Uses of Methods with Result Verification for Simplified Control-Oriented SOFC Models

Ekaterina Auer and Stefan Kiel

University of Technology, Business and Design Wismar

June 17, 2016

Introduction

SOFC: Devices converting chemical energy into electricity

An important characteristic: Temperature!

Control-oriented models: ODEs with unknown parameters to be identified from real-life data

Identification: Least squares/Global optimization

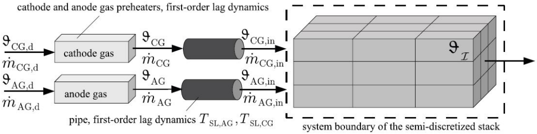
$$\Phi(p) = \sum_{k=1}^{T\approx 17500} \sum_{j=1}^{n_m} \Big(\underbrace{y_j(t_k, p)}_{\text{solution to ODEs}} - \underbrace{y_{j,m}(t_k)}_{\text{measured data}} \Big)^2 \rightarrow \text{min}$$

Traditional techniques: $y(t,p)$ reflects only stationary operating states

$\rightsquigarrow$ Goal: Models for a range of operating conditions (better control!)

A cooperation with the University of Rostock

A Temperature Model for SOFCs



A. Rauh et al., *Reliable control of high-temperature fuel cell systems...*, IMA J Math Control Info (2014)

SOFC stack: A distributed parameter system naturally described by a non-linear partial differential equation

Control-oriented models: ODEs obtained by spatial semi-discretization

Difficulty: The temperature can be measured only at few positions $\rightsquigarrow$ State/disturbance estimators are necessary

Basic control inputs: Gas preheaters (modeled or from measurements)

Types of models: Different kinds in dependence on

- the used arithmetics (floating point/interval/other)
- "accuracy" of the solution $y(t, p)$ (analytic/approximated/exact)

(Still) Modeling Temperature $\theta(t)$ for SOFCs

Important for θ: Heat capacities of gases modeled as

$$c_{gas}(\theta) = c_{gas,0} + c_{gas,1} \cdot \theta + c_{gas,2} \cdot \theta^2$$

Example of a model with (just) one finite volume element:

```
dtheta(1,1) = T_AG_inv*(v_N2_d-v_N2-d_N2_sch);dtheta(2,1) = T_SL_AG_inv*(v_N2-v_N2_in);dtheta(3,1) = 0;
dtheta(4,1) = T_AG_inv*(v_H2_d-v_H2-d_H2_sch);dtheta(5,1) = T_SL_AG_inv*(v_H2-v_H2_in);dtheta(6,1) = 0;
dtheta(7,1) = T_AG_inv*(v_H2O_d-v_H2O-d_H2O_sch);dtheta(8,1) = T_SL_AG_inv*(v_H2O-v_H2O_in);dtheta(9,1) = 0;
dtheta(10,1) = T_CG_inv*(v_CG_d-v_CG-d_CG_sch);dtheta(11,1) = T_SL_CG_inv*(v_CG-v_CG_in);dtheta(12,1) = 0;
dtheta(13,1)=-c_m_inv*(-234000*alpha_i*z*F*theta_A-448500*alpha_j*z*F*theta_A-I_ges^2*R_el*z*F
    -z*F*v_CG_in*c_CG_0+234000*alpha_i*z*F*theta_1_1_1-z*F*v_N2_in*c_N2_0-345000*alpha_k*z*F*theta_A
    +448500*alpha_j*z*F*theta_1_1_1+345000*alpha_k*z*F*theta_1_1_1-z*F*v_H2O_in*c_H2O_1*theta_1_1_1
    -z*F*v_H2O_in*c_H2O_2*theta_1_1_1^2+z*F*m_dot_AG_H2O_in*theta_1_1_1*c_H2O_0+z*F*m_dot_AG_H2O_in*
    theta_1_1_1^2*c_H2O_1+z*F*m_dot_AG_H2O_in*theta_1_1_1^3*c_H2O_2-I_ges*R_delta_H_H2_0-z*F*v_H2_in*
    c_H2_1*theta_1_1_1-z*F*v_H2_in*c_H2_2*theta_1_1_1^2+z*F*m_dot_AG_H2_in*theta_1_1_1*c_H2_0+
    z*F*m_dot_AG_H2_in*theta_1_1_1^2*c_H2_1+z*F*m_dot_AG_H2_in*theta_1_1_1^3*c_H2_2+z*F*m_dot_CG_in*
    theta_1_1_1*c_CG_0-z*F*v_CG_in*c_CG_2*theta_1_1_1^2-z*F*v_CG_in*c_CG_1*theta_1_1_1+z*F*m_dot_CG_in*
    theta_1_1_1^3*c_CG_2-I_ges*R_delta_H_H2_1*theta_1_1_1-I_ges*R_delta_H_H2_2*theta_1_1_1^2
    +z*F*m_dot_CG_in*theta_1_1_1^2*c_CG_1-z*F*v_N2_in*c_N2_1*theta_1_1_1+z*F*m_dot_AG_N2_in
    *theta_1_1_1^3*c_N2_2-z*F*v_N2_in*c_N2_2*theta_1_1_1^2+z*F*m_dot_AG_N2_in*theta_1_1_1*c_N2_0+
    z*F*m_dot_AG_N2_in*theta_1_1_1^2*c_N2_1-z*F*v_H2_in*c_H2_0-z*F*v_H2O_in*c_H2O_0)/z/F;
```

(Still) Modeling Temperature $\theta(t)$ for SOFCs

Important for θ: Heat capacities of gases modeled as

$$c_{gas}(\theta) = c_{gas,0} + c_{gas,1} \cdot \theta + c_{gas,2} \cdot \theta^2$$

Example of a model with (just) one finite volume element:

```
dtheta(1,1) = T_AG_inv*(v_N2_d-v_N2-d_N2_sch);dtheta(2,1) = T_SL_AG_inv*(v_N2-v_N2_in);dtheta(3,1) = 0;
dtheta(4,1) = T_AG_inv*(v_H2_d-v_H2-d_H2_sch);dtheta(5,1) = T_SL_AG_inv*(v_H2-v_H2_in);dtheta(6,1) = 0;
dtheta(7,1) = T_AG_inv*(v_H2O_d-v_H2O-d_H2O_sch);dtheta(8,1) = T_SL_AG_inv*(v_H2O-v_H2O_in);dtheta(9,1) = 0;
dtheta(10,1) = T_CG_inv*(v_CG_d-v_CG-d_CG_sch);dtheta(11,1) = T_SL_CG_inv*(v_CG-v_CG_in);dtheta(12,1) = 0;
dtheta(13,1)=-c_m_inv*(-234000*alpha_i*z*F*theta_A-448500*alpha_j*z*F*theta_A-I_ges^2*R_el*z*F
    -z*F*v_CG_in*c_CG_0+234000*alpha_i*z*F*theta_1_1_1-z*F*v_N2_in*c_N2_0-345000*alpha_k*z*F*theta_A
    +448500*alpha_j*z*F*theta_1_1_1+345000*alpha_k*z*F*theta_1_1_1-z*F*v_H2O_in*c_H2O_1*theta_1_1_1
    -z*F*v_H2O_in*c_H2O_2*theta_1_1_1^2+z*F*m_dot_AG_H2O_in*theta_1_1_1*c_H2O_0+z*F*m_dot_AG_H2O_in*
    theta_1_1_1^2*c_H2O_1+z*F*m_dot_AG_H2O_in*theta_1_1_1^3*c_H2O_2-I_ges*R_delta_H_H2_0-z*F*v_H2_in*
    c_H2_1*theta_1_1_1-z*F*v_H2_in*c_H2_2*theta_1_1_1^2+z*F*m_dot_AG_H2_in*theta_1_1_1*c_H2_0+
    z*F*m_dot_AG_H2_in*theta_1_1_1^2*c_H2_1+z*F*m_dot_AG_H2_in*theta_1_1_1^3*c_H2_2+z*F*m_dot_CG_in*
    theta_1_1_1*c_CG_0-z*F*v_CG_in*c_CG_2*theta_1_1_1^2-z*F*v_CG_in*c_CG_1*theta_1_1_1+z*F*m_dot_CG_in*
    theta_1_1_1^3*c_CG_2-I_ges*R_delta_H_H2_1*theta_1_1_1-I_ges*R_delta_H_H2_2*theta_1_1_1^2
    +z*F*m_dot_CG_in*theta_1_1_1^2*c_CG_1-z*F*v_N2_in*c_N2_1*theta_1_1_1+z*F*m_dot_AG_N2_in
    *theta_1_1_1^3*c_N2_2-z*F*v_N2_in*c_N2_2*theta_1_1_1^2+z*F*m_dot_AG_N2_in*theta_1_1_1*c_N2_0+
    z*F*m_dot_AG_N2_in*theta_1_1_1^2*c_N2_1-z*F*v_H2_in*c_H2_0-z*F*v_H2O_in*c_H2O_0)/z/F;
```

Idea: Simplify the models, but not too much (wide range!)

(Still) Modeling Temperature $\theta(t)$ for SOFCs

Important for θ: Heat capacities of gases modeled as

$$c_{gas}(\theta) = c_{gas,0} + c_{gas,1} \cdot \theta + c_{gas,2} \cdot \theta^2$$

Example of a model with (just) one finite volume element:

```
dtheta(1,1) = T_AG_inv*(v_N2_d-v_N2-d_N2_sch);dtheta(2,1) = T_SL_AG_inv*(v_N2-v_N2_in);dtheta(3,1) = 0;
dtheta(4,1) = T_AG_inv*(v_H2_d-v_H2-d_H2_sch);dtheta(5,1) = T_SL_AG_inv*(v_H2-v_H2_in);dtheta(6,1) = 0;
dtheta(7,1) = T_AG_inv*(v_H2O_d-v_H2O-d_H2O_sch);dtheta(8,1) = T_SL_AG_inv*(v_H2O-v_H2O_in);dtheta(9,1) = 0;
dtheta(10,1) = T_CG_inv*(v_CG_d-v_CG-d_CG_sch);dtheta(11,1) = T_SL_CG_inv*(v_CG-v_CG_in);dtheta(12,1) = 0;
dtheta(13,1)=-c_m_inv*(-234000*alpha_i*z*F*theta_A-448500*alpha_j*z*F*theta_A-I_ges^2*R_el*z*F
    -z*F*v_CG_in*c_CG_0+234000*alpha_i*z*F*theta_1_1_1-z*F*v_N2_in*c_N2_0-345000*alpha_k*z*F*theta_A
    +448500*alpha_j*z*F*theta_1_1_1+345000*alpha_k*z*F*theta_1_1_1-z*F*v_H2O_in*c_H2O_1*theta_1_1_1
    -z*F*v_H2O_in*c_H2O_2*theta_1_1_1^2+z*F*m_dot_AG_H2O_in*theta_1_1_1*c_H2O_0+z*F*m_dot_AG_H2O_in*
    theta_1_1_1^2*c_H2O_1+z*F*m_dot_AG_H2O_in*theta_1_1_1^3*c_H2O_2-I_ges*R_delta_H_H2_0-z*F*v_H2_in*
    c_H2_1*theta_1_1_1-z*F*v_H2_in*c_H2_2*theta_1_1_1^2+z*F*m_dot_AG_H2_in*theta_1_1_1*c_H2_0+
    z*F*m_dot_AG_H2_in*theta_1_1_1^2*c_H2_1+z*F*m_dot_AG_H2_in*theta_1_1_1^3*c_H2_2+z*F*m_dot_CG_in*
    theta_1_1_1*c_CG_0-z*F*v_CG_in*c_CG_2*theta_1_1_1^2-z*F*v_CG_in*c_CG_1*theta_1_1_1+z*F*m_dot_CG_in*
    theta_1_1_1^3*c_CG_2-I_ges*R_delta_H_H2_1*theta_1_1_1-I_ges*R_delta_H_H2_2*theta_1_1_1^2
    +z*F*m_dot_CG_in*theta_1_1_1^2*c_CG_1-z*F*v_N2_in*c_N2_1*theta_1_1_1+z*F*m_dot_AG_N2_in
    *theta_1_1_1^3*c_N2_2-z*F*v_N2_in*c_N2_2*theta_1_1_1^2+z*F*m_dot_AG_N2_in*theta_1_1_1*c_N2_0+
    z*F*m_dot_AG_N2_in*theta_1_1_1^2*c_N2_1-z*F*v_H2_in*c_H2_0-z*F*v_H2O_in*c_H2O_0)/z/F;
```

Idea: Simplify the models, but not too much (wide range!)

Approach: $c_{gas}(\theta) = c_{gas,0}$ or $c_{gas}(\theta) = c_{gas,0} + c_{gas,1} \cdot \theta \leftarrow$ Topic 1

How Good are the Simplified Models?

Question 0: How good is the correspondence to reality? What about the validity range?

Question 1: Can we take uncertainty in parameters into account?

Question 2: Can we find out which parameters these models are most sensitive to?

Question 3: Is it possible/useful to solve the equations analytically?

Approaches to answers $\leftarrow$ Topic 2:

1$\rightarrow$ Result verification (intervals, affine forms, Taylor models)

2$\rightarrow$ Sensitivity analysis (compute $\frac{\partial \theta}{\partial p_i}$)

3$\rightarrow$ Yes in some cases

Tool: UniVerMeC allowing for an easy reuse and flexibility

Characteristics of SOFC Temperature Models

$$\Phi(p) = \sum_{k=1}^{T} \sum_{j=1}^{n_m} \Big(\underbrace{\theta_j(t_k, p)}_{\text{solution to ODEs}} - \underbrace{\theta_{j,m}(t_k)}_{\text{measured data}} \Big)^2 \to \min$$

F1 How exactly is $\theta(t_k, p)$ obtained?

- F1.a A closed-form solution
- F1.b Approximation by an expression (e.g., with the Euler method)
- F1.c Numerical solution from a “black-box” solver

F2 What is the underlying technique for the implementation?

- F2.a Floating point
- F2.b Interval
- F2.c Other verified

Examples: High verification degrees

F1.b& F2.b-c “Verified approximation”

- [+] Rounding cared for
- [−] Overestimation
- [+] Easy derivatives (AD)
- [+] Easily portable to the GPU

F1.c&F2.b Verified IVP solvers

- [+] Verifies the whole model
- [−] Derivatives require solving an additional ODE
- [−] High computational effort

Good: UniVerMeC allows us to implement all combinations

Simplified Models

Trade-off between verification degree and computing time necessary

$\Downarrow$

Higher verification degrees in real-time $\rightsquigarrow$ Model simplification

Simplifying assumptions

- → The nitrogen is the only anode gas for the heating phase
- → One volume element for the whole stack
- → Heat capacities of all gases are constant ($c_{gas}(\theta) = c_{0,gas}$) or linear ($c_{gas}(\theta) = c_{0,gas} + c_{1,gas} \cdot \theta$)

Generally, we are interested in all combinations of F1& F2

Actually, we consider F1.a-b with

F2.a	for parameter identification
F2.a-c	for simulation

Parameters to Be Identified

α_i, α_j, α_k	coefficients of heat convection
$c_{N_2,0}$, $c_{N_2,1}$	heat capacity of N_2 as $c_{N_2}(\theta) = c_{N_2,0}(+c_{N_2,1} \cdot \theta)$
$c_{CG,0}$, $c_{CG,1}$	heat capacity of the cathode gas
T_{AG}^{inv}	inverse time constant of the anode gas preheater
T_{CG}^{inv}	inverse time constant of the cathode gas preheater
$T_{SL,AG}^{inv}$	inverse time constant of the anode gas supply line
$T_{SL,CG}^{inv}$	inverse time constant of the cathode gas supply line
c	specific heat capacity of the stack module
m	mass of the stack module
c_m^{inv}	$= \frac{1}{c \cdot m}$ with the mass m and heat capacity c of the stack

Control Variables

$\dot{m}_{N_2}^{in}$	mass flow of anode gas (recorded data)
$\dot{m}_{CG}^{in}$	mass flow of cathode gas (recorded data)
θ_A	ambient temperature
θ_{AG}^{d}	desired temperature of the anode gas (recorded data)
θ_{CG}^{d}	desired temperature of the cathode gas (recorded data)
$u_1 = v_{N_2}^{d}$	desired $v_{N_2} = \theta_{AG}^{d} \cdot \dot{m}_{N_2}^{in}$
$u_2 = v_{CG}^{d}$	desired $v_{CG} = \theta_{CG}^{d} \cdot \dot{m}_{CG}^{in}$

The Simplified Model with $c_{gas}(\theta) = c_{0,gas}$

Model:

$$\dot{y}_0 = T^{\text{inv}}_{\text{AG}} \cdot (u_1 - y_0), \quad \dot{y}_1 = T^{\text{inv}}_{\text{SL,AG}} \cdot (y_0 - y_1)$$

$$\dot{y}_2 = T^{\text{inv}}_{\text{CG}} \cdot (u_2 - y_2), \quad \dot{y}_3 = T^{\text{inv}}_{\text{SL,CG}} \cdot (y_2 - y_3)$$

$$\dot{\theta} = -c_m^{\text{inv}} \cdot (k_{\text{const}} - (c_{\text{CG},0} \cdot y_3 + c_{\text{N2},0} \cdot y_1) + k_{\text{lin}}\theta) \ ,$$

$$k_{\text{const}} = \theta_A \cdot (-234000\alpha_i - 448500\alpha_j - 345000\alpha_k) \ ,$$

$$k_{\text{lin}} = 234000\alpha_i + 448500\alpha_j + 345000\alpha_k + \dot{m}^{\text{in}}_{\text{CG}} \cdot c_{\text{CG},0} + \dot{m}^{\text{in}}_{\text{N2}} \cdot c_{\text{N2},0} \ ,$$

initial conditions $y_i(t_0) = y_i^{\text{ic}}$, $i = 0, 1, 2, 3$, $\theta(t_0) = \theta^{\text{ic}}$

Solution:

$$y_0(t) = u_1 - (u_1 - y_0^{\text{ic}}) e^{-T^{\text{inv}}_{\text{AG}}(t-t_0)}$$

$$y_1(t) = u_1 - \frac{T^{\text{inv}}_{\text{SL,AG}}}{T^{\text{inv}}_{\text{SL,AG}} - T^{\text{inv}}_{\text{AG}}} (u_1 - y_0^{\text{ic}}) e^{-T^{\text{inv}}_{\text{AG}}(t-t_0)} + k_{\text{N2}} e^{-T^{\text{inv}}_{\text{SL,AG}}(t-t_0)},$$

$$\theta(t) = \mathcal{I}_{\text{N2}}(t) + \mathcal{I}_{\text{CG}}(t) + k_\theta e^{-c_m^{\text{inv}} \cdot k_{\text{lin}}(t-t_0)} - \frac{k_{\text{const}}}{k_{\text{lin}}} \ ,$$

with defined expressions for k_{N2}, $\mathcal{I}_{\text{N2/CG}}(t)$, k_θ

The MATLAB-generated solution is too unstable!

The Simplified Model with $c_{gas}(\theta) = c_{0,gas} + c_{1,gas} \cdot \theta$

⚠ ODEs for preheaters $\rightsquigarrow$ a Riccati equation for θ

→ Measured values for preheaters $\rightsquigarrow$ an expression for θ

Model: $\dot{\theta} = -c_m^{inv} \cdot (k_{co} + k_{lin}\theta + k_{sq}\theta^2)$ with

$$\begin{aligned} k_a &= (234\alpha_i + 448.5\alpha_j + 345\alpha_k) \cdot 10^3\,,\ k_{co} = -\theta_A k_a - (y_{3,m} c_{CG,0} + y_{1,m} c_{N_2,0}) \\ k_{lin} &= k_a + \dot{m}_{CG}^{in} \cdot c_{CG,0} + \dot{m}_{N_2}^{in} \cdot c_{N_2,0} - (y_{3,m} c_{CG,1} + y_{1,m} c_{N_2,1}) \\ k_{sq} &= \dot{m}_{CG}^{in} \cdot c_{CG,1} + \dot{m}_{N_2}^{in} \cdot c_{N_2,1} \end{aligned}$$

Solution in dependence on $D = k_{lin}^2 - 4 \cdot k_{co} \cdot k_{sq}$

$$D > 0 : \theta(t) = \frac{\sqrt{D}/k_{sq}}{1 - e^{-c_m^{inv}(t-t_0)\sqrt{D}} \cdot \left(1 - \frac{2\sqrt{D}}{2k_{sq}\theta^{ic} + k_{lin} + \sqrt{D}}\right)} - \frac{k_{lin} + \sqrt{D}}{2k_{sq}}$$

$$D < 0 : \theta(t) = \frac{\sqrt{-D}\tan\left(-\frac{\sqrt{-D}}{2} c_m^{inv}(t-t_0) + \theta^c\right) - k_{lin}}{2k_{sq}}\,,\ \theta^c = \text{atan}\left(\frac{2k_{sq}\theta^{ic} + k_{lin}}{\sqrt{-D}}\right)$$

$$D = 0 : \theta(t) = \frac{2\theta^{ic} + k_{lin}/k_{sq}}{2 + c_m^{inv}(t-t_0)(2k_{sq}\theta^{ic} + k_{lin})} - \frac{k_{lin}}{2k_{sq}}$$

Overview

Two stages: Parameter identification and simulation

Two models: M1 ($c_{gas}(\theta) = c_{0,gas}$) and M2 ($c_{gas}(\theta) = c_{0,gas} + c_{1,gas} \cdot \theta$)

Identification: Goal A parameter set p with the smallest $\Phi(p)$

Possibilities: Analytical solution (F1.a) or approximation (F1.b) with doubles/intervals (F2.a-b)

Note: F1.c is still too expensive computationally ($\leadsto$ parallelization)

Means: MatLab; IpOpt and GlobOpt in UniVerMeC

Simulation: Goals Identify ps the models are most sensitive to; take into account uncertainty

Possibilities: Analytical (F1.a) or numerical (F1.c) solution with doubles/intervals/AA/TM (F2.a-c)

Note: F1.b is not interesting here!

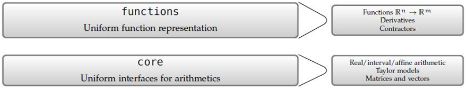
This helps to identify the overall usage areas for M1 and M2!

UniVerMeC: Function Specification

Important: Interoperability

The capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units

Necessary: Formalizations for arithmetics, types of enclosures, etc.



$f : \mathbb{R}^n \mapsto \mathbb{R}^m$, tools for user-defined functions (inductive), analytical expressions or C++ code blocks

Function extensions: Evaluated with all arithmetics supported by `core`

Features: set of functionalities associated with f (e.g., differentiability)

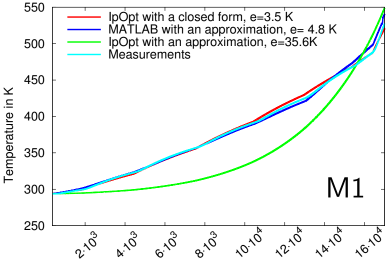
FR object: Tuple $F_{f,n,m} = (\mathcal{I}, \mathcal{F})$ where $\mathcal{I}$ is the set of inclusion functions, $\mathcal{F}$ is a choice out of r supported features

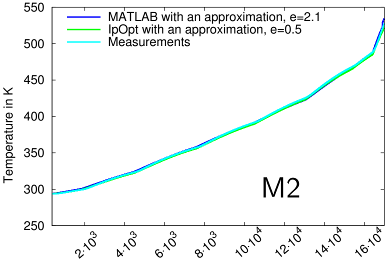
If $\mathcal{I}$ and $\mathcal{F}$ are defined appropriately, e.g., probabilistic arithmetics can be used!

Parameter Identification

The best obtained parameter sets for M1 and M2

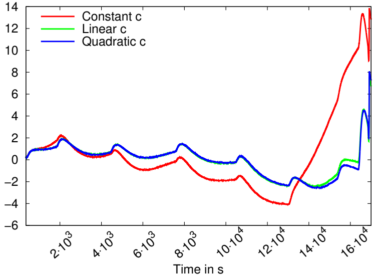
Model	M1 (F1.a&F2.a)	M2 (F1.b&F2.a)
c	$4.53503 \cdot 10^6$	$3.38579 \cdot 10^4$
$\alpha_i = \alpha_j = \alpha_k$	$1.49548 \cdot 10^{-3}$	$3.60235 \cdot 10^{-5}$
$c_{N_2,0}$	$1.96452 \cdot 10^6$	$5.52725 \cdot 10^6$
$c_{N_2,1}$	–	-11221.8
$c_{CG,0}$	$1.52826 \cdot 10^7$	$-8.88817 \cdot 10^4$
$c_{CG,1}$	–	483.868
e	3.5K	0.5K



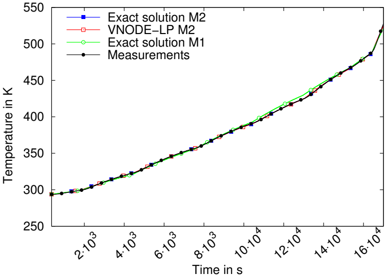


Comparison

Difference btw. measured and simulated temperatures in K



Temperature modeled by M1 and M2 (F2.a)



M2 seems to be even better than the normal model!

Sensitivity

Sensitivities are obtained easily in UniVerMeC
by algorithmic differentiation

Φ	$\lvert\frac{\partial\Phi}{\partial c}\rvert$	$\lvert\frac{\partial\Phi}{\partial\alpha_i}\rvert$	$\lvert\frac{\partial\Phi}{\partial c_{N_2,0}}\rvert$	$\lvert\frac{\partial\Phi}{\partial c_{N_2,1}}\rvert$	$\lvert\frac{\partial\Phi}{\partial c_{CG,0}}\rvert$	$\lvert\frac{\partial\Phi}{\partial c_{CG,1}}\rvert$
F1.a, M2	$2.94 \cdot 10^1$	$3.81 \cdot 10^{10}$	$4.96 \cdot 10^{-2}$	$2.64 \cdot 10^1$	1.47	$1.26 \cdot 10^2$
F1.b, M2	$4.77 \cdot 10^{-3}$	$6.88 \cdot 10^5$	$5.36 \cdot 10^{-5}$	$2.21 \cdot 10^{-2}$	$1.30 \cdot 10^{-3}$	$5.25 \cdot 10^{-1}$
F1.a, M1	$3.09 \cdot 10^{-1}$	$2.45 \cdot 10^8$	$8.49 \cdot 10^{-2}$	–	$1.36 \cdot 10^{-1}$	–
F1.b, M1	$3.09 \cdot 10^{-1}$	$2.45 \cdot 10^8$	$8.49 \cdot 10^{-2}$	–	$1.36 \cdot 10^{-1}$	–

→ Both models are most sensitive to α_i, α_j, α_k

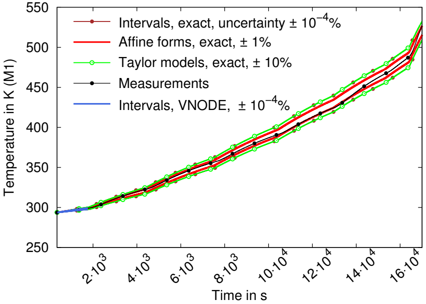
→ For M1, almost no difference in sensitivity between F1.a and F1.b

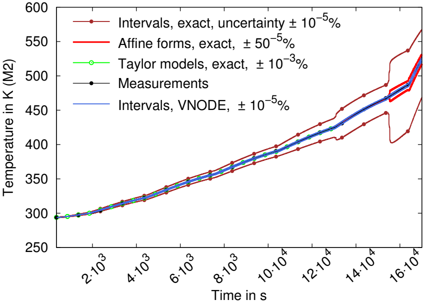
→ For M2, F1.b is up to 10^5 less sensitive

Uncertainty

Fact: We were intereseted in $c_{gas}(\theta) = c_{0,gas}(+c_{1,gas} \cdot \theta)$

Consider uncertainty in $c_{N_2,0}$ and $c_{CG,0}$





CPU times

	Intervals	Affine forms	Taylor models	VNODE-LP
M1	0.34 s $(u \approx 10^{-4})$	112.09 s $(u \approx 1)$	207 s $(u \approx 10)$	>1 day $(u \approx 10^{-4})$
M2	0.12 s $(u \approx 10^{-5})$	23.34 s $(u \approx 50^{-5})$	120.5* s $(u \approx 10^{-3})$	228.42 s $(u \approx 10^{-5})$

Conclusion

What did we do?

- → Simplified models for SOFC temperature wrt. heat capacity of gases with analytical solutions considered
- → Parameter sets with $e_{M1} = 3.5$K and $e_{M2} = 0.5$K identified
- → Models analysed wrt. to their areas of validity (F1.ab / F2.abc)

What did we gain?

- M1: F1.a-b are equivalent; good for short time intervals; fast
- M2: Very good correspondence with measured data; F1.a structurally complex (but fast)
- → M1,M2 are very sensitive to coefficients of heat convection
- → M1,M2 handle different magnitudes of uncertainty in $c_{gas,0}$
- → M2 with F1.a more prone to overestimation than M1

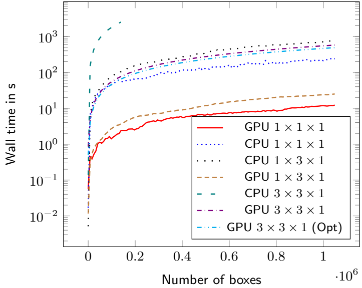
What now?

Outlook

Goal 1: More verification ($\rightsquigarrow$ consider F1.c)

Goal 2: Less overestimation ($\rightsquigarrow$ subdivide uncertainty regions)

Possible solution: Use the GPU to speed up computations!



Reference system:

Xeon E5-2680, 8 cores, gcc 4.7 on Linux, GeForce GTX 580, 512 cores, CUDA 4.2

Results (evaluation of $\Phi([p])$):

$1 \times 1 \times 1$: speedup of 19

$1 \times 3 \times 1$: speedup of 30

$3 \times 3 \times 1$: speedup of 33

Problem: Software!

Besides: Models for degradation of SOFC stacks?