Physics Engine Gap Analysis

Combined Repository Analysis for Production-Grade Physics \& Numerical Algorithms Library

December 18, 2025

Contents

1 Executive Summary

1.1 Repository Overview

This analysis examines three repositories intended to form a coherent physics engine and numerical algorithms library:

Jet Fluid Engine (doyubkim-fluid-engine-dev)
: Production-grade fluid simulation with SPH, FLIP/PIC/APIC, grid-based solvers, level sets, and FDM linear system solvers. Strong in computational physics but lacks rigid body dynamics.

- **Game Physics Cookbook (packtpublishing-game-physics-cookbook):** Rigid body physics with SAT collision detection, impulse-based response, cloth simulation, and spatial partitioning. Strong in game physics but lacks advanced numerical methods.

Numerical Recipes (notoriousjayy-delete)
: Collection of numerical algorithms including ODE solvers (RK4), integration routines, linear solvers, and statistical functions. Standalone algorithms without physics framework integration.

1.2 Key Findings

Critical Gap
: No unified constraint solver architecture bridges rigid bodies and fluids

Major Overlap
: Duplicate math primitives (vec2/vec3/mat4 vs Vector2/Vector3/Matrix) requiring unification

Architectural Mismatch
: Different coordinate conventions, memory layouts, and integration patterns

Partial Implementation
: Broadphase collision limited to QuadTree/Octree; no sweep-and-prune or dynamic BVH

1.3 Strategic Recommendation

Adopt Jet's modular architecture as the foundation, integrate Game Physics Cookbook's rigid body system as a new module, and incorporate Numerical Recipes algorithms into a unified MathPrimitives/Numerics layer. Estimated total effort: 16-24 person-months for full integration.

2 Capability Coverage Matrix

Legend: ✓= Implemented — ○= Partial — ×= Missing — ⋆= Duplicate

Table 1: Core Math & Numerics — Capability Coverage

Capability	Jet	Cookbook	NumRecipes
Vector/Matrix/Quaternion	✓Vector2/3, Matrix, Quaternion	⋆vec2/3, mat2/3/4	×
Transform Types	✓Transform2/3	○mat4 functions	×
Dense Linear Algebra	○Basic ops	✓Inverse, Determinant	×
Sparse Linear Solvers	✓CG, ICCG, Jacobi, MG	×	○tridag, sor
SVD/Eigendecomp	✓svd.h	×	×
ODE Solvers	○Semi-implicit Euler	○Euler/Verlet	✓rk4, stepper
Numerical Integration	×	×	✓qgaus, romberg, quad3d
Interpolation/Splines	✓Cubic interpolation	×	✓interp_linear, savgol
Random/Sampling	×	○Basic Random()	✓multinormaldev, ranpt

2.1 Core Math & Numerics

2.2 Rigid Body Physics

Table 2: Rigid Body Physics — Capability Coverage

Capability	Jet	Cookbook	NumRecipes
Rigid Body State	○RigidBodyCollider (static)	✓RigidbodyVolume	×
Mass Properties/Inertia	×	✓InvMass, inertia	×
Broadphase Collision	✓PointHashGrid, KdTree	✓QuadTree, Octree	×
Narrowphase (GJK/EPA)	×	×	×
Narrowphase (SAT)	×	✓SAT for OBB	×
Contact Manifold	×	✓CollisionManifold	×
Impulse Resolution	×	✓ApplyImpulse()	×
Friction/Restitution	×	○bounce param	×
Constraints/Joints	×	○DistanceJoint	×
Sequential Impulses/PGS	×	○ImpulseIteration loop	×
Sleeping/Islands	×	×	×
CCD (Continuous)	×	×	×

2.3 Soft Body / Cloth / Particles

2.4 Fluids

3 Gap List

3.1 Critical Gaps (High Severity)

3.1.1 Gap 1: GJK/EPA Narrowphase Collision

Description No Gilbert-Johnson-Keerthi or Expanding Polytope Algorithm for convex hull collision. SAT implementation only handles box primitives.

Table 3: Soft Body / Cloth / Particles — Capability Coverage

Capability	**Jet**	**Cookbook**	**NumRecipes**
Particle System Data	✓ParticleSystemData2/3	✓Particle class	×
Mass-Spring Cloth	×	✓Cloth.h with springs	×
Bending/Shear Springs	×	✓SetBendSprings, SetShearSprings	×
Constraint Solving	×	✓SolveConstraints()	×
Position-Based Dynamics	×	×	×

Table 4: Fluids — Capability Coverage

Capability	**Jet**	**Cookbook**	**NumRecipes**
Grid-Based Eulerian	✓GridFluidSolver2/3	×	×
Advection Solvers	✓SemiLagrangian, Cubic	×	×
Pressure Projection	✓GridPressureSolver	×	×
Diffusion	✓Forward/Backward Euler	×	×
SPH	✓SphSolver2/3	×	×
PCISPH	✓PciSphSolver2/3	×	×
FLIP/PIC/APIC	✓FlipSolver, ApicSolver	×	×
Level Sets	✓LevelSetSolver, FMM, ENO	×	×
Marching Cubes	✓marching_cubes.h	×	×
Boundary Conditions	✓Blocked, Fractional	×	×

Current State
Missing entirely. Cookbook has SAT for OBB/AABB but not general convex shapes.

Target Module
Geometry (or new CollisionDetection module)

Impact Correctness - Cannot handle arbitrary convex colliders required for realistic rigid body simulation

Implementation
Implement GJK with Minkowski difference, EPA for penetration depth, warm-starting with cached simplices

Acceptance Unit tests for sphere-sphere, box-box, convex-convex; benchmark vs SAT

Effort Large (3-4 weeks)

3.1.2 Gap 2: Unified Constraint Solver

Description No centralized constraint solver architecture. Cookbook has ad-hoc impulse iteration; Jet has FDM solvers but no rigid body constraint support.

Current State
Partial - ImpulseIteration loop in PhysicsSystem.cpp but no proper PGS/MLCP formulation

Target Module
New ConstraintSolver module

Impact Correctness/Stability - Joint stacking, complex constraint scenarios fail

Implementation
Sequential Impulses with warm-starting, constraint graph, bias factors, Baumgarte stabilization

Acceptance Stack of 10+ boxes stable; pendulum chain test; joint motor accuracy

Effort Large (4-5 weeks)

3.1.3 Gap 3: Continuous Collision Detection (CCD)

Description No time-of-impact calculation for fast-moving objects. Tunneling occurs at high velocities.

Current State
Missing entirely in all repositories

Target Module
CollisionDetection

Impact Correctness - Bullets, fast projectiles pass through thin walls

Implementation
Conservative advancement with GJK, speculative contacts, TOI root finding

Acceptance High-velocity sphere vs thin plane test; frame-rate independent collision

Effort Medium (2-3 weeks)

3.1.4 Gap 4: Island Management & Sleeping

Description No grouping of interacting bodies into islands or sleep state management for stationary objects.

Current State
Missing entirely

Target Module
PhysicsCore

Impact Performance - $O(n^2)$ collision checks for all bodies regardless of activity

Implementation
Union-find for islands, velocity/energy threshold for sleeping, wake propagation

Acceptance 100 stacked boxes achieve 60fps; proper wake-on-impact

Effort Medium (2 weeks)

3.2 Medium Severity Gaps

3.2.1 Gap 5: Advanced Joint Types

Description Only DistanceJoint implemented. Missing hinge, slider, ball-socket, motors, limits.

Current State
Partial - DistanceJoint.h exists (Evidence: Code/DistanceJoint.h)

Implementation
Derive from base Constraint class; implement position/velocity constraints with Jacobians

Effort Medium (2-3 weeks per joint type)

3.2.2 Gap 6: Position-Based Dynamics (PBD)

Description No PBD implementation for unified soft/rigid body simulation.

Current State
Missing - Cloth uses spring-damper model, not PBD constraints

Implementation
XPBD with compliant constraints, iterative projection, damping

Effort Large (4+ weeks)

3.2.3 Gap 7: Dynamic BVH / Sweep-and-Prune

Description Broadphase limited to QuadTree/Octree. No dynamic AABB tree or incremental SAP.

Current State
Partial - Static BVH in Geometry3D.h (BVHNode struct), QuadTree/Octree in Scene.h

Implementation
Self-balancing AABB tree with incremental updates, or sorted axis lists with insertion sort

Effort Medium (2 weeks)

4 Overlap/Conflict List

4.1 Math Type System Duplication

Overlap Jet uses Vector2¡T¿/Vector3¡T¿/Matrix3x3¡T¿/Quaternion¡T¿ (templated). Cookbook uses vec2/vec3/mat2/mat3/mat4 (float-only structs).

Evidence Jet: include/jet/vector.h, include/jet/matrix.h. Cookbook: vectors.h, matrices.h

Risk Type mismatches at module boundaries, implicit conversions causing bugs, maintenance burden

Recommendation
Adopt Jet's templated types as canonical. Create thin adapters vec2 → Vector2F, mat4 → Matrix4x4F. Gradually migrate Cookbook code.

Migration Phase 1: typedef aliases. Phase 2: Replace usages. Phase 3: Remove Cookbook math headers.

4.2 Particle System Data Structures

Overlap Jet's ParticleSystemData2/3 vs Cookbook's Particle class

Evidence Jet: include/jet/particle_system_data.h. Cookbook: Particle.h

Risk Memory layout conflicts, incompatible neighbor search

Recommendation
Keep both but establish clear boundaries: Jet for fluid particles, Cookbook for rigid/cloth particles. Create common IParticle interface if unified iteration needed.

4.3 Coordinate System / Handedness

Conflict Jet does not explicitly define handedness. Cookbook uses left-handed OpenGL conventions with Y-up.

Evidence Cookbook matrices.cpp: Projection(), LookAt() assume OpenGL conventions

Risk Incorrect physics when mixing modules; inverted normals in collision

Recommendation
Document and enforce right-handed Y-up convention throughout. Add coordinate transform utilities at module boundaries.

4.4 Neighbor Search Structures

Overlap Jet: PointHashGridSearcher, PointKdTreeSearcher. Cookbook: QuadTree, Octree (Scene.h)

Recommendation
Consolidate into SpatialPartitioning module. Use Jet's parallel hash grid for SPH, Cookbook's Octree for broadphase rigid body.

5 Architectural Misalignment Findings

5.1 Module Boundary Violations

Issue Cookbook's Geometry3D.h is a monolithic 4000+ line header containing collision detection, spatial structures, mesh handling

Decomposition Target
Should be split per module view: primitives → Geometry, BVH → SpatialPartitioning, collision → CollisionDetection

Refactor Extract struct definitions to primitives.h, BVH to bvh.h, collision tests to collision.h

Issue PhysicsSystem.cpp directly accesses RigidbodyVolume internals (position, orientation)

Decomposition Target
PhysicsCore should use abstract Rigidbody interface

Refactor Introduce IRigidbody interface with GetPosition(), SetPosition(), GetInverseMass() etc.

5.2 Dependency Direction Issues

Issue Cookbook's rendering code (FixedFunctionPrimitives) embedded in physics demos, creating circular dependency

Required Rendering should depend on Physics, not vice versa

Refactor Move all Render() methods to separate visualization module; physics types return geometry data only

5.3 Missing Abstractions

ISurface Interface
: Jet has Surface2/3 base classes; Cookbook lacks equivalent abstraction for collision shapes

IIntegrator Interface
: No common interface for Euler/Verlet/RK4 integration; hardcoded in each system

IConstraint Interface
: DistanceJoint is concrete class; no base Constraint type for solver to iterate

6 Prioritized Roadmap

6.1 Phase 0: Safety & Baseline (4-6 weeks)

Goal: Establish testing infrastructure, determinism, and documentation baseline

CI/CD Pipeline
: GitHub Actions for build/test on Linux/Windows/macOS

Unit Test Framework
: Adopt Google Test; port existing tests; achieve 60% coverage on math primitives

Determinism Audit
: Fixed-point RNG seeds; verify bitwise reproducibility across platforms

Module README Files
: Document public API, usage examples for each module

Acceptance All CI green; same simulation produces identical results on all platforms

6.2 Phase 1: Core Math Unification (3-4 weeks)

Goal: Single source of truth for math types and operations

Math Type Consolidation
: Jet templates as base; adapter headers for Cookbook compatibility

Numeric Utilities Integration
: Port relevant NumRecipes algorithms (rk4, romberg) to Jet's namespace

Coordinate System Documentation
: Enforce right-handed Y-up; add transform utilities

Acceptance Single math.h header usable by all modules; numerical precision tests pass

6.3 Phase 2: Rigid Body Integration (6-8 weeks)

Goal: Production-quality rigid body module integrated with Jet architecture

Rigid Body Module
Port Cookbook's RigidbodyVolume, adapt to Jet's coding style

GJK/EPA Implementation
Full convex collision with warm-starting

Sequential Impulses Solver
Replace ad-hoc iteration with proper SI implementation

Joint System
Distance, Hinge, Ball-socket with motors and limits

Acceptance Box stacking demo stable; ragdoll example working; benchmark vs Box2D/Bullet

6.4 Phase 3: Collision Optimization (4-5 weeks)

Goal: Scalable broadphase and optional CCD

Dynamic BVH
Self-balancing AABB tree with incremental updates

Island Management
Union-find islands with sleeping

CCD Implementation
Conservative advancement; speculative contacts

Acceptance 1000 body scene at 60fps; no tunneling with fast projectiles

6.5 Phase 4: Advanced Features (8+ weeks)

Goal: Unified soft body, fluid coupling, and production polish

PBD/XPBD System
Unified cloth/soft body with compliant constraints

Fluid-Rigid Coupling
Two-way interaction between SPH/FLIP and rigid bodies

GPU Acceleration
CUDA/OpenCL kernels for broadphase and solver

Profiling Integration
Tracy/Optick markers throughout pipeline

Acceptance Cloth-fluid interaction demo; GPU 10x speedup on appropriate workloads

7 Recommended Module Decomposition Updates

Based on this analysis, the following updates to the module decomposition are recommended:

7.1 New Modules to Add

RigidBody Module
: Rigid body state, mass properties, inertia tensor management (from Cookbook)

CollisionDetection Module
: Broadphase (BVH, SAP), narrowphase (GJK/EPA, SAT), contact manifold generation

ConstraintSolver Module
: Sequential impulses, constraints interface, joint implementations, islands

Numerics Module
: ODE integrators, quadrature, root finding (consolidate from NumRecipes)

7.2 Revised Dependency Graph

Proposed layer ordering (lower depends on higher):

Layer 0: CoreFoundation

Layer 1: MathPrimitives, Numerics

Layer 2: Geometry, Fields

Layer 3: CollisionDetection, Grids

Layer 4: RigidBody, Particles

Layer 5: ConstraintSolver, Solvers (Fluid)

Layer 6: Animation, PhysicsWorld (unified)

Layer 7: Serialization, Bindings, Examples

7.3 Key Interfaces to Define

End of Gap Analysis Report

Table 5: Key Interfaces to Define

Interface	**Module**	**Purpose**
ICollider	CollisionDetection	Abstract collision shape with support mapping
IBroadphase	CollisionDetection	Query interface for spatial acceleration
IConstraint	ConstraintSolver	Base for all constraints and joints
IIntegrator	Numerics	Step function for various ODE methods
IRigidBody	RigidBody	State access for physics world iteration
IPhysicsWorld	Animation	Unified world managing rigid + fluid + soft