MOS ESS101
December 2017 (re-exam)

This exam contains 9 pages (including this cover page) and 5 problems.

You are allowed to use the following books:

- **β-handbook**
- **"Formeln+Hilfen Höhere Mathematik"**
- **"Physics handbook for science and engineering"**

and a calculator. Some formula specific to this course are provided in the end as an appendix

- Organize your work in a reasonably neat and coherent way. Work scattered all over the page without a clear ordering may receive less credit.
- Mysterious or unsupported answers will not receive credit, but an incorrect answer supported by substantially correct calculations and explanations will receive partial credit.
- None of the proposed questions require extremely long computations. If you get caught in endless algebra, you have probably missed the simple way of doing it.
- The passing grade will a priori be given at 28 points, and the top grade at 42 points. These limits may be lowered depending on the outcome of the exam.

Problem	Points	Score
1	9	
2	10	
3	14	
4	4	
5	13	
Total:	50	

Best of luck to all !!

1. **Lagrange modelling** Consider a mass "1" (of mass m) moving on a sphere of equation $\varphi(\mathbf{p}) = \frac{1}{2}\left(\mathbf{p}^\top\mathbf{p} - R^2\right) = 0$ and a mass "2" (also of mass m) connected to the mass "1" with a rigid, massless link of length L. The problem is illustrated in Fig. 1

   (a) (4 points) Write down the model equations of this system in the form of a semi-explicit index-3 DAE.

   Note: try to keep your notations compact. You do not need to provide $\frac{\partial \mathbf{C}}{\partial \mathbf{q}}$ explicitly (where $\mathbf{q}$ will be your set of generalised coordinates), but you need to detail the model enough that one would understand how to code it symbolically in the computer (i.e. using basic operations like Jacobians and matrix-vector multiplications)

   (b) (3 points) Propose an equivalent model in the form of a fully-implicit index-1 DAE. Specify its consistency conditions.

   (c) (2 points) After running a simulation of the index-1 DAE equations, we observe the behavior depicted in Fig. 2. This can arguably be caused by two problems. Explain both.

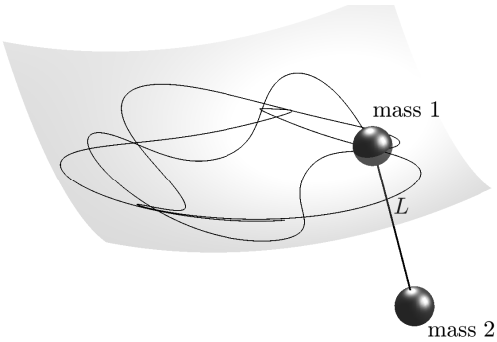

Figure 1: Illustration of the system. The surface φ is depicted as the see-through grey surface. The trajectory of the mass m_1 is depicted as a black trace on the surface φ.

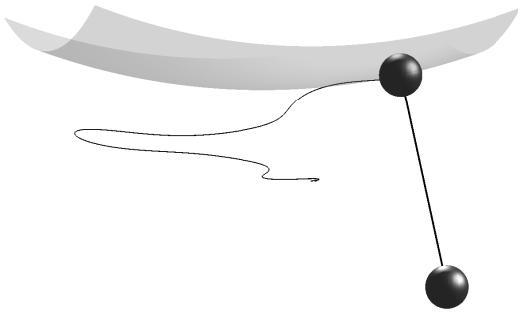

Figure 2: Simulation of the system over 10 s. The trajectory of the mass m_1 depicted as a black trace leaves the surface φ. We want to understand what can cause this.

Solution:

(a) Let us describe the system via the generalized coordinates (z-axis up):

$$\mathbf{q} = \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \end{bmatrix} \in \mathbb{R}^6 \tag{1}$$

specifying the position of each ball. The chain then has the kinetic and potential energy:

$$\mathcal{T} = \frac{1}{2} m \sum_{i=1}^{2} \dot{\mathbf{p}}_i^\top \dot{\mathbf{p}}_i \; = \frac{1}{2} m \dot{\mathbf{q}}^\top \dot{\mathbf{q}}, \qquad \mathcal{V} = mg\mathbf{a}^\top \mathbf{q} \tag{2}$$

where $\mathbf{a}^\top = \begin{bmatrix} 0 & 0 & m & 0 & 0 & m & \dots \end{bmatrix}$. We have the constraint functions

$$\mathbf{C}(\mathbf{q}) = \begin{bmatrix} \varphi(\mathbf{p}_1) \\ \frac{1}{2}\left(\|\mathbf{p}_2 - \mathbf{p}_1\|^2 - L^2 \right) \end{bmatrix} \tag{3}$$

The Lagrange function then reads as:

$$\mathcal{L}(\dot{\mathbf{q}}, \mathbf{q}, \mathbf{z}) = \frac{1}{2} m \dot{\mathbf{q}}^\top \dot{\mathbf{q}} - mg\mathbf{a}^\top \mathbf{q} - \mathbf{z}^\top \mathbf{C}(\mathbf{q}) \tag{4}$$

The Lagrange equation then provides the dynamics, using:

$$\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}} = m\dot{\mathbf{q}}, \quad \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}} = m\ddot{\mathbf{q}}, \quad \frac{\partial \mathcal{L}}{\partial \mathbf{q}} = -mg\mathbf{a} - \mathbf{z}^\top \frac{\partial \mathbf{C}}{\partial \mathbf{q}} \tag{5}$$

Hence we have the index-3 DAE model:

$$\dot{\mathbf{q}} = \mathbf{v} \tag{6a}$$

$$\dot{\mathbf{v}} = -g\mathbf{a} - \frac{1}{m}\frac{\partial \mathbf{C}}{\partial \mathbf{q}}^\top \mathbf{z} \tag{6b}$$

$$0 = \mathbf{C}(\mathbf{q}) \tag{6c}$$

This result could also be obtained directly from (62) by observing that $W(\mathbf{q}) = I$. We can additionally further explicitly provide (though this is not required from the question):

$$\frac{\partial \mathbf{C}}{\partial \mathbf{q}}^\top = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_1 - \mathbf{p}_2 \\ 0 & \mathbf{p}_2 - \mathbf{p}_1 \end{bmatrix} \tag{7}$$

(b) The index-reduced model is obtained by performing 2 time differentiations of $\mathbf{C} = 0$. They provide:

$$\dot{\mathbf{C}} = \begin{bmatrix} \mathbf{p}_1^\top \dot{\mathbf{p}}_1 \\ (\mathbf{p}_2 - \mathbf{p}_1)^\top (\dot{\mathbf{p}}_2 - \dot{\mathbf{p}}_1) \end{bmatrix} = 0 \tag{8a}$$

$$\ddot{\mathbf{C}} = \underbrace{\begin{bmatrix} \mathbf{p}_1^\top \ddot{\mathbf{p}}_1 \\ (\mathbf{p}_2 - \mathbf{p}_1)^\top (\ddot{\mathbf{p}}_2 - \ddot{\mathbf{p}}_1) \end{bmatrix}}_{= \frac{\partial \mathbf{C}}{\partial \mathbf{q}}\ddot{\mathbf{q}}} + \begin{bmatrix} \dot{\mathbf{p}}_1^\top \dot{\mathbf{p}}_1 \\ (\dot{\mathbf{p}}_2 - \dot{\mathbf{p}}_1)^\top (\dot{\mathbf{p}}_2 - \dot{\mathbf{p}}_1) \end{bmatrix} = 0 \tag{8b}$$

We can then assemble the semi-explicit index-1 DAE model e.g.

$$\begin{bmatrix} \dot{\mathbf{q}} \\ \dot{\mathbf{v}} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ -g\mathbf{a} - \frac{1}{m}\frac{\partial \mathbf{C}}{\partial \mathbf{q}}^\top \mathbf{z} \end{bmatrix} \tag{9a}$$

$$0 = \ddot{\mathbf{C}} \tag{9b}$$

or in a complete, matrix form:

$$\dot{\mathbf{q}} = \mathbf{v} \tag{10a}$$

$$\begin{bmatrix} mI & \frac{\partial \mathbf{C}}{\partial \mathbf{q}} \\ \frac{\partial \mathbf{C}}{\partial \mathbf{q}}^\top & 0 \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}} \\ \mathbf{z} \end{bmatrix} = - \begin{bmatrix} mg\mathbf{a} \\ \dot{\mathbf{p}}_1^\top \dot{\mathbf{p}}_1 \\ (\dot{\mathbf{p}}_2 - \dot{\mathbf{p}}_1)^\top (\dot{\mathbf{p}}_2 - \dot{\mathbf{p}}_1) \end{bmatrix} \tag{10b}$$

For simulation purposes, one would typically introduce $\mathbf{p}_1, \mathbf{p}_2, \dot{\mathbf{p}}_1, \dot{\mathbf{p}}_2, \mathbf{z}$ in the state-space. The consistency conditions require:

$$\mathbf{C}(\mathbf{q})|_{t=0} = 0, \qquad \dot{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})\Big|_{t=0} = 0 \tag{11}$$

(c) This could be example of constraints drift. Because the index-reduced model imposes $\ddot{\mathbf{C}} = 0$ instead of $\mathbf{C} = 0$, numerical noise tends to accumulate as $\ddot{\mathbf{C}}$ cannot be imposed at perfect accuracy. This numerical noise is integrated twice by the integrator and yields a drift of the constraint. However, the constraint drifts typically occurs over a long simulation time, while this is a simulation over only 10 s. In this Fig., another problem is at play. The initial conditions provided to the simulations are not satisfying the consistency conditions (**??**). As a result, the constraints $\mathbf{C}$ follow the first-order dynamics:

$$\mathbf{C}(\mathbf{q}(t)) = \mathbf{C}(\mathbf{q}(0)) + t\dot{\mathbf{C}}(\mathbf{q}(0), \dot{\mathbf{q}}(0)) \tag{12}$$

We can observe from the trajectory that probably $\mathbf{C}(\mathbf{q}(0)) = 0$ held, as mass 1 seems to be on the surface φ, but $\dot{\mathbf{C}}(\mathbf{q}(0), \dot{\mathbf{q}}(0)) \neq 0$, resulting in mass 1 leaving the surface.

2. **System Identification**

(a) (3 points) Consider the ARX model:

$$y_k + a_1 y_{k-1} = b_0 u_k + e_k \tag{13}$$

and the associated data $y_{0,\dots,N}$ and $u_{0,\dots,N}$ obtained from applying the input sequence $u_{0,\dots,N}$ to the real system, started with $y_{k<0} = 0$.

Write the problem delivering the maximum-likelihood estimator of $\boldsymbol{\theta} = \begin{bmatrix} a_1 & b_0 \end{bmatrix}^\top$ according to the one-step ahead prediction when the additive noise e_k is uncorrelated and uniformly distributed in the interval $[-1,\, 1]$, i.e. the probability density function of e_k is:

$$f(e_k) = \begin{cases} 0.5 & \text{if} \quad e_k \in [-1,\, 1] \\ 0 & \text{if} \quad e_k \notin [-1,\, 1] \end{cases} \tag{14}$$

Describe the solution to the max-likelihood problem, in particular, is the solution unique?

(b) (2 points) Provide the one-step ahead prediction of (13) when the additive error e_k is normal centered and uncorrelated (white noise). Is the least-squares problem linear or nonlinear ? How does the input sequence $u_{0,\dots,N}$ impact the resulting least-square problem?.

(c) (3 points) Consider a linear least-squares problem delivering a parameter estimation $\hat{\boldsymbol{\theta}}$:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \quad \frac{1}{2} \left\| A\boldsymbol{\theta} - \mathbf{y} \right\|^2_{\Sigma^{-1}} \tag{15}$$

1. What are we assuming about the data when using a least-squares fitting problem like (15).
2. Explain in detail the meaning of formula (78) in the Formula Sheet
3. How does the matrix Σ^{-1} enter in the least-squares problem, i.e. how is it used? How should it be selected?

(d) (2 points) Consider the following ARX one-step-ahead predictor

$$\hat{y}(t) = ay(t-1) + bu(t-1). \tag{16}$$

Assume that the following data set is available

$$[y(0),\, y(1)] = [0,\, 1] \tag{17}$$
$$[u(0),\, u(1)] = [1,\, 0] \tag{18}$$

Write the predictor (16) in the linear regression form $\hat{y}(t) = h(t)^T\theta$ and find the least-squares estimate for a, b, given the available data.

Solution:

(a) The one-step ahead predictor reads as:

$$\hat{y}_k = -a_1 y_{k-1} + b_0 u_k \tag{19}$$

and the mismatch between the data and the predictor is given by:

$$e_k = \hat{y}_k - y_k = -y_k - a_1 y_{k-1} + b_0 u_k = \mathbf{d}_k^\top \boldsymbol{\theta} - y_k \tag{20}$$

where $\mathbf{d}_k = \begin{bmatrix} -y_{k-1} & u_k \end{bmatrix}$. According to our model, e_k is uniformly distributed. As the noise

is uncorrelated, the probability density of observing a sequence $e_{0,\dots,N-1}$ is given by:

$$\mathbb{P}\left[e_{0,\dots,N}\right] = \prod_{k=0}^{N-1} f(e_k) = \begin{cases} 0.5 & \text{if} \quad e_{0,\dots,N} \in [-1,\, 1] \\ 0 & \text{if} \quad e_k \notin [-1,\, 1] \text{ for some } k \end{cases} \tag{21}$$

For a given set of parameters $\boldsymbol{\theta} = \begin{bmatrix} a_1 & b_0 \end{bmatrix}^\top$, the probability of obtaining a given noise sequence is then:

$$\mathbb{P}\left[e_k = \mathbf{d}_k^\top \boldsymbol{\theta} - y_k \quad \text{for} \quad k = 0, \dots, N-1 \,\middle|\, \boldsymbol{\theta} \right] = \begin{cases} \frac{1}{2^N} & \text{if} \quad \mathbf{d}_k^\top \boldsymbol{\theta} - y_k \in [-1,\, 1] \quad \forall\, k \\ 0 & \text{otherwise} \end{cases} \tag{22}$$

The maximum-likelihood problem is then:

$$\max_{\boldsymbol{\theta}} \quad \begin{cases} \frac{1}{2^N} & \text{if} \quad \mathbf{d}_k^\top \boldsymbol{\theta} - y_k \in [-1,\, 1] \quad \forall\, k \\ 0 & \text{otherwise} \end{cases} \tag{23}$$

If there exists $\boldsymbol{\theta}$ such that $|\mathbf{d}_k^\top \boldsymbol{\theta} - y_k| \leq 1$ for all k, then the solution set is

$$\boldsymbol{\theta}^\star = \left\{ \boldsymbol{\theta} \quad \text{s.t} \quad |\mathbf{d}_k^\top \boldsymbol{\theta} - y_k| \leq 1 \quad \forall\, k \right\} \tag{24}$$

and the solution is (possibly) not unique. If the solution set $\boldsymbol{\theta}^\star$ is empty, then any $\boldsymbol{\theta}$ has probability 0.

(b) The least-squares problem for the ARX model (13) reads as:

$$\min_{\boldsymbol{\theta}} \quad \frac{1}{2} \| D\boldsymbol{\theta} - \mathbf{y} \|^2 \tag{25}$$

where

$$A = \begin{bmatrix} \mathbf{d}_0^\top \\ \vdots \\ \mathbf{d}_N^\top \end{bmatrix}, \qquad \mathbf{y} = \begin{bmatrix} y_0 \\ \vdots \\ y_N \end{bmatrix} \tag{26}$$

The least-squares problem is linear because the regressor $A\boldsymbol{\theta}$ is linear. The input sequence is the second column of matrix A. This impacts the least-squares problem as the covariance of the resulting parameter estimation

$$\Sigma_{\hat{\boldsymbol{\theta}}} = \left(A^\top \Sigma^{-1} A \right)^{-1} \tag{27}$$

i.e. the input sequence u is directly impacting the "quality" of the estimation.

(c) 1. We are assuming that the data $\mathbf{y}$ are corrupted by normal centred (not necessarily white) noise, and by that only. I.e. the data sequence $\mathbf{y}_{0,\dots,N}$ reads as:

$$\mathbf{y} = A\boldsymbol{\theta}_{\text{true}} + \mathbf{e} \tag{28}$$

where $\boldsymbol{\theta}_{\text{true}}$ is the true model parameters.

2. The expression:

$$\Sigma_{\hat{\boldsymbol{\theta}}} = \left(A^\top \Sigma^{-1} A \right)^{-1} \tag{29}$$

describes the covariance of the parameter estimation, labelled $\Sigma_{\hat{\boldsymbol{\theta}}}$. The formula can be understood as follows. The normal centred noise sequence $\mathbf{e}$ is a vector or random variables.

In that sense, the data we feed into the least-squares problem $\mathbf{y} = A\boldsymbol{\theta}_{\text{true}} + \mathbf{e}$ is also a vector or random variables. Hence the outcome $\hat{\boldsymbol{\theta}}$ of the least-squares problem (15) (see formula (77))

$$\hat{\boldsymbol{\theta}} = \left(A^\top \Sigma^{-1} A\right)^{-1} A^\top \Sigma^{-1} \mathbf{y} \tag{30}$$

is itself a random variable (it is in fact a linear function of the random vector $\mathbf{y}$). Equation (**??**) provides the covariance of that random variable, i.e.

$$\mathbb{E}\left[\left(\hat{\boldsymbol{\theta}} - \boldsymbol{\mu}_{\hat{\boldsymbol{\theta}}}\right)\left(\hat{\boldsymbol{\theta}} - \boldsymbol{\mu}_{\hat{\boldsymbol{\theta}}}\right)^\top\right] \tag{31}$$

where $\boldsymbol{\mu}_{\hat{\boldsymbol{\theta}}} = \mathbb{E}\left[\hat{\boldsymbol{\theta}}\right]$.

3. Problem (15) can also be written as

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \quad \frac{1}{2}\left(A\boldsymbol{\theta} - \mathbf{y}\right)^\top \Sigma^{-1} \left(A\boldsymbol{\theta} - \mathbf{y}\right) \tag{32}$$

hence matrix Σ^{-1} is a "weighting" of the error vector $A\boldsymbol{\theta} - \mathbf{y}$. Matrix Σ ought to be selected as the covariance of the noise $\mathbf{e}$ corrupting the data.

(d) The linear regression form is

$$y(t) = \begin{bmatrix} y(t-1) & u(t-1) \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}, \tag{33}$$

and the least squares estimate of the parameters can be found by $\hat{\theta}_{LS} = (H^T H)^{-1} H^T \mathbf{y}$. Using the available data we have that

$$H = \begin{bmatrix} y(0) & u(0) \\ y(1) & u(1) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y(0) \\ y(1) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{34}$$

Hence the least square estimate is $\hat{\theta}_{LS} = [a \quad b] = [1 \quad 0]$.

3. **Differential-Algebraic and Implicit Differential Equations**

(a) (2 points) Consider the fully implicit DAE:

$$\mathbf{F}\left(\dot{\mathbf{x}},\, \mathbf{x}, \mathbf{z}, \mathbf{u}\right) = 0 \tag{35}$$

where $\mathbf{x} \in \mathbb{R}^{n_\mathbf{x}}$, $\mathbf{u} \in \mathbb{R}^{n_\mathbf{u}}$ and $\mathbf{z} \in \mathbb{R}^{n_\mathbf{z}}$. The model function $\mathbf{F}$ it then in the form:

$$\mathbf{F} : \underbrace{\mathbb{R}^n \times \mathbb{R}^{n_\mathbf{x}} \times \mathbb{R}^{n_\mathbf{z}} \times \mathbb{R}^{n_\mathbf{u}}}_{\text{size of the arguments}} \quad \mapsto \quad \underbrace{\mathbb{R}^m}_{\text{“size of the function output”}} \tag{36}$$

Specify what n and m are. In other words, what is the size of $\dot{\mathbf{x}}$ and what is the size of the vector resulting from evaluating $\mathbf{F}$. Justify!

(b) (2 points) Consider the semi-explicit DAE:

$$\dot{\mathbf{x}} = \mathbf{f}\left(\mathbf{x}, \mathbf{z}, \mathbf{u}\right) \tag{37a}$$
$$0 = \mathbf{g}\left(\mathbf{x}, \mathbf{z}, \mathbf{u}\right) \tag{37b}$$

Rewrite it in the fully-implicit form (35). What would function $\mathbf{F}$ be in this case?

(c) (2 points) Let us do the opposite work, i.e. consider a fully-implicit DAE (35), rewrite it in a semi-explicit form (37). *Hints: don't look for something complicated. You will need to introduce new algebraic variables.*

(d) (3 points) Consider the differential equation:

$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \dot{\mathbf{x}} = \mathbf{x} \tag{38}$$

1. Is (38) an implicit ODE or a DAE? Justify.
2. Show that we can rewrite this equation in a semi-explicit form having 2 algebraic variables and one differential variable. *Hint: you need to do algebraic manipulations and time-differentiations.*

(e) (3 points) Perform an index reduction of the DAE:

$$\dot{\mathbf{x}} = A\mathbf{x} + \mathbf{b}z \tag{39a}$$
$$0 = \mathbf{g}\left(\mathbf{x}\right) \tag{39b}$$

where $z \in \mathbb{R}$ and function $\mathbf{g} : \mathbb{R}^{n_\mathbf{x}} \times \mathbb{R} \mapsto \mathbb{R}^m$. Specify m. Assume that $\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\mathbf{b}$ is full rank. What are the consistency conditions? What condition is needed for the DAE to be of index larger than 2?

(f) (2 points) Consider the semi-explicit DAE:

$$\dot{\mathbf{x}} = \mathbf{f}\left(\mathbf{x}, \mathbf{z}, \mathbf{u}\right) \tag{40a}$$
$$0 = \mathbf{g}\left(\mathbf{x}\right) \tag{40b}$$

Prove that (40) is at least of index 2.

Solution:

(a) 1. Clearly, the size of $\dot{\mathbf{x}}$ is the same as the one of $\mathbf{x}$, i.e. $n = n_\mathbf{x}$.

2. Equation (35) ought to deliver $\dot{\mathbf{x}}$ and $\mathbf{z}$ for given $\mathbf{x}$ and $\mathbf{u}$. That is, equation (35) has $n + n_\mathbf{z} = n_\mathbf{x} + n_\mathbf{z}$ “unknowns”. Function $\mathbf{F}$ is essentially delivering m expressions $\mathbf{F}_1, \ldots, \mathbf{F}_m$ that must be set to zero in order to provide $\dot{\mathbf{x}}$ and $\mathbf{z}$. In order to provide enough “equations” to solve, $\mathbf{F} = 0$ must deliver as many equations as we have unknowns, i.e. $m = n_\mathbf{x} + n_\mathbf{z}$. This can be justified in the light of the Implicit Function Theorem, which requires the

invertibility of $\frac{\nabla \mathbf{F}}{\partial \mathbf{w}}$ is full rank, where $\mathbf{w} = \begin{bmatrix} \dot{\mathbf{x}} \\ \mathbf{z} \end{bmatrix}$. In order to be full invertible, this matrix ought to be square. An alternative acceptable answer is that $m \geq n_{\mathbf{x}} + n_{\mathbf{z}}$, and the set of equations defined by $\mathbf{F} = 0$ has more equations than unknowns.

(b) We can simply write:

$$\mathbf{F}(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{z}, \mathbf{u}) = \begin{bmatrix} \dot{\mathbf{x}} - \mathbf{f}(\mathbf{x}, \mathbf{z}, \mathbf{u}) \\ \mathbf{g}(\mathbf{x}, \mathbf{z}, \mathbf{u}) \end{bmatrix} = 0 \tag{41}$$

(c) This is a bit trickier, but fairly straightforward. We introduce additional algebraic variables $\boldsymbol{\delta}$, and write:

$$\dot{\mathbf{x}} = \boldsymbol{\delta} \tag{42a}$$
$$0 = \mathbf{F}(\boldsymbol{\delta}, \mathbf{x}, \mathbf{z}, \mathbf{u}) \tag{42b}$$

This equation is a semi-explicit DAE.

(d) 1. Since matrix E is rank deficient (3^{rd} line is the sum of the 1^{st} and 2^{nd} lines), (38) is a DAE

2. We observe that DAE (38) reads as:

$$\dot{\mathbf{x}}_1 + \dot{\mathbf{x}}_2 = \mathbf{x}_1 \tag{43a}$$
$$\dot{\mathbf{x}}_3 = \mathbf{x}_2 \tag{43b}$$
$$\dot{\mathbf{x}}_1 + \dot{\mathbf{x}}_2 + \dot{\mathbf{x}}_3 = \mathbf{x}_3 \tag{43c}$$

Subtracting (**??**) and (**??**) to (**??**), we get:

$$\dot{\mathbf{x}}_1 + \dot{\mathbf{x}}_2 = \mathbf{x}_1 \tag{44a}$$
$$\dot{\mathbf{x}}_3 = \mathbf{x}_2 \tag{44b}$$
$$0 = \mathbf{x}_1 + \mathbf{x}_2 - \mathbf{x}_3 \tag{44c}$$

A time-differentiation of (**??**) yields:

$$\dot{\mathbf{x}}_1 + \dot{\mathbf{x}}_2 = \mathbf{x}_1 \tag{45a}$$
$$\dot{\mathbf{x}}_3 = \mathbf{x}_2 \tag{45b}$$
$$0 = \mathbf{x}_1 + \mathbf{x}_2 - \mathbf{x}_3 \tag{45c}$$
$$\dot{\mathbf{x}}_1 + \dot{\mathbf{x}}_2 - \dot{\mathbf{x}}_3 = 0 \tag{45d}$$

We then do (**??**) - (**??**) + (**??**) to get:

$$\dot{\mathbf{x}}_1 + \dot{\mathbf{x}}_2 = \mathbf{x}_1 \tag{46a}$$
$$\dot{\mathbf{x}}_3 = \mathbf{x}_2 \tag{46b}$$
$$0 = \mathbf{x}_1 + \mathbf{x}_2 - \mathbf{x}_3 \tag{46c}$$
$$0 = \mathbf{x}_1 - \mathbf{x}_2 \tag{46d}$$

A time-differentiation of (**??**) yields:

$$\dot{\mathbf{x}}_1 + \dot{\mathbf{x}}_2 = \mathbf{x}_1 \tag{47a}$$
$$\dot{\mathbf{x}}_3 = \mathbf{x}_2 \tag{47b}$$
$$0 = \mathbf{x}_1 + \mathbf{x}_2 - \mathbf{x}_3 \tag{47c}$$
$$0 = \mathbf{x}_1 - \mathbf{x}_2 \tag{47d}$$
$$\dot{\mathbf{x}}_1 - \dot{\mathbf{x}}_2 = 0 \tag{47e}$$

We then do (**??**)+(**??**) to get the semi-explicit DAE:

$$\dot{\mathbf{x}}_1 = \frac{1}{2}\mathbf{x}_1 \tag{48a}$$
$$0 = \mathbf{x}_1 + \mathbf{x}_2 - \mathbf{x}_3 \tag{48b}$$
$$0 = \mathbf{x}_1 - \mathbf{x}_2 \tag{48c}$$

(e) We are dealing with a semi-explicit DAE, hence the index reduction requires time-differentiations of the algebraic equation (39b). The first step of the index reduction reads as:

$$\dot{\mathbf{x}} = A\mathbf{x} + \mathbf{b}z \tag{49a}$$
$$0 = \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}}\dot{\mathbf{x}} = \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}}(A\mathbf{x} + \mathbf{b}z) \tag{49b}$$

We note that $\frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}}\mathbf{b}$ is scalar here, and different than zero since it is full rank. It follows that (**??**) is of index 1. The consistency condition is simply:

$$\mathbf{g}(\mathbf{x}(0)) = 0 \tag{50}$$

DAE (39) is of index 2 if $\frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}}\mathbf{b} \neq 0$, if $\frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}}\mathbf{b} = 0$, then it is of index at least 3.

(f) In order for DAE (40) to be of index 1, it would require $\frac{\partial \mathbf{g}}{\partial \mathbf{z}}$ to be full rank. However, since $\mathbf{g}$ is not a function of $\mathbf{z}$, $\frac{\partial \mathbf{g}}{\partial \mathbf{z}} = 0$ holds, hence $\frac{\partial \mathbf{g}}{\partial \mathbf{z}}$ is rank deficient. It therefore has to be of index higher than 1.

4. **Newton** The Newton methods aims at solving a set of equation $\mathbf{r}(\mathbf{x}) = 0$ numerically. To that end, iterates the recursion:

$$\frac{\partial \mathbf{r}(\mathbf{x})}{\partial \mathbf{x}} \Delta \mathbf{x} + \mathbf{r}(\mathbf{x}) = 0 \tag{51a}$$

$$\mathbf{x} \leftarrow \mathbf{x} + \alpha \Delta \mathbf{x} \tag{51b}$$

where $\alpha \in]0,\, 1]$ is the step-size.

(a) (2 points) Explain in words what condition(s) is (are) required for Newton to converge with $\alpha = 1$.

(b) (2 points) The local convergence rate of an exact, full-step Newton method can be summarized as:

$$\|\mathbf{x}_+ - \mathbf{x}_\star\| \leq c \|\mathbf{x} - \mathbf{x}_\star\|^2 \tag{52}$$

where $\mathbf{x}_\star$ is a solution of $\mathbf{r}(\mathbf{x}_\star)$. What is the meaning of this formula? When does it (doesn't it) occur?

Solution:

(a) Full Newton steps are guaranteed to converge in a neighborhood of a solution only. The "size" of that neighborhood depends on how nonlinear $\mathbf{r}(\mathbf{x})$ is, and the Jacobian $\frac{\partial \mathbf{r}(\mathbf{x})}{\partial \mathbf{x}}$ must be full rank throughout this neighborhood.

(b) This formula states that the exact, full-step Newton iteration converges quadratically to a solution. That is, the number of accurate digits in the $\mathbf{x}$ is doubled at every iteration. Achieving the quadratic contraction rate requires basically what is stated in the question, namely:

- Exact Newton steps, i.e. an exact Jacobian $\frac{\partial \mathbf{r}(\mathbf{x})}{\partial \mathbf{x}}$ is used and system (51a) is solved to machine precision.
- Full steps are taken, i.e. $\alpha = 1$ throughout the iterations.
- The quadratic convergence rate is local, i.e. it occurs in a neighborhood of the solution $\mathbf{x}_\star$.

5. **Simulation**

(a) (4 points) Write a pseudo-code (algorithm) that would deploy an IRK scheme for an implicit DAE autonomous (no input)

$$\mathbf{F}\left(\dot{\mathbf{x}},\, \mathbf{x}, \mathbf{z}\right) = 0 \tag{53}$$

Be specific enough that someone could code it without knowing what the algorithm is about.

(b) (2 points) What is the maximum order (for a given number of stages s) that an IRK method can achieve? What one needs to do to achieve that order?

(c) (2 points) Why are IRK methods with a large number of stages not favoured in practice? One point is given for a short answer, an extra point for a more detailed discussion.

(d) (2 points) Why are high-order explicit RK methods often not the optimal choice? (the answer can be in the form of a discussion, without formula)

(e) (3 points) Write the IRK equations for a semi-explicit DAE:

$$\dot{\mathbf{x}} = \mathbf{f}\left(\mathbf{x}, \mathbf{z}, \mathbf{u}\right) \tag{54a}$$
$$0 = \mathbf{g}\left(\mathbf{x}, \mathbf{z}, \mathbf{u}\right) \tag{54b}$$

and specify what the Jacobian in the resulting Newton step looks like (no need for explicit nor detailed expressions, just specify what function is differentiating with respect to which variables. Keep your answer simple.)

Solution:

(a) The pseudo-code will look like

Algorithm: Integration of implicit ODE

Input: $\mathbf{x}_0$, α and Δt
Set $K = 0$
for $k = 0 : N-1$ **do**
  while $\|\mathbf{r}\left(\mathbf{K}, \mathbf{x}_k, \mathbf{u}(.)\right)\| > \text{tol}$ **do**
    Evaluate:

$$\mathbf{r}\left(\mathbf{K}, \mathbf{z}, \mathbf{x}_k\right) = \begin{bmatrix} \mathbf{F}\left(\mathbf{K}_1, \mathbf{x}_k + \Delta t \sum_{i=1}^{s} a_{1i}\mathbf{K}_i,\, \mathbf{z}_1\right) \\ \vdots \\ \mathbf{F}\left(\mathbf{K}_s, \mathbf{x}_k + \Delta t \sum_{i=1}^{s} a_{si}\mathbf{K}_i,\, \mathbf{z}_s\right) \end{bmatrix} = 0$$

    and

$$M = \begin{bmatrix} \frac{\partial \mathbf{r}(\mathbf{K},\mathbf{z},\mathbf{x}_k)}{\partial \mathbf{K}} & \frac{\partial \mathbf{r}(\mathbf{K},\mathbf{z},\mathbf{x}_k)}{\partial \mathbf{z}} \end{bmatrix}$$

    Take the Newton step

$$\begin{bmatrix} \mathbf{K} \\ \mathbf{z} \end{bmatrix} \leftarrow \begin{bmatrix} \mathbf{K} \\ \mathbf{z} \end{bmatrix} - \alpha M^{-1}\mathbf{r} \tag{55}$$

  Take the integrator step:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \sum_{i=1}^{s} b_i \mathbf{K}_i \tag{56}$$

return $\mathbf{x}_{0,\ldots N}$

Obs: pseudo-code adequately tailored to implicit DAEs will also be counted as

correct.

(b) The family of IRK methods includes the Gauss-Legendre collocation methods (this is easy to verify from the equations provided in the appendix), which achieve an order up to $2s$. That is the maximum order that IRK methods can achieve for a given number of stages s. Gauss-Legendre collocation schemes yield a very specific Butcher tableau (a, b, c) to be used in the IRK scheme. The order $2s$ is achieved only if this specific Butcher tableau is used.

(c) IRK methods suffer from the complexity of factorizing the Jacobian matrices involved in the Newton method underlying the integration scheme. A large number of stages provides a very high order, but requires also a heavy linear algebra. To detail this statement, we ought to specify that IRK methods need to solve linear systems of the form (**??**). The matrix factorization is dominating the computations involved in deploying an IRK scheme and are in the order of the cube of the size of the matrix, i.e. $\mathcal{O}\left(n^3 s^3\right)$ (where n is the number of states involved in the ODE). This complexity is "to be paid" at every time step of the integrator, i.e. $N = \frac{t_f}{\Delta t}$, i.e. the overall complexity of the integration scheme is dominated by $\mathcal{O}\left(\Delta t^{-1} n^3 s^3\right)$. The ratio complexity-order is then $\mathcal{O}\left(\Delta t^{-1} n^3 s^3\right)$ versus $\mathcal{O}\left(\Delta t^{2s}\right)$. A detailed complexity analysis shows then that this ratio is unfavorable for s large, and is typically best at $s \in \{2, 3\}$. If more accuracy is needed, reducing the step size Δt is then usually preferable than increasing the order beyond 3.

(d) For a semi-explicit DAE, the IRK equations read as:

$$\mathbf{r}\left(\mathbf{K}, \mathbf{z}, \mathbf{x}_k, \mathbf{u}(.)\right) = \begin{bmatrix} \mathbf{K}_1 - \mathbf{f}\left(\mathbf{x}_k + \Delta t \sum_{i=1}^{s} a_{1i}\mathbf{K}_i,\, \mathbf{z}_1,\, \mathbf{u}(t_k + c_1 \Delta t)\right) \\ \mathbf{g}\left(\mathbf{x}_k + \Delta t \sum_{i=1}^{s} a_{1i}\mathbf{K}_i,\, \mathbf{z}_1,\, \mathbf{u}(t_k + c_1 \Delta t)\right) \\ \vdots \\ \mathbf{K}_s - \mathbf{f}\left(\mathbf{x}_k + \Delta t \sum_{i=1}^{s} a_{si}\mathbf{K}_i,\, \mathbf{z}_s,\, \mathbf{u}(t_k + c_s \Delta t)\right) \\ \mathbf{g}\left(\mathbf{x}_k + \Delta t \sum_{i=1}^{s} a_{si}\mathbf{K}_i,\, \mathbf{z}_s,\, \mathbf{u}(t_k + c_s \Delta t)\right) \end{bmatrix} = 0 \tag{57}$$

and have to be solved with respect to the variables $\mathbf{K}_{1,\ldots,s}$ and $\mathbf{z}_{1,\ldots,s}$. If we gather all these variables in a single vector, e.g.

$$\mathbf{w} = \begin{bmatrix} \mathbf{K}_1 \\ \mathbf{z}_1 \\ \vdots \\ \mathbf{K}_s \\ \mathbf{z}_s \end{bmatrix} \tag{58}$$

then the Jacobian we need to use in the Newton step will be taken as:

$$\frac{\partial \mathbf{r}}{\partial \mathbf{w}} \tag{59}$$

Appendix: some possibly useful formula

- Lagrange mechanics is built on the equations:

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}} - \frac{\partial \mathcal{L}}{\partial \mathbf{q}} = \mathbf{Q}, \qquad \mathcal{L}\left(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{z}\right) = \mathcal{T} - \mathcal{V} - \mathbf{z}^\top \mathbf{C}, \qquad \mathbf{C} = 0, \qquad \langle \delta\mathbf{q},\, \mathbf{Q} \rangle = \delta W,\, \forall\, \delta\mathbf{q} \tag{60}$$

  The kinetic and potential energy of a point mass are given by:

$$\mathcal{T} = \frac{1}{2} m \dot{\mathbf{p}}^\top \dot{\mathbf{p}}, \qquad \mathcal{V} = mg\mathbf{p}_3 \tag{61}$$

  respectively, where $\mathbf{p} \in \mathbb{R}^3$ is the position of the mass in a cartesian reference frame having the third coordinate as the vertical axis pointing up. The generalized forces are identical to the external forces applied to a point mass if the position of that point is expressed in cartesian coordinates in the generalized coordinates $\mathbf{q}$.

- In the case $\mathcal{T} = \frac{1}{2} m \dot{\mathbf{q}}^\top W \dot{\mathbf{q}}$ with W constant $\mathcal{V} = \mathcal{V}\left(\mathbf{q}\right)$ and $\mathbf{C} = \mathbf{C}\left(\mathbf{q}\right)$, the Lagrange equations simplify to the dynamics in the semi-explicit index-3 DAE form:

$$\dot{\mathbf{p}} = \mathbf{v} \tag{62a}$$

$$W\dot{\mathbf{v}} + \frac{\partial \mathbf{C}}{\partial \mathbf{q}}^\top \mathbf{z} = \mathbf{Q} - \frac{\partial \mathcal{V}}{\partial \mathbf{q}}^\top \tag{62b}$$

$$0 = \mathbf{C}\left(\mathbf{q}\right) \tag{62c}$$

- The Implicit Function Theorem (IFT) guarantees that a nonlinear set of equations

$$\mathbf{r}\left(\mathbf{y}, \mathbf{z}\right) = 0 \tag{63}$$

  "can be solved" in terms of $\mathbf{z}$ for a given $\mathbf{y}$ iff the Jacobian $\frac{\partial \mathbf{r}(\mathbf{y},\mathbf{z})}{\partial \mathbf{z}}$ is full rank at the solution. More specifically, it guarantees that there is a function $\phi\left(\mathbf{y}\right)$ such that

$$\mathbf{r}\left(\mathbf{y}, \phi\left(\mathbf{y}\right)\right) = 0 \tag{64}$$

  holds in the neighborhood of the point $\mathbf{y}$ where the Jacobian is evaluated. Furthermore, the IFT specifies that:

$$\frac{\partial \mathbf{z}}{\partial \mathbf{y}} = -\frac{\partial \mathbf{r}}{\partial \mathbf{z}}^{-1} \frac{\partial \mathbf{r}}{\partial \mathbf{y}} \tag{65}$$

- For solving a problem $\mathbf{r}\left(\mathbf{x}\right) = 0$, Newton iterates:

$$\mathbf{x} \leftarrow \mathbf{x} - \alpha \frac{\partial \mathbf{r}}{\partial \mathbf{x}}^{-1} \mathbf{r} \tag{66}$$

  until $\mathbf{r}\left(\mathbf{x}\right) \approx 0$ where $\alpha \in [0, 1]$

- Runge-Kutta methods are described by:

$$\begin{array}{c|ccc} c_1 & a_{11} & \dots & a_{1s} \\ \vdots & \vdots & & \vdots \\ c_s & a_{s1} & \dots & a_{ss} \\ \hline & b_1 & \dots & b_s \end{array}$$

$$\mathbf{K}_j = \mathbf{f}\left(\mathbf{x}_k + \Delta t \sum_{i=1}^{s} a_{ji}\mathbf{K}_i,\, \mathbf{u}\left(t_k + c_j \Delta t\right)\right), \quad j = 1, \dots, s \tag{67a}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \sum_{i=1}^{s} b_i \mathbf{K}_i \tag{67b}$$

- For ERK methods, the relationship between the (minimum) number of stages s to the order o is given by:

s	1	2	3	4	6	7	9	11	...
o	1	2	3	4	5	6	7	8	...

Table 1: Stage to order of ERK methods

- Collocation methods use:

$$\dot{\mathbf{x}}(t_k + \Delta t \cdot \tau) \approx \dot{\hat{\mathbf{x}}}(t_k + \Delta t \cdot \tau) = \sum_{i=1}^{s} \mathbf{K}_i \ell_i(\tau), \quad \tau \in [0, 1] \tag{68}$$

$$\mathbf{x}(t_k + \Delta t \cdot \tau) \approx \hat{\mathbf{x}}(t_k + \Delta t \cdot \tau) = \mathbf{x}_k + \Delta t \sum_{i=1}^{s} \mathbf{K}_i L_i(\tau) \tag{69}$$

where the Lagrange polynomials are given by:

$$\ell_i(\tau) = \prod_{j=1, j\neq i}^{s} \frac{\tau - \tau_j}{\tau_i - \tau_j}, \quad \text{and} \quad L_i(\tau) = \int_0^{\tau} \ell_i(\xi) \mathrm{d}\xi \tag{70}$$

The Lagrange polynomials satisfy the conditions of

$$\text{Orthogonality:} \quad \int_0^1 \ell_i(\tau)\ell_j(\tau)\,\mathrm{d}\tau = 0 \quad \text{for} \quad i \neq j \tag{71a}$$

$$\text{Punctuality:} \quad \ell_i(\tau_j) = \begin{cases} 1 & \text{if} \quad j = i \\ 0 & \text{if} \quad j \neq i \end{cases} \tag{71b}$$

and enforce the collocation equations (for $j = 1, \ldots, s$):

$$\dot{\hat{\mathbf{x}}}(t_k + \Delta t \cdot \tau_j) = \mathbf{f}\left(\hat{\mathbf{x}}(t_k + \Delta t \cdot \tau_j),\, \mathbf{u}\left(t_k + \Delta t \cdot \tau_j\right)\right), \qquad \text{in the explicit ODE case} \tag{72a}$$

$$\mathbf{F}\left(\dot{\hat{\mathbf{x}}}(t_k + \Delta t \cdot \tau_j), \hat{\mathbf{x}}(t_k + \Delta t \cdot \tau_j),\, \mathbf{u}\left(t_k + \Delta t \cdot \tau_j\right)\right) = 0, \qquad \text{in the implicit ODE case} \tag{72b}$$

$$\mathbf{F}\left(\dot{\hat{\mathbf{x}}}(t_k + \Delta t \cdot \tau_j), \hat{\mathbf{z}}_j, \hat{\mathbf{x}}(t_k + \Delta t \cdot \tau_j),\, \mathbf{u}\left(t_k + \Delta t \cdot \tau_j\right)\right) = 0, \qquad \text{in the fully-implicit DAE case} \tag{72c}$$

- Gauss-Legendre collocation methods select the set of points $\tau_{1,\ldots,s}$ as the zeros of the (shifted) Legrendre polynomial:

$$P_s(\tau) = \frac{1}{s!} \frac{\mathrm{d}^s}{\mathrm{d}\tau^s} \left[\left(\tau^2 - \tau\right)^s\right] \tag{73}$$

They achieve the order $\|\mathbf{x}_N - \mathbf{x}(t_\mathrm{f})\| = \mathcal{O}\left(\Delta t^{2s}\right)$.

- Maximum-likelihood estimation is based on

$$\max_{\boldsymbol{\theta}} \quad \mathbb{P}\left[e_k = y_k - \hat{y}_k \quad \text{for} \quad k = 1, \ldots, N \mid \boldsymbol{\theta}\right] \tag{74}$$

If the noise sequence is uncorrelated, then

$$\mathbb{P}\left[e_k = y_k - \hat{y}_k \quad \text{for} \quad k = 0, \ldots, N \mid \boldsymbol{\theta}\right] = \prod_{k=1}^{N} \mathbb{P}\left[e_k = y_k - \hat{y}_k \mid \boldsymbol{\theta}\right] \tag{75}$$

- The solution of a linear least-squares problem

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{2} \|A\boldsymbol{\theta} - \mathbf{y}\|^2_{\Sigma_e^{-1}} \tag{76}$$

reads as:

$$\hat{\boldsymbol{\theta}} = \left(A^\top \Sigma_e^{-1} A\right)^{-1} A^\top \Sigma_e^{-1} \mathbf{y} \tag{77}$$

and the covariance of the parameter estimation based is given by the formula:

$$\Sigma_{\hat{\boldsymbol{\theta}}} = \left(A^\top \Sigma_e^{-1} A\right)^{-1} \tag{78}$$

- In system identification, given the a plant $G(z)$ and a noise $H(z)$ model description, the one-step-ahead predictor $\hat{y}(k|k-1)$ can be retrieved with

$$H(z)\hat{y}(z) = G(z)u(z) + (H(z) - 1)y(z) \tag{79}$$

- The Gauss-Newton approximation in an optimization problem

$$\min_{\mathbf{x}} \quad J(\mathbf{x}) = \frac{1}{2} \|\mathbf{R}(\mathbf{x})\|^2 \tag{80}$$

uses the approximation:

$$\frac{\partial^2 J}{\partial \mathbf{x}^2} \approx \frac{\partial R}{\partial \mathbf{x}}^\top \frac{\partial R}{\partial \mathbf{x}} \tag{81}$$

- The solution to an LTI system $\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$ is given by:

$$\mathbf{x}(t) = e^{At}\mathbf{x}(0) + \int_0^t e^{A(t-\tau)} B\mathbf{u}(\tau) \mathrm{d}\tau \tag{82}$$

and the transformation state-space to transfer function is given by:

$$G(s) = C\left(sI - A\right)^{-1} B + D \tag{83}$$