Animation Systems:
7. Physics

Charles A. Wüthrich

CogVis/MMC, Faculty of Media

Bauhaus-University Weimar

Introduction

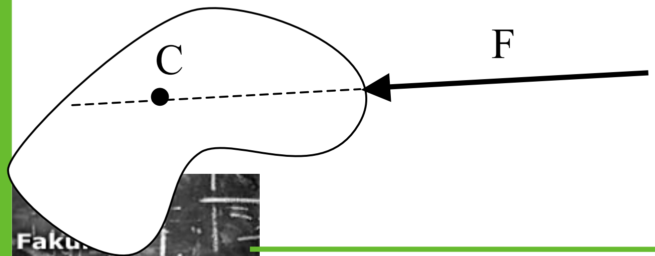
- While animating through kynematics may be interesting for plenty of applications, integrating physics is more difficult and a challenging problem
- The easiest way of integrating physics is rigid body simulation
- While physics is concerned with the exactness of the representation, animation is more interested in „credible" effects, and in rendering frame by frame
- Having to deal with the system at discrete time samples creates numerical problems in the solution methods which are not simple to deal with

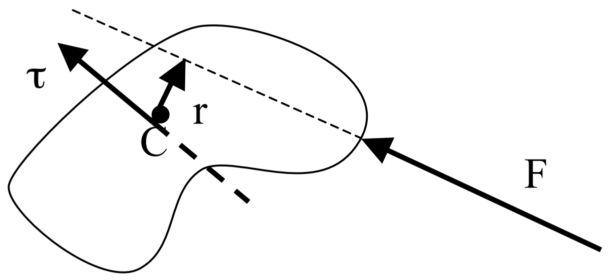
Recap on physics (physics 101)

- In the equations of motion, the following quantities play a role
  - Distance=speed · time
    time= frame# ·timeperframe
    averageVelocity=
      distancetraveled/time
- Linear motion
  - s=position
    v=velocity
    a=acceleration
  - s(t)=v(t) · t
    v(t)=a(t) · t
    s(t)= ·½ a(t) · t^2
- Circular motion
  - θ angular position
    ω angular velocity
    θ(t)= ω(t) · t
  - For a body in circular motion, we have
    $a(t)=(-\omega)^2 \cdot r$
- Newton's law:
  - $\underline{F} = m \cdot \underline{a}$
  - A body continues its own motion therefore if the sum of the forces acting on it =0
    $\Sigma F_i=0$

Recap on physics (physics 101)

- Remember the definition of center of mass
  - The point at which the object is balanced in all directions
  - If an external force is applied to a body in line with its center of mass, then the body would move as if it was a point at the center of mass C
- Torque is the tendency of a force to produce circular motion
  - It is produced by a force off center to the center of mass
  - $\tau = r \times F$
  - Clearly, $\tau \perp F$ and $\tau \perp r$
- An object does not move if $\Sigma F_i = 0$ and $\Sigma \tau_i = 0$





Recap on physics (physics 101)

- Linear springs:
  - Hooke‘s law:
    $F = -k \cdot x$,
    where x is the change from the equilibrium length of the spring
- Friction:
  - Static:
    $F_s = s \cdot f_N$
    where F_s = frictional force
    s=static friction coefficient
    f_N = normal force
  - Kinetic:
    $F_k = k \cdot f_N$
    with similar coefficient definitions as in static friction

- Momentum: $m \cdot v$
  - In a closed system, total momentum does not vary
- Angular momentum:
  $L = r \times p$
  where
  r=vector from center of rotation
  p=momentum $(m \cdot v)$
  - Note that
    $\tau = dL/dt$
  - In a closed system, total angular momentm does not vary
- Inertia tensor: the resistence of an object to change its angular momentum

Rigid body simulation

- If one wants to simulate rigid bodies, many forces act on them
- Such forces vary in time conti-nuously and in a non linear way
- Therefore it is not enough to evaluate velocities and acce-lerations at fixed timesteps Δt
- Evaluating the velocities at t_0, $t_0+\Delta t$, $t_0+2\Delta t$ does not generate a correct movement, and slowly drifts away from the correct solution
- This solution method is an example of the Euler integration method
- The accuracy of the method is determined by the size of the time step
- Obviously the shorter the time step, the more computations are needed
- A better way of integrating the equations bases on the Runge Kutta method
  - In particular, often 2nd order Runge Kutta (midpoint method) is used
  - Remember, the order of the RK method is the magnitude of the error term
  - Even higher order ones, 4th or 5th ones are used

Motion equations for a rigid body

- To develop the equation of motion for a rigid body, we have to apply some of the physics presented before
- When a force is applied to a rigid body, the force and the relative torque are applied to the body
- To uniquely solve for the resulting motions of interacting bodies, linear and angular momentum have to be conserved
- Finally, to calculate the angular momentum the distribution of an object mass in space has to be characterized with its inertia tensor.

Orientation and rotational movement

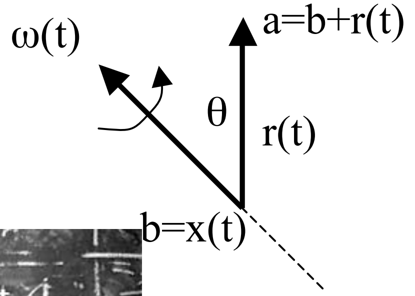
- Similar to position, velocity and acceleration, 3D objects have
  - orientation,
  - angular velocity and
  - angular acceleration
- which vary in time
- Let R(t) represent the object rotation
- *Angular velocity* $\underline{\omega}(t)$ is the rate at which the object is rotated (indep from linear velocity)
- The direction of $\underline{\omega}(t)$ indicates the orientation of the axis about which the object is rotating
- The magnitude of $\underline{\omega}(t)$ gives the speed of rotation in revs per unit time

Orientation and rotational movement

- Consider a point *a* whose position is defined in space relative to a point *b=x(t)*.
- Let *a*'s position be defined by *r(t)*
- Suppose that *a* is rotating and the axis passes through *b*
- The change in *r(t)* is the cross product of *r(t)* and *ω(t)*

  $\overset{\circ}{r}(t)=\omega(t) \times r(t)$

  $|\overset{\circ}{r}(t)|=|\omega(t)|\ |r(t)|\ \sin\theta$



- Now consider an object that has a distribution of mass in space
- Its orientation can be seen as a transformed version of the object local coord. system
- Its columns can be seen as vectors defining the relative positions of the object points
- Thus, the change of the rotation matrix can be computed by taking the cross product of *ω(t)* with each of the columns of *R(t)*

  $R(t)=[R_1(t)\ R_2(t)\ R_3(t)]$

  $\overset{\circ}{R}(t)=$

  $[\omega(t)\times R_1(t)\ \ \omega(t)\times R_2(t)\ \ \omega(t)\times R_3(t)]$

Orientation and rotational movement

- By defining a special matrix to represent cross products:

$$A \times B = \begin{bmatrix} A_y \cdot B_z - A_z \cdot B_y \\ -(A_z \cdot B_x) + A_x \cdot B_z \\ A_x \cdot B_y - A_y \cdot B_x \end{bmatrix} =$$

$$= \begin{bmatrix} 0 & -A_z & A_y \\ A_z & 0 & -A_x \\ -A_y & A_x & 0 \end{bmatrix} \begin{bmatrix} B_x \\ B_y \\ B_z \end{bmatrix} = A^* B$$

- It follows
  $R^{\circ}(t) = \omega(t)^* \; R(t)$

- Consider now a point Q on a rigid object
- Its position in local coord system does not change
- Its pos in world coords is
  $q(t)=R(t)q+x(t)$
- Differentiating this one obtains the velocity
- The change in orientation is given by the eq on the left
- Combining these, one obtains
  $q^{\circ}(t)=\omega(t)^* \; r(t)q+v(t)$
- Substituting one obtains
  $q^{\circ}(t)=\omega(t)(q(t)-x(t))+v(t)$

Center of mass

- The center of mass of a body is defined as the integral of the differential mass times its position in the object
- In a body with discrete masses, then the center of mass is *at* $q_i(t)$, the center of mass is at $x(t)=\Sigma m_i q_i(t)/\Sigma m_i$

Forces and torque

- A linear force applied to a mass gives rise to a linear acceleration

$$F=ma$$

- The various forces applied to a point sum up

$$F(t)=\Sigma f_i(t)$$

- The torque arising from the application of forces acting on a point of an object is given by

$$\tau_i(t)=(q(t)-x(t)) \times f_i(t)$$
$$\tau(t)=\Sigma\tau_i(t)$$

Momentum

- The momentum of an object (= mass times velocity) is decomposed into
  - linear component: acts on center of mass
  - angular components: acts WRT center
- Both are preserved in a closed system
- Linear momentum p=m v
- Total linear momentum of a rigid body: $P(t)=\Sigma m_i q^\circ_i(t)$
- Deriving p=mv we obtain
  $P^\circ(t)=M\,v^\circ(t)=F(t)$
- Angular momentum is a measure of the rotating mass weighted by the mass‘s distance from the axis of rotation
- $L(t)= \Sigma((q(t)-x(t)\times m_i(q^\circ(t)-v(t)))$
  $=\Sigma(R(t)q\times m_i(\omega(t)\times(q(t)-x(t))))$
  $=\Sigma(m_i(r(t)q\times(\omega(\tau)\times R(t)q)))$
- Similar to linear momentum, torque equals the change in angular momentum
  $L^\circ(t)=\tau(t)$
- Note that since angular momentum depends on distance to center of mass, to mantain constant angular momentum, the angular velocity increases if the distance of the mass decreases

Inertia tensor

- Angular momentum is related to angular velocity the same way linear momentum is related to lin. velocity $P(t)=M\cdot v(t)$
- We have
  $L(t)=I(t)\cdot\omega(t)$
- The distrib. of mass of the obj. in space is defined through a matrix, the inertia tensor $I(t)$

$$I_{obj} = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix}$$

  where the matrix terms are computed by integrating over the object, and I is symmetric

- In general, $I_{xx}=\iiint\rho(q)(q_y^2+q_z^2)dxdydz$ where ρ is the density of at an obj. point $q=(q_x,q_y,q_z)$
- In the case of discrete masses $I_{xx}=\Sigma m_i(y_i^2+z_i^2)$, $I_{xy}=\Sigma m_i x_i y_i$ $I_{yy}=\Sigma m_i(x_i^2+z_i^2)$, $I_{xz}=\Sigma m_i x_i z_i$ $I_{zz}=\Sigma m_i(x_i^2+y_i^2)$, $I_{yz}=\Sigma m_i y_i z_i$
- In a center of mass centered obj space, the intertia tensor of a transformed object depends on the obj orientation but not on its position, and therefore it depends on time
- It can be transformed with
  $I(t)=R(t)I_{obj}R(t)^T$

Motion equations

- The state of an object can be determined by the vector containing
  - Position
  - Orientation
  - Linear momentum
  - Angular momentum

$$S(t) = \begin{bmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{bmatrix}$$

- Object mass and its object space inertia tensor I_{obj} do not change in time
- At any time, the following quantities can be computed:
  Inertia tensor $I(t)=R(t)I_{obj}R(t)^T$
  Angular vel. $\omega(t)=I(t)^{-1}L(t)$
  Linear vel. $v(t)=P(t)/M$
- Now the time derivative can be formed:

$$\frac{d}{dt}S(t) = \frac{d}{dt}\begin{bmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{bmatrix} = \begin{bmatrix} v(t) \\ \omega(t)^{*}R(t) \\ F(t) \\ \tau(t) \end{bmatrix}$$

  This is enough to run a simulation
- A differential equation solver can now be used

Motion equations

- As the simplest solver, one can use Euler‘s method
  - The values of the state array are updated by multiplying their time derivatives by the length of the time step
- In practice, Runge-Kutta methods are used, especially 4th order ones
- Particular care has to be taken in updating the orientation of an object, because if the derivative info is used to update the orientation matrix, then the columns of the matrix can become nonorthogonal and not of unit length
- Here it is wise to renormalize after each step
- Alternatively, one can update
  - by applying the axis-angle rotation generated by angular velocity
  - By using quaternions

Ordinary differential equations

- Problem involving ordinary differential equations (ODE) can be reduced to the study of first order differential equations
- For example, the problem
  $d^2y/dx^2+q(x)dy/dx=r(x)$
  can be rewritten as the two first order equations
  $dy/dx=z(x)$
  $dz/dx=r(x)-q(x)z(x)$
- The generic problem in ordinary differential equations can be thus reduced to the study of a set of N first order differential equation for the functions y_i $(i=1,2,\ldots,N)$
  $dy_i(x)/dx=f'_i(x,y_1,\ldots y_N)$
  where the f_i are known
- A problem involving ODEs is not determined completely by its equations.
- The boundary conditions also determine how to solve the problem.
- Boundary conditions are algebraic conditions on the values of the functions y_i
- In general, they can be satisfied at the discrete specified points, but do not hold inbetween these points
- They can be as simple as requiring to pass through a certain point, or as complicated as a complex algebraic expression

Ordinary differential equations

- Boundary conditions divide into two categories:
  - Initial value problems: here all y_i are given at some starting value x_s, and one wants to find the y_i at some end point x_f, or at some discrete list of points
  - Two-point boundary value problems: here some conditions are set at x_s, and some at x_f
- We will deal only with the first class

- The main idea of a method for initial value is the following:
  - Rewrite dx and dy as finite Δy and Δx and multiply the equations by Δx
- This gives formulas for the change in the functions when the variable x is stepped at stepsize Δx
- For Δx small, one obtains a decent approximation of the diff. Equation
- Literal implementation of this method is called *Euler's method*
  - Euler method is not accurate compared to other methods using the same step size
  - It is not very stable either

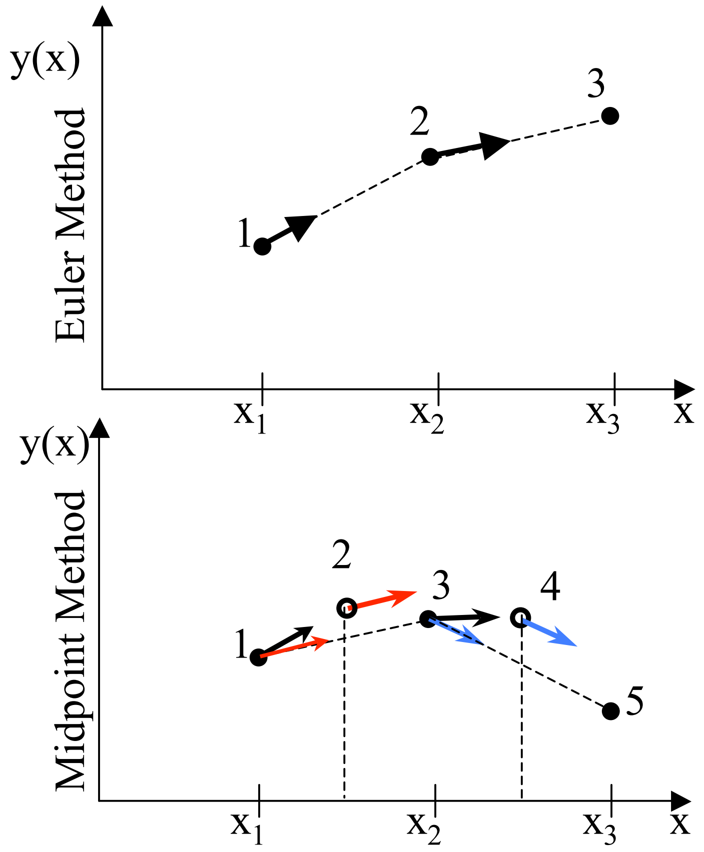
ODE: Runge Kutta

- One way or another, all methods do the following:
  - Add small increments to the functions corresponding to the derivatives multiplied by stepsizes
- Runge Kutta methods propagate a solution over an interval by combining the information from several Euler-style steps (each involving one evaluation of the f′ and then using the info obtained to match a taylor series up to some higher order
- The formula for the Euler method is
  $y_{n+1}=y_n+hf'(x_n,y_n)$
  where h is the step chosen
  (i.e. $x_{n+1}=x_n+h$)
- The problem with this is that info on the derivative change goes lost, since only info at the start of the interval (at x_n) is used
- This means that the step‘s error is only one power of h smaller than the correction, thus
  $y_{n+1}=y_n+hf'(x_n,y_n)+O(h^2)$
- By definition, we call a method such that its error term is $O(h^{n+1})$ as method of order n

ODE: Runge Kutta

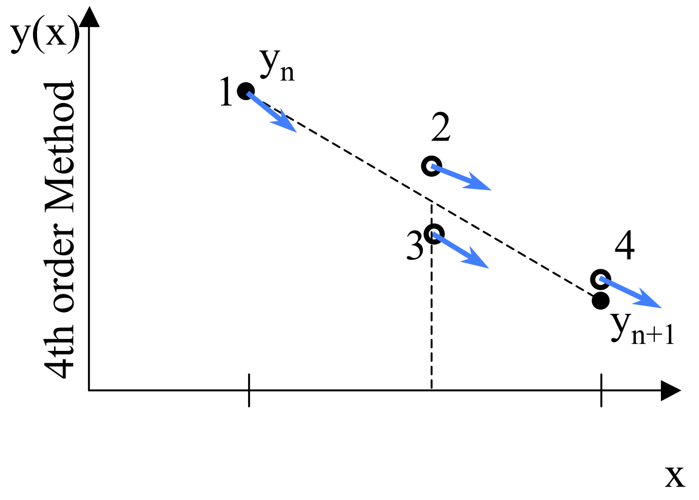
- Suppose to use a step like before ($y_{n+1}=y_n+hf'(x_n,y_n)$) to take a „trial“ step to the midpoint of the interval
- Then use the value of both x and y at the midpoint to compute the „real“ step across the whole interval
  $k_1=hf'(x_n,y_n)$
  $k_2=hf'(x_n+½h,y_n+½k_1)$
  then one can write
  $y_{n+1}=y_n+k_2+O(h^3)$
- Note how all of the sudden the error becomes of 3rd degree and therefore the method becomes of second order
- This by the way is second order Runge-Kutta (midpoint methd)

- In R-K the derivative at the midpt. is evaluated and used for the whole interval

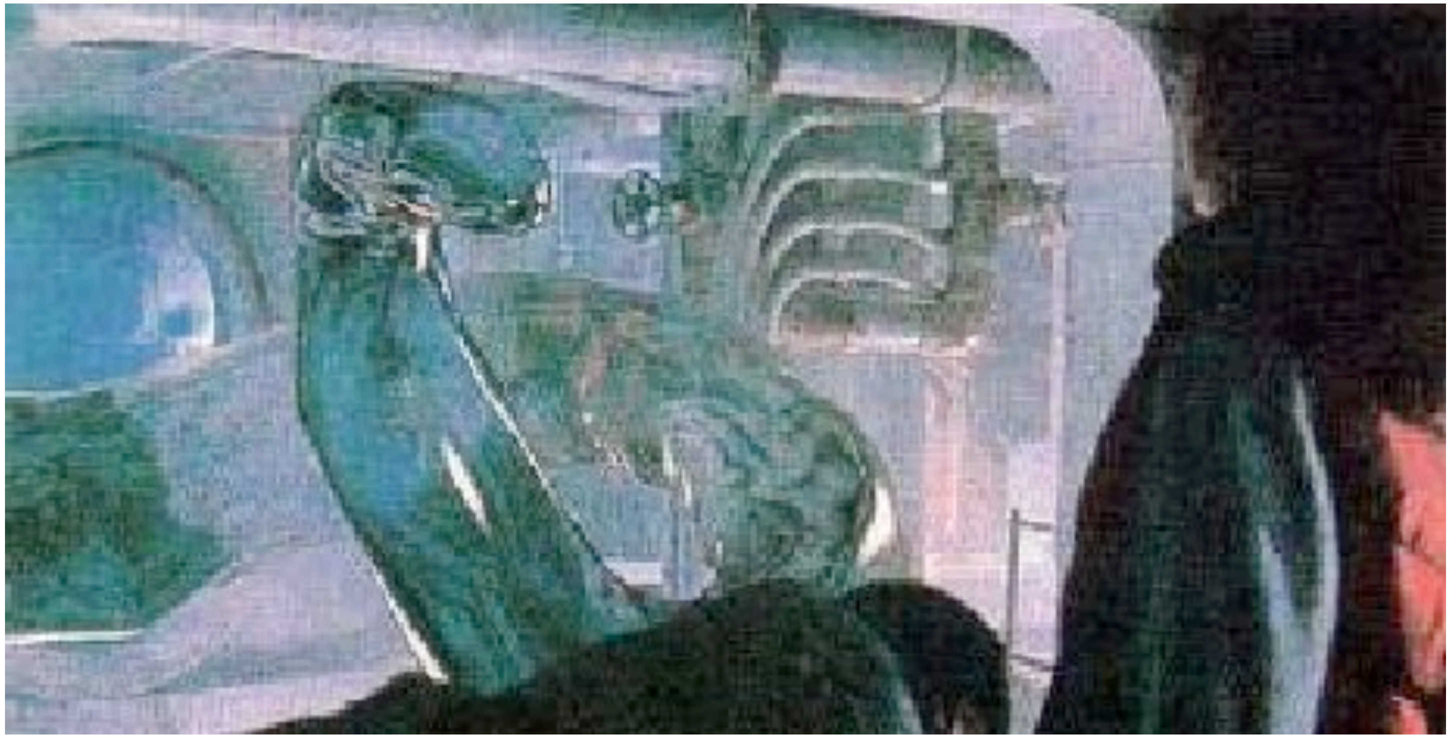


ODE: Runge Kutta

- Why does this work better?
  - Because by evaluating at the midpoint one takes an „average" of the derivative on the interval
  - This cancels the second order error
- Of course, one can decide now to push this elimination further
- Fourth order R-K evaluates the derivative four times:
  - Initial point, twice at trial midpoints, and once at the trial endpoint
  - From these derivatives the final function value is computed
- $k_1 = hf'(x_n, y_n)$
  $k_2 = hf'(x_n + h/2, y_n + k_1/2)$
  $k_3 = hf'(x_n + h/2, y_n + k_2/2)$
  $k_4 = hf'(x_n + h, y_n + k_3)$
  and the error term becomes
  $y_{n+1} = y_n + k_1/6 + k_2/3 + k_3/3 + k_4/6 + O(h^5)$



End



Copyright (c) 1988 ILM

+++ Ende - The end - Finis - Fin - Fine +++ Ende - The end - Finis - Fin - Fine +++