Università degli Studi di Trento

Department of Industrial Engineering

Mechatronic Systems Simulation

Prof.: Bertolazzi Enrico, Biral Francesco

Course Notes

Matteo Dalle Vedove
matteo.dallevedove@studenti.unitn.it

Academic Year 2021-2022
June 4, 2022

Contents

I Differential Algebraic Equations **1**

1 Ordinary Differential Equations and Numerical Solutions **2**
1.1 Existence of the solution . . . 4
1.2 Taylor expansion . . . 5
1.2.1 Multi-variable functions . . . 7
1.3 Numerical methods . . . 8
1.3.1 Taylor series based . . . 8
1.3.2 Runge-Kutta . . . 12

2 Introduction to Algebraic Differential Equations **16**
2.1 Linear differential algebraic equations . . . 18
2.1.1 Usage of the Kronecker normal form . . . 20
2.1.2 LU decomposition and Jacobi modification . . . 21
2.2 DAE index and index reduction . . . 22
2.2.1 Introduction of dummy variables . . . 28
2.2.2 Kernel computation and index reduction . . . 30
2.3 Semi-explicit form . . . 32
2.3.1 Implicit function theorem . . . 32

II Modelling & Simulation **35**

Part I

Differential Algebraic Equations

Chapter 1

Ordinary Differential Equations and Numerical Solutions

To start the description of the **differential algebraic equations** **DAEs** it's firstly necessary to recall what **ordinary differential equations** **ODEs** are, what they mean and how to solve them.

In particular ordinary differential equations is a particular equation that involves a function (for example $y(x)$ depending from the independent variable x) and it's derivative as shown in this example:

$$y''(x) + xy'(x) + y(x) = \sin x \qquad \leftrightarrow \qquad y'' + xy' + y = \sin x$$

Ordinary differential equations (or system of ODEs) can be written in a ***standard form*** made by the **differential part**, where the first derivative is a function of itself, and the **initial condition** that set a specified value of the solution of the problem:

$$\begin{cases} y' = f(x, y(x)) = f(x, y) & : \text{differential part} \\ y(a) = y_a & : \text{initial condition} \end{cases} \tag{1.1}$$

Initial conditions are mandatory: the solution of the differential part lonely gives a *family* of solutions (parametric results) whose specific value can be determined only by knowing the value of the function at certain time. Considering the simple case of the differential $x' = 0$ with independent variable t, then the general solution is the class of all the constant functions $x(t) = c$ (with $c \in \mathbb{R}$). If a boundary condition is set (example $x(1) = 3$) then we can chose the particular solution (in this case $c = 3$ and so $x(t) = 3$).

Ordinary differential equations can also come in system as in the following example (with t as dependent variable) composed by 2 differential and 2 initial condition terms:

$$\begin{cases} x' = x + y \\ y' = e^x - y \\ x(0) = 0 \\ y(0) = 1 \end{cases}$$

Vectorial notation Considering the system of $n = 3$ differential equation depending from the independent variable t in the form

$$\begin{cases} z'(t) = x(t) + z(t) \\ w'(t) = z(t) \\ x'(t) = w(t)z(t) + t(t) \\ x(0) = w(0) = z(0) = 1 \end{cases}$$

then it can be rewritten in a vectorial form; considering in fact the substitutions $x(t) = y_1(t)$, $w(t) = y_2(t)$ and $z(t) = y_3(t)$ we have obtain the system

$$\begin{cases} y_3' = y_1 + y_3 \\ y_2' = y_3 \\ y_1' = y_2 y_3 + t \\ y_1(0) = y_2(0) = y_3(0) = 1 \end{cases} \Rightarrow \begin{cases} y_1' = y_2 y_3 + t \\ y_2' = y_3 \\ y_3' = y_1 + y_3 \\ y_1(0) = y_2(0) = y_3(0) = 1 \end{cases}$$

Considering the vector $\boldsymbol{y} = (y_1, y_2, y_3)$ we can so rewrite the original system of ODEs as

$$\begin{cases} \boldsymbol{y}' = \boldsymbol{f}(t, \boldsymbol{y}) \\ \boldsymbol{y}(0) = \mathbf{1} \end{cases} \tag{1.2}$$

where

$$\boldsymbol{f} : \mathbb{R}^{n+1} \to \mathbb{R}^n = \begin{pmatrix} y_2 y_3 + t \\ y_3 \\ y_1 + y_3 \end{pmatrix}$$

In this case the domain of the function $\boldsymbol{f}$ is a vector of dimension $n + 1$: n related to the number of ODEs and 1 for the time dependency.

Order of an ODE The **order** of an ordinary differential equation is equal to the maximum order derivative appearing in the differential part of the system; considering as example

$$\begin{cases} y'' + y' + z' = 0 \\ z' + t = 0 \end{cases}$$

the order of such ordinary differential system is equal to 2 (associated to the term y'').

In general numerical methods are defined for 1^{st} order ODEs and so it's necessary (but most importantly possible) to convert any generic differential equation into a system of 1^{st} order ODEs by performing a change of variable.

Example 1.1: reduction to a system of ODEs of first order

Given the system of ODEs of the 3^{rd} order in the independent variable t defined as

$$\begin{cases} x''' + y' = x^2 + t \\ y'' + x = t^2 + 1 \\ x(0) = 0 \qquad y(0) = 0 \\ x'(0) = 0 \qquad y'(0) = 2 \\ x''(0) = 2 \end{cases}$$

the reduction to a system of first order ODEs is made by introducing the variable $z = x'$; defining instead the function $x'' = z' = w$ we also have that $z' = w$ hence $x''' = z'' = w'$; finally we can set

$y' = p$ hence $y'' = p$. The system so becomes

$$\begin{cases} w' + p = x^2 + t \\ p' + x = t^2 + 1 \\ x' = z \\ z' = w \\ y' = p \\ x(0) = 0 \qquad y(0) = 0 \\ z(0) = 0 \qquad p(0) = 2 \\ w(0) = 2 \end{cases}$$

This system present 3 more differential terms and so seems *more difficult*, however this formulation is numerically more suitable for the computation.

1.1 Existence of the solution

Given the general system of n ordinary differential equations in the standard form

$$\begin{cases} \boldsymbol{y}' = \boldsymbol{f}(t, \boldsymbol{y}) \\ \boldsymbol{y}(a) = \boldsymbol{y}_a \end{cases}$$

using **Peano's theorem** we can state that if the vectorial map $\boldsymbol{f} : \mathbb{R}^{n+1} \to \mathbb{R}^n$ is continuous, then a solution exists in the neighbourhood of the point $(t = a, \boldsymbol{y} = \boldsymbol{y}_a)$. This theorem provide a sufficient condition to determine if a solution exists, but it doesn't state that the solution is unique.

Example 1.2: ODE with multiple solution

Considering the ordinary differential equation in the independent variable t

$$\begin{cases} y' = \sqrt{|y|} \\ y(0) = 0 \end{cases}$$

we can see that the map $f(y) = \sqrt{|y|}$ is continuous for all $y \in \mathbb{R}$, hence for Peano's theorem a solution must exists for t *sufficiently close* to 0. In particular we can observe that the function

$$y(t) = 0$$

is a solution of the system, in fact it matches the initial condition and we have that it's derivative $y' = \frac{dy}{dt} = 0$ is equal to the function $f(y) = \sqrt{|0|} = 0$.
However this is not the lonely solution, considering in fact the function

$$y(t) = \frac{t^2}{4}\text{sign}(t) = \begin{cases} \frac{t^2}{4} & t \geq 0 \\ -\frac{t^2}{4} & t < 0 \end{cases}$$

Observing that the derivative of such function can be regarded as

$$y'(t) = \begin{cases} \frac{t}{4} & t \geq 0 \\ -\frac{t}{2} & t < 0 \end{cases} \quad \Rightarrow \quad y'(t) = \frac{|t|}{2}$$

we can also check that the provided solution solves the differential part of the system, in fact

$$\sqrt{|y(t)|} = \sqrt{\left|\frac{t^2}{4}\text{sign}(t)\right|} = \sqrt{\frac{t^2}{4}} = \frac{|t|}{2} = y'(t)$$

In general when solving system of differential equation we want to be sure that the solution exists (using as example Peano's theorem) but that is also unique. In order to do so we have to defined the **Lipschitz continuity**:

$$f : \mathbb{R}^n \to \mathbb{R} \text{ is Lip. cont.} \quad \text{if } \exists L \in \mathbb{R} \quad \text{such that} \quad \big\|f(\boldsymbol{x}) - f(\boldsymbol{y})\big\| \leq L\big\|\boldsymbol{x} - \boldsymbol{y}\big\| \qquad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R} \tag{1.3}$$

We can so state that a system of ordinary differential equation in the standard form has one solution if the map $\boldsymbol{f} : \mathbb{R}^{n+1} \to \mathbb{R}^n$ is continuous (from Peano) and is also lipschitzian.

Example 1.3: Lipschitz continuity

Considering the problem of example 1.2 we can prove that the system has multiple solution by checking that's not Lipschitz continuous. Known that the map $f(t, y) := \sqrt{|y|}$ is continuous we can apply Lipschitz definition in the particular case when one point is the origin of the the axis:

$$\begin{aligned} \big|f(0,y) - f(0,0)\big| &\leq L\big|x - 0\big| \\ \Big|\sqrt{|y|} - \sqrt{|0|}\Big| L &\leq |y| \\ \cancel{\sqrt{|y|}} &\leq L \cancel{\sqrt{|y|}} \sqrt{|y|} \\ L &\geq \frac{1}{\sqrt{|y|}} \end{aligned}$$

Observing that for $y \to 0$ the denominator converges to zero this means that L must diverge to ∞ meaning that the function is not lipschitzian: the solution exists (Peano's theorem) but it might not unique (for Lipschitz).

1.2 Taylor expansion

The main tool used for numerical approximate solution of (system of) ordinary differential equation is the **Taylor series expansion** that's used to approximate a function $f : \mathbb{R} \to \mathbb{R}$ of class $\mathcal{C}^\infty$ as a polynomial in the neighbourhood of a specific point x_0; given h as the deviation from the point x_0 for *small* values of h we can approximate the function

$$\begin{aligned} f(x_0 + h) &\approx a_0 + a_1 h + a_2 h^2 + a_3 h^3 + \cdots + a_n h^n \\ &\approx a_0 + \sum_{k=1}^{\infty} a_k h^k \end{aligned} \tag{1.4}$$

Evaluating both members allows to obtain the first coefficient $a_0 = f(x_0)$ of the Taylor expansion, in fact

$$f(x_0) = a_0 + \sum_{k=1}^{\infty} a_k 0^k = a_0$$

By differentiation we can also obtain other coefficients

$$\begin{aligned} f'(x_0 + h) &= a_1 + \sum_{k=2}^{\infty} a_k h^{k-1} k & \xrightarrow{h=0} \quad a_1 &= f'(x_0) \\ f''(x_0 + h) &= 2a_2 + \sum_{k=3}^{\infty} a_k h^{k-2} k(k-1) & \xrightarrow{h=0} \quad 2a_2 &= f''(x_0) \end{aligned}$$

Considering that in general each coefficient can be computed as $a_k = f^{(k)}(x_0)/k!$ we can better rewrite the Taylor series expansion of equation 1.4 as

$$f(x_0 + h) \approx \sum_{k=0}^{\infty} \frac{f^{(k)}(x_0)}{k!} h^k \tag{1.5}$$

Truncation From a numerical standpoint the computation of the Taylor series is truncated to a order n and so we can use the formulation

$$f(x_0 + h) = \sum_{k=0}^{n} \frac{f^{(k)}(x_0)}{k!} h^k + R_n(h) \tag{1.6}$$

where R_n is the reminder due to the truncation of the series that can be evaluated in multiple ways:

- using Peano's formulation the more formal definition of the reminder is

  $$R_n(h) = \int_{x_0}^{x_0+h} f^{(n+1)}(s) \frac{(s-h)^n}{n!}\, ds$$

  This formulation is still complex and numerically *unusable*;

- the Lagrange reminder in the form

  $$R_n(h) = f^{(n+1)}(\zeta) \frac{h^{n+1}}{(n+1)!} \qquad \text{with } \zeta \in (x_0, x_0 + h)$$

- the *big O* notation $R_n(h) = \mathcal{O}(h^{n+1})$; in particular we denote $g(x) = \mathcal{O}(f(x))$ if

  $$\exists C \in \mathbb{R} \quad \text{such that} \quad |g(x)| \leq C|f(x)|$$

- the *small o* notation $R_n(h) = o(h^n)$; we say that $g(x) = o(f(x))$ if f is lipschitzian (equation 1.3) and we have that

  $$\lim_{x \to 0} \frac{o(f(x))}{g(x)} = 0$$

Common Taylor expansions Examples of notable series expansion on the point $x_0 = 0$ for well-known functions are the exponential, the cosine and sine:

$$\begin{aligned} e^h &= \sum_{k=0}^{\infty} \frac{h^k}{k!} = 1 + h + \frac{h^2}{2} + \frac{h^3}{3!} + \frac{h^4}{4!} + \dots \\ \cos h &= \sum_{k=0}^{\infty} -1^k \frac{h^{2k}}{2k!} = 1 - \frac{h^2}{2!} + \frac{h^4}{4!} - \frac{h^6}{6!} + \dots \\ \sin h &= \sum_{k=0}^{\infty} -1^k \frac{h^{2k+1}}{(2k+1)!} = h - \frac{h^3}{3!} + \frac{h^5}{5!} - \frac{h^7}{7!} + \dots \end{aligned} \tag{1.7}$$

Existence of the expansion Considering the definition of the lagrangian reminder in the form $f^{(m)}(\zeta)\frac{h^m}{m!}$ if we truncate the series to higher order (bigger value of m) we observe that for $h < 1$ the term $h^m/m!$ tends to zero, and so if we ensure that the m-th derivative doesn't diverge we have that the Taylor series converge to the *real* function. However this sometimes can fail: considering as example the function

$$f(x) = \begin{cases} e^{-\frac{1}{x^2}} & x > 0 \\ 0 & \text{otherwise} \end{cases}$$

it's proven that the function is *smooth* ($f \in C^\infty$) and that the k-th derivative is in the form

$$\left(e^{-\frac{1}{x^2}}\right)^{(k)} = e^{-\frac{1}{x^2}} \frac{p_1(x)}{p_2(x)} \qquad p_1, p_2 \text{ polynomials}$$

and converges to zero for $x \to 0$. By applying the definition of the Taylor expansion in $x_0 = 0$ we so have that

$$f(0+h) = \sum_{k=0}^{\infty} f^{(k)}(0)\frac{h^k}{k!} = 0$$

This expansion correctly models the left-hand side of the function f but not the right side, and so the Taylor expansion fails.

1.2.1 Multi-variable functions

Until now we have defined the Taylor expansion of function with one variable in the form $f(x)$, but such concept should be extended to function of multiple variables. Considering the simple case of $f : \mathbb{R}^2 \to \mathbb{R}$ in order to perform the Taylor series we have to ensure a continuity up to an order m for the function, meaning

$$f(x,y) \in C^m \qquad \Leftrightarrow \qquad \frac{\partial^{i+j}}{\partial x^i \partial y^j} f(x,y) \in C \quad \forall i+j \leq m \tag{1.8}$$

We have the formal definition of the Taylor series for functions of one variable (equation 1.5) and so, as idea, we can *slice* the function f passing through a point (x_0, y_0) with a direction $(x - x_0, y - y_0) = (d_x, d_y)$ using a function

$$g(t) = f(x_0 + td_x, y_0 + td_y)$$

The idea is so to compute the Taylor series of this function in the neighbourhood of $t = 0$:

$$g(t) = g(0) + g'(t)t + g''(0)\frac{t^2}{2!} + g'''(0)\frac{t^3}{3!} + \dots$$

The term $g'(t)$relates to the total derivative of $f(x_0 + td_x, y_0 + td_y)$ respect to the variable t, meaning that

$$\begin{aligned}
g'(t) &= \frac{d}{dt} f\big(x_0 + t\,d_x, y_0 + t\,d_y\big) \\
&= \frac{\partial f(\dots)}{\partial x}\frac{d(x_0 + t\,d_x)}{dt} + \frac{\partial f(\dots)}{\partial y}\frac{d(y_0 + t\,d_y)}{dt} \\
&= \frac{\partial f(\dots)}{\partial x} d_x + \frac{\partial f(\dots)}{\partial y} d_y = \frac{\partial f(\dots)}{\partial x}(x - x_0) + \frac{\partial f(\dots)}{\partial y}(y - y_0) \\
g'(0) &= \frac{\partial f(x_0, y_0)}{\partial x}(x - x_0) + \frac{\partial f(x_0, y_0)}{\partial y}(y - y_0)
\end{aligned}$$

Using a similar methodology it's possible to compute the second order derivative of g as

$$\begin{aligned}
g''(t) &= \frac{d}{dt} g'(t) = \frac{d}{dt}\left(\frac{\partial f(\dots)}{\partial x}(x - x_0) + \frac{\partial f(\dots)}{\partial y}(y - y_0)\right) \\
&= \frac{\partial^2 f(\dots)}{\partial x^2}(x - x_0)^2 + \frac{\partial^2 f(\dots)}{\partial y^2}(y - y_0)^2 + 2\frac{\partial^2 f(\dots)}{\partial x\,\partial y}(x - x_0)(y - y_0)
\end{aligned}$$

Considering as more general statement to express the Taylor series respect to a point (x_0, y_0) moving with values $h = td_x$ and $k = td_y$ the series truncated to the second order is so

$$\begin{aligned}
f(x_0 + h, y_0 + k) = & f(x_0, y_0) + \left.\frac{\partial f}{\partial x}\right|_{(x_0,y_0)} h + \left.\frac{\partial f}{\partial y}\right|_{(x_0,y_0)} k \\
& + \frac{1}{2}\left.\frac{\partial^2 f}{\partial x^2}\right|_{(x_0,y_0)} h^2 + \frac{1}{2}\left.\frac{\partial^2 f}{\partial y^2}\right|_{(x_0,y_0)} k^2 + \left.\frac{\partial^2 f}{\partial x\,\partial y}\right|_{(x_0,y_0)} hk
\end{aligned} \tag{1.9}$$

This representation can be compacted using a vectorial/matrix notation condensing (x_0, y_0) in the vector $\boldsymbol{x}_0$ and the increment $(h, k) = \boldsymbol{h}$ we can define

$$f(\boldsymbol{x}_0 + \boldsymbol{h}) = f(\boldsymbol{x}_0) + \nabla f(\boldsymbol{x}_0)\,\boldsymbol{h} + \frac{1}{2}\boldsymbol{h}^t\,\nabla^2 f(\boldsymbol{x}_0)\,\boldsymbol{h} + R_3(\|\boldsymbol{h}\|) \tag{1.10}$$

where $\nabla f(\boldsymbol{x}) = (\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n})$ is the **gradient** of the function (and is a row vector) and $\nabla^2 f(\boldsymbol{x})$ is the **hessian matrix** of f. Note that this formulation is general and is valid for any multi-variable function $f : \mathbb{R}^n \to \mathbb{R}$ assuming that's at least C^2.

Higher order Taylor expansion In order to perform Taylor expansion with order greater than 2 it's necessary to use a **tensor** notation; in particular the expansion of a function $f : \mathbb{R}^n \to \mathbb{R}$ up to the order m is described by the equation

$$f(\boldsymbol{x}_0 + \boldsymbol{h}) = \sum_{k=0}^{m} \sum_{|\alpha|=k} \partial_\alpha f(\boldsymbol{x}_0) \frac{\boldsymbol{h}^\alpha}{\alpha!} + R_m \tag{1.11}$$

where $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ is the multi-index vector that consists of non-negative integers; the condition $|\alpha| = k$ based on the norm $|\alpha| = \alpha_1 + \alpha_2 + \cdots + \alpha_n$ choses all combination of α satisfying such relation and the multi-index variable is used for the computation following the expressions

$$\partial_\alpha := \partial_{x_1}^{\alpha_1} \partial_{x_2}^{\alpha_2} \ldots \partial_{x_n}^{\alpha_n} f(x) \qquad\qquad \boldsymbol{h}^\alpha := h_1^{\alpha_1} h_2^{\alpha_2} \ldots h_n^{\alpha_n}$$

1.3 Numerical methods

1.3.1 Taylor series based

In practise (systems of) ordinary differential equations are numerically solved by computes using algorithms some of which are based on the Taylor series expansion. Considering for the simplicity a first order ordinary differential equation in the standard form (equation 1.1)

$$\begin{cases} y' = f(x, y) \\ y(a) = y_a \end{cases}$$

If we consider $y(x)$ the solution of the ODE system, that can be expanded up to the second order with Taylor as

$$y(x + h) = y(x) + y'(x)h + \mathcal{O}(h^2) = y(x) + f(x, y)h + \mathcal{O}(h^2)$$

The most basic idea to numerically compute the solution is to subdivide the interval $[a, b]$ of integration into N pieces having each a width $h = \frac{b-a}{N}$; this discretization of the x axis determines so the sequence of point $x_k = a + hk$. With this idea in mind we can see that the yet computed Taylor expansion can be regarded as

$$y(x_{k+1}) = y(x_k) + f\big(x_k, y(x_k)\big)h + \mathcal{O}(h^2) \tag{1.12}$$

Numerical methods determines a sequence of output y_k that tends to approximate the real behaviour of the solution, hence $y_k \approx y(x_k)$. The simplest numerical method to solve the ordinary differential equation is by simply using equation 1.12 neglecting the reminder:

$$y_{k+1} = y_k + f(x_k, y_k)h \tag{1.13}$$

Error Numerical methods are approximation of the analytical solutions, hence intrinsically contains an error that should be somehow defined in order to determine *how bad* or *good* the numerical solution is. If we new define the error ϵ_k on the k-th step as the difference between the analytical and numerical solution

$$\varepsilon_{k+1} = y(x_{k+1}) - y_{k+1} = y(x_k) - y_k + \Big(f\big(x_k, y(x_k)\big) - f(x_k, y_k)\Big)h + \frac{y''(\zeta_k)}{2}h^2 \tag{1.14}$$

where the reminder $\mathcal{O}(h^2)$ as been substituted with the lagrangian one and hence $\zeta_k \in (x_k, x_{k+1})$. Considering that the function f is assumed to be lipschitzian, then it means that exists $L \in \mathbb{R}$ such that

$$\left|f\big(x_k, y(x_k)\big) - f(x_k, y_k)\right| \leq L\,|y(x_k) - y_k|$$

Considering this inequality, knowing that $y(x_k) - y_k = \epsilon_k$ and using the triangular inequality we can rewrite equation 1.14 as

$$|\varepsilon_{k+1}| = |\varepsilon_k| + hL\big|y(x_k) - y_k\big| + \frac{h^2}{2}|y''(\zeta_k)| = A|\varepsilon_k| + B \tag{1.15}$$

where $A = 1 + hL$ and $B = \frac{h^2}{2}M_2$. In particular M_2 is the constant that bounds the second derivative of y in the domain of integration, meaning

$$M_2 = \sup_{x\in[a,b]}\{y''(x)\}$$

Starting with the theoretical assumption that $\varepsilon_0 = 0$ (the initial error is null given the initial condition) and observing that $A \to 1$ and $B \to 0$, then we have that $|\varepsilon_1| \leq A0 + B = B$; the sequent error is so $|\varepsilon_2| \leq A|\varepsilon_1| + B = B(1 + A)$. Computing $|\varepsilon_3| \leq B(1 + A + A^2)$ it's possible to prove by induction that the error has a formulation

$$|\varepsilon_k| \leq \big(1 + A + A^2 + \cdots + A^{k-1}\big)B$$

The maximum error E_h of this numerical method is so determined by considering the maximum error respect to all discretization steps:

$$E_h = \max_{k=0,\ldots,N}|\varepsilon_k| \leq \max_{k=0,\ldots,N}\big(1 + A + A^2 + \cdots + A^{k-1}\big)B = \big(1 + A + A^2 + \cdots + A^{N-1}\big)B$$

Considering the geometrical series determined by $A + A^2 + \ldots$ we obtain that such sequence sums to the value $\frac{1-A^N}{1-A}$ and so the error can be considered as

$$E_h \leq \frac{A^N - 1}{A - 1}B = \frac{A^N - 1}{1 + hL - 1}\frac{h^2}{2}M_2 = \frac{A^h - 1}{L}\frac{h}{2}M_2$$

All we need now is to quantity the error related to the term A^N; considering that the Taylor series of the exponential sequence $e^x = 1 + x + \frac{x^2}{2!} + \ldots$ we have that such quantity is always greater than $1 + x$, and so knowing that $A = 1 + hL$ we have that $(1 + hL)^N \leq \left(e^{hL}\right)^n = e^{LNh}$. Observing that $Nh = b - a$ we can finally state the total error as

$$E_h \leq \frac{e^{L(b-a)}M_2}{2L}h = Ch \tag{1.16}$$

where $C \in \mathbb{R}$ is a constant, meaning that for $h \to 0$ the error presents the expected behaviour of approaching zero. By a computation point of view this method isn't that good, because in order to halve the error we have to halve also the integration step n (doubling the number of intervals N). If we would have considered other methods truncated to higher orders of derivation what we would have obtained is an error in the form

$$E_h = Ch^p$$

hence by dividing by 2 h the error would have been reduces by 2^p.

System of ODEs Considering the more general case of a system of ordinary differential equation in the standard form using the vectorial notation

$$\begin{cases}\boldsymbol{y}' = \boldsymbol{f}(t, \boldsymbol{y})\\ \boldsymbol{y}(a) = \boldsymbol{y}_a\end{cases}$$

the yet described method can be still used by expanding each component of $\boldsymbol{y}$, meaning that the numerical solution can be approximated as

$$\boldsymbol{y}_{k+1} = \boldsymbol{y}_k + h\boldsymbol{f}(x_k, \boldsymbol{y}_k)$$

and the error can be regarded as $E_h = \max\|\boldsymbol{y}(x_k) - \boldsymbol{y}_k\| \leq Ch$.

Implicit method: back-backward Euler

The numerical method described until now is the **explicit Euler integration** for determining the solution of ordinary differential equations; the formulation as provided in equation 1.13 (page 8) is computationally lightweight (because by knowing x_k, y_k at the current stage allows to automatically compute y_{k+1}), however is unstable and can quickly diverges from the analytical solution.

A way to solve such problematic is by using **implicit method** that are constructed by performing the Taylor expansion *from the left*. Considering as example the **Euler back-backward** method, the computed Taylor series is

$$y(x-h) = y(x) - hy'(x) + \frac{h^2}{2}y''(\zeta) = y(x) - hf(x,y) + \frac{h^2}{2}y''(\zeta)$$

This gives origin to the iterative numerical method defined as

$$y_{k-1} = y_k - hf(x_k, y_k) \tag{1.17}$$

where the solution of the current output y_k is implicitly defined as function of the current *position* x_k and the previous value y_{k-1}. This formulation increases the computational complexity (at each iteration a non-linear system has to be solved in order to determine the implicit solution y_k) but strongly increases the robustness of the algorithm.

Other methods based on the Taylor series

In general given the ordinary differential equation

$$\begin{cases} y' = f(x,y) \\ y(a) = y_a \end{cases}$$

in order to have a solution we assume that f is continuous and lipschitzian; given $y(x)$ the exact solution of the problem, we can apply the Taylor expansion on such result obtaining

$$y(x+h) = y(x) + hy'(x) + \frac{h^2}{2}y''(x) + \cdots + \frac{h^p}{p!}y^{(p)}(x) + \mathcal{O}(h^{p+1})$$

Knowing that $y(x)$ is the solution of the ODE, then we have that $y'(x) = f\big(x, y(x)\big)$; considering now so it's derivative we have that

$$\begin{aligned} y''(x) &= \frac{d}{dx}y'(x) = \frac{d}{dx}f\big(x,y(x)\big) = \frac{\partial f}{\partial x}\big(x,y(x)\big) + \frac{\partial f}{\partial y}\big(x,y(x)\big)y'(x) \\ &= \frac{\partial f}{\partial x}\big(x,y(x)\big) + \frac{\partial f}{\partial y}\big(x,y(x)\big)f\big(x,y(x)\big) \end{aligned}$$

This allows to rewrite the Taylor expansion as

$$y(x+h) = y(x) + hy'(x) + \frac{h^2}{2}\left(\frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}f\right) + \cdots + \frac{h^p}{p!}y^{(p)}(x) + \mathcal{O}(h^{p+1})$$

The previously described explicit Euler method was determined by neglecting the terms with order higher then h, but in this case we have the possibility to express also y'' as function of f and x, y increasing hence the numerical accuracy. The numerical method is so

$$y_{k+1} = y_k + hf(x_k, y_k) + \frac{h}{2}\left(\frac{\partial f(x_k,y_k)}{\partial x} + \frac{\partial f(x_k,y_k)}{\partial y}f(x_k,y_k)\right) \tag{1.18}$$

where so in this case we dropped an error in the form $\mathcal{O}(h^3)$. Increasing the order of the numerical method increases the solution but requires the symbolical computation of the derivatives $\frac{\partial f}{\partial x}$, $\frac{\partial f}{\partial y}$; we can

carry on the process to explicitly determine all the derivative $y^{(p)}$ (up to a certain order) as function of x, y and partial derivatives of f. Considering as example the third derivative of y we see that

$$\begin{aligned} y'''(x) &= \frac{d}{dx}y''(x) \\ &= \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial x\,\partial y}y' + \frac{\partial^2 f}{\partial x\,\partial y}f + \frac{\partial^2 f}{\partial y^2}yf + \frac{\partial f}{\partial y}\left(\frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}y'\right) \end{aligned}$$

We can see that numerical method based on the Taylor series are conceptually easy but requires *a lot* of symbolical computation in order to express the derivatives as function of the known variables.

Example 1.4: explicit Euler scheme of the second order

Given the ODE

$$\begin{cases} y' = xy + x \\ y(0) = 1 \end{cases}$$

to express the 2^{nd} order Euler scheme we need to compute the partial derivatives

$$\frac{\partial f}{\partial x} = y + 1 \qquad\qquad \frac{\partial f}{\partial y} = x$$

and hence using equation 1.18 we obtain the method

$$y_{k+1} = y_k + h\big(x_k y_k + x_k\big) + \frac{h^2}{2}\Big[y_k + 1 + x_n\big(x_n y_n + x_n\big)\Big]$$

Higher order methods for this problem can be implemented explicitly computing the derivatives

$$\begin{aligned} y'(x) &= xy(x) + x \\ y''(x) &= y(x) + xy'(x) + 1 \\ y'''(x) &= y'(x) + y'(x) + xy''(x) = 2y'(x) + xy''(x) \\ y^{(4)}(x) &= 2y''(x) + y''(x) + xy'''(x) = 3y''(x) + xy'''(x) \\ &\vdots \end{aligned}$$

The idea is so to determine the numerical method as

$$y_{k+1} = y_k + hy'_k + \frac{h^2}{2}y''_k + \frac{h^3}{6}y'''_k + \frac{h^4}{24}y^{(4)}_k + \dots$$

where

$$\begin{aligned} y'_k = x_k y_k + x_k \quad &\to \quad y''_k = y_k + x_k y'_k + 1 \\ \to \quad y'''_k = 2y'_k + x_k y''_k \quad &\to \quad y^{(4)}_k = 3y''_k + xy'''_k \quad \to \quad \dots \end{aligned}$$

Economic Taylor scheme Recalling the higher order method shown in the previous example, a way to simplify the notation on the numerical method based on the Taylor series we consider the scheme

$$y_{k+1} = y_k + hy'_k + \frac{h^2}{2}y''_k + \dots + \frac{h^p}{p!}y^{(p)}_k$$

where the numerical derivatives are recursively defined considering that $y'_k = f(x_k, y_k)$:

$$\begin{aligned}
y'_k = D_1\big(x, y(x)\big) &= f(x,y) \\
y''_k = D_2\big(x, y(x), y'(x)\big) &= \frac{\partial D_1}{\partial x}(x,y) + \frac{\partial D_1}{\partial y}(x,y)\, y'(x,y) \\
y'''_k = D_3\big(x, y(x), y'(x), y''(x)\big) &= \frac{\partial D_2}{\partial x}(x,y) + \frac{\partial D_2}{\partial y} y' + \frac{\partial D_2}{\partial y'} y'' \\
&\vdots \\
y^{(p)}_k = D_p\big(x, y(x), y'(x), \dots, y^{(p-1)}(x)\big) &= \frac{\partial D_{p-1}}{\partial x} + \frac{\partial D_{p-1}}{\partial y} + \cdots + \frac{\partial D_{p-1}}{\partial y^{(p-2)}} y^{(p-1)}
\end{aligned}$$

1.3.2 Runge-Kutta

Considering the statements seen for numerical methods derived from the Taylor series expansion, we can see that the function f can be expanded by parameters α, β as

$$f(x+\alpha, y+\beta) = f(x,y) + \frac{\partial f}{\partial x}\alpha + \frac{\partial f}{\partial y}\beta + \frac{1}{2}\frac{\partial^2 f}{\partial x^2}\alpha^2 + \frac{\partial^2 f}{\partial x\, \partial y}\alpha\beta + \frac{1}{2}\frac{\partial^2 f}{\partial y^2}\beta^2 + \mathcal{O}\Big(\sqrt{\alpha^2+\beta^2}^{\,3}\Big)$$

The idea that originates the Runge-Kutta method is so to combine many evaluation of $f(x+\alpha, y+\beta)$ in order to match the results provided by the Taylor series; in general this match determines an error that however should be at maximum comparable to the order $\mathcal{O}(x^n)$ required. Considering the expansion previously discussed

$$y(x+h) = y(x) + hf\big(x, y(x)\big) + \frac{h^2}{2}\left(\frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} f\right) + \mathcal{O}(h^3)$$

we can consider the term

$$(i): \qquad h\, f\big(x, y(x)\big) + \frac{h^2}{2}\left(\frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} f\big(x, y(x)\big)\right)$$

and it's Taylor expansion

$$\begin{aligned}
h\omega f\big(x, y(x)\big) + h\gamma f\big(x+\alpha h, y(x)+\beta h\big) &= h\omega f + h\gamma\left(f + \frac{\partial f}{\partial x}\alpha h + \frac{\partial f}{\partial y}\beta h + \mathcal{O}(h^2)\right) \\
(ii): \qquad &= h\omega f + h\gamma f + h^2\gamma\alpha\frac{\partial f}{\partial x} + h^2\gamma\beta\frac{\partial f}{\partial y} + \mathcal{O}(h^3)
\end{aligned}$$

By now subtracting (ii) to (i) we determine

$$hf(1-\omega-\gamma) + h^2\frac{\partial f}{\partial x}\left(\frac{1}{2} - \gamma\alpha\right) + h^2\frac{\partial f}{\partial y}\left(\frac{f}{2} - \beta\gamma\right) + \mathcal{O}(h^3)$$

In order to reduce the error with a threshold $\mathcal{O}(h^3)$ we have to set to zero all the multiplicative terms of f (and it's derivatives) determining the following non-linear system:

$$\begin{cases} 1-\omega-\gamma = 0 \\ \frac{1}{2} - \gamma\alpha = 0 \\ \frac{f}{2} - \beta\gamma = 0 \end{cases}$$

If we determine parameters $\omega, \gamma, \alpha, \beta$ that satisfy such conditions we have that the expansion

$$y(x+h) = y(x) + \omega\, f\big(x, y(x)\big) + \gamma\, f\big(x+\alpha h, y(x)+\beta h\big) + \mathcal{O}(h^3)$$

matches the result obtained with Taylor; the system has 3 equation but the unknowns are 4, having so the a parametric solution in the form $\omega = 1-\gamma$, $\alpha = \frac{1}{2\gamma}$ and $\beta = \frac{f}{2\gamma}$ is always satisfied. Substituting this in the original expression we have that all the 2^{nd} order numerical method that do not use *explicitly* the partial derivatives of $f(x,y)$ are

$$y(x+h) = y(x) + h(1-\gamma)f(x,y(x)) + h\gamma f\left(x + \frac{h}{2\gamma}, y(x) + \frac{h}{2\gamma}f(x,y(x))\right) + \mathcal{O}(h^3)$$

determining the numerical method

$$y_{k+1} = y_k + h(1-\gamma)f(x_k,y_k) + h\gamma f\left(x_k + \frac{h}{2\gamma}, y_k + \frac{h}{2\gamma}f(x_k,y_k)\right) \tag{1.19}$$

Definition The idea of the **Runge-Kutta** method is use a combination of *displacements* in order to match *as much as possible* the Taylor expansion of the exact solution; in particular the numerical steps are written as

$$\boldsymbol{y}_{k+1} = \boldsymbol{y}_k + \sum_{i=1}^{s} b_j \boldsymbol{k}_j \tag{1.20}$$

where the s vectors $\boldsymbol{k}_j$ (where s is the order of the Runge-Kutta method) are obtained as

$$\begin{cases} \boldsymbol{k}_1 = h\,\boldsymbol{f}\left(x_k + c_1 h, y_k + \sum_{j=1}^{s} A_{1j}\boldsymbol{k}_j\right) \\ \boldsymbol{k}_2 = h\,\boldsymbol{f}\left(x_k + c_2 h, y_k + \sum_{j=1}^{s} A_{2j}\boldsymbol{k}_j\right) \\ \quad\vdots \\ \boldsymbol{k}_s = h\,\boldsymbol{f}\left(x_k + c_s h, y_k + \sum_{j=1}^{s} A_{sj}\boldsymbol{k}_j\right) \end{cases} \tag{1.21}$$

where the coefficients c_j, b_j, A_{ij} are computed in such a way that $y_{k+1} - y(x_{k+1}) = \mathcal{O}(h^p)$ (so the error between the computed value and the theoretical solution) where p is as large as possible. Such values are already tabled in the ***Runge-Kutta tableaus*** represented as

$$\begin{array}{c|c} \boldsymbol{c} & A \\ \hline & \boldsymbol{b}^t \end{array} \tag{1.22}$$

where $\boldsymbol{c} = (c_1, \ldots, c_s)$, $\boldsymbol{b} = (b_1, \ldots, b_s)$ (with $\boldsymbol{c}, \boldsymbol{b} \in \mathbb{R}^s$) and $A \in \mathbb{R}^{s\times s}$.

Runge-Kutta of order 4 A tableau for the Runge-Kutta method of order 4 is defined as

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{3} & \frac{1}{3} & & & \\ \frac{2}{3} & -\frac{1}{3} & -1 & & \\ 1 & 1 & -1 & 1 & \\ \hline & \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8} \end{array}$$

where the non-represented terms are zeros. Recalling equation 1.20 as the definition of the Runge-Kutta, we have that the numerical method derived from that is

$$y_{k+1} = y_j + \frac{1}{8}k_1 + \frac{3}{8}k_2 + \frac{3}{8}k_3 + \frac{1}{8}k_4$$

where

$$\begin{cases} k_1 &= h\,f(x_k, y_k) \\ k_2 &= h\,f\left(x_k + \frac{1}{3}h, y_k + \frac{1}{3}k_1\right) \\ k_3 &= h\,f\left(x_k + \frac{2}{3}h, y_k - \frac{1}{3}k_1 - k_2\right) \\ k_4 &= h\,f\left(x_k + h, y_k + k_1 - k_2 + k_3\right) \end{cases}$$

In this case the method is **explicit**: in fact we can sequentially compute k_1 (that depends on the knowns h, x_k, y_k) and sequentially k_2, then k_3 and lastly k_4.

Euler methods If we consider the *strange* tableau

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$$

what we obtain is the explicit Euler method, in fact having $k_1 = h\,f(x_k, y_k)$ determines the method

$$y_{k+1} = y_k + 1k_1 = y_k + hf(x_k, y_k)$$

and perfectly matches the definition provided in equation 1.13 at page 8. Considering now instead the tableau

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}$$

it determines an **implicit** method: we in fact have that $k_1 = h\,f(x_k + h, y_k + k_1)$ where the solution is implicit in k_1; considering that such relation can be regarded as $k_1 = hf(x_{k+1}, y_{k+1})$ what we determine is the implicit Euler method (equation 1.17, page 10)

$$y_{k+1} = y_k + h\,f\big(x_{k+1}, y_{k+1}\big)$$

If we in general have that all the elements A_{ij} with $j \geq j$ are all zeros, then the Runge-Kutta scheme is explicit meaning that all the coefficients $\boldsymbol{k}_1, \boldsymbol{k}_2, \ldots, \boldsymbol{k}_s$ can be computed consecutively. *Graphically* this means that the matrix A must have non-zero terms only below its principal matrix not included (it means that A_{ii} must always be zero).

One step methods

Usually we refer to ***one step methods*** the ones that allow to explicitly compute the next step as function of the current step in the form

$$y_{k+1} = \phi(x_k, y_h, h)$$

An example is the explicit Euler method (equation 1.13, page 8) that's characterized by the function $\phi(x_k, y_k, h) = y + hf(x_k, y_k)$. Also implicit methods can be one step; considering the implicit Euler (equation 1.17, page 10) it can be considered as

$$y_{k+1} = y_k + K_1 \qquad \text{with } K_1 = h\,f\big(x_{k+1}, y_k + K_1\big)$$

the value K_1 is formally a function of the parameters x_k, y_k, h, hence

$$G\big(K_1, x_k, y_k, h\big) = K - h\,f\big(x_k + hy_k + K\big) \qquad \Rightarrow \qquad \phi(x, y, h) = y + K(x, y, h)$$

In general all Runge-Kutta methods (both explicit and implicit) can be formally written in the form $y_{k+1} = \phi(x_k, y_k, h)$ and so are one step methods.

Error propagation

Known that each Runge-Kutta is a one-step method, then we can express the **local truncation error** $\tau_k(h)$ as the difference between the theoretical computed value and the numerical result obtained:

$$y\big(x_{k+1}\big) = \phi\big(x_k, y(x_k), h\big) + \tau_k(h) \qquad \Rightarrow \qquad \tau(h) = y(x+h) - \phi\big(x, y(x), h\big)$$

Considering the **error** $\varepsilon_k = y\big(x_k\big) - y_k$ as the difference between the analytical solution and the numerical approximated one, we can regard the two cases as

$$\begin{aligned} (i)\,: &\qquad y_{k+1} = \phi(x_k, y_k, h) \\ (ii)\,: &\qquad y\big(x_{k+1}\big) = \phi\big(x_k, y(x_k), h\big) + \tau_k(h) \end{aligned}$$

performing the difference $(i) - (ii)$ what we obtain is

$$\varepsilon_{k+1} = \Big(\phi\big(x_k, y(x_k), h\big) - \phi(x_k, y_k, h)\Big) + \tau_k(h)$$

Considering the simple case of the explicit Euler method characterized by a function $\phi(x, y, h) = y + h\, f(x, y)$, the difference in the parenthesis is in the form $\phi(x, z, h) - \phi(x, y, h) = z - y + h\big(f(x, z) - f(x, y)\big)$; computing it's absolute value we have and considering that f is lipschitzian we have

$$\big|\phi(x, z, h) - \phi(x, y, h)\big| \leq |z - y| + h\big|f(x, z) - f(x, y)\big| \leq (1 + hL)|z - y|$$

We can so rewrite the magnitude of the error as

$$|\varepsilon_{k+1}| \leq (1 + hL)\big|y(x_k) - y_k\big| + |\tau_k(h)| \leq (1 + hL)|\varepsilon_k| + |\tau_k(h)|$$

MIN 7.08

Chapter 2

Introduction to Algebraic Differential Equations

The **algebraic differential equations** **DAEs** can be regarded as a system of ordinary differential equations combined with *general* algebraic equations; as example a DAE system is

$$\begin{cases} y' = f(x,y) \\ y(a) = y_a \\ f(x,y) = 0 \\ x^2 - y = 3 \end{cases}$$

In general for ODEs and algebraic equations a lot of numerical methods have been implemented with consolidated theory regarding existence, stability... The problem is that when combining such theories in the algebraic differential equations, the numerical results that we might wanna retrieve are a *nightmare* to compute.

DAE with an example: simple pendulum Considering the simple pendulum of a mass m fixed by a bar of length l to a pivot point; considering such center as the origin of a reference frame, the coordinates of the mass can be described as function of using the minimal number of coordinates (associated in this case to lagrangian coordinate θ as the angle between the bar and the vertical line) as

$$x = l\sin\theta \qquad\qquad y = -l\cos\theta$$

The idea is that this coordinates satisfy the constraint $x^2 + y^2 = l^2$, meaning that the mass m can move only on the circle of radius l. Taking the velocities

$$\dot{x} = \frac{dx}{dt} = l\cos\theta\,\dot{\theta} \qquad\qquad \dot{y} = \frac{dy}{dt} = l\sin\theta\,\dot{\theta}$$

Computing the kinematic and potentials energy as

$$T = \frac{m}{2}(\dot{x}^2 + \dot{y}^2) = \frac{m}{2}l^2\dot{\theta}^2 \qquad\qquad V = mgy = -mgl\cos\theta$$

With this we can build the lagrangian $\mathcal{L} = T - V = \frac{m}{2}l^2\dot{\theta}^2 + mgl\cos\theta$ and using the Euler-Lagrange equation (following the minimal action principle) the differential equation describing the motion is

$$\frac{d}{dt}\frac{\partial\mathcal{L}}{\partial\dot{\theta}} - \frac{\partial\mathcal{L}}{\partial\theta} = ml^2\ddot{\theta} + mgl\sin\theta = l\ddot{\theta} + g\sin\theta = 0$$

where the ordinary differential equation, in order to be solved/integrated, requires the initial conditions $\theta(0) = \theta_0$ and $\dot{\theta}(0) = \dot{\theta}_0$. Introducing $\omega = \dot{\theta}$ we can reduce the system of ODEs to the first order

that can be numerically solved:

$$\begin{cases} \dot{\theta} = \omega \\ l\dot{\omega} + g\sin\theta = 0 \\ \theta(0) = \theta_0, \quad \omega(0) = \omega_0 = \dot{\theta}_0 \end{cases}$$

Observe that we obtained the solution as an ordinary differential equation because we found the *minimal* set of coordinates which describes the system.

As alternative approach we could have used simpler independent ordinary differential equations and add some constraints; considering the independent mass m described by the point (x, y) in the plane and constrained to move on a circle of radius l, it's kinetic and potential energies are still $T = \frac{m}{2}(\dot{x}^2 + \dot{y}^2)$ and $V = mgy$ (note that no transformation in terms of θ has been applied), then the lagrangian is

$$\mathcal{L} = T - V = \frac{m}{2}(\dot{x}^2 + \dot{y}^2) - mgy$$

The constraint is described by the equation $\phi(x, y) = x^2 + y^2 - l^2 = 0$. Adding to the constraint the least action principle stating that the functional $\mathcal{A} = \int_{t_0}^{t_1} \mathcal{L}(x, y, \dot{x}, \dot{y}, t)\, dt$ we can find the *stationary point* of the action $\mathcal{A}$ that's subject to $\phi(x, y) = 0$. Expanding the definition

$$\int_{t_0}^{t_1} \mathcal{L}(x, y, \dot{x}, \dot{y}, t) - \lambda\phi(x, y)\, dt$$

we can build the hamiltonian $\mathcal{H} = \mathcal{L} - \lambda\phi$ that, after the first variation, determines the system

$$\begin{cases} \frac{d}{dt}\frac{\partial \mathcal{H}}{\partial \dot{x}} - \frac{\partial \mathcal{H}}{\partial x} & = m\ddot{x} + \lambda x = 0 \\ \frac{d}{dt}\frac{\partial \mathcal{H}}{\partial \dot{y}} - \frac{\partial \mathcal{H}}{\partial y} & = m\ddot{y} + \lambda y = -mg \\ \phi(x, y) & = x^2 + y^2 - l^2 = 0 \end{cases}$$

that is a **differential algebraic equation**; introducing $\dot{x} = u$ and $\dot{y} = v$ we can simplify to a differential algebraic equation

$$\begin{cases} m\dot{u} + \lambda x = 0 \\ m\dot{v} + \lambda y = -mg \\ \dot{x} = u, \qquad \dot{y} = v \\ x^2 + y^2 - l^2 = 0 \end{cases} \tag{2.1}$$

Introduction to numerical methods for DAEs Considering the example of the pendulum, we can rewrite the differential equations as

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{u} \\ \dot{v} \\ 0 \end{pmatrix} = \begin{pmatrix} u \\ v \\ -\lambda x/m \\ -\lambda y/m - g \\ x^2 + y^2 - l^2 \end{pmatrix}$$

Defining $\boldsymbol{z} = (x, y, u, b, \lambda)$, such relation is similar to the form $\dot{\boldsymbol{z}} = \boldsymbol{F}(t, \boldsymbol{z})$. The main idea is in fact to **transform DAEs to ODEs**; note in fact that the last equation is the lonely one that's not already a ordinary differential equation: deriving it in time determines

$$\frac{d}{dt}(x^2 + y^2 - l^2) = 2x\dot{x} + 2y\dot{y} = 2xu + 2yv$$

but stull we observe that the variable λ is missing in the equation. Deriving one more time respect to t we obtain

$$\begin{aligned} \frac{d}{dt}(2xu + 2yv) &= 2\dot{x}u + 2x\dot{u} + 2\dot{y}v + 2y\dot{v} = 2u^2 + 2v^2 - 2x\frac{\lambda x}{m} - 2y\left(\frac{\lambda y}{m} + g\right) \\ &= 2(u^2 + v^2) - \frac{2\lambda}{m}(x^2 + y^2) - 2yg \end{aligned} \tag{2.2}$$

By substituting the different known relations for $\dot{x}, \dot{y}, \dot{u}, \dot{v}$ we so obtain a derivative that's function of λ, but not of $\dot{\lambda}$. Deriving one more time respect to the variable t

$$\begin{aligned}\frac{d}{dt}(2.2) &= 4\left(u\dot{u} + v\dot{v}\right) - \frac{4\lambda}{m}\left(x\dot{x} + y\dot{y}\right) - 2\dot{y}g - \frac{2}{m}\dot{\lambda}\left(x^2 + y^2\right) \\ &= 4\left(-u\frac{\lambda x}{m} - v\frac{\lambda y}{m} - vg\right) - \frac{4\lambda}{m}\left(xu + yv\right) - 2vg - \frac{2}{m}\dot{\lambda}\left(x^2 + y^2\right) = 0\end{aligned}$$

Solving for $\dot{\lambda}$ so gives

$$\dot{\lambda} = \frac{-4\lambda\left(xy + yv\right) - 3vmg}{x^2 + y^2}$$

We can so rewrite the differentia algebraic system in equation 2.1 as a system of ODE only as

$$\begin{cases} \dot{x} = u \\ \dot{y} = v \\ \dot{u} = -\frac{l}{m}x \\ \dot{v} = -\frac{\lambda}{m}y - g \\ \dot{\lambda} = \frac{-4\lambda\left(xy+yv\right)-3vmg}{x^2+y^2} \end{cases}$$

With such definition we can use numerical methods to solve the form $\dot{\boldsymbol{z}} = \boldsymbol{F}(t, \boldsymbol{z})$. This formulations however introduces some problems:

- the initial condition on λ is not set. This problem can be overcome considering that given x, the coordinate y is constrained by $x^2 + y^2 = l^2$. If we moreover know $\dot{x} = u$ then using the derivative of the constraint $2xy + 2yv = 0$, then also $v = \dot{y}$ is constrained. Using the second derivative of the constraint (equation 2.2) we can finally solve for λ_0. We see that the initial conditions must satisfy the *original* constraints and the *hidden ones* determined by the derivatives of the algebraic equations.
- another problem is that if we considered a constraint in the form

  $$\phi(x, y) = x^2 + y^2 - l^2 + a + bt + ct^2 = 0$$

  in order to obtain the ODE equivalent system we have to derive ϕ three times over time resulting in the cancellations of the polynomial terms $a + bt + ct^2$ (we in fact would have obtained the same ODE system).

2.1 Linear differential algebraic equations

Starting from the simplest cases of study, a generic differential algebraic equation can be written as

$$\begin{cases} \boldsymbol{F}(t, \boldsymbol{y}, \boldsymbol{y}') = 0 \\ \boldsymbol{y}(a) = \boldsymbol{y}_a \end{cases}$$

with $\boldsymbol{y}(t) \in \mathbb{R}^n$. We can say that the map $\boldsymbol{F}$ is **linear** in y if it can be expressed as linear combination of $\boldsymbol{y}'$ in the form

$$\boldsymbol{F}(t, \boldsymbol{y}, \boldsymbol{y}') = \mathbf{E}(t, \boldsymbol{y})\boldsymbol{y}' + \boldsymbol{G}(t, \boldsymbol{y}) \tag{2.3}$$

Moreover the map $\boldsymbol{F}$ is linear in both $\boldsymbol{y}$ and $\boldsymbol{y}'$ if it can be regarded as

$$\boldsymbol{F}(t, \boldsymbol{y}, \boldsymbol{y}') = \mathbf{E}(t)\boldsymbol{y}' + \mathbf{A}(t)y - \boldsymbol{C}(t) \tag{2.4}$$

where in general $\mathbf{E}, \mathbf{A}$ are matrices that can sometimes be singular. The map $\boldsymbol{F}$ is said **linear with constant coefficients** if it happens that the matrices $\mathbf{E}, \mathbf{A}$ are time independent.

Example 2.1: linear DAEs

An example of linear differential algebraic equation is the one that can be described as

$$\begin{bmatrix} 1 & t \\ t^2 & 2 \end{bmatrix} \begin{pmatrix} y_1' \\ y_2' \end{pmatrix} + \begin{bmatrix} \sin t & \cos t \\ t^2 & 1 \end{bmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} - \begin{pmatrix} e^t \\ 1+t \end{pmatrix}$$

Observe that if the matrix $\boldsymbol{E}$ in non singular, expression 2.3 corresponds to a *simple* ordinary differential equation: it can be in fact rewritten as

$$\boldsymbol{y}' = -\mathbf{E}^{-1}(t, \boldsymbol{y})\boldsymbol{G}(t, \boldsymbol{y}) = \boldsymbol{f}(t, \boldsymbol{y}) \tag{2.5}$$

For the moment we can assume that if $\mathbf{E}$ is singular the system is not an ODE. In this first part we will focus on linear differential algebraic equations in order to exploit linear algebra tools to ease the calculations.

Jordan normal form As a recall from the linear algebra, given a matrix $\mathbf{B} \in \mathbb{R}^{n \times n}$ there exists always a non-singular matrix $\mathbf{T} \in \mathbb{R}^{n \times n}$ such that

$$\mathbf{T}^{-1}\mathbf{B}\mathbf{T} = \mathbf{J}$$

where $\mathbf{J}$ is the **Jordan matrix form** defined as

$$\mathbf{J} = \begin{bmatrix} J_1 & & & 0 \\ & J_2 & & \\ & & \ddots & \\ 0 & & & J_m \end{bmatrix} \qquad \text{where } J_k = \begin{bmatrix} \lambda_k & 1 & & 0 \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ 0 & & & \lambda_k \end{bmatrix} \tag{2.6}$$

Regular pencil Given the matrices $\mathbf{B}, \mathbf{C} \in \mathbb{R}^{n \times n}$, the couple $(\mathbf{B}, \mathbf{C})$ is a **regular pencil** if

$$f(\lambda) = \det\left(\mathbf{B} - \lambda\mathbf{C}\right) \neq 0$$

is not identically null, or equivalently if there exists a λ such that $f(\lambda) \neq 0$. Considering as example the matrices

$$\mathbf{B} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \qquad \mathbf{C} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

is not a regular pencil, in fact the polynomial $f(\lambda) = \det(\mathbf{B} - \lambda\mathbf{C}) = \det \begin{bmatrix} 1-\lambda & 1 \\ 0 & 0 \end{bmatrix} = 0$ always evaluates to zero.

Nilpotent matrix A matrix $\mathbf{B} \in \mathbb{R}^{n \times n}$ is **nilpotent** of order p if

$$\mathbf{B}^p = 0 \quad \text{and} \quad \mathbf{B}^j \neq 0 \; \forall j < p \tag{2.7}$$

where $\mathbf{B}^p$ is the product $\mathbf{B}\mathbf{B}\ldots\mathbf{B}$ p times. Considering as example

$$\mathbf{B} = \begin{bmatrix} 0 & 1 & 2 \\ 0 & 0 & 3 \\ 0 & 0 & 0 \end{bmatrix} \qquad \mathbf{B}^2 = \begin{bmatrix} 0 & 0 & 3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \mathbf{B}^3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

we have that such matrix $\mathbf{B}$ is nilpotent of order 3. Observe that if the matrix $\mathbf{B}$ is non-singular, than it can't be nilpotent.

Kronecker normal form If we consider two regular pencil matrices $(\mathbf{B}, \mathbf{C}) \in \mathbb{R}^{n\times n}$, then there exists two non-singular matrices $\mathbf{P}, \mathbf{Q} \in \mathbb{R}^{n\times n}$ such that

$$\mathbf{PBQ} = \begin{bmatrix} \mathbf{N} & 0 \\ 0 & \mathbf{I} \end{bmatrix} \qquad \text{and} \qquad \mathbf{PCQ} = \begin{bmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{J} \end{bmatrix} \tag{2.8}$$

where $\mathbf{N}$ is a nilpotent matrix, $\mathbf{I}$ is the identity matrix and $\mathbf{J}$ is a Jordan normal form matrix. Considering that the *blocks* $\mathbf{N}, \mathbf{J}$ can be empty, as extreme cases we have $\mathbf{PBQ} = \mathbf{I}$, $\mathbf{PCQ} = \mathbf{J}$, $\mathbf{PBQ} = \mathbf{N}$ and $\mathbf{PCQ} = \mathbf{I}$.

2.1.1 Usage of the Kronecker normal form

To ease the computation of linear differential algebraic equation, we can use the Kronecker normal form assuming that the couple of matrices $(\mathbf{E}, \mathbf{A})$ (equation 2.4) are a regular pencil (in order not to have an *inconsistent* DAE).

Example 2.2: inconsistent DAE

Using the matrices defined used in the theory of regular pencil, we can build a DAE system of the form

$$\begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \begin{pmatrix} y_1' \\ y_2' \end{pmatrix} + \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} t \\ 1 \end{pmatrix}$$

the associated system is

$$\begin{cases} y_1' + y_2' + y_2 = t \\ 0 = 1 \end{cases}$$

that's inconsistent.

With such assumption we can compute the Kronecker normal form $\mathbf{PEQ} = \begin{bmatrix} \mathbf{N} & \\ & \mathbf{I} \end{bmatrix}$ and $\mathbf{PAQ} = \begin{bmatrix} \mathbf{I} & \\ & \mathbf{J} \end{bmatrix}$; premultiplying so equation 2.4 by $\mathbf{P}$ results in $\mathbf{PE}\boldsymbol{y}' + \mathbf{PA}\boldsymbol{y} = \mathbf{P}\boldsymbol{C}$. Performing the change of variable $\mathbf{Q}\boldsymbol{z} = \boldsymbol{y}$ (hence $\boldsymbol{z} = \mathbf{Q}^{-1}$) and observing that $\mathbf{Q}\boldsymbol{z}' = \boldsymbol{y}'$ we obtain the expression $\mathbf{PEQ}\boldsymbol{z}' + \mathbf{PAQ}\boldsymbol{z} = \mathbf{P}\boldsymbol{C}$ on top of which we can apply the Kronecker normal form:

$$\begin{bmatrix} \mathbf{N} & \\ & \mathbf{I} \end{bmatrix} \boldsymbol{z}' + \begin{bmatrix} \mathbf{I} & \\ & \mathbf{J} \end{bmatrix} \boldsymbol{z} = \mathbf{P}\boldsymbol{C} \tag{2.9}$$

Splitting both vectors $\boldsymbol{z} = (\boldsymbol{\alpha}, \boldsymbol{\beta})$ and $\mathbf{P}\boldsymbol{C} = (\boldsymbol{d}, \boldsymbol{e})$ we can rewrite this expression as

$$\begin{bmatrix} \mathbf{N} & \\ & \mathbf{I} \end{bmatrix} \begin{pmatrix} \boldsymbol{\alpha}' \\ \boldsymbol{\beta}' \end{pmatrix} + \begin{bmatrix} \mathbf{I} & \\ & \mathbf{J} \end{bmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{pmatrix} = \begin{pmatrix} \boldsymbol{d} \\ \boldsymbol{e} \end{pmatrix}$$

The associated linear system representation is

$$\begin{cases} i) & \mathbf{N}\boldsymbol{\alpha}' + \boldsymbol{\alpha} = \boldsymbol{d}(t) \\ ii) & \boldsymbol{\beta}' + \mathbf{J}\boldsymbol{\beta} = \boldsymbol{e} \end{cases}$$

$ii)$ represent a *standard* system of ordinary differential equations; the term $i)$ is instead more complex but we can invert the relation to obtain $\boldsymbol{\alpha} = \boldsymbol{d} - \mathbf{N}\boldsymbol{\alpha}'$. Observing that the k-th derivative in time of this expression evaluates to $\boldsymbol{\alpha}^{(k)} = \boldsymbol{d}^{(k)} - \mathbf{N}\boldsymbol{\alpha}^{k+1}$, we can substitute the derivatives $\boldsymbol{\alpha}^{(k)}$ determining

$$\begin{aligned} \boldsymbol{\alpha} &= \boldsymbol{d} - \mathbf{N}\boldsymbol{\alpha}' = \boldsymbol{d} - \mathbf{N}(\boldsymbol{d}' - \mathbf{N}\boldsymbol{\alpha}'') = \boldsymbol{d} - \mathbf{N}\boldsymbol{d}' + \mathbf{N}^2\boldsymbol{\alpha}'' = \boldsymbol{d} - \mathbf{N}\boldsymbol{d}' + \mathbf{N}^2(\dots) = \dots \\ &= \boldsymbol{d} - \mathbf{N}\boldsymbol{d}' + \mathbf{N}^2\boldsymbol{d}'' - \mathbf{N}^3\boldsymbol{d}''' + \mathbf{N}^4\boldsymbol{d}^{(4)} - \mathbf{N}^5\boldsymbol{d}^{(5)} + \dots \end{aligned}$$

This series is infinite, however being $\mathbf{N}$ a nilpotent matrix of order p we have only that the first p terms remains and so

$$\boldsymbol{\alpha} = \boldsymbol{d} - \mathbf{N}\boldsymbol{d}' + \mathbf{N}^2\boldsymbol{d}'' + \cdots + (-1)^{p-1}\mathbf{N}^{p-1}\boldsymbol{d}^{(p-1)} + \cancel{(-1)^p\mathbf{N}^p\boldsymbol{d}^{(p)}} = \sum_{j=0}^{p-1}(-1)^j\mathbf{N}^j\boldsymbol{d}^{(j)} \tag{2.10}$$

With this relation we determined $\boldsymbol{\alpha}$ without using the initial conditions (as was required for the DEA solution using the conversion to ODE), depends only on $\boldsymbol{d}$ (and it's derivatives up to the $p-1$ order). Also observe that $ii)$ is a *regular* ODE, hence the initial values $\boldsymbol{\beta}(0)$ must be specified.

The order of the nilpotency of the matrix $\mathbf{N}$ is the so called **index** of the differential algebraic equation (for the linear ones) and is a sort of *measure* of the *difficulty* of solving numerically the DAE. In particular if $p = 0$ then what we have is a system of ordinary differential equation while if $p = n$ we have a set of only algebraic equations.

2.1.2 LU decomposition and Jacobi modification

In general the Kronecker normal form is *hard* to compute; computationally we can use the *simpler* **LU decomposition** in order to reduce the index of a differential algebraic equation, or better transform the DAE in an ordinary differential equation.

Recall on the LU decomposition Considering $\mathbf{A} \in \mathbb{R}^{n\times n}$ a square matrix, then there exists 2 permutation matrices $\mathbf{P}, \mathbf{Q}$ such that

$$\mathbf{PAQ} = \mathbf{LU} \tag{2.11}$$

where $\mathbf{L}$ is a **lower triangular matrix** and $\mathbf{U}$ is an **upper** triangular one (in particular if the matrix $\mathbf{A}$ is singular only the first $m < n$ rows of $\mathbf{U}$ are non-zero and are still triangular, having a *trapezoidal shape*).

In the case that $\mathbf{A}$ is non-singular, then the algorithm can be reduced to the form $\mathbf{PA} = \mathbf{LU}$; alternatively we can use the **Echelon form** determined as $\mathbf{PA} = \mathbf{LUQ}^T = \mathbf{L\tilde{U}}$.

A **permutation matrix** S_{ij} are used to exchange the i-th and j-th row/column of a matrix A; in particular $S_{ij}A = \tilde{A}$ results in a swapping of the rows while $AS_{ij} = \tilde{\tilde{A}}$ is the exchange of the columns i and j. Observe that $S_{ij}S_{ij} = \mathcal{I}$ results in the identity matrix and that permutations matrices are symmetric, in the sense that $S_{ij}^T = S_{ij}$.

With that said if we consider a series of multiplication on permutation matrix we have that

$$P = S_{ij}S_{kl}\ldots S_{mp} \qquad \Rightarrow \qquad P^T = S_{mp}^T\ldots S_{kl}^TS_{ij}^T = S_{mp}\ldots S_{kl}S_{ij}$$

because we have that $P^TP = \mathcal{I}$. In general a permutation matrix if a series of product of exchanges collected in a single matrix P such that $P^{-1} = P^T$.

Jacobi modification The main idea of the LU decomposition is to find $n-1$ matrices $\mathbf{L}_i$ such that $\mathbf{L}_{n-1}\ldots\mathbf{L}_2\mathbf{L}_1\mathbf{PAQ} = \mathbf{U}$ (where $\mathbf{L} = (\mathbf{L}_{n-1}\ldots\mathbf{L}_2\mathbf{L}_1)^{-1}$), where $\mathbf{U}$ is upper triangular and L_i are all lower ones. The **Jacobi modification** leverage the same idea, but the permuted matrix $\mathbf{PAQ}$ is pre-multiplied by a series of Jordan normal matrices $\mathbf{J}_i$ resulting in a matrix of the form:

$$\mathbf{J}_n\ldots\mathbf{J}_2\mathbf{J}_1\mathbf{PAQ} = \left[\begin{array}{c|c} \mathbf{I} & \\ \hline 0 & \end{array}\right]$$

where the *blank spaces* can be filled with non-zero elements.

2.2 DAE index and index reduction

As already discussed, a linear differential algebraic equation can be regarded as $\mathbf{E}\boldsymbol{y}' + \mathbf{A}\boldsymbol{y} = \boldsymbol{c}(t)$ (where we assume that the pair $(\mathbf{E}, \mathbf{A})$ is a regular pencil), then with the Kronecker decomposition we had the formulation shown in equation 2.9 (page 20). That allowed to re-state the original DAE problem into an algebraic part $\boldsymbol{\alpha} = \sum_{j=0}^{p-1}(-1)^j \mathbf{N}^j \boldsymbol{d}^{(j)}$ and an ordinary differential one $\boldsymbol{\beta}'(t) + \mathbf{J}\boldsymbol{\beta}(t)\boldsymbol{e}(t)$. Deriving the algebraic part evaluates to $\boldsymbol{\alpha}'(t) = \sum_{j=0}^{p-1}(-\mathbf{N})^j \boldsymbol{d}^{(j+1)}(t)$, meaning that the Kronecker normal form allows to transform the original DAE problem into a system of ordinary differential equations:

$$\begin{pmatrix} \boldsymbol{\alpha}' \\ \boldsymbol{\beta}' \end{pmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{J} \end{bmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{pmatrix} = \begin{pmatrix} \boldsymbol{e}(t) \\ \sum_{j=0}^{p-1}(-\mathbf{N})^j \boldsymbol{d}^{(j+1)}(t) \end{pmatrix} \tag{2.12}$$

We can observe so that starting from a linear differential algebraic equation with constant coefficients $\mathbf{E}\boldsymbol{y}' + \mathbf{A}\boldsymbol{y} = \boldsymbol{c}(t)$ we have that after p derivation (where p is the nilpotency order of $\mathbf{N}$ in the Kronecker normal form) and *some algebraic manipulation* we obtain an ordinary differential equation; as definition we say that the DAE has a **Kronecker index** p.

Differential index The minimum number of derivations of the system $\mathbf{E}\boldsymbol{y}' + \mathbf{A}\boldsymbol{y} = \boldsymbol{c}(t)$ required to transform the DAE into an ODE is called **differential index**.

Assuming the general expression $\boldsymbol{F}(\boldsymbol{y}, \boldsymbol{y}', t) = 0$, but also the derivatives $\frac{d}{dt}\boldsymbol{F}(\boldsymbol{y}, \boldsymbol{y}', t) = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{y}}\boldsymbol{y}' \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{y}'}\boldsymbol{y}'' + \frac{\partial \boldsymbol{F}}{\partial t}$, $\frac{d^2}{dt^2}\boldsymbol{F}$ up to the p-th order $\frac{d^p}{dt^p}\boldsymbol{F}$, such values can be combined to obtain an expression in the form $\boldsymbol{y}' = \boldsymbol{G}(\boldsymbol{y}, t)$. The differential index is the minimum number of derivations p required to transform $\boldsymbol{F}(\boldsymbol{y}, \boldsymbol{y}', t) = 0$ into $\boldsymbol{g}' = \boldsymbol{G}(\boldsymbol{y}, t)$.
As special cases

- if the Kronecker normal form is of type $\mathbf{PEQ} = \mathbf{I}$ and $\mathbf{PAQ} = \mathbf{J}$, then we have that

$$\begin{aligned} \mathbf{PE}\boldsymbol{y}' + \mathbf{PA}\boldsymbol{y} &= \mathbf{P}\boldsymbol{c} \\ \mathbf{PEQQ}^T\boldsymbol{y}' + \mathbf{PAQQ}^T\boldsymbol{y} &= \mathbf{P}\boldsymbol{c} \qquad \leftarrow \quad \boldsymbol{z} = \mathbf{Q}^T\boldsymbol{y} \\ \mathbf{I}\boldsymbol{z}' + \mathbf{J}\boldsymbol{z} &= \mathbf{P}\boldsymbol{c} \\ \Rightarrow \qquad \boldsymbol{z}' &= \mathbf{P}\boldsymbol{c} - \mathbf{J}\boldsymbol{z} \end{aligned}$$

  This is a purely ordinary differential equation.

- alternatively if the Kronecker normal form is such that $\mathbf{PEQ} = \mathbf{N}$ and $\mathbf{PAQ} = \mathbf{I}$, then we have

$$\begin{aligned} \mathbf{PEQQ}^T\boldsymbol{y}' + \mathbf{PAQQ}^T\boldsymbol{y} &= \mathbf{P}\boldsymbol{c} \qquad \leftarrow \quad \boldsymbol{z} = \mathbf{Q}^T\boldsymbol{y} \\ \mathbf{N}\boldsymbol{z}' + \boldsymbol{z} &= \mathbf{P}\boldsymbol{c} = \boldsymbol{d} \end{aligned}$$

  Knowing that $\mathbf{N}$ is a nilpotent matrix of order p, recalling previous *tricks we have that*

$$\boldsymbol{z} = \sum_{j=0}^{p-1}(-\mathbf{N})^j \boldsymbol{d}^{(j)}$$

  This is a purely algebraic equation.

Example 2.3: DAE to ODE

Considering the differential algebraic equation

$$\begin{cases} \dot{x}_1 + \dot{x}_2 + x_1 = 1 \\ \dot{x}_1 + \dot{x}_2 + x_1 + x_2 = t \end{cases}$$

subtracting from the second equation the first one results in $x_2 = t - 1$ that derived determines $\dot{x}_2 = 1$: we have so $\dot{x}_2$ expressed as an ordinary differential equation. Substituting this result in the first equation we have $\dot{x}_1 + 1 + x_1 = 0$, hence $\dot{x}_1 = -x_1$. After 1 derivation the resulting ODE is

$$\begin{cases} \dot{x}_1 = -x_1 \\ \dot{x}_2 = 1 \end{cases}$$

The differential index in this case is 1 (because we only derived $\dot{x}_2 = t - 1$ once).

Systematic index reduction algorithm Given a linear DAE with constant coefficients $\mathbf{E}\boldsymbol{y}' + \mathbf{A}\boldsymbol{y} = \boldsymbol{c}$ (where usually $\mathbf{E}$ is singular), we can find two permutation matrices $\mathbf{P}, \mathbf{Q}$ in order to have the factorization $\mathbf{PEQ} = \mathbf{LU}$ where $\mathbf{L}$ is lower triangular and $\mathbf{Q}$ is *upper trapezoidal* due to singularity of E. Knowing that $\mathbf{Q}^{-1} = \mathbf{Q}^T$, we also have that $\mathbf{PE} = \mathbf{LUQ}^T$: this last permutation $\mathbf{Q}^T$ corresponds simply to column commutations, meaning that we can regard

$$\mathbf{PE} = \mathbf{LUQ}^T = \mathbf{LM}$$

where $\mathbf{M}$ is a matrix that in general is composed in two *vertically stacked rectangular blocks*: the upper one that' generally non zero and the lower one identically null, hence $\mathbf{M} = \left[\begin{array}{c} \mathbf{M}_1 \neq 0 \\ \hline \mathbf{M}_2 = 0 \end{array}\right]$. With this consideration we can pre-multiply the linear DAE by $\mathbf{L}^{-1}\mathbf{P}$ what we obtain is the formulation

$$\begin{aligned} \mathbf{L}^{-1}\mathbf{PE}\boldsymbol{y}' + \mathbf{L}^{-1}\mathbf{PA}\boldsymbol{y} &= \mathbf{L}^{-1}\mathbf{P}\boldsymbol{c} \\ \mathbf{M}\boldsymbol{y}' + \mathbf{N}\boldsymbol{y} &= \boldsymbol{d} \end{aligned} \tag{2.13}$$

Splitting the matrices/vector $\mathbf{M}, \mathbf{N}, \boldsymbol{d}$ in order to *match* the dimension of the blocks of $\mathbf{M}$, then we can consider the initial differential algebraic equations as made of

$$\begin{cases} \mathbf{M}_1\boldsymbol{y}' + \mathbf{N}_1 = \boldsymbol{d}_1 & : \text{differential part} \\ \mathbf{N}_2\boldsymbol{y} = \boldsymbol{d}_2 & : \text{algebraic part} \end{cases}$$

The easiest thing to do to reduce the DAE into a system of ordinary differential equation is to derive in time the algebraic part, that evaluates to $\mathbf{N}_2\boldsymbol{y}' = \boldsymbol{d}_2'$: with that we obtain a *pure* system of ordinary differential equations in the form

$$\begin{cases} \mathbf{M}_1\boldsymbol{y}' + \mathbf{N}_1 = \boldsymbol{d}_1 & : \text{differential part} \\ \mathbf{N}_2\boldsymbol{y}' = \boldsymbol{d}_2' & : \text{algebraic part} \end{cases}$$

This results in a *new* differential algebraic system in the form $\mathbf{E}_1\boldsymbol{y}' + \mathbf{A}_1\boldsymbol{y} = \boldsymbol{e}_1$ where

$$\mathbf{E}_1 = \begin{bmatrix} \mathbf{M}_1 \\ \mathbf{N}_2 \end{bmatrix} \qquad \mathbf{A}_1 = \begin{bmatrix} \mathbf{N}_1 \\ 0 \end{bmatrix} \qquad \boldsymbol{e}_1 = \begin{pmatrix} \boldsymbol{d}_1 \\ \boldsymbol{d}_2' \end{pmatrix}$$

The so computed matrix $\mathbf{E}_1$ can still be singular, but we can apply this same algorithm until the obtained system is non-singular, hence solvable: the number of required derivations is the **index** of the differential algebraic equation.

Example 2.4: index computation and reduction

Considering a differential algebraic system in the form

$$\begin{cases} \dot{x}_1 + \dot{x}_2 + \dot{x}_3 + x_1 = \sin t \\ \dot{x}_1 + \dot{x}_2 + \dot{x}_3 + x_3 = t \\ x_1 + x_3 = \cos t \end{cases}$$

in order to perform the index reduction we can rewrite the system in a more *compact* form in order to perform more easily the Gauss-Jordan solution of the system:

$$\begin{array}{ccccccc} \dot{x}_1 & \dot{x}_2 & \dot{x}_3 & x_1 & x_2 & x_3 & \text{RHS} \\ \left[\begin{array}{ccc|ccc|c} 1 & 1 & 1 & 1 & 0 & 0 & \sin t \\ 1 & 1 & 1 & 0 & 0 & 1 & t \\ 0 & 0 & 0 & 1 & 0 & 1 & \cos t \end{array}\right] \end{array}$$

Performing the transformation on the rows of such matrix $(2) \mapsto (2) - (1)$ we obtain the matrix

$$\left[\begin{array}{ccc|ccc|c} 1 & 1 & 1 & 1 & 0 & 0 & \sin t \\ 0 & 0 & 0 & -1 & 0 & 1 & t - \sin t \\ 0 & 0 & 0 & 1 & 0 & 1 & \cos t \end{array}\right]$$

Observing that the last two rows are identically zero for what concerns the entries in $\mathbf{E}$, then we can derive such algebraic part by *shifting* to the left the block $\mathbf{N}_2$ and deriving in time the last column associated to the right hand side $\boldsymbol{d}$, obtaining

$$\left[\begin{array}{ccc|ccc|c} 1 & 1 & 1 & 1 & 0 & 0 & \sin t \\ -1 & 0 & 1 & 0 & 0 & 0 & 1 - \cos t \\ 1 & 0 & 1 & 0 & 0 & 0 & -\sin t \end{array}\right]$$

Continuing the Gauss-Jordan reduction

$$\xrightarrow[(3)\mapsto(3)-(1)]{(2)\mapsto(2)+(1)} \left[\begin{array}{ccc|ccc|c} 1 & 1 & 1 & 1 & 0 & 0 & \sin t \\ 0 & 1 & 2 & 1 & 0 & 0 & 1 - \cos t + \sin t \\ 0 & -1 & 0 & -1 & 0 & 0 & -2\sin t \end{array}\right]$$

$$\xrightarrow[(3)\mapsto(3)+(2)]{(1)\mapsto(1)-(2)} \left[\begin{array}{ccc|ccc|c} 1 & 0 & -1 & 0 & 0 & 0 & \cos t - 1 \\ 0 & 1 & 2 & 1 & 0 & 0 & 1 - \cos t + \sin t \\ 0 & 0 & 2 & 0 & 0 & 0 & 1 - \cos t - \sin t \end{array}\right]$$

$$\xrightarrow[(2)\mapsto(2)-(3)]{(1)\mapsto(1)+\frac{1}{2}(3)} \left[\begin{array}{ccc|ccc|c} 1 & 0 & 0 & 0 & 0 & 0 & \frac{\cos t - \sin t - 1}{2} \\ 0 & 1 & 0 & 1 & 0 & 0 & 2\sin t \\ 0 & 0 & 2 & 0 & 0 & 0 & 1 - \cos t - \sin t \end{array}\right]$$

After all this reductions we obtained a non-singular matrix $\mathbf{E}$ determined just after one differentiation of the original problem (hence the index of the DAE is $p = 1$) and the resulting ordinary differential equation is

$$\begin{cases} \dot{x}_1 = \frac{1}{2}(\cos t - \sin t - 1) \\ \dot{x}_2 = 2\sin t - x_1 \\ \dot{x}_3 = \frac{1}{2}(1 - \cos t - \sin t) \end{cases}$$

When the algorithm fails Considering now a differential algebraic system characterized by the equations

$$\begin{cases} x' + y' + z' + w' + x = t \\ x' - x = t^2 \\ y' - x = t \\ x' + y' = 0 \end{cases} \quad \leftrightarrow \quad \left[\begin{array}{cccc|cccc|c} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & t \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & t^2 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & t \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

after some steps in the Jordan-Gauss reduction for linear systems we obtain the form

$$\left[\begin{array}{cccc|cccc|c} 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & t^2 \\ 0 & -1 & -1 & -1 & -2 & 0 & 0 & 0 & t^2 - t \\ 0 & 0 & -1 & -1 & -1 & 0 & 0 & 0 & -t \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t - t^2 \end{array}\right]$$

Clearly the last row is a contradiction, in fact what it says is that $0 = t - t^2$ that's not verified in general. This is due to the fact that the initial choice of the pair $(\mathbf{A}, \mathbf{E})$ was **not** a **regular pencil**: we in fact have that

$$\det(\mathbf{E} - \lambda\mathbf{A}) = \begin{vmatrix} 1-\lambda & 1 & 1 & 1 \\ 1+\lambda & 0 & 0 & 0 \\ 1+\lambda & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{vmatrix} = - \begin{vmatrix} 1+\lambda & 0 & 0 \\ 1+\lambda & 0 & 0 \\ 1 & 1 & 0 \end{vmatrix} = 0$$

where the expansions used to compute the determinant are made along the last column on each submatrix.

Linear DAE with non-constant coefficients Considering the more general case of non-constant coefficient for linear differential algebraic equation, hence in the form $\mathbf{E}(t)\boldsymbol{y}' + \mathbf{A}\boldsymbol{y} = \boldsymbol{c}$, we might want to know if the condition on the regular pencil $(\mathbf{A}, \mathbf{E})$ must still be required. Considering the simple system $\mathbf{E}(t) = \begin{bmatrix} 1 & t \\ 0 & 0 \end{bmatrix}$ and $\mathbf{A}(t) = \begin{bmatrix} 0 & 0 \\ 1 & t \end{bmatrix}$ we can show that they are not regular pencil, in fact

$$\det(\mathbf{E} - \lambda\mathbf{A}) = \begin{vmatrix} 1 & t \\ 1-\lambda & t-\lambda \end{vmatrix} = t - \lambda - t(1-\lambda) = 0$$

However writing explicitly the differential algebraic equation we have

$$\begin{cases} x' + ty' = c_1 & : \text{differential part} \\ x + ty = c_2 & : \text{algebraic part} \end{cases}$$

Deriving in time the algebraic equation results in the system

$$\begin{cases} x' + ty' = c_1 \\ x' + ty' + y = c_2' \end{cases} \xrightarrow{(2)\mapsto(2)-(1)} \begin{cases} x' + ty' = c_1 \\ y = c_2' - c_1 \end{cases}$$

Deriving one more time the second equation (that's algebraic) allows to compute a *non-singular* system of ODEs characterized by solutions

$$y' = c_2'' - c_1' \qquad x' = c_1 - (c_2'' - c_1')t$$

The index of the original DAE is so 2.
In general not being regular pencil for non-constant linear DAEs is not a problem; in fact the the system $\mathbf{E}(t)\boldsymbol{x}' + \mathbf{A}(t)\boldsymbol{x} = \boldsymbol{c}$, after some algebraic manipulation, can be reduced to a form

$$\begin{bmatrix} \tilde{\mathbf{E}}_1(t) \\ 0 \end{bmatrix} \boldsymbol{x}' + \begin{bmatrix} \tilde{\mathbf{A}}_1(t) \\ \tilde{\mathbf{A}}_2(t) \end{bmatrix} \boldsymbol{x} = \begin{pmatrix} \tilde{c}_1 \\ \tilde{c}_2 \end{pmatrix}$$

Deriving the algebraic part this time doesn't mean just *shifting* $\tilde{\mathbf{A}}_2(t)$ on the left to complex the matrix $\tilde{\mathbf{E}}$, but also introduces the derivative of the entries of $\tilde{\mathbf{A}}_2(t)$:

$$\begin{bmatrix} \tilde{\mathbf{E}}_1(t) \\ \tilde{\mathbf{A}}_2(t) \end{bmatrix} \boldsymbol{x}' + \begin{bmatrix} \tilde{\mathbf{A}}_1(t) \\ \tilde{\mathbf{A}}_2'(t) \end{bmatrix} \boldsymbol{x} = \begin{pmatrix} \tilde{c}_1 \\ \tilde{c}_2' \end{pmatrix}$$

This makes the system in general non-singular, having the possibility always to have a solution.

Tricks in the computation Recalling example 2.4, after writing the linear system we described it in a *unique* matrix containing both matrices $\mathbf{E}, \mathbf{A}$ and the right-hand side of the equation. This can be done for linear system, however we can note that the Gauss reduction is performed on the matrix $\mathbf{E}$ in order to obtain a form $\begin{bmatrix} \mathbf{I} & \\ & \mathbf{0} \end{bmatrix}$. The general idea is so to compute the Gauss steps on the system

$$\left[\begin{array}{c|c} \mathbf{E} & \mathbf{I} \end{array}\right] \xrightarrow{\text{Gauss reduction}} \left[\begin{array}{c|c} \mathbf{T}_m \dots \mathbf{T}_2\mathbf{T}_1\mathbf{E} & \mathbf{T}_m \dots \mathbf{T}_2\mathbf{T}_1\mathbf{I} \end{array}\right] = \left[\begin{array}{cc|c} \mathbf{I} & & \\ & \mathbf{0} & \mathbf{T} \end{array}\right]$$

With such transformation matrix $\mathbf{T}$ computing $\mathbf{TE}\boldsymbol{y}' + \mathbf{TA}\boldsymbol{y} = \mathbf{T}\boldsymbol{c}$ results in a separation of the differential part from the algebraic one and we can apply the algorithm (by deriving the algebraic part and iterate).

Example 2.5: simple pendulum and index reduction

Recalling the simple pendulum described at the start of this chapter, we retrieved the differential algebraic equation describing the system as

$$\begin{cases} x' = u \\ y' = v \\ my' + \lambda x = 0 \\ mv' + \lambda y = -mg \\ x^2 + y^2 - l^2 = 0 \end{cases}$$

Describing the time-dependent coordinates in the vector $\boldsymbol{z}' = (x, y, u, v, \lambda)$, the first thing to to is to re-state the problem in a form $\mathbf{E}(\boldsymbol{z}, t)\boldsymbol{z}' = \boldsymbol{G}(\boldsymbol{z}, t)$, so

$$\begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & m & & \\ & & & m & \\ & & & & 0 \end{bmatrix} \begin{pmatrix} x' \\ y' \\ u' \\ v' \\ \lambda' \end{pmatrix} = \begin{pmatrix} u \\ v \\ -\lambda x \\ -\lambda y - mg \\ l^2 - x^2 - y^2 \end{pmatrix}$$

Exploiting the yet-described trick we consider the linear system

$$\left[\begin{array}{ccccc|ccccc} 1 & & & & & 1 & & & & \\ & 1 & & & & & 1 & & & \\ & & m & & & & & 1 & & \\ & & & m & & & & & 1 & \\ & & & & 0 & & & & & 1 \end{array}\right]$$

The first step of the index reduction is quite simple and consists in the multiplication by $\frac{1}{m}$ of both the 3^{rd} and 4^{th} rows, resulting in

$$\left[\begin{array}{cc|c} \mathbf{I} & & \\ & \mathbf{0} & \mathbf{T}_1 \end{array}\right] = \left[\begin{array}{ccccc|ccccc} 1 & & & & & 1 & & & & \\ & 1 & & & & & 1 & & & \\ & & 1 & & & & & \frac{1}{m} & & \\ & & & 1 & & & & & \frac{1}{m} & \\ & & & & 0 & & & & & 1 \end{array}\right]$$

Applying the transformation matrix $\mathbf{T}$ on the initial system determines so the form

$$\begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 0 \end{bmatrix} \begin{pmatrix} x' \\ y' \\ u' \\ v' \\ \lambda' \end{pmatrix} = \mathbf{T}_1\boldsymbol{G} = \begin{pmatrix} u \\ v \\ -\frac{\lambda}{m}x \\ -\frac{\lambda}{m}y - g \\ l^2 - x^2 - y^2 \end{pmatrix} = \boldsymbol{G}_1$$

Deriving with respect to time the algebraic equation (that's only the last row) evaluates to $\frac{d}{dt}(l^2 - x^2 - y^2) = -2xx' - 2yy'$; this determines the new system

$$\begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ -2x & -2y & 0 & 0 & 0 \end{bmatrix} \begin{pmatrix} x' \\ y' \\ u' \\ v' \\ \lambda' \end{pmatrix} = \begin{pmatrix} u \\ v \\ -\frac{\lambda}{m}x \\ -\frac{\lambda}{m}y - g \\ 0 \end{pmatrix}$$

We can apply the same method to reduce this system, by so determining a second transformation matrix $\mathbf{T}_2$ for this system using the Gauss reduction:

$$\left[\begin{array}{ccccc|ccccc} 1 & & & & & 1 & & & & \\ & 1 & & & & & 1 & & & \\ & & 1 & & & & & 1 & & \\ & & & 1 & & & & & 1 & \\ -2x & -2y & 0 & 0 & 0 & & & & & 1 \end{array}\right]$$

$$\xrightarrow{(5)\mapsto(5)+2x(1)+2y(1)} \left[\begin{array}{ccccc|ccccc} 1 & & & & & 1 & & & & \\ & 1 & & & & & 1 & & & \\ & & 1 & & & & & 1 & & \\ & & & 1 & & & & & 1 & \\ 0 & 0 & 0 & 0 & 0 & 2x & 2y & 0 & 0 & 1 \end{array}\right]$$

This new transformation matrix $\mathbf{T}_2$ determines a new system

$$\begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 0 \end{bmatrix} \begin{pmatrix} x' \\ y' \\ u' \\ v' \\ \lambda' \end{pmatrix} = \mathbf{T}_2 \boldsymbol{G}_1 = \begin{pmatrix} u \\ v \\ -\frac{\lambda}{m}x \\ -\frac{\lambda}{m}y - g \\ 2xu + 2yv \end{pmatrix} = \boldsymbol{G}_2$$

Deriving the algebraic part $\frac{d}{dt}(2xu + 2yv) = 2x'u + 2xu' + 2y'v + 2yv'$ determines a new differential matrix $\mathbf{E}_3$ that can so be reduced using the Gauss method:

$$\left[\begin{array}{ccccc|ccccc} 1 & & & & & 1 & & & & \\ & 1 & & & & & 1 & & & \\ & & 1 & & & & & 1 & & \\ & & & 1 & & & & & 1 & \\ 2u & 2v & 2x & 2y & 0 & & & & & 1 \end{array}\right]$$

$$\xrightarrow{(5)\mapsto(5)-2u(1)-2v(1)-2x(3)-2y(4)} \left[\begin{array}{ccccc|ccccc} 1 & & & & & 1 & & & & \\ & 1 & & & & & 1 & & & \\ & & 1 & & & & & 1 & & \\ & & & 1 & & & & & 1 & \\ 0 & 0 & 0 & 0 & 0 & -2u & -2v & -2x & -2y & 1 \end{array}\right]$$

determining

$$\begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 0 \end{bmatrix} \begin{pmatrix} x' \\ y' \\ u' \\ v' \\ \lambda' \end{pmatrix} = \mathbf{T}_3 \boldsymbol{G}_2 = \begin{pmatrix} u \\ v \\ -\frac{\lambda}{m}x \\ -\frac{\lambda}{m}y - g \\ -2u^2 - 2v^2 + 2\frac{\lambda}{m}x^2 + 2\frac{\lambda}{m}y^2 + 2yg \end{pmatrix} = \boldsymbol{G}_3$$

Deriving one more time the lonely algebraic equation results in

$$\frac{d}{dt}\left(-2u^2-2v^2+2\frac{\lambda}{m}x^2+2\frac{\lambda}{m}y^2+2yg\right)=2\lambda'\frac{x^2+y^2}{m}+4\lambda\frac{xx'+yy'}{m}-4uu'-4vv'+2y'g$$

We so reduce the system

$$\left[\begin{array}{ccccc|ccccc} 1 & & & & & 1 & & & & \\ & 1 & & & & & 1 & & & \\ & & 1 & & & & & 1 & & \\ & & & 1 & & & & & 1 & \\ & & & & 1 & & & & & 1 \\ 4\frac{\lambda}{m}x & 4\frac{\lambda}{m}y+2g & -4u & -4v & 2\frac{x^2+y^2}{m} & & & & & \end{array}\right]$$

$$\xrightarrow{(5)\mapsto(5)-4\frac{\lambda}{m}x(1)-\left(4\frac{\lambda}{m}y+2g\right)(2)+4u(3)+4v(4)} \left[\begin{array}{ccccc|ccccc} 1 & & & & & 1 & & & & \\ & 1 & & & & & 1 & & & \\ & & 1 & & & & & 1 & & \\ & & & 1 & & & & & 1 & \\ 0 & 0 & 0 & 0 & 2\frac{x^2+y^2}{m} & -4\frac{\lambda}{m}x & -4\frac{\lambda}{m}y-2g & 4u & 4v & 1 \end{array}\right]$$

$$\xrightarrow{(5)\mapsto\frac{m}{2(x^2+y^2)}(5)} \left[\begin{array}{ccccc|ccccc} 1 & & & & & 1 & & & & \\ & 1 & & & & & 1 & & & \\ & & 1 & & & & & 1 & & \\ & & & 1 & & & & & 1 & \\ 0 & 0 & 0 & 0 & 1 & -\frac{2\lambda x}{x^2+y^2} & -\frac{2\lambda y}{x^2+y^2}-\frac{mg}{x^2+y^2} & \frac{2mu}{x^2+y^2} & \frac{2mv}{x^2+y^2} & \frac{m}{2(x^2+y^2)} \end{array}\right]$$

Finally we have a transform matrix **E** that's non-singular and the ordinary differential equation originated from the initial DAE is so

$$\begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix}\begin{pmatrix} x' \\ y' \\ u' \\ v' \\ \lambda' \end{pmatrix}=\mathbf{T}_4\boldsymbol{G}_3=\begin{pmatrix} u \\ v \\ -\frac{\lambda}{m}x \\ -\frac{\lambda}{m}y-g \\ \frac{-4\lambda(xy+yv)-3mgv}{x^2+y^2} \end{pmatrix}$$

2.2.1 Introduction of dummy variables

Considering the following differential algebraic equation retrieved by a fairly simple mechanical system that's in the form

$$\begin{cases} x_1'=u_1 \\ y_1'=v_1 \\ x_2'=u_2 \\ u_1'=2\lambda_1(x_1-x_2)+2\lambda_2x_1 \\ v_1'=2y_1(\lambda_1-\lambda_2)-y \\ u_2'=2\lambda_2(x_2-x_1) \\ x_1^2+y_1^2-1=0 \\ (x_1-x_2)^2+y_1^2-1=0 \end{cases} \tag{2.14}$$

The difficult part for manually solving this problems lies in the implicit differentiation of the right hand sides of the ODEs: the three ones are quite simple (in fact $\frac{d}{dt}x_1'=u_1'$) while the others are more complex ($\frac{d}{dt}u_2'=2\lambda_2'(x_2-x_1)+2\lambda_2(x_2'-x_1')$) and so is not in general a good idea to perform the index reduction on such system (because each steps requires the differentiation in time, increasing the complexity of the calculations).

The idea is so to introduce some ***dummy variables*** in the form $\dot{z}$ that allows to rewrite the second three ODEs as

$$u_1' = \dot{u}_1 \qquad v_1' = \dot{v}_1 \qquad u_2' = \dot{u}_2$$

subjected to the following algebraic constraints:

$$0 = 2\lambda_1(x_1 - x_2) + 2\lambda_2 x_1 - \dot{u}_1 \qquad 0 = 2y_1(\lambda_1 - \lambda_2) - y - \dot{v}_1 \qquad 0 = 2\lambda_2(x_2 - x_1) - \dot{u}_2$$

With this idea the initial differential algebraic equation can be regarded as

$$\begin{cases} \left.\begin{aligned} x_1' &= u_1 \\ y_1' &= v_1 \\ x_2' &= u_2 \\ u_1' &= \dot{u}_1 \\ v_1' &= \dot{v}_1 \\ u_2' &= \dot{u}_2 \end{aligned}\right\} & \text{: simpler ODE part} \\ \left.\begin{aligned} 0 &= 2\lambda_1(x_1 - x_2) + 2\lambda_2 x_1 - \dot{u}_1 \\ 0 &= 2y_1(\lambda_1 - \lambda_2) - y - \dot{v}_1 \\ 0 &= 2\lambda_2(x_2 - x_1) - \dot{u}_2 \end{aligned}\right\} & \text{: additional algebraic constraint} \\ \left.\begin{aligned} 0 &= x_1^2 + y_1^2 - 1 \\ 0 &= (x_1 - x_2)^2 + y_1^2 - 1 \end{aligned}\right\} & \text{: original algebraic constraint} \end{cases} \tag{2.15}$$

In order to reduce the index we can so differentiate in time the algebraic constraints: starting off with the newly added one what we obtain is (mathematical simplification are already performed)

$$\begin{aligned} \frac{d}{dt}\left(2\lambda_1(x_1 - x_2) + 2\lambda_2 x_1 - \dot{u}_1\right) &= 2\lambda_1'(x_1 - x_2) + 2\lambda_1(u_1 - u_2) + 2\lambda_2' x_1 + 2\lambda_2 u_1 + \dot{u}_1' \\ \frac{d}{dt}\left(2y_1(\lambda_1 - \lambda_2) - y - \dot{v}_1\right) &= 2v_1(\lambda_1 - \lambda_2) + 2y_1(\lambda_1' - \lambda_2') - \dot{v}_1' \\ \frac{d}{dt}\left(2\lambda_2(x_2 - x_1) - \dot{u}_2\right) &= 2\lambda_2'(x_2 - x_1) + 2\lambda_2(u_2 - u_1) - \dot{u}_2' \end{aligned}$$

In this case the differentiation already presents differential terms (in the variables $\dot{u}_1', \dot{v}_1', \dot{u}_2', \lambda_1', \lambda_2'$) and so *it doesn't make sense* to continue with the differentiation in time of this expressions. Differentiating instead the *original* algebraic constraints what we obtain is

$$\frac{d}{dt}\left(x_1^2 + y_1^2\right) = 2x_1 x_1' + 2y_1 y_1' = 2x_1 u_1 + 2y_1 v_1 \tag{a}$$

$$\frac{d}{dt}\left((x_1 - x_2)^2 + y_1^2 - 1\right) = 2(x_1 - x_2)(x_1' - x_2') + 2y_1 y_1' = 2(x_1 - x_2)(u_1 - u_2) + 2y_1 v_1 \tag{b}$$

This equations (after substituting all the variables $x_i' = u_i$) presents no differential part, hence we have to reduce one more time the index by differentiating only this two algebraic equations in time:

$$\frac{d}{dt}(a) = 2u_1^2 + 2x_1\dot{u}_1 + 2v_1^2 + 2y_1\dot{v}_1 \tag{c}$$

$$\frac{d}{dt}(b) = 2(u_1 - u_2)^2 + 2(x_1 - x_2)(\dot{u}_1 - \dot{u}_2) + 2v_1^2 + 2y_1\dot{v}_1 \tag{d}$$

We still need to reduce the index (because this expression are still algebraic) and after this more differentiation we finally obtain an ordinary differential equation:

$$\frac{d}{dt}(c) = 6u_1\dot{u}_1 + 6v_1\dot{v}_1 + 2x_1\dot{u}_1' + 2y_1\dot{v}_1'$$
$$\frac{d}{dt}(d) = 6(u_1-u_2)(\dot{u}_1-\dot{u}_2) + 6v_1\dot{v}_1 + 2(x_1-x_2)(\dot{u}_1'-\dot{u}_2') + 2y_1\dot{v}_1'$$

Ended the index reduction used to transform the algebraic constraints in ordinary differential part, we can rewrite such obtained ODEs in a matrix form as

$$\underbrace{\begin{bmatrix} 2(x_1-x_2) & 2x_1 & -1 & 0 & 0 \\ 2y_1 & -2y_1 & 0 & -1 & 0 \\ 0 & -2(x_1-x_2) & 0 & 0 & -1 \\ 0 & 0 & 2x_1 & 2y_1 & 0 \\ 0 & 0 & 2(x_1x_2) & 2y_1 & -2(x_1-x_2) \end{bmatrix}}_{\mathbf{E}} \underbrace{\begin{pmatrix} \lambda_1' \\ \lambda_2' \\ \dot{u}_1' \\ \dot{v}_1' \\ \dot{u}_2' \end{pmatrix}}_{\boldsymbol{z}'} = \underbrace{\begin{pmatrix} 2\lambda_2(u_2-u_1) - 2\lambda_2 u_1 \\ 2v_1(\lambda_2-\lambda_1) \\ 2\lambda_2(u_1-u_2) \\ -6u_1\dot{u}_1 - 6v_1\dot{v}_1 \\ -6(u_1-u_2)(\dot{u}_1-\dot{u}_2) - 6v_1\dot{v}_1 \end{pmatrix}}_{\boldsymbol{G}}$$

After all this steps we can so consider that the equivalent ordinary differential system of the initial DAE problem reported in equation 2.14 is the one determined as

$$\begin{cases} x_1' = u_1 \\ y_1' = v_1 \\ x_2' = u_2 \\ u_1' = \dot{u}_1 \\ v_1' = \dot{v}_1 \\ u_2' = \dot{u}_2 \\ \mathbf{E}\boldsymbol{z}' = \boldsymbol{G} \end{cases} \tag{2.16}$$

The initial condition of such ordinary differential equation satisfy the hidden *original* constraints

$$0 = x_1^2 + y_1^2 - 1 \qquad 0 = (x_1-x_2)^2 + y_1^2 - 1$$
$$0 = 2\lambda_1(x_1-x_2) + 2\lambda_2 x_1 - \dot{u}_1 \qquad 0 = 2y_1(\lambda_1-\lambda_2) - y - \dot{v}_1 \qquad 0 = 2\lambda_2(x_2-x_1) - \dot{u}_2$$

but also the expression retrieved while performing the index reduction on the system $(a), (b), (c), (d)$, so

$$2x_1u_1 + 2y_1v_1 = 0 \qquad 2(x_1-x_2)(u_1-u_2) + 2y_1v_1 = 0$$
$$2u_1^2 + 2x_1\dot{u}_1 + 2v_1^2 + 2y_1\dot{v}_1 = 0 \qquad 2(u_1-u_2)^2 + 2(x_1-x_2)(\dot{u}_1-\dot{u}_2) + 2v_1^2 + 2y_1\dot{v}_1 = 0$$

2.2.2 Kernel computation and index reduction

Given a differential algebraic equation $\mathbf{E}(t,\boldsymbol{y})\boldsymbol{y}' = \boldsymbol{G}(t,\boldsymbol{y})$, if the matrix $\mathbf{E}$ is singular then it means that exists at least one vector $\boldsymbol{v} \neq 0$ such that $\mathbf{E}\boldsymbol{v} = 0$. This operation is indeed the **kernel computation** of the matrix $\mathbf{E}$. We define the **kernel** of such matrix the sub-space described by the set

$$\ker\{\mathbf{E}\} = \{\boldsymbol{v} \text{ such that } \mathbf{E}\boldsymbol{v} = 0\} = \text{span}\{\boldsymbol{v}_1, \dots, \boldsymbol{v}_k\}$$

where the vectors $\boldsymbol{v}_1, \dots, \boldsymbol{v}_k$ are determining a basis of $\ker\{\mathbf{E}\}$. Computing the kernel of $\mathbf{E}^T$ evaluates to $\ker\{\mathbf{E}^T\} = \{\boldsymbol{w} \mid \boldsymbol{w}^T\mathbf{E} = 0\} = \text{span}\{\boldsymbol{w}_1^T, \dots, \boldsymbol{w}_p^T\}$. Building so the matrix $\mathbf{K} = \begin{bmatrix} \boldsymbol{w}_1 & \dots & \boldsymbol{w}_p \end{bmatrix}$ (considering all the vectors of the basis of $\ker\{\mathbf{E}^T\}$) as the *concatenations of the vectors* $\boldsymbol{w}_i$ implies that

$$\mathbf{K}^T\mathbf{E} = \begin{bmatrix} \boldsymbol{w}_1^T\mathbf{E} \\ \vdots \\ \boldsymbol{w}_p^T\mathbf{E} \end{bmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

The main idea is that if we are indeed able to compute such matrix $\mathbf{K}$ associated to the kernel of $\mathbf{E}^T$, then we have a way to easily compute the invariant of the differential algebraic equation, in fact

$$\mathbf{K}^T(t,\boldsymbol{y})\Big(\mathbf{E}(t,\boldsymbol{y})\boldsymbol{y}' = \boldsymbol{G}(t,\boldsymbol{y})\Big) \qquad \Rightarrow \qquad \mathbf{0} = \mathbf{K}^T(t,\boldsymbol{y})\boldsymbol{G}(t,\boldsymbol{y})$$

In general $\mathbf{K} \in \mathbb{R}^{n\times p}$ is a rectangular matrix (where $p < n$) and so if we can determine a matrix $\mathbf{L} \in \mathbb{R}^{n\times(p-n)}$ such that $\mathbf{M} = \begin{bmatrix}\mathbf{K} & \mathbf{L}\end{bmatrix} \in \mathbb{R}^{n\times n}$ is non-singular, then we can observe that we can separate the algebraic part of the DAE from the differential part, in fact

$$\begin{aligned}&\mathbf{M}^T\Big(\mathbf{E}\boldsymbol{y}' = \boldsymbol{G}\Big)\\ &\mathbf{M}^T\mathbf{E}\boldsymbol{y}' = \mathbf{M}^T\boldsymbol{G} \qquad \Rightarrow \begin{cases}\cancel{\mathbf{K}^T\mathbf{E}\boldsymbol{y}'} = \mathbf{K}^T\boldsymbol{G}\\ \mathbf{L}^T\mathbf{E}\boldsymbol{y}' = \mathbf{L}^T\boldsymbol{G}\end{cases}\end{aligned}$$

that can be reduced to a form

$$\begin{cases}\tilde{\mathbf{E}}_1\boldsymbol{y}' = \tilde{\boldsymbol{G}}_1 & \text{differential part}\\ 0 = \tilde{\boldsymbol{G}}_2 & \text{algebraic part}\end{cases} \tag{2.17}$$

We have so separated the algebraic part from the differential one, so we can differentiate this second one in order to reduce the index and obtain a system in the form

$$\begin{cases}\tilde{\mathbf{E}}_1\boldsymbol{y}' = \tilde{\boldsymbol{G}}_1\\ \tilde{\mathbf{E}}_1\boldsymbol{y}' = \tilde{\boldsymbol{G}}_1 - \frac{\partial\tilde{\boldsymbol{G}}_2}{\partial\boldsymbol{y}}\boldsymbol{y}' = \frac{\partial\tilde{\boldsymbol{G}}_2}{\partial t}\end{cases}$$

This newly constructed differential algebraic problem can be reduce following the same procedure until when the kernel of $\mathbf{E}^T$ has a null dimension (it is composed only by the zero vector).

Computation of the kernel With this premise being said, we have to find a way to compute the kernel of the square matrix $\mathbf{E}$; this can be achieved, as example, using the LU decomposition. Considering in fact the decomposition $\mathbf{PEQ} = \mathbf{LU}$ (where $\mathbf{P}, \mathbf{Q}$ are permutation matrices), as was previously discussed, can be inverted to a form $\mathbf{E} = \mathbf{P}^{-1}\mathbf{LUQ}^{-1} = \mathbf{P}^{-1}\mathbf{L}M$ where $\mathbf{M}$ can has the *upper rectangle* that's non-null and the lower one that's null (so is in the form $\mathbf{M} = \begin{bmatrix}\mathbf{M}_1\\ 0\end{bmatrix}$); knowing that $\mathbf{L}$ is non-singular (for the definition of the LU decomposition), then also $\mathbf{P}^{-1}\mathbf{L}$ is nonsingular and so we can build the matrix

$$\mathbf{M} = \mathbf{L}^{-1}\mathbf{PE} = \begin{bmatrix}\mathbf{M}_1\\ 0\end{bmatrix}$$

Multiplying by appropriately matrices of the form $\begin{bmatrix}\mathcal{I} & 0\end{bmatrix}$ and $\begin{bmatrix}0 & \mathcal{I}\end{bmatrix}$ this expressions we can observe that

$$\begin{aligned}\underbrace{\begin{bmatrix}\mathcal{I} & 0\end{bmatrix}\mathbf{L}^{-1}\mathbf{P}}_{=\mathbf{R}^T}\mathbf{E} &= \begin{bmatrix}\mathcal{I} & 0\end{bmatrix}\begin{bmatrix}\mathbf{M}_1\\ 0\end{bmatrix} = \mathbf{M}_1 \neq 0\\ \underbrace{\begin{bmatrix}0 & \mathcal{I}\end{bmatrix}\mathbf{L}^{-1}\mathbf{P}}_{=\mathbf{K}^T}\mathbf{E} &= \begin{bmatrix}0 & \mathcal{I}\end{bmatrix}\begin{bmatrix}\mathbf{M}_1\\ 0\end{bmatrix} = 0\end{aligned}$$

This method allowed us to compute the required matrices as $\mathbf{K} = \mathbf{P}^T\mathbf{L}^{-T}\begin{bmatrix}0\\ \mathcal{I}\end{bmatrix}$ and $\mathbf{R} = \mathbf{P}^T\mathbf{L}^{-T}\begin{bmatrix}\mathbf{I}\\ 0\end{bmatrix}$ that *stacked* determined the non-singular matrix $\begin{bmatrix}\mathbf{K} & \mathbf{R}\end{bmatrix} = \mathbf{P}^T\mathbf{L}^{-T}\begin{bmatrix}0 & \mathcal{I}\\ \mathcal{I} & 0\end{bmatrix}$.

2.3 Semi-explicit form

Given a differential algebraic equation in the form $\boldsymbol{F}(\boldsymbol{z}, \boldsymbol{z}', t) = 0$ that can be expressed as

$$\begin{cases} \boldsymbol{x}' &= \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}, t) \\ 0 &= \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{y}, t) \end{cases}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are respectively the differential and algebraic variables of the problem and are such that $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y})$. Let us so consider $x \subseteq z$ the variables in $\boldsymbol{F}(\boldsymbol{z}, \boldsymbol{z}', t)$ that are appearing as derivative, then we can defined $\boldsymbol{x}'$ as $\dot{\boldsymbol{x}}$ thus

$$\boldsymbol{F}(\boldsymbol{z}, \boldsymbol{z}', t) = \boldsymbol{F}(\boldsymbol{z}, \dot{\boldsymbol{x}}, t) = \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{y}, \dot{\boldsymbol{x}}, t) = 0$$

This can be rewritten as

$$\begin{cases} \boldsymbol{x}' &= \dot{\boldsymbol{x}} \\ 0 &= \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{y}, \dot{\boldsymbol{x}}, t) \end{cases}$$

Moreover calling $\boldsymbol{w} = (\dot{\boldsymbol{x}}, \boldsymbol{y})$ what we obtain is a **differential algebraic equation** in the so called **semi-explicit form**:

$$\begin{cases} \boldsymbol{x} &= \dot{\boldsymbol{x}} = \boldsymbol{H}(\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{t}) \\ 0 &= \boldsymbol{G}(\boldsymbol{x}, \boldsymbol{w}, t) \end{cases} \tag{2.18}$$

Semi-explicit DAE of index 1 Considering a DAE in the form

$$\begin{cases} \boldsymbol{x}' &= \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}, t) \\ 0 &= \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{y}, t) \end{cases}$$

then if it is possible to solve $\boldsymbol{g}$ to obtain $\boldsymbol{y}$ so in the form $\boldsymbol{y} = \boldsymbol{H}(\boldsymbol{x}, t)$, then what we have is that $\boldsymbol{g}\big(\boldsymbol{x}, \boldsymbol{H}(\boldsymbol{x}, t), t\big) = 0$ for all values $\boldsymbol{x}, t$, and so we can consider the DAE as

$$\begin{cases} \boldsymbol{x}' &= \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}, t) \\ \boldsymbol{y} &= \boldsymbol{H}(\boldsymbol{x}, t) \end{cases}$$

Differentiating the second equation (the algebraic part) in time determines

$$\boldsymbol{y}' = \frac{d\boldsymbol{H}(\boldsymbol{x}, t)}{dt} = \frac{\partial \boldsymbol{H}(\boldsymbol{x}, t)}{\partial \boldsymbol{x}} \boldsymbol{x}' + \frac{\partial \boldsymbol{H}(\boldsymbol{x}, t)}{\partial t} = \frac{\partial \boldsymbol{H}(\boldsymbol{x}, t)}{\partial \boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}, t) + \frac{\partial \boldsymbol{H}(\boldsymbol{x}, t)}{\partial t} = \boldsymbol{L}(\boldsymbol{x}, \boldsymbol{y}, t)$$

and so the differential algebraic equation after one differentiation (so with a index reduction) becomes

$$\begin{cases} \boldsymbol{x}' &= \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}, t) \\ \boldsymbol{y}' &= \boldsymbol{L}(\boldsymbol{x}, \boldsymbol{y}, t) \end{cases}$$

In general an ODE in semi-explicit form of index 1 can be solved without performing the derivation by simply considering

$$\boldsymbol{x}' = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}, t) = \boldsymbol{f}\big(\boldsymbol{x}, \boldsymbol{H}(\boldsymbol{x}, t), t\big) = \tilde{\boldsymbol{f}}(\boldsymbol{x}, t) \tag{2.19}$$

2.3.1 Implicit function theorem

Equation 2.19 relies on the fact that from $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{y}, t) = 0$ allows us to extract the map $\boldsymbol{H}(\boldsymbol{x}, t)$ that allows to explicitly compute $\boldsymbol{y}$ (so satisfying $\boldsymbol{g}\big(\boldsymbol{x}, \boldsymbol{H}(\boldsymbol{x}, t), t\big) = 0$), however up to now we cannot be sure that such map $\boldsymbol{H}$ really exists.

Simplified version Considering a function $f(x,y) \in \mathcal{C}^1(\mathbb{R}^2)$ and a point (x_0, y_0) such that $f(x_0, y_0) = 0$ and $\frac{\partial f}{\partial y}(x_0, y_0) \neq 0$, then there exists an open interval $(x_0 - \delta, x_0 + \delta)$ such that

$$y = H(x) \qquad \text{and} \qquad f\big(x, H(x)\big) = 0 \quad \forall x \in (x_0 - \delta, x_0 + \delta)$$

Moreover we have that

$$\frac{dH(x)}{dx} = -\left(\frac{\partial f(x,y)}{\partial y}\right)^{-1} \frac{\partial f(x,y)}{\partial x} \tag{2.20}$$

Considering the simple example of the function $f(x,y) = y^2 - x$, the point $(x_0, y_0) = (1,1)$ satisfies the conditions of the implicit function theorem, in fact

$$f(x_0, y_0) = 0 \qquad \text{and} \qquad \left.\frac{\partial f}{\partial y}\right|_{x=x_0, y=y_0} = \left. 2y \right|_{x=1,y=1} = 2 \neq 0$$

then we are sure that there exists a matrix H that allows to express y as function of x, in fact rewriting $y^2 = x$ and so $y = \pm\sqrt{x}$. The choice of the sign strictly depends on the value of the point (x_0, y_0): in this case we have that $y = H(x) = \sqrt{x}$. In contrary, if we would have chosen the initial point $(x_0, y_0) = (1, -1)$ the condition for the theorem would have been still verified but the resulting map would have been $y = -\sqrt{x}$.
Choosing instead the point $(x_0, y_0) = (0,0)$ would have resulted in the violation of the condition $\frac{\partial f}{\partial y} \neq 0$, thus the theorem cannot be applied: in such point we can't in fact discriminate the sign of the map $y = \pm\sqrt{x}$. However by inverting x and y in the definition, we can define a map $H(y)$ that determines x, in fact

$$f(0,0) = 0 \qquad \text{and} \qquad \left.\frac{\partial f}{\partial x}\right|_{x=x_0, y=y_0} = 1 \neq 0$$

and so $x = H(y) = y^2$.

Implicit function theorem Given a function $\boldsymbol{f} : \mathcal{A} \subseteq \mathbb{R}^{n+m} \to \mathbb{R}^m$ (where n, m are respectively the number of *independent* and *dependent* variables) and a point $(\boldsymbol{x}_0, \boldsymbol{y}_0) \in \mathcal{A}$ (with $\boldsymbol{x}_0 \in \mathbb{R}^n, \boldsymbol{y}_0 \in \mathbb{R}^m$) such that $\boldsymbol{f}(\boldsymbol{x}_0, \boldsymbol{y}_0) = 0$ and the matrix $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{y}}(\boldsymbol{x}_0, \boldsymbol{y}_0)$ is non-singular, then there exists two open sets $\mathcal{U} \subseteq \mathbb{R}^n, \mathcal{V} \subseteq \mathbb{R}^m$ on top of which is defined a map $\boldsymbol{\phi} : \mathcal{U} \to \mathcal{V}$, $\boldsymbol{\phi} \in \mathcal{C}^1(\mathcal{U}, \mathcal{V})$, such that

$$\boldsymbol{y}_0 = \boldsymbol{\phi}(\boldsymbol{x}_0) \qquad \Leftrightarrow \qquad \boldsymbol{f}\big(\boldsymbol{x}_0, \boldsymbol{\phi}(\boldsymbol{x}_0)\big) = 0 \quad \forall \boldsymbol{x} \in \mathcal{U}, \boldsymbol{\phi}(\boldsymbol{x}) \in \mathcal{V}$$

Moreover

$$\frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{x}} = -\left[\frac{\partial \boldsymbol{f}\big(\boldsymbol{x}, \boldsymbol{\phi}(\boldsymbol{x})\big)}{\partial \boldsymbol{x}}\right]^{-1} \frac{\partial \boldsymbol{f}\big(\boldsymbol{x}, \boldsymbol{\phi}(\boldsymbol{x})\big)}{\partial \boldsymbol{x}} \tag{2.21}$$

General form The same theorem can be stated similarly considering an initial function $\boldsymbol{\phi}(\boldsymbol{z}) : \mathbb{R}^n \to \mathbb{R}^m$ where $n > m$. The vector $\boldsymbol{z} = (x_1, x_2, \ldots, x_n)$ can be partitioned, upon reordering of the variables, in 2 sets $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y})$ (where $\boldsymbol{x} \in \mathbb{R}^{n-m}$ contains the independent variables while $\boldsymbol{y} \in \mathbb{R}^m$ the dependent one) in such a way that the matrix $\frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{y}}(\boldsymbol{x}_0, \boldsymbol{y}_0)$ is non-singular: this so allows to use the implicit function theorem to determine a map $\boldsymbol{\psi}$ such that $\boldsymbol{y} = \boldsymbol{\psi}(\boldsymbol{x})$.

We can observe in fact that if the matrix $\frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{z}} \in \mathbb{R}^{m \times n}$ is full rank (hence all rows of the jacobian are linearly independent), then there exists m linearly independent columns while the remaining $n - m$ are linearly dependent from the others: reordering so the independent/dependent variables allows us to distinguish $\boldsymbol{x}$ from $\boldsymbol{y}$.
For this reason the map $\boldsymbol{\psi}$ is in general not unique because there exists multiple partitioned sets $(\boldsymbol{x}, \boldsymbol{y})$ determining a non-singular jacobian.

Semi-explicit DAE Recalling a differential algebraic equation in semi-explicit form

$$\begin{cases} \boldsymbol{x}' &= \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}, t) \\ 0 &= \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{y}, t) \end{cases}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ are the so called *differential states*, $\boldsymbol{y} \in \mathbb{R}^m$ the *algebraic states* and the maps are $\boldsymbol{f} :\in \mathbb{R}^{n+m+1} \to \mathbb{R}^n$ and $\boldsymbol{g} :\in \mathbb{R}^{n+m+1} \to \mathbb{R}^m$. If the matrix $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{y}} \in \mathbb{R}^{m \times m}$ is non-singular, then for the implicit function theorem we can define the independent and dependent variables respectively as $\boldsymbol{z}^i = (\boldsymbol{x}, t), \boldsymbol{z}^d = \boldsymbol{y}$ such that exists a map $\boldsymbol{\psi}$ for which $\boldsymbol{y} = \boldsymbol{\psi}(\boldsymbol{x}, t)$ and

$$\boldsymbol{y}' = \frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{x}} \boldsymbol{x}' + \frac{\partial \boldsymbol{\psi}}{\partial t} = \frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{x}} f + \frac{\partial \boldsymbol{\psi}}{\partial t} = \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{y}, t)$$

This means that after 1 derivation the initial DAE in explicit form can be regarded as following ODE:

$$\begin{cases} \boldsymbol{x}' &= \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}, t) \\ \boldsymbol{y}' &= \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{y}, t) \end{cases} \quad + \quad \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{y}, t) = 0 : \text{hidden invariant}$$

Part II

Modelling & Simulation