MOS ESS101
October 2017

This exam contains 14 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 26 points, and the top grade at 38 points. These limits may be lowered depending on the outcome of the exam.

Problem	Points	Score
1	10	
2	10	
3	8	
4	4	
5	12	
Total:	44	

Best of luck to all !!

1. **Lagrange modelling** Consider a hanging chain made of N dimensionless balls of mass m (see Fig. 1 for an illustration). The balls are connected to each other by rigid cables of length L, and the two balls at the extremities of the chain are connected to the two points $\mathbf{p}_0(t)$, $\mathbf{p}_{N+1}(t) \in \mathbb{R}^3$ with rigid cables of length L as well.

   (a) (4 points) Write down the model equations of the chain (in 3D) in the form of a semi-explicit index-3 DAE. Note that the end points can move in time!

   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) (1 point) How can one assess in a simulation whether the cables linking the masses are always under tension?

   (d) (2 points) After running a simulation of the index-1 DAE equations, we observe that the constraints $\mathbf{C}(\mathbf{q})$ have the behavior depicted in Fig. 2. Explain this graph.

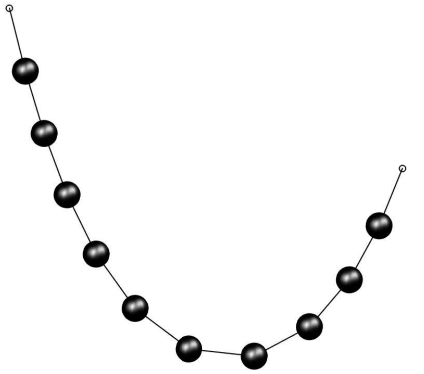

Figure 1: Hanging chain with $N = 10$.

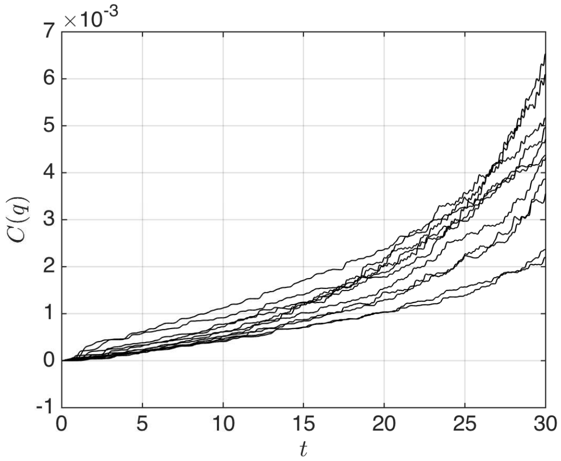

Figure 2: Constraints from simulation

Solution:

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

$$\mathbf{q} = \begin{bmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_N \end{bmatrix} \in \mathbb{R}^{3N} \tag{1}$$

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

$$\mathcal{T} = \frac{1}{2} m \sum_{i=1}^{N} \dot{\mathbf{p}}_i^\top \dot{\mathbf{p}}_i = \frac{1}{2} m \dot{\mathbf{q}}^\top \dot{\mathbf{q}}, \qquad \mathcal{V} = mg \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \sum_{i=1}^{N} \mathbf{p}_i = mg\mathbf{a}^\top \mathbf{q} \tag{2}$$

where $\mathbf{a}^\top = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & \dots \end{bmatrix}$, and has the constraint functions

$$\mathbf{C}(\mathbf{q},t) = \begin{bmatrix} C_1 \\ \vdots \\ C_{N+1} \end{bmatrix}, \qquad C_k(\mathbf{q},t) = \frac{1}{2}\left(\left\|\mathbf{p}_k - \mathbf{p}_{k-1}\right\|^2 - L^2\right) \tag{3}$$

for $k = 1, \dots, N+1$. The Lagrange function then reads as:

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

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:

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

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

$$0 = \mathbf{C}\left(\mathbf{q}, t\right) \tag{6c}$$

This result could also be obtained directly from (48) by observing that $W\left(\mathbf{q}\right) = I$. We can additionally further explicitly provide (though this is not required from the question):

$$\frac{\partial \mathbf{C}}{\partial \mathbf{q}}^\top = \begin{bmatrix} \mathbf{p}_1 - \mathbf{p}_0 & \mathbf{p}_1 - \mathbf{p}_2 & 0 & 0 & \dots & & \\ 0 & \mathbf{p}_2 - \mathbf{p}_1 & \mathbf{p}_2 - \mathbf{p}_3 & 0 & \dots & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \\ & & \vdots & 0 & \mathbf{p}_{N-1} - \mathbf{p}_{N-2} & \mathbf{p}_{N-1} - \mathbf{p}_N & 0 \\ & & \vdots & 0 & 0 & \mathbf{p}_N - \mathbf{p}_{N-1} & \mathbf{p}_N - \mathbf{p}_{N+1} \end{bmatrix} \tag{7}$$

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

$$\dot{\mathbf{C}} = \begin{bmatrix} \left(\mathbf{p}_1 - \mathbf{p}_0\right)^\top \left(\dot{\mathbf{p}}_1 - \dot{\mathbf{p}}_0\right) \\ \vdots \\ \left(\mathbf{p}_{N+1} - \mathbf{p}_N\right)^\top \left(\dot{\mathbf{p}}_{N+1} - \dot{\mathbf{p}}_N\right) \end{bmatrix} = 0 \tag{8a}$$

$$\ddot{\mathbf{C}} = \underbrace{\begin{bmatrix} \left(\mathbf{p}_1 - \mathbf{p}_0\right)^\top \left(\ddot{\mathbf{p}}_1 - \ddot{\mathbf{p}}_0\right) \\ \vdots \\ \left(\mathbf{p}_{N+1} - \mathbf{p}_N\right)^\top \left(\ddot{\mathbf{p}}_{N+1} - \ddot{\mathbf{p}}_N\right) \end{bmatrix}}_{=\frac{\partial \mathbf{C}}{\partial \mathbf{q}}\ddot{\mathbf{q}}} + \begin{bmatrix} \left(\dot{\mathbf{p}}_1 - \dot{\mathbf{p}}_0\right)^\top \left(\dot{\mathbf{p}}_1 - \dot{\mathbf{p}}_0\right) \\ \vdots \\ \left(\dot{\mathbf{p}}_{N+1} - \dot{\mathbf{p}}_N\right)^\top \left(\dot{\mathbf{p}}_{N+1} - \dot{\mathbf{p}}_N\right) \end{bmatrix} = 0 \tag{8b}$$

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

$$\dot{\mathbf{q}} = \mathbf{v} \tag{9a}$$

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

$$0 = \begin{bmatrix} \left(\mathbf{p}_1 - \mathbf{p}_0\right)^\top \left(\ddot{\mathbf{p}}_1 - \ddot{\mathbf{p}}_0\right) \\ \vdots \\ \left(\mathbf{p}_{N+1} - \mathbf{p}_N\right)^\top \left(\ddot{\mathbf{p}}_{N+1} - \ddot{\mathbf{p}}_N\right) \end{bmatrix} + \begin{bmatrix} \left(\dot{\mathbf{p}}_1 - \dot{\mathbf{p}}_0\right)^\top \left(\dot{\mathbf{p}}_1 - \dot{\mathbf{p}}_0\right) \\ \vdots \\ \left(\dot{\mathbf{p}}_{N+1} - \dot{\mathbf{p}}_N\right)^\top \left(\dot{\mathbf{p}}_{N+1} - \dot{\mathbf{p}}_N\right) \end{bmatrix} \tag{9c}$$

or in a complete, matrix form:

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

$$\begin{bmatrix} mI & \frac{\partial \mathbf{C}}{\partial \mathbf{q}} \\ \frac{\partial \mathbf{C}}{\partial \mathbf{q}}^\top & 0 \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}} \\ \mathbf{z} \end{bmatrix} = - \begin{bmatrix} mg\mathbf{a} \\ \left(\dot{\mathbf{p}}_1 - \dot{\mathbf{p}}_0\right)^\top \left(\dot{\mathbf{p}}_1 - \dot{\mathbf{p}}_0\right) - \left(\mathbf{p}_1 - \mathbf{p}_0\right)^\top \ddot{\mathbf{p}}_0 \\ \left(\dot{\mathbf{p}}_2 - \dot{\mathbf{p}}_1\right)^\top \left(\dot{\mathbf{p}}_2 - \dot{\mathbf{p}}_1\right) \\ \vdots \\ \left(\dot{\mathbf{p}}_N - \dot{\mathbf{p}}_{N-1}\right)^\top \left(\dot{\mathbf{p}}_N - \dot{\mathbf{p}}_{N-1}\right) \\ \left(\dot{\mathbf{p}}_{N+1} - \dot{\mathbf{p}}_N\right)^\top \left(\dot{\mathbf{p}}_{N+1} - \dot{\mathbf{p}}_N\right) + \left(\mathbf{p}_{N+1} - \mathbf{p}_N\right)^\top \ddot{\mathbf{p}}_{N+1} \end{bmatrix} \tag{10b}$$

In this model, $\ddot{\mathbf{p}}_0$ and $\ddot{\mathbf{p}}_{N+1}$ act as inputs. For simulation purposes, one would typically introduce $\mathbf{p}_0, \mathbf{p}_{N+1}, \dot{\mathbf{p}}_0, \dot{\mathbf{p}}_{N+1}$ in the state-space. The consistency conditions require:

$$\mathbf{C}\left(\mathbf{q}, \mathbf{p}_0, \mathbf{p}_{N+1}\right)\Big|_{t=0} = 0, \qquad \dot{\mathbf{C}}\left(\mathbf{q}, \mathbf{p}_0, \mathbf{p}_{N+1}, \dot{\mathbf{q}}, \dot{\mathbf{p}}_0, \dot{\mathbf{p}}_{N+1}\right)\Big|_{t=0} = 0 \tag{11}$$

(c) The tension in the cables is reflected by the variables $\mathbf{z} \in \mathbb{R}^{N+1}$. The sign of these variables specifies if the cables are "pushing" or "pulling". An inspection of (9b) allows us to conclude that $\mathbf{z} > 0$ corresponds to the cables "pulling" (the question did not require this statement). One could inspect the outcome of a simulation and decide whether the cables went slack during the motion.

(d) This is a case of constraint drift common when simulating index-reduced DAEs. 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. The drift is typically following a second-order polynomial in time (since the noise is integrated twice).

2. **System Identification**

(a) (4 points) Consider the ARX model:

$$y_k + a_1 y_{k-1} + a_2 y_{k-2} = b_0 u_k + e_k \tag{12}$$

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 $\boldsymbol{\theta} = \begin{bmatrix} a_1 & a_2 & b_0 \end{bmatrix}^\top$ according to the one-step ahead prediction when the additive noise e_k is uncorrelated and uniformly distributed in the interval $[-0.5,\, 0.5]$, i.e. the probability density function of e_k is:

$$f(e_k) = \begin{cases} 1 & \text{if} \quad e_k \in [-0.5,\, 0.5] \\ 0 & \text{if} \quad e_k \notin [-0.5,\, 0.5] \end{cases} \tag{13}$$

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

(b) (2 points) Write the least-squares problem associated to the one-step ahead prediction of (12) when the additive error e_k as normal centered and uncorrelated (white noise). Is it a linear or nonlinear least-squares problem? Is $u_k = 0$ for all k an admissible input? Explain why.

(c) (2 points) Consider a linear least-squares problem

$$\min_{\boldsymbol{\theta}} \quad \frac{1}{2} \left\| A\boldsymbol{\theta} - \mathbf{y} \right\|^2_{\Sigma^{-1}} \tag{14}$$

What are/is the condition(s) for the problem to yield a solution? What does this condition mean in System Identification?

(d) (2 points) Consider the following system generating the data

$$y_k + 0.5 y_{k-1} = u_{k-1} + 1.5 u_{k-2} + e_k - 0.2 e_{k-1} \tag{15}$$

Find the plant model $G(z)$ and the noise model $H(z)$. Find the one-step-ahead predictor for the system.

Solution:

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

$$\hat{y}_k = -a_1 y_{k-1} - a_2 y_{k-2} + b_0 u_k \tag{16}$$

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

$$e_k = \hat{y}_k - y_k = -y_k - a_1 y_{k-1} - a_2 y_{k-2} + b_0 u_k = \mathbf{d}_k^\top \boldsymbol{\theta} - y_k \tag{17}$$

where $\mathbf{d}_k = \begin{bmatrix} -y_{k-1} & -y_{k-2} & u_k \end{bmatrix}$. According to our model, e_k is uniformly distributed. As the noise is uncorrelated, the probability density of observing a sequence $e_{0,\dots,N}$ is given by:

$$\mathbb{P}\left[e_{0,\dots,N}\right] = \prod_{k=0}^{N} f(e_k) = \begin{cases} 1 & \text{if} \quad e_{0,\dots,N} \in [-0.5,\, 0.5] \\ 0 & \text{if} \quad e_k \notin [-0.5,\, 0.5] \text{ for some } k \end{cases} \tag{18}$$

For a given set of parameters $\boldsymbol{\theta} = \begin{bmatrix} a_1 & a_2 & b_0 \end{bmatrix}^\top$, the probability of obtaining a given noise sequence is then:

$$\mathbb{P}\left[e_k = \mathbf{d}_k^\top \boldsymbol{\theta} - y_k \quad \text{for} \quad k = 0, \dots, N \,\middle|\, \boldsymbol{\theta} \right] = \begin{cases} 1 & \text{if} \quad \mathbf{d}_k^\top \boldsymbol{\theta} - y_k \in [-0.5,\, 0.5] \quad \forall\, k \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

The maximum-likelihood problem is then:

$$\max_{\boldsymbol{\theta}} \quad \begin{cases} 1 & \text{if} \quad \mathbf{d}_k^\top \boldsymbol{\theta} - y_k \in [-0.5,\, 0.5] \quad \forall\, k \\ 0 & \qquad\qquad \text{otherwise} \end{cases} \tag{20}$$

If there exists $\boldsymbol{\theta}$ such that $|\mathbf{d}_k^\top \boldsymbol{\theta} - y_k| \leq 0.5$ for all k, then the solution set is

$$\boldsymbol{\theta}^\star = \Big\{ \boldsymbol{\theta} \quad \text{s.t} \quad |\mathbf{d}_k^\top \boldsymbol{\theta} - y_k| \leq 0.5 \quad \forall\, k \Big\} \tag{21}$$

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 (12) reads as:

$$\min_{\boldsymbol{\theta}} \quad \frac{1}{2}\|D\boldsymbol{\theta} - \mathbf{y}\|^2 \tag{22}$$

where

$$D = \begin{bmatrix} \mathbf{d}_0^\top \\ \vdots \\ \mathbf{d}_N^\top \end{bmatrix}, \qquad \mathbf{y} = \begin{bmatrix} y_0 \\ \vdots \\ y_N \end{bmatrix} \tag{23}$$

The least-squares problem is linear because the regressor $D\boldsymbol{\theta}$ is linear.

(c) Since the solution to the linear least-squares problem is given by (63), it requires that $A^\top \Sigma_{\mathrm{e}}^{-1} A$ is full rank. One can additionally see from (64) that if the matrix $A^\top \Sigma_{\mathrm{e}}^{-1} A$ is close to being rank deficient (very small eigenvalues), then the covariance of the parameter estimation is large, and therefore the parameters are poorly estimated. In terms of system identification, the requirement that matrix $A^\top \Sigma_{\mathrm{e}}^{-1} A$ is full rank (and preferably does not have very small eigenvalues) boils down to having a sufficiently large amount of data, and data that are rich enough to avoid a “poor” matrix $A^\top \Sigma_{\mathrm{e}}^{-1} A$ (low eigenvalues).

(d) The plant and noise model are

$$G(z) = \frac{z^{-1} + 1.5z^{-2}}{1 + 0.5z^{-1}} \quad H(z) = \frac{1 - 0.2z^{-1}}{1 + 0.5z^{-1}}, \tag{24}$$

and the one-step-ahead predictor can be found using $H(z)\hat{y}(t) = G(z)u(t) + (H(z) - 1)y(t)$.

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

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

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \dot{\mathbf{x}} = \mathbf{x} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u \tag{25}$$

1. Is (25) an implicit ODE or a DAE? Justify.
2. If it is an implicit ODE, what is its solution for $u = 0$? If it is a DAE, what is its differential index?

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

$$\dot{\mathbf{x}} = A\mathbf{x} + \mathbf{b}z \tag{26}$$

$$0 = \frac{1}{2}\left(\mathbf{x}^\top \mathbf{x} - L^2\right) \tag{27}$$

where L may be time-varying. What are the consistency conditions? What condition is needed on $\mathbf{b}$ and L for the DAE to be well-posed?

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

$$\dot{\mathbf{x}} = \mathbf{f}\left(\mathbf{x}, \mathbf{z}, \mathbf{u}\right) \tag{28a}$$

$$0 = \mathbf{g}\left(\mathbf{x}\right) \tag{28b}$$

Give a specific condition on $\mathbf{f}$ and $\mathbf{g}$ ensuring that (28) is of index 2. *Hint:* (28) *is of index 2 if a single time-differentiation of its algebraic constraint results in an index-1 DAE.*

Solution:

(a) 1. Since matrix E is rank deficient (one column of 0), (25) is a DAE

2. We observe that (25) reads as:

$$\dot{\mathbf{x}}_2 = \mathbf{x}_1 + u \tag{29a}$$

$$\dot{\mathbf{x}}_3 = \mathbf{x}_2 \tag{29b}$$

$$0 = \mathbf{x}_3 \tag{29c}$$

and is a semi-explicit DAE where $\mathbf{x}_1$ is the algebraic variable. It cannot be computed readily from (29) (equivalently the Jacobian of the algebraic constraint (29c) with respect to $\mathbf{x}_1$ is zero, i.e. rank deficient), hence we have a “high-index DAE” (index more than 1). In order to assess the index, we perform a $\frac{\mathrm{d}}{\mathrm{d}t}$ on the algebraic constraint (29c) (the other equations are explicit ODEs, we leave them alone). We obtain

$$\dot{\mathbf{x}}_3 = 0 \tag{30}$$

and use (29b) to get the new algebraic constraint $\mathbf{x}_2 = 0$. We perform a new $\frac{\mathrm{d}}{\mathrm{d}t}$, to obtain

$$\dot{\mathbf{x}}_2 = 0 \tag{31}$$

and use (29a) to obtain the new algebraic constraint:

$$\mathbf{x}_1 + u = 0 \tag{32}$$

We need yet another $\frac{\mathrm{d}}{\mathrm{d}t}$ to obtain:

$$\dot{\mathbf{x}}_1 + \dot{u} = 0 \tag{33}$$

The ODE corresponding to (25) is therefore:

$$\dot{\mathbf{x}}_2 = \mathbf{x}_1 + u \tag{34a}$$
$$\dot{\mathbf{x}}_3 = \mathbf{x}_2 \tag{34b}$$
$$\dot{\mathbf{x}}_1 = -\dot{u} \tag{34c}$$
$$\tag{34d}$$

Since we have performed 3 time differentiation to get there, our DAE is of index 3.

(b) We perform the index reduction by taking the time derivative of the algebraic constraint $g = \frac{1}{2}\left(\mathbf{x}^\top\mathbf{x} - L^2\right)$:

$$\dot{g} = \mathbf{x}^\top\dot{\mathbf{x}} = \mathbf{x}^\top A\mathbf{x} + \mathbf{x}^\top\mathbf{b}z = 0 \tag{35}$$

which is solvable for z as long as $\mathbf{x}^\top\mathbf{b} \neq 0$. We then have the index-1 DAE

$$\dot{\mathbf{x}} = A\mathbf{x} + \mathbf{b}z \tag{36a}$$
$$0 = \mathbf{x}^\top A\mathbf{x} + \mathbf{x}^\top\mathbf{b}z \tag{36b}$$

with the consistency condition $g(\mathbf{x}) = \frac{1}{2}\left(\mathbf{x}^\top\mathbf{x} - L^2\right) = 0$. For the DAE to be well-posed we need $\mathbf{x}^\top\mathbf{b} \neq 0$, i.e. the trajectory of $\mathbf{x}$ cannot end-up at a point where the vector $\mathbf{x}$ is orthogonal to the vector $\mathbf{b}$.

(c) DAE (28) is of index 2 if one step of index reduction yields an index-1 DAE. The first step of an index-reduction on (25) performs the operation:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{g}\left(\mathbf{x}\right) = \frac{\partial\mathbf{g}\left(\mathbf{x}\right)}{\partial\mathbf{x}}\dot{\mathbf{x}} = \frac{\partial\mathbf{g}\left(\mathbf{x}\right)}{\partial\mathbf{x}}\mathbf{f}\left(\mathbf{x},\mathbf{z},\mathbf{u}\right) \tag{37}$$

resulting in the DAE:

$$\dot{\mathbf{x}} = \mathbf{f}\left(\mathbf{x},\mathbf{z},\mathbf{u}\right) \tag{38a}$$
$$0 = \frac{\partial\mathbf{g}\left(\mathbf{x}\right)}{\partial\mathbf{x}}\mathbf{f}\left(\mathbf{x},\mathbf{z},\mathbf{u}\right) \tag{38b}$$

DAE (38) is of index 1 if the Jacobian of the new algebraic constraint $\frac{\partial\mathbf{g}(\mathbf{x})}{\partial\mathbf{x}}\mathbf{f}\left(\mathbf{x},\mathbf{z},\mathbf{u}\right)$ with respect to $\mathbf{z}$ is full rank, i.e. if the square matrix

$$\frac{\partial\mathbf{g}\left(\mathbf{x}\right)}{\partial\mathbf{x}}\frac{\partial\mathbf{f}\left(\mathbf{x},\mathbf{z},\mathbf{u}\right)}{\partial\mathbf{z}} \tag{39}$$

is full rank, or equivalently if:

$$\det\left(\frac{\partial\mathbf{g}\left(\mathbf{x}\right)}{\partial\mathbf{x}}\frac{\partial\mathbf{f}\left(\mathbf{x},\mathbf{z},\mathbf{u}\right)}{\partial\mathbf{z}}\right) \neq 0 \tag{40}$$

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

$$\frac{\partial \mathbf{r}\left(\mathbf{x}\right)}{\partial \mathbf{x}}\Delta\mathbf{x} + \mathbf{r}\left(\mathbf{x}\right) = 0 \tag{41a}$$

$$\mathbf{x} \leftarrow \mathbf{x} + \alpha\Delta\mathbf{x} \tag{41b}$$

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:

$$\left\|\mathbf{x}_+ - \mathbf{x}_\star\right\| \leq c\left\|\mathbf{x} - \mathbf{x}_\star\right\|^2 \tag{42}$$

where $\mathbf{x}_\star$ is a solution of $\mathbf{r}\left(\mathbf{x}_\star\right)$. 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 $\mathbf{r}\left(\mathbf{x}\right)$ 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 (41a) 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 ODE

$$\mathbf{F}\left(\dot{\mathbf{x}},\, \mathbf{x}, \mathbf{u}\right) = 0 \tag{43}$$

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) Specify what information the Butcher tableau readily provides on the resulting RK scheme, and what information is not obviously available (answers can be short but must be specific).

(d) (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.

(e) (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)

Solution:

(a) The pseudo-code will look like

```
Algorithm: Integration of implicit ODE
Input: $\mathbf{x}_0$, $\mathbf{u}(t_0), \ldots, \mathbf{u}(.)$, $\alpha$ and $\Delta t$
Set $K = 0$
for $k = 0 : N-1$ do
    while $\|\mathbf{r}(\mathbf{K}, \mathbf{x}_k, \mathbf{u}(.))\| > \text{tol}$ do
        Evaluate:
            $\mathbf{r}(\mathbf{K}, \mathbf{x}_k, \mathbf{u}(.)) = \begin{bmatrix} \mathbf{F}\left(\mathbf{K}_1, \mathbf{x}_k + \Delta t \sum_{i=1}^{s} a_{1i}\mathbf{K}_i,\, \mathbf{u}(t_k + c_1\Delta t)\right) \\ \vdots \\ \mathbf{F}\left(\mathbf{K}_s, \mathbf{x}_k + \Delta t \sum_{i=1}^{s} a_{si}\mathbf{K}_i,\, \mathbf{u}(t_k + c_s\Delta t)\right) \end{bmatrix} = 0$
        and
            $\frac{\partial \mathbf{r}(\mathbf{K}, \mathbf{x}_k, \mathbf{u}(.))}{\partial \mathbf{K}}$
        Take the Newton step
            $\mathbf{K} \leftarrow \mathbf{K} - \alpha \frac{\partial \mathbf{r}(\mathbf{K}, \mathbf{x}_k, \mathbf{u}(.))}{\partial \mathbf{K}}^{-1} \mathbf{r}$        (44)
    Take the integrator step:
        $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \sum_{i=1}^{s} b_i \mathbf{K}_i$        (45)
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) The Butcher tableau specifies: the number of stages of the RK method, whether the method is implicit or explicit and enough information to code the RK scheme. It does not (readily) provide the order of the integration method, as the order depends on the specific entries used in the tableau. To assess the order from the tableau, one needs to perform (possibly) involved computations.

(d) 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 (45). 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. $\mathcal{O}\left(n^3s^3\right)$ (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. $N = \frac{t_\mathrm{f}}{\Delta t}$, i.e. the overall complexity of the integration scheme is dominated by $\mathcal{O}\left(\Delta t^{-1}n^3s^3\right)$. The ratio complexity-order is then $\mathcal{O}\left(\Delta t^{-1}n^3s^3\right)$ 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.

(e) The answer lies in Table 1. Up to order $o = 4$, ERK methods require $s = o$ stages, hence $s = o$ evaluations of the model equations. Because the global error follows $\|\mathbf{x}_N - \mathbf{x}\left(t_\mathrm{f}\right)\| = \mathcal{O}\left(\Delta t^o\right)$, each extra function evaluation readily delivers an extra order of accuracy, and allows for reducing the total number of function evaluation required. This trend is broken for $o > 4$. Indeed, at higher orders, the required number of stages (and hence the number of function evaluations) progresses faster than o. Then the overall computational cost of obtaining a given accuracy tends to not improve (or even increase) for orders higher orders. In practice, it is often observed that the minimal computational complexity is achieved at orders of 4 to 5 (hence e.g. the ode45 method in Matlab).

Appendix: some possibly useful formula

- Lagrange mechanics is built on the equations:

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}} - \frac{\partial \mathcal{L}}{\partial \mathbf{q}} = \mathbf{Q}, \qquad \mathcal{L}\left(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{z}\right) = \mathcal{T} - \mathcal{V} - \mathbf{z}^\top \mathbf{C}, \qquad \mathbf{C} = 0, \qquad \langle \delta\mathbf{q}, \mathbf{Q}\rangle = \delta W, \,\forall\, \delta\mathbf{q} \tag{46}$$

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

$$\mathcal{T} = \frac{1}{2} m \dot{\mathbf{p}}^\top \dot{\mathbf{p}}, \qquad \mathcal{V} = m g \mathbf{p}_3 \tag{47}$$

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:

$$\dot{\mathbf{p}} = \mathbf{v} \tag{48a}$$

$$W\dot{\mathbf{v}} + \frac{\partial \mathbf{C}}{\partial \mathbf{q}}^\top \mathbf{z} = \mathbf{Q} - \frac{\partial \mathcal{V}}{\partial \mathbf{q}}^\top \tag{48b}$$

$$0 = \mathbf{C}\left(\mathbf{q}\right) \tag{48c}$$

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

$$\mathbf{r}\left(\mathbf{y}, \mathbf{z}\right) = 0 \tag{49}$$

"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

$$\mathbf{r}\left(\mathbf{y}, \phi\left(\mathbf{y}\right)\right) = 0 \tag{50}$$

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

$$\frac{\partial \mathbf{z}}{\partial \mathbf{y}} = -\frac{\partial \mathbf{r}}{\partial \mathbf{z}}^{-1} \frac{\partial \mathbf{r}}{\partial \mathbf{y}} \tag{51}$$

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

$$\mathbf{x} \leftarrow \mathbf{x} - \alpha \frac{\partial \mathbf{r}}{\partial \mathbf{x}}^{-1} \mathbf{r} \tag{52}$$

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

- Runge-Kutta methods are described by:

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

$$\mathbf{K}_j = \mathbf{f}\left(\mathbf{x}_k + \Delta t \sum_{i=1}^{s} a_{ji}\mathbf{K}_i,\, \mathbf{u}\left(t_k + c_j \Delta t\right)\right), \quad j = 1, \dots, s \tag{53a}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \sum_{i=1}^{s} b_i \mathbf{K}_i \tag{53b}$$

- 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:

$$\dot{\mathbf{x}}(t_k + \Delta t \cdot \tau) \approx \dot{\hat{\mathbf{x}}}(t_k + \Delta t \cdot \tau) = \sum_{i=1}^{s} \mathbf{K}_i \ell_i(\tau), \quad \tau \in [0,\, 1] \tag{54}$$

$$\mathbf{x}(t_k + \Delta t \cdot \tau) \approx \hat{\mathbf{x}}(t_k + \Delta t \cdot \tau) = \mathbf{x}_k + \Delta t \sum_{i=1}^{s} \mathbf{K}_i L_i(\tau) \tag{55}$$

where the Lagrange polynomials are given by:

$$\ell_i(\tau) = \prod_{j=1, j\neq i}^{s} \frac{\tau - \tau_j}{\tau_i - \tau_j}, \quad \text{and} \quad L_i(\tau) = \int_0^\tau \ell_i(\xi)\mathrm{d}\xi \tag{56}$$

The Lagrange polynomials satisfy the conditions of

$$\text{Orthogonality:} \quad \int_0^1 \ell_i(\tau)\ell_j(\tau)\,\mathrm{d}\tau = 0 \quad \text{for} \quad i \neq j \tag{57a}$$

$$\text{Punctuality:} \quad \ell_i(\tau_j) = \begin{cases} 1 & \text{if} \quad j = i \\ 0 & \text{if} \quad j \neq i \end{cases} \tag{57b}$$

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

$$\dot{\hat{\mathbf{x}}}(t_k + \Delta t \cdot \tau_j) = \mathbf{f}\left(\hat{\mathbf{x}}(t_k + \Delta t \cdot \tau_j),\, \mathbf{u}\left(t_k + \Delta t \cdot \tau_j\right)\right), \qquad \text{in the explicit ODE case} \tag{58a}$$

$$\mathbf{F}\left(\dot{\hat{\mathbf{x}}}(t_k + \Delta t \cdot \tau_j), \hat{\mathbf{x}}(t_k + \Delta t \cdot \tau_j),\, \mathbf{u}\left(t_k + \Delta t \cdot \tau_j\right)\right) = 0, \qquad \text{in the implicit ODE case} \tag{58b}$$

$$\mathbf{F}\left(\dot{\hat{\mathbf{x}}}(t_k + \Delta t \cdot \tau_j), \hat{\mathbf{z}}_j, \hat{\mathbf{x}}(t_k + \Delta t \cdot \tau_j),\, \mathbf{u}\left(t_k + \Delta t \cdot \tau_j\right)\right) = 0, \qquad \text{in the fully-implicit DAE case} \tag{58c}$$

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

$$P_s\left(\tau\right) = \frac{1}{s!}\frac{\mathrm{d}^s}{\mathrm{d}\tau^s}\left[\left(\tau^2 - \tau\right)^s\right] \tag{59}$$

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

- Maximum-likelihood estimation is based on

$$\max_{\boldsymbol{\theta}} \quad \mathbb{P}\left[e_k = y_k - \hat{y}_k \quad \text{for} \quad k = 1, \ldots, N \mid \boldsymbol{\theta}\right] \tag{60}$$

If the noise sequence is uncorrelated, then

$$\mathbb{P}\left[e_k = y_k - \hat{y}_k \quad \text{for} \quad k = 0, \ldots, N \mid \boldsymbol{\theta}\right] = \prod_{k=1}^{N} \mathbb{P}\left[e_k = y_k - \hat{y}_k \mid \boldsymbol{\theta}\,\right] \tag{61}$$

- The solution of a linear least-squares problem

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{2}\left\|A\boldsymbol{\theta} - \mathbf{y}\right\|^2_{\Sigma_e^{-1}} \tag{62}$$

reads as:

$$\hat{\boldsymbol{\theta}} = \left(A^\top \Sigma_e^{-1} A\right)^{-1} A^\top \Sigma_e^{-1} \mathbf{y} \tag{63}$$

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

$$\Sigma_{\hat{\boldsymbol{\theta}}} = \left(A^\top \Sigma_e^{-1} A\right)^{-1} \tag{64}$$

- 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

$$H(z)\hat{y}(z) = G(z)u(z) + (H(z)-1)y(z) \tag{65}$$

- The Gauss-Newton approximation in an optimization problem

$$\min_{\mathbf{x}} \quad J\left(\mathbf{x}\right) = \frac{1}{2}\left\|\mathbf{R}\left(\mathbf{x}\right)\right\|^2 \tag{66}$$

uses the approximation:

$$\frac{\partial^2 J}{\partial \mathbf{x}^2} \approx \frac{\partial R}{\partial \mathbf{x}}^\top \frac{\partial R}{\partial \mathbf{x}} \tag{67}$$

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

$$\mathbf{x}(t) = e^{At}\mathbf{x}(0) + \int_0^t e^{A(t-\tau)} B\mathbf{u}(\tau)\mathrm{d}\tau \tag{68}$$

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

$$G(s) = C\left(sI - A\right)^{-1} B + D \tag{69}$$