



Praktikumsbericht

Modellbildung und Simulation

3. Praktikumsaufgabe: Numerische Integration von Differenzial-Algebraischen-Systemen

Gruppe 10:

Konstantin Kuhl

Nils Leimbach

Sebastian Schwabe

Konstantin Wrede

Eingereicht am: 25. Juni 2020

Inhaltsverzeichnis

1	Aufgabe: Semi-explizites Differenzial-Algebraisches-System	3
1.1	Standardform des semi-expliziten DAE-Systems	3
1.2	Differenzieller Index des Modells	3
2	Aufgabe: Numerische Integration mit impliziten Verfahren	4
2.1	Trapez-Methode mit Newton-Raphson-Verfahren	4
2.2	Anfangswerte der algebraischen Variablen	5
2.3	Simulationsverifikation mit Sollverlauf der Ausgangsspannung	5
2.4	Konvergenzverhalten des Newton-Raphson-Verfahrens	6
2.5	Untersuchung der numerischen Genauigkeit	7

1 Aufgabe: Semi-explizites Differenzial-Algebraisches-System

1.1 Standardform des semi-expliziten DAE-Systems

Für das gegebene RC-Netzwerk ergibt sich die Standardform eines semi-expliziten DAE-Systems wie folgt:

$$\dot{\underline{x}} = \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \underline{f}(\underline{z}) = \begin{pmatrix} \frac{z_1}{C_1} \\ \frac{z_2}{C_2} \end{pmatrix} \quad \text{und} \quad \underline{g}(\underline{x}, \underline{z}, u) = \begin{pmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \end{pmatrix} = \begin{pmatrix} Rz_4 - z_3 \\ x_1 + z_3 - u \\ x_1 - x_2 \\ z_1 + z_2 - z_4 \end{pmatrix} = \underline{0}.$$

1.2 Differenzieller Index des Modells

Der differenzielle Index des DAE-Systems wird im Folgenden mit der Methode der Indexreduktion bestimmt. Zunächst können die algebraischen Gleichungen g_1 , g_2 und g_4 zusammengefasst werden, so dass sich damit ein vereinfachter Vektor $\tilde{\underline{g}}$ ergibt:

$$\tilde{\underline{g}} = \begin{pmatrix} x_1 + R(z_1 + z_2) - u \\ x_1 - x_2 \end{pmatrix} = \underline{0}.$$

Dies entspricht noch keinem explizitem DGL-System, der Index ist also größer als null. Durch einfaches Ableiten und Einsetzen der rechten Seite von $\underline{f}(\underline{z})$ ergeben sich zwei Gleichungen, wobei $\dot{\tilde{\underline{g}}}_2$ noch keiner DGL entspricht:

$$\dot{\tilde{\underline{g}}} = \begin{pmatrix} \frac{1}{C_1}z_1 + R(\dot{z}_1 + \dot{z}_2) - \dot{u} \\ \frac{1}{C_1}z_1 - \frac{1}{C_2}z_2 \end{pmatrix} = \underline{0}.$$

Deshalb ist der Index dieses Systems größer als eins und $\dot{\tilde{\underline{g}}}_2$ wird erneut abgeleitet:

$$\ddot{\tilde{\underline{g}}}_2 = \frac{1}{C_1}\dot{z}_1 - \frac{1}{C_2}\dot{z}_2 = 0 \Rightarrow \dot{\tilde{\underline{g}}}_1 = \frac{1}{C_1}z_1 + R\left(\dot{z}_1 + \frac{C_2}{C_1}\dot{z}_1\right) - \dot{u} = 0.$$

Damit entsteht eine explizite DGL in z_1 , die ebenso auf gesamt \underline{z} übertragen werden kann. Der differenzielle Index i , also die minimale Anzahl zeitlicher Differenziationen der algebraischen Gleichungen um das DAE-System als explizites DGL-System darzustellen, ist zu $i = 2$ bestimmt.

2 Aufgabe: Numerische Integration mit impliziten Verfahren

2.1 Trapez-Methode mit Newton-Raphson-Verfahren

Für die implizite numerische Integration mit dem Trapezverfahren ergibt sich nach MODSIM08 Seite 8-9 die folgende Gleichung:

$$\varphi(\underline{p}) = \begin{pmatrix} \hat{x}_{i+1} - \hat{x}_i - \frac{h}{2} [f(\hat{z}_i) + f(\hat{z}_{i+1})] \\ \underline{g}(\hat{x}_{i+1}, \hat{z}_{i+1}, u_{i+1}) \end{pmatrix} = \underline{0}.$$

Dabei wird das Newton-Raphson-Verfahren iterativ über den Index n ausgeführt:

$$\underline{p}_{n+1} = \underline{p}_n - \underline{J}^{-1}(\underline{p}_n) \cdot \varphi(\underline{p}_n) \text{ mit } \underline{p}_n = (\hat{x}_{i+1}, \hat{z}_{i+1})_n^T.$$

Die Iteration wird abgebrochen, falls eine maximale Anzahl an Schritten überstiegen wurde, beziehungsweise die Differenz aufeinanderfolgender Iterationen innerhalb einer Toleranzschranke ε liegt:

$$\|\underline{p}_{n+1} - \underline{p}_n\|_\infty = \|\underline{J}^{-1}(\underline{p}_n) \cdot \varphi(\underline{p}_n)\|_\infty < \varepsilon$$

Als Ableitung im mehrdimensionalen wird eine Jacobi-Matrix genutzt:

$$\underline{J}(\underline{p}) = \begin{pmatrix} \left[I - \frac{h}{2} \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \right] \cdot \frac{-h}{2} \cdot \begin{bmatrix} \frac{\partial f_1}{\partial z_1} & \frac{\partial f_1}{\partial z_2} & \frac{\partial f_1}{\partial z_3} & \frac{\partial f_1}{\partial z_4} \\ \frac{\partial f_2}{\partial z_1} & \frac{\partial f_2}{\partial z_2} & \frac{\partial f_2}{\partial z_3} & \frac{\partial f_2}{\partial z_4} \end{bmatrix} \\ \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} \\ \frac{\partial g_3}{\partial x_1} & \frac{\partial g_3}{\partial x_2} \\ \frac{\partial g_4}{\partial x_1} & \frac{\partial g_4}{\partial x_2} \end{bmatrix} \begin{bmatrix} \frac{\partial g_1}{\partial z_1} & \frac{\partial g_1}{\partial z_2} & \frac{\partial g_1}{\partial z_3} & \frac{\partial g_1}{\partial z_4} \\ \frac{\partial g_2}{\partial z_1} & \frac{\partial g_2}{\partial z_2} & \frac{\partial g_2}{\partial z_3} & \frac{\partial g_2}{\partial z_4} \\ \frac{\partial g_3}{\partial z_1} & \frac{\partial g_3}{\partial z_2} & \frac{\partial g_3}{\partial z_3} & \frac{\partial g_3}{\partial z_4} \\ \frac{\partial g_4}{\partial z_1} & \frac{\partial g_4}{\partial z_2} & \frac{\partial g_4}{\partial z_3} & \frac{\partial g_4}{\partial z_4} \end{bmatrix} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \frac{-h}{2C_1} & 0 & 0 & 0 \\ 0 & 1 & 0 & \frac{-h}{2C_2} & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & R \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & -1 \end{pmatrix}.$$

2.2 Anfangswerte der algebraischen Variablen

In der Aufgabenstellung ist der Initialzustand des Systems gegeben mit:

$$\underline{x}(0) = \begin{pmatrix} x_1(0) \\ x_2(0) \end{pmatrix} = \underline{0}.$$

Demnach folgt für die algebraischen Variablen durch sequentielles Einsetzen in die Komponenten von \underline{g} der folgende Anfangszustand:

$$\begin{aligned} z_4(0) &= \frac{u(0)}{R} = \frac{10}{R} \\ z_3(0) &= u(0) = 10 \\ z_2(0) &= \frac{C_2}{C_1 + C_2} z_4(0) = \frac{10 C_2}{R(C_1 + C_2)} \\ z_1(0) &= \frac{C_1}{C_1 + C_2} z_4(0) = \frac{10 C_1}{R(C_1 + C_2)}. \end{aligned}$$

2.3 Simulationsverifikation mit Sollverlauf der Ausgangsspannung

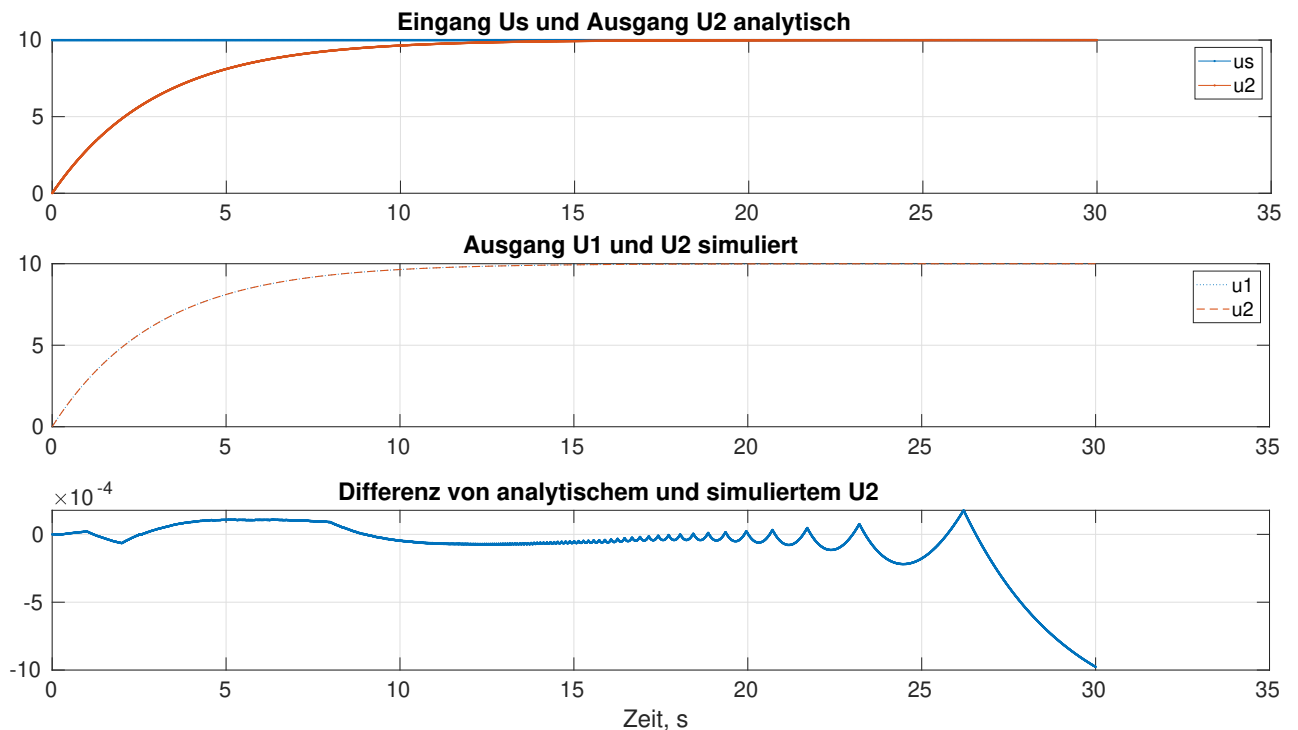


Abbildung 2.1: Simulationsverifikation für $h = 0.001$ s, $\varepsilon = 10^{-5}$

Durch die Betrachtung eines Spannungsteilers im Bildbereich kann zur analytischen Lösung eine Übertragungsfunktion ermittelt werden:

$$G_U^{X_2}(s) = \frac{\frac{1}{sC_1} \parallel \frac{1}{sC_2}}{R + \frac{1}{sC_1} \parallel \frac{1}{sC_2}} = \frac{1}{1 + R(sC_1 + sC_2)}$$

$$\text{für } U(s) = \frac{10}{s} \Rightarrow X_2(s) = \frac{10}{s(1 + sR(C_1 + C_2))}.$$

Durch Transformation in den Zeitbereich ergibt sich der Zeitverlauf der Ausgangsspannung zu:

$$x_2(t) = 10 \left(1 - e^{-\frac{t}{R(C_1 + C_2)}} \right).$$

Das Simulationsergebnis in Abbildung 2.1 zeigt nur geringe Abweichungen zwischen simuliertem und analytischem Sollverlauf von $x_2 = u_2$ in der Größenordnung 10^{-4} für eine Schrittweite $h = 0.001$ s und eine Toleranzschranke $\varepsilon = 10^{-5}$, der sich allerdings zum Ende des Simulationsteitraums im Aufschwingen befindet. Außerdem bestätigt die strukturelle Parallelität von u_1 und u_2 (auch $g_3 = x_1 - x_2 = 0$) den sich überlappenden Simulationsverlauf.

2.4 Konvergenzverhalten des Newton-Raphson-Verfahrens

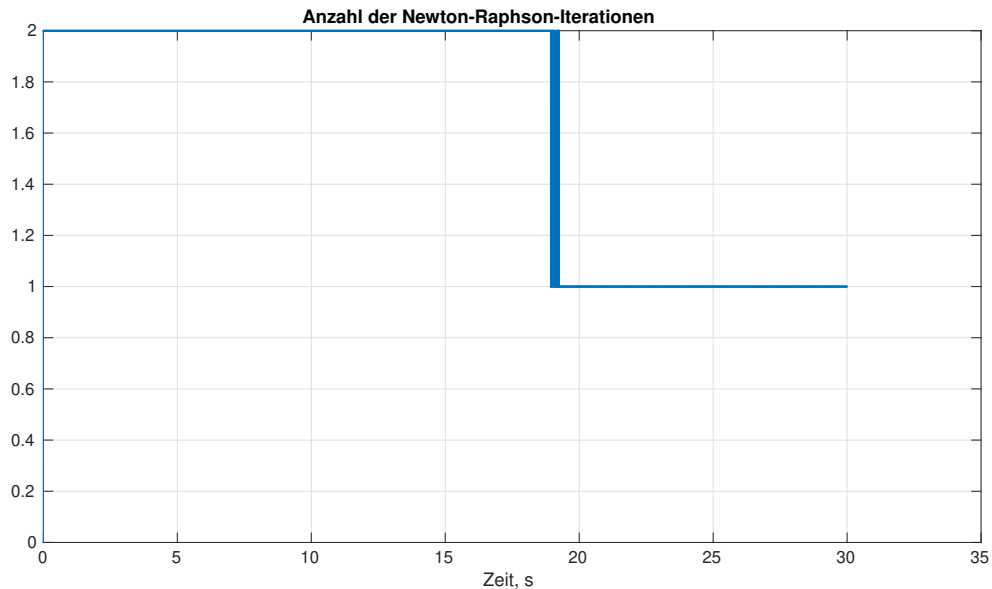


Abbildung 2.2: Newton-Raphson-Iterationen für $h = 0.001$ s, $\varepsilon = 10^{-5}$

Abbildung 2.2 zeigt, dass die Anzahl der notwendigen Iterationsschritte im gegebenen Simulationsrahmen für eine Schranke von $\varepsilon = 10^{-5}$ und einer Schrittweite von $h = 0.001$ s einen maximalen Wert von 2 aufweist. Während eines zum Simulationsende stets kleiner werdenden Anstieges von u_2 genügt sogar der erste Iterationsschritt, um innerhalb der Toleranz zu bleiben.

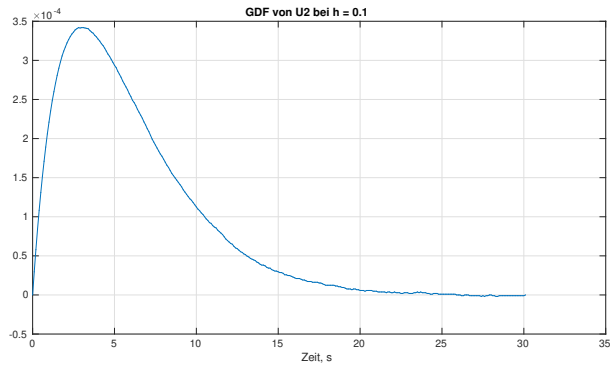
2.5 Untersuchung der numerischen Genauigkeit

Ein grundlegendes Kriterium für die Konvergenz und im Übertragenen Sinne auch für die numerische Genauigkeit ist die Existenz der Inversen von $\underline{J}(\underline{p})$, welche für das Newton-Raphson-Verfahren benötigt wird. Demnach muss für die Determinante $\det(\underline{J}(\underline{p})) \neq 0$ gelten. In diesem Index-2-System ergibt sich dabei folgender Wert:

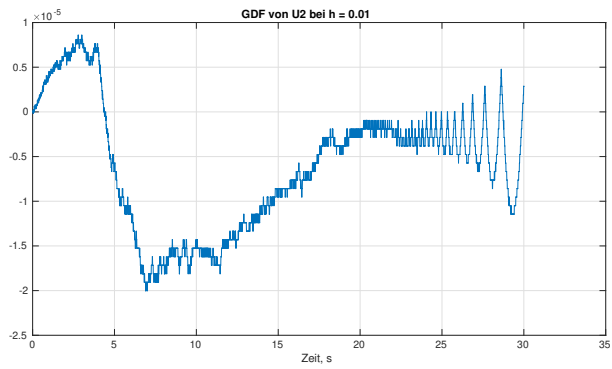
$$\det(\underline{J}(\underline{p})) = \frac{h}{2} \frac{\partial f}{\partial \underline{z}} \frac{\partial g}{\partial \underline{x}} = -\frac{h(2R(C_1 + C_2) + h)}{4C_1C_2}.$$

Für kleine Schrittweiten $h \rightarrow 0$ wird das Newtonverfahren wie bei Index-2-Systemem typisch also zunehmend divergent, weil sich die Jacobimatrix einer singulären Matrix annähert. Dies führt zu einer steigenden numerischen Ungenauigkeit, bei der bereits kleine Rundungsfehler zu einer Verschlechterung der numerischen Konvergenz führen.

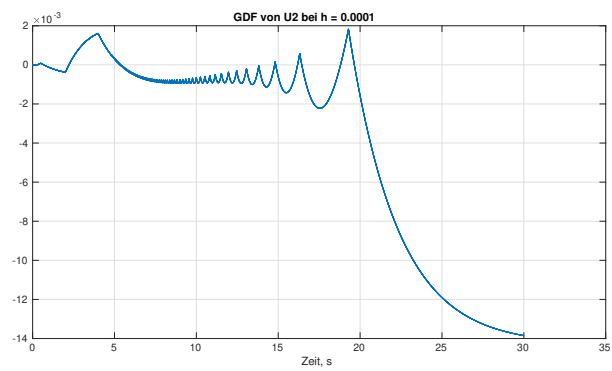
Durch die geringe Auflösungen der Gleitkomma-Variablen als Single führen Rundungsfehler schon bei etwa $h = 0.001$ s zu einem wachsenden globalen Diskretisierungsfehler, wie in Abbildung 2.3 erkennbar ist. Dabei kommt es bei $h = 10^{-5}$ s zu einem GDF von $|d| = 0.14$. Zudem steigt die Simulationsdauer auch stark durch die feste kleinere Schrittweite an.



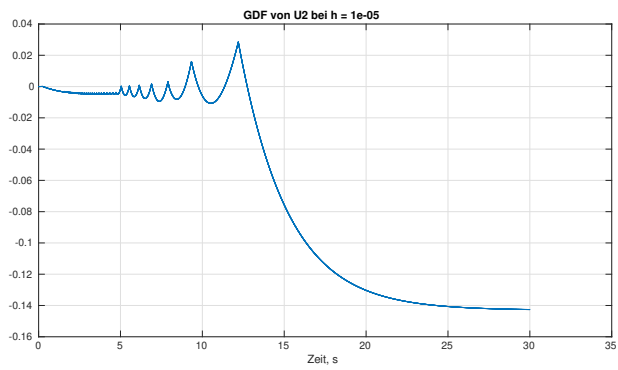
(a) $h = 0.1\text{s}$



(b) $h = 0.01\text{s}$



(c) $h = 10^{-4}\text{s}$



(d) $h = 10^{-5}\text{s}$

Abbildung 2.3: Globale Diskretisierungsfehler bei verschiedenen Schrittweiten