



HAL
open science

Simulation of a time dependent advection-reaction-diffusion problem using operator splitting and discontinuous Galerkin methods with application to plant root growth

Emilie Peynaud

► **To cite this version:**

Emilie Peynaud. Simulation of a time dependent advection-reaction-diffusion problem using operator splitting and discontinuous Galerkin methods with application to plant root growth. BIOMATH, 2018, 7 (2), pp.1812037. 10.11145/j.biomath.2018.12.037 . hal-01960018

HAL Id: hal-01960018

<https://hal.umontpellier.fr/hal-01960018v1>

Submitted on 19 Dec 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Operator splitting and discontinuous Galerkin methods for advection-reaction-diffusion problem. Application to plant root growth

Emilie Peynaud

CIRAD, UMR AMAP, Yaoundé, Cameroun

AMAP, University of Montpellier, CIRAD, CNRS, INRA, IRD, Montpellier, France

University of Yaoundé 1, National Advanced School of Engineering, Yaoundé, Cameroon

emilie.peynaud@cirad.fr

Received: 19 September 2017, accepted: 3 December 2018, published: 18 December 2018

Abstract—Motivated by the need of developing numerical tools for the simulation of plant root growth, this article deals with the numerical resolution of the C-Root model. This model describes the dynamics of plant root apices in the soil and it consists in a time dependent advection-reaction-diffusion equation whose unique unknown is the density of apices. The work is focused on the implementation and validation of a suitable numerical method for the resolution of the C-Root model on unstructured meshes. The model is solved using Discontinuous Galerkin (DG) finite elements combined with an operator splitting technique. After a brief presentation of the numerical method, the implementation of the algorithm is validated in a simple test case, for which an analytic expression of the solution is known. Then, the issue of the positivity preservation is discussed. Finally, the DG-splitting algorithm is applied to a more realistic root system and the results are discussed.

Keywords—Time dependent advection-reaction-diffusion; Operator splitting; Discontinuous

Galerkin method; Plant root growth simulation;

I. INTRODUCTION

The article is devoted to the numerical modeling of plant root growth. This work has been originally motivated by the need of developing numerical tools for the simulation of plant growth dynamics. Due to the difficulty of doing non-destructive observations of the underground part of plants (that allow to do long term studies of the dynamics of tree roots for example), mathematical models are achieving an essential role. Several theoretical and numerical challenges arise in the field of the simulation of the dynamics of plant roots [48], [47], [38], [2], [39]. The mathematical description of plant root is not trivial, due to the presence of many interactions arising in the rhizosphere and also due to the diversity of plant root types. Mathematical models based on the use of partial differential equations are useful tools to simulate the evolution of root densities in

Copyright: © 2018 Peynaud. This article is distributed under the terms of the Creative Commons Attribution License (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Citation: Emilie Peynaud, Operator splitting and discontinuous Galerkin methods for advection-reaction-diffusion problem. Application to plant root growth, Biomath 7 (2018), 1812037, <http://dx.doi.org/10.11145/j.biomath.2018.12.037>

space and time [43], [44], [44], [45], [46], [41], [40], [1]. This formalism facilitates the coupling with physical models such as water and nutrient transports [42], [43], [44], [41], [49]. And the computational time for the simulation of such models is not dependent on the number of roots which is useful for applications at large scale. The C-Root model [1] is a generic model of the dynamics of root density growth. This model takes only one unknown which is related to root densities such as the density of apices, root length density or biomass density. It has only three parameters. The model is said to be generic in the sense that it can apply to a wide variety of root system types. The model consists in a single time-dependent advection-reaction-diffusion equation, and one of the challenge is to numerically solve the equation. In [1] and [2] the authors solved the problem with the finite difference method on Cartesian mesh grids combined with an operator splitting technique. Unfortunately, Cartesian mesh grids do not allow easily to mesh complex soil geometries. From the theoretical and computational point of view, Cartesian grids also lead to difficulties for a rigorous study and validation of the model. That is why this article focus on the development and implementation of a suitable numerical method for the resolution of the C-Root model on triangular mesh grids, that allow to mesh complex geometries. However, one of the main difficulties in the C-root model is that the advection and diffusion terms are not always of the same order of magnitude. It depends on the phase of the root system development [2]. As a result, the properties of the equation may vary along the simulation: it can be either close to a hyperbolic problem or close to a parabolic problem.

In a previous work [3], the use of the Discontinuous Galerkin method has been implemented and validated. Indeed, the usual choice of the classical Lagrange finite element method suffers from a lack of stability when the advection term is dominant [4]. For this reason, we implemented a discontinuous Galerkin (DG) method for both the advection and diffusion terms. All the three

operators were solved simultaneously using the same time approximation scheme (θ -scheme).

However, as explained in [6], for multi-biophysic problems it is not efficient to use the same numerical scheme for the different operators of the system. For example, we may want to use the Euler explicit scheme for the advection term and an Euler implicit scheme for the diffusion. The operator splitting technique [7], [8] is a well known alternative for the resolution of equations having a multi-biophysic behaviour that allows the use of different time schemes for each operator of the equation. The idea of the splitting technique is to split the problem into smaller and simpler parts of the problem so that each part can be solved by an efficient and suitable time scheme. This method has been used for a wide range of applications dealing with the advection-reaction-diffusion equation [9]. Operator splitting techniques have been extensively used in combination with finite difference methods [10], [2], finite volume methods [11], [12] but also with Continuous Galerkin methods [13], [14], [15], [16], [17]. To the best of my knowledge, only very few articles deal with the use of the operator splitting technique in combination with the discontinuous Galerkin approximation [18], [19], [20], [21]. In this paper, we present a new application of the operator splitting technique combined with discontinuous finite elements.

The paper is structured as follows. In section II, the C-root growth model [1], [2], [3] is briefly described. An analysis is also provided, where I showed the existence and uniqueness of a positive real solution. In section III, the splitting operator technique is introduced and applied to the C-Root model, combined with the use of discontinuous Galerkin approximations. In section IV, the algorithm is implemented and validated using a simple test case for which an analytic expression of the solution is known. As an application, I provide simulations of the development of eucalyptus roots in section V. Finally, the paper ends with a conclusion and further improvements.

II. THE MODEL

A. Modelling root growth with PDE: the C-Root model

The C-Root model [1] was developed to simulate the growth of dense root networks, usually composed of fine roots, with negligible secondary thickening. As presented in [1], the unknown variable u is the number of apices per unit volume, but it can also stand for the density of fine root biomass. The soil is considered as a subdomain of \mathbb{R}^d (with $d = 1, 2$ or 3). It is assumed that Ω has smooth boundaries (Lipschitz boundaries) denoted $\partial\Omega$. The C-Root model combines advection, diffusion and reaction, which aggregate the main biological processes involved in root growth, such as primary growth, ramification and root death. The reaction operator gives the quantity of apices (or root biomass) produced in time, whereas advection and diffusion operators spatially distribute the whole apices (or biomass) in the domain.

The reaction operator describes the evolution in time of the root biomass in a given domain. In the C-Root model it is a linear term characterized by the scalar parameter ρ which is the growth rate of the root system. The diffusion corresponds to the spread of the root biomass over space. It is described by the parameter σ which is a $d \times d$ matrix that characterizes the growth of the root biomass in any direction exploiting free space in the soil. The advection corresponds to the displacement of the root biomass in a direction and velocity given by \mathbf{v} which is a vector in \mathbb{R}^d .

On the boundaries of Ω , what happens for the quantity being transported is different depending if the growth makes the roots to come inside Ω or to go outside of Ω . If \mathbf{v} is going inside Ω (at the inlet boundary) the root biomass u will enter the domain and increase. On edges where \mathbf{v} is going out of the domain (outlet boundary) the root biomass u is going to be pushed out of Ω . Since this phenomena is oriented (causality) and the behaviour of the solution is different on inlet and outlet boundaries, we need to specify in the model these parts of the boundaries. Mathematically, it is required to define the inlet boundary with respect

to \mathbf{v} as

$$\partial\Omega^- = \{\mathbf{x} \in \partial\Omega : (\mathbf{v} \cdot \mathbf{n})(\mathbf{x}) < 0\}. \quad (1)$$

The outlet boundary Ω^+ is given by $\partial\Omega^+ = \partial\Omega \setminus \partial\Omega^-$. The dynamics of the root system is studied between the time t_0 and t_1 with $0 \leq t_0 < t_1$. The problem reads as follow: find u such that

$$\begin{cases} \partial_t u + \mathbf{v} \cdot \nabla u - \nabla \cdot (\sigma \nabla u) + \rho u = 0 & \text{in }]t_0, t_1[\times \Omega \\ u(t_0) = u_0 \text{ at } \{t_0\} \times \Omega \\ \mathbf{n} \cdot \sigma \nabla u = g \text{ on }]t_0, t_1[\times \partial\Omega \\ (\mathbf{n} \cdot \mathbf{v})u = g_{in} \text{ on }]t_0, t_1[\times \partial\Omega^- \end{cases} \quad (2)$$

where $g \in L^2(\partial\Omega)$ and $g_{in} \in L^2(\partial\Omega^-)$ are given. And u_0 is the given initial solution.

Problem (2) is known as the time dependent advection- reaction-diffusion problem and belongs to the class of parabolic partial differential equations. This equation is a model problem that often occurs in fluid mechanics but also in many other applications in life sciences (see for instance [22], [23], [24]).

Depending on the boundary conditions, the problem has different meanings. To simplify the presentation we only consider the Neumann boundary condition combined with an inlet boundary condition at the inlet of the domain. The Neumann condition specifies the value of the normal derivative of the solution at the boundary of the domain. The inlet boundary condition specifies the quantity of u convected by \mathbf{v} that enters in the domain.

B. The weak problem

Since the goal is to solve the problem on unstructured meshes, the spatial operators are approximated using finite element methods. Within this framework, it is classical to write the problem in a variational form. Let us first introduce some functional spaces [50].

- The space $H^1(\Omega)$ defined such that $H^1(\Omega) = \{v \in L^2(\Omega) : \nabla v \in L^2(\Omega)\}$ is a Hilbert space when equipped with the norm $\|\cdot\|_{1,\Omega}$. We recall that $\forall v \in H^1(\Omega), \|v\|_{1,\Omega} = (v, v)_{1,\Omega}$

and the scalar product $(\cdot, \cdot)_{1,\Omega}$ is defined by $\forall v \in H^1(\Omega)$,

$$(u, v)_{1,\Omega} = \int_{\Omega} uv \, dx + \int_{\Omega} \nabla u \cdot \nabla v \, dx.$$

- We denote $L^2(]t_0, t_1[, H)$ the space of H -valued functions whose norm in H is in $L^2(]t_0, t_1[)$. This space is a Hilbert space for the norm

$$\|u\|_{L^2(]t_0, t_1[, H)} = \left(\int_{t_0}^{t_1} \|u(t)\|_H^2 \, dt \right)^{1/2}.$$

- Let $B_0 \subset B_1$ be two reflexive Hilbert spaces with continuous embedding, we denote $\mathcal{W}(B_0, B_1)$ the space of functions $v :]t_0, t_1[\rightarrow B_0$ such that $v \in L^2(]t_0, t_1[, B_0)$ and $d_t v \in L^2(]t_0, t_1[, B_1)$. Equipped with the norm

$$\|u\|_{\mathcal{W}(B_0, B_1)} = \|u\|_{L^2(]t_0, t_1[, B_0)} + \|d_t u\|_{L^2(]t_0, t_1[, B_1)},$$

the space $\mathcal{W}(B_0, B_1)$ is a Hilbert space [25].

Using the previous functional spaces, I now define the problem in the following weak form: Find u in W such that $\forall v \in H$

$$\langle d_t u(t), v(t) \rangle_{H', H} + a(t, u, v) = \ell(t, v) \text{ a.e. } t \in]t_0, t_1[\\ u(t_0) = u_0, \tag{3}$$

where $W = \mathcal{W}(H^1(\Omega), (H^1(\Omega))')$ and $H = H^1(\Omega)$ and

$$\ell(t, v) = \int_{\partial\Omega} g(t) v d\gamma \tag{4}$$

$$a(t, u, v) = a_A(t, u, v) + a_D(t, u, v) + a_R(t, u, v) \tag{5}$$

with

$$a_A(t, u, v) = \int_{\Omega} v(\mathbf{v}(t, \mathbf{x}) \cdot \nabla u) \, dx, \tag{6}$$

$$a_D(t, u, v) = \int_{\Omega} \nabla v \cdot \sigma(t, \mathbf{x}) \cdot \nabla u \, dx, \tag{7}$$

$$a_R(t, u, v) = \int_{\Omega} \rho(t) uv \, dx. \tag{8}$$

One can prove that problems (3) and (2) are equivalent almost everywhere in $]t_0, t_1[\times \Omega$. Let us

assume that there is a constant $\sigma_0 > 0$ such that

$$\forall \xi \in \mathbb{R}^d, \sum_{i,j=1}^d \sigma_{ij} \xi_i \xi_j \geq \sigma_0 \|\xi\|_d^2 \text{ a.e. in } \Omega. \tag{9}$$

In addition, I assume that

$$\inf_{\mathbf{x} \in \Omega} \left(\sigma - \frac{1}{2} (\nabla \cdot \mathbf{v}) \right) > 0 \text{ and } \inf_{\mathbf{x} \in \partial\Omega} (\mathbf{v} \cdot \mathbf{n}) \geq 0. \tag{10}$$

Under assumption (9) and (10), one can prove that the problem is well-posed for sufficiently smooth \mathbf{v} , σ and ρ (see for instance [25]).

C. The positivity preserving property of the solution

In the framework of our applications to the simulation of root biomass densities one of the crucial property of the problem is the preservation of the positivity of the solution along time. For a positive initial solution u_0 , the solution of (3) stays positive.

Proposition II-C.1. *Let $u_0 \in L^2(\Omega)$ and $f \in L^2(]t_0, t_1[, L^2(\Omega))$. We consider u the solution of (3) in W . We assume that $u_0(\mathbf{x}) \geq 0$ a.e. in Ω and $g(t, \mathbf{x}) \geq 0$ a.e. in $]t_0, t_1[\times \partial\Omega$. Then $u(t, \mathbf{x}) \geq 0$ a.e. in $]t_0, t_1[\times \Omega$.*

Proof: I follow [25]. See also [26], [27]. We consider the function u^- defined by

$$u^- = \frac{1}{2} (|u| - u).$$

Let us note that

$$u^- = \begin{cases} 0 \text{ a.e. in }]t_0, t_1[\times \Omega, & \text{if } u \geq 0 \text{ a.e. in }]t_0, t_1[\times \Omega, \\ -u \text{ a.e. in }]t_0, t_1[\times \Omega, & \text{if } u < 0 \text{ a.e. in }]t_0, t_1[\times \Omega. \end{cases}$$

That is we have

$$u^- \geq 0 \text{ a.e. in }]t_0, t_1[\times \Omega. \tag{11}$$

We verify that u^- is an admissible test function in W . Using the following obvious equations

$$\begin{aligned} (\nabla |u|)^2 &= (\nabla u)^2 \\ \nabla |u| \cdot \nabla |u| &= \nabla u \cdot \nabla u \\ u \nabla |u| &= |u| \nabla u \\ u \nabla u &= |u| \nabla |u| \end{aligned}$$

that are valid a.e in $]t_0, t_1[\times \Omega$ we can verify that

$$a(t, u^-, u^-) = -a(t, u, u^-).$$

By adding the same quantity on both sides of the equation we get

$$\langle d_t u^-, u^- \rangle + a(t, u^-, u^-) = \langle d_t u^-, u^- \rangle - a(t, u, u^-).$$

Since u satisfy (3) we have

$$\begin{aligned} \langle d_t u^-, u^- \rangle + a(t, u^-, u^-) &= \langle d_t u^-, u^- \rangle + \langle d_t u, u^- \rangle \\ &\quad - \ell(t, u^-). \end{aligned}$$

One can notice that $\langle d_t u^-, u^- \rangle + \langle d_t u, u^- \rangle = 0$. Then we have

$$\frac{1}{2} d_t \|u^-\|_{0,\Omega}^2 + a(t, u^-, u^-) = -\ell(t, u^-) \leq 0,$$

with $g(t, \mathbf{x}) \geq 0$ a.e. in $]t_0, t_1[\times \partial\Omega$. Now from the coercivity of the bilinear form a we obtain

$$\begin{aligned} \frac{1}{2} d_t \|u^-\|_{0,\Omega}^2 + c \|u^-\|_{0,\Omega}^2 \\ \leq \frac{1}{2} d_t \|u^-\|_{0,\Omega}^2 + a(t; u^-, u^-) \leq 0, \end{aligned}$$

where c is a strictly positive constant. The estimate is then

$$\frac{1}{2} d_t \|u^-\|_{0,\Omega}^2 \leq -c \|u^-\|_{0,\Omega}^2.$$

By the Gronwall lemma we have that

$$\forall t \in [t_0, t_1] \times \Omega, \|u^-(t)\|_{0,\Omega}^2 \leq e^{-2ct} \|u^-(0)\|_{0,\Omega}^2.$$

Since $c > 0$ and $t \geq t_0 \geq 0$, we have that $e^{-2ct} \leq 1$, so we obtain

$$\forall t \in [t_0, t_1] \times \Omega, \|u^-(t)\|_{0,\Omega}^2 \leq \|u^-(0)\|_{0,\Omega}^2.$$

Since $u(0) = u_0 \geq 0$ a.e in Ω we have $u^-(0) = 0$ a.e in Ω . So we deduce that

$$\forall t \in [t_0, t_1] \times \Omega, \|u^-(t)\|_{0,\Omega}^2 \leq 0.$$

But from the definition of u^- we have $u^- \geq 0$ a.e in $]t_0, t_1[\times \Omega$. So we deduce that $\|u^-(t)\|_{0,\Omega}^2 = 0$ and thus $u^-(t) = 0$ a.e in $]t_0, t_1[\times \Omega$. It means that $u \geq 0$ a.e in $]t_0, t_1[\times \Omega$ by definition of u^- . ■

III. APPROXIMATION OF THE MODEL

A. The operator splitting technique

Here we focus on the implementation of the operator splitting technique. The time interval $[t_0, t_1]$ is divided in N subspaces of size δt such that $[t_0, t_1] = \cup_{n=1, N}]t_n, t_{n+1}[$ with $\cap_{n=1, N}]t_n, t_{n+1}[= \emptyset$. At each iteration step we solve the following problems

- Find $u_A \in H$ such that $\forall v \in H$, for a.e $t \in]t_n, t_{n+1}[$,

$$\begin{aligned} \langle d_t u_A(t), v(t) \rangle_{H', H} + a_A(t, u_A, v) &= 0 \\ u_A(t_n) &= u(t_n). \end{aligned}$$
- Find $u_D \in H$ such that $\forall v \in H$, for a.e $t \in]t_n, t_{n+1}[$,

$$\begin{aligned} \langle d_t u_D(t), v(t) \rangle_{H', H} + a_D(t, u_D, v) &= \ell(t, v) \\ u_D(t_n) &= u_A(t_{n+1}). \end{aligned}$$
- Find $u_R \in H$ such that $\forall v \in H$, for a.e $t \in]t_n, t_{n+1}[$,

$$\begin{aligned} \langle d_t u_R(t), v(t) \rangle_{H', H} + a_R(t, u_R, v) &= 0 \\ u_R(t_n) &= u_D(t_{n+1}). \end{aligned}$$
- Set $u(t_{n+1}) = u_R(t_{n+1})$.

The bilinear forms $a_A(t, u, v)$, $a_D(t, u, v)$ and $a_R(t, u, v)$ are respectively given by (6), (7) and (8). And $\ell(t, v)$ is the linear form (4). If the operators are commutative, then the splitting error vanishes. Otherwise, if the operators are not commutative, then the splitting error does not vanish and a second order splitting would be required (see [6]). In the following, I present the different schemes related to each operator.

B. The advection step: DG upwind scheme

The advection step consists in solving the following transport problem : Find u such that $\forall v \in H$, for a.e $t \in]t_n, t_{n+1}[$,

$$\langle d_t u(t), v(t) \rangle_{H', H} + a_A(t, u, v) = 0 \quad (12)$$

$$u_A(t_n) = u_R(t_n) \quad (13)$$

where $a_A(t, u, v)$ is the bilinear form (6). For the space approximation of this problem, we implemented the DG upwind method presented below.

Let \mathcal{T}_h be a regular family of decomposition in triangles of the domain Ω such that

$$\Omega = \bigcup_{i=1}^N \bar{K}_i \text{ and } K_i \cap K_j = \emptyset, \forall i \neq j.$$

The h subscript in \mathcal{T}_h denotes the size of the mesh cells and it is defined by

$$h = \max_{K \in \mathcal{T}_h} h_K$$

where h_K is the diameter of the element K . Let \mathcal{E}_h be the set of edges of the elements of \mathcal{T}_h . Among the elements of \mathcal{E}_h we denote by \mathcal{E}_h^b the set of edges belonging to $\partial\Omega$. The sets $\mathcal{E}_h^{b,-}$ and $\mathcal{E}_h^{b,+}$ are the sets of edges belonging to $\partial\Omega^-$ and $\partial\Omega^+$ respectively. And \mathcal{E}_h^i is the set of interior edges. Let us consider an element of \mathcal{E}_h^i . We denote by T^+ and T^- the two mesh elements sharing the edge e so that $e = \partial T^+ \cap \partial T^-$ where the minus and plus superscripts depend on the direction of the advection vector. By convention we suppose that \mathbf{v} goes from T^- to T^+ that is $\mathbf{v} \cdot \mathbf{n}_e^+ < 0$ and $\mathbf{v} \cdot \mathbf{n}_e^- > 0$ where \mathbf{n}_e^+ (resp. \mathbf{n}_e^-) is the outward normal vector of e in T^+ (resp. T^-). When it is not necessary to distinguish the orientation of the normal vectors \mathbf{n}_e^+ and \mathbf{n}_e^- we denote by \mathbf{n} the unitary normal of e .

Let us consider the advection problem on each element K_i of the domain : for all $K_i, i = 1, N$ we look for u the solution of the equation (12) defined on K_i . Similarly to the problem defined on all the domain Ω , we look for a solution u that is in $L^2(K_i)$ and such that ∇u is in $L^2(K_i)$ for all K_i in \mathcal{T}_h . Let us introduce the following broken Sobolev space:

$$H^1(\mathcal{T}_h) = \left\{ v \in L^2(\Omega) : \nabla v \in L^2(K_i) \text{ and } v \in H^{1/2+\varepsilon}(K_i), \forall K_i \in \mathcal{T}_h \right\}$$

with ε a positive real number. The trace of the functions of $H^1(\mathcal{T}_h)$ are meaningful on $e \subset K_i, \forall K_i \in \mathcal{T}_h$. The functions v of $H^1(\mathcal{T}_h)$ have two traces along the edges e . We denote v_e^+ the trace of v along e on the side of triangle T^+ and v_e^- the trace of v along e on the side of T^- . On edges

that are subsets of $\partial\Omega$ the trace is unique and we can note

$$v_e^+ = v \text{ if } e \in \mathcal{E}_h^{b,-} \text{ and } v_e^- = v \text{ if } e \in \mathcal{E}_h^{b,+},$$

and by convention, we set

$$v_e^- = 0 \text{ if } e \in \mathcal{E}_h^{b,-} \text{ and } v_e^+ = 0 \text{ if } e \in \mathcal{E}_h^{b,+}.$$

The jump of functions of $H^1(\mathcal{T}_h)$ across the internal edge e is defined by:

$$[[v]] = v_e^+ - v_e^-, \forall e \in \mathcal{E}_h^i.$$

For edges belonging to the boundary of Ω we take

$$[[v]] = v_e, \forall e \in \mathcal{E}_h^{b,-} \text{ and } [[v]] = -v_e, \forall e \in \mathcal{E}_h^{b,+},$$

with v_e the trace of v along e . The mean value of u on e is defined by

$$\{\{v\}\} = \frac{1}{2}(v_e^+ + v_e^-), \forall e \in \mathcal{E}_h^i.$$

Besides for edges on the boundaries we take

$$\{\{v\}\} = v_e, \forall e \in \mathcal{E}_h^b.$$

Let us denote by \mathcal{X} the functional space defined such that

$$\mathcal{X} = \{v :]t_0, t_1[\rightarrow H^1(\mathcal{T}_h) : v \in L^2(]t_0, t_1[, H^1(\mathcal{T}_h)); \text{ and } d_t v \in L^2(]t_0, t_1[, H^1(\mathcal{T}_h)')\}.$$

This space is a Hilbert space equipped with the norm

$$\|v\|_{\mathcal{X}} = \|v\|_{L^2(]t_0, t_1[, H^1(\mathcal{T}_h))} + \|v\|_{L^2(]t_0, t_1[, H^1(\mathcal{T}_h)')}.$$

The DG variational formulation of the advection step written on the broken Sobolev space takes the following form: Find u in \mathcal{X} such that for a.e $t \in]t_0, t_1[, \forall v \in H^1(\mathcal{T}_h)$

$$\langle d_t u(t), v(t) \rangle_{H^1(\mathcal{T}_h), H^1(\mathcal{T}_h)} + a_h^{up}(t; u, v) = \ell_h^{up}(t; v),$$

$$u(t_0) = u_0,$$

where the form $a_h^{up}(t; u, v)$ is the approximation of the advection term. It consists in the upwind formulation of the DG method [28]. It reads:

$$a_h^{up}(t; u, v) = \sum_{K \in \mathcal{T}_h} \int_K u(\rho v - \mathbf{v} \cdot \nabla v) dx - \sum_{e \in \mathcal{E}_h^{b,\pm} \cup \mathcal{E}_h^{b,+}} \int_e |\mathbf{v} \cdot \mathbf{n}_e^+| u_e^- [[v]] ds. \quad (14)$$

The approximated linear form of the the right hand side reads

$$\ell_h^{up}(t; v) = - \sum_{e \in \mathcal{E}_h^{b,-}} \int_e (\mathbf{v} \cdot \mathbf{n}_e^+) g_{in} v_e^+ ds.$$

The DG-formulation (14) is consistent and stable, see for example [32]. The discontinuous Galerkin method consists in searching the solution in the approximation space \mathcal{X}_h defined such that

$$\mathcal{X}_h = \left\{ v :]t_0, t_1[\longrightarrow W_h^k; v \in L^2(]t_0, t_1[, W_h^k); \text{ and } d_t v \in L^2(]t_0, t_1[, (W_h^k)') \right\},$$

where W_h^k is given by

$$W_h^k = \left\{ v_h \in L^2(\Omega); \forall K \in \mathcal{T}_h, v_h|_K \in \mathbb{P}^k \right\}.$$

Let us note that the functions of W_h^k can be discontinuous from one element of the mesh to the other. Let us note that W_h^k is embedded in $H^1(\mathcal{T}_h)$ so that $\mathcal{X}_h \subset \mathcal{X}$. This problem can be written in a matrix form. Let us denote $(\lambda_i)_{i=1,n}$ the basis of the finite dimensional subspace W_h^k where $n = \dim(W_h^k)$. In this basis the approximated solution takes the form:

$$u_h(t, x, y) = \sum_{i=1}^n \xi_i(t) \lambda_i(x, y),$$

where the $\xi_i(t)$ are the degrees of freedom. Let us define X the vector of degrees of freedom:

$$X(t) = (\xi_1(t), \dots, \xi_n(t))^T.$$

The approximated problem then reduces to find $X(t) \in [\mathcal{C}^2(0, T)]^n$ such that

$$\begin{aligned} \mathbf{M} \frac{dX(t)}{dt} + \mathbf{A}^{up}(t) X(t) &= L_h^{up}(t) \\ \mathbf{M} X(0) &= \mathbf{M} X_0 \end{aligned}$$

where \mathbf{M} and $\mathbf{A}^{up}(t)$ are two matrices defined such that

$$\mathbf{M} = (M_{i,j})_{i,j} \text{ and } M_{i,j} = \sum_{T \in \mathcal{T}_h} \int_K \lambda_i \lambda_j dx, \tag{15}$$

$$\mathbf{A}^{up} = (A_{i,j}^{up})_{i,j} \text{ and } A_{i,j}^{up} = a_h^{up}(t; \lambda_i, \lambda_j), \tag{16}$$

and $L_h^{up}(t)$ is the vector of size n defined such that $(L_h^{up}(t))_i = \ell_h^{up}(t; \lambda_i)$ for $i = 1, n$. The problem reduces to a linear system of ordinary differential equations. The time approximation is based on a finite difference scheme.

At each iteration step we solve the following problem: Find $X^{N+1} \in \mathbb{R}^n$ such that

$$\begin{aligned} \frac{1}{\delta t} \mathbf{M} (X^{N+1} - X^N) &+ (1 - \theta) \mathbf{A}^{up} X^N + \theta \mathbf{A}^{up} X^{N+1} \\ &= (1 - \theta) L_h^{up,N} + \theta L_h^{up,N+1} \end{aligned} \tag{17}$$

and $\mathbf{M} X^0 = \mathbf{M} X_0$,

where θ is a real parameter taken in $[0, 1]$. For $\theta = 0$, we have the explicit Euler schema. For $\theta = 1$, it is the implicit Euler schema. For $\theta = 1/2$, it is the Crank-Nicolson schema.

C. The diffusion step

The diffusion step consists in solving the following problem : Find u such that $\forall v \in H$, for a.e $t \in]t_n, t_{n+1}[$,

$$\begin{aligned} \langle d_t u(t), v(t) \rangle_{H', H} + a_D(t; u, v) &= \ell(t; v) \\ u(t_n) &= u_A(t_n) \end{aligned}$$

where $a_D(t; u, v)$ is the bilinear form (7) and $\ell(t; v)$ is the linear form (4). In the setting introduced before, the DG variational formulation of the diffusion step written in the broken Sobolev space takes the following form: Find u in \mathcal{X} such that $\forall v \in H^1(\mathcal{T}_h)$, for a.e. $t \in]t_0, t_1[$

$$\begin{aligned} \langle d_t u(t), v \rangle_{H^1(\mathcal{T}_h)', H^1(\mathcal{T}_h)} + a_h^{ip}(t; u, v) &= \ell_h^{ip}(t; v) \\ u(t_0) &= u_0. \end{aligned}$$

The form $a_h^{ip}(t; u, v)$ is the approximation of the diffusion term. It consists in the interior penalty

formulation (IP) that reads

$$\begin{aligned}
 a_h^{ip}(t; u, v) &= \sum_{K \in \mathcal{T}_h} \int_K \sigma \nabla u \cdot \nabla v \, dx \\
 &\quad - \sum_{e \in \mathcal{E}_h^i} \int_e \{ \sigma \nabla u \} \cdot \mathbf{n}_e^+ [v] \, ds \\
 &\quad + \sum_{e \in \mathcal{E}_h^i} \int_e \{ \sigma \nabla v \} \cdot \mathbf{n}_e^+ [u] \, ds \\
 &\quad + \sum_{e \in \mathcal{E}_h^i} \frac{\eta}{h_e} \int_e [u][v] \, ds,
 \end{aligned}$$

where η is a positive penalization factor. The linear form $\ell_h^{ip}(t; v)$ is given by $\ell_h^{ip}(t; v) = \sum_{e \in \mathcal{E}_h^b} \int_e gv \, ds$.

This formulation was introduced in [31] and is known as the non-symmetric interior penalty (NSIP) formulation, see [30], [32]. In matrix form the problem reduces to find $X(t) \in [C^2(0, T)]^n$ such that

$$\begin{aligned}
 \mathbf{M} \frac{dX(t)}{dt} + \mathbf{A}^{ip}(t)X(t) &= L_h^{ip}(t) \\
 \mathbf{M}X(0) &= \mathbf{M}X_0
 \end{aligned}$$

where \mathbf{M} is defined by (15) and \mathbf{A}^{ip} is defined such that

$$\mathbf{A}^{ip} = \left(A_{i,j}^{ip} \right)_{i,j} \quad \text{and} \quad A_{i,j}^{up} = a_h^{ip}(t; u, v).$$

The vector $L_h^{ip}(t)$ is such that $\left(L_h^{ip}(t) \right)_i = \ell_h^{ip}(t; \lambda_i)$ for $i = 1, n$. Similarly to the advection step, the time approximation of the problem is based on a finite difference scheme of the form (17).

D. The reaction step

The reaction step consists in solving the following problem : Find u such that $\forall v \in H$, for a.e. $t \in]t_n, t_{n+1}[$

$$\begin{aligned}
 \langle d_t u(t), v(t) \rangle_{H', H} + a_R(t; u, v) &= 0 \\
 u(t_n) &= u_D(t_n)
 \end{aligned}$$

where $a_R(t; u, v)$ is the bilinear form (8). This problem takes the following matrix form find

$X(t) \in [C^2(0, T)]^n$ such that

$$\begin{aligned}
 \frac{dX(t)}{dt} + \rho X(t) &= 0 \\
 X(0) &= X_0
 \end{aligned}$$

where we recall that ρ is a constant real parameter. This problem can be solved by an exact scheme (a kind of schemes that provide exact solutions, i.e. a solution equal to the analytical solution). At each iteration we find X^{N+1} such that

$$\frac{1}{\Phi(\delta t)} (X^{N+1} - X^N) = -\rho X^N$$

with $\Phi(\delta t) = \frac{1}{\rho}(1 - \exp(-\rho\delta t))$. This scheme is unconditionally stable, meaning that we can choose the time step independently from the space step. It is also positively stable, meaning that if $X^N \geq 0$ so is X^{N+1} .

IV. VALIDATION OF THE SPLITTING ALGORITHM WITH A SIMPLE TEST CASE

Problem (3) has been already solved using discontinuous Galerkin elements (DG) [3]. Advection and Diffusion operators were solved simultaneously using the Crank-Nicolson scheme providing stable results. However, even for simple test cases some simulations did not always provide positive numerical solutions. One reason is that the same time approximation scheme is not necessarily suitable for both the advection and for the diffusion. That is why a new operator splitting algorithm has been implemented with a different time scheme for each operator.

The goal of this section is to validate the implementation of the code. To this end I compare the convergence of the approximation with and without the splitting technique. I briefly explore the question of the positivity of the approximated solution.

A. Description of the simple test-case

First let me introduce a simplified test-case for the validation of the splitting algorithm. Set $L > 0$, and $\Omega =]-L; L[^2$. Let $\mathbf{v} = (v_1, v_2) \in \mathbb{R}^2$ and

$d \in \mathbb{R}$ be a constant and $0 \leq t_0 \leq t_1$. Find u such that

$$\begin{aligned} \frac{\partial u}{\partial t} + \mathbf{v} \cdot \nabla u + \rho u &= d\Delta u \text{ in }]t_0, t_1[\times \Omega, \\ u(x, y, 0) &= u_0(x, y) \text{ on } \{t_0\} \times \Omega, \\ \mathbf{n} \cdot \nabla u &= g \text{ on }]t_0, t_1[\times \partial\Omega, \\ \mathbf{n} \cdot \mathbf{v} u &= g_{in} \text{ on }]t_0, t_1[\times \partial\Omega^-. \end{aligned} \tag{18}$$

The initial condition and the boundary condition are chosen such that the solution of problem (18) is explicitly given by $\forall (x, y, t) \in \Omega \times]t_0, t_1[$

$$u(x, y, t) = c_0 \left(\frac{a^2}{a^2 + td} \right) \kappa(x, y, t) e^{-\rho t}.$$

with

$$\begin{aligned} \kappa(x, y, t) &= c_0 \exp \left(- \frac{(x - x_0 - tv_1)^2 + (y - y_0 - tv_2)^2}{4(a^2 + td)} \right) \end{aligned}$$

where $c_0 > 0$, $a > 0$, x_0 and y_0 are real parameters and v_1 and v_2 are the two components of \mathbf{v} . Notice that $u(x, y, t) > 0$ for all (x, y, t) in $\Omega \times]t_0, t_1[$.

B. Numerical validation and convergence

To validate the implementation of the splitting technique, I ran the previous test case with different mesh sizes and time steps and I computed the global L^2 -errors such that

$$e_h = \left(\delta t \sum_{k=1}^N \|u(t_k) - u_h(t_k)\|_{0,\Omega}^2 \right)^{1/2}$$

where $t_k = t_0 + k\delta t$, with $k \in \mathbb{N}_*^+$ and $t_N = t_1$.

The flexibility of the splitting technique allows to choose different time schemes for each operator. I consider a θ -scheme with $\theta = 0$ (explicit Euler), $\theta = 1$ (implicit Euler), and $\theta = \frac{1}{2}$ (Crank-Nicolson) for both the advection step and the diffusion step, and I consider an exact scheme for the reaction step. For the simulations I took the parameters such that $\mathbf{v} = (0.1, 0)^T$, $\sigma = 0.01$ and $\rho = -1$. The triangular meshes used for the simulations are identified by h which is the size of the biggest triangle of the mesh. Table I, page 9, gives the number of triangles and the number of nodes of each mesh used for the simulations. Choosing

h (\approx)	number of triangles	number of nodes
2.63×10^{-1}	68	45
1.31×10^{-1}	272	157
6.57×10^{-2}	1 088	585
3.29×10^{-2}	4 352	2 257
1.64×10^{-2}	17 408	8 865
8.22×10^{-3}	69 632	35 137
4.11×10^{-3}	278 528	139 905

TABLE I
TRIANGULAR MESHES USED FOR THE SIMULATIONS.

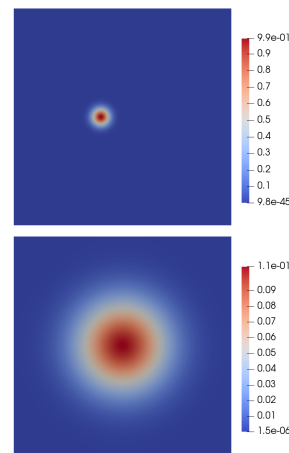


Fig. 1. Solution of the validation test case at $t = t_0$ (left) and $t = t_1$ (right) computed using the DG method with p^1 -finite elements and the Euler implicit scheme ($\theta = 1$) and the operator splitting technique with $h \approx 8.2 \times 10^{-3}$ and $\delta t = 10^{-2}$.

$L = 1/2$, the simulations are performed between $t_0 = 0$ and $t_1 = 1$ for different values of the time step δt . Fig. 1, page 9, shows the solution at $t = t_0$ and $t = t_1$. The code is implemented in Fortran 90 and it is run under a 64-bit Linux operating system on a 8-core processor Intel®Core™i7-7820HQ at a frequency of 2.9GHz and with 32 GB of RAM. The sparse matrices resulting from the finite element approximation are inverted using a solver provided by the library MUMPS [51], [52].

According to Fig. 2, page 10, all the three temporal schemes provide results with approximately the same level of accuracy with a spatial convergence rate of 2 computed with the global L^2 -

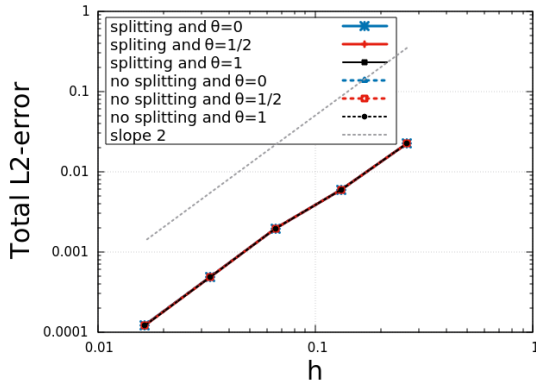


Fig. 2. Convergence of the solution with respect to the mesh size: plot of the total L^2 -error computed between $t = 0$ and $t = 1$ with and without the splitting technique for the explicit Euler scheme ($\theta = 0$), the implicit Euler scheme ($\theta = 1$) and the Crank-Nicolson scheme ($\theta = 1/2$) for $\delta t = 5 \times 10^{-5}$.

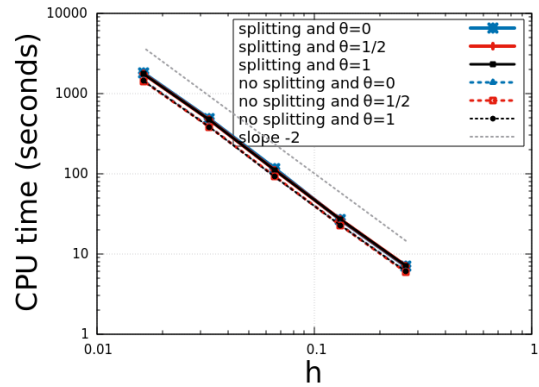


Fig. 4. Validation of the test case: plot of the CPU time against the mesh size (h) for the computations performed with a processor Intel®Core™i7-7820HQ at 2.9 GHz and RAM 32 GB, between $t = 0$ and $t = 1$ with and without the splitting technique for the explicit Euler scheme ($\theta = 0$), the implicit Euler scheme ($\theta = 1$) and the Crank-Nicolson scheme ($\theta = 1/2$) for $\delta t = 5 \times 10^{-5}$.

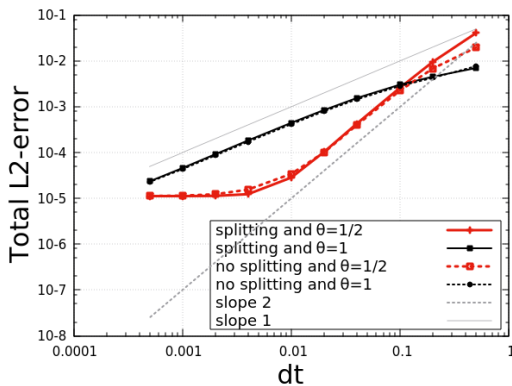


Fig. 3. Convergence of the solution with respect to the time step: plot of the total L^2 -error computed between $t = 0$ and $t = 1$ with and without the splitting technique for the implicit Euler scheme ($\theta = 1$) and the Crank-Nicolson scheme ($\theta = 1/2$) for $h = 4.1 \times 10^{-3}$.

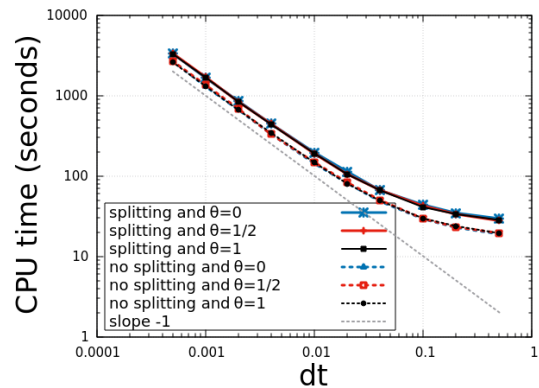


Fig. 5. Validation of the test case: plot of the CPU time against the time step (δt) for the computations performed with a processor Intel®Core™i7-7820HQ at 2.9 GHz and RAM 32 GB, between $t = 0$ and $t = 1$ with and without the splitting technique for the explicit Euler scheme ($\theta = 0$), the implicit Euler scheme ($\theta = 1$) and the Crank-Nicolson scheme ($\theta = 1/2$) for $h \approx 4.1 \times 10^{-3}$ and δt ranging from 5×10^{-1} to 5×10^{-4} . Note that the computations performed here with $\theta = 0$ gave unstable results.

norm. The same order of convergence is obtained when the problem is solved without the splitting technique.

Figure 3 on page 10 shows that the Crank-Nicolson scheme ($\theta = 1/2$) converges in δt^2 while the Euler Implicit scheme ($\theta = 1$) converges in δt , with and without the splitting technique. The convergence rate in time has to be computed with a really refined mesh grid (here $h \approx 4.1 \times 10^{-3}$).

δt	global L^2 -errors	min dof	t_+	CPU time
$1 \cdot 10^{-3}$	unstable	unstable	-	95 s
$2 \cdot 10^{-4}$	$1.24 \cdot 10^{-4}$	$-1 \cdot 10^{-4}$	0.221	475 s
$1 \cdot 10^{-4}$	$1.22 \cdot 10^{-4}$	$-1 \cdot 10^{-4}$	0.218	838 s
$5 \cdot 10^{-5}$	$1.21 \cdot 10^{-4}$	$-1 \cdot 10^{-4}$	0.216	1672 s
$2.5 \cdot 10^{-5}$	$1.20 \cdot 10^{-4}$	$-9 \cdot 10^{-5}$	0.215	3376 s
$1 \cdot 10^{-5}$	$1.20 \cdot 10^{-4}$	$-9 \cdot 10^{-5}$	0.215	10183 s

TABLE II

COMPUTATIONS PERFORMED WITH THE SPLITTING TECHNIQUE AND THE EXPLICIT EULER SCHEME ($\theta = 0$) WITH $h \approx 1.6 \cdot 10^{-2}$ (INTEL®CORE™I7-7820HQ AT 2.9 GHZ, RAM 32 GB).

It results an additional cost in term of CPU time, since it behaves like $1/h^2$, as shown on figure 4 page 10. For bigger values of h the plot of the errors gave convergence order in time less than 1 and 2 for the implicit Euler scheme and the Crank-Nicolson scheme respectively. As expected, the explicit Euler scheme is conditionally stable, such that, when the CFL condition is fulfilled, the computational time becomes prohibitive. Indeed, it behaves like $1/\delta t$, as shown on figure 5. For instance, the computation with $h \approx 4.1 \times 10^{-3}$ and $\delta t = 10^{-5}$ takes more than 30 hours with the device specified above. That is why, in the rest of the paper, we will only focus on implicit Euler and Crank-Nicolson schemes. However I present here additional computations performed with a bigger mesh size ($h \approx 1.6 \times 10^{-2}$) and smaller time steps chosen such that the CFL condition is fulfilled. The global L^2 -errors and the CPU time are shown on table II. Clearly, the mesh is not fine enough to recover the convergence order in δt , indeed decreasing the time step results only in an increase of the computational time but not in a significant decrease of the errors.

C. Some comments on the positivity

1) Positivity of the full problem: Table III on page 11 and table V on page 12 give the minimum values of the degrees of freedom (dof) obtained during the simulations performed respectively with and without the splitting technique. The minimum value of the dof is defined such that $\min_{t_k}(\min_{i=1,n} X_i^k)$ where X_i^k is the i^{th} dof at time t_k . This quantity gives an idea about the

δt	$h \approx 1.6 \cdot 10^{-2}$	$h \approx 8.2 \cdot 10^{-3}$	$h \approx 4.1 \cdot 10^{-3}$
$5 \cdot 10^{-1}$	$-7 \cdot 10^{-1}$	$-7 \cdot 10^{-1}$	$-7 \cdot 10^{-1}$
$2 \cdot 10^{-1}$	$-2 \cdot 10^{-1}$	$-2 \cdot 10^{-1}$	$-2 \cdot 10^{-1}$
$1 \cdot 10^{-1}$	$-8 \cdot 10^{-4}$	$-1 \cdot 10^{-3}$	$-2 \cdot 10^{-3}$
$4 \cdot 10^{-2}$	$-2 \cdot 10^{-4}$	$-6 \cdot 10^{-4}$	$-1 \cdot 10^{-3}$
$2 \cdot 10^{-2}$	$-3 \cdot 10^{-13}$	$-2 \cdot 10^{-4}$	$-6 \cdot 10^{-4}$
$1 \cdot 10^{-2}$	$-6 \cdot 10^{-5}$	$-2 \cdot 10^{-20}$	$-1 \cdot 10^{-4}$
$4 \cdot 10^{-3}$	$-3 \cdot 10^{-4}$	$4 \cdot 10^{-65}$	$5 \cdot 10^{-66}$
$2 \cdot 10^{-3}$	$-3 \cdot 10^{-4}$	$-9 \cdot 10^{-11}$	$5 \cdot 10^{-88}$
$1 \cdot 10^{-3}$	$-2 \cdot 10^{-4}$	$-8 \cdot 10^{-10}$	$8 \cdot 10^{-114}$
$1 \cdot 10^{-4}$	$-9 \cdot 10^{-5}$	$-1 \cdot 10^{-11}$	$-4 \cdot 10^{-35}$

Crank-Nicolson scheme ($\theta = 1/2$)

δt	$h \approx 1.6 \cdot 10^{-2}$	$h \approx 8.2 \cdot 10^{-3}$	$h \approx 4.1 \cdot 10^{-3}$
$5 \cdot 10^{-1}$	$-5 \cdot 10^{-8}$	$-9 \cdot 10^{-9}$	$-2 \cdot 10^{-9}$
$2 \cdot 10^{-1}$	$4 \cdot 10^{-9}$	$1 \cdot 10^{-9}$	$4 \cdot 10^{-10}$
$1 \cdot 10^{-1}$	$3 \cdot 10^{-11}$	$3 \cdot 10^{-11}$	$1 \cdot 10^{-11}$
$4 \cdot 10^{-2}$	$6 \cdot 10^{-17}$	$5 \cdot 10^{-17}$	$5 \cdot 10^{-17}$
$2 \cdot 10^{-2}$	$3 \cdot 10^{-23}$	$2 \cdot 10^{-23}$	$2 \cdot 10^{-23}$
$1 \cdot 10^{-2}$	$1 \cdot 10^{-31}$	$4 \cdot 10^{-32}$	$3 \cdot 10^{-32}$
$4 \cdot 10^{-3}$	$-3 \cdot 10^{-5}$	$1 \cdot 10^{-48}$	$6 \cdot 10^{-49}$
$2 \cdot 10^{-3}$	$-5 \cdot 10^{-5}$	$2 \cdot 10^{-65}$	$2 \cdot 10^{-66}$
$1 \cdot 10^{-3}$	$-7 \cdot 10^{-5}$	$-1 \cdot 10^{-13}$	$2 \cdot 10^{-88}$
$1 \cdot 10^{-4}$	$-9 \cdot 10^{-5}$	$-7 \cdot 10^{-12}$	$-3 \cdot 10^{-43}$

Implicit Euler scheme ($\theta = 1$)

TABLE III

MINIMUM VALUE OF THE DOF ($\min_{i,k} X_i^k$) COMPUTED WITH THE SPLITTING ALGORITHM WITH THE CRANK-NICOLSON SCHEME (TOP) AND THE IMPLICIT EULER SCHEME (BOTTOM).

stability and the positivity preserving behaviour of the schemes. Tables III and V clearly show that the schemes are not always positivity preserving. In case where the approximated solution is not positive for all $t > t_0$ I also check if it becomes non-negative for larger time ie. if there is $t_+ > t_0$

δt	$h \approx 1.6 \cdot 10^{-2}$	$h \approx 8.2 \cdot 10^{-3}$	$h \approx 4.1 \cdot 10^{-3}$
$5 \cdot 10^{-1}$	-	-	-
$2 \cdot 10^{-1}$	-	-	-
$1 \cdot 10^{-1}$	-	-	-
$4 \cdot 10^{-2}$	-	-	-
$2 \cdot 10^{-2}$	0.24	-	-
$1 \cdot 10^{-2}$	0.07	0.10	-
$4 \cdot 10^{-3}$	0.172	t_0	t_0
$2 \cdot 10^{-3}$	0.202	0.044	t_0
$1 \cdot 10^{-3}$	0.210	0.079	t_0
$1 \cdot 10^{-4}$	0.2140	0.0939	0.0297

Crank-Nicolson scheme ($\theta = 1/2$)

δt	$h \approx 1.6 \cdot 10^{-2}$	$h \approx 8.2 \cdot 10^{-3}$	$h \approx 4.1 \cdot 10^{-3}$
$5 \cdot 10^{-1}$	$-4 \cdot 10^{-1}$	$-4 \cdot 10^{-1}$	$-4 \cdot 10^{-1}$
$2 \cdot 10^{-1}$	$-1 \cdot 10^{-1}$	$-1 \cdot 10^{-1}$	$-1 \cdot 10^{-1}$
$1 \cdot 10^{-1}$	$1 \cdot 10^{-13}$	$1 \cdot 10^{-13}$	$1 \cdot 10^{-13}$
$4 \cdot 10^{-2}$	$1 \cdot 10^{-21}$	$1 \cdot 10^{-21}$	$9 \cdot 10^{-22}$
$2 \cdot 10^{-2}$	$3 \cdot 10^{-30}$	$1 \cdot 10^{-30}$	$1 \cdot 10^{-30}$
$1 \cdot 10^{-2}$	$-6 \cdot 10^{-5}$	$1 \cdot 10^{-42}$	$9 \cdot 10^{-43}$
$4 \cdot 10^{-3}$	$-3 \cdot 10^{-4}$	$3 \cdot 10^{-64}$	$5 \cdot 10^{-65}$
$2 \cdot 10^{-3}$	$-3 \cdot 10^{-4}$	$-8 \cdot 10^{-11}$	$3 \cdot 10^{-87}$
$1 \cdot 10^{-3}$	$-2 \cdot 10^{-4}$	$-7 \cdot 10^{-10}$	$3 \cdot 10^{-113}$
$1 \cdot 10^{-4}$	$-9 \cdot 10^{-5}$	$-1 \cdot 10^{-11}$	$-8 \cdot 10^{-35}$

Crank-Nicolson scheme ($\theta = 1/2$)

δt	$h \approx 1.6 \cdot 10^{-2}$	$h \approx 8.2 \cdot 10^{-3}$	$h \approx 4.1 \cdot 10^{-3}$
$5 \cdot 10^{-1}$	1	1	1
$2 \cdot 10^{-1}$	t_0	t_0	t_0
$1 \cdot 10^{-1}$	t_0	t_0	t_0
$4 \cdot 10^{-2}$	t_0	t_0	t_0
$2 \cdot 10^{-2}$	t_0	t_0	t_0
$1 \cdot 10^{-2}$	t_0	t_0	t_0
$4 \cdot 10^{-3}$	0.072	t_0	t_0
$2 \cdot 10^{-3}$	0.132	t_0	t_0
$1 \cdot 10^{-3}$	0.173	0.029	t_0
$1 \cdot 10^{-4}$	0.2102	0.0874	0.0217

Implicit Euler scheme ($\theta = 1$)

δt	$h \approx 1.6 \cdot 10^{-2}$	$h \approx 8.2 \cdot 10^{-3}$	$h \approx 4.1 \cdot 10^{-3}$
$5 \cdot 10^{-1}$	$2 \cdot 10^{-5}$	$2 \cdot 10^{-5}$	$2 \cdot 10^{-5}$
$2 \cdot 10^{-1}$	$1 \cdot 10^{-7}$	$1 \cdot 10^{-7}$	$1 \cdot 10^{-7}$
$1 \cdot 10^{-1}$	$6 \cdot 10^{-10}$	$5 \cdot 10^{-10}$	$5 \cdot 10^{-10}$
$4 \cdot 10^{-2}$	$1 \cdot 10^{-15}$	$1 \cdot 10^{-15}$	$1 \cdot 10^{-15}$
$2 \cdot 10^{-2}$	$6 \cdot 10^{-22}$	$5 \cdot 10^{-22}$	$5 \cdot 10^{-22}$
$1 \cdot 10^{-2}$	$1 \cdot 10^{-30}$	$7 \cdot 10^{-31}$	$6 \cdot 10^{-31}$
$4 \cdot 10^{-3}$	$-2 \cdot 10^{-5}$	$2 \cdot 10^{-47}$	$8 \cdot 10^{-48}$
$2 \cdot 10^{-3}$	$-4 \cdot 10^{-5}$	$2 \cdot 10^{-64}$	$2 \cdot 10^{-65}$
$1 \cdot 10^{-3}$	$-7 \cdot 10^{-5}$	$-3 \cdot 10^{-13}$	$2 \cdot 10^{-87}$
$1 \cdot 10^{-4}$	$-9 \cdot 10^{-5}$	$-7 \cdot 10^{-12}$	$-1 \cdot 10^{-42}$

Implicit Euler scheme ($\theta = 1$)

TABLE IV

POSITIVITY THRESHOLD (t_+) COMPUTED WITH THE SPLITTING ALGORITHM AND THE CRANK-NICOLSON SCHEME (TOP) AND THE IMPLICIT EULER SCHEME (BOTTOM).

TABLE V

MINIMUM VALUE OF THE DOF ($\min_{i,k} X_i^k$) COMPUTED WITHOUT THE SPLITTING ALGORITHM AND THE CRANK-NICOLSON SCHEME (TOP) AND IMPLICIT EULER SCHEME (BOTTOM).

such that $X_i^k \geq 0, \forall i = 1, n$ for all $t_k > t_+ > t_0$. The smallest such t_+ , if it exists, is referred as the positivity threshold, as defined in [36]. Table IV on page 12 and table VI on page 13 give the positivity thresholds computed with and without the splitting technique respectively.

For the Crank-Nicolson scheme ($\theta = 1/2$) and the implicit Euler scheme ($\theta = 1$) the positivity is obtained under a specific condition on the time step and the mesh size. For a given mesh size, the time step δt must be bounded from above, but also from below to guarantee that the solution stays positive all along the simulation. In the case of the splitting technique those bounds are more restrictive than in the case of the resolution of the full problem without splitting. Those bounds are also more restrictive in the case of the Crank-Nicolson ($\theta = 1/2$) scheme than in the case of the

implicit Euler scheme ($\theta = 1$). Refining the mesh results in less restrictions on the time step but also lead to additional computational time.

With the Crank-Nicolson scheme ($\theta = 1/2$), for a given mesh size, if δt is too big, there is no threshold of positivity in $t_k \in]t_0, t_1]$ and the computed solution is not non-negative all along the simulation. For $\theta = 1/2$ and $\theta = 1$, still with a given mesh size, if δt is too small, the simulations showed that there is a threshold of positivity t_+ such that the approximated solution becomes non-negative for $t_k \geq t_+$. The thresholds of positivity slightly depend on the time step and tend to increase when the time step δt is decreased. The computations clearly showed that the positivity thresholds diminish with the mesh size h (see for example [36]).

Altogether, the positivity of the approximated

δt	$h \approx 1.6 \cdot 10^{-2}$	$h \approx 8.2 \cdot 10^{-3}$	$h \approx 4.1 \cdot 10^{-3}$
$5 \cdot 10^{-1}$	-	1	1
$2 \cdot 10^{-1}$	0.8	0.8	0.8
$1 \cdot 10^{-1}$	t_0	t_0	t_0
$4 \cdot 10^{-2}$	t_0	t_0	t_0
$2 \cdot 10^{-2}$	t_0	t_0	t_0
$1 \cdot 10^{-2}$	0.08	t_0	t_0
$4 \cdot 10^{-3}$	0.188	t_0	t_0
$2 \cdot 10^{-3}$	0.208	0.050	t_0
$1 \cdot 10^{-3}$	0.213	0.083	t_0
$1 \cdot 10^{-4}$	0.2143	0.0942	0.0300

Crank-Nicolson scheme ($\theta = 1/2$)

δt	$h \approx 1.6 \cdot 10^{-2}$	$h \approx 8.2 \cdot 10^{-3}$	$h \approx 4.1 \cdot 10^{-3}$
$5 \cdot 10^{-1}$	$-5 \cdot 10^{-1}$	$-5 \cdot 10^{-1}$	$-5 \cdot 10^{-1}$
$2 \cdot 10^{-1}$	$-2 \cdot 10^{-1}$	$-2 \cdot 10^{-1}$	$-2 \cdot 10^{-1}$
$1 \cdot 10^{-1}$	$4 \cdot 10^{-15}$	$4 \cdot 10^{-15}$	$4 \cdot 10^{-15}$
$4 \cdot 10^{-2}$	$6 \cdot 10^{-23}$	$5 \cdot 10^{-23}$	$4 \cdot 10^{-23}$
$2 \cdot 10^{-2}$	$2 \cdot 10^{-31}$	$8 \cdot 10^{-32}$	$6 \cdot 10^{32}$
$1 \cdot 10^{-2}$	$-4 \cdot 10^{-5}$	$1 \cdot 10^{-43}$	$6 \cdot 10^{43}$
$4 \cdot 10^{-3}$	$-3 \cdot 10^{-4}$	$4 \cdot 10^{-65}$	$5 \cdot 10^{-66}$
$2 \cdot 10^{-3}$	$-3 \cdot 10^{-4}$	$-5 \cdot 10^{-11}$	$5 \cdot 10^{-88}$
$1 \cdot 10^{-3}$	$-2 \cdot 10^{-4}$	$-6 \cdot 10^{-10}$	$8 \cdot 10^{-114}$
$1 \cdot 10^{-4}$	$-9 \cdot 10^{-5}$	$-1 \cdot 10^{-11}$	$-1 \cdot 10^{-34}$

Crank-Nicolson scheme ($\theta = 1/2$)

δt	$h \approx 1.6 \cdot 10^{-2}$	$h \approx 8.2 \cdot 10^{-3}$	$h \approx 4.1 \cdot 10^{-3}$
$5 \cdot 10^{-1}$	t_0	t_0	t_0
$2 \cdot 10^{-1}$	t_0	t_0	t_0
$1 \cdot 10^{-1}$	t_0	t_0	t_0
$4 \cdot 10^{-2}$	t_0	t_0	t_0
$2 \cdot 10^{-2}$	t_0	t_0	t_0
$1 \cdot 10^{-2}$	t_0	t_0	t_0
$4 \cdot 10^{-3}$	0.0720	t_0	t_0
$2 \cdot 10^{-3}$	0.1380	t_0	t_0
$1 \cdot 10^{-3}$	0.1760	0.0290	t_0
$1 \cdot 10^{-4}$	0.2104	0.0877	0.0221

Implicit Euler scheme ($\theta = 1$)

δt	$h \approx 1.6 \cdot 10^{-2}$	$h \approx 8.2 \cdot 10^{-3}$	$h \approx 4.1 \cdot 10^{-3}$
$5 \cdot 10^{-1}$	$4 \cdot 10^{-6}$	$4 \cdot 10^{-6}$	$4 \cdot 10^{-5}$
$2 \cdot 10^{-1}$	$2 \cdot 10^{-8}$	$2 \cdot 10^{-8}$	$2 \cdot 10^{-8}$
$1 \cdot 10^{-1}$	$2 \cdot 10^{-11}$	$2 \cdot 10^{-11}$	$2 \cdot 10^{-11}$
$4 \cdot 10^{-2}$	$5 \cdot 10^{-17}$	$5 \cdot 10^{-17}$	$5 \cdot 10^{-17}$
$2 \cdot 10^{-2}$	$3 \cdot 10^{-23}$	$2 \cdot 10^{-23}$	$2 \cdot 10^{-23}$
$1 \cdot 10^{-2}$	$1 \cdot 10^{-31}$	$4 \cdot 10^{-32}$	$3 \cdot 10^{-32}$
$4 \cdot 10^{-3}$	$-2 \cdot 10^{-5}$	$1 \cdot 10^{-48}$	$6 \cdot 10^{-49}$
$2 \cdot 10^{-3}$	$-4 \cdot 10^{-5}$	$2 \cdot 10^{-65}$	$2 \cdot 10^{-66}$
$1 \cdot 10^{-3}$	$-7 \cdot 10^{-5}$	$-3 \cdot 10^{-13}$	$2 \cdot 10^{-88}$
$1 \cdot 10^{-4}$	$-9 \cdot 10^{-5}$	$-7 \cdot 10^{-12}$	$-2 \cdot 10^{-42}$

Implicit Euler scheme ($\theta = 1$)

TABLE VI

POSITIVITY THRESHOLD (t_+) COMPUTED WITHOUT THE SPLITTING ALGORITHM AND THE CRANK-NICOLSON SCHEME (TOP) AND THE IMPLICIT EULER SCHEME (BOTTOM).

TABLE VII

MINIMUM VALUE OF THE DOF ($\min_{i,k} X_i^k$) COMPUTED FOR THE PURE DIFFUSION PROBLEM WITH THE CRANK-NICOLSON SCHEME (TOP) AND THE IMPLICIT EULER SCHEME (BOTTOM).

solution is obtained at the expense of the computational cost, but for a given mesh size h computations performed with too small time step can also lead to a loss of positivity for small t_k . In [36] (and references therein), Thomée showed that threshold values of $t_k > 0$ may exist such that $X(t) > 0$ when $t > t_k$.

At this stage, one may wonder how each term of the splitting behaves in terms of positivity preservation. The reaction term is approximated using an exact scheme, so obviously the positivity of the solution is preserved. What about the diffusion and the advection term ?

2) Positivity of the pure diffusion problem:

Here I set $\mathbf{v} = (0, 0)$ and $\rho = 0$, while keeping all others parameters to the same values as previously. Table VII clearly shows that the Crank-Nicolson scheme ($\theta = 1/2$) is positivity preserving under a

CFL-like condition with upper and lower bounds, like in the previous test. The implicit Euler scheme ($\theta = 1$) seems to be more favorable, since it preserves the positivity even for big values of the time step. For both the Crank-Nicolson ($\theta = 1/2$) and implicit Euler ($\theta = 1$) schemes, the approximated solution suffers from a loss of positivity for small values of t_k when the time step is too small. According to table VIII, there are positivity thresholds, like in [36] which indeed deals with the heat equation.

3) Positivity of the pure advection problem:

Here I set $\sigma = 0$ and $\rho = 0$, while keeping all others parameters to the same values as in the first test. Table IX shows that none of the computations performed gave a non negative solutions, even though the minimum value of the dof can be really close to zero for small mesh sizes. Besides, I did

δt	$h \approx 1.6 \cdot 10^{-2}$	$h \approx 8.2 \cdot 10^{-3}$	$h \approx 4.1 \cdot 10^{-3}$
$5 \cdot 10^{-1}$	-	-	-
$2 \cdot 10^{-1}$	0.8	0.8	0.8
$1 \cdot 10^{-1}$	t_0	t_0	t_0
$4 \cdot 10^{-2}$	t_0	t_0	t_0
$2 \cdot 10^{-2}$	t_0	t_0	t_0
$1 \cdot 10^{-2}$	0.1	t_0	t_0
$4 \cdot 10^{-3}$	0.2	t_0	t_0
$2 \cdot 10^{-3}$	0.216	0.054	t_0
$1 \cdot 10^{-3}$	0.219	0.085	t_0
$1 \cdot 10^{-4}$	0.2203	0.0956	0.0303

Crank-Nicolson scheme ($\theta = 1/2$)

δt	$h \approx 1.6 \cdot 10^{-2}$	$h \approx 8.2 \cdot 10^{-3}$	$h \approx 4.1 \cdot 10^{-3}$
$5 \cdot 10^{-1}$	$-2 \cdot 10^{-1}$	$-2 \cdot 10^{-1}$	$-2 \cdot 10^{-1}$
$2 \cdot 10^{-1}$	$-4 \cdot 10^{-2}$	$-4 \cdot 10^{-2}$	$-4 \cdot 10^{-2}$
$1 \cdot 10^{-1}$	$-4 \cdot 10^{-3}$	$-2 \cdot 10^{-3}$	$-1 \cdot 10^{-3}$
$4 \cdot 10^{-2}$	$-1 \cdot 10^{-3}$	$-4 \cdot 10^{-7}$	$-5 \cdot 10^{-7}$
$2 \cdot 10^{-2}$	$-1 \cdot 10^{-3}$	$-6 \cdot 10^{-8}$	$-2 \cdot 10^{-13}$
$1 \cdot 10^{-2}$	$-1 \cdot 10^{-3}$	$-4 \cdot 10^{-8}$	$-3 \cdot 10^{-28}$
$4 \cdot 10^{-3}$	$-1 \cdot 10^{-3}$	$-3 \cdot 10^{-8}$	$-1 \cdot 10^{-28}$
$2 \cdot 10^{-3}$	$-1 \cdot 10^{-3}$	$-3 \cdot 10^{-8}$	$-1 \cdot 10^{-28}$
$1 \cdot 10^{-3}$	$-1 \cdot 10^{-3}$	$-3 \cdot 10^{-8}$	$-1 \cdot 10^{-28}$
$1 \cdot 10^{-4}$	$-1 \cdot 10^{-3}$	$-3 \cdot 10^{-8}$	$-1 \cdot 10^{-28}$

Crank-Nicolson scheme ($\theta = 1/2$)

δt	$h \approx 1.6 \cdot 10^{-2}$	$h \approx 8.2 \cdot 10^{-3}$	$h \approx 4.1 \cdot 10^{-3}$
$5 \cdot 10^{-1}$	t_0	t_0	t_0
$2 \cdot 10^{-1}$	t_0	t_0	t_0
$1 \cdot 10^{-1}$	t_0	t_0	t_0
$4 \cdot 10^{-2}$	t_0	t_0	t_0
$2 \cdot 10^{-2}$	t_0	t_0	t_0
$1 \cdot 10^{-2}$	t_0	t_0	t_0
$4 \cdot 10^{-3}$	0.084	t_0	t_0
$2 \cdot 10^{-3}$	0.148	t_0	t_0
$1 \cdot 10^{-3}$	0.184	0.032	t_0
$1 \cdot 10^{-4}$	0.2165	0.0893	0.0224

Implicit Euler scheme ($\theta = 1$)

δt	$h \approx 1.6 \cdot 10^{-2}$	$h \approx 8.2 \cdot 10^{-3}$	$h \approx 4.1 \cdot 10^{-3}$
$5 \cdot 10^{-1}$	$-1 \cdot 10^{-4}$	$-1 \cdot 10^{-10}$	$-8 \cdot 10^{-34}$
$2 \cdot 10^{-1}$	$-2 \cdot 10^{-4}$	$-6 \cdot 10^{-10}$	$-3 \cdot 10^{-32}$
$1 \cdot 10^{-1}$	$-4 \cdot 10^{-4}$	$-1 \cdot 10^{-9}$	$-2 \cdot 10^{-31}$
$4 \cdot 10^{-2}$	$-6 \cdot 10^{-4}$	$-4 \cdot 10^{-9}$	$-9 \cdot 10^{-31}$
$2 \cdot 10^{-2}$	$-8 \cdot 10^{-4}$	$-7 \cdot 10^{-9}$	$-2 \cdot 10^{-30}$
$1 \cdot 10^{-2}$	$-9 \cdot 10^{-4}$	$-1 \cdot 10^{-8}$	$-5 \cdot 10^{-30}$
$4 \cdot 10^{-3}$	$-1 \cdot 10^{-3}$	$-1 \cdot 10^{-8}$	$-8 \cdot 10^{-30}$
$2 \cdot 10^{-3}$	$-1 \cdot 10^{-3}$	$-2 \cdot 10^{-8}$	$-2 \cdot 10^{-29}$
$1 \cdot 10^{-3}$	$-1 \cdot 10^{-3}$	$-2 \cdot 10^{-8}$	$-4 \cdot 10^{-29}$
$1 \cdot 10^{-4}$	$-1 \cdot 10^{-3}$	$-3 \cdot 10^{-8}$	$-9 \cdot 10^{-29}$

Implicit Euler scheme ($\theta = 1$)

TABLE VIII

POSITIVITY THRESHOLD (t_+) COMPUTED FOR THE PURE DIFFUSION PROBLEM WITH THE CRANK-NICOLSON SCHEME (TOP) AND THE IMPLICIT EULER SCHEME (BOTTOM).

TABLE IX

MINIMUM VALUE OF THE DOF ($\min_{i,k} X_i^k$) COMPUTED FOR THE PURE ADVECTION PROBLEM WITH THE CRANK-NICOLSON SCHEME (TOP) AND THE IMPLICIT EULER SCHEME (BOTTOM).

not observe any positivity threshold. The approximated solution stays non positive all along the simulation. However I run additional simulations with even smaller mesh size ($h \approx 2.0 \times 10^{-3}$ and $\delta t = 10^{-4}$). This time the computed solution was positive at the beginning of the simulation (before $t_- = 1.9 \times 10^{-3}$), pointing the existence of a threshold of negativity, to finally reaching a negative minimum values of dof (around -10^{-44}). Unfortunately, this threshold of negativity is really small compared to the ending time of the computation ($t_1 = 1$), while the computational time was reaching more than 14 hours (Intel®Core™i7-7820HQ at 2.9 GHz, RAM 32 GB) for both the Crank-Nicolson and the implicit Euler schemes.

In fact it is well known that for the advection term the solution can be polluted by overshoot and undershoot oscillations near a discontinuity or a

sharp layer, see [34], [33], [35], [30]. For low order accurate spacial approximations one can prove the positivity preserving property of the scheme [33]. But for high order schemes slopes limiters are often required to guarantee the positivity of the approximated solution. When slope limiters are used, explicit time schemes seem to be suitable for the advection [6]. However, in the next section we will only privilege a numerical scheme that is unconditionally stable, i.e. the Crank-Nicolson scheme, that is a two-order scheme.

V. APPLICATION TO THE SIMULATION OF ROOT SYSTEM GROWTH

In this section, I apply the previous DG-splitting approach to solve numerically the C-Root model. First, I detail the parameters used for the simulations, then, I present and validate the results of the

simulations.

A. The C-Root parameters for Eucalyptus root growth

The parameters and operators' coefficients are chosen based on the previous calibration done in [2]. The diffusion coefficient, σ , is build using the following Gaussian function

$$f_{\alpha,\mu}(x, y) = \frac{\alpha}{\sqrt{2\pi}} \exp\left(-\frac{(r(x, y) - \mu)^2}{2}\right)$$

where $r(x, y) = \sqrt{(x - x_0)^2 + (y - y_0)^2}$ and $(x_0, y_0) \in \Omega =] - L, L[$. The function $f_{\alpha,\mu}(x, y)$ depends on two real and positive parameters: α , related to the maximum amplitude of $f_{\alpha,\mu}$, and μ , the distance from (x_0, y_0) to the point where the function $f_{\alpha,\mu}$ reaches its maximum.

The diffusion tensor is taken such that

$$\sigma(x, y) = f_{\alpha_d, \mu_d}(x, y) \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

for all $(x, y) \in \Omega$, and $\alpha_d, \mu_d \in \mathbb{R}^+$ are given parameters. The advection vector is taken such that $\mathbf{v}(x, y) = (0, -v_0)^T$, for all $(x, y) \in \Omega$, with v_0 a positive constant. The reaction term is constant in space and splitted into two contributions: β_r and μ_r , the branching and mortality rates, respectively. That is

$$\rho = \beta_r - \mu_r \in \mathbb{R}.$$

The branching rate, β_r , is estimated from biological knowledge: it is equal to zero before 9 months and equal to 1/3 after, since no roots die before 9 month. However, for the following simulations we will not distinguish the contribution of β_r and μ_r , so that the reaction term will only be described by the parameter ρ .

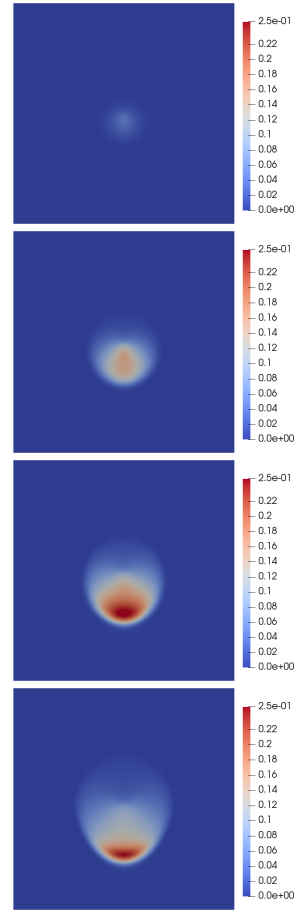


Fig. 6. Density of apices computed at $t = 6, t = 12, t = 18$ and $t = 24$ months (from the left to the right and from the top to the bottom).

B. Some simulations

For the simulation the initial solution is chosen equal to the following function:

$$u_0(x, y) = A \left[\frac{\exp(b(1 - x))}{(\exp(-b(1 - x)) + \exp(b(1 - x)))} - \frac{\exp(b(-1 - x))}{(\exp(-b(-1 - x)) + \exp(b(-1 - x)))} \right] \times \left[\frac{\exp(b(1 - y))}{(\exp(-b(1 - y)) + \exp(b(1 - y)))} - \frac{\exp(b(-1 - y))}{(\exp(-b(-1 - y)) + \exp(b(-1 - y)))} \right]$$

with $A = 2 \cdot 10^{-4}$ and $b = 1$. The parameters' values μ_r, α_d, μ_d are estimated using the code

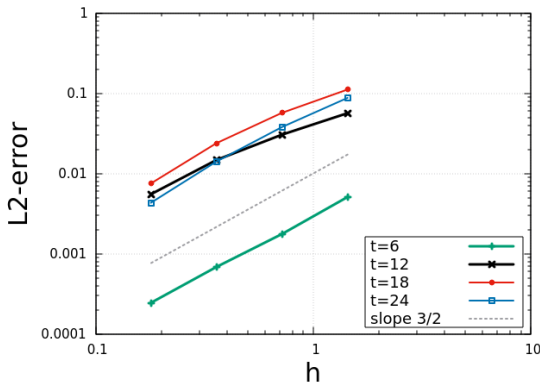


Fig. 7. L^2 -error with respect to the solution obtained with the mesh of size $h \approx 9.87 \times 10^{-2}$ and $\delta t = 10^{-3}$ computed at $t = 6$, $t = 12$, $t = 18$ and $t = 24$ months and plotted against the mesh size.

described in [2]. I run the simulations from $t_0 = 1$ to $t_1 = 24$ months, with $L = 13$. The simulations are performed for different values of the mesh size. Fig. 6, page 15, shows the solution computed at four different stages of the root system development. One can notice the diffusion of the apices in the soil and also the transport of the apices from the top to the bottom of the soil layer. Since there is no analytic solution, the convergence of the computation is evaluated by measuring the L^2 -errors with respect to the approximated solution computed with the finest mesh ($h \approx 8.97 \times 10^{-2}$) and with $\delta t = 10^{-3}$. The curves of the errors against the mesh size are plotted on figure 7 and clearly show that the DG-splitting algorithm converges with a convergence rate of almost two. However, one can note that the mesh sizes and the time steps chosen for the simulations presented here might not be small enough. The positivity of the solution is not preserved at all times and the full convergence might not be achieved. Unfortunately, refining the mesh sizes and the time steps can lead to prohibitive computational time as shown on table X. On top of that simulation of root system growth can last for a long period of time, particularly for trees. Finally, this application shows promising results for future simulations of

h (\approx)	$\delta t = 10^{-1}$	$\delta t = 10^{-2}$	$\delta t = 10^{-3}$
1.44	1 s.	7 s.	66 s.
7.18×10^{-1}	16 s.	46 s.	6 min.
3.59×10^{-1}	70 s.	3.5 min.	27 min.
1.79×10^{-1}	7 min.	17 min.	2 h.
8.97×10^{-2}	50 min.	2h30	9 h.

TABLE X
COMPUTATIONAL TIMES FOR THE SIMULATIONS OF A ROOT SYSTEM GROWTH PERFORMED (WITH THE PROCESSOR INTEL®CORE™I7-7820HQ AT 2.9 GHZ, RAM 32 GB) BETWEEN $t = 1$ AND $t = 24$ MONTHS WITH THE DG-SPLITTING ALGORITHM AND THE CRANK-NICOLSON SCHEME ($\theta = 1/2$).

the root system growth, provided that the computational cost is not limiting. Further simulations requiring much more computational power has to be done to check if the convergence is achieved. This application also point out the difficulties related to the rigorous simulation validation in realistic test-cases of root system growth.

VI. CONCLUSION

In this work, a discontinuous Galerkin approximation method based on unstructured mesh combined with operator splitting has been described, implemented and tested, to solve an advection-diffusion-reaction equation used to model the growth of root systems. The code has been validated in a simple test case for which an analytic expression of the solution is known. The computations showed that the method converges with a convergence rate of two in space with P^1 -finite elements. A convergence rate of one and two in time were obtained for respectively the implicit Euler scheme and the Crank-Nicolson scheme both with and without the splitting technique. The computations of those convergence rates required the use of fine mesh grids. For the explicit Euler scheme, such fine mesh computations were not performed since they require really small time steps to fulfill the CFL condition, resulting in huge additional computational cost. Indeed the computational time of the DG-splitting algorithm behaves like $1/\delta t$ and $1/h^2$ where δt and h are respectively the time step and the mesh size.

Similarly, the positivity of the approximated solution is obtained at the expense of the computational time since it requires meshes of small size and small time steps. In fact, there is a CFL-like condition for positivity that has to be fulfilled to guarantee the positivity of the approximated solution. But for a given mesh size computations performed with too small time step can also lead to a loss of positivity at the beginning of the computation [36]. In that cases, the computations showed that there is a positivity threshold in time after which the solution becomes positive. This positivity threshold clearly appeared to diminish with the mesh size. This behavior is specific to the diffusion term. For the advection term, the computations also showed that the positivity of the solution can be preserved, but only at the beginning of the simulation and it required a really small mesh size and time step leading to huge computational time. Further studies in terms of numerical analysis has to be done in that direction.

I also performed a more realistic simulation of root system growth. The computations showed that the algorithm converged but additional simulations with smaller time steps and mesh sizes might be performed to recover the full convergence order and positivity. Validation of the computation, but above all the computational time appeared to be the major limitations of the root growth simulation based on the C-Root model, particularly when it comes to deal with trees for which the life span is rather a long period of time. Further improvements on the numerical method has to be done so that the scheme preserves the positivity of the approximated solution under acceptable CFL conditions in terms of computational time. However, our work shows promising results for the simulation of the C-Root model which appears to be an appropriate methodology for future improvements, like root-soil coupling or nonlinear terms arising to handle competition phenomena.

ACKNOWLEDGMENT

The author would like to thank Y. Dumont (CIRAD, University of Pretoria) for discussions

and valuable comments about the numerical schemes.

REFERENCES

- [1] A. Bonneau, Y. Dumont, H. Rey, C. Jourdan and T. Fourcaud, A minimal continuous model for simulating growth and development of plant root systems, Plant and Soil, Springer, 2012, 354, 211-22. <https://doi.org/10.1007/s11104-011-1057-7>.
- [2] E. Tillier and A. Bonneau, Operator splitting for solving C-Root, a minimalist and continuous model of root system growth, Plant Growth Modeling, Simulation, Visualization and Applications (PMA), 2012 IEEE Fourth International Symposium on, 2012, 396-402. <https://doi.org/10.1109/PMA.2012.6524863>.
- [3] E. Peynaud, T. Fourcaud and Y. Dumont Numerical resolution of the C-Root model using Discontinuous Galerkin methods on unstructured meshes: application to the simulation of root system growth, 2016 IEEE International Conference on Functional-Structural Plant Growth Modeling, Simulation, Visualization and Applications (FSPMA), IEEE, FSPMA. Qingdao: IEEE, 2016, 158-166. <https://doi.org/10.1109/FSPMA.2016.7818302>.
- [4] H.-G. Roos, M. Stynes and L. Tobiska, Robust numerical methods for singularly perturbed differential equations: convection-diffusion-reaction and flow problems Springer Science & Business Media, 2008, 24. <https://doi.org/10.1007/978-3-540-34467-4>.
- [5] X. Zhang and CW. Shu, Maximum-principle-satisfying and positivity-preserving high-order schemes for conservation laws: survey and new developments. Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, 2011, 467, 2752-2776. <https://doi.org/10.1098/rspa.2011.0153>.
- [6] W. Hundsdorfer and J. G. Verwer, Numerical solution of time-dependent advection-diffusion-reaction equations. Springer Science & Business Media, 2013, 33. DOI 10.1007/978-3-662-09017-6
- [7] G. Strang, On the construction and comparison of difference schemes. SIAM Journal on Numerical Analysis, SIAM. 5, 506-517. 1968. <https://doi.org/10.1137/0705041>.
- [8] N. Janenko, The method of fractional steps. Springer. 1971. <https://doi.org/10.1007/978-3-642-65108-3>.
- [9] A. Chertock and A. Kurganov, On splitting-based numerical methods for convection-diffusion equations Numerical methods for balance laws, Aracne Editrice Srl Rome, 2010, 24, 303-343.
- [10] D. Lanser and J.G. Verwer, Analysis of operator splitting for advection-diffusion-reaction problems from air pollution modelling, Journal of computational and applied mathematics, Elsevier, 111, 1-2, 1999, 201-216. [https://doi.org/10.1016/S0377-0427\(99\)00143-0](https://doi.org/10.1016/S0377-0427(99)00143-0).
- [11] J. Kačur, B. Malengier, M. Remešiková, Convergence of an operator splitting method on a bounded domain for a convection-diffusion-reaction system, Journal of

- Mathematical Analysis and Applications. 348, 894-914, 2008. <https://doi.org/10.1016/j.jmaa.2008.08.017>.
- [12] M. Remešková, Numerical solution of two-dimensional convection-diffusion-adsorption problems using an operator splitting scheme, Applied mathematics and computation, 184, 116-130, 2007. <https://doi.org/10.1016/j.amc.2005.06.018>.
- [13] J.C. Chrisspell, V. Ervin V and E. Jenkins, A fractional step θ -method for convection-diffusion problems, Journal of Mathematical Analysis and Applications, 333, 204-218, 2007. <https://doi.org/10.1016/j.jmaa.2006.11.059>.
- [14] S. Ganesan and L. Tobiska, Operator-splitting finite element algorithms for computations of high-dimensional parabolic problems, Applied Mathematics and Computation, Elsevier, 219, 2013. 6182-6196. <https://doi.org/10.1016/j.amc.2012.12.027>.
- [15] M. Wheeler and C. Dawson, An operator-splitting method for advection-diffusion-reaction problems, MAFELAP Proceedings, 6, 463-482, 1987.
- [16] R. Anguelov, C. Dufourd and Y. Dumont, Simulations and parameter estimation of a trap-insect model using a finite element approach, Mathematics and Computers in Simulation, 2017, 133, 47-75. <https://doi.org/10.1016/j.matcom.2015.06.014>.
- [17] C. Dufourd and Y. Dumont, Impact of environmental factors on mosquito dispersal in the prospect of sterile insect technique control, Computers & Mathematics with Applications, Elsevier, 2013, 66, 1695-1715. <https://doi.org/10.1016/j.camwa.2013.03.024>.
- [18] V. Girault, B. Rivière and M.F. Wheeler, A splitting method using discontinuous Galerkin for the transient incompressible Navier-Stokes equations, ESAIM: Mathematical Modelling and Numerical Analysis, EDP Sciences, 39, 1115-1147, 2005. <https://doi.org/10.1051/m2an:2005048>.
- [19] J. Zhu, Y.T. Zhang, S.A. Newman and M. Alber, Application of discontinuous Galerkin methods for reaction-diffusion systems in developmental biology, Journal of Scientific Computing, Springer, 2009, 40, 391-418. <https://doi.org/10.1007/s10915-008-9218-4>.
- [20] N. Ahmed, G. Matthies and L. Tobiska, Finite element methods of an operator splitting applied to population balance equations Journal of Computational and Applied Mathematics, Elsevier. 236, 1604-1621, 2011. <https://doi.org/10.1016/j.cam.2011.09.025>.
- [21] R. Zhang, J. Zhu, A.F Loula and X. Yu, Operator splitting combined with positivity-preserving discontinuous Galerkin method for the chemotaxis model, Journal of Computational and Applied Mathematics. 302, 312 - 326, 2016. <https://doi.org/10.1016/j.cam.2016.02.018>.
- [22] L. Edelstein-Keshet, Mathematical models in biology, SIAM, 46, 1988. ISBN 0-89871-554-7.
- [23] B. Perthame, Transport equations in biology. Springer Science & Business Media, 2006. <https://doi.org/10.1007/978-3-7643-7842-4>.
- [24] B. Perthame, Parabolic equations in biology. Growth, reaction, mouvement and diffusion, Springer, 2015. https://doi.org/10.1007/978-3-319-19500-1_1
- [25] A. Ern, J.L. Guermond, Theory and practice of finite elements, Springer, 159, 2004. <https://doi.org/10.1007/978-1-4757-4355-5>.
- [26] V. Volpert, Elliptic Partial Differential Equations: Volume 2: Reaction-Diffusion Equations, Springer, 104, 2014. <https://doi.org/10.1007/978-3-0348-0813-2>.
- [27] D. Kuzmin, A guide to numerical methods for transport equations, Friedrich-Alexander-Universität, Erlangen-Nrnberg, 2010.
- [28] F. Brezzi, L. D. Marini and E. Süli, Discontinuous Galerkin methods for first-order hyperbolic problems, Mathematical models and methods in applied sciences, World Scientific, 2004, 14, 1893-1903. <https://doi.org/10.1142/S0218202504003866>.
- [29] W. H. Reed and T. R. Hill, Triangular mesh methods for the neutron transport equation, Los Alamos Scientific Lab., N. Mex.(USA), report LA-UR-73-479, 1973.
- [30] B. Rivière, Discontinuous Galerkin methods for solving elliptic and parabolic equations: theory and implementation, Society for Industrial and Applied Mathematics, 2008. <https://doi.org/10.1137/1.9780898717440>
- [31] J. Oden, I. Babuška and C. E. Baumann, A discontinuous hp -finite element method for diffusion problems, Journal of computational physics, Elsevier 146, 491-519, 1998. <https://doi.org/10.1006/jcph.1998.6032>.
- [32] D. A. Di Pietro and A. Ern, Mathematical aspects of discontinuous Galerkin methods Springer, 69, 2011. <https://doi.org/10.1007/978-3-642-22980-0>.
- [33] X. Zhang and CW. Shu CW. Maximum-principle-satisfying and positivity-preserving high-order schemes for conservation laws: survey and new developments, Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, 2011, 467, 2752-2776. <https://doi.org/10.1098/rspa.2011.0153>.
- [34] B. Cockburn and CW. Shu, Runge-Kutta discontinuous Galerkin methods for convection-dominated problems. Journal of scientific computing, Springer, 2001, 16, 173-261. <https://doi.org/10.1023/A:1012873910884>.
- [35] JS. Hesthaven and T. Warburton, Nodal discontinuous Galerkin methods: algorithms, analysis, and applications. Springer, 2007, 54. <https://doi.org/10.1007/978-0-387-72067-8>.
- [36] V. Thomée, On positivity preservation in some finite element methods for the heat equation. International Conference on Numerical Methods and Applications, 2014, 13-24. https://doi.org/10.1007/978-3-319-15585-2_2.
- [37] J. Zhu, YT. Zhang, SA. Newman and M. Alber, Application of discontinuous Galerkin methods for reaction-diffusion systems in developmental biology. Journal of Scientific Computing, Springer, 2009, 40, 391-418. <https://doi.org/10.1007/s10915-008-9218-4>.
- [38] J.-F. Barczy, H. Rey, S. Griffon and C. Jourdan, DigR: a generic model and its open source simulation software to mimic three-dimensional root-system architecture

- diversity. *Annals of Botany*, 2018, 121, 5, 1089-1104, <https://doi.org/10.1093/aob/mcy018>.
- [39] L. X. Dupuy, M. Vignes, An algorithm for the simulation of the growth of root systems on deformable domains. *Journal of Theoretical Biology*, 2012, 310, 164-174. <https://doi.org/10.1016/j.jtbi.2012.06.025>.
- [40] L. Dupuy, P. J. Gregory, A. G. Bengough, Root growth models: towards a new generation of continuous approaches. *Journal of experimental botany, Soc Experiment Biol*, 2010. <https://doi.org/10.1093/jxb/erp389>.
- [41] P. Bastian, A. Chavarria-Krauser, C. Engwer, W. Jäger, S. Marnach, M. Ptashnyk, Modelling in vitro growth of dense root networks *Journal of theoretical biology*, Elsevier, 2008, 254, 99-109. <https://doi.org/10.1016/j.jtbi.2008.04.014>.
- [42] T. Roose, A. Schnepf, Mathematical models of plant-soil interaction *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, The Royal Society, 2008, 366, 4597-4611, <https://doi.org/10.1098/rsta.2008.0198>.
- [43] S. G. Adiku, R. D. Braddock, C. W. Rose, 1996, Simulating root growth dynamics, *Environmental Software* 11 : 99-103. [https://doi.org/10.1016/S0266-9838\(96\)00041-X](https://doi.org/10.1016/S0266-9838(96)00041-X).
- [44] H. Hayhoe, 1981, Analysis of a diffusion model for plant root growth and an application to plant soil-water uptake, *Soil Science* 131 : 334-343.
- [45] M. Heinen, A. Mollier, P. De Willigen, 2003 Growth of a root system described as diffusion numerical model and application, *Plant and Soil* 252 : 251-265. <https://doi.org/10.1023/A:1024749022761>.
- [46] V. R. Reddy, Ya. A. Pachepsky, 2001, Testing a convective dispersive model of two dimensional root growth and proliferation in a greenhouse experiment with mare plants, *Annals of Botany* 87 : 759-768. <https://doi.org/10.1006/anbo.2001.1409>.
- [47] P. -H. Tournier, F. Hecht, M. Comte, Finite element model of soil water and nutrient transport with root uptake: explicit geometry and unstructured adaptive meshing. *Transp. Porous Media* 106 (2), 487504 (2015). <https://doi.org/10.1007/s11242-014-0411-7>.
- [48] M. Comte, Analysis and Simulation of a Model of Phosphorus Uptake by Plant Roots in Current Research in Nonlinear Analysis: In Honor of Haim Brezis and Louis Nirenberg, Rassias, T. M. (Ed.), Springer International Publishing, 2018, 85-97. https://doi.org/10.1007/978-3-319-89800-1_4.
- [49] F. Gérard, Cé Blitz-Frayret, P. Hinsinger, L. Pagès, Modelling the interactions between root system architecture, root functions and reactive transport processes in soil *Plant and Soil*, 2017, 413, 161-180. <https://doi.org/10.1007/s11104-016-3092-x>.
- [50] H. Brezis, Functional analysis, Sobolev spaces and partial differential equations. Springer Science & Business Media, 2010. <https://doi.org/10.1007/978-0-387-70914-7>.
- [51] P. R. Amestoy, I. S. Duff, J. Koster and J.-Y. L'Excellent, A fully asynchronous multifrontal solver using distributed dynamic scheduling, *SIAM Journal of Matrix Analysis and Applications*, Vol 23, No 1, pp 15-41 (2001). <https://doi.org/10.1137/S0895479899358194>.
- [52] P. R. Amestoy, A. Guermouche, J.-Y. L'Excellent and S. Pralet, Hybrid scheduling for the parallel solution of linear systems. *Parallel Computing* Vol 32 (2), pp 136-156 (2006). <https://doi.org/10.1016/j.parco.2005.07.004>.