Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2009, Article ID 728105, 34 pages
doi:10.1155/2009/728105

Research Article

Adaptive Step-Size Control in Simulation of Diffusive CVD Processes

Jürgen Geiser and Christian Fleck

*Department of Mathematics, Humboldt-Universität zu Berlin, Unter den Linden 6,
D-10099 Berlin, Germany*

Correspondence should be addressed to Jürgen Geiser, geiser@mathematik.hu-berlin.de

Received 3 September 2008; Revised 5 January 2009; Accepted 28 January 2009

Recommended by José Roberto Castilho Piqueira

We present control strategies of a diffusion process for chemical vapor deposition for metallic bipolar plates. In the models, we discuss the application of different models to simulate the plasma-transport of chemical reactants in the gas-chamber. The contribution are an optimal control problem based on a PID control to obtain a homogeneous layering. We have taken into account one- and two-dimensional problems that are given with constraints and control functions. A finite-element formulation with adaptive feedback control for time-step selection has been developed for the diffusion process. The optimization is presented with efficient algorithms. Numerical experiments are discussed with respect to the diffusion processes of the macroscopic model.

1. Introduction

We motivate our studying on simulating a low-temperature low-pressure plasma that can be found in chemical vapor deposition (CVD) processes. In the last years the research and optimization in producing high-temperature films by depositing low-pressure processes have increased by using simulation tools, see [1, 2]. Theoretical models exist for deposition processes and can be modeled by coupled transport and flow equations, see [3, 4]. Further interest on standard applications to deposit titanium-nitrogen (TiN) and titanium-carbon (TiC) on metallic layers are immense, see [5]. Recently more and more focus on deposition with new material classes known as MAX-phases are becoming important, see [6, 7]. The MAX-phase are nanolayered terniar metal-carbides or -nitrids, where M is a transition metal, A is an A-group element (e.g., Al, Ga, In, Si, etc.) and X is C (carbon) or N (nitride), see [8]. Such materials combine ceramic and metallic behavior and can be implanted in the metallic bipolar plates to obtain a new material with noncorrosive and good metallic conductivity behavior.

We discuss a model for low-temperature and low-pressure plasma that can be used to implant or deposit thin layers of important materials, see [9]. This model is used for the implantation process. First, we present the process in the plasma-reactor that transport the contaminants to the wafer surface, see [10]. We deal with a continuous flow model, while we assume a vacuum- and a diffusion-dominated processes. Second, the process at the wafer-surface is modeled by the heavy particles problem with underlying drift. This model deals more with the atomic behavior and we do not allow $p = 0$, see [10].

To solve such optimization problems, we present a PID-controller (proportional, integral, differential) to control our deposition process, see [11]. We improved heuristic methods of deriving the PID parameters, while we discuss the posteriori error estimates respecting the time-step size control. Our contribution is a modified automatically step-size control and a best approximation is obtained with the time-dependent control method based on the Chien-Hrones-Reswick algorithm, see [12].

Numerical methods are described in the context of time- and spatial-discretization methods for the mesoscopic-scale model. We discussed different experiments and their convergence rates.

For the simulations we apply analytical and also numerical methods to obtain results to control the grow of thin layers.

The paper is outlined as follows. In Section 2 we present our mathematical model and a possible reduced model for the further approximations. In Section 3 we discuss the theoretical background for the simulation of CVD processes. The optimal control and their control paths based on the PID-control approach are discussed in Section 4. The software and program-tools are discussed in Section 5. The numerical experiments are given in Section 6. In the contents that are given in Section 7, we summarize our results.

2. Mathematical Model

In the following, the models are discussed in two directions of far-field and near-field problems:

(1) reaction-diffusion equations, see [13] (far-field problem);

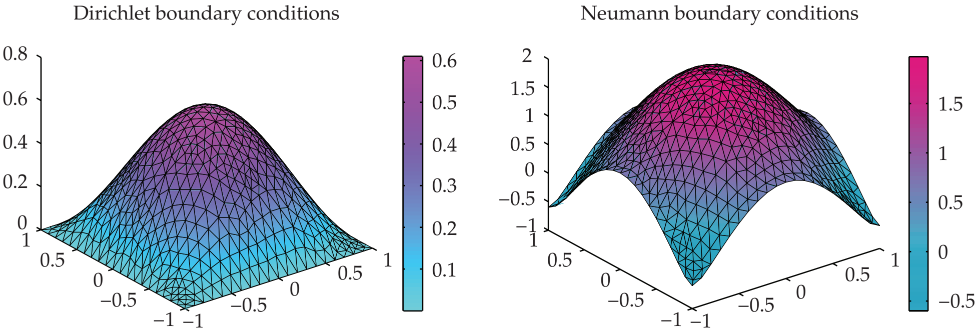
(2) Boltzmann-Lattice equations, see [9] (near-field problem).

The modeling is considered by the Knudsen Number (Kn), which is the ratio of the mean free path λ over the typical domain size L. For small Knudsen numbers $\text{Kn} \approx 0.01 - 1.0$, we deal with a Navier-Stokes equation or with the convection-diffusion equation, see [5, 14], whereas for large Knudsen numbers $\text{Kn} \geq 1.0$, we deal with a Boltzmann equation, see [4].

2.1. Modeling with Partial Differential Equations

Dynamic processes with modifications in time and space will be reshaped by partial differential equations. There is (i) the PDE-formula itself which describes the physical laws of nature that influence the process and (ii) initial and boundary conditions in which specific characteristics of the process, like boundary behavior, can be coded.

There are two types of boundary conditions, namely, Dirichlet and Neumann boundary. With the Dirichlet type the exact value of the boundary is known, however, with



(a)

(b)

Figure 1: Dirichlet and Neumann boundary conditions.

Neumann boundaries the spatial derivation of the boundary values in normal direction is known, see an example of the boundary conditions in Figure 1.

2.2. *Model for Optimal Control of the Layer*

We will concentrate us on a continuum model of mass transportation and assume that the energy and momentum is conserved, see [13]. Therefore, the continuum flow of the mass can be described as diffusion reaction equation given as

$$\partial_t c - \nabla D \nabla c - R_g = 0, \quad \text{in } \Omega \times [0,T], \tag{2.1}$$

$$c(x,0) = c_0(x), \quad \text{on } \Omega, \tag{2.2}$$

$$\frac{\partial c(x,t)}{\partial n} = c_1(x,t), \quad \text{on } \partial\Omega \times [0,T], \tag{2.3}$$

where c is the molar concentration, D is the diffusion parameter, and R_g is the reaction and source term.

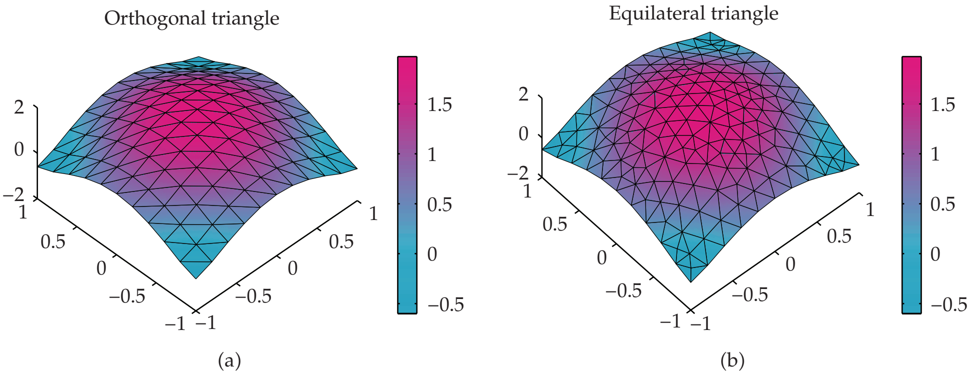
We modify our model equation (2.1) to a control problem with an additionally right-hand side source:

$$\begin{gathered} \partial_t c - \nabla D \nabla c = c_{\text{source}}, \quad \text{in } \Omega \times [0,T], \\ c(x,0) = c_0(x), \quad \text{on } \Omega, \\ \frac{\partial c(x,t)}{\partial n} = c_1(x,t), \quad \text{on } \partial\Omega \times [0,T], \end{gathered} \tag{2.4}$$

where $c_{\text{source}}(x,t) = R_g$ is the discontinuous or continuous source term of the concentration c and we neglect a reaction term of this concentration.

We assume an optimal concentration at the layer

$$c_{\text{opt}}(x,t), \tag{2.5}$$



(a)

(b)

Figure 2: Spatial-discretization.

where the layer is given as $x \in \Omega_{\text{layer}}$ and our constraints are given as

$$c_{\text{source,min}} \leq c_{\text{source}} \leq c_{\text{source,max}}. \tag{2.6}$$

Additionally, we have to solve the minimization problem:

$$\min(J(c, c_{\text{source}})) := \frac{1}{2}\int_T \int_{\Omega_{\text{layer}}} \left|c(x,t) - c_{\text{opt}}(x,t)\right|^2 dx\, dt + \frac{\lambda}{2}\int_T \int_{\Omega} \left|c_{\text{source}}(x,t)\right|^2 dx\, dt, \tag{2.7}$$

where T is the time period of the process.

Remark 2.1. We choose the L_2-error to control our minimization problem. In literature, see [15, 16], there exists further control-errors, which respect the time behavior.

In a first part, we only solve the transport equation with UG software-tool (unstructed grid software, see [17]) and try to find out the optimal control of the sources to obtain the best homogeneous layer.

In a second part, we consider the optimal control problem and solve also the backward problem.

3. Theoretical Background for Simulation of Diffusive CVD Processes

In what follows we discuss the approximation methods and errors for the simulation of the CVD processes.

3.1. Approximation and Discretization

For the numerical solutions we need to apply approximation methods, for example, finite-difference methods and iterative solver methods for the nonlinear differential equations, see [18, 19].

The finite-element discretization is based on Ω_h the variational boundary value problem reduces to find $u_h \in V_h$ satisfying the initial condition $u_h(0)$ such that

$$\int_{\Omega_h} \left(\frac{\partial u_h}{\partial t} v_h + D \nabla u_h \cdot \nabla v_h \right) dx = 0, \quad \forall v_h \in V_h. \tag{3.1}$$

We define the minimal length of triangle which we get from the spatial-discretization with Δx.

This leads to the following linear semidiscretized system of ordinary differential equations:

$$M \frac{du^*}{dt} + A u^* = 0, \tag{3.2}$$

where M is the mass and A the M-matrix.

Here we have taken into account the Courant-Friedrichs-Levy- (CFL-) condition, which is given as

$$\mathrm{CFL} = 2 D_{\max} \frac{\Delta t}{\min_{e \in E} \Delta x_e^2}, \tag{3.3}$$

where $D_{\max}$ is the maximal diffusion parameter, E is the set of the edges of the discretization. We restrict the CFL-condition to 1, if we use an explicit time-discretization and can lower the condition, if we use an implicit discretization.

For the explicit time-discretization, we apply explicit Euler or Runge-Kutta methods.

We use the explicit lower-order Runge-Kutta methods:

$$\begin{array}{c|cc} 0 & & \\ \frac{1}{2} & \frac{1}{2} & \\ \hline & 0 & 1 \end{array} \tag{3.4}$$

Furthermore we use the following Heun method (third-order):

$$\begin{array}{c|ccc} 0 & & & \\ \frac{1}{3} & \frac{1}{3} & & \\ \frac{2}{3} & 0 & \frac{2}{3} & 0 \\ \hline & \frac{1}{4} & 0 & \frac{3}{4} \end{array} \tag{3.5}$$

The implicit time-discretization is done with implicit Euler or Runge-Kutta methods.

Here, we use the implicit trapezoidal rule:

$$\begin{array}{c|cc} 0 & & \\ 1 & \frac{1}{2} & \frac{1}{2} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \tag{3.6}$$

Furthermore we use the following Gauss Runge-Kutta method:

$$\begin{array}{c|cc} \frac{1}{2}-\frac{\sqrt{3}}{6} & \frac{1}{4} & \frac{1}{4}-\frac{\sqrt{3}}{6} \\ \frac{1}{2}+\frac{\sqrt{3}}{6} & \frac{1}{4}+\frac{\sqrt{3}}{6} & \frac{1}{4} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \tag{3.7}$$

Remark 3.1. We apply implicit time-discretization methods for the pure diffusion part, where we apply explicit time-discretization methods for the pure convection part. Here we have to respect the CLF-condition, see [20].

3.2. Errors and Convergence Rate

For studying the errors and the convergence-rates in our test example, we have to define the following norm in two space-dimensions:

(i) discrete $L_{\max}$-norm:

$$\mathrm{err}_{L_{\max},\Delta x,\Delta t} = \max_{i=1}^{p} \left| c_{\mathrm{num}}(x_i, T) - c_{\mathrm{ref}}(x_i, T) \right|, \tag{3.8}$$

(ii) discrete L_1-norm:

$$\mathrm{err}_{L_1,\Delta x,\Delta t} = \sum_{i=1}^{p} \Delta x^2 \left| c_{\mathrm{num}}(x_i, T) - c_{\mathrm{ref}}(x_i, T) \right|, \tag{3.9}$$

(iii) discrete L_2-norm:

$$\mathrm{err}_{L_2,\Delta x,\Delta t} = \sqrt{\sum_{i=1}^{p} \Delta x^2 \left| c_{\mathrm{num}}(x_i, T) - c_{\mathrm{ref}}(x_i, T) \right|^2}, \tag{3.10}$$

where Δx is the spatial-step of the discretization, Δt is the time-step of the discretization, and T is the end-time of the computation. p is the number of grid points in the discretization method. c_{num} is the numerical solution and c_{ref} is the reference solution, computed at fine spatial- and time-grids.

The numerical convergence rate are given as follows.

(i) For the spatial error, we define

$$\rho_{L_2,\Delta x_1,\Delta x_2,\Delta t} = \frac{\log\left(\text{err}_{L_2,\Delta x_1,\Delta t}/\text{err}_{L_2,\Delta x_2,\Delta t}\right)}{\log(\Delta x_1/\Delta x_2)}, \tag{3.11}$$

where Δx_1 is the coarse, Δx_2 is the fine spatial grid-step, and Δt is the time-grid step for both results.

(ii) For the time error, we define

$$\rho_{L_2,\Delta x,\Delta t_1,\Delta t_2} = \frac{\log\left(\text{err}_{L_2,\Delta x,\Delta t_1}/\text{err}_{L_2,\Delta x,\Delta t_2}\right)}{\log(\Delta t_1/\Delta t_2)}, \tag{3.12}$$

where Δt_1 is the coarse, Δt_2 is the fine-time-step, and Δx is the spatial-grid step for both results.

We often use $\Delta x_2 = \Delta x_1/2$. In this case, we have $\rho_{L_2,\Delta x_1,\Delta x_2,\Delta t} = \rho_{L_2,\Delta x_1,\Delta t}$. Further we have to choose Δx_1 with respect to the $\Delta t \in I = [0, \Delta t_{\max}]$, which have maximal ρ. Thus we define $\operatorname{Arg\,Max}(\Delta x)$:

$$\operatorname{Arg\,Max}(\Delta x) := \underset{\Delta t \in I}{\arg\max}\, \rho_{L_2,\Delta x,\Delta x/2,\Delta t}. \tag{3.13}$$

4. Optimal Control Methods

Here we discuss the control of a diffusion equation with a feedback based on a PID-controller.

4.1. Forward Controller (Simple P-Controller)

The first controller we discuss is the simple P-controller, see [11]. A first idea is to control linearly the error of the solved PDE.

In Figure 3, we present the P-controller.

Our control problem is given with the control of the error to the optimal concentration of the layer and correct the source-flux:

$$\begin{gathered} \partial_t c - \nabla D \nabla c = c_{\text{source}}, \quad \text{in } \Omega \times [0,T], \\ c(x,0) = c_0(x), \quad \text{on } \Omega, \\ \frac{\partial c(x,t)}{\partial n} = c_1(x,t), \quad \text{on } \partial\Omega \times [0,T], \end{gathered} \tag{4.1}$$

where $c_{\text{source}}(x,t)$ is a discontinuous source flow of the concentration c.

We assume an optimal concentration at the layer with the concentration $c_{\text{opt}}(x,t)$, where the layer is given as $x \in \Omega_{\text{layer}}$ and 0 elsewhere.

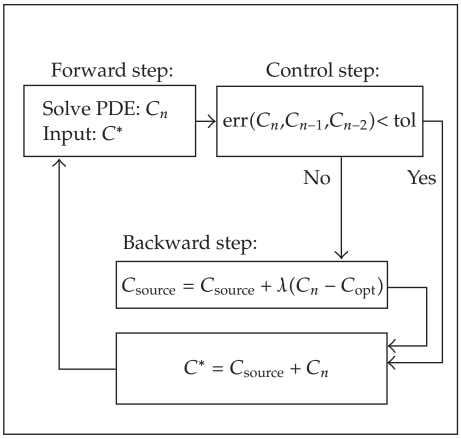

Forward step:
Solve PDE: C_n
Input: C^*
Control step:
err(C_n,C_{n-1},C_{n-2})< tol
No
Yes
Backward step:
$C_{\text{source}} = C_{\text{source}} + \lambda(C_n - C_{\text{opt}})$
$C^* = C_{\text{source}} + C_n$

Figure 3: P-controller for the solution C.

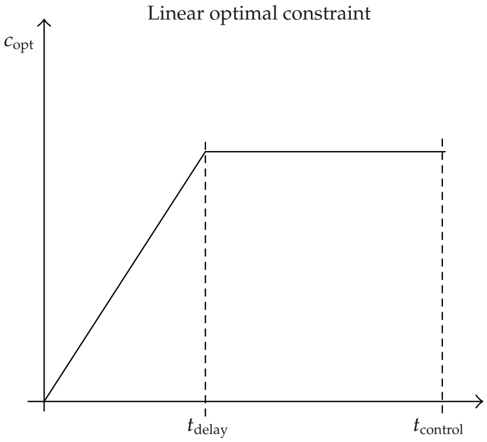


Figure 4: Linear constraint c_{opt} for the deposition process. x-axis: Time, y-axis: c (concentration).

Our constraints are bounded as

$$c_{\text{source,min}} \leq c_{\text{source}} \leq c_{\text{source,max}}. \tag{4.2}$$

Remark 4.1. Taken into account the hysteresis of the deposition process, we apply a linear increase of $c_{\text{source,max}}$ in the optimal control with respect to time, see Figure 4.

4.2. PID-Controller

The PID-controller is used to control temperature, motion, and flow. The controller is available in analog and digital forms, see [16]. The aim of the controller is to get the output (velocity, temperature, position) in the area of the constraint output, in a short time, with minimal overshoot, and with small errors.

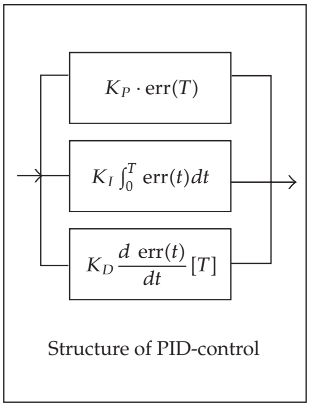

$K_P \cdot \mathrm{err}(T)$
$K_I \int_0^T \mathrm{err}(t)dt$
$K_D \dfrac{d\ \mathrm{err}(t)}{dt}[T]$
Structure of PID-control

Figure 5: PID-control. Effect on control system: the main influence in a control loop, K_P reduces a large part of the overall error. K_I reduces the final error in a system. Summing even a small error over time produces a drive signal large enough to move the system toward a smaller error. K_D counteracts the K_P and K_I terms when the output changes quickly. This helps reduce overshoot and ringing.

We have three elements in the PID-control, where P is the proportional part, I the integral part, and D is the derivative part of the controller, see [16].

These terms describe three basic mathematical functions applied to the error signal, error $= C_{\text{optimal}} - C_{\text{computed}}$.

The errors represented the difference between constraint (optimal set) and computed results in the simulation.

To accelerate a PID-controller means to adjust the three multipliers K_P, K_I, and K_D adding in various amounts of these functions to get the system to behave the way you want, see [11].

Figure 5 summarizes the PID terms and their effect on a control system.

Initialization of the PID-Controller

The algorithm of the initialization of the PID-control (i.e., search K_P, K_I, K_D) is given as in Algorithm 4.2 (see [15]).

Algorithm 4.2. (1) We initialize the P-controller: $K_I = 0.0, K_D = 0.0$.

(2) The amplifying factor K_P is increased till we reached the permanent oscillations as a stability boundary of the closed control system.

(3) We obtain for K_P the critical value $K_{P,\text{crit.}}$.

(4) The period-length of the permanent oscillation is given as T_{crit}.

(5) We obtain the parameters from Table 1.

Further we compute the rest parameters as $K_I = K_P/T_n, K_D = K_P * T_v$, see [21].

4.3. Adaptive Time-Control

Often the heuristic assumptions of the PID-parameters are too coarse.

One can improve the method by applying an adaptive step-size control.

Table 1: Heuristic derivation of the control parameters (Nichols-Ziegler).

Controller	K_P	T_n	T_v
P	$0.5K_{P,\mathrm{crit}}$		
PI	$0.45K_{P,\mathrm{crit}}$	$0.85T_{\mathrm{crit}}$	
PID	$0.6K_{P,\mathrm{crit}}$	$0.5T_{\mathrm{crit}}$	$0.12T_{\mathrm{crit}}$

We discuss the step-size control with respect to our underlying error, that is, given by the computed and optimal output of our differential equation.

Based on the adaptive control, we can benefit to accelerate the control problem.

According to Hairer and Wanner [22], we apply the automatic control problem with a PID-controller.

The automatically step-size is given as (see [11])

$$\Delta t_{n+1} = \left(\frac{e_{n-1}}{e_n}\right)^{K_P} \left(\frac{\mathrm{tol}}{e_n}\right)^{K_I} \left(\frac{e_{n-1}^2}{e_n e_{n-2}}\right)^{K_D} \Delta t_n, \tag{4.3}$$

where tol is the tolerance, e_n is the error of the quantities of interest in time-step Δt.

We can control the step-size with respect to our heuristically computed K_P, K_I, and K_D parameters. Initialization of the adaptive control can be seen in Algorithm 4.3.

Algorithm 4.3. (1) Define Tolerance, Min and Max of the concentration.

(2) Apply the parameters: K_P, K_D, K_I form a first run.

(3) Optimize the computations with a first feedback.

4.4. Identification of the Control Path

In the forward problem, we computed a PDE to simulate the CVD process in the underlying domain. For our control problem, we have the behavior of two points in the underlying domain, C_{source} our source and C_x the source restricted to the response of the controlled system. To analyze the differences between given source the C_{source} and the computed source in the control process C_x, we study the step response from our system:

$$x_a(t) := \left\{C_x(t) \mid C_{\mathrm{Source}}(t) = x_{e0}\right\}. \tag{4.4}$$

Algorithm 4.4. (1) We determine the model of the control path, for example, PT1 (Proportional time 1), PT2 (Proportional time 2), see [16]. For that purpose we investigate the step response x_a, his first and second derivate [16, pages 117 and 331].

(2) We determine the parameters of our model: Kp, T1, T2.

(3) Our goal is to control the system with a controller. Also we have to determine the control-parameter [16, page 405].

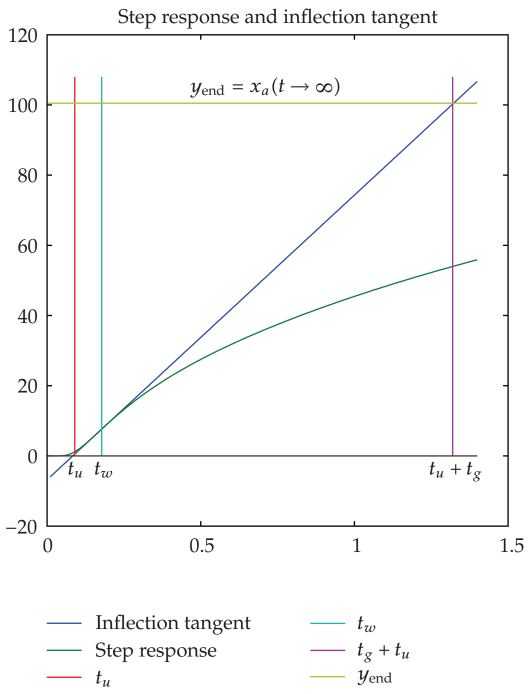


Figure 6: Step response and inflection tangent.

Identification of the Control Path

With the step response x_a, we identify the control path as a PT2 element.

Because $\lim_{t\to\infty} x_a(t) = \text{constant}$ (also $\neq 0$ nor ∞), the basic behavior of our element is P (and neither D nor I). Now we have to determine the number of time delays in the control path.

Identification of the Parameters

For such control path which has PT2 behavior, there exists a model as ordinary differential equation. Here a generalization can be obtained with four parameters and be divided by a_0 a normed form with three parameters ω_0 = angular frequency, A = attenuation, and K_S = proportional factor:

$$a_2 \cdot \ddot{x}_a(t) + a_1 \cdot \dot{x}_a(t) + a_0 \cdot x_a(t) = b_0 \cdot x_{e0},$$
$$\frac{1}{\omega_0^2} \cdot \ddot{x}_a(t) + \frac{2A}{\omega_0} \cdot \dot{x}_a(t) + x_a(t) = K_S \cdot x_{e0}. \tag{4.5}$$

The solution of the differential equation results in the characteristic equation:

$$\frac{\alpha^2}{\omega_0^2} + 2A \cdot \frac{\alpha}{\omega_0} + 1 = 0 \quad \text{with } \alpha_{1,2} = -A \cdot \omega_0 \pm \omega_0 \cdot \sqrt{A^2 - 1}. \tag{4.6}$$

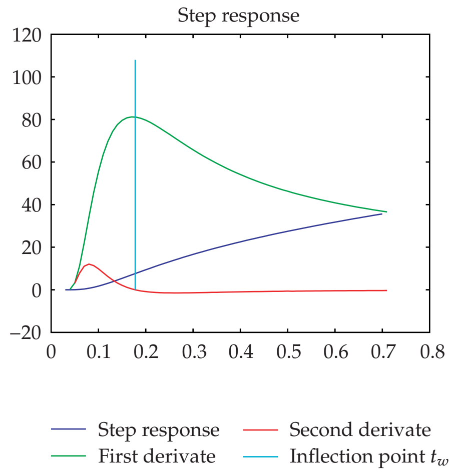


Figure 7: Empirical determination of the model of the control path. The three main cases: (1) $x_a(t=0)\neq 0:$ $\mathrm{P}, \mathrm{PDT}_1, \mathrm{PPT}_1, \mathrm{PIDT}_1$, (2) $x_a(t=0)=0, \dot{x}_a(t=0)\neq 0: \mathrm{I}, \mathrm{PT}_1, \mathrm{PDT}_2$, (3) $x_a(t=0), \dot{x}_a(t=0)=0, \ddot{x}_a(t=0)\neq 0:$ $\mathrm{IT}_1, \mathrm{PT}_2, \ldots$,(1) $x_a(0)=0$, (2) $\dot{x}_a(0)=0$, (3) $\ddot{x}_a(0)\neq 0$.

So that we have the analytical solution for the normed step response:

$$\frac{x_a(t)}{x_{e0}} = K_S \cdot \left[1 - \frac{T_1}{T_1 - T_2}\cdot e^{-t/T_1} + \frac{T_2}{T_1 - T_2}\cdot e^{-t/T_2}\right], \qquad \text{with } \alpha_1 = -\frac{1}{T_1}, \quad \alpha_2 = -\frac{1}{T_2}. \tag{4.7}$$

We go now the inverse way to identify the system. We start with the step response. Using the inflected tangent at $(t_w, x_w, \dot{x}_a(t_w))$ and the limit $x_a(t \to \infty)$, we obtain our parameters.

Firstly, we have $K_S = x_a(t \to \infty)/x_{e0}$, $t_g = K_S \cdot (x_{e0}/\dot{x}_a(t_w))$, and $t_u = t_w - t_g \cdot (x_a(t_w)/x_a(t \to \infty))$, see Figure 6. In the second step, we obtain T_1 and T_2 by the following formulas:

$$\frac{t_u}{t_g} = \frac{\alpha^{\alpha/(1-\alpha)}\cdot(\alpha\cdot\ln(\alpha)+\alpha^2-1)}{\alpha-1} - 1, \qquad \frac{t_g}{T_1} = \alpha^{\alpha/(1-\alpha)}, \qquad \alpha = \frac{T_2}{T_1}. \tag{4.8}$$

Control Parameters

The parameter occurrence also in the transfer function $G(s)$, which is the Laplace transformation of the differential equation. Our first choice to determine the control parameter is here the PT2 element. Alternatively, we use an approximation to a PT1 element, but here we have additional a scan-time parameter:

$$\begin{aligned} G(s) &= \frac{K_S}{(1+T_1 s)(1+T_2 s)} \text{ PT2 element},\\ G(s) &\approx K_S \cdot \frac{e^{-T_u s}}{1 + s\cdot T_g} \text{ PT1 element}. \end{aligned} \tag{4.9}$$

Table 2: Derivation of the control parameters (Chien-Hrones-Reswick and Takahashi).

PID	(PO) = 0	(PO) = 20	Takahashi
K_R	$0.6 \cdot \dfrac{t_g}{t_u \cdot K_S}$	$0.95 \cdot \dfrac{t_g}{t_u \cdot K_S}$	$\dfrac{1,2 \cdot t_g}{K_s \cdot (t_u + T)}$
T_n	$1.0 \cdot t_g$	$1.35 \cdot t_g$	$\dfrac{2 \cdot (t_u + T/2)^2}{t_u + T}$
T_v	$0.5 \cdot t_u$	$0.47 \cdot t_u$	$0.5(t_u + T)$

Table 3: MATLAB-toolbox pdetool.

MATLAB function	Description	Enhancement
Parabolic	Solve parabolic PDE (heat equation)	In the future we plan to use academical code to compute the PDE-solutions (e.g., θ-method). Furthermore we will use Comsol/Femlab
Pdetool	MATLAB toolbox to create the geometry of the FEM-structure and the boundary conditions	We have also implemented an alternative mesh, with orthogonal triangles. MATLAB always uses equilateral triangles
Refine	This function refines the geometry of the mesh. All triangles, were replaced with four new triangles	For convergence rate calculations we must guarantee that the geometry near the points, which are changed in the backward step (source), are similar
Guide	MATLAB toolbox to create graphical user interfaces (gui)	

Independent from the behavior which is being wished, we obtain values for the parameters K_P, K_I, and K_D, respectively. $K_R = K_P$, T_n and $T_v$are of the PID-controller. In Table 2 are given the values in dependence to the percentage overshoot (PO). There are the values of PO = 0, respectively, PO = 20. The rest can be determined by linear interpolation. An alternative schema is given by Takahashi. Here we have in addition the parameter scan-time T. This is a general form of the Nichols-Ziegler method (see Table 1). Further are $K_P = K_R$, $K_I = K_P/T_n$, and $K_D = K_P * T_v$.

5. Software and Program-Tools

MATLAB Functions

We use the MATLAB-toolbox pdetool for the time- and spatial-discretizations, where we have a finite-element method with $P1$-elements and an implicit Euler method for the time-discretization.

DEPOSIT-PID Toolbox

The PID-controller is also programmed in MATLAB. Our combined code is given in the DEPOSIT-PID toolbox and described in what follows. The DEPOSIT-PID toolbox is

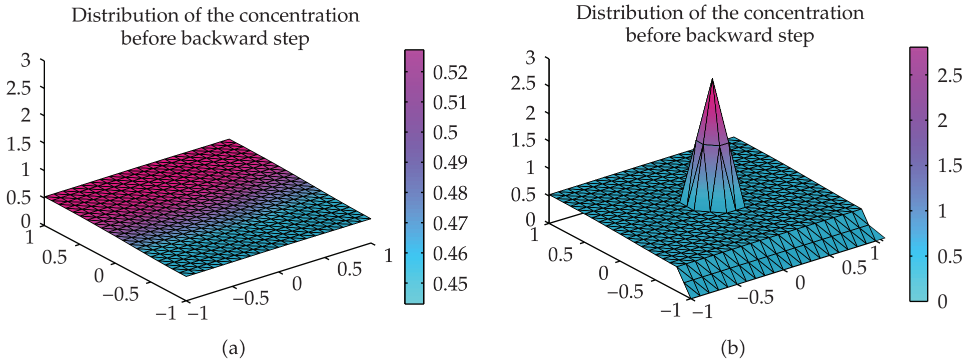


Figure 8: Backward step.

manipulated by a graphical user interface, by which simulation- and control-models are chosen and corresponding parameters can be manually adjusted.

The objective is to simulate the diffusion and deposition of the vapor in the apparatus and to obtain the optimal vapor concentration at the measuring point.

We can divide the process into three phases, namely, forward step, control step, backward step, which have a cyclic repetition. In the forward step, the diffusion takes place, which can be simulated by a time-step of the heat equation.

After that, in the control step, the actual concentration at the under boundary $(0, -1)$ can be measured (shown by computed in the graph) and compared with the optimal value (shown by optimal).

The control step is followed by the backward step: from the error in control step and the control model, the optimal alteration of the source can be computed. The vapor flows through the source point $(0, 0)$ in the apparatus and this value is shown by the SourceOutput in Figure 12. This will be simulated through the addition of SourceOutput to actual concentration at the source point (see Figure 8). Further, as a simplification, we set the concentration at the under boundary to zero, because here the gas transforms into solid matter .

In Section 4 we introduced some fundamentals in order to control the apparatus. The models and corresponding parameters can be altered by the Gui.

The Layout of the DEPOSIT-PID Gui

This Gui contains the following:

(1) short-time plot (2D) of computed, optimal, and SourceOutput;

(2) long-time plot (2D);

(3) listbox with names parameters:

   (a) textbox with actual value of the parameter chosen in [12],
   (b) textbox to change the value of the parameter chosen in [12];

(4) listbox with names parameters:

   (a) textbox with actual value of the parameter chosen in [23],
   (b) textbox to change the value of the parameter chosen in [23];

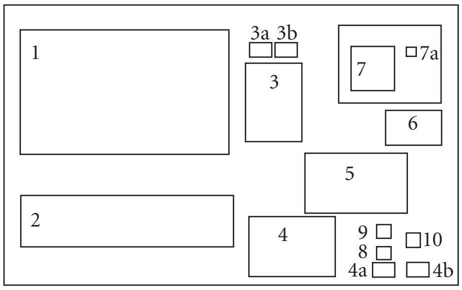


Figure 9: Layout of the DEPOSIT-PID Gui.

(5) 3D plot of distribution;

(6) 3D grid plot of distribution;

(7) listbox with names of parameters:

(a) checkbox with actual value of the parameter chosen in [24];

(8) push button: save;

(9) push button: reset;

(10) radio button: start.

KONTOOL

Another software-tool, KONTOOL, is programmed to compute the numerical convergence rates of the applications. The software-tool has implemented the errors and convergence rates defined in Section 3.2. An error-analysis based on successive refinement of space and time is done and the resulting errors and convergence rates are computed. Optimal convergence rates with respect to balance the time- and spatial-grids are calculated, see Algorithm 5.2.

Remark 5.1. The software-tool can be modified and applied to arbitrary spatial- and time-discretization methods. The interface of KONTOOL needs at least the parameters of the spatial- and time-grid and the starting parameters of the underlying methods.

Algorithm 5.2 (The algorithm of the computation of the numerical convergence tableau). (1) We compute reference solutions: (a) numerically: fine time and spatial steps or (b) analytically (if there exists an analytical solution).

(2) We apply one spatial discretization of step Δx and apply all time discretization with steps Δt, where the coarsest Δt is given by the CFL condition or till first non-numerical results as oscillations. We compute the error $c_{\text{num}} - c_{\text{ref}}$ in the L_2-norm.

(3) We continue the next fine spatial steps, for example, $\Delta x/2$.

(4) We compute the convergence tableau with time and space.

In the next section, we discuss the numerical experiments.

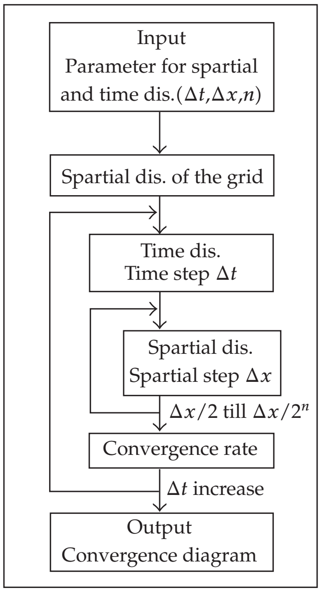


Figure 10: Convergence diagram tool.

6. Experiment for the Plasma Reactor

In this section, we present our numerical experiments for the CVD processes in a plasma reactor.

6.1. Simulation of a Diffusion Equation with Analytical Solution (Neumann Boundary Conditions)

Here we simulate a diffusion equation with Neumann boundary conditions and right-hand side 0. Our control problem has only the forward problem to solve and we consider the accuracy of our simulations.

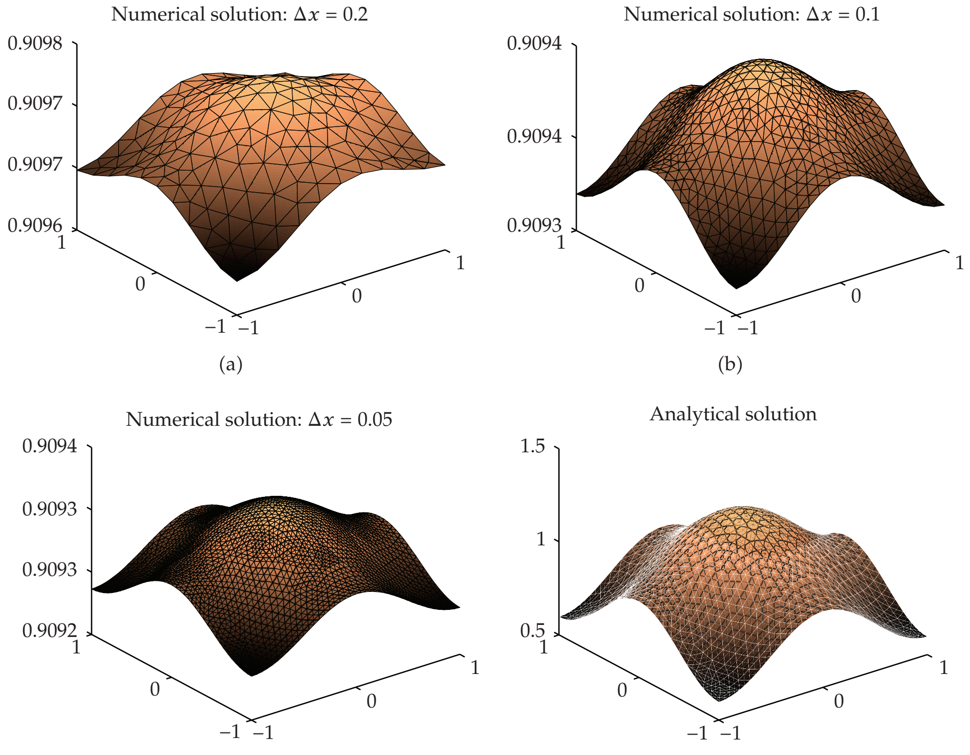
We have the following equation:

$$\begin{gathered}\partial_t c - \beta^2(\partial_{xx} + \partial_{yy})c = 0, \quad \text{in } \Omega \times [0,T], \\ c(x,y,0) = \cos(2x) + \cos(2y), \quad \text{on } \Omega, \\ \partial_n c(x,y,t) = 0, \quad \text{on } \partial\Omega \times [0,T],\end{gathered} \tag{6.1}$$

where c is the molar concentration, $\Omega = [-1,1] \times [-1,1]$, and $t \in (0,T)$. $D = \beta^2$ is the diffusion parameter of the diffusion equation.

We have the following analytical solution:

$$c_{\text{ana}}(x,y,t) = \sin(2) + \sum_{n=1}^{\infty} A_n \exp(-\beta^2 n^2 \pi^2 t)(\cos(n\pi x) + \cos(n\pi y)), \tag{6.2}$$



(a) (b) (c) (d)

Figure 11: 2D experiment of the diffusion equation at the end-time $T = 12.0$, $\beta = 0.1$. x-axis: x-coordinate, y-axis: y-coordinate and z-axis: c (concentration).

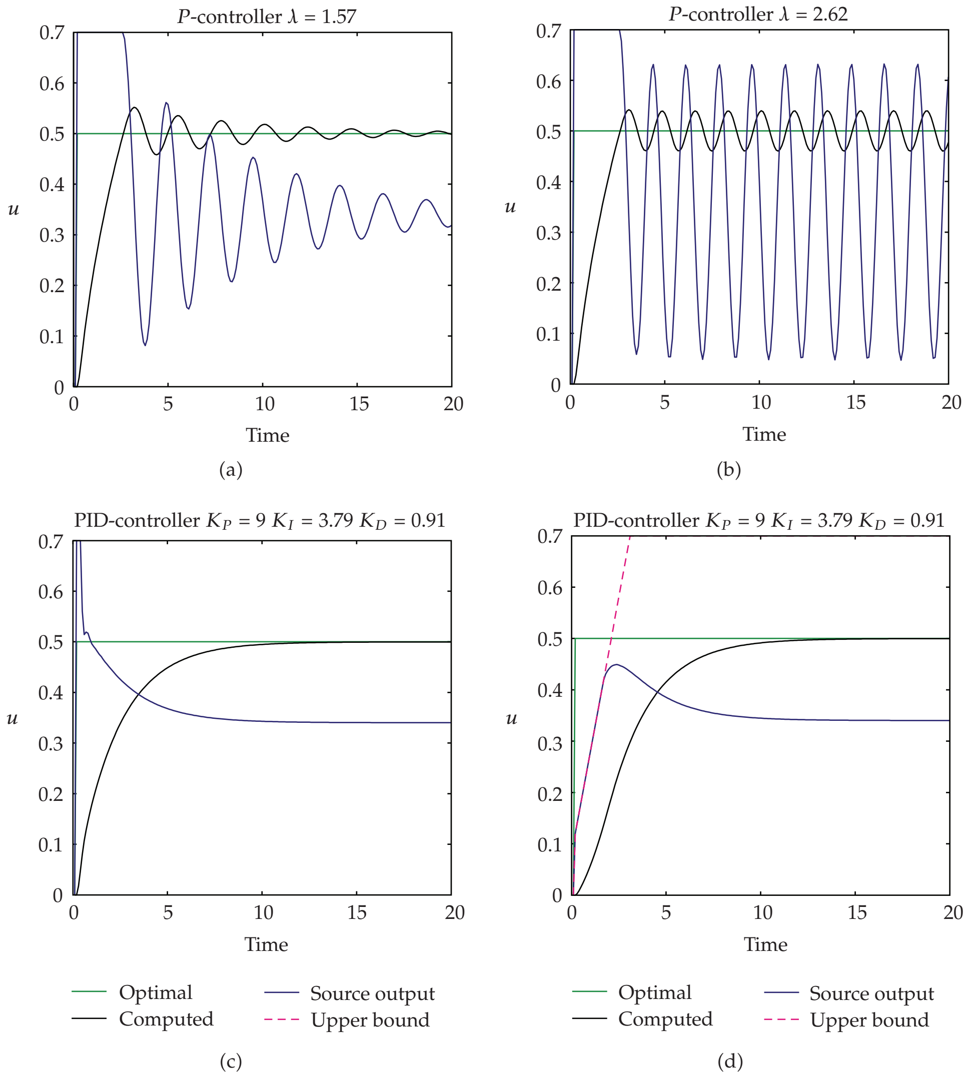
Table 4: Offset convergence ($\Delta t = 0.1$, time-step = 100, $\beta = 0.61644$).

Δx	offset(analy.)	max(num.)	min(num.)	L_1-error	L_2-error
0.20	0.90**92**97426826	0.90**21**08407643	0.90**21**08381295	7.189*e*–3	5.17*e*–5
0.10	0.90**92**97426826	0.90**75**14994036	0.90**75**14987008	1.782*e*–3	3.18*e*–6
0.05	0.90**92**97426826	0.90**88**58511500	0.90**88**58509666	4.389*e*–4	1.93*e*–7

where $A_n = (-1)^n(-4\sin(2)/(n^2\pi^2 - 4))$. We apply the diffusion coefficient $D = 0.01$, respectively, $\beta = 0.1$. We obtain the following result after time $T = 12.0$, see Figure 11.

We see that for large Δx the numerical solution converge faster to the stable constant endsolution (offset) than solution for smaller Δx and the analytical solution. The error between the constant analytical endsolution (offset, $\sin(2) = 0.9093$) is also greater than for smaller Δx. The L_2-error is given in Table 4.

Remark 6.1. We test for the pure diffusion equation our underlying discretization methods and apply finite elements for the spatial-discretization and implicit Runge-Kutta methods for the time-discretization. In the results, we obtain decreasing errors for the different time- and spatial-steps.



(a)

(b)

(c)

(d)

Figure 12: 2D experiment of the diffusion equation and control of a single point.

6.2. Simulation of an Optimal Control of a Diffusion Equation with Heuristic Choise of the Control Parameters

Here we simulate a first example of a diffusion equation and control the concentrations in the deposition process.

We have the following equation:

$$\begin{aligned} \partial_t c - \nabla D \nabla c &= f(t), \quad \text{in } \Omega \times [0, T], \\ c(x, 0) &= c_0(x), \quad \text{on } \Omega, \\ \frac{\partial c(x,t)}{\partial n} &= c_1(x, t), \quad \text{on } \partial\Omega \times [0, T], \end{aligned} \tag{6.3}$$

where c is the molar concentration, D is the diffusion parameter of the diffusion equation, and $f(t)$ is the right-hand side or diffusion source. We have the following constraint: $c_{\text{optimal}}(x_{\text{point}}, y_{\text{point}}) = 0.5$, where $(x_{\text{point}}, y_{\text{point}})$ is the control point in our domain. The parameters are given as $D = 0.01$ and $f(t) = at + b$, so we deal with a linear source, $a = 0.2$ and $b = 0.1$ are constants. In the following tests, we propose the 3 possibilities to control the optimal temperature:

(i) P-control with constant optimal constraint,

(ii) PID-control with constant optimal constraint,

(iii) PID-control with Linear optimal constraint.

The results for the control methods are given in Figure 12.

6.3. Preliminary Remark for Simulations for the Convergence Order

To determine the function Arg Max, firstly, we have to determine a practicable interval I for

$$\operatorname{Arg\,Max}(\Delta x) := \underset{\Delta t \in I}{\arg\max}\, \rho_{L_2, \Delta x, \Delta x/2, \Delta t}. \tag{6.4}$$

One possibility is the interval

$$I_{\text{CFL}} := (0, \Delta t_{\text{CLF}}], \tag{6.5}$$

with $\Delta t_{\text{CFL}} := \Delta s^2 / 2D_{\max}$ from the CFL-condition. In the experiments, we find another interval, where the convergence rate function is relative stable convex:

$$I_{\text{stable}} := (0, \Delta t_{\text{stable}}], \tag{6.6}$$

where

$$\Delta t_{\text{stable}} := \max_{\Delta t}\{\rho : (0, \Delta t) \longrightarrow \mathbb{R} \ \text{ is } \ \text{convex}\}. \tag{6.7}$$

It is clear that we can in I_{stable} take some restrictions to a subinterval $I_{\text{sub}} := [\Delta t_{s\min}, \Delta t_{s\max}] \subset I_{\text{stable}}$, when we know that some $\Delta t \in I_{\text{sub}}$ with $\rho(\Delta t) > \rho(\Delta t_{s\min}), \rho(\Delta t_{s\max})$.

The function of the numerical convergence rate is discrete in the spatial-discretization variable Δx since we get a finer discretization with a bisection of Δx. With a finer discretization, a triangle is replaced by four subtriangles.

In the temporal discretization variable Δt, we are in contrast not restricted to such conditions and Δx could be chosen to any arbitrary value above 0. To get a first glance, we have selected the methods of bisection. Subsequently, we consider finer discretizations in intervals which are of special interest.

Table 5: Numerical results for the P-controller for different spatial-steps with $D = 0.1$ and $\lambda = 1.0$ as P-value for the controller.

Δx	Δt	err = $\lvert u_{\mathrm{num},\Delta x,\Delta t} - u_{\mathrm{num,fine}\Delta x,\Delta t}\rvert$	Convergence rate
0.1	0.1	0.077007	3.2531
0.1	0.05	0.016153	4.0077
0.1	0.025	0.0020085	3.9447
0.1	0.0125	0.00026087	1.8276
0.1	0.00625	0.00014699	0
0.05	0.1	0.27873	2.8591
0.05	0.05	0.076833	3.7481
0.05	0.025	0.011437	4.052
0.05	0.0125	0.001379	3.7776
0.05	0.00625	0.00020111	0
0.025	0.1	0.6564	2.4552
0.025	0.05	0.23939	3.2449
0.025	0.025	0.050505	4.008
0.025	0.0125	0.0062781	3.9999
0.025	0.00625	0.00078482	0
0.0125	0.1	1.04	2.1424
0.0125	0.05	0.4711	2.7173
0.0125	0.025	0.14327	3.7251
0.0125	0.0125	0.021668	4.0516
0.0125	0.00625	0.0026133	0
0.00625	0.1	1.2092	1.586
0.00625	0.05	0.80554	2.4179
0.00625	0.025	0.30148	3.2179
0.00625	0.0125	0.064803	4.005
0.00625	0.00625	0.0080726	0

6.4. Simulations for the Convergence Order

We consider the convergence rate ρ as a function in dependent of Δx and Δt. This function also depends on some parameters, for instance, the diffusion coefficient D and the control parameter λ. We now present the results of two chosen experiments. For the first experiment, the parameters are $D = 0.1$ and $\lambda = 1$, while we increase the propagation velocity in such a way that we increase D to 1 for the second experiment.

We observe that, for instance, in the case of spatial-discretization $\Delta x = 0.05$ and $x = 0.25$, the maximal convergence rate lies between $\Delta t = 0.05$ and $\Delta t = 0.0125$. The maximum itself lies close to 0.025. To determine the precise value of $\Delta t = \operatorname{Arg\,Max}(\Delta x)$, we refine our method in these interesting intervals.

In Figure 13, we observe that there is a relatively stable convex area starting at 0. This area then switches over to an area with strong fluctuations. We now compare the stable area with the CFL-condition. The area where CFL < 1 lies in the stable convex area of the function ($\Delta t_{\mathrm{CFL}} < \Delta t_{\mathrm{stable}}$). For $\Delta x = 0.05$, $D = 0.1$, the CFL-area ends at $\Delta t_{\mathrm{CFL}} = 0.0125$, the stable convex area range to about $\Delta t_{\mathrm{stable}} = 0.2$. For our convergence diagram, we try to find for

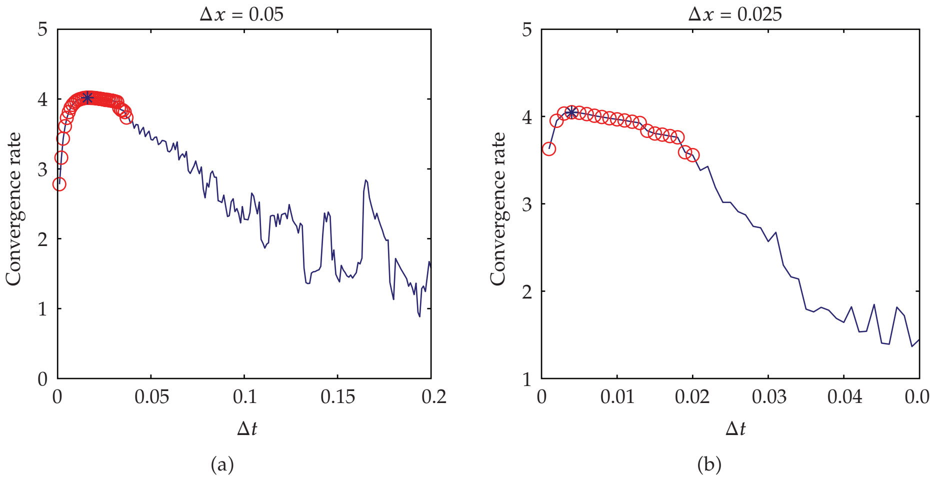


Figure 13: ρ for a P-controller, $D = 0.1$.

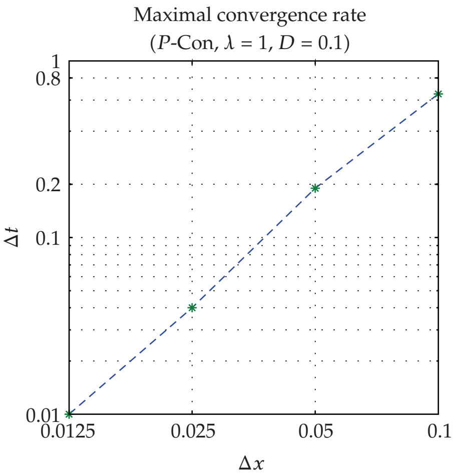


Figure 14: Convergence diagram (KONTOOL). We can see in the loglog-plot $\mathrm{Arg\,Max}(\Delta x)$ linear dependence.

every Δx the value Δt, where the convergence rate becomes maximal (ArgMax). It is clear that we only use Δt at which the convergence rate function is stable: $\Delta t < \Delta t_{\text{stable}}$.

Remark 6.2. The experiment shows the linear convergence rate of the P-controller with different λ values. So we obtain a stable method with respect to the P-controller. In the examples, we apply heuristic methods to derive the control parameters for the P- and PID-controller. We show that we have reached the linear order of the underlying finite element discretization method. We have higher control errors if we did not compute the correct control parameters and the numerical errors are smaller than our control error. To prohibit this problem, we have to compute in the next example the control parameters by a feedback equation, see [25].

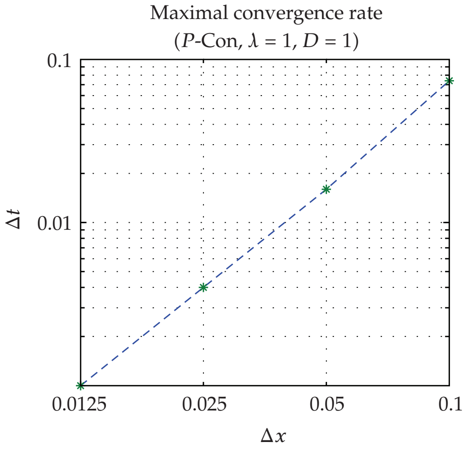


Figure 15: Convergence diagram (KONTOOL). The loglog-plot Arg Max(Δx) with linear dependence.

Table 6: The associated Δt, which have maximal convergence rate (Arg Max), for Δx = 0.1, 0.05, 0.025, 0.0125.

Δx	Arg Max	– log(Max)
.10000	.65000	0.62
.05000	.19000	2.4
.02500	0.04	4.64
.01250	.01000	6.64

Table 7: The associated Δt, which have maximal convergence rate (Arg Max), for Δx = 0.1, 0.05, 0.025, 0.0125.

Δx	Arg Max	– log(Max)
.10000	0.0700	3.84
.05000	0.0170	5.88
.02500	0.0040	7.97
.01250	0.0010	9.97

6.5. Simulation of an Optimal Control of a Diffusion Equation with Adaptive Control

In the second example, we simulate the diffusion equation and control the temperature with and adaptive control based on a PID-controller, see [11].

We have the following equation:

$$\begin{aligned} \partial_t c - \nabla D \nabla c &= f(t), \quad \text{in } \Omega \times [0,T], \\ c(x,t) &= c_0(x), \quad \text{on } \Omega, \\ \frac{\partial c(x,t)}{\partial n} &= c_1(x,t), \quad \text{on } \partial\Omega \times [0,T], \end{aligned} \tag{6.8}$$

where c is the molar concentration, D is the diffusion parameter of the diffusion equation, and $f(t)$ is the right hand side or source.

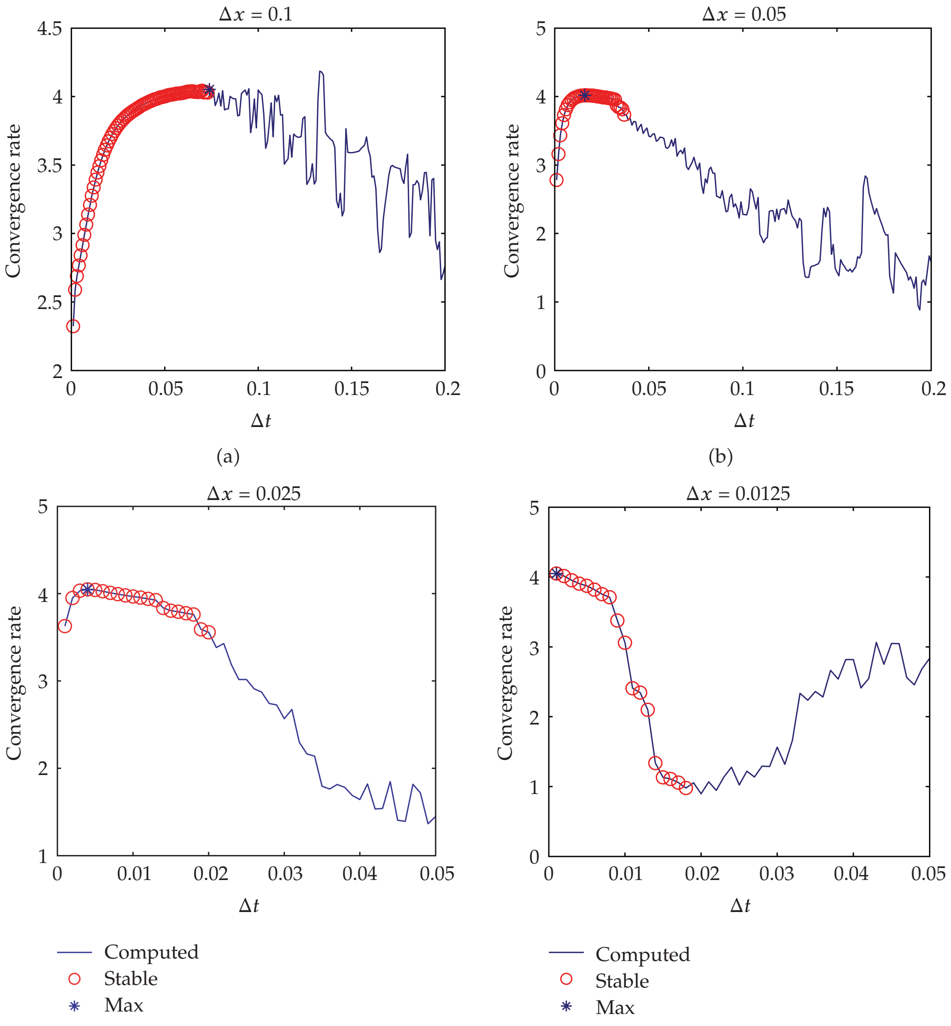


Figure 16: ρ for a P-controller, $D = 1$.

We have the following constraint:

$$c_{\text{optimal}}(x_{\text{point}}, y_{\text{point}}) = 0.5, \tag{6.9}$$

where $(x_{\text{point}}, y_{\text{point}}) = (0, -1)$ is the control point in our domain.

The automatically step-size is given as (see [11])

$$\Delta t_{n+1} = \left(\frac{e_{n-1}}{e_n}\right)^{K_P} \left(\frac{\text{tol}}{e_n}\right)^{K_I} \left(\frac{e_{n-1}^2}{e_n e_{n-2}}\right)^{K_D} \Delta t_n, \tag{6.10}$$

where tol = 1 is the tolerance, e_n is the error of the quantities of interest in time-step Δt.

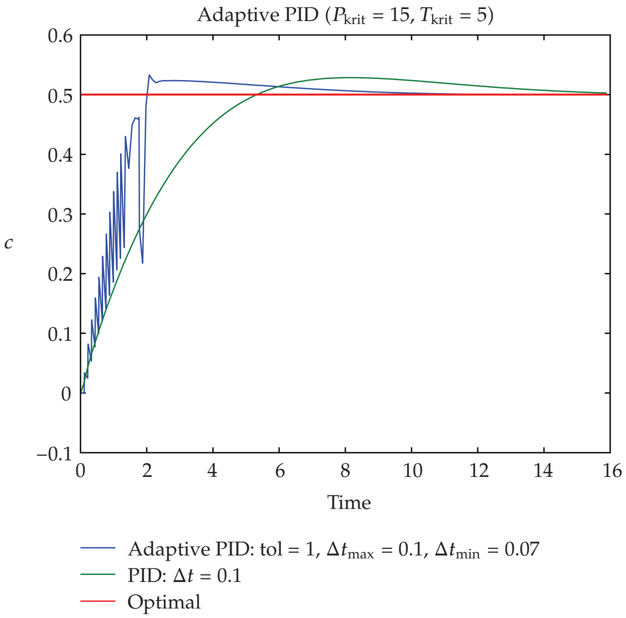


Figure 17: 2D experiment with and without the adaptive time-step control.

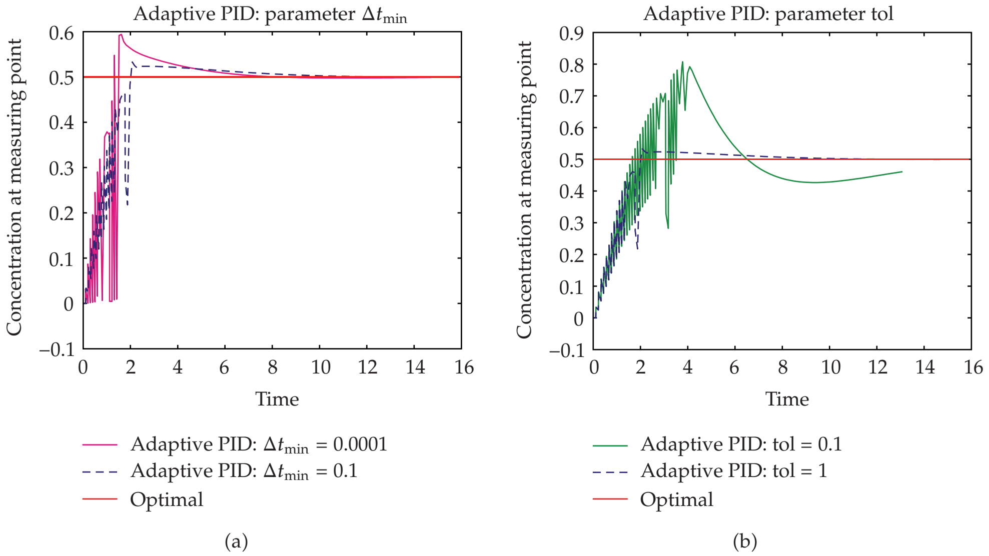


Figure 18: Adaptive PID with modified parameters.

The errors are given as

$$e_n = \frac{||u_n - u_{n-1}||}{||u_n||}, \tag{6.11}$$

where u_n is the result at time-step t^n.

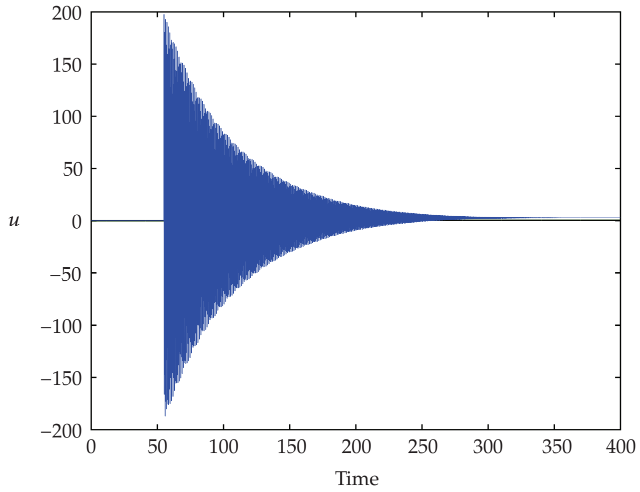


Figure 19: Error of the time-step control.

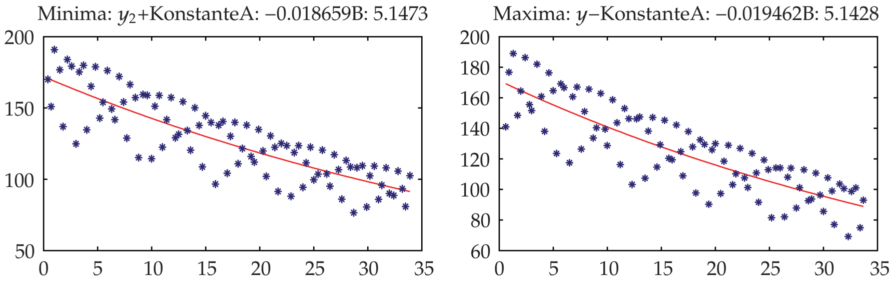
Table 8: Convergence of λ with bisection.

λ	a(max)	a(min)	Time-step
790.000	0.031267612875889	0.031262218711872	**2**
592.500	0.017928791447504	0.018012608268341	3
493.750	0.009755520357640	0.009663115651239	4
444.375	0.005011636633412	0.005007035757358	5
419.687	0.002464209438841	0.002504970973550	6
407.343	0.001131628414131	0.001182533773415	7
401.171	0.000466907455284	0.000465794576191	8
398.085	0.000109260068267	0.000189637255604	9
397.314	0.000080312399314	0.000043201893613	11
396.928	0.000041628241361	0.000012118492674	12
396.735	0.000002923384293	0.000006047318776	13
396.729			**18**
396.723	−0.000017536337301	−0.000000444521435	17
396.711	−0.000016718130883	−0.000004912794617	16
396.687	−0.000002204413679	−0.000022279535981	15
396.639	−0.000015370659931	−0.000016743767093	14
396.542	−0.000042906107641	−0.000015117641002	10
395.000	−0.000163960451645	−0.000164998927189	**1**

The parameters are given as $D = 0.1$, $P_{\text{krit}} = 15$, $T_{\text{krit}} = 5$, Δt (PID-control), tol $=$ 1 (adaptive PID-control), $\Delta t_{\max} = 0.1$ (adaptive PID-control), $\Delta t_{\min} = 0.01$ (adaptive PID-control) (see Figure 17).

Furthermore, we change the parameter tol $= 1$ to tol $= 0.1$ and $\Delta t_{\min} = 0.01$ to $\Delta t_{\min} =$ 0.0001 (see Figure 18).

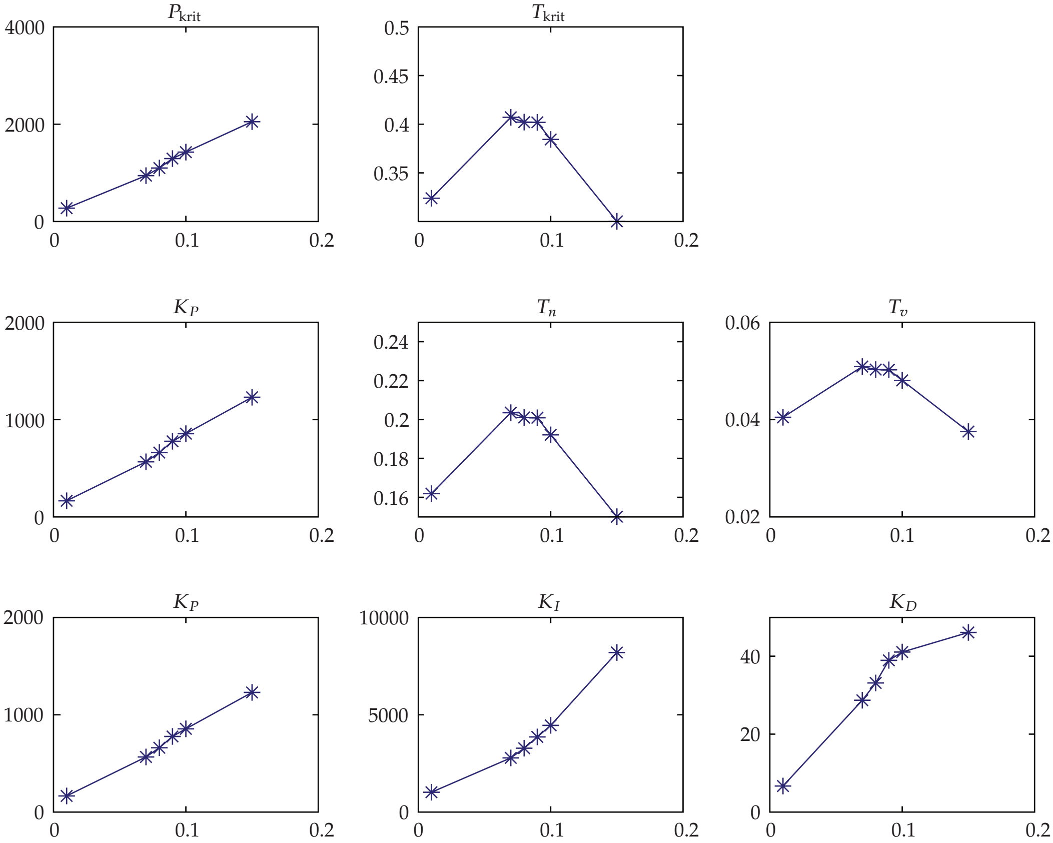
Remark 6.3. In Figures 17 and 18, we see an oscillating time interval at the beginning of the automatically step-size control. In the first experiments, we had only taken into account a



(a)

(b)

Figure 20: Exponential regression of min and max. Behavior of λ.



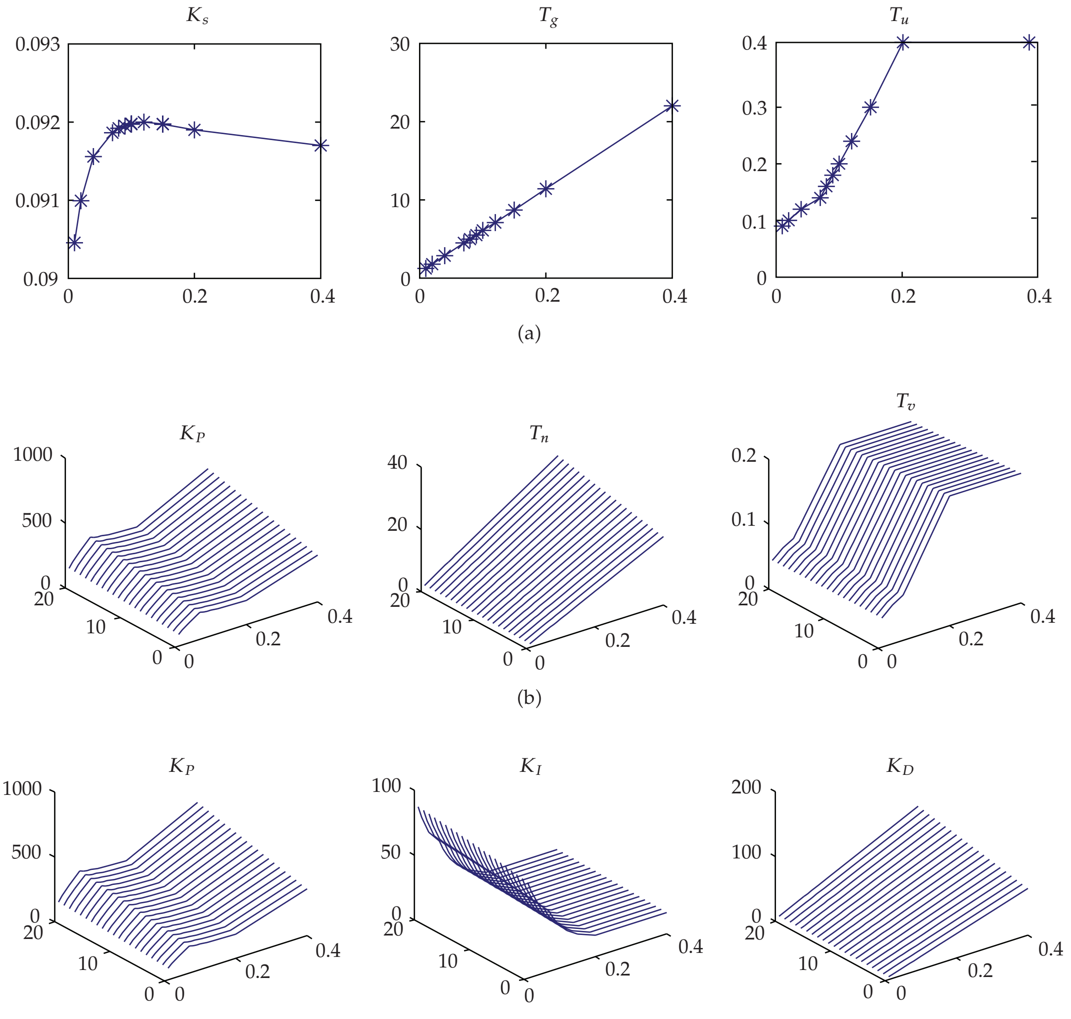
(a)

(b)

(c)

Figure 21: Δt dependence parameters determined by Nichols-Ziegler.

previous heuristically computation of the control parameters K_P, K_I, and K_D before the step-size control for the whole time-interval $0, T$. The optimal control parameters are given to the whole time-interval $0, T$. A modified algorithm to compute the control parameters for the initialization time-interval $0, t_{\text{begin}}$ and the whole time-interval $0, T$ overcome the oscillation problems.



(a)

(b)

(c)

Figure 22: Δt dependence parameters determined by Chien-Hrones-Reswick with PO = $0, \ldots, 20$.

A modified automatically step-size control, which minimize the oscillations is given in the following algorithm.

Algorithm 6.4. (1) We compute the reference control parameters $K_{P,\text{global}}$, $K_{I,\text{global}}$ and $K_{D,\text{global}}$ for the time-interval $0, T$.

(2) We apply the automatically step-size control for the global control parameters with $\text{tol}_{\text{global}}$, $\Delta t_{\max,\text{global}}$ and $\Delta t_{\min,\text{global}}$, which can by chosen large.

(3) We stop the computation till we reach the optimal solution and mark remember the time t_{oszill}.

(4) We compute the local control parameters $K_{P,\text{local}}$, $K_{I,\text{local}}$ and $K_{D,\text{local}}$ for the time-interval $0, t_{\text{oszill}}$.

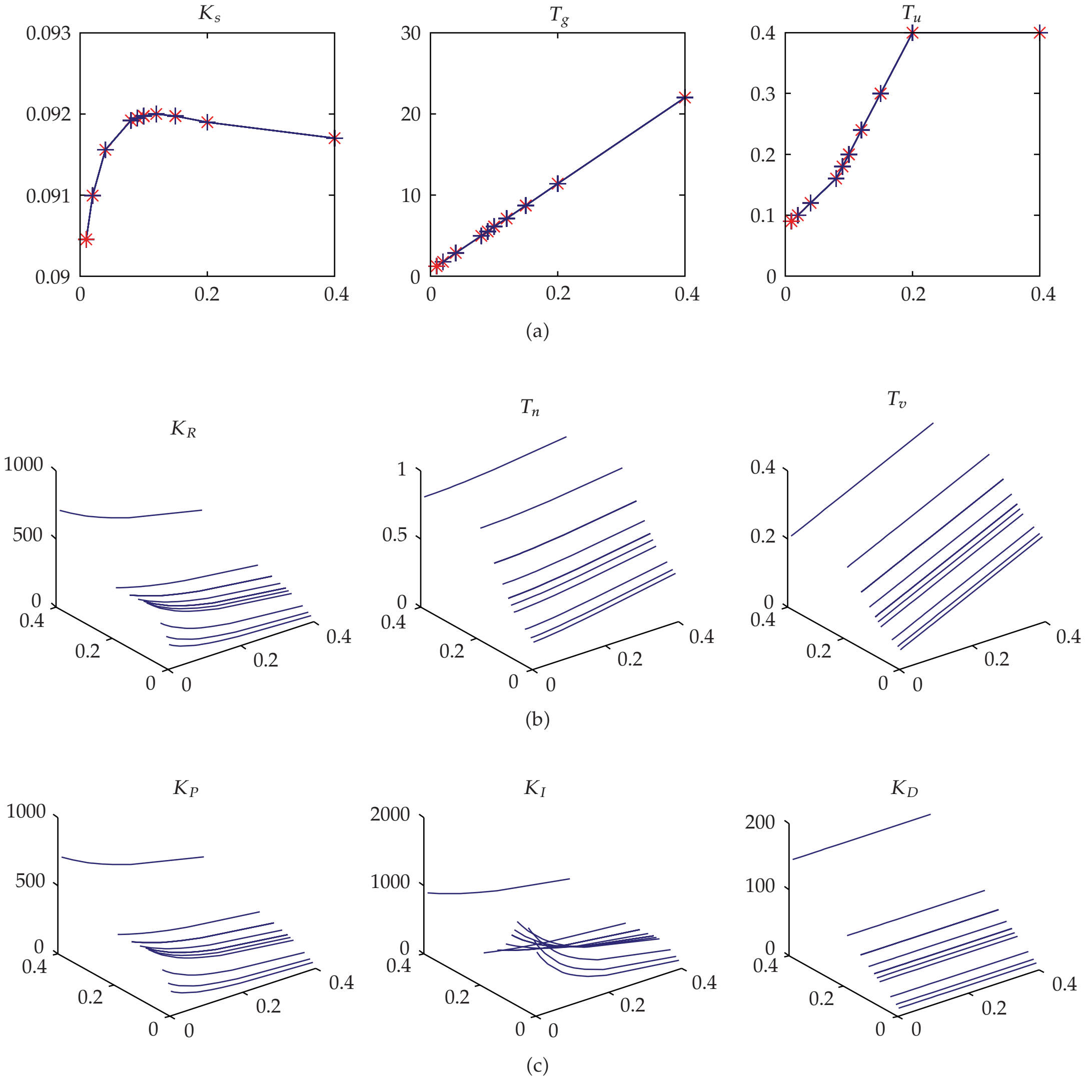


Figure 23: Δt dependence parameters determined by Takahashi.

(5) We restart the computation with the local control parameters and smaller step-size parameters $\text{tol}_{\text{local}}$, $\Delta t_{\text{max,local}}$ and $\Delta t_{\text{min,local}}$ till we reach t_{oszill} and continue the computation with the global parameters.

(6) We stop the computation if we reach $t = T$. If we obtain also high oscillation with the local parameters, we refine the local interval and go to step (3).

Remark 6.5. The modified automatically step-size control had taken into account the local behavior of the control problem. We could adapt the control parameters K_P, K_I, and K_D with respect to the local time-intervals. This modified algorithm considers a local time behavior more accurate and reduces oscillations at the initialization process.

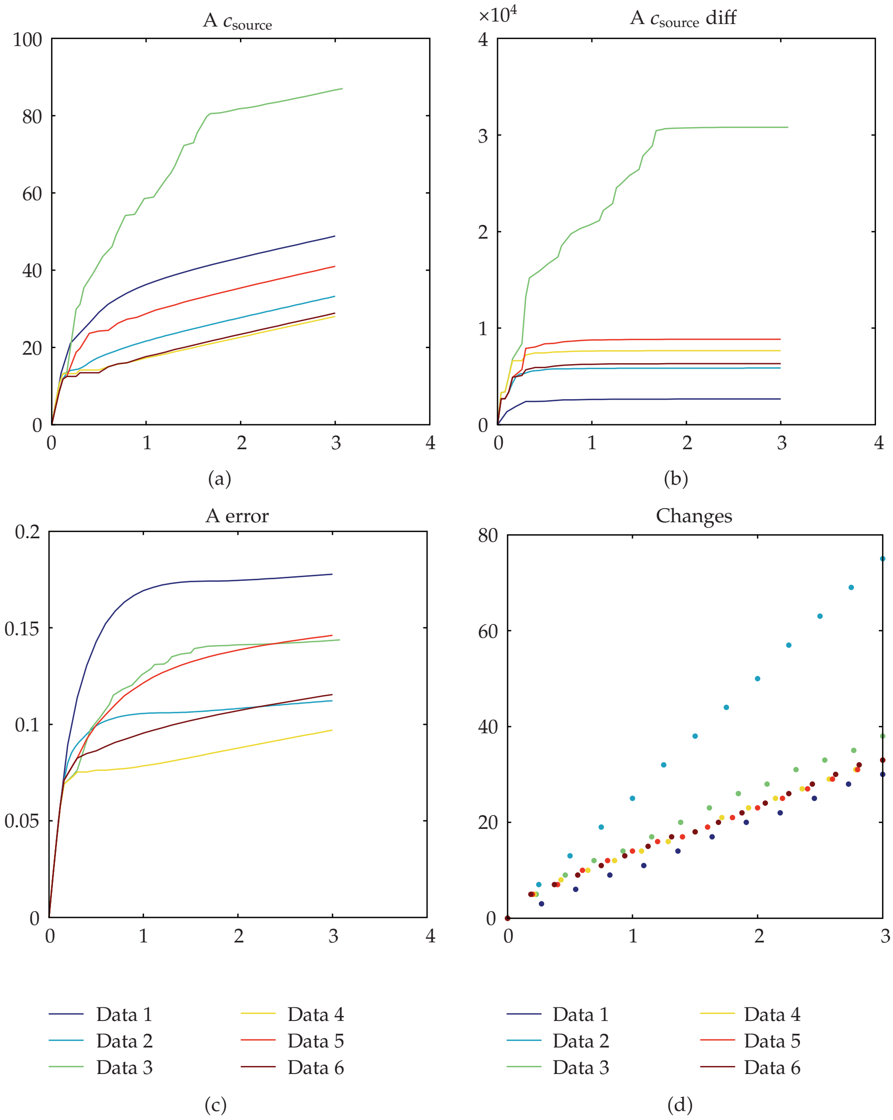


Figure 24: Parameters chosen by Chien-Hrones-Reswick: Summands of the minimization problem.

6.6. Simulation of an Optimal Control of a Diffusion Equation with Adaptive Control and Recovering of the Control Parameters

In what follows, we present the adaptive control based on Algorithm 6.4.

To be more precise the computation of global control parameters $K_{P,\text{global}} = K_P(\Delta t)$, $K_{I,\text{global}} = K_I(\Delta t)$, and $K_{D,\text{global}} = K_D(\Delta t)$ can be automatized in an interval $\Delta t \in (0, T)$.

We improve the automatically step-size, which is given in Section 4.3 to

$$(\Delta t)_{n+1} = \left(\frac{e_{n-1}}{e_n}\right)^{K_P(\Delta t)} \left(\frac{\text{tol}}{e_n}\right)^{K_I(\Delta t)} \left(\frac{e_{n-1}^2}{e_n e_{n-2}}\right)^{K_D(\Delta t)} (\Delta t)_n, \tag{6.12}$$

where tol is the tolerance, e_n is the error of the quantities of interest in time-step n.

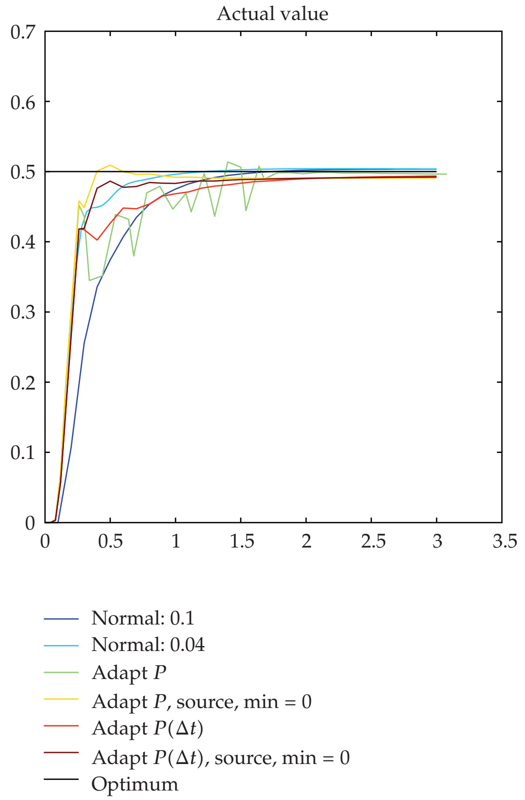


Figure 25: Parameters chosen by Chien-Hrones-Reswick: Actual value.

With approximations to each subintervals of $(0, T)$, we derive the control parameters $K_P(\Delta t)$, $K_I(\Delta t)$, and $K_P(\Delta t)$.

The improved globalized control parameters are computed in the following algorithm.

Algorithm 6.6. Computing of the control parameters $K_{P,\text{global}} = K_P(t)$, $K_{I,\text{global}} = K_I(t)$ and $K_{D,\text{global}} = K_D(t)$ for the time-interval $0, T$.

(1) Compute the critical control function with the minima and maxima. Based on the optimal oscillated function, we can derive the $K_P(T/3)$ for the interval $(0, T/3)$.

(2) Redo the step (1) for the intervals $(T/3, 2T/3)$ and $2T/3, T$ and approximate the function $K_P(t)$. Based on the $K_P(t)$ we can derive the $K_I(t)$ and $K_D(t)$ function.

(3) The optimal control parameters are used in Algorithm 6.4.

Remark 6.7. Further ideas to determine the time-dependent parameters $K_P(\Delta t)$, $K_I(\Delta t)$, and $K_D(\Delta t)$ can be done by using our identification method from Algorithm 4.4 in Section 4.4. There we have discussed two methods to determine the parameters from the control path, namely, Chien-Hrones-Reswick and Takahashi. Alternatively, we have the method from Algorithm 4.2 (Nichols-Ziegler) in Section 4.2. For the identification of the control path, we have only look to the step response. The alternative method demanded to find the critical P-value.

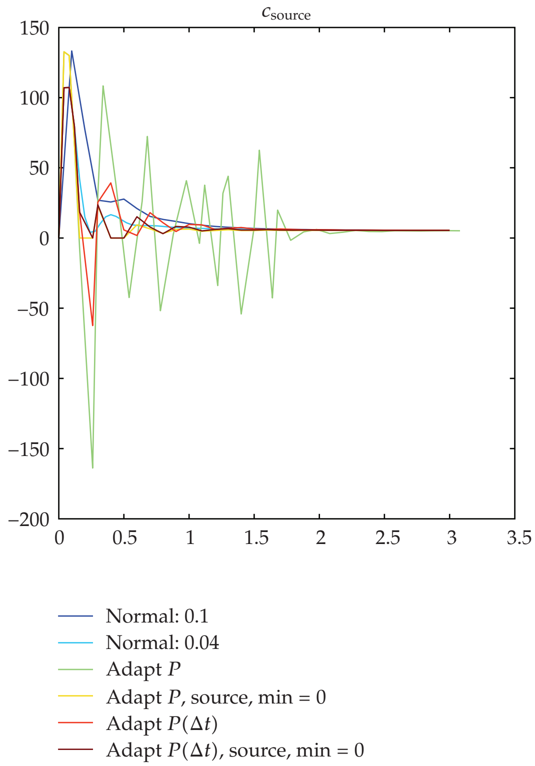


Figure 26: Parameters chosen by Chien-Hrones-Reswick: Source.

6.7. Determinate of Critical P with Exponential Regression

To initialize the PID-controller, we have to derive the K_P, K_D, and K_I parameters.

An idea is derived with the step function response, see [15].

This idea has also be included into the time-step control of the PID-controller.

In Figure 19, we can see that the maximum and minimum of the error function, done with the step function response, have a exponential behavior.

So we use the exponential regression

$$y = b \cdot \exp(a \cdot t) + \epsilon, \tag{6.13}$$

with $\epsilon \sim N(0, \sigma^2)$ normal distributed. Via log the exponential regression can be formed in a linear regression:

$$\begin{aligned} y &= b \cdot \exp(a \cdot t), \\ \log(y) &= \log(b) + a \cdot t. \end{aligned} \tag{6.14}$$

Before we have to convert the minima into the positive area.

Let $\lambda(n)$ be the actual lambda value, $\lambda(n)^+$ the next greater λ, and $\lambda(n)^-$ the next smaller one. Firstly, we use

$$\lambda(n+1) = \begin{cases} \dfrac{(\lambda(n)+\lambda(n)^+)}{2} : & a(\max, n) < 0,\ a(\min, n) > 0\ (\lambda \text{ to small}), \\ \dfrac{(\lambda(n)+\lambda(n)^-)}{2} : & a(\max, n) > 0,\ a(\min, n) < 0\ (\lambda \text{ to great}). \end{cases} \tag{6.15}$$

In the first step and in limit case, we use $\lambda(n+1) = 2 \cdot \lambda(n)$, respectively, $\lambda(n+1) = \lambda(n)/2$. In the first experiment, we start with $\lambda = 359$, which we have found with manual tests. The automatic found that the value 396.729 the a-value of $\lambda = 3.59$ have 3 zeros, the new less than five.

Remark 6.8. The adaptive step-size control of the PID-controller had taken into account the optimization in each local time-interval. We consider a larger time-interval and derive the optimal control parameters $K_{P,\text{global}}$, $K_{I,\text{global}}$, and $K_{D,\text{global}}$. Based on this parameterization, we apply the adaptation in time to be optimal in such a large time-interval.

6.8. Results of Our Experiments to Find the Time-Dependent Parameters

To automatize the adaptive time-step algorithm, we have to compute the step response and derive the K_P, K_I, and K_D parameters of this process. Here, we present the experiments in Figure 21, which had taken into account the critical values P_{crit} and T_{crit} to derive the optimal K_P, K_D, and K_I parameters based on Algorithm 6.6.

A further algorithm to derive the control parameters is done by Chien et al., see [12].

The results are given in Figure 23.

The K_P, K_D, and K_I parameters are derived with Algorithm 6.6.

6.9. Experiments with Time-Dependent Parameters

In these experiments, we concentrate on the Chien-Hrones-Reswick method, see [12].

To automatize the adaptive time-step algorithm, we have to compute the step response and derive the K_P, K_I, and K_D parameters of this process. The minimization problem is presented in Figure 24, which had taken into account the critical parameters.

In Figure 25, we see the step function response and the best fit of this function with the adaptive time control method. Because of the sensitivity to the initial problem, we can overcome the oscillation with the Chien-Hrones-Reswick method.

In Figure 26, we see the comparison of our different adaptive methods. Here the best method is the benefit of the time adaptive control method based on Algorithm 6.6.

Remark 6.9. The benefit of the time-dependent control parameters $K_P(\Delta t)$, $K_D(\Delta t)$, and $K_I(\Delta t)$ are important in the initialization of the control problem. Based on the fine-time scales globally chosen, parameters neglect the local time behavior. In the experiments, we found out the accelerated relaxation behavior and the fast control to the optimal value. Such adaptive algorithms to optimize the control parameters can improve the control methods and had taken into account the local-time scales.

7. Conclusions and Discussions

We present a continuous or kinetic model, due to the fare-field or near-field effect of our deposition process. We discuss the PID-controller to automatize our deposition process. Due to heuristic methods of deriving the PID parameters, we discuss aposteriori error estimates to automatize the time-stepping methods. A modified automatically step-size control is discussed and the best approximations are obtained with the time-dependent control method based on the Chien-Hrones-Reswick algorithm. For the mesoscopic-scale model, we discussed different experiments and their convergence rates. In future, we will analyze the validity of the models with physical experiments.

References

[1] V. Hlavacek, J. Thiart, and D. Orlicki, "Morphology and film growth in CVD reactions," *Journal de Physique IV*, vol. 5, pp. 3–44, 1995.

[2] H. Rouch, "MOCVD research reactor simulation," in *Proceedings of the COMSOL Users Conference*, pp. 1–7, Paris, France, November 2006.

[3] M. A. Lieberman and A. J. Lichtenberg, *Principle of Plasma Discharges and Materials Processing*, John Wiley & Sons, New York, NY, USA, 2nd edition, 2005.

[4] M. Ohring, *Materials Science of Thin Films*, Academic Press, San Diego, Calif, USA, 2nd edition, 2002.

[5] S. Middleman and A. K. Hochberg, *Process Engineering Analysis in Semiconductor Device Fabrication*, McGraw-Hill, New York, NY, USA, 1993.

[6] M. W. Barsoum and T. El-Raghy, "Synthesis and characterization of a remarkable ceramic: Ti_3SiC_2," *Journal of the American Ceramic Society*, vol. 79, no. 7, pp. 1953–1956, 1996.

[7] C. Lange, M. W. Barsoum, and P. Schaaf, "Towards the synthesis of MAX-phase functional coatings by pulsed laser deposition," *Applied Surface Science*, vol. 254, no. 4, pp. 1232–1235, 2007.

[8] P. Eklund, A. Murugaiah, J. Emmerlich, et al., "Homoepitaxial growth of Ti-Si-C MAX-phase thin films on bulk Ti_3SiC_2 substrates," *Journal of Crystal Growth*, vol. 304, no. 1, pp. 264–269, 2007.

[9] T. K. Senega and R. P. Brinkmann, "A multi-component transport model for non-equilibrium low-temperature low-pressure plasmas," *Journal of Physics D*, vol. 39, no. 8, pp. 1606–1618, 2006.

[10] J. Geiser and M. Arab, "Modelling, Optimzation and Simulation for a Chemical Vapor Deposition," *Journal of Porous Media, Begell House Inc., Redding, USA*, June 2008.

[11] A. M. P. Valli, G. F. Carey, and A. L. G. A. Coutinho, "Control strategies for timestep selection in simulation of coupled viscous flow and heat transfer," *Communications in Numerical Methods in Engineering*, vol. 18, no. 2, pp. 131–139, 2002.

[12] K. L. Chien, J. A. Hrones, and J. B. Reswick, "On the automatic tuning of generalized passive systems," *Transactions of the ASME*, vol. 74, pp. 175–185, 1952.

[13] M. K. Gobbert and C. A. Ringhofer, "An asymptotic analysis for a model of chemical vapor deposition on a microstructured surface," *SIAM Journal on Applied Mathematics*, vol. 58, no. 3, pp. 737–752, 1998.

[14] H. H. Lee, *Fundamentals of Microelectronics Processing*, McGraw-Hill, New York, NY, USA, 1990.

[15] N. B. Nichols and J. G. Ziegler, "Optimum settings for automatic controllers," *Transactions of the ASME*, vol. 64, pp. 759–768, 1942.

[16] H. Lutz and W. Wendt, *Taschenbuch der Regelungstechnik*, Harri-Deutsch, Frankfurt, Germany, 6th edition, 2005.

[17] P. Bastian, K. Birken, K. Johannsen, et al., "UG—a flexible software toolbox for solving partial differential equations," *Computing and Visualization in Science*, vol. 1, no. 1, pp. 27–40, 1997.

[18] J. Geiser, "Discretization methods with embedded analytical solutions for convection-diffusion dispersion-reaction equations and applications," *Journal of Engineering Mathematics*, vol. 57, no. 1, pp. 79–98, 2007.

[19] R. J. LeVeque, *Finite Volume Methods for Hyperbolic Problems*, Cambridge Texts in Applied Mathematics, Cambridge University Press, Cambridge, UK, 2002.

[20] W. Hundsdorfer and J. Verwer, *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*, vol. 33 of *Springer Series in Computational Mathematics*, Springer, Berlin, Germany, 2003.

[21] J. Lee and Th. F. Edgar, "Continuation method for the modified Ziegler-Nichols tuning of multiloop control systems," *Industrial and Engineering Chemistry Research*, vol. 44, no. 19, pp. 7428–7434, 2005.

[22] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations. II: Stiff and Differential-Algebraic Problems*, vol. 14 of *Springer Series in Computational Mathematics*, Springer, Berlin, Germany, 2nd edition, 1996.

[23] I. Faragó and Á. Havasi, "On the convergence and local splitting error of different splitting schemes," *Progress in Computational Fluid Dynamics*, vol. 5, no. 8, pp. 495–504, 2005.

[24] K.-J. Engel and R. Nagel, *One-Parameter Semigroups for Linear Evolution Equations*, vol. 194 of *Graduate Texts in Mathematics*, Springer, New York, NY, USA, 2000.

[25] T. Yamaguchi and K. Shimizu, "Asymptotic stabilization by PID control: stability analysis based on minimum phase and high-gain feedback," *Electrical Engineering in Japan*, vol. 156, no. 1, pp. 44–53, 2006.