

Angewandte Mathematik

Projekt Zeitaufgelöste Photolumineszenz

Vorbemerkungen

Die Projektarbeit soll im Laufe des Semesters von jedem Studierenden selbständig bearbeitet werden. Schritt für Schritt werden die notwendig Grundlagen gelegt, die dann direkt am Projekt bearbeitet werden können. Die notwendigen Skripte und Funktionen zur Lösung des Problems sind Teil der Abgabe und müssen selbständig erstellt werden. Eine Übernahme aus anderen Quellen ist nicht zulässig.

Die einzelnen Schritte müssen begründet und Ergebnisse dokumentiert werden, wobei die Anforderungen einer schriftlichen Arbeit gelten. Insbesondere ist auf eine verständliche, strukturierte Darstellung zu achten: die Arbeit ist so zu schreiben, dass Studierende des Kurses “Angewandte Mathematik” diese verstehen und selbst nachvollziehen können.

Die Dokumentation kann in Form eines PDFs erfolgen oder als Matlab Live Script. Zusammen mit den programmierten Matlab-Routinen erfolgt die Abgabe in einem ZIP-Archiv über Moodle mit den dort angegebenen Fristen. Plagiate werden als Täuschungsversuch gewertet. Die Ausarbeitung geht in die Modulnote ein und wird neben allgemeinen fachlichen Fragen in der mündlichen Modulprüfung besprochen.

1 Einleitung

Perowskit-Solarzellen sind der neue Stern am Himmel der Photovoltaik. Innerhalb von wenigen Jahren sind diese neuartigen Materialien zu konkurrenzfähigen Dünnschicht-Solarzellen mit weit über 20% Wirkungsgrad erwachsen. Daraus begründen sich die enorme Popularität und auch die vielen Hoffnungen an diese Halbleiter-Materialklasse. Dennoch gibt es noch zahlreiche Probleme und ungeklärte Fragestellungen, die die Forschung adressieren kann. Andreas Bartelt und seine Gruppe untersuchen an der HTW Berlin diese Materialien mithilfe von zeitaufgelösten Photolumineszenz-Spektren (TRPL, *time resolved photoluminescence*): das Material wird mithilfe eines Lasers angeregt und die aus der Anregung entstehende Aussendung von Licht in Abhängigkeit von der Zeit gemessen.

Um aus den experimentellen Daten Rückschlüsse auf Lebensdauern und Hinweise auf effizienzlimitierende Prozesse in den neuen Materialien zu ziehen, ist eine Simulation der Dynamiken im Halbleiter notwendig. In dieser Hausarbeit sollen Sie die Grundlagen einer solchen Simulation entwickeln.

Der Halbleiter der Dicke d zwischen zwei Materialien wird gleichmäßig in der Fläche bestrahlt und erzeugt so eine Ladungsträgerdichte u im Halbleiter (siehe Abb. 1). Die Abhängigkeit von u in der Ebene senkrecht zur Dicke kann in guter Näherung vernachlässigt werden, so dass die Ladungsträgerdichte eine Funktion der Zeit t und der Tiefe z ist. Die Elektronendynamik im Halbleiter nach der Bestrahlung wird durch eine Diffusionsgleichung modelliert [2]. Die Ladungsträgerdichte $u(t, z)$ im Halbleiter genügt der parabolischen Differenzialgleichung

$$\frac{\partial u}{\partial t} = D \frac{\partial^2 u}{\partial z^2} - (k_1 + k_2 N_D)u - k_2 u^2 + s(t, z), \quad t \geq t_0, 0 < z < d \quad (1)$$

mit der Anfangsbedingung

$$u(t_0, z) = 0, \quad 0 \leq z \leq d \quad (2)$$

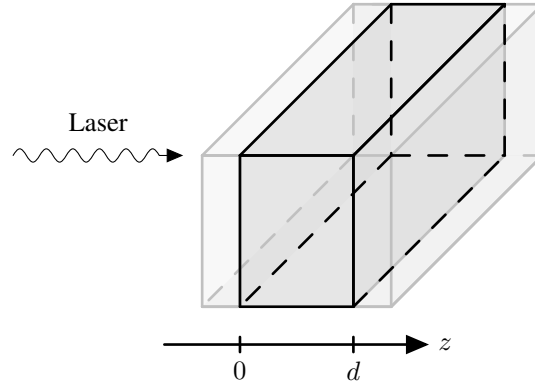


Abbildung 1: Skizze des Aufbaus der Probe. Der Halbleiter aus Peroskit liegt zwischen zwei Materialien und wird in der Ebene senkrecht zu z als unendlich ausgedehnt angenommen. Die Bestrahlung mit Laserlicht erfolgt gleichmäßig aus der Richtung $z < 0$.

und den Randbedingungen

$$D \frac{\partial u}{\partial z}(t, 0) = S_L u(t, 0), \quad D \frac{\partial u}{\partial z}(t, d) = -S_R u(t, d), \quad t \geq t_0. \quad (3)$$

Hier ist $s(t, z)$ die Ladungsträgerdichte, die durch die Bestrahlung pro Zeiteinheit erzeugt wird, D die Diffusionskonstante, N_D die Dotierungsdichte, k_1 bzw. k_2 die Rekombinationskonstanten (Shockley Read Hall Rekombination, bzw. direkte Rekombination) und α die Absorptionskonstante. Die Konstanten S_L ($z = 0$) bzw. S_R ($z = d$) bestimmen die Rekombinationsraten an den jeweiligen Grenzschichten.

In dieser Arbeit sollen folgende Parameter verwendet werden

Parameter	d	D	k_1	k_2	N_D	S_L	S_R	α
Einheit	$[\mu\text{m}]$	$[\frac{\text{cm}^2}{\text{s}}]$	$[\frac{1}{\text{s}}]$	$[\frac{\text{cm}^3}{\text{s}}]$	$[\frac{1}{\text{cm}^3}]$	$[\frac{\text{cm}}{\text{s}}]$	$[\frac{\text{cm}}{\text{s}}]$	$[\frac{1}{\text{cm}}]$
Wert	0.3	0.003	10^6	10^{-8}	10^{15}	10	10^5	10^5

Achtung: Verwenden Sie in der Simulation durchgehend μm und μs als Einheit!

Die Simulation von Gleichung (1) wird schrittweise entwickelt. In Abschnitt 2 wird zunächst die Zeitabhängigkeit vernachlässigt und eine Diskretisierung im Raum der stationären Gleichung mittels finiter Differenzen entwickelt. Der folgende Abschnitt 3 untersucht eine Klasse von numerischen Verfahren zur Lösung von Anfangswertproblemen, die besonders gut für den vorliegenden Fall geeignet sind. Im letzten Abschnitt 4 werden beide Ansätze mithilfe der Linienmethode kombiniert und eine numerische Lösung der Ausgangsgleichung berechnet.

2 Finite Differenzen der stationären Gleichung

In diesem Teil der Arbeit soll die stationäre Verteilung der Ladungsträger, d.h. für den Fall $\partial u / \partial t = 0$, bei kontinuierlicher Bestrahlung modelliert werden. In diesem Fall ist die Gleichung durch

$$D \frac{\partial^2 u}{\partial z^2} - (k_1 + k_2 N_D) u - k_2 u^2 = -s(z), \quad 0 < z < d \quad (4)$$

mit Randbedingungen

$$D \frac{\partial u}{\partial z}(0) = S_L u(0), \quad D \frac{\partial u}{\partial z}(d) = -S_R u(d) \quad (5)$$

gegeben. Hier ist $s(z)$ die Ladungsträgerdichte, die pro Zeiteinheit durch die externe Quelle erzeugt wird.

Die Gleichung soll näherungsweise mithilfe der Finite Differenzen Methode gelöst werden [1] (benötigte Abschnitte werden auf Moodle bereitgestellt). Dazu wird das Gebiet $z \in [0, d]$ in M gleich große Intervalle der Länge h aufgeteilt und die Knotenpunkte mit $0 = z_0 < z_1 < \dots < z_N = d$ bezeichnet. Die genäherte Ladungsdichte an den Stellen z_i wird mit u_i bezeichnet, d.h. es soll

$$u_i \approx u(z_i), \quad i = 0, 1, \dots, N$$

gelten. Für dies Werte wird ein Gleichungssystem hergeleitet und anschließend numerisch gelöst. Zwischenzeitlich werden Hilfsknoten an den Stellen $z_{-1} = -h$ und $z_{N+1} = d + h$ mit Werten u_{-1} und u_{N+1} verwendet, die jedoch später durch die Randbedingungen eliminiert werden.

2.1 Lineare stationäre Gleichung

Im ersten Schritt soll nur der in u lineare Anteil der Gleichung (4) ohne den quadratischen Term $-k_2 u^2$ untersucht werden:

$$D \frac{\partial^2 u}{\partial z^2} - ku = -s(z), \quad 0 < z < d, \quad (6)$$

mit $k = k_1 + k_2 N_D$. Die Randbedingungen sind weiterhin durch Gleichung (5) gegeben.

Die Behandlung der stationären Gleichung erfolgt ähnlich zu der in [1, Abschnitt 8.8] beschriebenen Methode. Jedoch werden die Randbedingungen unterschiedlich behandelt. Gehen Sie wie folgt vor.

1. Erarbeiten Sie sich Abschnitt 8.8 aus [1] und beschreiben Sie Ihr Vorgehen für die Anwendung der Methode auf Gleichung (6).
2. Leiten Sie analog zu Gleichung (8.128) die Gleichungen für die gesuchten Werte u_0, u_1, \dots, u_N an den Punkten z_0, \dots, z_N her. Die Gleichungen enthalten u_{-1} und u_{N+1} , die im nächsten Schritt eliminiert werden. Verwenden Sie dabei die Abkürzung $s_i = s(z_i)$.
3. Approximieren Sie die ersten Ableitungen an den Randbedingungen (5) durch

$$u'(0) \approx \frac{u_1 - u_{-1}}{2h}, \quad u'(d) \approx \frac{u_{N+1} - u_{N-1}}{2h}$$

und lösen Sie die Gleichungen nach u_{-1} bzw. u_{N+1} auf. Setzen Sie diese Ausdrücke in die Gleichungen für die Knoten z_0 und z_N der letzten Teilaufgabe ein.

4. Stellen Sie das lineare Gleichungssystem für die Größen u_0, \dots, u_N analog zu Gleichung (8.133) in [1] und folgende dar:

$$A\mathbf{u} = \mathbf{b}, \quad \mathbf{u} = [u_i], \quad \mathbf{b} = [b_i] \quad (7)$$

Ordnen Sie die Gleichungen analog zu den Knotenpunkten z_0, z_1, \dots, z_N .

5. Implementieren Sie eine Routine zur Berechnung der Matrix A der letzten Teilaufgabe. Verwenden Sie dabei dünnbesetzte Matrizen (*sparse matrix*, siehe z.B. `spdiags`).

$$A = \text{fd_lin_matrix}(N)$$

Eingabe:

N Anzahl von Teilintervallen N .

Ausgabe:

A Finite Differenzen Matrix für \mathbf{u} der Größe $(N + 1) \times (N + 1)$ als *sparse matrix*.

6. Implementieren Sie eine Routine zur Lösung des entsprechenden Gleichungssystems (7) unter Verwendung `fd_lin_matrix`. Lösen Sie das lineare Gleichungssystem mithilfe des Operators `\`.

$$[z, u] = \text{stationaer_lin}(s, N)$$

Eingabe:

`s` Funktionshandle auf Funktion $s(z)$
`function sz=s(z)` mit Spaltenvektoren z und sz
`N` Anzahl von Teilintervallen N .

Ausgabe:

`z` Knotenpunkte (z_0, z_1, \dots, z_N) der Größe $(N+1) \times 1$
`u` Vektor \mathbf{u} der Größe $(N+1) \times 1$

7. Testen Sie Ihre Routine, indem Sie sich eine Funktion $u(z)$ und Konstanten d, D, k vorgeben und dann eine geeignete rechte Seite s und Konstanten S_L, S_R berechnen und die Anzahl der Teilintervalle n variieren.

8. Berechnen Sie mit Hilfe der obigen Routine die Lösung für die Fälle

$$s(z) = S_0 \cdot e^{-\alpha z}, \quad S_0 = 10^2, 10^3, 10^4 \frac{1}{\mu\text{m}^3 \mu\text{s}}.$$

Bestimmen Sie näherungsweise experimentell ein geeignetes N , so dass der relative Fehler in u_i maximal 1‰ beträgt, und stellen Sie die Lösung graphisch dar.

2.2 Nichtlineare stationäre Gleichung

Im zweiten Schritt soll nun die volle nichtlineare Gleichung (4) mit Randbedingung (5) gelöst werden. Analog zum letzten Abschnitt wird ein nichtlineares Gleichungssystem aufgestellt, das mithilfe des Newton-Verfahrens aus der Belegarbeit gelöst werden soll. Gehen Sie wie folgt vor.

1. Leiten Sie die nichtlinearen Gleichungen für die gesuchten Werte u_0, u_1, \dots, u_N her und stellen Sie sie in Form

$$\mathbf{F}(\mathbf{u}) = \mathbf{b}. \quad (8)$$

dar. Berücksichtigen Sie die Randbedingungen wie im letzten Abschnitt.

2. Implementieren Sie eine Routine zur Berechnung von \mathbf{F} der letzten Teilaufgabe.

$$\mathbf{F} = \text{fd_nonlin}(\mathbf{u}, N)$$

Eingabe:

`u` Vektor \mathbf{u} der Größe $(N+1) \times 1$
`N` Anzahl von Teilintervallen N .

Ausgabe:

`F` Vektor $\mathbf{F}(\mathbf{u})$ der Größe $(N+1) \times 1$

3. Berechnen Sie die Jacobi-Matrix $D\mathbf{F}$ von \mathbf{F} bezüglich \mathbf{u} und geben Sie sie an.
4. Implementieren Sie eine Routine zur Berechnung der Jacobi-Matrix $D\mathbf{F}$ von \mathbf{F} der letzten Teilaufgabe. Verwenden Sie dabei dünnbesetzte Matrizen.

$$\mathbf{J} = \text{fd_nonlin_jac}(\mathbf{u}, N)$$

Eingabe:

`u` Vektor \mathbf{u} der Größe $(N+1) \times 1$
`N` Anzahl von Teilintervallen N .

Ausgabe:

`J` Jacobi-Matrix $D\mathbf{F}(\mathbf{u})$ der Größe $(N+1) \times (N+1)$ als *sparse matrix*.

5. Implementieren Sie eine Routine zur Lösung des Gleichungssystems (8) unter Verwendung `fd_nonlin` und `fd_nonlin_jac` und des Newton-Verfahrens wie in der Belegaufgabe. Wählen Sie einen geeigneten Startwert.

`[z,u]=stationaer_nonlin(s,N,tol,nmax)`

Eingabe:

`s` Funktionshandle auf Funktion $s(z)$
`function sz=s(z)` mit Spaltenvektoren z und sz
`N` Anzahl von Teilintervallen N .
`tol` Toleranz für Newton-Verfahren
`nmax` maximale Anzahl an Schritten des Newton-Verfahrens

Ausgabe:

`z` Knotenpunkte (z_0, z_1, \dots, z_N) der Größe $(N+1) \times 1$
`u` Vektor u der Größe $(N+1) \times 1$

6. Testen Sie Ihre Routine, indem Sie sich eine Funktion $u(z)$ und Konstanten d, D, k vorgeben und dann eine geeignete rechte Seite s und Konstanten S_L, S_R berechnen und die Anzahl der Teilintervalle n variieren.
7. Berechnen Sie mit Hilfe der obigen Routine die Lösung für die Fälle

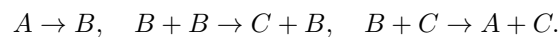
$$s(z) = S_0 \cdot e^{-\alpha z}, \quad S_0 = 10^2, 10^3, 10^4 \frac{1}{\mu\text{m}^3 \mu\text{s}}.$$

Bestimmen Sie näherungsweise experimentell ein geeignetes N , so dass der relative Fehler in u_i maximal 1% beträgt, und stellen Sie die Lösung graphisch dar. Vergleichen Sie die Lösung mit den entsprechenden Lösungen der linearen Gleichung.

3 Implizite Einschrittverfahren

In diesem Abschnitt sollen Löser für Systeme von sogenannten steifen Anfangswertproblemen untersucht werden. In Abschnitt 4 wird die partielle Differenzialgleichung (1) mithilfe der finiten Differenzen des letzten Abschnitts näherungsweise als ein System von Anfangswertproblemen formuliert, das sich steif verhält.

Steife Systeme treten aber auch in der Modellierung von Reaktionen in der Chemie auf. Als Beispiel soll eine chemische Reaktion dreier Stoffe A, B, C aus [3] dienen:



Die Dynamik der Konzentrationen der einzelnen Komponenten können durch Anfangswertproblem

$$\begin{aligned} A: \quad y_1' &= -0.04 y_1 + 10^4 y_2 y_3 \\ B: \quad y_2' &= 0.04 y_1 - 10^4 y_2 y_3 - 3 \cdot 10^7 y_2^2 \\ C: \quad y_3' &= 3 \cdot 10^7 y_2^2 \end{aligned} \tag{9}$$

mit den Anfangswerten

$$y_1(0) = 1, y_2(0) = 0, y_3(0) = 0 \tag{10}$$

modelliert werden. Anhand dieses Systems sollen im folgenden steife Differenzialgleichungen untersucht und ein effizientes numerisches Verfahren entwickelt werden.

3.1 Entwicklung

Ein allgemeines System von Anfangswertproblemen für eine Funktion $\mathbf{y} : \mathbb{R} \rightarrow \mathbb{R}^k$ kann in der Form

$$\mathbf{y}' = \mathbf{f}(t, \mathbf{y}), \quad t \in [a, b], \quad \mathbf{y}(a) = \mathbf{y}_a \quad (11)$$

geschrieben werden. Wie in der Vorlesung wird zur näherungsweisen numerischen Lösung das Intervall $[a, b]$ in n gleich große Teilintervalle der Länge h aufgeteilt mit Knoten $a = t_0 < t_1 < \dots < t_n = b$. Der Wert von \mathbf{y} wird näherungsweise an diesen Zeitpunkten berechnet:

$$\mathbf{y}^{(i)} \approx \mathbf{y}(t_i), \quad i = 0, 1, \dots, N.$$

Für die Berechnung der Werte $\mathbf{y}^{(i)}$ sollen implizite Einschrittverfahren wie in verwendet werden [1, Abschnitt 8.4]. Insbesondere das implizite Euler-Verfahren [1, Gleichung (8.51)]

$$\mathbf{y}^{(i+1)} = \mathbf{y}^{(i)} + h \mathbf{f}(t_{i+1}, \mathbf{y}^{(i+1)}) \quad (12)$$

und die implizite Trapezregel [1, Gleichung (8.57)]

$$\mathbf{y}^{(i+1)} = \mathbf{y}^{(i)} + \frac{h}{2} [\mathbf{f}(t_i, \mathbf{y}^{(i)}) + \mathbf{f}(t_{i+1}, \mathbf{y}^{(i+1)})] \quad (13)$$

Im Unterschied zu [1] soll die Lösung der auftretenden nichtlinearen Gleichungssystemen nicht mithilfe der Fixpunktiteration sondern mit dem Newton-Verfahren aus der Belegarbeit näherungsweise bestimmt werden.

Gehen Sie für die Entwicklung der beiden Methoden wie folgt vor.

1. Erarbeiten Sie sich Abschnitt 8.4 aus [1] und fassen Sie in eigenen Worten zusammen, was Sie unter steifen Differenzialgleichung verstehen.
2. Setzen Sie in den Gleichungen des impliziten Euler-Verfahrens und der impliziten Trapezregel (Gln. (12) und (13))

$$\mathbf{y}^{(i+1)} = \mathbf{y}^{(i)} + \mathbf{z}$$

und formulieren Sie jeweils die Gleichung für \mathbf{z} als Nullstellenproblem:

$$\mathbf{F}_{\text{euler}}(\mathbf{z}) = \mathbf{0}, \quad \mathbf{F}_{\text{trapez}}(\mathbf{z}) = \mathbf{0} \quad (14)$$

3. Bestimmen Sie die Jacobi-Matrizen $D\mathbf{F}_{\text{euler}}$ und $D\mathbf{F}_{\text{trapez}}$ von $\mathbf{F}_{\text{euler}}$ und $\mathbf{F}_{\text{trapez}}$ in Abhängigkeit von $D_{\mathbf{y}}\mathbf{f}$, d.h. der Jacobi-Matrix von \mathbf{f} bezüglich \mathbf{y} .
4. Implementieren Sie die folgenden zwei Routinen. Verwenden Sie für die Jacobi-Matrix \mathcal{J} dünnbesetzte Matrizen.

$$\begin{aligned} [F, J] &= F_{\text{euler}}(z, ti, h, yi, f, df) \text{ und} \\ [F, J] &= F_{\text{trapez}}(z, ti, h, yi, f, df) \end{aligned}$$

Eingabe:

z	Vektor \mathbf{z} der Größe $k \times 1$
ti	Zeitpunkt t_i
h	Schrittweite h
yi	Vektor $\mathbf{y}^{(i)}$ der Größe $k \times 1$
f	Funktionshandle für $\mathbf{f}(t, \mathbf{y})$
df	Funktionshandle für $D_{\mathbf{y}}\mathbf{f}(t, \mathbf{y})$

Ausgabe:

F	Vektor $\mathbf{F}_{\text{euler}}(\mathbf{z})$ bzw. $\mathbf{F}_{\text{trapez}}(\mathbf{z})$ der Größe $k \times 1$
J	Jacobi-Matrix $D\mathbf{F}_{\text{euler}}(\mathbf{z})$ bzw. $D\mathbf{F}_{\text{trapez}}(\mathbf{z})$ der Größe $k \times k$ als <i>sparse matrix</i> .

5. Implementieren Sie die Löser aufbauend auf den Routinen des letzten Abschnitts und des Newton-Verfahrens aus der Belegarbeit. Als Startwert der Newtoniteration wählen Sie für \mathbf{z} den Nullvektor.

$[t, y] = \text{impl_euler}(f, \text{tspan}, y_a, n, df, \text{tol}, n_{\max})$ und
 $[t, y] = \text{impl_trapez}(f, \text{tspan}, y_a, n, df, \text{tol}, n_{\max})$

Eingabe:

f Funktionshandle für $\mathbf{f}(t, \mathbf{y})$
 tspan Intervall $[a, b]$
 y_a Anfangswert \mathbf{y}_a
 n Anzahl von Teilintervallen.
 df Funktionshandle für $D_{\mathbf{y}}\mathbf{f}(t, \mathbf{y})$ bezüglich \mathbf{y}
 tol Toleranz für Newton-Verfahren
 n_{\max} maximale Anzahl an Schritten des Newton-Verfahrens

Ausgabe:

t Vektor der Stützpunkte (t_0, t_1, \dots, t_n) der Größe $(n+1) \times 1$
 y Matrix der approximierten Lösungswerte $\mathbf{y}^{(i)}$ mit $i = 0, \dots, n$ der Größe $(n+1) \times k$

6. Bestimmen Sie wie in den Übungsblättern anhand des Anfangswertproblems $y' = -y$ mit Anfangsbedingung $y(0) = 1$ auf dem Intervall $[0, 1]$ graphisch die Ordnung der beiden Verfahren.
7. Testen Sie Ihre Routine anhand eines geeigneten nichtlinearen, zeitabhängigen Systems von Anfangswertproblemen.

3.2 Anwendung

In diesem Abschnitt sollen die entwickelten Löser auf das Modell der chemischen Reaktion angewandt werden. Gehen Sie dabei wie folgt vor.

- Formulieren Sie Gleichung (9) als System der Form (11) mit Funktion $\mathbf{f}_{\text{chem}}(t, \mathbf{y})$
- Implementieren Sie die Funktion als Routine

$\text{dy} = \text{f_chem}(t, \mathbf{y})$

Eingabe:

t Zeit t
 \mathbf{y} Vektor \mathbf{y} der Größe 3×1

Ausgabe:

dy Vektor $\mathbf{f}_{\text{chem}}(t, \mathbf{y})$ der Größe 3×1

- Berechnen Sie die Jacobi-Matrix $D_{\mathbf{y}}\mathbf{f}_{\text{chem}}(t, \mathbf{y})$ von \mathbf{f}_{chem} bezüglich \mathbf{y} .
- Implementieren Sie die Funktion als Routine

$J = \text{f_chem_jac}(t, \mathbf{y})$

Eingabe:

t Zeit t
 \mathbf{y} Vektor \mathbf{y} der Größe 3×1

Ausgabe:

J Matrix $D_{\mathbf{y}}\mathbf{f}_{\text{chem}}(t, \mathbf{y})$ der Größe 3×3

5. Lösen Sie das Anfangswertproblem auf dem Intervall $[0, 1]$ mit den folgenden Integratoren: explizites Euler-Verfahren, Mittelpunktregel (siehe Übungen), implizites Euler-Verfahren und implizite Trapezregel. Bestimmen Sie experimentell jeweils näherungsweise ein geeignetes n , so dass so dass der absolute Fehler jeder einzelnen Komponente und Zeitschritts höchstens 10^{-4} , bzw. 10^{-6} ist. Erstellen Sie eine Tabelle der notwendigen Intervalleinteilungen.
6. Diskutieren Sie Ihre Ergebnisse im Hinblick auf die Performanz.

4 Zeitaufgelöste Simulation

In diesem Abschnitt werden die entwickelten Methoden kombiniert, um die ursprüngliche Gleichung (1) zu lösen. Dazu wird diese Gleichung an den Knotenpunkten z_0, \dots, z_N aus Abschnitt 2 ausgewertet und man erhält

$$\frac{\partial u}{\partial t}(t, z_i) = D \frac{\partial^2 u}{\partial z^2}(t, z_i) - (k_1 + k_2 N_D)u(t, z_i) - k_2 u^2(t, z_i) + s(t, z_i), \quad t \geq t_0, i = 0, 1, \dots, N$$

Wie bei den finiten Differenzen setzen wir

$$u_i(t) = u(t, z_i)$$

und approximieren die rechte Seite durch finite Differenzen wie in Abschnitt 2. Wir erhalten somit ein System von Anfangswertproblemen für die Funktionen $u_i(t)$, $i = 0, \dots, N$. Gehen Sie wie folgt vor.

1. Formulieren Sie das System in der Form

$$\mathbf{u}' = \mathbf{F}(t, \mathbf{u}).$$

2. Stellen Sie Routinen zur Berechnung von $\mathbf{F}(t, \mathbf{u})$ und $D_{\mathbf{u}}\mathbf{F}(t, \mathbf{u})$ bereit.
3. Verwenden Sie die implizite Trapezregel und simulieren Sie das System für $t \in [-0.05, 0.2]\mu\text{s}$ mit

$$s(t, z) = S_0 \cdot e^{-\frac{t^2}{2 \cdot 0.01^2}} \cdot e^{-\alpha z}, \quad S_0 = 10^4, 10^5, 10^6 \frac{1}{\mu\text{m}^3 \mu\text{s}}.$$

Wählen Sie geeignete Zahl an Orts- und Zeitschritten.

4. Stellen Sie die Ergebnisse geeignet dar (siehe `mesh`).

Literatur

- [1] Atkinson, Kendall E. und Han, Weimin: “Elementary Numerical Analysis”, J. Wiley & Sons, Hoboken, NJ (2004).
- [2] Baloch, Ahmer A. B. *et al*: “Analysis of Photocarrier Dynamics at Interfaces in Perovskite Solar Cells by Time-Resolved Photoluminescence”, The Journal of Physical Chemistry C, Seiten 26805–26815 (2018).
- [3] Hairer, Ernst und Wanner, Gerhard: “Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems”, Springer, Berlin (1991).