MOS ESS101
August 2018 (re-exam)

This exam contains 13 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	10	
2	7	
3	9	
4	4	
5	13	
Total:	43	

Best of luck to all !!

1. **Lagrange modelling** Consider the Δ-robot, common in ultra-fast packaging applications, illustrated in Fig. 1. The three yellow arms of length l are actuated and can pivot in their vertical planes. They drive the three double thin rods of length L (typically made of ultra-light carbon fibre), connected to the nacelle (triangular shape at the bottom). The geometry imposes the nacelle to always remain horizontal. The pivots on the nacelle are forced to remain at a distance L from the pivot at the extremities of the yellow arms. The goal here will be to write the Lagrange function of the Δ-robot (at a high level). To help you, we specify next the *position* of the extremities of the yellow arms in a cartesian frame (x axis aligned with the arm labelled "1", z axis up).

$$\mathbf{p}_k = R_k \begin{bmatrix} d + l\cos\alpha_k \\ 0 \\ -l\sin\alpha_k \end{bmatrix}, \qquad R_k = \begin{bmatrix} \cos\gamma_k & -\sin\gamma_k & 0 \\ \sin\gamma_k & \cos\gamma_k & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

where $\gamma_{1,2,3} = \left\{0, \frac{2\pi}{3}, \frac{4\pi}{3}\right\}$, d is the (constant) distance from the center of the upper platform to the axis of the motors, and α_k are the angles of the yellow arms with respect to the horizontal plane. The yellow arms together with the motors have an inertia J, i.e. their kinetic energy is $T_k = \frac{1}{2}J\dot{\alpha}_k^2$. The nacelle has a mass m. For simplicity, we will consider that the nacelle is a point where (long) arms are all connected.

(a) (2 points) What are the generalized coordinates of the Δ robot?

(b) (4 points) Write down the Lagrange function $\mathcal{L}$ of the Δ-robot, with the associated constraints $\mathbf{c}$.

(c) (2 points) What would be the differential index of the resulting DAE that would be obtained from $\mathcal{L}$ and $\mathbf{c}$?

(d) (2 points) What would be the consistency conditions of the Δ robot?

Solution:

(a) The generalized coordinates for this system can be defined as:

$$\mathbf{q} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \mathbf{p} \end{bmatrix}$$

where $\mathbf{p} \in \mathbb{R}^3$ is the position of the nacelle.

(b) Let us define the constraint function first. It must capture the knowledge that the nacelle position $\mathbf{p}$ is at a distance L of the arms end-points $\mathbf{p}_k$. It can be written as:

$$\mathbf{c}(\mathbf{q}) = \begin{bmatrix} \mathbf{c}_1 \\ \vdots \\ \mathbf{c}_3 \end{bmatrix} \quad \text{where} \quad \mathbf{c}_k = \|\mathbf{p} - \mathbf{p}_k\|^2 - L^2$$

To build the Lagrange function, we write the kinetic and potential energy of the robot:

$$T(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2}J\sum_{k=1}^{3}\alpha_k^2 + \frac{1}{2}m\mathbf{p}^\top\mathbf{p}$$
$$V(\mathbf{q}) = mg\mathbf{p}_3$$

The Lagrange function the reads as:

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\lambda}) = T - V + \boldsymbol{\lambda}^\top\mathbf{c}$$

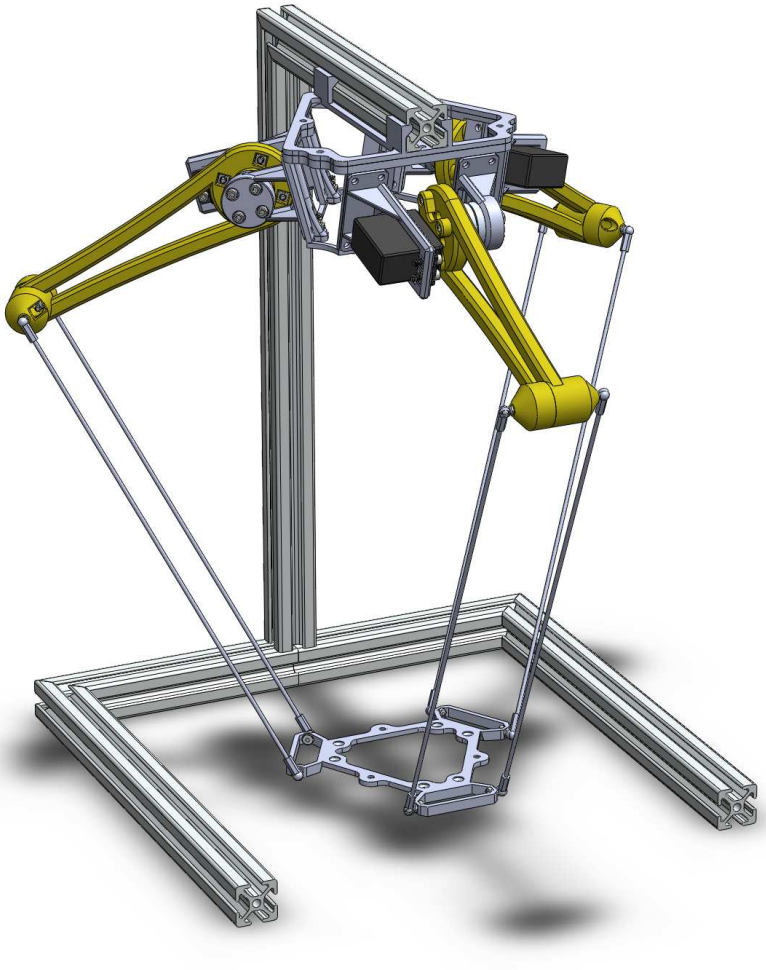

Figure 1: Illustration of the Δ-robot. The yellow arms (length l) are actuated by the motors on top (black boxes), and rotate

(c) Since the DAE would result from a constrained Lagrange formulation, it would have to be of index 3.

(d) The consistency conditions would need to impose:

$$C\left(\mathbf{q}, \dot{\mathbf{q}}\right) = \left[\begin{array}{c} \mathbf{c} \\ \dot{\mathbf{c}} \end{array} \right] = 0 \tag{2}$$

2. **System Identification**

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

$$y_k + 0.5y_{k-1} = u_{k-1} + 1.5u_{k-2} + e_k - 0.2e_{k-1} \tag{3}$$

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

$$x_k = A + e_k, \quad k = 0, \cdots, N-1 \tag{4}$$

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

$$\hat{A} = a\frac{1}{N}\sum_{k=0}^{N-1} x[k], \tag{5}$$

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

$$\text{mse}(\hat{A}) = \text{var}(\hat{A}) + b^2(\hat{A}), \tag{6}$$

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 $\hat{\boldsymbol{\theta}}$:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \quad \frac{1}{2}\left\|A\boldsymbol{\theta} - \mathbf{y}\right\|^2_{\Sigma^{-1}} \tag{7}$$

1. What are we assuming about the data when using a least-squares fitting problem like (16).
2. Explain in detail the meaning of formula (56) 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

$$G(z) = \frac{z^{-1} + 1.5z^{-2}}{1 + 0.5z{-}1} \quad H(z) = \frac{1 - 0.2z^{-1}}{1 + 0.5z{-}1}, \tag{8}$$

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

(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

$$\text{mse}(\hat{A}) = \frac{a^2\sigma^2}{N} + (a-1)^2A^2. \tag{9}$$

Differentiating the MSE with respect to a yields

$$\frac{\partial \text{mse}(\hat{A})}{\partial a} = \frac{2a\sigma^2}{N} + 2(a-1)A^2, \tag{10}$$

which setting to zero and solving for a yields the value

$$a_{opt} = \frac{A^2}{A^2 + \sigma^2/N}. \tag{11}$$

(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:

$$\mathbf{y} = A\boldsymbol{\theta}_{\text{true}} + \mathbf{e} \tag{12}$$

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

2. The expression:

$$\Sigma_{\hat{\boldsymbol{\theta}}} = \left(A^\top \Sigma^{-1} A\right)^{-1} \tag{13}$$

   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 (16) (see formula (55))

$$\hat{\boldsymbol{\theta}} = \left(A^\top \Sigma^{-1} A\right)^{-1} A^\top \Sigma^{-1} \mathbf{y} \tag{14}$$

   is itself a random variable (it is in fact a linear function of the random vector $\mathbf{y}$). Equation (13) provides the covariance of that random variable, i.e.

$$\mathbb{E}\left[\left(\hat{\boldsymbol{\theta}} - \boldsymbol{\mu}_{\hat{\boldsymbol{\theta}}}\right)\left(\hat{\boldsymbol{\theta}} - \boldsymbol{\mu}_{\hat{\boldsymbol{\theta}}}\right)^\top\right] \tag{15}$$

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

3. Problem (16) can also be written as

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \quad \frac{1}{2}\left(A\boldsymbol{\theta} - \mathbf{y}\right)^\top \Sigma^{-1}\left(A\boldsymbol{\theta} - \mathbf{y}\right) \tag{16}$$

   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:

$$\mathbf{F}\left(\dot{\mathbf{x}},\, \mathbf{x}, \mathbf{z}, \mathbf{u}\right) = 0 \tag{17}$$

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:

$$\mathbf{F} : \underbrace{\mathbb{R}^n \times \mathbb{R}^{n_\mathbf{x}} \times \mathbb{R}^{n_\mathbf{z}} \times \mathbb{R}^{n_\mathbf{u}}}_{\text{size of the arguments}} \quad \mapsto \quad \underbrace{\mathbb{R}^m}_{\text{“size of the function output”}} \tag{18}$$

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:

$$\dot{\mathbf{x}} = \mathbf{f}\left(\mathbf{x}, \mathbf{z}, \mathbf{u}\right) \tag{19a}$$
$$0 = \mathbf{g}\left(\mathbf{x}, \mathbf{z}, \mathbf{u}\right) \tag{19b}$$

Rewrite it in the fully-implicit form (17). What would function $\mathbf{F}$ be in this case?

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

$$\dot{\mathbf{x}} = A\mathbf{x} + \mathbf{b}z \tag{20a}$$
$$0 = \mathbf{g}\left(\mathbf{x}\right) \tag{20b}$$

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?

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

$$\dot{\mathbf{x}} = \mathbf{f}\left(\mathbf{x}, \mathbf{z}, \mathbf{u}\right) \tag{21a}$$
$$0 = \mathbf{g}\left(\mathbf{x}\right) \tag{21b}$$

Prove that (21) 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 (17) ought to deliver $\dot{\mathbf{x}}$ and $\mathbf{z}$ for given $\mathbf{x}$ and $\mathbf{u}$. That is, equation (17) 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:

$$\mathbf{F}\left(\dot{\mathbf{x}},\, \mathbf{x}, \mathbf{z}, \mathbf{u}\right) = \begin{bmatrix} \dot{\mathbf{x}} - \mathbf{f}\left(\mathbf{x}, \mathbf{z}, \mathbf{u}\right) \\ \mathbf{g}\left(\mathbf{x}, \mathbf{z}, \mathbf{u}\right) \end{bmatrix} = 0 \tag{22}$$

(c) We are dealing with a semi-explicit DAE, hence the index reduction requires time-differentiations of the algebraic equation (20b). The first step of the index reduction reads as:

$$\dot{\mathbf{x}} = A\mathbf{x} + \mathbf{b}z \tag{23a}$$

$$0 = \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}}\dot{\mathbf{x}} = \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}}(A\mathbf{x} + \mathbf{b}z) \tag{23b}$$

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

$$\mathbf{g}(\mathbf{x}(0)) = 0 \tag{24}$$

DAE (20) 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.

(d) DAE (21) is of index 2 if one step of index reduction yields an index-1 DAE. The first step of an index-reduction on (**??**) performs the operation:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{g}(\mathbf{x}) = \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}}\dot{\mathbf{x}} = \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}, \mathbf{z}, \mathbf{u}) \tag{25}$$

resulting in the DAE:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{z}, \mathbf{u}) \tag{26a}$$

$$0 = \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}, \mathbf{z}, \mathbf{u}) \tag{26b}$$

DAE (26) is of index 1 if the Jacobian of the new algebraic constraint $\frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}, \mathbf{z}, \mathbf{u})$ with respect to $\mathbf{z}$ is full rank, i.e. if the square matrix

$$\frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}}\frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{z}, \mathbf{u})}{\partial \mathbf{z}} \tag{27}$$

is full rank, or equivalently if:

$$\det\left(\frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}}\frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{z}, \mathbf{u})}{\partial \mathbf{z}}\right) \neq 0 \tag{28}$$

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

$$\frac{\partial \mathbf{r}(\mathbf{x})}{\partial \mathbf{x}} \Delta \mathbf{x} + \mathbf{r}(\mathbf{x}) = 0 \tag{29a}$$

$$\mathbf{x} \leftarrow \mathbf{x} + \alpha \Delta \mathbf{x} \tag{29b}$$

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:

$$\|\mathbf{x}_+ - \mathbf{x}_\star\| \leq c \|\mathbf{x} - \mathbf{x}_\star\|^2 \tag{30}$$

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 (29a) 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)

$$\mathbf{F}\left(\dot{\mathbf{x}},\, \mathbf{x}, \mathbf{z}\right) = 0 \tag{31}$$

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) How can one spot from the Butcher tableau if the RK method it describes is explicit or implicit?

(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:

$$\dot{\mathbf{x}} = \mathbf{f}\left(\mathbf{x}, \mathbf{z}, \mathbf{u}\right) \tag{32a}$$

$$0 = \mathbf{g}\left(\mathbf{x}, \mathbf{z}, \mathbf{u}\right) \tag{32b}$$

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

$$\begin{bmatrix} \mathbf{K} \\ \mathbf{z} \end{bmatrix} \leftarrow \begin{bmatrix} \mathbf{K} \\ \mathbf{z} \end{bmatrix} - \alpha M^{-1}\mathbf{r} \tag{33}$$

  Take the integrator step:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \sum_{i=1}^{s} b_i \mathbf{K}_i \tag{34}$$

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) Butcher tableaus describing explicit methods have to be (lower) triangular. If they are not then they describe an implicit 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. To detail this statement, we ought to specify that IRK methods need to solve linear systems of the form (34). 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.

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

$$\mathbf{r}\left(\mathbf{K}, \mathbf{z}, \mathbf{x}_k, \mathbf{u}(.)\right) = \begin{bmatrix} \mathbf{K}_1 - \mathbf{f}\left(\mathbf{x}_k + \Delta t \sum_{i=1}^{s} a_{1i}\mathbf{K}_i,\, \mathbf{z}_1,\, \mathbf{u}(t_k + c_1\Delta t)\right) \\ \mathbf{g}\left(\mathbf{x}_k + \Delta t \sum_{i=1}^{s} a_{1i}\mathbf{K}_i,\, \mathbf{z}_1,\, \mathbf{u}(t_k + c_1\Delta t)\right) \\ \vdots \\ \mathbf{K}_s - \mathbf{f}\left(\mathbf{x}_k + \Delta t \sum_{i=1}^{s} a_{si}\mathbf{K}_i,\, \mathbf{z}_s,\, \mathbf{u}(t_k + c_s\Delta t)\right) \\ \mathbf{g}\left(\mathbf{x}_k + \Delta t \sum_{i=1}^{s} a_{si}\mathbf{K}_i,\, \mathbf{z}_s,\, \mathbf{u}(t_k + c_s\Delta t)\right) \end{bmatrix} = 0 \tag{35}$$

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.

$$\mathbf{w} = \begin{bmatrix} \mathbf{K}_1 \\ \mathbf{z}_1 \\ \vdots \\ \mathbf{K}_s \\ \mathbf{z}_s \end{bmatrix} \tag{36}$$

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

$$\frac{\partial \mathbf{r}}{\partial \mathbf{w}} \tag{37}$$

Appendix: some possibly useful formula

- Lagrange mechanics is built on the equations:

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}} - \frac{\partial \mathcal{L}}{\partial \mathbf{q}} = \mathbf{Q}, \qquad \mathcal{L}\left(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{z}\right) = \mathcal{T} - \mathcal{V} - \mathbf{z}^\top \mathbf{C}, \qquad \mathbf{C} = 0, \qquad \langle \delta\mathbf{q},\, \mathbf{Q}\rangle = \delta W,\, \forall\, \delta\mathbf{q} \tag{38}$$

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

$$\mathcal{T} = \frac{1}{2} m \dot{\mathbf{p}}^\top \dot{\mathbf{p}}, \qquad \mathcal{V} = mg\mathbf{p}_3 \tag{39}$$

  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:

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

$$W\dot{\mathbf{v}} + \frac{\partial \mathbf{C}}{\partial \mathbf{q}}^\top \mathbf{z} = \mathbf{Q} - \frac{\partial \mathcal{V}}{\partial \mathbf{q}}^\top \tag{40b}$$

$$0 = \mathbf{C}\left(\mathbf{q}\right) \tag{40c}$$

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

$$\mathbf{r}\left(\mathbf{y}, \mathbf{z}\right) = 0 \tag{41}$$

  "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

$$\mathbf{r}\left(\mathbf{y}, \phi\left(\mathbf{y}\right)\right) = 0 \tag{42}$$

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

$$\frac{\partial \mathbf{z}}{\partial \mathbf{y}} = -\frac{\partial \mathbf{r}}{\partial \mathbf{z}}^{-1} \frac{\partial \mathbf{r}}{\partial \mathbf{y}} \tag{43}$$

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

$$\mathbf{x} \leftarrow \mathbf{x} - \alpha \frac{\partial \mathbf{r}}{\partial \mathbf{x}}^{-1} \mathbf{r} \tag{44}$$

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

- Runge-Kutta methods are described by:

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

$$\mathbf{K}_j = \mathbf{f}\left(\mathbf{x}_k + \Delta t \sum_{i=1}^{s} a_{ji}\mathbf{K}_i,\, \mathbf{u}\left(t_k + c_j \Delta t\right)\right), \quad j = 1, \dots, s \tag{45a}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \sum_{i=1}^{s} b_i \mathbf{K}_i \tag{45b}$$

- 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:

$$\dot{\mathbf{x}}(t_k + \Delta t \cdot \tau) \approx \dot{\hat{\mathbf{x}}}(t_k + \Delta t \cdot \tau) = \sum_{i=1}^{s} \mathbf{K}_i \ell_i(\tau), \quad \tau \in [0,\, 1] \tag{46}$$

$$\mathbf{x}(t_k + \Delta t \cdot \tau) \approx \hat{\mathbf{x}}(t_k + \Delta t \cdot \tau) = \mathbf{x}_k + \Delta t \sum_{i=1}^{s} \mathbf{K}_i L_i(\tau) \tag{47}$$

where the Lagrange polynomials are given by:

$$\ell_i(\tau) = \prod_{j=1, j\neq i}^{s} \frac{\tau - \tau_j}{\tau_i - \tau_j}, \quad \text{and} \quad L_i(\tau) = \int_0^{\tau} \ell_i(\xi) \mathrm{d}\xi \tag{48}$$

The Lagrange polynomials satisfy the conditions of

$$\text{Orthogonality:} \quad \int_0^1 \ell_i(\tau)\ell_j(\tau)\,\mathrm{d}\tau = 0 \quad \text{for} \quad i \neq j \tag{49a}$$

$$\text{Punctuality:} \quad \ell_i(\tau_j) = \begin{cases} 1 & \text{if} \quad j = i \\ 0 & \text{if} \quad j \neq i \end{cases} \tag{49b}$$

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

$$\dot{\hat{\mathbf{x}}}(t_k + \Delta t \cdot \tau_j) = \mathbf{f}\left(\hat{\mathbf{x}}(t_k + \Delta t \cdot \tau_j),\, \mathbf{u}\left(t_k + \Delta t \cdot \tau_j\right)\right), \qquad \text{in the explicit ODE case} \tag{50a}$$

$$\mathbf{F}\left(\dot{\hat{\mathbf{x}}}(t_k + \Delta t \cdot \tau_j), \hat{\mathbf{x}}(t_k + \Delta t \cdot \tau_j),\, \mathbf{u}\left(t_k + \Delta t \cdot \tau_j\right)\right) = 0, \qquad \text{in the implicit ODE case} \tag{50b}$$

$$\mathbf{F}\left(\dot{\hat{\mathbf{x}}}(t_k + \Delta t \cdot \tau_j), \hat{\mathbf{z}}_j, \hat{\mathbf{x}}(t_k + \Delta t \cdot \tau_j),\, \mathbf{u}\left(t_k + \Delta t \cdot \tau_j\right)\right) = 0, \qquad \text{in the fully-implicit DAE case} \tag{50c}$$

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

$$P_s\left(\tau\right) = \frac{1}{s!}\frac{\mathrm{d}^s}{\mathrm{d}\tau^s}\left[\left(\tau^2 - \tau\right)^s\right] \tag{51}$$

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

$$\max_{\boldsymbol{\theta}} \quad \mathbb{P}\left[e_k = y_k - \hat{y}_k \quad \text{for} \quad k = 1, \ldots, N \mid \boldsymbol{\theta}\right] \tag{52}$$

If the noise sequence is uncorrelated, then

$$\mathbb{P}\left[e_k = y_k - \hat{y}_k \quad \text{for} \quad k = 0, \ldots, N \mid \boldsymbol{\theta}\right] = \prod_{k=1}^{N} \mathbb{P}\left[e_k = y_k - \hat{y}_k \,|\, \boldsymbol{\theta}\,\right] \tag{53}$$

- The solution of a linear least-squares problem

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{2}\left\|A\boldsymbol{\theta} - \mathbf{y}\right\|_{\Sigma_e^{-1}}^2 \tag{54}$$

reads as:

$$\hat{\boldsymbol{\theta}} = \left(A^\top \Sigma_e^{-1} A\right)^{-1} A^\top \Sigma_e^{-1} \mathbf{y} \tag{55}$$

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

$$\Sigma_{\hat{\boldsymbol{\theta}}} = \left(A^\top \Sigma_e^{-1} A\right)^{-1} \tag{56}$$

- 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

$$H(z)\hat{y}(z) = G(z)u(z) + (H(z) - 1)y(z) \tag{57}$$

- The Gauss-Newton approximation in an optimization problem

$$\min_{\mathbf{x}} \quad J(\mathbf{x}) = \frac{1}{2} \|\mathbf{R}(\mathbf{x})\|^2 \tag{58}$$

uses the approximation:

$$\frac{\partial^2 J}{\partial \mathbf{x}^2} \approx \frac{\partial R}{\partial \mathbf{x}}^\top \frac{\partial R}{\partial \mathbf{x}} \tag{59}$$

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

$$\mathbf{x}(t) = e^{At}\mathbf{x}(0) + \int_0^t e^{A(t-\tau)} B\mathbf{u}(\tau) \mathrm{d}\tau \tag{60}$$

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

$$G(s) = C\left(sI - A\right)^{-1} B + D \tag{61}$$