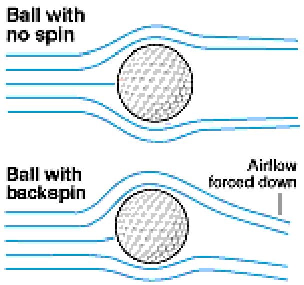
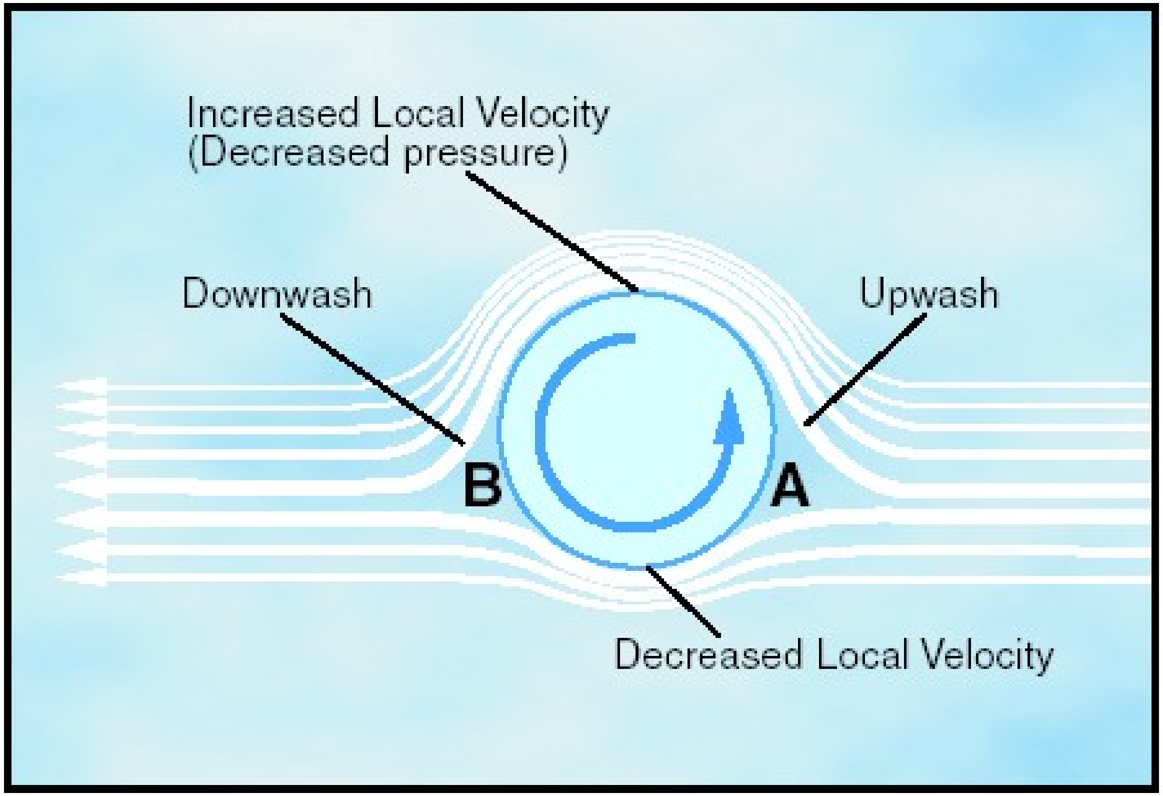
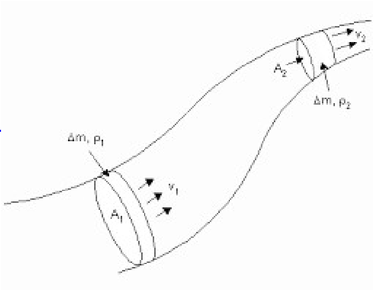
Projectiles

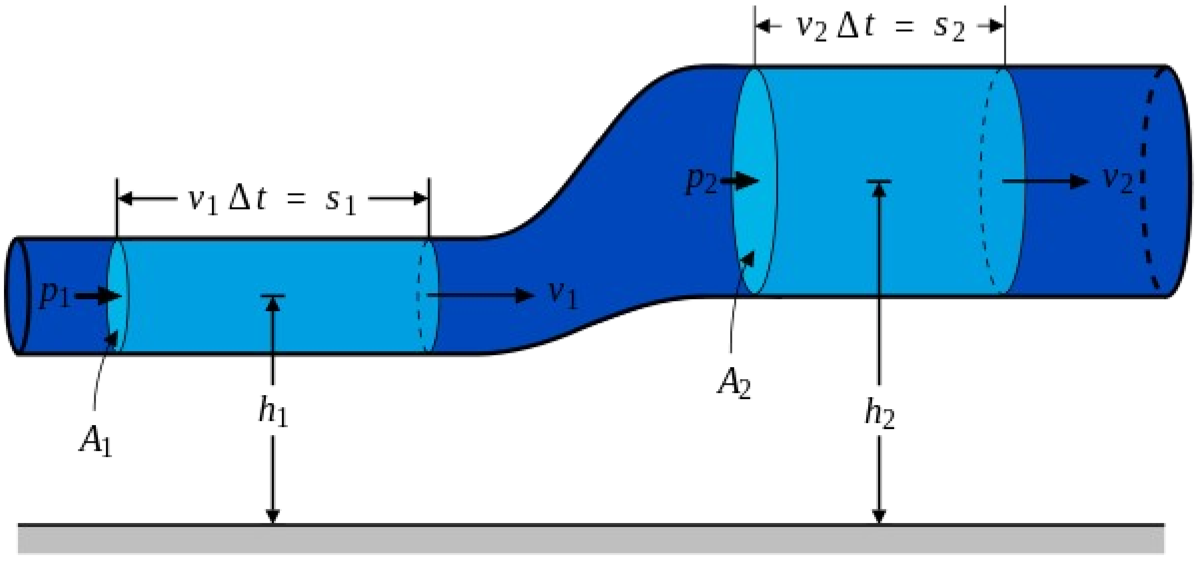
- **Projectiles**
  - **Spin Effect**





Bernoulli's equation



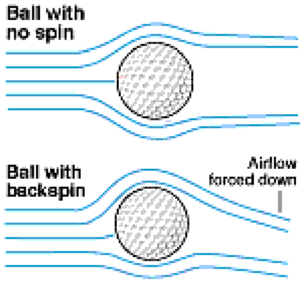


Bernoulli's equation: $p + \frac{1}{2}\rho v^2 + \rho g h = const$ ⇨ $p + \frac{1}{2}\rho v^2 = const$

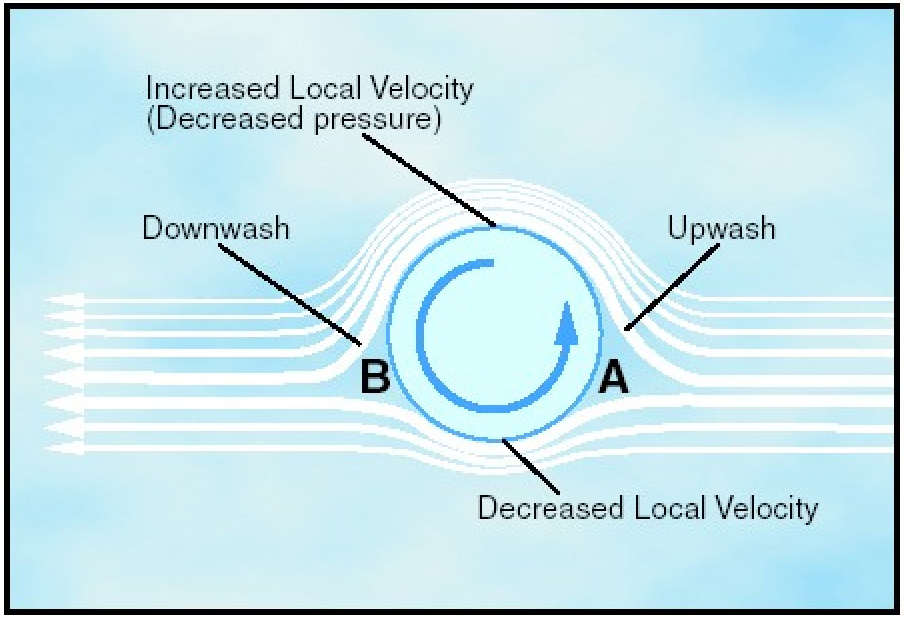
(h=const)

pressure, **the fluid density**, **velocity**, **altitude**

Projectiles



- **Projectiles**
  - **Spin Effect**



⇨ **A spinning object generates lift**

Bernoulli's equation: $p+\frac{1}{2}\rho v^2+\rho gh=const$ ⇨ $p+\frac{1}{2}\rho v^2=const$

$(h=const)$

pressure (p), **the fluid density** (ρ), **velocity** (v), **altitude** (h)

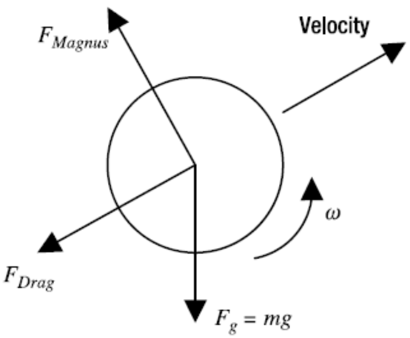
Projectiles

Spin Effect

- **Magnus force**

$$\vec{F}_M = C_L \rho \frac{v^2}{2} A \frac{[\vec{\omega}\times\vec{v}]}{|[\vec{\omega}\times\vec{v}]|}$$

The Magnus force lift coefficient



For a sphere: $C_L = \frac{r\omega}{v}$

For a cylinder: $C_L = \frac{2\pi r\omega}{v}$

- **Equation of motion**

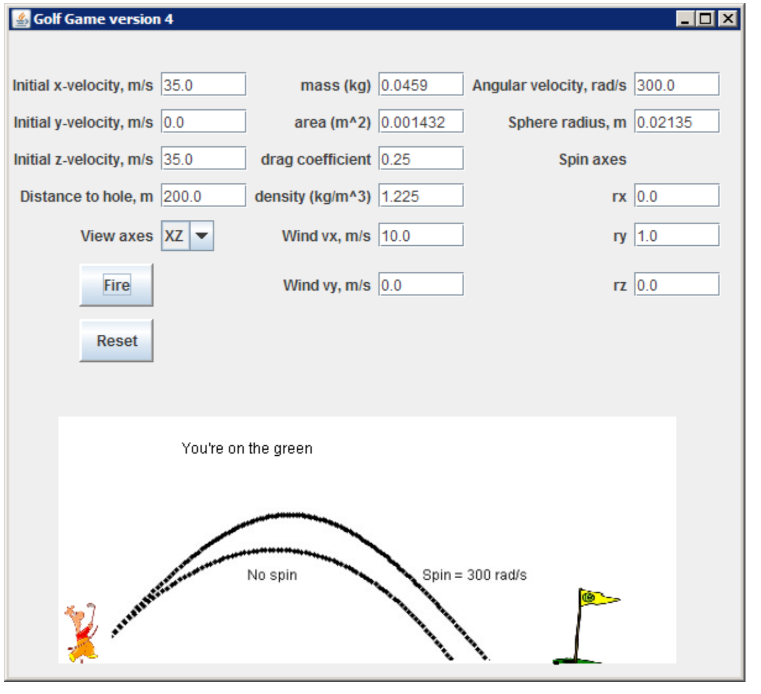
$$\vec{F} = m\,\vec{g} - |\vec{F}_D|\frac{\vec{v}}{|\vec{v}|} + C_L \rho \frac{v^2}{2} \frac{[\vec{\omega}\times\vec{v}]}{|[\vec{\omega}\times\vec{v}]|}$$

$$\Downarrow$$

$$\ddot{\vec{r}} = \vec{g} - \frac{|\vec{F}_D|}{m}\frac{\dot{\vec{r}}}{|\dot{\vec{r}}|} + C_L \rho \frac{|\dot{\vec{r}}|^2}{2m} \frac{[\dot{\vec{\theta}}\times\dot{\vec{r}}]}{|[\dot{\vec{\theta}}\times\dot{\vec{r}}]|}$$

Projectiles

- **Spin Effect**
  - **Golf Game Version 4**



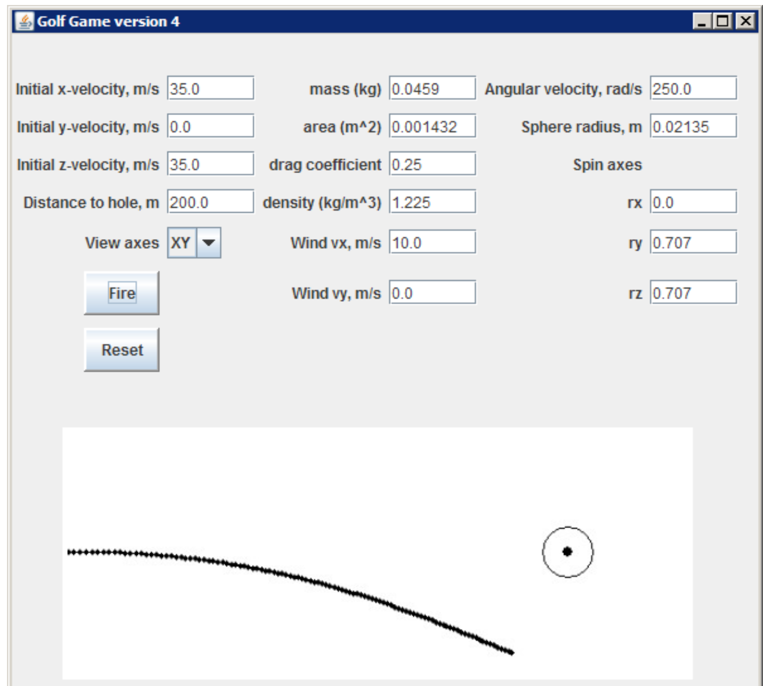
The effect of spin on golf ball flight

…Java_Code\Chapter05_Projectile\GolfGame4.java (from www.apress.com/book/downloadfile/2078)

Projectiles

Spin Effect

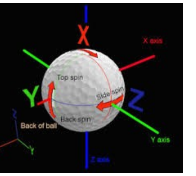
- **Golf Game Version 4**



A tilt in the spin axis causes the ball to curve

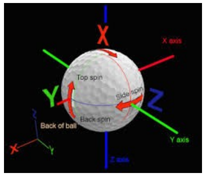
…Java_Code\Chapter05_Projectile\GolfGame4.java (from www.apress.com/book/downloadfile/2078)

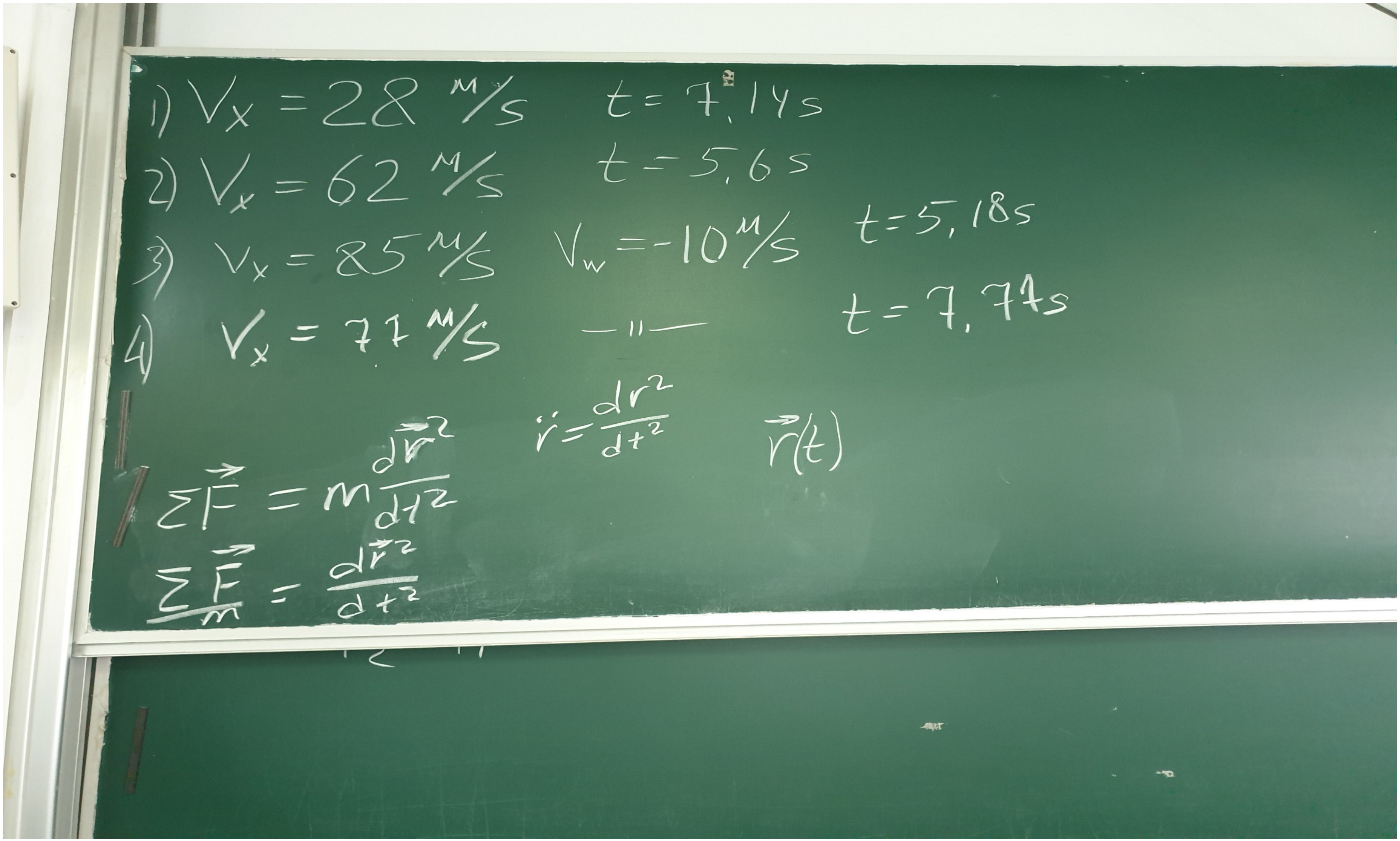
Projectiles



- **Spin Effect**
  - **Summary:**
    - An object given backspin will generate a lifting force. An object given topspin will generate a force that will push the object downwards.
    - The acceleration that results from Magnus force is inversely proportional to mass. A heavier object will experience less acceleration than a similar, lighter object.
    - The magnitude of Magnus force depends on the geometry. All other things being equal, larger objects will generate a larger Magnus force than will smaller objects.

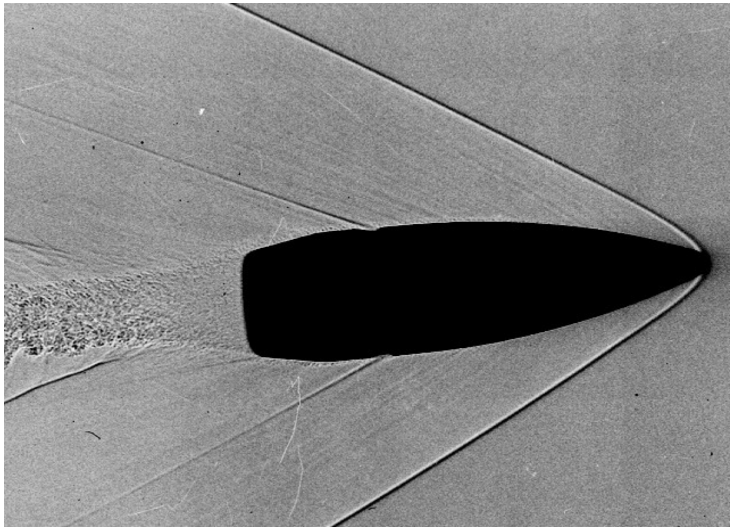
Projectiles



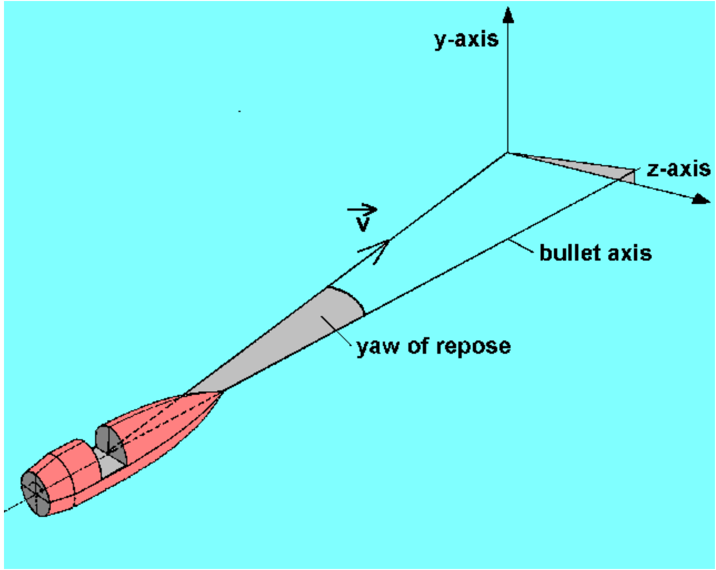


Projectiles

- **Details on Specific Types of Projectiles**
  - **Bullets**



Shadowgraph of .308 Winchester FMJ bullet traveling at approximately 850 m/s (from www.nennstiel-ruprecht.de)

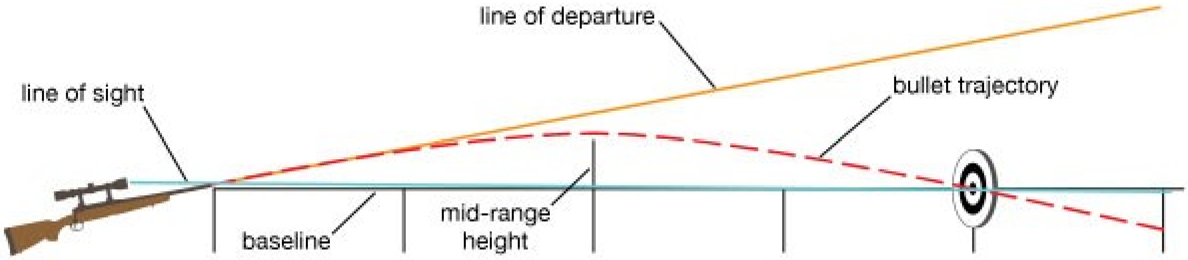


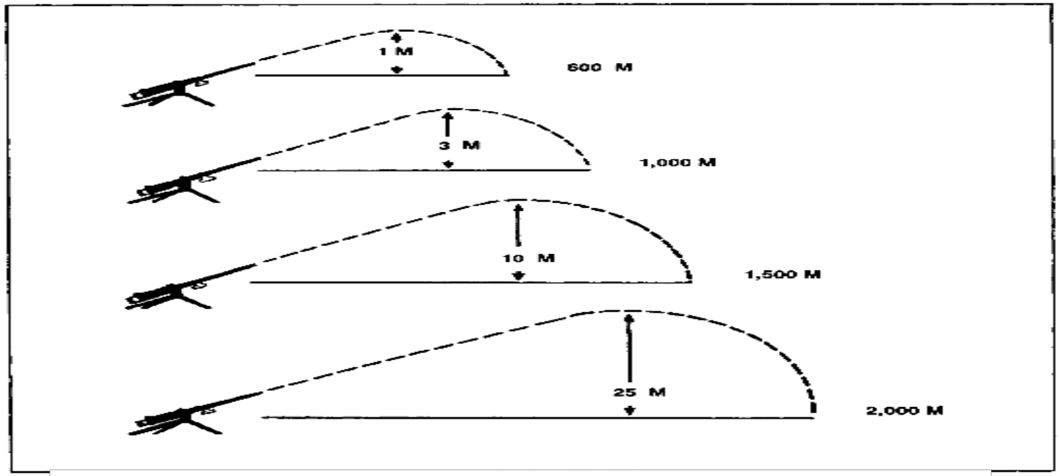
Bullets usually have a yaw angle during flight

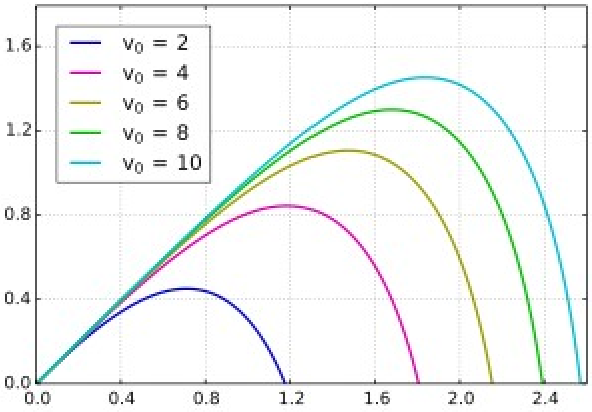
Projectiles

- **Trajectory of bullets**

Elements of a trajectory

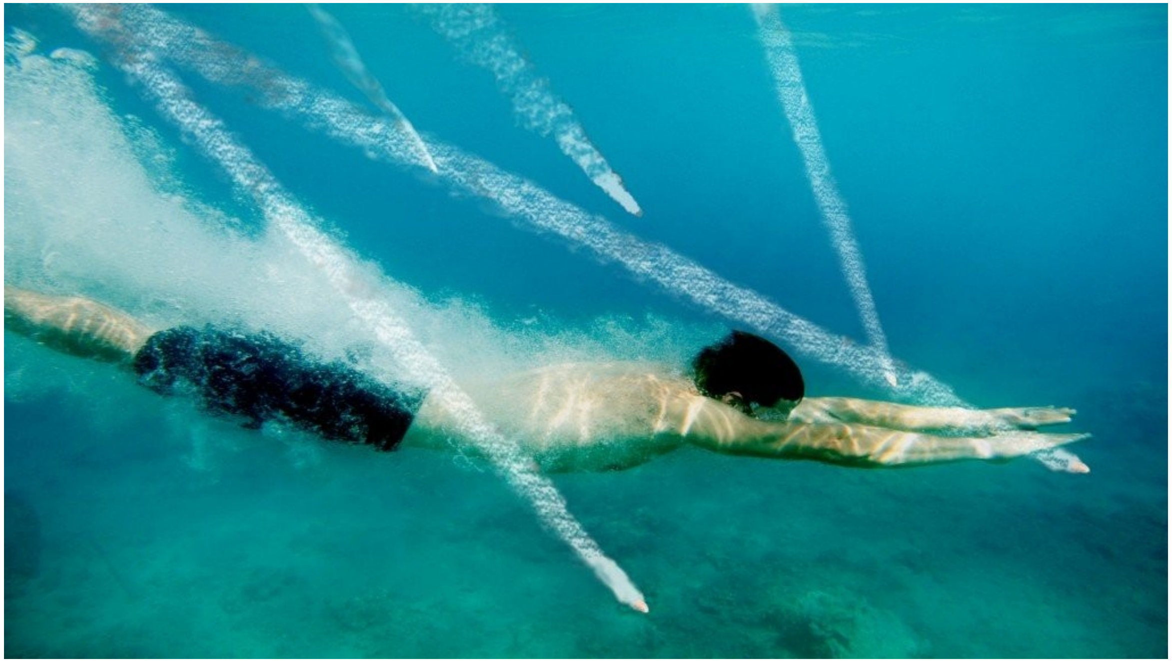






Projectiles

- **Underwater bullets**



Projectiles

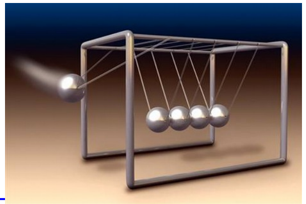
- **Details on Specific Types of Projectiles**
  - **Bullets**

Bullet	Muzzle Velocity (*m/s*)	Mass (*gm*)	Diameter (*mm*)
.22 Long rifle round nose	330	2.6	5.6
.32 ACP FMJ round nose	262	4.7	7.84
.357 Magnum	506	7.3	9
.38 ACP FMJ round nose	322	6.2	9.7
9 *mm* FMJ	341–373	8.0	9
9 *mm* FMJ high velocity	436	8.2	9
.44 Magnum	436	15.6	11.2
M74 (5.45 *mm*)	917	3.44	5.64
M80 (7.62 *mm* FMJ)	877	9.5	7.82
M2 .30 armor piercing	869	10.8	7.7

Collisions

- **Specific topics**
  - **Linear momentum and impulse**
  - **Conservation of linear momentum**
  - **Two-body linear collisions**
  - **Elastic and inelastic collisions**
  - **Determining when a collision occurs**
  - **Angular momentum and impulse**
  - **Conservation of angular momentum**
  - **General two-body collisions**

Collisions

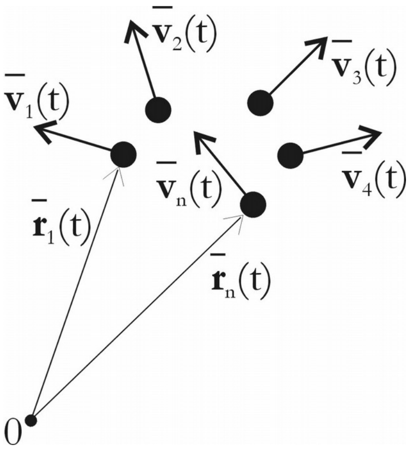


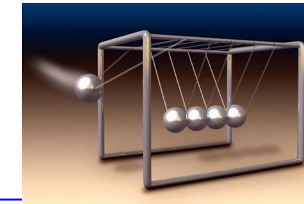
- **Linear momentum and impulse**

The linear momentum $\vec{P} = m\vec{v}$

Newton's 2nd law $\vec{F} = \frac{d\vec{P}}{dt}$

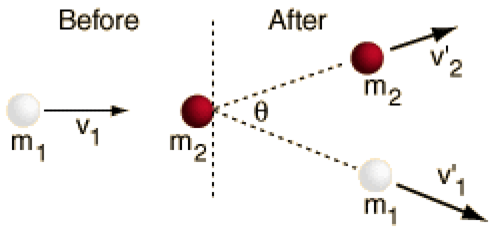
Conservation of Linear Momentum $\sum_i \vec{P}_i = const$





Collisions

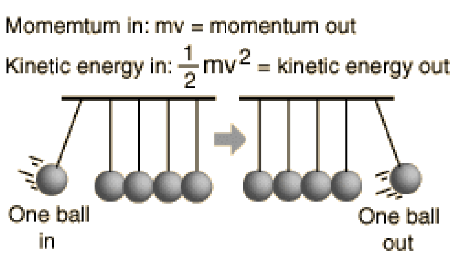
Elastic Collisions

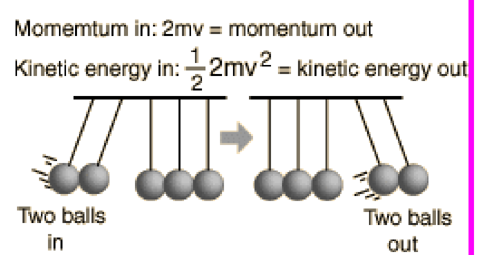


A perfectly elastic collision is defined as one in which there is no loss of kinetic energy in the collision.

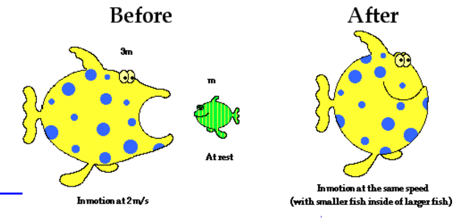
$$\sum_i \frac{m_i^- v_i^{-2}}{2} = \sum_i \frac{m_i^+ v_i^{+2}}{2}$$

Before After

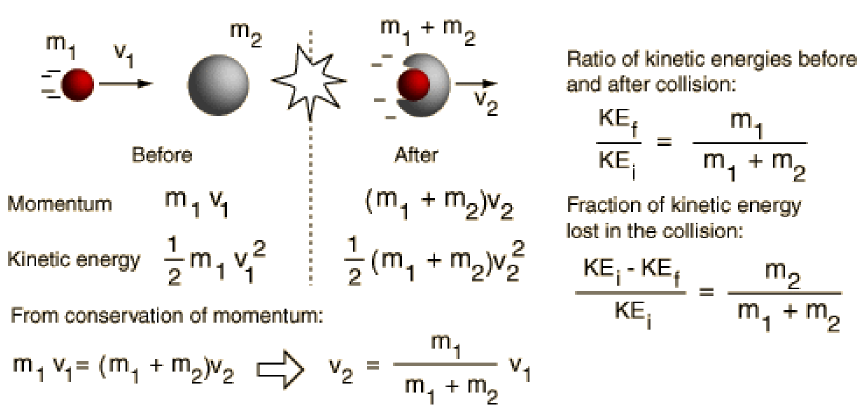


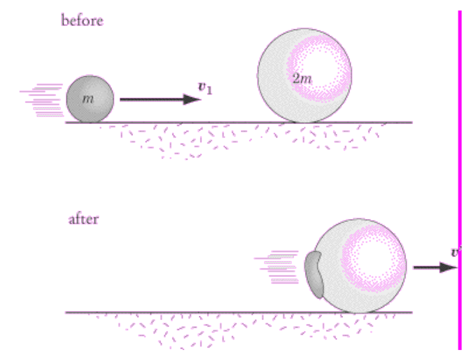


Collisions



Inelastic Collisions





$$\sum_i \frac{m_i^- {v_i^-}^2}{2} > \sum_i \frac{m_i^+ {v_i^+}^2}{2}$$

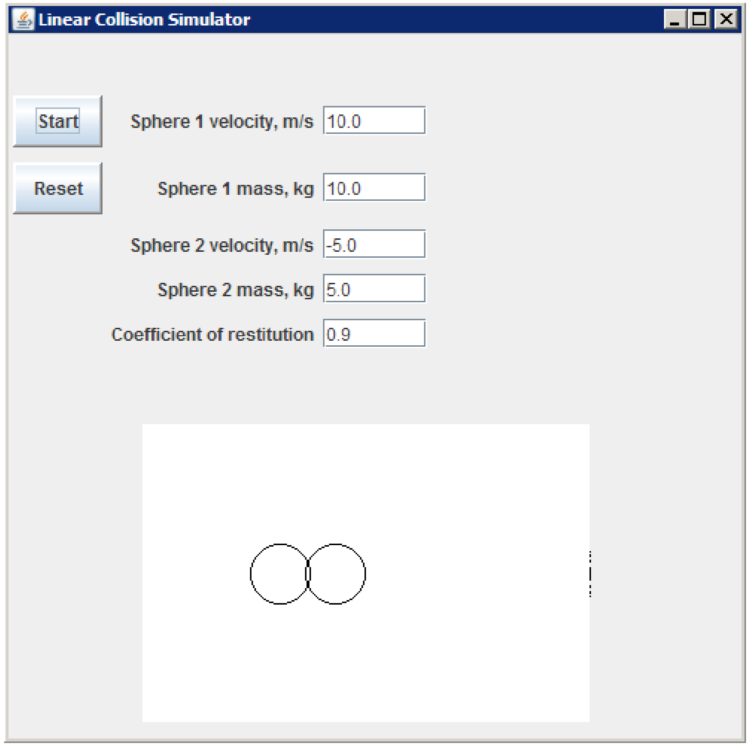
Before the collision   After the collision

An inelastic collision is one in which part of the kinetic energy is changed to some other form of energy in the collision

Collisions

Elastic and Inelastic Collisions

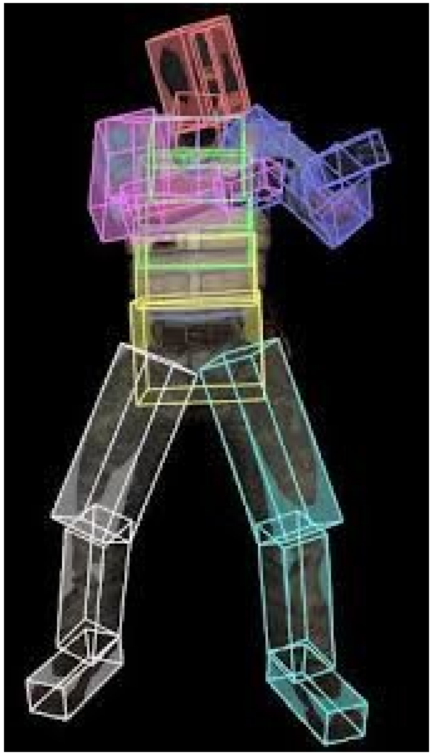
coefficient of restitution $e = \frac{|\vec{v}_{1^+} - \vec{v}_{2^+}|}{|\vec{v}_{1^-} - \vec{v}_{2^-}|}$ $0 \le e \le 1$



…Java_Code\Chapter06_Collision\SphereCollision.java (from www.apress.com/book/downloadfile/2078)

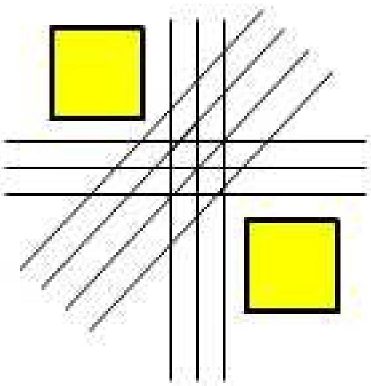
Collisions

- **Collision detection**



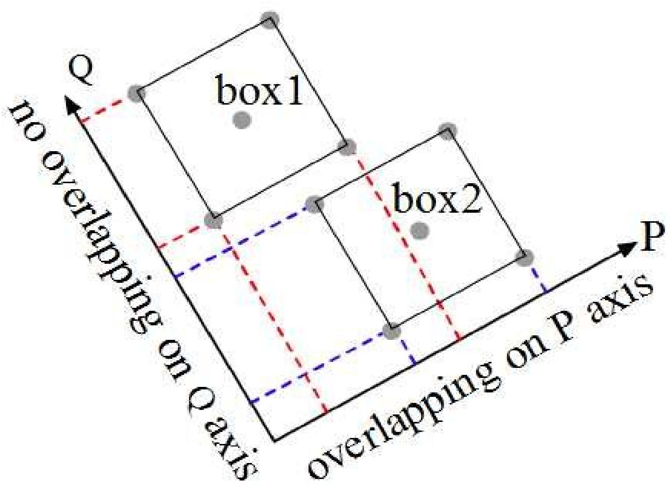
Collisions

- **Collision detection**
  - Separating axis



if you are able to draw a line to separate two polygons, then they do not collide.

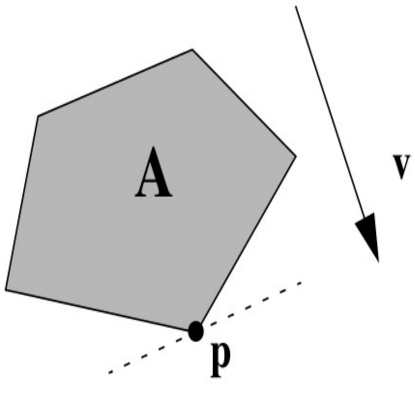
Projection Along an Arbitrary Axis



Collisions

- **Collision detection**
  - Gilbert-Johnson-Keerthi (GJK) algorithm
    - Suppor mapping

SA(v) = pi where i maximizes v•pi



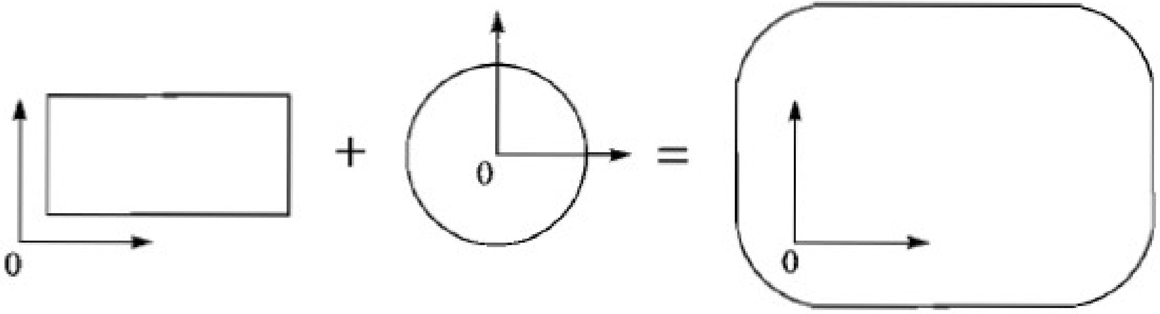
The support mapping of the shape A along the vector v is the point p

Collisions

Collision detection

– Gilbert-Johnson-Keerthi (GJK) algorithm
  - Minkowski Addition

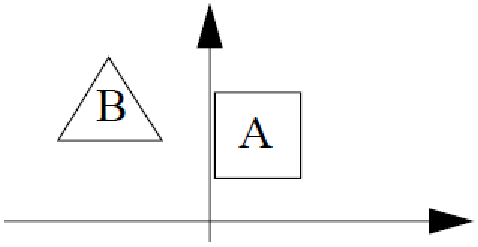
$$A + B = \{\mathbf{x} + \mathbf{y} : \mathbf{x} \in A, \mathbf{y} \in B\}.$$



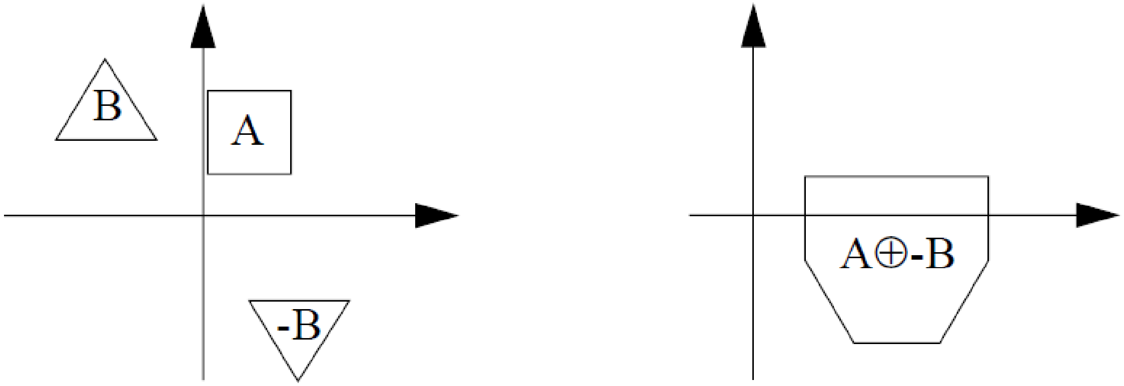
The Minkowski sum of a box and a sphere.

Collisions

- **Collision detection**
  - Gilbert-Johnson-Keerthi (GJK) algorithm



The two shapes for our GJK example.

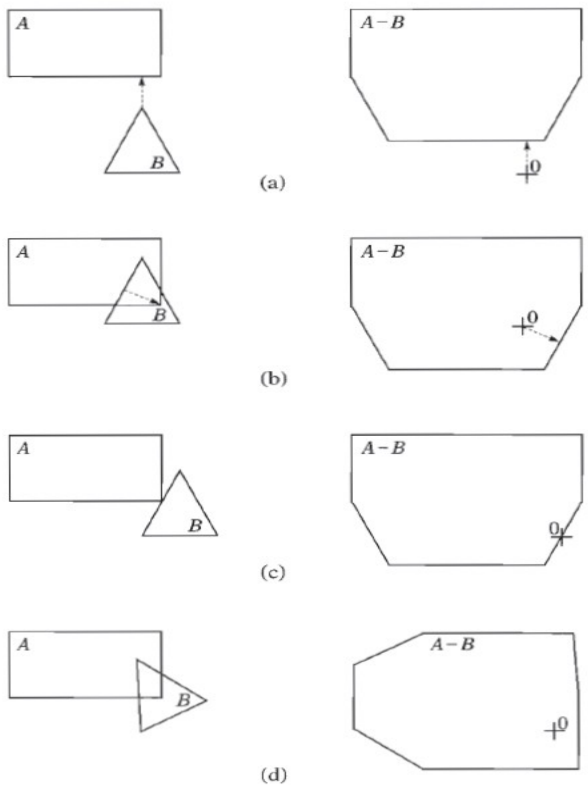


The negated shape -B and the Minkowski sum A⊕(-B).

Collisions

Collision detection

- Gilbert-Johnson-Keerthi (GJK) algorithm
  - Minkowski ”difference” (Configuration space obstacle (CSO))

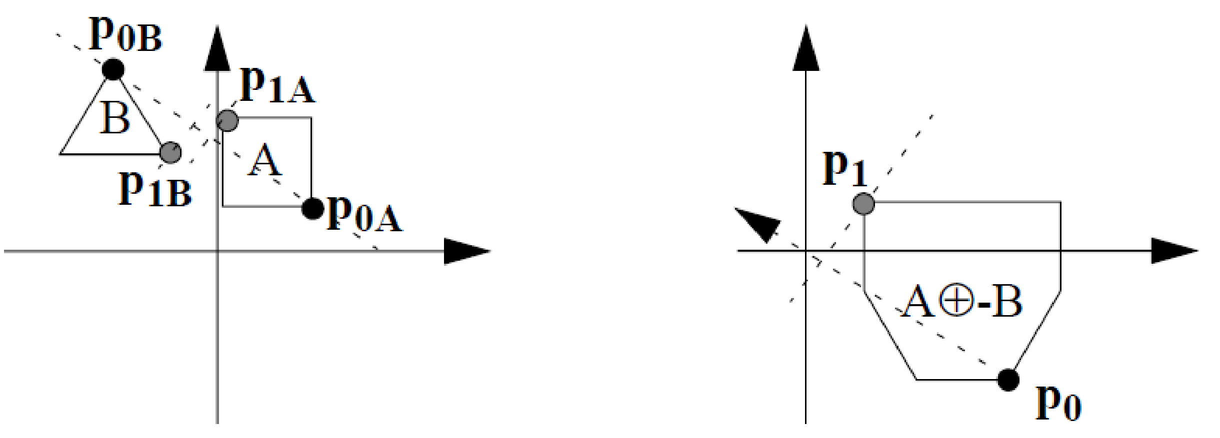


A pair of convex objects and the corresponding CSO. (a) Nonintersecting: The origin is outside the CSO. The arrow denotes the distance. (b) Intersecting: The origin is inside the CSO. The arrow denotes the penetration depth. (c) After a translation of B over the penetration depth vector, the objects are in contact. The origin lies on the boundary of the CSO. (d) After a rotation of B, the shape of the CSO changes.

Collisions

- **Collision detection**
  - Gilbert-Johnson-Keerthi (GJK) algorithm

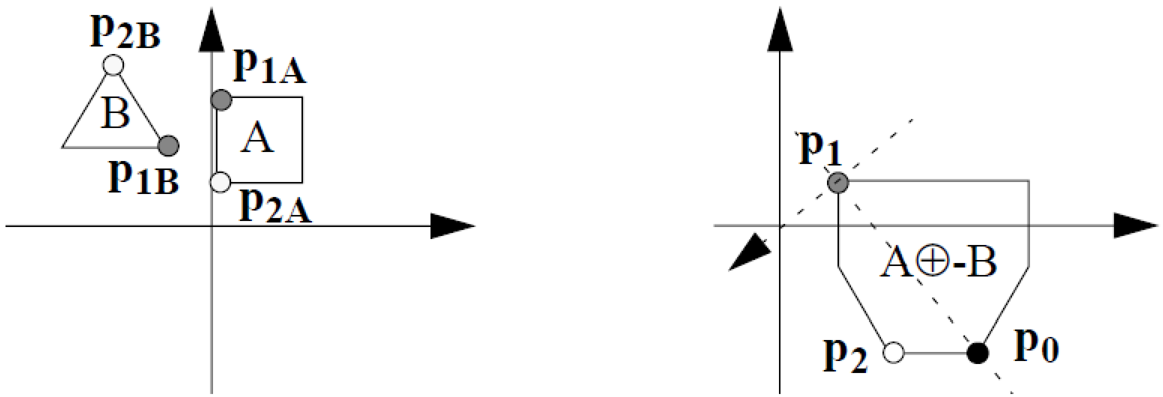
$$S_{A\oplus -B}(\mathbf{p_0}) = S_A(\mathbf{p_{0A}} - \mathbf{p_{0B}}) - S_B(\mathbf{p_{0B}} - \mathbf{p_{0A}}) = S_A(\mathbf{v}) - S_B(-\mathbf{v})$$



The first step of the GJK algorithm, on separate objects (left) and combined (right)

Collisions

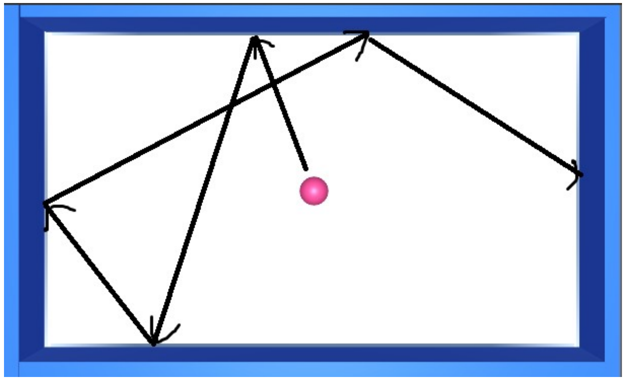
- **Collision detection**
  - Gilbert-Johnson-Keerthi (GJK) algorithm



The first step of the GJK algorithm, on separate objects (left) and combined (right)

Collisions

- **1) Collision of a ball with a wall**

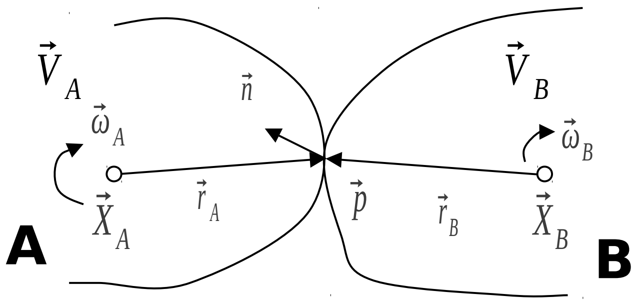


- **2) Colission of two (several) balls**

Consider both tasks in the
coordinate free (vector) form

Collisions

Collision response



$$\vec{r}_A = \vec{p} - X_A(t) \qquad \vec{r}_B = \vec{p} - X_B(t)$$

$$\vec{v}_{pA} = \vec{v}_A^- + \vec{\omega}_A^- \times \vec{r}_A \qquad \vec{v}_{pB} = \vec{v}_B^- + \vec{\omega}_B^- \times \vec{r}_B$$

$$\vec{v}_{rel} = \vec{v}_{pA} - \vec{v}_{pB} \qquad \vec{v}_{rel}^- = (\vec{v}_{pA} - \vec{v}_{pB}) \cdot \vec{n}$$

$$\vec{P}_{imp} = j\vec{n}$$

$$\vec{\omega}_A^+ = \vec{\omega}_A^- + \hat{I}_A^{-1}[\vec{r}_A \times j\vec{n}] = \vec{\omega}_A^- + j\,\hat{I}_A^{-1}[\vec{r}_A \times \vec{n}]$$

$$\vec{v}_A^+ = \vec{v}_A^- + j\,\vec{n}/m_A \qquad \vec{v}_B^+ = \vec{v}_B^- - j\,\vec{n}/m_B$$

$$\vec{\omega}_B^+ = \vec{\omega}_B^- - \hat{I}_B^{-1}[\vec{r}_B \times j\vec{n}] = \vec{\omega}_B^- - j\hat{I}_B^{-1}[\vec{r}_B \times \vec{n}]$$

$$v_{rel}^+ = -\varepsilon \cdot v_{rel}^-$$

$$v_{rel}^- = (\vec{v}_{pA}^- - \vec{v}_{pB}^-) \cdot \vec{n} = (\vec{v}_A^- + \vec{\omega}_A^- \times \vec{r}_A - \vec{v}_B^- - \vec{\omega}_B^- \times \vec{r}_B) \cdot \vec{n}$$

$$v_{rel}^+ = (\vec{v}_{pA}^+ - \vec{v}_{pB}^+) \cdot \vec{n} = (\vec{v}_A^+ + \vec{\omega}_A^+ \times \vec{r}_A - \vec{v}_B^+ - \vec{\omega}_B^+ \times \vec{r}_B) \cdot \vec{n} =$$

$$= v_{rel}^- + j\left(\vec{n}/m_A + \hat{I}_A^{-1}\left[[\vec{r}_A \times \vec{n}] \times \vec{r}_A\right] + \vec{n}/m_B + \hat{I}_B^{-1}\left[[\vec{r}_B \times \vec{n}] \times \vec{r}_B\right]\right) \cdot \vec{n}$$

$$-(\varepsilon + 1)v_{rel}^- = j/m_A + j/m_B + j\left(\hat{I}_A^{-1}\left[[\vec{r}_A \times \vec{n}] \times \vec{r}_A\right] + \hat{I}_B^{-1}\left[[\vec{r}_B \times \vec{n}] \times \vec{r}_B\right]\right) \cdot \vec{n}$$

$$j = \frac{-(\varepsilon + 1)v_{rel}^-}{1/m_A + 1/m_B + \left(\hat{I}_A^{-1}\left[[\vec{r}_A \times \vec{n}] \times \vec{r}_A\right] + \hat{I}_B^{-1}\left[[\vec{r}_B \times \vec{n}] \times \vec{r}_B\right]\right) \cdot \vec{n}}$$

Collisions

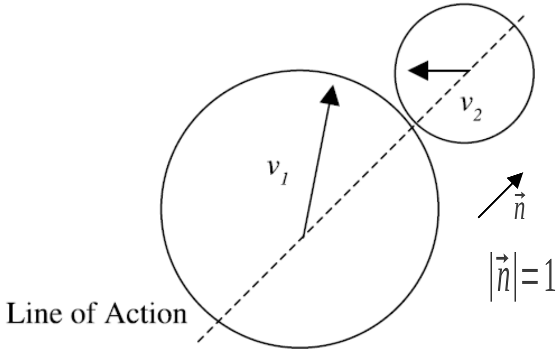
The Simulation Loop Pseudocode

```
while(simulating) {
   DeltaTime = CurrentTime - LastTime
   while(LastTime < CurrentTime) {
        calculate all forces and torques @ LastTime+DeltaTime
        compute linear and angular accelerations @ LastTime+DeltaTime
        integrate accelerations and velocities over DeltaTime @ LastTime+DeltaTime
        if(objects are interpenetrating) { subdivide DeltaTime}
                                        else {
                                             if(objects are colliding) {
                                                 resolve collisions using Eqs}
                                             LastTime = LastTime + DeltaTime
                                             DeltaTime = CurrentTime - LastTime
                                             update positions and velocities
                                             }
                                           }
   draw objects in current positions
}
```

Collisions

Collisions with Friction

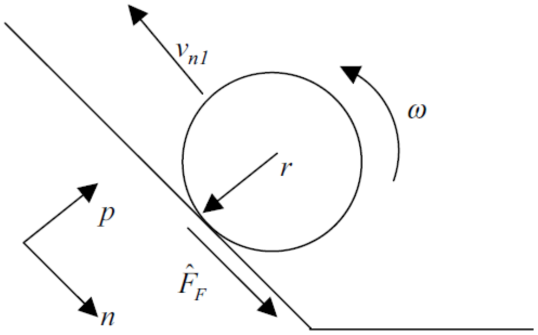
When two objects collide obliquely, they will slide against each other for a brief period of time.



The frictional impulse $\hat{F}_{\mu}=\mu\vec{n}\cdot(\vec{P}_b-\vec{P}_a)$

$$\hat{F}_{\mu}=-\frac{\hat{I}_1\Delta\omega_1}{r_1}\qquad \hat{F}_{\mu}=-\frac{\hat{I}_2\Delta\omega_2}{r_2}$$

The frictional impulse acts in the direction normal to the line of action and causes rotations of the objects



Collisions

- **Summary**

  - The change in velocity that results from a collision can be characterized by a linear or angular impulse.

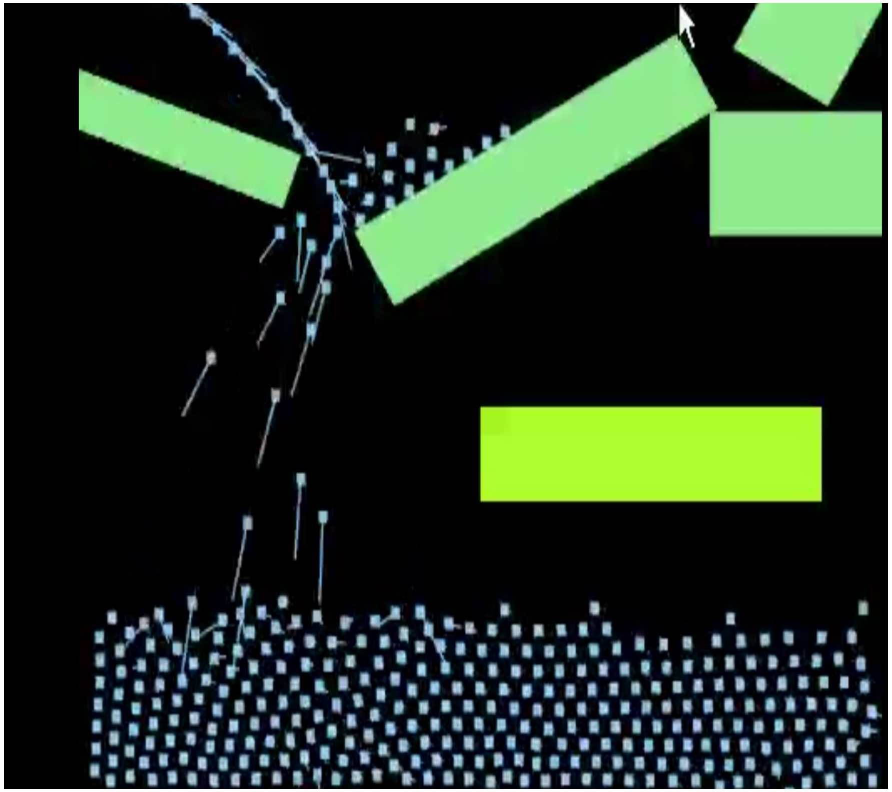
  - The post-collision velocities of two objects after a collision can be determined from the principle of conservation of momentum and the coefficient of restitution for the collision.

  - For frictionless collisions, only the velocity in the direction of the line of action of a collision is affected by the collision. The other velocity components normal to the line of action are unchanged.

  - For collisions that involve friction, the resulting frictional impulse reduces the magnitude of the velocity in the direction normal to the line of action and causes the objects to spin.

Fluid dynamics

Example of a solution of the Euler (or Navier–Stokes) equation



Fluid dynamics

Euler equations

$$\frac{\partial \rho}{\partial t} + \sum_{i=1}^{3} \frac{\partial(\rho u_i)}{\partial x_i} = 0,$$

$$\frac{\partial(\rho u_j)}{\partial t} + \sum_{i=1}^{3} \frac{\partial(\rho u_i u_j)}{\partial x_i} + \frac{\partial p}{\partial x_j} = 0,$$

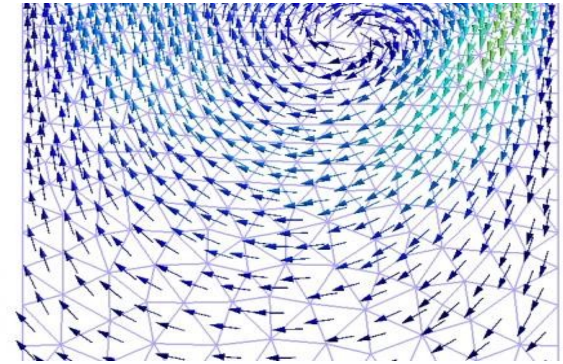
$$\frac{\partial E}{\partial t} + \sum_{i=1}^{3} \frac{\partial((E+p)u_i)}{\partial x_i} = 0,$$

Velocity components

Mass density

Pressure

Energy

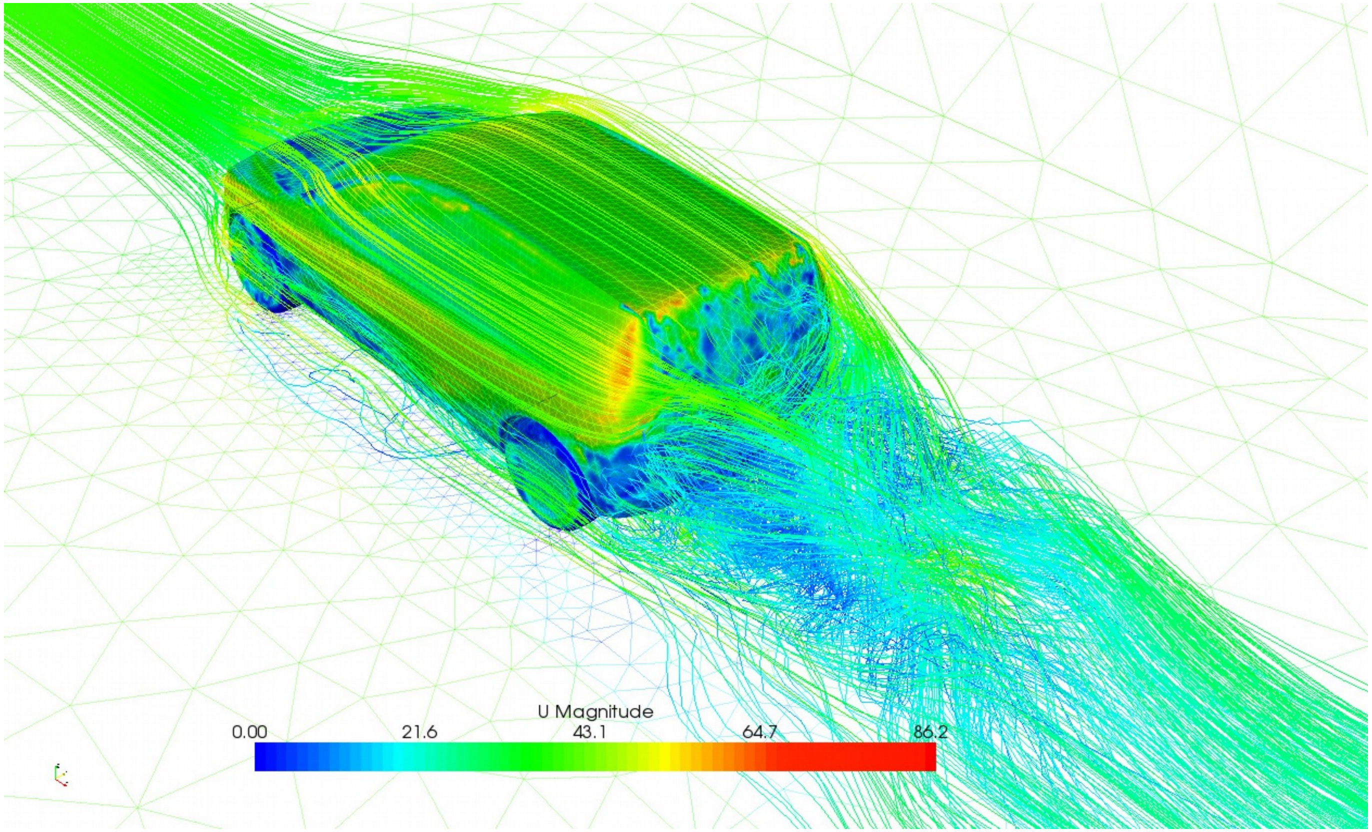


Navier–Stokes equation

$$\frac{\partial \vec{v}}{\partial t} = -(\vec{v}\cdot\nabla)\vec{v} + \nu\Delta\vec{v} - \frac{1}{\rho}\nabla p + \vec{f},$$

Fluid dynamics

Example of a solution of the Navier–Stokes equation

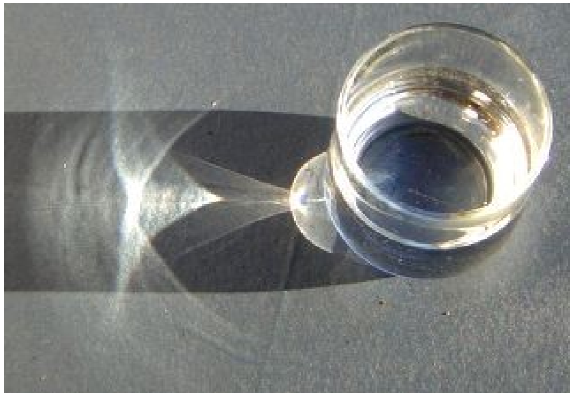


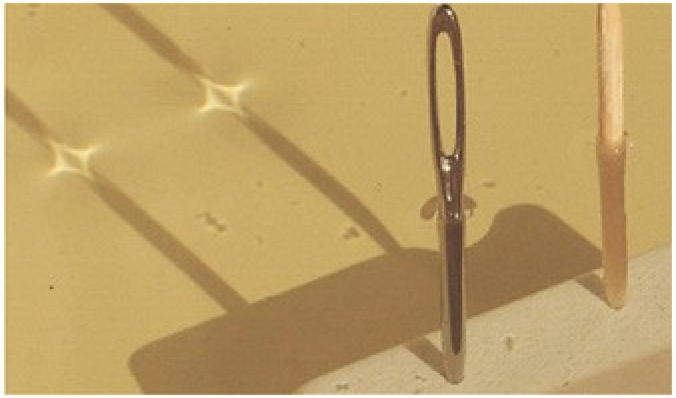
Fluid dynamics

Caustic (or caustic network) is the envelope of light rays reflected or refracted by a curved surface or object

Simulation of a liquid

Examples of caustics

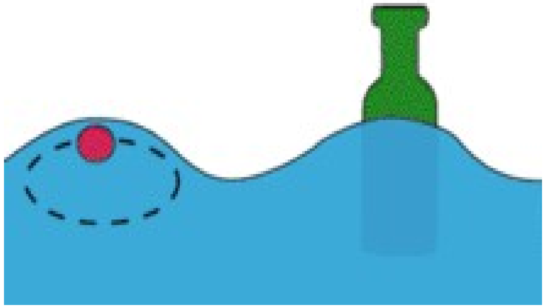




Simulation of a liquid

Wake

Wind/gravitation

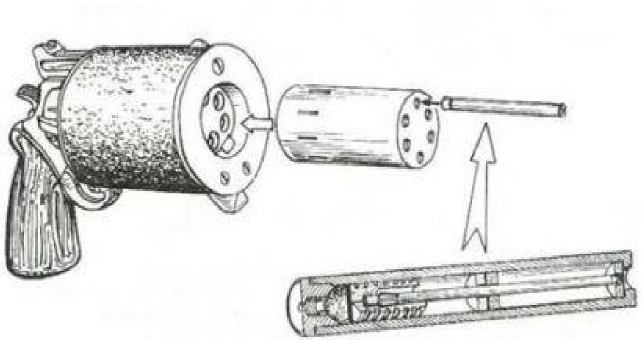


Continuity equation:

$$\rho_1 v_1 \cdot S_1 = \rho_2 v_2 \cdot S_2$$

Simulation of a liquid

Underwater weapon

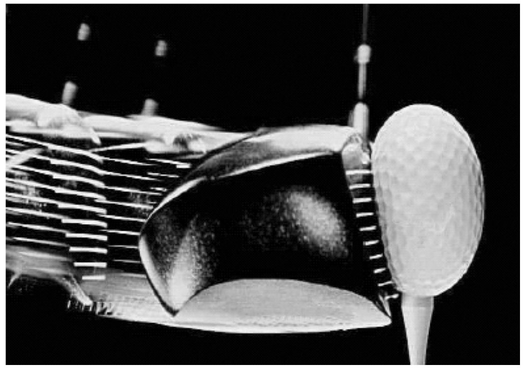


Sports Simulations

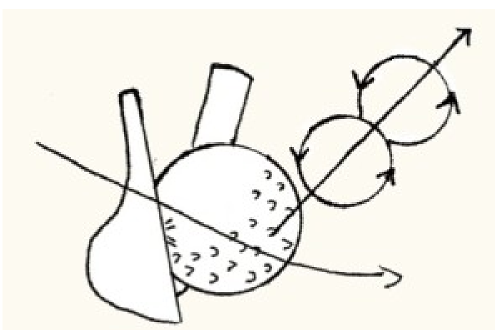
- **Examples**
  - Golf
  - Soccer
  - Basketball
  - Other games

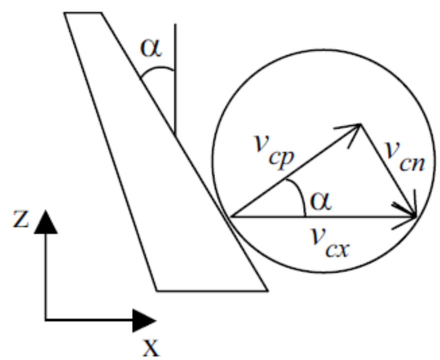
Sports Simulations

Golf



A model for simulation includes the mass of the club head, the mass of the ball, the velocity of the club head at impact, and the angle of the impact.





Schematic of a club head—golf ball collision

$$v_{cp} = (v_{Club} \cdot n)n$$

$$v_{cn} = v_{Club} - (v_{Club} \cdot n)n$$

$$\vec{v}^{+}_{Ball} = \left(\frac{(1+e)m_{Club}}{m_{Club} + m_{Ball}} \vec{v}^{-}_{Club} \cdot \vec{n} \right) \vec{n}$$

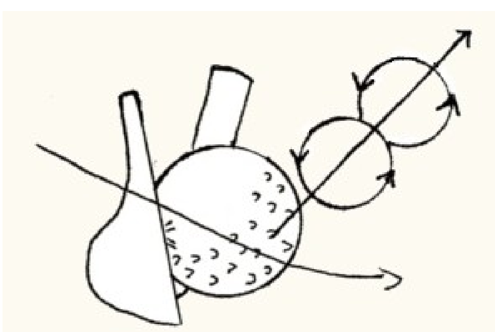
Sports Simulations

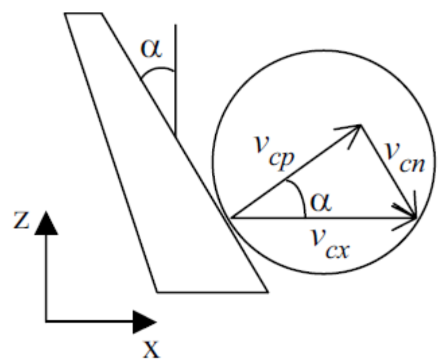
- Golf

Golf Ball Specifications

Quantity	Two-Piece Ball	Three-Piece Ball
Mass	0.0459 *kg* (1.62 *oz*)	0.0459 *kg* (1.62 *oz*)
Diameter	0.0427 *m* (1.68 *in*)	0.0427 *m* (1.68 *in*)
Coefficient of restitution	0.78	0.68
Drag coefficient	0.21–0.25	0.22–0.35

A model for simulation includes the mass of the club head, the mass of the ball, the velocity of the club head at impact, and the angle of the impact.





Schematic of a club head—golf ball collision

$$\vec{v}_{cp} = \left(\vec{v}_{Club}\cdot\vec{n}\right)\vec{n}$$

$$\vec{v}_{cn} = \vec{v}_{Club} - \left(\vec{v}_{Club}\cdot\vec{n}\right)\vec{n}$$

$$\vec{v}^{+}_{Ball} = \left(\frac{(1+e)\,m_{Club}}{m_{Club}+m_{Ball}}\,\vec{v}^{-}_{Club}\cdot\vec{n}\right)\vec{n}$$

Sports Simulations

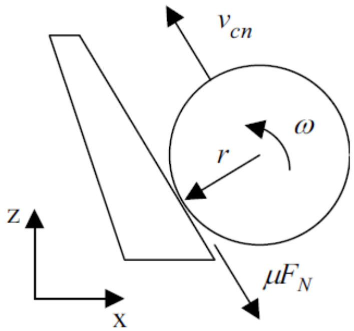
Golf

Golf Clubs Specifications

Club	Loft (Degrees)	Club Head Mass
1 wood	11	0.2 *kg* (7.05 *oz*)
3 wood	15	0.208 *kg* (7.34 *oz*)
5 wood	18	0.218 *kg* (7.69 *oz*)
2 iron	18	0.232 *kg* (8.18 *oz*)
3 iron	21	0.239 *kg* (8.43 *oz*)
4 iron	24	0.246 *kg* (8.67 *oz*)
5 iron	27	0.253 *kg* (8.92 *oz*)
6 iron	31	0.260 *kg* (9.17 *oz*)
7 iron	35	0.267 *kg* (9.42 *oz*)
8 iron	39	0.274 *kg* (9.66 *oz*)
9 iron	43	0.281 *kg* (9.91 *oz*)
Pitching wedge	48	0.285 *kg* (10.05 *oz*)
Sand wedge	55	0.296 *kg* (10.44 *oz*)
Putter	4	0.33 *kg* (11.64 *oz*)

Sports Simulations

- Golf
  - Friction Effects



Friction between the ball and club face causes the ball to spin

$$m\left(v_n^+ - v_n^-\right) = -\frac{I\omega^+}{r}$$

$$v_n^+ = r\omega^+$$

$$v_n^+ = \frac{v_n^-}{1 + \frac{I}{mr^2}}$$

$$I = \frac{2}{5}mr^2$$

The friction force does two things:
1) it reduces the relative velocity between the club and ball, and
2) it generates a torque on the ball that causes it to spin.

$$v_n^+ = \frac{5}{7}v_n^- \qquad \omega^+ = \frac{5v_n^-}{7r}$$

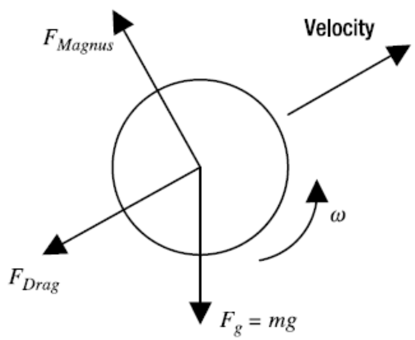
$$v_{Bx}^+ = v_{Cx}^- \frac{m_C}{m_C + m_B}\left(\left(1+e\right)\cos^2\alpha + \frac{2}{7}\sin^2\alpha\right)$$

$$v_{Bz}^+ = v_{Cx}^- \frac{m_C}{m_C + m_B}\sin\alpha\cos\alpha\left(e + \frac{5}{7}\right)$$

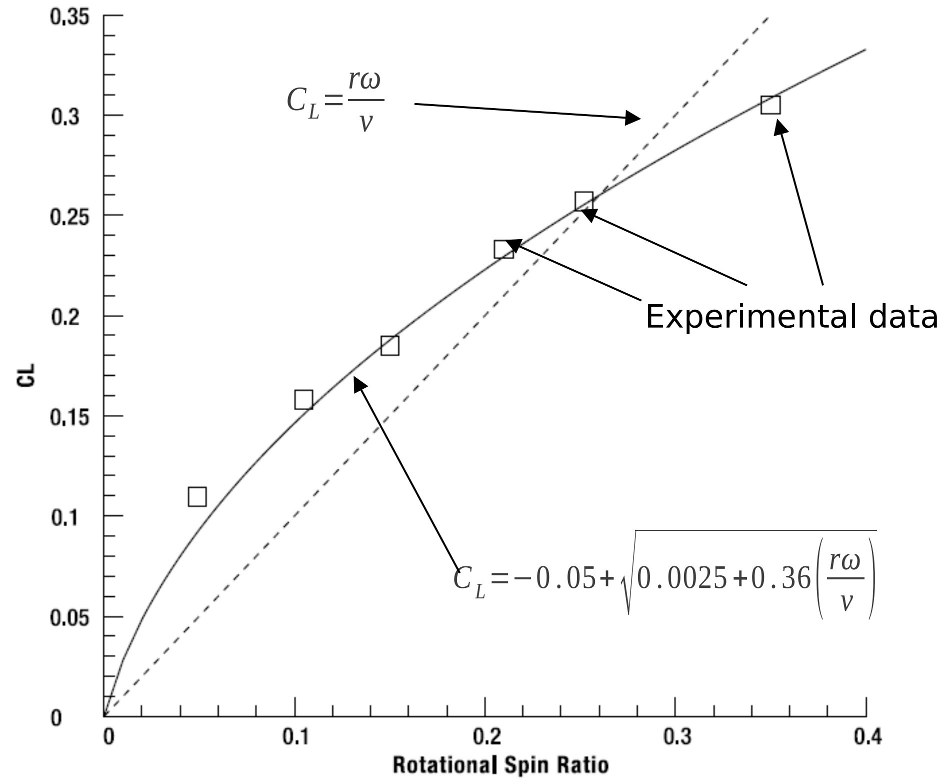
Sports Simulations

- Golf
  - Modeling the Golf Ball in Flight

$$\vec{F}_M = C_L \rho \frac{v^2}{2} A \frac{[\vec{\omega} \times \vec{v}]}{\left\| [\vec{\omega} \times \vec{v}] \right\|}$$



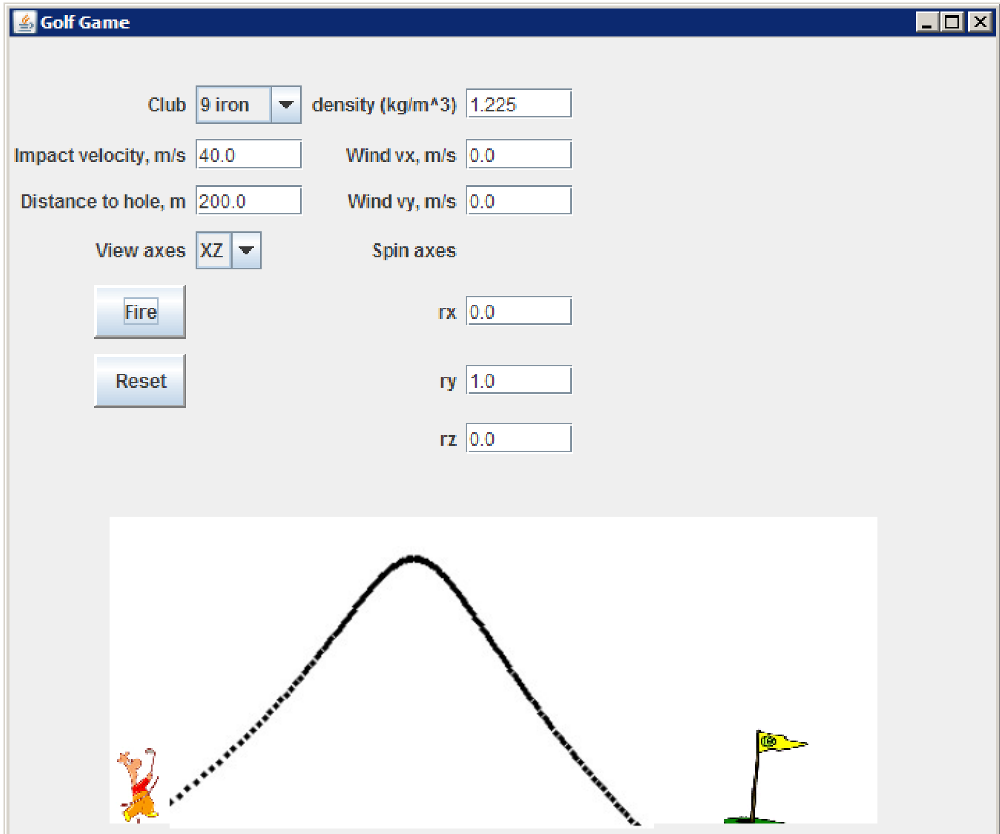
Force diagram for a golf ball in flight



Experimental and computed lift coefficients for a standard golf ball

Sports Simulations

- Golf

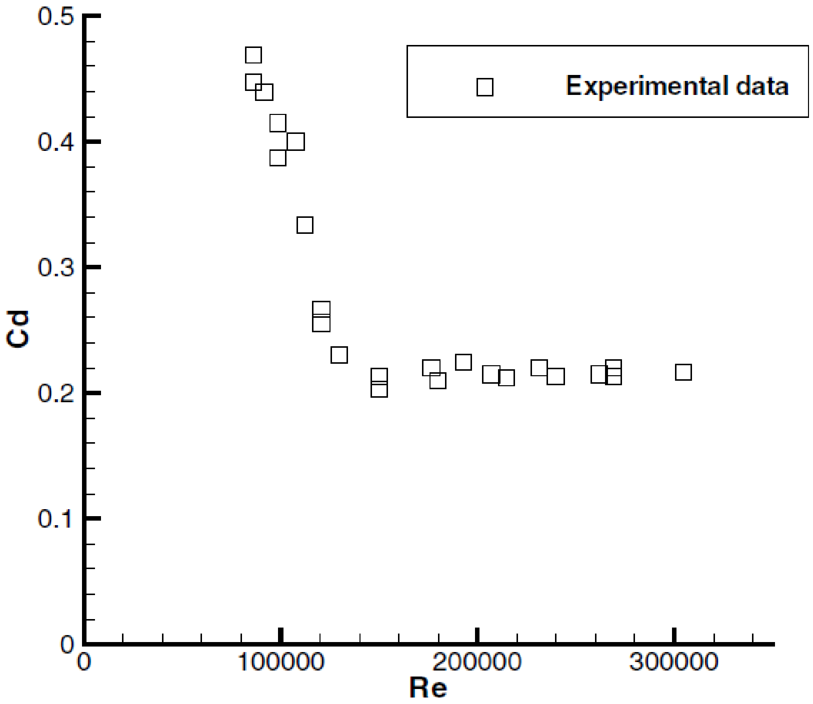


A blow-up shot results from too much spin on the ball

…Java_Code\Chapter07_Sports\GolfGame.java (from www.apress.com/book/downloadfile/2078)

Sports Simulations

- Soccer
  - Modeling the Soccer Ball in Flight
    - Laminar and Turbulent Drag



Drag coefficient of a nonspinning soccer ball

$$F_D = C_D \rho \frac{v^2}{2} A$$

The Reynolds number: $\mathrm{Re} = \frac{\rho v L}{\mu}$

The viscosity of air: $\mu = 1.456 \cdot 10^{-6} \frac{T^{1.5}}{T + 110.4}$

$C_D = 0.47$ for $\mathrm{Re} < 100000$

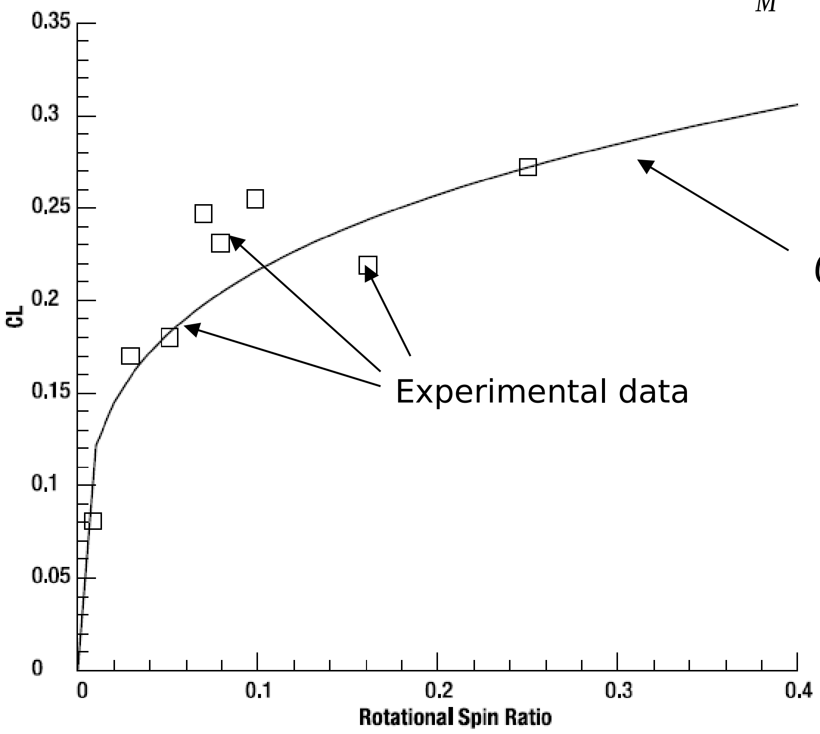
$C_D = 0.47 - 0.25 * \frac{\mathrm{Re} - 100000}{35000}$ for $100000 < \mathrm{Re} < 135000$

$C_D = 0.22$ for $\mathrm{Re} > 135000$

Sports Simulations

- Soccer
  - Modeling the Soccer Ball in Flight
    - Magnus Force

$$\vec{F}_M = C_L \rho \frac{v^2}{2} A \frac{[\vec{\omega} \times \vec{v}]}{\|[\vec{\omega} \times \vec{v}]\|}$$

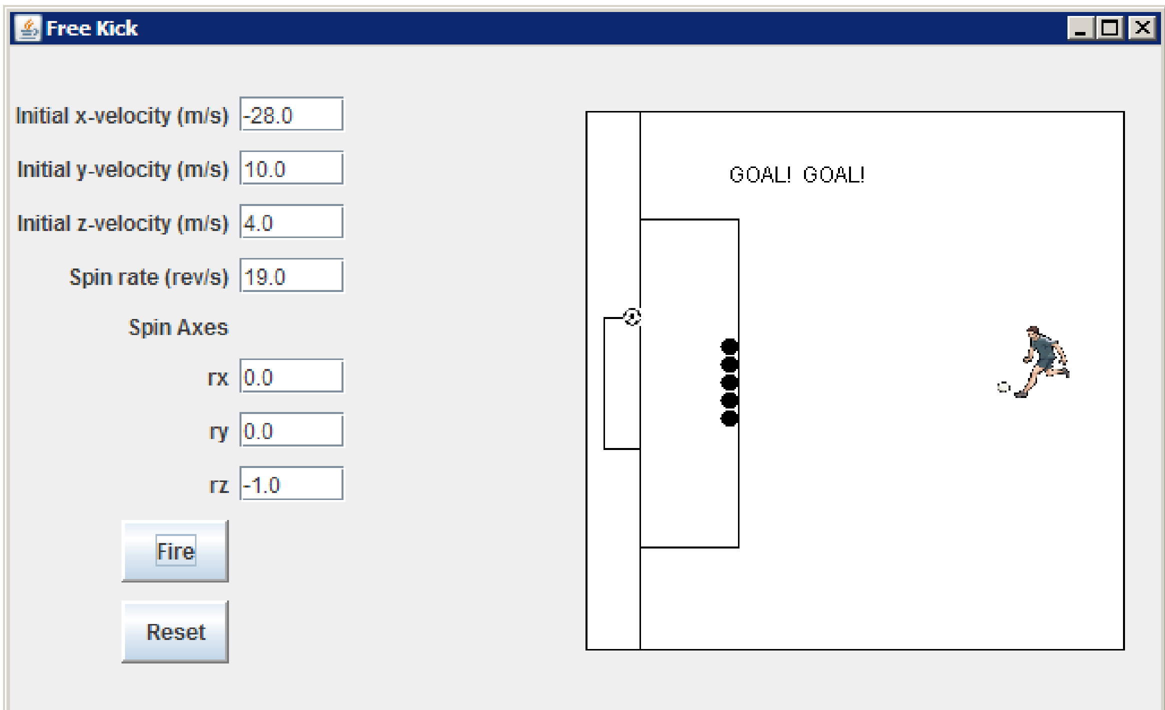


$$C_L = 0.35\left(\frac{r\omega}{v}\right)^{0.25}$$

Experimental and computed soccer ball lift coefficients

Sports Simulations

- Soccer
  - Free-Kick Game



The Free-Kick Game screen shot

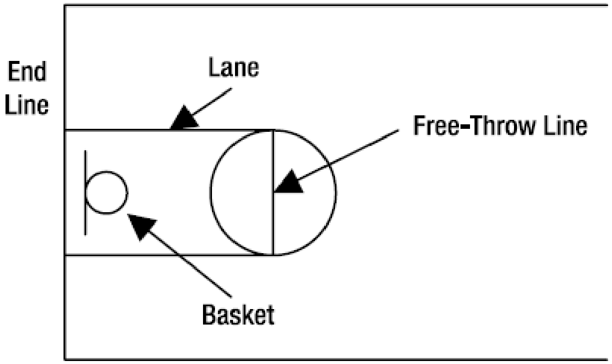
…Java_Code\Chapter07_Sports\FreeKick.java (from www.apress.com/book/downloadfile/2078)

Sports Simulations

- Basketball
  - Equipment Specifications

The Radius, Diameter, and Mass of Regulation Men's Basketballs

	FIBA	NBA	NCAA
Circumference (m)	0.78	0.749–0.762	0.76
Radius (m)	0.124	0.119–0.121	0.121
Mass (kg)	0.567–0.650	0.567–0.624	0.624



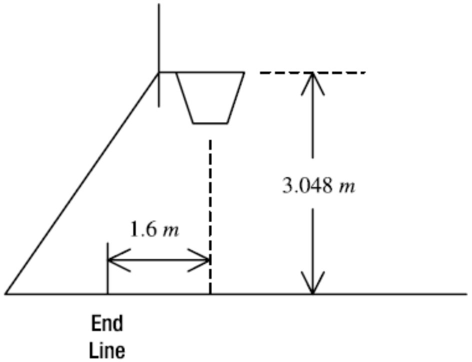
A schematic of the location of the basket, lane, and free-throw line

Court Dimensions

	FIBA	NBA	NCAA
Court length (m)	28	28.65	28.65
Court width (m)	15	15.24	15.24
Lane length (m)	5.8	5.79	5.79
Lane width (m)	6.0	4.88	3.66
3-point line distance (m)	6.25	6.71–7.24	6.02

Sports Simulations

- Basketball
  - Equipment Specifications



Basket and backboard schematics

Basket and Backboard Dimensions

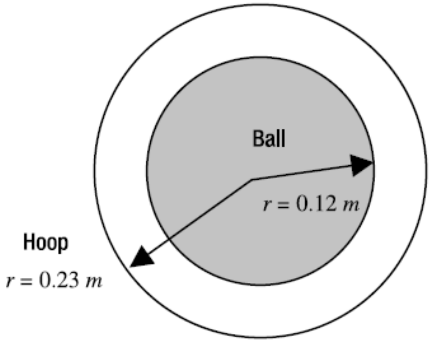
	FIBA	NBA/NCAA
Basket inside diameter (m)	0.45–0.475	0.4572
Hoop diameter (m)	0.016–0.02	0.016–0.02
Backboard height (m)	1.05	1.07
Backboard width (m)	1.8	1.83

Sports Simulations

- Basketball
  - Evaluating the Forces on a Basketball in Flight

Force and Acceleration Components Acting on a Basketball

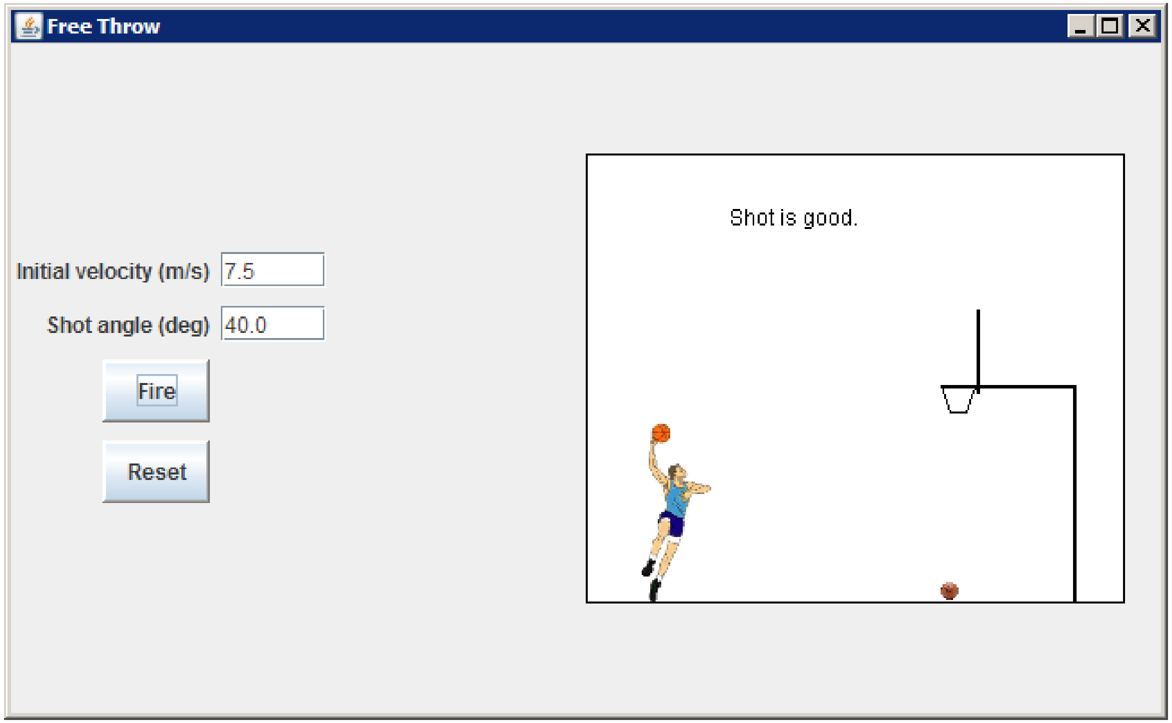
Force Type	Force Value (N)	Acceleration Value (m/s²)
Gravity	$F_g = mg = -6.08$	$a_g = g = -9.81$
Drag	$F_D = \frac{1}{2} C_D \rho v^2 A = 0.76$	$a_D = 1.23$
Spin	$F_M = \frac{1}{2} C_L \rho v^2 A = 0.23$	$a_M = 0.37$



For a shot to be good, it must travel through the hoop

Sports Simulations

- Basketball
  - **A Free-Throw Game**



A screen shot of the Free-Throw Game

…Java_Code\Chapter07_Sports\FreeThrow.java (from www.apress.com/book/downloadfile/2078)

Sports Simulations

- **Specific of simulation of other games**
  - **Baseball**
  - **Football**
  - **Hockey**
  - **Tennis**

Sports Simulations

Summary

- When a ball (or person for that matter) is in the air, it can be treated as projectile and will be subject to the forces due to gravity, aerodynamic drag, wind, and spin.

- The Magnus force due to spin is very important for the sports of golf, soccer, and baseball. The magnitude of the force due to spin can be obtained by determining the lift coefficient for the object in question.

- At times the effects of wind and spin can be ignored, for example, when simulating the flight of a basketball.

- There are also instances, for example soccer and baseball, when it is probably better for game programming purposes not to try to model the initial collision, but rather to begin the simulation by specifying the post-collision velocity, spin rate, and spin axis of the ball.