GDC 09

learn
network
inspire

www.GDConf.com

Game Developers Conference®

March 23-27, 2009 | Moscone Center, San Francisco

Anatomy of a Physics Engine

Erwin Coumans

erwin_coumans@playstation.sony.com

How it fits together

- Terminology
  - Rigid Body Dynamics
  - Collision Detection
- Software Design Decisions
  - Trip through the Physics Pipeline
  - Typical Optimizations
- Beyond Rigid Bodies
  - Softbody and Fluid simulation
  - Networked Physics

Modeling and Simulation

Physical System

Idealization ⇓

Mathematical Model

Discretization ⇓

Discrete Model

Computation ⇓

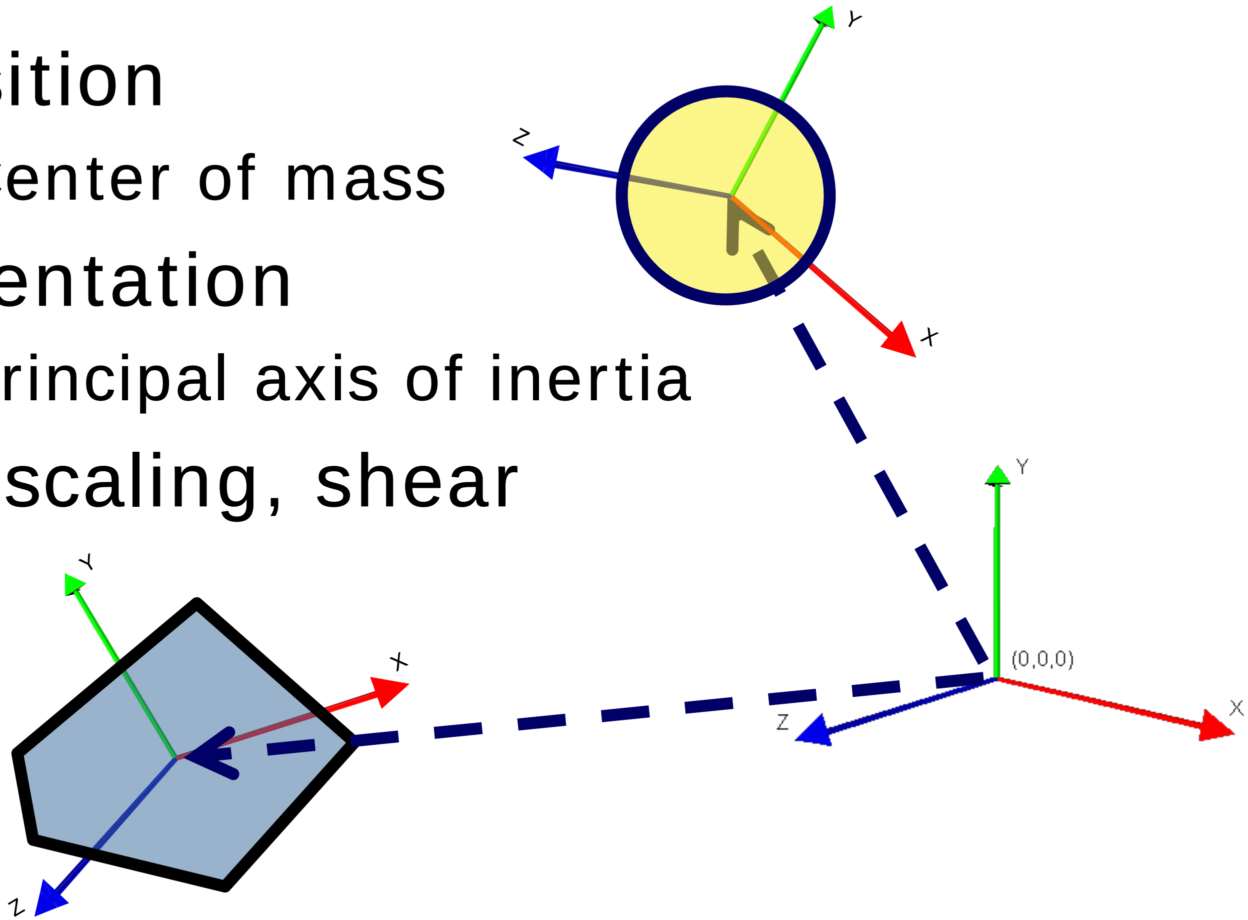
Discrete Solution

Simplified Model

- Objects are rigid, no deformation
- The static environment is indestructible
- Approximate the shape of objects
  - collision detection, ray cast, inertia
  - Can have multiple representations
- Can add deformation, destruction and more realistic representation later

Rigid Body Transform

- Position
  - Center of mass
- Orientation
  - Principal axis of inertia
- No scaling, shear

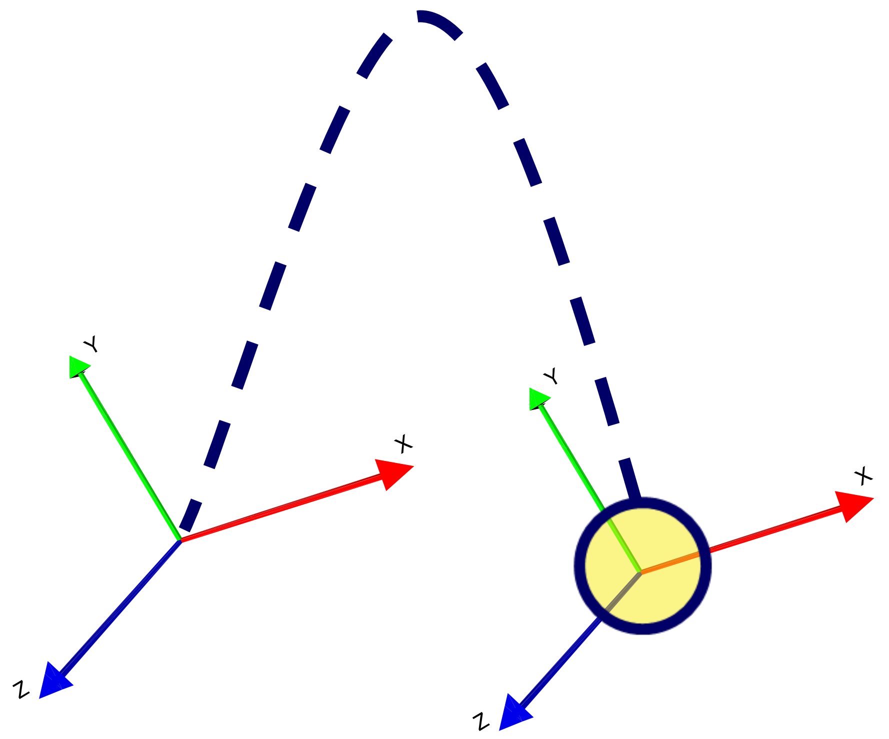


Moving Things Around

» Kinematics

  Describes motion

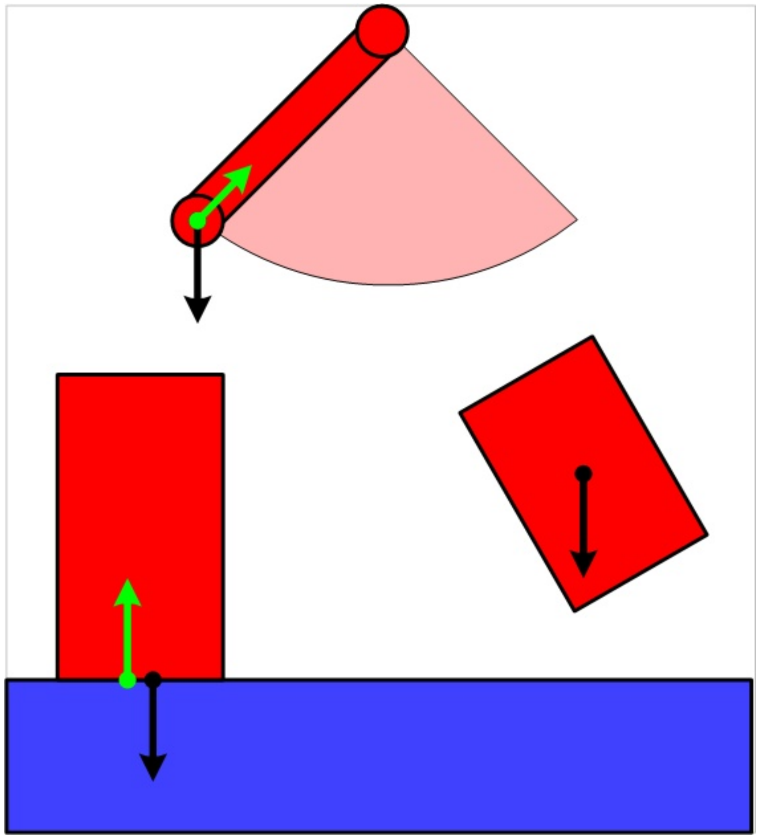
  Uses position, velocity, momentum, acceleration



$$x_t = x_0 + v_i t + \frac{1}{2}at^2$$

Moving Things Around

- Kinematics
  - *Describes* motion
  - Uses position, velocity, momentum, acceleration
- Dynamics
  - *Explains* motion
  - Uses forces
  - ...and impulses

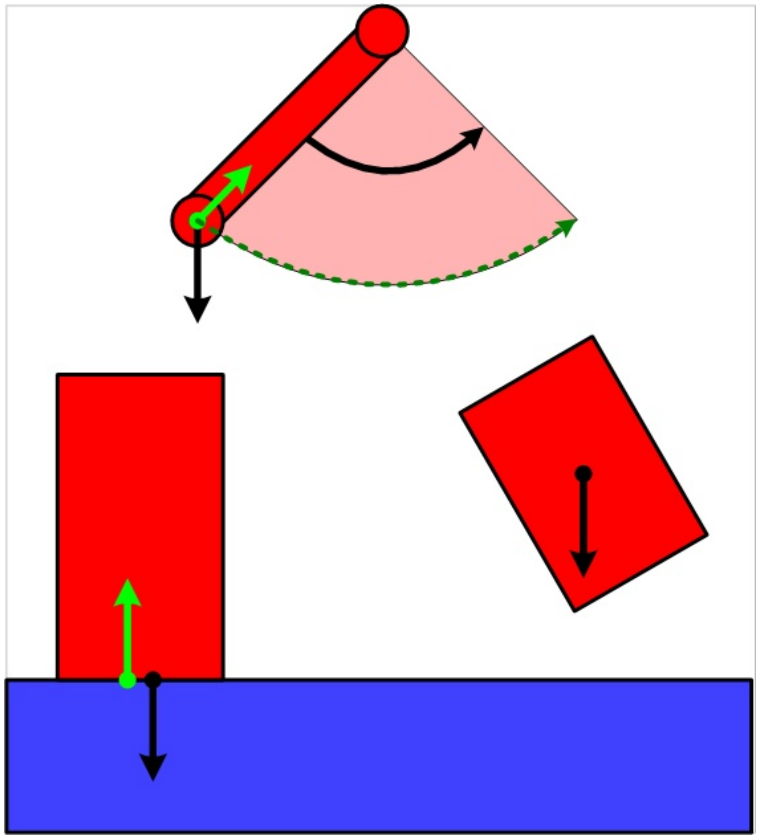


$$F = m * a$$

www.GDConf.com

Moving Things Around

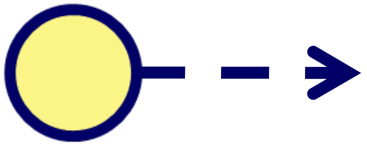
- Kinematics
  - *Describes* motion
  - Uses position, velocity, momentum, acceleration
- Dynamics
  - *Explains* motion
  - Forces (F=ma)
  - Impulses
- Rotation
  - Torque
  - Angular momentum
  - Moment of inertia



$$\tau = I * \alpha$$

Newton’s Laws of Motion

1. Law of Inertia

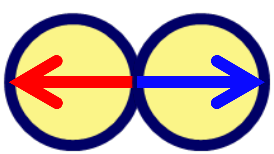


Velocity stays constant without force

2. Force is mass times acceleration

$$F = m * a$$

3. Action = - Reaction



2nd Law for Rotation

- Torque = Inertia times angular acceleration

$$\tau = I * \alpha$$

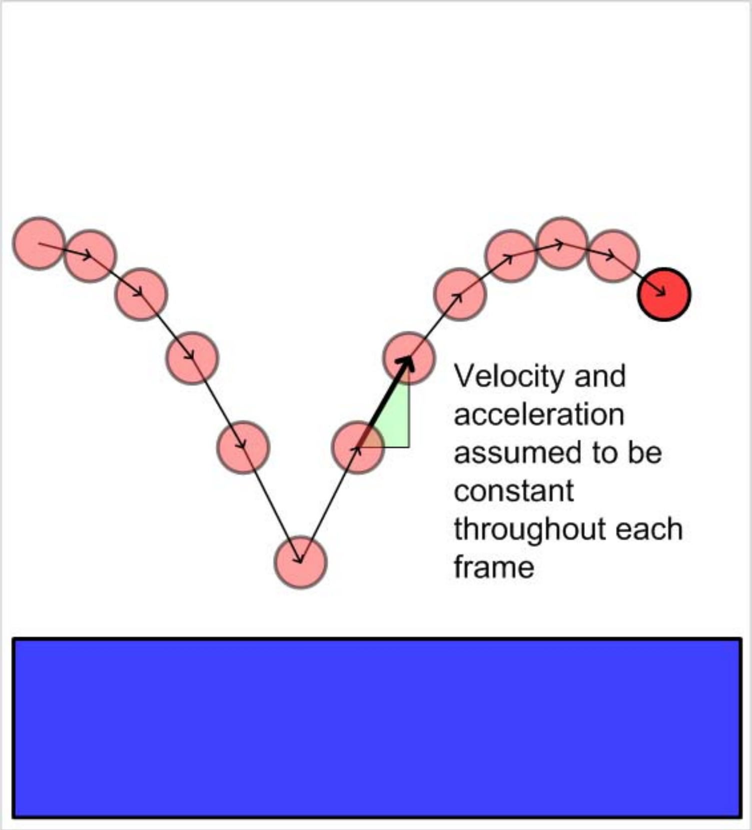
- Around the principal axis, center of mass
- Inertia is mass distribution over the shape

Newton's Laws and Spatial, Temporal Scale

- Object size should be not too small
  - Bigger than an atom, or pebble
- And not to large
  - Unlike planets or the universe
- Speed of objects < < light speed

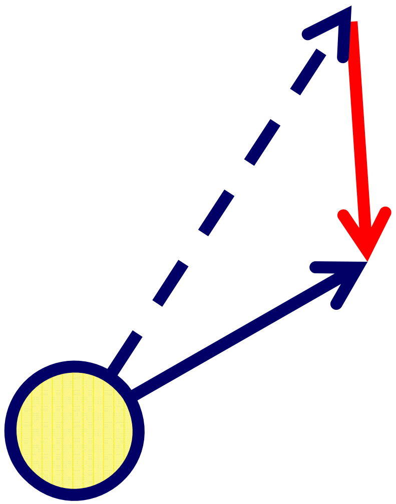
Simulation Baggage

- Flipbook syndrome
- Things *mostly* happen in-between snapshots
- Curved trajectories treated as piecewise linear
- Terms often assumed to be constant throughout the frame

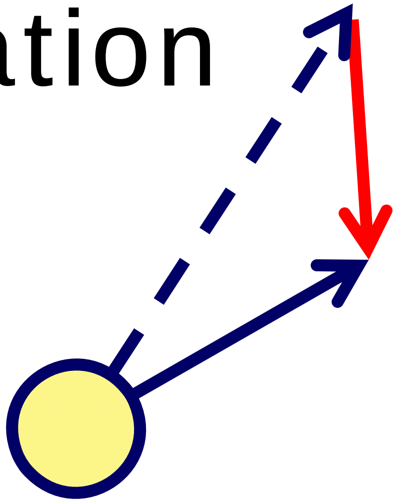


Unconstrained Motion

- How to compute the next position and velocity from current position and velocity?
- This process is called *integration*;
- An algorithm for doing this in an *integrator*
- Which integration method to use?



Position Integration



» Euler Integrator

$$x_{t+1} = x_t + v_t * dt$$

» Simplectic Euler

$$x_{t+1} = x_t + v_{t+1} * dt$$

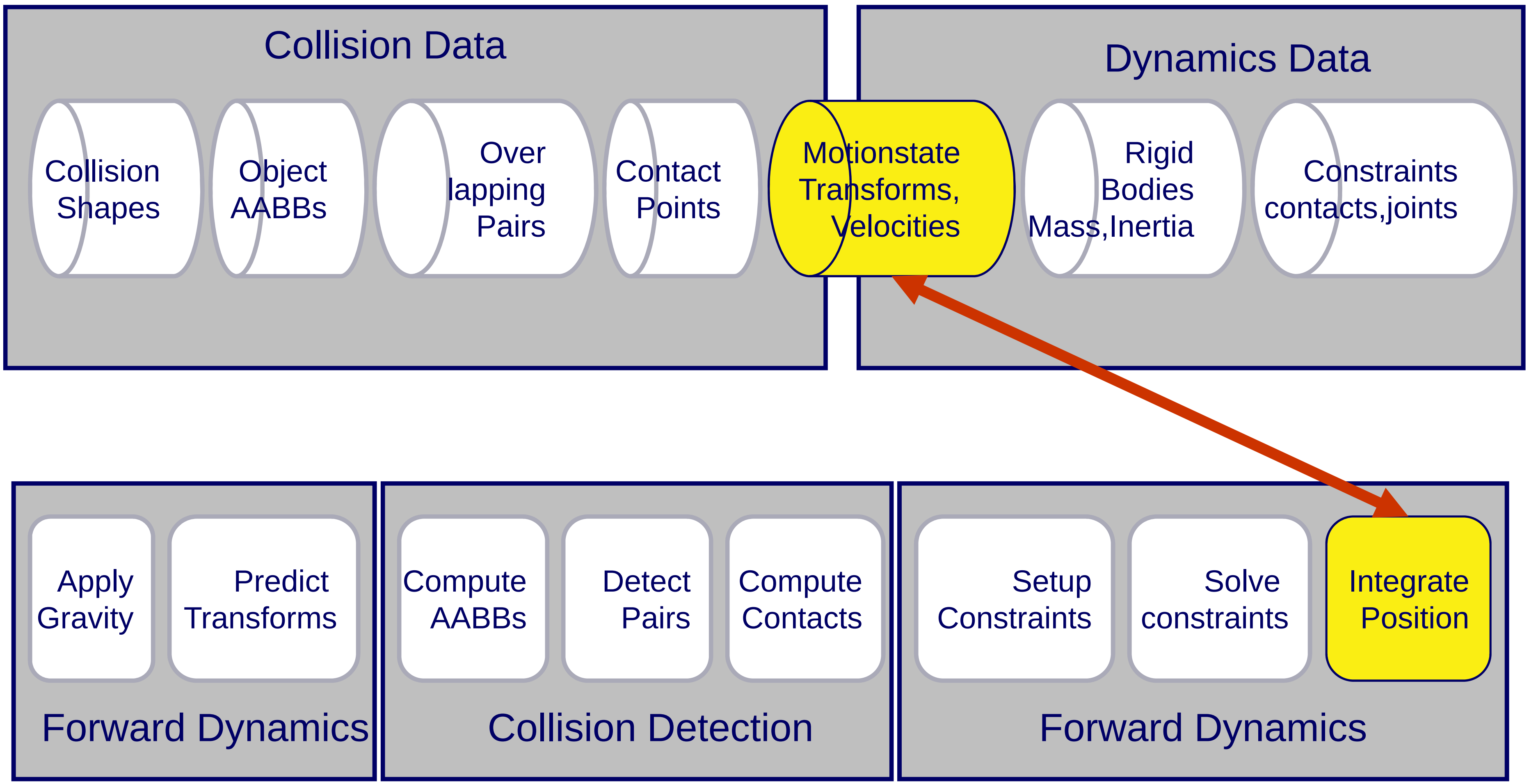
First update velocity, then position

» Runge Kutta

Accuracy often not worth it

» See Erin Catto's talk on integrators

Position Integration



Applying Gravity

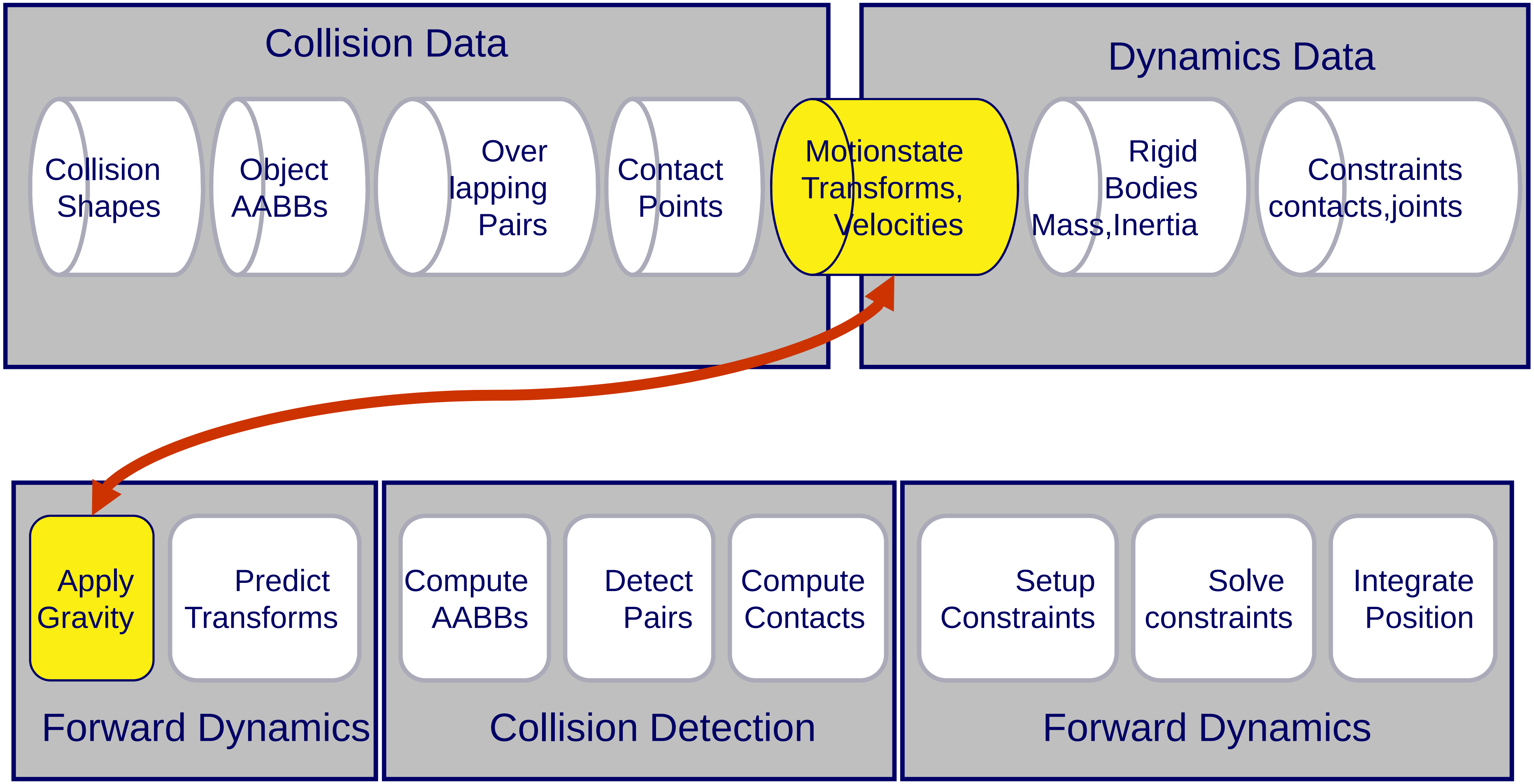
- Update the acceleration

$$a_{t+1} = a_t + F / m$$

- Then update the velocity

$$v_{t+1} = v_t + a\ * dt$$

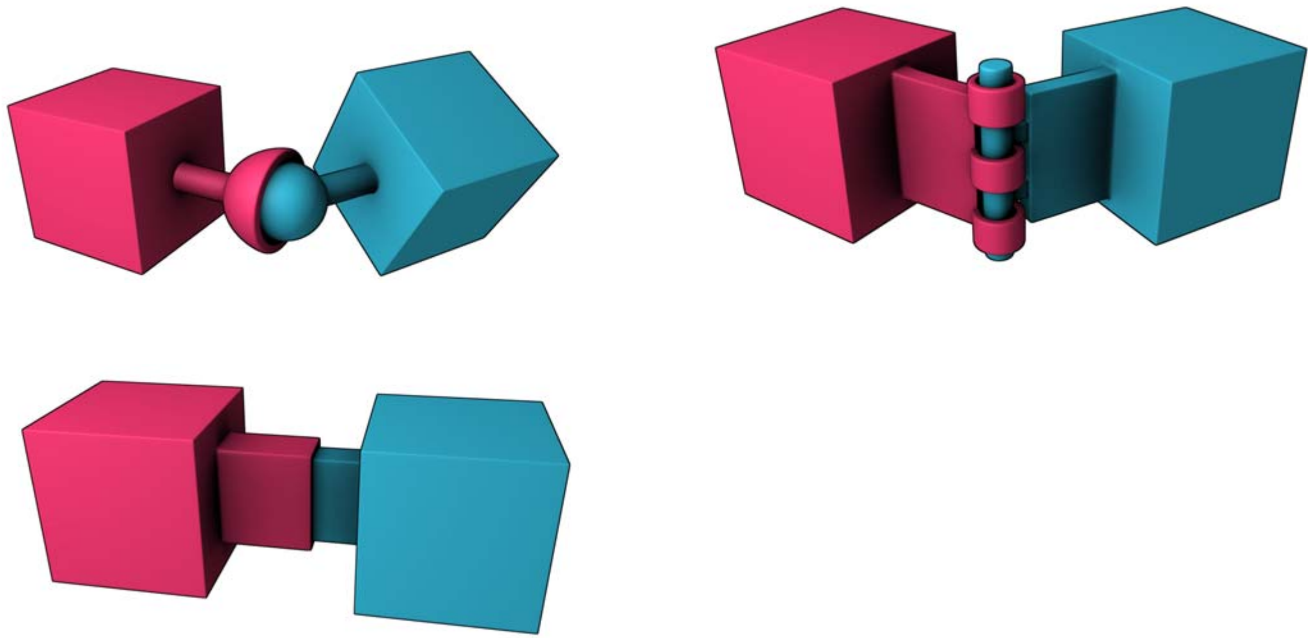
Apply Gravity



Constrained Motion

- Reducing the degrees of freedom
- Collision Impact
- Non-penetration
- Friction
- Joint connections

Constraints and Degrees of Freedom



Constraint Solver

» Calculate new velocities to satisfy constraints between objects

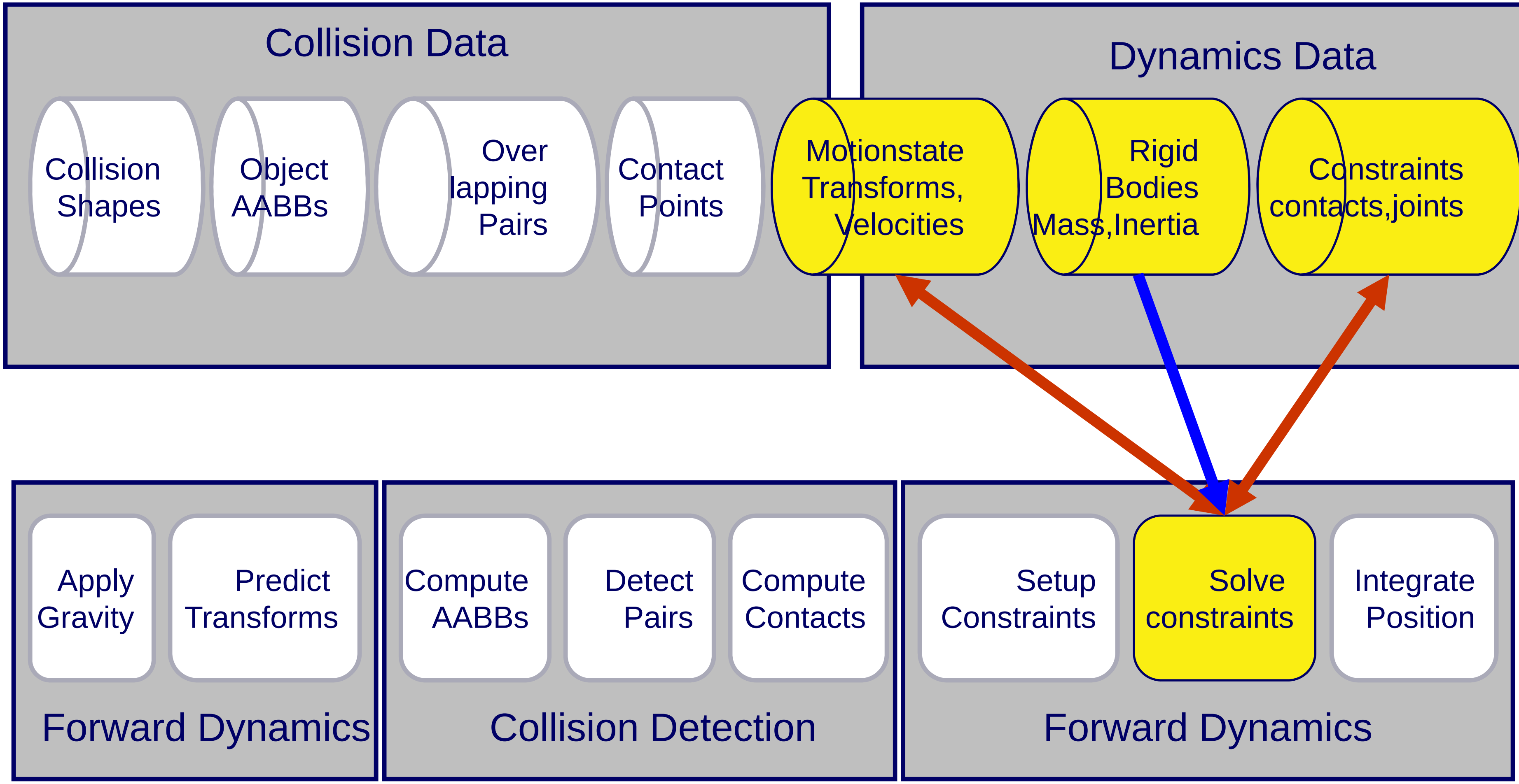
1. Impact/collision constraints
2. Non-penetration constraints
3. Friction constraints
4. Joint constraints

Iterative Constraint Solver

1. Measure the relative velocity and penetration error between pairs of objects
2. Apply impulse to correct this error
3. Repeat step 1 and 2 for all pairs, a small number of times (10)

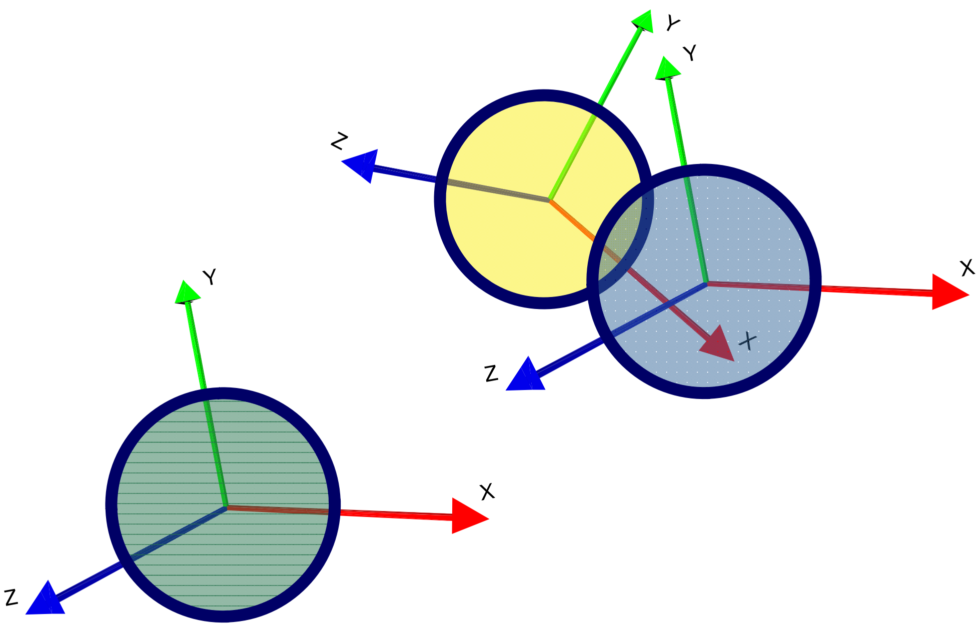
» See Erin Catto's talk for details

Constraint Solver



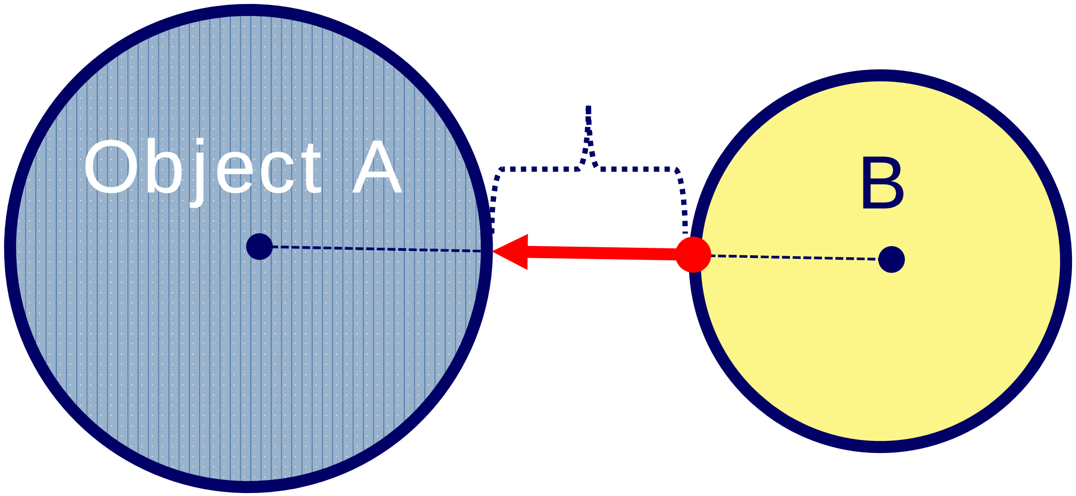
Collision Detection

» Check if any objects overlap



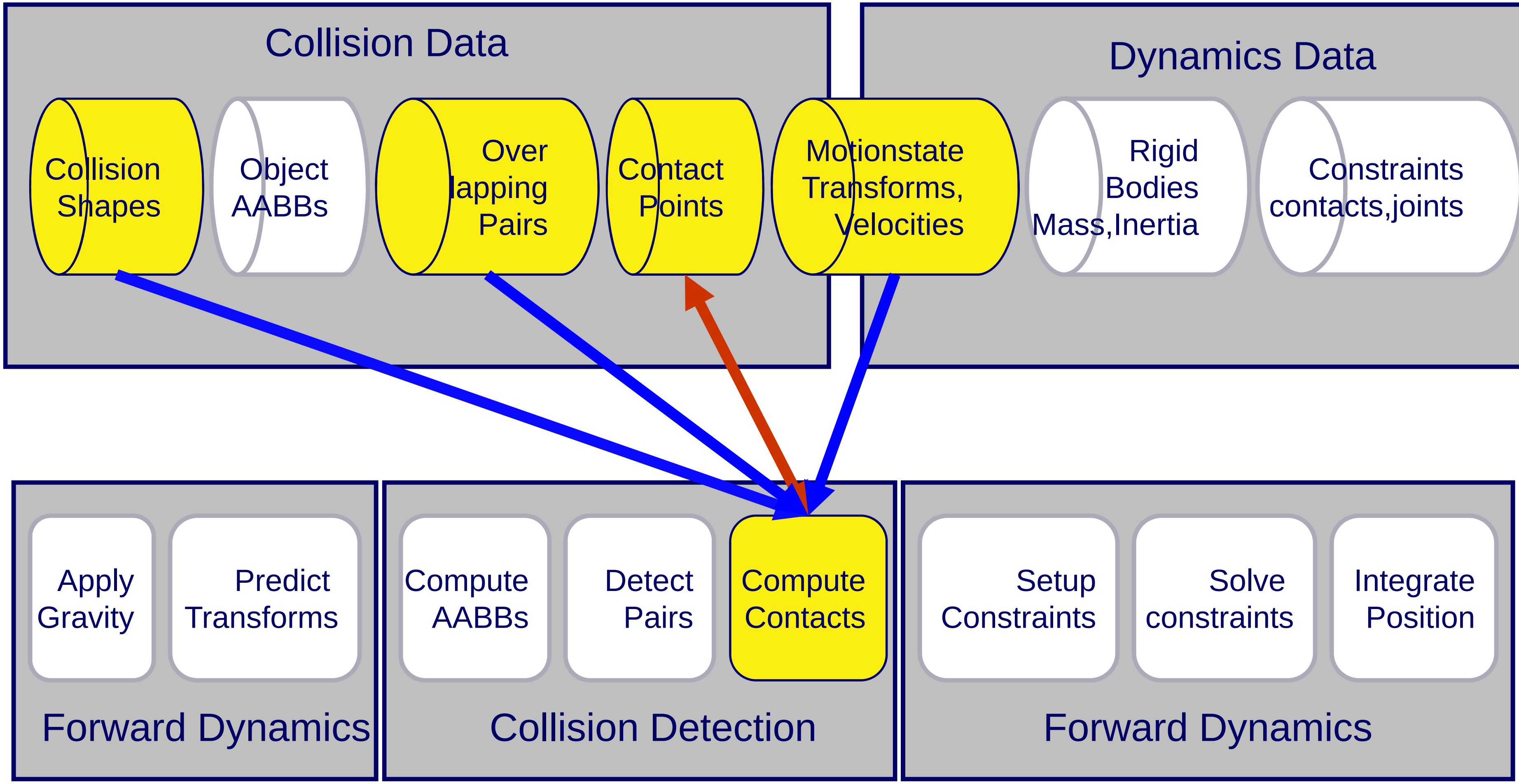
Collision Narrowphase: Contact Points

- Position, normal and distance of closest points



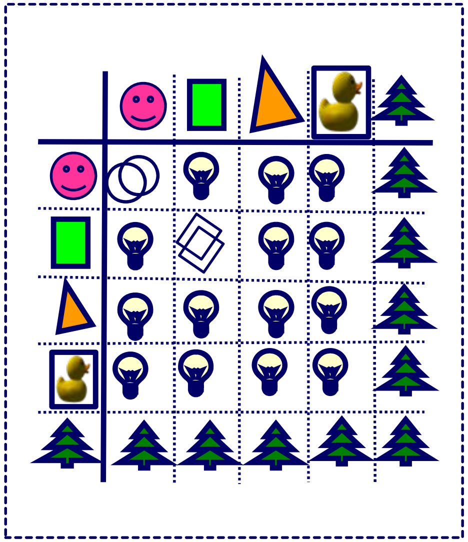
- Conventions:
  - positive distance = separation
  - negative distance = penetration
  - normal points from B to A

Narrowphase Collision Detection



Pairwise tests

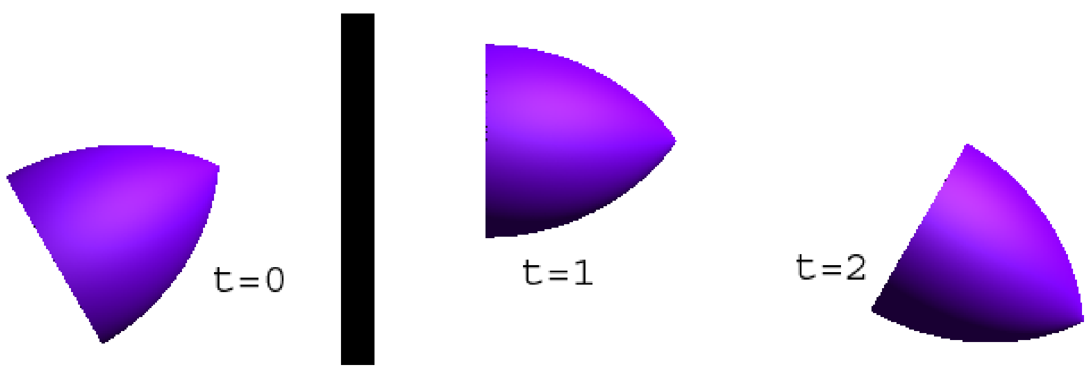
- Need to compute contact info based on both type of collision shapes



- See Gino van den Bergen's talk

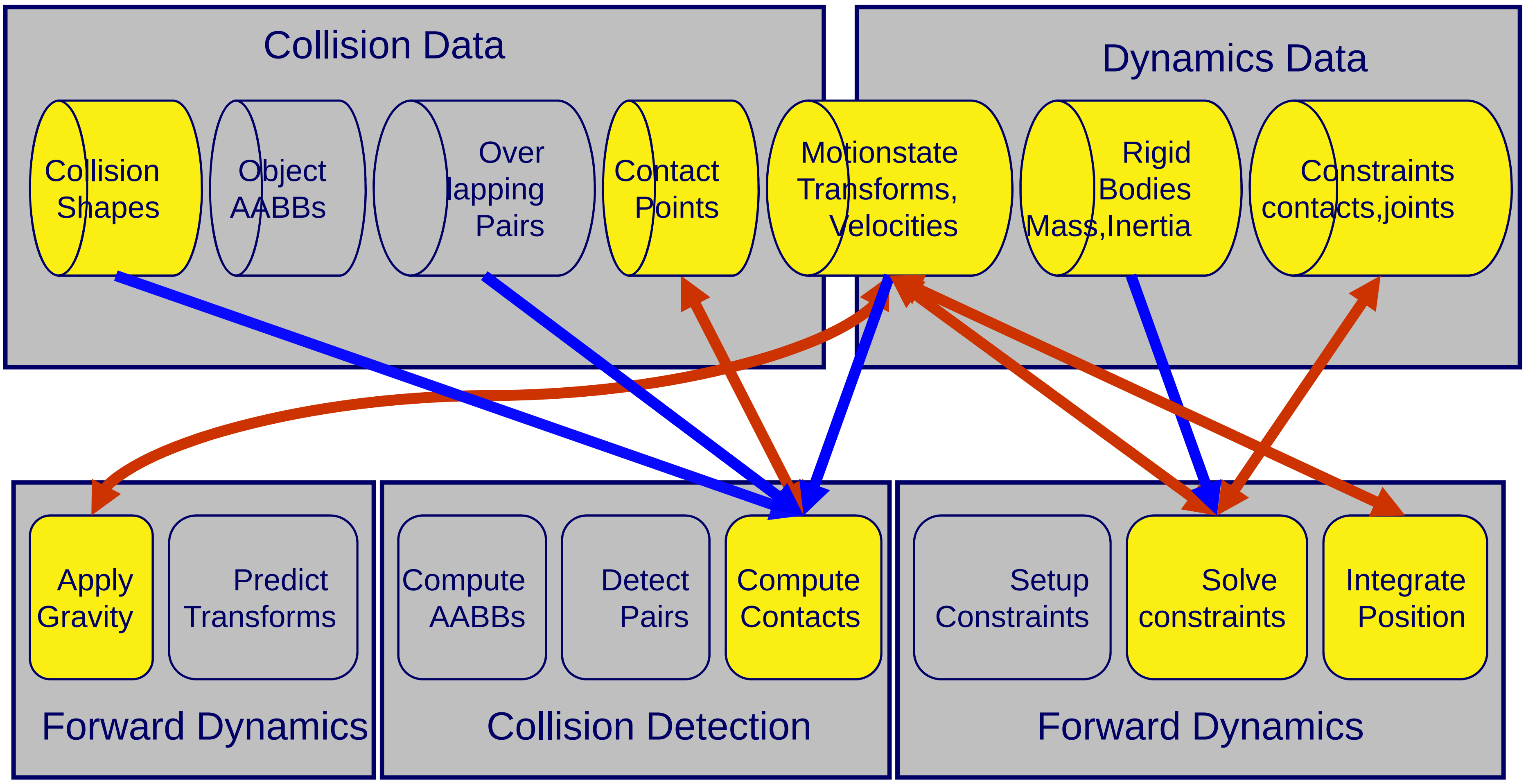
The Problem

- If collisions are checked only for the sampled moments, some collisions are missed (tunneling).
- Humans easily spot such artifacts.



- See Gino van den Bergen's talk

Brute Force Physics Pipeline



Counteracting Goals

- Fast Computations
  - Interactivity
- Robustness
  - Works no matter what
- Accuracy
  - Physical correct or plausible
- Problem size
  - The whole world

Common Optimizations

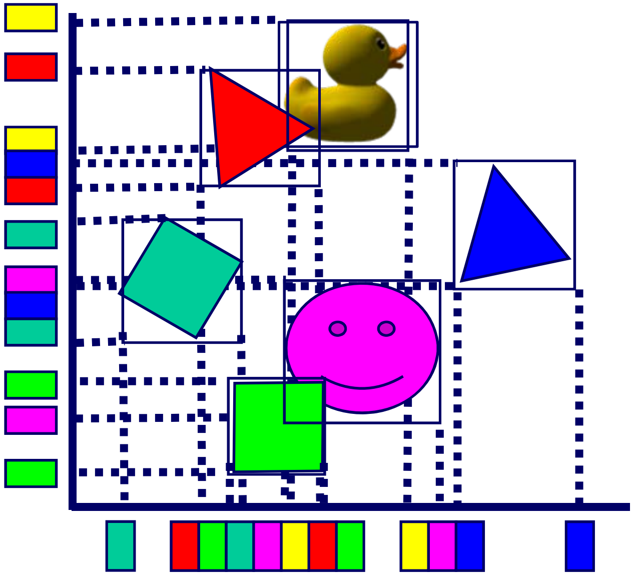
Collision Broadphase

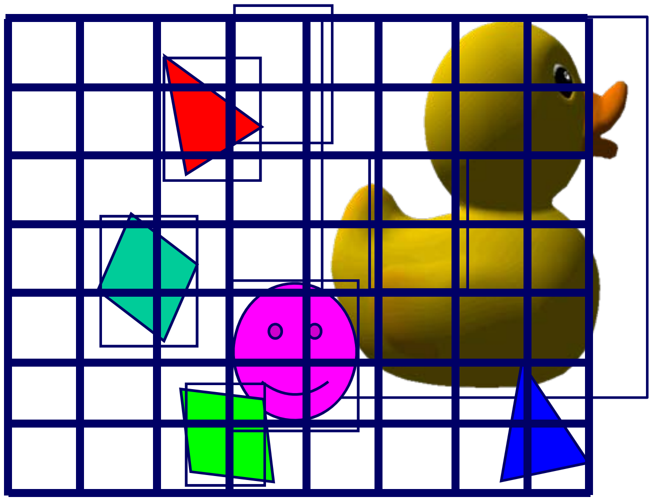
» Avoid costly narrowphase using

1. Approximate bounding shape
2. Spatial data structure and/or spatial sorting to avoid n^2 tests

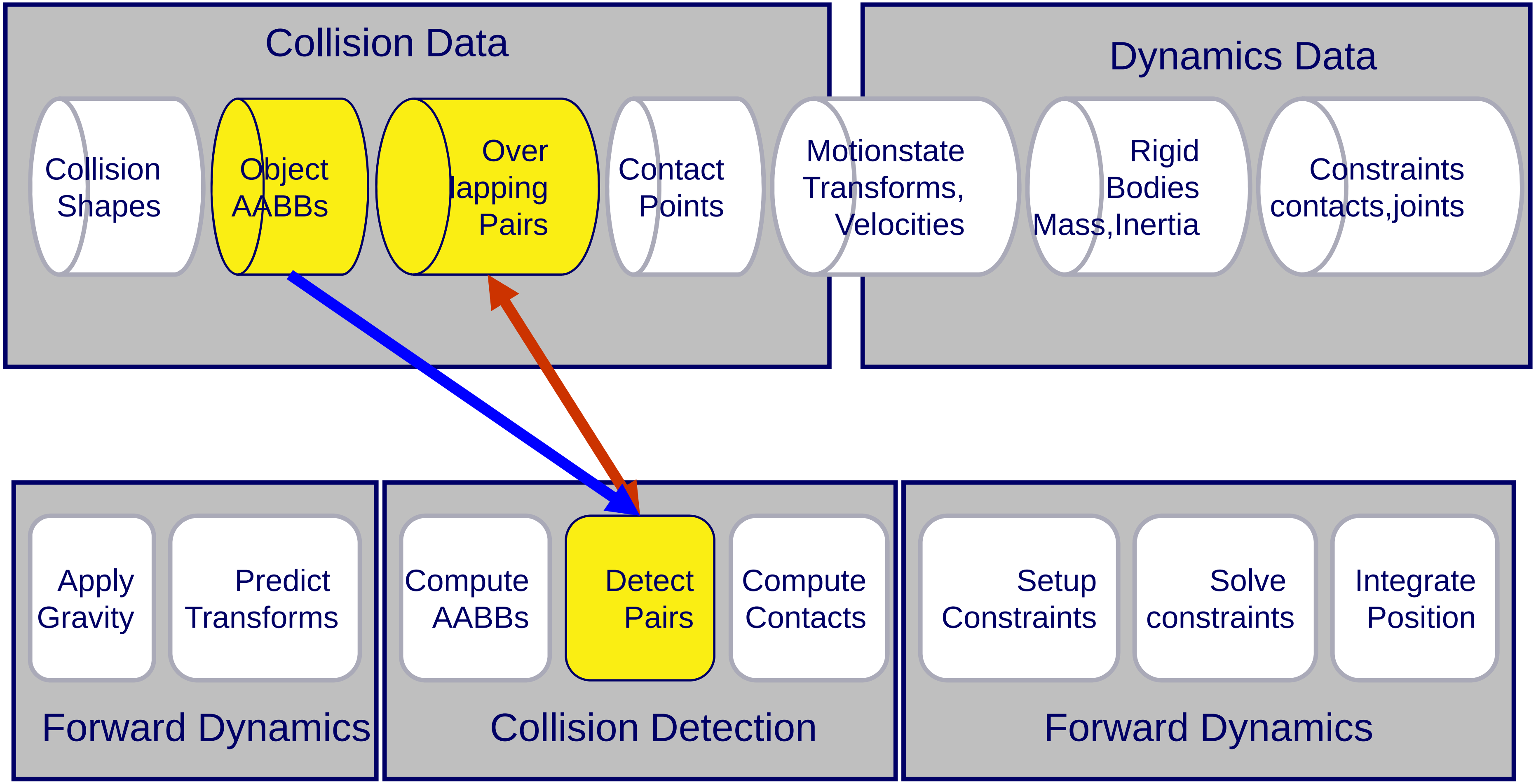
Broadphase acceleration structures

» Sweep and prune, uniform grid, dynamic BVH tree





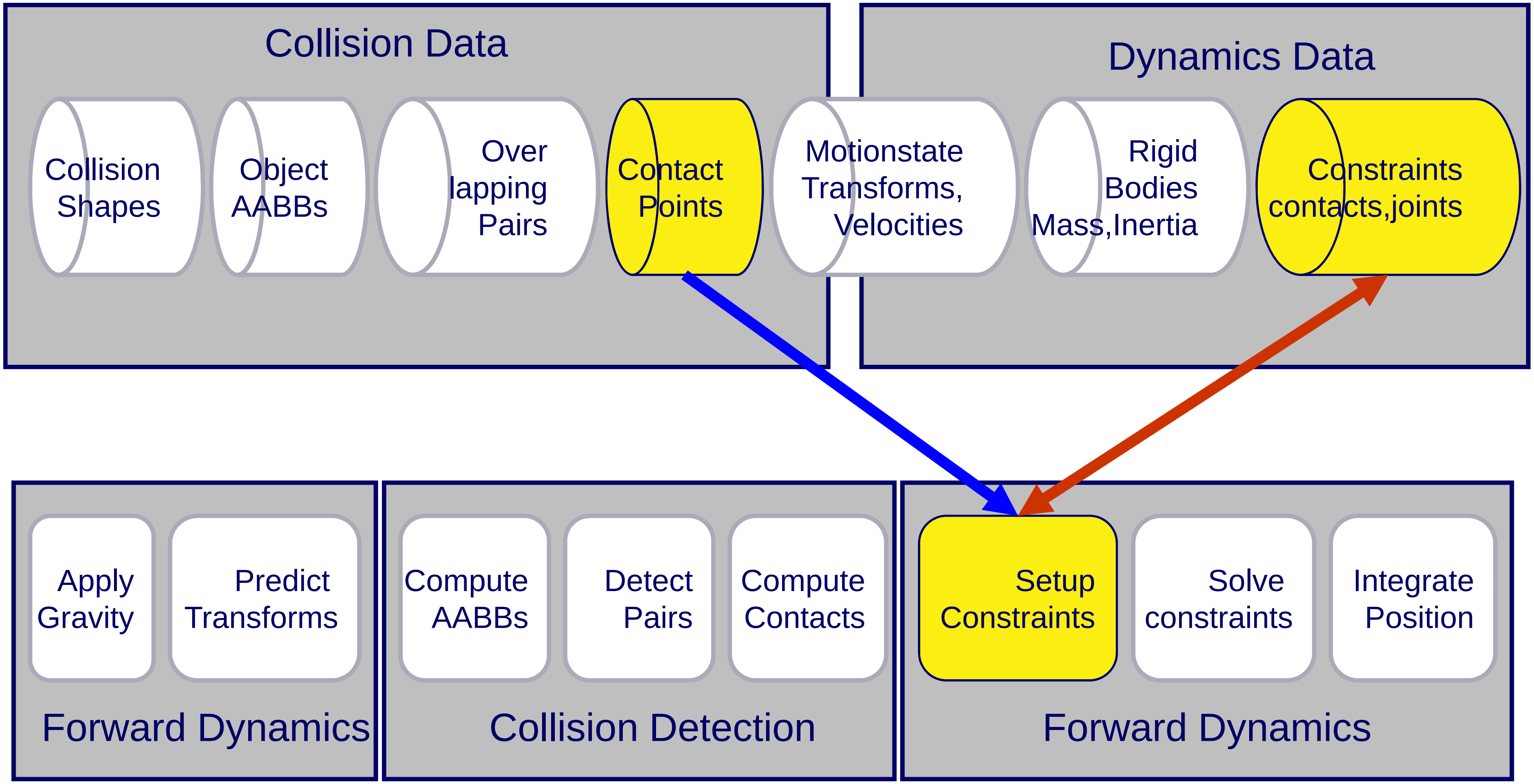
Broadphase Collision Detection



Constraint Setup

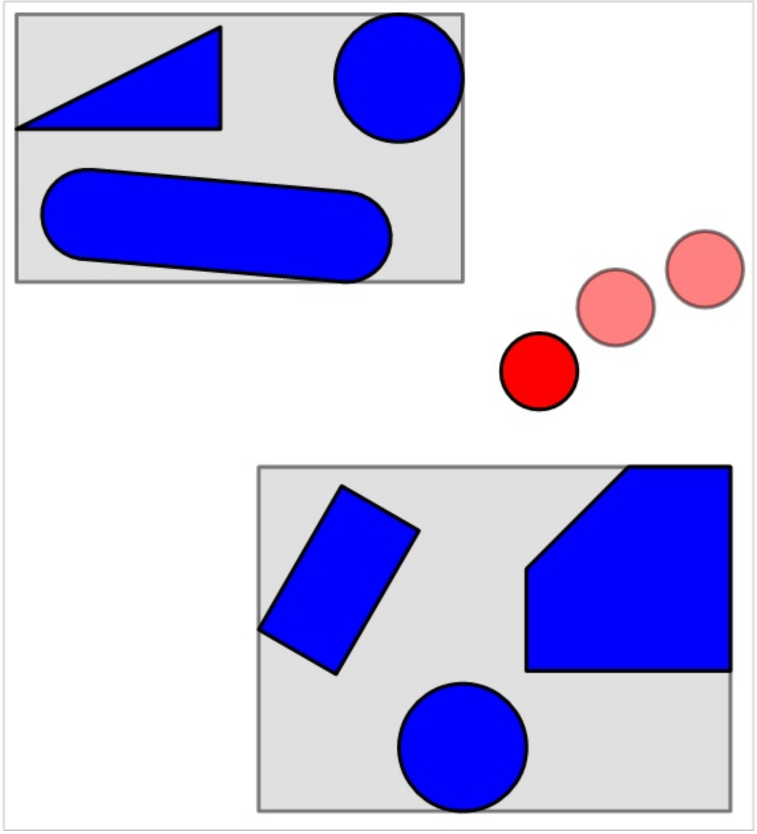
- Precompute info that is constant during all the iterations
  - Friction, jacobian data, cross products
- Split the constraints for parallel processing (batching)

Constraint Setup



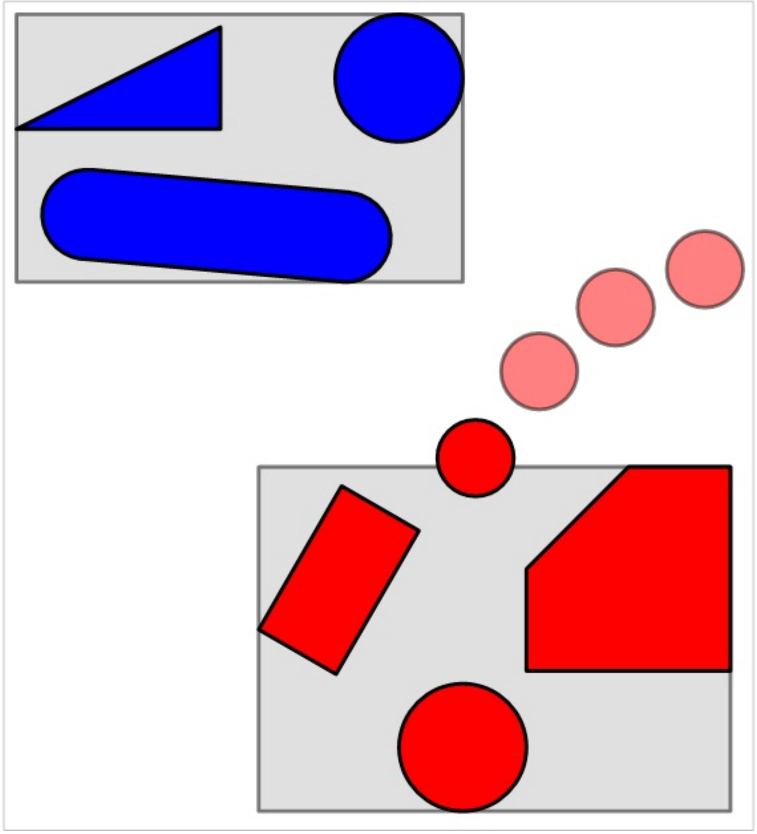
Simulation Islands

- Simulation islands can “go to sleep” when they become stable
  - i.e. when forces and motion remain unchanged
- When an object enters the island’s bounds…

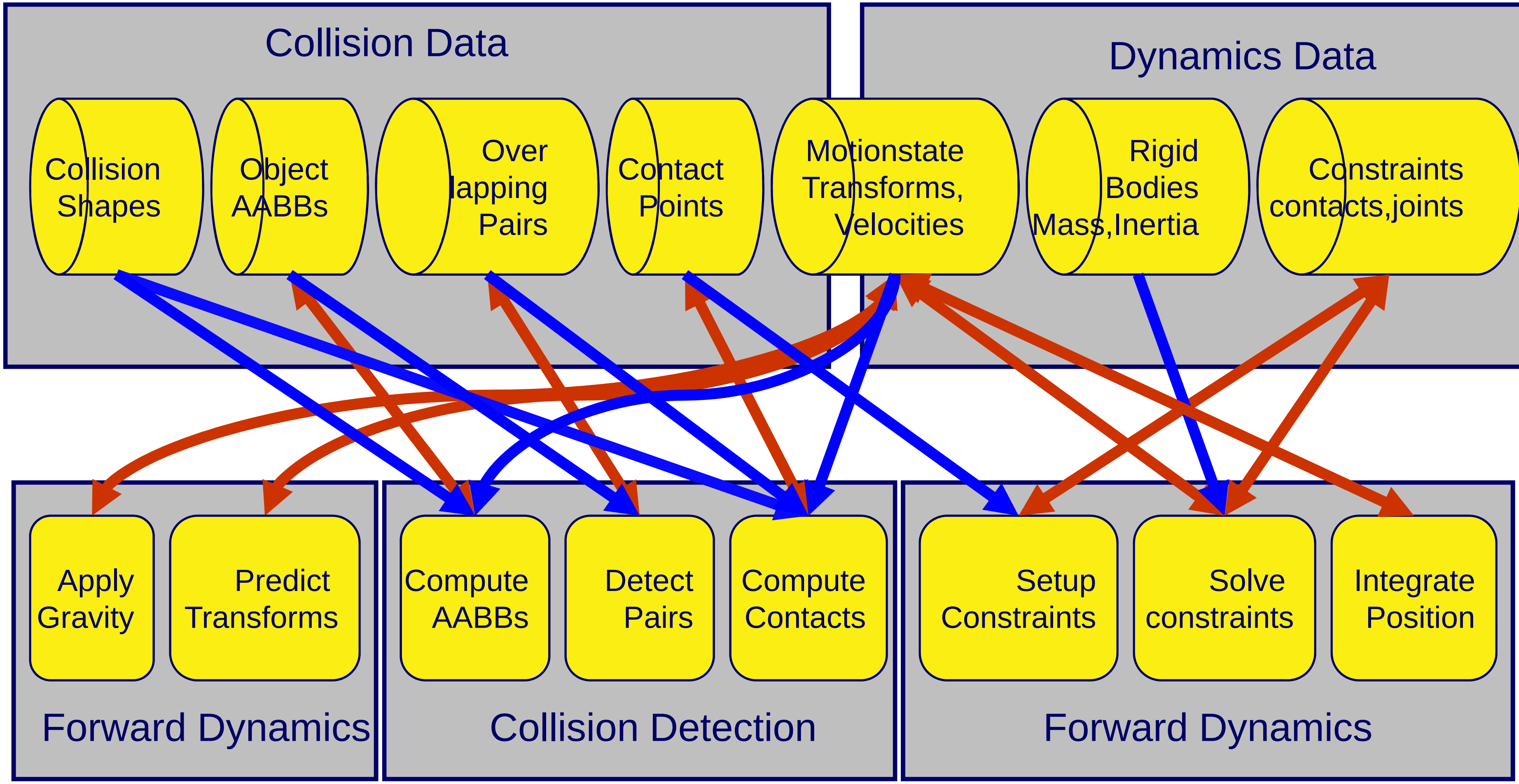


Simulation Islands

- Simulation islands can "go to sleep" when they become stable
  - i.e. when forces and motion remain unchanged
- When an object enters the island's bounds…
- …the island wakes up



Trip through the Physics Pipeline



Questions?

http://bulletphysics.org