

TECHNISCHE UNIVERSITÄT DRESDEN

FAKULTÄT ELEKTROTECHNIK UND INFORMATIONSTECHNIK

**Elemente der Modellbildung und
Simulationstechnik**

PRAKTIKUMSAUFGABE III

Gruppe 11

Cao,Bozhi Gao,Yue Jia,Xuehua Zhu,Jinyao

Dresden, 8. Juli 2018

1. Aufgabe

1.1 DAE-System

Konstitutive Gleichung:

$$\begin{aligned}\dot{u}_1 &= \frac{1}{C_1} \cdot i_1 \\ \dot{u}_2 &= \frac{1}{C_2} \cdot i_2\end{aligned}$$

Zwangsbedingungen:

$$\begin{aligned}i_s - i_1 - i_2 &= 0 \\ u_s - u_R - u_1 &= 0 \\ u_1 - u_2 &= 0 \\ u_R - R \cdot i_s &= 0\end{aligned}$$

Definition:

$$x_1 = u_1, \quad x_2 = u_2, \quad z_1 = i_1, \quad z_2 = i_2, \quad z_3 = u_R, \quad z_4 = i_s, \quad u = u_s$$

\Rightarrow semi-explizites DAE-System:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{z}) \Leftrightarrow \begin{cases} \dot{x}_1 = \frac{1}{C_1} \cdot z_1 \\ \dot{x}_2 = \frac{1}{C_2} \cdot z_2 \end{cases} \quad (1.1)$$

$$(1.2)$$

$$\mathbf{0} = \mathbf{g}(\mathbf{x}, \mathbf{z}) \Leftrightarrow \begin{cases} x_1 - x_2 = 0 \\ u - z_3 - x_1 = 0 \\ z_3 - R \cdot z_4 = 0 \\ z_4 - z_1 - z_2 = 0 \end{cases} \quad \begin{aligned} (1.3) \\ (1.4) \\ (1.5) \\ (1.6) \end{aligned}$$

1.2 Bestimmung des differentiellen Index

Nach Gl.(1.6) Gl.(1.4) und Gl.(1.5) können die Variablen z_3, z_4 eliminiert werden, die umgeformten Zwangsbedingungen sind:

$$x_1 - x_2 = 0 \quad (1.7)$$

$$u - R \cdot (z_1 + z_2) - x_1 = 0 \quad (1.8)$$

offensichtlich aus Gl.(1.7) Gl.(1.8) kann keine explizite DGL erzeugt werden. $\Rightarrow \text{Index} > 0$.

1.Differenziation

Durch Zeitableitung von Gl.(1.7) Gl.(1.8) erhält man:

$$\dot{x}_1 - \dot{x}_2 = 0 \quad (1.9)$$

$$\dot{u} - R \cdot (\dot{z}_1 + \dot{z}_2) - \dot{x}_1 = 0 \quad (1.10)$$

mit Gl.(1.1) Gl.(1.2):

$$\frac{1}{C_1} \cdot z_1 - \frac{1}{C_2} \cdot z_2 = 0 \quad (1.11)$$

$$\dot{u} - R \cdot (\dot{z}_1 + \dot{z}_2) - \frac{1}{C_1} \cdot z_1 = 0 \quad (1.12)$$

Gl.(1.11) in Gl.(1.12) einsetzen:

$$\dot{u} - R \cdot (\dot{z}_1 + \dot{z}_2) - \frac{1}{C_2} \cdot z_2 = 0 \quad (1.13)$$

bisher ist die Zusammenhang zwischen \dot{z}_1 und \dot{z}_2 noch unbekannt, deshalb entsteht keine explizite DGL.
 $\Rightarrow \text{Index} > 1$.

2.Differenziation

Um die Beziehung zwischen \dot{z}_1 und \dot{z}_2 zu finden, kann man die Gl.(1.11) differenzieren:

$$\frac{1}{C_1} \cdot \dot{z}_1 - \frac{1}{C_2} \cdot \dot{z}_2 = 0 \quad (1.14)$$

setzt man die Gl.(1.14) in Gl.(1.13) ein:

$$\begin{aligned} \dot{u} - R \cdot \left(\frac{C_1}{C_2} + 1 \right) \dot{z}_2 - \frac{1}{C_2} \cdot z_2 &= 0 \\ \Rightarrow \dot{z}_2 &= -\frac{1}{R \cdot (C_1 + C_2)} \cdot z_2 + \frac{C_2}{R \cdot (C_1 + C_2)} \cdot \dot{u} \end{aligned} \quad (1.15)$$

mit Gl.(1.1) Gl.(1.2) Gl.(1.9) und Gl.(1.15) entsteht ein explizites DGL-System:

$$\tilde{\mathbf{f}}(z_2, \dot{u}) \Leftrightarrow \begin{cases} \dot{x}_1 = \frac{1}{C_2} \cdot z_2 \\ \dot{x}_2 = \frac{1}{C_2} \cdot z_2 \\ \dot{z}_2 = -\frac{1}{R \cdot (C_1 + C_2)} \cdot z_2 + \frac{C_2}{R \cdot (C_1 + C_2)} \cdot \dot{u} \end{cases}$$

jetzt ist der Index des originalen DAE-Systems auf Null reduziert. $\Rightarrow \text{Index} = 2$.

Mit den Zwangsbedingungen Gl.(1.6) Gl.(1.5) und Gl.(1.11) können die übrigen Variablen regeneriert werden:

$$\begin{cases} z_1 = \frac{C_1}{C_2} \cdot z_2 \end{cases} \quad (1.16)$$

$$\begin{cases} z_4 = z_1 + z_2 \end{cases} \quad (1.17)$$

$$\begin{cases} z_3 = R \cdot z_4 \end{cases} \quad (1.18)$$

2. Aufgabe

2.1 Trapezmethode mit Newton-Rapson-Verfahren

Problemstellung:

Nullstelle von $\varphi_{k+1}(\mathbf{p}_{k+1})$ iterativ zu finden, wobei:

$$\varphi_{k+1}(\mathbf{p}_{k+1}) := \begin{pmatrix} \varphi_{1,k+1} \\ \varphi_{2,k+1} \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{x}}_{k+1} - \hat{\mathbf{x}}_k - \frac{h}{2} [\mathbf{f}(\hat{\mathbf{z}}_k) + \mathbf{f}(\hat{\mathbf{z}}_{k+1})] \\ \mathbf{g}(\hat{\mathbf{x}}_{k+1}, \hat{\mathbf{z}}_{k+1}) \end{pmatrix}$$

$$\mathbf{p}_{k+1} = (\hat{\mathbf{x}}_{k+1}, \hat{\mathbf{z}}_{k+1})^\top$$

Lösung:

Iterationsvorschrift nach Newton-Rapson-Verfahren:

$$\mathbf{p}_{k+1,i+1} = \mathbf{p}_{k+1,i} - \mathbf{J}(\mathbf{p}_{k+1,i})^{-1} \cdot \varphi_{k+1}(\mathbf{p}_{k+1,i}) \quad (1.19)$$

Iteration über i bis:

$$\|\mathbf{p}_{k+1,i+1} - \mathbf{p}_{k+1,i}\|_\infty < \varepsilon$$

wobei \mathbf{J} ist die Jacobi-Matrix von φ_{k+1} an der Stelle $\mathbf{p}_{k+1,i}$:

$$\mathbf{J}(\mathbf{p}_{k+1,i}) = \left. \frac{\partial \varphi_{k+1}}{\partial \mathbf{p}_{k+1}} \right|_{\mathbf{p}_{k+1} = \mathbf{p}_{k+1,i}} \quad (1.20)$$

nach Gl.(1.1)~Gl.(1.5) und Gl.(1.20), erhält man:

$$\mathbf{J} = \begin{pmatrix} 1 & 0 & -\frac{h}{2C_1} & 0 & 0 & 0 \\ 0 & 1 & 0 & -\frac{h}{2C_2} & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -R \\ 0 & 0 & -1 & -1 & 0 & 1 \end{pmatrix} \quad (1.21)$$

nach Gl.(1.19) muss die Jacobi-Matrix invertierbar(regulär) sein, deshalb muss:

$$\det(\mathbf{J}(\mathbf{p}_{k+1,i})) = \frac{h \cdot [h + 2R \cdot (C_1 + C_2)]}{4C_1C_2} \neq 0 \quad (1.22)$$

offensichtlich für jede $h \neq 0$ die Bedingung(1.22) gilt.

2.2 Bestimmung der Anfangswerten

Weil es sich um ein Index-2 DAE-System handelt, kann \mathbf{x}_0 und \mathbf{z}_0 nicht frei gewählt werden.

Aus der vorgegebenen Bedingung:

$$\mathbf{x}_0 = \begin{pmatrix} x_{10} \\ x_{20} \end{pmatrix} = \mathbf{0}$$

mit den Zwangsbedingungen Gl.(1.4) und Gl.(1.16)~(1.18), erhält man die Anfangswerten von \mathbf{z} :

$$\mathbf{z}_0 = \begin{pmatrix} z_{10} \\ z_{20} \\ z_{30} \\ z_{40} \end{pmatrix} = \begin{pmatrix} \frac{C_1}{C_1+C_2} \cdot \frac{u_0}{R} \\ \frac{C_2}{C_1+C_2} \cdot \frac{u_0}{R} \\ u_0 \\ \frac{u_0}{R} \end{pmatrix}$$

2.3 Analytische Lösung von $u_2(t)$

Aus dem gegebenen RC-Netzwerk, erhält man:

$$(C_1 + C_2) \cdot \dot{u}_2(t) = i_s(t)$$

$$R \cdot i_s(t) + u_2(t) = u_s(t)$$

$$\Rightarrow \dot{u}_2(t) + \underbrace{\frac{1}{R(C_1 + C_2)}}_{\tau} \cdot u_2(t) = \underbrace{\frac{1}{R(C_1 + C_2)}}_{\tau} \cdot u_s(t)$$

Lösung der DGL mit der Anfangsbedingung $u_2(0) = 0$ und $u_s(0) = 10$:

$$u_2(t) = 10 \cdot \left(1 - e^{-\frac{t}{\tau}}\right)$$

2.4 Kriterium für die numerische Genauigkeit

Die Singularität der Jacobi-Matrix spielt eine wichtige Rolle beim Newton-Raphson-Verfahren für die numerische Genauigkeit der Lösung. Normalerweise, fördert man eine hohe numerische Genauigkeit, soll eine möglichst kleine Schrittweite verwenden. Aus der Jacobi-Matrix(1.21), wenn $h \ll 1$, sind jedoch die dritte und die vierte Spalte der Jacobi-Matrix fast linear abhängig, also führt zu einer singulären Jacobi-Matrix. In gewissem Maße kann die Determinante der Jacobi-Matrix als ein Kriterium der Singularität dienen. Im Experiment haben wir gefunden dass, dies ist nicht genug, da eine Matrix kann skaliert beliebig werden, dadurch ändert sich auch die Determinante, aber die Singularität hat sich nicht verändert. Ein besseres Kriterium für die numerische Genauigkeit bzw. die Singularität der Jacobi-Matrix ist die **Konditionszahl** der Jacobi-Matrix. Die Größe der Konditionszahl beschreibt die Zusammenhang der relativen Fehler an der Ein- und Ausgangsgröße. Eine sehr große Konditionszahl($\text{cond}(\mathbf{J}) \gg 1$) impliziert eine hohe Empfindlichkeit der Lösung gegenüber die Eingangsfehler(z.B. Rundungsfehler bei der 32-Bit-Gleitkommazahl), und führt zu einer **schlecht konditionierten** Jacobi-Matrix. Der Solver besitzt deshalb eine schlechte numerische Genauigkeit. Dagegen eine Konditionszahl nahe 1 führt zu einer **gut konditionierten** Jacobi-Matrix. In diesem Experiment gibt es ein Dilemma zwischen der Schrittweite und der Konditionszahl der Jacobi-Matrix, welcher die numerische Genauigkeit der Simulation beschränkt.

2.5 Simulationsergebnisse

Bedingungen:

Fehlerschranke des Newton-Raphson-Verfahrens:

$$\varepsilon = 1 \times 10^{-5}$$

Schrittweite der Trapezmethode:

$$h = 1 \times 10^{-2} \text{s}$$

Simulieren mit Gleitkommazahl einfacher Genauigkeit(32-Bit).

Ergebnisse:

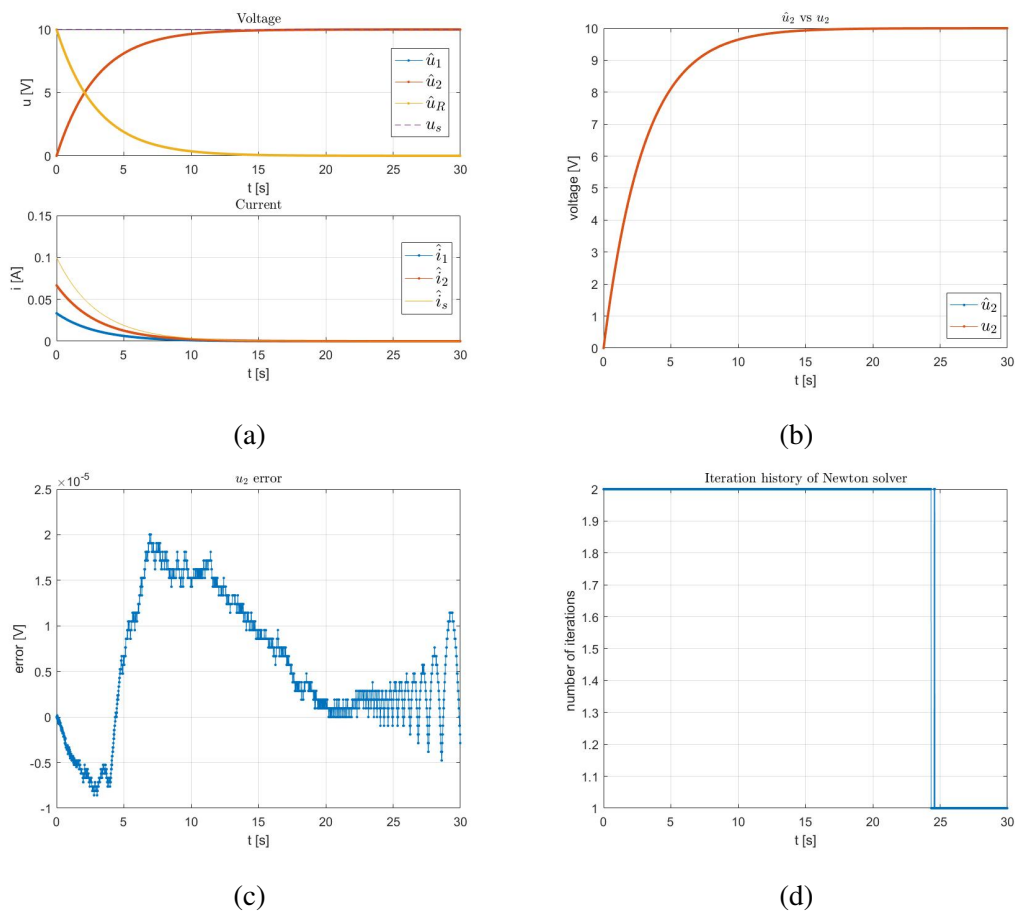


Abbildung 1: (a) Verlauf aller Spannungen und Ströme; (b) Simuliert-/Sollverlauf von $u_2(t)$; (c) Differenz zum Sollverlauf von $u_2(t)$; (d) Verlauf der Iterationsschritte beim Newton-Raphson-Verfahren;

Untersuchung der numerischen Genauigkeit:

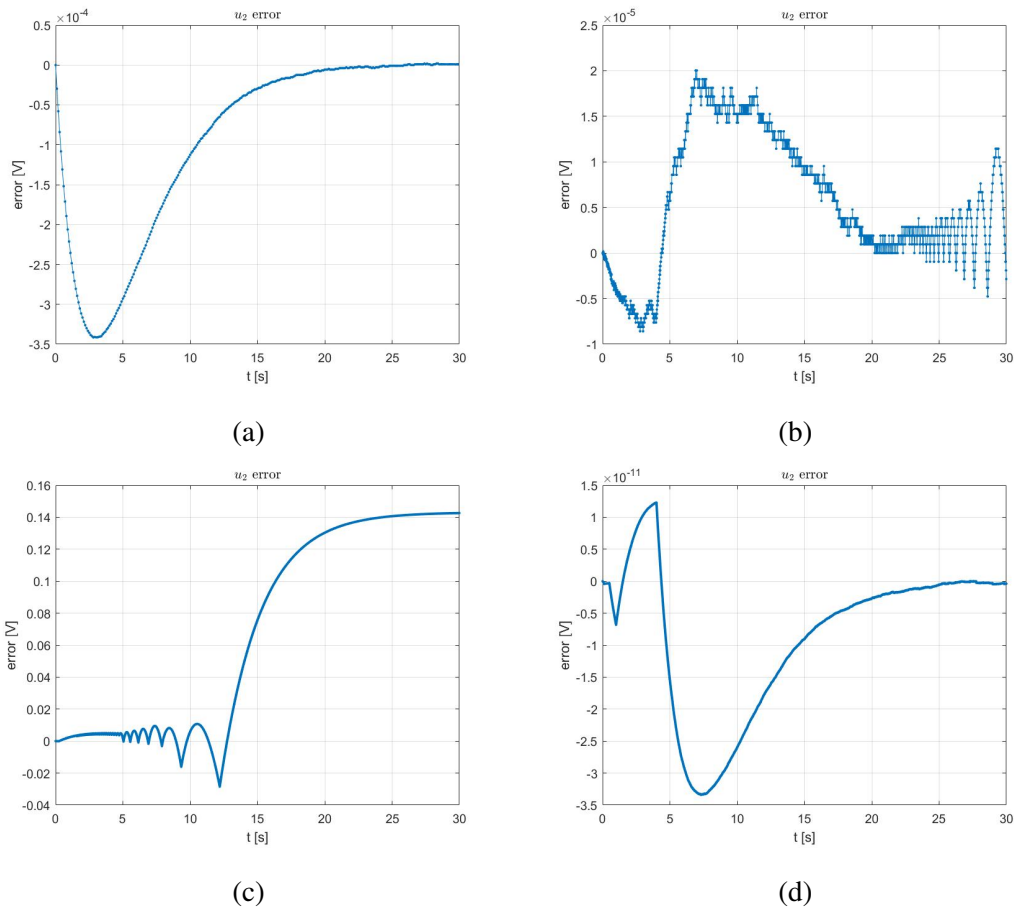


Abbildung 2: (a) $h = 1 \times 10^{-1}$ s, einfache Genauigkeit; (b) $h = 1 \times 10^{-2}$ s, einfache Genauigkeit; (c) $h = 1 \times 10^{-5}$ s, einfache Genauigkeit; (d) $h = 1 \times 10^{-5}$ s, doppelte Genauigkeit;

h [s]	Datentype	$\det(\mathbf{J})$	$\text{cond}(\mathbf{J})$	max. $\ \text{error}\ $ [V]
1×10^{-1}	single	762.5	327.9	3.4×10^{-4}
1×10^{-2}	single	75.13	339.1	2.0×10^{-5}
1×10^{-5}	single	0.075	3.3×10^5	0.142 !!!
1×10^{-5}	double	0.075	3.3×10^5	3.3×10^{-11}

Tabelle 1: Numerische Genauigkeit, $\text{cond}(\mathbf{J}) = (2\text{-Norm})$ Konditionszahl der Jacobi-Matrix

Aus den Simulationsergebnisse kann man sehen, dass bei sehr kleinen Schrittweite ($h = 1 \times 10^{-5}$ s) verschlechtert sich der numerische Genauigkeit dramatisch. Die Rundungsfehler der Gleitpunktzahl werden sich von der schlecht konditionierten Jacobi-Matrix erheblich vergrößert.

Nach mehreren Versuche haben wir gefunden, dass bei $h \approx 1.3 \times 10^{-2}$ s besitzt die Simulation (mit einfacher Gleitkommazahl) eine höchst numerische Genauigkeit von $\max. \|\text{error}\| \approx 1.05 \times 10^{-5}$ V.