MATHEMATICS OF COMPUTATION
Volume 75, Number 253, Pages 209–222
S 0025-5718(05)01782-5
Article electronically published on September 29, 2005

AN EFFICIENT NUMERICAL SCHEME FOR PRECISE TIME INTEGRATION OF A DIFFUSION-DISSOLUTION/PRECIPITATION CHEMICAL SYSTEM

BLAISE FAUGERAS, JÉRÔME POUSIN, AND FRANCK FONTVIEILLE

Abstract. A numerical scheme based on an operator splitting method and a dense output event location algorithm is proposed to integrate a diffusion-dissolution/precipitation chemical initial-boundary value problem with jumping nonlinearities. The numerical analysis of the scheme is carried out and it is proved to be of order 2 in time. This global order estimate is illustrated numerically on a test case.

1. Introduction

In this article we address the problem of the numerical integration of a complex diffusion-dissolution/precipitation chemical system of equations constituted of partial differential equations and ordinary differential equations with nonlinear discontinuous right hand side. Such systems arise in models describing the retention capacity of concrete matrices in which wastes and pollutants are embedded. The particular model we have in mind is described and studied from a mathematical point of view in [7] and [8]. It takes into account the influence of the chemical context evolution on the dissolution/precipitation rates and expresses the necessary presence of solid for dissolution by an obstacle problem. The multi-species diffusion-dissolution/precipitation model takes the form of an initial-boundary value problem in which partial differential equations (PDEs) and ordinary differential equations (ODEs) are coupled through nonlinear discontinuous terms. The system of equations for N_s species is formulated as follows. $\mathbf{C} = (C_i)_{i=1,\ldots,N_s}$ is the vector of chemical species concentrations in liquid phase and $\mathbf{S} = (S_i)_{i=1,\ldots,N_s}$ is the vector of chemical species concentrations in solid phase. C_i^* are nonlinear functions of C representing saturation concentrations, α_i and D_i are strictly positive constants. The following notation is also used

$$\forall z \in \mathbb{R},\ z^+ = \max(z,0) \quad \text{and } z^- = z^+ - z \geq 0,$$

and

$$\operatorname{sgn}^+(z) = \begin{cases} 1 \text{ if } z > 0, \\ 0 \text{ otherwise.} \end{cases}$$

Received by the editor December 2, 2003 and, in revised form, September 20, 2004.

2000 *Mathematics Subject Classification.* Primary 65M12, 65G99, 35K57.

Key words and phrases. Numerical time integration, operator splitting, dense output, high order, error analysis, reaction-diffusion, jumping nonlinearities.

For $i = 1$ to N_s, we have

$$\begin{cases} \partial_t C_i = D_i \Delta C_i + \operatorname{sgn}^+(S_i)\alpha_i(C_i^*(C) - C_i)^+ - \alpha_i(C_i^*(C) - C_i)^- & \text{in } (0,T) \times \Omega, \\ \partial_t S_i = -\operatorname{sgn}^+(S_i)\alpha_i(C_i^*(C) - C_i)^+ + \alpha_i(C_i^*(C) - C_i)^- & \text{in } (0,T) \times \Omega, \\ C_i(0,x) = C_i^0(x) > 0, \ S_i(0,x) = S_i^0(x) > 0 & \text{in } \Omega, \\ C_i(t,x) = 0 & \text{in } (0,T) \times \partial\Omega. \end{cases} \tag{1.1}$$

The purpose of this article is to present an efficient numerical scheme of order 2 in time to integrate systems such as system (1.1). The scheme proposed in [7] is based on a simple implicit Euler method and has two main drawbacks. First a large nonlinear system has to be solved at each time step and second it is only of order 1 in time. We propose a scheme combining an operator splitting method ([13], [9]) and an event location algorithm using a dense output formula. Operator splitting methods are known to provide cheap and high order approximations to reaction-diffusion equations [2], [10], [6]. Therefore, they represent an interesting tool for dealing with large chemical systems. The event location algorithm presented in Section 3 enables us to determine the switching times at which the discontinuities occur in the reaction terms with a desired accuracy.

Throughout this article we consider a semi-discretized system of equations. Indeed a difficulty appears in the fully continuous case that we are not able to cope with easily. The switching time, t_d, is an unknown function of x, the space variable. Dealing with the continuous case then means considering reaction-diffusion equations defined on a noncylindrical domain. One can bring back the problem to a cylindrical domain by rescaling the time variable but then time and space dependent coefficients with unknown regularity appear in the equations. Thus, we consider that the chemical system is already discretized in space, using, for example, a finite difference or a finite element method. The system of ODEs we consider then reads

$$\begin{cases} \dfrac{d\mathbf{C}}{dt} = \mathbf{A}\mathbf{C} + \mathbf{F}(\mathbf{C}, \mathbf{S}), \\ \dfrac{d\mathbf{S}}{dt} = -\mathbf{F}(\mathbf{C}, \mathbf{S}), \\ \mathbf{C}(0) = \mathbf{C}_0, \ \mathbf{S}(0) = \mathbf{S}_0. \end{cases} \tag{1.2}$$

$\mathbf{C}$ and $\mathbf{S}$ are vectors of $\mathbb{R}^N$ and $\mathbf{A}$ is the $N \times N$ matrix resulting from the spatial discretization of the Δ operator which is symmetric negative definite. The nonlinear terms read

$$\mathbf{F}(\mathbf{C}, \mathbf{S}) = (F_k(\mathbf{C}, \mathbf{S}))_{k=0,\dots,N},$$

with

$$F_k(\mathbf{C}, \mathbf{S}) = \begin{cases} G_k^1(\mathbf{C}), \text{ if } S_k > 0, \\ \\ G_k^2(\mathbf{C}), \text{ if } S_k \leq 0. \end{cases} \tag{1.3}$$

This paper is organized as follows. In Section 2 we present the operator splitting method and show that it can be applied to a system in which PDEs and ODEs are coupled. Estimates of the local errors are given. Section 3 deals with the numerical treatment of the discontinuities in the nonlinear reaction terms. We formalize the event location algorithm suggested in [4] and give estimates of the local errors. We then describe the scheme we propose, combining an operator splitting method and an adaptation of the event location algorithm, and show it is of order 2. In the final Section 4 the effectiveness of the scheme is illustrated numerically.

2. Operator splitting

The main topic of this section is to present the operator splitting method which constitutes the first ingredient of the scheme we propose. We first concentrate on a classical reaction-diffusion equation and give estimates of the local errors. We then show that the method can still be applied without any order reduction ([3], [12], [5]) if an ODE is coupled to the first equation.

2.1. **Strang operator splitting.** In this section we only consider the semi-discretized diffusion-reaction equation for $\mathbf{C}$. The problem of the switching of the nonlinear discontinuous reaction terms is also left aside. We assume that locally the reaction term $\mathbf{F}(\mathbf{C},\mathbf{S})$ is given by a smooth function $\mathbf{G}(\mathbf{C})$. As in [1], $\mathbf{G}$ is a Lipschitz function with constant L of class C^∞ such that $\mathbf{G}(0)=0$ and the first four derivatives of $\mathbf{G}$ are bounded.

Let R^t denote the flow (also called fundamental solution operator) of the system

$$\begin{cases} \dfrac{d\mathbf{C}}{dt} = \mathbf{A}\mathbf{C} + \mathbf{G}(\mathbf{C}),\ t>0, \\ \mathbf{C}(0) = \mathbf{C}_0. \end{cases} \tag{2.1}$$

Let Y^t denote the flow of system (2.2),

$$\begin{cases} \dfrac{d\mathbf{C}_1}{dt} = \mathbf{G}(\mathbf{C}_1),\ t>0, \\ \mathbf{C}_1(0) = \mathbf{C}_{0,1}, \end{cases} \tag{2.2}$$

and X^t denote the flow of system (2.3),

$$\begin{cases} \dfrac{d\mathbf{C}_2}{dt} = \mathbf{A}\mathbf{C}_2,\ t>0, \\ \mathbf{C}_2(0) = \mathbf{C}_{0,2}. \end{cases} \tag{2.3}$$

The idea of splitting methods is to approximate R^t by combining the two flows X^t and Y^t. Two classical approximations are given by the Strang formulas [13],

$$Z^t_{DRD} = X^{t/2}Y^tX^{t/2} \quad \text{and} \quad Z^t_{RDR} = Y^{t/2}X^tY^{t/2}$$

(which we also denote by diffusion-reaction-diffusion or DRD-splitting and RDR-splitting in the remaining part of this paper). The following result holds.

Lemma 2.1. *Let $\mathbf{C}_0 \in \mathbb{R}^N$. For t sufficently small, the local errors for the two splitting schemes satisfy,*

$$R^t\mathbf{C}_0 - Z^t_{DRD}\mathbf{C}_0 = O(t^3)$$

$$R^t\mathbf{C}_0 - Z^t_{RDR}\mathbf{C}_0 = O(t^3).$$

Proof. This result is the particular finite dimensional case of local error estimation results obtained by Besse et al. in [1]. It can be derived using the same tools, essentially Taylor expansions and judicious estimations of the rest, with some minor adaptations due to finite dimension. For the sake of completeness we give here the main ideas of the proof. Let us denote by $||.||$ the euclidean norm on $\mathbb{R}^N$. We may write a Duhamel formula for problem (2.1), which reads

$$R^t\mathbf{C}_0 = X^t\mathbf{C}_0 + \int_0^t X^{t-s}\mathbf{G}(R^s\mathbf{C}_0)ds,$$

and express the difference between the exact solution and the splitting solution $Z^t\mathbf{C}_0$ (DRD or RDR) as

$$R^t\mathbf{C}_0 - Z^t\mathbf{C}_0 = \int_0^t X^{t-s}(\mathbf{G}(R^s\mathbf{C}_0) - \mathbf{G}(Z^s\mathbf{C}_0))ds + \mathbf{W}(t).$$

Since $\mathbf{G}$ is Lipschitz with constant $L > 0$ such that for all $\mathbf{C}_1, \mathbf{C}_2 \in \mathbb{R}^N$

$$||\mathbf{G}(\mathbf{C}_1) - \mathbf{G}(\mathbf{C}_2)|| \leq L||\mathbf{C}_1 - \mathbf{C}_2||.$$

The matrix $\mathbf{A}$ is negative definite. Thus for all $\mathbf{V} \in \mathbb{R}^N$ and all $t \geq 0$ the following inequality holds for the semi-group $e^{t\mathbf{A}}$,

$$||X^t\mathbf{V}|| = ||e^{t\mathbf{A}}\mathbf{V}|| \leq ||\mathbf{V}||.$$

It follows that

$$||R^t\mathbf{C}_0 - Z^t\mathbf{C}_0|| \leq L\int_0^t ||\mathbf{G}(R^s\mathbf{C}_0) - \mathbf{G}(Z^s\mathbf{C}_0)||ds + ||\mathbf{W}(t)||.$$

Then the estimates of Lemma 2.1 are obtained in the same way as in the proof of Lemma 3.1 p.13 of [1] by accounting for the following changes. In Lemma 2.2.1 p.4 from [1] the L^2, H^2, and H^4 norms are replaced by the Euclidean norm $\| \cdot \|$, the A-norm $\| \cdot \|_A$=$\|A \cdot \|$ and the A^2-norm $\| \cdot \|_{A^2}$=$\|A^2 \cdot \|$, respectively . Then by using the Gronwall Lemma (p.3 from [1]), the rest $W(t)$ is estimated as $||W(t)|| = O(t^3)$ for t small and Lemma 2.1 reduces to Lemma 3.1 from [1] with the norms previously introduced. □

2.2. **Coupling reaction-diffusion equations and ODEs.** Let us now consider the coupling of equation (2.1) with the equation for $\mathbf{S}$:

$$\left\{\begin{array}{l} \dfrac{d\mathbf{C}}{dt} = \mathbf{AC} + \mathbf{G}(\mathbf{C}), \\ \dfrac{d\mathbf{S}}{dt} = -\mathbf{G}(\mathbf{C}), \\ \mathbf{C}(0) = \mathbf{C}_0, \ \mathbf{S}(0) = \mathbf{S}_0. \end{array}\right. \tag{2.4}$$

The solution $(\mathbf{C}(t), \mathbf{S}(t))$ to (2.4) is denoted by $R^t(\mathbf{C}_0, \mathbf{S}_0) = (R^t_C\mathbf{C}_0, R^t_S(\mathbf{C}_0, \mathbf{S}_0))$. $\mathbf{S}(t)$ is given explicitly by

$$\mathbf{S}(t) = \mathbf{S}_0 - \int_0^t \mathbf{G}(\mathbf{C}(s))ds,$$

which can also be written $\mathbf{S} = \mathbf{H}(\mathbf{C})$. In such a situation Descombes and Massot [3] show that order reduction occurs in the DRD-splitting but not in the RDR-splitting. The problem we consider is quite similar. However, because of the particular form of function $\mathbf{H}$ which should be written as

$$\mathbf{S}(t) = \mathbf{H}(t, \mathbf{S}_0, \mathbf{C}(.)),$$

no order reduction occurs as is shown in Lemma 2.2. Let us denote by

$$(\mathbf{C}_1(t), \mathbf{S}_1(t)) = (Y^t_C\mathbf{C}_{0,1}, \mathbf{H}(t, \mathbf{S}_{0,1}, Y^{\cdot}_C\mathbf{C}_{0,1}))$$

the solution to system

$$\left\{\begin{array}{l}\dfrac{d\mathbf{C}_1}{dt} = \mathbf{G}(\mathbf{C}_1),\\ \dfrac{d\mathbf{S}_1}{dt} = -\mathbf{G}(\mathbf{C}_1),\\ \mathbf{C}_1(0) = \mathbf{C}_{0,1},\ \mathbf{S}_1(0) = \mathbf{S}_{0,1}.\end{array}\right. \tag{2.5}$$

The DRD-splitting for system (2.4) can be written as

$$Z^t_{DRD}(\mathbf{C}_0, \mathbf{S}_0) = (Z^t_{DRD,C}\mathbf{C}_0, Z^t_{DRD,S}(\mathbf{C}_0, \mathbf{S}_0)),$$

with

$$Z^t_{DRD,C}\mathbf{C}_0 = X^{t/2}Y^t_C X^{t/2}\mathbf{C}_0, \tag{2.6}$$

$$Z^t_{DRD,S}(\mathbf{C}_0, \mathbf{S}_0) = \mathbf{H}(t, \mathbf{S}_0, Y^{\cdot}_C X^{t/2}\mathbf{C}_0), \tag{2.7}$$

and the RDR-splitting as

$$Z^t_{RDR}(\mathbf{C}_0, \mathbf{S}_0) = (Z^t_{RDR,C}\mathbf{C}_0, Z^t_{RDR,S}(\mathbf{C}_0, \mathbf{S}_0)), \tag{2.8}$$

with

$$Z^t_{RDR,C}\mathbf{C}_0 = Y^{t/2}_C X^t Y^{t/2}_C \mathbf{C}_0, \tag{2.9}$$

$$Z^t_{RDR,S}(\mathbf{C}_0, \mathbf{S}_0) = \mathbf{H}(t/2, \mathbf{H}(t/2, \mathbf{S}_0, Y^{\cdot}_C\mathbf{C}_0), Y^{\cdot}_C X^t Y^{t/2}_C\mathbf{C}_0)). \tag{2.10}$$

The following result holds.

Lemma 2.2. *Let $\mathbf{C}_0 \in \mathbb{R}^N$ and $\mathbf{S}_0 \in \mathbb{R}^N$. For t sufficiently small, the local errors for the two splitting schemes satisfy*

$$R^t_C\mathbf{C}_0 - Z^t_{DRD,C}\mathbf{C}_0 = O(t^3), \tag{2.11}$$

$$R^t_S(\mathbf{C}_0, \mathbf{S}_0) - Z^t_{DRD,S}(\mathbf{C}_0, \mathbf{S}_0) = O(t^3), \tag{2.12}$$

$$R^t_C\mathbf{C}_0 - Z^t_{RDR,C}\mathbf{C}_0 = O(t^3), \tag{2.13}$$

$$R^t_S(\mathbf{C}_0, \mathbf{S}_0) - Z^t_{RDR,S}(\mathbf{C}_0, \mathbf{S}_0) = O(t^3). \tag{2.14}$$

Proof. Equations (2.11) and (2.13) follow directly from Lemma 2.1. Let us show (2.12) and (2.14). Using the Duhamel formula, $\mathbf{C}(t)$ is given explicitly by

$$\mathbf{C}(t) = e^{t\mathbf{A}}\mathbf{C}_0 + \int_0^t e^{(t-s)\mathbf{A}}\mathbf{G}(\mathbf{C}(s))ds.$$

It follows from classical expansions that

$$\mathbf{C}(t) = \mathbf{C}_0 + t(\mathbf{A}\mathbf{C}_0 + \mathbf{G}(\mathbf{C}_0)) + O(t^2), \tag{2.15}$$

and since $\mathbf{S}(t) = \mathbf{H}(t, \mathbf{S}_0, \mathbf{C}(.))$, we obtain

$$\mathbf{S}(t) = \mathbf{S}_0 - t\mathbf{G}(\mathbf{C}_0) - \frac{t^2}{2}\mathbf{G}'(\mathbf{C}_0)(\mathbf{A}\mathbf{C}_0 + \mathbf{G}(\mathbf{C}_0)) + O(t^3). \tag{2.16}$$

From

$$Y^t_C\mathbf{C}_0 = \mathbf{C}_0 + t\mathbf{G}(\mathbf{C}_0) + O(t^2)$$

and

$$X^t\mathbf{C}_0 = e^{t\mathbf{A}}\mathbf{C}_0 = \mathbf{C}_0 + t\mathbf{A}\mathbf{C}_0 + O(t^2),$$

we deduce that

$$Y^s_C X^{t/2\mathbf{A}}\mathbf{C}_0 = \mathbf{C}_0 + \frac{t}{2}\mathbf{A}\mathbf{C}_0 + s\mathbf{G}(\mathbf{C}_0) + O(t^2) + O(s^2) + O(st)$$

and

$$\mathbf{G}(Y_C^s X^{t/2\mathbf{A}}\mathbf{C}_0) = \mathbf{G}(\mathbf{C}_0)+\frac{t}{2}\mathbf{G}'(\mathbf{C}_0)\mathbf{A}\mathbf{C}_0+s\mathbf{G}'(\mathbf{C}_0)\mathbf{G}(\mathbf{C}_0)+O(t^2)+O(s^2)+O(st).$$

Eventually since

$$Z^t_{DRD,S}(\mathbf{C}_0,\mathbf{S}_0) = \mathbf{H}(t,\mathbf{S}_0,Y_C^{\cdot}X^{t/2}\mathbf{C}_0),$$

we obtain

$$\mathbf{S}(t) - Z^t_{DRD,S}(\mathbf{C}_0,\mathbf{S}_0) = O(t^3),$$

which proves (2.12).

The same type of arguments are used to prove (2.14). □

3. Numerical treatment of the discontinuities

The purpose of this section is twofold. First we present the event location algorihtm for a discontinuous ODE suggested in Hairer et al. [4] and prove that indeed it leads to an accurate numerical method. We then combine this algorithm to a splitting scheme and obtain a method of order 2 to integrate system (1.2).

3.1. **An event location algorithm for ODEs.**

In this section we present a numerical scheme of order $p \geq 2$ to solve a nonlinear discontinuous ODE. The main numerical tool used is an explicit Runge-Kutta method of order p with a dense output of order $p^* \geq 2$. The reader is referred to Sections II-1 to II-6 of the book by Hairer et al. [4] for a detailed description of these methods. We assume here that $p = p^*$.

Let us give some notation. An explicit Runge-Kutta method of order p to solve the ordinary differential equation

$$\left\{\begin{array}{l} y' = f(t,y), \\ y(t_0) = y_0, \end{array}\right. \tag{3.1}$$

is represented by the increment function of the method, $F(t,y,h)$. Given an initial value (t_0,y_0) and a step size h, one computes a numerical solution y_1 approximating $y(t_0+h)$ by $y_1 = y_0 + hF(t_0,y_0,h)$. The numerical solution for a point $T > t_0$ is then obtained by a step-by-step procedure

$$y_{i+1} = y_i + hF(t_i,y_i,h).$$

If the method is of order p, then the local error

$$e_{i+1} = y(t_i+h) - (y(t_i) + hF(t_i,y(t_i),h)),$$

statisfies

$$e_{i+1} = O(h^{p+1}). \tag{3.2}$$

A Runge-Kutta method with a dense output formula provides a cheap numerical approximation to $y(t_i+\theta h)$ for the whole integration interval $0 \leq \theta \leq 1$. We denote this approximation by $u_i(\theta)$, and we have

$$u_i(\theta) = y(t_i+\theta h) + O(h^{p+1}). \tag{3.3}$$

Let us now concentrate on the numerical integration on a time interval $[t_0,T]$ of the following ordinary differential equation:

$$\left\{\begin{array}{l} y' = f_1(t,y) \text{ if } g(y) > 0, \\ y' = f_2(t,y) \text{ if } g(y) \leq 0, \\ y(t_0) = y_0 \text{ and } g(y_0) > 0. \end{array}\right. \tag{3.4}$$

We assume that f_1, f_2 and g are C^∞ functions. The function g is called the switching function. We also assume that the solution $y(t)$ to (3.4) crosses the surface $\Sigma = \{y;\ g(y) = 0\}$ only once, at the point $y_d = y(t_d)$. Therefore, $y(t)$ may be written as

$$\begin{cases} y(t) = y_1(t) \text{ in } [t_0, t_d[, \\ y(t) = y_2(t) \text{ in } [t_d, T], \\ y_1(t_d) = y_2(t_d) = y_d, \end{cases}$$

where y_1 is the solution to

$$\begin{cases} y'(t) = f_1(t, y),\ t > t_0, \\ y(t_0) = y_0, \end{cases}$$

and y_2 is the solution to

$$\begin{cases} y'(t) = f_2(t, y),\ t > t_d, \\ y(t_d) = y_d. \end{cases}$$

The derivative of the solution y is, in general, discontinuous on Σ. The difficulty in the numerical integration of such a discontinuous equation is that the point (t_d, y_d) is not known in advance but has to be detected. Moreover, in order to obtain a method of order p, this point has to be detected with a precision of order p. The method proposed here relies on the event location algorithm suggested in the book by Hairer et al. [4] (Algorithm 6.4 page 195).

Algorithm 3.1.

- *Using f_1, define a Runge-Kutta method of order p with increment function F_1,*
$$y_{i+1} = y_i + hF_1(t_i, y_i, h).$$
- *Compute the solution step-by-step $y_0, y_1, \ldots$ until a sign change appears between $g(y_{n-1})$ and $g(y_n)$.*
- *Using the dense output, find θ such that $g(u_{n-1}(\theta)) = 0$.*
- *Reset $y_n = u_{n-1}(\theta)$ and $t_n = t_{n-1} + \theta h$.*
- *Using f_2, define a Runge-Kutta method of order p with increment function F_2,*
$$y_{i+1} = y_i + hF_2(t_i, y_i, h),$$
and carry on the computation from t_n to $t_N = T$.

The key point in this algorithm is that, thanks to the dense output, we are able to compute $y_n = y_d + O(h^{p+1})$ and $t_n = t_d + O(h^{p+1})$. Thus, we can show the following technical result.

Lemma 3.1. *At each time step of the scheme provided by Algorithm* 3.1, *the local error satisfies*

$$e_i = O(h^{p+1}).$$

Proof. From (3.2) it is clear that for $i = 1$ to $n - 1$,

$$e_i = y_1(t_i) - (y_1(t_{i-1}) + hF_1(t_{i-1}, y_1(t_{i-1}), h)) = O(h^{p+1}),$$

and that for $i = n + 2$ to N,

$$e_i = y_2(t_i) - (y_2(t_{i-1}) + hF_2(t_{i-1}, y_2(t_{i-1}), h)) = O(h^{p+1}).$$

It remains to show the result for e_n and e_{n+1}. Since we only know that $t_n = t_d + O(h^{p+1})$, there are two cases.

- Case $t_{n-1} < t_d \le t_n < t_{n+1}$:
  The local error e_n reads
  $$\begin{aligned} e_n &= y_2(t_n) - (y_1(t_{n-1}) + hF_1(t_{n-1}, y_1(t_{n-1}), h)) \\ &= [y_1(t_n) - (y_1(t_{n-1}) + hF_1(t_{n-1}, y_1(t_{n-1}), h))] + [y_2(t_n) - y_1(t_n)]. \end{aligned}$$
  Moreover, we have $y_1(t_n) = y_d + O(h^{p+1})$, $y_2(t_n) = y_d + O(h^{p+1})$, and we can conclude that $e_n = O(h^{p+1})$.
  Concerning the next step it is also clear that $e_{n+1} = O(h^{p+1})$.
- Case $t_{n-1} < t_n \le t_d < t_{n+1}$:
  It is clear that $e_n = O(h^{p+1})$. The local error e_{n+1} reads
  $$e_{n+1} = y_2(t_{n+1}) - (y_1(t_n) + hF_2(t_n, y_1(t_n), h)).$$
  Since $y_1(t_n) = y_d + O(h^{p+1})$, $t_n = t_d + O(h^{p+1})$, and $h = (t_{n+1} - t_d) + O(h^{p+1})$, we have that
  $$hF_2(t_n, y_1(t_n), h) = (t_{n+1} - t_d)F_2(t_d, y_d, (t_{n+1} - t_d)) + O(h^{p+1})$$
  and
  $$\begin{aligned} e_{n+1} &= y_2(t_{n+1}) - (y_d + (t_{n+1} - t_d)F_2(t_d, y_d, (t_{n+1} - t_d))) + O(h^{p+1}) \\ &= O((t_{n+1} - t_d)^{p+1}) + O(h^{p+1}). \end{aligned}$$
  Therefore, $e_{n+1} = O(h^{p+1})$.

□

The third step of Algorithm 3.1 is crucial. The computation of θ, such that $g(u_{n-1}(\theta)) = 0$, can be done using a dichotomy method or, for example, a second order Muller method. This latter requires that the zeros of g are separated and might require many iterations to converge depending on the "flatness" of g between t_{n-1} and t_n. However, for the applications we considered the desired accuracy on θ, y_n, and t_n can always be achieved.

3.2. **Combining the event location algorithm and the splitting scheme.** In this section we formulate the scheme proposed to integrate system (1.2). The method combines either the RDR-splitting or the DRD-splitting described in Section 2 and an event location algorithm similar to Algorithm 3.1 of Section 3.1. With those tools we construct a numerical scheme of order 2 in time for system (1.2).

Since the discontinuous nonlinear reaction terms only come up in the R-stages, it is tempting to try to detect the switching times only during these stages. However, this is not possible since the intermediate $\mathbf{C}$ or $\mathbf{S}$ values computed after the first two stages of the splitting scheme are not yet in $O(h^3)$. The computed switching time, therefore, cannot be an $O(h^3)$ approximation of the exact switching time, and we need to construct a dense output for a whole time step including the three stages of the splitting method. Hermite interpolation (Shampine [11]) provides an efficient way to construct dense output formulas. Whatever the splitting is, at each time step we have two function values $\mathbf{S}^0$, $\mathbf{S}^1$ and two derivatives $\frac{d\mathbf{S}^0}{dt}$, $\frac{d\mathbf{S}^1}{dt}$ at our disposal and can thus do cubic polynomial interpolation. The resulting formula is

$$\mathbf{u}^S(\theta) = (1-\theta)\mathbf{S}^0 + \theta\mathbf{S}^1 + \theta(\theta-1)\left((1-2\theta)(\mathbf{S}^1 - \mathbf{S}^0) + (\theta-1)h\frac{d\mathbf{S}^0}{dt} + \theta h\frac{d\mathbf{S}^1}{dt}\right).$$

A similar formula, $\mathbf{u}^C$ can be computed for $\mathbf{C}$. Since the splitting is of order 2, we have

$$\mathbf{u}^S(\theta) - \mathbf{S}(t_0 + \theta h) = O(h^3),$$

$$\mathbf{u}^C(\theta) - \mathbf{C}(t_0 + \theta h) = O(h^3).$$

These dense output formulas are used to detect the switching times. This detection is performed component by component, as illustrated in the following algorithm.

Algorithm 3.2.

- *Start from* $\mathbf{C}^0 > 0$*,* $\mathbf{S}^0 > 0$ *and thus with* $\mathbf{F}(\mathbf{C}, \mathbf{S}) = (G_k^1(\mathbf{C}))_{k=0,\ldots,N}$*.*
- *Using either the RDR-splitting or the DRD-splitting, compute the solution step-by-step* $(\mathbf{C}^0, \mathbf{S}^0), (\mathbf{C}^1, \mathbf{S}^1), \ldots$ *until a sign change appears, for a component* k_1*, between* $\mathbf{S}_{k_1}^{n-1}$ *and* $\mathbf{S}_{k_1}^n$*.*
- *Using the dense output polynomial* $\mathbf{u}^S$*, find* θ *such that* $u_{k_1}^S(\theta) = 0$*.*
- *Reset* $t_n = t_{n-1} + \theta h$*,* $\mathbf{S}^n = \mathbf{u}^S(\theta)$*, and* $\mathbf{C}^n = \mathbf{u}^C(\theta)$*.*
- *Change* $G_{k_1}^1$ *to* $G_{k_1}^2$ *and carry on the computation using the new reaction term* $\mathbf{F}$ *until a new sign change appears for another component* S_{k_2}*.*

We denote by

$$(\mathbf{C}^{n+1}, \mathbf{S}^{n+1}) = Z^h(\mathbf{C}^n, \mathbf{S}^n) = (Z_C^h \mathbf{C}^n, Z_S^h(\mathbf{C}^n, \mathbf{S}^n))$$

the numerical scheme provided by Algorithm 3.2. Let us now state a result concerning the estimation of the local errors.

Lemma 3.2. *At each time step of the scheme provided by Algorithm* 3.2*, the local error satisfies*

$$e_i^C = R_C^h \mathbf{C}(t_i) - Z_C^h \mathbf{C}(t_i) = O(h^3), \tag{3.5}$$

$$e_i^S = R_S^h(\mathbf{C}(t_i), \mathbf{S}(t_i)) - Z_S^h(\mathbf{C}(t_i), \mathbf{S}(t_i)) = O(h^3). \tag{3.6}$$

Proof. We restrict ourselves to a time interval $[0, T]$ on which only one component, S_{k_1}, switches at time t_d. Other switchings can be treated in the same way. The exact solution, $(\mathbf{C}(t), \mathbf{S}(t)) = R^t(\mathbf{C}_0, \mathbf{S}_0)$ may be written as

$$\begin{cases} (\mathbf{C}(t), \mathbf{S}(t)) = (\mathbf{C}_1(t), \mathbf{S}_1(t)) = R_1^t(\mathbf{C}_0, \mathbf{S}_0) = (R_{1C}^t \mathbf{C}_0, R_{1S}^t(\mathbf{C}_0, \mathbf{S}_0)), \text{ in } [0, t_d[, \\ (\mathbf{C}(t), \mathbf{S}(t)) = (\mathbf{C}_2(t), \mathbf{S}_2(t)) = R_2^t(\mathbf{C}_d, \mathbf{S}_d) = (R_{2C}^t \mathbf{C}_d, R_{2S}^t(\mathbf{C}_d, \mathbf{S}_d)), \text{ in } [t_d, T], \\ (\mathbf{C}_1(t_d), \mathbf{S}_1(t_d)) = (\mathbf{C}_2(t_d), \mathbf{S}_2(t_d)) = (\mathbf{C}_d, \mathbf{S}_d), \end{cases}$$

where $(\mathbf{C}_1(t), \mathbf{S}_1(t))$ is the solution to

$$\begin{cases} \dfrac{d\mathbf{C}}{dt} = \mathbf{A}\mathbf{C} + \mathbf{F}^1(\mathbf{C}), \ t > 0, \\ \dfrac{d\mathbf{S}}{dt} = -\mathbf{F}^1(\mathbf{C}), \\ \mathbf{C}(0) = \mathbf{C}_0, \ \mathbf{S}(0) = \mathbf{S}_0, \end{cases}$$

and $(\mathbf{C}_2(t), \mathbf{S}_2(t))$ is the solution to

$$\begin{cases} \dfrac{d\mathbf{C}}{dt} = \mathbf{A}\mathbf{C} + \mathbf{F}^2(\mathbf{C}), \ t > t_d, \\ \dfrac{d\mathbf{S}}{dt} = -\mathbf{F}^2(\mathbf{C}), \\ \mathbf{C}(t_d) = \mathbf{C}_d, \ \mathbf{S}(t_d) = \mathbf{S}_d, \end{cases}$$

where

$$\mathbf{F}^1(\mathbf{C}) = (G_k^1(\mathbf{C}))_{k=0,\dots,N},$$

and

$$\mathbf{F}^2(\mathbf{C}) = (G_1^1(\mathbf{C}), \dots, G_{k_1-1}^1(\mathbf{C}), G_{k_1}^2(\mathbf{C}), G_{k_1+1}^1(\mathbf{C}), \dots).$$

Before the switching, (2.13) and (2.14) (or (2.11) and (2.12)) directly show that for $i = 1$ to $n-1$,

$$e_i^C = O(h^3), \qquad e_i^S = O(h^3).$$

In the same way after the switching time we have for $i = n+2$ to N,

$$e_i^C = O(h^3), \qquad e_i^S = O(h^3).$$

It remains to show that e_n^C, e_{n+1}^C, e_n^S, and e_{n+1}^S are $O(h^3)$. Since we only know that $t_n = t_d + O(h^3)$ there are two cases.

- Case $t_{n-1} < t_d \le t_n < t_{n+1}$:

  The local error e_n^C reads

$$\begin{aligned} e_n^C &= R_C^{(t_n-t_{n-1})}\mathbf{C}(t_{n-1}) - Z_C^{(t_n-t_{n-1})}\mathbf{C}(t_{n-1}) \\ &= (R_{1C}^{(t_n-t_{n-1})}\mathbf{C}(t_{n-1}) - Z_C^{(t_n-t_{n-1})}\mathbf{C}(t_{n-1})) \\ &\quad + (R_C^{(t_n-t_{n-1})}\mathbf{C}(t_{n-1}) - R_{1C}^{(t_n-t_{n-1})}\mathbf{C}(t_{n-1})). \end{aligned}$$

  Again, from (2.13) we know that

$$(R_{1C}^{(t_n-t_{n-1})}\mathbf{C}(t_{n-1}) - Z_C^{(t_n-t_{n-1})}\mathbf{C}(t_{n-1})) = O(h^3).$$

  Moreover, since

$$\begin{aligned} &(R_C^{(t_n-t_{n-1})}\mathbf{C}(t_{n-1}) - R_{1C}^{(t_n-t_{n-1})}\mathbf{C}(t_{n-1})) \\ &\qquad = (R_{2C}^{(t_n-t_d)} - R_{1C}^{(t_n-t_d)})R_{1C}^{(t_d-t_{n-1})}\mathbf{C}(t_{n-1}) \\ &\qquad = (R_{2C}^{(t_n-t_d)} - R_{1C}^{(t_n-t_d)})\mathbf{C}_d, \end{aligned}$$

  we obtain using the same expansion as in (2.15),

$$(R_C^{(t_n-t_{n-1})}\mathbf{C}(t_{n-1}) - R_{1C}^{(t_n-t_{n-1})}\mathbf{C}(t_{n-1})) = O(t_n - t_d) = O(h^3),$$

  and this proves that

$$e_n^C = O(h^3).$$

  The same type of manipulations enable us to show that

$$e_n^S = O(h^3).$$

  It is also clear that $e_{n+1}^C = O(h^3)$, and that $e_{n+1}^S = O(h^3)$.

- Case $t_{n-1} < t_n \le t_d < t_{n+1}$:

  The arguments of the proof are similar to those of the previous case.

□

4. Global error estimate and numerical illustration

Theorem 4.1. *For all $\mathbf{C}_0$, $\mathbf{S}_0 \in \mathbb{R}^N$ and all $T > 0$, there exists h_0 such that for all $h \in]0, h_0]$, for all n such that $nh \leq T$*

$$R_C^{nh}\mathbf{C}_0 - (Z_C^h)^n\mathbf{C}_0 = O(h^2), \tag{4.1}$$

$$R_S^{nh}(\mathbf{C}_0, \mathbf{S}_0) - (Z_S^h)^n(\mathbf{C}_0, \mathbf{S}_0) = O(h^2). \tag{4.2}$$

Proof. We only prove (4.1). As noticed in [2] the triangle inequality yields

$$||(Z_C^h)^n\mathbf{C}_0 - R_C^{nh}\mathbf{C}_0|| \leq \sum_{j=0}^{n-1} ||(Z_C^h)^{n-1-j}(Z_C^h)R_C^{jh}\mathbf{C}_0 - (Z_C^h)^{n-1-j}R_C^{(j+1)h}\mathbf{C}_0||.$$

By using the fact that X^t is unitary with respect to the Euclidean norm and that the functions G^i are Lipchitzian with constant L, we refer to [1, p.8] where, for deriving, there exists a constant K such that for $\mathbf{C}_1$ and $\mathbf{C}_2 \in \mathbb{R}^N$ and all $t \in [0, 1]$

$$||Z_C^t\mathbf{C}_1 - Z_C^t\mathbf{C}_2|| \leq (1 + Kt)||\mathbf{C}_1 - \mathbf{C}_2||.$$

Therefore,

$$||(Z_C^h)^n\mathbf{C}_0 - R_C^{nh}\mathbf{C}_0|| \leq \sum_{j=0}^{n-1}(1 + Kh)^{n-1-j}||(Z_C^h - R_C^h)R_C^{jh}\mathbf{C}_0||.$$

Now from Lemma 3.2 we deduce that there exists a constant $\tilde{K}$ such that for all j such that $jh \leq T$

$$||(Z_C^h - R_C^h)R_C^{jh}\mathbf{C}_0|| \leq \tilde{K}h^3,$$

and eventually

$$\begin{aligned} ||(Z_C^h)^n\mathbf{C}_0 - R_C^{nh}\mathbf{C}_0|| &\leq \tilde{K}\sum_{j=0}^{n-1} e^{(n-1-j)Kh}h^3, \\ &\leq \tilde{K}e^{nhK}(nh)h^2, \\ &\leq \tilde{K}(T)h^2. \end{aligned}$$ □

Let us illustrate this result by a numerical experiment with a simple test case. We consider the following system of equations set on the one dimensional domain $(0, 1)$,

$$\left\{ \begin{array}{llll} \partial_t C & = \Delta C & +\alpha C(1 - C) & \text{if } S > S_d, \\ & = \Delta C & +\beta C & \text{if } S \leq S_d, \\ \partial_t S & = & -\alpha C(1 - C) & \text{if } S > S_d, \\ & = & -\beta C & \text{if } S \leq S_d, \end{array} \right. \tag{4.3}$$

where $\alpha = 0.5, \beta = 0.25$, and $S_d = 1$ are constants. Initial and boundary conditions for C are determined by the exact solution

$$C(t, x) = (\frac{1}{1 + \exp(\sqrt{\frac{\alpha}{6}}x - \frac{5}{6}\alpha t)})^2$$

to Fisher's equation

$$\partial_t C = \Delta C + \alpha C(1 - C).$$

Hence,

$$C(0,x) = (\frac{1}{1+\exp(\sqrt{\frac{\alpha}{6}}x)})^2,$$

$$C(t,0) = (\frac{1}{1+\exp(-\frac{5}{6}\alpha t)})^2,$$

and

$$C(t,1) = (\frac{1}{1+\exp(\sqrt{\frac{\alpha}{6}}-\frac{5}{6}\alpha t)})^2.$$

Initial conditions for S are given by

$$S(0,x) = 1+\exp(-(x-1/2)^2).$$

The diffusion operator is discretized using second order finite differences with a step size $\Delta x = 10^{-2}$, and its time integration is performed using the unconditionally stable second order Crank-Nicolson scheme. Reaction terms are integrated with a second order explicit Runge-Kutta scheme. A reference solution is computed for the classical splitting method and for the method proposed in this paper with a time step $h_{\text{ref}} = \frac{0.1}{2^{14}}$.

Figure 1 shows a zoom in on $S(t,x)$ where the discontinuity of the derivative $\partial_t S$ clearly appears when S crosses the surface $S = S_d = 1$.

Solutions are computed using five different time steps, $h = \frac{0.1}{2^9}, \frac{0.1}{2^{10}}, \frac{0.1}{2^{11}}, \frac{0.1}{2^{12}}$ and $h = \frac{0.1}{2^{13}}$. For each solution the global errors

$$E_C = ||\mathbf{C}_h(T) - \mathbf{C}_{h_{\text{ref}}}(T)||,$$

$$E_S = ||\mathbf{S}_h(T) - \mathbf{S}_{h_{\text{ref}}}(T)||,$$

are computed at $T = 0.1$ ($||.||$ denoted the euclidian norm). Figure 2 shows $-\log(E_C)$ and $-\log(E_S)$ versus $-\log(h)$ when the classical splitting method is used to compute the solution to problem (4.3). The convergence curve is very

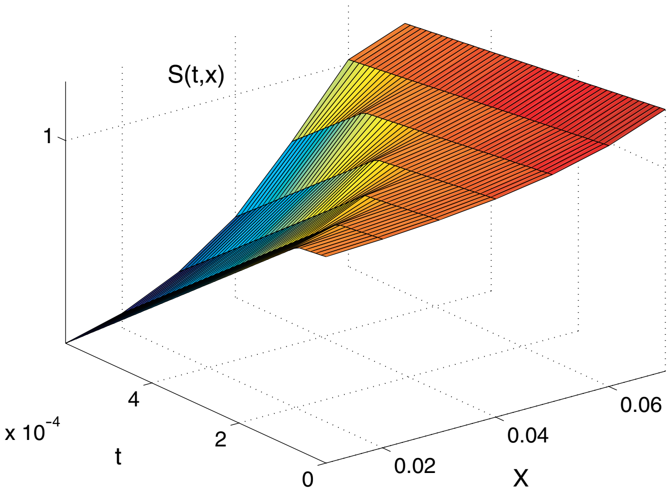

FIGURE 1. Zoom in on $S(t,x)$ crossing the surface $S = 1$, computed with the proposed scheme.

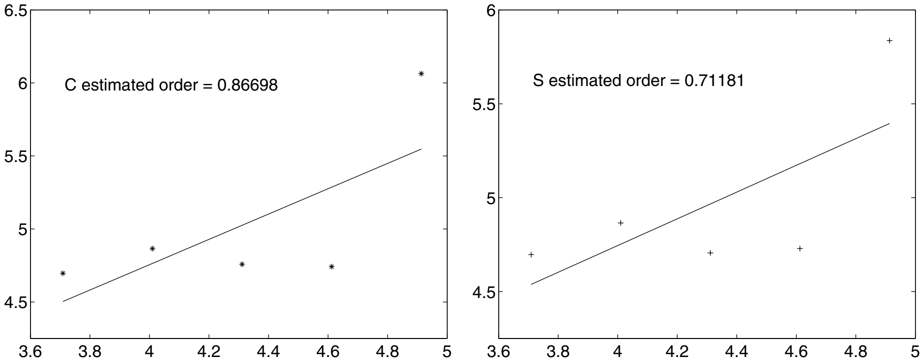

FIGURE 2. $-\log(E)$ versus $-\log(h)$. Convergence curve for the classical splitting (left C and right S).

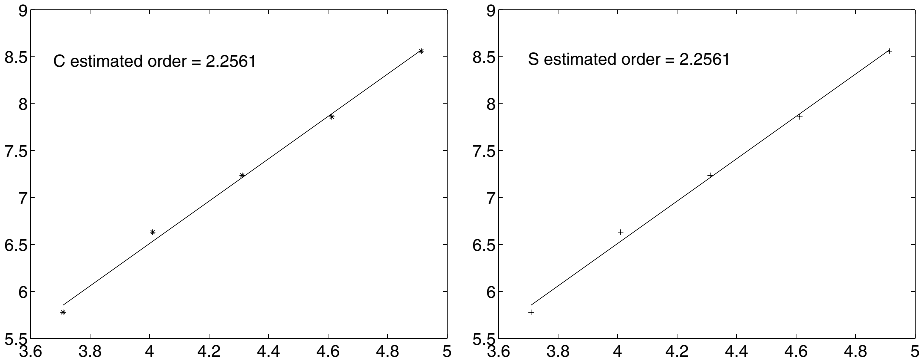

FIGURE 3. $-\log(E)$ versus $-\log(h)$. Convergence curve for the proposed scheme (left C and right S).

perturbed and the estimated order of the scheme is less then 1. This is not surprising since the method is not able to deal with the discontinuities correctly. On the other hand Figure 3 shows $-\log(E_C)$ and $-\log(E_S)$ versus $-\log(h)$ when the method proposed in this paper is used. The estimated order is about 2, which is in agreement with the theoretical result.

REFERENCES

1. C. Besse, B. Bidegaray, and S. Descombes, *Order estimates in time of splitting methods for the nonlinear schrödinger equation*, SIAM J. Numer. Anal. **40** (2002), no. 5, 26–40. MR1921908 (2003k:65099)
2. S. Descombes, *Convergence of a splitting method of high order for reaction-diffusion systems*, Math. Comp. **70** (2001), no. 236, 1484–1501. MR1836914 (2002c:65152)
3. S. Descombes and M. Massot, *Operator splitting for nonlinear reaction-diffusion systems with an entropic structure : singular perturbation and order reduction*, Numerische Mathematik (2004) Vol. 97, No. 4, 667–698.
4. E. Hairer, S.P. Norsett, and G. Wanner, *Solving Ordinary Differential Equations I, Nonstiff Problems*, Springer Series in Computational Mathematics, Springer Verlag, 1993. MR1227985 (94c:65005)

5. E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems*, Springer Series in Computational Mathematics, Springer Verlag, 1993. MR1111480 (92a:65016)
6. C. Lubich and T. Jahnke, *Error bounds for exponential operator splitting*, Technical report, Universitat Tubingen, Germany, 1999; BIT **40** (2000), 735–744. MR1799313 (2001k:65143)
7. E. Maisse, *Analyse et simulations numériques de phénomènes de diffusion-dissolution/précipitation en milieux poreux, appliquées au stockage de déchets*, Ph.D. thesis, Université Claude Bernard Lyon I, 1998.
8. E. Maisse and J. Pousin, *Diffusion and dissolution/precipitation in an open porous reactive medium*, J. Comp. Appl. Math. **82** (1997), 279–280. MR1473546 (98g:35170)
9. G.I. Marchuk, *Splitting and alternating direction methods*, Handbook of numerical analysis, vol. I, North-Holland, Amsterdam, 1990, pp. 197–462. MR1039325
10. M. Schatzman, *Toward non commutative numerical analysis : high order integration in time*, Journal of Scientific Computing **17** (2002), no. 1-3, 107–125. MR1910554
11. L.F. Shampine, *Interpolation for Runge-Kutta methods*, SIAM J. Numer. Anal. **22** (1985), 1014–1027. MR0799125 (86j:65014)
12. B. Sportisse, *An analysis of operator splitting techniques in the stiff case*, J. Comput. Phys. **161** (2000), no. 1, 140–168. MR1762076 (2001f:65107)
13. G. Strang, *On the construction and comparison of difference schemes*, SIAM J. Numer. Anal. **5** (1968), 506–517. MR0235754 (38:4057)

CNRS I35, Les Algorithmes, 2000 Route des Lucioles, BP 121, 06903 Sophia Antipolis cedex France
E-mail address: Blaise.Faugeras@unice.fr

MAPLY, Centre de Mathématique INSA de Lyon, Bat. Léonard de Vinci, 21, Av. Jean Capelle, 69100 Villeurbanne Cedex, France
E-mail address: Jerome.Pousin@insa-lyon.fr

MAPLY, Centre de Mathématique INSA de Lyon, Bat. Léonard de Vinci, 21, Av. Jean Capelle, 69100 Villeurbanne Cedex, France
E-mail address: Franck.Fontvieille@insa-lyon.fr