



Integration methods in systems simulation

Prof. Cesar de Prada

Dpt. Systems Engineering and Automatic
Control

University of Valladolid

prada@autom.uva.es



Outline

- ✓ Model types
- ✓ Integration methods
 - Explicit
 - Implicit
 - Variable step size
 - Stiff problems
 - DAE
- ✓ Dealing with events and discontinuities



Model types

- ✓ **Steady state models**
 - They describe the process in equilibrium. Variables do not change over time.
 - Represented usually with algebraic equations.
- ✓ **Dynamic models**
 - They describe the time evolution of a process
 - Represented with a wide range of differential or difference equations
- ✓ **Discrete event models**
 - Variables change at certain time instant from one state to another as a result of the occurrence of an event



Lumped parameter models

- ✓ The properties of the process are assumed not to change with spatial positions. Variables are assumed to be homogeneous over space, but time dependent.
- ✓ They are formulated as sets of differential and algebraic variables (ODEs or DAEs)
- ✓ Generally the equations represent mass, energy, momentum,... balances, equilibriums, ...

$$F\left(\frac{dx}{dt}, x, u, t\right) = 0$$

x states

u inputs, actions

$$G(y, x, u) = 0$$

y outputs, responses



Distributed Parameter Systems

- ✓ The value of some variables depends on their spatial position. In a dynamic distributed parameter system, the variables can depend on both, space and time.
- ✓ Their formulation includes sets of partial differential equations (PDE's)
- ✓ They appear in many chemical processes, transport, ...

$$F\left(\frac{\partial x}{\partial t}, \frac{\partial x}{\partial z}, x, u, t, z\right) = 0$$

z spatial
coordinate



Continuous/discrete time models

- ✓ In continuous models the variables are real numbers that can change continuously over time.
- ✓ In discrete time models, the variables change at given time instants.
- ✓ Discrete time models are described with difference equations
- ✓ They appear in computer controlled systems, finance, population growth,...

$$F(x(k+1), x(k), u(k), t_k) = 0$$

$$G(y(k), x(k), u(k)) = 0$$

$$k = 0, 1, 2, \dots =$$

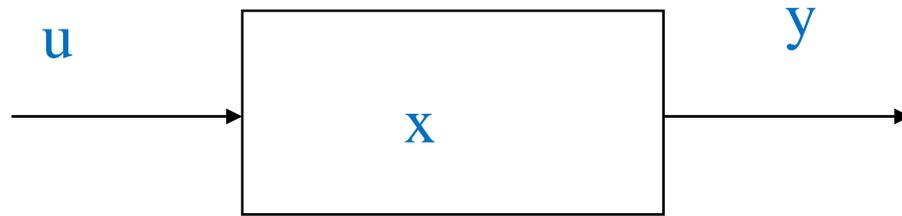
0, T, 2T, ...

T sampling time



Types of problems

- ✓ Different types of problems according to the variables assumed to be known



$u, x_0 \rightarrow y$ Simulation, What happens if ..?

$u, y \rightarrow x$ State estimation, identification

$y, x_0 \rightarrow u$ Control, What should I do for...?



ODE's Integration

- ✓ In a Ordinary Differential Equation (ODE), the derivative of the state appears in explicit form. There are many methods for the integration of sets of equations in state space form.

$$\frac{dx}{dt} = f(x, u, t) \quad x(0) = x_0$$

Quite often, in order to integrate a set of differential equations, it must be transformed previously to this format



Example

This second order differential equation:

$$m \frac{d^2 x}{dt^2} = F \quad x(0) = x_0 \quad v(0) = v_0$$

Can be converted into two first order differential equations, in state space format:

$$\frac{dx}{dt} = v \quad x = x_0$$

$$\frac{dv}{dt} = \frac{F}{m} \quad v = v_0$$

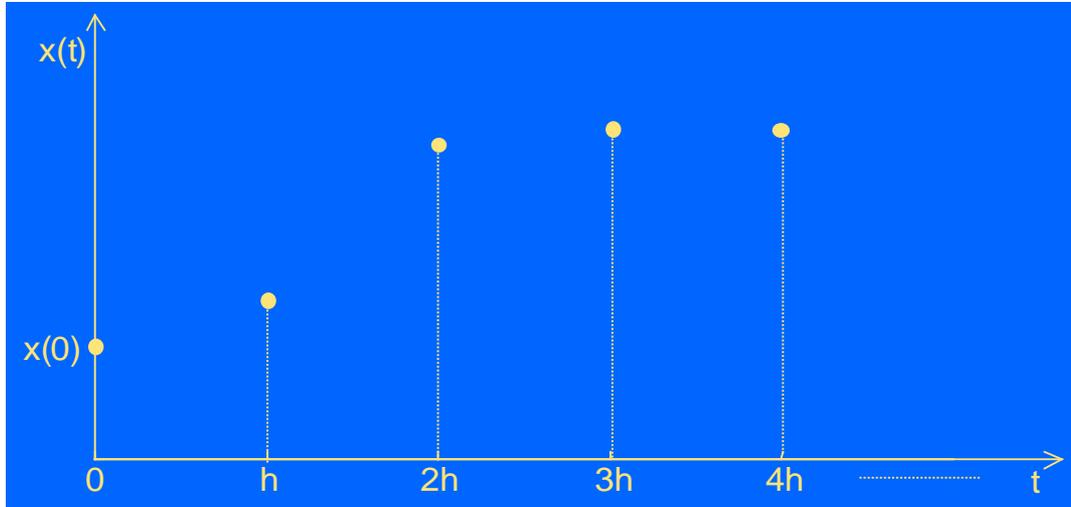


Integration methods

- ✓ Explicit methods
- ✓ Errors
- ✓ Variable step-size methods
- ✓ Implicit methods
- ✓ Stability
- ✓ “Stiff” systems
- ✓ Methods for DAE



Concept



$$\frac{dx}{dt} = f(x, u, t) \quad x(0) = x_0 \quad h \text{ integration step}$$

Iterative approach: Knowing x at $t = 0$, the method estimate x at $t = h$. With the value of x at $t = h$, the method estimate x at $t = 2h$, etc.

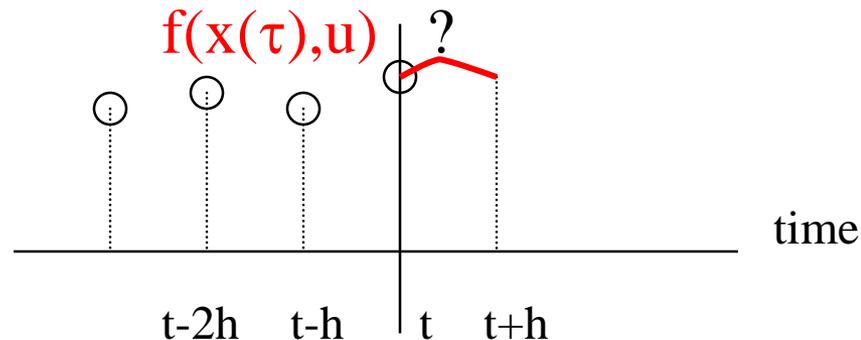


Integration of ODE's

$$\frac{dx}{dt} = f(x, u) \quad x(0) = x_0$$

$$x(t+h) = x(t) + \int_t^{t+h} f(x(\tau), u) d\tau$$

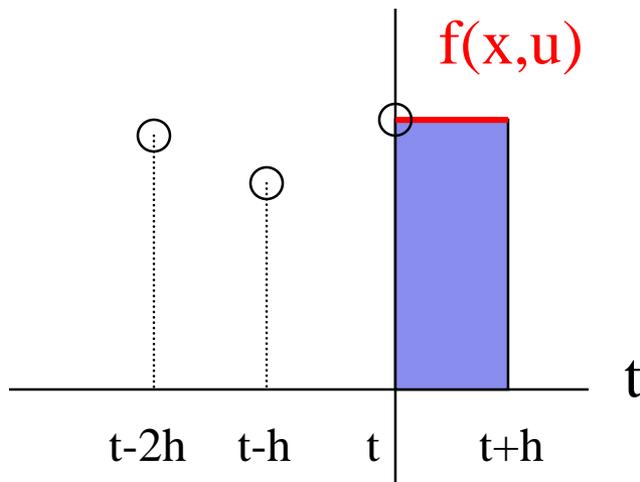
$f(x, u)$ is assumed to be continuous and differentiable and u a known function of time





Explicit methods

- ✓ $f(x(\tau), u)$ is approximated in $[t, t+h]$ by an **extrapolation** polynomial, taking advantage of the continuity of f



Zero order
approximation

$$f(x, u) = f(x(t), u)$$

$$x(t+h) = x(t) + \int_t^{t+h} f(x(\tau), u) d\tau$$

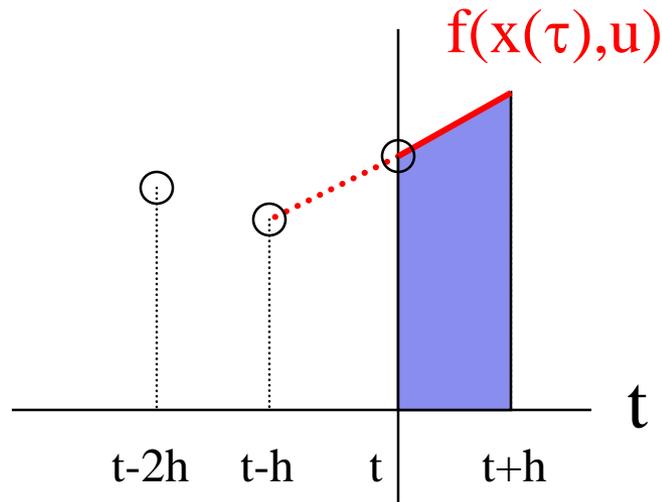
Euler's method:

$$x(t+h) = x(t) + hf(x(t), u)$$

$f(x(t), u)$ is called residual in the integration jargon



Adams-Bashforth 2nd order



First order
approximation:

$$f(x(\tau), u) = f(x(t), u) + \frac{f(x(t), u) - f(x(t-h), u)}{h}(\tau - t)$$

$$x(t+h) = x(t) + \int_t^{t+h} f(x(\tau), u) d\tau$$

Trapezoidal
formula:

$$x(t+h) = x(t) + 0.5h[3f(x(t), u) - f(x(t-h), u)]$$



Explicit methods

$x(t+h)$ is computed as a explicit function of x and u in the previous time instants

$$x(t + h) = x(t) + hf(x(t), u)$$

$$x(t + h) = x(t) + 0.5h[3f(x(t), u) - f(x(t - h), u)]$$

.....

Adams-Bashforth
4th order

$$x(t + h) = x(t) + \frac{h}{24} \left[55f(x(t), u) - 59f(x(t - h), u) + 37f(x(t - 2h), u) - 9f(x(t - 3h), u) \right]$$

Initialization problem: To start a n -order method, it is necessary to use formulas of $n - i$ orders in the previous n steps



Taylor series expansion

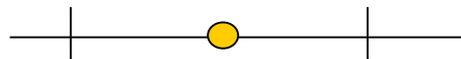
As an alternative to the polynomial extrapolation of $f(x(t), u)$, one can use a Taylor series:

$$x(t+h) = x(t) + \left. \frac{dx}{dt} \right|_t h + \left. \frac{d^2x}{dt^2} \right|_t \frac{h^2}{2!} + \dots + O(h^{n+1})$$

$$x(t+h) = x(t) + f(x(t), u)h + \left. \frac{\partial f}{\partial x} \right|_t \frac{h^2}{2!} + \dots$$

Runge-Kutta methods replace the estimation of the higher order derivatives by a linear combination of values of the function f evaluated at time instants between t and $t+h$ with the constraint of obtaining the same precision as the truncated Taylor series.

$$x(t+h) = x(t) + w_1 f(x(t), u)h + w_2 f(x(t+ah), u)h + \dots$$





2nd order Runge-Kutta

$$x(t+h) = x(t) + w_1 f(x(t), u)h + w_2 f(x(t+ah), u)h$$

$$f(x(t+ah), u) \approx f(x(t), u) + \left. \frac{\partial f}{\partial x} f \right|_t ah$$

→
$$x(t+h) = x(t) + (w_1 + w_2) f(x(t), u)h + w_2 \left. \frac{\partial f}{\partial x} f \right|_t ah^2$$

→
$$x(t+h) = x(t) + f(x(t), u)h + \left. \frac{\partial f}{\partial x} f \right|_t \frac{h^2}{2!} + \dots$$

Both expressions are equal up to
2nd order if

$$w_1 + w_2 = 1$$

$$aw_2 = 0.5$$



2nd order Runge-Kutta

$$w_1 + w_2 = 1 \qquad aw_2 = 0.5$$

Several choices:

$$\text{If } w_1=0 \Rightarrow w_2=1, a=0.5$$

$$x(t+h) = x(t) + w_1 f(x(t), u)h + w_2 f(x(t+ah), u)h$$

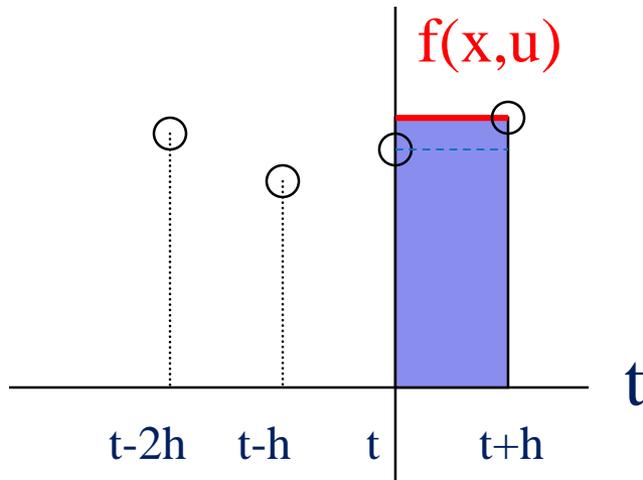
$$x(t+h) = x(t) + f(x(t+0.5h), u)h$$

with $x(t+0.5h) = x(t) + f(x(t), u)0.5h$



Implicit methods

The formulation of the extrapolation polynomial involves future values of $x(t)$



Implicit Euler's
method

Zero order polynomial:

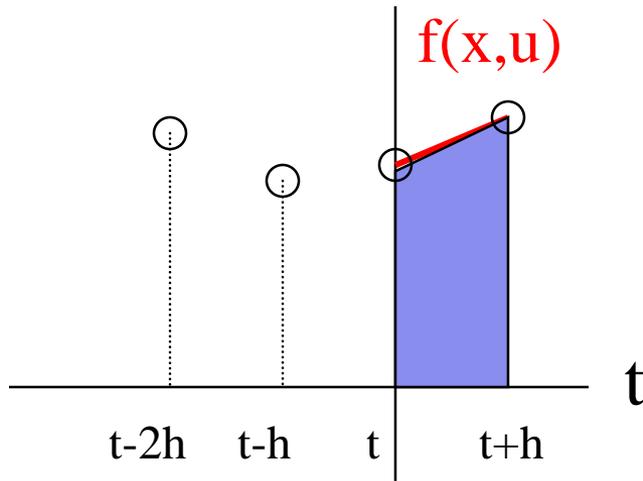
$$f(x(\tau), u) = f(x(t+h), u)$$

$$x(t+h) = x(t) + \int_t^{t+h} f(x(\tau), u) d\tau$$

$$x(t+h) = x(t) + hf(x(t+h), u)$$



Crank-Nicolson



First order polynomial

$$f(x(\tau), u) = f(x(t), u) + \frac{f(x(t+h), u) - f(x(t), u)}{h}(\tau - t)$$

$$x(t+h) = x(t) + \int_t^{t+h} f(x(\tau), u) d\tau$$

Implicit
trapezoidal
method

$$x(t+h) = x(t) + 0.5h[f(x(t+h), u) + f(x(t), u)]$$



Implicit methods

- ✓ With the implicit integration methods it is not possible, in general, to compute $x(t+h)$ explicitly from the integration formula. A non-linear algebraic equation in $x(t+h)$ must be solve:

$$x(t+h) = x(t) + hf(x(t+h), u)$$

$$x(t+h) = x(t) + 0.5h[f(x(t+h), u) + f(x(t), u)]$$

Implicit methods are computationally more expensive, but have better numerical stability.

Adams- Moulton methods are the implicit versions of Adams – Bashforth ones



Solving implicit algebraic equations



Implicit integration methods give rise to equations with the format:

$$\mathbf{x} = h\mathbf{f}(\mathbf{x}) + \mathbf{a}$$

Where \mathbf{x} must be computed numerically

Two popular numerical methods for solving non-linear algebraic equations are:

Successive approximations

Newton-Raphson



Successive approximations

$$\mathbf{x} = h\mathbf{f}(\mathbf{x}) + \mathbf{a}$$

Starting from an initial guess \mathbf{x}_0 :

$$\mathbf{x}_{i+1} = h\mathbf{f}(\mathbf{x}_i) + \mathbf{a}$$

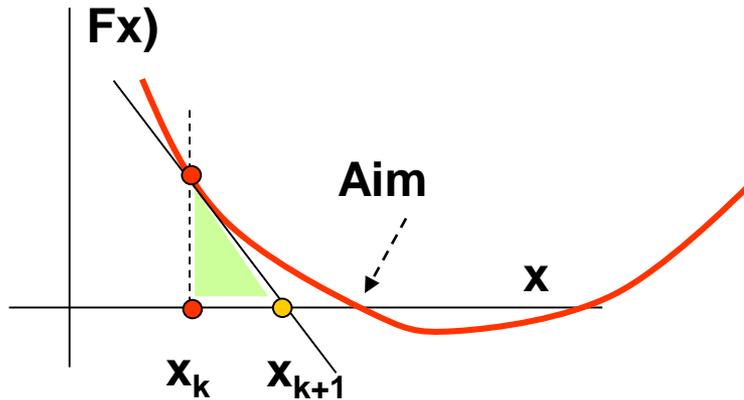
Iterating until $|\mathbf{x}_{i+1} - \mathbf{x}_i| \leq \varepsilon$

The method converges assumed that $h \left| \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right| < 1$

If there is no convergence in N iterations, it is always possible to decrease h in order to fulfil the convergence condition.



Newton-Raphson



Requires solving a linear system of equations using the Jacobian at every iteration

$$F(x) = 0$$

$$F(x_{i+1}) = F(x_i) + \left. \frac{\partial F}{\partial x} \right|_{x_i} (x_{i+1} - x_i) + \dots = 0$$

$$x_{i+1} = x_i - \left[\left. \frac{\partial F}{\partial x} \right|_{x_i} \right]^{-1} F(x_i)$$

$$\left. \frac{\partial F}{\partial x} \right|_{x_i} \Delta x_i = -F(x_i)$$

$$x_{i+1} = x_i + \Delta x_i$$



Newton-Raphson

$$\mathbf{x} = h\mathbf{f}(\mathbf{x}) + \mathbf{a}$$

Starting from an initial guess \mathbf{x}_0 :

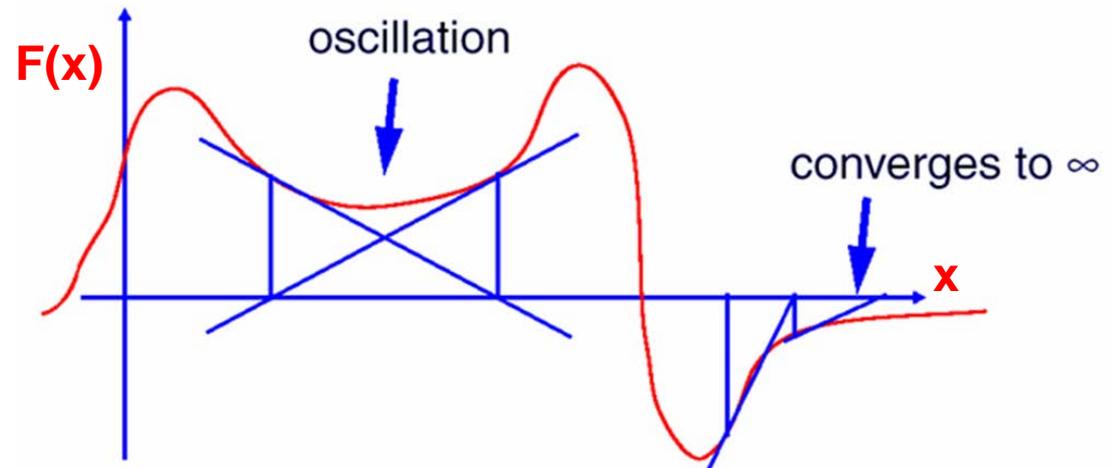
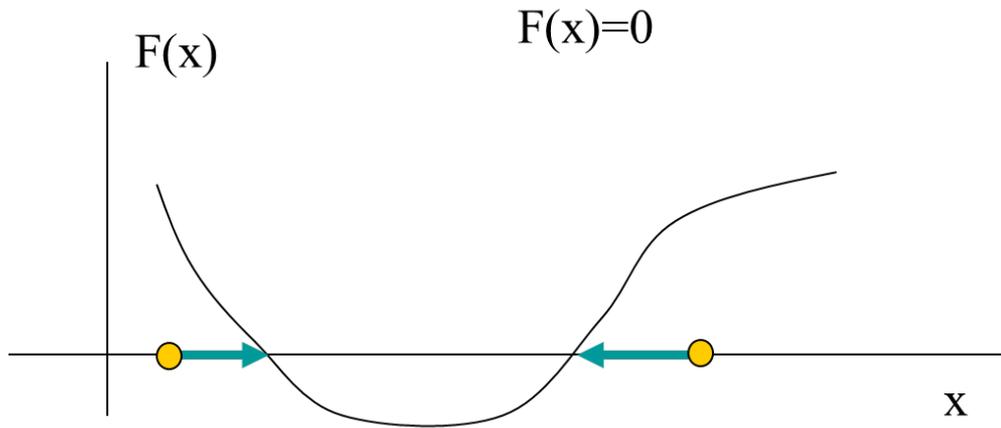
$$\mathbf{x}_{i+1} = \mathbf{x}_i - \left[\mathbf{I} - h \frac{\partial \mathbf{f}(\mathbf{x}_i)}{\partial \mathbf{x}} \right]^{-1} (\mathbf{x}_i - h\mathbf{f}(\mathbf{x}_i) - \mathbf{a})$$

It converges for h small enough

It requires estimating the Jacobian, but has better numerical robustness properties than the successive approximation method



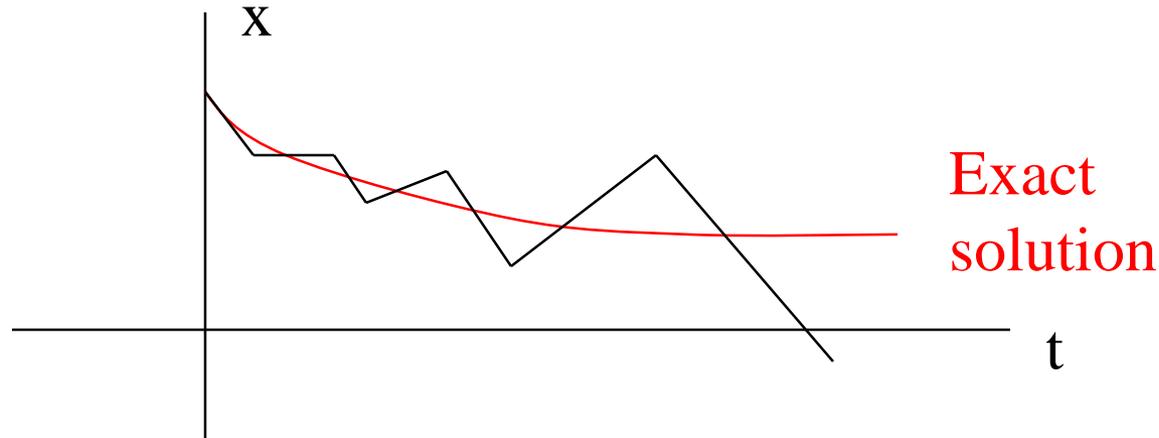
Initial value problems /Convergence





Stability

- ✓ Good integration implies that the error between the numerical and analytical solutions should tend to zero over the integration horizon.
- ✓ Solution stability depends on the integration method and on h
- ✓ A non-stable solution implies oscillations or divergence from the exact one.





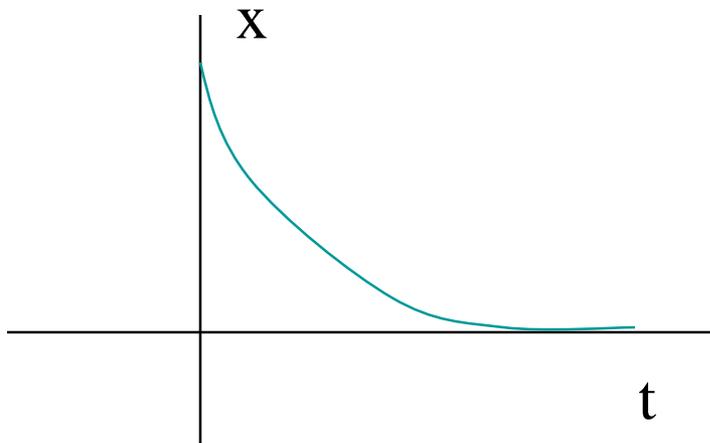
Stability: Example

$$\frac{dx}{dt} = -\mu x \quad x(0) = 1 \quad \text{analytical solution } x(t) = e^{-\mu t}$$

Explicit Euler's
method:

$$x(t+h) = x(t) - h\mu x(t)$$

$$\left| \frac{x(t+h)}{x(t)} \right| = |1 - h\mu| < 1$$



Decreasing $x(t)$
implies: $h < 2/\mu$
Non-oscillatory $x(t)$
implies: $h < 1/\mu$

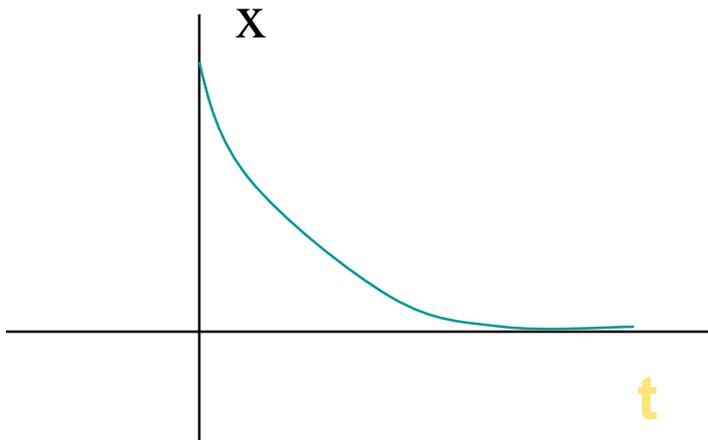


Stability: Example

$$\frac{dx}{dt} = -\mu x \quad x(0) = 1 \quad \text{solución analítica } x(t) = e^{-\mu t}$$

Implicit Euler's
method

$$x(t+h) = x(t) - h\mu x(t+h)$$



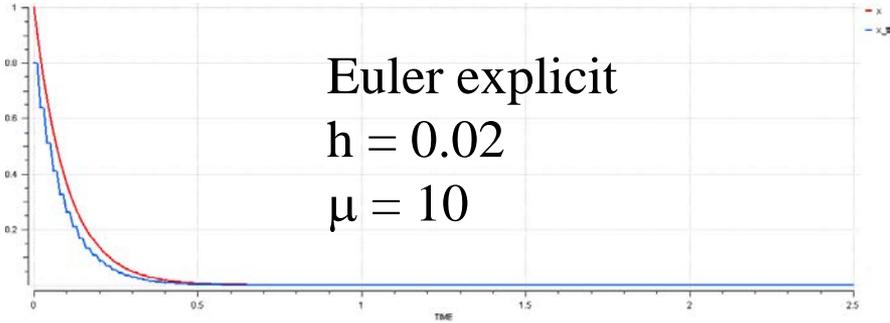
$$\left| \frac{x(t+h)}{x(t)} \right| = \left| \frac{1}{1+h\mu} \right| < 1$$

$x(t)$ is decreasing and non-oscillatory for all $h > 0$

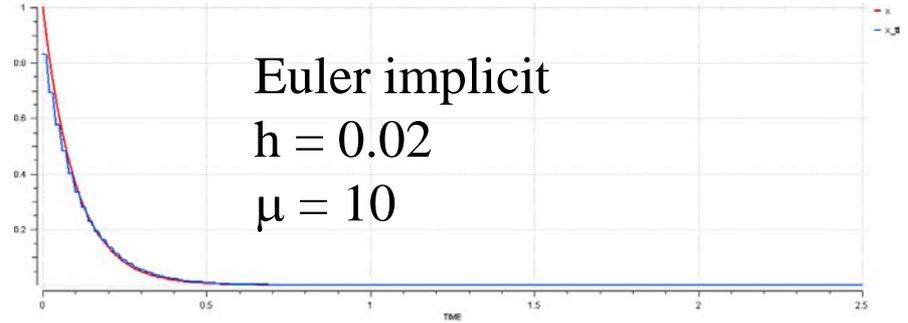


Example

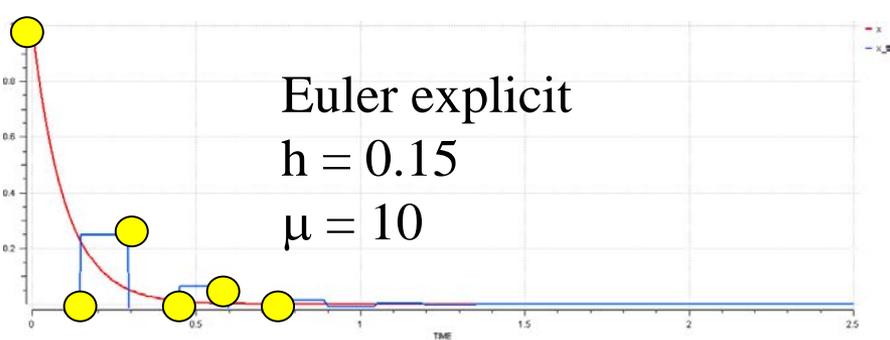
Euler explicit
 $h = 0.02$
 $\mu = 10$



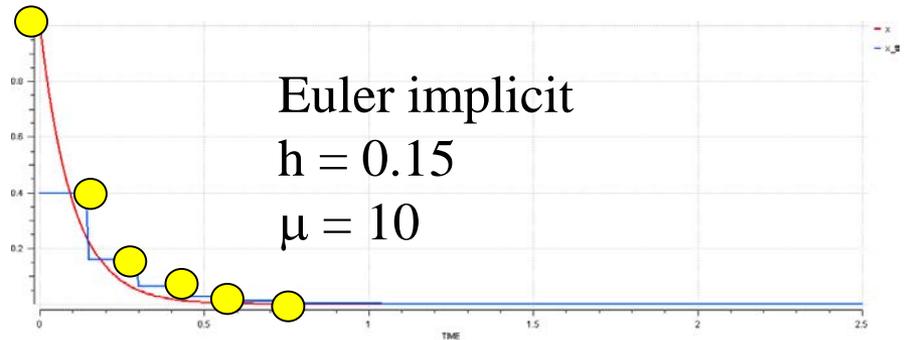
Euler implicit
 $h = 0.02$
 $\mu = 10$



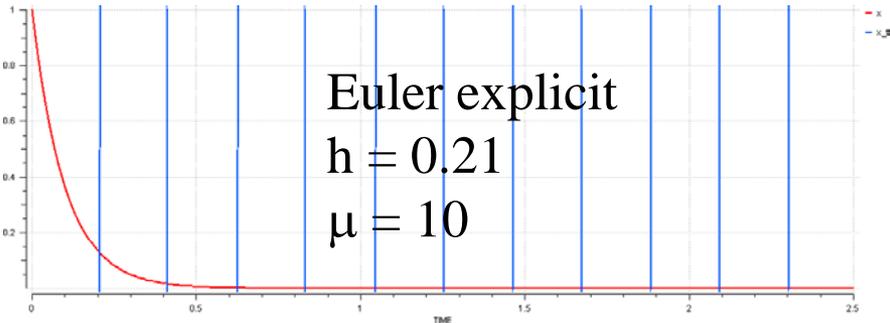
Euler explicit
 $h = 0.15$
 $\mu = 10$



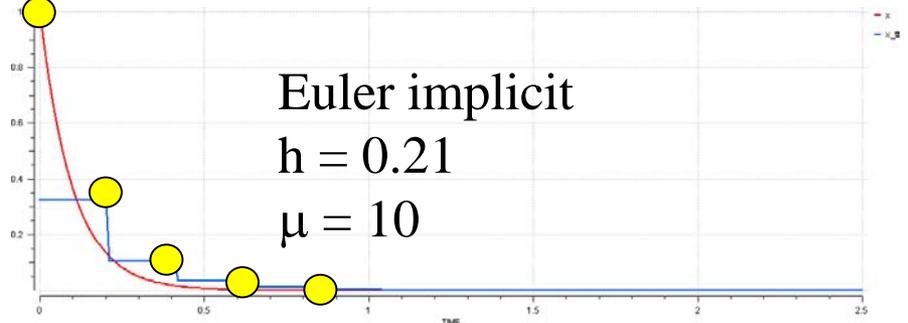
Euler implicit
 $h = 0.15$
 $\mu = 10$



Euler explicit
 $h = 0.21$
 $\mu = 10$



Euler implicit
 $h = 0.21$
 $\mu = 10$





Stability

- ✓ It is difficult, in general, to study the stability of a set of non-linear ODEs. But a local approximation can be made by linearization

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x}, \mathbf{u}) \rightarrow \frac{d\Delta\mathbf{x}}{dt} = \mathbf{A}\Delta\mathbf{x} + \mathbf{B}\Delta\mathbf{u}$$

- ✓ Eigenvalues of the matrix \mathbf{A} allow studying the stability of the solution. It is expressed normally as a bound in the value of $|\lambda h|$ with λ the largest eigenvalue of \mathbf{A}
- ✓ In general, implicit methods give wider ranges of h for the same stability margin, but at the expense of more intensive calculus.



Predictor-corrector methods

These methods combine explicit and implicit formulas to improve the stability of the explicit methods and reduce the computational load of the implicit ones.

They use an explicit formula for prediction and an implicit one for correction

Example: Euler's predictor-corrector:

Prediction:
$$x_p(t+h) = x(t) + hf(x(t), u)$$

Correction:
$$x(t+h) = x(t) + hf(x_p(t+h), u)$$

It is possible to iterate with the corrections



“Stiff” systems

- ✓ Quite often, the model includes fast and slow dynamics
- ✓ If they are very different, this can create integration problems: Slow dynamics requires large h values, but this creates stability problems with the fast dynamics
- ✓ A model is called “stiff” if the ratio of the largest to the smallest eigenvalue of the linearized model is larger than 100 or 1000

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \rightarrow \frac{d\Delta\mathbf{x}}{dt} = \mathbf{A}\Delta\mathbf{x} + \mathbf{B}\Delta\mathbf{u} \quad \text{stiff if } \frac{\max|\operatorname{Re}(\lambda_i)|}{\min|\operatorname{Re}(\lambda_i)|} > 100$$

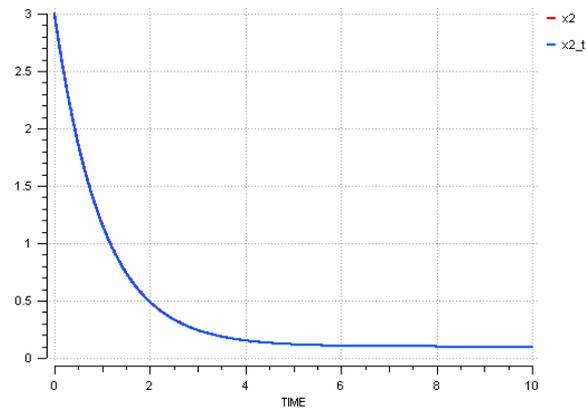
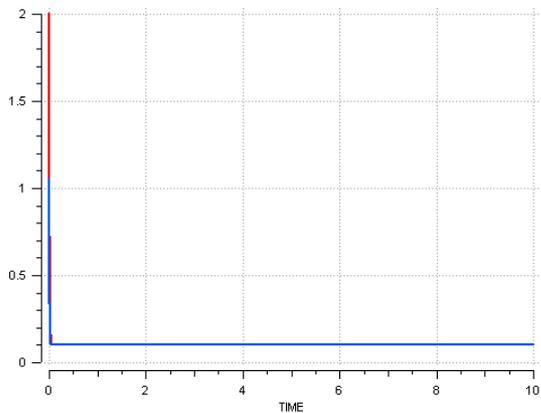


Stiff models

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -100 & 0 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 10 \\ 0 \end{bmatrix} \quad x(0) = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

$$x_1(t) = -1.8(1 - e^{-100t}) + 2$$

$$x_2(t) = 0.1 - \frac{1.8}{99}e^{-100t} + 2.9181e^{-t}$$



Eigenvalues
-1, -100



Stiff models: Example

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -100 & 0 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 10 \\ 0 \end{bmatrix} \quad x(0) = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

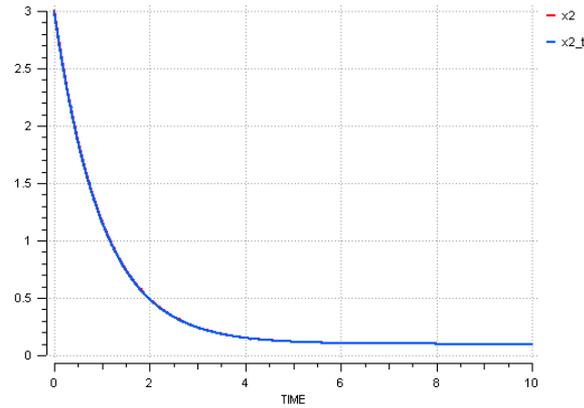
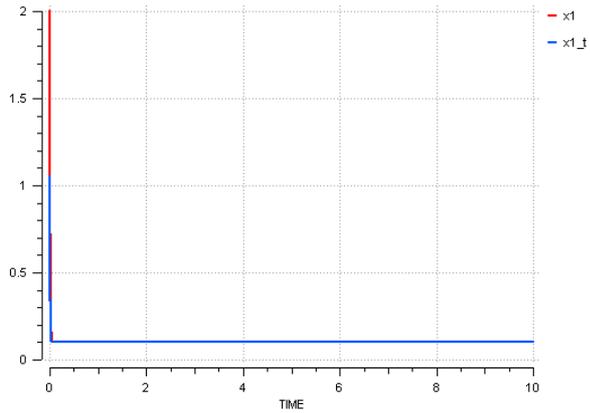
Eigenvalues: -100, -1 Using Euler: $h < 1/100=0.01$

Integrating up to $t = 10$ requires $10/0.01=1000$ iterations, which increases the accumulation errors and integration time

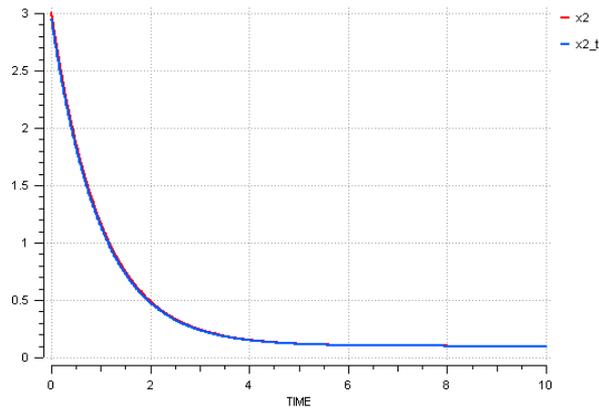
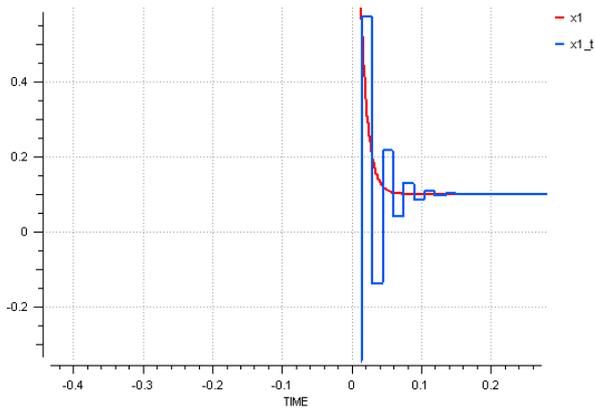
Implicit methods allows integrating with larger values of h



Example stiff



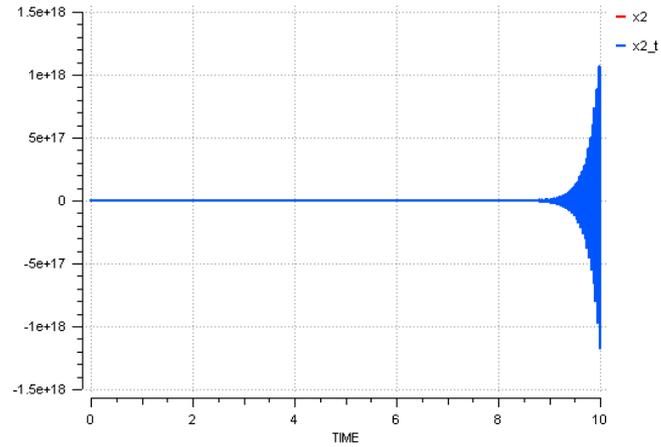
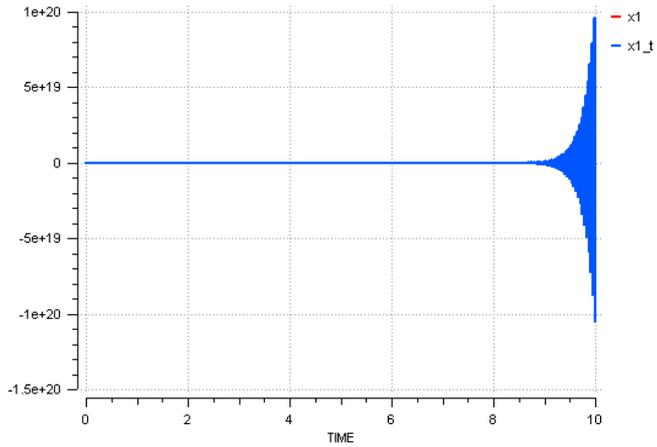
Euler explicit
 $h = 0.005$



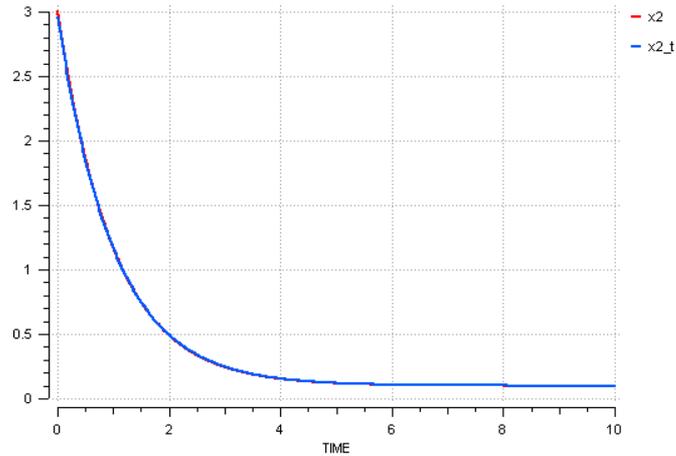
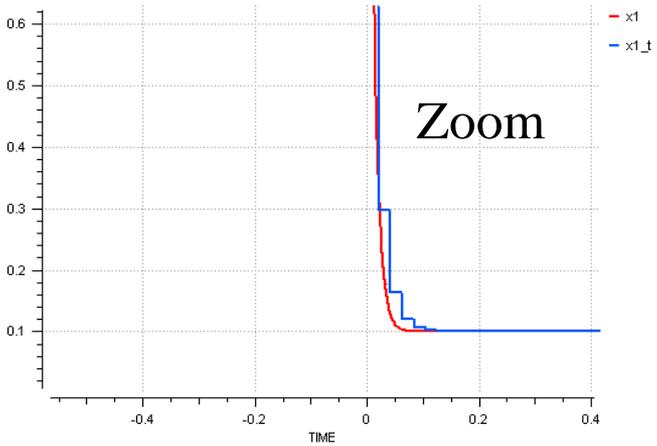
Euler explicit
 $h = 0.015$



Example stiff



Euler explicit
 $h = 0.021$



Euler implicit
 $h = 0.021$



Approximate model

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -100 & 0 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 10 \\ 0 \end{bmatrix} \quad x(0) = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

If the slow and fast dynamics can be separated, the model can be approximated by:

Fast
equation
converted to
algebraic

$$\longrightarrow \begin{bmatrix} 0 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -100 & 0 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 10 \\ 0 \end{bmatrix}$$

$$-100x_1 = 10 \quad \Rightarrow \quad x_1 = 0.1$$

$$\dot{x}_2 = 0.1 - x_2$$

$$x_1(t) = 0.1$$

$$x_2(t) = 0.1 + 2.9e^{-t}$$

Approximation: converting
fast dynamic equations
into algebraic ones

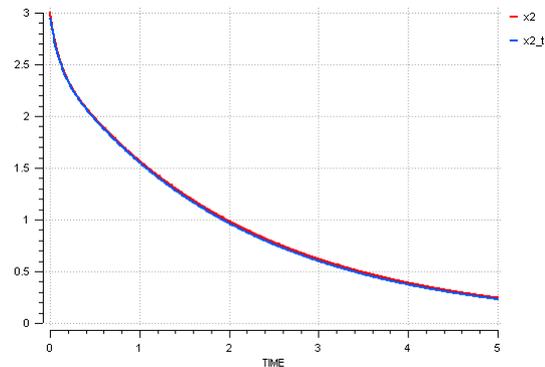
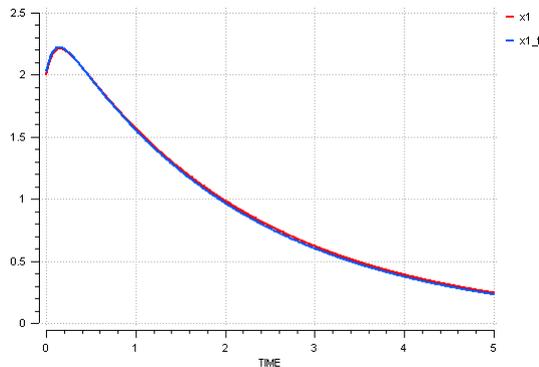
Now the stability limit is $h < 1/1=1$



Stiff models: Example2

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -5 & 4.5363 \\ 4.5363 & -5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad x(0) = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

Eigenvalues: -9.5364, -0.4636

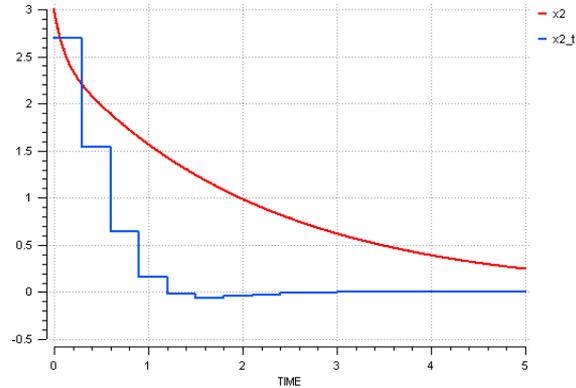
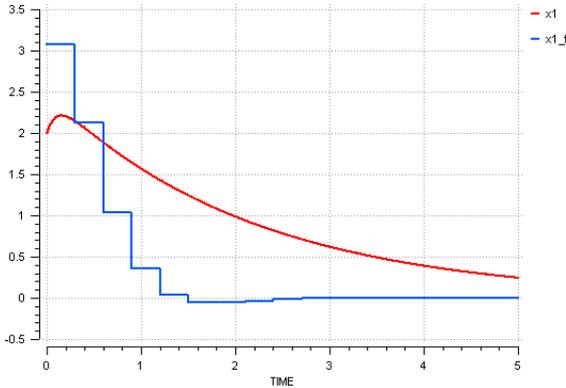


Euler explicit
 $h = 0.01$

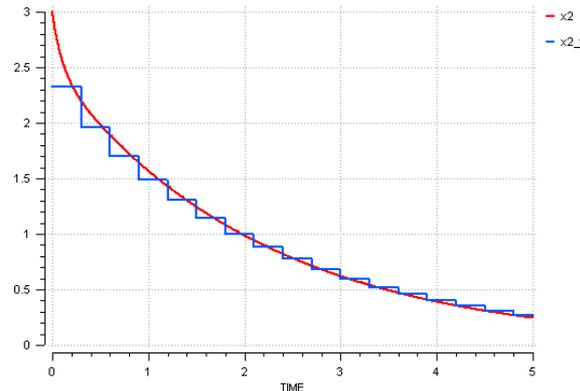
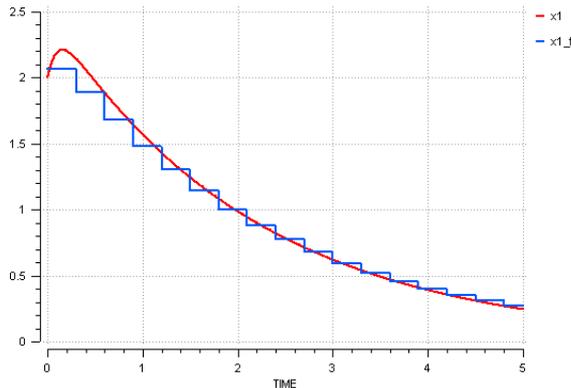
In this example, both, fast and slow dynamics are non-separable
Specific integration methods for these problems are required



Example 2



Euler explicit
 $h = 0.3$

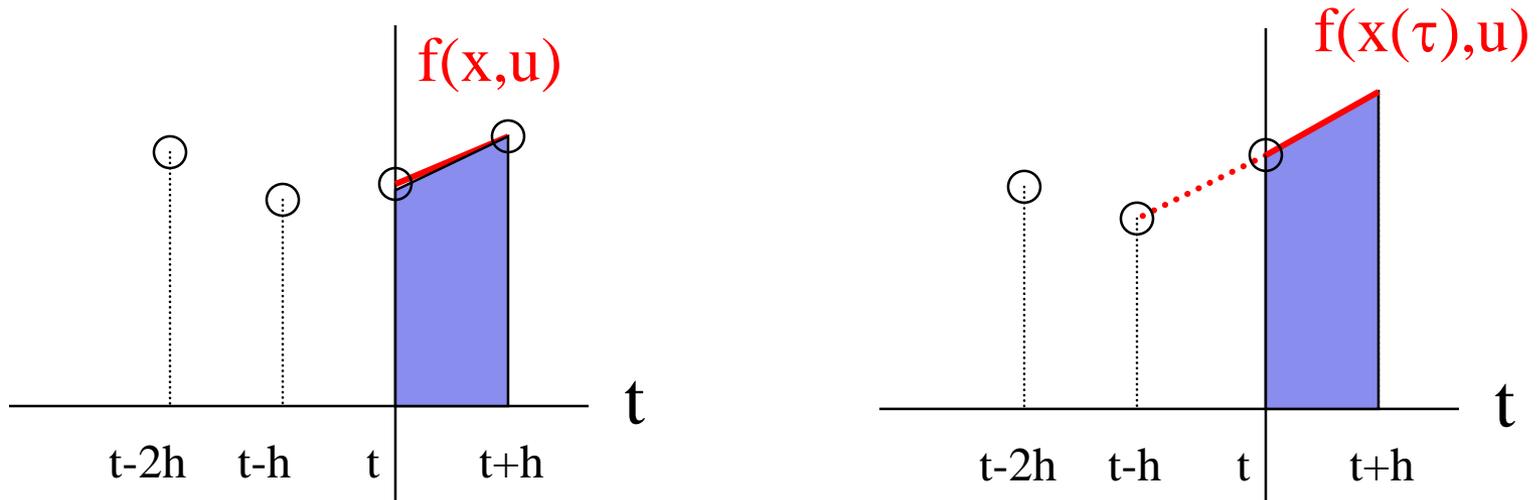


Euler implicit
 $h = 0.3$

A small h would be required at the beginning to capture the x_1 dynamics and then expand h to follow quicker the slow dynamics: variable integration steps



Approximating derivatives



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

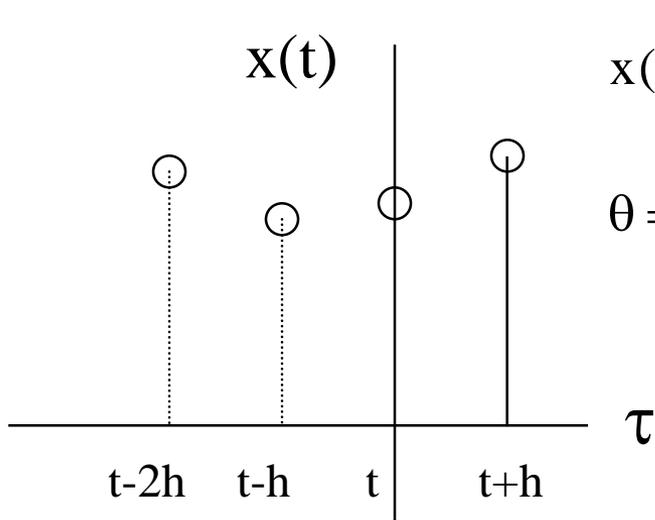
Implicit and explicit integration formulas basically approximate the derivative in $[t, t+h]$ with polynomials of different orders and integrate it.



Backward Difference Formulas



BDF



$$x(\tau) = \left[1 + \theta\Delta + \frac{\theta(\theta+1)}{2!}\Delta^2 + \frac{\theta(\theta+1)(\theta+2)}{3!}\Delta^3 + \dots \right] x(t+h)$$

$$\theta = \frac{\tau - (t+h)}{h}$$

$$\Delta = 1 - q^{-1}$$

Polynomial going through $x(t+h)$ and n previous values.

Differentiating:
$$\dot{x}(\tau) = \frac{1}{h} \left[\Delta + \left(\theta + \frac{1}{2} \right) \Delta^2 + \left(\frac{\theta^2}{2} + \theta + \frac{1}{3} \right) \Delta^3 + \dots \right] x(t+h)$$

At $\tau = t+h$:
$$\dot{x}(t+h) = \frac{1}{h} \left[\Delta + \frac{1}{2} \Delta^2 + \frac{1}{3} \Delta^3 + \dots \right] x(t+h) = f(x(t+h), u)$$



Backward difference formulas



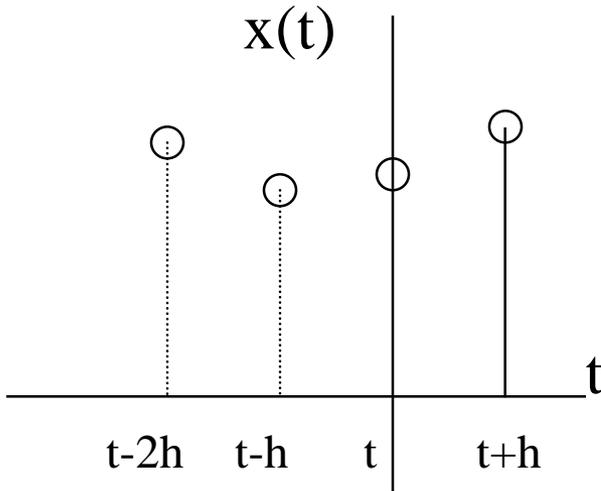
BDF

This provides a formula for computing derivatives:

$$\dot{x}(t+h) = \frac{1}{h} \left[\Delta + \frac{1}{2} \Delta^2 + \frac{1}{3} \Delta^3 + \dots \right] x(t+h)$$

As well as $x(t+h)$ in a ODE model:

$$\frac{1}{h} \left[\Delta + \frac{1}{2} \Delta^2 + \frac{1}{3} \Delta^3 + \dots \right] x(t+h) = f(x(t+h), u(t+h))$$



For order 3: BDF3, implicit method

above $n=5$ the method
is unstable

$$x(t+h) = \frac{18}{11} x(t) - \frac{9}{11} x(t-h) + \frac{2}{11} x(t-2h) + \frac{6}{11} f(x(t+h), u(t+h))h$$



Errors

- ✓ There are several sources of errors in the integration of ODEs:
- ✓ **Precision** errors in the integration formulas:
 - Integration step-size h
 - Order of the approximation (polynomial, Taylor series,..)
- ✓ **Rounding** errors due to machine precision
- ✓ **Accumulation** errors due to the iterative nature of the integration methods



Cutting off errors

Cutting off or precision errors are related to the precision of the approximation of $f(x(\tau), u)$ by a polynomial or Taylor series expansion.

$$x(t+h) = x(t) + hf(x(t), u)$$

$$x(t+h) = x(t) + 0.5h[3f(x(t), u) - f(x(t-h), u)]$$

In a method of order n the error is of order h^{n+1}

Using higher order methods, it is possible to integrate the model with a larger h , but maintaining the precision.



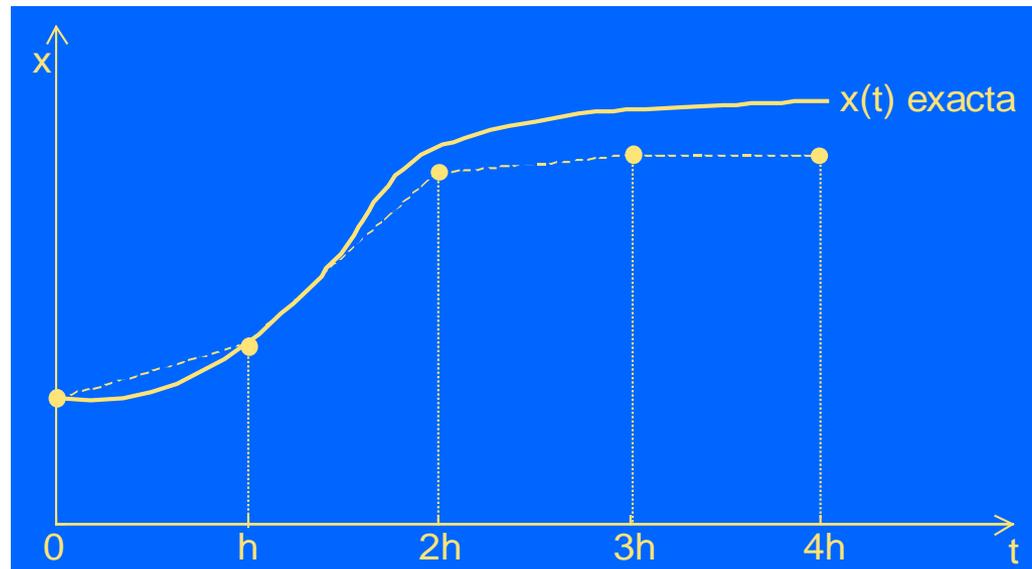
Rounding errors

They depend on the computer numerical precision, which is affected by:

- Rounding due to finite word length
- Accumulation of errors
- Machine type
- Compiler type
- Data type: integer, single precision, double precision,...
- ...



Accumulation errors



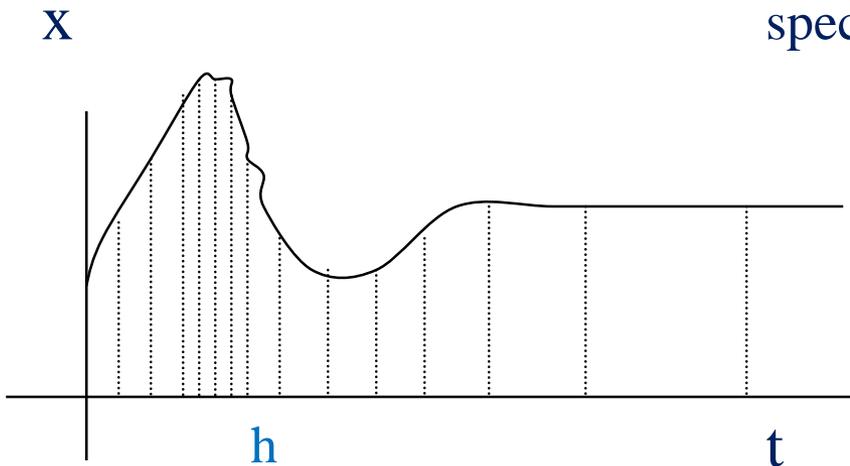
Except at $t = 0$, the solution at $t + ih$ is affected by the error in $x(t+(i-1)h)$, accumulating the errors over the integration horizon, which is proportional to $1/h$



Variable step-size methods

Taking into account all type of errors, an integration method of order n has an error of order h^n

Variable step-size methods adjust continuously the integration step size h , in order to keep the total error bellow a pre-specified level.



Runge-Kutta-Fehlberg RK45 compares the results at $t+h$ of RK4 and RK5 and accepts, reduces or enlarges h according to the difference w.r.t. a certain tolerance



Adjusting h

Two problems:

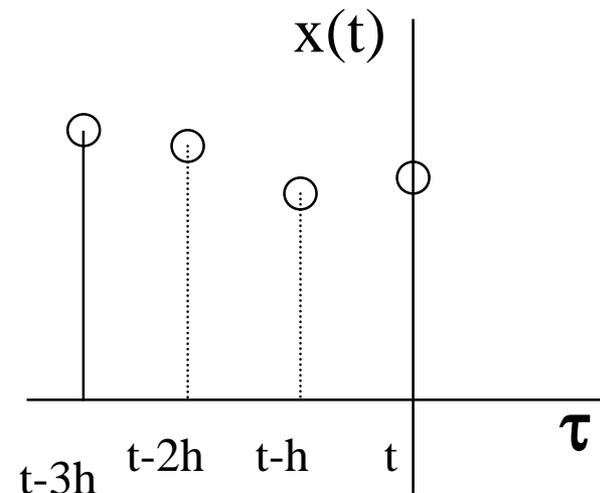
- ✓ Estimate the error.
- ✓ Re-start the algorithm, that is based on previous values of $x(t)$ computed at regular intervals h

Error estimation is based on the use of a BDF extrapolation polynomial going through $x(t)$:

$$x(\tau) = \left[1 + \theta\Delta + \frac{\theta(\theta+1)}{2!}\Delta^2 + \frac{\theta(\theta+1)(\theta+2)}{3!}\Delta^3 + \dots \right] x(t)$$

$$\theta = \frac{\tau - t}{h}$$

$$\Delta x(t) = (1 - q^{-1})x(t) = x(t) - x(t - h)$$





Nordsieck' vector

Derivatives of different orders can be computed by differentiation:

$$\dot{\mathbf{x}}(\tau) = \frac{1}{h} \left[\Delta + \left(\theta + \frac{1}{2} \right) \Delta^2 + \left(\frac{\theta^2}{2} + \theta + \frac{1}{3} \right) \Delta^3 + \dots \right] \mathbf{x}(t) \quad \theta = \frac{\tau - t}{h}$$

$$\ddot{\mathbf{x}}(\tau) = \frac{1}{h^2} \left[\Delta^2 + (\theta + 1) \Delta^3 + \dots \right] \mathbf{x}(t)$$

At $\tau = t$:

$$\dot{\mathbf{x}}(t)h = \left[\Delta + \frac{1}{2} \Delta^2 + \frac{1}{3} \Delta^3 + \dots \right] \mathbf{x}(t)$$

$$\ddot{\mathbf{x}}(t)h^2 = \left[\Delta^2 + \Delta^3 + \dots \right] \mathbf{x}(t)$$

For a third order approximation:

$$\begin{pmatrix} \mathbf{x}(t) \\ \dot{\mathbf{x}}(t)h \\ \ddot{\mathbf{x}}(t)\frac{h^2}{2} \\ \ddot{\mathbf{x}}(t)\frac{h^3}{6} \end{pmatrix} = \frac{1}{6} \begin{pmatrix} 6 & 0 & 0 & 0 \\ 11 & -18 & 9 & -2 \\ 6 & -15 & 12 & -3 \\ 1 & -3 & 3 & -1 \end{pmatrix} \begin{pmatrix} \mathbf{x}(t) \\ \mathbf{x}(t-h) \\ \mathbf{x}(t-2h) \\ \mathbf{x}(t-3h) \end{pmatrix}$$



Adjusting step-size h

Nordsieck' vector allows computing the error e made with step-size h

$$e \leq \frac{1}{n!} \left| \frac{d^{n+1}x}{dt^{n+1}} \right| h^n_{x \in [x(t), x(t+h)]}$$

If the maximum desired error is E, the step-size h should be corrected to h_1 so that:

$$E = \frac{1}{n!} \frac{d^{n+1}x}{dt^{n+1}} h_1^n$$

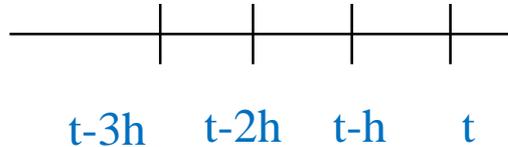
$$\begin{pmatrix} x(t) \\ \dot{x}(t)h \\ \ddot{x}(t)\frac{h^2}{2} \\ \ddot{\ddot{x}}(t)\frac{h^3}{6} \end{pmatrix} = \frac{1}{6} \begin{pmatrix} 6 & 0 & 0 & 0 \\ 11 & -18 & 9 & -2 \\ 6 & -15 & 12 & -3 \\ 1 & -3 & 3 & -1 \end{pmatrix} \begin{pmatrix} x(t) \\ x(t-h) \\ x(t-2h) \\ x(t-3h) \end{pmatrix}$$

Computing the ratio of both expressions, the new step-size h_1 can be computed from:

$$\frac{e}{E} = \left(\frac{h}{h_1} \right)^n$$



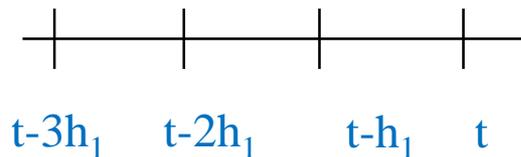
Re-start with h_1



It is possible to change h to h_1 without reducing the order of the BDF to 1 to re-start the algorithm

$$\begin{pmatrix} x(t) \\ \dot{x}(t)h \\ \ddot{x}(t)\frac{h^2}{2} \\ \ddot{\ddot{x}}(t)\frac{h^3}{6} \end{pmatrix} = \frac{1}{6} \begin{pmatrix} 6 & 0 & 0 & 0 \\ 11 & -18 & 9 & -2 \\ 6 & -15 & 12 & -3 \\ 1 & -3 & 3 & -1 \end{pmatrix} \begin{pmatrix} x(t) \\ x(t-h) \\ x(t-2h) \\ x(t-3h) \end{pmatrix}$$

Past values of $x(t-h_1)$ can be estimated using the inverse Nordsieck's formula:



$$\begin{pmatrix} x(t) \\ x(t-h_1) \\ x(t-2h_1) \\ x(t-3h_1) \end{pmatrix} = 6 \begin{pmatrix} 6 & 0 & 0 & 0 \\ 11 & -18 & 9 & -2 \\ 6 & -15 & 12 & -3 \\ 1 & -3 & 3 & -1 \end{pmatrix}^{-1} \begin{pmatrix} x(t) \\ \dot{x}(t)h_1 \\ \ddot{x}(t)\frac{h_1^2}{2} \\ \ddot{\ddot{x}}(t)\frac{h_1^3}{6} \end{pmatrix}$$



Gear's method

- ✓ Oriented to stiff model
- ✓ Implicit BDF formulas of several orders to approximate derivatives
- ✓ Variable step-size
- ✓ Selection of the order and step-size h is made from an index that considers the integration error, number of steps required and associated computing time.
- ✓ The implicit equation is solved using Newton-Raphson. If there is no convergence after three iterations, h is reduced and the computation is repeated.
- ✓ Implemented as DIFFSUB routine.



DAE models

DAE: Differential-algebraic equations are coupled algebraic and differential equations or implicit differential equations, where the derivatives of all variables cannot be explicitly computed.

Implicit	$F\left(\frac{dx}{dt}, x, u\right) = 0$	$\frac{dx}{dt} = f(x, z, u)$	Semi-explicit,
		$0 = g(x, z, u)$	z algebraic variables

Integration methods for ODEs require explicit values of the derivatives (and do not consider constraints on the state imposed by algebraic equations), so that they cannot be used directly in the integration of DAEs.

DAE models require special integration methods



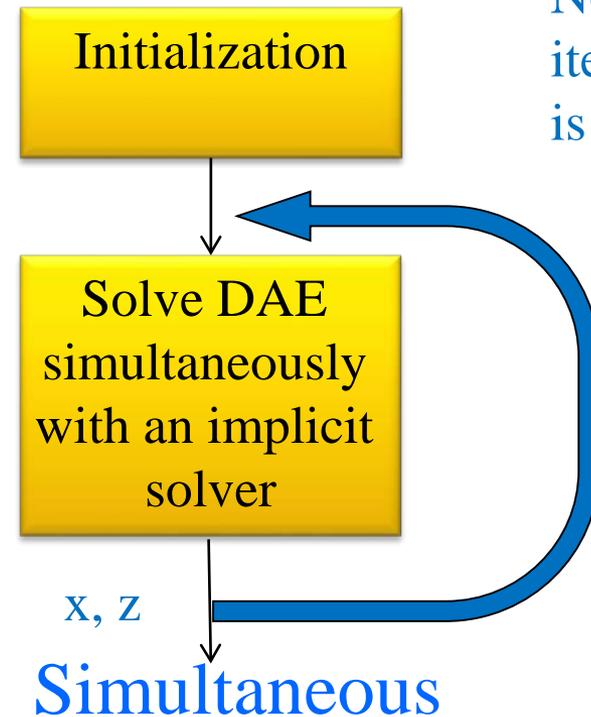
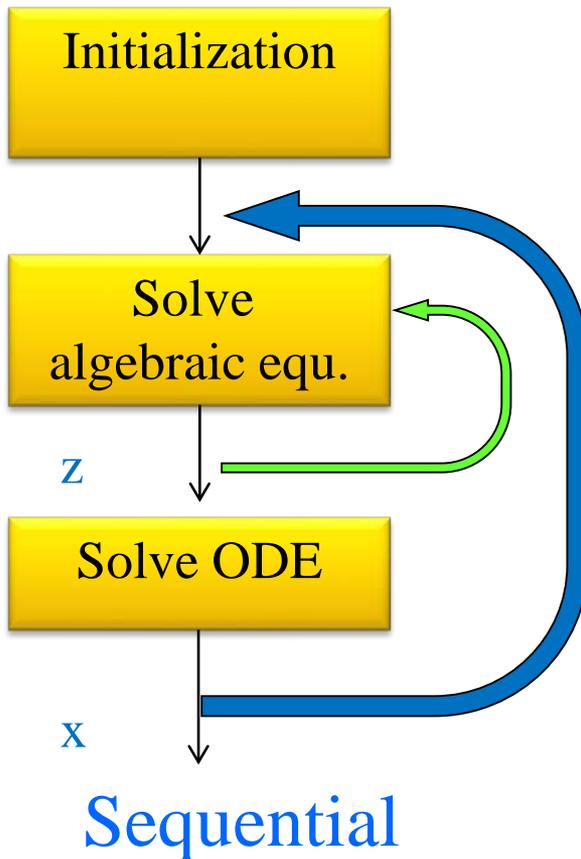
Integration of semi-explicit DAEs

✓ Approaches:

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

$$0 = g(x, z, u)$$

Only one
single
Newton
iteration
is needed





Initial conditions

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

$$g(x, z, u) = 0$$

In general, even for solving semi-explicit DAE equations using a sequential approach, good initial values of the algebraic variables z are required as the solution of $g(x, z, u) = 0$ may depend on them

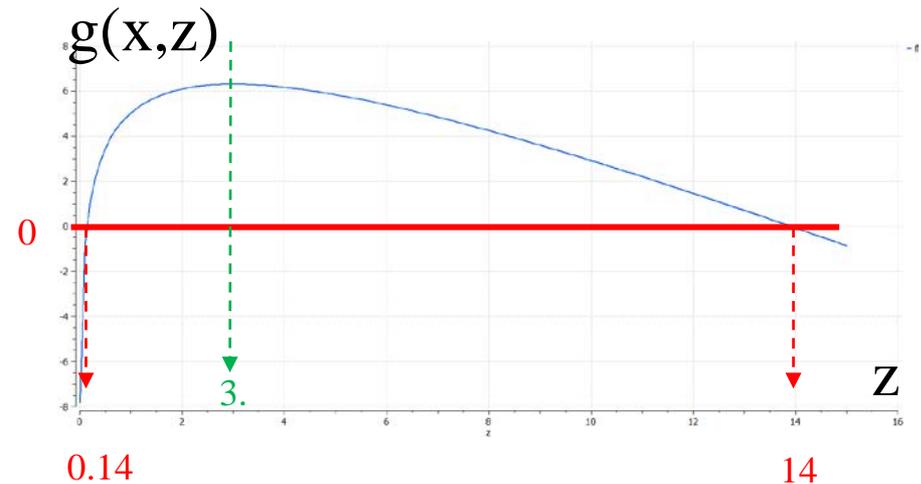
$$\frac{dx}{dt} + 3xz - x^2 = u$$

$$xu + x \log(z) = z$$

$$u(t) = 2$$

$$x(0) = 3$$

$$z(0) = ?$$



Starting with values of z below or above $z = 3$ will give very different results

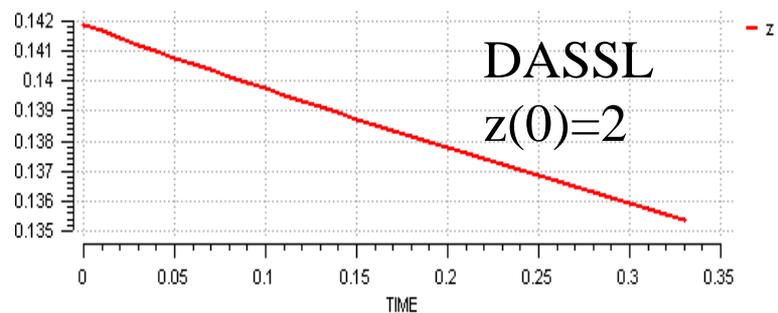
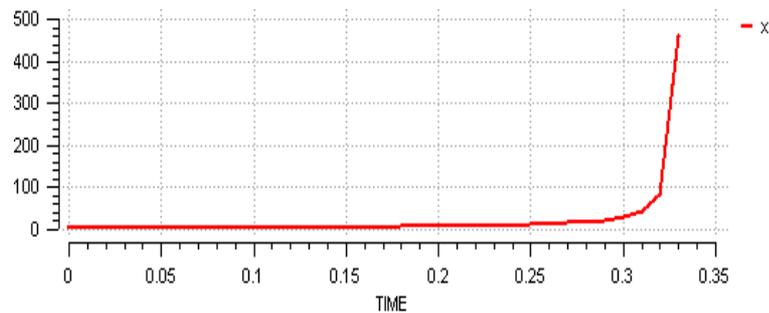
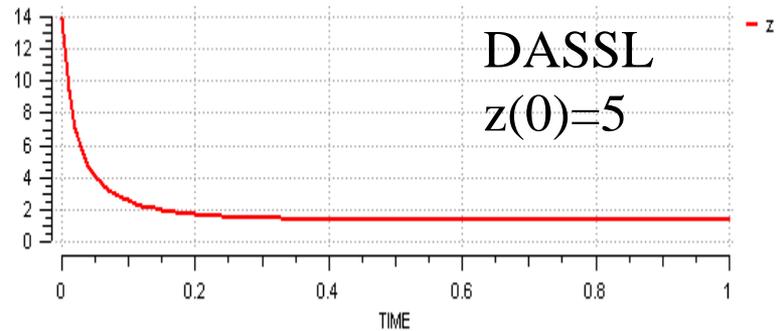
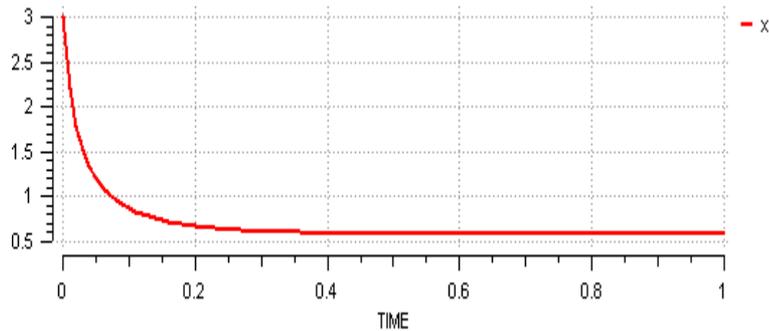


DAE example

$$\frac{dx}{dt} + 3xz - x^2 = u \quad x(0) = 3$$

$$xu + x \log(z) = z \quad z(0) = ?$$

$$u(t) = 2$$





Integration of semi-explicit DAEs



A set of DAEs can be converted to a set of ODEs if the algebraic equations are differentiated with respect to time.

$$\begin{array}{l} \frac{dx}{dt} = f(x, z, u) \\ g(x, z, u) = 0 \end{array} \quad \begin{array}{l} \text{Computing the time} \\ \text{derivative of } g, \text{ the} \\ \text{DAE can} \\ \xrightarrow{\text{yellow arrow}} \\ \text{be converted into an} \\ \text{ODE if } \partial g / \partial z \neq 0 \end{array} \quad \begin{array}{l} \frac{dx}{dt} = f(x, z, u) \\ \frac{\partial g}{\partial x} \frac{dx}{dt} + \frac{\partial g}{\partial z} \frac{dz}{dt} + \frac{\partial g}{\partial u} \frac{du}{dt} = 0 \\ \frac{\partial g}{\partial x} f(x, z, u) + \frac{\partial g}{\partial z} \frac{dz}{dt} + \frac{\partial g}{\partial u} \frac{du}{dt} = 0 \end{array}$$

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

$$g(x, z, u) = 0$$



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

$$\frac{dz}{dt} = r(x, z, u)$$



Initial conditions



$$\frac{d x}{d t} = f(x, z, u)$$

$$g(x, z, u) = 0$$



$$\frac{d x}{d t} = f(x, z, u)$$

$$\frac{d z}{d t} = r(x, z, u)$$

z cannot be given any initial condition $z(0)$, it requires **consistent initial conditions** fulfilling

$$g(x(0), z(0), u(0)) = 0$$

- ✓ Notice that the ODE converted system uses values of du/dt and not only of u
- ✓ If $\partial g/\partial z$ is singular, the algebraic equations can be differentiated more times in order to obtain an ODE



Index of a DAE

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

$$g(x, z, u) = 0$$



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

$$\frac{\partial g}{\partial x} \frac{dx}{dt} + \frac{\partial g}{\partial z} \frac{dz}{dt} + \frac{\partial g}{\partial u} \frac{du}{dt} = 0$$

The **index** of a DAE is the number of times needed to differentiate the DAEs in order to obtain a system of ODEs.

A differential index of 1 is called low index, while it is called High index if it is 2 or larger.



DAE

$$\frac{d \mathbf{x}}{d t} = \mathbf{f}(\mathbf{x}, \mathbf{u})$$

Example

$$g(\mathbf{x}, \mathbf{u}) = 0$$

If the algebraic equation involve only the \mathbf{x} variables, then it is not possible to integrate independently both equations and DEA integration methods and index reduction are required.

$$\frac{d x_1}{d t} = f_1(x_1, x_2, u)$$

$$\frac{d x_2}{d t} = f_2(x_1, x_2, u)$$

$$x_1^2 + x_2^2 = 25$$

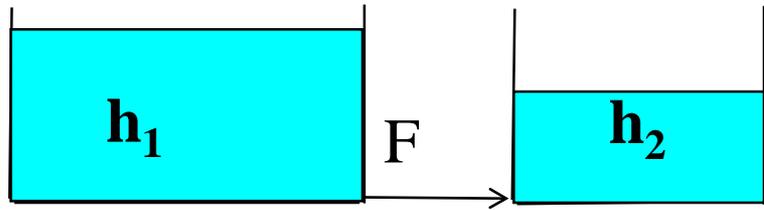
Consistent initial values of the state variables \mathbf{x} are also required

$$x_1^2(0) + x_2^2(0) = 25$$



Communicating vessels

Index 1



$$A_1 \frac{dh_1}{dt} = -k\sqrt{h_1 - h_2}$$

Constant total volume V

$$A_2 \frac{dh_2}{dt} = k\sqrt{h_1 - h_2}$$

$$A_1 \frac{dh_1}{dt} + A_2 \frac{dh_2}{dt} = 0$$

$$A_1 h_1 + A_2 h_2 = V$$

$$h_1 = (V - A_2 h_2) / A_1$$

← h_1 and dh_1/dt can be recovered from here

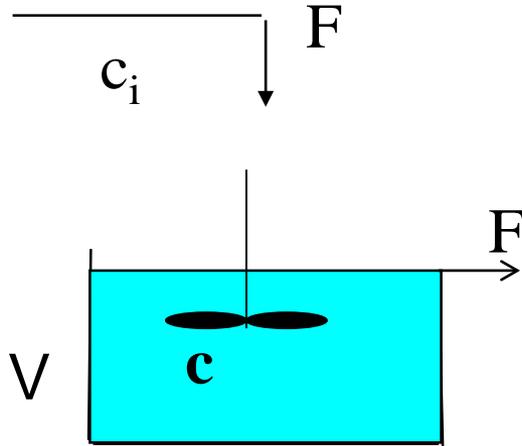
h_1 and h_2 linked by an algebraic equation

$$A_2 \frac{dh_2}{dt} = k\sqrt{h_1 - h_2} = k\sqrt{(V - A_2 h_2) / A_1 - h_2}$$

Differentiating once the algebraic equation, and working out h_1 , after substitution in the second equation, results a single ODE in h_2 :



Index 1 problem



Now the algebraic equation is on c (control problem: given c_i , find F such that $c = \phi(t)$ time function)

$$V \frac{dc}{dt} = F(c_i - c)$$

$$c = \phi(t)$$

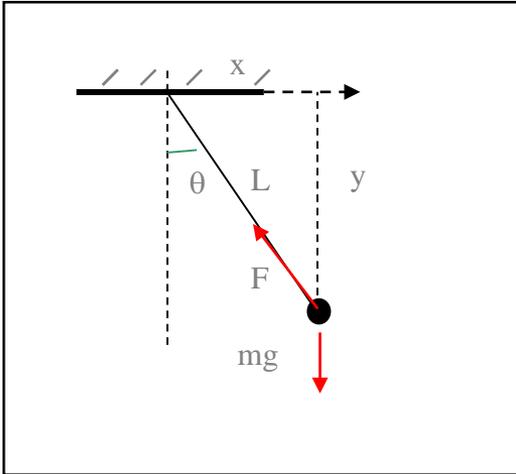


Differentiating
once

$$\left. \begin{array}{l} V \frac{dc}{dt} = F(c_i - c) \\ \frac{dc}{dt} = \frac{d\phi(t)}{dt} = \varphi(t) \end{array} \right\} F = \frac{V}{c_i - c} \varphi(t)$$



Pendulum (Index 2 problem)



The structural singularity is created by inadequate modelling: using cylindrical coordinate, the problem can be described with a single variable θ without bonds

$$m \frac{dv_x}{dt} = -F \frac{x}{L}$$

$$\frac{dx}{dt} = v_x$$

$$m \frac{dv_y}{dt} = -F \frac{y}{L} - mg$$

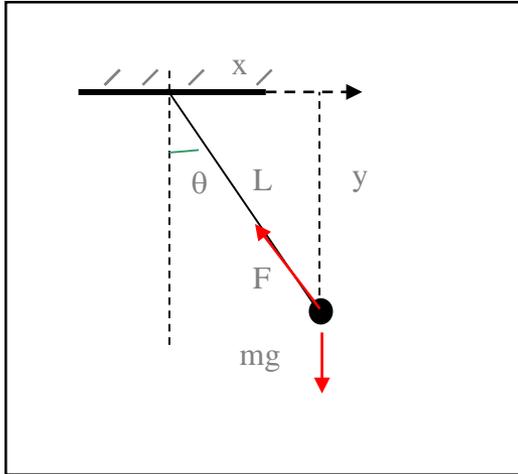
$$\frac{dy}{dt} = v_y$$

$$x^2 + y^2 = L^2 \quad \text{x and y are bond by the string}$$

5 equations, and 5 variables. The algebraic equation imposes a link among state variables. The bond equation is differentiated twice to find equations that provide the value of two state variables and the other two are computed from the bond and its derivative, as direct integration of ODEs cannot be performed.



Pendulum (Index 2 problem)



$$m \frac{dv_x}{dt} = -F \frac{x}{L} \quad \frac{dx}{dt} = v_x$$

$$m \frac{dv_y}{dt} = -F \frac{y}{L} - mg \quad \frac{dy}{dt} = v_y$$

$$x^2 + y^2 = L^2$$

$$x^2 + y^2 = L^2 \Rightarrow 2xv_x + 2yv_y = 0 \Rightarrow$$

$$\Rightarrow 2x \frac{dv_x}{dt} + 2v_x^2 + 2y \frac{dv_y}{dt} + 2v_y^2 = 0$$

Obtained by
differentiation
(twice)

It is possible to find a subset without redundancy and, then, compute the other variables from the bond and the differentiated equations

$$\rightarrow y = \sqrt{L^2 - x^2}$$

$$m \frac{dv_x}{dt} = -F \frac{x}{L} \quad \frac{dx}{dt} = v_x$$

$$v_y = -\frac{xv_x}{y}$$

$$\frac{dv_y}{dt} = \frac{-1}{y} \left[-x \frac{F_x}{mL} + v_x^2 + v_y^2 \right]$$



DAE integration methods: DASSL



$$F\left(\frac{dx}{dt}, x, u\right) = 0$$

$$\dot{x}(t+h) \approx \frac{x(t+h) - \text{old}(x)}{h}$$

Implicit or semi-explicit differential equations can be solved approximating the derivatives at $t+h$ by their variable order BDF expressions and solving the resulting implicit algebraic equation by Newton-Raphson. Initial values for $x(t+h)$ for solving this algebraic equation are computed by extrapolation.

$$F\left(\frac{x(t+h) - \text{old}(x(t))}{h}, x(t+h), u(t+h)\right) = 0$$

An initial value for $x(t+h)$ is required for NR

Variable order BDF approximations of dx/dt (BDF1 to 5) and variable step size h in order to bound the integration error



DASSL

3rd order BDF
approximation of
derivatives:

$$\begin{pmatrix} \mathbf{x}(t+h) \\ \dot{\mathbf{x}}(t+h)h \\ \ddot{\mathbf{x}}(t+h)\frac{h^2}{2} \\ \ddot{\mathbf{x}}(t+h)\frac{h^3}{6} \end{pmatrix} = \frac{1}{6} \begin{pmatrix} 6 & 0 & 0 & 0 \\ 11 & -18 & 9 & -2 \\ 6 & -15 & 12 & -3 \\ 1 & -3 & 3 & -1 \end{pmatrix} \begin{pmatrix} \mathbf{x}(t+h) \\ \mathbf{x}(t) \\ \mathbf{x}(t-h) \\ \mathbf{x}(t-2h) \end{pmatrix}$$

Initialization of DASSL: $\mathbf{F}\left(\frac{\mathbf{x}(h) - \mathbf{x}(0)}{h}, \mathbf{x}(h), \mathbf{u}(h)\right) = 0$ 1st order

$$\mathbf{F}\left(\frac{1}{h}\left(\frac{3}{2}\mathbf{x}(2h) - 2\mathbf{x}(h) + \frac{1}{2}\mathbf{x}(0)\right), \mathbf{x}(2h), \mathbf{u}(2h)\right) = 0$$
 2nd order

$$\mathbf{F}\left(\frac{1}{6h}(11\mathbf{x}(3h) - 18\mathbf{x}(2h) + 9\mathbf{x}(h) - 2\mathbf{x}(0)), \mathbf{x}(3h), \mathbf{u}(3h)\right) = 0$$
 3rd order

... Variable step h as in Gear's



DASSL: initialization

Initial values for the Newton-Raphson iteration:

It may be difficult to propose good initial values of the algebraic variables and also of $\mathbf{x}(t+h)$ when significant dynamic changes are present, if the derivatives are not given explicitly.

$$\frac{\mathbf{x}(h) - \mathbf{x}(0)}{h} \approx f(\mathbf{x}(h), \mathbf{u}(0))??$$

$$\mathbf{F}\left(\frac{\mathbf{x}(h) - \mathbf{x}(0)}{h}, \mathbf{x}(h), \mathbf{u}(h)\right) = 0 \quad \text{1st order}$$

$$\mathbf{F}\left(\frac{1}{h}\left(\frac{3}{2}\mathbf{x}(2h) - 2\mathbf{x}(h) + \frac{1}{2}\mathbf{x}(0)\right), \mathbf{x}(2h), \mathbf{u}(2h)\right) = 0$$

2nd order

...

Whenever possible, a reduction of the DAE to index zero (ODE) should be made in order to facilitate the integration.

Numerical problems are normally associated to index 1 or higher DAEs and small h



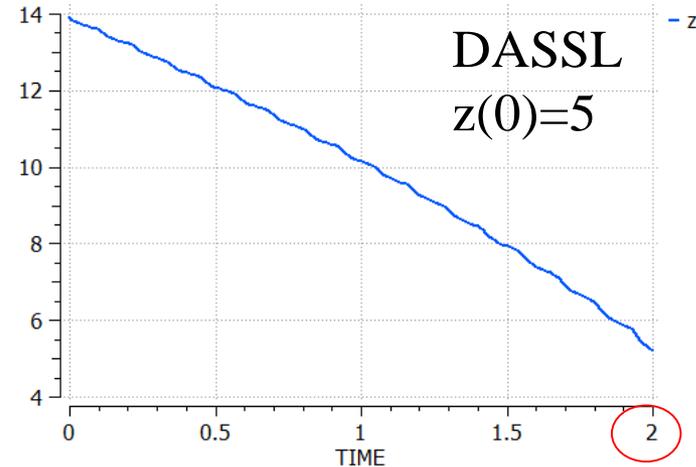
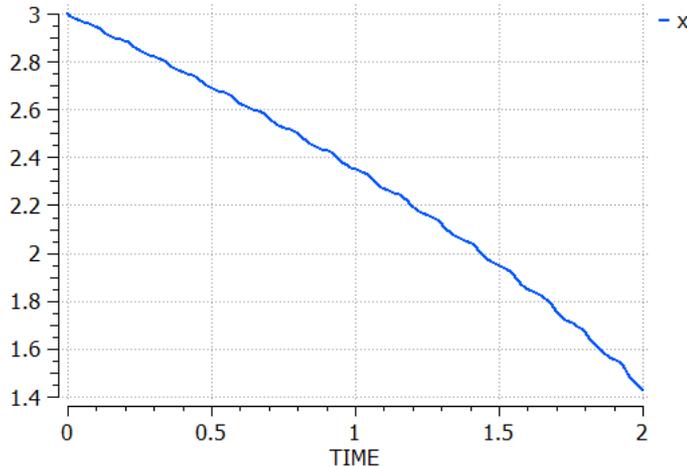
DAE, example

$$\sin\left(\frac{dx}{dt} + 3xz\right) - x \frac{dx}{dt} = u \quad x(0) = 3 \quad \text{Implicit DAE}$$

$$xu + x \log(z) = z \quad z(0) = ?$$

$$u(t) = 2$$

Stiff: fast and slow dynamics





EcosimPro, gProms, Dymola, Jacobian, Abacus, Aspen Dynamics...



1. Compute the DAE model index
2. Reduce it to zero (or one) by differentiation of the bound equations.
3. Select the best states for the reduced problem
4. Solve the reduced DAE problem with a method such as DASSL, DASPK, IDAS, etc.

$$F\left(\frac{\mathbf{x} - \text{old}(\mathbf{x})}{h}, \mathbf{x}, \mathbf{u}\right) = 0 \quad \mathbf{x}(t_0) = \mathbf{x}_0 \quad \mathbf{J}_x = \frac{\partial F}{\partial \mathbf{x}}$$

Jacobian

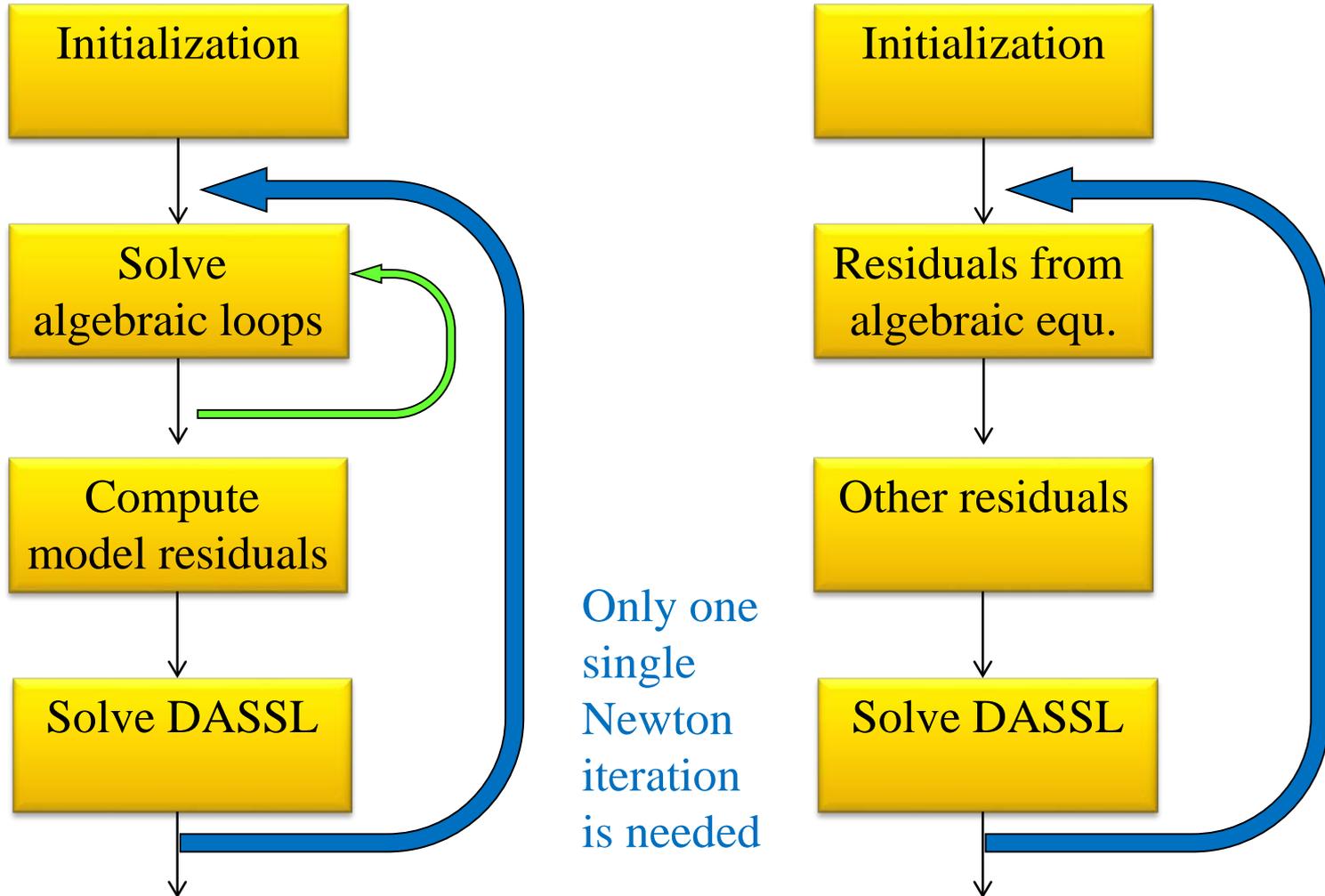
Compute the Jacobian
and obtain $\Delta \mathbf{x}$ with a
linear solver, iterating
until convergence

$$\left(\frac{\mathbf{J}_{\dot{\mathbf{x}}}}{h} + \mathbf{J}_x\right) \Delta \mathbf{x}_i = -F(\dot{\mathbf{x}}_i, \mathbf{x}_i, t)$$

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \Delta \mathbf{x}_i \quad \text{Newton step}$$



DAEs and algebraic loops





Integration methods in EcosimPro



METHOD	DAE solver	ODE solver	Variable/ Fixed integration Step	IMPlicit EXPlicit	Stiff system oriented	Sparse solver
DASSL	YES	YES	Var-Step	IMP	YES	NO
DASSL_SPARSE	YES	YES	Var-Step	IMP	YES	YES
IDAS	YES	YES	Var-Step	IMP	YES	NO
IDAS_SPARSE	YES	YES	Var-Step	IMP	YES	YES
AM1,AM2,AM4	YES	YES	Fixed-Step	IMP	NO	NO
CVODE_BDF, CVODE_AM	NO	YES	Var-Step	IMP	YES	NO
CVODE_BDF_SPARSE	NO	YES	Var-Step	IMP	YES	YES
RK4	NO	YES	Fixed-Step	EXP	NO	NO
EULER	NO	YES	Fixed-Step	EXP	NO	NO



Problems in DAE integration

DAE solvers generate the algebraic system:

$$F\left(\frac{\mathbf{x} - \text{old}(\mathbf{x})}{h}, \mathbf{x}, \mathbf{u}\right) = 0 \quad \mathbf{x}(t_0) = \mathbf{x}_0$$

and solve the set of implicit equations in \mathbf{x} using Newton-Raphson to bring the residuals F to zero.

$$\left(\frac{J_{\dot{\mathbf{x}}}}{h} + J_{\mathbf{x}}\right) \Delta \mathbf{x}_i = -F(\dot{\mathbf{x}}_i, \mathbf{x}_i, \mathbf{u})$$

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \Delta \mathbf{x}_i$$

$$\dot{\mathbf{x}}_{i+1} = \dot{\mathbf{x}}_i + \frac{\Delta \mathbf{x}_i}{h}$$

\mathbf{x}_i value of \mathbf{x} in the Newton iteration i

$$J_{\mathbf{x}} = \partial F / \partial \mathbf{x}$$

$$\begin{aligned} & (\mathbf{x}_{i+1} - \text{old}(\mathbf{x})) / h - (\mathbf{x}_i - \text{old}(\mathbf{x})) / h \\ & = \Delta \mathbf{x}_i / h \end{aligned}$$

Newton iteration, selection of BDF order, step h , etc. are managed by the integrator (DASS...)



Problems in DAE integration



$$\left(\frac{J_{\dot{x}}}{h} + J_x\right)\Delta x_i = -F(\dot{x}_i, x_i, u)$$

$$x_{i+1} = x_i + \Delta x_i$$

$$\dot{x}_{i+1} = \dot{x}_i + \frac{\Delta x_i}{h}$$

It may be difficult to find consistent initial conditions of x and its derivative

$$F(\dot{x}_0, x_0, u) = 0$$

In stiff systems, when $h \rightarrow 0$, one can expect numerical problems associated to the updating of derivatives and the Newton matrix calculation

$$(J_{\dot{x}} + hJ_x)\Delta x_i = -hF(\dot{x}_i, x_i, u)$$

With $h \rightarrow 0$, The Newton matrix reduces to $J_{\dot{x}}$, which maybe singular if algebraic equations are involved in the calculations. One can expect badly conditioned equations to solve in this case.



DAE, example

$$\sin\left(\frac{d x}{d t} + 3xz\right) - x \frac{d x}{d t} = u \quad x(0) = 3 \quad \text{Implicit DAE}$$

$$xu + x \log(z) = z \quad z(0) = ? \quad \text{DASSL}$$

$$u(t) = 2 \quad z(0)=5$$

Expanding the integration time above 2:

[TIME: 2.5582027368550806] *** KILLPOINT level 2 (code ESI:9:201:32:98)

Cannot converge on the integration step from time 2.558203 to time 2.558203

Reason:

DASSL solver message: The corrector could not converge, probably due to an inaccurate or ill-conditioned Jacobian.



Problems in DAE integration



$$\left. \begin{array}{l} \frac{dx}{dt} = f(x, z, u) \\ g(x, z, u) = 0 \end{array} \quad \left. \begin{array}{l} \frac{x - \text{old}(x)}{h} = f(x, z, u) \\ g(x, z, u) = 0 \end{array} \right\} \Rightarrow \begin{array}{l} x - \text{old}(x) - hf(x, z, u) = 0 \\ g(x, z, u) = 0 \end{array}$$

Solving with
Newton-Raphson

$$\begin{bmatrix} 1 - h \frac{\partial f}{\partial x} & -h \frac{\partial f}{\partial z} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial z} \end{bmatrix} \begin{bmatrix} \Delta x_i \\ \Delta z_i \end{bmatrix} = \begin{bmatrix} -x_i + \text{old}(x) + hf(x_i, z_i, u) \\ -g(x_i, z_i, u) \end{bmatrix}$$

Singular matrix
with $h \rightarrow 0$ if
 $\partial g / \partial z$ is singular

$$\begin{array}{l} \frac{dx}{dt} = f(x, u) \\ g(x, u) = 0 \end{array}$$

Example:
High index
problem



DAE integration



Order reduction to index 0 or 1 is advisable before integration of DAE systems.

For higher order systems:

Consistent initial conditions of x , z and its derivatives can be found solving the system and their derivatives at $t = 0$

$$F(\dot{x}_0, x_0, u) = 0$$

$$\dot{F}(\ddot{x}_0, \dot{x}_0, x_0, u) = 0$$

$$\ddot{F}(\dddot{x}_0, \ddot{x}_0, \dot{x}_0, x_0, u) = 0$$

...

Solutions of x , z and its derivatives over time are found solving the system and their derivatives for h , $t+h$, $t+2h, \dots$



In-line integration



In order to facilitate the integration of the equations, dynamic (x) and algebraic (w) variables are considered separately:

$$F(\dot{x}, x, w, u) = 0 \quad x(t_0) = x_0$$

When discretizing the system, the derivative of x , that is \dot{x} , is not replaced by its BDF approximation, but x is replaced by the corresponding expression as a function of its derivative:

$$F(\dot{x}, h\dot{x} + \text{old}(x), w, u) = 0 \quad \dot{x} \approx \frac{x - \text{old}(x)}{h}$$

The system of equations is solved for w and \dot{x} , the derivative of x , using Newton-Raphson method. Finally, x is updated from:

$$x = \dot{x}h + \text{old}(x)$$



In-line integration



$$F(\dot{x}, h\dot{x} + \text{old}(x), w, u) = 0$$

$$[J_{\dot{x}} + hJ_x, J_w] \begin{bmatrix} \Delta\dot{x}_i \\ \Delta w_i \end{bmatrix} = -F(\dot{x}_i, x_i, w_i, u)$$

$$\dot{x}_{i+1} = \dot{x}_i + \Delta\dot{x}_i$$

$$w_{i+1} = w_i + \Delta w_i$$

$$x_{i+1} = x_i + h\Delta\dot{x}_i$$

There is no problem
with $h \rightarrow 0$, or
presence of algebraic
loops

This algorithm provides a way of finding consistent initial conditions with x_0 for the derivatives and algebraic variables, solving for them the same equations with $h=0$ and $\text{old}(x) = x_0$

$$F(\dot{x}_0, x_0, w_0, u) = 0$$

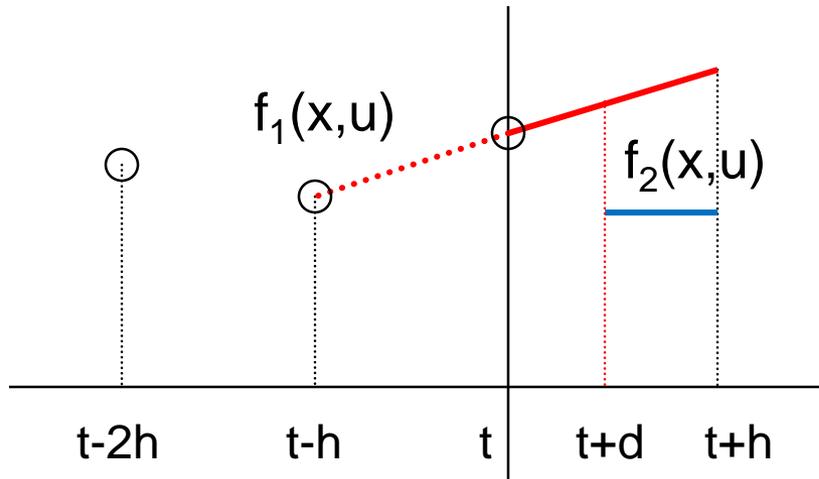


Events and Discontinuities

- ✓ Quite often, when some event takes place, the system's dynamics changes in such a way that the model describing the process requires discontinuous expressions in $f(x,u)$ or its derivative.
- ✓ When such events appear, the $f(x,u)$ of the model changes from $f_1(x,u)$ to $f_2(x,u)$ at a certain time instant.
- ✓ In these cases, direct application of the integration formulas can lead to wrong results



Events and Discontinuities



$$x(t+h) = x(t) + \int_t^{t+h} f(x(\tau), u) d\tau$$

If integration is performed up to $t+h$, this generates an error

$$\frac{dx}{dt} = f_1(x, u) \quad \text{time} < t+d$$

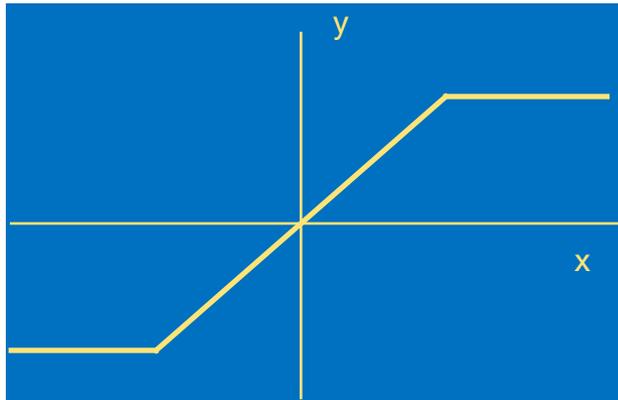
$$\frac{dx}{dt} = f_2(x, u) \quad \text{time} \geq t+d$$

Event at
time $t+d$

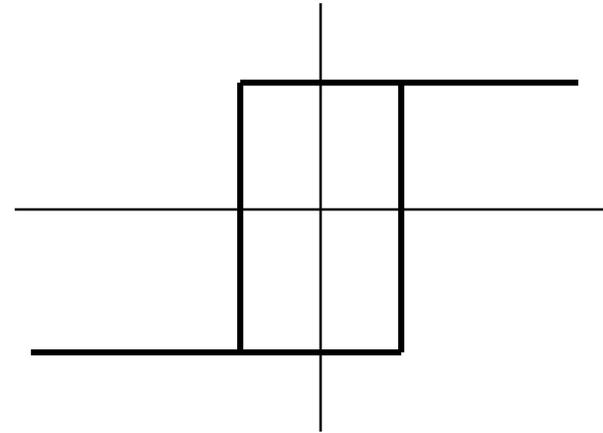


Events and Discontinuities

Discontinuities appear in many processes



Bounds



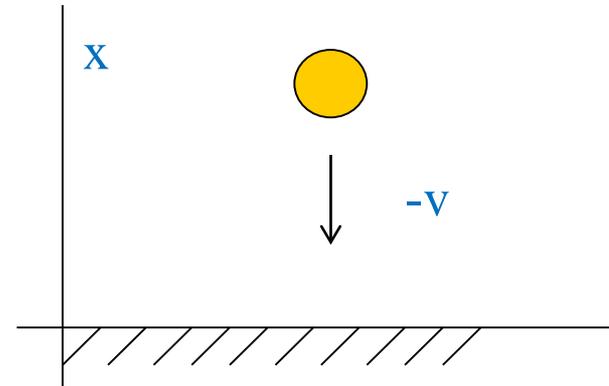
Hysteresis



Events and Discontinuities

When an event takes place, new dynamics can appear

$$\frac{dv}{dt} = \begin{cases} -g & \text{si } x > 0 \\ 0 & \text{si } x \leq 0 \end{cases}$$
$$\frac{dx}{dt} = v$$
$$v = -kv \quad \text{when } x = 0$$

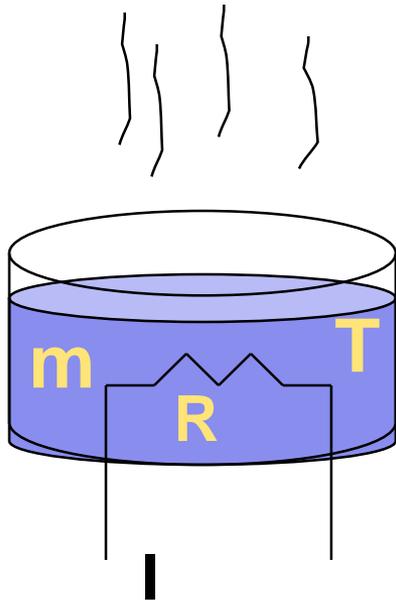


Crash / Bounce



Events and Discontinuities

Heating and boiling at constant pressure



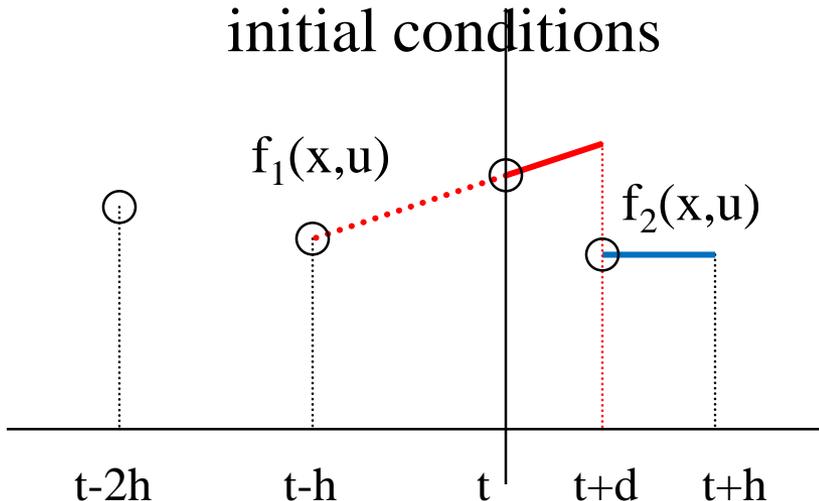
$$\frac{dT}{dt} = \begin{cases} I^2 R / (m c_e) & \text{si } T < T_e \\ 0 & \text{si } T \geq T_e \end{cases}$$
$$\frac{dm}{dt} = \begin{cases} 0 & \text{si } T < T_e \\ -I^2 R / \lambda & \text{si } T \geq T_e \end{cases}$$



Discontinuities

In order to integrate correctly a model with discontinuities, one must:

- Locate precisely the time instant when the discontinuity takes place
- Re-start the integration with the new set of equations and initial conditions



$$x(t+d) = x(t) + \int_t^{t+d} f_1(x(\tau), u) d\tau$$
$$x(t+h) = x(t+d) + \int_{t+d}^{t+h} f_2(x(\tau), u) d\tau$$

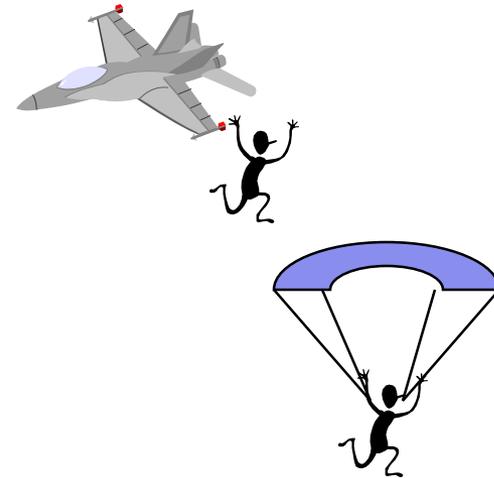


Time events

The model changes at a predefined time instant.

Example: The parachute opens after 3sec. from jumping.

$$\frac{dv}{dt} = \begin{cases} -g & \text{si } t < 3 \\ -g + F(v) & \text{si } t \geq 3 \end{cases}$$

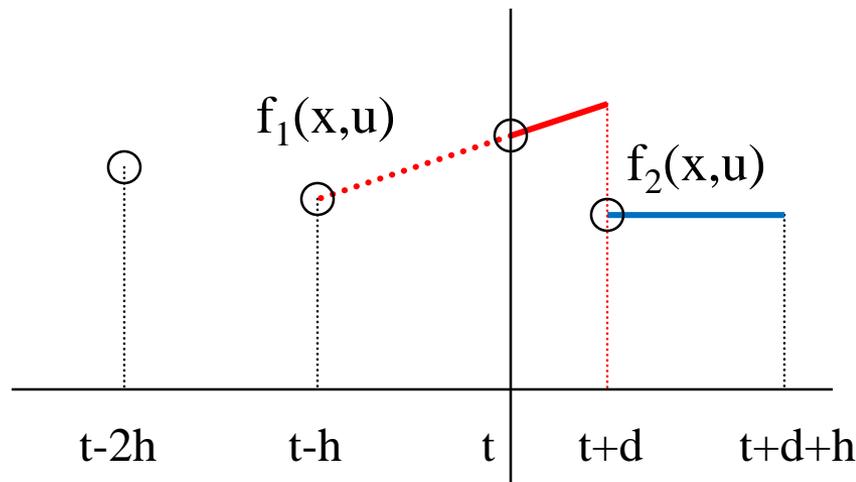


The time instant when the discontinuity takes place is known precisely



Time events

As the time instant $t+d$ is known, in order to integrate properly the model, one can adjust the step h so that it coincides with the event, compute $x(t+d)$, and re-start the integration at $t+d$

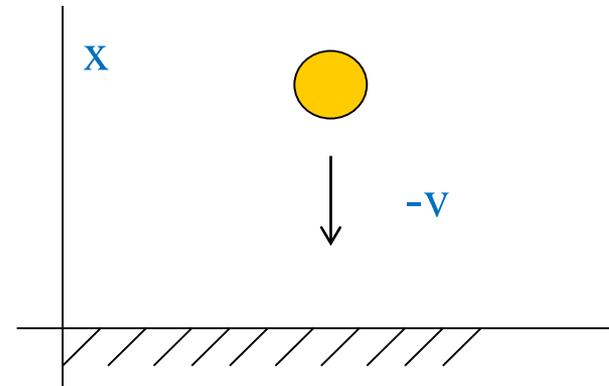




State events

- ✓ The model change takes place when a model variable crosses a certain threshold.
- ✓ The time instant of the discontinuity is not known a priori and must be estimated.

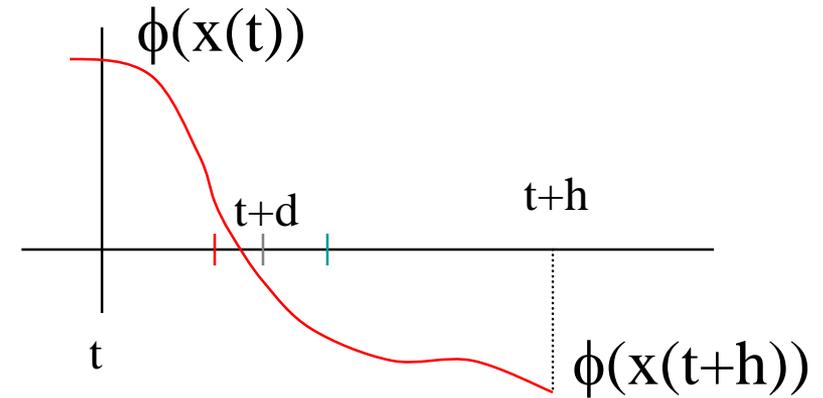
$$\frac{dv}{dt} = \begin{cases} -g & \text{si } x > 0 \\ 0 & \text{si } x \leq 0 \end{cases}$$
$$\frac{dx}{dt} = v$$
$$v = -kv \quad \text{when } x = 0$$





State events

$$\begin{cases} \frac{dx}{dt} = f_1(x, u) \text{ si } \phi(x) > 0 \\ \frac{dx}{dt} = f_2(x, u) \text{ si } \phi(x) \leq 0 \end{cases}$$



The change in the transition function $\phi(x)$ is detected at $t+h$

In order to locate the time instant $t+d$, the step size h must be reduced and several iterations must be performed until $t+d$ is found with enough precision.

Then, the integration is re-started at $t+d$ with the new model



Discontinuities in EcosimPro

There are sentences in the simulation language that manage automatically the location of the time instants when the discontinuities takes place, as well as the changes in the model equations and the re-initialization of the integration.

Discrete events: DISCRETE region, WHEN, WHILE, AFTER

Discontinuities in continuous models: ZONE



Discontinuities in EcosimPro

Discrete events

WHEN (logic condition) THEN

sentences

WHILE (logic condition)

sentences

END WHEN

END WHILE

sentence AFTER time

Changes in a continuous model

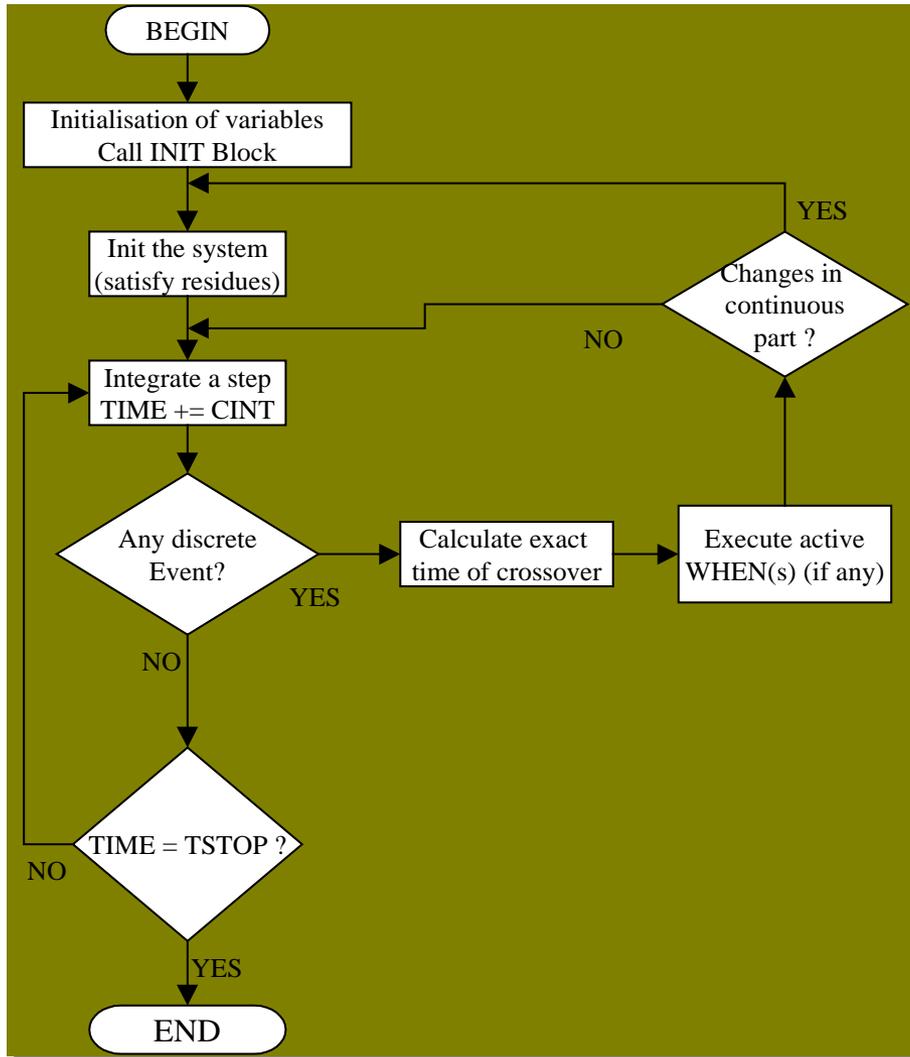
x = ZONE (condition 1) equation 1

ZONE (condition 2) equation 2

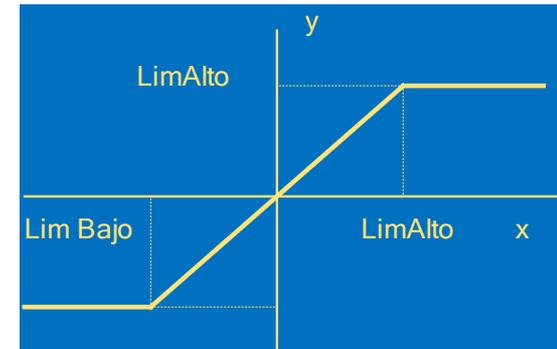
OTHERS equation 3



Events and discontinuities



Typical execution path



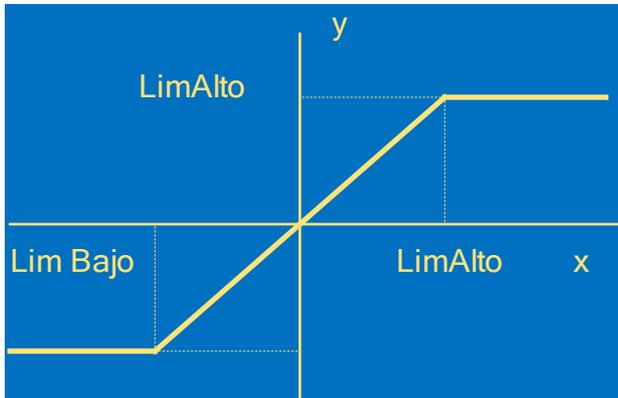
$y = \text{ZONE } (x > \text{HiLim }) \text{ Hilim}$
 $\text{ZONE } (x < \text{LowLim }) \text{ LowLim}$
 $\text{OTHERS } x$



Treatment of discontinuities

Some simulation languages do not incorporate an explicit treatment of discontinuities, and only consider conditional expressions, which can lead to integration errors.

Example: A bounded signal



MACRO Limitador (y, x, LimAlto, LimBajo)

```
if (x .GT. LimAlto) then y = LimAlto
else if (x .LT. LimBajo) then y = LimBajo
else y = x
endif
```

macro end



Example

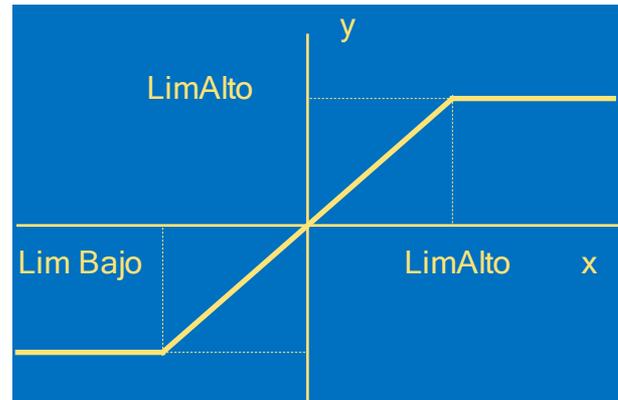
With correct discontinuity integration

$$\frac{dx}{dt} = \tau y(x)$$

Without correct discontinuity integration

TIME	x	y
0	0.1	0.1
0.3	0.18214	0.18214
0.6	0.3317498	0.3317498
0.9	0.60424908	0.60424908
1.15197374	1.00000024	1.00000024
1.15197374	1.00000024	1
1.2	1.09605276	1
1.5	1.69605276	1
1.8	2.29605276	1
2	2.69605276	1

-1



TIME	x	y
0	0.1	0.1
0.3	0.18214	0.18214
0.6	0.3317498	0.3317498
0.9	0.60424908	0.60424908
1.2	1.08975999	1
1.5	1.68975999	1
1.8	2.28975999	1
2	2.68975999	1

$$x' = 2*y$$

$y = \text{ZONE } (x > x_{\max}) \ x_{\max}$
 $\text{ZONE } (x < x_{\min}) \ x_{\min}$
 $\text{OTHERS } \ x$

$$x' = 2*y$$

$y = \text{IF } (x \geq x_{\max}) \ x_{\max}$
 $\text{ELSEIF } (x \leq x_{\min}) \ x_{\min}$
 $\text{ELSE } \ x$

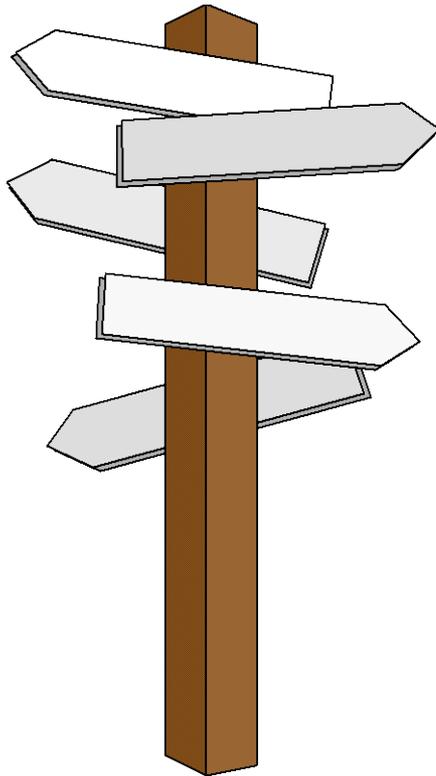


Variable structure models

- ✓ Models in which the equations change over time when some events takes place.
- ✓ The number of state variables is kept over the integration horizon



End



Thank you for your
attention