DELFT UNIVERSITY OF TECHNOLOGY

REPORT 07-08

Transient Chemical Vapor Deposition Simulations

S. van Veldhuizen C. Vuik C.R. Kleijn

ISSN 1389-6520

Reports of the Department of Applied Mathematical Analysis

Delft 2007

Transient Chemical Vapor Deposition Simulations

S. van Veldhuizen* C. Vuik* C.R. Kleijn†

Abstract

The numerical modeling of laminar reacting gas flows in thermal Chemical Vapor Deposition (CVD) processes commonly involves the solution of convection-diffusion-reaction equations for a large number of reactants and intermediate species. These equations are stiffly coupled through the reaction terms, which typically include dozens of finite rate elementary reaction steps with largely varying rate constants. The solution of such stiff sets of equations is difficult, especially when time-accurate transient solutions are required. In this study various numerical schemes for multidimensional transient simulations of laminar reacting gas flows with homogeneous and heterogeneous chemical reactions are compared in terms of efficiency, accuracy and robustness. As a test case, we study the CVD process of silicon from silane, modeled according to the classical 16 species, 27 reactions chemistry model for this process as published by Coltrin et al. (1989).

1 Introduction

The growth of thin solid films via Chemical Vapor Deposition (CVD) is of considerable importance in the micro-electronics industry. Other applications of thin solid films via CVD can for instance be found in the glass industry as protective and decorative layers. The CVD process considered in this paper involves the deposition of silicon in an atmospheric pressure, cold wall, stagnation flow single wafer reactor, starting from the thermal decomposition of silane at the heated susceptor surface. This CVD process was one of the very first for which a detailed chemistry mode, based on a large number of elementary reaction steps leading to the formation of many intermediate species, has been proposed in literature by Coltrin et al. (1989). The numerical modeling realistic CVD processes and equipment, based on such detailed chemistry models, involves the solution of multi-dimensional convection-diffusion-reaction equations for a large number of reactants and intermediate species. These equations are stiffly coupled through the reaction terms, which typically include dozens of finite rate elementary reaction steps with largely varying

*Delft University of Technology, Delft Institute of Applied Mathematics, Mekelweg 4, 2628 CD Delft, The Netherlands (s.vanveldhuizen@tudelft.nl, c.vuik@tudelft.nl)

†Delft University of Technology, Department of Multi Scale Physics, Prins Bernardlaan 6, 2628 BW Delft, The Netherlands (c.r.kleijn@tudelft.nl)

rate constants. The solution of such stiff sets of equations is difficult, and the numerical solvers present in most commercial CFD codes have great problems in handling such stiff systems of equations. This is especially the case when time-accurate transient solutions are required. The latter is important for the study of start-up and shut-down cycli, but also for the study of inherently transient CVD processes, such as Rapid Thermal CVD (RTCVD), see Bouteville (2005), and Atomic Layer Deposition (ALD), see Alam & Green (2003).

In this paper we focus on solving the system of species equations, which describe mass transport due to convective and diffusive transport, and their conversion due to chemical reactions, in a time accurate way. Since there may be orders of magnitude difference between the time scales of advection, diffusion and the various chemical reactions, the system of species equations is extremely stiff. To stably integrate the species equations in time, a suitable time integration method has to be found. Moreover, we demand that negative species concentrations are not allowed in the transient solution, because they cause blow up of the solution in finite time, see, for instance, Hundsdorfer & Verwer (2003). Since we do not want to apply clipping, and thus artificially add mass to the system, this extra property puts a severe restriction on time integration methods.

This report, which is a detailed description of van Veldhuizen et al. (2007a), is organized as follows. First we give details of the CVD process considered in this paper, followed by a brief overview of the numerical methods that we used to do the experiments. We conclude with some transient numerical results.

2 Model Equations

The model assumptions and equations used have been described in great detail in Kleijn (2000). The gas mixture is assumed to behave as a continuum, Newtonian fluid. The composition of the N component gas mixture is described in terms of mass fractions ω_i, $i = 1, \ldots, N$. In this paper we focus on the time accurate numerical solution of the nonlinearly, stiffly coupled set of species equations, $i = 1, \ldots, N$,

$$\frac{\partial(\rho\omega_i)}{\partial t} = -\nabla \cdot (\rho \mathbf{v} \omega_i) + \nabla \cdot [(\rho \mathbb{D}_i' \nabla \omega_i) + (\mathbb{D}_i^T \nabla(\ln T))] + m_i \sum_{k=1}^{K} \nu_{ik} R_k^g, \tag{1}$$

where the diffusive mass flux is composed of concentration and thermal diffusion.

The studied reactor is illustrated in Figure 1, where as computational domain one half of the (r-z) plane is taken. From the top a gas-mixture, consisting of 0.1 mole% silane diluted in helium, enters the reactor with a uniform temperature $T_{\text{in}} = 300$ K and velocity $u_{\text{in}} = 0.1\ \frac{\text{m}}{\text{s}}$. In the hot region above the susceptor with temperature $T_s = 1000$ K the reactive gas silane decomposes into silylene and hydrogen. In the model of Coltrin et al. (1989), which was used in this paper, this first gas phase reaction initiates a chain of 25 homogeneous gas phase reactions leading to the (de)formation of 14 silicon containing gas phase species. Each of these silicon containing species may diffuse towards the susceptor to produce a thin solid film.

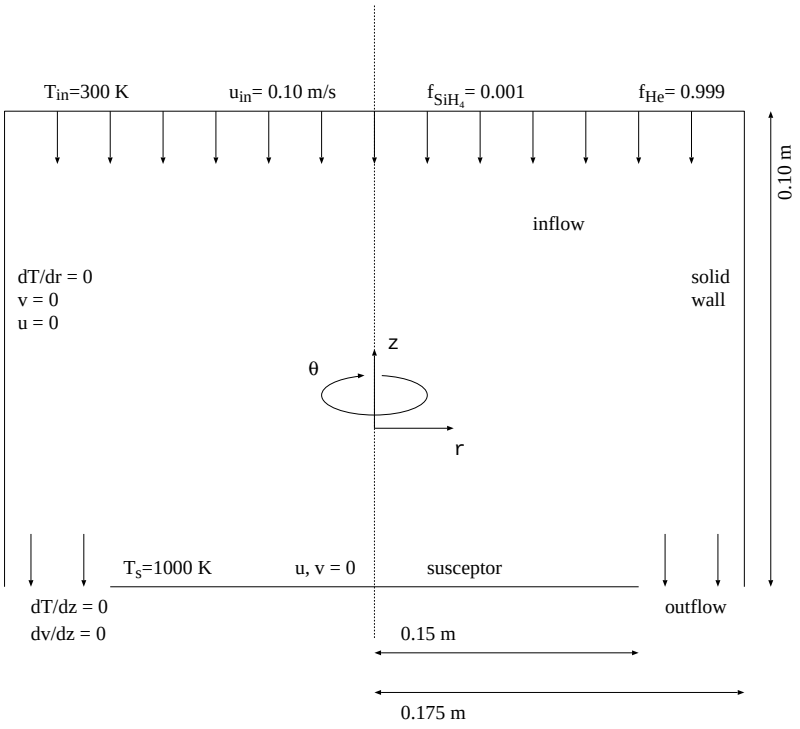

Figure 1: Reactor geometry and boundary conditions.

There is some ambiguity as to which values were used in Coltrin et al. (1989), in the present work we followed the approach used in Kleijn (2000), i.e., we set the sticking coefficient of Si_2H_5 equal to one, the sticking coefficient of Si_3H_8 equal to zero and for the other species the values as were used in Coltrin et al. (1989).

3 Numerical Methods

The species equations are first discretized in space, and thereafter integrated in time. For spatial discretization a hybrid Finite Volume (FV) scheme has been used, which uses central differences if possible and first order upwinding if necessary. More information on the hybrid FV scheme can be found in, for instance, Patankar (1980). It should be noted that the hybrid FV discretization conserves the non-negativity of the solution. In van Veldhuizen et al. (2005) all details of the hybrid FV spatial discretization have been written down.

Implicit treatment of the reaction terms, when integrating in time, is needed for stability reasons. When, in addition, also the positivity of the solution is needed, this results in an extra, severe condition on the time step size. Moreover, it has been proven in Hundsdorfer & Verwer (2003), that the first order accurate Euler Backward time integration method is the only known method being unconditionally positive (and stable). Every higher order time integration method will need impractically small time steps to integrate the solution positively. However, in this paper we test next to EB, also the second order accurate Rosenbrock scheme ROS2, the second order BDF2 scheme, and the second order IRKC scheme. To test these schemes seems to contradict with the previous remark, but each of these higher order methods have their advantages. As has been experienced in several tests, see Hundsdorfer & Verwer (2003), the ROS2 scheme performs well with respect to positivity. For the BDF2 scheme the positivity condition can be computed explicitly, and the IRKC scheme is designed to integrate convection-diffusion-reaction schemes very efficiently. More information on these time integration schemes can be found in Hundsdorfer & Verwer (2003) or van Veldhuizen et al. (2007b). For the recently developed IRKC scheme, which integrates the moderately stiff advection-diffusion part of the species equations explicitly, and the reaction part implicitly, we refer to Verwer et al. (2004). In van Veldhuizen et al. (2006) the time integration methods EB, BDF2, ROS2 and IRKC are comprehensively described, as well as relevant properties as stability, positivity, etc.

3.1 Positivity for Included Surface Chemistry

Spatial discretization of the species equations along the reacting surface can give some problems with respect to positivity. In particular, when the species surface reaction flux is computed with the cell centered species concentrations, the surface reaction flux can be too large. Consequently, in the next time step we might obtain negative concentrations.

For the type of surface reactions in the present paper, the molar reactive surface flux is linearly proportional to the species molar concentration at the wafer. Consequently, the

reactive surface mass flux is linearly proportional to the surface mass fraction, which is denoted as $F_{\text{wall}} = K_1\omega_{\text{wall}}$, with ω_{wall} the unknown species mass fraction at the wafer. Since advective transport of the species mass fraction is negligible near the wafer, we have diffusion transport only, see Kleijn (2000). At the reacting surface will therefore hold that the total transport mass flux should be equal to F_{wall}, or in discretized form

$$\frac{\mathbb{D}}{\Delta z}(\omega_{\text{center}} - \omega_{\text{wall}}) + \mathbb{D}^T\nabla(\ln T) = F_{\text{wall}}, \tag{2}$$

where

- $\mathbb{D}$ is the effective ordinary diffusion coefficient,
- $\mathbb{D}^T$ is the multi-component thermal diffusion coefficient,
- ω_{center} the species mass fraction in the cell center, and,
- ω_{wall} the unknown species mass fraction at the wafer.

From Figure (2) the meaning of ω_{center}, ω_{wall}, Δz, etc. should be clear.

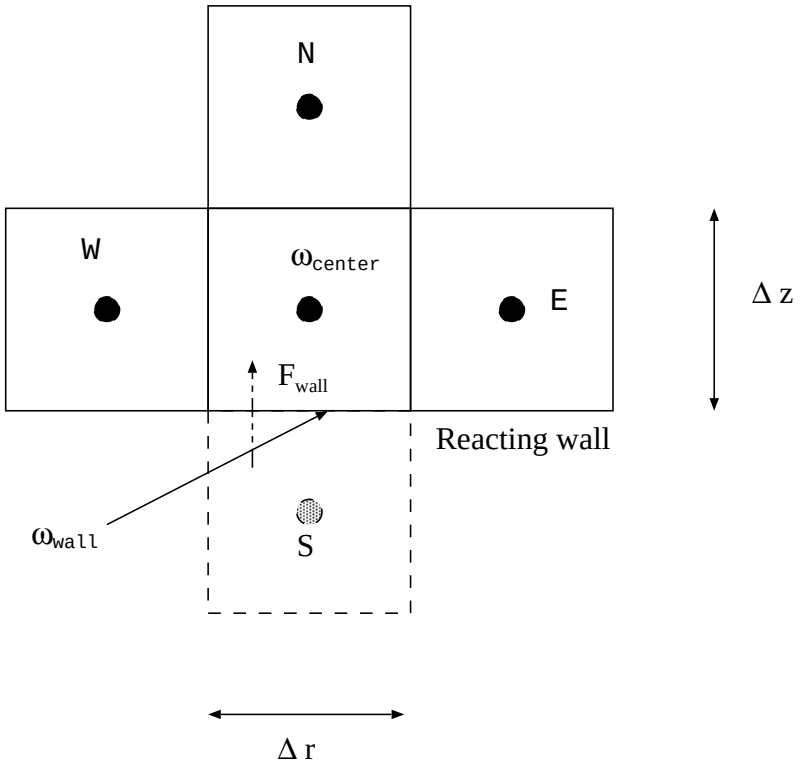

Figure 2: Grid cells near the reacting surface

The multi-component thermal diffusion coefficient $\mathbb{D}^T$ is linearly proportional with the mass fraction, see Kleijn (2000), and therefore we can write $\mathbb{D}^T = K_2\omega_{\text{wall}}$. The unknown mass fraction at the wafer can easily be derived from (2) as

$$\omega_{\text{wall}} = \frac{\omega_{\text{center}}}{1 + \frac{K_1\Delta z}{\mathbb{D}} - \frac{K_2\nabla(\ln T)\Delta z}{\mathbb{D}}} \tag{3}$$

From (3) follows easily that ω_{wall} is positive when ω_{center} is positive, and $\omega_{\text{wall}} \leq 1$ as long as $\omega_{\text{center}} \leq 1$ and $K_1 - K_2\nabla(\ln T) \geq 0$. The latter is easily satisfied along the reacting boundary because the size of Δz of the corresponding grid cell, see Figure 2, is relatively small. Therefore, $\nabla(\ln T)$ will be small in comparison with K_1 and K_2 and thus $K_1 - K_2\nabla(\ln T)$ will remain positive. Note that this is *not* a proof, but a heuristic argument.

To summarize, by replacing the diffusive mass flux by $F_{\text{wall}} = R^S\omega_{\text{wall}}$, with ω_{wall} as in (3), one obtains a positive semi-discretization near the wafer.

3.2 Nonlinear Solver in Euler Backward and BDF2: Newton's method

The nonlinear systems arising from the implicit treatment of the species equations are solved by means Newton's method, which, if necessary, uses the global convergence technique *line search*. The line search technique, or back tracking, is explained below. The algorithm as used in our code is Algorithm 1. Convergence of Algorithm 1 is declared when $\|F(x)\| > \text{TOL}$, where TOL the termination tolerance. As default the termination tolerance is given as

$$\text{TOL} = \text{TOL}_{\text{rel}}\|F(x_0)\| + \text{TOL}_{\text{abs}}, \tag{4}$$

where

- TOL_{rel} is the relative termination tolerance,
- TOL_{abs} is the absolute termination tolerance, and,
- $\|F(x_0)\|$ the norm of F evaluated in the initial guess x_0.

Global convergence of Newton's method can, for instance, be obtained by augmenting the algorithm by a sufficient decrease condition on $\|F\|$:

> Find a $\lambda \in [\lambda_{\min}, \lambda_{\max}]$ such that
>
> $$\|F(x_k + \lambda s_k)\| \leq (1 - \alpha\lambda)\|F(x_k)\|, \tag{5}$$
>
> with α a small number such that (5) is satisfied as easy as possible.

Algorithm 1: Globally Convergent Newton's method

1 Evaluate $F(x)$
2 $\text{TOL} \leftarrow \text{TOL}_{\text{rel}}\|F(x)\| + \text{TOL}_{\text{abs}}$
3 **while** $\|F(x)\| > \text{TOL}$ **do**
4 Solve $F'(x)d = -F(x)$
5 If no such d can be found, terminate with failure.
6 Put $\lambda = 1$.
7 **while** $\|F(x + \lambda d)\| > (1 - \alpha\lambda)\|F(x)\|$ **do**
8 $\lambda \leftarrow \alpha\lambda$, where $\lambda \in [1/10, 1/2]$ is computed by minimizing the polynomial model of $\|F(x + \lambda d)\|^2$.
9 $x \leftarrow x + \lambda d$

This condition provides a test for acceptability of a Newton step that is used. If (5) is satisfied for a certain $\lambda \in [\lambda_{\min}, \lambda_{\max}]$, then the Newton step is replaced by $s_k \leftarrow \lambda s_k$.

The minimum step-length reduction is one half, and in our code the maximum step-length reduction is 1/10. In the second **while**-loop in Algorithm 1 (see line 7), the step-length reductioner λ is computed by minimizing the quadratic polynomial model of

$$p(\lambda) = \|F(x_k + \lambda d)\|^2, \tag{6}$$

which is based on the last two values of λ. The while-loop in Algorithm 1 terminates when for a certain λ holds

$$\|F(x_k + \lambda s_k)\| > (1 - \alpha\lambda)\|F(x_k)\|. \tag{7}$$

3.2.1 Quadratic Polynomial Model of $\|F(x_k + \lambda d)\|^2$

In this section we provide details on the minimization of the quadratic polynomial model of

$$p(\lambda) = \|F(x_k + \lambda d)\|^2, \tag{8}$$

which is based on the last two values of λ. The quadratic polynomial model is based on the three values

- $F_0 = \|F(x_k)\|^2$,
- $F_{\text{prev}} = \|F(x_k + \lambda_{\text{prev}} d)\|^2$, where λ_{prev} is the previous step-length, and,
- $F_{\text{cur}} = \|F(x_k + \lambda_{\text{cur}} d)\|^2$, with λ_{cur} the current step-length.

The second order interpolation polynomial through F_0, F_{prev} and F_{cur} is

$$p(\lambda) = F_0 + c_1\lambda + c_2\lambda^2. \tag{9}$$

Computing the constants c_1 and c_2 in (9) gives the polynomial

$$p(\lambda) = F_0 + \frac{\tilde{c}_1\lambda + \tilde{c}_2\lambda}{\tilde{c}_3}, \tag{10}$$

with

- $\tilde{c}_1 = \lambda_{\text{cur}}^2(F_{\text{cur}} - F_0) - \lambda_{\text{prev}}^2(F_{\text{prev}} - F_0)$,
- $\tilde{c}_2 = \lambda_{\text{prev}}(F_{\text{cur}} - F_0) - \lambda_{\text{cur}}(F_{\text{prev}} - F_0)$,
- $\tilde{c}_3 = (\lambda_{\text{cur}} - \lambda_{\text{prev}})\lambda_{\text{cur}}\lambda_{\text{prev}}$.

Note that $\tilde{c}_3 < 0$. The next step-length λ_{new} is the minimizer of (10), i.e.,

$$\lambda_{\text{new}} = -\frac{\tilde{c}_1}{2\tilde{c}_2}. \tag{11}$$

In the case that

- $\tilde{c}_2 \geq 0$, we have negative curvature, and thus $\lambda_{\text{new}} \leftarrow 1/2\ \lambda_{\text{cur}}$,
- $\lambda_{\text{new}} < 1/10\ \lambda_{\text{cur}}$, then $\lambda_{\text{new}} \leftarrow 1/10\ \lambda_{\text{cur}}$, and,
- $\lambda_{\text{new}} > 1/2\ \lambda_{\text{cur}}$, then $\lambda_{\text{new}} \leftarrow 1/2\ \lambda_{\text{cur}}$.

3.3 Direct Linear Solver

At the deepest level of each time integration method considered in this paper, at least one linear system has to be solved per time step. This section is subdivided into two parts, where in the first part the linear solvers that appear in Newton's method in the case of EB and BDF2 time integration are described. The linear systems in the ROS2 time integration method are solved in the same way as the linear systems appearing in Newton's method for EB and BDF2. In the second part of this section we discuss the linear systems as they appear in the IRKC solver.

3.3.1 Linear Solver in EB and BDF2

When considering the semi-discretization $w' = F(t, w)$, with w containing all species in all grid points and $F(t, w)$ the spatially discretized advection, diffusion and reaction terms, the resulting linear system is of the following form

$$Ax = b, \tag{12}$$

with

- $A = I - \tau F'(t, w)$, with $F'(t, w)$ the Jacobian of F, and,

- $b = -F(t, w)$.

Since in this paper only two dimensional simulations are considered, we use a direct solver. Because, in particular, the reaction part of F' is not-symmetric, an LU-factorization of A will be used. The amount of work of computing an LU-factorization of A depends highly on the bandwidth of A. Therefore, the ordening of unknowns in w, and thus implicitly the ordening of equations, is important, since it determines the bandwidth of A.

On a rectangular, non-equidistant, structured grid, as presented in Figure 6, with nr grid cells in the r direction, and nz grid cells in the z direction, we call the number of grid points $n = nr \cdot nz$. Then, it follows that the total number of unknowns is $totn = s \cdot n$, with s the number of species.

The most obvious ordening of unknowns would be the 'natural ordening', which means per species a sequential numbering of the grid lines and the points within each grid line. The resulting nonzero pattern of A is presented in Figure 3. It can easily be seen that the bandwidth of A is then $n(s-1)$.

The most 'optimal' ordening in terms of minimal bandwidth is, however, given by: per grid point a sequential numbering of all species, and then walk through by sequentially numbering all grid points per line. The resulting bandwidth is $s \cdot nr$, and the corresponding nonzero pattern can be found in Figure 4.

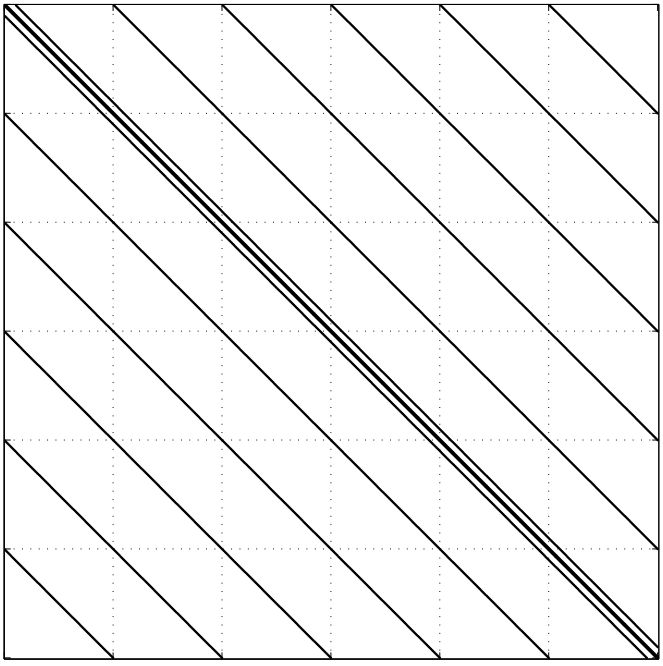

Figure 3: Nonzero pattern of the Jacobian matrix with a natural ordening of the unknowns. In this case the number of species $s = 6$.

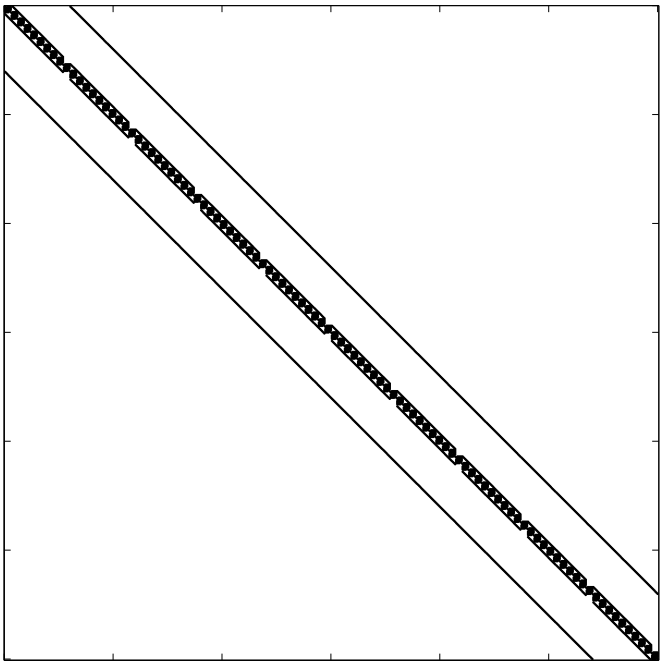

Figure 4: Nonzero pattern of the Jacobian matrix with an optimal reordening.

3.3.2 Linear systems appearing in IRKC

The efficiency of the IRKC time integrator highly depends on the ordening of the unknowns. The only way to make this scheme efficient, and that is then also the way it should be used, is to arrange the unknowns and equations as described above as being the most 'optimal' ordening. Since the advection and diffusion part(s) of the species equations are integrated explicitly, the two upper and two lower diagonals in Figure 4 will drop out. Then remains a matrix A with bandwidth s, i.e., s the number of species. If one looks more carefully to the structure of A, then one observes that there are actually small, independent, linear systems per grid point, which have to be inverted, see also Figure 5 The cheapest way to do this is to build an LU factorization of these small subsystems per grid point, which is a cheap operation. For the sake of clarity, these small systems are *not* sparse.

3.3.3 Rounding Errors in the Linear Solver

A very well known result on the error analysis of Gaussian elimination is the following. Suppose that $\hat{L}$ and $\hat{U}$ are the computed factors of the LU factorization of A, and let $\hat{y}$ be the computed solution of $\hat{L}\hat{y} = b$, and $\hat{x}$ be the computed solution of $\hat{U}\hat{x} = \hat{y}$. Then the computed solution $\hat{x}$ and the exact solution x of $Ax = b$ can are related in the following

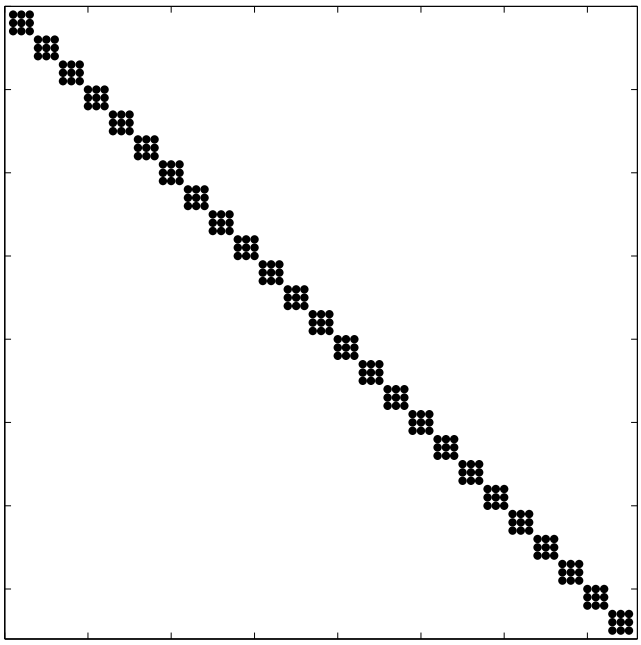

Figure 5: Nonzero pattern of the Jacobian matrix arising in the IRKC method. The current Jacobian is constructed for the case with 5 grid points in both radial and axial directions, and 3 species.

way

$$\frac{\|x - \hat{x}\|}{\|x\|} \leq C\kappa(A), \tag{13}$$

where $C \in \mathbb{R}$ is the machine precision.

To use this error analysis in practice it is needed to estimate the condition number of A. We follow the algorithm proposed in Higham (1988), which gives a reliable estimation of the order of magnitude of the condition number of A.

The condition number of A is defined as

$$\kappa(A) = \|A\|\|A^{-1}\|. \tag{14}$$

The L_1-norms of A and A^{-1} are respectively

$$\|A\| = \max_x \frac{\|Ax\|}{\|x\|}, \quad \text{and} \tag{15}$$

$$\|A^{-1}\| = \max_z \frac{\|z\|}{\|Az\|}. \tag{16}$$

The L_1-norm of a square matrix A of dimension m can be computed easily as

$$\|A\| = \max_{1 \leq k \leq m} \sum_{j=1}^{m} |A_{ij}|. \tag{17}$$

Basically, the condition estimator's task is to obtain a good approximation for $\|A^{-1}\|$. The algorithm to do this is described in Higham (1988) and presented below as Algorithm 2. In Higham (1988) can be found that it terminates in at most $m+1$ iterations. If A is an M-matrix, then Algorithm 2 terminates after at most 2 iterations and $\|A^{-1}\|$ is computed exactly.

Algorithm 2: Estimation γ of $\kappa(A^{-1})$

```
Choose x such that ||x||_1 = 1
repeat
    Solve Ay = x
    ζ = sign(y) (componentwise)
    Solve A^T z = ζ
    if ||z||_∞ ≤ z^T x then
        quit with γ = ||y||_1
    x = e_j, where |z_j| = ||z||_∞
until finished ;
```

4 Results

Since the reactants are highly diluted in the carrier gas helium, we use the steady state velocity fields, temperature field, pressure field and density field computed by Kleijn (2000). For such systems, the computation of the laminar flow and temperature fields etc., is, in comparison with computation of the species mass fractions, a relatively trivial task. All simulations are done on a spatial grid with $nr = 35$ equidistant grid points in radial direction with $\Delta r = 5 \cdot 10^{-3}$ m, and $nz = 45$ non-equidistant grid points in axial direction. The axial distance from the wafer to the first grid point is $1 \cdot 10^{-5}$ m, with the grid spacing gradually increasing to $5 \cdot 10^{-3}$ m for $z > 0.04$ m. The computational grid is presented in Figure 6, and the steady state streamlines and temperature field in Figure 7.

The simulations start from the instant that the reactor is completely filled with the carrier gas helium and a mixture of helium and silane starts to enter the reactor, and stop when steady state is obtained. Correctness of our solution is then validated against the steady state solution obtained with the software of Kleijn (2000). All simulations presented in this paper are test cases where the wafer is *not* rotating.

In Figure 9 up to and including 13 we present transient deposition rates for simulations with wafer temperatures varying from 900 K up to 1100 K. The time dependent behavior of all deposition rates is monotonically increasing until the species concentrations are in steady state. Note that the relative contributions of the various silicon containing species to the total deposition rate is a function of the wafer temperatures, with the relative contribution of Si_2H_2 increasing with increasing temperature, and the relative contribution

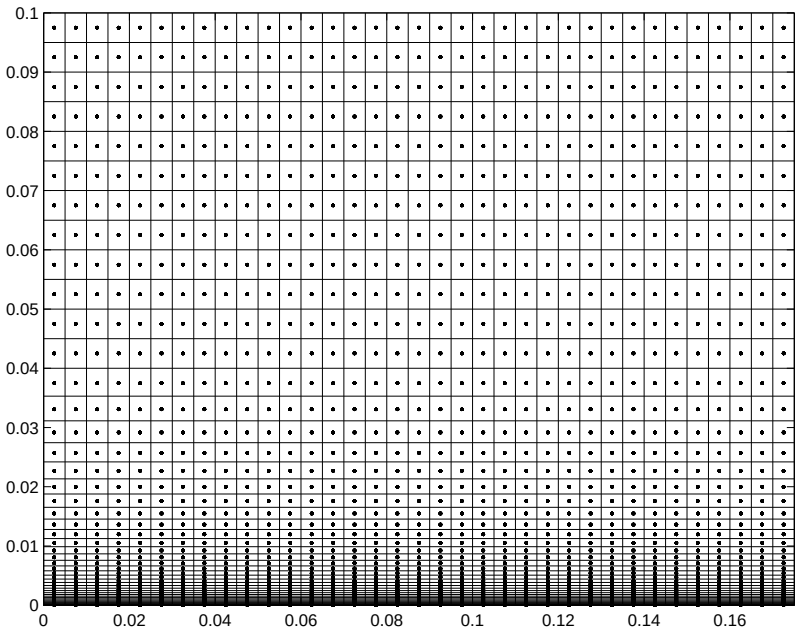

Figure 6: The computational grid.

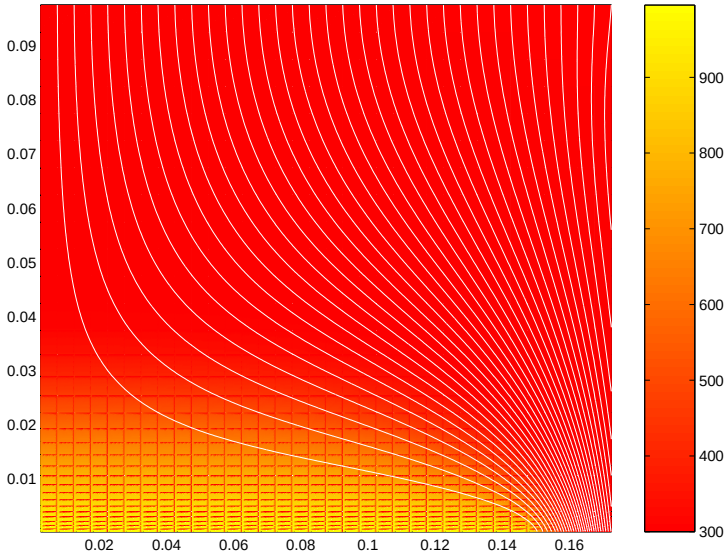

Figure 7: Steady state streamlines and temperature field.

of H_2SiSiH_2 decreasing with increasing temperature. In Table 1 the total steady state deposition rates for wafer temperatures from 900 K up to 1100 K are given.

	900 K	950 K	1000 K	1050 K	1100 K
via long term time integration	0.65	1.44	1.93	2.08	2.15
Kleijn's steady state computations	0.603	1.42	1.88	2.14	2.21

Table 1: Total steady state deposition rates $\left(\frac{\text{nm}}{\text{s}}\right)$ for wafer temperatures varying from 900 K up to and including 1100 K.

Figure 15 shows radial profiles of the total steady state deposition rates of both Kleijn's steady state computations, see Kleijn (2000), and our steady state results obtained with the time integration methods as discussed in Section 3, for wafer temperatures varied from 900 K up to and including 1100 K. Again, the agreement is for all wafer temperatures excellent. For all studied temperatures, the steady state growth rates obtained with the present transient solution method were found to differ less than 5% from those obtained with Kleijn's steady state code.

The integration statistics of the various time integration methods mentioned in Section 3 are presented in Table 2. Based upon these experiments we conclude that the unconditionally stable and positive time integration method Euler Backward is the cheapest in terms of computational costs. However, the second order ROS2 scheme performs also quite well, although it is not unconditionally positive for the species equations, see van Veldhuizen et al. (2007b). When the convection part is omitted, then the ROS2 scheme becomes unconditionally positive. This property explains probably the good behavior with respect to positivity for the convection-diffusion-reaction case. The performance of the IRKC scheme is between BDF-2 and ROS2. However, when going from 2 to 3 spatial dimensions, we expect that the IRKC scheme performs much better in comparison with EB and ROS2. This is due to the fact that the linear systems to be solved in IRKC remain of the same dimension, but only more of them have to be solved. This dimension corresponds to the number of species in the mixture, see also Section 3.3.2.

For the other schemes like EB, ROS2 and BDF2 holds that when going from 2 to 3 spatial dimensions, the direct linear solver as presented in this paper is no longer feasible. Therefore, one has to apply iterative linear solvers, like for instance Krylov Subspace methods. The success of a Krylov Subspace method largely depends on an effective preconditioner, which efficiently clusters the eigenvalues of the iteration matrix, resulting in speed-up of the Krylov method. To find such an effective preconditioner is a challenging task for future research in this field.

5 Conclusions

In this paper we presented two dimensional transient simulations of a Chemical Vapor Deposition problem taken from Kleijn (2000). The solutions presented in Kleijn (2000),

Number of	EB	BDF-2	ROS2	IRKC
F	190	757	424	427911
F'	94	417	142	2008
Linesearch	11	0	0	30
Newton iters	94	417	0	17331
Rej. time steps	1	10	2	728
Acc. time steps	38	138	140	1284
CPU Time	6500	30500	8000	19500

Table 2: Integration statistics for EB, BDF-2, ROS2 and IRKC, with full Newton solver

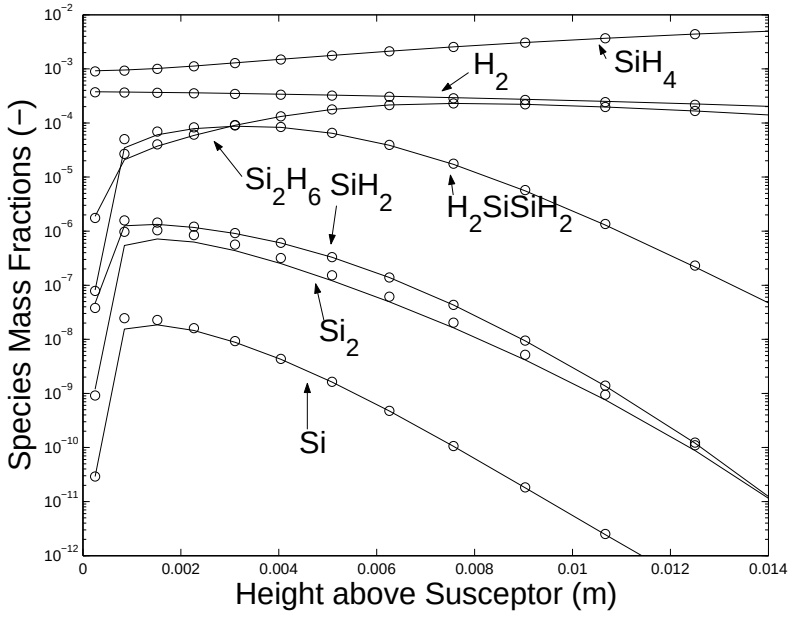

Figure 8: Axial steady state concentration profiles along the symmetry axis for some selected species. Solid lines are solutions from Kleijn (2000), circles are long time steady state results obtained with the present transient time integration method.

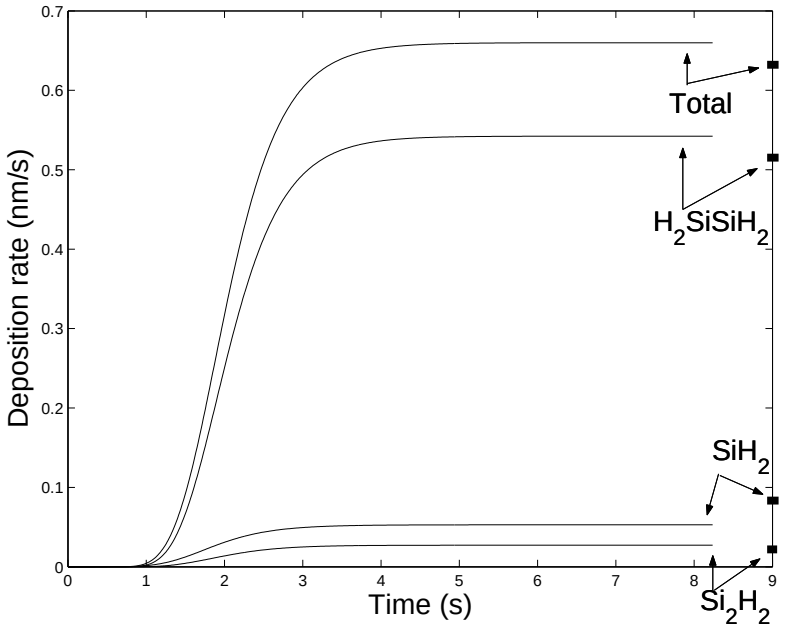

Figure 9: Transient deposition rates due to some selected species on the symmetry axis for simulations with a non-rotating wafer at 900 K. On the right vertical axis: steady state deposition rates obtained with the steady state code from Kleijn (2000).

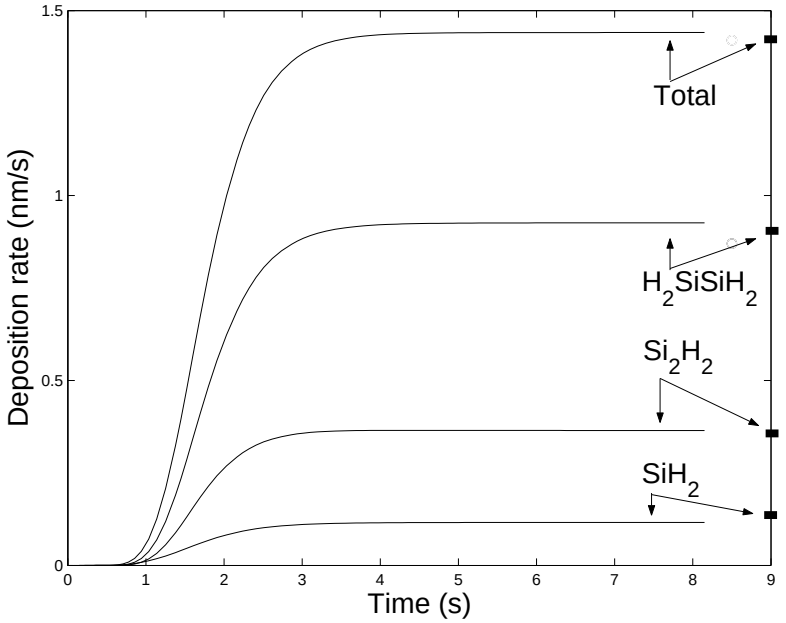

Figure 10: Transient deposition rates due to some selected species on the symmetry axis for simulations with a non-rotating wafer at 950 K. On the right vertical axis: steady state deposition rates obtained with the steady state code from Kleijn (2000).

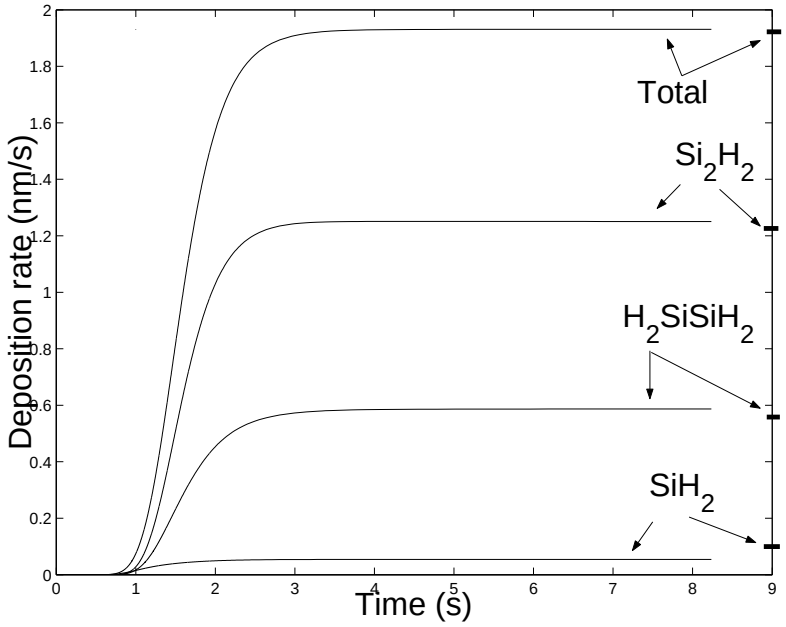

Figure 11: Transient deposition rates due to some selected species on the symmetry axis for simulations with a non-rotating wafer at 1000 K. On the right vertical axis: steady state deposition rates obtained with the steady state code from Kleijn (2000).

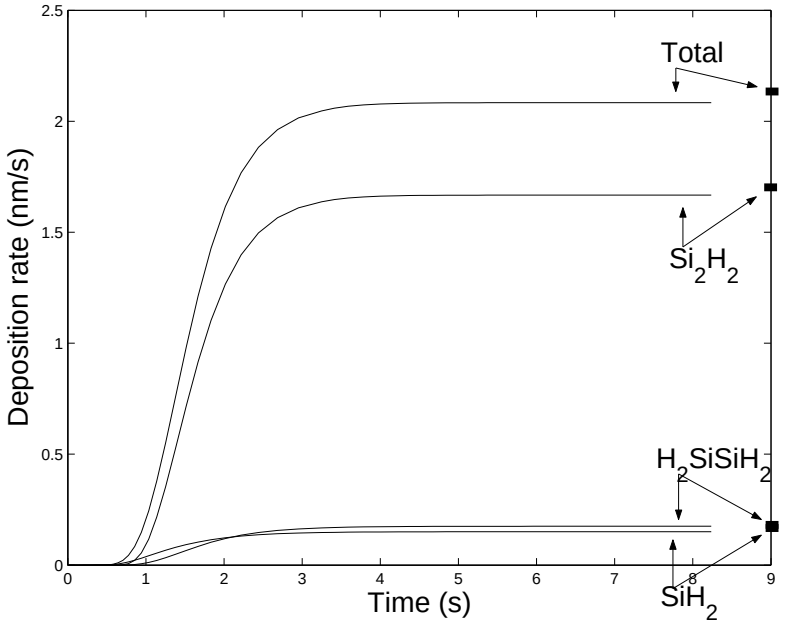

Figure 12: Transient deposition rates due to some selected species on the symmetry axis for simulations with a non-rotating wafer at 1050 K. On the right vertical axis: steady state deposition rates obtained with the steady state code from Kleijn (2000).

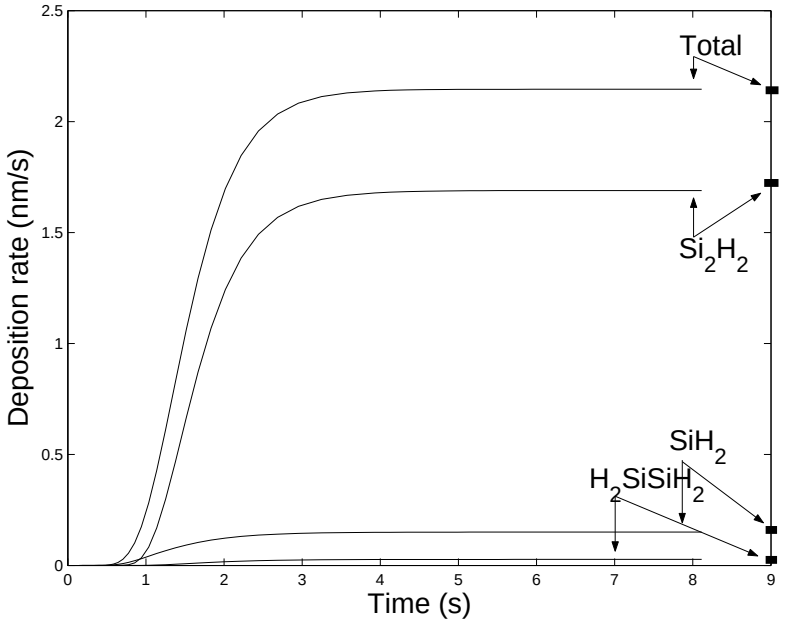

Figure 13: Transient deposition rates due to some selected species on the symmetry axis for simulations with a non-rotating wafer at 1100 K. On the right vertical axis: steady state deposition rates obtained with the steady state code from Kleijn (2000).

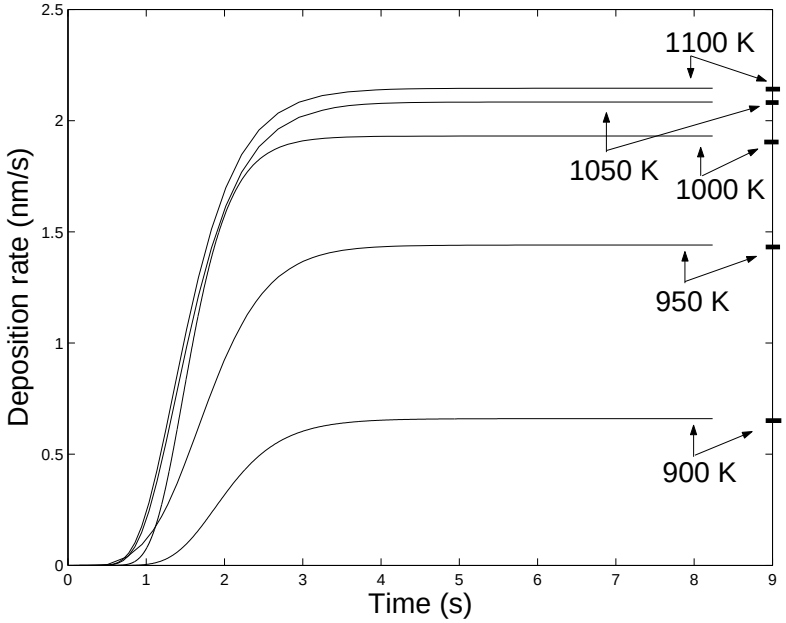

Figure 14: Transient deposition rates for some selected species on the symmetry axis for wafer temperatures varying from 900 K up to 1100 K. On the right vertical axis: steady state deposition rates obtained with the steady state code from Kleijn (2000).

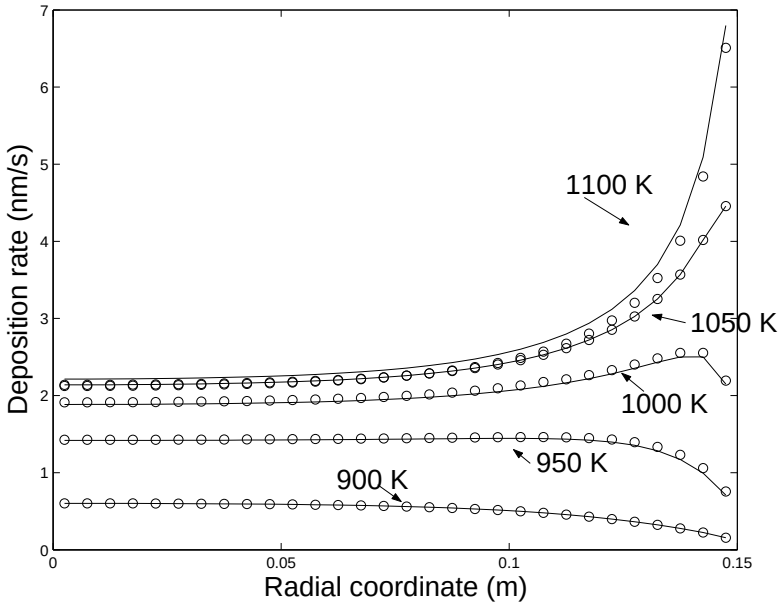

Figure 15: Radial profiles of the total steady state deposition rate for wafer temperatures varied from 900 K up to 1100 K. Solid lines are steady state results from Kleijn (2000), circles are long time steady state results obtained with the present transient time integration method.

however, were steady state. Correctness of our code is validated by comparing our long term time integration steady state solutions with the steady state solutions from Kleijn (2000). The steady state growth rates obtained with our code(s) were found to differ less than 5% from those obtained in Kleijn (2000).

Another topic considered in this paper is the efficiency, in terms of total computational costs, of the time integration method used. The time integration methods considered in this paper, are selected on stability issues and positivity properties. In terms of computational costs the Euler Backward scheme is the best choice. In spite of its conditional positivity, the ROS2 scheme performed quite well in comparison with the other higher order integration methods. However, for time accurate simulations on 3D geometries, we expect that the IRKC scheme will perform better, because the dimension of the linear systems appearing in this method remain the same. The other time integration methods have to switch to iterative linear solvers, where appropriate preconditioners have to be developed. For problems from chemistry, like the one in this paper, this is still a challenging task for future research.

References

Alam, M. & Green, M. (2003). Mathematical description of atomic layer deposition and its application to the nucleation and growth of HfO2 gate dielectric layers. *J. Appl. Phys*, *94*, pp. 3403–3413.

Bouteville, A. (2005). Numerical simulation applied to Chemical Vapour Deposition process. Rapid Thermal CVD and Spray CVD. *J. Optoelectronics and Advanced Materials*, *7*, pp. 599–606.

Coltrin, M., Kee, R., & Evans, G. (1989). A mathematical model of the fluid mechanics and gas-phase chemistry in a rotating Chemical Vapor Deposition reactor. *J. Electrochem. Soc*, *136*, pp. 819–829.

Higham, N. (1988). FORTRAN codes for estimating the one-norm of a real or complex matrix, with applications to condition estimation. *ACM Trans. on Math. Software*, *14*, pp. 381–396.

Hundsdorfer, W. & Verwer, J. (2003). *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*. Number 33 in Springer Series in Computational Mathematics. Berlin: Springer.

Kleijn, C. (2000). Computational modeling of transport phenomena and detailed chemistry in Chemical Vapor Deposition- A benchmark solution. *Thin Solid Films*, *365*, pp. 294–306.

Patankar, S. (1980). *Numerical Heat Transfer and Fluid Flow*. Hemisphere Publishing Corp.

van Veldhuizen, S., Vuik, C., & Kleijn, C. (2005). Efficient solution methods for stiff systems of advection-diffusion-reaction equations, literature study. Technical Report 05-05, Delft University of Technology, Delft Institute of Applied Mathematics, Delft.

van Veldhuizen, S., Vuik, C., & Kleijn, C. (2006). Numerical methods for CVD simulation. Report 06-07, Delft University of Technology, Delft Institute of Applied Mathematics, Delft.

van Veldhuizen, S., Vuik, C., & Kleijn, C. (2007a). Comparison of numerical methods for transient CVD simulations. *Surface and Coatings Technology*. doi: 10.1016/j.surfcoat.2007.04.022.

van Veldhuizen, S., Vuik, C., & Kleijn, C. (2007b). Comparison of ode methods for for laminar reacting gas flow simulations. *Num. Meth. Part. Diff. Eq.* submitted.

Verwer, J., Sommeijer, B., & Hundsdorfer, W. (2004). RKC time-stepping for advection-diffusion-reaction problems. *J. of Comp. Physics*, *201*, pp. 61–79.