Lecture 1

Simulation of differential-algebraic equations

Erik Frisk
erik.frisk@.liu.se

Department of Electrical Engineering
Linköping University

October 31, 2017

Is and is not

What this part of the course is (hopefully):

- Understand what a DAE is, characteristics, and structure
- Understand why they are useful
- Understand why they are (sometimes) more difficult to simulate than an ODE
- Understand the origins of the difficulties and how to detect them
- Know how and when one can expect your favourite solver for ODE:s to work well also for DAE:s
- How to simulate models described in object orients languages, like Modelica

What this part is not:

- detailed derivations and analysis of specific methods for simulation of DAE:s

Outline of the DAE module, lectures

1) Basic properties
   - principles
   - differences between ODE:s and DAE:s
   - differential index
2) Simulation methods
   - principal problems with high index problems
   - simulation of low-index problems
   - index reduction techniques
3) DASSL och Modelica, simulation of object-oriented models
4) Modelica continued
   - Simulation of Modelica moidels, structural analysis
   - index reduction using dummy-derivatives

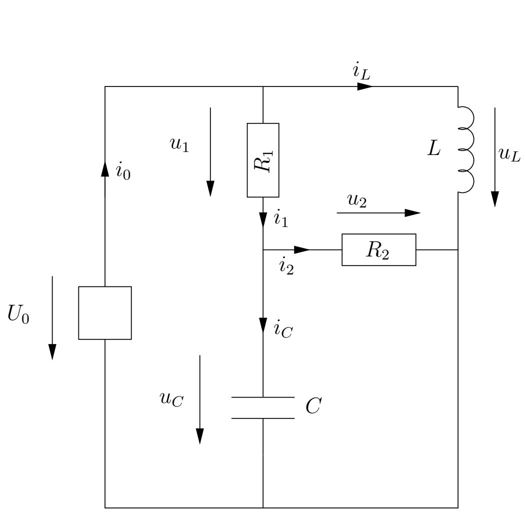
Outline

- *Introduction to differential-algebraic models*
- *Briefly; solution to differential-algebraic equations*
- *Illustrative example in three acts*
- *Differential index*
- *Initial conditions*
- *Simulation of DAE:s with low index*
- *Implicit and semi-explicit forms*

Why DAE?

- Object oriented modelling
- Basic physics
- structure and numerics
- Invariants
- Simplification of an ODE, e.g., assume a physical connection is stiff instead of flexible. Can result in a DAE that is much simple to solve than the original ODE
- Singular perturbation problems (SPP)
- Inverse problems, given $y(t)$, simulate corresponding u
- Many names: singular, implicit, descriptor, generalized state-space, non-causal, semi-state, ...

A simple electrical circuit



$$
\begin{aligned}
u_0 &= f(t)\\
u_1 &= R_1 i_1\\
u_2 &= R_2 i_2\\
i_C &= C\frac{du_C}{dt}\\
u_L &= L\frac{di_L}{dt}\\
i_0 &= i_1 + i_L\\
i_1 &= i_2 + i_C\\
u_0 &= u_1 + u_C\\
u_L &= u_1 + u_2\\
u_C &= u_2
\end{aligned}
$$

10 equation in 10 unknown $(u_0, u_1, u_2, u_L, u_C, i_0, i_1, i_2, i_L, i_C)$

Modelica model of the circuit

```
model Circuit
  import Modelica.Electrical.Analog.Basic.*;
  import Modelica.Electrical.Analog.Sources.*;
  Resistor R1;
  Resistor R2;
  Capacitor C;
  Inductor L;
  Ground G;
  SineVoltage src;
equation
  connect(G.p, src.n);
  connect(src.p, R1.p);
  connect(src.p, L.p);
  connect(R1.n, R2.p);
  connect(R1.n,C.p);
  connect(L.n, R2.n);
  connect(L.n, C.n);
  connect(C.n, G.p);
end Circuit;
```

Equations generated from the Modelica model (33 eqs.)

```
R1.R * R1.i = R1.v;
R1.v = R1.p.v - R1.n.v;
0.0 = R1.p.i + R1.n.i;
R1.i = R1.p.i;
R2.R * R2.i = R2.v;
R2.v = R2.p.v - R2.n.v;
0.0 = R2.p.i + R2.n.i;
R2.i = R2.p.i;
C.i = C.C * der(C.v);
C.v = C.p.v - C.n.v;
0.0 = C.p.i + C.n.i;
C.i = C.p.i;
L.L * der(L.i) = L.v;
L.v = L.p.v - L.n.v;
0.0 = L.p.i + L.n.i;
L.i = L.p.i;
G.p.v = 0.0;
src.signalSource.y = sin();
src.v = src.signalSource.y;
src.v = src.p.v - src.n.v;
0.0 = src.p.i + src.n.i;
src.i = src.p.i;
L.n.i + R2.n.i + C.n.i + G.p.i
+ src.n.i = 0.0;
L.n.v = R2.n.v;
R2.n.v = C.n.v;
C.n.v = G.p.v;
G.p.v = src.n.v;
R1.n.i + R2.p.i + C.p.i = 0.0;
R1.n.v = R2.p.v;
R2.p.v = C.p.v;
src.p.i + R1.p.i + L.p.i = 0.0;
src.p.v = R1.p.v;
R1.p.v = L.p.v;
```

Differential-algebraic models

A general DAE can be written in the form

$$F(y, \frac{d}{dt}y, t) = 0$$

which is kind of similar to an ODE

$$\frac{d}{dt}y = f(y, t)$$

How big difference could there be?

Why not apply an explicit Euler-forward

$$F(y_{t-h}, (y_t - y_{t-h})/h, t - h) = 0$$

and solve for y_t?
Or, probably even better, an Euler-backwards

$$F(y_t, (y_t - y_{t-h})/h, t) = 0$$

Seems pretty straightforward and similar to the ODE case. Unfortunately, it does not work!

A simple case

Assume a DAE

$$\dot{x}_1 = f(x_1, x_2, t)$$
$$0 = g(x_1, x_2, t)$$

If you can solve for x_2 in the second equation $x_2 = g^{-1}(x_1, t)$ you'll have an ODE

$$\dot{x}_1 = f(x_1, g^{-1}(x_1, t), t)$$

Perhaps better to apply Euler-backwards directly?

$$F(y_n, (y_n - y_{n-1})/h, t_n) = 0$$

Loss of structure when transforming into an ODE!

Differential-algebraic models

A general DAE

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

is pretty similar to an ODE

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

What is the difference? When can an ODE solver work also for DAE:s?

Answer: Sometimes

This first lecture deals with these differences, characteristics of DAE:s and when ODE methods can be directly applied

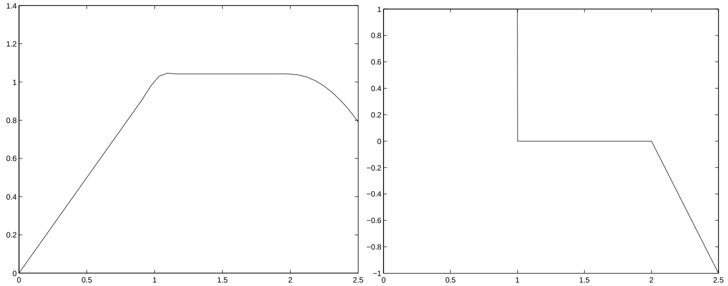
Next time more on how to simulate DAE:s and how to transform them into a form suitable for an ODE solver. Euler backwards is a common approach.

Outline

DAE and ODE

$$\dot{y}(t) = z(t)$$

- Integration, gives smoother solutions; differentiation gives more non-smooth solutions.
- Differentiation is "simpler" than integration analytically; numerically it is the other way around
- ODE - pure integration.
  DAE - mix between integration and differentiation



Solutions

Assume a DAE

$$\begin{aligned} z_1 &= g(t) \\ \dot{z}_1 &= z_2 \end{aligned}$$

You can easily see that it is not direct to numerically derive solutions $(z_1(t), z_2(t))$ if the function $g(t)$ has discontinouties.

For ODE:s the situation is more simple

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

Different DAE formulations

Implicit ODE

$$F(y, \dot{y}, t) = 0, \quad F_{y'} \text{ invertible}$$

Linear time-invariante DAE

$$E\dot{y} = Ay, E \text{ singular}$$

Semi-explicit DAE

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, x_2, t) \\ 0 &= f_2(x_1, x_2, t) \end{aligned}$$

Solvability/solutions

Definitions on solvability for DAE is similar to solvability for ODE:s.

Require consistency!

One difference worth noting: An ODE solution is always at least once differentiable, this is not true for DAE:s and all components are not as smooth.

Consider

$$\begin{aligned} \dot{y} &= x \\ y &= f \end{aligned} \Leftrightarrow \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ f \end{pmatrix}$$

where $f \in \mathcal{C}^1$. Then y will be 1 time differentiable and x not differentiable.

Solvability

A linear and time-invariant DAE

$$A\dot{y} + By = f(t)$$

is solvable if and only if $\lambda A + B$ has full rank for any $\lambda \in \mathbb{C}$ (think Laplace-transform) for a smooth $f(t)$.

$$(sA + B)Y(s) = F(s)$$

However, the DAE

$$\begin{bmatrix} -t & t^2 \\ -1 & t \end{bmatrix} \frac{d}{dt} y + y = 0$$

is not solvable on the interval $t > 0$ in spite of $|\lambda A(t) + B(t)| \equiv 1$. Home assignment: figure out why. Hint: uniqueness.

That this is a DAE and not an (implicit) ODE is due to

$$\det A(t) \equiv 0$$

Time-variable DAE:s can be difficult, both in defining solutions and numerically.

DAE vs. stiff problems

A semi-explicit DAE

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, x_2, t) \\ 0 &= f_2(x_1, x_2, t) \end{aligned}$$

is similar to the stiff ODE (ϵ small)

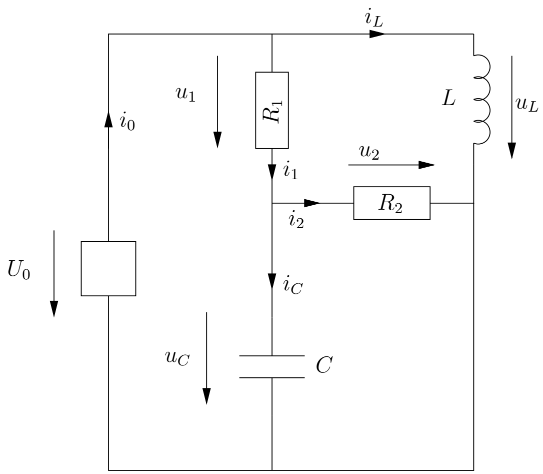
$$\begin{aligned} \dot{x}_1 &= f_1(x_1, x_2, t) \\ \epsilon \dot{x}_2 &= f_2(x_1, x_2, t) \end{aligned}$$

- similarities
- differences
- when do ODE methods work for DAE:s?
- In this presentation, I will for simplicity mainly illustrate using one-step Euler-backwards

Outline

- *Introduction to differential-algebraic models*
- *Briefly; solution to differential-algebraic equations*
- *Illustrative example in three acts*
- *Differential index*
- *Initial conditions*
- *Simulation of DAE:s with low index*
- *Implicit and semi-explicit forms*

The simple circuit model, act 1



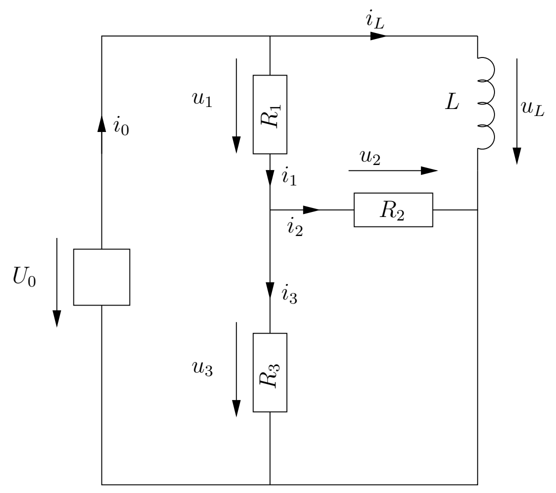
$$\begin{aligned} u_0 &= f(t) \\ u_1 &= R_1 i_1 \\ u_2 &= R_2 i_2 \\ i_C &= C\frac{du_c}{dt} \\ u_L &= L\frac{di_L}{dt} \\ i_0 &= i_1 + i_L \\ i_1 &= i_2 + i_C \\ u_0 &= u_1 + u_C \\ u_L &= u_1 + u_2 \\ u_C &= u_2 \end{aligned}$$

$x_1 = (u_c, i_L)$, $x_2 = (u_2, i_2, u_0, u_1, u_L, i_1, i_C, i_0)$

Reformulate equations into computational form

$$
\begin{aligned}
e_1 &: u_0 = f(t) \\
e_2 &: u_1 = R_1 i_1 \\
e_3 &: u_2 = R_2 i_2 \\
e_4 &: i_C = C\frac{du_c}{dt} \\
e_5 &: u_L = L\frac{di_L}{dt} \\
e_6 &: i_0 = i_1 + i_L \\
e_7 &: i_1 = i_2 + i_C \\
e_8 &: u_0 = u_1 + u_C \\
e_9 &: u_L = u_1 + u_2 \\
e_{10} &: u_C = u_2
\end{aligned}
\quad \Rightarrow \quad
\begin{aligned}
e_4 &: \frac{du_c}{dt} = \frac{1}{C} i_C \\
e_5 &: \frac{di_L}{dt} = \frac{1}{L} u_L \\
\\
e_{10} &: u_2 := u_C \\
e_3 &: i_2 := \frac{1}{R_2} u_2 \\
e_1 &: u_0 := f(t) \\
e_8 &: u_1 := u_0 - u_C \\
e_9 &: u_L := u_1 + u_2 \\
e_2 &: i_1 := \frac{1}{R_1} u_1 \\
e_7 &: i_C := i_1 - i_2 \\
e_6 &: i_0 := i_1 + i_L
\end{aligned}
$$

The simple circuit model, act 2 ($C \to R_3$)



$$
\begin{aligned}
u_0 &= f(t) \\
u_1 &= R_1 i_1 \\
u_2 &= R_2 i_2 \\
u_3 &= R_3 i_3 \\
u_L &= L\frac{di_L}{dt} \\
i_0 &= i_1 + i_L \\
i_1 &= i_2 + i_3 \\
u_0 &= u_1 + u_3 \\
u_L &= u_1 + u_2 \\
u_3 &= u_2
\end{aligned}
$$

$x_1 = i_L$, $x_2 = (u_L, u_2, i_2, u_0, u_1, u_L, i_1, i_C, i_0)$

Reformulate equations into computational form

$$\frac{di_L}{dt} = \frac{1}{L} u_L$$

$$u_0 := f(t)$$

Solve for $\{u_1, u_2, u_3, i_1, i_2, i_3\}$ in (6 unknowns, 6 equations)

$$
\begin{aligned}
u_1 &= R_1 i_1 \\
u_2 &= R_2 i_2 \\
u_3 &= R_3 i_3 \\
i_1 &= i_2 + i_3 \\
u_0 &= u_1 + u_3 \\
u_3 &= u_2
\end{aligned}
$$

$$
\begin{aligned}
i_0 &:= i_1 + i_L \\
u_L &:= u_1 + u_2
\end{aligned}
$$

Reformulate equations into computational form

$$\frac{di_L}{dt} = \frac{1}{L} u_L$$

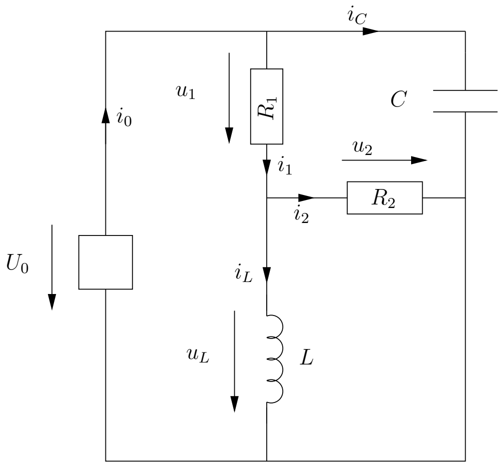
$$u_0 := f(t)$$

Solve for $\{u_1, u_2, u_3, i_1, i_2, i_3\}$ in (6 unknowns, 6 equations)

$$
\begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ i_1 \\ i_2 \\ i_3 \end{pmatrix} := \frac{1}{R_1R_2 + R_1R_3 + R_2R_3} \begin{pmatrix} R_1(R_2 + R_3) \\ R_2R_3 \\ R_2R_3 \\ R_2 + R_3 \\ R_3 \\ R_2 \end{pmatrix} u_0
$$

$$
\begin{aligned}
i_0 &:= i_1 + i_L \\
u_L &:= u_1 + u_2
\end{aligned}
$$

The simple circuit model, act 3



$$
\begin{aligned}
u_0 &= f(t)\\
u_1 &= R_1 i_1\\
u_2 &= R_2 i_2\\
i_C &= C\frac{du_c}{dt}\\
u_L &= L\frac{di_L}{dt}\\
i_0 &= i_1 + i_C\\
i_1 &= i_2 + i_L\\
u_0 &= u_1 + u_L\\
u_C &= u_1 + u_2\\
u_L &= u_2
\end{aligned}
$$

$x_1 = (u_C, i_L)$, $x_2 = (u_2, i_2, u_0, u_1, u_L, i_1, i_C, i_0)$

Reformulate equations into computational form

Is not possible to, in the same way as before, to obtain a computational form. If you write the model in the form

$$
\begin{aligned}
\dot{x}_1 &= g(x_1, x_2)\\
0 &= h(x_1, x_2)
\end{aligned}
$$

where $x_1 = (u_C, i_L)$ och $x_2 = (u_0, u_1, u_2, u_L, i_0, i_1, i_2, i_C)$. Then

$$
\text{rank } h_{x_2} = \text{rank } \frac{\partial h(x_1, x_2)}{\partial x_2} =
$$

$$
= \text{rank} \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0\\
0 & 1 & 0 & 0 & 0 & -1 & -1 & 0\\
0 & 0 & 1 & 0 & 0 & 0 & -1 & -1\\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 1\\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\
0 & -R1 & 0 & -1 & 1 & 0 & 0 & 0\\
0 & 0 & -R2 & 0 & -1 & 0 & 0 & 0\\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0
\end{pmatrix} = 7 < 8
$$

Summary of the three acts

- Act 1: simple, very similar to an ODe
- Act 2: bit more difficult, took some algebra but we were OK
- Act 3: significantly more difficult

The difference between these three acts were changes in components.

Important: All three are well formed models!

A main property that separates them is: *differential index*

Outline

Index, one example

A linear example that illustrates an important difference between DAE:er and ODE:er

$$\begin{aligned} \dot{x}_1 + x_2 + x_3 &= f_1 \\ \dot{x}_2 + x_1 &= f_2 \\ x_2 &= f_3 \end{aligned} \quad \Rightarrow \quad \begin{aligned} \dot{x}_1 &= \dot{f}_2 - \ddot{f}_3 \\ \dot{x}_2 &= -x_1 + f_2 \\ \dot{x}_3 &= x_1 - f_2 - \ddot{f}_2 + \dot{f}_1 - f_3^{(3)} \end{aligned}$$

- Initial conditions. For an ODE they can be chosen freely, not for DAE:s
- "hidden" algebraic conditions

$$\begin{aligned} x_1 &= f_2 - \dot{f}_3 \\ x_2 &= f_3 \\ x_3 &= f_1 - \dot{f}_2 - f_3 + \ddot{f}_3 \end{aligned}$$

Something called (differential) *index* proves good to characterize DAE:s

(Differential-) Index

A DAE is almost an ODE, just need some differentiation

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

Differentiate the second equation

$$0 = g_x \dot{x} + g_y \dot{y} = g_x f + g_y \dot{y}$$

If g_y^{-1} exists we can rewrite as

$$\begin{aligned} \dot{x} &= f(x, y) \\ \dot{y} &= -g_y^{-1} g_x f \end{aligned}$$

Comments: solutions sets, equivalence.

Index, cont.

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

Definition

The minimum number of times the DAE has to be differentiated with respect to t to be able to determine $\dot{y}$ as a function of t och y is called the (differential-) index of the DAE.

- index might be solution dependent, uniform index
- There are several types of index, the abpove is called differetial index.
- Perturbation index
- variants of the above (see paper)

Anyhow: index is a measure how far from an ODE the DAE is.

Linear constant DAE:s of any index

$$E\dot{x} = Jx + Ku$$

Then there exists a non-singular matrix P and a change of variables $z = Qx$ such that

$$\begin{pmatrix} I & 0 \\ 0 & N \end{pmatrix} \begin{pmatrix} \dot{z}_1 \\ \dot{z}_2 \end{pmatrix} = \begin{pmatrix} A & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} + \begin{pmatrix} B \\ D \end{pmatrix} u$$

Where matrix N is nilpotent, i.e., there is an integer m such that $N^i \neq 0$ for $i < m$ and $N^m = 0$.

A simple algebra exercise gives that the solution to the DAE is

$$\begin{aligned} \dot{z}_1 &= Az_1 + Bu \\ z_2 &= -\sum_{i=0}^{m-1} N^i D u^{(i)} \end{aligned}$$

How is the degree of nilpotency m related to the index? Transfer function, how does it relate to the degrees of numerators and denominators?

Sufficient condition for index

$$F(y, \dot{y}) = 0$$
$$\frac{d}{dt}F(y, \dot{y}) = 0$$
$$\vdots$$
$$\frac{d^{j-1}}{dt^{j-1}}F(y, \dot{y}) = 0$$

which can be collected to $\mathbf{F}_j(t, y, \mathbf{y}_j) = 0$. Algebraicly $\mathbf{F}_j(t, y, \mathbf{y}_j) = 0$ consists of nj equations in $nj + n$ unknown variables.
A sufficient condition for $\dot{y}$ is a unique function (locally) if t and y is that

$$\frac{\partial \mathbf{F}_j}{\partial \mathbf{y}_j}$$

is 1-full column rank

DAE:n has index no larger than v if $\partial \mathbf{F}_{v+1}/\partial \mathbf{y}_{v+1}$ has 1-full rank and $\mathbf{F}_{v+1} = 0$ is consistent.

1-full rank

When has the equation

$$\begin{pmatrix} A_1 & A_2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = b$$

a unique solution for x_1?

Unique x_1 solution if and only if

$$\text{rang } A = n_1 + \text{rang } A_2$$

Example:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = b$$

Now, back to the last slide, what does 1-full rank mean there?

Common forms for differential equations

- ODE
$$\dot{y} = f(y, t)$$
- Hessenberg index 1/semi-explicit index 1
$$\dot{x} = f(x, z, t)$$
$$0 = g(x, z, t), \quad g_z \text{ nonsingular for all } t$$
- Hessenberg index 2
$$\dot{x} = f(x, z, t)$$
$$0 = g(x, t), \quad g_x f_z \text{ nonsingular for all } t$$

Oure index 2 equation, all algebraic variables are “index 2” variables.

Remainder of the lecture

The remainder of the lecture will introduce some important differences between ODE:s and DAE:s from a simulation perspective. We will come back to these in detail in upcoming lectures.

1 Initial conditions
2a Simulation of equations with index 0 and 1
2b Simulation of equations with index ≥ 2

Outline

Bullet 1: Initial conditions

For the DAE:n

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

is it sufficient that the initial conditions y_0 and $\dot{y}_0$ satisfies

$$F(t, y_0, \dot{y}_0) = 0?$$

- Index and "hidden" conditions
- Methods to determine consistent initial conditions
- Pantelides algorithm

$$\begin{aligned} \dot{x}_1 + x_2 + x_3 &= f_1 \\ \dot{x}_2 + x_1 &= f_2 \\ x_2 &= f_3 \end{aligned}$$

Initial conditions, cont.

What degrees of freedom do we have for the initial condition? In the equations

$$\begin{aligned} \dot{x}_1 + x_2 + x_3 &= f_1 \\ \dot{x}_2 + x_1 &= f_2 \\ x_2 &= f_3 \end{aligned}$$

there is no freedom at all and the solution was uniquely determined (in the class of smooth functions) directly by the equations.

If we have m equations/variables, it holds that the degrees of freedom l that $0 \leq l \leq m$ anbd it is not trivial to find *consistent* initial conditions.

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

Pantelides algorithm

We will come back to a possible solution later

Outline

Bullet 2a: Index 1 "as easy" as ODE

Will combe back to this on the next lecture, but the basic principle is easily illustrated.
Assume a semi-explicit DAE in the form

$$\dot{x}_1 = f_1(x_1, x_2, t)$$
$$0 = f_2(x_1, x_2, t)$$

with index 1. Then,

$$\frac{\partial f_2}{\partial x_2}$$

has full column rank and it exists a (local) inverse.

The algebraic variable can then be inserted in the dynamic equation resulting in an ODE which can be solved using any standard ODE method.

A lot more about this next lecture

Bullet 2a: Index 1 "as easy" as ODE, cont.

Consider an index 1 DAE

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

Apply a basic implicit Euler

$$F((x_t - x_{t-1})/h_t, x_t, t) = 0$$

and solve numerically for x_t. Index 1 property ensures that a solution exists.

Procedure *no different* than implicit Euler for ODE:s.

Bullet 2a: Index 1 "as easy" as ODE, cont.

One conclusion: BDF and other typical implicit solvers will work approximately the same for DAE:s of index 1 as for ODE:s.

There are practical differences though, see Hairer/Wanner and the following papers for further details

- Petzold, "Differential/algebraic equations are not ODEs"
- Brenan, Campbell and Petzold Petzold, "Numerical Solution of Initial-Value Problems in Differential Algebraic Equations"

Bullet 2b: Why is index > 1 so difficult?

Equations you, generally, can solve using basic ODE methodology is

- Index 1 DAE:s (more to follow)
- Linear DAE:s with constant coefficients of any index (kind of)

  $$A\dot{y} + By = f$$

  Will not pursue this here. More details in "ODE methods for the solution of differential/algebraic systems".
- For index > 1, direct ODE methodology does not work at all. We need new techniques and index reduction is one possibility we will discuss a lot in upcoming lectures.

Outline

Implicit and semi-explicit forms

A fully implicit DAE

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

can always be rewritten, asa semi-explicit DAE by introducing a new variable x' (algebraic, should not be confused with $\dot{x}$)

$$\begin{aligned} \dot{x} &= x' \\ F(x', x) &= 0 \end{aligned}$$

Q

Does this mean that we can forget about implicit forms and focus on semi-explicit?

A

No, not really.

An implicit example

Consider the implicit index-1 DAE

$$\begin{aligned} e_1 &: \dot{x}_1 + \dot{x}_2 = u_1 \\ e_2 &: x_1 - x_2 = u_2 \end{aligned}$$

From equations $(e_1, e_2, \dot{e}_2)$ we can solve for the highest derivatives.

Transform the DAE into a semi-explicit DAE by introducing x_1' and x_2'

$$\begin{aligned} e_1 &: x_1' + x_2' = u_1 \\ e_2 &: x_1 - x_2 = u_2 \\ e_3 &: \frac{d}{dt}x_1 = x_1' \\ e_4 &: \frac{d}{dt}x_2 = x_2' \end{aligned}$$

Q

What is the index of this one?

An implicit example, cont'd

Turns out that

$$\begin{aligned} e_1 &: x_1' + x_2' = u_1 \\ e_2 &: x_1 - x_2 = u_2 \\ e_3 &: \frac{d}{dt}x_1 = x_1' \\ e_4 &: \frac{d}{dt}x_2 = x_2' \end{aligned}$$

has index 2.

Assignment: Verify that you need $(e_1, \dot{e}_1, e_2, \dot{e}_2, e_3, \dot{e}_3, e_4, \dot{e}_4, \ddot{e}_4)$ to be able to solve for highest derivatives.

Rule of thumb

Going from fully implicit to semi-explicit increases index by 1

Lecture 1

Simulation of differential-algebraic equations

Erik Frisk
erik.frisk@.liu.se

Department of Electrical Engineering
Linköping University

October 31, 2017