Models as the Basis for Visual Representation

- “realistic” 3D visualisation
- “insight” at high abstraction level
- link visualisation to model
  1. structure
  2. entitity attributes

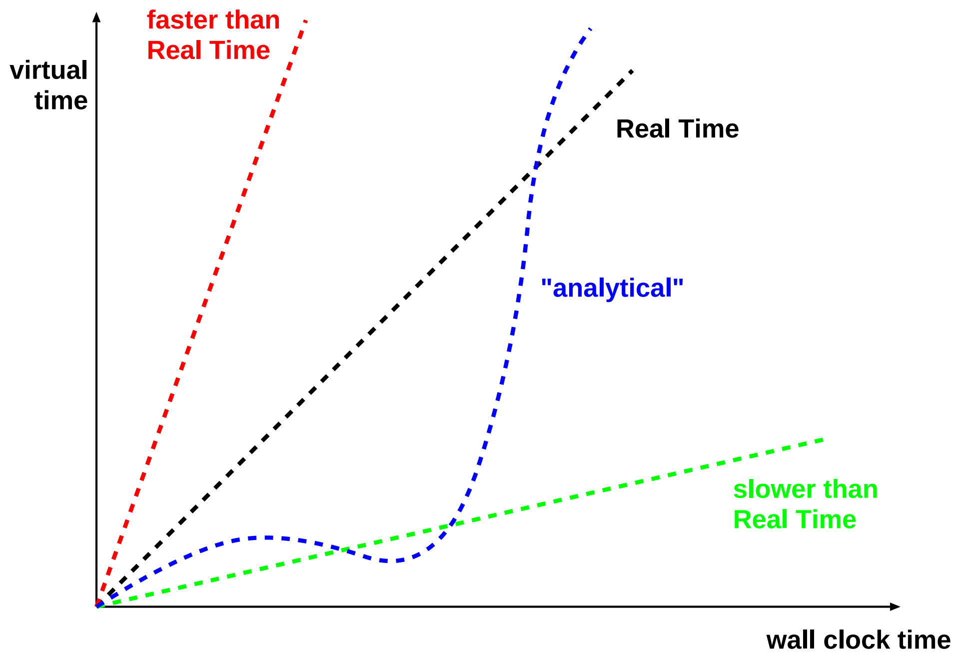
Categories of Simulation Animation Implementation

- Animation using a post-processor
- Direct simulation animation
  - integrated program (one thread)
  - cooperating programs (multiple threads, reactive subject pattern)
- Visual Interactive Simulation: user in the loop
  - interrupt, modify (parameters, IC, . . .), re-start
  - discrete event: statistical relevance ?
  - discrete event: transient behaviour
  - need to keep track of modifications
    (generate script logging modifications)

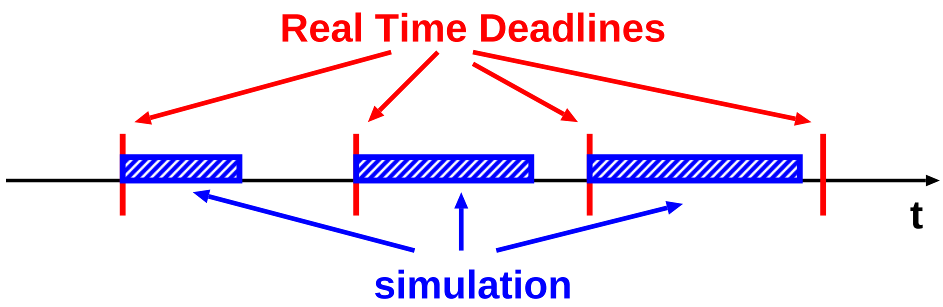
Technical Problems of Simulation Animation

- Transformation of time for animation: non-equidistant, speedup/slowdown
- Suspension of animation on multi-tasking systems: buffer

Real Time Simulation



Real Time Deadlines: Rate Monotonic Scheduling (RMS)



Continuous Models: ODE

$$\frac{d^n x}{dt^n} = f(\frac{d^{n-1}x}{dt_{n-1}}, \ldots, x, u, t)$$

f and $x(t)$ may be vectors

$$\begin{cases} x & = & x_0 \\ \frac{dx}{dt} & = & x_1 \\ \frac{dx_1}{dt} & = & x_2 \\ & \ldots & \\ \frac{dx_{n-1}}{dt} & = & x_n = f(x_{n-1}, x_{n-2}, \ldots, x_1, x_0, u, t) \end{cases}$$

Euler discretisation

$$\frac{dx}{dt} = f(x,u,t)$$

$$\Downarrow$$

$$\frac{x(t_i+\Delta t)-x(t_i)}{\Delta t} \cong f(x(t_i),u(t_i),t_i)$$

$$\Downarrow$$

$$x(t_i+\Delta t) \cong x(t_i)+f(x(t_i),u(t_i),t_i)\Delta t$$

Taylor Series Expansion

$$x(t_i+\Delta t) = x(t_i)+\frac{\Delta t}{1!}\frac{dx}{dt}|_{t_i}+\frac{\Delta t^2}{2!}\frac{d^2x}{dt^2}|_{t_i}+\ldots$$

$$\text{ERROR} = O(N+1) \text{ if chopped after } N$$

$$\text{ERROR } \varepsilon_N \cong approx_{N+1} - approx_N$$

Integration Methods: Euler

Single-step

$x_0 = \alpha_0$

$x_{i+1} = x_i + hf(t_i, x_i), i \geq 0$

Unsymmetrical: uses only derivative in begin point.

Integration Methods:Modified Euler

Single-step

$x_0 = \alpha_0$

$k1 = hf(t_i, x_i)$

$k2 = hf(t_i + h, x_i + k1)$

$x_{i+1} = x_i + \frac{k1}{2} + \frac{k2}{2}, i \geq 0$

Symmetrical: uses derivative in begin and end point.

Integration Methods: Midpoint

Single-step

$x_0 = \alpha_0$

$k1 = hf(t_i, x_i)$

$k2 = hf(t_i + \frac{h}{2}, x_i + \frac{k1}{2})$

$x_i + 1 = x_i + k2, i \geq 0$

Symmetrical: halfway point.

Integration Methods: Heun

Single-step

$x_0 = \alpha_0$

$k1 = hf(t_i, x_i)$

$k2 = hf(t_i + \frac{2h}{3}, x_i + \frac{2k1}{3})$

$x_{i+1} = x_i + \frac{k1}{4} + \frac{3k2}{4}, i \geq 0$

Integration Methods: Runge-Kutta 4

Single-step

$x_0 = \alpha_0$

$k1 = hf(t_i, x_i)$

$k2 = hf(t_i + \frac{h}{2}, x_i + \frac{k1}{2})$

$k3 = hf(t_i + \frac{h}{2}, x_i + \frac{k2}{2})$

$k4 = hf(t_i + h, x_i + k3)$

$x_{i+1} = x_i + \frac{k1}{6} + \frac{k2}{3} + \frac{k3}{3} + \frac{k4}{6}, i \geq 0$

"Optimal" choice of intermediate points.

Integration Methods: Adams-Bashforth

Multi-step: Need one-step methods for *start-up*

2-step

$x_0 = \alpha_0$

$x_1 = \alpha_1$

$x_{i+1} = x_i + \frac{h}{2}(3f(t_i, x_i) - f(t_{i-1}, x_{i-1}), i \geq 2$

3-step

$x_0 = \alpha_0$

$x_1 = \alpha_1$

$x_2 = \alpha_2$

$x_{i+1} = x_i + \frac{h}{12}(23f(t_i, x_i) - 16f(t_{i-1}, x_{i+1}) + 5f(t_{i-2}, x_{i-2})), i \geq 2$

4-step

$x_0 = \alpha_0$

$x_1 = \alpha_1$

$x_2 = \alpha_2$

$x_3 = \alpha_3$

$x_{i+1} =$
$x_i + \frac{h}{24}(55 f(t_i, x_i) - 59 f(t_{i-1}, x_{i+1}) + 37 f(t_{i-2}, x_{i-2}) - 9 f(t_{i-3}, x_{i-3})), i \geq 3$

Integration Methods: Milne

Predictor-corrector

- Predictor
  $x_0 = \alpha_0$
  $x_1 = \alpha_1$
  $x_2 = \alpha_2$
  $x_3 = \alpha_3$
  $x_{i+1}^{(0)} = x_{i-3} + \frac{4h}{3}(2 f(t_i, x_i) - f(t_{i-1}, x_{i-1}) + 2 f(t_{i-2}, x_{i-2})), i \geq 3$
- Corrector
  $x_{i+1}^{(k+1)} = x_{i-1} + \frac{h}{3}(f(t_{i+1}, x_{i+1}^{(k)}) + 4 f(t_i, x_i) + f(t_{i-1}, x_{i-1})),$
  $i \geq 2, k = 1, 2, \ldots$

Adaptive Step-size Control

Use accuracy indicator to double/halve step-size

1. step halving
2. $\varepsilon_N \cong approx_{N+1} - approx_N$

Adaptive Step-size Control

RK4 + RK5 $\rightarrow$ Runge-Kutta Fehlberg (embedded)

$k_1 = hf(t_i, x_i)$

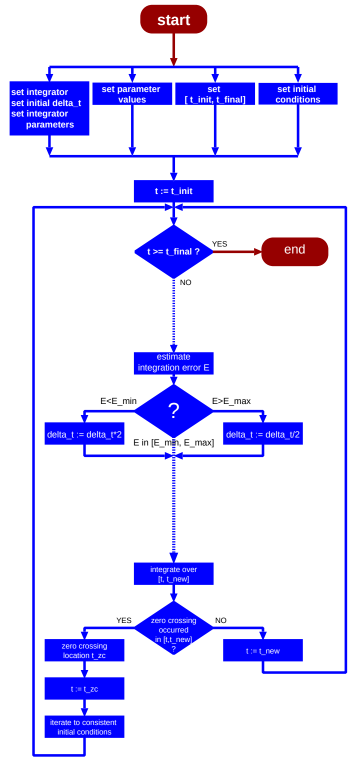
$k_2 = hf(t_i + a_2 h, x_i + b_1 2 k_1)$

$\cdots$

$k_6 = hf(t_i + a_6 h, x_i + b_6 1 k_1 + b_6 2 k_2 + \cdots + b_6 5 k_1)$

$x_{i+1} = x_i + c_1 k_1 + c_2 k_2 + \cdots + c_6 k_6 + O(h^6)$

$x^*_{i+1} = x_i + c^*_1 k_1 + c^*_2 k_2 + \cdots + c^*_6 k_6 + O(h^5)$



Stiff Systems

$u' = 998u + 1998v$

$v' = -999u - 1999v$

$u(0) = 1, v(0) = 0$

$u = 2y - z, v = -y + z$

$u = 2e^{-t} - e^{-1000t}$

$v = -e^{-t} + e^{-1000t}$

$x' = -cx$

Explicit: Forward Euler: $x_{i+1} = x_i + hx'_i$

$x_{i+1} = (1 - ch)x_i$

Implicit: Backward Euler: $x_{i+1} = x_i + hx'_{i+1}$

$x_{i+1} = \frac{x_i}{1+ch}$

Rosenbrock, Gear, ... methods

Differential Algebraic Equations (DAE)

$$f(\frac{d^n x}{dt^n}, \frac{d^{n-1} x}{dt_{n-1}}, \ldots, x, u, t) = 0$$

$$g(x, t) = 0$$

Residual Solvers
DASSL (Petzold)