

---

# 多結晶SI太陽電池中の鉄不純物の電圧印加オペランド観察での時系列シミュレーション

---

はじまりを、つくる



静岡理科大学

吉田 惇輝, 小林 正, 大場 春佳, 水野 信也, 吉田 豊



NTTデータ数理システム学生研究奨励賞  
2020/12/18



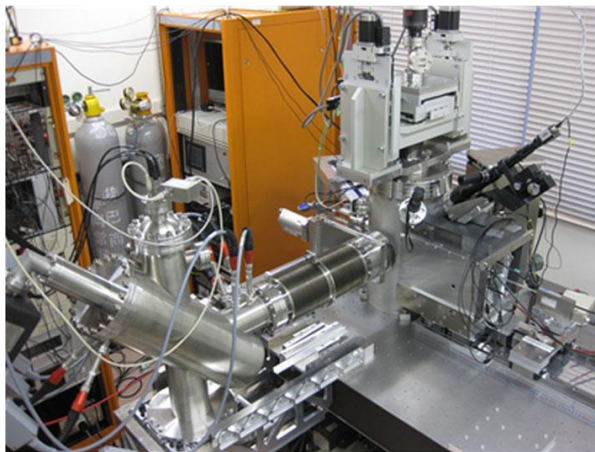
- 1,はじめに
- 2,本研究の目的
- 3,本研究に用いた反応式・データ
- 4,Kの推定
  - 4-1,差分方程式と勾配法を用いてKをモデル化
  - 4-2,Numerical Optimizerを使用
- 5,Kの推定の結果
- 6,まとめと今後の課題

# 1,はじめに

「多結晶Si太陽電池中の鉄不純物の電圧印加オペランド観察」とは、

- 静岡理科大学 理工学部 物質生命科学科  
吉田研究室で行った研究

顕微メスバウア分光装置を利用して、多結晶シリコン太陽電池に外部電圧を印加したときに生じる、電子・正孔・鉄不純物を特殊な装置を用いて、異なる化学状態ごとにオペランド観察をしたもの



顕微メスバウア分光装置

引用文献

Y. Yoshida・ Y. Ino・K. Matsumuro・T. Watanabe・H. Fujita・K. Hayakawa・K. Yukihiro・K. Ogai・K. Moriguchi・ Y. Harada・H. Soejima (2016),  
“Feasibility study to investigate diffusion of Fe in Si using a Mossbauer spectroscopic microscope,”  
Hyperfine Interact(2016) 237: 130

# 1,はじめに

## 多結晶シリコン太陽電池の特徴



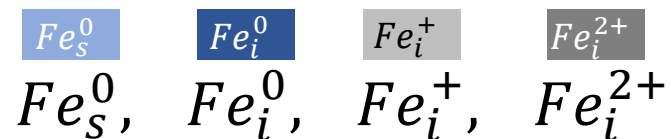
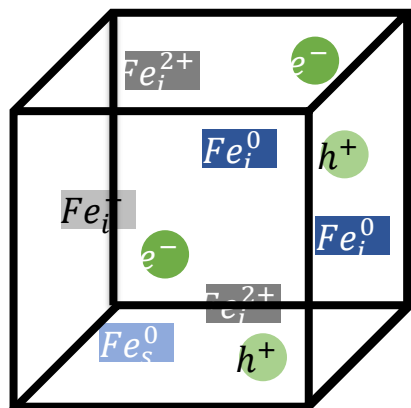
- ICの端材やシリコン半導体の端材を利用して製造しているため、大理石のような青く輝く見た目
- 製造単価が安価に抑えられる点から、住宅用太陽光発電システムに多く使われる
- 製造過程で不可避免的に鉄不純物が混入



発電効率の低下↘

## 2,本研究の目的

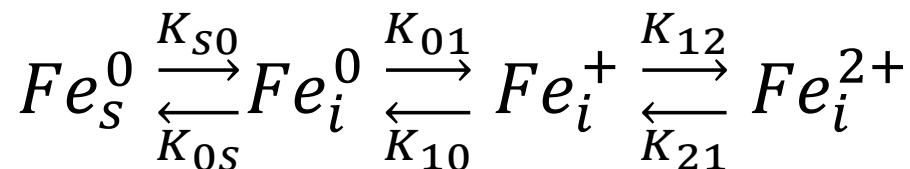
多結晶シリコン太陽電池には4種類の状態の鉄不純物



が存在する。

本研究では、

電圧を印加した際に起こる、鉄(Fe)の状態変化とキャリア捕獲反応のしやすさ  
「 $K_{s0}, K_{0s}, K_{01}, K_{10}, K_{12}, K_{21}$ 」のモデル化と推定を行う。



# 3,反応式



鉄の状態変化とキャリア捕獲反応のしやすさ「 $K_{s0}, K_{0s}, K_{01}, K_{10}, K_{12}, K_{21}$ 」  
は電子濃度、正孔濃度と反応のしやすさの積である。

$$K_{s0} = k_{s0}$$

$$K_{0s} = k_{0s}$$

$$K_{01} = k_{01}n_h$$

$$K_{10} = k_{10}n_e$$

$$K_{12} = k_{12}n_h$$

$$K_{21} = k_{21}n_e$$

- $k_{s0} \rightarrow Fe_s^0$  から  $Fe_i^0$  の反応のしやすさ
- $k_{0s} \rightarrow Fe_i^0$  から  $Fe_s^0$  の反応のしやすさ
- $k_{01} \rightarrow Fe_i^0$  から  $Fe_i^+$  の反応のしやすさ
- $k_{10} \rightarrow Fe_i^+$  から  $Fe_i^0$  の反応のしやすさ
- $k_{12} \rightarrow Fe_i^+$  から  $Fe_i^{2+}$  の反応のしやすさ
- $k_{21} \rightarrow Fe_i^{2+}$  から  $Fe_i^{2+}$  の反応のしやすさ
- $n_e \rightarrow$  電子( $e^-$ )濃度
- $n_h \rightarrow$  正孔( $h^+$ )濃度

# 3, 反応式

鉄濃度( $C_{Fe_s^0}, C_{Fe_i^0}, C_{Fe_i^+}, C_{Fe_i^{2+}}$ )の反応式は以下のように表す。

$$\frac{dC_{Fe_i^0}}{dt} = -k_{01}C_{Fe_i^0}n_h + k_{10}C_{Fe_i^+}n_e - (k_{s0}C_{Fe_s^0} + k_{0s}C_{Fe_i^0})$$

$$\frac{dC_{Fe_i^+}}{dt} = k_{01}C_{Fe_i^0}n_h - (k_{10}n_e + k_{12}n_h)C_{Fe_i^+} + k_{21}C_{Fe_i^{2+}}n_e$$

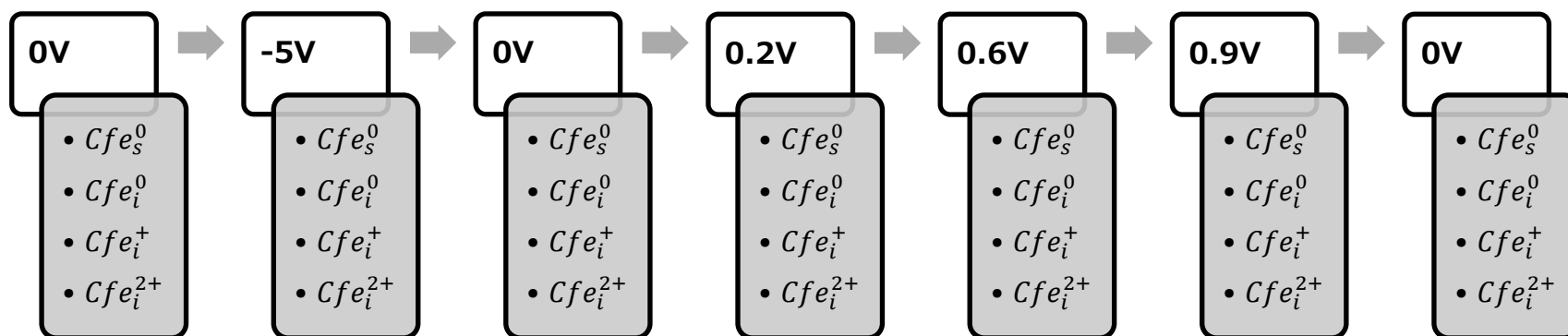
$$\frac{dC_{Fe_i^{2+}}}{dt} = k_{12}C_{Fe_i^+}n_h - k_{21}C_{Fe_i^{2+}}n_e$$

$$\frac{dC_{Fe_s^0}}{dt} = k_{s0}C_{Fe_s^0} + k_{0s}C_{Fe_i^0}$$

- $C_{Fe_i^0} \rightarrow$  鉄( $Fe_i^0$ )濃度
- $C_{Fe_i^1} \rightarrow$  鉄( $Fe_i^1$ )濃度
- $C_{Fe_i^2} \rightarrow$  鉄( $Fe_i^2$ )濃度
- $C_{Fe_s^0} \rightarrow$  鉄( $Fe_s^0$ )濃度
- $n_e \rightarrow$  電子( $e^-$ )濃度
- $n_h \rightarrow$  正孔( $h^+$ )濃度

# 3,研究データ

- 鉄濃度( $C_{Fe_s^0}, C_{Fe_i^0}, C_{Fe_i^+}, C_{Fe_i^{2+}}$ )は多結晶Si太陽電池中の鉄不純物の電圧印加オペランド観察で得られたデータを用いる。
- 7種類の電圧印加時の鉄濃度( $C_{Fe_s^0}, C_{Fe_i^0}, C_{Fe_i^+}, C_{Fe_i^{2+}}$ )を一つの鉄濃度あたり、6400セルごとに表したデータである。



$(8.0 \times 10^{-2})cm$

$(8.0 \times 10^{-2})cm$

$C_{fe}$

	1	2	3	4	5...	76	77	78	79	80
1	###	###	###	###	###	###	###	###	###	###
2	###	###	###	###	###	###	###	###	###	###
3	###	###	###	###	###	###	###	###	###	###
4	###	###	###	###	###	###	###	###	###	###
5	###	###	###	###	###	###	###	###	###	###
...	...	...	...	...	...	...	...	...	...	...
76	###	###	###	###	###	###	###	###	###	###
77	###	###	###	###	###	###	###	###	###	###
78	###	###	###	###	###	###	###	###	###	###
79	###	###	###	###	###	###	###	###	###	###
80	###	###	###	###	###	###	###	###	###	###

80\*80セル





# 4, Kの推定

## 4-1, 差分方程式と勾配法を用いてKをモデル化

- 6つのパターンでシミュレーションの初期値、目的値を設定し  
3つのプロセスで各セルごとのKの最適化を行う。

	シミュレーション値 の初期値	目的値
パターン1	0V	-5V
パターン2	-5V	0V
パターン3	0V	0.2V
パターン4	0.2V	0.6V
パターン5	0.6V	0.9V
パターン6	0.9V	0V

### I. 差分方程式による $C_{Fe}^n$ のシミュレーション

$$C_{Fe}^n(t+h) = C_{Fe}^n(t) + f_n(C_{Fe}(t), K)h$$

### II. 目的関数

(目的値とシミュレーション値の誤差を評価)

$$P(K) = \frac{1}{2} \sum_{n=s}^2 (C_{Fe}^n - S_{Fe}^n)^2$$

### III. 勾配法より $K = (K_{s0}, K_{0s}, K_{01}, K_{10}, K_{12}, K_{21})$ を更新

$$K(m+1) = K(m) - \alpha(\nabla P(K))$$

### IV. I ~ IIIを繰り返し、最適値Kを得る

# 4, $K$ の推定



## 4-1, 差分方程式と勾配法を用いて $K$ をモデル化

### I. 差分方程式による $C_{Fe}^n$ のシミュレーション

$$C_{Fe}^n(t+h) = C_{Fe}^n(t) + f_n(C_{Fe}(t), K)h \quad n = s, 0, 1, 2$$

詳細は

$$C_{Fe}^s(t+h) = C_{Fe}^s(t) + f_s(C_{Fe}^s(t), C_{Fe}^0(t), C_{Fe}^1(t), C_{Fe}^2(t), K)h$$

$$C_{Fe}^0(t+h) = C_{Fe}^0(t) + f_0(C_{Fe}^s(t), C_{Fe}^0(t), C_{Fe}^1(t), C_{Fe}^2(t), K)h$$

$$C_{Fe}^1(t+h) = C_{Fe}^1(t) + f_1(C_{Fe}^s(t), C_{Fe}^0(t), C_{Fe}^1(t), C_{Fe}^2(t), K)h$$

$$C_{Fe}^2(t+h) = C_{Fe}^2(t) + f_2(C_{Fe}^s(t), C_{Fe}^0(t), C_{Fe}^1(t), C_{Fe}^2(t), K)h$$

ただし、

$$f_s(C_{Fe}(t), K) = -k_{s0}C_{Fe}^s(t) + k_{0s}C_{Fe}^0(t)$$

$$f_0(C_{Fe}(t), K) = k_{01}C_{Fe}^0(t)n_h + k_{10}C_{Fe}^1(t)n_e + k_{s0}C_{Fe}^s(t) - k_{0s}C_{Fe}^0(t)$$

$$f_1(C_{Fe}(t), K) = k_{01}C_{Fe}^0(t)n_h - (k_{10}n_e + k_{12}n_h)C_{Fe}^1(t) + k_{21}C_{Fe}^2(t)n_e$$

$$f_2(C_{Fe}(t), K) = k_{12}C_{Fe}^1(t)n_h - k_{21}C_{Fe}^2(t)n_e$$

$$K = (K_{s0}, K_{0s}, K_{01}, K_{10}, K_{12}, K_{21})$$

# 4, Kの推定



## 4-1, 差分方程式と勾配法を用いてKをモデル化

- 関数fの偏微分

$$\frac{\partial f_n(C_{Fe}, K)}{\partial K_{s0}} = \begin{cases} -C_{Fe}^s & (n = s) \\ C_{Fe}^s & (n = 0) \\ 0 & (n = 1) \\ 0 & (n = 2) \end{cases} \quad \frac{\partial f_n(C_{Fe}, K)}{\partial K_{0s}} = \begin{cases} C_{Fe}^0 & (n = s) \\ -C_{Fe}^0 & (n = 0) \\ 0 & (n = 1) \\ 0 & (n = 2) \end{cases}$$

$$\frac{\partial f_n(C_{Fe}, K)}{\partial K_{01}} = \begin{cases} 0 & (n = s) \\ -C_{Fe}^0 n_h & (n = 0) \\ C_{Fe}^0 n_h & (n = 1) \\ 0 & (n = 2) \end{cases} \quad \frac{\partial f_n(C_{Fe}, K)}{\partial K_{10}} = \begin{cases} 0 & (n = s) \\ C_{Fe}^1 n_e & (n = 0) \\ -C_{Fe}^1 n_e & (n = 1) \\ 0 & (n = 2) \end{cases}$$

$$\frac{\partial f_n(C_{Fe}, K)}{\partial K_{12}} = \begin{cases} 0 & (n = s) \\ 0 & (n = 0) \\ -C_{Fe}^1 n_h & (n = 1) \\ C_{Fe}^1 n_h & (n = 2) \end{cases} \quad \frac{\partial f_n(C_{Fe}, K)}{\partial K_{21}} = \begin{cases} 0 & (n = s) \\ 0 & (n = 0) \\ C_{Fe}^2 n_e & (n = 1) \\ -C_{Fe}^2 n_e & (n = 2) \end{cases}$$



# 4, $K$ の推定

## 4-1, 差分方程式と勾配法を用いて $K$ をモデル化

II. 目的関数(目的値とシミュレーション値の誤差を評価)

$$P(K) = \frac{1}{2} \sum_{n=s}^2 (C_{Fe}^n - S_{Fe}^n)^2$$

- 目的関数の微分

→ 目的関数の微分

$$\begin{aligned} \frac{\partial P}{\partial K_x} &= \sum_{n=s}^2 (C_{Fe}^n - S_{Fe}^n(K)) (C_{Fe}^n - S_{Fe}^n(K))' \\ &= \sum_{n=s}^2 (C_{Fe}^n - S_{Fe}^n(K)) (f_n(C_{Fe}^n(t), K)h)' \\ &= \sum_{n=s}^2 (C_{Fe}^n - S_{Fe}^n(K)) \frac{\partial f_n(C_{Fe}^n(t), K)}{\partial K_x} h \\ C_{Fe}^n(t, K) - S_{Fe}^n(t, K) &\cong f_n(C_{Fe}^n(t, K))h \end{aligned}$$

# 4, $K$ の推定



## 4-1, 差分方程式と勾配法を用いて $K$ をモデル化

III. 勾配法より  $K = (K_{s0}, K_{0s}, K_{01}, K_{10}, K_{12}, K_{21})$  を更新

$$K(m+1) = K(m) - \alpha(\nabla P(K))$$

$$\begin{pmatrix} K_{s0}(m+1) \\ K_{0s}(m+1) \\ K_{01}(m+1) \\ K_{10}(m+1) \\ K_{12}(m+1) \\ K_{21}(m+1) \end{pmatrix} = \begin{pmatrix} K_{s0}(m) \\ K_{0s}(m) \\ K_{01}(m) \\ K_{10}(m) \\ K_{12}(m) \\ K_{21}(m) \end{pmatrix} - \alpha \begin{pmatrix} \frac{\partial P(K)}{\partial K_{s0}} \\ \frac{\partial P(K)}{\partial K_{0s}} \\ \frac{\partial P(K)}{\partial K_{01}} \\ \frac{\partial P(K)}{\partial K_{10}} \\ \frac{\partial P(K)}{\partial K_{12}} \\ \frac{\partial P(K)}{\partial K_{21}} \end{pmatrix}$$

$m$ はシミュレーション回数

# 4, $K$ の推定

## 4-1, 差分方程式と勾配法を用いて $K$ をモデル化

### IV. I ~ IIIを繰り返し、推定値 $K$ を得る

I. 差分方程式による  $C_{Fe}^n$  のシミュレーション

$$C_{Fe}^n(t+h) = C_{Fe}^n(t) + f_n(C_{Fe}(t), K)h$$

II. 目的関数

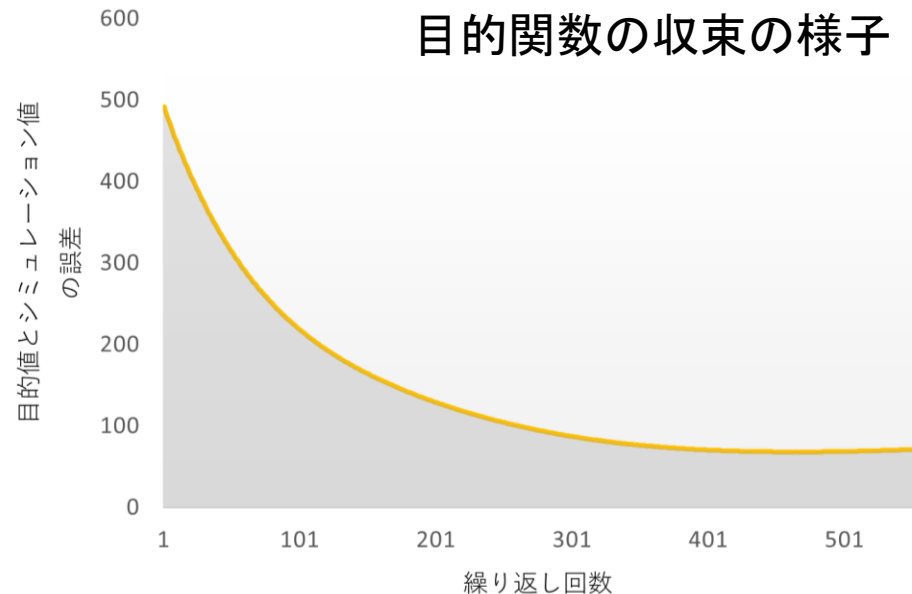
(目的値とシミュレーション値の誤差を評価)

$$P(K) = \frac{1}{2} \sum_{n=s}^2 (C_{Fe}^n - S_{Fe}^n)^2$$

III. 勾配法より  $K = (K_{s0}, K_{0s}, K_{01}, K_{10})$

$$K(m+1) = K(m) - \alpha(\nabla P(K))$$

目的関数の収束の様子



# 4, $K$ の推定



## 4-2, Numerical Optimizerを使用

- さらなる最適化を図るため、Numerical Optimizerを使用。

→ NTTデータ数理システムの数理計画法パッケージ

```
1 //データのセット ↓
2 Set S; ↓
3 Element i(set=S); ↓
4 //目的値fe ↓
5 Parameter df0(name="df0", index=i); ↓
6 Parameter df1(name="df1", index=i); ↓
7 Parameter df2(name="df2", index=i); ↓
8 Parameter dfs(name="dfs", index=i); ↓
9 Parameter ne; .....ne=1.0; ↓
10 Parameter nh; .....nh=1.0; ↓
11 //変数 ↓
12 Variable cfe0(name="cfe0", index=i); .....cfe0[i]>=0; ↓
13 Variable cfe1(name="cfe1", index=i); .....cfe1[i]>=0; ↓
14 Variable cfe2(name="cfe2", index=i); .....cfe2[i]>=0; ↓
15 Variable cfes(name="cfesub", index=i); .....cfes[i]>=0; ↓
16 Variable k01(index=i); .....k01[i]>=0; ↓
17 Variable k10(index=i); .....k10[i]>=0; ↓
18 Variable k12(index=i); .....k12[i]>=0; ↓
19 Variable k21(index=i); .....k21[i]>=0; ↓
20 Variable ks0(index=i); .....ks0[i]>=0; ↓
21 Variable k0s(index=i); .....k0s[i]>=0; ↓
22 //微分式 ↓
23 Expression getcfe0(index=i); ↓
24 getcfe0[i]=k01[i]*cfe0[i]*nh+k10[i]*cfe1[i]*ne-(-ks0[i]*cfes[i]+k0s[i]*cfe0[i]); ↓
25 getcfe0[i]>=0; ↓
26 Expression getcfe1(index=i); ↓
27 getcfe1[i]=k01[i]*cfe0[i]*nh-(k10[i]*ne+k12[i]*nh)*cfe1[i]+k21[i]*cfe2[i]*ne; ↓
28 getcfe1[i]>=0; ↓
29 Expression getcfe2(index=i); ↓
30 getcfe2[i]=-k12[i]*cfe1[i]*nh-k21[i]*cfe2[i]*ne; ↓
31 getcfe2[i]>=0; ↓
32 Expression getsub(index=i); ↓
33 getsub[i]=-cfes[i]*ks0[i]+cfe0[i]*k0s[i]; ↓
34 getsub[i]>=0; ↓
35 //目的関数 ↓
36 Objective P(index=i, type=minimize); ↓
37 P[i]=-(df0[i]-getcfe0[i])*(df0[i]-getcfe0[i])+(df1[i]-getcfe1[i])*(df1[i]-getcfe1[i])+(df2[i]-getcfe2[i])*(df2[i]-getcfe2[i])+(dfs[i]-getsub[i])*(dfs[i]-getsub[i])/2; ↓
38 ↓
39 //結果の表示 ↓
```

} 目的値[定数]

} 初期値[変数]

} 求解[変数]

} 微分式

```
17 DiffusionMain_1212.smp:36:info: 展開中 制約式 ... (13/14) name="" ↓
18 DiffusionMain_1212.smp:39:info: 展開中 制約式 ... (14/14) name="" ↓
19 ↓
20 [About Numerical Optimizer] ↓
21 MSI Numerical Optimizer 22.1.0 (license expire in 73 days) ↓
22 >>>> ↓
23 >>>><with META-HEURISTICS engine="wccsp"/"rcpsp"> ↓
24 >>>><with Netlib-BLAS> ↓
25 , Copyright (C) 1991 NTT DATA Mathematical Systems Inc. ↓
26 ↓
27 [Problem and Algorithm] ↓
28 PROBLEM_NAME ..... DiffusionMain_1212 ↓
29 NUMBER_OF_VARIABLES ..... 10 ↓
30 NUMBER_OF_FUNCTIONS ..... 4 ↓
31 PROBLEM_TYPE ..... MINIMIZATION ↓
32 METHOD ..... BFGS_LINE_SEARCH ↓
33 QUASI-NEWTON_TYPE ..... BFGS ↓
34 ↓
35 [Progress] ↓
36 <preprocess begin> ..... <preprocess end> ↓
37 <iteration begin> ↓
38 .....res=8.8e+00 .....1.8e-01 .....1.6e-04 .....1.6e-04 .....2.1e-03 ..... ↓
39 .....1.9e-03 .....1.8e-03 .....1.8e-03 .....1.7e-03 .....1.6e-03 ..... ↓
40 .....1.5e-03 .....1.5e-03 .....1.4e-03 .....1.4e-03 .....1.4e-03 ..... ↓
41 .....1.3e-03 .....1.3e-03 .....1.3e-03 .....1.2e-03 .....1.2e-03 ..... ↓
42 .....1.2e-03 .....1.1e-03 .....1.1e-03 .....1.1e-03 .....1.0e-03 ..... ↓
```

# 4, $K$ の推定



## 4-2, Numerical Optimizerを使用

- 目的値

$$\begin{cases} f_0 = Fe_i^0 \text{濃度[目的]} \\ f_1 = Fe_i^1 \text{濃度[目的]} \\ f_2 = Fe_i^2 \text{濃度[目的]} \\ f_{sub} = Fe_s^0 \text{濃度[目的]} \end{cases}$$

- 求解

$$\begin{cases} k_{01}, & k_{01} \geq 0 \\ k_{10}, & k_{10} \geq 0 \\ k_{12}, & k_{12} \geq 0 \\ k_{21}, & k_{21} \geq 0 \\ k_{s0}, & k_{s0} \geq 0 \\ k_{0s}, & k_{0s} \geq 0 \end{cases}$$

- 初期値

$$\begin{cases} Cf_0 = Fe_i^0 \text{濃度[初期]}, & Cfe_0 \geq 0 \\ Cf_1 = Fe_i^1 \text{濃度[初期]}, & Cfe_1 \geq 0 \\ Cf_2 = Fe_i^2 \text{濃度[初期]}, & Cfe_2 \geq 0 \\ Cf_{sub} = Fe_s^0 \text{濃度[初期]}, & Cfe_{sub} \geq 0 \end{cases}$$



# 4, Kの推定



## 4-2, Numerical Optimizerを使用

- 目的関数

→ 目的値とシミュレーション値の誤差を最小化する。

$$P = \frac{1}{2} \left( (f_0 - (C_{Fe_i^0})')^2 + (f_1 - (C_{Fe_i^+})')^2 + (f_2 - (C_{Fe_i^{2+}})')^2 + (f_s - (C_{Fe_s^0})')^2 \right)$$

$$P \geq 0$$

$(C_{fe})' \rightarrow$  シミュレーション値

- 微分式

→ シミュレーション値として反応式を微分式に用いる。

$$\frac{dC_{Fe_i^0}}{dt} = -k_{12}C_{Fe_i^0}n_h + k_{21}C_{Fe_i^+}n_e - (k_{s0}C_{Fe_s^0} + k_{0s}C_{Fe_i^0})$$

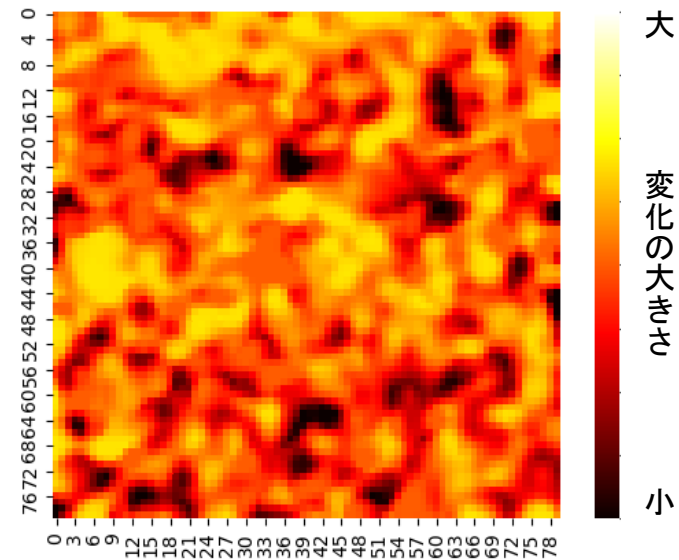
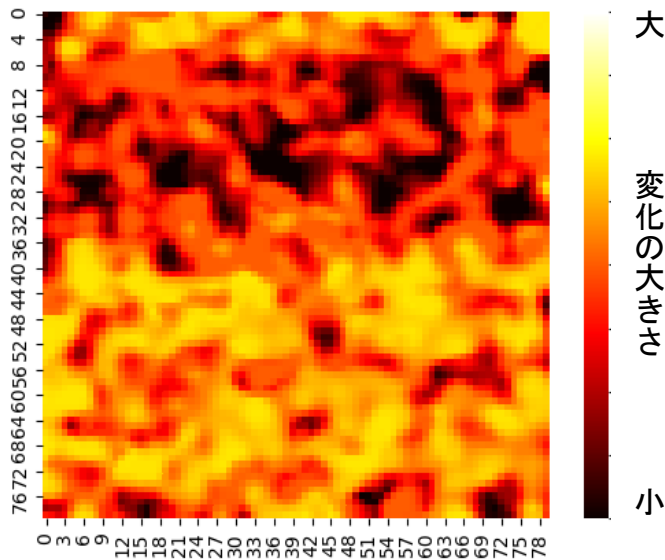
$$\frac{dC_{Fe_i^+}}{dt} = k_{12}C_{Fe_i^0}n_h - (k_{21}n_e + k_{23}n_h)C_{Fe_i^+} + k_{32}C_{Fe_i^{2+}}n_e$$

$$\frac{dC_{Fe_i^{2+}}}{dt} = k_{23}C_{Fe_i^+}n_h - k_{32}C_{Fe_i^{2+}}n_e$$

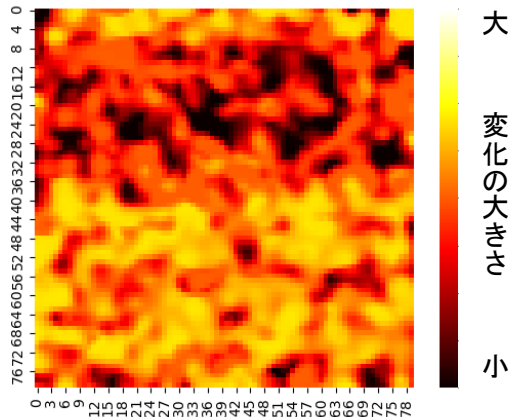
$$\frac{dC_{Fe_s^0}}{dt} = k_{s0}C_{Fe_s^0} + k_{0s}C_{Fe_i^0}$$

# 5, $K$ の推定の結果

- 最適化した鉄(Fe)の状態変化とキャリア捕獲反応のしやすさ「 $K_{s0}, K_{0s}, K_{01}, K_{10}, K_{12}, K_{21}$ 」をプロット化。
- 以下の画像は  
シミュレーション値の初期値[-5V]→目的値[0V]の $K_{10}, K_{12}$ を  
変化の大きさごとに色分けを行った。

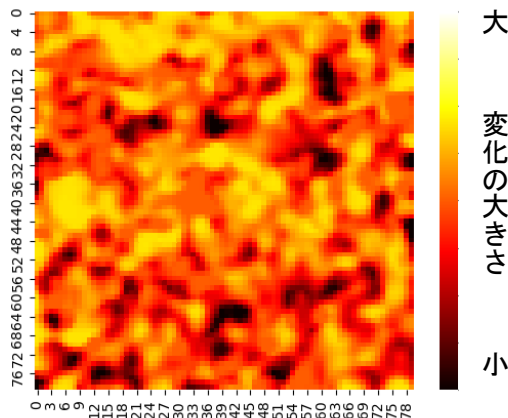


# 5, $K$ の推定の結果



シミュレーション値の初期値[-5V]→目的値[0V]の $K_{10}$

上半分が変化しにくい、  
下半分が変化しやすいと考えられる。



シミュレーション値の初期値[-5V]→目的値[0V]の $K_{12}$

変化しやすい部分と変化しにくい部分が  
広く分布していると考えられる。

# 5,まとめと今後の課題



## 今回

- 変化の現象のモデル化と、ソフトウェアを用いて電子濃度・正孔濃度での状態変化とキャリア捕獲のしやすさ「 $K_{s0}, K_{0s}, K_{01}, K_{10}, K_{12}, K_{21}$ 」を得られた。

## 今後は

- 電子濃度「 $n_e$ 」、正孔濃度「 $n_h$ 」のそれぞれの値を求めていく必要がある。
- 専門家と話し合い、より詳細なモデル化と推定をする必要がある。

また結果本研究は最適化する個数が多いため、並列化ができる環境が必要である。