Numerical methods for dynamical systems

Alexandre Chapoutot

ENSTA Paris
master CPS IP Paris

2020-2021

Part VI

Numerical methods for IVP-DAE

Part 6. Section 1
Introduction fo Differential Algebraic Equations

We consider a differential system of equation

$$F(\dot{x}, x, t) = \begin{pmatrix} F_1(\dot{x}(t), x(t), t) \\ F_2(\dot{x}(t), x(t), t) \\ \vdots \\ F_n(\dot{x}(t), x(t), t) \end{pmatrix} = 0$$

with $\dot{x}(t), x(t) \in \mathbb{R}^n$.

This system is a **DAE** if the Jacobian matrix

$$\frac{\partial F}{\partial \dot{x}} \quad \text{is } \textbf{singular}$$

The following system is a DAE

$$\begin{aligned} x_1 - \dot{x}_1 + 1 = 0 \\ \dot{x}_1 x_2 + 2 = 0 \end{aligned} \quad \Rightarrow \quad F(\dot{x}, x, t) = \begin{pmatrix} x_1 - \dot{x}_1 + 1 \\ \dot{x}_1 x_2 + 2 \end{pmatrix} \quad \text{with} \quad x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

The Jacobian of F w.r.t. $\dot{x}$ is

$$\frac{\partial F}{\partial \dot{x}} = \begin{pmatrix} \frac{\partial F_1}{\partial \dot{x}_1} & \frac{\partial F_1}{\partial \dot{x}_2} \\ \frac{\partial F_2}{\partial \dot{x}_1} & \frac{\partial F_2}{\partial \dot{x}_2} \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ x_2 & 0 \end{pmatrix} \quad \Rightarrow \quad \det\left(\frac{\partial F}{\partial \dot{x}}\right) = 0$$

Note in this example $\dot{x}_2$ is not explicitly defined.

Solving DAE is a hard challenge either symbolically or numerically.

Special DAE forms are usually considered: linear, Hessenberg form, etc.

Example, we rewrite the previous system

- solving for $\dot{x}_1$ the equation $x_1 - \dot{x}_1 + 1 = 0 \Rightarrow \dot{x}_1 = x_1 + 1$
- Substitute $\dot{x}_1$ in $\dot{x}_1 x_2 + 2 = 0$ we get

$$\begin{aligned} \dot{x}_1 &= x_1 + 1 \quad \texttt{Ordinary differential equation} \\ (x_1 + 1)\, x_2 + 2 &= 0 \quad \texttt{Algebraic equation} \end{aligned}$$

Note: this form of DAE is used in many engineering applications.

- mechanical engineering, process engineering, electrical engineering, etc.
- **Usually:** dynamics of the process + laws of conservation

Engineering examples of DAE - Chemical reaction

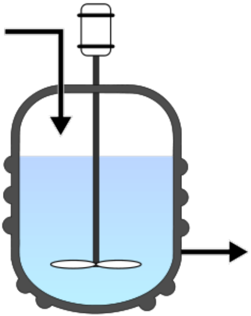
An isothermal continuous flow stirred-tank reactor[1] (CSTR) with elementary reactions:

$$A \rightleftharpoons B \rightarrow C$$

assuming

- reactant A with a in-flow rate F_a and concentration C_{A_0}
- Reversible reaction $A \rightleftharpoons B$ is much faster that $B \rightarrow C$, i.e., $k_1 \gg k_2$

$$\begin{aligned}
\dot{V} &= F_a - F \\
\dot{C}_A &= \frac{F_a}{V}\left(C_{A_0} - C_A\right) - R_1 \\
\dot{C}_B &= -\frac{F_a}{V} C_B + R_1 - R_2 \\
\dot{C}_C &= -\frac{F_a}{C}{}_C + R_2 \\
0 &= C_A - \frac{C_B}{K_{eq}} \\
0 &= R_2 - k_2 C_B
\end{aligned}$$



[1] Control of Nonlinear DAE Systems with Applications to Chemical Processes

- R_1 and R_2 rates of reactions
- F output flow
- C_A, C_B and C_C are concentrations of A, B and C.

Let

$$\begin{aligned} x &= (V, C_A, C_B, C_C) \\ z &= (R_1, R_2) \end{aligned}$$

we get

$$\begin{aligned} \dot{x} &= f(x, z) \\ 0 &= g(x, z) \end{aligned}$$

Pendulum

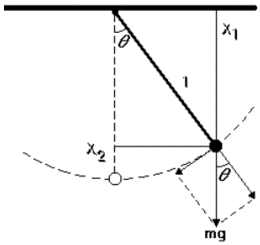
- second Newton's law

$$m\ddot{x} = -\frac{F}{\ell}x$$
$$m\ddot{y} = mg\frac{F}{\ell}y$$

- Mechanical energy conservation

$$x^2 + y^2 = \ell^2$$

$$\dot{x}_1 = x_3$$
$$\dot{x}_2 = x_4$$
$$\dot{x}_3 = -\frac{F}{\ell}x_1$$
$$\dot{x}_4 = g\frac{F}{\ell}x_2$$
$$0 = x_1^2 + x_2^2 - \ell^2$$



Ohm's law

$$C\dot{V}_C = i_C, \quad L\dot{V} = i_L, V_R = Ri_R$$

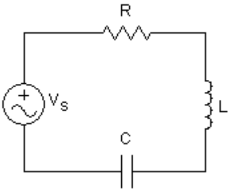
Kirchoff's voltage and current laws

- Conservation of current

$$i_E = i_R, \quad i_R = i_C, \quad i_C = i_L$$

- Conservation of energy

$$V_R + V_L + V_C + V_E = 0$$



And we get

$$\begin{aligned} \dot{V}_C &= \frac{1}{C} i_L \\ \dot{V}_L &= \frac{1}{L} i_L \\ 0 &= V_R + Ri_E \\ 0 &= V_E + V_R + V_C + V_L \\ 0 &= i_L - i_E \end{aligned}$$

Let

$$x = (V_C, V_L, V_R, i_L, i_E)$$

we have

$$\dot{x} = \begin{pmatrix} \frac{1}{C} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{L} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} x$$

$$0 = \begin{pmatrix} 0 & 0 & 1 & 0 & R \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} V_E$$

which is of the form

$$\dot{x} = Ax$$
$$0 = Bx + Dz$$

Consider the linear PDE (diffusion equations)

$$\frac{\partial u}{\partial t}(x,t) = D\frac{\partial^2 u}{\partial x^2}(x,t) \quad \text{with} \quad \begin{cases} u(x = x_0, t) = u_b \\ \dfrac{\partial u}{\partial x}(x = x_f, t) = 0 \end{cases}$$

and D a constant.

Using method of lines, we have with an equally spaced grid for x (finite difference)

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2} + \mathcal{O}\left(\Delta x^2\right)$$

Hence, we get

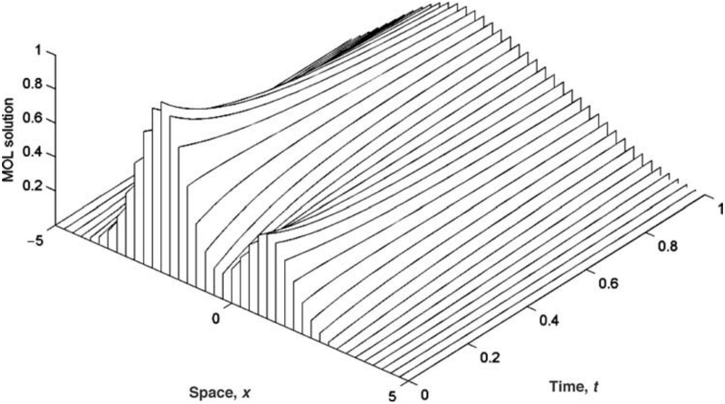
$$\frac{du_i}{dt} = D\frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2} \quad \text{for} \quad i = 1, 2, \ldots, M$$

In other words, we get the system

$$
\begin{aligned}
u_1 &= u_b \\
\frac{du_2}{dt} &= D\frac{u_3 - 2u_2 + u_b}{\Delta x^2} \\
\frac{du_3}{dt} &= D\frac{u_4 - 2u_3 + u_2}{\Delta x^2} \\
&\vdots \\
\frac{du_M}{dt} &= D\frac{u_{M+1} - 2u_M + u_{M-1}}{\Delta x^2} \\
u_{M+1} &= u_M
\end{aligned}
$$

Note u_{M+1} is outside of the grid so we add an extra constraints.

Hence we get a **DAE**


MOL solution
1
0.8
0.6
0.4
0.2
–5
0
5
Space, x
0
0.2
0.4
0.6
0.8
1
Time, t

- **Nonlinear DAE** if it is of the form

$$F(\dot{x}, x, t) = 0$$

  and it is nonlinear w.r.t. any one of $\dot{x}$, x, or t

- **Linear DAE** if it is of the form

$$A(t)\dot{x} + B(t)x = c(t)$$

  If $A(t) \equiv A$ and $B(t) \equiv B$ then the DAE is **time-invariant**

- **Semi-explicit DAE** it is of the form

$$\begin{aligned} \dot{x} &= f(t, x, z) \\ 0 &= g(t, x, z) \end{aligned}$$

  z is the **algebraic variable** and x is a **differential/state variable**

- **Fully implicit DAE** it is of the form

$$F(\dot{x}, x, t) = 0$$

Note any DAE can be written in a semi-explicit form.

Conversion of fully implicit form

$$F(\dot{x}, x, t) = 0 \quad \overset{\dot{x}=z}{\Leftrightarrow} \begin{cases} \dot{x} = z \\ 0 = F(z, x, t) \end{cases}$$

Remark this transformation does not make the solution more easier to get

But useful in case of linear DAE, see next.

Consider a linear time-invariant DAE

$$A\dot{x} + Bx + b(t) = 0$$

assuming that $\lambda A + B$ (**matrix pencil**) is not singular for some scalar λ.

Then it exists non-singular matrices G and H of size $n \times n$ such that:

$$GAH = \begin{pmatrix} I_m & 0 \\ 0 & N \end{pmatrix} \quad \text{and} \quad GBH = \begin{pmatrix} J & 0 \\ 0 & I_{n-m} \end{pmatrix}$$

- I_m is the identity matrix of size $m \times m$ ($m \leq n$)
- I_{n-m} is the identity matrix of size $(n-m) \times (n-m)$
- N is a **nilpotent matrix**, i.e., $\exists p \in \mathbb{N}^+, N^p = 0$
- $J \in \mathbb{R}^{m \times m}$

Classification of DAE - cont

Hence

$$
\begin{aligned}
A\dot{x} + Bx + b(t) = 0 &\Leftrightarrow (GAH)(H^{-1})\dot{x} + (GBH)(H^{-1})x + Gb(t) = 0 \\
&\Leftrightarrow \begin{pmatrix} I_m & 0 \\ 0 & N \end{pmatrix} H^{-1}\dot{x} + \begin{pmatrix} J & 0 \\ 0 & I_{n-m} \end{pmatrix} H^{-1}x + Gb(t) = 0 \\
&\Leftrightarrow \quad \text{with} \quad w(t) = H^{-1}x \\
&\begin{pmatrix} I_m & 0 \\ 0 & N \end{pmatrix} \dot{w} + \begin{pmatrix} J & 0 \\ 0 & I_{n-m} \end{pmatrix} w + Gb(t) = 0
\end{aligned}
$$

Let $w = (w_1, w_2)^T$ with $w_1 \in \mathbb{R}^m$ and $w_2 \in \mathbb{R}^{n-m}$, $b = (b_1, b_2)^T$ we get

$$
\begin{aligned}
\dot{w}_1 + Jw_1 + b_1(t) &= 0 \\
Nw_1 + w_2 + b_2(t) &= 0
\end{aligned}
$$

From Nilpotency property, we get

$$
\begin{aligned}
\dot{w}_1 &= -Jw_1 - b_1(t) \\
0 &= -(N^p)^{-1}w_2 - (N^p)^{-1}b_2(t)
\end{aligned}
$$

Part 6. Section 2
Notion of index for DAE

Remark

There are several definitions of an index.
Each measure a different aspect of the DAE.

- **Differential index** (δ) measure the degree of singularity.
- **Perturbation index** (π) measure the influence of numerical approximation.
- etc.

Definition of differential index

The index of a DAE system $F(\dot{x}, x, t) = 0$ is the minimum number of times certain equations in the DAE must be differentiated w.r.t. t, in order to transform the problem into an ODE.

Remark: (differential) index can be seen as a measure of the distance between the DAE and the corresponding ODE.

Remark: mathematical properties are lost with differentiation!

Definition of index

The differential index k of a sufficiently smooth DAE is the smallest k such that:

$$
\begin{aligned}
F(\dot{x}, x, t) &= 0 \\
\frac{\partial F}{\partial t}(\dot{x}, x, t) &= 0 \\
&\vdots \\
\frac{\partial^k F}{\partial t^k}(\dot{x}, x, t) &= 0
\end{aligned}
$$

uniquely determines $\dot{\mathbf{x}}$ as a continuous function of $(\mathbf{x}, t)$.

Let

$$\dot{x}_1 = x_1 + 1$$
$$(x_1 + 1)\, x_2 + 2 = 0$$

with x_2 the algebraic variable.
Differentiation of g w.r.t. t,

$$\frac{d}{dt} g(x_1, x_2) = 0 \quad \Rightarrow \quad \dot{x}_1 x_2 + (x_1 + 1)\dot{x}_2 = 0 \quad \Rightarrow \quad \dot{x}_2 = -\frac{\dot{x}_1 x_2}{x1_1} = -x_2$$

Only one differentiation is needed to define $\dot{x}_2$, this DAE is **index** 1

Other examples,

- CSTR is index 2
- Pendulum is index 3

There are **higher index** DAEs (index > 1)

Index reduction is used to go from higher index to lower index DAE (cf Khalil Ghorbal's lecture)

DAE family and differential index

Index 0

ODE system $\dot{x} = f\left(t, x(t)\right)$

Index 1

Algebraic equation $y = q(t)$

Index 1

DAE in Hessenberg form of index 1

$$\begin{aligned} \dot{x} &= f\left(t, x, y\right) \\ 0 &= g\left(x, y\right) \quad \text{with} \quad \frac{\partial g}{\partial y} \quad \text{is non-singular} \end{aligned}$$

Index 2

DAE in Hessenberg form of index 2

$$\dot{x} = f(t, x, y)$$

$$0 = g(t, x) \quad \text{with} \quad \frac{\partial g}{\partial x}\frac{\partial f}{\partial y} \quad \text{is non-singular}$$

Index 3

DAE in Hessenberg form of index 3

$$\dot{x} = f(t, x, y, z)$$

$$\dot{y} = g(t, x, y)$$

$$0 = h(t, y) \quad \text{with} \quad \frac{\partial h}{\partial y}\frac{\partial g}{\partial x}\frac{\partial f}{\partial z} \quad \text{is non-singular}$$

e.g., mechanical systems

The DAE has the perturbation index k along a solution x if k is the smallest integer such that,

for all functions $x(t)$ having the defect

$$f(\dot{x}_\delta, x_\delta, t) = \delta(t)$$

there exists an estimate

$$\| x(t) - x_\delta(t) \| \leq C \Big(\| x(t_0) - x_\delta(t_0) \| + \max_t \| \delta(t) \| + max_t \| \delta'(t) \| \\ + \cdots + \max_t \| \delta^{(k-1)}(t) \| \Big)$$

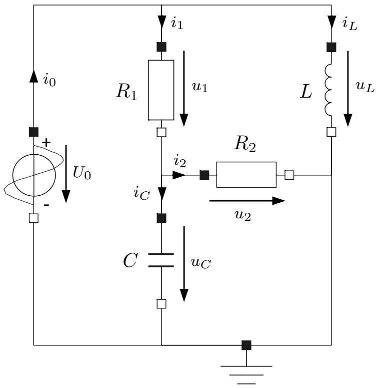
for a constant $C > 0$, if δ is small enough.

Property:

$$\delta \leq \pi \leq \delta + 1$$

Part 6. Section 3
Index reduction

RLC circuit



$$u_0 = f(t) \tag{1}$$
$$u_1 = R_1 i_1 \tag{2}$$
$$u_2 = R_2 i_2 \tag{3}$$
$$u_L = L\frac{di_L}{dt} \tag{4}$$
$$i_C = C\frac{du_C}{dt} \tag{5}$$
$$u_0 = u_1 + u_C \tag{6}$$
$$u_L = u_1 + u_2 \tag{7}$$
$$u_C = u_2 \tag{8}$$
$$i_0 = i_1 + i_L \tag{9}$$
$$i_1 = i_2 + i_C \tag{10}$$

We want to compute a state-space form of this RLC circuit.

$$
\begin{array}{c c}
 & \begin{array}{cccccccccc} u_0 & i_0 & u_1 & i_1 & u_2 & i_2 & u_L & \frac{di_L}{dt} & \frac{du_C}{dt} & i_C \end{array} \\
\begin{array}{c} \text{Eq. (1)} \\ \text{Eq. (2)} \\ \text{Eq. (3)} \\ \text{Eq. (4)} \\ \text{Eq. (5)} \\ \text{Eq. (6)} \\ \text{Eq. (7)} \\ \text{Eq. (8)} \\ \text{Eq. (9)} \\ \text{Eq. (10)} \end{array} &
\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1
\end{pmatrix}
\end{array}
$$

Structure incidence matrix

Relation between equations (rows) and unknowns (columns)

- if the i-th equation contains the j-th variable then the matrix coefficient (i, j) contains 1 and 0 otherwise.

By default all equations are implicit (or **acausal**)

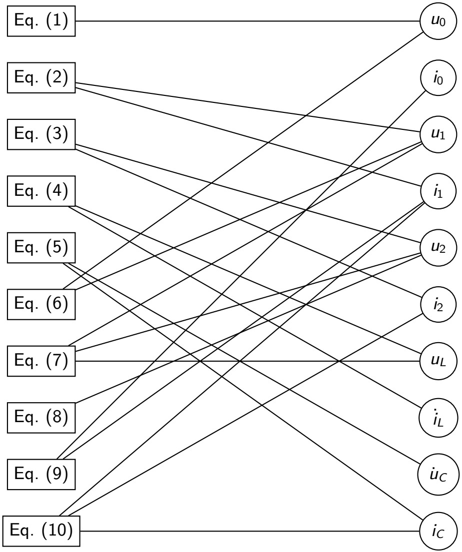
Two rules to choose the set of variables to solve

- if an equations contains only a single unknown then we need that variable to solve it (i.e., this equation is **causal**, e.g., Eq. (1))
- If an unknown only appears in one equation, that equation must use to solve it. E.g., Eq. (9) i_0 only appears in that equation.

Apply iteratively these rules:

- if a row only contains one 1, that equation needs to be solved for that variable so eliminate both row and column
- if a column only contains one 1, that variable needs to be solved for that equation so eliminate both row and column

Structure digraph



Remark the number of equations must always equal to the number of variables.

Building: There is a link between a node of equations and a node of variable is this variable appears in that equation.

Finding which variable needs to be solved from which equations, is based on a graph coloring algorithm (Tarjan)

- When a variable is selected to be solved from an equation the link between them is colored in red.
- When a variable is known or when the equation in which it occurs is being used to solve an other variable, the link is colored in blue

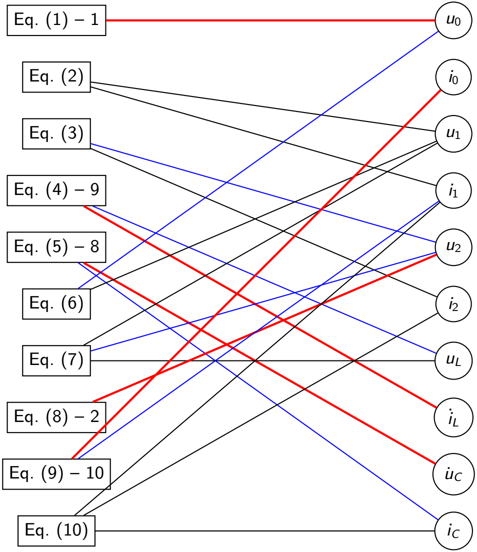
- A causal equation has exactly one red link connected to it
- An acausal equation has block or blue connected edges
- A known variable has exactly one red input edge
- An unknown variable has only black or blue input edges
- No equation or variable has more than one red edges

Rules to find variables and equations

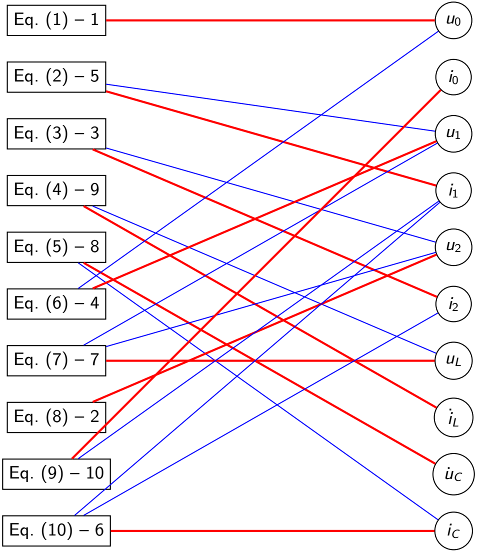
- For all acausal equations, if an equation has only one black line attached to it, color that line red, follow it to the variable it points at, and color all other connections ending in that variable in blue. Renumber the equation using the lowest free number starting from 1.
- For all unknown variables, if a variable has only one black line attached to it, color that line red, follow it back to the equation it points at, and color all other connections emanating from that equation in blue. Renumber the equation using the highest free number starting from n, where n is the number of equations.

These rules are applied recursively.

After one iteration of the algorithm.



At the end of the algorithm



At the end of the algorithm and the system of equations is written as

$$u_0 = f(t) \tag{11}$$
$$u_2 = u_C \tag{12}$$
$$i_2 = u_2/R_2 \tag{13}$$
$$u_1 = u_0 - u_C \tag{14}$$
$$i_1 = u_1 \ R1 \tag{15}$$
$$i_C = i_1 - i_2 \tag{16}$$
$$u_L = u_1 + u_2 \tag{17}$$
$$\frac{du_C}{dt} = i_C/C \tag{18}$$
$$\frac{di_L}{dt} = u_L/L \tag{19}$$
$$i_0 = i_1 + i_L \tag{20}$$

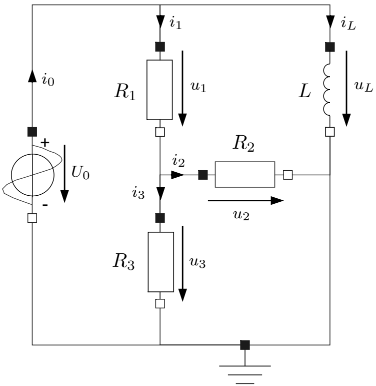
Note these equations are causal and in order to be evaluated.

$$
\begin{array}{r c}
 & \begin{array}{cccccccccc} u_0 & u_2 & i_2 & u_1 & i_1 & i_C & u_L & \frac{di_L}{dt} & \frac{du_C}{dt} & i_0 \end{array} \\
\begin{array}{r} \text{Eq. (11)} \\ \text{Eq. (12)} \\ \text{Eq. (13)} \\ \text{Eq. (14)} \\ \text{Eq. (15)} \\ \text{Eq. (16)} \\ \text{Eq. (17)} \\ \text{Eq. (18)} \\ \text{Eq. (19)} \\ \text{Eq. (20)} \end{array} &
\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}
\end{array}
$$

Note 1 the matrix is lower triangular (Tarjan $\Leftrightarrow$ matrix permutation)

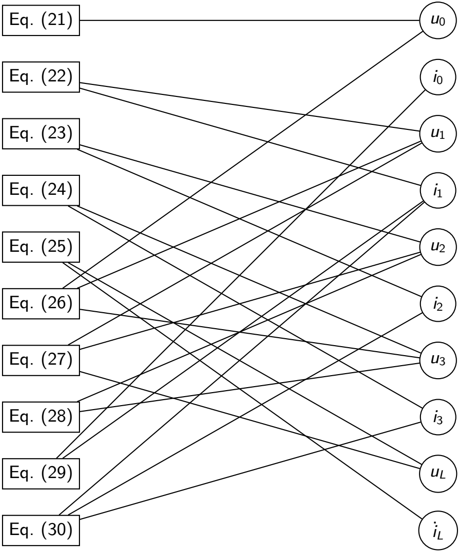
Note 2 Tarjan algorithm has a linear complexity in the number of equations.
Also used in **Pantelides algorithm**

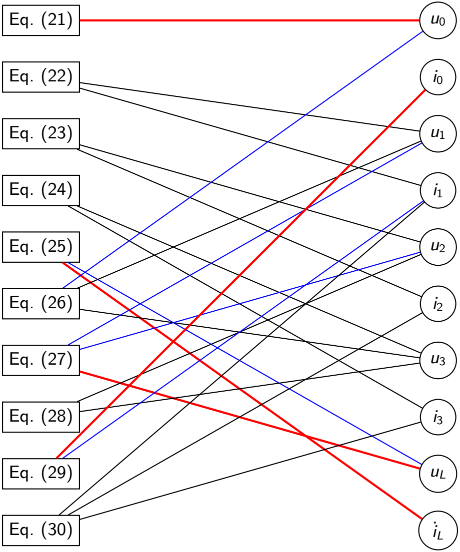
A tiny modification of the RLC circuit



$$u_0 = f(t) \tag{21}$$
$$u_1 = R_1 i_1 \tag{22}$$
$$u_2 = R_2 i_2 \tag{23}$$
$$u_3 = R_3 i_3 \tag{24}$$
$$u_L = L\frac{di_L}{dt} \tag{25}$$
$$u_0 = u_1 + u_3 \tag{26}$$
$$u_L = u_1 + u_2 \tag{27}$$
$$u_3 = u_2 \tag{28}$$
$$i_0 = i_1 + i_L \tag{29}$$
$$i_1 = i_2 + i_3 \tag{30}$$

Note the capacitor is replaced by a resistor.


Eq. (21)
Eq. (22)
Eq. (23)
Eq. (24)
Eq. (25)
Eq. (26)
Eq. (27)
Eq. (28)
Eq. (29)
Eq. (30)
u_0
i_0
u_1
i_1
u_2
i_2
u_3
i_3
u_L
$\dot{i}_L$



Remark after 2 iterations the Tarjan algorithm cannot progress any more.

$$u_0 = f(t) \tag{31}$$

$$u_1 - R_1 i_1 = 0 \tag{32}$$

$$u_2 - R_2 i_2 = 0 \tag{33}$$

$$u_3 - R_3 i_3 = 0 \tag{34}$$

$$u_1 + u_3 = u_0 \tag{35}$$

$$u_2 - u_3 = 0 \tag{36}$$

$$i_1 - i_2 - i_3 = 0 \tag{37}$$

$$u_L = u_1 + u_2 \tag{38}$$

$$\frac{di_L}{dt} = u_L / L \tag{39}$$

$$i_0 = i_1 + i_L \tag{40}$$

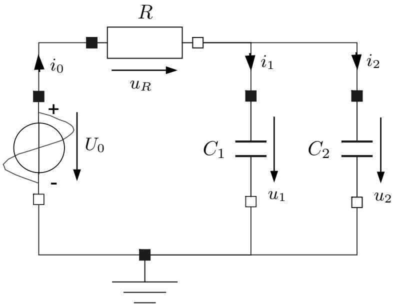
Note The last six equations form an algebraic loop and cannot be sorted then they must be solved all together.

$$
\mathbf{S} = \begin{array}{rc}
 & \begin{array}{cccccccccc} u_0 & u_1 & i_1 & u_2 & i_2 & u_3 & i_3 & u_L & \frac{di_L}{dt} & i_0 \end{array} \\
\begin{array}{r} Eq.(7.6a) \\ Eq.(7.6b) \\ Eq.(7.6c) \\ Eq.(7.6d) \\ Eq.(7.6e) \\ Eq.(7.6f) \\ Eq.(7.6g) \\ Eq.(7.6h) \\ Eq.(7.6i) \\ Eq.(7.6j) \end{array} &
\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}
\end{array}
\tag{7.7}
$$

Algebraic loops deserve special treatment:

- in case of linear system: Gauss elimination
- otherwise: Newton algorithm

Algebraic loops are very frequent in multi-body dynamics.



$$u_0 = f(t) \tag{41}$$
$$u_R = Ri_0 \tag{42}$$
$$i_1 = C_1 \frac{du_1}{dt} \tag{43}$$
$$u_2 = C_2 \frac{du_2}{dt} \tag{44}$$
$$u_0 = u_R + u_1 \tag{45}$$
$$u_2 = u_1 \tag{46}$$
$$i_0 = i_1 + i_2 \tag{47}$$

If the state variables are u_1 and u_2 then Eq. (46) is a **constraint** (a variable as only blue edges in the structure digraph).

Pantelides algorithm can can be used to handle this situation

If $u_2 = u_1$ is true for all t then

$$\frac{du_2}{dt} = \frac{du_1}{dt} \quad \text{for all t} \tag{48}$$

Idea use symbolic differentiation to compute Eq. (48) and replace the constraint by its derivative. Hence,

$$\begin{aligned}
u_0 &= f(t) && (49)\\
u_R &= Ri_0 && (50)\\
i_1 &= C_1 \frac{du_1}{dt} && (51)\\
u_2 &= C_2 \frac{du_2}{dt} && (52)\\
u_0 &= u_R + u_1 && (53)\\
\frac{du_2}{dt} &= \frac{du_1}{dt} && (54)\\
i_0 &= i_1 + i_2 && (55)
\end{aligned}$$

Using Tarjan algorithm we get an algebraic loop but we know how to deal with.

Structurally singular systems are also known as **higher index problems**.

- an index-0 contains neither algebraic loop nor structural singularities
- index 1 contains algebraic loops but no structural singularities

Pantelides is a **symbolic index reduction** algorithm. One application reduces the index by 1.

Issues of index reduction

Issues

- **Consistent initial conditions** finding initial value for differential and algebraic variables may be very difficult.
  For
  $$F(\dot{x}, x, t) = 0$$
  x_0 is a consistent initial value, if there exists a smooth solution that fulfills $x(0) = x_0$ and this solution is defined for all t.
  E.g., semi-explicit DAE with only $x(0) = x_0$ what about the algebraic variable?
- **Drift off effect** when applying index reduction the solution of the lower index DAE may not be of the original index.

In consequence, tools/methods to solve DAE should

- provide automatic index reduction
- be able to find consistent initial values

e.g., Dymola/Modelica

Let

$$\dot{u} = -0.5(u + v) + q_1(t)$$
$$0 = 0.5(u - v) - q_2(t)$$

If $u(0)$ is given we can determine $v(0) = u(0) - 2q_2(0)$ and so $\dot{u}(0)$.

Set $u = y_1 + y_2$ and $v = y_1 - y_2$ we get

$$\dot{y}_1 + \dot{y}_2 = -y_1 + q_1(t)$$
$$0 = y_2 - q_2(t)$$

For consistency we must have $y_2(0) = q_2(0)$ but we can choose $y_1(0)$ arbitrarily but we cannot determine $\dot{y}_1(0)$ without using $\dot{y}_2(0) = \dot{q}_2(0)$.

Going from index 3 pendulum to index 2 by differentiating the constraint $x_1^2 + x_2^2 - \ell^2 = 0$ leads to

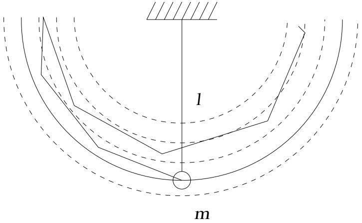
$$\dot{x}_1 = x_3 \tag{56}$$

$$\dot{x}_2 = x_4 \tag{57}$$

$$\dot{x}_3 = -\frac{F}{\ell} x_1 \tag{58}$$

$$\dot{x}_4 = g \frac{F}{\ell} x_2 \tag{59}$$

$$0 = x_1 x_3 + x_2 x_4 \tag{60}$$



Comments:

- solid line curve is the result of index 3 pendulum problem
- Constraint (60) says the velocity should orthogonal to the position. Index reduction increase the space of solution with dashed line curves

Part 6. Section 4
Sovability of IVP DAE

For ODE, we have a theorem applying on a large class of problem proving the existence and unicity of the solution

No such theorem exists for DAE

Instead we have some theorems of **solvability** of different kinds of DAE

- Linear constant coefficient DAE
- Linear time varying coefficient DAE
- Non-linear DAE

Definition

Let $\mathcal{I}$ be an open sub-interval of $\mathbb{R}$, Ω a connected open subset of $\mathbb{R}^{2m+1}$, and F a differentiable function from Ω to $\mathbb{R}^m$. Then the DAE $F(\dot{x}, x, t) = 0$ is *solvable* on $\mathcal{I}$ in Ω if there is an r-dimensional family of solutions $\phi(t, c)$ defined on a connected open set $\mathcal{I} \times \tilde{\Omega}$, $\tilde{\Omega} \subset \mathbb{R}^r$, such that

1. $\phi(t, c)$ is defined on all of $\mathcal{I}$ for each $c \in \tilde{\Omega}$
2. $(\phi'(t, c), \phi(t, c), t) \in \Omega$ for $(t, c) \in \mathcal{I} \times \tilde{\Omega}$
3. If $\psi(t)$ is any other solution with $(\psi'(t, c), \psi(t, c), t) \in \Omega$ then $\psi(t) = \phi(t, c)$ for some $c \in \tilde{\Omega}$
4. The graph of ϕ as a function of (t, c) is an $r + 1$-dimensional manifold.

Let

$$A\dot{x} + Bx = f$$

And consider the **matrix pencil** $\lambda A + B$

A matrix pencil is regular if $\det(\lambda A + B)$ is not identically zero as a function of λ.

Theorem

The linear constant coefficient DAE is solvable if and only if $\lambda A + B$ is regular pencil.

Note: the degree of nilpotency of the matrix N used in the decomposition is also the index number of the DAE.

DAE are a generalisation of ODE but

- there is no general theorem to prove existence of the solution of DAE
- differentiation used to index reduction can introduce singularities
- the class of numerical methods used to solve DAE is rather small compare to ODE.

Part 6. Section 5
Initial Value Problem for DAE – solving methods

We will consider DAE in Hessenberg form of index 1

$$\dot{\mathbf{y}} = f\left(t, \mathbf{y}, \mathbf{z}\right)$$

$$0 = g\left(\mathbf{y}, \mathbf{z}\right) \quad \text{with} \quad \frac{\partial g}{\partial \mathbf{z}} \quad \text{is non-singular}$$

$$\text{with} \quad \mathbf{z}(0) = \mathbf{z}_0 \quad \text{and} \quad \mathbf{y}(0) = \mathbf{y}_0$$

and sometimes, DAE of the following form can be considered

$$\mathbf{M}\dot{\mathbf{y}}(t) = f(\mathbf{y}(t))$$

M is known as the *Mass Matrix*

Relation between DAE and stiff ODE

Singularly perturbed ODE systems are of the form

$$\dot{\mathbf{y}} = f(t, \mathbf{y}, \mathbf{z}) \tag{61}$$

$$\varepsilon \dot{\mathbf{z}} = g(t, \mathbf{z}, \mathbf{y}) \tag{62}$$

When $\varepsilon = 0$ then we get a DAE but Eq. (61) is usually stiff.
DAE can be seen as infinitely stiff.

Consequence

not all numerical method to solve ODE can be used to solve DAE!

we want A-stable methods (event L-stable) but stiffly stable is enough (as for BDF)

$$\dot{\mathbf{y}} = f\left(t, \mathbf{y}, \mathbf{z}\right)$$
$$0 = g\left(\mathbf{y}, \mathbf{z}\right) \quad \text{with} \quad \frac{\partial g}{\partial \mathbf{z}} \quad \text{is non-singular}$$
$$\text{with} \quad \mathbf{z}(0) = \mathbf{z}_0 \quad \text{and} \quad \mathbf{y}(0) = \mathbf{y}_0$$

By *Implicit function theorem* there exists (at leat locally) a function $G(\mathbf{y})$ such that

$$\mathbf{z} = G(\mathbf{y})$$

By substitution we can have

$$\dot{\mathbf{y}} = f(t, \mathbf{y}, G(y))$$

which can be solved by any method for IVP ODE **but**

- you lose the structure of the problem
- G is not so simple to get

$$\dot{\mathbf{y}} = f\left(t, \mathbf{y}, \mathbf{z}\right)$$

$$\varepsilon \dot{\mathbf{z}} = g\left(\mathbf{y}, \mathbf{z}\right) \quad \text{with} \quad \frac{\partial g}{\partial \mathbf{z}} \quad \text{is non-singular}$$

$$\text{with} \quad \mathbf{z}(0) = \mathbf{z}_0 \quad \text{and} \quad \mathbf{y}(0) = \mathbf{y}_0$$

Applying a Runge-Kutta method,

$$\mathbf{Y}_{\mathrm{ni}} = \mathbf{y}_n + h \sum_{j=1}^{s} a_{ij} f(\mathbf{Y}_{\mathrm{nj}}, \mathbf{Z}_{\mathrm{nj}})$$

$$\varepsilon \mathbf{Z}_{\mathrm{ni}} = \varepsilon \mathbf{z}_n + h \sum_{j=1}^{s} a_{ij} g(\mathbf{Y}_{\mathrm{nj}}, \mathbf{Z}_{\mathrm{nj}})$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h \sum_{i=1}^{s} b_i f(\mathbf{Y}_i, \mathbf{Z}_i)$$

$$\varepsilon \mathbf{z}_{n+1} = \varepsilon \mathbf{z}_n + h \sum_{i=1}^{s} b_i g(\mathbf{Y}_i, \mathbf{Z}_i)$$

Applying a Runge-Kutta method,

$$\mathbf{Y}_{\mathrm{ni}} = \mathbf{y}_n + h \sum_{j=1}^{s} a_{ij} f(\mathbf{Y}_{\mathrm{nj}}, \mathbf{Z}_{\mathrm{nj}})$$

$$\varepsilon \mathbf{Z}_{\mathrm{ni}} = \varepsilon \mathbf{z}_n + h \sum_{j=1}^{s} a_{ij} g(\mathbf{Y}_{\mathrm{nj}}, \mathbf{Z}_{\mathrm{nj}})$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h \sum_{i=1}^{s} b_i f(\mathbf{Y}_i, \mathbf{Z}_i)$$

$$\varepsilon \mathbf{z}_{n+1} = \varepsilon \mathbf{z}_n + h \sum_{i=1}^{s} b_i g(\mathbf{Y}_i, \mathbf{Z}_i)$$

assuming the matrix $\mathbf{A}$ of coefficients a_{ij} is non singular,

$$hg(\mathbf{Y}_{\mathrm{ni}}, \mathbf{Z}_{\mathrm{ni}}) = \varepsilon \sum_{j=1}^{s} \omega_{ij} \left(\mathbf{Y}_{\mathrm{nj}} - \mathbf{z}_n\right) \quad \text{with} \quad \omega_{ij} = (a_{ij})^{-1}$$

From

$$hg(\mathbf{Y}_{\mathrm{ni}}, \mathbf{Z}_{\mathrm{ni}}) = \varepsilon \sum_{j=1}^{s} \omega_{ij} \left(\mathbf{Y}_{\mathrm{nj}} - \mathbf{z}_n\right) \quad \text{with} \quad \omega_{ij} = \left(a_{ij}\right)^{-1}$$

we get,

$$\begin{aligned}
\mathbf{Y}_{\mathrm{ni}} &= \mathbf{y}_n + h \sum_{j=1}^{s} a_{ij} f(\mathbf{Y}_{\mathrm{nj}}, \mathbf{Z}_{\mathrm{nj}}) \\
0 &= g\left(\mathbf{Y}_{\mathrm{ni}}, \mathbf{Z}_{\mathrm{ni}}\right) \\
\mathbf{y}_{n+1} &= \mathbf{y}_n + h \sum_{i=1}^{s} b_i f(\mathbf{Y}_i, \mathbf{Z}_i) \\
\mathbf{z}_{n+1} &= \left(1 - \sum_{i,j=1}^{s} b_i \omega_{ij}\right) \mathbf{z}_n + \sum_{i,j=1}^{s} b_i \omega_{ij} \mathbf{Z}_{\mathrm{nj}} \quad \text{independence wrt } \varepsilon
\end{aligned}$$

Remark: this approach can lead to numerical divergence as the solution may not respect the constraint $g(y, z) = 0$

Approximating state-space method can be reached by the formula

$$\mathbf{Y}_{\mathrm{ni}} = \mathbf{y}_n + h\sum_{j=1}^{s} a_{ij} f(\mathbf{Y}_{\mathrm{nj}}, \mathbf{Z}_{\mathrm{nj}})$$

$$0 = g\left(\mathbf{Y}_{\mathrm{ni}}, \mathbf{Z}_{\mathrm{ni}}\right)$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h\sum_{i=1}^{s} b_i f(\mathbf{Y}_i, \mathbf{Z}_i)$$

$$0 = g(\mathbf{y}_{n+1}, \mathbf{z}_{n+1})$$

Remarks

- For stiffly accurate methods (see next slide) ε-embedding method and state-space method are identical
- ε-embedding method can be generalized to other classes of DAE index 1 (mass matrix form or implicit form)

A Runge-Kutta is defined by its Butcher tableau

$$\begin{array}{c|cccc}
c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\
\vdots & \vdots & \vdots & & \vdots \\
c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\
\hline
 & b_1 & b_2 & \cdots & b_s \\
 & b'_1 & b'_2 & \cdots & b'_s & \quad \text{(optional)}
\end{array}$$

Remark

For DAE, we only consider fully implicit Runge-Kutta methods which are L-stable, with A non-singular and with $b_j = a_{sj}$ $(j = 1, 2, \ldots, s)$.

The most used method are **Backward Euler's method** and **Radau IIA order 5**.

Remark:

- the last condition $b_j = a_{sj}$ is good as the last step of RK method is not applied on algebraic variable.
- Stiffly accurate is sufficient for semi-explicit index 1 but not for higher index

Recall: single-step methods solve IVP using one value $\mathbf{y}_n$ and some values of f.

A multi-step method approximate solution $\mathbf{y}_{n+1}$ of IVP using k previous values of the solution $\mathbf{y}_n$, $\mathbf{y}_{n-1}$, ..., $\mathbf{y}_{n-k-1}$.

Different methods implement this approach

- Adams-Bashworth method (explicit)
- Adams-Moulton method (implicit)
- Backward Difference Method (implicit)

The general form of such method is

$$\sum_{j=0}^{k} \alpha_j \mathbf{y}_{n+j} = h \sum_{j=0}^{k} \beta_j f(t_{n+j}, \mathbf{y}_{n+j}) \ .$$

with α_j and β_j some constants and $\alpha_k = 1$ and $|\alpha_0| + |\beta_0| \neq 0$

We consider

$$\dot{\mathbf{y}} = f\left(t, \mathbf{y}, \mathbf{z}\right)$$

$$0 = g\left(\mathbf{y}, \mathbf{z}\right) \quad \text{with} \quad \frac{\partial g}{\partial \mathbf{z}} \quad \text{is non-singular}$$

$$\text{with} \quad \mathbf{z}(0) = \mathbf{z}_0 \quad \text{and} \quad \mathbf{y}(0) = \mathbf{y}_0$$

by using ε-embedding method.

$$\dot{\mathbf{y}} = f\left(t, \mathbf{y}, \mathbf{z}\right)$$

$$\varepsilon\dot{\mathbf{z}} = g\left(\mathbf{y}, \mathbf{z}\right) \quad \text{with} \quad \frac{\partial g}{\partial \mathbf{z}} \quad \text{is non-singular}$$

$$\text{with} \quad \mathbf{z}(0) = \mathbf{z}_0 \quad \text{and} \quad \mathbf{y}(0) = \mathbf{y}_0$$

Applying, multi-step method, we get

$$\sum_{i=0}^{k} \alpha_i \mathbf{y}_{n+i} = h \sum_{i=0}^{k} \beta_i f(\mathbf{y}_{n+i}, \mathbf{z}_{n+i})$$

$$\varepsilon \sum_{i=0}^{k} \alpha_i \mathbf{z}_{n+i} = h \sum_{i=0}^{k} \beta_i g(\mathbf{y}_{n+i}, \mathbf{z}_{n+i})$$

Applying, multi-step method, we get

$$\sum_{i=0}^{k} \alpha_i \mathbf{y}_{n+i} = h \sum_{i=0}^{k} \beta_i f(\mathbf{y}_{n+i}, \mathbf{z}_{n+i})$$
$$\varepsilon \sum_{i=0}^{k} \alpha_i \mathbf{z}_{n+i} = h \sum_{i=0}^{k} \beta_i g(\mathbf{y}_{n+i}, \mathbf{z}_{n+i})$$

and setting $\varepsilon = 0$ we get

$$\sum_{i=0}^{k} \alpha_i \mathbf{y}_{n+i} = h \sum_{i=0}^{k} \beta_i f(\mathbf{y}_{n+i}, \mathbf{z}_{n+i})$$
$$0 = h \sum_{i=0}^{k} \beta_i g(\mathbf{y}_{n+i}, \mathbf{z}_{n+i})$$

A state-space method can be applied by using

$$\sum_{i=0}^{k} \alpha_i \mathbf{y}_{n+i} = h \sum_{i=0}^{k} \beta_i f(\mathbf{y}_{n+i}, \mathbf{z}_{n+i})$$
$$0 = g(\mathbf{y}_{n+k}, \mathbf{z}_{n+k})$$

For BDF one has

$$\frac{1}{h\beta_0}\sum_{i=0}^{k}\alpha_i \mathbf{y}_{n+i} = f(\mathbf{y}_{n+k}, \mathbf{z}_{n+k})$$

$$0 = g(\mathbf{y}_{n+k}, \mathbf{z}_{n+k})$$

Remarks

- we still need stiffly accurate method so BDF has to be considered
- Can be applied on DAE index 2 also

Convergence

m-step BDF with $m < 6$ converge; *i.e.*,

$$\mathbf{y}(t_i) - \mathbf{y}_i \le \mathcal{O}(h^m) \quad \text{and} \quad \mathbf{z}(t_i) - \mathbf{z}_i \le \mathcal{O}(h^m)$$

for **consistent initial values**.