

## NUMERICAL SOLUTION AND FAST-SLOW DECOMPOSITION OF A POPULATION OF WEAKLY COUPLED SYSTEMS

ALEXANDRE CABOUSSAT

University of Houston, Department of Mathematics  
4800 Calhoun Rd, Houston, TX 77204-3008, USA

ALLISON LEONARD

University of Houston, Department of Mathematics  
4800 Calhoun Rd, Houston, TX 77204-3008, USA

**ABSTRACT.** The modeling of the microphysics of a population of atmospheric particles interacting through a common medium leads to the solution of a large system of weakly coupled differential-algebraic equations. An implicit time discretization of the system of differential-algebraic equations is solved with a Newton method at each time step. The structure of the global system and the sparsity of the Newton matrix allow the efficient use of a Schur complement approach for the decoupling of the various subsystems at the discrete level. A numerical approach for the decomposition of the population into fast and slow subsystems is proposed. Numerical results are presented for organic atmospheric particles to illustrate the properties of the method.

**1. Introduction.** The modeling of the dynamics of aerosol particles is a crucial step in the simulation of atmospheric processes [12, 16, 17]. The computation of the microphysics of a population of aerosol particles is an actual challenge, since particles of different sizes are interacting through a common medium (the gas phase), and directly by coagulation.

A numerical method for the resolution of large population of subsystems is presented here. The interactions between subsystems are assumed to hold through a common medium (or agent), and direct interactions are neglected. The model leads to a large system of differential equations, constrained by the global minimization of the internal energy of each subsystem [1, 2, 3]. The corresponding chemical model equations have been presented in [4].

Due to the wide range of particles' sizes and the diversity of the chemical components involved, the resulting system of differential equations is stiff [15]. Moreover large time steps are usually required to cover the large time intervals of simulation. A mathematical and numerical framework is proposed here. An implicit first order time discretization of the differential-algebraic system is considered. The corresponding system of nonlinear algebraic equations is solved with a Newton method. Numerical linear algebra techniques based on the Schur complement are very efficient for the resolution of the Newton system, when considering a large number of subsystems of different sizes [7, 8].

---

2000 *Mathematics Subject Classification.* Primary: 65L05, 65L80, 65F05; Secondary: 70K70.

*Key words and phrases.* Differential-algebraic equations, weakly coupling, Schur complement, fast/slow decomposition, air quality modeling.

The reaction speed and the time to reach a stationary solution vary from one subsystem to the other. A decomposition technique is advocated to decouple fast and slow subsystems, for both theoretical and numerical purposes. Numerical results exhibit a continuum of speeds among the various particles that depend mainly on the size of the particles. A decomposition of the subsystems into fast and slow entities [6, 13] is therefore proposed based on the behavior of the residuals and sizes for each subsystem. This captures the fast and slow scales without restriction on the time stepping procedure [5]. Numerical experiments in a simple, two particles, case, show that the fast-slow decomposition allows the computational cost of the algorithm to decrease, without sacrificing the accuracy.

**2. Model problem.** Let  $N > 0$  be a given integer denoting the number of subsystems. A population of  $N$  spherical particles, each with corresponding radius  $r_i$ ,  $i = 1, \dots, N$  is considered, as illustrated in Figure 1. The particles do not interact directly with each other, but each particle interacts with the common medium (or agent). Let us denote by  $\mathbf{b}_0$ , resp.  $\mathbf{b}_i$ , the chemical concentration vector in the common medium, resp. in the subsystem number  $i$ .

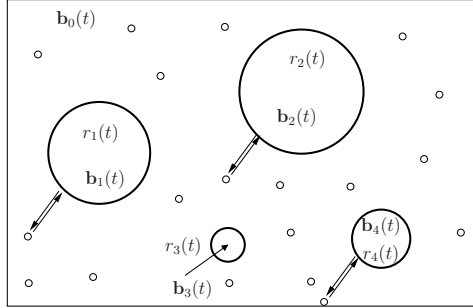


FIGURE 1. Population of  $N$  subsystems in a common medium ( $N = 4$ ).

The model problem reads as follows: find  $\mathbf{b}_i : (0, T) \rightarrow \mathbb{R}^n$ ,  $i = 0, \dots, N$ ,  $\mathbf{z}_i : (0, T) \rightarrow \mathbb{R}^m$  and  $r_i : (0, T) \rightarrow \mathbb{R}$ ,  $i = 1, \dots, N$  satisfying:

$$\begin{aligned} \frac{d}{dt} \mathbf{b}_0(t) &= - \sum_{i=1}^N f(t, \mathbf{b}_i(t), \mathbf{b}_0(t), \mathbf{z}_i(t), r_i), \quad \mathbf{b}_0(0) = \mathbf{b}_{0,0}, \\ \frac{d}{dt} \mathbf{b}_i(t) &= f(t, \mathbf{b}_i(t), \mathbf{b}_0(t), \mathbf{z}_i(t), r_i), \quad \mathbf{b}_i(0) = \mathbf{b}_{i,0}, \quad i = 1, \dots, N, \\ \mathbf{z}_i(t) &= \arg \min_{\mathbf{z}^*} g(\mathbf{z}^*) \quad \text{s. t. } A\mathbf{z}^* = \mathbf{b}_i(t), \quad \mathbf{z}^* \geq 0, \quad i = 1, \dots, N, \end{aligned} \quad (1)$$

where  $g$  is a non-convex, smooth internal energy function,  $f$  is a given smooth flux function, and  $A \in \mathbb{R}^{n \times m}$  is a matrix describing the chemical reactions and phase separations inside each subsystem; the variable  $\mathbf{z}_i(t)$  is the global minimum of the optimization problem at time  $t \in (0, T)$  corresponding to the right-hand side data  $\mathbf{b}_i(t)$  in the constraint. At each time  $t \in (0, T)$ , the radius  $r_i$  of the particle is directly computed from the variables  $\mathbf{b}_i$  as  $r_i(t) = h(\mathbf{b}_i(t))$ ,  $i = 1, \dots, N$ , where  $h : \mathbb{R}^n \rightarrow \mathbb{R}$  is a given mapping that depends on the physical properties of the

subsystem [4]. The parameters  $r_i$  in (1) are those governing the reaction speed of each subsystem.

By replacing the global minimization problem in (1) by its first order optimality conditions (describing the stationary points of the optimization problem), a system of nonlinear differential-algebraic equations of index one [10] is obtained, and reads: find  $\mathbf{b}_i : (0, T) \rightarrow \mathbb{R}^n$ ,  $i = 0, \dots, N$ ,  $\mathbf{z}_i : (0, T) \rightarrow \mathbb{R}^m$ ,  $\boldsymbol{\lambda}_i : (0, T) \rightarrow \mathbb{R}^n$ ,  $\boldsymbol{\theta}_i : (0, T) \rightarrow \mathbb{R}^m$  and  $r_i : (0, T) \rightarrow \mathbb{R}$ ,  $i = 1, \dots, N$  satisfying:

$$\begin{aligned} \frac{d}{dt}\mathbf{b}_0(t) &= -\sum_{i=1}^N f(t, \mathbf{b}_i(t), \mathbf{b}_0(t), \mathbf{z}_i(t), r_i), \quad \mathbf{b}_0(0) = \mathbf{b}_{0,0}, \\ \frac{d}{dt}\mathbf{b}_i(t) &= f(t, \mathbf{b}_i(t), \mathbf{b}_0(t), \mathbf{z}_i(t), r_i), \quad \mathbf{b}_i(0) = \mathbf{b}_{i,0}, \quad i = 1, \dots, N, \\ G(\mathbf{z}_i(t), \mathbf{b}_i(t), \boldsymbol{\lambda}_i(t), \boldsymbol{\theta}_i(t)) &= 0, \quad i = 1, \dots, N, \\ \boldsymbol{\theta}_i(t)\mathbf{z}_i(t) &= 0, \quad \boldsymbol{\theta}_i(t) \geq 0, \quad \mathbf{z}_i(t) \geq 0, \quad i = 1, \dots, N, \end{aligned} \tag{2}$$

where  $\boldsymbol{\lambda}_i(t)$  is the Lagrange multiplier related to the equality constraint  $A\mathbf{z}_i(t) = \mathbf{b}_i(t)$ ,  $\boldsymbol{\theta}_i(t)$  is the Kuhn-Tucker multiplier related to the inequality constraint  $\mathbf{z}_i(t) \geq 0$ , and  $G : \mathbb{R}^{m+n+n+m} \rightarrow \mathbb{R}^{m+n}$  is defined as:

$$G(\mathbf{z}, \mathbf{b}, \boldsymbol{\lambda}, \boldsymbol{\theta}) = \begin{pmatrix} \nabla g(\mathbf{z}) + A^T \boldsymbol{\lambda} - \boldsymbol{\theta} \\ A\mathbf{z} - \mathbf{b} \end{pmatrix}.$$

The set of stationary points, *i.e.* the set of solutions to  $G(\mathbf{z}, \mathbf{b}, \boldsymbol{\lambda}, \boldsymbol{\theta}) = 0$ , includes not only the global minimum of the original non-convex optimization problem in (1), but also any local minima, maxima or saddle-points, implying that the solution of the algebraic part of (2) is not necessarily unique, and a bifurcation phenomenon may exist when  $\mathbf{z}_i(t) = 0$  [2, 3].

An interior-point method for the relaxation of the inequality constraints in (2) has been described in [1]. Tracking techniques for the activation/deactivation of inequality constraints are reminiscent of the detection of discontinuities for ordinary differential equations or delay differential equations [9, 10, 14]. In order to focus on the behavior of a large population of subsystems, the inequality constraints appearing in the last equation of (2) are neglected in the present article; the unknowns  $\boldsymbol{\theta}_i$  are therefore also neglected and  $G : \mathbb{R}^{m+n+n} \rightarrow \mathbb{R}^{m+n}$  is redefined as:

$$G(\mathbf{z}, \mathbf{b}, \boldsymbol{\lambda}) = \begin{pmatrix} \nabla g(\mathbf{z}) + A^T \boldsymbol{\lambda} \\ A\mathbf{z} - \mathbf{b} \end{pmatrix}. \tag{3}$$

The numerical solution of (2)(3) is addressed for large values of  $N$ , and the repartition of subsystems into fast and slow entities is discussed.

**3. Numerical method and numerical linear algebra.** A first order implicit time scheme is advocated for the discretization of (2)(3). Consider a time step  $h > 0$ ,  $t^n = nh$ , and  $\mathbf{z}_i^{n+1}$ ,  $\mathbf{b}_i^{n+1}$ ,  $\boldsymbol{\lambda}_i^{n+1}$ ,  $r_i^{n+1}$  approximations of  $\mathbf{z}_i(t^{n+1})$ ,  $\mathbf{b}_i(t^{n+1})$ ,  $\boldsymbol{\lambda}_i(t^{n+1})$ ,  $r_i(t^{n+1})$  respectively. An implicit first order discretization of (2)(3) reads, for all  $n \geq 0$ :

$$\begin{aligned}
\frac{\mathbf{b}_0^{n+1} - \mathbf{b}_0^n}{h} &= - \sum_{i=1}^N f(t^{n+1}, \mathbf{b}_i^{n+1}, \mathbf{b}_0^{n+1}, \mathbf{z}_i^{n+1}, r_i^n), \quad \mathbf{b}_0^0 = \mathbf{b}_{0,0}, \\
\frac{\mathbf{b}_i^{n+1} - \mathbf{b}_i^n}{h} &= f(t^{n+1}, \mathbf{b}_i^{n+1}, \mathbf{b}_0^{n+1}, \mathbf{z}_i^{n+1}, r_i^n), \quad \mathbf{b}_i^0 = \mathbf{b}_{i,0}, \quad i = 1, \dots, N, \\
G(\mathbf{z}_i^{n+1}, \mathbf{b}_i^{n+1}, \boldsymbol{\lambda}_i^{n+1}) &= 0, \quad i = 1, \dots, N.
\end{aligned} \tag{4}$$

The discretization of (2)(3) can be achieved with more accurate methods than the first order implicit Euler scheme. For instance, in the case of one single subsystem, a Crank-Nicolson scheme has been used in [3], and a numerical method based on the RADAU5 method [11] has been presented in [14]. We plan to extend the latter to the case of a population in the future.

The solution of the (large) system (4) is obtained with a Newton method. Let us define  $\mathbf{x}_i^{(k)} := (\mathbf{z}_i^{(k)}, \boldsymbol{\lambda}_i^{(k)})$  and  $\mathbf{b}_i^{(k)}$  the Newton iterates for  $k = 0, 1, 2, \dots$ . Starting with an initial guess  $\mathbf{b}_0^{(0)} := \mathbf{b}_0^n$ ,  $(\mathbf{b}_i^{(0)}, \mathbf{x}_i^{(0)}) := (\mathbf{b}_i^n, \mathbf{x}_i^n)$ ,  $i = 1, \dots, N$ , the increments are computed as the solutions of the following Newton system:

$$\begin{pmatrix} H_1 & B_1 & 0 & 0 & \dots & \dots & C_1 \\ E_1 & O_1 & 0 & 0 & \dots & \dots & 0 \\ 0 & 0 & H_2 & B_2 & \dots & \dots & C_2 \\ 0 & 0 & E_2 & O_2 & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & & \vdots \\ \vdots & \vdots & \vdots & \vdots & & \ddots & \vdots \\ F_1 & D_1 & F_2 & D_2 & \dots & \dots & H_0 \end{pmatrix} \begin{pmatrix} p_{\mathbf{b}_1} \\ p_{\mathbf{x}_1} \\ p_{\mathbf{b}_2} \\ p_{\mathbf{x}_2} \\ \vdots \\ \vdots \\ p_{\mathbf{b}_0} \end{pmatrix} = \begin{pmatrix} r_{\mathbf{b}_1} \\ r_{\mathbf{x}_1} \\ r_{\mathbf{b}_2} \\ r_{\mathbf{x}_2} \\ \vdots \\ \vdots \\ r_{\mathbf{b}_0} \end{pmatrix}, \tag{5}$$

where the matrices in (5) are defined by

$$\begin{aligned}
H_i &= h^{-1} \mathbf{I} - \nabla_{\mathbf{b}_i} f(t^{n+1}, \mathbf{b}_i^{(k)}, \mathbf{b}_0^{(k)}, \mathbf{z}_i^{(k)}, r_i^n), \\
B_i &= -\nabla_{\mathbf{x}_i} f(t^{n+1}, \mathbf{b}_i^{(k)}, \mathbf{b}_0^{(k)}, \mathbf{z}_i^{(k)}, r_i^n), \\
C_i &= -\nabla_{\mathbf{b}_0} f(t^{n+1}, \mathbf{b}_i^{(k)}, \mathbf{b}_0^{(k)}, \mathbf{z}_i^{(k)}, r_i^n), \quad E_i = \nabla_{\mathbf{b}_i} G(\mathbf{z}_i^{(k)}, \mathbf{b}_i^{(k)}, \boldsymbol{\lambda}_i^{(k)}), \\
O_i &= \nabla_{\mathbf{x}_i} G(\mathbf{z}_i^{(k)}, \mathbf{b}_i^{(k)}, \boldsymbol{\lambda}_i^{(k)}), \quad D_i = \nabla_{\mathbf{x}_i} f(t^{n+1}, \mathbf{b}_i^{(k)}, \mathbf{b}_0^{(k)}, \mathbf{z}_i^{(k)}, r_i^n), \\
F_i &= \nabla_{\mathbf{b}_i} f(t^{n+1}, \mathbf{b}_i^{(k)}, \mathbf{b}_0^{(k)}, \mathbf{z}_i^{(k)}, r_i^n), \\
H_0 &= h^{-1} \mathbf{I} + \sum_{i=1}^N \nabla_{\mathbf{b}_0} f(t^{n+1}, \mathbf{b}_i^{(k)}, \mathbf{b}_0^{(k)}, \mathbf{z}_i^{(k)}, r_i^n),
\end{aligned}$$

and the right-hand sides are defined as

$$\begin{aligned}
r_{\mathbf{b}_i} &= - \left( \frac{\mathbf{b}_i^{(k)} - \mathbf{b}_i^n}{h} - f(t^{n+1}, \mathbf{b}_i^{(k)}, \mathbf{b}_0^{(k)}, \mathbf{z}_i^{(k)}, r_i^n) \right), \quad r_{\mathbf{x}_i} = - \left( G(\mathbf{z}_i^{(k)}, \mathbf{b}_i^{(k)}, \boldsymbol{\lambda}_i^{(k)}) \right) \\
r_{\mathbf{b}_0} &= - \left( \frac{\mathbf{b}_0^{(k)} - \mathbf{b}_0^n}{h} + \sum_{i=1}^N f(t^{n+1}, \mathbf{b}_i^{(k)}, \mathbf{b}_0^{(k)}, \mathbf{z}_i^{(k)}, r_i^n) \right).
\end{aligned}$$

Finally the Newton iterates are updated by setting

$$\mathbf{b}_0^{(k+1)} = \mathbf{b}_0^{(k)} + p_{\mathbf{b}_0}, \quad \mathbf{b}_i^{(k+1)} = \mathbf{b}_i^{(k)} + p_{\mathbf{b}_i}, \quad \mathbf{x}_i^{(k+1)} = \mathbf{x}_i^{(k)} + p_{\mathbf{x}_i}, \quad i = 1, \dots, N,$$

until some stopping criterion is satisfied. The specific features of this linear system come from the mixing of optimization and differential-algebraic equations.

The numerical method for the solution of (5) relies on the sparse structure of the linear system. In order to avoid additional errors induced by an iterative method for the solution of linear systems, a direct method is preferred. When the size  $n$  of the last block line is small compared to the number of diagonal blocks  $N$ , a Schur complement approach is known to be efficient [7, 8]. The underlying idea is to eliminate the weak coupling variable  $\mathbf{b}_0$ , and decouple the linear systems relative to each subsystem. The Schur complement system

$$\mathcal{S}p_{\mathbf{b}_0} = \mathcal{R}, \quad (6)$$

is first constructed and solved, where

$$\begin{aligned} \mathcal{S} &= H_0 - \sum_{i=1}^N \begin{pmatrix} F_i & D_i \end{pmatrix} \begin{pmatrix} H_i & B_i \\ E_i & O_i \end{pmatrix}^{-1} \begin{pmatrix} C_i \\ 0 \end{pmatrix} \in \mathbb{R}^{n \times n} \\ \mathcal{R} &= r_{\mathbf{b}_0} - \sum_{i=1}^N \begin{pmatrix} F_i & D_i \end{pmatrix} \begin{pmatrix} H_i & B_i \\ E_i & O_i \end{pmatrix}^{-1} \begin{pmatrix} r_{\mathbf{b}_i} \\ r_{\mathbf{x}_i} \end{pmatrix} \in \mathbb{R}^n \end{aligned} \quad (7)$$

For  $h > 0$  sufficiently small, the linear system (5) and the Schur complement  $\mathcal{S}$  are invertible, and therefore (6) is well-defined, and solved with a LU decomposition. Although  $\mathcal{S}$  is invertible, it may be ill-conditioned. If needed, a regularization of  $\mathcal{S}$  can be achieved by adding positive terms in its diagonal. Adding terms to the diagonal of  $\mathcal{S}$  corresponds to modifying either  $H_i$ ,  $O_i$  or  $H_0$  in the original system. Increasing the diagonal terms of  $H_i$  and  $H_0$  corresponds to an artificial decrease of the time step  $h$ , while increasing the diagonal terms of  $O_i$  can be interpreted as a convexification of the objective function  $g$  in the underlying optimization problem.

Once the solution of (6) is obtained, a sequence of linear systems is solved for given  $p_{\mathbf{b}_0}$ :

$$\begin{pmatrix} H_i & B_i \\ E_i & O_i \end{pmatrix} \begin{pmatrix} p_{\mathbf{b}_i} \\ p_{\mathbf{x}_i} \end{pmatrix} = \begin{pmatrix} r_{\mathbf{b}_i} \\ r_{\mathbf{x}_i} \end{pmatrix} - \begin{pmatrix} C_i \\ 0 \end{pmatrix} p_{\mathbf{b}_0}, \quad i = 1, \dots, N. \quad (8)$$

The solutions of systems similar to (8) are also required for the construction of the Schur complement system (7). The cost of the total algorithm is equivalent to the solution of one system of size  $n$  (6), and  $N(n+2)$  systems of size  $m+2n$  (8).

**Remark 1.** The use of a Schur complement strategy for this kind of matrices is standard. Note that the matrix  $\mathcal{S}$  is full, but of reasonable size  $n$ . On the other hand, we can take advantage of the special block structure of the matrices appearing in (8) in a second level of resolution as explained below.

**Remark 2.** The linear system (8) is block-structured. A *hierarchical* sequence of linear algebra techniques can be designed. Problem (8) is therefore solved by LU decomposition

$$\begin{pmatrix} H_i & B_i \\ E_i & O_i \end{pmatrix} = \begin{pmatrix} H_i & 0 \\ E_i & V_i \end{pmatrix} \begin{pmatrix} I & H_i^{-1}B_i \\ 0 & I \end{pmatrix}, \quad (9)$$

where  $V_i = O_i - E_i H_i^{-1} B_i$ . Optimization-like techniques similar to those presented in [1] can be developed for the solution of linear systems with matrix  $V_i$ .

The simulation of large population of subsystems ( $N \gg 1$ ) may suffer from scalability issues when decomposition methods are not used. The computational

cost of the algorithm for the solution of (5) is at least proportional to the number of diagonal blocks  $N$ , but may be larger when using direct methods without decomposition. Numerical results presented in the next section show the scalability properties of the Schur complement algorithm, and the computational cost when  $N$  becomes large.

**4. Numerical results.** We consider the three-component system ( $n = 3$ ) formed by 1-hexacosanol ( $\text{C}_{26}\text{H}_{54}\text{O}$ ), pinic acid ( $\text{C}_9\text{H}_{14}\text{O}_4$ ) and water ( $\text{H}_2\text{O}$ ) at temperature 298.15 [K] and pressure 1 [atm]. The energy function  $g$  is a non-convex function defined by the chemical UNIFAC model (see [1] and references therein). The flux  $f$  depends on the radius  $r$ ,  $\mathbf{b}_0$  and  $\mathbf{z}$  only, and is given by  $f(t, \mathbf{b}_0, \mathbf{z}, r) = \varphi(r)(\mathbf{b}_0 - \Gamma e^{\nabla g(\mathbf{z})})$ , where  $\varphi(r)$  is the mass transfer rate, and  $\Gamma$  is a constant depending on vapor pressures and temperature [2]. The radius is computed by the relation  $\frac{4}{3}\pi r_i(t)^3 = \sum_{k=1}^n \frac{b_{i,k}(t)m_k}{\rho_k}$ , where  $m_k$  is the molecular weight and  $\rho_k$  is the density of the chemical component  $k$ .

The evolution of three particles of different radii is considered, when the global system reaches its equilibrium. A time step of  $\tau = 0.5$  is chosen. Initial concentrations are given by  $\mathbf{b}_{1,0} = (2.0 \cdot 10^{-1}, 1.0 \cdot 10^{-1}, 7.0 \cdot 10^{-1})$ ,  $\mathbf{b}_{2,0} = (5.0 \cdot 10^0, 1.5 \cdot 10^1, 1.7 \cdot 10^0)$ ,  $\mathbf{b}_{3,0} = (7.0 \cdot 10^{-1}, 8.0 \cdot 10^{-1}, 8.00 \cdot 10^{-2})$  and  $\mathbf{b}_{0,0} = (2.00 \cdot 10^0, 0.11 \cdot 10^{-2}, 1.27 \cdot 10^0)$ . The corresponding radii of the particles are given by  $r_1 = 6.64 \cdot 10^{-6}$ ,  $r_2 = 22.74 \cdot 10^{-6}$  and  $r_3 = 10.35 \cdot 10^{-6}$  respectively, and  $n = 3$  and  $m = 9$ .

Figure 2 (left) shows the evolution of the normalized particle feed  $\mathbf{b}_i(t)/\mathbf{e}^T \mathbf{b}_i(t)$ ,  $i = 1, 2, 3$  in the two-dimensional simplex (triangle). Each vertex of this two-dimensional simplex corresponds to a composition without mixing, while any point in the interior of the simplex corresponds to a specific mixing of water, pinic acid and 1-hexacosanol. Regions separated by solid black lines are determined by the number of activated inequality constraints [2]. Figure 2 (right) illustrates the evolution and the convergence to a stationary solution for the radii  $r_i(t)$ ,  $i = 1, 2, 3$ .

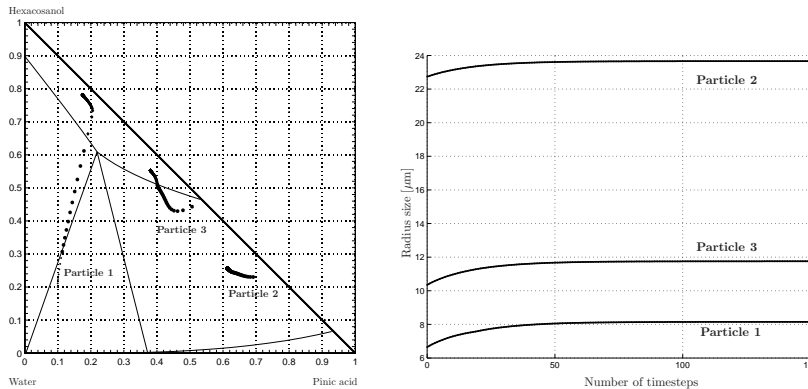


FIGURE 2. Dynamics for three particles and three components (water, pinic acid, 1-hexacosanol). Left: evolution of the normalized concentration vectors  $\mathbf{b}_i(t)/\mathbf{e}^T \mathbf{b}_i(t)$ ,  $i = 1, 2, 3$ , in the phase diagram; right: evolution of the radii  $r_i(t)$   $i = 1, 2, 3$ , of the particles.

The scalability of the algorithm is discussed by considering a population of  $N$  particles of identical initial concentrations  $\mathbf{b}_i(0) = (0.2, 0.1, 0.7)^T$ , together with  $\mathbf{b}_0(0) = (2.001, 0.00114, 1.27)^T$ . A simulation with 20 time steps of size  $h = 0.5$  is considered. Numerical results exhibited in Figure 3 show that the Schur complement approach exhibits a (*optimal*) linear growth of the CPU time with respect to the number of diagonal blocks  $N$ . On the other hand, when using the *monolithic approach* that consists of solving directly (5) with an LU-type method, the CPU time increases faster (but is not proportional to  $N^3$ , due to the sparsity of the matrix). Such a direct resolution of (5) with a LU decomposition is actually never used in practice, and the results in Figure 3 shows that it is computationally not tractable for that particular application.

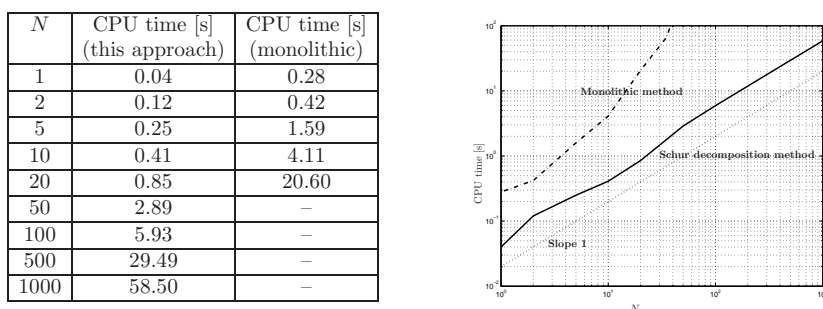


FIGURE 3. Scalability of the numerical method: Left: CPU times for the simulation of the system of differential-algebraic equations, as a function of the number of subsystems. Right: log-log plot of the CPU times for the simulation of the system of differential-algebraic equations, vs. the number of subsystems  $N$ ; decomposition approach (solid line); monolithic approach (dashed line).

**5. Fast/slow scales.** In atmospheric chemistry applications, the size of each particle strongly influences the time before reaching a stationary or nearly-stationary solution, and small particles are reaching equilibrium more rapidly [16]. From the numerical viewpoint, small particles therefore require smaller time steps to capture the fast scales, while larger ones allow larger time steps (see for instance [5]).

Since the sizes of particles are forming a continuum spectrum, the separation between fast and slow entities is not trivial. Fast subsystems evolving on a fast time scale require a small time step to allow the internal concentrations to reach a stationary solution, while the concentrations in slower subsystems stay approximately constant. When fast subsystems have reached a stationary state, a larger time step can be used for the simulation of the slow scales.

Particles of various sizes are considered to study the qualitative behavior of the population and design decomposition techniques into fast and slow subsystems. Let us define  $\bar{\mathbf{b}} = (0.2, 0.7, 0.1)^T$ ,  $\mathbf{b}_0(0) = (2.001, 0.00114, 1.27)^T$ , and  $\mathbf{b}_i(0) = 10^k \cdot \bar{\mathbf{b}}$  for  $k = -1, 0, 1, 2, 3, 6$  respectively. These initial values correspond to  $N = 6$  values of radii (reaction speeds). These values are typical values for large atmospheric particles.

Figure 4 show the evolution of the subsystems, by showing the difference between two successive (normalized) iterates  $\|\mathbf{b}_i^{n+1}/(\mathbf{e}^T \mathbf{b}_i^{n+1}) - \mathbf{b}_i^n/(\mathbf{e}^T \mathbf{b}_i^n)\|$  for  $i =$

$1, \dots, N$ , as a function of the time iterations. It shows that the transient time before reaching a stationary solution depends strongly on the size of the subsystem.

Particles with a smaller radius are represented with solid lines on Figure 4 and considered *fast variables*. The evolution of the increments shows a small overshoot that indicates that the time step cannot be larger for these subsystems. Particles with larger radii are represented with dashed lines on Figure 4, and considered *slow variables*. Figure 4 (left) shows the continuum of time scales among various particles. When considering only two particles with radii  $r_1 = 1.53 \cdot 10^{-6}$  and  $r_2 = 6.64 \cdot 10^{-4}$ , Figure 4 (right) shows that a clear distinction between the two regimes can be made. The time to reach a stationary state is approximately twice longer when the radius is  $10^2$  times larger. As expected, the radius of the particle can therefore be used as a criterion to distinguish between fast and slow scales.

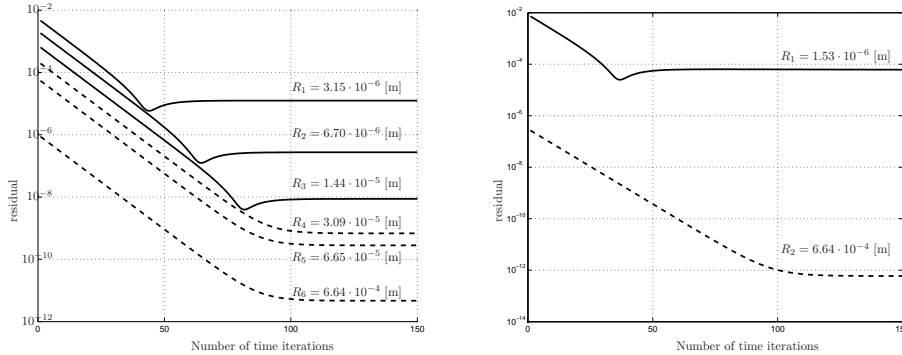


FIGURE 4. Fast/slow decomposition: evolution of the increments  $\|\mathbf{b}_i^{n+1}/(\mathbf{e}^T \mathbf{b}_i^{n+1}) - \mathbf{b}_i^n/(\mathbf{e}^T \mathbf{b}_i^n)\|$  for subsystems of various sizes. Left: System of  $N = 6$  particles with respective radii  $3.15 \cdot 10^{-6}$ ,  $6.70 \cdot 10^{-6}$ ,  $1.44 \cdot 10^{-5}$ ,  $3.09 \cdot 10^{-5}$ ,  $6.65 \cdot 10^{-5}$ ,  $6.64 \cdot 10^{-4}$  [m]; right: System of  $N = 2$  particles with radii  $r_1 = 1.53 \cdot 10^{-6}$  and  $r_2 = 6.64 \cdot 10^{-4}$ .

With such continuous spectrum, the separation of several scales is not trivial, and the subject of current research. We propose in the following a first numerical strategy for the decomposition of the subsystems into fast and slow entities when considering only two categories of sizes (or speeds) for the particles in the population. Note that our goal at this point is not to design decomposition methods for the accurate simulation of all scales, but to investigate how to reduce the computational time in some simple cases, without sacrificing the accuracy of the solution.

The system (5) is first reduced to a system of smaller size by freezing the slow subsystems. A small time step is used for the reduced system until the fast subsystems reach a stationary state. Then the slow subsystems are reintroduced into the system and a larger time step is used. This alternating procedure is repeated.

The numerical decomposition of a system of two particles, with one small and one large particle, is illustrated here. The initial conditions are  $\mathbf{b}_{1,0} = (2.0 \cdot 10^{-3}, 1.0 \cdot 10^{-3}, 7.0 \cdot 10^{-3})$ ,  $\mathbf{b}_{2,0} = (5.0 \cdot 10^0, 10.0 \cdot 10^0, 2.0 \cdot 10^0)$  and  $\mathbf{b}_{0,0} = (2.0 \cdot 10^0, 1.1 \cdot 10^{-3}, 1.27 \cdot 10^0)$ . We compute the trajectories over 300 time steps of size  $h = 0.005$ .

Figure 5 compares the evolution of the trajectories  $\mathbf{b}_i^n/(\mathbf{e}^T \mathbf{b}_i^n)$  when using two approaches: i) when solving the complete system with two particles, or ii) when



*deactivating* the slow particle until the fast particle reaches a stationary state. The trajectories of the fast particle by using both methods are very similar, and the maximal difference between the values of  $\mathbf{b}_1(t)/(\mathbf{e}^T \mathbf{b}_1(t))$  obtained by the two methods (in the Euclidean norm) is smaller than  $1.38 \cdot 10^{-4}$ . This implies that the influence of the slow subsystem on the fast one can be neglected in a first stage. On the other hand the CPU time when deactivating the slow particle is 1.16[s] vs. 2.02[s] when computing with both subsystems, implying a significant gain of CPU time.

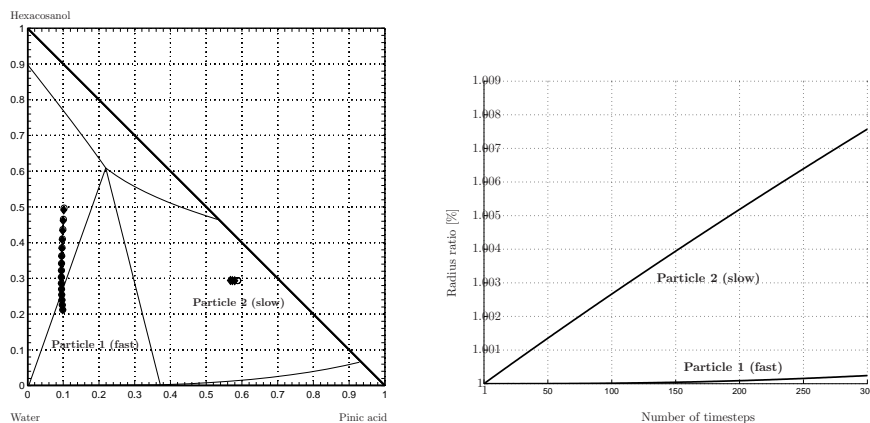


FIGURE 5. Fast/slow decomposition: Comparison of the trajectories of a two-particle system when freezing the slow particle. Left: comparison of the trajectories  $\mathbf{b}_i^n/(\mathbf{e}^T \mathbf{b}_i^n)$  for the resolution with two particles ( $\blacklozenge$ ) and with one fast particle only ( $\circ$ ). The trajectories for the fast particles are nearly superimposed. Right: Representation of the ratio between radii obtained with both methods. The ratio for the fast particle is close to one, implying less than 1% of error on the size of the slow particle.

As expected, the simulation of the fast particle with the decomposition method is very close to the original one. Errors introduced for the slow particles have to be compensated for by reintroducing the slow variables. More thorough strategies for the decoupling of the population into fast and slow entities will be investigated in the future, following [5] for instance. When the particles cannot be distinguished into only two categories (fast/slow), decomposition techniques are not straightforward. The amplitudes of the residuals have to be categorized into several classes (depending of their order of magnitude) at each time step. We plan to investigate a hierarchic approach to define *e.g.* fast/slow particles at several levels: among the fast particles, we will distinguish recursively the various fast/slow scales, and repeat the procedure. The number of repetitions of the procedure is still to be determined, but can rely on the number of *size bins* used in three-dimensional global climate models [16] (typically between 10 and 20 size bins).

**6. Conclusion.** The modeling of a population of subsystems at equilibrium has been addressed. The subsystems are weakly coupled through a common medium. An efficient numerical method has been proposed and optimal results for the computational cost have been exhibited. The decomposition of the subsystems between

various time scales has been investigated as a function of the size of the subsystems, and a continuum of scales throughout the population has been obtained. A first decomposition method, valid only when two kinds of particles (fast/slow) exist in the system, has shown to provide accurate results with significantly smaller computational cost. Application to atmospheric organic particles has been presented. A recursive approach for the decomposition of the various scales will be investigated in the future.

#### REFERENCES

- [1] N. R. Amundson, A. Caboussat, J. W. He, and J. H. Seinfeld. Primal-dual interior-point algorithm for chemical equilibrium problems related to modeling of atmospheric organic aerosols. *J. Optim. Theory Appl.*, 130(3):375–407, 2006.
- [2] N. R. Amundson, A. Caboussat, J.W. He, C. Landry, and J. H. Seinfeld. A dynamic optimization problem related to organic aerosols. *C. R. Acad. Sci.*, 344(8):519–522, 2007.
- [3] A. Caboussat and C. Landry. A second order scheme for solving optimization-constrained differential equations with discontinuities. *Numerical Mathematics and Advanced Applications, Springer Verlag, Berlin*, pages 761–768, 2008.
- [4] A. Caboussat and A. Leonard. Numerical method for a dynamic optimization problem arising in the modeling of a population of aerosol particles. *C. R. Acad. Sci.*, 346(11-12):677–680, 2008.
- [5] B. Engquist and Y Tsai. Heterogeneous multiscale methods for stiff ordinary differential equations. *Math. Comp.*, 74(252):1707–1742, 2005.
- [6] M. Golubitsky, K. Josic, and T. J. Kaper. *Global Analysis of Dynamical Systems*, chapter An Unfolding Theory Approach to Bursting in Fast-Slow Systems. Inst. Phys. Bristol, 2001.
- [7] J. Gondzio and A. Grothey. Direct solution of linear systems of size  $10^9$  arising in optimization with interior point methods. *Lecture Notes in Computer Science, Springer-Verlag, Berlin*, pages 513–525, 2006.
- [8] J. Gondzio and A. Grothey. Solving nonlinear portfolio optimization problems with the primal-dual interior point method. *European Journal of Operational Research*, 181(3):1019–1029, 2007.
- [9] N. Guglielmi and E. Hairer. Computing breaking points in implicit delay differential equations. *Advances in Computational Mathematics*, 29(3):229–247, 2008.
- [10] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems.*, volume 14 of *Springer Series in Computational Mathematics*. Springer-Verlag, 1996.
- [11] E. Hairer and G. Wanner. Stiff differential equations solved by Radau methods. *J. Comput. Appl. Math.*, 111:93–111, 1999.
- [12] M. Z. Jacobson. A solution to the problem of nonequilibrium acid/base gas-particle transfer at long time step. *Aerosol Science and Technology*, 39:92–103, 2005.
- [13] T. J. Kaper. An introduction to geometric methods and dynamical systems theory for singular perturbation problems. In *Analyzing multiscale phenomena using singular perturbation methods (Baltimore, MD, 1998)*, pages 85–131. Amer. Math. Soc., Providence, RI, 1999.
- [14] C. Landry, A. Caboussat and E. Hairer. Solving Optimization-Constrained Differential Equations with Discontinuity Points, with Application to Atmospheric Chemistry. *SIAM J. Sci. Comp.*, to appear, 2009.
- [15] A. Sandu, J. G. Verwer, J. G. Blom, E. J. Spee, G. R. Carmichael, and F. A. Potra. Benchmarking stiff ODE solvers for atmospheric chemistry problems, II, Rosenbrock solvers. *Atmos. Environ.*, 31:3459–3472, 1997.
- [16] J. H. Seinfeld and S. N. Pandis. *Atmospheric Chemistry and Physics: From Air Pollution to Climate Change*. Wiley, New York, 1998.
- [17] R. A. Zaveri, R. C. Easter, J. D. Fast, and L. K. Peters. Model for simulating aerosol interactions and chemistry (MOSAIC). *J. Geophys. Res.*, 113, D13204, 2008.

Received June 2008; revised June 2009.

*E-mail address:* caboussat@math.uh.edu

*E-mail address:* aleonard@math.uh.edu