walls.
What should a planar channel measure?	The no-slip laminar profile has $U_{\max}/U_{\text{mean}} = 1.5$.	Infinite-width or periodic-span assumptions exclude influential sidewalls; the SPH box preview is not this test.
Does vorticity identify a vortex?	Simple shear has vorticity but zero rotation-minus-strain Q; rigid rotation differs.	Q conventions differ when divergence is nonzero.
Does a streamline show particle history?	For $\mathbf{u} = (1, t, 0)$ in the stated SI scales at 2 s, streamline, pathline and streakline have different shapes.	Quantitative time snapshots and stable particle IDs cannot be recovered from preview bytes.
Can color hide a tenfold change?	At $x = 0.5$ m, the two prescribed speeds are 0.5 and 5 m/s.	Per-frame autoscaling can make both pictures look alike.
Does refinement establish accuracy?	The manufactured quantity $1 + 2h^2$ has observed order two and extrapolate one.	This is not an error estimate for Rheon or evidence of physical validation.
Why does a particle cloud look less dense at its edge?	The specified poly6 lattice gives about 1009.775 kg/m ³ at its center and 817.000 kg/m ³ at an outer-face particle.	Finite support and reconstruction affect the image; normalizing a display attribute does not repair density physics.

Keep geometry and engineering claims separate

A CAD design surface, the boundary used by the solver and a display mesh may all differ. A smooth surface colored from nearby fluid samples is not yet a conservative force integral. Pressure loads require a declared pressure reference, consistently oriented numerical boundary and supported stress model. Uniform pressure cancels on a closed surface with consistent normals; it generally does not cancel on an open patch. Wall shear additionally requires a qualified viscous wall treatment and wall gradient. Box-contact impulses are a different diagnostic, not a wall-shear map. These distinctions follow the separate [pressure, viscous and moment conventions](#) used in established analysis tools.

The rotating emitter is a prescribed source. Its pressure-head speed, midpoint orientation and fixed-volume parcel credits do not provide a nozzle collision solid or simulated nozzle walls. The source definitions and accepted publication are anchored in [emitter geometry, profile](#), [particle admission](#) and [transactional publication](#).

The first useful native extension is an unquantized, stage-labelled particle inspector and export, followed by reconstruction with visible support and invalid regions. Add histories only after retaining quantitative state, and loads only after establishing their geometry and stress contract. An analytic check verifies a calculation; a relevant refinement study estimates numerical error; matched experiments address physical validation. The [source bibliography](#) and [implementation order](#) preserve the evidence and prerequisites behind that sequence.

Draft checkpoint: inspect a retained obstacle projection

Unmerged [PR50](#), inspected at 3f49a158a0f1bec93f47edf6cc42ad86242e4aa5, adds a native [obstacle projection inspector](#). Its 12×8×6 grid has 0.125 m spacing and a fixed aligned obstacle. One accepted pressure projection over 0.05 s supplies a retained result; this is not an advancing wake or a viscous wall-load computation. Its [scope and validation record](#) records the fixture, gauge, rendering choices and remaining integration work.

A practical inspection sequence is to probe a fluid cell, rotate the camera, and revisit the same cell: presentation changes must not change its value. Then move the slice. Scalar colors retain a full-state valid-cell range, so a slice change cannot silently rescale a small feature into a large one. Pressure means gauge-fixed correction in Pa; speed is reconstructed from adjacent face velocities, not a native face flux. Solid cells report unavailable, not zero. Velocity arrows encode velocity times 0.1 s, with physical aspect ratio and camera foreshortening; they are neither normalized directions nor trajectories.

The lab's snapshot and camera are separate from live appearance history. Scrubbing an SPH appearance frame cannot retime the retained projection or recover quantitative particle fields. The draft uses CPU-drawn slices and wire overlays, and requires full-fluid or solid cells; it supplies no arbitrary CAD import, cut-cell reconstruction, calibrated pressure/viscous load, VTK exchange or GPU-performance result. Its reported local PR47 combination remains a tested draft integration, not a merge or shared accepted API. The existing browser chapter still presents its separate bounded carrier-flow lesson.

15 Exact mathematics and finite arithmetic

Lean real numbers are not IEEE floating values. Exact cancellation does not imply identical bits from reordered reductions. Goldberg is a foundational reference for rounding and exceptional values [S18]. The proof-to-runtime gap requires explicit analysis and tests.

Rounding

For suitable finite normal values, rounding assumptions and no exceptional range behavior, a common model is

$$\text{fl}(a \circ b) = (a \circ b)(1 + \delta), \quad |\delta| \leq \varepsilon.$$

Zero results require absolute-error interpretation; underflow, overflow and NaNs need separate treatment. For n-term accumulation, $\gamma_n = n\varepsilon/(1 - n\varepsilon)$ is a standard bound factor when $n\varepsilon < 1$. Cancellation can make final relative error large despite small absolute error.

Pressure subtracts nearly equal velocities near rest. Report absolute divergence and physical energy scales alongside relative residual. Every normalization needs a denominator and zero policy.

An executed arithmetic comparison

The companion's binary64 residual identity discrepancy is approximately 2.49×10^{-15} . Replaying the algebra with binary32 inputs and operations gives 1.70×10^{-6} . These are fixed-problem observations, not universal bounds.

The binary32 replay rounds the binary64 pressure; it does not run a binary32 CG solver. Its purpose is to demonstrate arithmetic discrepancy in a formally exact identity. Production binary32 needs its own convergence, breakdown and residual-drift qualification.

Invalid values and finite range

Validate finiteness explicitly before range checks for spacing, time, density, viscosity, cameras and initial data. NaN ordered comparisons are false. Negative zero may affect serialization or hashes even when harmless numerically; define canonicalization only if the replay contract needs it.

Do not silently replace nonfinite state with zero. Report stage and version, preserve accepted state and use only an explicitly authorized degraded-preview policy. Otherwise the intervention hides the cause.

Natural-number indexing omits machine range. Check dimensions, face counts, byte counts and offsets before allocation, then enforce resource limits. Huge finite world coordinates can overflow integer conversion or lose subcell precision. Clamp only under a declared boundary sampling policy; otherwise reject out-of-domain.

Tolerances

Pressure convergence, geometry classification, interpolation bounds and test comparisons need separate tolerances with units. A global epsilon cannot serve all. A tolerance changes the accepted mathematical set and should be documented.

Use exact rational or higher-precision small oracles. Test both sides of thresholds, zero step, minimum shapes, empty liquid, blocked domains, tiny coefficients and near-contact geometry. Derive tolerances from error models or accepted geometric budgets; never enlarge them merely until a visible defect passes.

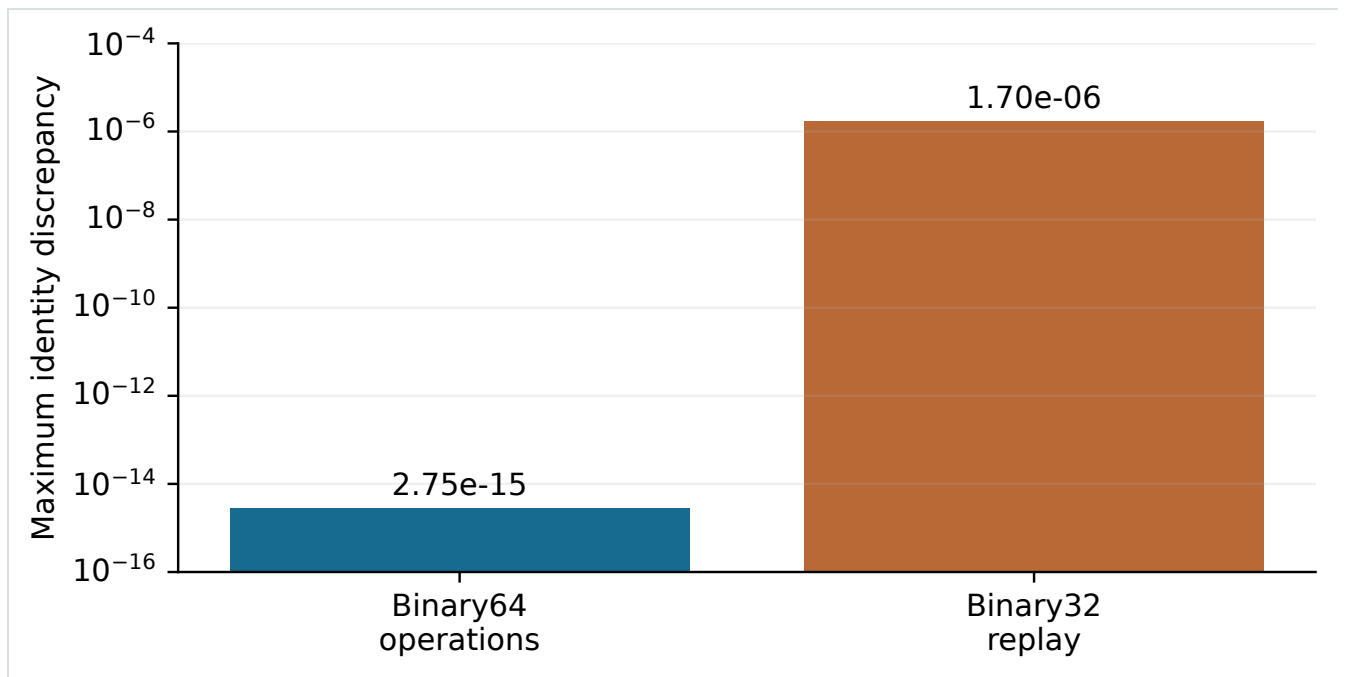
A unit lost in a reduction

Take $A = 2^{24}$ and sum $A, 1, -A$. The browser explicitly applies `Math.fround` to each input and addition in its binary32 mode. The trace becomes $(A, A, 0)$, so the executed sum differs from the exact result 1. In binary64 mode the same finite terms give 1. [ECMAScript's Math.fround specification](#) defines conversion through binary32 with ties-to-even rounding. Python's independent oracle uses struct conversion to check every trace element.

Cancel the large pair first

Change only the order to $A, -A, 1$. The binary32 trace is $(A, 0, 1)$. Inspect both bars at the same final-sum scale. This is a concrete counterexample to transferring the Lean exact-real associativity identity to floating-point reductions. Parallel reductions may choose other orders; compiler flags, precision, exceptional values and the executed reduction tree are part of the implementation contract. These finite checks do not bound a complete solver's accumulated error or license reassociation of production code.

► **Practical study proof scope**



Original reference illustration for 15 Exact mathematics and finite arithmetic

16 The checked Lean companion

The foundation has 20 public theorems in three modules; the expanded edition adds thirteen in `Physics.lean` and nine in `BoundedPhysics.lean`, for 42 public theorems in five modules. The bounded module also defines six helpers; its fifteen audited declarations are not fifteen theorems. Lean 4.19.0 and mathlib revision `c44e0c8ee63ca166450922a373c7409c5d26b00b` are pinned, with locked transitive revisions. Hosted qualification at source head `ecf97d3a943ebe30d20acc1cf01b94dc08ff96a2` compiled all modules and audited 31 declarations including generated equation and proof declarations.

Only `propext`, `Classical.choice` and `Quot.sound` are allowed transitive axioms. Custom assumptions, admitted proofs and native-evaluation assumptions are rejected. Negative tests inject a custom axiom and an admitted theorem and verify rejection. An exact reviewed source inventory requires review when the file surface changes. It is change control; Lean's kernel does the proof check.

Discrete operator inventory

`Adjoint_identity` proves the finite transpose identity for arbitrary real matrices. `Pressure_energy` rewrites the quadratic form as weighted squares. `Pressure_nonnegative` adds nonnegative weights. `Pressure_symmetric` establishes bilinear symmetry.

`Constant_gradient_zero` and `constant_pressure_nullspace` assume balanced columns. They establish constants in the kernel, not that constants are the entire kernel. Disconnected components and zero weights can enlarge it.

`Internal_flux_conservation` proves total incidence cancellation. `Conservative_update` proves unweighted total preservation for its exact update. Unequal physical cell volumes require separate weighting.

`Projection_residual` assumes input incidence equals time step times the declared right-hand side and derives corrected incidence from residual. `Exact_projection` additionally assumes an exact pressure solution. Neither guarantees a solution exists or an algorithm finds it.

Transport inventory

`Blend_lower`, `blend_upper` and `blend_constant` prove two-point bounds and constant preservation.

`Convex_sum_bounds` handles nonnegative finite weights summing to one. `Upwind_positive` proves one-dimensional positivity for Courant number in $[0, 1]$. `Explicit_diffusion_positive` proves one-dimensional three-point positivity for diffusion number in $[0, 1/2]$.

`Cfl_counterexample` checks a negative upwind value outside the restriction. `Interpolation_not_mass_conservative` checks the three-row matrix example of Chapter 6. Both are exact rational counterexamples proved by kernel-checked arithmetic tactics.

Indexing inventory

`Flatten_in_bounds` proves x-fastest natural-number indexing under coordinate bounds. `Staggered_count` proves the cubic face-count identity. They do not verify Rust accesses, allocator behavior or machine overflow.

What remains unproved

No theorem establishes geometry assembly, cut-cell quadrature, positive definiteness of every configured domain, pressure uniqueness, CG convergence, trajectory accuracy, liquid-volume conservation, IEEE error bounds, Rust or GPU equivalence, physical realism, runtime performance or diffusion-image fidelity. No Navier-Stokes existence claim is made.

These boundaries make the proofs useful: exact algebra removes sign and structure ambiguity while implementation and empirical claims retain appropriate evidence. Weighted cut-cell adjointness, component-aware gauges and refinement from array kernels to abstract operators are valuable future work.

Reproduction

From proofs, fetch the pinned mathlib cache, run lake build, then lake env lean AxiomAudit.lean and the negative audit script. The workflow checks unchanged dependency files. Historical success applies to its exact source revision. Changed proof sources require new qualification.

The expansion qualification and all new assumptions are recorded separately in Appendix F and `expansion/proof-qualification.json`. The integrated pinned project compiles all 42 public theorem statements and audits 60 declarations. The current source binding is `expansion/bounded-proof-qualification.json`; the earlier 33-theorem/45-declaration receipt is retained as historical evidence. BoundedPhysics derives planar first contact/clipping and dissipative work from finite-strain backward-Euler equations. Arbitrary-mesh queries, symmetric-gradient assembly and a coupled liquid implementation remain separate research work.

Reading the newer implementation bridges

The counts and revision identities above describe retained historical qualifications. They do not automatically include modules added for the current material preview or adaptive CPU scheduler. Each extension needs its own reviewed source inventory, normal pinned Lean build and allowed-axiom audit. A linked theorem statement, a native regression test and a fresh qualification receipt answer different questions.

For a first example, two particles exchange equal and opposite forces. In exact arithmetic that cancels their internal contribution to total momentum. It does not cancel gravity or a wall impulse, and it does not show that a parallel neighbor loop evaluated each intended pair. Read the assumptions, then locate the code and the independent test that connect the statement to the implementation.

Checked material-preview proof bridge

[Rheon.MaterialPreview](#) isolates several small contracts: nonnegative liquid pressure; reference ideal-gas pressure; symmetric pressure-pair coefficients; pair action/reaction; viscous work, momentum and a conditional finite-step energy bound; nozzle-profile bounds; the prescribed pressure-head identity; and a reservoir mass ledger. The [material guide](#) maps these statements to the actual formulas.

For an isolated viscous pair, nonpositive power is weaker than energy decrease after an explicit step. The latter also needs a bound on timestep, drag and inverse masses. The theorem expresses that missing condition; it does not certify the complete adaptive controller. No statement here proves the compression-energy derivative through the native density floor, multiphase interface accuracy, surface curvature, gas choking, phase change or physical calibration. Nor does a mass-ledger identity prove that source credits survive cancellation in Rust: that is a state-publication test.

Checked CPU-scheduling proof bridge

[Rheon.CpuScheduling](#) describes bounded integer work and dependency relationships. It provides a mathematical bridge for the [scheduling implementation](#), whose actual scoped pool, thread-local ownership and callbacks remain Rust execution concerns. Coverage of a finite set is not a proof of race freedom for arbitrary callbacks, and ready-wave ordering is not a throughput bound. `chunk_covers` and `chunk_unique` concern half-open intervals; `same_wave_has_no_dependency` concerns the declared nearest-axis predecessor relation. `successful_trace_explores` assumes a stable workload, serial successful invocations and exactly one counter record per invocation. It is not a fairness claim for overlapping or failed work.

The practical reading rule is the same as for pressure: keep the proven algebra, native execution evidence and measured performance distinct. A change to a source module invalidates an older source-bound qualification; it must not be hidden by reusing a theorem count or a successful historical screenshot.

Inspect the premise before the result

The pressure-proposal example derives residual $(e, -e)$ from a specific two-cell operator. A checked condition $|e| \leq t$ bounds both physical divergence components because both volumes are one. It does not establish zero divergence when $t > 0$. At $e = 1/16$ and $t = 1/8$ the condition passes with nonzero error; the Lean witness states this explicitly. A gauge shift leaves this particular pressure jump unchanged, while an error in that jump changes the corrected flux.

The browser is neither a proof assistant nor a general pressure backend. Read the theorem's operator, volumes and tolerance premises before using it for another mesh. Native source/receipt admission, floating-point refinement, model validation and runtime performance require their own evidence.

► Practical study proof scope

17 A staged Rust implementation

Production should begin in Rust, with Zig and Odin following a stable behavioral contract. This research package is not that production solver. The checked standards snapshot `dcc56f26e884ade260770beceba2501d3746200d` emphasizes owned state, explicit failures and claim-matched evidence [S20].

Core ownership

GridShape validates dimensions, spacing, origin, strides and capacity. BoundarySnapshot owns classification and prescribed data for one version. FluidState owns accepted fields and time. Workspace owns reusable scratch. Stepper produces a candidate and diagnostics. A separate renderer consumes an immutable accepted snapshot.

The numerical core should not depend on GUI, model weights or windowing. A demonstration application is one consumer. Physics configuration and presentation configuration remain separate.

Geometry and pressure first

Implement checked indexing, shapes, coordinates and boundary classification. Test uneven dimensions, minimum grids, overflow, every orientation and linear pressure ramps. Compare independently assembled small matrices with production stencils. Verify adjointness, nullspace and connected-component compatibility.

Then add a typed pressure solver, initially diagonal-preconditioned CG on a gauge-fixed system if selected. Exercise zero residual, incompatible flux, isolated cells, disconnected domains, invalid coefficients and budget exhaustion. Compare direct small solves and hand calculations. Compute residual and divergence through separate paths.

Binary64 simplifies diagnosis; binary32 may be required for memory but needs independent qualification. The real-arithmetic proofs select neither production precision nor hardware.

Transport and forces

Implement component-aware sampling, tracing, collisions and source scheduling. Bounded scalar transport precedes optional correction. Test constants, bounds, translation and limiter behavior. Gravity and constant viscosity follow with hydrostatic and decay tests. Projection and boundary reapplication follow the derived stage order, not buffer convenience.

A step publishes a complete accepted interval or an explicit incomplete outcome. Cancellation preserves accepted state and identifies completed physical time. Renderers must not see half-updated arrays.

Guidance and application

Export calibrated silhouette, depth and normals for primitives and state. Retain camera and state IDs so hosts cannot combine stale images with new metadata. A lagging renderer may be intentional but must consume a coherent snapshot.

Build the demonstration around the core. Acceptance exercises create, advance, pause, reset, parameter changes, cancellation and export. None is proved by the research scripts or Lean algebra.

Liquid and performance

Select an interface method after comparing volume drift, silhouette accuracy and memory. Add its boundary and surface tests before advertising liquid support. Then profile complete frames under intended host model allocation. Record hardware, build, scene, warm-up, variability and peak memory. Optimize measured contributors while retaining diagnostics.

Ports share fixtures, units, serialization and failure meanings. Matching signatures is insufficient. Stabilize Rust first so three languages do not evolve three interpretations of one equation.

Concrete type and function contracts

The following Rust-shaped interfaces are a design specification, not compiled library code. They make coordinate domains and ownership visible:

```
struct CellShape { nx: usize, ny: usize, nz: usize, len: usize }
struct FaceShape { axis: Axis, dims: [usize; 3], len: usize }
struct GridGeometry { cells: CellShape, faces: [FaceShape; 3],
                     origin: [f64; 3], spacing: [f64; 3] }
struct BoundarySnapshot { version: u64, cell_kind: Vec<CellKind>,
                         face_data: [Vec<FaceBoundary>; 3] }
struct FaceField { x: Vec<f32>, y: Vec<f32>, z: Vec<f32> }
struct FluidState { id: u64, time: f64, velocity: FaceField,
                   tracer: Vec<f32> }
struct PressureWorkspace { p: Vec<f64>, rhs: Vec<f64>,
                          residual: Vec<f64>, direction: Vec<f64>,
                          product: Vec<f64>, preconditioned: Vec<f64> }
```

Private constructors validate lengths and finite geometry. A `CellIndex` cannot be constructed from an unchecked integer through the public API. `FaceIndex` includes axis or comes from an axis-specific view. Avoid retaining both an authoritative len and independently mutable dimensions; derived capacities remain immutable after construction.

A proposed `apply_pressure_operator` takes immutable topology, immutable pressure and a mutable output slice of validated length. It must overwrite every active output, including isolated or boundary rows according to classification. A divergence function consumes velocity and the same boundary snapshot but uses an independent direct face-flux traversal for validation. A project function returns a `PressureOutcome` containing diagnostics and never silently reclassifies invalid geometry.

A `sample_velocity` function accepts a world position and boundary policy and returns either a finite vector with validity metadata or a typed sample failure. Advection accepts immutable old fields and separate outputs. A step function accepts a physical interval and returns either an accepted snapshot or a failure carrying the last complete time. It must not expose mutable workspace through callbacks.

Storage and invariants

Use x-fastest contiguous arrays initially. For cells, strides are $1, n_x, n_x n_y$; for x-faces they are $1, n_x + 1, (n_x + 1)n_y$. The y- and z-face strides follow their own dimensions. Validate the multiplication when building shapes, then retain strides for kernel use.

Boundary data should distinguish fluid neighbor, prescribed normal speed, prescribed pressure and inactive face. A floating coefficient of zero is not enough to encode all four meanings. PressureTopology owns reduced indices, component IDs, gauge choices and face transmissibilities. Its version depends on geometry, density policy and active-fluid classification.

State publication increments a version only after diagnostics accept the candidate. A renderer borrows an immutable accepted snapshot. Reset constructs a new coherent state rather than zeroing arrays piecemeal while consumers read them. Parameter changes that alter array shape or topology invalidate dependent workspace and preconditioners explicitly.

Executable acceptance fixtures

The companion provides expected mathematical results for a production test port:

- The three-cell chain in Appendix E must yield pressure $(0, 4, 2)$ and zero corrected internal velocity at time step one half.
- The cycle fixture must retain equal circulation rather than force every face to zero.
- All four lattice samplers must reproduce an affine scalar within precision tolerance; the intentionally wrong offset must fail.
- The pressure operator must match an independently assembled small matrix on rectangular dimensions.
- A closed component with nonzero prescribed net flux must return incompatible flux before iteration.
- A zero right-hand side must terminate without dividing by zero.
- The rational flux fixture must preserve total amount; floating ports compare with an appropriate absolute budget.
- A deliberately truncated particle support must expose moment failure rather than pass the full-support affine assertion.

Add integration fixtures for cancellation after each stage, resource rejection before mutation and stale boundary versions. Those are not mathematical experiments; they require the actual Rust state owner and host boundary.

Dependency and implementation choices

Use established sparse and numerical libraries for reference solves where licensing and deployment allow, while retaining a transparent matrix-free production operator if justified by the memory contract. Pin toolchains and dependencies for qualification, and record the exact compiler and feature set used in numerical comparisons.

No unsafe code is required for the first bounded CPU implementation. Introduce unsafe indexing or SIMD only after profiling identifies a meaningful contributor and the validated shape contract supports a reviewed safety argument. Bounds-check elimination by the compiler is preferable to a manually duplicated unchecked path when performance is equivalent.

GPU kernels require a separate buffer-layout and dispatch contract, including alignment, workgroup limits, synchronization and readback. A CPU test cannot establish a shader's binding layout or race freedom. Keep backend-independent fixtures but qualify each backend through its real execution path before advertising support.

Current material and desktop owners

The Rust-shaped interfaces earlier in this chapter remain a design specification. The current repository also has compiled owners with narrower, explicit contracts. [ParticleFluidSimulation](#) owns the bounded material preview, while the original carrier simulation and its pressure implementations remain a separate selectable mode. Switching the desktop's material mode changes the physical model; it is not a different preconditioner for the same pressure problem.

A desktop run takes an immutable copy of settings and creates one joinable worker with a scoped Rayon pool. That worker exclusively owns the numerical simulation. The renderer consumes immutable accepted `PreviewFrame` values published to history; it never borrows candidate particles. Editing settings for the next run therefore does not reinterpret the physical scale of frames already recorded. The [desktop contract](#) documents startup, cancellation, joining and retained history.

Follow one particle step through publication

The material owner first validates the requested interval and emitter. It chooses an interval constrained by its acoustic, motion and material scales, copies accepted particles to candidate storage, and prepares source parcels using candidate reservoir credits. Neighbor bins, density and forces are then computed from that candidate. A force check can request preparation again at a shorter interval. There are at most 16 preparation attempts in one call; exhausting them returns an explicit stability failure.

Only a successful preparation commits the particle arrays, accepted clock, generation, source credits and emission-sequence state. This last pair matters: if a cancelled attempt kept its source credit or disk-sampling progress, a retry could inject a different amount of material or emit it elsewhere even when the visible particles had been restored. Scratch may change during a failed attempt; accepted state must remain coherent. The [material tests](#) exercise cancellation, capacity failure and retry through the real public owner.

After commit, the worker captures the accepted state for display. Scrubbing history selects an earlier immutable frame while new frames arrive; Follow live selects the newest accepted frame. The material image splats parcels onto its stored preview grid. It is a visualization, not a full numerical checkpoint from which the particle state can be recovered exactly.

A small acceptance sequence for the new owners

Begin with a vacuum scene with no source and zero gravity: there should be no particle mass to display. A filled scene instead starts with mass determined by its physical volume and reference density. Enable the source, record actual accepted time and emitted mass, then cancel and inspect the retained frames. The measurable question is whether retained mass equals initial mass plus accepted emitted mass within the declared floating tolerance. A plausible jet image alone cannot answer it.

Next keep the physical inputs fixed while changing worker count. Compare accepted fields and reports through the numerical fixtures, not only preview pixels. The run-owned [CpuScheduler](#) may choose different batch boundaries from timing feedback, but that does not authorize different per-item arithmetic or reordered reductions. Finally raise only preview resolution and check the stated payload budget: numerical parcel spacing should remain fixed. These three exercises isolate physical state, execution order and display cost.

The [material-preview contracts](#) and [CPU-scheduling contracts](#) give the bounded formulas, source links and proof boundaries. They do not upgrade a native desktop preview into a new browser solver or establish equivalence to the paper designs earlier in this chapter.

Draft checkpoint: borrow one accepted transport snapshot

Unmerged [PR48](#), reviewed at f06a1270935b2ac285c68c0e594ab0742220913e, adds an engine-neutral [SimulationSession](#) for the existing full-box transport path. It owns the boundary workspace and borrows unchanged native step inputs for each synchronous call. A snapshot binds grid, publication number, immutable state, identity and the last accepted report. Stages distinguish constructor state from an accepted interval. Pressure is the last interval's projection pressure with its existing gauge; it is not a reconstructed end-surface pressure.

Try the draft's [headless and game-loop examples](#). Predict which clock should advance after a CFL-shortened request: the host must use accepted report dt, not requested dt. Next cancel an attempt and compare the published frame before retry. The draft's [contract tests](#) check publication preservation, retry, quotas and equivalence to existing encoding. Deadlines are checked cooperatively at native stage checkpoints and around callbacks; a long kernel or callback can overrun them. They provide no preemption, maximum wall time or frame-rate guarantee.

The snapshot borrows its owner. Retention requires an explicit consumer copy or encoding; there is no hidden frame archive. Stop is terminal but leaves inspection available. Caller-supplied run/revision strings are declarations, not verified provenance. Encoded frames alone do not satisfy the separately validated dense3D dataset-publication contract. The API is transport-only: it does not yet wrap PR47 particle diagnostics, PR45 experimental Stokes, CAD measurement or AI providers, and the Rust game-loop sample is not a Godot binding. These are separate unmerged drafts; the current embedded browser lesson continues to use its existing adapter.

18 Validation and remaining research

Formal algebra, numerical fixtures, integration tests and visual review prove different things. A proof build cannot replace camera-export validation; a beautiful render cannot replace divergence or volume checks.

Executed evidence

The companion builds a closed 8-cubed graph with 512 cells and 1344 faces. It gauge-fixes pressure and projects deterministic random velocity. CG uses 78 iterations at the chosen tolerance. Divergence L2 decreases from 49.74366159 to 4.4429×10^{-10} . Unweighted kinetic energy decreases from 642.9265376 to 401.9458778. The scalar update retains total 512 in the executed binary64 computation.

These are sparse-reference results, not Rust, cut-cell or laboratory evidence. Energy decrease is an observation consistent with projection, not a general proof. An exact rational three-cell flux example updates to 199/630, 96/175 and 61/450, summing exactly to one.

Transport and diffusion experiments are described in Chapters 7 and 10. Scripts assert stated properties and preserve data used by figures. No script timing is claimed as production performance.

Production evidence matrix

Geometry tests cover shape arithmetic, transforms and orientation. Operator tests cover nullspaces, adjointness, symmetry and compatibility. Solver tests cover residual/divergence, termination and singularity. Transport tests cover smooth and discontinuous data, variable velocity and boundaries.

Forces need rest and decay tests. Liquids need sphere transport, volume drift, merging, thin sheets, rest and contact. Rendering needs calibration, invalid depth, normals and temporal rest stability. Integration needs coherent state versions, resource rejection, cancellation and replay. Each class requires an owner and diagnosis.

Convergence and performance

Report grid, time step, norm, duration, boundaries and reference with each error plot. Several refinements reveal whether slopes stabilize. Coarse points can be pre-asymptotic and fine points can reach solver or rounding floors; neither should be discarded without explanation. Limiters and topology changes can lower observed order.

Choose representative sparse and dense scenes, obstacles, sources and output resolution. Measure simulation, pressure, rendering, readback and full-frame distributions, plus peak memory with host model allocations. The README supplies no numeric budget or target hardware. Until chosen, memory arithmetic is an estimate and real time remains an objective.

Next research

The highest-value integrated study links coarse simulation error to spatial conditioning quality. Measure how displaced interfaces, missing thin features and flicker affect a pinned diffusion pipeline. More simulation detail may consume memory without improving final guidance.

Numerical priorities are component-aware pressure, obstacle sampling and bounded liquid volume. Formal priority is connecting array assembly to the abstract model. Non-Newtonian stress, APIC and sparsity follow needed capability rather than the project's name.

Every result should retain its level of certainty: derived identity, checked theorem, executed fixture, measured implementation or proposed design. That distinction makes a small reusable solver dependable.

Match the benchmark boundary operator

The [research record](#) works through the official steady DFG Re20 specification. Its outlet uses a velocity-gradient traction; substituting full symmetric traction changes the problem. Before comparing a drag number, reproduce the declared geometry, normal convention, inlet, boundary operator and pressure samples. Reference values are not Rheon measurements. The linked record separates remaining spatial, iteration and model-validation requirements.

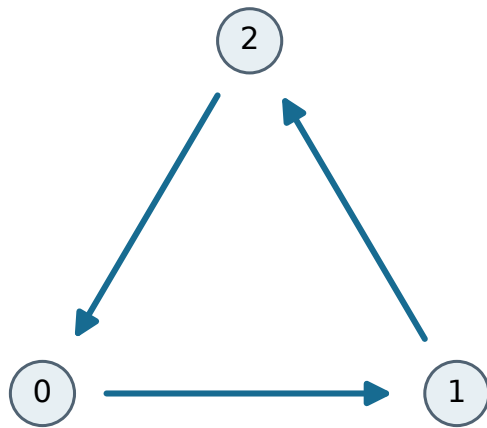
Midpoint refinement with a known answer

Integrate y^2 over $[0,1]$ by midpoint quadrature. The exact answer is $1/3$. With n cells the independently derived sum is $1/3 - 1/(12n^2)$. The browser actually samples and accumulates the rectangles, then computes three grids n , $2n$ and $4n$. At $n = 4$ their values are 0.328125, 0.33203125 and 0.3330078125. Successive differences give observed order 2; Richardson extrapolation gives $1/3$. The fine-grid absolute GCI estimate with factor 1.25 is approximately 0.000406901. The rectangles depict the selected coarse quadrature; the dots and curve show the exact integrand. [NASA's spatial-convergence procedure](#) supplies the GCI context. Here the smooth quadratic fixture establishes the error model directly; arbitrary CFD data require a separate asymptotic check.

Refining a biased model

Add a constant $1/16$ to every sampled integrand. The observed order remains 2 and the GCI estimate is unchanged, but extrapolation now gives $1/3 + 1/16$. The spatial estimator measures convergence to this biased model, not agreement with the original exact target. Report model/input/measurement discrepancy separately from discretization, iteration and temporal error. A GCI safety factor is not a probability statement or a universal bound on physical truth. Nonmonotonic differences, unresolved changes, varying ratios and nonsmooth solutions need explicit refusal or an appropriate alternative analysis. The control domain here intentionally contains only the two stated smooth fixtures; it does not silently fit an order to arbitrary results.

► **Practical study proof scope**



Before (3, 0, 0)

After (1, 1, 1)

A cycle can retain
zero-divergence flow

Original reference illustration for 18 Validation and remaining research

19 Forces and moving-boundary work

The accepted smoke demonstration supplies one localized acceleration in a fixed box. The separate [explicit body-force API](#) now distinguishes acceleration and force density in the fixed-box carrier. The [native wall/force progression](#) adds finite wall traction, compatible no-slip and uniform forcing in a separate fixed slab. This chapter's prescribed moving-solid and two-way coupling requirements remain research specifications. The first decision is whether motion is prescribed or dynamically coupled. A scripted paddle can supply arbitrarily large work without losing momentum. A dynamic paddle requires a body mass, inertia, reaction force and compatible time integration.

Units before parameters

Use metres, seconds and kilograms internally. Density ρ is kg/m³, dynamic viscosity μ is Pa s, and kinematic viscosity $\nu = \mu/\rho$ is m²/s. An acceleration a changes velocity by $\Delta t a$; a force density f , in N/m³, changes it by $\Delta t f/\rho$. A source of liquid volume is neither of these. Its addition must appear in the volume ledger and pressure compatibility condition.

For a face mass $m > 0$, an acceleration-only update satisfies

$$\frac{1}{2}m(u + \Delta t a)^2 - \frac{1}{2}mu^2 = m\Delta t a u + \frac{1}{2}m\Delta t^2 a^2.$$

The second term explains why logging old-velocity power alone does not reproduce a finite explicit step's energy change. `force_work_identity` checks this exact scalar identity. The reference fixture verifies it for a signed acceleration; it makes no general stability claim.

One geometry interval

A rigid prescribed boundary has world velocity $u_s(x) = v + \omega \times (x - x_c)$. Retain pose, translation, rotation and version for the interval consumed by transport, pressure, viscosity and output. Sampling only the final mesh pose can miss a swept collision. Sampling wall velocity from one pose and cut volumes from another breaks displaced-volume accounting even if pressure residual is tiny.

For an outward fluid normal, sealed moving-wall flux is positive when fluid-domain volume grows. Reynolds transport gives $\sum_e Q_e = \dot{V}$, with $Q_e = A_e u_s \cdot n_e$. Incompressibility on a moving control volume uses this geometric volume change, rather than imposing a zero sum of all wall fluxes blindly. A translating obstacle inside a fixed sealed container preserves total fluid-domain volume; a piston moving inward does not unless a compensating outlet or source exists.

For fixed-volume cell constraints choose an outward-positive internal incidence B (the negative of Chapter 16's head-positive pressure incidence) and write $Bu + Q = s$, where internal incidence columns sum to zero and s is an integrated volume source. Summing yields the necessary component condition $\sum Q = \sum s$. `sealed_flux_compatibility` proves this implication under explicit balance and solved-constraint hypotheses. It does not prove a solution exists. Moving-cell schemes must additionally supply their geometric conservation law.

Pressure and external work

Prescribed boundary data make divergence affine. The weighted adjoint relation includes pressure work $p^T Q$, as Chapter 8 derives. Separate known boundary terms before testing a homogeneous transpose identity. For two-way coupling use the same normals, areas and pressure interpolation for wall flux, force and torque. Otherwise an apparently correct pressure solution can exchange inconsistent momentum. The variational solid-fluid paper provides the primary coupled formulation [E1].

The original three-cell amount ledger stores balanced and incompatible prescribed flux vectors. Their sums illustrate the necessary component condition; the separate 27-cell projection reference executes a dense pressure solve. No moving-wall pressure solve is claimed. The 3D mesh lab shows prescribed translation and the velocity at a selected collision; it is not a moving-solid fluid solve.

Implementation contract and experiments

Proposed ForceField distinguishes acceleration from force density; BoundaryInterval owns immutable start/end geometry and motion. Reject nonfinite input, stale geometry, unclassified boundary and incompatible component source before publication. An accepted report retains force work, wall work, source volume, physical time and geometry version. Failure leaves the previously accepted state coherent.

Begin with gravity and a flat tank. Then use a translating closed obstacle, a rotating paddle and a piston with a specified outlet. Compare volume change against integrated boundary flux, include both signs of wall motion, and refine temporal sampling. Dynamic light-body added-mass tests belong to a later two-way implementation; prescribed-body examples do not establish them. Mesh query accuracy and liquid volume transport remain separate obligations even when the exact algebra passes.

Work in a frozen-geometry operator

Let M be a positive mass matrix, K a nonnegative strain operator, C a tangential sample map, and Λ a nonnegative wall-friction matrix. With wall samples w , generalized force f and equation residual e , suppose

$$M(v - u) + \Delta t \{ K v + C^T \Lambda (C v - w) - f \} = e.$$

For $r = C v - w$ and kinetic energy E , the finite work budget is

$$E(v) - E(u) + \frac{1}{2} \|v - u\|_M^2 + \Delta t v^T K v + \Delta t r^T \Lambda r = \Delta t v^T f - \Delta t w^T \Lambda r + v^T e.$$

The minus sign in actuator work follows from wall force $-\Lambda r$ on the fluid. Laboratory-frame energy can increase while relative friction dissipates. The residual must be taken from this unscaled equation, with its norm and normalization declared. This richer matrix statement is a next proof target, numerically checked in small synthetic systems; it is not one of Physics.lean's thirteen compiled contracts. Prescribed normal-wall pressure work requires its own affine constraint term.

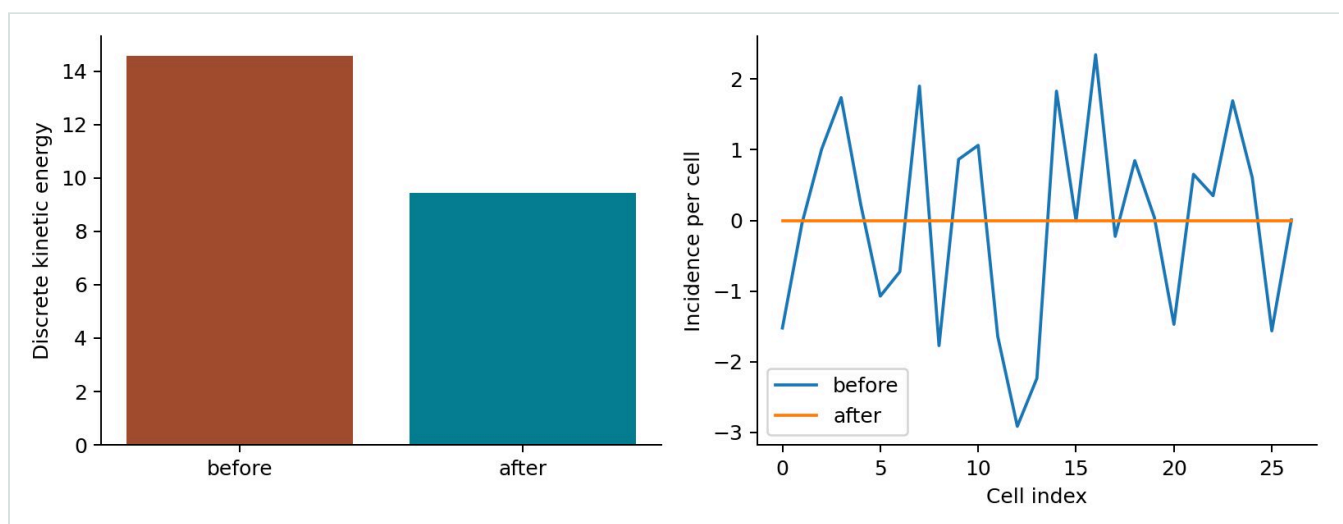
An affine projection $M(v-u) + A^T \lambda = 0$, $A v = b$ gives $E(v) - E(u) + \frac{1}{2} \|v - u\|_M^2 = -\lambda^T b$. Here λ is impulse-scaled. A homogeneous stationary constraint has zero right side; a moving constraint supplies work. Feasibility is separate from pressure gauge uniqueness. For two-way solids, combine fluid and body masses and use one adjoint constraint J .

Common translation conserves the corresponding generalized momentum only if J annihilates that mode; angular momentum additionally needs rotation modes and consistent interaction points. These are frozen-geometry specifications, not completed moving-solid proofs.

Signed work at a moving wall

Use the steady wall-load example with $\mu = U = 1$ and $G = -2$. Traction on the fluid are bottom 0 Pa and top 2 Pa; mean velocity is 1/3 m/s. The moving wall supplies 2 W while the opposing body force removes 2/3 W, leaving dissipation 4/3 W. Work is signed: body forcing need not inject energy. This one-metre, unit-area solution is a teachable traction and power oracle, not a moving mesh or two-way coupled body. At general boundaries use total Cauchy traction, including pressure, and distinguish the normal pointing out of the fluid from the normal pointing out of a solid. Prescribed outlet traction also needs a compatible inflow/backflow policy and pressure reference; importing a benchmark with a different boundary operator changes the problem.

► Practical study proof scope



Original reference illustration for 19 Forces and moving-boundary work

20 From collision meshes to fluid boundaries

A rendered triangle soup is not automatically a closed collision solid. Rheon's proposed boundary pipeline accepts an oriented watertight mesh in a declared world transform, or a deliberately separate thin-shell model. It does not invent an inside for open edges. Static and prescribed rigid motion are the first research targets; deforming meshes require vertex velocities and swept geometry.

Admission and representation

Validate finite vertices, valid triangle indices, nonzero triangle area at the declared tolerance, edge incidence, coherent orientation and supported topology. Self-intersections invalidate a simple signed-volume interpretation. Nonuniform scale affects distances and normals: transform a normal with the inverse transpose and normalize after checking its length. Negative determinant transforms require orientation handling. Record original topology and transformed geometry hashes.

An unsigned nearest distance comes from the closest point on triangles. Its sign requires an inside/outside policy, such as consistently oriented winding or parity with robust edge/vertex rules. A signed distance used for classification is a derived numerical field, not an authoritative replacement for the mesh. Thin features can disappear when sampled at cell centers; face openings and cell volumes need their own geometry queries.

Bridson's author-hosted mesh-query and geometric-predicate implementations provide relevant original implementation references [E2]. Their availability does not prove Rheon's independently written code. The present companion implements segment/triangle intersection and closed-cube tests directly, with explicit finite tolerances. It does not advertise a production robust-predicate package.

Earliest path collision

For a segment $x(t) = x_0 + t(x_1 - x_0)$, seek the smallest hit $t \in [0, 1]$. A triangle test solves the barycentric system with two in-plane coordinates and segment parameter; parallel or nearly degenerate configurations need a specified classification. Selecting the nearest endpoint distance is insufficient. An endpoint can be outside the solid while the segment crosses it twice.

The original fixture uses a triangulated closed cube, traces an outside-to-outside segment and finds its first hit at $t = 0.3125$. It also checks a miss and a trace beginning inside. The browser shows the complete segment, earliest hit and triangle mesh, so the evidence remains visible rather than hidden behind a texture effect. A boundary control translates the geometry and uses the corresponding regenerated hit records.

At a hit, collision response depends on what is transported. A passive tracer trace may stop at the surface; a liquid interface needs its own boundary/contact-angle extension; a velocity sample needs prescribed normal motion plus the chosen tangential law. Repeatedly clamping a point can pin a contact line unintentionally. No-slide and no-slip mean different operations.

Checked planar first contact and clipping

The mesh-query fixture and the exact collision contract have different domains. The new `BoundedPhysics.lean` contract uses an infinite stationary plane in three dimensions. Its oriented plane value is

$$g(x) = \sum_{i=0}^2 n_i x_i - c, \quad x(t) = (1-t)a + tb.$$

The permitted half-space is $g(x) \geq 0$. The normal points into that half-space and need not have unit length; g is a sign/plane-value function, not necessarily a signed distance. Assume strictly $g(a) > 0$ and $g(b) < 0$. These hypotheses exclude a zero denominator and imply

$$t_* = \frac{g(a)}{g(a) - g(b)}, \quad 0 < t_* < 1.$$

`wall_segment_affine` derives $g(x(t)) = (1-t)g(a) + tg(b)$ from the actual finite-vector definitions. `wall_hit_range` proves the range of t_* . `wall_hit_on_surface` proves $g(x(t_*)) = 0$. `wall_first_hit` proves every $0 \leq t < t_*$ remains strictly in the permitted side. Finally, `clipped_segment_in_halfspace` proves every point on the segment from a to $x(t_*)$, parameterized by $0 \leq s \leq 1$, has nonnegative plane value, including contact. The hit and clipping conclusions are derived, not assumed.

The exact-rational witness uses plane values 1 and -1 , giving $t_* = 1/2$. The unclipped continuation at $t = 3/4$ has value $-1/2$. If the proposed end instead has value $1/2$, the formula gives $t_* = 2$ outside the segment; if the start/end values are -1 and -2 , it gives $t_* = -1$. The endpoint sign assumptions are therefore essential. Starting on the plane, grazing, same-side traces and moving planes require separate policies.

This proves a bounded planar geometric contract. It does not prove triangle containment, ordering among mesh facets, watertightness, cut-cell construction or floating-point robustness. A finite-facet query must establish both that its intersection lies inside the facet and that no earlier relevant facet was missed. The closed-cube browser fixture remains an independently executed numerical query, not an implementation refinement proved by these planar lemmas.

Cut geometry and pressure

For a face opening A_e , fluid control volume V_i and sample separation ℓ_e , a candidate pressure weight is $w_e = A_e / (\rho_e \ell_e)$. Positive weights preserve the finite energy structure. Shared openings must be computed once for both adjacent cells. Independent face estimates can violate cancellation. The scalar volume and face metrics must agree with the chosen divergence/gradient pair, not merely look plausible.

Tiny volumes create stiffness and amplify residual-to-divergence scaling. A minimum fraction, cell merging or topology removal is a model policy requiring comparison against unclipped geometry. A binary staircase is a valid educational baseline with a disclosed geometry error; it is not continuous collision-mesh support. Basilisk's original embedded-boundary source exposes how geometry enters gradients, interpolation and small-cell handling [E3].

Qualification ladder

Compare cube intersections against hand arithmetic, then oblique planes and spheres against analytic distances. Add grazing rays, edge hits, reversed winding, repeated triangles, very small and large scales, and subcell wall offsets. Use refinement to distinguish geometry error from linear-solver error. Validate moving traces in relative coordinates for rigid translation, then rotation against an independent swept reference.

The research admission contract returns `UnsupportedOpenMesh`, `DegenerateTriangle`, `AmbiguousSign` or `UnresolvedThinFeature` where appropriate. It does not silently substitute a box. The finite Lean conservation contracts assume a valid assembled incidence and shared fluxes; they do not verify these geometry predicates. A complete mesh-to-cut-cell implementation remains future production work.

Mesh validity is part of the numerical method

A render mesh is not automatically a valid collision domain. The conversion should produce an explicit report containing:

- Unit scale, coordinate transform, winding convention, and sign convention.
- Watertightness and manifold checks for volumetric obstacles.
- Degenerate triangles, self-intersections, duplicate faces, and open boundaries.
- Minimum feature/gap scale relative to Δx , particle spacing, kernel support, and collision thickness.
- A declaration that a surface is a finite-volume solid, a zero-thickness shell, or a intentionally thickened approximation.

For a closed solid, a signed distance function can describe inside and outside if its sign construction is valid. A zero-thickness open sheet has no enclosed solid volume. A sub-grid sheet can disappear entirely if an algorithm samples only whether cell centers lie inside it. Inflating it avoids that disappearance but changes displaced volume, passage width, wetted area, and pressure forces.

Guendelman, Selle, Losasso, and Fedkiw’s thin-shell method uses ray-based visibility information to prevent interpolation and finite-difference stencils from crossing an infinitesimally thin triangulated barrier. It also treats incompressibility near that barrier. This is the important conceptual precedent: detect which samples may communicate across the shell, rather than treating every nearby sample as a neighbor. [R18](#)

For cut cells, store fluid cell volumes and open face areas. Also represent connectivity: an internal baffle can split one Cartesian cell into disconnected fluid pieces that cannot share a single pressure unknown without introducing a false connection. Scalar volume fractions alone are not enough for all sub-cell topologies. The correct fallback may be refinement, multi-region cut cells, or rejecting the feature as unresolved.

Small cut cells introduce their own explicit transport restriction. When fluid volume tends to zero faster than an open face area, the local transit time tends to zero. Basilisk’s embedded-boundary code documents this issue and implements a redistribution approach for tracer overflow. Any adopted merge/redistribution rule needs separate conservation, positivity, accuracy, and barrier-connectivity checks. Clamping a tiny volume to a larger value is a model change unless the resulting discrepancy is accounted. [R19](#)

Moving and deforming walls

For a rigid wall, evaluate local velocity at the actual interaction point:

$$U_w(x) = V + \Omega \times (x - c).$$

A single object translation velocity is incorrect for a rotating paddle. Use the same point and frame for wall traction, solid force, and solid torque. For a two-way solid, accumulate equal-and-opposite impulses and the moment arm. For a prescribed wall, accumulate the actuator work; do not report fluid energy growth as a solver defect without including it.

When geometry moves, a cut cell can be uncovered or swallowed. Update swept volumes and associated mass/momentum consistently. For a space-time control volume, a discrete geometric conservation law is the prerequisite for transporting a uniform field without generating a fictitious disturbance. Reinitializing newly uncovered cells by arbitrary extrapolation may be robust, but it is not automatically conservative or work-consistent.

Collision detection for advected particles should consider the segment or swept trajectory, not only its endpoint, whenever a particle can traverse a thin wall in one substep. Surface-only pushout can repair position while leaving pressure fluxes, density estimates, and wall work wrong. Do not use a smoothed rendering normal as the geometric contact normal without analyzing the resulting gap and traction error. Nonuniform scaling also requires correct normal transformation and a recomputed distance metric.

Precomputed rigid-boundary maps may be transformed with a rigid body; arbitrary deformation changes the precomputed geometry. A deforming mesh needs an updated representation and checks for new self-intersections or inverted elements. A narrow band must cover the actual query reach, including kernel support and swept motion, not merely the visible surface.

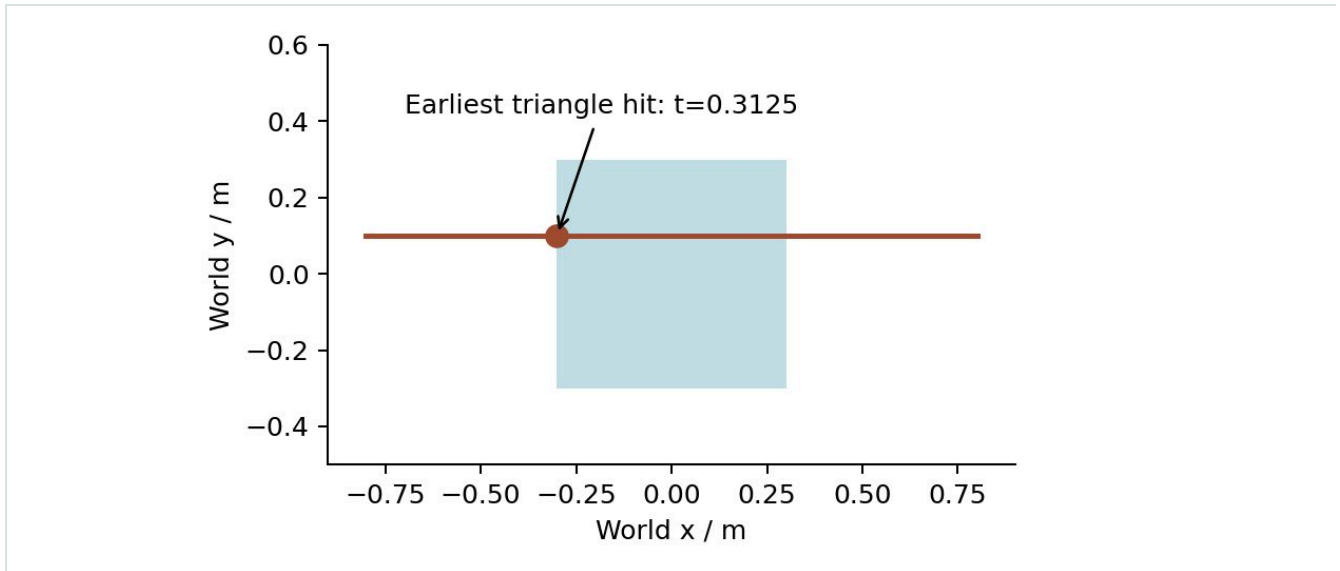
Implemented bounded static owner

The [static-obstacle geometry feature](#) now retains one admitted closed axis-aligned triangle box for collision, cell volumes, uniquely shared face openings, connectivity and conservative flux. Its exact-real overlap and complement formulas are checked in [StaticObstacle.lean](#); Rust evaluates represented world endpoints with explicit arithmetic refusal. A subcell box spanning two full axes can split one cell into two fluid pieces, so that topology is refused. Existing pressure and viscosity owners do not consume this geometry. The matching lab inspects recorded native controls; it advances no fluid.

Current native import and the remaining fluid geometry gate

The reviewed runtime composition includes static OBJ/gLTF/GLB import and a CollisionMesh with either validated convex-solid or explicitly two-sided-surface policy. Native file loading, triangle/resource validation and collision queries are separate capabilities. The fixed WASM browser adapter has no filesystem importer. The particle scene can use the static collider; the existing bounded obstacle laboratory uses its own admitted box geometry and operator domain. Neither path turns arbitrary triangles into shared fluid cell volumes, face openings, connected pressure components or embedded symmetric-stress interactions.

Use the [current requirements map](#) to choose the supported native exercise, then return to chapter 8's face-flux and reconstruction studies to identify the missing conservative assembly. A visually correct surface and a passed segment query cannot supply that assembly. The [integration review](#) keeps source admission and fresh numerical qualification explicit.



Original reference illustration for 20 From collision meshes to fluid boundaries

21 Density and real liquid interfaces

A liquid state needs more than a smoke concentration. Its interface locates a liquid domain, its density supplies inertial mass, and its volume ledger records transport and sources. Changing the grayscale opacity coefficient does not produce a liquid. This chapter specifies an incompressible, immiscible Newtonian research model. Compressible flow, phase change and dissolved materials are separate extensions.

Constant material and multiple phases

A single liquid with negligible air dynamics can use constant positive ρ_l in active liquid cells and a pressure boundary at the interface. A two-phase model solves momentum in both phases with densities $\rho_l, \rho_a > 0$. These choices have different pressure masks and inertia. The choice must remain visible in state and output metadata.

For variable density, the pressure coefficient uses inverse density at the pressure connection. Across a one-dimensional layered connection of lengths ℓ_1, ℓ_2 , integrating $\partial_x p = \rho a$ gives effective $1/\rho_e = (\ell_1 + \ell_2)/(\rho_1 \ell_1 + \rho_2 \ell_2)$. This is the harmonic transmissibility of the inverse-density coefficient, not a universal average independent of geometry. Volume-fraction mixing is another declared approximation. Use the same coefficient in assembly and velocity correction.

`positive_transmissibility` proves positivity from positive area, density and distance.

`weighted_pressure_energy` proves the corresponding weighted-square sum is nonnegative; Chapter 16's existing matrix identity explains its operator interpretation. Neither identifies the physically correct density interpolation at an unresolved interface.

Hydrostatic oracle

Take upward y , gravity $-g$, and free surface at H . For piecewise layers,

$$p(y) = p_a + g \int_y^H \rho(s) ds.$$

A discrete face increment $p_{j+1} - p_j = -\rho_{j+1/2} g \Delta y$ cancels a gravity step when the same face density is used. The reference tank has a lower layer of density 1000 and an upper layer of 800 kg/m³, each 0.5 m deep. With $g=9.81$ m/s² its bottom gauge pressure is 8829 Pa. These are illustrative parameters, not a measured material calibration or a stable stratified-flow simulation.

The educational tank displays the stored layer density and pressure samples. Increasing density changes pressure and inertia; it does not change equilibrium cap geometry at zero gravity. That distinction prepares the later capillary lab. `hydrostatic_increment` checks the exact cancellation identity for a prescribed face balance; it does not prove long-time rest after transport, interface reconstruction or rounding.

Conservative represented volume

Let $m_i = f_i V_i$, with $0 \leq f_i \leq 1$. Update $m_i^+ = m_i - \Delta t (BF)_i + \Delta t s_i$ using one signed shared face flux. In this amount ledger B is outward-positive (positive at the face tail, negative at its head), the negative of Chapter 16's pressure incidence. The conservation result is unchanged by this global sign choice. Balanced incidence columns imply

$$\sum_i m_i^+ = \sum_i m_i + \Delta t \sum_i s_i.$$

`volume_source_balance` proves this identity in exact arithmetic for arbitrary finite cell volumes. The proof operates on represented amounts; it does not establish fraction bounds or determine geometrically correct fluxes. Post-update clamping can violate the identity. The original three-cell volume fixture checks unequal volumes and a nonzero source using rational arithmetic before exporting floating values.

Chapter 9's axis-aligned PLIC example remains a local geometric oracle. Full three-dimensional bounded conservative transport needs one coherent multidimensional outflow scheme, consistent reconstruction normals and source treatment. A level-set branch instead needs measured volume drift and an explicit correction policy. Neither is selected merely because it yields a smoother surface image.

State and failure contract

Proposed `LiquidState` retains interface representation, positive material parameters, volume/source totals, pressure-unknown mask and velocity-validity mask. Those masks are distinct. Air velocity extension supports transport without making extended air cells physical liquid. Rebuild topology and preconditioners when classification or density policy changes. A failed interface reconstruction must not publish pressure and geometry from different versions.

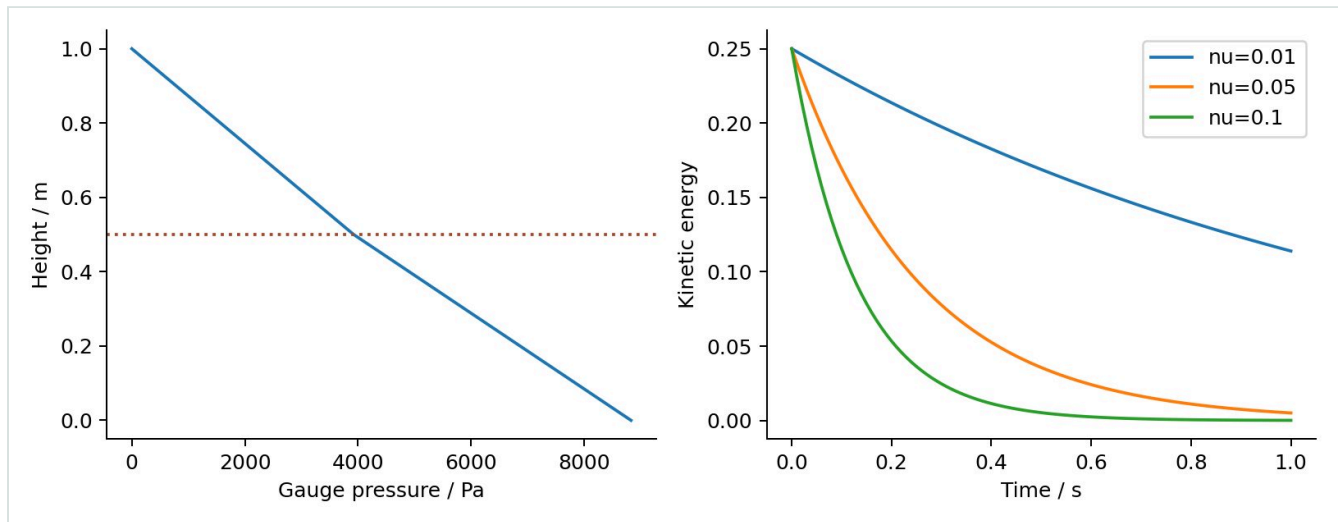
Acceptance progresses through hydrostatics, translating sphere, oblique interface transport, merging volumes and thin films. Report both represented volume error and spatial interface error; equal volume does not establish equal shape. Two-phase density-ratio tests additionally require momentum-transfer and pressure-jump checks. The original VOF and level-set works remain primary method references [S8–S10], while the present fixtures are independently constructed.

Holding one material quantity fixed

Increasing density at fixed dynamic viscosity lowers $v = \mu/\rho$; increasing density at fixed v raises μ . A single-phase prescribed gravity acceleration stays the same, while hydrostatic pressure and inertia change. A fluid material API must record which viscosity quantity is supplied. The inspected original `SPlisHSPlasH` Weiler2018 code multiplies its configured viscosity by rest density to obtain μ [R14](#). Copying a GUI value between solvers without converting its definition is therefore unsafe as a material comparison.

Represented particle mass, reference density, estimated density and particle count also differ. For uniform reference sampling $m = \rho_0 V_0$. Removing particles removes represented mass unless explicitly accounted as boundary/source flux. A planar 2D formulation must declare an out-of-plane thickness or use per-unit-depth quantities; a 3D coefficient cannot be transferred by a screenshot comparison.

For pressure-driven planar Poiseuille flow with $G = -dp/dx > 0$, $u(y) = Gy(H-y)/(2\mu)$ and flow per width $Q' = GH^3/(12\mu)$. The equivalent acceleration is G/ρ , with units m/s^2 . The [bounded native forced-column operation](#) now supplies this force/density/viscosity cross-check using a uniform body force, without solving a pressure gradient. At fixed G the steady speed depends on μ while density changes viscous relaxation; at fixed acceleration the effective force density changes with ρ . Its recorded startup lab compares the actual explicit algorithm against both the continuum steady profile and the distinct endpoint-basis equilibrium, plus a closed-form exact-real discrete transient. This fixed-slab operation does not qualify arbitrary channel meshes or variable-density liquid dynamics.



Original reference illustration for 21 Density and real liquid interfaces

22 Viscosity, stress and dissipation

Density and viscosity have different roles. Density weights inertia; dynamic viscosity relates deformation rate to stress; kinematic viscosity controls a simple diffusion time scale. Treating a viscosity slider as an arbitrary velocity damping factor hides both the material units and boundary work. The first liquid research model is Newtonian, with positive density and nonnegative viscosity.

Symmetric strain

For $D = (\nabla u + \nabla u^T)/2$, Newtonian stress is $\tau = 2\mu D$. Its power dissipation is $2\mu D : D \geq 0$. The cross derivatives matter: rigid rotation has zero symmetric strain even though velocity gradients are nonzero. A scalar component Laplacian is equivalent to the incompressible constant-viscosity interior equation only under the matching assumptions and boundary terms.

Batty and Bridson's original viscous-free-surface paper separates normal and tangential traction and builds a compatible variational discretization [E4]. Their later variable-coefficient Stokes formulation supports irregular moving boundaries [S11]. Basilisk's original viscosity implementation is another useful code reference for coupled stress components [E5]. These sources motivate a choice; they do not qualify Rheon's future stencil.

Finite dissipative update

Let face masses form positive diagonal M , let E map face velocity to strain samples, and let nonnegative quadrature coefficients form W . Put $K = E^T W E$. Homogeneous stationary boundaries and zero external force give backward Euler

$$M(v - u) + \Delta t K v = 0.$$

Taking the inner product with v and completing a square yields

$$\|u\|_M^2 - \|v\|_M^2 = \|u - v\|_M^2 + 2\Delta t v^T K v \geq 0.$$

`implicit_energy_identity` checks the finite identity from an explicit scalar work equation.

`implicit_energy_nonincrease` adds nonnegative time and dissipation to obtain the inequality. Its finite sums use arbitrary positive masses, so variable density is represented without assuming uniform kinetic weights. It proves neither that an iterative solve reaches the work equation nor that E is the actual symmetric-gradient stencil.

Moving walls and body forces add work on the right. Energy need not decrease then. A diagnostic should compare measured energy change against viscous dissipation and declared external work, with residual and rounding budgets, rather than reject every energy increase.

Checked bridge from strain to the implicit equations

The earlier Physics contracts require a scalar work identity. The new BoundedPhysics contracts derive that identity for a defined fixed finite-strain operator. For arbitrary finite dimensions n and m , let E map velocity degrees of freedom to strain samples, and let μ_e be fixed material/quadrature weights. Define

$$(Ev)_e = \sum_i E_{ei} v_i, \quad (Kv)_i = \sum_e E_{ei} \mu_e (Ev)_e, \quad \mathcal{D}(v) = \sum_e \mu_e (Ev)_e^2.$$

Thus $K=E^T \text{diag}(\mu)E$. `viscous_work` derives $\sum_i (Kv)_i v_i = \mathcal{D}(v)$. `dissipation_nonnegative` derives $\mathcal{D}(v) \geq 0$ when every weight is nonnegative. The exact coordinate backward-Euler hypothesis is

$$m_i v_i + \Delta t (Kv)_i = m_i u_i \quad \text{for every } i.$$

`backward_euler_work` derives $\sum_i m_i (u_i - v_i) v_i = \Delta t \mathcal{D}(v)$ directly from these equations.

`backward_euler_energy_nonincrease` then applies the existing finite energy theorem: with $m_i \geq 0$, $\Delta t \geq 0$ and $\mu_e \geq 0$, $\sum_i m_i v_i^2 \leq \sum_i m_i u_i^2$. The coordinate update and positive-weight hypotheses replace an assumed dissipative-work conclusion.

The theorem permits zero masses and weights, so it does not prove invertibility, existence or uniqueness. E is arbitrary and fixed during the step; no theorem here identifies it with an assembled symmetric-gradient stencil or proves a rigid-rotation null mode. This is an unforced homogeneous step. Moving-wall work, free-surface traction, nonlinear coefficients, approximate solves and IEEE rounding remain outside its statement.

For the exact witness $E=[-1,1]$, $\mu=2$, unit masses, $\Delta t=1/4$ and $u=(1,-1)$, the computed update is $v=(1/2,-1/2)$. The weighted squared norm falls from 2 to 1/2; kinetic energy with its 1/2 factor falls from 1 to 1/4. The derived work is $1/2=\Delta t \cdot 2$. A negative weight $\mu=-1/2$ instead gives squared norm 32/9, while an arbitrary candidate not satisfying the update can also increase it. These are exact rational witnesses, not liquid simulations or material calibration.

Shared shear-mode oracle

The reference code constructs a 32-cell periodic difference matrix with spacing $1/32$ m. A sine mode has discrete eigenvalue $\lambda_h = 4 \sin^2(\pi/32)/h^2$. With $\Delta t = 0.01$ s and ν chosen from 0.01, 0.05 or 0.1 m²/s, backward Euler damps amplitude by $(1 + \nu \Delta t \lambda_h)^{-n}$.

An independently assembled dense solve verifies the modal formula at each step. The browser displays the resulting velocity arrows in a three-dimensional slab and reports mass-weighted kinetic energy. The same profiles and energy records generate the book figure. A large implicit step remains damped but does not reproduce exact exponential decay; the error is shown rather than treated as a performance improvement.

Boundary and material tests

The [filled-box Newtonian implementation](#) uses symmetric strain under stationary free-slip box boundaries. The separate [native wall/force progression](#) qualifies fixed layer-uniform Navier/no-slip walls and external force, with an explicit simultaneous step. Its explicit squared-increment work correction differs in sign and interpretation from the

backward-Euler decay identity above. Neither implementation supplies arbitrary embedded/free-surface stress coupling.

Use affine shear to check the stress law and rigid rotation to check the strain nullspace. Add periodic decay, no-slip channel flow, prescribed-wall Couette flow and free-surface traction. The first two are local operator tests; a complete free surface additionally needs pressure and stress coupling. Explicitly record whether a split assumes zero normal viscous stress.

Generalized Newtonian and viscoelastic materials remain later research. A shear-dependent viscosity needs an admissible range and nonlinear residual; viscoelasticity needs history and its own stress evolution. The existing finite-budget nonlinear specification in Chapter 10 remains valid. This expansion does not quietly turn it into an implemented material law.

A stable operator can damp the wrong mode

Positive semidefiniteness only says that a quadratic energy decreases. It does not imply that the stencil preserves rigid rotation. For Ω about the z axis, points at $(-1,0,0)$ and $(1,0,0)$ have velocities $(0,-\Omega,0)$ and $(0,\Omega,0)$. Pairwise vector-difference smoothing sees a difference of magnitude $2|\Omega|$ and penalizes it, although the continuum symmetric strain of the rigid field vanishes. A future assembly theorem must show $Br=0$ for sampled rigid translation and rotation under explicit geometry assumptions. Only then does $K=B^T W B$ annihilate those modes.

Isolate viscosity for that fixture: a fully rotating drop still needs centripetal pressure, advection and capillary shape dynamics. Check an affine shear field separately, so a stencil that simply applies zero force everywhere cannot pass. At a free surface, normal and tangential traction belong to the stress condition; independent scalar diffusion with convenient component boundary conditions can violate it even when the interior formula matches $\mu \Delta u$.

For comparison with particle methods, the original Weiler et al. viscosity work evaluates conservation, spatial/temporal refinement and spurious viscous forces [R12](#). Those criteria transfer as questions to ask of a grid method; its published performance and accuracy do not transfer as Rheon results. PBF positional density constraints, artificial pressure and postprocessed smoothing offer a different real-time model [R15](#). Label their coefficients by their actual equations instead of silently assigning μ or σ units.

Draft checkpoint: stationary-box Stokes acceptance

Unmerged [PR45](#), reviewed at [bbc6089888e257962b968e70ed84ec61b7668d5b](#), implements an opt-in [aligned Stokes experiment](#). It combines bounded explicit symmetric-strain decay with a pressure proposal on a fixed aligned-box domain. The obstacle is stationary and no-slip; outer walls are sealed and free-slip. It is not the production liquid timestep or a moving-body solver, and this book branch does not import that implementation.

The [independent rational controls](#) construct a cell cycle around all four sides of the central obstacle in a 3^3 grid with 26 wet cells. With unit density and viscosity, initial energy 1 J and $dt = 1/32$ s, the ideal discrete step ends at **731/1024 J** with exactly zero cell flux. Pressure is zero by symmetry in this case. Predict the energy loss (293/1024 J), then compare with the separate asymmetric curl: it develops nonzero intermediate flux and needs a nonzero pressure correction. The ring alone does not demonstrate the ability to solve a nontrivial pressure field.

The qualification driver enumerates 13 one-step cases (eight accepted specimens and five refusal specimens) plus two three-step sequences. These are a finite acceptance roster, not 15 successful runs or a convergence study. During book coordination, ten exact mathematical controls were independently rerun in normal and optimized Python; native execution and the draft owner's source-bound native receipts remain separate evidence. Exact oracle flux is not a claim that every accepted binary64 field has identically zero divergence.

The [experimental Lean statements](#) assume sound enclosures of energy changes, flux and momentum defects. They derive conditional acceptance implications; they do not establish Rust/IEEE enclosure soundness, compiler correspondence or continuum convergence. They are outside this branch's root Lean inventory and theorem counts. The native interval path has explicit Linux x86_64/default floating-environment premises and can refuse unsupported or unresolved arithmetic. An unenclosed scalar stability estimate cannot authorize an actual step.

The experiment has no advection, source forcing, inflow/outflow, free surface, variable material, capillarity or moving geometry. It exposes no production clock/generation owner and is not integrated into PR48's transport facade. Calibrated wall loads, arbitrary meshes and two-way body/contact dynamics still need their own geometry, work, impulse and rollback contracts. A stationary viscous acceptance case does not complete those broader physics topics.

Affine rotation and shear

Use a centred unit cube as a measurement region, not a no-slip container. Prescribe $U=(-s\ y, s\ x, 0)$, constant viscosity μ , and pressure zero. This is rigid rotation: the velocity gradient is nonzero but $D=(\text{grad } U + \text{grad } U^T)/2$ vanishes. Consequently $\tau=2\ \mu\ D$, traction and constitutive dissipation vanish. Teal arrows show actual sampled velocity at a fixed scale; the field is planar and independent of z . Camera motion does not advance the field.

Boundary work supplies shear

Select $U=(s\ y, 0, 0)$. Its two off-diagonal strain entries are $s/2$. The unit-cube dissipation is $\mu\ s^2\ W$, which is 1 W at unit rate and viscosity. Orange arrows show $\tau\ n$ at the six face centres, using a separate labelled traction scale. The integral of traction dot velocity over those faces is also 1 W.

Constant stress has zero divergence here. Dissipation is supplied by boundary work, not an evolving bulk-force calculation. All affine component Laplacians are zero. Comparing $\mu\ \text{grad } U:\text{grad } U$ with $2\ \mu\ D:D$ while omitting the different boundary terms would be misleading. These imposed fields do not satisfy stationary no-slip conditions on the cube faces.

Extension control

For $U=(s\ x, -s\ y, 0)$, dissipation and supplied boundary power are $4\ \mu\ s^2\ W$. Changing the sign of s reverses the velocity and traction but leaves power unchanged. Setting viscosity to zero removes traction and dissipation while leaving the prescribed velocity intact. Read the complete matrices and six-face quadrature in the [research dossier](#). Batty and Bridson's [section 3, equations \(3.1\)–\(3.5\)](#) provides the constitutive/traction context; this lesson does not implement their embedded-boundary solver or inherit its convergence results.

► Planar stress proof scope

From viscosity to a specified turbulence closure

A turbulence-model name alone is not a reproducible constitutive law. The [constitutive research record](#) pins the SST-2003 limiter and pressure convention, distinguishes its primary 1992 motivation from later formulas, and identifies the transport and wall requirements still absent from a supplied-field constitutive probe. No SST transport or validation result is supplied by this chapter.

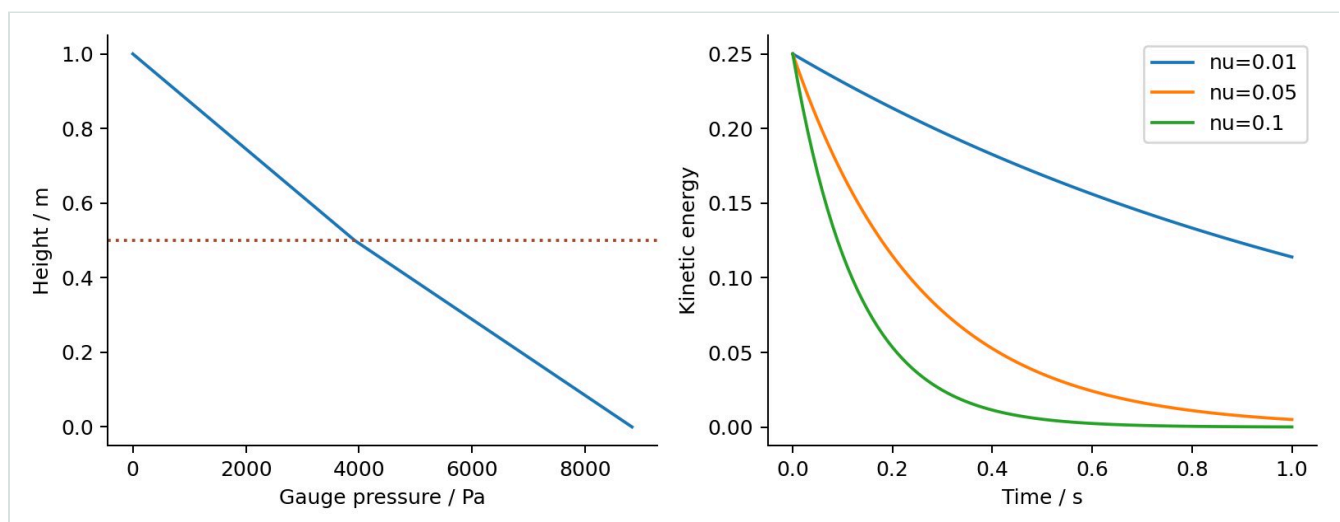
A specified SST-2003 limiter

The supplied-field probe uses density 1, $a_1 = 0.31$ and the SST-2003 denominator $\max(a_1\omega, SF_2)$. Select $k = \omega = 1$, $S = 4$ and $F_2 = 1$: eddy viscosity is 0.0775 Pa s, raw production is 1.24 W/m³, and the production cap gives 0.9 W/m³. The raw and limited production bars share one fixed scale, so their heights show clipping directly. The viscosity bar uses its own unit scale; compare the numerical values when comparing production and viscosity. They do not depict velocity or resolved eddies. The [variant and primary-reading record](#) pins coefficients, stress/pressure convention, source correction and wall data.

The zero-k constitutive limit

Set $k = 0$ with positive ω . Eddy viscosity and both production readouts are zero, and the denominator stays positive. This verifies the constitutive limit; the browser does not evaluate an undefined zero-viscosity ω source. A full transport implementation must define that limit, solve k/ω diffusion and cross-diffusion, compute wall distance/blending, enforce boundary and positivity policies, and report nonlinear residuals. The limiter inequalities prove nonnegative bounded algebraic production under stated premises; they cannot validate separation, turbulence spectra or a drag prediction.

► Practical study proof scope



Original reference illustration for 22 Viscosity, stress and dissipation

23 Wetting, adhesion and the contact line

Wetting describes a liquid's preference for solid contact in relation to surrounding phases. Adhesion names interfacial energetic attraction; slip describes tangential motion; viscosity describes bulk stress. These are not interchangeable controls. A liquid can have a small equilibrium contact angle and substantial slip, or a large angle with strong viscous drag. A wall that numerically pins every trace can look sticky without implementing an adhesion model.

Equilibrium surface energy

For a smooth homogeneous ideal solid, write interfacial energies γ_{sv} , γ_{sl} and liquid-vapor tension $\sigma > 0$, all in N/m. The equilibrium Young relation is

$$\gamma_{sv} - \gamma_{sl} = \sigma \cos \theta_e.$$

The reversible work of adhesion is $W_A = \gamma_{sv} + \sigma - \gamma_{sl}$, giving $W_A = \sigma(1 + \cos \theta_e)$. `young_adhesion_identity` proves this implication from a stated algebraic Young balance. `adhesion_bounds` assumes a cosine value in $[-1,1]$ and proves $0 \leq W_A \leq 2\sigma$. These are energetic identities, not a microscopic derivation, an existence theorem for a droplet, or a proof of dynamic contact angle.

Only the difference of solid interfacial energies is inferred by an angle. Their individual values remain unidentified. Nor does W_A specify a bulk attractive body force. Adding an arbitrary wall attraction while also imposing the same contact angle can count energetic effects twice. A chosen disjoining-pressure or precursor-film model requires its own potential, length scale and calibration.

Constant-volume cap geometry

In negligible gravity, an axisymmetric spherical cap with curvature radius R has height $H = R(1 - \cos \theta)$, contact radius $a = R \sin \theta$, and volume

$$V = \frac{\pi}{3} R^3 (2 - 3 \cos \theta + \cos^3 \theta).$$

For fixed V , solve this equation for R before changing angle. Drawing every cap with the same sphere radius changes liquid volume and confounds the demonstration. The original fixture holds V at 10^{-6} m^3 , samples angles from 30 to 150 degrees, and independently integrates circular cross sections to check volume. All cap meshes, pressure jumps and adhesion readouts use these same stored records.

Basilisk's original 3D sessile-drop test provides a primary implementation and analytic comparison [E6]. The present cap lab evaluates analytic equilibrium geometry; it does not simulate relaxation. Density and viscosity therefore do not affect its equilibrium shape. Gravity and finite Bond number invalidate this spherical-cap oracle.

Contact-angle reconstruction

A negative-inside level set gives an outward liquid normal. Let the solid normal point into the accessible fluid. At a flat horizontal solid, the cap's liquid normal obeys $n_l \cdot n_s = \cos \theta$. Curved or angled surfaces require a local wall frame and a contact-line tangent. A two-dimensional ghost slope copied directly into three dimensions can impose the wrong angle.

Afkhami and Bussmann's original height-function method accounts for the orientation of the contact-line projection in 3D [E7]. Its geometry and refinement results are method-specific. Basilisk's documented contact implementation includes restrictions at shallow angles and unresolved height support [E8]. Rheon's proposed admission rule reports unresolved normals rather than advertising all-angle accuracy.

Dynamics and hysteresis

A sharp moving contact line with strict no-slip has a classical stress-singularity difficulty. Cox's original analysis addresses a small-capillary-number regime with a microscopic slip or other regularization scale [E9]. A static angle alone is not a complete dynamic law. A proposed model must identify microscopic regularization, apparent-angle observation scale and contact-line mobility.

Advancing and receding angles can define a pinned interval with motion outside it. That is an additional hysteresis model, not an automatic consequence of Young's relation. Heterogeneous solids, roughness and contamination can change observed behavior. Start with a declared smooth equilibrium model, then qualify contact-line speed, angle and volume under refinement before adding hysteresis. No such dynamic contact-line solver is implemented or formally verified by this edition.

Moving contact lines

A static angle closes an equilibrium geometry problem. It does not determine how a contact line moves. Huh and Scriven's original creeping-flow analysis shows the divergent stress and dissipation associated with a sharp moving contact line under the classical adherence condition. A finite grid may conceal the singularity by supplying numerical slip; refining that grid changes the concealed length scale. [R04](#)

Choose and state one regularization:

- Resolve a physical or effective Navier slip length and a compatible dynamic contact-angle closure.
- Use a diffuse interface with specified interface thickness, mobility, wall free energy, and wall relaxation law.
- Use a precursor-film/disjoining-pressure model with its own microscopic length scale.
- Pair an explicitly labeled engineering angle-speed law with effective slip, diffusion or another declared microscopic cutoff. Advancing/receding limits and pinning are optional additional models; the angle-speed law alone does not remove the sharp no-slip singularity.

For a low-capillary-number hydrodynamic approximation, the familiar small-angle form is

$$\theta_{app}^3 - \theta_{micro}^3 \simeq 9 Ca \log(L/\ell_{micro}), \quad Ca = \mu U_{cl}/\gamma.$$

Here U_{cl} is signed, the angle is in radians, and L is the stated observation scale. This expression is not a universal all-angle, high-speed law. The general Cox construction concerns small Ca , scale separation, and a two-fluid viscosity ratio; at finite gas viscosity the full angle function replaces the cubic shorthand. Use the expression as a labeled asymptotic comparison only within those assumptions. [R05](#)

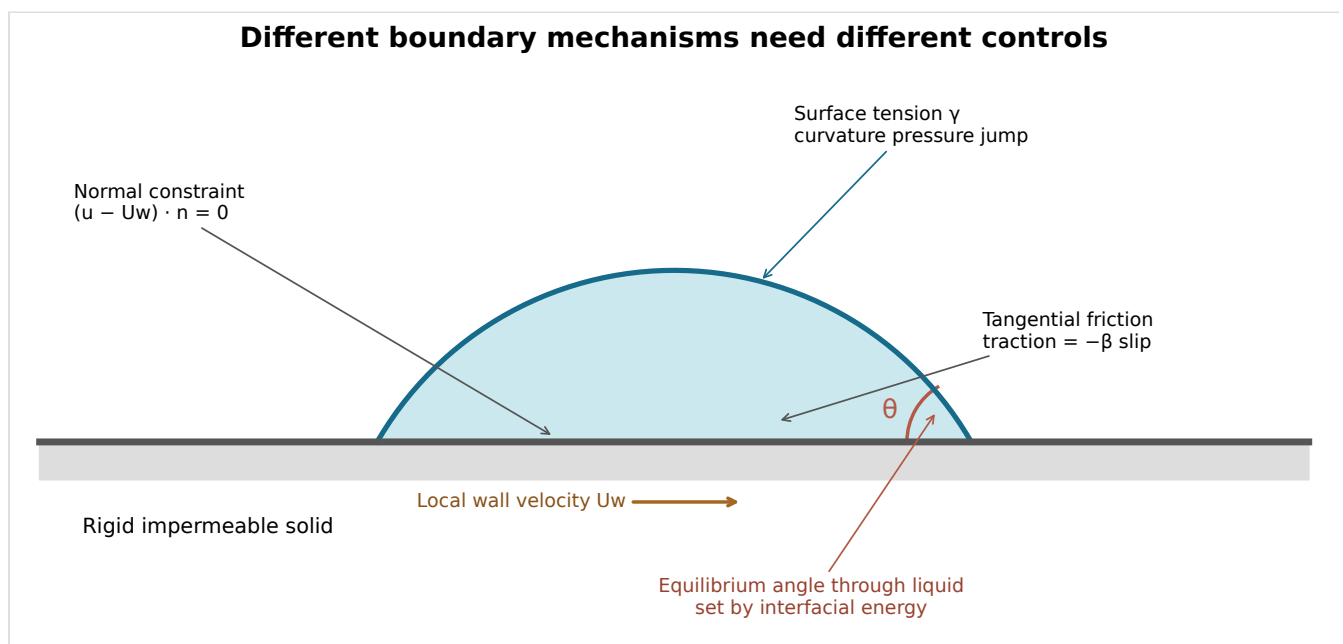
Qian, Wang, and Sheng combine Cahn–Hilliard interfacial free energy with a generalized Navier boundary condition containing uncompensated Young stress. They compare the continuum model with molecular-dynamics profiles, including Couette and Poiseuille configurations. That is stronger evidence than an attractive drop image, but it does not license transplanting their fitted molecular parameters to an arbitrary macroscopic liquid or treating diffuse-interface mobility as a harmless numerical knob. [R06](#)

For contact-angle hysteresis, an engineering pinned state may allow $\theta_R \leq \theta_{app} \leq \theta_A$ with $U_{cl} = 0$. An advancing or receding branch applies outside that interval. Store θ_R , θ_A , the observation scale, and the dynamic law separately. Static friction, capillary pinning, and high bulk viscosity can all delay motion while representing different physics.

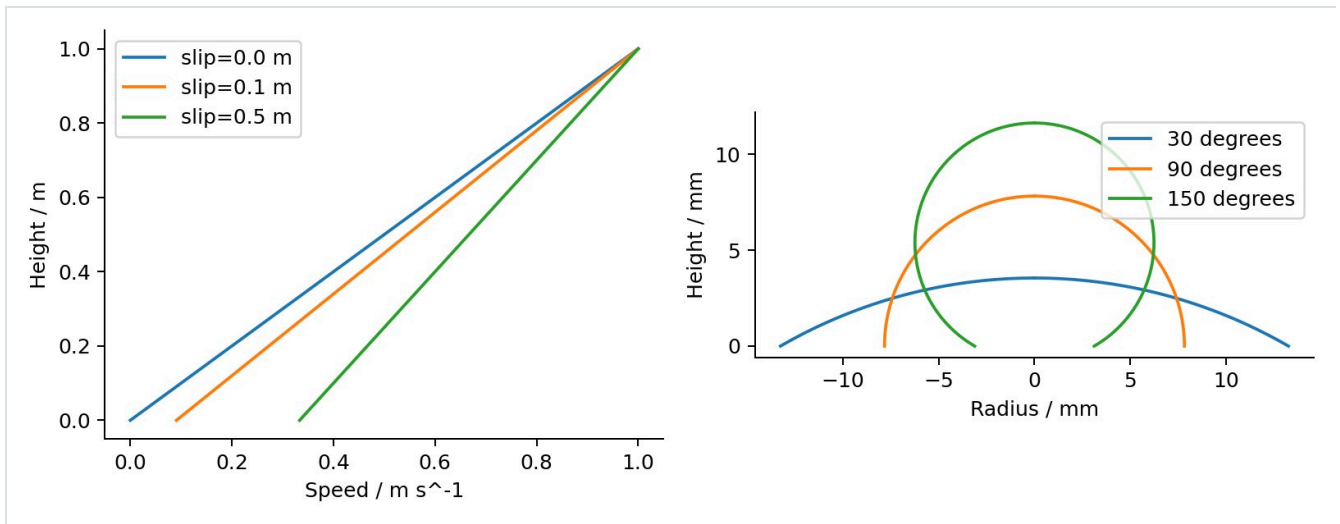
Partial wetting has an admissible range

Young balance requires $\sigma > 0$ and $|(\gamma_{sv} - \gamma_{sl})/\sigma| \leq 1$. Outside that interval, clamping the inferred cosine hides a changed model. The spreading coefficient $S = \gamma_{sv} - \gamma_{sl} - \sigma$ distinguishes the ideal complete-wetting tendency when $S \geq 0$; a precursor film or disjoining pressure then needs extra length scales. The fixed-volume cap reference deliberately stays within partial wetting and does not provide a film model.

With fixed total solid area, surface energy can be written $E_s = \sigma(A_{lv} - \cos\theta_e A_{sl})$ up to a constant. Changing μ changes the relaxation path, while changing θ_e changes this ideal energy landscape. A nearly stationary highly viscous drop may be far from equilibrium. An empirical SPH cohesion/adhesion force is a possible visual model [R11](#), but its coefficient is not automatically the gradient of this area energy or a calibrated interfacial tension.



Boundary geometry, tangential dissipation and interfacial energy supply different conditions; arrows are schematic.



Original reference illustration for 23 Wetting, adhesion and the contact line

24 Slip and wall friction

Impermeability sets relative normal velocity. A tangential boundary law must be selected separately. No-slip sets tangential fluid velocity equal to the wall. Free-slip sets tangential traction to zero. Navier slip relates traction and relative tangential speed. None of these selects a wetting angle.

Dissipative Navier law

With a normal/sign convention chosen so traction on the fluid opposes slip, let $t_t = -\beta(u - u_s)_t$, where $\beta \geq 0$ has units Pa s/m. Relative power is

$$t_t \cdot (u - u_s)_t = -\beta|(u - u_s)_t|^2 \leq 0.$$

`slip_power_nonpositive` proves the scalar component version, and component sums extend it to a tangent plane. The inequality concerns relative dissipation. A moving wall can still increase fluid kinetic energy through external work. It does not prove a spatially assembled slip discretization has this sign.

For positive viscosity, slip length $\ell_s = \mu/\beta$. The limiting cases must be represented explicitly: $\ell_s = 0$ is no-slip, while zero β is free-slip and corresponds to unbounded slip length. Avoid dividing by zero or approximating both limits with the same finite parameter. When $\mu = 0$, this relation cannot define the same viscous boundary model.

Couette reference

A lower stationary wall at $y=0$ and upper wall moving at U at $y=H$ give steady affine speed

$$u(y) = U \frac{y + \ell_s}{H + \ell_s}$$

when only the lower wall has Navier slip and the upper wall has no-slip. Then $u(0) = \ell_s u'(0)$, $u(H) = U$, and shear stress is $\mu U / (H + \ell_s)$. Increasing slip raises lower-wall speed and reduces shear at fixed U . It does not change equilibrium contact angle.

The reference generator verifies both boundary equations and exports profiles for slip lengths 0, 0.1 and 0.5 m in a unit-height slab. The browser uses actual profile samples for its arrows and shows the shear and dissipated power. These large illustrative slip lengths clarify the mathematics; they are not fitted values for water on a named material.

Friction is a model choice

Dry Coulomb friction bounds a tangential force using normal contact force. Applying it to a liquid-solid wall is a separate constitutive assumption, not the default Newtonian no-slip boundary law. A penalty damping term likewise has its own units and finite-step dissipation. Label it as a numerical/constitutive choice if used. A single “stickiness” parameter cannot consistently identify all of these mechanisms.

Cox’s primary contact-line analysis explains why a microscopic mobility mechanism matters [E9]. A macroscopic slip length selected for a resolved Couette profile need not serve as the microscopic regularization of a contact-line model. Using it for both requires an explicit scale argument and validation.

Proposed state and tests

`TangentialLaw` distinguishes no-slip, free-slip and finite Navier coefficient. `WettingLaw` owns angle/energies separately. Boundary elimination must include known moving-wall terms; diagnostics retain relative dissipation and external wall power independently.

The broader test plan includes both signs of U , rotated wall frames, uniform wall/fluid translation, curved-wall interpolation and zero/coefficient limits. A stationary relative state must dissipate zero. Refinement should approach the analytic Couette profile and traction. The finite proof does not establish those geometric limits; the native operations below qualify only the fixed flat-slab slice.

Both walls slipping: a distinct oracle

The native [flat-slab wall-friction update](#) now advances actual Rust velocities with finite nonnegative wall coefficients, positive density and viscosity, separate relative dissipation and actuator work. Its reproducible `column_wall_shear` laboratory compares the existing finite endpoint basis with the symmetric two-wall oracle below. This bounded model does not add wetting, contact lines or arbitrary collision-mesh fluid boundaries.

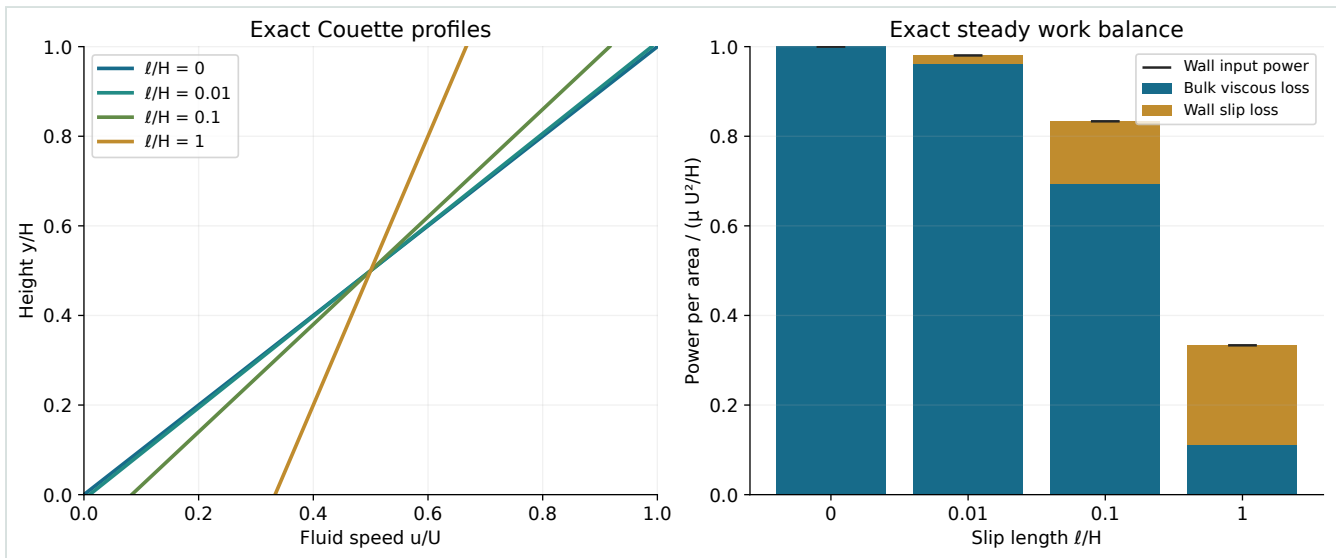
The browser lab uses lower-wall slip only. With equal finite slip lengths ℓ at both stationary-lower/moving-upper walls, the separate exact profile is

$$u(y) = U \frac{y + \ell}{H + 2\ell}, \quad \tau = \mu \frac{U}{H + 2\ell}.$$

The two relative wall speeds have magnitude $U\ell/(H+2\ell)$, and actuator power partitions as

$$\tau U = \mu \left(\frac{U}{H + 2\ell} \right)^2 H + 2 \frac{\mu}{\ell} \left(\frac{U\ell}{H + 2\ell} \right)^2.$$

Handle $\ell=0$ as a no-slip constraint instead of dividing by zero. This formula is an independent analytical fixture; it must not replace the one-wall denominator in the existing browser data. The original comparison figure below uses symmetric two-wall slip and displays bulk/wall power separately.



Exact fully developed Couette profiles with symmetric Navier slip; wall input partitions into bulk and wall dissipation.

Exact no-slip in the retained flat-slab basis

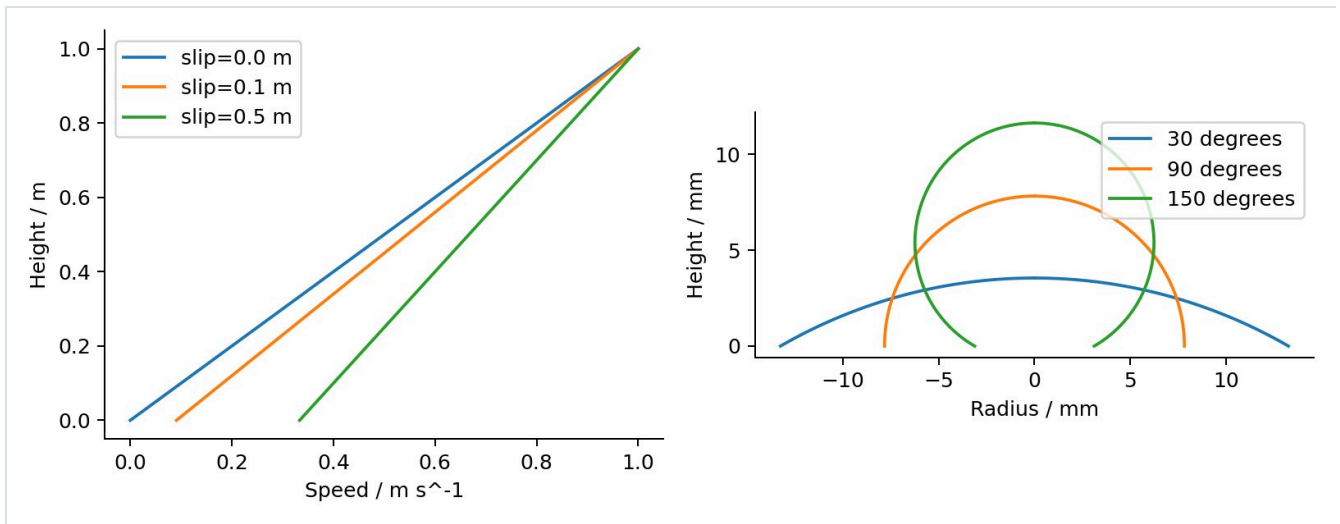
The [connected native progression](#) starts with both walls at finite Navier friction, adds compatible exact no-slip, then adds stationary-wall body-force-driven Poiseuille startup. These are three distinct reference problems. The original one-wall 3D analytic lab remains an equilibrium record, while native case/time controls select saved Rust profiles; none is a live general fluid solver.

The [compatible endpoint no-slip operation](#) now distinguishes an exact prescribed tangential constraint from finite Navier friction. It requires the incoming trace to match the fixed wall speed. Rather than introducing a large beta, it eliminates the endpoint increment and reports the reaction impulse $J = -dt F_{\text{unconstrained}}$. Delivered wall work is $w \cdot J$ and can have either sign. Relative reaction work $J \cdot (u - w)$ vanishes, while bulk viscosity continues to dissipate energy. Wetting and adhesion remain separate.

The native `column_no_slip` lab provides case selection and five actual stored snapshots for two-wall no-slip at 8/16 layers and lower-Navier/upper-no-slip at 16 layers. Its mixed continuum reference is the original one-wall oracle above. It includes compatible initial energy, reaction impulses, signed actuator work, bulk and finite-slip dissipation, and explicit/f32 rounding corrections.

The existing basis extends wet-center endpoint values constantly to each wall. Thus the wall trace is constrained exactly, but comparison at wet centers still has spatial endpoint error. The discrete equilibrium for two no-slip walls is $u_j = U j / (L - 1)$; for lower slip it is $u_j = U(jh + \text{ell}) / ((L - 1)h + \text{ell})$. Distance to these equilibria also includes remaining physical relaxation and numerical transient/rounding effects. A decreasing endpoint deviation under the bounded refinement is not temporal order.

Four new Lean statements prove compatible trace preservation, zero relative reaction work, momentum balance and the constrained finite work identity under explicit step/assembled-power assumptions. They do not certify stencil assembly, Rust floating point or general wall geometry. The native operation refuses prescribed speeds incompatible with the incoming trace and two no-slip walls sharing one wet node, where separate reactions would be nonunique. It retains no boundary-speed history and supplies no time-varying wall-speed evolution model.



Original reference illustration for 24 Slip and wall friction

25 Surface tension and integrated qualification

Surface tension supplies an interface stress. It is not viscosity or a wall adhesion force. With constant σ , outward liquid normal and curvature sum $\kappa = \nabla \cdot n$, a spherical interface has $\kappa = 2/R$. Neglecting normal viscous stresses gives $p_l - p_a = \sigma\kappa$. With viscous liquid stress and negligible air viscous stress, the normal jump instead includes $n^T \tau_l n$. Chapter 9’s sign convention remains authoritative.

Pressure-force balance

A discrete surface force and pressure gradient should use compatible locations, masks and differences. Otherwise even an accurately solved stationary drop can develop parasitic velocity. Basilisk’s original surface-tension implementation expresses an interfacial potential using curvature and applies an explicit capillary stability bound [E10]. Its code is a reference for examining balanced construction, not a borrowed Rheon backend.

The capillary lab holds the same cap volume and angle while varying tension through 0.03, 0.072 and 0.12 N/m. Shape remains the analytic zero-gravity equilibrium cap; pressure jump changes linearly with tension. For $V = 10^{-6}$ m³ and angle 90 degrees, R is approximately 0.007815926 m and the jump at 0.072 N/m is approximately 18.42392 Pa. The reference records use full precision, with these prose numbers rounded.

Time scales and resolution

The inertial capillary scale is $t_\sigma = \sqrt{\rho h^3 / \sigma}$. The viscous diffusion scale is $t_\nu = h^2 / \nu$. The gravity-capillary length is $\ell_c = \sqrt{\sigma / (\Delta \rho g)}$. These are dimensional scales, not universal accepted step sizes. Actual stability constants depend on discretization and time integration; implicit tension can change stability while retaining geometric error.

Dimensionless groups separate regimes: $Re = \rho UL / \mu$, $Ca = \mu U / \sigma$, $We = \rho U^2 L / \sigma$, and $Bo = \Delta \rho g L^2 / \sigma$. A spherical-cap reference assumes negligible Bo ; a shear decay reference excludes an interface. Combining their pictures does not validate a coupled solver.

Progressively richer teaching examples

The Pages edition presents six linked 3D laboratories: finite MAC projection; earliest triangle-mesh collision with prescribed motion; layered hydrostatic density; implicit viscous mode; Navier-slip Couette flow; and constant-volume wetting/capillarity. Each points to its chapter, original reference implementation, stored data and applicable proof. Parameters select or interpolate declared reference states, rather than pretending to run a general liquid simulator.

The original smoke render remains a separate accepted Rust demonstration with its original source identity. This edition does not relabel that smoke as a liquid calculation, or claim that the educational analytic caps establish production realism. Readers can compare the evidence levels in Appendix F.

Production integration sequence

The new [wall/force progression](#) qualifies local finite traction, compatible trace and uniform forcing; it does not complete this integration sequence or the proposed F1–F3 continuum gates. The [source-backed requirements map](#) recommends a static closed-obstacle fluid-geometry owner as the next feature, connecting triangle queries to shared volumes/open areas/connectivity before general pressure, viscous traction or wetting coupling.

After research review, implement one-way collision geometry and gravity first. Add free-surface classification/transport with volume and spatial metrics, then constant and variable density using a consistent pressure correction. Add tensor viscosity and wall traction, then a declared contact-angle reconstruction and balanced capillarity. Dynamic contact lines and two-way solids come after their assumptions are selected. This is a research implementation sequence, not a merge gate for the existing bounded smoke milestone.

An accepted liquid step must publish interface, material fields, velocity, time and geometry version together. Failure in geometry, transport, nonlinear stress or pressure keeps the last coherent state. Capillary, viscous and Courant restrictions must be reported with the actual accepted interval; no hidden parameter changes should imitate a successful solve.

Evidence that earns a capability claim

Use hydrostatic rest, variable-density layers, viscous decay/Couette flow, translating volume, a stationary droplet and a sessile cap before combined pouring scenes. Report pressure jump, parasitic speed, volume drift, contact angle, wall flux/work and boundary versions. Refine h and dt independently. Then add moving meshes, pinning/depinning and separation, with independent geometry references. Tests must name a physical regime and error budget before advertising calibrated material behavior.

The new Lean contracts remove ambiguity in finite identities and sign conditions. The reference programs qualify local mechanisms. Neither proves arbitrary mesh assembly, full three-dimensional liquid transport, contact-line convergence, floating-point production behavior or performance. Those are explicit research gaps, not reasons to reject the existing smoke demonstration.

A future liquid qualification suite

The following fixtures are planned for a quantitative liquid implementation. Proposed error targets are starting engineering choices, not universal tolerances or merge gates for the accepted smoke milestone. The local references qualify only the finite/analytical mechanisms described earlier.

Analytical fixtures

F1 Planar Couette flow

Use a periodic streamwise channel with $0 \leq y \leq H$, lower wall speed 0 and upper wall speed U . For stationary fully developed incompressible Newtonian flow with no slip,

$$u_x(y) = Uy/H, \quad \tau = \mu U/H, \quad P_{wall}/A = \mu U^2/H.$$

Use synthetic SI values $H = 0.01$ m, $U = 0.1$ m/s, $\mu = 0.1$ Pa·s, $\rho = 1000$ kg/m³. Then $\nu = 10^{-4}$ m²/s, $Re = 10$ using gap width, $t_v = 1$ s, $\tau = 1$ Pa, and $P/A = 0.1$ W/m². Initialize the analytic steady profile for a discrete equilibrium test; start from rest separately to test transient relaxation. Do not claim steady state merely after an arbitrary number of frames.

Proposed gates: relative velocity L2 error below 1%, wall shear error below 2%, and steady wall-work minus dissipation residual below 1% of wall power on the finest test grid. Also verify that changing ρ while holding μ fixed changes transient timescale but not the steady profile.

F2 Symmetric Navier slip in Couette flow

For equal slip lengths ℓ at both walls,

$$u_x(y) = U \frac{y + \ell}{H + 2\ell}, \quad \tau = \mu \frac{U}{H + 2\ell}.$$

Sweep $\ell/H = 0, 0.01, 0.1, 1$. Use a constraint for the zero-slip limit, not division by zero in $\beta = \mu/\ell$. Check both fluid slip velocities and both traction signs. For finite ℓ , test

$$\tau U = \mu \left(\frac{U}{H + 2\ell} \right)^2 H + 2 \frac{\mu}{\ell} \left(\frac{U\ell}{H + 2\ell} \right)^2.$$

This partitions actuator power into bulk and wall dissipation. A solver matching the profile but omitting wall dissipation is still missing an energy term.

F3 Pressure driven Poiseuille flow

For a planar channel, no slip, and $G = -dp/dx > 0$,

$$u_x(y) = \frac{G}{2\mu} y(H - y), \quad Q' = \frac{GH^3}{12\mu}.$$

Q' has units m²/s and is flow rate per out-of-plane width. With equal Navier slip length ℓ , add $G\ell H/(2\mu)$ to the velocity and multiply Q' by $1 + 6\ell/H$. A uniform body acceleration reproducing the pressure gradient is $a = G/\rho$, not G itself. Check profile, flow rate, wall shear, and input pressure power per unit out-of-plane width $GQ'L_x$ (W/m). Its time integral is work per unit width.

Use the Couette material values and $G = 120$ Pa/m for a no-slip mean speed 0.01 m/s. Proposed gates: below 1% relative flow-rate/profile error on the finest grid and decreasing errors under refinement. Repeat with a rotated or curved embedded channel only when an appropriate analytic or independently converged reference is supplied.

F4 Viscous mode decay and rigid motion

For a periodic mode $u = (U \sin(ky), 0)$, the convective term vanishes and

$$u_x(t) = U \sin(ky) e^{-\nu k^2 t}, \quad E(t)/E(0) = e^{-2\nu k^2 t}.$$

This distinguishes μ from ν and reveals excess numerical damping. With backward Euler and constant operator eigenvalue λ , the discrete amplitude ratio after N steps is $U_N/U_0 = (1 + \nu\lambda\Delta t)^{-N}$, where $\lambda \geq 0$ is the eigenvalue of the positive discrete negative-Laplacian operator before multiplication by ν , with units m^{-2} . Compare against the discrete formula for the integrator test and the exponential for the PDE convergence test.

Independently apply only the viscous step to uniform translation and a rigid rotation sampled on a free domain. Check zero strain, zero viscous force, and unchanged velocity. Rotation tests boundary traction and symmetric-gradient consistency; decay of a nonrigid sine mode tests actual dissipation.

F5 Static pressure jump and parasitic currents

Turn gravity off, place a circular drop away from walls, set zero velocity, and use constant γ . In a strictly two-dimensional planar cross-section with translational invariance, $\Delta p = \gamma/R$. For a three-dimensional sphere, $\Delta p = 2\gamma/R$. Do not use the sphere formula for a planar 2D disk.

Start with $R = 0.001$ m and synthetic $\gamma = 0.072$ N/m, giving 72 Pa in planar 2D and 144 Pa in 3D. These are declared synthetic parameters, not a measured fluid specification. Use $\rho l = 1000$ kg/m³ and $\tau\gamma = \sqrt{(\rho l R^3/\gamma)}$; select and report gas density/viscosity or the one-phase approximation. Sweep radii of 8, 16, 32, and 64 grid cells, drop offsets, and at least one density ratio beyond unity.

Measure bulk-phase mean pressure away from the interface band, maximum and RMS velocity, curvature bias/variance, volume drift, and interface error. Normalize parasitic velocity by $U_\gamma = \sqrt{(\gamma/(\rho l R))}$; additionally report $\mu U_{\max}/\gamma$ when $\mu > 0$. A proposed quantitative gate is pressure-jump error below 2%, relative volume drift below 0.1% over ten $\tau\gamma$, and $U_{\max}/U_\gamma$ below 10^{-3} on the finest fixture. These targets may need stricter tolerances for long capillary relaxation.

Basilisk's original spurious-current test supplies a reproducible scientific precedent with explicit diameter, Laplace number, viscous damping time, and resolution sweep. It is a reference design to reproduce, not a result already obtained here. [R24](#)

F6 Sessile drops at several contact angles

Set gravity to zero, keep fixed liquid volume, and use a smooth flat rigid substrate. For a 2D circular segment with angle θ through the liquid, using radians in the formula,

$$A = R^2(\theta - \sin \theta \cos \theta), \quad a = R \sin \theta, \quad h = R(1 - \cos \theta).$$

For a 3D spherical cap,

$$V = \frac{\pi R^3}{3}(2 - 3 \cos \theta + \cos^3 \theta), \quad a = R \sin \theta, \quad h = R(1 - \cos \theta).$$

At obtuse angles the spherical cap can overhang its footprint, so a height-field-only visualization cannot represent every shape. Compare $\theta = 30^\circ, 60^\circ, 90^\circ, 120^\circ$, and 150° first. Extreme angles are a separate capability claim and require appropriate curvature stencils and film resolution. Report angle measured at a stated distance from the wall, fitted radius, footprint, height, and remaining kinetic energy. Proposed gates are angle error below 2° and dimensions within 2% on the finest baseline grid, with converging errors and stable volume.

The original Basilisk 2D and 3D sessile-drop tests provide implementation-level references. Preserve the distinction between a circular segment and a spherical cap; the published 2D page uses an area-like volume convention. [R22](#), [R23](#)

F7 Dynamic capillary oscillation

Perturb a drop slightly and measure frequency and damping. Use the correct dimensionality and phase model. For an inviscid 3D spherical drop in a dynamically negligible exterior, the degree- l mode has $\omega l^2 = l(l-1)(l+2)\gamma/(\rho R^3)$, $l \geq 2$. With a dynamically important exterior or significant viscosity, use the corresponding two-fluid/damped theory rather than that formula. [R27](#)

Keep perturbation amplitude small and verify amplitude independence over a decreasing-amplitude sequence. Report oscillation frequency separately from amplitude decay. A method can produce the correct equilibrium radius while being badly overdamped. The original Basilisk oscillation fixture explicitly analyzes frequency error and equivalent numerical viscosity; use it as an additional reproducible baseline for its own dimensional setup. [R25](#)

F8 Moving contact line and capillary penetration

For a selected slip/diffuse-interface closure, compare spreading radius and apparent angle versus time while refining the grid at fixed physical regularization length. If ℓ_s or ϵ is tied to Δx , report that as a changing-model sequence. Do not label it ordinary convergence to a fixed physical problem.

For a long horizontal circular capillary of radius r , with negligible inertia/gravity/gas resistance, quasi-steady Poiseuille flow, no slip outside the regularized contact region, and constant θ , the Lucas–Washburn reference is

$$x^2(t) - x^2(0) = \frac{r\gamma \cos \theta}{2\mu} t.$$

Use $\theta < 90^\circ$ and a finite initial filled length to avoid the unphysical infinite startup speed of the inertia-free approximation. Fit only a time window where its hypotheses hold; entrance losses, inertia, gravity, dynamic angle, and trapped gas can all invalidate it. The original Washburn publication supplies the square-root penetration law and its assumptions. [R26](#)

F9 Moving walls and two way solids

This remains a qualification target. The unmerged stationary aligned-box Stokes checkpoint described in [Chapter 22](#) does not implement moving boundaries, coupled bodies or contact dynamics.

Translate a wall through initially stationary fluid and rotate a paddle with prescribed angular speed. Plot accumulated actuator work against changes in kinetic/potential/surface energy and dissipation. Then move both fluid and obstacle at the same constant velocity; a Galilean-consistent relative-velocity treatment should not generate drag from their common translation. Repeat as the obstacle crosses grid cells.

For a freely moving solid in a closed fluid system without external forces, check combined linear momentum. Check total angular momentum only with compatible rotational geometry and transfer. During a fixed-geometry implicit projection, T4 predicts a particular nonnegative numerical energy loss, not exact kinetic-energy conservation.

Geometry stress fixtures

Fixture	Sweep	Future capability observation
Thin closed plate	Thickness/ Δx from 4 to 0.1; sub-cell offsets and orientations	No unreported disappearance or leakage; displaced volume error disclosed
Zero-thickness two-sided shell	Crossings of every interpolation/advection stencil	Barrier respected or explicit unsupported status
Narrow slot	Width/ Δx = 0.25, 0.5, 1, 2, 4, 8	Conductance convergence when resolved; no invented sub-grid accuracy
Sliver cut cell	Fluid fraction approaching zero	Stability, flux conservation, redistribution locality, no negative mass
Moving cut surface	Swept cell fractions and common translation	Uniform-field preservation and mass budget under covering/uncovering
Rotating paddle	Angular speed and interaction-point offsets	Correct local wall speed, torque, and work signs
Dirty mesh	Open seam, self-intersection, reversed patch, degenerate triangles	Deterministic rejection or documented repair; never silently plausible output
Stale deformation map	Bending or scaling obstacle after preprocessing	Update/rebuild or explicit unsupported status

Collision leakage should be measured as fluid mass crossing a labeled impermeable barrier, not merely particles visibly inside a solid at the end of a frame. For a closed static tank, declare a finite reference observation time $T = L_{\text{ref}}/U_{\text{ref}}$ with $U_{\text{ref}} > 0$, or select t_y or t_v when there is no reference flow. Proposed regression gates are relative mass drift below 0.1% over ten such T and zero topological barrier crossings in exact-geometry tests; engineering fixtures should additionally report the numerical tolerance used to classify a crossing.

26 Compressible state, thermodynamics and waves

Compressibility introduces an evolving density and a thermodynamic energy equation. Pressure now follows a constitutive law and propagates through acoustic waves. The incompressible projection in Chapter 4 does not supply these dynamics. These three chapters specify a research path for a compressible branch; they do not establish a production compressible capability in Rheon.

A state that conserves the right quantities

For a single continuum in three dimensions, use the conservative state

$$U = (\rho, \mathbf{m}, \mathcal{E})^T, \quad \mathbf{m} = \rho \mathbf{u}, \quad \mathcal{E} = \rho e + \frac{1}{2} \rho |\mathbf{u}|^2.$$

Here e is specific internal energy in J/kg; $\mathcal{E}$ is total energy **per volume**, in J/m³. Density has units kg/m³ and momentum density kg/(m² s). A reader comparing papers must check their notation: many authors write ρE for what this chapter calls $\mathcal{E}$. Storing a pressure or temperature tracer instead of total energy changes the conservation contract.

For an outward unit normal $\mathbf{n}$, let $u_n = \mathbf{u} \cdot \mathbf{n}$. The Euler normal flux is

$$F_n(U) = (\rho u_n, \mathbf{m} u_n + p \mathbf{n}, (\mathcal{E} + p) u_n)^T, \quad \partial_t U + \nabla \cdot F(U) = 0.$$

Pressure transports momentum and performs work; the energy flux therefore contains $p u_n$. In a finite volume, a face transports one mass, one vector momentum and one energy flux. Computing them from different advecting states can conserve each array globally while breaking the intended thermodynamic coupling. NASA's governing-equation overview identifies the coupled continuity, momentum and energy system; the ALE paper by Benoit and Nadarajah makes the conservative state and ideal-gas pressure explicit. [NASA equations](#), [ALE formulation](#), §2.1

Close the equations before choosing a solver

A calorically perfect ideal gas assumes constant c_v , c_p and $\gamma = c_p/c_v > 1$, with $R = c_p - c_v$. Its closure is

$$p = (\gamma - 1) \left(\mathcal{E} - \frac{|\mathbf{m}|^2}{2\rho} \right), \quad e = c_v T, \quad p = \rho R T, \quad c^2 = \gamma p / \rho.$$

The admissible set has $\rho > 0$ and positive internal-energy density $\mathcal{E} - |\mathbf{m}|^2/(2\rho) > 0$. Absolute pressure enters the EOS. A plotting offset called gauge pressure must not replace it. Values such as $\gamma = 1.4$ define a teaching gas; constant heat capacities exclude temperature-dependent excitation, dissociation, ionization and chemical reaction. A thermally perfect gas can retain $p = \rho R T$ while requiring temperature-dependent internal energy. NACA Report 1135 treats both perfect and imperfect gases, so its perfect-gas shock tables must be selected deliberately. [NACA Report 1135](#)

For a general single-material EOS $p = p(\rho, e)$, the isentropic acoustic derivative follows from $de = p d\rho/\rho^2$:

$$c^2 = \left. \frac{\partial p}{\partial \rho} \right|_e + \left. \frac{\partial p}{\partial e} \right|_\rho \frac{p}{\rho^2}.$$

The implementation must provide this derivative, a valid thermodynamic domain and consistent temperature/energy conversions. A negative c^2 , an EOS inversion outside its domain or an undefined heat capacity is a material-model failure, not a sound-speed value to clamp silently. Wang and Li derive EOS-dependent thermodynamics and a local stiffened-gas approximation; their approximation is a numerical strategy for a declared EOS, not a universal liquid calibration. [Wang and Li, §§1–3](#)

One simple stiffened-gas convention is

$$p = (\gamma_s - 1)\rho e - \gamma_s p_\infty, \quad c^2 = \gamma_s(p + p_\infty)/\rho.$$

It reduces to an ideal gas when $p_\infty = 0$. A compatible constant- c_v caloric relation is $e = c_v T + p_\infty/\rho$; other conventions include an energy offset and must be documented with their inversion. Admissibility requires $\rho > 0$, $p + p_\infty > 0$ and the chosen caloric domain. Reusing the ideal-gas condition $p > 0$ without reviewing this EOS changes its domain. Parameters must be fitted over the intended pressure and temperature range. A single-phase fit does not introduce cavitation, latent heat or a liquid-vapour interface model. The shock-compression experiments of Prasanna and colleagues illustrate why a liquid EOS fit needs a named material, data range and uncertainty. Their perfluorohexane parameters are not water parameters. [Liquid EOS experiment, §§II–IV](#)

A Tait-type barotropic law instead prescribes pressure from density, for example

$$p(\rho) = p_0 + B[(\rho/\rho_0)^n - 1], \quad c^2 = \frac{nB}{\rho_0}(\rho/\rho_0)^{n-1}, \quad B > 0, \quad n > 0.$$

These equations define this chapter's convention; some sources absorb p_0 into the pressure offset. It can represent a specified compressibility range. In weakly compressible numerical models, artificially reducing c to enlarge the time step changes acoustic travel times and compressibility. This is a modelling choice, not calibrated acoustics. A barotropic pressure law alone supplies no temperature dependence or shock heating. For smooth isentropic motion one may construct an energy with $de/d\rho = p(\rho)/\rho^2$; that restricted mechanical energy does not close the general thermal Euler/Navier–Stokes system through shocks. Select a complete caloric EOS before claiming energy-resolved liquid shocks.

Acoustic waves, contacts and shocks

In direction $\mathbf{n}$, the three-dimensional Euler eigenvalues are

$$u_n - c, \quad u_n, \quad u_n, \quad u_n, \quad u_n + c.$$

The middle fields carry entropy and the two tangential velocity components. Acoustic fields travel relative to the fluid at $\pm c$. A contact can have different density and temperature on either side while pressure and normal velocity agree. A pressure pulse and an advected density contact therefore exercise different mechanisms. The hyperbolic wave structure is set by the EOS; the ideal-gas formulas cannot be transferred unchanged to arbitrary real materials. [Wang and Li, §2](#)

Linearize about uniform rest $(\rho_0, p_0, \mathbf{u} = 0)$ with $c_0^2 = \gamma p_0/\rho_0$. Along x , the acoustic subsystem is

$$\partial_t p' + \rho_0 c_0^2 \partial_x u' = 0, \quad \partial_t u' + \rho_0^{-1} \partial_x p' = 0.$$

Thus $p' = f(x - c_0 t)$, $u' = p' / (\rho_0 c_0)$, $\rho' = p' / c_0^2$ is a right-travelling linear wave. Its impedance is $Z = \rho_0 c_0$. The nonlinear Euler system only approaches this reference as amplitude tends to zero; Chapter 28 specifies how to separate the linearization error from numerical error.

Across a planar discontinuity moving at normal speed s , integrate conservation over a shrinking space-time control volume:

$$s[U] = [F_n(U)], \quad [a] = a_R - a_L.$$

These Rankine–Hugoniot conditions constrain shock speed and both states. Conservation is necessary but does not select the physical branch. For an ideal gas use entropy relative to fixed reference values,

$$s - s_{ref} = c_v \log [(p/p_{ref})(\rho_{ref}/\rho)^\gamma].$$

An admissible adiabatic shock increases specific entropy along the material crossing direction. Smooth inviscid motion transports it. The weak inequality is $\partial_t(\rho s) + \nabla \cdot (\rho s \mathbf{u}) \geq 0$. Mathematical entropy commonly uses the opposite sign, such as $\eta = -\rho s$, so its inequality is nonpositive. Always state the sign convention before plotting an entropy residual. NASA’s normal-shock reference explains the increase of entropy and loss of stagnation pressure across a calorically perfect shock. [NASA normal shocks](#)

For a stationary normal shock with upstream Mach $M_1 > 1$, ideal-gas conservation gives

$$\frac{\rho_2}{\rho_1} = \frac{(\gamma + 1)M_1^2}{(\gamma - 1)M_1^2 + 2}, \quad \frac{p_2}{p_1} = 1 + \frac{2\gamma}{\gamma + 1}(M_1^2 - 1).$$

With $\gamma = 1.4$, $M_1 = 2$, these ratios are $8/3$ and 4.5 . The temperature ratio is $27/16$, and $M_2^2 = 1/3$. These derived numbers are a local jump reference, not an executed Rheon shock calculation. Resolving a numerical shock over several cells does not reproduce its physical molecular thickness.

Viscous stress and heat are part of energy accounting

For a compressible Newtonian fluid let $D = (\nabla \mathbf{u} + \nabla \mathbf{u}^T)/2$, dynamic shear viscosity $\mu \geq 0$, bulk viscosity $\zeta \geq 0$, and conductivity $k \geq 0$. Use

$$\boldsymbol{\tau} = 2\mu \left(D - \frac{1}{3}(\nabla \cdot \mathbf{u})I \right) + \zeta(\nabla \cdot \mathbf{u})I, \quad \mathbf{q} = -k\nabla T.$$

Stokes’ hypothesis sets $\zeta = 0$; it is an assumption about bulk dissipation, not the condition $\nabla \cdot \mathbf{u} = 0$. With body acceleration $\mathbf{b}$ and volumetric heat source Q , the equations become

$$\partial_t \rho + \nabla \cdot (\rho \mathbf{u}) = 0,$$

$$\partial_t \mathbf{m} + \nabla \cdot (\mathbf{m} \otimes \mathbf{u} + pI) = \nabla \cdot \boldsymbol{\tau} + \rho \mathbf{b},$$

$$\partial_t \mathcal{E} + \nabla \cdot [(\mathcal{E} + p)\mathbf{u}] = \nabla \cdot (\boldsymbol{\tau} \mathbf{u} - \mathbf{q}) + \rho \mathbf{b} \cdot \mathbf{u} + Q.$$

Viscous work converts kinetic energy into internal energy while total energy changes through boundary work and heat. Adding a velocity diffusion step without its compatible energy term misses this conversion. For $T > 0$, the smooth entropy balance has production $\tau : D/T + k|\nabla T|^2/T^2 \geq 0$, with heat entropy flux $\mathbf{q}/T$. Numerical shock dissipation, physical viscosity and turbulent closure remain three distinct contributors. The original curvilinear Navier–Stokes study of Nolasco and colleagues explicitly constructs compatible viscous and entropy terms; its particular high-order discretization is not incorporated here. [Nolasco et al., §4](#)

The next chapter constructs conservative numerical fluxes. Chapter 28 adds boundaries and a verification ladder. Any material or solver claim must name the EOS, transport coefficients, boundary regime and evidence actually obtained.

27 Conservative compressible discretization

The numerical method must carry the state and thermodynamics of Chapter 26 through every face, time stage and boundary. A low-memory implementation can store five conservative scalars per fluid cell and compute one shared face flux at a time. This reduces temporary storage; it does not remove the need for reliable reconstruction, thermodynamic validation and conservative parallel assembly.

Start from a finite-volume balance

For a stationary cell K_i with fluid volume V_i , cell average $\bar{U}_i$, outward normals $\mathbf{n}_{if}$ and open face areas A_f , a forward-Euler inviscid step is

$$V_i \bar{U}_i^{n+1} = V_i \bar{U}_i^n - \Delta t \sum_{f \in \partial K_i} A_f \hat{F}_{if}.$$

The same interior flux must enter the neighbour with opposite orientation. It follows by cancellation that the sum of cell contents changes only by boundary fluxes and declared sources. Summing cell averages without volume weights is the wrong conservation audit when cells differ in size. A consistent face solver also satisfies

$$\hat{F}(U, U; \mathbf{n}) = F_n(U), \quad \hat{F}(U_R, U_L; -\mathbf{n}) = -\hat{F}(U_L, U_R; \mathbf{n}).$$

These are finite algebraic identities. They leave open admissibility, entropy, accuracy, solid geometry and floating-point summation error. Conservation of a computed total cannot establish correct shock speed if the EOS or physical flux is wrong. The positivity analysis of Johnson and Rossmannith starts from this local conservation structure and adds separate conditions on reconstruction and time step. [Johnson and Rossmannith, §§1–2](#)

Local Lax–Friedrichs gives a transparent baseline

For admissible left and right states, the LLF or Rusanov normal flux is

$$\hat{F}_{LLF} = \frac{1}{2} \{F_n(U_L) + F_n(U_R)\} - \frac{1}{2} \alpha_f (U_R - U_L),$$

where a customary ideal-gas interface estimate is

$$\alpha_f = \max(|u_{n,L}| + c_L, |u_{n,R}| + c_R).$$

For a claimed invariant-domain theorem, the speed estimate must satisfy that theorem’s required bound on the relevant wave fan, with its specified CFL and admissible states. Endpoint estimates and the label “LLF” alone are not a proof for every EOS or reconstruction. The dissipation term damps all conservative jumps, including contacts and tangential shear. That makes first-order LLF useful for debugging a small teaching fixture, while its contact broadening and acoustic damping are visible limitations. Degond and Tang formulate a first-order LLF-based all-speed method; their semi-implicit treatment is an additional method, not a property of the explicit flux written here. [Degond and Tang](#)

A dimensional check helps implementation: α_f has units m/s, so both F_n and $\alpha_f U$ have conservative-content flux units. On a uniform one-dimensional grid, each flux difference divides by Δx , not Δx^2 . A three-dimensional fixture must update all three momentum components even when its first verification case has one-dimensional variation.

HLL separates the outer signals; HLLC restores a contact

Estimate bounding wave speeds $S_L < S_R$. The HLL flux is

$$\hat{F}_{HLL} = \begin{cases} F_L, & 0 \leq S_L, \\ \frac{S_R F_L - S_L F_R + S_L S_R (U_R - U_L)}{S_R - S_L}, & S_L < 0 < S_R, \\ F_R, & S_R \leq 0. \end{cases}$$

Here $F_L = F_n(U_L)$, $F_R = F_n(U_R)$. The middle formula follows by integrating conservation over a two-wave fan. With $S_L = -\alpha_f$, $S_R = \alpha_f$, it reduces to LLF. HLLC denotes HLL with Einfeldt-type robust speed estimates. Its single middle state diffuses the contact and shear fields; suitable wave bounds make it an important admissible fallback for an ideal-gas method under the corresponding time-step assumptions. [HLL/HLLC construction, §4.1](#)

HLLC introduces a contact speed S_M and two star states. For each side $k = L, R$, impose the jump relations and common star pressure:

$$p_* = p_k + \rho_k (S_k - u_{n,k})(S_M - u_{n,k}), \quad \rho_{*,k} = \rho_k \frac{S_k - u_{n,k}}{S_k - S_M}.$$

The star normal velocity is S_M , and tangential velocity is transported consistently. In the left-star branch, for example, use $F_L + S_L(U_{*,L} - U_L)$. This structure resolves a contact more sharply, but small denominators, a non-admissible star state and multidimensional shock instabilities need explicit treatment. Sangeeth and Mandal analyse shock instability and develop antidiffusion control. Their strong-shock study demonstrates why passing Sod alone cannot qualify a contact-resolving flux for arbitrary multidimensional shocks. This book incorporates none of their source code or figures. [HLLC and shock-instability analysis, §§4–7](#)

Record how speed estimates, degeneracies and fallback choices are handled. A fallback flux must remain shared by both adjacent cells; independently selecting a fallback on each side breaks flux cancellation. Reporting how often a fallback activates helps distinguish a robust calculation from one that spends most of its run at first order.

Roe's resolution requires additional safeguards

Roe's construction replaces the nonlinear fan with waves from a linearized matrix $\tilde{A}$ satisfying

$$\tilde{A}(U_R - U_L) = F_R - F_L, \quad \hat{F}_{Roe} = \frac{1}{2}(F_L + F_R) - \frac{1}{2}|\tilde{A}|(U_R - U_L).$$

The flux-difference property gives good resolution of an isolated shock. It does not preserve positive density or internal energy for all rarefactions. An unmodified linearized fan can also admit an expansion shock near a transonic rarefaction; an entropy fix changes the offending wave dissipation. A positivity fallback addresses a different failure and does not replace the entropy fix. The authors' Riemann notebook exhibits both failures and contrasts Roe with HLLC; it is a mathematical reference, not an imported dependency. [Roe rarefactions and positivity examples](#)

The design choice should follow the required regime. For the first independently written Euler reference, LLF has the smallest explanatory surface. HLL/HLLC offers a useful next comparison. HLLC and Roe need stronger rarefaction, strong-shock and low-Mach evidence before they become defaults. A solver name is not a complete algorithm specification.

Reconstruction, limiting and time integration

Piecewise-constant cell states give first-order spatial accuracy. A linear reconstruction can approximate smooth face states more accurately. On a uniform one-dimensional grid a simple limited slope is

$$\delta W_i = \text{minmod}(W_i - W_{i-1}, W_{i+1} - W_i), \quad W_{i\pm 1/2} = W_i \pm \frac{1}{2}\delta W_i.$$

Minmod is zero when the two arguments have opposite signs, and otherwise takes the common sign and smaller magnitude. Specify whether W contains conservative, primitive or characteristic variables. Reconstructing primitive variables is convenient; the conservative cell average remains the evolved content. Nonlinear conversion means a primitive reconstruction does not automatically define a conservative polynomial with that average. A multidimensional cut-cell method additionally needs a geometry-aware gradient, stencil and limiter.

Limiting suppresses oscillations near discontinuities, usually sacrificing local order near extrema or shocks. A componentwise scalar TVD argument is not a universal Euler-system positivity or entropy proof. WENO and other high-order reconstructions require a separate smoothness selection, quadrature and admissibility analysis; increasing polynomial order does not remove these obligations.

For a conservative reconstruction U_h whose cell average is $\bar{U}$, a positivity limiter may use

$$U_h^\theta(x) = \bar{U} + \theta[U_h(x) - \bar{U}], \quad 0 \leq \theta \leq 1.$$

Choose θ to maintain density and internal-energy bounds at the required flux and quadrature points. This scaling preserves the cell average; it cannot repair an already inadmissible average. The flux, quadrature, CFL restriction and every time stage must satisfy the selected positivity argument. Zhang and Shu's original Euler paper is identified in Shu's publication list; Johnson and Rossmanith give an inspected primary derivation of the wider convex-state framework. Their theorem does not automatically cover newly invented cut-cell or moving-wall quadrature. [Original Euler positivity publication](#), [inspected positivity framework](#)

Use a time integrator with a declared spatial operator and order. Strong-stability-preserving Runge–Kutta stages can inherit a forward-Euler admissibility property only under their corresponding stage restrictions. Reducing Δt and retrying from the previous coherent state is preferable to accepting an invalid state. Blindly flooring density or pressure changes conservative content and may create a hidden energy source. Publish any deliberate nonconservative repair and its cumulative budgets; do not relabel it as a conservative positivity guarantee.

Positivity and entropy are different contracts

Positivity keeps the thermodynamic state in its valid domain. Entropy stability controls a specified discrete entropy inequality, including boundary fluxes and source contributions. A positive state can still contain an inadmissible expansion shock. A method with an entropy inequality may still need a positivity limiter to evaluate its logarithms and

EOS. Conservation alone proves neither property. Curvilinear entropy-stable methods also need compatible metric identities; Nolasco and colleagues give an original example of that coupling. [Entropy and geometry construction, §§3–4](#)

An entropy diagnostic should state whether it is cellwise or global, semi-discrete or fully discrete, and which entropy flux is used. A global entropy rise in an insulated shock tube is a useful observation, but it can hide local violations and is not a theorem for a reconstructed three-dimensional scheme.

CFL, diffusion and the low-Mach limit

A useful local advective-acoustic estimate for stationary cells is

$$\Delta t \leq C \min_i \frac{V_i}{\sum_f A_f \alpha_{if}}, \quad \alpha_{if} \sim |u_{n,if}| + c_{if}.$$

The allowed C depends on the actual flux, quadrature, dimension and time integrator. It is an engineering estimate until linked to a specific stability argument. Report the controlling cell and mechanism. Explicit stress and conduction bring further restrictions of order h^2/ν and h^2/χ , where $\nu = \mu/\rho$ and $\chi = k/(\rho c_p)$; compressible stress has additional bulk/longitudinal coefficients, so a scalar estimate is not a complete viscous bound. Use the minimum of the selected restrictions and hit the requested final time with the actual accepted final step.

At Mach $M = U/c \ll 1$, acoustic time steps are roughly M times the flow-advection scale. Sound-speed-scaled LLF diffusion can overwhelm slow vortical motion, while physical pressure variations can be small differences on a large thermodynamic background. High-speed total-energy subtraction presents a different conditioning problem: kinetic energy can dominate internal energy. Dual-energy policies and alternative thermodynamic variables must document which equation is authoritative and how their conservation error is measured.

Low-Mach remedies include preconditioning for steady or pseudo-time iteration, and carefully designed implicit-explicit or asymptotic-preserving time discretization for physical unsteady motion. Simply replacing c in a physical explicit flux changes acoustic dynamics and its stability argument. Noelle and colleagues demonstrate that a Mach-independent CFL and formal asymptotic consistency still leave an asymptotic stability/accuracy problem in their method. This is a concrete reason to test the Mach limit independently. [Noelle et al., §§1 and 3–6](#)

The initial qualification should retain the physical sound speed, expose its accepted CFL and show its dissipation on an acoustic pulse and a smooth vortex. A broader all-speed claim requires a Mach sweep and a matching error budget. Geometry and boundaries enter the same conservative balance in the next chapter.

The accompanying [Lean source](#) makes three narrow contracts explicit. `shared_face_conservation` proves cancellation for balanced incidence and one shared extensive flux. `admissible_blend` proves convexity of the ideal-gas admissible set over exact reals. `cfl_density_positive` proves strict positivity for its stated donor-density combination with nonnegative coefficients and strict CFL; it is **not** a positivity proof for the Rusanov Euler update. The source assumptions and current proof-qualification records govern these claims. They establish no entropy inequality, arbitrary geometry assembly, floating-point accuracy or production solver.

28 Compressible boundaries and a verification ladder

Boundaries complete the conservative balance; they are not a display mask added after the fluid step. A production compressible branch would need compatible geometry, characteristic flow boundaries, wall stress and heat, coherent accepted states, and quantitative verification. The program below is a proposed qualification sequence. An executed Euler teaching fixture qualifies only its declared cells, flux, EOS and boundary model.

Prescribe information on incoming characteristics

Let $\mathbf{n}$ point out of the fluid domain and $a = (\mathbf{u} - \mathbf{v}_b) \cdot \mathbf{n}$, where $\mathbf{v}_b$ is the velocity of the boundary/control surface. The inviscid characteristic speeds relative to that surface are $a - c, a, a, a, a + c$. A negative speed carries information into the domain. For a nonsonic ideal-gas Euler boundary in three dimensions the count is:

Local regime	Incoming fields	Physical data to supply
Supersonic inflow, $a < -c$	5	A complete compatible state
Subsonic inflow, $-c < a < 0$	4	Four independent data, with one acoustic field obtained from the interior
Subsonic outflow, $0 < a < c$	1	One incoming acoustic datum, often associated with target pressure
Supersonic outflow, $a > c$	0	Interior extrapolation, with no independent downstream state

This is a local inviscid count. Sonic points, stagnation, corners, recirculation and mixed regimes need explicit handling. Total pressure, total temperature and flow direction are one common subsonic-inlet parameterization; they must combine with the outgoing interior information and EOS to produce a valid state. Prescribing density, all velocity components and pressure independently at a subsonic inlet generally imposes too much information. A changing outflow can develop inflow, requiring an appropriate exterior entropy and tangential state rather than continuing an outflow-only rule.

NASA's Wind-US documentation uses characteristic theory for freestream boundaries and identifies one physical condition at a subsonic outflow. It also reports reflection from a constant-pressure outlet. A boundary described as “nonreflecting” therefore needs a measured reflection error for the incidence angles, frequencies and mean flow actually claimed. [NASA flow-boundary documentation](#)

Viscous Navier–Stokes additionally requires compatible velocity/stress and temperature/heat conditions. The hyperbolic table alone does not specify those parabolic data. An impermeable stationary Euler wall sets $\mathbf{u} \cdot \mathbf{n} = 0$ and allows slip. A moving Euler wall sets $(\mathbf{u} - \mathbf{v}_w) \cdot \mathbf{n} = 0$. A no-slip Newtonian wall sets $\mathbf{u} = \mathbf{v}_w$, with either an adiabatic condition $\mathbf{q} \cdot \mathbf{n} = 0$, a prescribed wall temperature, or another explicitly modelled thermal relation. These are different wall models. [NASA wall conditions](#)

Shared solid geometry owns volumes and fluxes

For a stationary solid intersecting a background grid, the fluid geometry should publish one versioned description: fluid volumes, open fluid-fluid areas, oriented wall patches, connectivity and face quadrature. A triangle hit distance alone does not provide this description. A velocity collision response followed by an unchanged full-cell flux update cannot claim conservative arbitrary-solid compressible flow.

The boundary of each valid cut control volume must close geometrically. For planar area quadrature this includes

$$\sum_{f \in \partial K_i} A_f \mathbf{n}_{if} = 0.$$

Neighbouring fluid cells share the same open face geometry with opposite normals. The same wall normals and areas supply pressure forces, viscous traction, heat flux and solid reaction. A uniform static pressure then has zero resultant on a closed surface when its quadrature closes. An imbalance can produce spurious momentum even though all fluid-fluid fluxes cancel. The original metric study of Nolasco and colleagues connects geometry identities with freestream preservation; its smooth mapped-grid analysis does not prove arbitrary polyhedral clipping or topology assembly.

Geometry identities and freestream preservation

For an inviscid impermeable stationary wall, the outward conservative flux is

$$F_{wall} = (0, p_w \mathbf{n}, 0)^T.$$

The pressure p_w is the boundary/Riemann pressure, with a declared reconstruction. In a ghost-state implementation, reflect the velocity component relative to wall motion while keeping the selected density and internal energy compatible with the EOS. Recompute total energy for the reflected velocity. A generic primitive-state copy does not enforce wall impermeability.

Small cut cells introduce a real stability cost: a flux acts on the fluid volume V_i , potentially far below the background cell volume. The local explicit time-scale estimate is $V_i / \sum_f A_f \alpha_f$. Holding the full-cell step while shrinking the fluid volume can violate the selected CFL or positivity condition. Cell merging, conservative redistribution or local stepping can address this, but each needs a specified exchange, admissibility and accuracy argument. Clamping the volume fraction changes geometry and content. Report the smallest fluid fraction, the controlling cell and any stabilization transfers.

Moving solids require swept geometry and wall work

For a moving control volume, Reynolds transport gives

$$\frac{d}{dt} \int_{K(t)} U dV + \int_{\partial K(t)} [F_n(U) - v_{b,n} U] dA = 0, \quad v_{b,n} = \mathbf{v}_b \cdot \mathbf{n}.$$

The ALE flux is therefore $F_n - v_{b,n} U$. Its acoustic speeds use fluid velocity relative to the face. At an inviscid impermeable solid wall $u_n = v_{w,n}$, it reduces to

$$F_{wall}^{ALE} = (0, p_w \mathbf{n}, p_w v_{w,n})^T.$$

The outward energy flux removes energy at rate $p_w v_{w,n}$ per area. With $\mathbf{n}$ outward from the fluid, an inward-moving piston has $v_{w,n} < 0$, adding energy to the fluid. Omitting this term loses pressure work even when mass leakage is zero. For a no-slip viscous wall, the total outward energy flux is $p_w v_{w,n} - (\boldsymbol{\tau} \mathbf{n}) \cdot \mathbf{v}_w + \mathbf{q} \cdot \mathbf{n}$. Prescribed wall motion uses an external actuator; a two-way solid calculation needs the opposite force and work on the solid. The ALE derivation in Benoit and Nadarajah provides the moving-face subtraction from which these wall expressions follow. [ALE Euler balance, §2.1](#)

One necessary discrete geometric conservation law is

$$V_i^{n+1} - V_i^n = \int_{t_n}^{t_{n+1}} \sum_f A_f(t) v_{b,n,f}(t) dt.$$

Use the same time-integrated swept geometry in the volume change and the ALE content update. A freestream on compatible moving control volumes should remain a freestream; stale areas or a different velocity quadrature can manufacture density and pressure. The moving-grid study explicitly analyses integrated face velocities and tests freestream preservation. That evidence is necessary for moving geometry, but does not by itself qualify newly uncovered cells, disappearing cells or cut-cell redistribution. [Moving geometry and GCL, §§3–4](#)

A moving physical piston need not preserve a uniform pressure: it compresses the fluid. Use a compatible translating fluid/wall configuration or free moving-mesh problem for a freestream test, and a separate piston work test for compression. Topology changes must transfer conservative contents over swept regions without duplicating or discarding fluid. Publish geometry, conservative state, material parameters and accepted time together. If assembly, EOS inversion or a stage fails, retain the previous coherent state and report the failure.

Turbulence is a separate modelling choice

Euler shock capture does not establish a turbulence model. Compressible Navier–Stokes with molecular stress specifies a continuum; a simulation only qualifies as DNS when its resolution and time sampling resolve the relevant dissipative motion for the claimed regime. LES evolves filtered quantities and needs a stated subgrid stress and heat-flux treatment, explicit or implicit, with a filter/grid interpretation. RANS evolves mean quantities and closes Reynolds/Favre stresses, turbulent heat transport and any additional model equations. LES, RANS and hybrid methods introduce model-form uncertainty beyond discretization error. NASA’s primary Wind-US model documentation exposes separate laminar, RANS, LES and hybrid options and their coefficients. [NASA turbulence-model documentation](#)

State the closure variant, coefficients, turbulent Prandtl treatment, inflow turbulence, transition assumptions and calibration data. Matching one drag value after retuning coefficients is calibration, not independent validation. For a wall model report $y^+ = \rho u_\tau y / \mu$, with $u_\tau = \sqrt{|\tau_w|/\rho}$, its evaluation convention in variable-density flow, and the full first-cell distribution. Also report streamwise/spanwise spacing and wall-normal layer counts. A universal y^+ target does not cover wall-resolved LES, wall-function RANS and DNS equally. Near-wall shear and heat flux can remain wrong while a bulk velocity plot looks smooth.

A reproducible verification ladder

Use analytical solutions and conservation identities for verification of equations and code; reserve physical validation for comparisons with independent experiments and their uncertainty. NASA distinguishes verification of a code from verification of a calculation and requires convergence and consistency checks. [NASA verification](#), [NASA validation](#)

Every case should retain the EOS and nondimensionalization, initial conservative cell averages, physical boundary data, geometry version, reconstruction/flux/time integrator, all grid sizes and actual accepted steps, final time, reference method and its error, norms, conservation residuals, minimum density/pressure/internal energy and limiter/fallback counts. Save machine-readable values and original plotting scripts. Thresholds should follow a declared error budget; the case names below supply no universal pass percentage.

The separate [computed reference and reproduction instructions](#) document the bounded first-order periodic ideal-gas experiments used by the [three-dimensional lab](#). Its planar Sod data end at $t = 0.08$; periodicity introduces a second jump, so the central exact-reference comparison is restricted to $0.2 < x < 0.8$. Its initial smooth fields are sampled at cell centres, and its acoustic viewer uses two nonparallel finite-amplitude modes. These are distinct from the proposed open-tube $t = 0.2$, pulse and smooth-vortex cases below. Periodic momentum conservation and open-tube pressure-force momentum change are both correct within their respective boundary models. The README records numerical checks and limitations; the lab displays their saved fields.

V0 Local algebra and negative cases

Before a flow animation, check conservative/primitive round trips, units, uniform-state flux, orientation reversal and shared-face cancellation. Check the normal-shock ratios in Chapter 26 and their jump residuals. Present inadmissible input and excessive-CFL cases as explicit rejected operations. For solids, exercise closed area-normal sums, zero mass leakage, wall work and a GCL freestream reference separately. Exact finite proofs establish their assumptions and algebra only; they do not prove floating-point EOS evaluation or arbitrary-mesh assembly.

V1 Sod shock tube

Use ideal gas $\gamma = 1.4$, $0 \leq x \leq 1$, discontinuity at $x_0 = 0.5$, left primitive state $(\rho, u, p) = (1, 0, 1)$, right state $(0.125, 0, 0.1)$, and final time $t = 0.2$. The remaining two velocities are zero. Choose grids with even cell counts, for example 100, 200, 400 and 800, so the initial jump lies on a face. Keep endpoint data at their undisturbed states through this time; the wave fan has not reached the ends. Use unit cross-section, or report the actual extruded volume factor.

Derive the exact ideal-gas Riemann reference by solving $f_L(p_*) + f_R(p_*) + u_R - u_L = 0$, where a rarefaction branch uses

$$f_k(p) = \frac{2c_k}{\gamma - 1} [(p/p_k)^{(\gamma-1)/(2\gamma)} - 1],$$

and a shock branch uses $f_k(p) = (p - p_k)\sqrt{A_k/(p + B_k)}$, $A_k = 2/[(\gamma + 1)\rho_k]$, $B_k = (\gamma - 1)p_k/(\gamma + 1)$. Bracket the positive root and retain the root tolerance. Approximate checkpoints are $p_* = 0.3031301781$, $u_* = 0.9274526200$, $\rho_{*,L} = 0.4263194282$, $\rho_{*,R} = 0.2655737117$; independently recompute them rather than treating rounded prose as the oracle. Wang and Li give the Riemann matching construction, including its ideal-gas limit. [Riemann reference, §2](#)

Compare **conservative cell averages** of the exact solution, integrating across wave breaks, with numerical cell averages. For primitive-field comparisons, derive the reference primitives from those conservative averages, or explicitly report a different primitive-average convention: nonlinear conversion and averaging do not commute. Record density, velocity and pressure L1 errors, shock/contact locations and widths, and overshoots. Initial integrated mass is 0.5625 and total energy 1.375 in these nondimensional units. Momentum changes by the net endpoint pressure flux: its integral should be $(1 - 0.1)t = 0.18$ at $t = 0.2$, not zero. Do not demand conservation of momentum in a tube with unequal boundary pressure forces. Smooth-region order and discontinuity-dominated norms need different interpretations. This case verifies one-dimensional wave structure; it says little about turbulence or multidimensional shock instability.

V2 Small acoustic pulse and outlet reflection

Set $\rho_0 = p_0 = 1$, $\gamma = 1.4$, $c_0 = \sqrt{1.4}$, and a Gaussian $f(x) = \epsilon \exp[-((x - 0.25)/0.025)^2]$ on $[0, 1]$. Initialize $p = 1 + f$, $u = f/(\rho_0 c_0)$, $\rho = 1 + f/c_0^2$, with other velocities zero and total energy recovered from the EOS. For $t = 0.2/c_0$ compare with the right-travelling linear reference $f(x - c_0 t)$, away from boundary contamination. Sweep $\epsilon = 10^{-3}, 10^{-4}, 10^{-5}$ and at least three grids resolving the pulse, such as 256, 512 and 1024 cells.

Measure propagation-speed/phase error, amplitude loss and the acoustic combinations $p' \pm \rho_0 c_0 u'$. The unwanted left-travelling component measures a different error from damping. The linear reference has nonlinear Euler discrepancy of order ϵ^2 over fixed pre-steepening time; refinement at fixed amplitude can plateau for that reason. Repeat with the pulse leaving a characteristic outlet, then compare with a longer-domain computation at the same resolution and time to estimate reflected amplitude. State probe location, time window and amplitude normalization. A planar pulse cannot qualify oblique-wave absorption.

V3 Smooth isentropic vortex convergence

Use $\gamma = 1.4$, background $\rho = p = 1$, convection velocity $(1, 1, 0.2)$, and vortex strength $\beta = 1$. With $X = x - t$, $Y = y - t$, $r^2 = X^2 + Y^2$, define

$$\delta u = -\frac{\beta}{2\pi} Y \exp[(1 - r^2)/2], \quad \delta v = \frac{\beta}{2\pi} X \exp[(1 - r^2)/2],$$

$$\Theta = 1 - \frac{(\gamma - 1)\beta^2}{8\gamma\pi^2} \exp(1 - r^2), \quad \rho = \Theta^{1/(\gamma-1)}, \quad p = \rho^\gamma, \quad \mathbf{u} = (1 + \delta u, 1 + \delta v, 0.2).$$

These equations specify this fixture independently. Use $[-10, 10]^2 \times [0, 1]$, periodic z , analytic exterior states/compatible incoming data in x, y , and $t = 1$. A truncated Gaussian vortex is not exactly periodic in x, y ; do not call a wrapped seam an exact smooth periodic solution. Compare 64^2 , 128^2 and 256^2 grids with a declared extruded z resolution. Integrate initial and reference conservative cell averages with quadrature more accurate than the expected numerical error. Nolasco and colleagues use an analytical three-dimensional isentropic vortex to assess metric accuracy, providing a primary reference for this type of smooth test. [Vortex study, §5.1](#)

For $q = \rho, p, u, v, w$, report volume-weighted L1/L2/L ∞ errors and the observed refinement rate $\log(e_h/e_{h/2})/\log 2$. Refine time independently so spatial and temporal orders can be distinguished. First-order LLF should not be advertised as a second-order method because the reference is smooth. A z -invariant extrusion exercises three momentum components but does not verify fully three-dimensional vortex interaction or turbulence.

V4 Oblique shock and inclined wall

Use a calorically perfect gas with $\gamma = 1.4$, $M_1 = 2$, $\rho_1 = p_1 = 1$, zero attack angle and a stationary slip wedge turning the flow through $\theta = 10^\circ$. Then $u_1 = M_1 \sqrt{\gamma p_1 / \rho_1}$. Solve the weak attached-shock root β_s of

$$\tan \theta = 2 \cot \beta_s \frac{M_1^2 \sin^2 \beta_s - 1}{M_1^2 (\gamma + \cos 2\beta_s) + 2}.$$

Use upstream normal Mach $M_{n1} = M_1 \sin \beta_s$, apply the normal-shock ratios, and obtain $M_2 = M_{n2} / \sin(\beta_s - \theta)$. The weak root is approximately 39.31393° . Retain branch, bracket and angular units in the reference. NASA's oblique-shock equations state the turning relation and normal/tangential decomposition and link to the original NACA report. [NASA oblique shocks](#), [NACA 1135](#)

Publish domain extents, wedge location, wall representation, farfield/outlet data and convergence criteria. Compare shock angle, pressure and density ratios, wall impermeability and global boundary-flux budgets under grid and boundary-distance refinement. An embedded wedge needs the same shared-geometry checks as other cut cells. Add a perturbed grid-aligned strong-shock case before claiming immunity to carbuncle instability. This inviscid attached-shock reference excludes separation, viscous wall heating and detached shocks beyond the attachment limit.

V5 Laminar boundary layer, then heat and turbulence

A first viscous check can use the NASA low-Mach flat-plate study: $M_\infty = 0.1$, static pressure 6.0 psia, temperature 700 Rankine, plate length 1 ft and reported $Re_L \approx 200000$. Its documented grid, inlet/outlet and wall choices make the case reproducible. Record the selected viscosity law and exact resulting Reynolds number; do not obtain a desired Reynolds number by silently changing material parameters. No archive code or figures are included here. [NASA laminar flat plate, Study 1](#)

For the incompressible, constant-property, zero-pressure-gradient approximation, independently solve Blasius' boundary-value problem

$$f''' + \frac{1}{2} f f'' = 0, \quad f(0) = f'(0) = 0, \quad f'(\infty) = 1, \quad \eta = y \sqrt{U_\infty / (\nu x)}.$$

Its reference values give local $C_f = 0.664 / \sqrt{Re_x}$ and $\delta^* / x = 1.7208 / \sqrt{Re_x}$. Compare velocity profiles, displacement thickness and wall shear at declared downstream stations, excluding the leading-edge singular region. Refine streamwise and wall-normal resolution separately and demonstrate steady convergence. At finite Mach this incompressible comparison is an approximation; for a tighter compressible claim use a compressible similarity/reference solution with the same thermal boundary and viscosity law. Adiabatic low-Mach agreement cannot qualify high-Mach heat transfer. Add heat-conduction/thermal-wall tests before claiming conductivity, then independent transition and turbulent-wall evidence before claiming a calibrated turbulence closure.

What the ladder establishes

Passing V1–V3 can support a bounded inviscid algorithm claim for the documented domains and regimes. V4 adds inclined-wall and shock evidence; V5 adds molecular-stress evidence under a declared thermal model. Arbitrary moving solids, EOS mixtures, shock–boundary-layer interaction, high-temperature gas and turbulence remain

additional questions. Original reference data, checked finite identities and a three-dimensional teaching visualization are useful evidence with distinct scopes; none alone establishes a general production compressible solver.

A Algorithm reference

These algorithms specify future implementation behavior. The numerical companion executes a smaller independent subset. The pseudocode is not a tested Rust solver. Each routine names ownership and failure boundaries.

Validated grid construction

Inputs are positive counts, finite positive spacing, finite origin and caller-supplied memory limit. Compute all capacities and bytes with checked arithmetic. Reject overflow, unsupported dimensions or budget excess before allocation. Allocate all required buffers before publishing a state.

```
make_grid(nx, ny, nz, spacing, origin, memory_limit):
    require finite(spacing) and spacing > 0
    require finite(origin) and positive counts
    cells = checked_product(nx, ny, nz)
    xfaces = checked_product(checked_add(nx, 1), ny, nz)
    yfaces = checked_product(nx, checked_add(ny, 1), nz)
    zfaces = checked_product(nx, ny, checked_add(nz, 1))
    bytes = checked_layout_bytes(cells, xfaces, yfaces, zfaces)
    if bytes > memory_limit: return ResourceLimit
    allocate state and workspace or return AllocationFailure
    return ValidatedGrid
```

The indexing theorem applies after finite-range checks. Each view retains its shape so face arrays cannot accidentally use cell strides.

Matrix free pressure

For each pressure cell gather incident faces with nonnegative coefficients. Closed walls contribute known flux to the right-hand side. Prescribed pressure is eliminated consistently.

```
apply_A(p, out, topology):
    for each pressure cell i:
        total = 0
        for neighbor j across internal face e:
            total += weight[e] * (p[i] - p[j])
        for each prescribed-pressure face e:
            total += boundary_diagonal[e] * p[i]
        out[i] = total
```

This gives nonnegative diagonals, positive only with at least one positive retained coefficient, and nonpositive off-diagonals. Known boundary pressure adds a matching right-hand-side term. A pinned pressure needs matching row and column elimination.

Pressure solve lifecycle

The lifecycle operates on the gauge-fixed SPD system or a separately justified admissible subspace. The preconditioner is assumed SPD by a separately justified construction. Check initial residual before any quotient. Final acceptance uses a freshly computed residual.

```
solve(A, M, b, p, tolerances, iteration_limit):
    r = b - A(p)
    if not finite(r): return NonfiniteState
    if accepted(r, b): return Converged
    z = M_inverse(r)
    d = z
    rz = dot(r, z)
    if not finite(rz) or rz <= 0: return Breakdown
    for k in 0 .. iteration_limit:
        Ad = A(d)
        curvature = dot(d, Ad)
        if not finite(curvature) or curvature <= 0:
            return Breakdown
        alpha = rz / curvature
        p += alpha * d
        r -= alpha * Ad
        if candidate_tolerance_met(r):
            true_r = b - A(p)
            if accepted(true_r, b): return Converged
            r = true_r
            z = M_inverse(r)
            d = z
            rz = dot(r, z)
            if not finite(rz) or rz <= 0: return Breakdown
            continue
        z = M_inverse(r)
        next_rz = dot(r, z)
        if not finite(next_rz) or next_rz <= 0:
            return Breakdown
        beta = next_rz / rz
        d = z + beta * d
        rz = next_rz
    return IterationLimit(p, b - A(p))
```

Production also validates updated vectors and supports cancellation. Tiny negative curvature still indicates breakdown of expected SPD arithmetic; precision escalation or retry must be explicit. The stopping function combines absolute and relative scales, then verifies physical divergence.

A fixed boundary substep

This first-order proposal holds geometry fixed over a substep. Reordering requires validation.

```
step(accepted, boundary_snapshot, requested_dt):
    validate configuration and state version
    choose dt and report its limiting reason
```

```

initialize candidate buffers
advect velocity into separate face arrays
apply declared body accelerations
apply viscosity if enabled
impose prescribed normal wall velocities
build pressure topology and component gauges
build rhs through the same divergence operator
solve pressure or return a declared failure
subtract the matching pressure gradient
verify boundary consistency
recompute divergence and finite-state diagnostics
transport scalars at the declared velocity time level
integrate sources with explicit amounts
validate field bounds and diagnostics
publish candidate atomically at time + dt

```

Liquids need separately derived interface transport, classification, velocity extension and free-surface pressure stages. Adding a surface renderer does not turn this into a liquid algorithm.

Conservative amounts

Compute each shared face flux once from immutable old state. A gather update reads a flux array without races:

```

for each face e:
    flux[e] = selected_flux(old, velocity, geometry, e)
for each cell i:
    amount_next[i] = amount_old[i]
    for incident face e:
        amount_next[i] += dt * signed_inflow(i, e) * flux[e]
    amount_next[i] += integrated_source[i]

```

The conserved quantity is amount, not concentration. Divide by cell volume only when constructing concentration. Moving cells require geometric transport terms absent from this static-cell argument.

Diagnostics

A packet should contain state ID, boundary version, start/end time, accepted substeps, limiting reason, maximum and RMS divergence, true residual, iterations, total tracer or liquid amount, kinetic-energy estimate, allocated bytes and termination reason. Define units and normalization. Missing measurements are unavailable, not zero.

Execute the iteration contract

Use the two-cell Jacobi mechanism as an executable algorithm specimen. The old x supplies both residual entries before either component is updated. At $\alpha = 1/2$ inspect one, two and four iterations; record x and the freshly recomputed full residual. A pressure-correction fraction is not this loop. The stated matrix and finite domain make the contract reproducible; larger systems need explicit boundary assembly and stopping rules.

► **Practical study proof scope**

B Reproduction and evidence

Source includes chapter Markdown, appendices, document and figure builders, experiments, covers and Lean. Word equations are native and editable; PDF is the rendered reading copy. Cover fluid imagery is illustrative, not simulated evidence.

Numerical environment

Executed fixtures use Python 3.12.14, NumPy 2.3.5 and SciPy 1.17.0 on Linux x86_64. Figures use Matplotlib 3.10.8. Requirements pin these packages. The deterministic seed is 20261003. The script writes results.json including full residual histories.

From this book directory, retrieve immutable historical observations into ignored storage before generating new ignored outputs. The qualified Linux x86_64 replay uses Haswell dispatch on hardware that supports that kernel, with both wheel BLAS libraries at OpenBLAS 0.3.30. These process-local settings reproduce the separately recorded CG profile, not the historical CG trajectory byte-for-byte; historical JSON stays unchanged.

```
python3 -m pip install -r companion/requirements.txt
baseline=$(python3 companion/historical_baselines.py)
OPENBLAS_CORETYPE=Haswell OPENBLAS_NUM_THREADS=1 python3 companion/experiments.py
OPENBLAS_CORETYPE=Haswell OPENBLAS_NUM_THREADS=1 python3 companion/depth_experiments.py
OPENBLAS_CORETYPE=Haswell OPENBLAS_NUM_THREADS=1 python3 companion/verify_results.py \
    --baseline-dir "$baseline" --cg-profile haswell-openblas-0.3.30
python3 companion/make_figures.py
```

No result files or recorded CG histories are tracked in the source checkout. Verification checks historical/source hashes and the loaded runtime, substitutes only the explicitly qualified CG history, and retains the original comparison tolerances for all fields. See the companion README for profile limits. Assertions cover graph projection, rational conservation, periodic transport bounds, diffusion decay and memory arithmetic. They do not qualify production runtime or a GPU backend. Run experiment assertions with normal Python, since -O disables those scientific experiment assertions.

Lean environment

The toolchain is leanprover/lean4:v4.19.0. Mathlib is pinned to c44e0c8ee63ca166450922a373c7409c5d26b00b and transitive revisions are locked. The workflow uses a pinned checkout action and official versioned elan, read-only permissions and no retained checkout credentials.

```
cd proofs
python3 scripts/check_sources.py
lake exe cache get
lake build
lake env lean AxiomAudit.lean
python3 scripts/test_audit.py
```

Inventory digests are reviewed, not regenerated in CI. Changed statements require mathematical review, inventory updates and full qualification. A digest establishes source identity, not truth.

The first successful qualification is [workflow run 37141752646](#), associated with source head `ecf97d3a943ebe30d20acc1cf01b94dc08ff96a2`. The runner checked the pull-request merge ref against then-current main. The retained log records actual checkout and all proof/audit stages. Later proof changes require new qualification.

Boundaries and rendering

Proofs establish properties of Lean definitions. Fixtures establish small reference calculations. Rust architecture, liquids and performance remain implementation work. No production throughput, latency or GPU-memory benchmark was executed.

Rebuild and render after layout edits, inspect every page, and check native equation structure separately. Export success alone does not establish readable notation or clean pagination.

Extended mechanism fixtures

Run `python companion/depth_experiments.py` for twelve restricted fixtures. They cover staggered affine sampling, rotating-field backtracing, a two-level Dirichlet multigrid mechanism, axis-aligned interface flux, sphere curvature, weighted viscous energy, frozen-position affine particle transfer and SPH constraint gradients.

The affine sampler errors are at most 2.23×10^{-16} , while the intentionally wrong x-face offset creates a -0.07 bias. Over one unit of rotation time, midpoint tracing error decreases from 0.00166645 at 10 steps to 0.0000260416 at 80; Euler decreases from 0.0511230 to 0.00626930.

The 31-unknown Dirichlet multigrid fixture reduces residual from 1.13620 to 0.00873160 in one cycle and near floating precision in eight. Its exact coarse-space correction discrepancy is 3.17×10^{-15} . This nodal one-dimensional mechanism fixture is not the proposed cell-centered three-dimensional fluid hierarchy.

For a radius-0.5 sphere, curvature errors decrease from 0.0392195 at $h=0.1$ to 0.000624805 at $h=0.0125$. The weighted viscous fixture decreases energy from 7 to 0.941037. Its tiny computed negative minimum eigenvalue, about -2.49×10^{-16} , is numerical roundoff around the exact constant nullspace, not a proof of negative dissipation.

The 27-node quadratic particle support reproduces its affine matrix to 3.85×10^{-16} and represented angular momentum to 1.16×10^{-16} . Removing one support node and renormalizing leaves a first-moment defect of 0.205778. The SPH analytical constraint gradient differs from finite differences by at most 1.12×10^{-10} . These results validate the specified local algebra and derivatives, not a full APIC, SPH or PBF simulation.

The boundary review fixtures distinguish linear from affine divergence: omitting pressure boundary work creates a 0.21 identity defect. A skew pressure connection gives normal gradient 3.2 instead of 2 despite positive geometry. An offset fraction step confirms opposite fraction-gradient and level-set-gradient signs. A compatible chain has retained residual maximum 0.001 but full maximum 0.003, so a 0.0015 threshold must reject it despite the reduced solve criterion.

Reproduce an arithmetic counterexample

Select exponent 24, left order and binary32. Record terms, trace and output zero; switch to cancel order and record output one. Run `test_practical_reference.py` in normal and optimized Python to compare against struct-based conversion. The browser and Python implementations are separate executions; matching these twelve arithmetic cases does not certify compiler or solver behavior.

► **Practical study proof scope**

C Primary sources

The works below provide historical and technical context. Graph definitions, worked arithmetic, recommendations, experiments and Lean contracts are independently specified for Rheon. No third-party figures or long quotations are reproduced. Publisher-only citations identify original works without implying full-paper reproduction.

[S1] Jos Stam. Stable Fluids. SIGGRAPH 1999, 121-128. DOI 10.1145/311535.311548. [Author paper](#). Semi-Lagrangian graphics simulation.

[S2] Francis H. Harlow and J. Eddie Welch. Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface. Physics of Fluids 8, 2182-2189, 1965. DOI 10.1063/1.1761178. [Paper](#). Marker-and-cell foundation.

[S3] Alexandre J. Chorin. Numerical Solution of the Navier-Stokes Equations. Mathematics of Computation 22, 745-762, 1968. [Author paper](#). Projection methods.

[S4] Robert Bridson and Matthias Müller-Fischer. Fluid Simulation. SIGGRAPH 2007 Course Notes. [Author notes](#). Broad numerical implementation context.

[S5] Jonathan Richard Shewchuk. An Introduction to the Conjugate Gradient Method Without the Agonizing Pain. 1994. [Author paper](#). Krylov methods and conditioning.

[S6] Andrew Selle, Ronald Fedkiw, ByungMoon Kim, Yingjie Liu and Jarek Rossignac. An Unconditionally Stable MacCormack Method. Journal of Scientific Computing 35, 350-371, 2008. [Publisher](#). Corrected advection.

[S7] Christopher Batty, Florence Bertails and Robert Bridson. A Fast Variational Framework for Accurate Solid-Fluid Coupling. ACM Transactions on Graphics, 2007. [Author paper](#). Boundary coupling.

[S8] Stanley Osher and James A. Sethian. Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations. Journal of Computational Physics 79, 12-49, 1988. [Author paper](#). Level sets.

[S9] C. W. Hirt and B. D. Nichols. Volume of Fluid Method for the Dynamics of Free Boundaries. Journal of Computational Physics 39, 201-225, 1981. [Publisher](#). Volume fractions.

[S10] Douglas Enright, Ronald Fedkiw, Joel Ferziger and Ian Mitchell. A Hybrid Particle Level Set Method for Improved Interface Capturing. Journal of Computational Physics 183, 83-116, 2002. [Publisher](#). Particle interface correction.

[S11] Christopher Batty and Robert Bridson. A Simple Finite Difference Method for Time-Dependent, Variable Coefficient Stokes Flow on Irregular Domains. 2010 preprint. [Paper](#). Coupled viscosity and pressure.

[S12] Ronald Fedkiw, Jos Stam and Henrik Wann Jensen. Visual Simulation of Smoke. SIGGRAPH 2001. [Author paper](#). Smoke and confinement.

[S13] Matthias Müller, David Charypar and Markus Gross. Particle-Based Fluid Simulation for Interactive Applications. SCA 2003. [Author paper](#). SPH.

- [S14] Lvmin Zhang, Anyi Rao and Maneesh Agrawala. Adding Conditional Control to Text-to-Image Diffusion Models. 2023. [Paper](#). Spatial conditioning; not a Rheon-specific error guarantee.
- [S15] Miles Macklin and Matthias Müller. Position Based Fluids. ACM Transactions on Graphics 32, 2013. [Author preprint](#). Density constraints.
- [S16] Chenfanfu Jiang, Craig Schroeder, Andrew Selle, Joseph Teran and Alexey Stomakhin. The Affine Particle-In-Cell Method. ACM Transactions on Graphics 34, 2015. [Publisher](#). APIC.
- [S17] Chenfanfu Jiang, Craig Schroeder and Joseph Teran. An Angular Momentum Conserving Affine-Particle-In-Cell Method. 2016 preprint; Journal of Computational Physics, 2017. [Paper](#). Specified transfer conservation.
- [S18] David Goldberg. What Every Computer Scientist Should Know About Floating-Point Arithmetic. ACM Computing Surveys, 1991. [Authorized reprint](#). Rounding and exceptional arithmetic.
- [S19] Rheon repository README, revision 0cb48429789699578b640d30d9d5dc99c6921ce4. [Pinned source](#). Project scope; no production solver at this revision.
- [S20] MrScripty. Coding Standards, revision dcc56f26e884ade260770beceba2501d3746200d. [Pinned source](#). Ownership, evidence and performance guidance.
- [S21] Lean community. [Lean 4.19.0](#) and [pinned mathlib](#). Formal environment.
- [S22] Yousef Saad. Iterative Methods for Sparse Linear Systems, second edition. SIAM, 2003. [Author hosted book](#). Multigrid and iterative-method context. The companion hierarchy and its measured results are independently constructed.

Read a model variant before using it

Inspect the supplied SST-2003 limiter, then read the primary model/variant record linked from chapter 22. Keep the publication motivation, maintained equation specification and locally executed probe as distinct evidence. The 1992 report does not specify all 2003 coefficients. An inaccessible proceedings text is identified as unread. No figure, code or dataset from those publications is included.

► Practical study proof scope

D Notation and provenance

Velocity, pressure, density, dynamic viscosity, kinematic viscosity, spacing and physical step are $\mathbf{u}, p, \rho, \mu, \nu, h, \Delta t$. Advected scalar, level set and liquid fraction are q, ϕ, f_i . Units follow the model, not the letter alone.

In the graph model, B is cells by oriented internal faces, negative at tail and positive at head. Physical divergence is $-B\mathbf{u}$ at unit spacing; pressure difference is $B^T p$; $A = BWB^T$. Euclidean graph norms are unweighted. Physical integrals require volume or face metrics. MiB is 1,048,576 bytes; GiB is 1,073,741,824 bytes.

This research edition is credited to Puma at the user's request. Covers were generated specifically for the book and visually reviewed. They are illustrative, not numerical evidence. Original prompts and hashes accompany the artwork. Explanatory diagrams are independently generated from definitions and experiment data; no cited-paper figures are reproduced.

The native-equation publishing approach follows the project family's research-book style. Scientific content and theorem claims are Rheon-specific. The proof environment reuses a previously qualified pinned Lean/mathlib combination with independently authored fluid contracts.

The dense baseline, typed outcomes, component gauges and immutable snapshots are recommendations for Rust implementation review. Sparse grids, APIC, non-Newtonian materials and integrated diffusion evaluation remain extensions. No application GUI, calibrated presets or measured production performance is claimed.

The repository's original license is Apache 2.0. Cited papers retain their own rights and are linked rather than redistributed. Generated illustration provenance is documented without asserting identical copyright treatment in every jurisdiction. No publisher affiliation, ISBN, peer-review status or physical validation is claimed.

Units in a force and power ledger

In the wall example μ has units Pa s, U has m/s, and G has N/m³. The gap is one metre and wall area one square metre. Thus wall traction times area is a force and traction times area times wall speed is power. Body force must multiply the volume-average speed and volume. Change G and inspect signed wall/body work before comparing it with nonnegative dissipation.

► **Practical study proof scope**

E A worked matrix laboratory

Small exact systems make useful implementation fixtures because every entry has a meaning. They reveal sign errors, gauge mistakes and false conservation claims without a large visual scene.

Three cells on a line

Orient two faces from cell 0 to 1 and from 1 to 2. With unit weights,

$$B = \begin{bmatrix} -1 & 0 \\ 1 & -1 \\ 0 & 1 \end{bmatrix}, \quad A = BB^T = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}.$$

Take face velocity $(2, -1)$ and time step $1/2$. Incidence is $(-2, 3, -1)$, so right-hand side is $(-4, 6, -2)$. It sums to zero, as required. Fixing $p_0 = 0$ leaves

$$\begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = \begin{bmatrix} 6 \\ -2 \end{bmatrix}.$$

The second equation gives $p_2 = p_1 - 2$, hence $p_1 = 4$, $p_2 = 2$. Gradient is $(4, -2)$. Subtracting half of it gives zero velocity. A closed one-dimensional chain has no nonzero divergence-free internal flux; the result follows from topology, not excessive numerical damping.

Now stop at $p = (0, 3.9, 2)$. Gradient is $(3.9, -1.9)$, corrected velocity is $(0.05, -0.05)$, and incidence is $(-0.05, 0.10, -0.05)$. Residual is $(-0.10, 0.20, -0.10)$, exactly twice the corrected incidence. This fixture checks the time-step factor and the sign of physical divergence.

A cycle retains circulation

A three-cell cycle has faces 0 to 1, 1 to 2 and 2 to 0:

$$B = \begin{bmatrix} -1 & 0 & 1 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \end{bmatrix}.$$

The constant face vector (c, c, c) has zero incidence. With unit metric, projecting velocity $(3, 0, 0)$ onto this nullspace yields $(1, 1, 1)$. Its circulation survives while compressive content is removed. Initial squared norm is 9; final squared norm is 3. A test expecting zero velocity in every closed domain would be wrong because it confuses a chain with a cycle.

This graph is an algebraic teaching fixture, not a spatially resolved three-cell physical vortex. Its role is to make the nullspace distinction visible. A production mesh must separately establish which cycles and coefficients correspond to its geometry.

Deriving energy reduction

Let W be positive definite, $u^+ = u - \Delta t W B^T p$, and $Bu^+ = 0$. Then the removed vector $v = \Delta t W B^T p$ is orthogonal to u^+ in the weighted inner product:

$$(u^+)^T W^{-1} v = \Delta t (Bu^+)^T p = 0.$$

Since $u = u^+ + v$,

$$\|u\|_{W^{-1}}^2 = \|u^+\|_{W^{-1}}^2 + \|v\|_{W^{-1}}^2.$$

This establishes exact norm nonincrease for the stated closed compatible projection. If some weights vanish, W^{-1} is undefined and the argument needs restriction to active faces or a different metric. Moving boundaries and prescribed flux add work; approximate solves leave a nonzero cross term. The result is derived here but is not one of the current checked Lean theorems.

For an approximate pressure, the cross term is tied to residual. In the unweighted case,

$$(u^+)^T (\Delta t B^T p) = \Delta t (Bu^+)^T p = \Delta t^2 r^T p.$$

Thus solver error can affect energy even when the algebraic operator is correct. A small residual in one norm does not bound this term without also controlling pressure in the dual norm.

Residual and solution error

On a gauge-fixed SPD system, exact solution p_* and approximate p satisfy $A(p_* - p) = r$. Therefore

$$\|p_* - p\|_2 \leq \lambda_{\min}(A)^{-1} \|r\|_2.$$

The smallest eigenvalue depends on geometry, gauge and scaling. A universal pressure-error claim from a residual threshold is invalid without this spectral factor. Velocity error can be better related through the energy norm:

$$\|B^T(p_* - p)\|_2^2 = (p_* - p)^T A(p_* - p).$$

These are exact linear-system relations. Floating arithmetic, variable weights and physical metrics need corresponding versions. Their practical lesson is to choose diagnostics from the consumer: divergence tolerance, velocity error and pressure accuracy are related but not identical acceptance criteria.

A conservation fixture with fractions

Take scalar amounts $(1/3, 1/2, 1/6)$, cyclic face fluxes $(2/7, -1/5, 1/9)$, and time step $1/10$. Each cell adds incoming minus outgoing amount. The result is $(199/630, 96/175, 61/450)$, whose sum is exactly one. The companion executes this with rational arithmetic.

This fixture isolates cancellation. It does not prove the selected flux is physically accurate or positivity-preserving for every input. A complete transport validation adds the required flux law, time restriction, geometry and boundary terms. Conservation is a structural property; accuracy depends on how the flux approximates the intended equation.

A Dirichlet matrix and its error modes

The displayed matrix has diagonal 2 and off-diagonal -1 . Its energy is $(x-y)^2+x^2+y^2$, while the Jacobi iteration uses a separate relaxation. Select alternating b and $\alpha = 3/2$: each iteration changes the alternating error by $-5/4$ and increases its energy. This explicitly contrasts an SPD operator with an unstable chosen stationary iteration. The diagram and residual arrows describe a two-unknown dimensionless system.

► **Practical study proof scope**

F Expansion contracts, sources and evidence

This edition adds original finite contracts and local numerical references. The historical 20 theorems retain their source-bound evidence. The previous expansion receipt expansion/proof-qualification.json retains its 33-theorem/45-declaration source binding. Current integration qualification is recorded separately in expansion/bounded-proof-qualification.json: all 42 public theorems compile, 60 declarations pass the transitive axiom audit, and negative audit/source-integrity fixtures pass. The latest bounded module contributes nine theorems and six definitions. All new derivations and implementations are Rheon research work. No cited researcher reviewed or endorsed this edition.

A map from requirements to evidence

Requirement	New exact statement	Executed local reference	Remaining production gap
External forces	force_work_identity	signed finite-step work	spatial force interpolation and IEEE execution
Moving boundary	sealed_flux_compatibility	balanced/unbalanced component flux	swept mesh and geometric conservation
Planar collision; numerical mesh query	wall_first_hit, wall_hit_on_surface, clipped_segment_in_halfspace	exact rational crossing/clipping plus independent cube queries	finite-facet containment, earliest mesh query, robust predicates and cut cells
Positive density	positive_transmissibility, weighted_pressure_energy	layered hydrostatic increments	phase-interface coefficient assembly
Liquid volume	volume_source_balance	unequal-volume rational ledger	bounded multidimensional interface transport
Viscosity	viscous_work, dissipation_nonnegative, backward_euler_work, backward_euler_energy_nonincrease	exact rational operator step; dense solve versus periodic eigenmode	symmetric-gradient assembly, free-surface traction, forcing and approximate solves
Slip	slip_power_nonpositive	affine Couette boundary equations	curved wall frame and assembly
Wetting/adhesion	young_adhesion_identity, adhesion_bounds	fixed-volume spherical caps	dynamic angle, hysteresis and microscopic law
Surface tension	no full capillary theorem claimed	pressure jump and independent volume integral	balanced moving-interface force and parasitic currents

Assumptions of the new theorems

The domain is a finite set and values are exact reals. Internal_amount_balance and volume_source_balance assume balanced incidence columns. Sealed_flux_compatibility additionally assumes every cell constraint is solved, proving a necessary global compatibility condition rather than existence. Weighted_pressure_energy assumes positive areas, densities and distances and bounds a finite weighted-square energy. It does not establish that geometric assembly yields these quantities.

Hydrostatic_increment assumes nonzero density/distance and a prescribed exact pressure increment. The implicit energy results assume an exact discrete work equation; nonincrease additionally assumes nonnegative masses, time and dissipation. BoundedPhysics now derives that work identity from defined finite strain and exact coordinate update equations with fixed nonnegative weights; it does not identify a production stencil or prove solver convergence. Young_adhesion_identity assumes Young balance; adhesion_bounds treats the cosine as a bounded scalar, without proving trigonometry or interfacial thermodynamics. Slip dissipation assumes a nonnegative coefficient. Force_work_identity is a finite-step algebraic identity without a stability assertion.

No theorem proves mesh orientation, robust arbitrary-triangle intersection, wetting existence, contact-line convergence, liquid volume bounds, production Rust correspondence, floating-point error or performance. The educational reference has no general liquid solver.

Bounded assumptions and witnesses

The planar lemmas use $g(x)=n \cdot x - c$, permitted side $g \geq 0$, a strictly positive start and strictly negative proposed endpoint. The normal need not be unit length. They prove an interior first hit and clipping to it; grazing, wall starts, finite-facet containment and moving-wall queries are excluded. The viscosity lemmas use arbitrary fixed finite E and μ , $K=E^T \text{diag}(\mu)E$, nonnegative masses/time/weights and the exact coordinate backward-Euler equations. Their conclusion is a weighted squared-norm inequality, not solvability, stencil validity or a full timestep result.

evidence/bounded-physics-contracts/witness.py checks rational first contact at $1/2$, the forbidden unclipped continuation, out-of-range hits without endpoint signs, an implicit squared-norm decrease $2 \rightarrow 1/2$, and counterexamples with negative viscosity or an omitted update equation. Normal and optimized Python outputs agree. These witnesses are separate from the nine kernel-checked theorems and the browser's triangle-query/shear fixtures.

Shared data and reproduction

Run `python3 docs/research-book/expansion/reference.py` from the repository. It writes `reference-data.json`, original figures and a `source/data-bound` receipt. The JSON schema identifies every lab as an analytic or local numerical oracle. The site consumes it directly; the figures use the same values. Numerical acceptance uses independent dense/modal, geometric and rational checks, not an aesthetic comparison.

Build the teaching edition with `python3 docs/education/build.py`; its README records math, browser and PDF steps. The generated full Markdown preserves all chapters and appendices, including this source map. Historical evidence files are not rewritten.

Primary sources added

[E1] Batty, Bertails and Bridson (2007). A Fast Variational Framework for Accurate Solid-Fluid Coupling. [Primary source](#). Prescribed/dynamic coupling, adjoint reaction, wall separation; Sections 2 and 4.

[E2] Robert Bridson. Author-hosted mesh_query and Tunicate implementations. [Primary source](#). Original geometry implementations linked under Downloads; independently written Rheon segment fixture uses no copied source.

[E3] Basilisk contributors. Embedded boundaries, original source. [Primary source](#). Geometry metrics, interpolation and small-cell policies; current online code, accessed 2026-10-04.

[E4] Batty and Bridson (2008). Accurate Viscous Free Surfaces for Buckling, Coiling, and Rotating Liquids. [Primary source](#). Section 3: normal/tangential traction, viscous variational construction.

[E5] Basilisk contributors. Implicit viscous diffusion, original source. [Primary source](#). Coupled stress, residual and relaxation implementation; current code, accessed 2026-10-04.

[E6] Basilisk contributors. 3D sessile drop, original test. [Primary source](#). Constant-volume spherical-cap comparison; no code/figures copied.

[E7] Afkhami and Bussmann (2009; author preprint dated 2008). Height functions for applying contact angles to 3D VOF simulations. IJNMF 61, 827–847. DOI 10.1002/fld.1974. [Primary source](#). Section 2.5: wall frame and 3D height-function geometry.

[E8] Basilisk contributors. Contact angles, original source. [Primary source](#). 3D orientation and shallow-angle support restrictions; current code, accessed 2026-10-04.

[E9] R. G. Cox (1986). The dynamics of the spreading of liquids on a solid surface. Part 1. Viscous flow. JFM 168, 169–194. DOI 10.1017/S0022112086000332. [Primary source](#). Publisher abstract: contact-line stress singularity, microscopic regularization and small-capillary-number asymptotics. No full-paper reproduction.

[E10] Basilisk contributors. Surface tension, original source. [Primary source](#). Interfacial potential and capillary restriction; current code, accessed 2026-10-04.

Verified expertise, without endorsement

[Christopher Batty](#). Author university page lists liquid/gas simulation, viscous flows, surface tension and embedded boundaries. This verifies the relevance of the source author, not an external review of Rheon.

[Shahriar Afkhami](#). Author NJIT page identifies computational interfacial flows, dynamic contact lines and complex liquids. This verifies the relevance of the source author, not an external review of Rheon.

[Robert Bridson](#). Author UBC page lists fluid simulation research, original implementations and primary publications. This verifies the relevance of the source author, not an external review of Rheon.

Additional primary-source ledger

The focused contribution inspected these original works and implementations on 2026-10-04. R identifiers avoid collisions with the historical S bibliography. Access depth is retained; moving code URLs are references, not vendored qualified dependencies.

[R01] Batty and Bridson, Accurate Viscous Free Surfaces for Buckling Coiling and Rotating Liquids, 2008. [Primary source](#). Full original text inspected, especially sections 3–5. Coupled symmetric-gradient viscosity, free-surface traction, variational dissipation, rigid-rotation counterexample.

[R02] Brackbill, Kothe, and Zemach, A Continuum Method for Modeling Surface Tension, JCP 100, 335–354, 1992. [Primary source](#). Original-paper text surfaced through academic-hosted PDF. Continuum surface force and wall adhesion boundary treatment. DOI 10.1016/0021-9991(92)90240-Y.

- [R03] Young, An Essay on the Cohesion of Fluids, 1805. [Primary source](#). Original archival metadata verified; the journal DOI endpoint returned 403. Used only for historical attribution of material-dependent contact angle, not to attribute modern notation verbatim. DOI 10.1098/rstl.1805.0005.
- [R04] Huh and Scriven, Hydrodynamic Model of Steady Movement of a Solid Liquid Fluid Contact Line, 1971. [Primary source](#). Original abstract inspected. Creeping-flow contact-line stress/dissipation singularity under adherence. DOI 10.1016/0021-9797(71)90188-3.
- [R05] Cox, The Dynamics of the Spreading of Liquids on a Solid Surface Part 1 Viscous Flow, JFM 168, 169–194, 1986. [Primary source](#). Original abstract and bibliographic details inspected. Small-Ca matched-asymptotic/slip assumptions verified. Full publisher text was not retrieved; do not present the cubic shorthand as its general finite-angle equation. DOI 10.1017/S0022112086000332.
- [R06] Qian, Wang, and Sheng, Molecular Scale Contact Line Hydrodynamics of Immiscible Flows, PRE 68, 016306, 2003. [Primary source](#). Original full text inspected. Generalized Navier boundary condition, Cahn–Hilliard framework, molecular comparison and fitted dynamic parameters. DOI 10.1103/PhysRevE.68.016306.
- [R07] Batty, Bertails, and Bridson, A Fast Variational Framework for Accurate Solid Fluid Coupling, 2007. [Primary source](#). Original project, paper text, and sample-code descriptions inspected. Kinetic projection, static irregular-boundary samples, and author hindsight on improved weights. Sample archive license not verified.
- [R08] Popinet, An Accurate Adaptive Solver for Surface Tension Driven Interfacial Flows, JCP 228, 5838–5866, 2009. [Primary source](#). Original publisher abstract/conclusion inspected. Balanced-force VOF with height-function curvature and stationary-drop equilibrium. Author PDF endpoint timed out. DOI 10.1016/j.jcp.2009.04.042.
- [R09] Basilisk original contact-angle implementation. [Primary source](#). Source and explicit limitations inspected. Height-function normal fallback and shallow-angle limitation are visible in the implementation.
- [R10] Basilisk original surface-tension implementation. [Primary source](#). Source inspected. Explicit capillary timestep restriction and curvature potential construction.
- [R11] Akinci, Akinci, and Teschner, Versatile Surface Tension and Adhesion for SPH Fluids, TOG 32(6), 2013. [Primary source](#). Original text inspected. Cohesion and boundary attraction as practical SPH models. DOI 10.1145/2508363.2508395.
- [R12] Weiler, Koschier, Brand, and Bender, A Physically Consistent Implicit Viscosity Solver for SPH Fluids, CGF 37(2), 2018. [Primary source](#). Author abstract and accepted-paper text inspected. Conservation, refinement, and spurious-viscosity evaluation criteria.
- [R13] Bender et al., Implicit Frictional Boundary Handling for SPH. [Primary source](#). Original manuscript passages inspected. Boundary viscosity/friction extension and separate boundary coefficient. Filename uses 2020; final issue metadata was not verified.
- [R14] SPlisHSPlasH Weiler2018 implementation. [Primary source](#). Source inspection confirms μ = configured viscosity $\times$ rest density and separate boundary coefficient. No build/execution.
- [R15] Macklin and Müller, Position Based Fluids, 2013. [Primary source](#). Original algorithm inspected. Positional density constraints, artificial pressure, velocity postprocessing, and performance/accuracy tradeoff.

- [R16] Koschier and Bender, Density Maps for Improved SPH Boundary Handling, SCA 2017. [Primary source](#). Original author abstract inspected. Precomputed continuous boundary density contribution.
- [R17] Bender, Kugelschadt, Weiler, and Koschier, Volume Maps An Implicit Boundary Representation for SPH, MIG 2019. [Primary source](#). Original text inspected. Kernel-independent geometric-volume contribution, rigid boundaries, comparison with density maps.
- [R18] Guendelman, Selle, Losasso, and Fedkiw, Coupling Water and Smoke to Thin Deformable and Rigid Shells, SIGGRAPH 2005. [Primary source](#). Original abstract and paper passages inspected. Infinitesimally thin surfaces, ray-based visibility, and incompressibility treatment.
- [R19] Basilisk original embedded-boundary implementation. [Primary source](#). Source inspected. Volume/face geometry, small-cell CFL restriction, tracer redistribution. Static implementation details are not asserted to establish arbitrary moving-boundary correctness.
- [R20] Basilisk core license text. [Primary source](#). GPL version 3 text verified. Per-file scope and downstream compatibility still require checking.
- [R21] SPLisHSPlasH repository license. [Primary source](#). MIT notice verified. Repository README separately warns that dependencies have their own terms.
- [R22] Basilisk original 2D sessile-drop test. [Primary source](#). Source and analytic geometry comparison inspected. The implementation tests 15° – 165° and notes shallower limitations.
- [R23] Basilisk original 3D sessile-drop test. [Primary source](#). Original source inspected as the 3D comparison. No local reproduction performed.
- [R24] Basilisk original parasitic-current test. [Primary source](#). Source inspected. Constant density, diameter 0.8, Laplace-number-controlled viscosity, viscous-time observation, resolution study.
- [R25] Basilisk original drop-oscillation test. [Primary source](#). Source/diagnostics inspected. Frequency error and equivalent numerical viscosity are separate quantities. Its dimensional setup must be retained when reproducing it.
- [R26] Washburn, The Dynamics of Capillary Flow, Physical Review 17, 273–283, 1921. [Primary source](#). Original abstract inspected. Cylindrical-capillary square-root penetration formula. Full text is access controlled; no access bypass attempted.
- [R27] Gong et al., Contactless Surface Tension Measurement of Molten Oxides Using Oscillating Drop Method in an Aerodynamic Levitator. [Primary source](#). Original measurement equations and experimental cautions inspected. Supports Rayleigh frequency convention and the need to control external forcing, mode splitting, and rotation. No material values imported as Rheon presets.
- [R28] François, Cummins, Dendy, Kothe, Sicilian, and Williams, A Balanced Force Algorithm for Continuous and Sharp Interfacial Surface Tension Models within a Volume Tracking Framework, JCP 213, 141–173, 2006. [Primary source](#). Original author-uploaded text inspected; publisher endpoint failed. Exact-curvature balanced-force experiment is an appropriate control before testing estimated curvature. DOI 10.1016/j.jcp.2005.08.004.

[R29] SPLisHSPlasH Akinci2013 implementation. [Primary source](#). Original source inspected. Separate fluid and boundary force coefficients, with boundary-method branches. No assumption that every branch reproduces every original-paper conservation claim.

Next operator proofs, not compiled claims

The nine focused targets are stationary implicit strain/wall dissipation; moving-wall work with residual norm; affine pressure constraint work; two-way impulse cancellation; rigid strain null modes; conservative redistribution/geometric conservation; balanced capillary rest; work-compatible surface energy; and contact-line friction/pinning. The current checked Physics module covers finite energy identities, conservation, positive coefficients, relative slip power and Young algebra. The new BoundedPhysics module supplies the unforced finite-strain work implication within the first target; wall terms, operator solvability, assembled strain validity and the larger surface-evolution targets remain future work.

In particular, a nonnegative stiffness is insufficient for a rigid-rotation null mode. A conservative redistribution matrix needs column sums one, but positivity and barrier connectivity need additional conditions. Balanced rest assumes M is symmetric positive definite, an exact solve, and a capillary force actually in the chosen gradient range. Surface-energy conservation requires a discrete gradient satisfying its exact work identity; triangle area is not globally convex in arbitrary vertex coordinates. A scalar contact-coordinate dissipation proof cannot establish a 3D contact-line PDE.

The contributed `expansion/contact/sanity_checks.py` executes 28 algebra/formula checks, including 100 synthetic SPD systems for each of the stationary, moving-wall and affine-projection identities. These remain numerical sanity checks, separate from Lean compilation and from a Rheon liquid run. No third-party research code or paper figure is bundled.

Proof assumptions and runtime admission

The nonfinite proposal preset is refused and the exact fallback is displayed. The Lean statements describe finite exact-real residuals, gauge invariance and a conditional tolerance bound; they do not admit NaN as a real number. A numerical candidate must satisfy finite/unit/source checks before those mathematical properties apply. Production native packets additionally require their reviewed source inventory and fresh qualification receipt. Historical results cannot be transferred to changed source bytes.

► Practical study proof scope

One progression: wall traction, exact trace and external force

Read [finite wall friction](#), then [compatible no-slip](#), then [uniform forcing and Poiseuille startup](#). These three native operations share one fixed flat-column model and one endpoint basis. They do not compose a general liquid solver. The original six 3D reference laboratories remain separate analytical/dense-solve teaching mechanisms.

What stays fixed, and what each step adds

The slab has fixed height H and area A , periodic lateral directions, zero normal motion and layer-uniform tangential velocity. Density ρ is constant and positive; dynamic viscosity μ is in Pa s. Liquid dual lengths ω_j give $V_j = A \omega_j$ and $m_j = \rho V_j$. Each viscous edge has stiffness $\mu A/h$. The retained first/last wet-center value extends constantly to the physical wall, so a constrained trace can be exact while the continuum field retains endpoint spatial error. This is not an evolving free-surface/pressure update.

Step	Native API	New physical term	What the recorded lab compares
PR22: finite Navier walls	update_with_walls	$F_b = -\beta_b A(u_b - w_b)$, β_b in Pa s/m	Six native cases; both walls have finite friction. Symmetric continuum denominator $H + 2\ell$, $\ell = \mu/\beta_b$.
PR23: exact compatible trace	update_with_boundaries	$\Delta u_b = 0$ and constraint reaction $J_b = -dt F_{\text{unconstrained}}$	Two no-slip cases and one lower-Navier/upper-no-slip case. Mixed continuum denominator $H + \ell$, not $H + 2\ell$.
PR24: one uniform tangential force	update_with_forcing	$f_{\text{body}} = ma$ for acceleration; $f_{\text{body}} = Vq$ for force density; include it before reaction	Seven stationary-wall Poiseuille cases, with continuum steady, discrete steady and finite-time discrete references.

The no-slip API requires the incoming trace to match its prescribed fixed speed. It is stateless and cannot infer an intercall wall-speed change when the caller supplies a matching state. Two no-slip walls on one wet node are refused because their separate reactions are nonunique. A mixed one-node law can have a unique reaction. No large artificial β_b replaces a constraint.

A common momentum and work language

Let δ be the combined explicit increment before storage rounding, and let J contain no-slip reaction impulses. The most general step in this progression is

$$m_j \delta_j = \Delta t (f_{\text{bulk},j} + f_{\text{Navier},j} + f_{\text{body},j}) + J_j.$$

Free nodes have $J=0$. Compatible constrained nodes have $\delta=0$, so reaction cancels every unconstrained contribution, including body force and any Navier traction sharing that node. Body force is applied to its actual liquid dual volume even at constrained endpoints; wall reaction is not obtained by dropping those nodes from the external

impulse.

With bulk and relative Navier dissipation measured at the incoming velocity,

$$\Delta T + \Delta t(P_{bulk} + P_{Navier}) - W_{wall} - W_{body,old} - \frac{1}{2}\delta^T M \delta - W_{round} = 0, \quad W_{body,old} = \Delta t \sum_j f_{body,j} \cdot u_j.$$

Momentum changes by body impulse plus total wall impulse plus stored-f32 rounding momentum. Prescribed wall work is wall velocity dotted with delivered wall impulse and can be negative. Compatible no-slip relative reaction work $J \cdot (u-w)$ vanishes without removing bulk dissipation. A stationary wall does no work while still exchanging momentum. From rest, body old-speed work is zero on the first step, but the explicit squared-increment energy is positive. This combined-step convention differs from midpoint applied work in the separate [box force stage](#).

The free-row explicit admission is unchanged by a constant body source. No-slip rows have zero increment; arithmetic checks still apply. All gates and cancellation callbacks precede output publication, with the same five fixed scratch vectors and no per-call allocation. These are native runtime contracts, not an implicit integrator, general geometry theorem or whole-program memory certificate.

Keep the reference problems separate

For symmetric finite slip, $u(y)=U(y+ell)/(H+2ell)$. The native discrete equilibrium is $U(jh+ell)/((L-1)h+2ell)$. For lower slip and an upper no-slip wall, the respective denominators are $H+ell$ and $(L-1)h+ell$. Both no-slip Couette references are Uy/H and $Uj/(L-1)$.

For stationary walls driven by uniform $q=G$, the continuum steady reference is $Gy(H-y)/(2\mu)$, with flow per width $GH^3/(12\mu)$. The retained discrete equilibrium is $Gh^2j(L-1-j)/(2\mu)$. At fixed G , density changes relaxation but not steady amplitude; at fixed acceleration, $G=\rho a$ changes too. The Poiseuille lab uses a body source and does not solve a pressure gradient.

Only the forcing lab supplies a closed-form exact-real explicit discrete startup comparison. Its binary64 evaluation has trigonometric, power and sum rounding, while the native assembly also rounds before f32 storage. There is no certified reference-evaluation bound. A steady reference at an early saved time is not a transient oracle. Distance to equilibrium includes unfinished relaxation; decreasing endpoint deviation does not establish temporal order.

Proofs and recorded demos

Lean source	Checked statement family	Explicit limits
WallFriction.lean	Force/work decomposition, translation, relative loss, explicit work	Assumes the finite step and assembled power; no stencil/IEEE/wetting proof.
NoSlip.lean	Compatible trace, zero relative reaction work, momentum, constrained work	Compatible fixed trace and per-node step; no wall-speed evolution or geometry proof.
ColumnForcing.lean	Units, body-inclusive reaction, forced momentum/work, parabola/stencil algebra, equilibrium recurrence	Step, internal cancellation and assembled power remain explicit; no spectral-reference/convergence or IEEE proof.

The integrated native proof source has 57 public theorems and 77 audited declarations at accepted PR24 source. The older expansion receipt's 42/60 counts remain historical, with their own source identity. Definitions and generated helpers are not added to the public-theorem count.

The [native lab hub](#) connects all three recorded packets when supplied to the book builder. Every case/time control is explicitly snapshot playback. The PR22 plot is its final saved profile; its CSV contains four saved times. PR23/24 sliders choose five actual saved profiles. They do not recompute arbitrary parameters or run a live fluid solver. The builder verifies frozen input bytes and performs zero numerical calls. A source-only CI build without recorded packets states that they are absent, rather than silently substituting new runs or fabricated results.

These steps qualify local wall/force mechanisms. Continue to the [requirements map and next geometry contract](#) before extending beyond the fixed slab.

Native finite wall friction in a fixed flat slab

This implements the wall-traction slice of Chapters 19 and 24. The existing explicit body-force API and collision triangle queries retain their own scopes. The new operation advances liquid shear velocities, rather than changing their visualization. It reuses the fixed flat-column mass/basis from [column shear quadrature](#).

API and physical units

`ColumnShearWorkspace::update_with_walls` consumes two `ColumnShearWall` values in lower/upper order. Velocity is a world-space tangential vector in m/s; normal motion is rejected. Friction beta is finite and nonnegative in Pa s/m. Zero beta is free slip. Positive beta requires positive dynamic viscosity mu in Pa s; positive density rho in kg/m³ weights inertia. For mu,beta > 0 the continuum slip length is mu/beta. Exact no-slip is not represented by a large artificial coefficient; it remains unsupported by this finite-coefficient operation. The separate [compatible endpoint no-slip operation](#) supplies an explicit constraint with reaction impulse/work accounting. Wetting and adhesion are separate, unimplemented mechanisms, not synonyms for this wall friction.

The supplied column geometry stays fixed. Its cap becomes a confining wall for this call; this is not a moving free-surface or a sealed MAC carrier update. Lateral directions are periodic, normal velocity is zero, and tangential velocity must be constant across each layer. The inherited varying-height and unsupported-field refusals remain. Output is copied only after all gates.

Equation, momentum and energy

Let A be slab area, $m_j = \rho A \omega_j$ the existing liquid-only lumped mass, and $k = \mu A/h$ the interior edge stiffness. With endpoint trace u_b and prescribed wall velocity w_b , the force on the liquid is

$$\begin{aligned} F_b &= -\beta_b A (u_b - w_b). \\ M \delta &= dt (-K u + \sum_b \text{endpoint}_b F_b), \quad v = u + \delta. \end{aligned}$$

The existing constant endpoint extensions make the wall trace equal to the first/last wet-center value. This deliberately retains the existing basis; it has a finite spatial endpoint error and is not an exact continuum wall trace reconstruction. If one wet node remains, both walls act on that same mass.

The bulk dissipative power is $P_{\text{bulk}} = u^T K u$; relative wall power loss is $P_{\text{wall}} = \sum_b \beta_b A |u_b - w_b|^2$. Actuator power delivered to the liquid is $W = \sum_b w_b \cdot F_b$, which can have either sign. Exact-real algebra gives

$$\begin{aligned} T(v) - T(u) + dt(P_{\text{bulk}} + P_{\text{wall}}) - dt W - \delta^T M \delta / 2 &= 0. \\ \text{momentum}(v) - \text{momentum}(u) &= dt \sum_b F_b. \end{aligned}$$

Rust accounts separately for actual stored-f32 rounding work and momentum. `ColumnWallShearReport` exposes each wall force, bulk/relative-wall dissipation and actuator work. Its nested shear report includes total dissipation and wall impulse in the balance checks. Moving walls may add energy; this is not hidden behind a damping parameter.

Common wall/fluid translation generates no drag.

The existing 64-epsilon identity budgets are retained with explicit wall-work and impulse magnitudes added to their scale. The explicit admission number is $\text{dt max}_j((\text{adjacent edge weights} + \text{incident beta } A)/m_j)$, required ≤ 1 . Active wall velocities extend the convex bound. There are no automatic substeps or coefficient adjustments. No new workspace vectors or per-step allocations are introduced; the existing five normal-count arrays are reused.

`proofs/Rheon/WallFriction.lean` proves exact wall-force/work decomposition, translation invariance, nonnegative relative dissipation, and the finite explicit work identity under declared step and power hypotheses. It does not prove Rust/IEEE arithmetic, stencil assembly, spatial convergence or wetting.

Reproducible native laboratory

Build and run from the repository, selecting a fresh directory outside Git:

```
cargo build --release --no-default-features --example column_wall_shear
target/release/examples/column_wall_shear /tmp/rheon-native-wall-lab
```

The lab performs exactly six cases of 8192 steps over 96 seconds, with a unit height/area slab, lower wall speed zero and upper speed 1 m/s. Baselines use 8 and 16 layers, $\rho=3$, $\mu=0.15$ and $\beta=0.3$ on both walls. Four additional 16-layer cases double density, double viscosity, halve friction and double friction independently. Four transient snapshots and all parameter units are exported. HTML shows actual Rust profiles alongside a labeled steady reference; it does not pretend the browser runs a liquid solver.

For symmetric slip length $\ell=\mu/\beta$ the continuum steady reference is $u(y)=(y+\ell)/(H+2\ell)$. The discrete fixed-basis equilibrium at node j is $u_j=(j h+\ell)/((L-1)h+2\ell)$. Distance from the latter checks closeness to the known discrete equilibrium and combines unfinished physical relaxation, explicit-step transient error and stored-f32 rounding; it does not isolate integration error or establish temporal order. The difference between the two steady references exposes the endpoint spatial error. The fixed engineering acceptance criteria require decreasing continuum max error at $8 \rightarrow 16$ layers, finest error <0.01 m/s, and both discrete-equilibrium errors $<5e-4$ m/s. The other parameter cases are reported as finite transients, not automatically advertised as equilibrium or continuum convergence.

Open generated `index.html`, with the adjacent CSV/HTML files kept together. All generated lab outputs, logs, plots and compiled proof products belong outside Git. Existing failed numerical assertions, E2 evidence and pressure selection refusal remain preserved in their separate frozen branches. This milestone does not qualify variable density, general 3D strain, collision-mesh fluid topology, interface evolution, contact-angle physics or material fits.

Primary-source scope

Qian, Wang and Sheng, *Physical Review E* 68, 016306 (2003), [original author-hosted paper](#), describes ordinary Navier slip away from a contact line and gives slip length as viscosity divided by the wall coefficient. Its generalized contact-line condition additionally includes uncompensated Young stress. This operation implements only the single-phase linear wall-traction law, not that generalized condition or the paper's diffuse-interface model.

The original [Huh–Scriven publisher abstract](#) (1971, DOI 10.1016/0021-9797(71)90188-3) reports the moving-contact-line stress and dissipation singularity under adherence. The original [Cox publisher abstract](#) (1986, DOI 10.1017/S0022112086000332) treats small-capillary-number spreading with microscopic slip or another local mechanism. Those two abstract-level checks motivate the contact-line exclusions; no full-text formula from either is claimed to be implemented here.

Exact compatible endpoint no-slip in a fixed slab

`ColumnShearWorkspace::update_with_boundaries` extends the independently accepted finite-Navier operation with explicit tangential endpoint constraints. It retains the same five normal-count f64 scratch arrays, liquid-only lumped mass, interior viscosity stiffness, output publication gate and 64-epsilon identity budgets. The old `update` and `update_with_walls` APIs remain available.

Boundary data and limits

Lower/upper `ColumnShearBoundary` values select either `Navier(ColumnShearWall)` or `NoSlip { velocity: [f32; 3] }`. Navier coefficients remain finite and nonnegative in Pa s/m; zero is free slip. A no-slip velocity is in m/s, representable in the stored f32 field, and constant for this step. Normal motion, nonfinite/subnormal speeds and zero viscosity at a no-slip wall are refused. Positive density and the fixed-flat-slab prerequisites still apply.

The incoming endpoint trace must equal its prescribed no-slip velocity. The operation does not silently project an incompatible state or account for an impulsive change in prescribed wall speed. A compatible initial field may be nonzero at a moving endpoint; its initial energy and momentum must be retained. Two no-slip walls on the same wet node are refused even when their speeds agree, because their separate reaction impulses are nonunique. One no-slip wall and one finite Navier wall on a single wet node have a unique reaction and are supported. All refusals and cancellation leave the caller's output unchanged.

The retained basis extends the first/last wet-center value constantly to its wall. No-slip is exact for that trace at both ends of the step. This does not make the interior continuum representation exact: the basis still has spatial endpoint error. The cap is confining, geometry stays fixed, lateral directions are periodic, normal velocity is zero, and tangential fields are layer-uniform.

Reaction impulse, momentum and work

Let f_j be the bulk plus finite-Navier force before constraint elimination, $m_j = \rho A \omega_j$ the lumped mass, and δ_j the exact-real increment. At a free node, $m_j \delta_j = dt f_j$. At a compatible constrained node, $\delta_j = 0$, so

$$\begin{aligned} J_j &= -dt f_j, & m_j \delta_j &= dt f_j + J_j. \\ W_{\text{constraint}} &= \sum_j w_j \text{dot } J_j. \end{aligned}$$

J is reaction impulse on the liquid in N s, and W is delivered work in J. The reported reaction force is J/dt , assembled directly as $-f$ rather than through an artificial large coefficient. Moving walls can supply or extract energy. No-slip relative reaction work is $J \text{dot } (u-w) = 0$; this is not zero bulk viscous dissipation. A finite Navier wall keeps its relative slip dissipation separately.

The new report exposes each total wall force, total wall impulse and no-slip reaction impulse. Total wall impulse includes finite Navier traction; reaction impulse is zero at Navier walls. Actuator work includes all walls. The balance is

```
P(v)-P(u) = sum_b I_b + stored-f32 rounding momentum.
T(v)-T(u) + dt(P_bulk+P_Navier) - W_all
- delta^T M delta/2 - stored-f32 rounding work = 0.
```

For mixed laws on a single node, the no-slip reaction also cancels the other wall's finite Navier force. Their impulses and work remain separate. No-slip rows have zero increment and are excluded from the explicit admission maximum. Free rows retain $dt(\text{sum adjacent stiffness} + \text{incident Navier beta } A)/m \leq 1$, including an edge connected to a constrained endpoint. A fully constrained two-node slab has no explicit free row; its admission number is zero. No hidden substeps, coefficient changes or automatic numerical retries are introduced.

`proofs/Rheon/NoSlip.lean` proves four exact-real contracts: compatible reaction preserves the trace, relative reaction work vanishes, finite momentum balances, and the constrained finite work identity. The latter two assume the per-node step equation; the energy statement additionally assumes assembled bulk/Navier power and compatible reaction work. These algebraic statements do not prove Rust/IEEE behavior, assembly, stability, spatial convergence or temporal order.

Native interactive laboratory

Build the example and select a fresh directory outside Git:

```
cargo build --release --no-default-features --example column_no_slip
target/release/examples/column_no_slip /tmp/rheon-no-slip-lab
```

Open `index.html` beside `results.json` and the three CSV files. The case selector chooses 8/16-layer two-wall no-slip or 16-layer lower-Navier/upper-no-slip. The time control selects five actual native snapshots, including the compatible initial field. A prominent snapshot-playback note beside the controls states that the three recorded cases are fixed and provide no live recomputation or general liquid solver. It neither interpolates nor recomputes a trajectory in JavaScript. The ledger displays cumulative reaction and total impulses, signed work, bulk/relative-wall losses, initial energy, explicit and rounding corrections.

Exactly three cases run 8192 steps each through 96 s. All use a unit height/area, $\rho=3 \text{ kg/m}^3$, $\mu=0.15 \text{ Pa s}$, lower wall speed zero and upper speed 1 m/s. The mixed case uses lower $\beta=0.3 \text{ Pa s/m}$ and $\ell=\mu/\beta=0.5 \text{ m}$. The upper endpoint starts at 1 m/s; other wet nodes start at zero, so initial energy is $\rho A h/2$.

With $H=1$ and $h=H/L$, the continuum and discrete steady references are

```
both no-slip:    u(y)=y/H,      u_j=j/(L-1).
mixed lower slip: u(y)=(y+ell)/(H+ell),
                  u_j=(j h+ell)/((L-1)h+ell).
```

The mixed continuum denominator matches Chapter 24's original one-wall oracle; it does not use the symmetric two-wall Navier denominator. Native states at wet centers compare against $y=(j+1/2)h$. Distance from the discrete equilibrium includes unfinished relaxation, explicit transient error and f32 rounding. It is not an isolated integration

error or a temporal-order measurement.

New engineering gates are declared before the run: two-no-slip continuum max deviation must decrease $8 \rightarrow 16$ and be <0.032 m/s at 16 layers; mixed deviation must be <0.021 m/s; all three discrete-equilibrium deviations must be $<5e-5$ m/s. These bounds acknowledge the retained endpoint basis and do not alter any original finite-Navier gates. Endpoint no-slip is checked at every native step. Generated data, plots, logs, ZIPs and compiled artifacts stay outside Git.

Exclusions

The separate [uniform tangential forcing operation](#) uses these compatible constraints and includes body force in their reaction, with explicit force units and a Poiseuille startup lab.

This is neither wetting nor adhesion. It does not qualify contact-line physics, arbitrary collision-mesh fluid topology, moving normal walls, time-varying prescribed speeds, variable-density inertia, general 3D stress or evolving interfaces. Historical failed assertions, pressure selection refusal, E2 evidence and PR22 refinement outputs remain separate and preserved. No E2 roster or new reference integration is part of this feature.

Uniform tangential forcing and Poiseuille startup

`ColumnShearWorkspace::update_with_forcing` adds one uniform tangential vector to the compatible fixed-wall column operation. `UniformColumnForce` uses the existing `ForceUnits` enum: acceleration in m/s^2 or force density in N/m^3 . For actual liquid dual volume $V_j = A \omega_j$ and $m_j = \rho V_j$,

```
acceleration: f_body,j=m_j a.  
force density: f_body,j=V_j q, with equivalent a=q/rho.
```

The implementation assembles these products directly using the selected units. It does not apply the same numeric vector as both quantities. Positive constant ρ , finite normal-or-zero vector components and zero normal force are required. No support regions, pressure gradient solve or variable-density inertia is added.

Actual simultaneous explicit step

On the unchanged endpoint basis, an edge transfers $k(u_{\{j+1\}} - u_j)$, $k = \mu A/h$. Its two equal/opposite node contributions retain internal momentum. Finite Navier traction is assembled next, then body force on every liquid dual node, including constrained endpoints. A free node has

```
m_j delta_j=dt(f_bulk,j+f_Navier,j+f_body,j).
```

At a compatible fixed no-slip endpoint $\delta_j=0$ and

```
J_j=-dt(f_bulk,j+f_Navier,j+f_body,j).
```

This is one combined explicit step, not a force/viscosity splitting scheme. No hidden substeps or retries are supplied. Free rows retain the existing $dt(\text{sum adjacent stiffness} + \text{incident beta } A)/m \leq 1$ admission; a constant source does not change the homogeneous row coefficients. The unforced convex velocity interval is enlarged by the signed $dt f_{\text{body}}/m$ for each row, with the corresponding rounded-f32 interval checked separately. There is no clamp. Normal/dry velocities stay zero. All checks/callbacks precede output publication; the same five fixed scratch vectors are reused with no per-call allocation. Old APIs and zero-forcing results remain compatible.

`ColumnForcedShearReport` contains the original boundary report and supplied force units/vector, total body force, body impulse and signed old-speed work. Its nested `shear.force_sum` is the net force after constraint elimination, including body forces and wall reaction. Body totals include the liquid dual volume at constrained nodes; omitting those nodes would misstate wall reaction.

Momentum and work

The stored-velocity ledgers retain all terms independently:

```

Delta momentum = body impulse + total wall impulse + rounding momentum.
W_body,old = dt sum(f_body,j dot u_j).
Delta T + dt(P_bulk+P_Navier) - W_wall - W_body,old
- sum(m_j |delta_j|^2)/2 - rounding_work = 0.

```

Work delivered by moving walls and body old-speed work can each be negative. Stationary walls transfer momentum but perform zero work. From rest, the first body old-speed work is zero while explicit increment energy is positive. The squared-increment correction belongs to the combined step, including cross terms; it cannot be assigned solely to a separately applied body-force stage. This convention differs from midpoint applied work in the existing box force stage. Runtime budgets retain the original 64-epsilon identity policy and include the actual body-force and absolute body-power scales.

Steady oracle and finite-time behavior

With stationary no-slip walls at 0,H, uniform $q=G$, constant positive ρ/μ , zero normal motion and periodic lateral directions, the continuum equation is $\rho u_t = \mu u_{yy} + G$. Its steady profile and flow per unit width are

$$u(y) = G y(H-y)/(2\mu), \quad Q' = G H^3/(12\mu).$$

No pressure field is solved here. A pressure-gradient drive $G = -dp/dx$ has the same tangential momentum source under these assumptions. Holding G fixed gives the same steady speed at different ρ , but relaxation changes with $\nu = \mu/\rho$. Holding acceleration a fixed makes $G = \rho a$, changing steady amplitude as well. Doubling μ at fixed G halves steady speed and changes the relaxation rate.

The retained basis fixes wet-center endpoint values and extends them constantly to the physical wall. For full dual lengths $h=H/L$ its distinct equilibrium is

$$u^*_j = G h^2 j(L-1-j)/(2\mu), \quad j=0, \dots, L-1.$$

The physical continuum reference is evaluated at $y_j = (j+1/2)h$; these are different references. Decreasing endpoint deviation is spatial evidence, not temporal-order qualification. Partial dual lengths are covered by contracts, not by this full-layer analytic equilibrium.

From rest, the exact-real explicit recurrence has an independent closed form. For $M=L-1$, $b_k = (2/M) \sum_{j=1}^{M-1} u^*_j \sin(k \pi j/M)$,

$$\begin{aligned} \lambda_{k=1-4} &= \nu dt/h^2 \sin^2(k \pi/(2M)). \\ u_j^n &= u^*_j - \sum_{k=1}^{M-1} b_k \lambda_k^n \sin(k \pi j/M). \end{aligned}$$

column_poiseuille evaluates these finite sums at saved times; it never integrates a second reference trajectory. This exact-real discrete startup reference compares native storage/arithmetic against a binary64 evaluation of the closed-form exact-real discrete recurrence. Observed deviation includes f32 storage, native assembly rounding and reference sin/power/sum rounding; no certified reference-evaluation bound is supplied. It does not measure error relative to a continuum transient or establish time order. The runtime row condition bounds the homogeneous amplification; no new formal spectral convergence or IEEE theorem is claimed.

Recorded native laboratory

Build cargo build --release --no-default-features --example column_poiseuille and run the executable with a fresh directory outside Git. It writes JSON, seven CSVs and a standalone HTML playback lab. Case/time controls select actual saved native profiles and ledger entries, with a visible snapshot-playback notice. They do not recompute parameters, interpolate a trajectory or provide a general solver. Native, continuum steady, discrete steady and exact-real discrete startup references remain visibly distinct.

All cases use $H=A=1$, initial velocity zero, stationary walls, 8192 steps through 96 s, $dt=96/8192$ s, and save steps 0,128,683,2731,8192.

Case	Layers	rho (kg/m ³)	mu (Pa s)	Supplied quantity
q-8	8	3	0.15	q=0.3 N/m ³
q-16	16	3	0.15	q=0.3 N/m ³
a-16	16	3	0.15	a=0.1 m/s ²
q-rho6-16	16	6	0.15	q=0.3 N/m ³
a-rho6-16	16	6	0.15	a=0.1 m/s ²
q-mu2-16	16	3	0.3	q=0.3 N/m ³
q-negative-16	16	3	0.15	q=-0.3 N/m ³

Predeclared gates: saved discrete-startup deviation and final equilibrium deviation $<3e-5$ m/s; saved cumulative energy/momentum residuals $<1e-9$ SI units; baseline continuum deviation decreases $8 \rightarrow 16$ and every 16-layer deviation is $<0.031 |G/\mu|/2$ m/s. Final equivalent-unit and sign-reversal profiles agree within $2e-7$ m/s; fixed-G density, fixed-a doubling and doubled-mu controls agree within $3e-5$ m/s with their respective steady predictions. At step683, fixed-G density doubling has lower peak speed and fixed-a doubling has higher peak speed than the baseline. These new engineering gates leave earlier gates intact.

The frozen original run passed all gates. Baseline continuum deviations at 8/16 layers were 0.05859480798 / 0.03027459979 m/s, and their separate discrete equilibrium deviations were $1.057982445e-6$ / $1.162290573e-6$ m/s. Across all seven cases the largest saved discrete-startup deviation was $4.738568319e-6$ m/s. At step683 (8.00390625 s), the 16-layer baseline peak was 0.2162252367 m/s; fixed-G density doubling gave 0.1947998554 m/s, whereas fixed-a density doubling gave 0.3895997107 m/s. The two baseline unit formulations produced identical stored profiles. Generated receipts and full data remain outside Git.

Proof assumptions and scope

`Rheon.ColumnForcing` proves exact-real unit conversion, compatible reaction including body force, finite momentum cancellation, forced work, zero parabola endpoints, quadratic-stencil balance and recurrence about equilibrium. Mass/ volume meaning, the step equation, internal-force cancellation and assembled power are explicit inputs. Polynomial identities do not prove derivative assembly, Rust floating point, the DST reference or transient convergence.

This is a bounded fixed-column force/viscosity/wall coupling. It does not add arbitrary collision-mesh fluid geometry, pressure composition, evolving liquid interfaces, adhesion, wetting or contact lines, and does not complete the capstone integration roadmap. Historical numerical refusals and outputs remain separate and preserved.

One static obstacle geometry for collision and flux

`StaticObstacleGeometry` is a bounded geometry foundation. It consumes a `GridGeometry` and the original immutable `TriangleSurface`, admitting a closed axis-aligned box with eight unique corners and twelve outward triangles. Each face must have two complementary triangles sharing a rectangle diagonal. World-space vertices express translation and axis-preserving scaling in metres. Open, duplicate, inward, non-box and unresolved geometry is refused. The existing two-sided collision query API retains its broader, separate contract.

The owner retains that exact surface and stamp for collision queries. Cell volumes, one opening per MAC face, connectivity and the conservative flux helper derive from the same admitted vertices. Borrowed accessors do not reconstruct a second mesh. This closes a bounded part of the collision-mesh requirement: geometry and flux can agree about the stationary solid. Existing pressure, transport and viscosity workspaces still require their original filled-box or column domains. A [separate bounded follow-on](#) now borrows this owner for sealed pressure and reduced extruded shear; it does not retrofit the old carrier.

Measures at represented endpoints

The physical cell boundaries are the evaluated world endpoints $x_d(k) = \text{fl}(o_d + kh_d)$. Define the exact-real interval overlap at those represented endpoints by

$$L([a, b], [c, d]) = \max(0, \min(b, d) - \max(a, c)).$$

For cell intervals $I_{i,d}$, the admitted solid intervals S_d , physical widths $H_{i,d}$, and overlaps $q_{i,d} = L(I_{i,d}, S_d)$,

$$V_i = H_{i,x}H_{i,y}H_{i,z} - q_{i,x}q_{i,y}q_{i,z}.$$

For a face normal to axis d at coordinate x_f , with tangential axes a, b ,

$$A_f = H_{f,a}H_{f,b} - \mathbf{1}_{x_f \in S_d} L(I_{f,a}, S_a) L(I_{f,b}, S_b).$$

The box is closed: equality on its boundary blocks that face intersection. Volume has units m^3 and area m^2 . Outer faces report geometric openings; wall or inlet speeds remain explicit caller data. A box touching the container can leave its interior volume unchanged while blocking a boundary opening.

These formulas are exact-real specifications. Rust evaluates them in binary64. Products, complements and flux products refuse nonfinite, subnormal or lost positive measures, including a positive obstruction rounded away by subtraction. There is no minimum-fraction clamp. Dyadic fixtures have zero comparison allowance. The non-dyadic control compares rational measures at the represented endpoints using an assembly allowance of 64ϵ times the corresponding physical cell, face or domain measure. This is a declared fixture check, not a universal IEEE error theorem or a temporal convergence result.

Shared incidence and connected fluid

Each internal face stores one area. Its negative-side cell uses $+A_f u_f$ and its positive-side cell uses $-A_f u_f$. Thus internal flux cancels in exact arithmetic. `outward_flux` borrows the three velocity arrays and computes

$$Q_i = \sum_{f \in \partial i} \sigma_{if} A_f u_f, \quad [Q_i] = \text{m}^3/\text{s}.$$

It performs no field mutation, allocation or pressure correction. The recorded controls exercise one interior face at (1,1,1) on each axis with 2 m/s speed and zero other speeds, checking the native neighboring flux pair against the shared area. This checks spatial incidence, not a fluid time step.

Positive-volume cells receive deterministic component labels through positive shared openings. Dry cells have no component. A box that spans a cell completely in two directions and lies strictly inside its third interval splits that cell's fluid into disconnected pieces. One cell unknown cannot represent those pieces; construction refuses `UnresolvedCellTopology`. A whole grid-aligned separator instead admits two distinct fluid components. No topology cleanup silently removes a sliver or connects it across the solid.

For a later closed-domain pressure solve, each isolated component needs its own compatibility condition and pressure gauge. Componentwise constant pressures have zero weighted internal jump when every active edge joins equal labels. That observation alone does not establish coercivity, the complete nullspace, boundary compatibility or convergence of a future pressure operator.

If a future integrated residual obeys $V_i D_i = \Delta t r_i$ in an active cell, then

$$D_i = \frac{\Delta t r_i}{V_i}.$$

Tiny positive volumes amplify divergence for a fixed integrated residual. Zero-volume cells cannot be pressure unknowns. The present feature assembles no such pressure system and makes no residual qualification claim.

Resources and publication

Construction accounts for actual retained `Vec` capacities of the consumed surface, volumes, all face arrays and labels. The constructor peak adds its simultaneous BFS queue, which is released before returning the owner. Checked arithmetic plans these payloads before allocation; fallible reservation and actual capacity checks can refuse the configured managed-buffer limit. An allocator can round a reservation upward before its actual capacity is checked. Allocator metadata, fixed stack, caller/callback storage and process RSS are excluded; this is not a hard process peak bound or a whole-program memory certificate.

Cancellation is checked in cell, face and connectivity loops. A refused construction does not modify a separately retained earlier immutable owner. Publication is explicit: the caller replaces its owner only after success.

Analytic controls and the recorded lab

The [native exporter](#) constructs nine geometry controls: partial box intersections; X/Y/Z separators; dry, outside and container-touching boxes; translated/scaled geometry; and a non-dyadic case. Three thin subcell separators are recorded refusals. No time integration, pressure solve or reference integration runs. Outputs must be fresh and outside Git. The [rational checker](#) independently checks all cell and face measures, global volume, oriented closed edge incidence, union-find components, analytic segment/box first hits and flux cancellation. Its receipt binds the source, executable and records.

The [interactive geometry lab](#) selects those native records, cells and face/collision axes. Cell boxes show labels, while the original triangulated box shows the solid. Readouts expose physical measures, the fixed native flux pair and the same-source collision hit. There are no live arbitrary obstacle controls or fluid advances. A source-only edition without a qualified packet states that the records are absent.

Exact contracts and original research

[StaticObstacle.lean](#) adds seven public theorems for nonnegative, symmetric and bounded overlap; bounded volume complement; shared flux cancellation; component-constant weighted jumps; and residual/divergence scaling. Labels, active edges, nonnegative dimensions and the integrated residual equation are stated assumptions. The geometry-stage local pinned Lean build audited 64 public theorems and 87 declarations across that stage's project with the existing axiom allowlist. Rust admission, triangulation, binary64 predicates, BFS correctness and solver refinement are unproved. The earlier PR24 source and 57/77 kernel qualification remain historical and separate.

Batty, Bertails and Bridson formulate pressure through kinetic energy and use fluid mass in staggered velocity volumes; §2.1 also discusses face-area-based approximations. This owner supplies primal cell volumes and shared face openings, not their complete dual-mass projection or solid coupling. Their original paper motivates retaining subgrid geometry instead of binary voxel masks. [Original paper, 2007](#).

Basilisk's original embedded-boundary source stores cell and face fractions and guards interpolation against coupling disconnected regions. Its topology cleanup can change fractions; Rheon's bounded owner instead refuses unresolved subcell topology. This source supports the need for consistent geometry and topology, not an implementation-equivalence claim. [Original embedded-boundary source](#).

The [bounded operator follow-on](#) adds stationary sealed pressure assembly/correction with component compatibility, gauges, rest and field controls, plus a separate reduced shear traction/work ledger. General viscous embedded-wall traction still needs additional interaction geometry. Arbitrary closed meshes, multiple fluid pieces within a cell, moving or two-way solids, variable material coefficients and contact-angle evolution remain unresolved. See the [requirements map](#).

Stationary obstacle pressure and reduced wall shear

The same [immutable static box owner](#) now supplies geometry to two bounded field operators. Both borrow its retained triangle surface, physical cell volumes and unique shared face openings. Neither creates another collision mesh or substitutes a binary solid mask. The original geometry lab remains a construction-only record. The new [field-response lab](#) plays native pressure and reduced-shear results with independent rational checks.

These operators have **different declared boundary models**. Pressure uses a sealed stationary container. Shear is a fully developed, tangentially invariant/periodic extrusion above one stationary obstacle wall. They are not a composed three-dimensional fluid step. Arbitrary embedded tensor viscosity, evolving liquid interfaces, moving solids, general material variation and adhesion/contact-angle evolution remain open.

Pressure: geometry, incidence, units and inertia

For each internal open face, let a be its negative-side cell, b its positive-side cell, $A_f > 0$ the owner's opening in m^2 , and $d_f > 0$ the separation in metres of its represented nominal grid pressure centers. The unknowns are the positive-volume cells only. A pressure point can lie inside the cut-away part of a cell: this face-area method does not relocate general cut-cell pressure centers or establish a continuum consistency order there.

With constant density $\rho > 0$ in kg/m^3 , define

$$w_f = \frac{A_f}{\rho d_f}, \quad m_f = \rho A_f d_f, \quad (Lp)_i = \sum_{f \sim i} w_f (p_i - p_j).$$

m_f is a declared face-area lumped mass. It is **not** the exact clipped staggered dual-cell mass. The operator is symmetric in its integrated form; $p^T Lp = \sum_f w_f (p_b - p_a)^2 \geq 0$ in exact arithmetic. No minimum volume, area or distance clamp changes the admitted geometry. Invalid, subnormal, nonfinite or lost-positive coefficients refuse construction.

The owner's outward flux uses $+A_f u_f$ at a and $-A_f u_f$ at b . All outer container faces and fully blocked faces must have exactly zero speed. For an explicit positive step Δt in seconds,

$$b_i = -\frac{Q_i^*}{\Delta t}, \quad Lp = b, \quad u_f = u_f^* - \frac{\Delta t}{\rho d_f} (p_b - p_a).$$

Pressure has units Pa. Q has units m^3/s ; the integrated right-hand side, operator result and residual have units m^3/s^2 . Pressure correction changes the actual stored binary64 field, rather than merely reporting a residual.

Components, gauges and acceptance

The retained geometry labels active cells by positive-opening connectivity. Each sealed component has its own constant-pressure mode and compatibility condition $\sum_{i \in C} b_i = 0$. The workspace pins a largest-volume wet cell of each component to zero, with the lowest index winning exact volume ties, including isolated singleton components. Dry cells are not unknowns and must have zero right-hand side. An all-dry owner is a valid zero operation.

Compatibility is checked independently for every component with a compensated sum and a declared $64\epsilon \sum |b_i|$ allowance. Incompatible data are refused; no global mean subtraction quietly transfers volume between isolated regions.

The component sum may be nonzero within that arithmetic allowance. In exact residual accounting, if the non-gauge rows are solved, the pinned row retains that component sum. Using a largest-volume cell minimizes its local divergence amplification $\Delta t |\sum_C b|/V_{\text{gauge}}$ among the component's cells. It does not remove incompatibility, alter the RHS, relax a tolerance or guarantee convergence. The full residual and actual-field gates still apply. An exhausted/nonpositive internal PCG product is classified as `ArithmeticFailure`, not caller `InvalidParameter`.

Deterministic regressions use a first-cell volume 2^{-30} , an explicitly roundoff-compatible RHS $[-1, 1 + \epsilon]$, and a three-cell flux field with speeds 0.1 and 1 m/s whose represented RHS has a -2^{-55} component defect. They distinguish unavoidable compatibility roundoff from amplification caused by choosing a tiny gauge cell. Equal-volume and disconnected-component controls retain the stable tie break. Earlier first-cell-gauge records remain historical; new packets must qualify against the current gauge oracle and source.

Jacobi-preconditioned conjugate gradients operate on the gauge-eliminated system. Every candidate acceptance recomputes the **full** integrated residual, including gauge rows. With $r = b - Lp$, the exact identities have the signs

$$Q_i^{\text{new}} = -\Delta t r_i, \quad D_i^{\text{new}} = -\frac{\Delta t r_i}{V_i}.$$

Both the full residual norm and $\max_i \Delta t |r_i|/V_i$ must satisfy the caller's tolerances. After correction, the owner independently differentiates the actual stored face speeds and checks $\max_i |Q_i^{\text{new}}|/V_i$ again. A tiny positive cell cannot hide behind a small integrated residual. A contract with volume 2^{-30} deliberately fails a loose integrated-residual gate because its local divergence is too large. This is a numerical acceptance control, not a lower bound on all future cut-cell conditioning.

The face-area kinetic energy is $E(u) = \frac{1}{2} \sum_f m_f u_f^2$. In exact arithmetic the correction satisfies

$$E(u^{\text{new}}) - E(u^*) + E(u^{\text{new}} - u^*) = \Delta t \sum_i p_i Q_i^{\text{new}}.$$

The right side is numerical residual work. Stationary impermeable walls do no mechanical work. Rust reports all terms, checks the identity with its declared binary64 allowance, and permits no unexplained energy increase. This ledger does not change the declared lumped-mass approximation into an exact geometric mass theorem.

Reduced shear: when scalar diffusion really is Newtonian strain

`StaticObstacleShear` admits only a box covering the entire two-dimensional cross-section and extending from below the container to one lower wall. The fluid above it is a connected channel. The normal direction may be X, Y or Z; the laboratory uses Y. The upper wall is stationary no-slip. Tangential directions are invariant/periodic, so no sealed end-wall response is implied.

For the exact reduced continuum mode $\mathbf{u} = (u(y), 0, 0)$, constant Newtonian symmetric strain gives shear stress $\tau = \mu u_y$ and dissipation μu_y^2 . Thus this scalar reduction is physically justified **within that mode**. It is not a license to smooth each component independently near arbitrary embedded or free-surface boundaries.

Each fluid layer has volume V_i obtained by summing the owner's cell volumes, centroid y_i from its actual clipped interval, and mass $M_i = \rho V_i$. Interlayer area $A_{i+1/2}$ is the sum of the owner's shared open normal faces. The conductance and equal-opposite fluid forces are

$$k_{i+1/2} = \frac{\mu A_{i+1/2}}{y_{i+1} - y_i}, \quad F_{i \leftarrow i+1} = k_{i+1/2}(u_{i+1} - u_i).$$

For wall-to-centroid distance $d_w > 0$, stationary no-slip gives $k_w = \mu A_w / d_w$. A finite Navier friction coefficient $\beta \geq 0$, in Pa s/m, gives the series fluid/wall closure

$$k_w = \frac{A_w \mu \beta}{\mu + \beta d_w}.$$

At $\beta = 0$ there is zero tangential wall traction. NoSlip is its own boundary mode, not an enormous finite beta. The effective boundary dissipation combines the fluid half-layer loss and wall friction. Increasing beta changes friction; it does not introduce adhesion, wall/interface energy or wetting.

Forcing, one update and its work ledger

Dynamic viscosity μ is Pa s; kinematic viscosity is μ/ρ in m²/s. Force input is explicit: Density (q) means N/m³, while Acceleration (a) means m/s² and converts to $q = \rho a$. With the symmetric traction matrix K , including stationary-wall diagonal terms, the update is

$$(M + \Delta t K)v = Mu + \Delta t qV.$$

A tridiagonal direct solve performs one backward-Euler step. There are no hidden substeps. The exact discrete energy and momentum identities are

$$E(v) - E(u) + E(v - u) + \Delta t v^T K v = \Delta t \sum_i q V_i v_i,$$

$$\sum_i M_i (v_i - u_i) = \Delta t \sum_i q V_i - \Delta t (k_{\text{lower}} v_0 + k_{\text{upper}} v_{n-1}).$$

The source checks both identities and the true momentum equation residual before publishing the field. Unforced kinetic energy cannot increase beyond the declared arithmetic allowance. This is a discrete forced/dissipative response, not a transient continuum-reference or temporal-convergence qualification.

A one-layer hand control has area 1 m², fluid thickness 0.5 m, density 2 kg/m³, viscosity 1 Pa s, and stationary no-slip walls. Its mass is 1 kg and total conductance is 8 kg/s. At $\Delta t = 0.25$ s, $q = 2$ N/m³, starting from 1 m/s, the exact discrete answer is $v = (1 + 0.25)/(1 + 0.25 \cdot 8) = 5/12$ m/s. Changing the lower closure to finite Navier friction or free slip changes the actual velocity, force and energy ledgers.

Native records and independent controls

The [exporter](#) records four pressure cases: partial-box gradient removal, an outside-box control, a disconnected separator, and partial-box rest. Six shear cases record initial rest and eight updates, comparing no-slip/Navier/free-slip and fixed-q/fixed-a density responses. These are 4 pressure projections and 48 bounded shear updates. No historical seven-equation campaign is run or replaced.

The [checker](#) recomputes box measures and components with exact rational arithmetic. It checks analytic affine pressures, actual corrected speeds, cellwise divergence and pressure work. A tiny dense rational Gaussian-elimination oracle checks each shear field independently of the Rust tridiagonal algorithm, along with energy and impulse balances. Its 48 rational linear solves are fixture-level discrete controls, not continuum reference integrations. Separately, the checker substitutes the **actual recorded old and new fields**, plus the independently checked represented coefficients, into the momentum equations and energy/impulse ledgers. The exact reference trajectory never substitutes for those stored fields.

The actual shear equation gate is the native $4096\epsilon \max_i(|\text{inertia}_i| + |\text{stress}_i| + |\text{load}_i|)$. Pressure energy uses 1024ϵ times the actual ledger-term scale; shear energy and momentum use their documented 2048ϵ scales. Exported maxima and budgets must be finite and nonnegative. Every exported budget is checked against its formula from the reported terms, with a separate 16ϵ relative allowance for that short arithmetic expression. Reported ledger terms/residuals are compared to exact rational recomputation using 64ϵ times their contributing-term scales. These checks have **no unit-sized absolute floor**. They are explicit fixture arithmetic allowances, not new operator tolerances or universal IEEE error theorems. The independent ideal-trajectory field allowance remains $4096\epsilon \max(1, |\text{exact control}|)$; pressure additionally requires independent actual and predicted divergence at most 10^{-10} s^{-1} .

Records, executable hashes, qualification receipts, browser captures, generated book/PDF editions and logs stay outside Git. Publication requires a frozen clean source, exact record bytes and matching source hashes. Missing packets produce an explicit source-only page. Browser controls select saved native fields; they do not solve fluid equations in JavaScript. Corruption tests must reject changed geometry, gauges, fields, force semantics and ledgers in normal and optimized Python.

Proof boundary and original research

The existing `Discrete` and `StaticObstacle` Lean modules establish exact-real graph symmetry/energy, incidence cancellation, component-constant jumps and conditional positive-volume scaling. [ObstacleOperators.lean](#) adds 24 public theorems and 15 definitions. With pinned Lean 4.19.0 and the unchanged axiom allowlist, all ten modules and the root compile; 88 public theorems, 118 explicitly expected declarations and 135 total declarations are audited. The three real negative audit probes and four source/pin tests pass in normal and optimized Python. This local qualification invoked the pinned compiler directly with private dependency artifacts; it is not a new `lake build` or hosted-CI receipt.

The new contracts derive signed/absolute residual-divergence scaling, per-label compatibility, matched-mass pressure work, forced shear energy and two-wall momentum, the Navier traction/loss split, and the reduced 3-by-3 symmetric-strain identity. Exact shear coordinate equations and physical coefficient assumptions remain explicit hypotheses. No

theorem here proves Rust mesh admission, IEEE arithmetic, PCG/Thomas convergence, complete graph-nullspace identification or arbitrary-domain continuum accuracy. All historical proof receipts and failed numerical evidence retain their own source identities.

Batty, Bertails and Bridson's original pressure work formulates projection as a kinetic-energy minimization and discusses subgrid geometry and face-area approximations. It motivates the matched coefficient/mass relation; this bounded stationary implementation does not reproduce their moving rigid/deformable coupling. [Original paper, 2007](#). The author's [reference implementation](#) explicitly distinguishes face-area weights from fluid-volume weights.

Batty and Bridson's viscosity work derives an implicit variational update from symmetric-strain dissipation and emphasizes the coupled traction condition at free surfaces. The present exact shear reduction has neither general free-surface traction nor arbitrary embedded tensor assembly. [Original paper, 2008](#).

The next physical gate is shared staggered interaction geometry for general embedded symmetric strain and an integrated pressure/viscosity step under one consistent boundary model. Variable-density transport, moving/two-way solids, capillarity and contact-angle dynamics need their own contracts and evidence.

Requirements map and the next solid-fluid geometry contract

The first table records the earlier PR24/main and bounded static-obstacle checkpoint; the current reviewed runtime composition is described below. The new native wall/force sequence is qualified within its fixed-column model; the broader liquid capability is incomplete. Source links below freeze the inspected PR24 source, so a future implementation cannot silently change these claims.

User requirement	Actual implementation	Research/teaching coverage	Remaining capability
Collision meshes	TriangleSurface provides immutable two-sided earliest segment queries/clipping and stamps; translation has its own prescribed-motion query scope. Open/disconnected/intersecting surfaces are admitted.	Chapter20 explains shared cut volumes, face openings and connectivity. BoundedPhysics proves infinite-plane crossing/clipping, not arbitrary triangle-solid assembly.	StaticObstacleGeometry now admits a single closed axis-aligned box and derives shared volumes, openings, labels and conservative flux from the retained surface. General mesh cuts remain absent. A bounded follow-on now adds sealed obstacle pressure and a separate reduced extruded-shear response from the same owner; general 3D coupling remains absent. A hit normal alone is not fluid topology.
Forces	BodyForce / ForceUnits supports explicit acceleration/force-density vectors and optional regions in the fixed box; LiquidStepInputs borrows them. PR24 adds uniform tangential forcing to the separate constrained slab.	Chapter19 explains external and moving-boundary work; the Poiseuille lab separates fixed-q and fixed-a density controls.	Mesh-based traction, moving-solid displaced volume/work and two-way solid impulse/torque are not supplied by these force APIs.
Viscosity	ViscosityWorkspace assembles symmetric Newtonian strain on a fully filled sealed MAC box with stationary free-slip walls. step_viscous retains that domain restriction. PR22–24 qualify a separate layer-uniform fixed slab with Navier/no-slip.	Chapter22 and finite-strain Lean identities distinguish dissipation from stencil correctness; fitted/coupled prototypes retain separate research scopes.	The same static owner now supplies a strictly admitted reduced shear channel , with actual cut-layer inertia, traction, forced response and work/impulse ledgers. No arbitrary embedded-wall stress, general evolving-free-surface traction, spatially varying/nonlinear viscosity or viscosity/mesh/interface composition in the general carrier. Passing an isolated slab does not remove this gap.
Stickiness / adhesion	Finite beta is mechanical slip friction. It has units Pa s/m and is not an adhesive material law. No implemented contact-angle/adhesion evolution is exposed by the slab.	Chapter23 defines wall/interface energies, Young angle and adhesion work; the cap lab selects analytic equilibrium records at fixed volume.	A declared wall energy/contact-angle reconstruction, balanced capillarity and moving contact-line regularization are needed. An attractive force or larger beta cannot substitute for these.

User requirement	Actual implementation	Research/teaching coverage	Remaining capability
Density	LiquidTransportConfig uses a constant carrier density and represented density, with equality required in surface modes. Box/slab inertia and $\nu=\mu/\rho$ have explicit units.	Chapter21 includes hydrostatic layers and weighted pressure identities; the forced lab checks density scaling within a single constant material.	No general advected variable-density/two-phase inertia and matching variable pressure/stress coefficients, conservative interphase momentum or density-ratio liquid qualification.

The [liquid composition](#) and fixed-box viscosity milestones already exist; they are not the remaining requirement or a new completion claim. Volume/pressure classifications, interval geometry and output publication are real bounded components. General three-dimensional fluid topology and coupled physical boundaries remain separate prerequisites. The unresolved research qualifications and failed numerical evidence are retained; this book integration executes no new E2 roster or reference integrations.

Geometry foundation and bounded operator follow-on

The [bounded geometry owner](#) retains an admitted closed box surface and supplies physical cell volumes, one shared opening per MAC face, component labels and a conservative flux helper. It refuses unsupported meshes and disconnected fluid pieces within one cell. The [operator follow-on](#) borrows this owner for sealed pressure and a separately bounded reduced shear mode. It does not add those operators to the existing carrier or compose their different boundary models. World-space vertices express translation and axis-preserving scaling; no moving-body work is implied.

Nine recorded native controls and independent rational checks cover wet/dry and partial cuts, three separator orientations, translations/scaling, container contact, collision source agreement and shared flux incidence. Three thin baffles are recorded topology refusals. The non-dyadic fixture has a declared binary64 assembly allowance; exact-real Lean identities do not prove implementation refinement. Managed buffer capacities and constructor queue lifetimes are reported with an explicit scope, rather than a whole-program cap claim.

The pressure follow-on checks component compatibility/gauges, full integrated and cellwise residuals, actual corrected fields and a face-area kinetic ledger. The reduced shear follow-on checks owner-derived inertia/traction and a forced momentum/work ledger. The next gate is general embedded symmetric-strain interaction geometry and a composed step under consistent boundaries. Material variation and balanced capillarity/contact-angle reconstruction come after their coefficient/interface contracts exist. Moving/two-way solids add swept geometry and reaction/torque obligations. This sequence connects the remaining mesh, force, viscosity, density and adhesion requirements without claiming capstone completeness.

Current reviewed runtime composition

The historical links above remain evidence for their recorded checkpoint. [Runtime PR65](#) and the [book integration PR67](#) now provide a reviewable combined source. Qualification/publication of that combined source is tracked separately; a capability being present is not a physical-validation claim.

Capability in the combined source	Exact scope and remaining obligation
Native OBJ/GLTF/GLB import and CollisionMesh	Finite bounded static geometry, convex watertight solid or explicitly two-sided surface barrier. Native filesystem import is excluded from the WASM browser host. Particle collision is available; general fluid cut-cell topology and swept moving-volume assembly are still absent.
Bounded obstacle and plate laboratories	Owner-derived sealed pressure, separate reduced shear, and finite-mass moving-plate coupling remain their explicitly reduced models. They do not compose arbitrary imported CAD into a general 3D FSI solver.
MaterialSession/SimulationSession/PressureSession	Version-two bounded APIs expose units, immutable accepted snapshots, cooperative cancellation and explicit capability flags. Constructor pressure is unavailable; the particle material model does not call the grid pressure provider.
Qualified pressure provider in a full step	Opt-in sealed full-box carrier path retains pressure and final-step acceptance before publication. Boundary flux, surface modes and viscosity composition are refused on this path. Provider/model/schema/resource provenance and real inference qualification are distinct from physical accuracy and speed.
Quantitative desktop views and VTK	Particle diagnostics, bounded obstacle/transport slices/probes and immutable snapshot exports identify their evaluation/publication stages. A pressure evaluated before integration is not a pressure recomputed at later accepted particle positions.
Vector/moving emitter controls	Jet velocity, source translation and seeded perturbations are independent prescribed controls. The perturbations are not a solved or validated SST turbulence model.

Consequently, static collision import is no longer the next missing implementation in this composition. The remaining geometry gate is conservative fluid topology, shared face flux/stress assembly and interval geometry under consistent physical boundaries. These requirements differ from collision queries and inspection. The [engineering studies](#) supply bounded worked examples and scoped proofs without presenting that missing backend as done.

Implementation guide: the bounded material preview

The desktop can emit physical material parcels into an initially empty box. This is the role of [ParticleFluidSimulation](#). It is a bounded SPH preview with its own density and force model. The original carrier solver, pressure methods and recorded replay contracts remain available in the other desktop mode. The browser's 24³ concentration lesson still uses that carrier model; it does not run this particle simulation.

Read [Chapter 11](#) for the wider particle-method design space, [material scenes](#) for controls and units, and the [desktop owner](#) for live history and thread lifecycle. No PBF, PIC, FLIP or APIC implementation is implied by this preview's use of particles.

What one parcel represents

A parcel stores position in metres, velocity in m/s, mass in kg, temperature in K and a material identity. The numerical spacing determines a reference parcel volume. Emitted parcels receive mass equal to source reference density times that volume. Initial filled scenes distribute the declared material mass over their seeded parcels. The emitter supplies mass from an external reservoir; it is not a closed-system mass-conservation experiment while injection is on.

For a filled cube of side L and reference density ρ_0 , the intended initial mass is $\rho_0 L^3$. Doubling L therefore multiplies this mass by eight at fixed density. Changing a display unit or camera scale does not do so. The native fixture checks physical mass at several domain sizes. It does not establish that geometrically scaled moving scenes share the same dynamics: gravity, acoustic limits and source pressure also set physical scales.

Density and pressure are declared approximations

Let W_h be the compact poly6 kernel and $\rho_{0,j}$ the reference density of neighbor j 's material. The current estimate is

$$f_i = \sum_j \frac{m_j}{\rho_{0,j}} W_h(x_i - x_j), \quad \rho_i = \rho_{0,i} \max(f_i, 0.05).$$

Self density is included. Bins identify candidate neighbors, and an exact support-distance test chooses contributors. The reference-volume factor prevents a denser material's parcel from being interpreted as extra volume of the receiving material. The floor keeps estimated density positive; it changes the formula and must be included in an implementation comparison.

The implemented pressure laws are

$$p_i = \max(c^2(\rho_i - \rho_{0,i}), 0) \quad \text{for liquid}, \quad p_i = \rho_i R_i T_i \quad \text{for gas}.$$

Here c is a deliberately reduced liquid acoustic stiffness, and R_i is the gas-specific constant. The gas reference density is chosen to reproduce its specified reference pressure and temperature. A liquid can compress in this model; a zero tensile pressure does not impose exact incompressibility. Gas expansion into vacuum is an approximate parcel behavior, not a qualified model of shocks, nozzle choking or rarefied gas kinetics.

Pair forces: momentum and energy answer different questions

With the native density estimates held fixed, define

$$A_{ij} = \frac{p_i \rho_{0,i}}{\rho_{0,j} \rho_i^2} + \frac{p_j \rho_{0,j}}{\rho_{0,i} \rho_j^2}, \quad F_{ij}^p = -m_i m_j A_{ij} \nabla W_h(x_i - x_j).$$

Exchanging i and j leaves A_{ij} unchanged and reverses the kernel gradient. Thus the pair forces cancel in exact arithmetic, even when masses differ. Their accelerations need not match: each force is divided by that particle's own mass. The pressure term uses the gradient of the same poly6 kernel used in density. Replacing it with a different kernel gradient would require a new consistency argument.

For the viscous pair, the code corresponds to

$$b_{ij} = \frac{m_i m_j (\mu_i + \mu_j)}{2 \rho_i \rho_j} L_h(r), \quad F_{ij}^v = b_{ij} (v_j - v_i),$$

where the supported kernel Laplacian L_h is nonnegative. Positive densities, nonnegative viscosities and masses give $b_{ij} \geq 0$. Pair power is $-b_{ij} |v_i - v_j|^2$, but an explicit timestep still needs a size restriction. For one isolated coordinate with frozen $b = b_{ij}$,

$$E^+ - E = -\Delta t b \left[1 - \frac{\Delta t b}{2} \left(\frac{1}{m_i} + \frac{1}{m_j} \right) \right] (v_i - v_j)^2.$$

This is nonpositive when $\Delta t b (1/m_i + 1/m_j) \leq 2$, with positive masses and nonnegative timestep. For 1 kg masses, velocities ± 1 m/s and $b = 1$ kg/s, an interval of 1/4 s gives velocities $\pm 1/2$ m/s and energy 1/4 J instead of 1 J. An interval of 5/4 s gives energy 9/4 J. These exact teaching witnesses isolate viscous drag; they are not proposed timesteps for the complete native scene.

Gravity, injection, pressure and wall contact can change a full scene's energy. The liquid cohesion term is a symmetric attraction approximation for matching liquid materials. It does not calculate surface curvature or contact angle. Wall drag damps tangential contact motion, and wall heat exchange relaxes contact temperature. There is no bulk conduction field, vacuum cooling model, phase change or elastic slime model.

Source credit explains intermittent emission

Each end of the cylinder has its own accumulated volume credit. The source uses a nonuniform disk profile, a pressure-head speed and an approximate 0.43 profile-average factor for volume flux. If the current credit plus supplied volume is smaller than one reference parcel volume, no particle appears that step. The credit is retained rather than discarded.

Write C for previous credit, Q for prescribed volume flux, V_p for parcel volume and n for emitted parcel count. The update is

$$C^+ = C + Q \Delta t - n V_p, \quad n \rho_0 V_p + \rho_0 C^+ = \rho_0 (C + Q \Delta t).$$

For a hand calculation, measure volumes in units of V_p : credit 0.6 plus new supply 0.7 permits one parcel and leaves credit 0.3. Credit is future source volume, not mass already inside the box. This identity accounts for the prescribed source; it does not validate the approximate flux coefficient or prove how many parcels a floating implementation must select near an integer threshold. The visible cylinder is a source illustration, not a rotating collision solid.

Prepare, retry, then publish

step may shorten the requested interval using acoustic, advective, gravity, viscosity, cohesion and emitter-rotation limits. It prepares candidate particles, source credits, neighbor bins, density and forces. An additional acceleration or damping check may restart preparation at a shorter interval. At most 16 preparations occur in one call; unresolved stability returns an error.

The accepted state includes the particles, clock, generation, two source credits and deterministic disk-sequence state. A failure or cancellation must preserve them together. Candidate scratch may be overwritten. A successful commit publishes one coherent state, then the desktop worker captures its immutable preview. The report records the actual interval, particle count, emitted and total mass, and speed/pressure/temperature diagnostics. Its `max_pressure` comes from the force-evaluation densities and temperatures, before positions and contact temperatures are updated. In contrast, `max_speed` is measured after the accepted velocity/contact update. Those two maxima are not simultaneous samples of a freshly recomputed accepted pressure field.

The native tests in [material_contract.rs](#) exercise vacuum/fill mass, scaling, source accounting, cancellation, capacity failure, retry and thread-count replay. Unit fixtures beside the implementation exercise gravity, pressure against an independent compression-energy gradient, unlike-material inertia, viscous energy, cohesion and contact temperature. Their scope and numerical tolerances matter; a finite successful trajectory is not a continuum validation or a global stability theorem.

The particle thread-count fixture compares one source-free filled-scene step at 26 samples per axis using one and four workers. It compares time and parcel state directly. That is a bounded replay witness, not a claim that every material scene, compiler or hardware configuration replays identically.

[material_reference_contract.rs](#) adds a small independent route: an unordered all-pairs reference compares every parcel's full step for liquid and gas fixtures; a separate compression energy finite difference checks each pressure-acceleration coordinate, including unlike materials; and a first-emission witness checks actual disk sampling, midpoint rotation, prescribed head and retained credit. These fixtures avoid reusing the production linked-cell traversal. Their small scope makes errors easier to locate, rather than replacing larger scene tests.

Memory and display are different budgets

The numerical owner reserves accepted and candidate particles, thermodynamic and force scratch, linked neighbor indices and bin heads at construction. Its array budget uses actual compiled element sizes and capacities. It does not silently cap neighbor contributions or remove particles to keep a picture running. A particle or buffer limit is an explicit failure with accepted state retained.

Separately, the desktop splats accepted parcels into an eight-bit voxel preview. Surrounding-material parcels may participate in forces while being excluded from the visible emitted-material surface. The image therefore does not show all the simulated material or calibrated optical density. Changing the surface threshold or preview resolution

does not change particle sampling. The planned history can lower its stored resolution to retain every accepted step within its own budget. Renderer meshes, allocations and process RSS lie outside the payload estimate.

Three bounded experiments to read alongside the desktop

Experiment	Expected comparison	Measurable limit
Vacuum versus filled initial contents, source disabled	Initial parcel mass is zero versus density × volume	This checks initialization, not interface accuracy
More preview voxels at fixed physical samples and inputs	Stored image detail and payload change; numerical parcel spacing stays fixed	Voxel resolution cannot recover missing physical samples
More physical samples at matched physical time	Parcel count, neighbor work and accepted intervals can change	Compare mass, density/pressure and shape; do not infer convergence from a smoother image

Record material, domain size, source, gravity, accepted time, physical sampling, stored preview size and worker count with every comparison. A run’s settings are immutable; use a new run for changed physical parameters. Source injection and adaptive timesteps make equal step counts an unreliable matching rule.

Proof and execution boundaries

[MaterialPreview.lean](#) gives exact-real statements for the declared formulas. Representative names are `pressure_pair_coefficient_symmetric`, `pressure_pair_action_reaction`, `viscosity_pair_work_nonpositive`, `viscosity_pair_momentum`, `viscosity_pair_energy_nonincrease`, `nozzle_profile_bounds`, `nozzle_pressure_head_identity` and `reservoir_mass_ledger`.

The pressure-pair symmetry statements tolerate the declared positive density regularization. They do not prove that the pressure force is an energy derivative across that floor or the liquid pressure kink. The viscosity energy statement is for an isolated pair with frozen coefficients and its stated timestep restriction. Source identities do not certify gas discharge physics, the 0.43 approximation or floating parcel-count selection.

The independent native energy-gradient witnesses stay away from the active density floor, the liquid-pressure kink and the near-coincident pair exclusion at distances below $10^{-10}h$. Those hypotheses must remain visible when interpreting a successful finite-difference comparison.

Source statements are not a proof-run receipt. Publication must bind these modules to the current reviewed source inventory, pinned normal Lean build and allowed-axiom audit. Native execution tests separately connect the code to independent equations and exercise ownership failures. Neither evidence layer establishes a physically calibrated multiphase solver, full-scene convergence, IEEE equivalence to Lean or a performance guarantee.

CPU scheduling contracts and the Pantograph comparison

Rheon’s CPU scheduler changes the decomposition of a numerical kernel. It does not choose physical timesteps, reorder physics stages, own accepted fields, or schedule inference models. This note separates the implemented mechanism, its exact finite contracts, and observations that remain tests rather than proofs.

Actual ownership and execution order

Layer	Actual owner and decision	Required boundary
Accepted physical state	Simulation and its step/acceptance path	Scratch work may fail; publication still requires the existing acceptance gates.
Linear scratch work	parallel.rs and Rayon	Each mutable chunk is disjoint; the whole invocation joins before its consumer.
Batch feedback	scheduling.rs	One run owns per-stage timing metadata; no numerical buffers are stored here.
SGS triangular dependencies	sgs.rs	Each coordinate-sum wave joins before the dependent wave; reverse order for the backward sweep.
Scalar numerical reductions	pressure.rs	Products and compensated accumulation retain their existing serial order.
Worker display	observation.rs	Relaxed atomic markers are sampled diagnostics, not utilization estimates or a complete execution trace.

The scheduler’s composition roots bind the run owner around every scoped Rayon worker. A nested binding restores its predecessor on normal return and unwinding. `snapshot()` locks each stage independently; it is not an atomic snapshot of the entire simulation.

For n items and w workers, the linear policy proposes the caller’s preferred grain capped at $\text{ceil}(n/\max(w, 1))$, then approximately one, two and four batches per worker. Every proposal is clamped to 64–16,384 items. These are four *candidate indices*: clamping can produce duplicate batch sizes. A small grid or one-worker invocation can take the serial path and never enter this policy. Particle-density and particle-force kernels have a lower parallel threshold than grid vector kernels.

The feedback clock measures a complete joined kernel invocation, including scheduling overhead. It is not a sum of worker times. Under the assumptions below, the first eight successful calls try indices 0, 1, 2, 3, 0, 1, 2, 3; subsequent calls use the candidate with the smallest recorded trial time. The trial scores reset at phase 0 of every 128 successful calls. Changing the tracked (`item_count`, `worker_count`) shape resets exploration. The displayed smoothed cost is separate from these minimum trial scores. Neither statistic establishes an optimum or a repeatable speedup.

SGS calibrates independently: for an eligible multiworker grid it compares 8^3 and 4^3 tile sweeps in the order 8, 4, 4, 8, then runs the selected sweep. These scratch-only passes contribute real work and diagnostics but do not add physical timesteps or PCG iterations. The workspace retains the selected tile edge and last calibrated worker count. One-worker

or thin-grid fallback returns before calibration; passing through that fallback does not necessarily erase a previous multiworker choice.

Finite contracts and explicit assumptions

[CpuScheduling.lean](#) defines natural-number reference contracts, with seven definitions and fifteen public theorems:

Checked statement	Concrete interpretation	Boundary not proved
<code>batch_size_bounds</code>	Every clamped grain is positive and at most 16,384.	Rust integer conversion, allocation, performance.
<code>chunk_covers</code> , <code>chunk_unique</code>	For positive grain, each in-bounds index belongs to exactly one clipped half-open chunk.	Refinement of Rayon or Rust borrowing to the finite model.
<code>predecessor_rank</code> , <code>same_wave_has_no_dependency</code>	A read crossing one lower-axis tile boundary has rank exactly one smaller; same-rank tiles cannot have that relation.	Other stencils, diagonal couplings, or arbitrary task graphs.
<code>tile_wave_in_bounds</code> , <code>reverse_wave_dependency</code>	Valid rectangular tiles have bounded rank; descending waves reverse the dependency order.	Machine overflow and memory-model synchronization.
<code>trial_candidate_bounded</code> , <code>trial_first_round</code> , <code>trial_second_round</code> , <code>trial_periodic</code> , <code>no_trial_after_exploration</code>	The exact phase formula selects only indices 0–3, tries each twice in the first eight phases, and repeats after 128.	Four distinct grains or four equally good alternatives.
<code>candidate_revisited_within_period</code> , <code>successful_trace_count</code> , <code>successful_trace_explores</code>	An eligible candidate is selected within fewer than 128 further successful-completion positions in the stated trace.	Wall-clock progress, OS/Rayon fairness, or progress when calls keep failing.

For example, ten indices at grain four occupy $[0, 4)$, $[4, 8)$, and $[8, 10)$. A $2 \times 2 \times 1$ tile grid has waves $\{(0, 0, 0)\}$, $\{(1, 0, 0), (0, 1, 0)\}$, and $\{(1, 1, 0)\}$. The middle two tiles have no nearest-axis triangular dependency on each other. This example is about ready work, not an independence claim for arbitrary physics operations.

The successful-trace premise has operational content: one outstanding invocation of the same stage; stable item count, worker count and preferred grain; a nonempty workload; and exactly one record after each successful joined call. The natural-number model has no machine-counter overflow. A delay counts successful completions before selection of the next invocation, never seconds.

Those premises cannot be removed. Two overlapping `BatchRuns` can both observe phase 0; recording both moves the next call to phase 2. The new adversarial test executes this counterexample. A dropped or failed pass does not call `complete`, so repeated failure can retain one phase indefinitely. A whole timestep may later be rejected even though some earlier kernels completed and supplied valid kernel timings. Batch exploration is not an accepted-timestep counter.

Other tuning limits follow from the actual metadata key. Preferred grain is not part of the tracked shape, so a changed preference can reuse an index scored under an earlier proposal. Overlapping shape changes $A \rightarrow B \rightarrow A$ can also make an old completion look current because there is no plan-generation identity. These are limits on interpreting the timing

policy; the scheduler owns no fields that could become a second accepted-state authority. This work changes no production policy and makes no claim that these cases are fair or performance-optimal.

For SGS, preserving the per-cell X/Y/Z summation order is as important as the topological order. Relaxed atomic bit loads/stores avoid aliased mutable f64 storage; they do not perform floating-point accumulation. Visibility relies on the joined task execution between waves. The Lean rank statements do not prove the Rust/Rayon memory model. Likewise, swapping the current serial compensated reductions for unordered parallel reductions would need a new numerical contract; the scheduling theorems would not justify it.

Rayon documents disjoint mutable chunk iteration and early termination on fallible parallel work. If several tasks fail, its returned error is not specified to be the lexicographically earliest one. Rheon’s bitwise replay checks for successful computations do not establish deterministic error choice. See the pinned [Rayon 1.11 slice API](#) and [fallible iterator contract](#).

Pantograph: inference/workflow scheduling is a different layer

Here Pantograph means the user’s [MrScripty/Pantograph repository](#), reviewed at commit 1fba975ab20cb032949aa7221e373f0719e3264b on 2026-10-12. The primary sources are its authored architecture and actual policy code, not a performance paper. It is unrelated to the similarly named Lean proof-search interface in arXiv:2410.16429.

Pantograph’s scheduler owns workflow-task readiness, resource admission, compatible batching, runtime/device selection and lifecycle policy. Its architecture distinguishes these decisions from runtime launch, model storage, and frontend presentation. The document also labels a transition toward durable task execution; that target architecture alone is not evidence that every path is already implemented. [Primary architecture](#).

The reviewed readiness function defers a task when dependency evidence is absent. Its completion-ranking policy compares bounded eligible offers using supplied preparation, transfer and execution costs. It distinguishes measured from synthetic evidence and checks comparison conditions, snapshot association, sample count and age. Selection remains advisory until ordinary admission and dispatch validation; the pure ranking function does not acquire a runtime lease. [Readiness code](#), [completion-ranking code](#).

The separate frozen-cohort evaluator explores two legal serialized completion events and then uses a deterministic greedy continuation. It has explicit work bounds and refuses an incomplete comparison when required evidence is missing, invalid or over budget. Its source explicitly distinguishes bounded work units from a wall-time guarantee. This is not an unrestricted global-optimum or starvation-freedom theorem. [Bounded cohort code](#).

Pantograph concern	Rheon mapping	Limit of the analogy
Evidence before task admission	Complete predecessor waves before consuming their values.	Runtime readiness evidence is not an SGS memory barrier.
Comparable completion-cost evidence	Measure the entire joined kernel pass, not worker busyness.	Rheon’s four trial minima do not implement Pantograph’s resource/residency model.
Explicit bounded alternatives and refusal	Four candidate indices, fixed exploration phases; failed passes do not train.	Rheon does not refuse all scheduling when timing is absent; it uses a hardware/workload proposal.

Pantograph concern	Rheon mapping	Limit of the analogy
State/owner identity across dispatch	Keep numerical state ownership outside timing metadata.	Adding queued workflow tasks or model residency to a fluid step would be new architecture.

Engineering inference: Pantograph’s explicit comparison identities are useful when considering a future Rheon feedback key that includes preferred grain and a generation for overlapping plans. That is a design suggestion, not an implemented change or measured improvement. Neither codebase’s scheduling policy proves Rheon’s timestep stability, conservation, pressure convergence, continuum accuracy, or CPU utilization.

Focused qualification

The new module compiled directly with pinned Lean 4.19.0 (6caae842e9495688c1567e78c0e68dbb96942aa). Printing the axioms of all 22 new declarations reported only the repository’s existing allowed axioms: `propext`, `Classical.choice`, and `Quot.sound`. This standalone check is distinct from the root import, reviewed inventory and complete project audit, which own the integrated qualification.

Six new internal Rust tests exercise the private scheduling decisions without exposing a new public API:

- 256 successful nonoverlapping calls cover two full tuning periods.
- A real failed parallel pass leaves no successful sample; retry visits every index once, including the partial final chunk.
- Nested scheduler scopes restore the correct owner after unwinding.
- Two overlapping invocations witness the missing nonoverlap premise.
- All 125 rectangular tile shapes with axes 1–5 satisfy both sweep directions’ cross-tile dependency checks, with completion marked only after a whole wave.
- Tile edges 1/4/8/16 on one/four threads overwrite NaN-poisoned scratch, produce identical output bits, and satisfy an independently reconstructed triangular residual within the stated test tolerance on the anisotropic fixture.

The focused command below passed ten selected tests (six new and four existing) with Rust 1.92.0, no default features, and `parallel`. This is a contract test, not a timing benchmark. Existing [parallel_contract.rs](#) contains broader accepted-state, diagnostic, cancellation and replay checks; those are not claimed as newly executed by this focused run.

```
cargo test --locked --no-default-features --features parallel --lib scheduling -- --test-threads=1
lean proofs/Rheon/CpuScheduling.lean
```

Production scheduler and solver behavior are unchanged. No speedup, CPU fairness, strict latency bound, race-detector qualification or Rust-to-Lean refinement is claimed. Historical timing observations remain tied to their original source and machine in the [scheduling evidence](#).

CAD and engineering visualization contracts

Research integration for Rheon · 12 October 2026

[Chapter 14](#) is the short reading route. This guide retains the full algorithms, assumptions and proposed lesson designs. The native material inventory below is anchored to the stated main commit; it does not erase the separate browser carrier volume renderer or its instantaneous velocity curves. The chapter's existing sticky 24³ carrier example is not a material/SPH or CAD solver. **The seven lessons in Section 15 are specifications, not seven newly implemented interfaces.**

[Annotated bibliography](#) · [44 primary sources and 21 code anchors](#) · [implementation order](#) · [lesson specifications](#) · [analytic fixture generator](#)

Reader's route

Begin with Sections 1–3 if you know fluid animation but are new to CFD. Sections 4–10 explain what each view measures and how to implement it. Sections 11–14 cover scientific honesty, exchange formats, and a practical architecture. Section 15 contains seven interactive lessons with exact expected results. Section 16 is the release checklist. The linked implementation plan gives a short delivery order, and the annotated bibliography explains what each primary source actually establishes. CAD01–CAD44 are local source IDs, distinct from the book's existing citation labels.

This chapter distinguishes verified implementation inventory from proposed extensions. Rheon's current inventory is anchored to commit 2209012bc5f902693d0f695ec1c2d854126af317, as reviewed on 12 October 2026; [the particle implementation](#) and the other commit-anchored links below provide the implementation citations. Prior discussion supplies the research question, not independent repository evidence. General formulas and visualization methods below do not establish that a particular Rheon mode currently supports them.

1. Start with the engineering question

A fluid image and an engineering result answer different questions. A volume image might reveal the location of a transported marker. An engineer may need to know whether a channel carries the requested flow, where a part experiences pressure loading, how much pumping power a constriction requires, or whether a result changes under refinement. All of those require quantities with definitions, coordinates, units, and repeatable measurements.

An effective first engineering workspace can be modest:

1. A geometry view showing the actual computational domain and named boundaries.
2. A movable scalar slice, initially colored by speed.
3. Optional velocity arrows, with a visible scale rule.
4. A pressure slice or pressure profile with an explicit reference convention.
5. A section measurement and a point probe.
6. A time-history chart and conservation diagnostics.
7. Export that allows the same fields to be inspected independently.

These features make a small CPU reference solver useful as a laboratory. They do not require a photorealistic liquid renderer, a general CAD kernel, or a promise of industrial certification. They also do not make an incomplete model complete. For example, thermal visualization requires a computed temperature field and energy model; a visually reconstructed interface does not supply surface tension or liquid–air coupling.

ParaView offers the established reference workflow: scalar slices, contours, glyphs, vector integration, sampling, and charts belong to a processing pipeline whose output is separate from presentation. Use it to check a first exporter and to learn which outputs are useful before rebuilding every operation. [CAD01](#)

What “accurate” needs to mean

There are at least four questions:

- **Implementation verification:** does the code implement the stated mathematical method correctly?
- **Solution verification:** is this particular run sufficiently converged in iteration, space, and time for the quantity of interest?
- **Physical validation:** does the model predict relevant experimental observations, within measured uncertainty, for the intended class of problems?
- **Visualization verification:** do the displayed and exported values faithfully represent the accepted solver state?

NASA’s verification guidance distinguishes code checks and calculation checks; its validation guidance restricts validation claims to supported applications and experimental comparisons. These are useful distinctions for product wording as well as technical practice. [CAD02](#), [CAD03](#)

An analytical Poiseuille profile verifies a laminar-flow calculation. It does not validate every imported valve, a turbulent wake, or a free-surface impact. A low linear-system residual is evidence about that algebraic solve, not a percentage error on drag. A convincing picture provides neither kind of evidence on its own.

Rheon’s present starting point: two distinct data paths

The reviewed implementation has a new bounded **reference-volume particle/SPH preview** and a separate **staggered carrier projection/transport path**. Do not combine their fields as if they describe one coupled state.

The particle preview uses a poly6 density/pressure neighborhood calculation, clamped liquid compression pressure, ideal-gas rho-R-T pressure, symmetric viscosity, same-material liquid cohesion, symplectic Euler, and inelastic box-wall contact. Its public particle view includes position in meters, velocity in m/s, mass in kg, temperature in K, and material ID. Per-particle density, pressure, acceleration, and damping data are presently private. Public report pressure maxima are evaluated before motion and wall-temperature updates, while reported speed refers to the accepted state. Therefore even an apparently simple pressure-versus-speed comparison needs evaluation-stage metadata or an additional synchronized diagnostic evaluation. See [public accessors](#), [pressure assembly](#), [motion/contact](#), and [report assembly](#).

The rotating emitter is a prescribed source with pressure-head-based speed, midpoint orientation, fixed-volume parcels, and transactional volume credits. See [axis orientation](#), [prescribed profile](#), [particle admission](#), and [accepted publication](#). The particle source evaluates orientation at the interval midpoint and its own disk profile; the separate carrier source applies its declared forcing profile. It is not a collision solid or a simulated nozzle wall. The liquid

compression response and gas absolute-pressure response do not yet establish one common ambient gauge or a CAD wall-pressure map. Particle temperature exists, but that alone is not evidence of a resolved heat-transfer or conjugate-thermal solver. See [material equations of state](#).

The separate carrier path has face velocities and a pressure correction on a fixed box. Its pressure correction must not be relabeled as particle pressure, and its native face flux must not be reported as SPH flux. Existing carrier output includes opacity images and step/run records; dimensional flux/wall-gradient export support still needs checking. See [declared carrier exports](#) and [their implementation](#).

The native material UI at the inventory commit uses eframe/egui Glow with CPU-projected, depth-sorted voxel faces. Particles are quantized into an appearance view, with ambient material hidden. It has no verified CAD/BREP/NURBS pipeline, liquid reconstructed surface, volume ray marcher, or wall-gradient view. Its immutable history stores preview bytes, report, time, and step, rather than full particle checkpoints. There is no verified public particle CSV, mesh, or surface export in the material UI; that is an inventory finding rather than an explicit prohibition in its API. See [desktop architecture](#), [preview payload](#), [appearance filtering](#), [history frames](#), and [projection/depth sorting](#).

Immediate consequence: first expose quantitative particle inspection and a synchronized, unquantized snapshot/export contract. Speed/material/temperature particle coloring and inventory/source diagnostics are available earlier than pressure slices, complete particle histories, or wall loads. Grid-based methods remain useful for the carrier branch and as independent teaching fixtures, but they are not a shortcut around the particle data prerequisites.

2. What is actually stored?

2.1 The field dictionary

The following is a proposed field dictionary, not a claim that every field is available.

Quantity	Meaning and SI units	First useful view	Important qualification
Velocity $\mathbf{u}$	Fluid velocity, m/s	Speed slice and arrows	Record lab/body frame; components may be staggered
Speed	$ \mathbf{u} $, m/s	Sequential scalar map	Magnitude after vector interpolation differs from interpolated magnitude
Physical pressure p	Normal isotropic stress variable, Pa	Slice, profile, surface load	State gauge and whether hydrostatic terms are included
Kinematic pressure π	p/ρ , m^2/s^2 , for the stated reference density	Slice with its real units	Do not label as Pa before conversion
Mass density ρ	Mass per physical fluid volume, kg/m^3	Slice if variable	Constant density can conceal all interesting motion
Liquid fraction α	Fraction occupied by a phase, dimensionless	Cell slice; interface display	Specify denominator: full cell or available fluid volume
Tracer c	Passive marker or species variable	Slice, outlet history	Dimensionless appearance marker is not automatically mass fraction
Solid/open fraction β	Available volume or face opening, dimensionless	Geometry diagnostic	Distinct from liquid fraction

Quantity	Meaning and SI units	First useful view	Important qualification
Vorticity ω	$\nabla \times \mathbf{u}$, s^{-1}	Magnitude slice or component map	Nonzero in simple shear; not a turbulence detector
Divergence d	$\nabla \cdot \mathbf{u}$, s^{-1}	Diagnostic slice and norms	Use the solver's discrete flux divergence for conservation checks
Q , λ_2	Velocity-gradient invariants, s^{-2}	Thresholded isosurfaces	Definition, threshold, and gradient scheme matter
Wall shear τ_w	Tangential traction, Pa	Wall map	Requires a valid viscous/wall model and wall derivative
Temperature T	Thermodynamic temperature, K	Thermal slice	Requires an energy equation and thermal boundary conditions

Pressure naming is particularly treacherous. OpenFOAM explicitly distinguishes kinematic and physical pressure, and supports adding or subtracting hydrostatic contributions. Treat that as a reason to carry units and semantics, not as a reason to copy another solver's variable names blindly. [CAD04](#)

Do not use “density” as a collective label for mass density, fraction, tracer, and opacity. Volume-of-fluid methods represent occupancy through a fractional volume; the historical VOF papers describe that geometric role separately from a complete physical model. [CAD05](#)

2.2 Four states a field can have

A field selector needs more than a list of array names. Distinguish:

1. **Native:** directly stored by the accepted simulation state.
2. **Derived:** computed from native data with an identified operation.
3. **Prescribed:** supplied by a boundary condition or analytic example, not predicted.
4. **Unavailable:** not produced or not meaningful in this model.

A pressure array full of zeros in a transport-only mode is not a measured zero-pressure solution. An unsupported wall-shear field should say “not available for this model,” rather than silently show blue everywhere. A stale derived field should not survive a timestep or geometry change under its old label.

2.3 A minimal data contract

Every exported snapshot should answer these questions without reading its source code:

- What solver mode, version/commit, configuration, geometry, and run produced it?
- What accepted simulation time and timestep does it represent?
- Which coordinate system, origin, orientation, and length unit are used?
- Where does each component live: cell center, node, face, boundary quadrature point, or particle?
- Does an array include halos, inactive cells, air cells, or other nonphysical values?
- Which material and boundary models are active?
- What do pressure zero, connected-component IDs, and phase fractions mean?
- What diagnostic tolerances were used, and did this state pass its acceptance checks?

Recommended field metadata includes a stable ID, human label, unit dimensions, components, association, spatial offset, validity mask, native/derived/prescribed status, reference frame, value semantics, and source revision. Derived metadata should identify input fields, sampling/gradient method, parameters, and a reproducible cache key.

An analysis result additionally needs the measurement geometry, orientation, weighting convention, normalizing scales, selected boundary IDs, numerical precision, invalid-sample count, and integration method. Save these definitions with the run. A screenshot alone cannot reproduce a pressure-drop value.

3. Sampling without lying about the grid

Rheon’s particle preview should use the particle-specific reconstruction in Section 3.5. Sections 3.1–3.4 describe the separate staggered carrier route and generic structured-grid analysis. A regular visualization lattice reconstructed from particles is a derived display grid, not evidence that the particle solver has native face fluxes or a pressure-projection lattice.

3.1 Cell centers and staggered velocities

In a common uniform MAC arrangement, pressure lies at cell centers and velocity components lie at faces. For a domain origin $\mathbf{o}$, spacings (h_x, h_y, h_z) , and cell indices (i, j, k) :

- Cell center: $\mathbf{o} + ((i+\frac{1}{2})h_x, (j+\frac{1}{2})h_y, (k+\frac{1}{2})h_z)$
- x-face speed u : $\mathbf{o} + (i h_x, (j+\frac{1}{2})h_y, (k+\frac{1}{2})h_z)$
- y-face speed v : $\mathbf{o} + ((i+\frac{1}{2})h_x, j h_y, (k+\frac{1}{2})h_z)$
- z-face speed w : $\mathbf{o} + ((i+\frac{1}{2})h_x, (j+\frac{1}{2})h_y, k h_z)$

Their logical dimensions differ: u has (N_x+1, N_y, N_z) , v has (N_x, N_y+1, N_z) , and w has (N_x, N_y, N_z+1) . The MAC layout is designed around discrete pressure gradients and flux divergence; it is not a set of colocated vector samples. Bridson and Müller-Fischer’s course notes provide a practical explanation of that arrangement. [CAD06](#)

For interior visualization, a cell-centered vector can be reconstructed as:

$$u_c(i, j, k) = \frac{1}{2}[u(i, j, k) + u(i+1, j, k)]$$

with analogous formulas for v and w . This average is suitable only under the stated uniform/interior assumptions. Cut cells, irregular faces, and boundary conditions require a reconstruction consistent with their geometry. Label the output “reconstructed cell-centered velocity.” Preserve the face arrays separately whenever they are needed for conservation-sensitive analysis.

3.2 The sampler is shared infrastructure

Implement one tested sampler for slices, probes, seed integration, and export previews. For an array with location offset δ in grid units, transform a physical point into its array coordinates:

$$\mathbf{q} = (\mathbf{x} - \mathbf{o}) / \mathbf{h} - \delta.$$

For each axis, take the floor index and fractional coordinate. Trilinear interpolation combines the eight valid neighboring samples using the product of linear weights. Sample each staggered component at its own offset. This avoids an extra center-resampling stage and retains a clearer relationship to the native representation.

The sampler should return a value plus validity/provenance, not merely a float. Possible failures include outside the domain, inside a solid, absent timestep, unavailable field, and insufficient boundary stencil. Never convert these failures into zero. Handle periodic wrapping deliberately and separately from wall clamping. Clamp only when that behavior is the stated numerical extension; blind clamping creates plausible but fabricated near-wall values.

For a speed display, choose and record whether to compute $|\text{interpolate}(\mathbf{u})|$ or $\text{interpolate}(|\mathbf{u}|)$. They are different operations. Opposing vectors can average to zero while their magnitudes remain positive. The first is normally the appropriate speed of the reconstructed local velocity; the second describes a smoothed sampled-speed scalar.

3.3 Cell-to-point conversion is a transformation

VTK’s cell-to-point filter averages contributing cell data onto mesh points. That is a useful display operation, but it changes the represented field. [CAD07](#) On an obstacle boundary, ordinary averaging may blend values across regions that should not communicate. Keep a piecewise-constant “native cells” view beside a smooth reconstructed view for diagnosis.

Use native face fluxes for flux totals. Use native cell-volume weights for integral quantities. Use raw valid data, not a display-decimated volume, for min/max reports. The renderer may upload Float32 while the CPU retains Float64 measurements, but this conversion must be explicit and tested.

3.4 A practical sampler test suite

- Constant fields remain constant at every valid sample.
- Linear fields interpolate exactly in the interior of a uniform grid, up to floating-point error.
- Each staggered component samples correctly under anisotropic spacing and a nonzero origin.
- Outside/solid samples are invalid, not zero or extrapolated unless specified.
- Sampling does not bridge two disconnected fluid regions.
- A pressure gauge shift changes displayed p but preserves valid within-component pressure differences.
- UI probes, exported probes, and headless probes return the same values and validity states.

3.5 Reconstructing a field from SPH particles

Start by displaying and probing actual particles. A point cloud colored by accepted-state speed or material is more honest than inventing a smooth field before its sampling rules exist. Give selected particles stable IDs, show their unquantized positions and velocities, and distinguish hidden ambient material from material absent in the simulation. A filtered-out particle must not disappear from a physical inventory merely because it is invisible.

A common SPH density estimate is $\rho(\mathbf{x}) = \sum_j m_j W(\mathbf{x} - \mathbf{x}_j, h)$. A common volume-weighted scalar interpolant is $A(\mathbf{x}) \approx \sum_j (m_j / \rho_j) A_j W(\mathbf{x} - \mathbf{x}_j, h)$. A normalized display interpolant divides that sum by $\sum_j (m_j / \rho_j) W$. These are different estimators: the normalized version can preserve a constant sampled attribute but does not by itself provide a conservative mass field or a consistent gradient. Price’s SPH review explains the distinction between interpolation and the discretization of the physical equations. [CAD41](#)

For a conventional three-dimensional poly6 kernel with support radius h ,

$W(\mathbf{r}, h) = 315 / (64\pi h^9) (h^2 - r^2)^3$ for $0 \leq r \leq h$, and zero otherwise.

The kernel has units m^{-3} . This formula is a reference from the particle-fluid literature, not a substitute for inspecting Rheon's exact implementation, radius convention, self contribution, material selection, and normalization. Müller, Charypar, and Gross describe these particle-fluid kernel ideas and a related interactive formulation. [CAD42](#)

For Rheon's first scalar slice, the required new data include each particle's synchronized density and pressure, kernel/support metadata, material rules, and a valid support/coverage indicator. Evaluate the interpolant at slice samples using a spatial hash or cell-linked list. The bin width and neighbor stencil must cover the entire support; boundary particles must not be missed because they lie in adjacent bins. Brute-force sums over tiny fixtures provide an independent reference for a neighbor-accelerated implementation.

Reject nonfinite/nonpositive density where the interpolant divides by density, and report a zero or insufficient support denominator as invalid rather than silently dividing by a convenient epsilon. A spatial index changes candidate search cost, not the kernel definition: it must return the same contributing neighbors and weights as the reference calculation. For a bounded small preview, a CPU-generated slice may be sufficient before any GPU renderer change.

Separate the following fields in both API and UI:

- Native particle density at the solver's evaluation stage.
- Native particle pressure and its material-specific equation-of-state semantics.
- Reconstructed mass density on a display lattice.
- Reconstructed velocity/temperature/pressure attributes, with their weighting rule.
- Kernel support/normalization measure and neighbor count.
- Particle count or occupancy used for appearance.

They can look similar and have very different meanings. In particular, a particle-count image is not mass density when particle masses or parcel volumes differ. A pressure maximum computed before motion should not be placed at a particle's post-motion position without disclosure. Prefer a pure diagnostic re-evaluation at accepted positions/temperatures, with no effect on the accepted simulation, or keep the original stage's positions and arrays together in a distinct snapshot.

Near a free surface or solid boundary, kernel support can be incomplete. Normalizing a display field may hide that deficiency without fixing the physics. Boundary treatments such as Ghost SPH explicitly address this numerical issue; they are additional methods, not effects obtained by drawing a smoother surface. [CAD43](#) Do not average across an impermeable wall or blend different material pressure conventions by default. Use material/region masks and explain samples whose support crosses a phase boundary.

For particle-derived vorticity or Q , first decide whether to use a native SPH gradient, a corrected particle gradient, or differences of a reconstructed grid. These have different errors. A coarse display grid adds kernel reconstruction, resampling, and finite-difference errors before the invariant calculation. Label the method and record h , output-grid spacing, neighborhood counts, correction policy, and invalid regions. Do not apply the MAC derivative recipe to a particle occupancy image.

Particle-based measurements need their own definitions. Mass inventory is $\sum m_j$ over the selected material/region. A sum of reference parcel volumes is a reference-volume inventory; $\sum m_j / \rho_j$ is a density-based volume estimate and need not equal it. Report these separately. The emitter's transactional volume credits can be checked against admitted parcel volume and pending remainder, but a credit balance is not a measured downstream flow rate.

An approximate Eulerian section flux can be obtained by integrating reconstructed $\rho \cdot u$ over a plane. A different estimate counts signed particle mass crossings over a time interval and divides by its duration. Crossing counts require stable particle IDs, complete positions/times, births/deletions, and sufficiently resolved trajectories; multiple crossings between sparse snapshots can be missed. Label the two methods and compare them rather than call either a native finite-volume face flux.

For box-wall reaction, logging the actual contact momentum change can produce an interval-averaged contact-force diagnostic: $\text{reaction on wall} = -\Sigma \Delta(\text{momentum}_{\text{particle,contact}})/\Delta t$. That is a contact-impulse quantity, not a resolved hydrodynamic wall-shear or pressure distribution. Log pressure/viscosity/body/source/contact contributions separately where the integrator allows it. Excluding emitter momentum or counting gravity as wall contact can make a superficially plausible force balance wrong.

Finally, SPH resolution studies need particle spacing, smoothing support/neighbor behavior, timestep, and boundary treatment to be specified together. Merely making an isosurface finer, or increasing neighbor count at fixed particles, is not a general convergence guarantee. Dehnen and Aly analyze particle disorder, kernel choice, neighbor counts, and pairing stability in SPH; use their findings as a warning against simplistic resolution sliders, not as proof of Rheon's accuracy. [CAD44](#)

4. Scalar slices, contours, and clipping

4.1 A slice is a measurement surface

Start with XY, XZ, and YZ slices. Add an arbitrary plane when basic indexing and masking are correct. An arbitrary plane can be defined by a point $\mathbf{x}_0$ and an orthonormal basis $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{n})$. Sample at

$$\mathbf{x}(a,b) = \mathbf{x}_0 + a\mathbf{e}_1 + b\mathbf{e}_2.$$

The resolution of this display grid is independent of the simulation resolution. A 2048-pixel slice through a 32-cell domain does not reveal 2048 independent samples. Offer “show source cells” and retain a tooltip showing the physical coordinate, interpolated value, and contributing cell/location.

Fast implementation: upload a scalar 3D texture and sample it on a plane in a shader, with explicit coordinate transforms and an appropriate validity mask. **Reference implementation:** evaluate the same plane on the CPU using the shared sampler. Compare their sampled values before judging image similarity. For very small CPU runs, producing a 2D texture on demand can be simpler and sufficiently fast.

An unlit slice is a good quantitative default. Lighting modulates the very brightness used to read values. If geometry shading is helpful, use it on the surrounding solid, keep the quantitative plane unlit, and make any alternative shading switch obvious.

4.2 Slice versus clip

A slice extracts a lower-dimensional cross-section. A clip removes part of a dataset and exposes what remains. They answer different questions and have different memory costs. ParaView's filtering guide explicitly distinguishes them and warns that some volume-subsetting operations convert compact structured grids into larger unstructured representations. [CAD01](#)

In a native viewer, visual clipping can initially be a shader-plane test rather than a new volumetric mesh. Do not use the clipped display's triangle count or silhouette as a geometric measurement unless actual intersection geometry has been generated and validated.

4.3 Contour lines and isosurfaces

A contour line on a 2D slice satisfies $f(x,y)=c$. An isosurface in 3D satisfies $f(x,y,z)=c$. They are level sets of a chosen scalar, not automatically material boundaries. A pressure isosurface is not a solid wall; a Q isosurface is not a material tube; an $\alpha=0.5$ surface is a display interpretation of a phase-fraction field.

Marching-cubes-style extraction classifies a grid cell's corner values relative to the level, intersects crossing edges, and emits triangles. Interpolation on an edge from values f_0 to f_1 uses $t=(c-f_0)/(f_1-f_0)$. Specify tie handling when a value equals the level, degenerate cases, consistent shared-edge indexing, and a topology policy for ambiguous configurations. The source samples must be meaningfully located at the extraction lattice vertices; cell-centered data needs a declared reconstruction or an equivalent shifted lattice.

For a mature CPU implementation, VTK's Flying Edges is a useful reference. Its edge-oriented multi-pass structure counts output, allocates once, and avoids repeated edge intersections; its documentation also warns that degenerate triangles can occur. Do not promise published speedups on Rheon's data without benchmarking them. [CAD08](#), [CAD09](#)

4.4 Liquid surfaces deserve a separate label

If the numerical method has a native geometric interface reconstruction, exporting it can preserve more of the method's meaning than contouring a smoothed fraction. If it has only fractions, an isosurface is still useful, provided it is labeled as a display surface and its threshold is shown.

Compute liquid volume from native fractions and their documented volume denominator. Do not infer it from the volume enclosed by a display surface. For example, if α is the fraction of the available fluid portion of a cut cell, $V_{\text{liquid}} = \sum \alpha_i \beta_i V_{\text{cell},i}$. If α already measures liquid relative to the entire cell, multiplying by β again is wrong. This is a data-contract decision, not a rendering preference.

Smoothing, closing holes, decimation, and normal smoothing can improve appearance while changing shape. Preserve the unmodified extracted mesh for measurement, and describe processed meshes as display derivatives. A reconstructed free surface cannot compensate for absent gas pressure, surface tension, contact-angle, gravity, or interface-advection physics.

5. Direction: arrows, streamlines, pathlines, and streaklines

5.1 Velocity glyphs

Place arrows on a sparse lattice, a slice, or selected points. Use stable deterministic sampling so arrows do not flicker randomly as time changes. Store whether length is proportional to speed, normalized, capped, or logarithmic. A normalized arrow plot is a direction plot, not a speed plot. If arrows use fixed length, color by speed and make that convention visible.

On a slice, distinguish the full 3D velocity from its in-plane projection:

$$\mathbf{u}_{\text{parallel}} = \mathbf{u} - (\mathbf{u} \cdot \mathbf{n})\mathbf{n}.$$

A nearly zero projected arrow may indicate strong out-of-plane flow. Show the normal component separately or offer full 3D glyphs. A streamline of the projected slice field is not necessarily the intersection of a 3D streamline with that slice.

Glyph instancing is preferable to creating a separate object for every arrow. Display limits should be configurable, but decimation should not affect numerical statistics. At zero speed, omit orientation or show a neutral marker; normalizing a zero vector creates NaNs and arbitrary directions.

5.2 Three curves, three meanings

The classical distinctions are essential in unsteady flow. MIT’s fluid-mechanics notes and NASA’s unsteady-visualization work explicitly separate them. [CAD10](#), [CAD11](#)

- **Streamline:** a curve tangent to $\mathbf{u}(\mathbf{x}, t_0)$ with time frozen at one instant. An arc-length parameterization is $d\mathbf{x}/ds = \mathbf{u}/|\mathbf{u}|$.
- **Pathline:** the trajectory of one ideal passive particle through changing fields, $d\mathbf{X}/dt = \mathbf{u}(\mathbf{X}, t)$, from a specified release time and point.
- **Streakline:** at observation time t , the locus of particles released from a specified source at different earlier times.

For a steady velocity field these curves can coincide. Recomputing streamlines for successive snapshots and playing them as a movie still does not make them particle histories. Likewise, a “travel time” computed along a frozen streamline is a frozen-field estimate, not a prediction through an unsteady run.

5.3 A trustworthy streamline integrator

For a small CPU reference implementation, begin with RK2 midpoint integration and then add RK4 or adaptive RK45 if needed. Keep a testable fixed-step implementation as a baseline. VTK’s stream tracer offers these integrator choices and records termination reasons; it uses arc-length-oriented stepping and configurable limits. [CAD12](#)

An arc-length RK2 step is conceptually:

1. Sample a valid velocity at $\mathbf{x}$ and reject speeds below ϵ_{speed} .
2. Let $\mathbf{k}_1 = \mathbf{u}(\mathbf{x})/|\mathbf{u}(\mathbf{x})|$.
3. Sample at $\mathbf{x} + \frac{1}{2}h \mathbf{k}_1$; normalize to $\mathbf{k}_2$.
4. Propose $\mathbf{x}_{\text{new}} = \mathbf{x} + h \mathbf{k}_2$.
5. Check the segment against fluid-domain boundaries, not only its endpoint.
6. Append a point and continue, subject to length, step, error, and validity limits.

Set h relative to local cell dimensions and geometry features, not camera zoom. A step that starts and ends in fluid can still cross a thin solid wall. A robust implementation uses segment traversal/intersection or a conservative step bound, and rejects or shortens a crossing step. Do not reflect a particle off a wall unless that is the intended model.

Store termination reasons such as left domain, hit wall, stagnant, invalid stencil, maximum arc length, maximum steps, or tolerance failure. A truncated curve with an unexplained endpoint can be misread as flow stagnation.

For adaptive methods, tighter integration tolerance cannot repair a coarse or inconsistent interpolated velocity field. Refine both the streamline integration and the underlying field separately. Record seed coordinates, direction, field snapshot, integrator, step units, tolerance, and termination limits with a saved view.

5.4 Time-dependent integration

Pathlines require at least two bracketing snapshots for simple time interpolation, actual timestamps, and consistent geometry/frame conventions. A basic interpolation is

$$\mathbf{u}(\mathbf{x},t)=(1-\theta)\mathbf{u}(\mathbf{x},t_n)+\theta\mathbf{u}(\mathbf{x},t_{n+1}), \theta=(t-t_n)/(t_{n+1}-t_n).$$

Use enough snapshots to resolve relevant motion; temporal interpolation cannot recover events that were never saved. Do not continue past a missing time interval by pretending the field is steady. If mesh geometry changes, transform or locate samples on the appropriate geometry at each time. VTK's particle tracer explicitly handles unsteady vector fields and distinguishes mesh-over-time behaviors. [CAD13](#)

For streaklines, release particles at defined physical-time intervals and integrate each to the common observation time. Seed density represents your visualization choice unless weighted injection rates are explicitly modeled. Ideal tracer paths neglect inertia, buoyancy, finite-size collisions, and diffusion. They should not be labeled droplets, sand, or bubbles without the necessary dynamics.

5.5 Dense direction textures

Line integral convolution (LIC) can reveal a dense 2D direction pattern by filtering noise along the vector field. It is a useful optional slice mode when arrows become cluttered. It does not inherently show direction sign or speed, and an in-plane LIC inherits the projection caveat. Cabral and Leedom's original method is the primary reference. [CAD14](#) Keep LIC after the simpler, testable arrow and streamline tools; it adds visual density rather than a new engineering measurement.

6. Pressure, flow rate, and losses

6.1 Pressure needs a reference

In an incompressible pressure solve, an additive constant may be undetermined. The implementation usually fixes a reference pressure, constrains a mean, or anchors a degree of freedom. If multiple sealed disconnected regions have independent gauges, subtracting their pressures does not yield a physical pressure difference without an additional convention.

Show pressure kind, units, reference, connected component, and any hydrostatic transformation near the legend or in the field details. Negative gauge pressure is not by itself a vacuum or cavitation prediction. Cavitation needs an appropriate absolute-pressure interpretation and phase-change model. A projection multiplier may also differ from the pressure one intends to use for mechanical loads; verify its temporal and physical meaning in the solver.

For constant density, physical pressure p and kinematic pressure π are related by $p=\rho\pi$. If the solver nondimensionalizes, first use its complete reference scales. Multiplying a dimensionless multiplier by a convenient density is insufficient. OpenFOAM's documented pressure conversions are useful examples of explicit rather than implicit semantics. [CAD04](#)

6.2 Section flux

For a stationary oriented section A:

$$Q = \int_A \mathbf{u} \cdot \mathbf{n} \, dA, \text{ with units } \text{m}^3/\text{s}.$$

Mass flow is $\dot{m} = \int_A \rho \mathbf{u} \cdot \mathbf{n} \, dA$. Only for a known constant density does $\dot{m} = \rho Q$. For a moving measurement surface, use the relative normal velocity $(\mathbf{u} - \mathbf{v}_{\text{surface}}) \cdot \mathbf{n}$ for flux through that surface. Phase flux requires the phase-specific numerical flux or a stated reconstruction, not an undocumented multiplication of tracer by total flux.

On a face-aligned section of a finite-volume/staggered method:

$$Q_h = \sum_f s_f A_{\text{open},f} u_{n,f},$$

where s_f accounts for the chosen section orientation and A_{open} is the actual fluid aperture. If the solver already stores integrated volume flux q_f , sum q_f ; do not multiply by area again. If it stores area-weighted momentum or a mass flux, do not relabel it as velocity.

This aligned measurement is the best first feature because it can share the solver's discrete conservation definition. OpenFOAM's surface analysis distinguishes sums, area integrals, weighted averages, and normal integrals; copying the right reduction is more important than copying its interface labels. [CAD15](#)

For an arbitrary plane, intersect each active cell or fluid subcell with the plane; form nonoverlapping polygons; triangulate consistently; and integrate the reconstructed normal velocity with an identified quadrature. Clip against the real fluid region and report total sampled/open area. Use analytic uniform/linear fields to verify quadrature and an axis-aligned limit test to compare against native flux. An arbitrary-plane estimate can differ from native discrete flux even when both algorithms are implemented correctly.

6.3 Averages and pressure drop

Area mean:

$$\bar{p}_A = (\int_A p \, dA) / (\int_A dA).$$

Forward-flow-weighted mean:

$$\bar{p}_{Q+} = (\int_A p \max(\mathbf{u} \cdot \mathbf{n}, 0) dA) / (\int_A \max(\mathbf{u} \cdot \mathbf{n}, 0) dA).$$

A signed flux-weighted mean uses $\mathbf{u} \cdot \mathbf{n}$ without max, and can become unstable when inflow and outflow nearly cancel. Report forward, reverse, and net flow separately in recirculating sections. Avoid silently switching between signed and absolute weights. A denominator near zero should produce an unavailable measurement with a reason.

Define pressure drop as $\Delta p = \bar{p}_{\text{in}} - \bar{p}_{\text{out}}$ using explicitly stored sections and a common gauge. State whether p is static pressure or a total-pressure quantity. A speed change in an ideal flow can change static pressure without an irreversible loss. Under an appropriate incompressible interpretation, $p_{\text{total}} = p + \frac{1}{2} \rho |\mathbf{u}|^2$; a loss coefficient requires a defined total-pressure difference and reference dynamic pressure. Treat these as model-qualified measurements, not universal labels.

Hydraulic power is $\Delta p Q$ under the stated steady incompressible device interpretation and consistent section convention. It is not automatically motor electrical power; efficiency and other losses are separate. A nondimensional loss coefficient $K = \Delta p_{\text{loss}} / (\frac{1}{2} \rho U_{\text{ref}}^2)$ must retain its U_{ref} , density, and pressure-loss definition.

6.4 Conservation views

For a closed control volume, compare change of stored mass with boundary flux and sources. For an incompressible fixed geometry, show native flux divergence and net boundary balance. In cut cells, the divergence denominator is the active volume appropriate to the method. An extreme value in a tiny active volume deserves investigation; hiding tiny cells in a display must not erase their contribution from totals.

Present absolute and normalized discrepancies. A percentage normalized by a nearly zero net flow is meaningless. Use a declared scale such as total throughflow, initial inventory, or reference flow, with a small-denominator policy. Conservation is necessary evidence, but a conservative wrong model can still give a wrong engineering answer.

7. Loads on parts: pressure, shear, force, and torque

7.1 Fix the normal convention first

Let $\mathbf{n}_b$ point outward from a solid body into the surrounding fluid, and let fluid stress be $\sigma = -p\mathbf{I} + \boldsymbol{\tau}$. The force exerted by fluid on the body is

$$\mathbf{F} = \int_S [-p\mathbf{n}_b + \boldsymbol{\tau}\mathbf{n}_b] dA.$$

Moment about $\mathbf{x}_0$ is

$$\mathbf{M}(\mathbf{x}_0) = \int_S (\mathbf{x} - \mathbf{x}_0) \times [-p\mathbf{n}_b + \boldsymbol{\tau}\mathbf{n}_b] dA.$$

If using normals outward from the fluid domain instead, reverse the appropriate sign. A closed body's outward normals should satisfy $\sum A_f \mathbf{n}_b, \mathbf{f} \approx 0$; uniform pressure should yield near-zero net force and moment on that closed surface. This is a powerful geometry and sign test, not a universal zero-load rule for an open patch.

OpenFOAM's force utility separates pressure, viscous, and optional porous contributions and records coordinate/moment conventions. That separation is an appropriate model for a trustworthy UI. [CAD16](#)

7.2 Viscous stress and wall shear

For a Newtonian incompressible fluid of dynamic viscosity μ :

$$\boldsymbol{\tau} = \mu(\nabla \mathbf{u} + (\nabla \mathbf{u})^T).$$

More general compressible or modeled stresses need their own constitutive definition. At a wall, tangential traction is

$$\boldsymbol{\tau} \mathbf{w} = (\mathbf{I} - \mathbf{n}_b \mathbf{n}_b^T)(\boldsymbol{\tau} \mathbf{n}_b).$$

Do not estimate this simply by coloring the nearest interior speed. Wall gradients depend strongly on spatial resolution, wall position, slip/no-slip conditions, ghost reconstruction, and any turbulence wall model. A correct-looking bulk velocity profile can still yield poor wall stress.

Some solvers expose stress divided by density. OpenFOAM's wallShearStress documentation is a concrete warning: the listed field can have m^2/s^2 units rather than Pa. Inspect the actual model/field dimensions before converting or comparing. [CAD17](#) Its wall-function documentation also makes near-wall modeling an explicit numerical choice. [CAD18](#)

If Rheon's verified model supports pressure projection but not a viscous wall treatment, offer pressure inspection and perhaps a carefully qualified pressure-only load experiment. Do not call it total drag. If the geometry is a voxel approximation, integrate on the method's actual boundary representation, or clearly label a transferred CAD-surface load as approximate.

7.3 CAD surface coloring versus load integration

There are three different objects:

1. The original design surface.
2. The boundary actually used by the numerical method.
3. A display mesh used for rendering.

They may differ. A transfer map can color the display CAD mesh from nearby fluid samples, but every sample needs a valid source region, distance, and extrapolation policy. Such a map is not automatically a conservative force transfer. For integrated loads, prefer solver boundary faces/quadrature and preserve patch IDs. Avoid duplicate faces, hidden backfaces, and inconsistent winding.

Constant gauge shifts cancel over a truly closed consistently oriented surface, but generally do not cancel over an open patch. Therefore a patch pressure force requires an explicit pressure reference or opposing-side pressure. Reporting the load on one side of a plate without the other pressure may not answer the structural question.

7.4 Drag, lift, and coefficients

Choose orthonormal directions $\mathbf{e}_D$, $\mathbf{e}_L$, and a moment axis. Then $D = \mathbf{F} \cdot \mathbf{e}_D$, $L = \mathbf{F} \cdot \mathbf{e}_L$, and

$$C_D = D / (\frac{1}{2} \rho_{\text{ref}} U_{\text{ref}}^2 A_{\text{ref}}),$$

$$C_L = L / (\frac{1}{2} \rho_{\text{ref}} U_{\text{ref}}^2 A_{\text{ref}}),$$

$$C_M = (\mathbf{M} \cdot \mathbf{e}_M) / (\frac{1}{2} \rho_{\text{ref}} U_{\text{ref}}^2 A_{\text{ref}} L_{\text{ref}}).$$

No coefficient is meaningful without its reference density, speed, area, length, axes, and moment origin. Reject zero reference scales. OpenFOAM's forceCoeffs implementation makes those inputs explicit and checks them; it is a useful implementation precedent. [CAD19](#)

When a result is two-dimensional or extruded, state whether force is per unit span (N/m) or for a finite span (N). Do not compare them directly. For unsteady flows, show both history and a clearly defined averaging interval. Averaging after transients and estimating sampling uncertainty are additional tasks, not rendering options.

8. Rotation, deformation, vorticity, Q, and λ_2

8.1 Begin with the velocity gradient

Define $A_{ij} = \partial u_i / \partial x_j$. Its symmetric and antisymmetric parts are

$$S = (A + A^T)/2, \quad \Omega = (A - A^T)/2.$$

S describes local strain rate; Ω describes local rigid-rotation content. Vorticity is $\nabla \times \mathbf{u}$. All entries of A , S , Ω , and ω have s^{-1} units. A uniform translational velocity has zero gradient. A rigid rotation $\mathbf{u} = (-a y, a x, 0)$ has $\omega = (0, 0, 2a)$.

On a regular interior grid, centered differences are a reasonable reference derivative. Near walls and cut cells, use the method's boundary-aware reconstruction or an explicitly separate diagnostic approximation. Never use screen-space differences of a rendered image. A derivative amplifies noise and interpolation artifacts; smoothing before differentiation changes the analysis and must be recorded.

8.2 Why vorticity is insufficient

Simple shear $\mathbf{u} = (y, 0, 0)$ has nonzero vorticity $(0, 0, -1)$, although local shear can dominate the interpretation. A vorticity-magnitude isosurface highlights rotational/shear regions but does not prove a coherent vortex or turbulence. Prefer a beginner explanation with these simple fields to a gallery of dramatic but unexplained vortex tubes.

8.3 There are two commonly conflated Q formulas

The rotation-minus-strain quantity is

$$Q_{\text{rot}} = \frac{1}{2}(\|\Omega\|_F^2 - \|S\|_F^2).$$

The second invariant of A is

$$Q_{\text{inv}} = \frac{1}{2}[(\text{tr } A)^2 - \text{tr}(A^2)] = Q_{\text{rot}} + \frac{1}{2}(\nabla \cdot \mathbf{u})^2.$$

For divergence-free flow they coincide. With compressibility or numerical divergence they do not. OpenFOAM documents its Q as the second invariant. [CAD20](#) A Rheon field should name its convention rather than assume every “ Q ” from another program matches it.

Positive Q_{rot} marks regions where the local rotation norm exceeds the strain norm under this definition. It is a local criterion, not a unique or universal definition of a physical vortex. Choose a nonzero threshold based on the problem, and display it with units. For cross-case comparison one can also define $Q^* = Q L_{\text{ref}}^2 / U_{\text{ref}}^2$, provided those scales are explicit and fixed.

8.4 λ_2

Form the symmetric matrix $B = S^2 + \Omega^2$ and order its real eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3$. The λ_2 criterion identifies a region with $\lambda_2 < 0$; visualizations often choose a particular negative isovalue. Jeong and Hussain's original paper defines this approach; OpenFOAM's implementation description confirms the middle-eigenvalue operation. [CAD21](#), [CAD22](#)

Use a tested symmetric 3×3 eigenvalue algorithm. Sort explicitly, handle near-degenerate eigenvalues, and check against analytically known tensors. In general, Q and λ_2 need not select identical regions. Thresholds have s^{-2} dimensions. Their invariance to uniform translation does not make them invariant to arbitrary rotating observer frames; document the velocity reference frame.

8.5 Exact gradient fixtures

These values are analytical consequences of the definitions above, intended as implementation tests:

Velocity field	Divergence	Vorticity	Q_rot	Q_inv	λ_2
(U,0,0)	0	(0,0,0)	0	0	0
(-a y,a x,0)	0	(0,0,2a)	a^2	a^2	$-a^2$
(y y,0,0)	0	(0,0,-y)	0	0	0
(a x,-a y,0)	0	(0,0,0)	$-a^2$	$-a^2$	a^2
(a x,a y,a z)	3a	(0,0,0)	$-1.5a^2$	$3a^2$	a^2

The last row deliberately violates incompressibility and catches the Q-convention bug. Add a constant vector to any row: the gradients and invariants should not change. Rotate both coordinates and vectors consistently: scalar invariants should remain the same within tolerance.

9. Geometry and mesh boundaries are part of the result

A CAD viewer is not yet a CFD geometry pipeline. A rendered triangle surface may contain leaks, self-intersections, duplicate faces, wrong normals, tiny slivers, or disconnected pieces. A topologically valid mesh can still have geometric defects. CGAL’s polygon-mesh documentation makes this distinction and provides separate operations for intersection tests, connected components, normals, repair, and remeshing. [CAD23](#)

For a voxel/cut-cell reference solver, show:

- Domain box, coordinate triad, and physical dimensions.
- Cell resolution and nonuniform/anisotropic spacing where present.
- Solid mask, open volume, and face apertures.
- Inlet, outlet, wall, symmetry, periodic, and prescribed-velocity regions.
- Connected fluid components and pressure-reference components.
- Geometry approximation relative to the imported/display surface.
- Excluded or invalid cells and the reason for exclusion.

Preserve stable boundary IDs through processing. Names like inlet_A and wall_blade should connect boundary conditions, saved measurements, selected loads, and reports. A geometry edit may invalidate a section or surface selection; require remapping or mark it stale instead of silently attaching it to a different face.

A diagnostic cut plane showing cell occupancy and the numerical wall beside the smooth CAD outline is often more useful than transparent glass geometry. It reveals a passage that disappeared under voxelization, a wall one cell too thick, or an unintended connection. Those can dominate results even when the rendering looks excellent.

Do not automatically “repair” engineering geometry by closing every hole: an opening may be an inlet. Any topology-changing preprocessing belongs in a visible provenance record with before/after dimensions, region counts, and user-reviewable changes.

10. Probes, profiles, histories, and design comparison

10.1 Point and line probes

A point probe should report position, time, field, units, association, sampling method, and validity. A line probe should save endpoints, physical arc length, sample count, and sampling policy. Gaps outside fluid should remain gaps. Connecting a chart line across a wall or missing interval creates false continuity.

Probe count is a sampling control, not a simulation-resolution control. Sampling 10,000 points from a 32-cell profile produces a smoother chart, not more resolved flow. Offer a native-sample overlay where practical. Point samples exactly at a discontinuity or wall need a specified side/limit; “the value at the wall” may be a boundary quantity rather than an interpolated interior quantity.

10.2 Time histories

Use physical time on the horizontal axis. Store timestep and accepted-state IDs, because output frames may be unevenly spaced. Separate solver iteration histories from physical-time histories. Pressure residual versus Krylov iteration is not pressure at a sensor versus seconds.

Recommended first histories are section flow, section pressure difference, total liquid volume, native conservation error, probe pressure, and a chosen velocity component. Later, add forces and statistical quantities only when supported.

For irregular times, a time mean needs time weighting. For example, use trapezoidal integration of a scalar history divided by the duration, or a stated piecewise-constant convention. An arithmetic mean of output frames overweights densely saved intervals. Report averaging start/end times, sample count, missing intervals, and whether the startup transient is excluded.

An RMS fluctuation, standard deviation, and uncertainty of the mean are different quantities. A long correlated time series does not provide as many independent samples as timestamps. Do not call min/max envelopes or a resolution spread a confidence interval without a defined statistical procedure.

10.3 A measurement definition is reusable state

Make section and probe definitions first-class records. A design comparison should reuse the same physical interpretation, even if a surface’s numerical representation changes. Save:

- Common operating conditions and material parameters.
- Reference length, speed, area, pressure, and coordinate axes.
- Probe/section geometry and correspondence rules.
- Field and averaging definitions.
- Time window and transient treatment.
- Mesh/timestep/convergence evidence.

- Common color ranges and camera settings where visual comparison matters.

The essential output is a defensible change in a quantity of interest, together with its uncertainty. “The second plot is redder” is not a design conclusion if the legends differ.

11. Honest color and numerical uncertainty

11.1 Color policy

Use a sequential map for ordered nonnegative quantities such as speed or fraction. Use a diverging map when a physically meaningful central value matters, such as a signed velocity component or pressure deviation from a defined reference. Use qualitative colors for boundary IDs and connected components. Avoid a rainbow default whose unequal perceptual transitions invent apparent bands. Moreland’s color-map research explains this failure mode and offers carefully designed alternatives. [CAD24](#)

Lock ranges across time and compared runs by default. Provide explicit alternatives: current frame, selected time window, whole run, or manual. Show the selected policy. Mark under/over-range values rather than silently making them indistinguishable from endpoints. Give NaN/invalid data a distinct treatment. Log mapping needs an appropriate strictly positive domain; symmetric-log transforms must show their definition and linear threshold. ParaView’s transfer-function controls demonstrate that color, opacity, and range-rescaling are separate choices. [CAD25](#)

Every exported figure should include field, component, units, physical time, range, orientation/section location, and whether data are native or reconstructed. Do not let the camera crop the only scale information. Provide a numerical readout so color is never the only accessible channel. Moreland’s current advice also notes the tension between scalar luminance and 3D shape shading. [CAD26](#)

11.2 What to display about reliability

Put numerical diagnostics near the main result rather than in an obscure debugging console:

- Solver acceptance/rejection and residual definition.
- Native divergence norms and boundary flux balance.
- Fraction/tracer bounds and any corrections/clamping.
- Minimum active cell volume and geometry-resolution warnings.
- Spatial and temporal resolution.
- The latest available refinement comparison for the selected quantity.

Do not use a single green “accurate” badge. Prefer statements such as “flow measurement agreed within X under this tested refinement” once such evidence actually exists. A result without a refinement study can still be explored, but it should not inherit an unrelated benchmark’s error estimate.

11.3 A compact refinement derivation

For a smooth quantity $f(h)=f^*+C h^p$ in an asymptotic regime, use three systematically refined grids $h_1 < h_2 < h_3$ with common ratio $r=h_2/h_1=h_3/h_2$. If differences have a consistent sign and the assumptions hold:

$$p_{\text{obs}} = \ln |(f_3 - f_2)/(f_2 - f_1)| / \ln r,$$

$$f_{\text{extrap}} = f_1 + (f_1 - f_2) / (r^{\text{p_obs}} - 1).$$

A fine-grid GCI-style estimate is $F_s |(f_1 - f_2) / f_1| / (r^{\text{p_obs}} - 1)$, often expressed as a percentage. NASA’s tutorial describes the asymptotic assumptions, recommends three levels, and gives a safety factor of 1.25 for studies using three or more grids. [CAD27](#)

Do not apply this formula mechanically to oscillatory/nonmonotone convergence, changes of physical regime, discontinuities, unaligned geometry changes, a near-zero f_1 , or a chaotic instantaneous field at mismatched phase. Report the raw refinement values first. Compare suitable statistics for genuinely unsteady/turbulent cases. Refine timestep separately from grid where feasible; otherwise their errors can mask one another. NASA’s space-time assessment work is a useful primary example of coupled refinement planning. [CAD28](#)

Experimental validation needs experimental uncertainty and matched conditions. Keep numerical, experimental, input, and model-form uncertainty distinct. A small discretization estimate is not a bound on all physical error. [CAD03](#)

12. Volume rendering and surface rendering

12.1 Two fundamentally different representations

Surface rendering draws explicit geometry: a CAD boundary, section plane, isosurface, or streamline tube. It is well suited to geometry context and specific level sets.

Direct volume rendering accumulates color and opacity through a 3D scalar field without first constructing an explicit surface. Its appearance depends on the transfer function, ray sampling, and compositing. NVIDIA’s volume-rendering chapter provides the emission–absorption interpretation and practical sampling considerations. [CAD29](#)

Neither representation reconstructs missing data. A dense 3D volume can contain little engineering information if its scalar is just appearance tracer. A transparent volume may hide a small high-pressure region behind a broad opaque foreground. Pair volume views with quantitative slices and probes.

12.2 A minimal ray marcher

For each pixel, intersect the viewing ray with the volume bounds. Restrict the ray interval using clipping planes and opaque-scene depth. At sample points, reconstruct scalar f , map it to color $\mathbf{c}$ and opacity α , then composite front to back:

$$\mathbf{C} \leftarrow \mathbf{C} + (1 - A) \alpha \mathbf{c},$$

$$A \leftarrow A + (1 - A) \alpha.$$

Here $\mathbf{C}$ is accumulated premultiplied color. Stop early when remaining transmittance is negligible. If the reference opacity α_0 was defined for distance Δs_0 , preserve the corresponding absorption under a new step length using

$$\alpha(\Delta s) = 1 - (1 - \alpha_0)^{(\Delta s / \Delta s_0)}.$$

This prevents a resolution/step-size change from arbitrarily changing the apparent opacity. The formula is the discretized exponential attenuation relationship. Keep transfer-function units and reference sample length in the saved view. NVIDIA’s implementation discussion stresses opacity correction when sampling rate changes. [CAD29](#)

Depth clipping must use the same camera transforms and depth convention as the opaque geometry. Otherwise fluid can appear through a wall or disappear in front of it. NVIDIA's 3D-fluids chapter gives a concrete implementation of limiting ray traversal by scene depth. [CAD30](#)

12.3 Performance without scientific ambiguity

Useful optimizations include early termination, empty-space skipping, coarse preview resolution during interaction, brick min/max ranges, and cached transfer functions. They need correctness boundaries:

- A brick is not empty merely because its current transfer function happens to hide it; changing the transfer function invalidates that skip decision.
- A coarse mipmap can erase thin jets, walls, and peaks. Label interactive previews and recompute final views at the requested quality.
- Lighting gradients are rendering derivatives; do not reuse them as certified physical gradients.
- Color/opacity edits should not mutate solver arrays.
- A camera change should not change a numerical measurement.

Scientific volume rendering can be valuable for tracer distribution and spatial context. Place it after slices, measurements, and basic validation in the implementation order. Surface reflections, refraction, foam, caustics, and cinematic lighting are optional presentation work with no direct bearing on pressure-drop accuracy.

13. Export and scalable web/Rust architecture

13.1 First export: VTK XML plus a manifest

For the current particle preview, precede the volume export with a particle export. A .vtp dataset with explicit vertex cells can hold positions and point arrays for velocity, mass, temperature, material ID, and newly exposed stage-aligned density/pressure. Include stable particle IDs and release/source metadata when implemented. Export the actual unquantized arrays, not the appearance voxelization. For early debugging, a small CSV plus JSON manifest is also useful; do not imply the existing UI already provides this export. Reconstructed .vti fields should be separate derived products with kernel and sampling metadata.

For a uniform Cartesian dataset, VTK ImageData (.vti) can represent three-dimensional scalar/vector fields in physical coordinates. PolyData (.vtp) is suitable for surfaces and curves; UnstructuredGrid (.vtu) is suitable for explicit arbitrary cells. VTK's XML specification defines dimensions/extents, origin/spacing, field association, and binary/appended representations. [CAD31](#)

For $N_x \times N_y \times N_z$ cells, an ImageData point lattice normally has $(N_x+1) \times (N_y+1) \times (N_z+1)$ points and cell arrays have $N_x N_y N_z$ tuples. A different valid choice is a point lattice of cell-center samples, but then its geometry describes a sample lattice rather than the original cell volumes. Name that choice explicitly. Test physical locations, not merely tuple counts.

A single cell/point association cannot faithfully encode all three native staggered components as one colocated vector without transformation. Recommended first export:

- Main cell grid: native cell scalars, masks, component IDs, and a clearly labeled reconstructed vector for standard viewers.

- Optional separate component grids: native u/v/w with their own extents and physical offsets.
- Surface dataset: numerical boundary geometry, patch IDs, available native boundary fields.
- CSV/JSON: measurements and diagnostics with real timestamps and definitions.
- Manifest: units, solver/model provenance, pressure gauges, fraction semantics, transforms, array layout, and checksums.

Use an actual-time collection/series mechanism supported by the target ParaView version. Do not equate frame number with time. Before adding compression or large parallel files, round-trip a tiny asymmetric fixture and verify coordinates, values, components, time, and association in an independent reader. An asymmetric field catches transposed axes that a sphere or symmetric vortex can conceal.

13.2 Later formats

VTKHDF is an evolving HDF5-based option with explicit temporal layouts and an implementation-status matrix. Check the exact reader/writer versions and dataset types rather than assuming the specification implies support in every installed ParaView or VTK build. Current documentation describes temporal ImageData and its time-first storage layout. [CAD32](#), [CAD33](#)

Large transient datasets benefit from chunked arrays, selective field loading, and separate static geometry. A mesh format for presentation is not automatically an analysis archive: a colored GLB, STL, or screenshot generally does not preserve the original volume field, staggered association, pressure reference, or measurement definitions. Keep scientific data and presentation exports separate even when both are convenient.

13.3 Three implementation routes

Route A: external ParaView first. The lowest-risk reference route. Write a small correct exporter, reproduce the intended analysis pipeline in ParaView, and use analytical fixtures and numerical comparisons to isolate solver/export/viewer issues. It supplies an independent implementation, not automatic validation of the solver.

Route B: web viewer using vtk.js or VTK.wasm. vtk.js provides browser-oriented visualization components and remote-geometry/rendering integration; VTK.wasm makes C++ VTK pipelines available through WebAssembly. Feature sets, bundle size, and browser/backend support should be tested for the chosen version. [CAD34](#), [CAD35](#) Do not assume vtk.js contains every desktop VTK filter. Its documented ImageStreamline is a specific RK2 ImageData algorithm and does not currently interpolate input arrays onto output streamline points, so speed coloring and richer termination/validity semantics may need additional work. [CAD36](#)

Route C: shared Rust analysis plus wgpu presentation. Keep a pure analysis library for metadata, sampling, reductions, derivatives, integration, and export. Use wgpu for slice/volume textures, geometry, glyph instances, and curve rendering. wgpu supports native backends and browser WebGPU/WebGL paths, but support levels and available features differ. [CAD37](#), [CAD38](#) Query limits and formats on the actual adapter. Do not hard-code a 3D texture size or assume compute shaders and float filtering are universally available. [CAD39](#)

For Rheon, choose based on existing UI architecture and deployment constraints, after code review. A small custom native inspector can be appropriate; a browser prototype can use vtk.js; a desktop reference can remain ParaView. The scientific contract should be common to all three.

13.4 Memory budgets that matter

These are arithmetic estimates for uncompressed Float32 cell scalars, excluding halos, geometry, masks, GPU padding, and temporary arrays:

Grid	Cells	One scalar	Three-component vector	Six scalars
64 ³	262,144	1 MiB	3 MiB	6 MiB
128 ³	2,097,152	8 MiB	24 MiB	48 MiB
256 ³	16,777,216	64 MiB	192 MiB	384 MiB
512 ³	134,217,728	512 MiB	1.5 GiB	3 GiB

Two time snapshots, CPU and GPU copies, gradients, staging buffers, and an extracted surface can multiply these totals. A full 3×3 gradient alone is nine scalars per cell. Compute derived fields on demand or over regions, cache with bounds, and avoid retaining every temporary array.

A 256³ scalar at 30 uploads/s is roughly 1.875 GiB/s of raw upload data, before overhead. Upload once per accepted displayed frame, not once per camera movement. Changing a color map should normally update a small lookup texture, not resend the volume.

Run heavy CPU analysis in a worker/task with cancellation. Coalesce rapid slice-slider updates and never show a stale result under a new timestamp. Publish immutable accepted snapshots, or use a ownership/synchronization scheme that prevents the renderer from reading a partially updated state. For truly large data, server-side extraction/rendering can reduce client load; ParaView distinguishes transmitting geometry from transmitting rendered images. Its documented server mode also requires proper access/network security rather than public exposure of an unprotected server. [CAD40](#)

14. A proposed analysis API

The important design choice is not a particular class name. It is that the numerical meaning lives outside the renderer.

Snapshot

Contains immutable run/state IDs, physical time, grid/domain/boundary definitions, field descriptors, native arrays, masks, materials, reference conventions, and diagnostics.

Analysis operations

- `sample(field, positions, policy)` → values + validity + provenance
- `reconstruct_velocity(location, policy)` → vector field + provenance
- `gradient(field, boundary_policy)` → derivatives + quality/validity
- `section_flux(section, phase, method)` → signed/forward/reverse flux + area
- `section_average(field, section, weight)` → value + denominator + validity
- `surface_load(boundaries, origin, stress_model)` → pressure/viscous/total force and moment
- `trace_streamlines(snapshot, seeds, integrator)` → polylines + sample arrays + termination reasons
- `trace_pathlines(time_source, releases, integrator)` → trajectories + time/validity
- `extract_isosurface(field, level, reconstruction)` → mesh + provenance

- `history(measurement, time_range) → series + missing intervals`

Presentation state

Contains camera, visible representations, selected scalar/component, units display, range policy, transfer functions, clipping, glyph size, and display quality. It refers to immutable analysis outputs. It does not own pressure-drop formulas or silently rewrite native arrays.

Common error policy

Return typed unavailable/invalid results. Examples: “pressure reference differs between sections,” “no wall stress model,” “section has zero valid area,” “flux-weighted denominator is near zero,” and “pathline crosses an unsaved time interval.” The UI can explain these plainly; headless tests can assert them exactly.

Suggested acceptance hierarchy

1. Mathematical unit tests on exact fields.
2. Data-layout and physical-coordinate tests.
3. Boundary/mask and invalid-data tests.
4. Independent export/readback comparisons.
5. Qualified solver examples and refinement checks.
6. Rendering regression tests, including legends and invalid masks.
7. Usability/performance tests on representative hardware.

An image-regression test cannot replace the first five. Two wrong samplers can produce identical images.

15. Seven interactive worked lessons

These are proposed controls and acceptance criteria. The companion computes analytic/synthetic numbers only; it launches no UI or native simulation. Run it from the repository root with `python3 docs/research-book/companion/cad_visualization_fixtures.py --output .generated/cad-visualization/numerical-fixtures.json`. The output is a generated artifact, not a committed numerical baseline.

These lessons are deliberately designed so a reader can reason about the answer before trusting the picture. The first, third, fourth, fifth, sixth, and seventh can use analytic or synthetic fields without claiming they were produced by Rheon. Label data origin conspicuously. The second becomes a solver demonstration only after the specified model and comparison are implemented and tested.

Lab 1. A bright volume can hide a moving fluid

Question: why is a density image insufficient?

Field: on a unit box, set $\rho=1000 \text{ kg/m}^3$ and $\alpha=1$ everywhere. Compare $u=(0,0,0)$, $u=(1,0,0) \text{ m/s}$, and a prescribed linear shear $u=(y,0,0)$ with y in meters and shear rate 1 s^{-1} .

Controls: field selector, slice location, glyph toggle, normalization toggle, and a fixed color range.

Expected observation: density/fraction views are identical across all three fields; speed and arrows distinguish them immediately. A normalized arrow view cannot distinguish speeds at different y without color or labels.

Measurement: through the unit x -normal face, the three volume fluxes are 0, 1, and $0.5 \text{ m}^3/\text{s}$. Reverse the normal and every sign reverses. This demonstrates an engineering measurement that opacity alone cannot provide.

Tests: zero-vector glyph handling; units; orientation; constant/linear sampling; flux integration. No claim is made that the shear field satisfies the box's chosen wall conditions.

Lab 2. Pressure-driven planar channel

Question: can a solver reproduce a known profile, flow, and wall shear simultaneously?

Model: steady, incompressible, Newtonian, fully developed laminar flow between parallel plates $y=0$ and $y=H$; no slip; no transverse variation in the ideal model. Let $G=-dp/dx>0$.

From $0=G+\mu d^2u/dy^2$ and $u(0)=u(H)=0$:

$u(y)=G y(H-y)/(2\mu)$, $U_{\text{mean}}=G H^2/(12\mu)$, $U_{\text{max}}=1.5 U_{\text{mean}}$.

For a represented width W , $Q=W G H^3/(12\mu)$, $\Delta p=GL$, and $|\tau_w|=GH/2$. If sidewalls are physically present and influential, this planar model is no longer the exact reference; use a periodic spanwise direction or explicitly state the infinite-width assumption.

Numerical example: $H=0.01 \text{ m}$, $W=0.02 \text{ m}$, $L=0.1 \text{ m}$, $\mu=0.001 \text{ Pa}\cdot\text{s}$, $\rho=1000 \text{ kg/m}^3$, $G=1 \text{ Pa/m}$. Then $U_{\text{mean}}=0.0083333333 \text{ m/s}$, $U_{\text{max}}=0.0125 \text{ m/s}$, $Q=1.6666667\times 10^{-6} \text{ m}^3/\text{s}$, $\Delta p=0.1 \text{ Pa}$, and $|\tau_w|=0.005 \text{ Pa}$. Reynolds number based on H and mean speed is 83.3333; based on $2H$ it is 166.6667. State which definition is used.

Controls: grid resolution, driving gradient, viscosity, slice position, profile overlay, native/smooth display, and pressure reference shift.

Linked views: channel geometry, speed slice, cross-channel profile against the analytical curve, pressure along x , Q history, wall-shear values, and refinement plot.

Expected lessons: doubling G doubles speed, flux, drop, and shear in this model; doubling μ halves speed/flux while the prescribed pressure gradient and shear remain unchanged; pressure gauge shifts do not alter within-component Δp or velocity. A refined display does not substitute for a refined solver.

Failure probes: if the speed profile passes but shear fails, inspect wall placement/gradients; if Q differs from integrated profile, inspect native flux areas, staggering, or quadrature; if a closed domain is used accidentally, the setup is not this channel problem.

Lab 3. Vorticity is not a vortex detector

Question: what do vorticity, Q , and λ_2 distinguish?

Fields: rigid rotation ($-a y, a x, 0$), simple shear ($\gamma y, 0, 0$), planar strain ($a x, -a y, 0$), and isotropic expansion ($a x, a y, a z$).

Controls: field type, a or γ , translation velocity, derivative resolution, and the selected invariant.

Expected numbers: use the exact table in Section 8.5. For $a=2\text{ s}^{-1}$, rigid rotation has $|\omega|=4\text{ s}^{-1}$, $Q=4\text{ s}^{-2}$, and $\lambda_2=-4\text{ s}^{-2}$. For $\gamma=2\text{ s}^{-1}$, simple shear has $|\omega|=2\text{ s}^{-1}$ but $Q=\lambda_2=0$. For isotropic expansion with $a=2$, divergence is 6 s^{-1} , $Q_{\text{rot}}=-6\text{ s}^{-2}$, and $Q_{\text{inv}}=12\text{ s}^{-2}$.

Lesson: an attractive nonzero-vorticity region is not enough to characterize a vortex, and a divergence error can change the meaning of an unlabeled Q implementation. The expansion fixture is deliberately not an incompressible flow.

Lab 4. Streamline, pathline, and streakline disagree

Question: what is lost by freezing a time-dependent velocity field?

Use the prescribed field $u=(U, a t, 0)$, with $U>0$ and a a constant acceleration. A streamline at $t=T$ is a straight line with slope aT/U . A particle released from the origin at time τ has

$$X(t)=U(t-\tau), Y(t)=\frac{1}{2}a(t^2-\tau^2).$$

At observation T , the streakline of particles released continuously from the origin is

$$Y=(aT/U)X - aX^2/(2U^2), \text{ for } 0 \leq X \leq U(T-\tau_{\text{min}}).$$

A pathline of the particle released at $\tau=0$ follows $Y=aX^2/(2U^2)$. For $U=1\text{ m/s}$, $a=1\text{ m/s}^2$, and $T=2\text{ s}$: the current streamline through the origin is $Y=2X$; the $\tau=0$ pathline is $Y=X^2/2$ and ends at $(2,2)$; the streakline is $Y=2X-X^2/2$, also ending at that oldest particle.

Controls: observation time, release interval, saved-snapshot spacing, integration step, and curve type. Show a shared seed and use distinct labels rather than only color.

Lesson: the three curves answer different questions even in a very simple field. Coarsening integration or saved time samples demonstrates numerical error without invoking complicated turbulence. This is an analytic visualization test, not a boundary-conditioned Rheon solution.

Lab 5. The color legend can conceal a tenfold change

Field: $f(x,t)=a(t)x$ on $x \in [0,1]\text{ m}$, where f is an illustrative speed scalar and a changes from 1 to 10 s^{-1} .

Controls: animate amplitude; switch current-frame autoscale versus locked $0\text{--}10\text{ m/s}$; hover a fixed x ; display min/max history.

Expected observation: autoscaled images can look identical while the value at $x=0.5\text{ m}$ rises from 0.5 to 5 m/s . Locked range and the probe expose the difference. Add an invalid region and an above-range region to test whether the display falsely maps either to an ordinary endpoint color.

Lesson: a legend is part of the data, not decorative furniture. This lab is especially effective before asking readers to compare design screenshots.

Lab 6. Does the analysis converge, or only look smoother?

Use a synthetic quantity $f(h)=1+2h^2$ for $h=0.25,0.125,0.0625$. The values are 1.125,1.03125,1.0078125. With fine/medium/coarse ordering, the observed order is 2 and Richardson extrapolation gives exactly 1 under this constructed model. A three-grid fine GCI formula with $F_s=1.25$ gives about 0.9690%.

Controls: add an $O(h^3)$ term; add a fixed model bias; add small roundoff-like perturbations; switch to nonmonotone values; change display interpolation independently.

Expected lesson: display smoothing does not change the quantity; a constant model bias survives grid refinement; a near-zero reference value or nonmonotone sequence can make a rote relative estimate misleading. Clearly label this a manufactured convergence lesson rather than evidence about Rheon's current error.

Lab 7. Particle density, coverage, and appearance are different

Question: why can the edge of a particle cloud show lower reconstructed density even though every particle has the same mass and reference volume?

Fixture: place 125 particles at all combinations of $x,y,z \in \{-0.2,-0.1,0,0.1,0.2\}$ m. Give each particle mass 1 kg and reference volume 0.001 m³. Use the conventional poly6 kernel from Section 3.5 with support radius 0.2 m. This is a static mathematical fixture, not a Rheon fluid state or equilibrium claim.

Controls: probe position, kernel support radius, display-grid resolution, mass-density versus normalized constant-attribute reconstruction, and point/appearance/slice representations.

Expected values: total particle mass is always 125 kg and reference parcel-volume inventory is always 0.125 m³. With the stated kernel, the density sum at the origin is approximately 1009.775167 kg/m³; at the outer-face particle position $(-0.2,0,0)$ m it is approximately 816.999908 kg/m³. The radially integrated kernel is one, while the finite particle lattice does not exactly reproduce a continuous uniform medium and has truncated support at its boundary.

Lesson: a normalized display interpolant can reproduce a constant attribute despite missing support; it does not correct the physical density calculation. More display pixels do not add particles. Changing opacity or hiding a material does not change mass. Record the kernel, support radius, and data origin beside the legend.

Rheon connection: once quantitative snapshots exist, use the same layout to show particle-native density, reconstructed density, support, and pressure evaluation stage side by side. Do not feed the synthetic fixture's expected values into a Rheon test until its exact kernel/material/self-contribution conventions have been matched.

16. What to ship, and what to defer

Rheon-specific prerequisite milestone

For the reviewed particle preview, ship the following before advertising a general engineering field inspector:

1. Explicit model/branch labels, with particle SPH separated from carrier projection/transport.
2. Public read-only, stage-aligned particle diagnostics and unquantized export; never pair a pre-motion pressure with a post-motion location without its stage label.

3. Stable particle IDs and source/birth metadata; selectable particle values and material-aware speed/temperature views.
4. Histories for accepted particle mass, reference parcel volume, emitted/admitted volume, pending emitter credits, and clearly named contact/source diagnostics.
5. Full quantitative snapshots at a controlled cadence if replay, pathlines, or mass-crossing histories are required. Existing preview-byte history cannot supply missing particle states retroactively.
6. Kernel reconstruction and support diagnostics before smooth particle-field slices, volume rendering, or isosurfaces.

The carrier branch can independently gain structured-grid export, slices, and native flux measurements after its field units and retained boundary data are verified. Keep its outputs, examples, and evidence visibly separate.

First usable engineering inspector

Ship only after the field contract, physical coordinates, and validity masks are tested:

- Geometry/domain/boundary overlay.
- Field selector with units and availability status.
- Axis slices with speed, velocity component, pressure if meaningful, tracer, fraction, and diagnostics.
- Sparse velocity glyphs and a numerical probe.
- Fixed/manual color ranges and invalid-data handling.
- One face-aligned native section flux and its time history.
- VTK export plus metadata and a tiny independent round-trip test.
- At least one analytic/prescribed fixture clearly separated from a solver run.

Next quantitative layer

Add arbitrary slices, line profiles, saved section definitions, area/flux-weighted averages, pressure-drop histories, conservation charts, and a qualified channel case. Add gradients, vorticity, and streamlines with explicit reconstruction and integration tests. The renderer, headless path, and exported results must agree numerically.

Later, conditional features

Add Q/λ_2 after the gradient/boundary implementation is verified. Add pathlines/streaklines after time sampling and state retention are adequate. Add wall shear and integrated loads after wall geometry, constitutive stress, pressure meaning, and boundary treatment are established. Add thermal, multiphase, or turbulence outputs only when the solver really supplies those models and has supporting evidence.

Volume rendering, advanced interface shading, LIC, and high-density glyphs can follow as presentation work. They should not block the first useful measurements. Scalable formats, bricking, in-situ analysis, and remote rendering should be driven by measured data sizes and bottlenecks rather than speculative architecture.

Release review questions

1. Can a reader tell which fields were solved, prescribed, reconstructed, or unavailable?
2. Are all physical quantities labeled with units, frames, references, and time?
3. Does the same probe/measurement agree across UI, headless analysis, and export?

4. Do invalid cells, walls, disconnected regions, and missing time intervals remain explicit?
5. Are pressure differences, flux signs, surface normals, and moment origins tested?
6. Is the reported engineering quantity checked under relevant spatial/time refinement?
7. Is a rendered interface clearly separated from interface physics and conservation?
8. Are comparisons reproducible with saved operating conditions and measurement definitions?
9. Do fast-preview settings alter only presentation, never authoritative measurements?
10. Do capability claims match the currently verified solver mode?

The strongest first result is a small, reproducible laboratory in which a user can inspect a field, ask a quantitative question, see the corresponding measurement, and understand the evidence and limitations. That is the foundation on which a CAD-oriented CFD workflow can grow.

References

The linked evidence bibliography supplies access notes and scopes for these sources. Links here are direct primary-source destinations.

- **CAD01.** ParaView, [Filtering Data](#).
- **CAD02.** NASA NPARC Alliance, [Verification Assessment](#).
- **CAD03.** NASA NPARC Alliance, [Validation Assessment](#).
- **CAD04.** OpenCFD, [pressure function object](#).
- **CAD05.** Hirt and Nichols (1981), [Volume of fluid method for the dynamics of free boundaries](#), DOI 10.1016/0021-9991(81)90145-5.
- **CAD06.** Bridson and Müller-Fischer (2007), [Fluid Simulation course notes](#), especially grid and pressure sections.
- **CAD07.** VTK, [vtkCellDataToPointData](#).
- **CAD08.** VTK, [vtkFlyingEdges3D](#).
- **CAD09.** Schroeder and Tsalikis, Kitware (2023), [Really Fast Isocontouring](#), discussing Flying Edges and the Schroeder–Maynard–Geveci Lдав 2015 paper.
- **CAD10.** MIT OpenCourseWare, [Fluids Lecture 8: Streamlines, Pathlines, Streaklines](#).
- **CAD11.** NASA Ames, [Visualization of Numerical Unsteady Fluid Flows](#).
- **CAD12.** VTK, [vtkStreamTracer](#).
- **CAD13.** VTK, [vtkParticleTracer](#).
- **CAD14.** Cabral and Leedom (1993), [Imaging vector fields using line integral convolution](#), SIGGRAPH '93, DOI 10.1145/166117.166151.
- **CAD15.** OpenCFD, [surfaceFieldValue](#).
- **CAD16.** OpenCFD, [forces](#).
- **CAD17.** OpenCFD, [wallShearStress](#).
- **CAD18.** OpenCFD, [Wall functions](#).
- **CAD19.** OpenCFD, [forceCoeffs implementation](#).
- **CAD20.** OpenCFD, [Q](#).
- **CAD21.** Jeong and Hussain (1995), [On the identification of a vortex](#), Journal of Fluid Mechanics 285, 69–94, DOI 10.1017/S0022112095000462.
- **CAD22.** OpenCFD, [Lambda2](#).
- **CAD23.** CGAL, [Polygon Mesh Processing user manual](#).

- **CAD24.** Moreland (2009), [Diverging Color Maps for Scientific Visualization](#), expanded paper, DOI 10.1007/978-3-642-10520-3_9.
- **CAD25.** ParaView, [Color maps and transfer functions](#).
- **CAD26.** Moreland, [Color Map Advice for Scientific Visualization](#).
- **CAD27.** NASA NPARC Alliance, [Examining Spatial \(Grid\) Convergence](#).
- **CAD28.** Housman et al. (2011), NASA, [Space Time Accuracy Assessment of CFD Simulations for the Launch Environment](#).
- **CAD29.** NVIDIA GPU Gems, [Chapter 39: Volume Rendering Techniques](#).
- **CAD30.** NVIDIA GPU Gems 3, [Chapter 30: Real-Time Simulation and Rendering of 3D Fluids](#).
- **CAD31.** VTK, [XML File Formats](#).
- **CAD32.** VTK, [VTKHDF format specification](#).
- **CAD33.** VTK, [VTKHDF implementation status](#).
- **CAD34.** Kitware, [vtk.js overview](#).
- **CAD35.** Kitware, [VTK.wasm](#).
- **CAD36.** vtk.js, [ImageStreamline API](#).
- **CAD37.** wgpu, [project overview](#).
- **CAD38.** wgpu, [Backends](#).
- **CAD39.** wgpu, [Limits](#).
- **CAD40.** ParaView, [Remote and parallel visualization](#).
- **CAD41.** Price (2012), [Smoothed Particle Hydrodynamics and Magnetohydrodynamics](#), Journal of Computational Physics 231, 759–794; [author manuscript](#).
- **CAD42.** Müller, Charypar, and Gross (2003), [Particle-Based Fluid Simulation for Interactive Applications](#).
- **CAD43.** Schechter (2012), [Enhancing Particle Methods for Fluid Simulation in Computer Graphics](#), especially the Ghost SPH boundary-condition work.
- **CAD44.** Dehnen and Aly (2012), [Improving convergence in smoothed particle hydrodynamics simulations without pairing instability](#).