MOS ESS101
January 2019

This exam contains 17 pages (including this cover page) and 5 problems.

You are allowed to use the following books:

- **β-handbook**
- **"Physics handbook for science and engineering"**
- **"Lecture Notes for Modeling and Simulation"**
- **One A4 page of notes (cannot be a copy of old exams)**

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, unclear or unsupported answers will not receive credit. It is up to you to show that you understand your answer.
- The passing grade will a priori be given at 30 points, and the top grade at 45 points. These limits may be lowered depending on the outcome of the exam.

Problem	Points	Score
1	11	
2	7	
3	14	
4	6	
5	12	
Total:	50	

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) What is the force in the link between the masses? What is the condition for mass 1 to not detach from the surface?

   (d) (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.

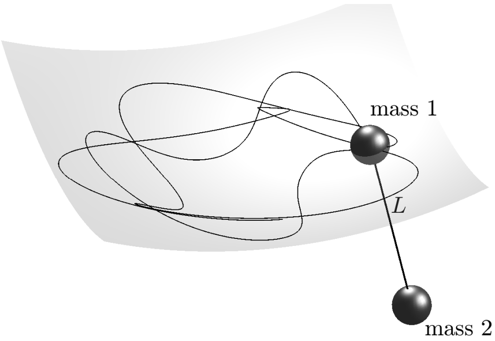

Figure 1: Illustration of the system. The surface φ is depicted as the see-through grey surface. The trajectory of the mass m_1 is depicted as a black trace on the surface φ.

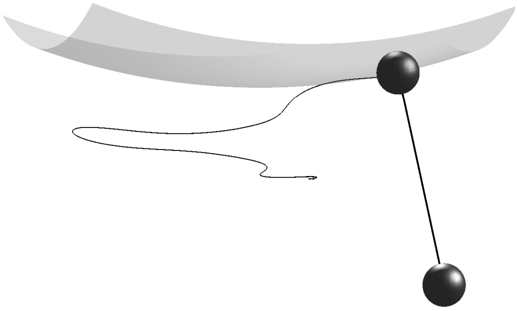

Figure 2: Simulation of the system over 10 s. The trajectory of the mass m_1 depicted as a black trace leaves the surface φ. We want to understand what can cause this.

Solution:

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

$$\mathbf{q} = \left[\begin{array}{c} \mathbf{p}_1 \\ \mathbf{p}_2 \end{array}\right] \in \mathbb{R}^6 \tag{1}$$

specifying the position of each ball. The system 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 $\mathbf{a}^\top = \left[\begin{array}{ccccccc} 0 & 0 & m & 0 & 0 & m & \dots \end{array}\right]$. We have the constraint functions

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

The Lagrange function then reads as:

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

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:

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

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

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

This result could also be obtained directly from (61) by observing that $W\left(\mathbf{q}\right) = I$. We can additionally further explicitly provide (though this is not required from the question):

$$\frac{\partial\mathbf{C}}{\partial\mathbf{q}}^\top = \left[\begin{array}{cc} \mathbf{p}_1 & \mathbf{p}_1 - \mathbf{p}_2 \\ 0 & \mathbf{p}_2 - \mathbf{p}_1 \end{array}\right] \tag{7}$$

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

$$\dot{\mathbf{C}} = \left[\begin{array}{c} \mathbf{p}_1^\top\dot{\mathbf{p}}_1 \\ \left(\mathbf{p}_2 - \mathbf{p}_1\right)^\top\left(\dot{\mathbf{p}}_2 - \dot{\mathbf{p}}_1\right) \end{array}\right] = 0 \tag{8a}$$

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

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:

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

(c) The forces delivered by the constraints in the system (surface and link) are given by:

$$\mathbf{F} = -\frac{1}{m}\frac{\partial \mathbf{C}}{\partial \mathbf{q}}^\top \mathbf{z} \tag{12}$$

and drive the accelerations in (9a). The force acting on the second mass is the second part of that vector and reads as:

$$\|\mathbf{F}_{\text{link}}\| = \frac{1}{m}\|\mathbf{p}_1 - \mathbf{p}_2\| \mathbf{z}_2 = \frac{L}{m}\mathbf{z}_2 \tag{13}$$

Similarly, the force delivered by the surface on mass 1 is given by:

$$\mathbf{F}_{\text{surf}} = -\frac{1}{m}\mathbf{p}_1 \mathbf{z}_1 \tag{14}$$

We will consider that mass 1 is "resting" on the surface, i.e. it would detach if the surface has to "pull down" the mass. The normal to the surface (pointing "up") is given by:

$$-\nabla\varphi(\mathbf{p}_1) = -\mathbf{p}_1 \tag{15}$$

and the condition for mass 1 to not detach from the surface is given by:

$$-\nabla\varphi(\mathbf{p}_1)^\top \mathbf{F}_{\text{surf}} > 0 \tag{16}$$

i.e.

$$\frac{1}{m}\mathbf{p}_1^\top \mathbf{p}_1 \mathbf{z}_1 = \frac{R^2}{m}\mathbf{z}_1 > 0 \tag{17}$$

or more simply $\mathbf{z}_1 > 0$.

(d) 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 (11). As a result, the constraints $\mathbf{C}$ follow the first-order dynamics:

$$\mathbf{C}\left(\mathbf{q}(t)\right) = \mathbf{C}\left(\mathbf{q}(0)\right) + t\dot{\mathbf{C}}\left(\mathbf{q}(0), \dot{\mathbf{q}}(0)\right) \tag{18}$$

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

2. **System Identification**

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

$$y_k + 0.5y_{k-1} = u_{k-1} + 1.5u_{k-2} + e_k - 0.2e_{k-1} \tag{19}$$

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

(b) (2 points) Consider the following data samples

$$x_k = A + e_k, \quad k = 0, \cdots, N-1 \tag{20}$$

with e_k being normal centered and uncorrelated (white noise) with variance σ^2. Consider the following estimator for the parameter A,

$$\hat{A} = a\frac{1}{N}\sum_{k=0}^{N-1} x[k], \tag{21}$$

for some constant a. The *mean square error* (MSE) of the parameter estimate is defined as

$$\mathrm{mse}(\hat{A}) = \mathrm{var}(\hat{A}) + b^2(\hat{A}), \tag{22}$$

where var is the variance and b is the bias of the estimate. What is the value of a that minimizes the MSE?

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

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \quad \frac{1}{2}\left\|A\boldsymbol{\theta} - \mathbf{y}\right\|^2_{\Sigma^{-1}} \tag{23}$$

1. What are we assuming about the data when using a least-squares fitting problem like (32).
2. Explain in detail the meaning of formula (77) 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?

Solution:

(a) 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 $H(z)\hat{y}(t) = G(z)u(t) + (H(z) - 1)y(t)$.

$$\frac{1 - 0.2z^{-1}}{1 + 0.5z^{-1}}\hat{y} = \frac{z^{-1} + 1.5z^{-2}}{1 + 0.5z^{-1}}u + \Big(\frac{1 - 0.2z^{-1}}{1 + 0.5z^{-1}} - 1\Big)y$$

$$(1 - 0.2z^{-1})\hat{y} = (z^{-1} + 1.5z^{-2})u + (-0.7^{-1})y$$

$$\rightarrow \hat{y}_k = 0.2\hat{y}_{k-1} + u_{k-1} + 1.5u_{k-2} - 0.7y_{k-1}$$

(b) The variance of $\hat{A}$ is $a^2\sigma^2/N$, while the bias is $\mathbb{E}(\hat{A}) - A = aA - A = (a-1)A$. Hence we have that

$$\mathrm{mse}(\hat{A}) = \frac{a^2\sigma^2}{N} + (a-1)^2A^2. \tag{25}$$

Differentiating the MSE with respect to a yields

$$\frac{\partial \mathrm{mse}(\hat{A})}{\partial a} = \frac{2a\sigma^2}{N} + 2(a-1)A^2, \tag{26}$$

which setting to zero and solving for a yields the value

$$a_{opt} = \frac{A^2}{A^2 + \sigma^2/N}. \tag{27}$$

(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 $\mathbf{y}_{0,\ldots,N}$ reads as:

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

where $\boldsymbol{\theta}_{\mathrm{true}}$ 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 $\mathbf{y} = A\boldsymbol{\theta}_{\mathrm{true}} + \mathbf{e}$ is also a vector or random variables. Hence the outcome $\hat{\boldsymbol{\theta}}$ of the least-squares problem (32) (see formula (76))

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

is itself a random variable (it is in fact a linear function of the random vector $\mathbf{y}$). Equation (29) 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 (32) 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.

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

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

$$\mathbf{F}\left(\dot{\mathbf{x}},\,\mathbf{x},\mathbf{z},\mathbf{u}\right)=0 \tag{33}$$

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:

$$\mathbf{F}\,:\,\underbrace{\mathbb{R}^{n}\times\mathbb{R}^{n_\mathbf{x}}\times\mathbb{R}^{n_\mathbf{z}}\times\mathbb{R}^{n_\mathbf{u}}}_{\text{size of the arguments}}\qquad\mapsto\qquad\underbrace{\mathbb{R}^{m}}_{\text{“size of the function output”}} \tag{34}$$

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:

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

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

Rewrite it in the fully-implicit form (33). 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 (33), rewrite it in a semi-explicit form (35). *Hints: don't look for something complicated. You will need to introduce new algebraic variables.*

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

$$\begin{bmatrix}1&1&0\\0&0&1\\1&1&1\end{bmatrix}\dot{\mathbf{x}}=\mathbf{x} \tag{36}$$

1. Is (36) 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:

$$\dot{\mathbf{x}}=A\mathbf{x}+\mathbf{b}z \tag{37a}$$

$$0=\mathbf{g}\left(\mathbf{x}\right) \tag{37b}$$

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:

$$\dot{\mathbf{x}}=\mathbf{f}\left(\mathbf{x},\mathbf{z},\mathbf{u}\right) \tag{38a}$$

$$0=\mathbf{g}\left(\mathbf{x}\right) \tag{38b}$$

Prove that (38) is at least of index 2.

Solution:

(a)
- Clearly, the size of $\dot{\mathbf{x}}$ is the same as the one of $\mathbf{x}$, i.e. $n=n_\mathbf{x}$.
- Equation (33) ought to deliver $\dot{\mathbf{x}}$ and $\mathbf{z}$ for given $\mathbf{x}$ and $\mathbf{u}$. That is, equation (33) 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. $m=n_\mathbf{x}+n_\mathbf{z}$. An alternative and valid answer is to justify this statement in the light of the Implicit

Function Theorem, which requires that $\frac{\partial \mathbf{F}}{\partial \mathbf{w}}$ is invertible, where $\mathbf{w} = \left[\begin{array}{c} \dot{\mathbf{x}} \\ \mathbf{z} \end{array}\right]$. In order to be invertible, this matrix ought to be square.

(b) We can simply write:

$$\mathbf{F}\left(\dot{\mathbf{x}},\, \mathbf{x}, \mathbf{z}, \mathbf{u}\right) = \left[\begin{array}{c} \dot{\mathbf{x}} - \mathbf{f}\left(\mathbf{x}, \mathbf{z}, \mathbf{u}\right) \\ \mathbf{g}\left(\mathbf{x}, \mathbf{z}, \mathbf{u}\right) \end{array}\right] = 0 \tag{39}$$

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

$$\dot{\mathbf{x}} = \mathbf{v} \tag{40a}$$

$$0 = \mathbf{F}\left(\mathbf{v},\, \mathbf{x}, \mathbf{z}, \mathbf{u}\right) \tag{40b}$$

This equation is a semi-explicit DAE.

(d)

1. Since matrix E is rank deficient (3rd line is the sum of the 1st and 2nd lines), (36) is a DAE

2. We observe that DAE (36) reads as:

$$\dot{\mathbf{x}}_1 + \dot{\mathbf{x}}_2 = \mathbf{x}_1 \tag{41a}$$

$$\dot{\mathbf{x}}_3 = \mathbf{x}_2 \tag{41b}$$

$$\dot{\mathbf{x}}_1 + \dot{\mathbf{x}}_2 + \dot{\mathbf{x}}_3 = \mathbf{x}_3 \tag{41c}$$

Subtracting (41a) and (41b) to (41c), we get:

$$\dot{\mathbf{x}}_1 + \dot{\mathbf{x}}_2 = \mathbf{x}_1 \tag{42a}$$

$$\dot{\mathbf{x}}_3 = \mathbf{x}_2 \tag{42b}$$

$$0 = \mathbf{x}_1 + \mathbf{x}_2 - \mathbf{x}_3 \tag{42c}$$

A time-differentiation of (42c) yields:

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

$$\dot{\mathbf{x}}_3 = \mathbf{x}_2 \tag{43b}$$

$$0 = \mathbf{x}_1 + \mathbf{x}_2 - \mathbf{x}_3 \tag{43c}$$

$$\dot{\mathbf{x}}_1 + \dot{\mathbf{x}}_2 - \dot{\mathbf{x}}_3 = 0 \tag{43d}$$

We then do (43d) - (43a) + (43b) to get:

$$\dot{\mathbf{x}}_1 + \dot{\mathbf{x}}_2 = \mathbf{x}_1 \tag{44a}$$

$$\dot{\mathbf{x}}_3 = \mathbf{x}_2 \tag{44b}$$

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

$$0 = \mathbf{x}_1 - \mathbf{x}_2 \tag{44d}$$

A time-differentiation of (44d) yields:

$$\begin{aligned}\dot{\mathbf{x}}_1 + \dot{\mathbf{x}}_2 &= \mathbf{x}_1 && (45a)\\ \dot{\mathbf{x}}_3 &= \mathbf{x}_2 && (45b)\\ 0 &= \mathbf{x}_1 + \mathbf{x}_2 - \mathbf{x}_3 && (45c)\\ 0 &= \mathbf{x}_1 - \mathbf{x}_2 && (45d)\\ \dot{\mathbf{x}}_1 - \dot{\mathbf{x}}_2 &= 0 && (45e)\end{aligned}$$

We then do (45e)+(45a) to get the semi-explicit DAE:

$$\begin{aligned}\dot{\mathbf{x}}_1 &= \frac{1}{2}\mathbf{x}_1 && (46a)\\ 0 &= \mathbf{x}_1 + \mathbf{x}_2 - \mathbf{x}_3 && (46b)\\ 0 &= \mathbf{x}_1 - \mathbf{x}_2 && (46c)\end{aligned}$$

(e) In this case m = 1. We are dealing with a semi-explicit DAE, hence the index reduction requires time-differentiations of the algebraic equation (37b). The first step of the index reduction reads as:

$$\begin{aligned}\dot{\mathbf{x}} &= A\mathbf{x} + \mathbf{b}z && (47a)\\ 0 &= \frac{\partial \mathbf{g}\left(\mathbf{x}\right)}{\partial \mathbf{x}}\dot{\mathbf{x}} = \frac{\partial \mathbf{g}\left(\mathbf{x}\right)}{\partial \mathbf{x}}\left(A\mathbf{x} + \mathbf{b}z\right) && (47b)\end{aligned}$$

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 (47) is of index 1. The consistency condition is simply:

$$\mathbf{g}\left(\mathbf{x}(0)\right) = 0 \qquad (48)$$

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 (38) 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 $\mathbf{r}\left(\mathbf{x}\right) = 0$ numerically. To that end, iterates the recursion:

$$\frac{\partial \mathbf{r}\left(\mathbf{x}\right)}{\partial \mathbf{x}}\Delta\mathbf{x} + \mathbf{r}\left(\mathbf{x}\right) = 0 \tag{49a}$$

$$\mathbf{x} \leftarrow \mathbf{x} + \alpha\Delta\mathbf{x} \tag{49b}$$

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:

$$\left\|\mathbf{x}_+ - \mathbf{x}_\star\right\| \leq c\left\|\mathbf{x} - \mathbf{x}_\star\right\|^2 \tag{50}$$

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?

(c) (2 points) Consider the optimization problem:

$$\mathbf{x}(\mathbf{p}) = \min_{\mathbf{v}}\ \Phi(\mathbf{v}, \mathbf{p}) \tag{51}$$

for some parameters $\mathbf{p}$. Provide a sufficient condition for (51) to have a unique solution for a given $\mathbf{p}$. Give an expression for the Jacobian:

$$\frac{\partial \mathbf{x}(\mathbf{p})}{\partial \mathbf{p}} \tag{52}$$

What is required for (52) to exist?

Solution:

(a) Full Newton steps are guaranteed to converge in a neighborhood of a solution only. The convergence in a neighborhood of the solution requires that $\frac{\partial \mathbf{r}(\mathbf{x})}{\partial \mathbf{x}}$ is full rank at the solution, and Lipschitz continuous in the neighborhood of the solution.

(b) This formula states that the exact, full-step Newton iteration converges quadratically to a solution. 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 (49a) 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 only in a neighborhood of the solution $\mathbf{x}_\star$, and requires the conditions under a) to hold.

(c) Problem (51) is guaranteed to have a unique solution if it is convex, i.e. if $\nabla_{\mathbf{xx}}\Phi(\mathbf{x}, \mathbf{p})$ is positive definite. We observe that the solution of (51) satisfies the implicit function:

$$\nabla_{\mathbf{x}}\Phi(\mathbf{x}, \mathbf{p}) \tag{53}$$

We can then use the Implicit Function Theorem and observe that:

$$\left(\frac{\partial}{\partial \mathbf{x}}\nabla_{\mathbf{x}}\Phi(\mathbf{x}, \mathbf{p})\right)\frac{\partial \mathbf{x}(\mathbf{p})}{\partial \mathbf{p}} + \frac{\partial}{\partial \mathbf{p}}\nabla_{\mathbf{x}}\Phi(\mathbf{x}, \mathbf{p}) = 0 \tag{54}$$

or more simply:

$$\frac{\partial \mathbf{x}(\mathbf{p})}{\partial \mathbf{p}} = -\nabla_{\mathbf{xx}}\Phi(\mathbf{x},\mathbf{p})^{-1}\nabla_{\mathbf{xp}}\Phi(\mathbf{x},\mathbf{p}) \tag{55}$$

This calculation is only possible if the square matrix $\nabla_{\mathbf{xx}}\Phi(\mathbf{x},\mathbf{p})$ is full rank.

5. **Simulation**

   (a) (4 points) Write a pseudo-code (algorithm) that would deploy an IRK scheme for an implicit DAE

$$\mathbf{F}\left(\dot{\mathbf{x}},\,\mathbf{x},\,\mathbf{z},\mathbf{u}\right)=0 \tag{56}$$

   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

   (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?

Solution:

(a) Modify The pseudo-code will look like

```
Algorithm: Integration of implicit ODE
Input: $\mathbf{x}_0$, $\mathbf{u}(t_0),\dots,\mathbf{u}(.)$, $\alpha$ and $\Delta t$
Set $\mathbf{K},\mathbf{z}=0$ (or any better initial guess)
for $k=0:N-1$ do
    while $\|\mathbf{r}\left(\mathbf{K},\mathbf{z},\mathbf{x}_k,\mathbf{u}(.)\right)\|>$ tol do
        Evaluate:
            $\mathbf{r}\left(\mathbf{K},\mathbf{x}_k,\mathbf{u}(.)\right)=\begin{bmatrix}\mathbf{F}\left(\mathbf{K}_1,\mathbf{z}_1,\mathbf{x}_k+\Delta t\sum_{i=1}^{s}a_{1i}\mathbf{K}_i,\,\mathbf{u}(t_k+c_1\Delta t)\right)\\ \vdots \\ \mathbf{F}\left(\mathbf{K}_s,\mathbf{z}_s,\mathbf{x}_k+\Delta t\sum_{i=1}^{s}a_{si}\mathbf{K}_i,\,\mathbf{u}(t_k+c_s\Delta t)\right)\end{bmatrix}=0$
        and
            $\frac{\partial\mathbf{r}\left(\mathbf{K},\mathbf{z},\mathbf{x}_k,\mathbf{u}(.)\right)}{\partial\mathbf{w}}$
        where $\mathbf{w}$ gathers $\mathbf{K}_{1,\dots,s}$ and $\mathbf{z}_{1,\dots,s}$.
        Take the Newton step
            $\mathbf{w}\leftarrow\mathbf{w}-\alpha\frac{\partial\mathbf{r}\left(\mathbf{K},\mathbf{z},\mathbf{x}_k,\mathbf{u}(.)\right)}{\partial\mathbf{w}}^{-1}\mathbf{r}$    (57)
    Take the integrator step:
        $\mathbf{x}_{k+1}=\mathbf{x}_k+\Delta t\sum_{i=1}^{s}b_i\mathbf{K}_i$    (58)
return $\mathbf{x}_{0,\dots N}$
```

(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.

(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. The trade off between having a high order and taking fewer steps, or having a lower order but taking more steps is not straightforward, but it tends to favor fairly low order methods.

(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. 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 $o > 4$. 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 higher orders.

Appendix: some possibly useful formula

- Lagrange mechanics is built on the equations:

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}} - \frac{\partial \mathcal{L}}{\partial \mathbf{q}} = \mathbf{Q}, \qquad \mathcal{L}\left(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{z}\right) = \mathcal{T} - \mathcal{V} - \mathbf{z}^\top \mathbf{C}, \qquad \mathbf{C} = 0, \qquad \langle \delta\mathbf{q}, \mathbf{Q}\rangle = \delta W, \forall\, \delta\mathbf{q} \tag{59}$$

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

$$\mathcal{T} = \frac{1}{2} m \dot{\mathbf{p}}^\top \dot{\mathbf{p}}, \qquad \mathcal{V} = mg\mathbf{p}_3 \tag{60}$$

  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:

$$\dot{\mathbf{p}} = \mathbf{v} \tag{61a}$$

$$W\dot{\mathbf{v}} + \frac{\partial \mathbf{C}}{\partial \mathbf{q}}^\top \mathbf{z} = \mathbf{Q} - \frac{\partial \mathcal{V}}{\partial \mathbf{q}}^\top \tag{61b}$$

$$0 = \mathbf{C}\left(\mathbf{q}\right) \tag{61c}$$

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

$$\mathbf{r}\left(\mathbf{y}, \mathbf{z}\right) = 0 \tag{62}$$

  "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

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

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

$$\frac{\partial \mathbf{z}}{\partial \mathbf{y}} = -\frac{\partial \mathbf{r}}{\partial \mathbf{z}}^{-1} \frac{\partial \mathbf{r}}{\partial \mathbf{y}} \tag{64}$$

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

$$\mathbf{x} \leftarrow \mathbf{x} - \alpha \frac{\partial \mathbf{r}}{\partial \mathbf{x}}^{-1} \mathbf{r} \tag{65}$$

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

- Runge-Kutta methods are described by:

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

$$\mathbf{K}_j = \mathbf{f}\left(\mathbf{x}_k + \Delta t \sum_{i=1}^{s} a_{ji}\mathbf{K}_i,\, \mathbf{u}\left(t_k + c_j\Delta t\right)\right), \quad j = 1, \dots, s \tag{66a}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \sum_{i=1}^{s} b_i \mathbf{K}_i \tag{66b}$$

- 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:

$$\dot{\mathbf{x}}(t_k + \Delta t \cdot \tau) \approx \dot{\hat{\mathbf{x}}}(t_k + \Delta t \cdot \tau) = \sum_{i=1}^{s} \mathbf{K}_i \ell_i(\tau), \quad \tau \in [0,\, 1] \tag{67}$$

$$\mathbf{x}(t_k + \Delta t \cdot \tau) \approx \hat{\mathbf{x}}(t_k + \Delta t \cdot \tau) = \mathbf{x}_k + \Delta t \sum_{i=1}^{s} \mathbf{K}_i L_i(\tau) \tag{68}$$

where the Lagrange polynomials are given by:

$$\ell_i(\tau) = \prod_{j=1, j\neq i}^{s} \frac{\tau - \tau_j}{\tau_i - \tau_j}, \quad \text{and} \quad L_i(\tau) = \int_0^{\tau} \ell_i(\xi) \mathrm{d}\xi \tag{69}$$

The Lagrange polynomials satisfy the conditions of

$$\text{Orthogonality:} \quad \int_0^1 \ell_i(\tau)\ell_j(\tau)\,\mathrm{d}\tau = 0 \quad \text{for} \quad i \neq j \tag{70a}$$

$$\text{Punctuality:} \quad \ell_i(\tau_j) = \begin{cases} 1 & \text{if} \quad j = i \\ 0 & \text{if} \quad j \neq i \end{cases} \tag{70b}$$

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

$$\dot{\hat{\mathbf{x}}}(t_k + \Delta t \cdot \tau_j) = \mathbf{f}\left(\hat{\mathbf{x}}(t_k + \Delta t \cdot \tau_j),\, \mathbf{u}\left(t_k + \Delta t \cdot \tau_j\right)\right), \qquad \text{in the explicit ODE case} \tag{71a}$$

$$\mathbf{F}\left(\dot{\hat{\mathbf{x}}}(t_k + \Delta t \cdot \tau_j), \hat{\mathbf{x}}(t_k + \Delta t \cdot \tau_j),\, \mathbf{u}\left(t_k + \Delta t \cdot \tau_j\right)\right) = 0, \qquad \text{in the implicit ODE case} \tag{71b}$$

$$\mathbf{F}\left(\dot{\hat{\mathbf{x}}}(t_k + \Delta t \cdot \tau_j), \hat{\mathbf{z}}_j, \hat{\mathbf{x}}(t_k + \Delta t \cdot \tau_j),\, \mathbf{u}\left(t_k + \Delta t \cdot \tau_j\right)\right) = 0, \qquad \text{in the fully-implicit DAE case} \tag{71c}$$

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

$$P_s\left(\tau\right) = \frac{1}{s!} \frac{\mathrm{d}^s}{\mathrm{d}\tau^s} \left[\left(\tau^2 - \tau\right)^s\right] \tag{72}$$

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

$$\max_{\boldsymbol{\theta}} \quad \mathbb{P}\left[e_k = y_k - \hat{y}_k \quad \text{for} \quad k = 1, \ldots, N \mid \boldsymbol{\theta}\right] \tag{73}$$

If the noise sequence is uncorrelated, then

$$\mathbb{P}\left[e_k = y_k - \hat{y}_k \quad \text{for} \quad k = 0, \ldots, N \mid \boldsymbol{\theta}\right] = \prod_{k=1}^{N} \mathbb{P}\left[e_k = y_k - \hat{y}_k \mid \boldsymbol{\theta}\,\right] \tag{74}$$

- The solution of a linear least-squares problem

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{2} \left\|A\boldsymbol{\theta} - \mathbf{y}\right\|^2_{\Sigma_e^{-1}} \tag{75}$$

reads as:

$$\hat{\boldsymbol{\theta}} = \left(A^\top \Sigma_e^{-1} A\right)^{-1} A^\top \Sigma_e^{-1} \mathbf{y} \tag{76}$$

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

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

- 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

$$H(z)\hat{y}(z) = G(z)u(z) + (H(z) - 1)y(z) \tag{78}$$

- The Gauss-Newton approximation in an optimization problem

$$\min_{\mathbf{x}} \quad J\left(\mathbf{x}\right) = \frac{1}{2}\left\|\mathbf{R}\left(\mathbf{x}\right)\right\|^2 \tag{79}$$

uses the approximation:

$$\frac{\partial^2 J}{\partial \mathbf{x}^2} \approx \frac{\partial R}{\partial \mathbf{x}}^\top \frac{\partial R}{\partial \mathbf{x}} \tag{80}$$

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

$$\mathbf{x}(t) = e^{At}\mathbf{x}(0) + \int_0^t e^{A(t-\tau)} B\mathbf{u}(\tau)\mathrm{d}\tau \tag{81}$$

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

$$G(s) = C\left(sI - A\right)^{-1} B + D \tag{82}$$