



**HAL**  
open science

**Solid propellant combustion in the low Mach  
one-dimensional approximation: from an index-one  
differential-algebraic formulation to high-fidelity  
simulations through high-order time integration with  
adaptive time-stepping**

Laurent François, Joël Dupays, Dmitry Davidenko, Marc Massot

► **To cite this version:**

Laurent François, Joël Dupays, Dmitry Davidenko, Marc Massot. Solid propellant combustion in the low Mach one-dimensional approximation: from an index-one differential-algebraic formulation to high-fidelity simulations through high-order time integration with adaptive time-stepping. 2020. hal-02888035v1

**HAL Id: hal-02888035**

**<https://hal.science/hal-02888035v1>**

Preprint submitted on 3 Jul 2020 (v1), last revised 4 Jan 2024 (v2)

**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.

# Solid propellant combustion in the low Mach one-dimensional approximation : from an index-one differential-algebraic formulation to high-fidelity simulations through high-order time integration with adaptive time-stepping

Laurent François<sup>a,b</sup>, Joël Dupays<sup>a</sup>, Dmitry Davidenko<sup>a</sup>, Marc Massot<sup>b</sup>

<sup>a</sup>ONERA, DMPE, 6 Chemin de la Vauve aux Granges, 91120 Palaiseau, France

<sup>b</sup>CMAP, CNRS, Ecole polytechnique, Institut Polytechnique de Paris, Route de Saclay, 91128 Palaiseau Cedex, France

---

## Abstract

An unsteady one-dimensional model of solid propellant combustion, based on a low-Mach assumption, is presented and semi-discretised in space via a finite volume scheme. The mathematical nature of this system is shown to be differential-algebraic of index one. A high-fidelity numerical strategy with stiffly accurate singly diagonally implicit Runge-Kutta methods is proposed, and time adaptation is made possible using embedded schemes. High-order is shown to be reached, while handling the constraints properly, both at the interface and for the mass conservation in the gaseous flow field. Three challenging test-cases are thoroughly investigated: ignition transients, growth of combustion instabilities through a Hopf bifurcation leading to a limit cycle periodic solution and the unsteady response of the system when detailed gas-phase kinetics are included in the model. The method exhibits high efficiency for all cases in terms of both computational time and accuracy compared to first- and second-order schemes traditionally used in the combustion literature, where the time step adaptation is CFL- or variation-based.

---

## 1. Introduction

Solid propellant combustion is a key element in rocket propulsion and has been extensively studied since the 1950s [1, 2, 3, 4, 5]. It involves a solid phase and a gas phase, separated by an interface. The solid is heated up by thermal conduction and radiation from the gas phase. At its surface, the solid propellant decomposes, melts and evaporates through a pyrolysis process as the interface regresses. The resulting gaseous products react and form a flame which heats back the solid, allowing for a sustained combustion.

It is essential to understand the physics of this phenomenon to allow for clever combustion chamber designs leading to efficient solid rocket motors. A key element is the regression speed of the propellant surface and its dependence on the combustion chamber conditions. This has been extensively studied in a steady-state context through the use of analytical models [6, 7, 8]. Such models served as initial tools for the ground analysis of unsteady combustion dynamics, for example with the computation of a linearised frequency response [9], the characterisation of intrinsic combustion instabilities, and the stability analysis of a combustion chamber [10]. Ignition transients have originally been treated with analytical models considering only the unsteady heating of the solid phase, with heat being generated by surface reactions, bulk reactions, or simplified gas flame heat feedback [11, 12]. Unfortunately,

---

\*Corresponding author

Email address: [laurent.francois@onera.fr](mailto:laurent.francois@onera.fr) (Laurent François)

these simple models are limited in accuracy and struggle to predict the combustion dynamics of a propellant without extensive experimental data. They are also not suited to the detailed study of transient dynamics. Since the 1990s, developments have therefore focused on CFD tools, both for steady and unsteady applications. Recent one-dimensional CFD codes make use of complex chemical mechanisms [13, 14, 15, 16], however such kinetics are still difficult to evaluate and validate, and the heterogeneity of widely used propellants, such as AP-HTPB, tends to limit the use of such mechanisms to lower pressures [14] due to an increase of three-dimensional effects involving diffusion flames at higher pressures. Detailed two- to three-dimensional CFD approaches taking into account the heterogeneity of the propellant and relying on simplified kinetics [17, 18, 19, 20] have started to emerge, however they are costly and limited to the simulation of a very small burning area. All these models allow for the precise analysis of the unsteady combustion of a propellant that is assumed to be semi-infinite in the direction normal to the surface, and either infinite (for one-dimensional models) or periodic symmetric (for two- and three-dimensional models) in the directions tangent to the surface. They have been developed to simulate laboratory experiments, for example the burning of a sufficiently large and thick propellant sample, such that these assumptions are realistic. Cook-off studies of enclosed samples have also been considered [14, 21].

The previous models cannot be used for large-scale simulations, and there is still a need for a precise and robust solid propellant combustion model and numerical strategy for the simulation of the ignition of a complete combustion chamber. A three-dimensional CFD tool can hardly be used to solve the flow field inside the chamber and the surface combustion, due to extreme computational requirements. Indeed the near-surface combustion typically happens in a flame that is a few hundred micrometers thick, across which the temperature may increase by 2000K, requiring mesh cells thinner than 1 micrometers for an adequate resolution. Using such a refined mesh along the complete length of combustion chamber (on the order of 1-10 meters) would be prohibitive. Therefore, the only viable option is to use a coarse mesh near the surface, and solve all the surface combustion within a simpler submodel, e.g. a one-dimensional CFD tool. Such a coupling with a very simplified propellant modelling has been reported [22, 23], and has been implemented in the CEDRE code [24] at ONERA, with a propellant model based on analytical formulae [25]. More accurate models are needed to better resolve the coupled dynamics. These models also need to be as efficient as possible to allow for short restitution times. For instance, the time integration procedure must be accurate enough without requiring prohibitively small time steps. Eventually, there is a convergent need for a one-dimensional model, either for the coupling with the previous three-dimensional model for combustion chamber predictive simulations, or for the parametric / detailed study of flame dynamics and unsteady combustion dynamics in a purely one-dimensional context. Such a model should involve the proper level of physics, while being amenable to high-fidelity simulation by relying on a tailored numerical strategy taking into account the fact that most models involve systems with algebraic constraints.

This is specifically what we wish to investigate in this paper. It is instructive to conduct a short overview of the various numerical approaches presented in the literature for the time integration of one-dimensional solid propellant combustion models. One of the earliest detailed one-dimensional model is presented by Erikson and Beckstead [26]. A splitting method is implemented to integrate the gas phase equation, using the ICE scheme [27] to compute the pressure and velocity fields with an implicit scheme of first-order accuracy in time. The stiff chemical source terms are handled with DVODE [28]. The solid phase energy equation is integrated implicitly. The surface temperature

and regression speed are then iterated upon until the interface conditions are met, each time performing the split integration of both phases. Due to poor computational performance and large splitting errors, they transition in [29] to a fully implicit resolution of the gas phase, using the TWOPNT [30] algorithm to solve the system discretised in time with the first-order implicit Euler method. The authors mention the attempt to use DASSL [31] instead, which is a high-order adaptive method based on a backward-differentiation formula, however difficulties led them to use TWOPNT most of the time. Both phases still need to be iterated upon at each time step. V. Yang and his colleagues (see the review [15]) used a similar approach with an iterative coupling of both phases. The solid phase species equations, the gas phase continuity and species equations are solved explicitly with the fourth-order RK4 method, while the remaining equations are solved with PREMIX [32], which uses the first-order implicit Euler scheme. A variant of this approach [33] solves the gas phase implicitly with a dual time step to improve the convergence of the Newton solver, however implicit Euler is still the method used for the time integration. Meredith and Beckstead [16] and Smyth [14] use the same iterative approach and the same resolution method for the condensed phase as Erikson and Beckstead, but the solid phase is allowed to take multiple smaller steps to improve convergence. The gas phase is solved implicitly with a Newton algorithm and time integration methods provided in the PETSc library [34]. No precision is given on the order and properties of the chosen methods. To our knowledge, most of the one-dimensional unsteady CFD codes for the combustion of a solid propellant use a similar iterative procedure between the different phases, with a time integration based on splitting and/or implicit methods, which are usually limited to first-order accuracy in time. No mathematical analysis has been reported regarding the nature of the system of coupled equations obtained after semi-discretisation in space, where the handling of some variables (e.g. surface temperature) is usually problematic. Besides, relying on low-order integration methods may prove disadvantageous in terms of accuracy, performance and ability to resolve fine dynamics. High-order methods are especially important when investigating instabilities and non-linear behaviours, e.g. limit cycles, where growth of some modes can only be captured by high-fidelity numerical methods [35].

Even if solid propellant combustion brings in additional difficulties and constraints related to the heterogeneous nature of the flow, homogeneous combustion or two-phase flow combustion for one-dimensional low-Mach flows involve in fact the same problematic. We briefly mention it since the results provided in this paper can equally be applied to these fields of research. Solving for steady and unsteady homogeneous and spray combustion in one-dimensional flame simulations has attracted enormous attention in the combustion community starting with the seminal work conducted by M.D. Smooke and collaborators between Sandia and Yale University [32, 36, 37, 38, 39]. The main use of low-order time integrators as well as relatively ad hoc treatment of algebraic constraints lead to similar conclusions compared to the observation in the previous paragraph, except for the work on the dynamics of non-premixed counterflow flames [40]. In this interesting piece of work, high-order time integrators for differential-algebraic equations (DAE) were introduced, but the constraint formulation was rather involved (introduction of compressibility effects - link with the index) and the details of the convergence / efficiency were not the main purpose of the paper.

The present contribution focuses on the one-dimensional solid propellant combustion in the Low-Mach number approximation, and introduces a high-order time integration strategy based on singly-diagonally implicit Runge-Kutta methods, with a fully coupled approach involving algebraic constraints. The accuracy on the algebraic

variables, i.e. surface variables and flow velocity, is guaranteed by requiring certain specific properties for the integration scheme.

In the first part of the paper, we focus on the mathematical nature of the set of coupled equations obtained after semi-discretisation in space. We show that it is a differential-algebraic system of index one. The knowledge of this particular property is decisive when looking for high-order time integration methods. Many suited methods are reported in the literature for this particular class of problems<sup>1</sup> [42, 31]. In the second part of this paper, a progressive description of the different requirements for a high-accuracy time integration method is given, leading to the choice of a family of singly-diagonally implicit Runge-Kutta methods with embedded lower-order solutions providing native time adaptation capabilities based on objective mathematical criteria. Other methods may be applied, however Runge-Kutta methods are preferred, as they usually offer strong convergence and stability properties and allow for a natural treatment of the mass conservation algebraic constraint. A one-dimensional CFD code that can accurately compute the time-dependent solution of simplified combustion models is then presented and the method and its implementation are verified on two test-cases, where we have analytic or quasi-analytic solutions: steady-state solution and response of the solid propellant combustion to pressure oscillations, a classical problem in the field. In order to investigate the potential and efficiency of the method, we tackle three challenging test-cases. First, we focus on the simulation of ignition transients with a simplified modelling, showing that the native time adaptation capability of embedded methods is more performant than traditional CFL- or variation-limited time step evaluation strategies. High-order adaptive methods provide more accurate results than traditional low-order methods for similar computational times. Second, we identify physical parameters for which we expect an unstable behaviour of the steady-state solution through a Hopf bifurcation, thus exhibiting a nonlinear growth toward a limit-cycle periodic solution. A procedure is detailed for the generation of such a configuration based on the Zeldovich-Novozhilov (ZN) analysis [2] and an optimisation tool we have designed. Simulations of the growth of the instability toward the limit cycle show that high-order time integration methods offer a major improvement over low-order methods, in terms of quality of the result, robustness, and ability to capture such dynamics. If a precise temporal resolution is sought, these methods also offer important benefits in both computational time and amount of engineering work required to setup and conduct an accurate simulation of such non-linear systems, the time response of which is a key issue for applications. Eventually, the proposed numerical strategy is tested with the unsteady combustion of an ammonium perchlorate (AP) monopropellant with detailed kinetics involving 25 species and 80 reactions in the gas phase. Even if this case is mainly presented as a proof of concept for the numerical strategy and cannot be considered as a very accurate and relevant model in terms of physics, it allows us to show that the convergence orders and efficiency of the method are not affected by the increased complexity and stiffness of the detailed modelling level. Important gains in the quality of the result, ease of simulation setup, and computational times are also observed. We can then conclude and assess the numerical strategy proposed in the paper.

---

<sup>1</sup>It is important to stress the fact that, when the problem is considered in more than one dimension, the system may become differential-algebraic of index two, in particular due to the pressure field [31], requiring a deeper analysis of the problem and more advanced time integration methods [41]. However, the main results of the one-dimensional case should remain applicable.

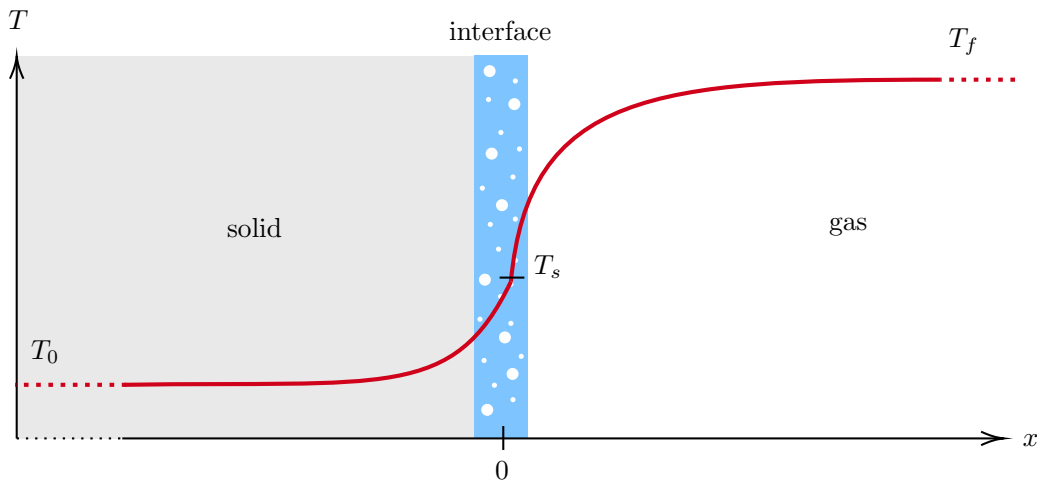
## 2. Formulation of the discrete unsteady problem

In this section, we first present the modelling assumptions and derive a 1D mathematical model of solid propellant combustion. A method of lines based on finite volumes is used to obtain the set of semi-discrete equations.

### 2.1. General modelling

The temperature profile of the one-dimensional combustion of a solid propellant is presented schematically in Figure 1. The space variable is  $x$ . The solid phase represents the propellant and is semi-infinite towards  $x = -\infty$ . As the surface heats up, it is generally agreed that thermal decomposition occurs inside the solid [14]. However, due to the high activation energy of this process, the pyrolysis is often assumed to be concentrated in an infinitely thin zone at the surface, while the propellant remains inert. Experiments have shown that many solid propellants exhibit a thin liquid layer with gaseous bubbles (“foam” layer), typically a few micrometers thick for AP-based propellants [14]. For HMX/RDX propellants at ambient pressure, the melt layer can be around 200 micrometers thick [43], while the gas flame thickness is around 5 millimeters. At higher pressure, both thicknesses decrease [44]. Due to difficulties with the experimental determination of the melt layer properties, many models have assumed this melt layer to be infinitely thin as well [45, 13, 46]. Overall, the modelled interface is the location of a singularity, where all decomposition and gasification phenomena are gathered. We follow the same reasoning for simplicity, however the addition of a liquid phase and in-depth decomposition reactions should not affect the considerations discussed in Section 3. Note that due to computational costs, all multi-dimensional tools for the solid propellant combustion [17, 18, 19, 20] also use these interface assumptions.

Flows near the surface of a solid propellant are generally at high temperatures but slow velocity (1-10 m/s), therefore the gas phase is modelled as a low-Mach one-dimensional reactive flow with  $n_e$  species and uniform pressure  $P$ . Radiative emission from the gas phase and surface are neglected. The inert solid phase is simply modelled with the heat equation.



**Figure 1** One-dimensional model of solid propellant combustion

We recall the equations for both phases and their coupling at the interface. They are originally expressed in a Galilean reference frame, where the interface position  $\sigma$  varies in time:  $d_t\sigma = -r$  with  $r \geq 0$  the surface

regression rate. We perform the simple variable change  $x = x_{galilean} - \int d_t \sigma dt$ , so as to keep the interface at  $x = 0$ . This introduces a convective term in the solid phase heat equation, which represents the interface regression. The temperature field  $T$  in the solid phase at  $x < 0$  is subject to:

$$\rho_c c_c \partial_t T + \rho_c c_c r \partial_x T - \partial_x (\lambda_c \partial_x T) = 0 \quad (1)$$

with  $\rho_c$  the propellant density,  $c_c$  its heat capacity,  $\lambda_c$  the thermal conductivity. Far below the surface, the solid is at its resting temperature:

$$T(-\infty) = T_0 \quad (2)$$

The gas phase at  $x > 0$  is subject to the following partial differential equations:

$$\begin{cases} \partial_t \rho + \partial_x \rho(v+r) = 0 & (3) \\ \partial_t \rho Y_{\mathfrak{k}} + \partial_x (\rho(v+r) Y_{\mathfrak{k}}) = -\partial_x J_{\mathfrak{k}} + \omega_{\mathfrak{k}} \quad \forall \mathfrak{k} \in [1, n_e] & (4) \\ \partial_t \rho h + \partial_x (\rho(v+r) h) = -d_t P + \partial_x (\lambda \partial_x T) - \partial_x (\sum_1^{n_e} h_{\mathfrak{k}} J_{\mathfrak{k}}) & (5) \end{cases}$$

Here  $Y_{\mathfrak{k}}$  is the mass fraction of  $\mathfrak{k}$ -th species, and  $J_{\mathfrak{k}}$  is the corresponding diffusion flux, approximated by a generalised Fick's law:  $J_{\mathfrak{k}} = \rho \sum_{j=1}^{n_e} D_{\mathfrak{k}j} \partial_x Y_j$ , where the  $D_{\mathfrak{k}j}$ 's are the components of the species diffusion matrix. The volumetric production rate of the  $\mathfrak{k}$ -th species is  $\omega_{\mathfrak{k}}$ . The enthalpy  $h$  is the sum of the chemical and sensible enthalpies:  $h = \sum_{\mathfrak{k}=1}^{n_e} Y_{\mathfrak{k}} h_{\mathfrak{k}}$ , where  $h_{\mathfrak{k}} = (\Delta h_{f,\mathfrak{k}}^0 + \int_{T_0}^T c_{p,\mathfrak{k}}(a) da)$ , with  $c_{p,\mathfrak{k}}$  the heat capacity of the  $\mathfrak{k}$ -th species, and  $\Delta h_{f,\mathfrak{k}}$  its formation enthalpy at  $T_0$  the standard temperature. Soret and Dufour effects are neglected. The following conditions are automatically satisfied at infinity if we are looking for smooth enough solutions and cannot be considered strictly speaking as boundary conditions:

$$\partial_x T(-\infty) = 0, \quad \partial_x T(+\infty) = 0, \quad \partial_x Y_{\mathfrak{k}}(+\infty) = 0 \quad \forall \mathfrak{k} \in [1, n_e] \quad (6)$$

Both phases are coupled at the interface by the following conditions, expressing the continuity of the mass flow rate and temperature, as well as the thermal and species flux balance around the interface:

$$\begin{cases} \rho_c r = \rho(0^+)(v(0^+) + r) & (7) \\ T(0^-) = T(0^+) = T_s & (8) \\ (mh - \lambda_c \partial_x T)_{0^-} = (mh - \lambda \partial_x T - \sum_1^{n_e} h_{\mathfrak{k}} J_{\mathfrak{k}})_{0^+} & (9) \\ (mY_{in,j,\mathfrak{k}})_{0^-} = (mY_{\mathfrak{k}} + J_{\mathfrak{k}})_{0^+} \quad \forall \mathfrak{k} \in [1, n_e] & (10) \end{cases}$$

with  $m$  the mass flow rate ( $\rho_c r$  in the solid,  $\rho(v+r)$  in the gas) and  $Y_{in,j}$  the product mass fractions generated by decomposition and gasification processes, which can be constant or functions of the surface temperature as in [13].

The ideal gas law relates the various state variables in the gas phase:

$$\rho = P / \left( RT \sum_{\mathfrak{k}=1}^{n_e} \frac{Y_{\mathfrak{k}}}{\mathcal{M}_{\mathfrak{k}}} \right) \quad (11)$$

with  $\mathcal{M}_\mathfrak{k}$  the molar mass of the  $\mathfrak{k}$ -th species. The surface pyrolysis mass flow rate is given by the pyrolysis law:

$$m(0) = \rho_c r = f(T_s, P) \quad (12)$$

A few remarks may be useful. First, in this one-dimensional framework, the momentum equation is redundant, as the mass flow rate spatial variation is already determined by the continuity equation from the  $\rho$  temporal evolution, which itself is known from the temporal variations of  $T$  and  $Y_\mathfrak{k}$  and the thermodynamic pressure  $P$  through the equation of state (11). Hence  $\rho$ , although its time derivative is specified by the continuity equation, is not a true variable. As we will see further on in Section 3, the continuity equation only acts as a constraint that determines the velocity field. The momentum equation could be used to determine the hydrodynamic pressure field, however it is known that this pressure perturbation is of the order of  $\text{Ma}^2$ , with  $\text{Ma}$  the Mach number. In solid propellant flames, the Mach number is typically of the order of  $10^{-3}$ , hence the pressure perturbation can be completely neglected.

Second, the variable change we have performed is not a change of reference frame, therefore no inertial body forces appear. Indeed the gas-phase velocity  $v$  is still the one observed in the original Galilean reference frame. We see that all the convective terms involve  $(v + r)$  and  $v$  never appears on its own. Consequently in the rest of the paper we replace it with  $u = v + r$ , which is the gas velocity relative to the interface. Furthermore, using the gas mass flow rate  $m = \rho u$  further simplifies the notations.

Third, all the partial differential equations presented above are in conservative form, which we will rely on for the finite volume discretisation. However, for the theoretical analysis, we will replace equation (5) for the enthalpy  $h$  with the simpler non-conservative equation for the temperature, allowing for a more direct analysis:

$$\rho c_p \partial_t T + \rho c_p (v - r) \partial_x T - \partial_x (\lambda \partial_x T) + \sum_{\mathfrak{k}=1}^{n_e} J_\mathfrak{k} \partial_x h_\mathfrak{k} = - \sum_{\mathfrak{k}=1}^{n_e} h_\mathfrak{k} \omega_\mathfrak{k} \quad (13)$$

The interface thermal balance equation (9) is also replaced by an equivalent equation for the temperature  $T$ :

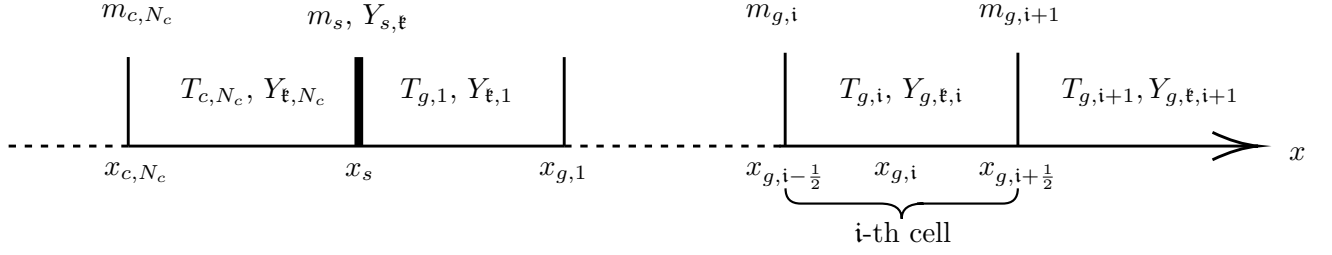
$$(\lambda_c \partial_x T)_{0-} = m Q_s + (\lambda \partial_x T)_{0+} \quad (14)$$

with  $Q_s$  the heat of reaction associated with the pyrolysis and gasification processes (surface reactions), which depends on the temperature  $T_s$  and surface mass fractions  $Y_{s,\mathfrak{k}}$ .

## 2.2. Concept of eigenvalue in steady-state

In the case where the kinetic mechanism is reduced to two species with one potentially reversible reaction, and if the gas phase has a unitary Lewis number, we have shown in [47] that the steady-state travelling wave solution exists and is unique. The steady-state problem belongs to the class of nonlinear eigenvalue problems. The eigenvalue is the regression speed  $r$ , or equivalently the surface temperature  $T_s$ , as both are related through the pyrolysis law (12). The associated eigenfunction is the temperature profile  $T(x)$ . The problem can be solved separately in each phase for any value of  $r$ , but the interface condition (14) can only be satisfied for a unique value of  $r$ . This interface boundary condition can be considered an algebraic constraint and, as we will see later on, understanding this notion in steady-state allows for a proper mathematical interpretation of the unsteady problem.





**Figure 2** Localisation of the discretised variables in the finite volume mesh. The thick vertical line represents the interface.

### 2.3. Discretisation with a finite volume approach

For the numerical implementation of a solid propellant combustion model, we apply the method of lines to obtain a set of discrete evolution equations. The semi-discretisation in space is obtained with a finite volume approach, however other approaches could be applied without affecting the conclusions drawn in this paper.

#### 2.3.1. Gas phase

The set of conservative equations for the gas phase is semi-discretised in space with a finite volume approach: the domain is split in  $N_g$  cells (control volumes). The discretised variables are the temperature  $T$ , the mass fractions  $Y_{\xi}$ , the mass flow rate  $m = \rho u$ . The temperature and mass fractions are taken at the centers of each cell, while the mass flow rate is taken at the left face of each cell. This staggered-grid approach helps decreasing the numerical discretisation error, and is convenient for the free boundary problem with a flux defined at the surface. The surface temperature  $T_s$ , the surface mass fraction of the  $\xi$ -th species  $Y_{s,\xi}$  are taken at  $x = 0$  (rightmost face of the solid domain, leftmost face of the gas domain). The localisation of each variable is sketched in Figure 2.

Using the notation  $q$  to identify any of the conservative variables:  $\rho Y_{\xi}$  and  $\rho h$ , or  $\rho$ , and the subscript  $i$  as the index of the mesh cell considered, the conservative equations (3) to (5) become:

$$\frac{dq_i}{dt} = [F_{d,q} + F_{c,q}]_{i-\frac{1}{2}}^{i+\frac{1}{2}} + s_{q,i} \quad (15)$$

with  $F_{d,q}$  the diffusive fluxes,  $F_{c,q}$  the convective fluxes and  $s_{q,i}$  the source term.

Thermodynamic and transport properties are evaluated at the cell centers, and their values at the interface are taken as averages of the adjacent cells values. The gradient at the  $(i - \frac{1}{2})$ -th interface of a variable  $q$  discretised at the cell centers is:

$$\nabla q_{i-\frac{1}{2}} = \frac{q_i - q_{i-1}}{x_i - x_{i-1}}$$

The interface values of the transported variables are computed via a numerical scheme. For any conservative variable  $q$ , its interface value at the abscissa  $x_{i-\frac{1}{2}}$  is defined as:

$$q_{i-\frac{1}{2}} = \Phi_{i-\frac{1}{2}}^- q_{i-1} + \Phi_{i-\frac{1}{2}}^+ q_i$$

where  $\Phi_{i-\frac{1}{2}}^-$  and  $\Phi_{i-\frac{1}{2}}^+$  are the scheme coefficients and sum up to one. If both these coefficients are equal to 0.5, the scheme is centered. This scheme is the only second-order accurate scheme with a 2 point stencil. MUSCL-type scheme need at least a 3 point stencil. Due to stability requirements, the centered scheme needs to be locally

upwinded if the flow is convection-dominated, ensuring stability at the expense of falling back to first-order accuracy. This is done dynamically via a Péclet-weighted average of the first order upwind and second order centered schemes, similar to what is done in [48] with the concept “mesh Reynolds number”. The local Péclet number at the center of the  $i$ -th cell is  $Pe_i = c_{p,i} m_i / \lambda_i$ , where the reference length is chosen unitary. The Péclet number at the  $(i - \frac{1}{2})$ -th interface is computed as:

$$Pe_{i-\frac{1}{2}} = \frac{1}{2} (Pe_i + Pe_{i-1}) (x_i - x_{i-1})$$

i.e. it is an average Péclet number with a reference length taken as the distance between the centers of the neighbouring cells. If  $|Pe_{i-\frac{1}{2}}| < 0.5$ , we use the centered scheme:  $\Phi^+ = \Phi^- = 0.5$ , i.e. the centered scheme is used as the thermal diffusion is locally the dominating form of energy transfer compared to convection. If  $|Pe_{i-\frac{1}{2}}| > 1$ , we use the upwind scheme:  $\Phi^+ = 0$  if  $Pe > 0$ , else  $\Phi^+ = 1$ . The transition between these two cases is smooth with respect to  $Pe_{i-\frac{1}{2}}$ , so as to not cause numerical issues later on. This allows to use a second-order accurate scheme where the mesh is sufficiently refined. The first order scheme is only used in poorly resolved areas far away from the surface, where precise representation of the flow is not required.

Convective fluxes are then computed as:

$$F_{c,q,i-\frac{1}{2}} = m_i q_{i-\frac{1}{2}}$$

Diffusive fluxes are approximated by a second-order centered scheme. Source terms are evaluated at the cell centers.

The final semi-discrete equations for the evolution of the conservative variables in the  $i$ -th cell read as follows:

$$\Delta x_i \frac{d\rho_i}{dt} = m_i - m_{i-1}, \quad \Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}} \quad (16)$$

$$\begin{aligned} \Delta x_i \frac{d\rho_i Y_{\ell,i}}{dt} = & m_i \left( \Phi_{i-\frac{1}{2}}^+ Y_{\ell,i} + \Phi_{i-\frac{1}{2}}^- Y_{\ell,i-1} \right) - m_{i+1} \left( \Phi_{i+\frac{1}{2}}^+ Y_{\ell,i+1} + \Phi_{i+\frac{1}{2}}^- Y_{\ell,i} \right) \\ & + J_{\ell,i-\frac{1}{2}} - J_{\ell,i+\frac{1}{2}} + \Delta x_i \omega_{\ell,i} \quad \forall \ell \in [1, n_e] \end{aligned} \quad (17)$$

$$\begin{aligned} \Delta x_i \frac{d\rho_i h_i}{dt} = & \Delta x_i d_t P + m_i \left( \Phi_{i-\frac{1}{2}}^+ h_i + \Phi_{i-\frac{1}{2}}^- h_{i-1} \right) - m_{i+1} \left( \Phi_{i+\frac{1}{2}}^+ h_{i+1} + \Phi_{i+\frac{1}{2}}^- h_i \right) \\ & - \frac{\lambda_{i-1} + \lambda_i}{2} \frac{T_i - T_{i-1}}{x_i - x_{i-1}} + \frac{\lambda_i + \lambda_{i+1}}{2} \frac{T_{i+1} - T_i}{x_{i+1} - x_i} \\ & + \sum_{\ell=1}^{n_e} \left( \frac{h_{\ell,i} + h_{\ell,i-1}}{2} J_{\ell,i-\frac{1}{2}} - \frac{h_{\ell,i+1} + h_{\ell,i}}{2} J_{\ell,i+\frac{1}{2}} \right) \end{aligned} \quad (18)$$

### 2.3.2. Solid Phase

The solid phase energy equation (1) is replaced by a conservative equation for the enthalpy and discretised in the same way as the conservative equations in the gas phase. The solid mesh contains  $N_c$  cells. As explained further in Section 4.3, the block-tridiagonal Jacobian of the complete system leads to an improved computational efficiency. In order to keep a consistent Jacobian structure between the gas and solid phases, the mass flow rate field  $m$  and

the species profile  $Y_{\mathfrak{k}}$  are kept as dummy variables. As the solid propellant is assumed inert and incompressible, the continuity equation is equivalent to  $d_x m = 0$  which is discretised as  $m_i = m_{i+1}$ , with the boundary condition  $m_{N_c+1} = m(T_s, P)$ . Species evolution is not considered in the solid phase, therefore the simple equation  $Y_{\mathfrak{k},i} = 0$  is used, for all  $i \in [1, N_c]$ , and  $\mathfrak{k} \in [1, n_e]$ .

### 2.3.3. Surface

The surface variables are the surface temperature  $T_s$  and the surface mass fractions  $Y_{s,\mathfrak{k}}$  on the gas side. The surface matching conditions (7), (10) and (14) and the continuity equation (3) are discretised as follows, with a first-order approximation of the gradients:

$$\begin{cases} 0 = g_{th} := -\lambda_c \frac{T_s - T_{c,-1}}{x_s - x_{c,1}} + \lambda \frac{T_{g,1} - T_s}{x_{g,1} - x_s} + m(T_s) Q_s(T_s) & (19) \\ 0 = g_{sp,\mathfrak{k}} := m(T_s)(Y_{in,j,\mathfrak{k}} - Y_{s,\mathfrak{k}}) + J_{s,\mathfrak{k}} \quad \forall \mathfrak{k} \in [1, n_e] & (20) \end{cases}$$

with  $T_{c,-1}$  the temperature in the last cell of the solid phase below the surface, and  $T_{g,1}$  the temperature in the first cell of the gas phase, just above the surface. The species surface diffusion fluxes are computed as:

$$J_{s,\mathfrak{k}} = \rho(T_s, Y_s, P) \sum_{j=1}^{n_e} D_{\mathfrak{k}j}(T_s) \frac{Y_{g,i,1} - Y_{s,j}}{x_{g,1} - x_s}$$

## 3. Differential-algebraic nature of the semi-discretised system

We aim at developing a 1D unsteady CFD tool for high-fidelity simulations of transient phenomena. Relying on a finite-volume space discretisation, we have obtained a system of ODEs. In this section, we show that a difficulty arises from the nature of this system: some variables are not defined by differential, but only through algebraic equations. The system thus belongs to the class of Differential-Algebraic Equations (DAE). We refer the reader to [42, 31] for details on DAEs. Only the necessary aspects of this class of problem will be discussed hereafter.

### 3.1. Identification of the constraints

The discretised surface variables  $Y_{s,\mathfrak{k}}$  and  $T_s$  only appear in equations (19) and (20), however no time derivative appear. Such variables are called algebraic and will “instantly” adapt to the variations of the other variables in the cells adjacent to the surface, i.e. they are not directly affected by their time histories.

The remaining algebraic equations come from the discrete gas continuity equation (16), which we recall here:

$$\begin{cases} 0 = g_{m,1} := m_1 - m(T_s) & (21) \\ 0 = g_{m,i} := \frac{d\rho_i}{dt} + \frac{m_i - m_{i-1}}{x_i - x_{i-1}} \quad \forall i \in [2, N_g] & (22) \end{cases}$$

Equation (21) is the boundary condition for the gaseous mass flow rate field and is equivalent to equation (7).

As underlined in Section 2.1, the density  $\rho_i$  in the  $i$ -th cell is a function of  $T_i$  and  $Y_{\mathfrak{k},i}$  via the ideal gas law (11). These variables are governed by the discretised form of equations (4) and (13) and by the pressure  $P$ , assumed spatially constant and given as an input to the model. Therefore  $\rho_i$  is not a true differential variable. Its time

derivative  $d_t \rho_i$  appearing in equation (22) is entirely determined from the variations of the other gas-phase variables  $T_i$  and  $Y_{\mathfrak{t},i}$ . The continuity equation is solely used to constrain the flow rate field in the gas phase  $m$ , the time derivative of which does not appear in this one-dimensional low-Mach model. As a consequence, the discrete values of the mass flow rate are also algebraic variables, which adapt instantly to variations of the other variables in coherence with the parabolic nature of the low-Mach number limit, where all pressure waves propagate at infinite speed and are relaxed instantly.

### 3.2. Analogy with a singular perturbation problem

To better understand the origin of the appearance of this algebraic character, and for the sake of the example, let us focus on the surface temperature. If we relax the tacit assumption that no accumulation of mass or energy takes place at the surface, we can derive a differential equation for  $T_s$ . Let us for instance assume that the interface is a thin volume of thickness  $\delta x$  with uniform temperature  $T_s$ , heat capacity  $c_s$  and density  $\rho_s$ , and is able to accumulate energy. The interface energy condition (14) becomes:

$$d_t(\rho_s c_s T_s) = \frac{1}{\delta x} [-\lambda_c \partial_x T(0^-) + \lambda \partial_x T(0^+) - m(T_s) Q_s(T_s)] \quad (23)$$

If we consider the surface properties  $\rho_s$  and  $c_s$  constant, and introduce the parameter  $\epsilon = \rho_s c_s \delta x$ , we obtain:

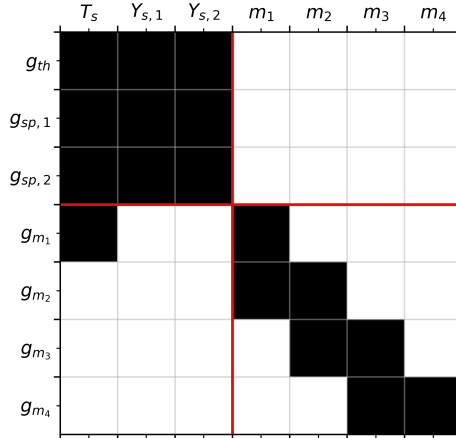
$$\epsilon d_t T_s = -\lambda_c \partial_x T(0^-) + \lambda \partial_x T(0^+) - m(T_s) Q_s(T_s) \quad (24)$$

When  $\delta x$  tends to 0 (infinitely thin interface), we perform a singular perturbation on the previous equations ( $\epsilon \rightarrow 0$ ). The time derivative of  $T_s$  disappears and the interface thermal balance reduces to equation (14). The equation, originally of the differential type, becomes algebraic; this is a classical result of singular perturbation theory, which is also valid for the surface mass fractions  $Y_{s,\mathfrak{t}}$ . This simplified analysis helps to clarify the origin of the algebraic character of equations (10) and (14): no accumulation of mass or energy in the infinitely thin interface. The algebraic nature of the gas-phase mass flow rate  $\rho u = m$  comes from the zero-Mach limit with uniform thermodynamic pressure.

### 3.3. Index of the algebraic equations

The presence of algebraic equations usually brings in various types of additional difficulties in the numerical resolution of the system of equations. A useful mathematical criterion, the ‘‘index’’, is used to classify the different types of algebraic equations and is essential in order to choose the proper numerical method. It is defined as the number of times a given algebraic equation must be differentiated with respect to time to obtain a differential equation for the corresponding algebraic variable [31]. Algebraic equations obtained via singular perturbations are generally of index 1, therefore following the reasoning of Section 3.2 we expect  $T_s$  and  $Y_{s,\mathfrak{t}}$  to be defined by index 1 algebraic equations.

We can verify that it is indeed the case by deriving our discretised equations (19) to (22) with respect to time. We see that terms in  $d_t T_s$ ,  $d_t Y_{s,\mathfrak{t}}$  and  $d_t m_i$  appear, and that an explicit expression can be obtained for each of them.



**Figure 3** Sparsity pattern of the Jacobian of  $g$ , the vector of constraints

Therefore the index is 1. Note that when differentiating equation (22) with respect to time, the second-derivative  $d_{tt}\rho_i$  appears. It can however be expressed by differentiating the ideal gas law and the other conservation equations.

Another approach is to reorganise our system by clearly separating the algebraic equations from the differential ones. Let  $W$  be the vector of differential variables, containing the discretised values of  $T$  and  $Y_{\ell}$  in each cell, and let  $Z$  be the vector containing our algebraic variables  $T_s$ ,  $Y_{s,\ell}$  and  $m_i$ :  $W = (T, Y_{\ell})^t$  and  $Z = (T_s, Y_{s,\ell}, m_i)^t$ . The system of DAEs can be recast into the following semi-explicit form:

$$\begin{cases} d_t W = f(W, Z) & (25) \\ 0 = g(W, Z) & (26) \end{cases}$$

with  $g = (g_{th}, g_{sp,1} \dots g_{sp,n_e}, g_{m_0} \dots g_{m_{N_g}})^t$  the vector of algebraic equations. In this form, the index of the DAE is known to be 1 if the Jacobian  $\partial_Z g$  is non-singular [42]. We can verify this by forming this Jacobian, however for the sake of simplicity, we only show in Figure 3 its sparsity pattern when considering only 2 species. The labels on the vertical axis describe the constraint being derived, and the labels on the horizontal axis denote the differentiation variable. We can decompose the matrix into smaller blocks following the red lines. The first block on the diagonal corresponds to the Jacobian of the nonlinear system  $(g_{th}, g_{sp,1} \dots g_{sp,n_e})^t = 0$ . A numerical investigation has shown that in all studied cases, the corresponding solution is unique and the Jacobian, invertible. The second block on the diagonal is lower triangular with non-zero elements on the diagonal, therefore it is invertible. Overall we claim that the Jacobian can be considered as invertible, hence the index of the corresponding algebraic equations is 1.

As previously stated, for readability reasons, we have used the thermal interface balance (14) involving the temperature, instead of eq. (9) involving the enthalpy. This however does not change the nature of the system.

Here a parallel may be drawn with the eigenvalue found when investigating the steady-state problem shortly addressed in Section 2.2 and detailed in [47]. In the steady-state problem, the unitary Lewis assumption allows us to drop the mass fractions in the gas phase and at the interface, as they are directly related to the temperature through enthalpy conservation. The continuity equation simply tells us that the mass flow rate in the gas phase

is uniform and equal to the pyrolysis mass flow rate given by the pyrolysis law. The thermal interface condition (14) in the steady case determines the single value of the regression speed  $r$  at the interface, or equivalently the surface temperature  $T_s$ , since both are linked through the pyrolysis law (12), and we thus call it the eigenvalue of the problem. In the unsteady case, the constraints involve a larger number of surface variables (the surface mass fractions  $Y_{s,\text{f}}$  are added to the surface temperature), but also lead to a single regression speed, which can still be called an eigenvalue of the problem. The semi-discretised mass conservation equation (16) leads to a vector of algebraic variables, the discretised mass flow rate values in all the domain, which are directly determined from the regression rate and constitute a vector of eigenvalues evolving with time and uniquely determined as a function of time. This parallel allows to shed some light on the unsteady problem as another nonlinear eigenvalue problem.

#### 4. Requirements for the time integration method

We have shown that our semi-discretised problem is an index 1 differential-algebraic system of equations. We are now looking for a high-order time integration method to maximise the performance of our code, while ensuring high-accuracy. Adaptive time stepping is an additional capability that we seek, so as to minimise the computational time. We focus on Runge-Kutta methods and shortly discuss the applicability of other methods.

##### 4.1. Ensuring high order of convergence for DAEs

###### 4.1.1. Theoretical formulation

Let us consider a generic  $s$ -stage Runge-Kutta integration method applied to equations (25) and (26):

$$\left\{ \begin{array}{l} w_{ni} = W_n + \Delta t \sum_{j=1}^s a_{ij} f(w_{nj}, z_{nj}) \\ 0 = g(w_{ni}, z_{ni}) \\ W_{n+1} = w_n + \Delta t \sum_{i=1}^s b_i f(w_{ni}, z_{ni}) \\ Z_{n+1} = (1 - \sum_{i,j=1}^s b_i \omega_{ij}) Z_n + \sum_{i,j=1}^s b_i \omega_{ij} z_{nj} \end{array} \right. \quad \begin{array}{l} (27) \\ (28) \\ (29) \\ (30) \end{array}$$

where  $w_{ni}$  and  $z_{ni}$  represent the values of  $W$  and  $Z$  at stage  $i$ , and  $a_{ij}$ ,  $b_i$  and  $c_i$  are the coefficients of the Runge-Kutta method, and  $\omega_{ij}$  are the coefficients of the inverse of  $A = (a_{ij})$ , which we assume to exist, i.e. we temporarily restrict ourselves to implicit methods. The same system is obtained in [42], on page 375. Equation (28) indicates that, at each stage, the algebraic variables gathered in  $Z$  are determined such that the constraints are all verified. This can be interpreted as a systematic projection of the algebraic variables onto the set  $\{z \mid g(W, z) = 0\}$ . After all stages are computed, the advancement to the next time step is performed via equations (29) and (30).

An issue arises from equation (30) which does not necessarily ensure that  $g(W_{n+1}, Z_{n+1}) = 0$ , hence a deviation from the correct solution may occur. Runge-Kutta methods that are *stiffly accurate* satisfy the following conditions:  $W_{n+1} = w_{ns}$ ,  $Z_{n+1} = z_{ns}$ , i.e. the last stage is the solution at the next time step. With such methods, we directly obtain  $0 = g(W_{n+1}, Z_{n+1})$ . Non-stiffly accurate methods have their order of convergence severely reduced or might even become unstable when applied to index 1 DAEs [42], whereas stiffly accurate methods retain the same order of

convergence as for ODEs. Therefore such methods are of particular interest for our problem. Implicit Runge-Kutta methods also allow for a natural treatment of the mass conservation constraint, preserving the order of convergence on both differential and algebraic variables; the details are provided in Appendix B.

Finally, to ensure proper convergence in the case of a stiff system, L-stability is a very advantageous property. It ensures that modes with time scales much shorter than the time step are instantaneously relaxed. The singular perturbation analogy from Section 3 shows that DAEs of index 1 may correspond to modes that are infinitely fast ( $\epsilon \rightarrow 0$ ), hence L-stability is also beneficial when integrating DAEs. On the opposite, non L-stable methods may relax these modes on multiple time steps, thus altering the dynamics of the system, or may even diverge.

#### 4.1.2. Examples

The most widely used implicit method is implicit Euler (or backward Euler), which is first-order accurate, stiffly-accurate and L-stable. It is a single-stage method with coefficients  $a_{11} = 1$ ,  $b_1 = 1$ ,  $c_1 = 1$ . If we apply it following the previous formulation, we obtain:

$$\begin{cases} W_{n+1} = W_n + \Delta t f(W_{n+1}, Z_{n+1}) & (31) \\ 0 = g(W_{n+1}, Z_{n+1}) & (32) \end{cases}$$

A classical second-order scheme is the Crank-Nicolson method [49], which yields:

$$\begin{cases} W_{n+1} = W_n + \frac{\Delta t}{2} (f(W_{n+1}, Z_{n+1}) + f(W_n, Z_n)) & (33) \\ g(W_{n+1}, Z_{n+1}) = -g(W_n, Z_n) & (34) \end{cases}$$

Equation (34) may be surprising, however it shows that this method is sensitive to error accumulation on the algebraic variables. In particular, it is crucial that the initial condition satisfies  $g(Y_0, Z_0) = 0$ . This method is not L-stable and may encounter difficulties when applied to stiff systems.

#### 4.2. Optimising the computational cost

When advancing forward in time, equations (27) and (28) must be solved, usually via a Newton algorithm, which iterates on the values  $w_{ni}$  and  $z_{ni}$  for  $i \in [1, s]$ . Fully implicit Runge-Kutta methods are such that all the stages must be solved simultaneously. The very popular stiffly accurate method Radau5 [42], a 3-stage fifth-order fully implicit method based on Gauss-Radau quadrature points, is one such method. It possesses very interesting properties, however if the problem has  $N$  unknowns, each time step requires solving a  $3N \times 3N$  system, which can be rather costly. An interesting subclass of Runge-Kutta methods is the class of diagonally-implicit Runge-Kutta methods (DIRK) [50]. These methods are such that the summations in equations (27) and (28) for the  $i$ -th stage only go up to  $i$  instead of  $s$ , i.e. any stage can be solved by knowing the values at the previous stages. Such methods require more stages (typically twice as many) to reach the same order of convergence as fully implicit methods, however a complete time step only requires the resolution of  $s$  systems of size  $N \times N$ , which is usually more computationally efficient. Therefore we narrow down our choices to DIRK methods.

### 4.3. Newton algorithm

For DIRK methods, the stages can be solved sequentially. For each stage we need to solve equations (27) and (28) for the unknowns  $w_{ni}$  and  $z_{ni}$ . These equations can be combined to form the nonlinear problem  $F(X) = 0$ , with  $X = (w_{ni}, z_{ni})^t$  the vector of unknowns. This problem is solved iteratively using a modified Newton-Raphson method with damping, simply referred to as Newton method in the rest of the article. Starting from an initial guess, the method generates a sequence of iterates  $(X_\alpha)$  such that:

$$X_{\alpha+1} = X_\alpha - \tau_\alpha J^{-1} F(X_\alpha) \quad (35)$$

The Jacobian  $J = \partial_X F$  is only updated when the convergence is poor or if the iterates diverge. It is computed by finite differences. The damping coefficient  $\tau_\alpha$  is initially set to 1. It is reduced as long as the norm of the Newton step is not decreasing, i.e. if  $\|J^{-1}F(X_{\alpha+1})\| > \|J^{-1}F(X_\alpha)\|$ . An error is raised if the Jacobian is computed more than a maximum allowed number of times (typically 5), or if the number of iterations is too high.

The Newton increment  $\Delta X_\alpha = J^{-1}F(X_\alpha)$  is obtained by solving the linear system  $J\Delta X_\alpha = F(X_\alpha)$ . As explained in Section 2.3, the computation of the interface fluxes only relies on two adjacent cells, therefore  $J$  is block-tridiagonal. A Thomas algorithm is used to solve this system, after having performed a block LU-decomposition.

Being able to reuse the Jacobian as many times as possible is important to save computational time. Let us take the example of a simple nonlinear problem  $F(X) = 0$  representing the residuals of a discretised ODE, for instance  $d_t X = f(X)$ , with  $X$  the vector of variables. The equation for the  $i$ -th stage of a DIRK method is:

$$X_{ni} = X_n + \Delta t \sum_{j=1}^i a_{ij} f(X_{nj})$$

We can rearrange this equation by regrouping terms involving the unknown  $X_{ni}$  to one side:

$$X_{ni} - a_{ii}\Delta t f(X_{ni}) = \underbrace{X_n + \Delta t \sum_{j=1}^{i-1} a_{ij} f(X_{nj})}_{R, \text{ independent of } X_{ni}}$$

The corresponding nonlinear problem is  $F(X_{ni}) = 0$  with:  $F(X) = X - a_{ii}\Delta t f(X) - R$ .

For each stage, this nonlinear problem is solved via the Newton method, which generates a sequence of iterates  $(X_\alpha)$  defined recursively by equation (35), taking  $X_n$  as initial condition, or a more precise extrapolation based on former steps. The Jacobian of the nonlinear problem is:

$$J = \left( \frac{\partial F}{\partial X} \right) = \mathbb{I} - a_{ii}\Delta t \left( \frac{\partial f}{\partial X} \right)$$

If the original ODE is not too nonlinear, it is highly likely that  $(\partial f/\partial X)$  will not vary much as  $y$  changes from one iteration to the next. However, if the method is composed of multiple stages and the coefficients  $a_{ii}$  are not equal, the Newton algorithm will need to update the Jacobian at each stage. That is the reason why we focus on singly-diagonally implicit Runge-Kutta methods (SDIRK), which satisfy the property  $a_{ii} = a_{jj} \forall (i, j) \in [1, s]$ .



#### 4.4. Time adaptation

##### 4.4.1. Motivation

A typical ignition transient is shown in Figure 6a for a solid propellant ignited by a laser source. The evolution of the surface temperature is very rapid at the beginning of the propellant heating, before slowing down. At the time of ignition, the temperature rises again quickly before settling to its steady-state value. The ignition transient can be split into successive phases with very different time scales. Consequently, ensuring the time step is adapted to the time scale of each phase is very important to guarantee a precise resolution and to save computational time.

Some simulations presented in the literature use a constant time step, taken as sufficiently low compared to the ignition time [26, 51]. When time adaptation is used, it usually relies on a CFL criteria [14]. CFL limitation may be irrelevant before ignition, as the gas-phase flow velocity is negligible, therefore it may be supplemented with an additional requirement that the relative variation of the solution between two successive time steps is sufficiently small (e.g. 1%). However such an approach requires fine-tuning and does not provide any guarantee regarding the accuracy of the solution. Embedded Runge-Kutta methods provide a local error estimate by comparing two solutions at different orders. The process is described in [52] and is briefly recalled here for completeness.

##### 4.4.2. Embedded methods

An embedded Runge-Kutta method provides two approximations of different orders for the solution. Let us consider a generic scalar ODE  $y' = f(y)$ , with the Runge-Kutta method providing a first approximation  $y_{n+1}$  of order  $p$ , and a second one  $\hat{y}_{n+1}$  of order  $q < p$ . Let us denote as  $y(t)$  the exact solution. Assuming we start with the exact solution  $y(t_n)$  as initial condition for the time step  $n$ , the local errors for the two Runge-Kutta solutions are:  $y_{n+1} = y(t_{n+1}) + \mathcal{O}(\Delta t^{p+1})$  and  $\hat{y}_{n+1} = y(t_{n+1}) + \mathcal{O}(\Delta t^{q+1})$ . Therefore the difference is  $\epsilon(\Delta t) = |y_{n+1} - \hat{y}_{n+1}| = |\mathcal{O}(\Delta t^{p+1}) - \mathcal{O}(\Delta t^{q+1})| \approx \mathcal{O}(\Delta t^{q+1}) = \alpha \Delta t^{q+1}$ , with  $\alpha > 0$  a constant. We can assume the time evolution of the solution is sufficiently well resolved when this difference is smaller than a specified tolerance:  $\epsilon(\Delta t) \leq tol$ . Defining the integration error as  $err(\Delta t) = \epsilon(\Delta t)/tol$ , this is equivalent to  $err(\Delta t) \leq 1$ . If we choose a time step  $\Delta t_1$  such that this condition is not met, we can estimate a time step  $\Delta t_{opt}$  such that the error matches the tolerance:

$$\epsilon(\Delta t_1) = \alpha \Delta t_1^{q+1} > tol, \quad \epsilon(\Delta t_{opt}) = \alpha \Delta t_{opt}^{q+1} = tol$$

Dividing the second equation by the first one yields:  $(\Delta t_{opt}/\Delta t_1)^{q+1} = tol/\epsilon(\Delta t_1)$ , therefore  $\Delta t_{opt} = \Delta t_1 (tol/\epsilon(\Delta t_1))^{1/(q+1)}$ .

If  $\Delta t_1$  is such that the asymptotic regime of convergence for the integration method is already reached, then the estimated local integration error for  $\Delta t = \Delta t_{opt}$  will be close to the prescribed tolerance. This allows the time step to be dynamically reduced or increased, ensuring that the solution is solved at least as precisely as specified, while minimising computational cost. In practice, to avoid overcorrecting the time step, we do not allow it to change by a factor lower than 0.2 or higher than 5 between two successive error estimations. A safety factor of 0.9 is also applied to  $\Delta t_{opt}$  to ensure the tolerance is strictly satisfied. If a time step fails due to floating arithmetic errors or results in non-convergence of the Newton iterates, the current step is started over again with a decreased step length. If the time step becomes smaller than  $10^{-15}$  s, an error is raised to avoid important numerical rounding errors.

In the case of ODEs, the estimated integration error is defined as in [52]. Relative tolerances  $rtol$  and absolute tolerances  $atol$  are defined as scalars and a tolerance vector is defined as  $tol = atole + rtol|y_n|$ ,  $y_n$  the solution at the former time step and  $\mathbf{e}$  the vector of ones (same size as  $y_n$ ). The integration error is then estimated as:

$$err = \left\| \frac{y_{n+1} - \hat{y}_{n+1}}{tol} \right\|_2 \quad (36)$$

where the fraction stands for division element by element. In this article we use the 2-norm.

#### 4.5. Final choice of the method

Considerations on the accuracy of the method for a system of DAE of index 1 has led us to consider stiffly-accurate Runge-Kutta methods. Minimisation of the computational cost due to Newton iterations is ensured by using singly-diagonally implicit methods. Finally, the requirement of native time adaptation capabilities favours embedded methods. Overall, based on the criteria presented so far, we look for embedded stiffly-accurate SDIRK methods. Additionally, improvements in the error estimation for stiff system and DAEs can be obtained if the lower-order embedded method is stiffly-accurate as well, as discussed in [53]. This reference introduces several such schemes with an additional interesting property, which is that the first stage values are equal to the former time step values. This allows to have a “free” stage to improve the accuracy of the method without any additional cost. We retain three schemes from this reference:

- ESDIRK-32A, a four-stage, third-order method with a second-order embedded scheme
- ESDIRK-43B, a five-stage, fourth-order method with a third-order embedded scheme
- ESDIRK-54A, a seven-stage, fifth-order method with a fourth-order embedded scheme.

Other methods may be applicable. Rosenbrock and multistep methods can be applied directly to the semi-explicit form of the system. In particular the DASSL algorithm [31] has been extensively applied to DAE systems of index 1 with great success and may be an interesting alternative to our approach. Another possibility is to take advantage of the fact that the DAE system is of index 1, thus the algebraic variables can be uniquely determined from the differential variables, i.e. we can consider that there exists a function  $p$  such that  $Z = p(W)$  satisfies  $g(W, Z) = 0$ . Usually this relation is not explicitly known, therefore  $Z$  is determined iteratively via a Newton method. The differential variables  $W$  are then governed by the differential equation  $d_t W = f(W, p(W))$ , which can be integrated with any ODE solver, in particular explicit ones. However we have found that our fully implicit approach is able to produce accurate results while having CFL numbers much higher than 1, therefore explicit integration algorithms would be relatively inefficient due to stability requirements. A partially implicit algorithm, e.g. IMEX methods [54], could be used to remove such numerical instabilities induced by the convection operator, however diffusion and reaction operators would also cause stability issues if treated explicitly, thus defeating the purpose of IMEX methods. Splitting methods could be used so that each phenomena (diffusion, convection, reaction) is integrated with an adequate efficient method, however the order of accuracy in time would generally not exceed 2, and additional difficulties may appear when handling the interface conditions.

## 5. Numerical verification

In this section, the complete one-dimensional code is verified in two steps. First, the spatial discretisation is verified in terms of steady-state profiles by comparing it to a semi-analytical model and enforcing the same level of modelling. Second, the time integration is then verified by comparing the numerically determined response of the surface mass flow rate to pressure oscillations with the classical linearised response.

### 5.1. Steady-state solution

We use a simple test case to validate the finite volume discretisation in steady-state. In the case of temperature independent properties in both phases, surface reaction, unitary Lewis number in the gas phase and equal properties (molar mass, heat capacity) for both gaseous species, we have developed a semi-analytical model [47], which solves the complete problem in steady-state and determines the regression speed, surface temperature and solution profiles with arbitrary precision. We use this model to generate a reference steady-state solution, and we observe how the CFD solution converges to this reference as the mesh is refined. This comparison has been presented in details in [47] and we only summarise the most important aspects here for the sake of completeness.

#### 5.1.1. Model parameters

We consider a simple one-dimensional model of an AP-HTPB-Al propellant [47]. The main characteristics of the simplified model are summarised hereafter.

*Solid phase.* The solid phase is composed of the solid species  $P$  and has the following properties:  $\rho_c = 1806 \text{ kg}\cdot\text{m}^{-3}$ ,  $\Delta h_f^o(P) = 0 \text{ J/kg}$  at  $T = 0 \text{ K}$ ,  $c_c = 1253 \text{ J/kg}\cdot\text{K}$ ,  $\lambda_c = 0.65 \text{ W/m}\cdot\text{K}$ . The initial temperature is  $T_0 = 300 \text{ K}$ .

*Surface.* The pyrolysis mass flow rate is computed as:  $m = A_p \exp(-T_{ap}/T_s)$ , with  $A_p = 6.07 \times 10^7 \text{ kg/s}\cdot\text{m}^2$  and  $T_{ap} = 15082 \text{ K}$ . The pyrolysis process converts the solid phase into the gaseous species  $G_1$ .

*Gas phase.* Two global species are considered: the reactant  $G_1$  and product  $G_2$ , which have the same properties except standard enthalpies. Their molar mass is  $\mathcal{M} = 74 \text{ g/mol}$ , and their heat capacities are  $c_p = c_c$ . The standard enthalpies at  $T = 0 \text{ K}$  are  $\Delta h_f^o(G_1) = -1.80 \times 10^5 \text{ J/kg}$  et  $\Delta h_f^o(G_2) = -4.06 \times 10^6 \text{ J/kg}$ . The unique global reaction is  $G_1 \rightarrow G_2$  and irreversible. The reaction rate is computed as:  $\omega = A[G_1] \exp(-T_a/T)$ , with  $A = 435.5 \text{ s}^{-1}$ ,  $T_a = 7216 \text{ K}$ , and  $[G_1]$  the concentration of  $G_1$ . The diffusion coefficients are equal for both species and taken as a linear function of  $T$  such that the Lewis number is one throughout the gas phase. The thermal conductivity is  $\lambda = 0.464 \text{ W/m}\cdot\text{K}$ .

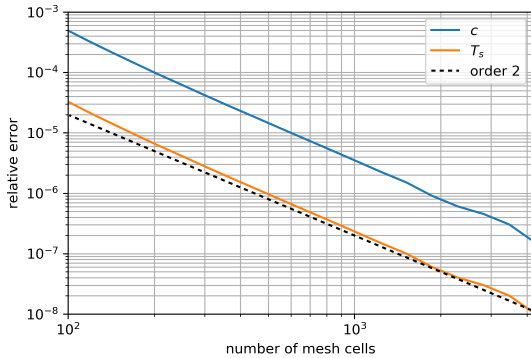
#### 5.1.2. Verification process

First, the complete steady-state problem is solved with the semi-analytical model. Multiple meshes are then generated for the one-dimensional CFD code: knowing the temperature profile from the semi-analytical solution and starting from an initial grid point at  $x = 0$  (interface), the other grid points are placed such that the difference in interpolated temperature between two successive grid points is equal to or below a given threshold. By varying this threshold (from 0.05K to 50K), grids with varying levels of refinement are obtained, whose point distribution is relatively well adapted to the problem. The finite volume mesh is then generated by taking these grid points as

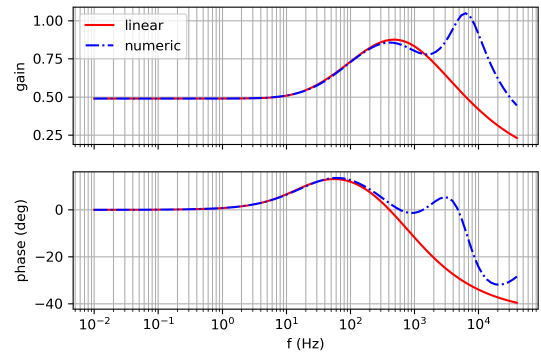
the positions of the cell faces. In this reference case, the thermal layer in the solid phase and the flame in the gas phase both have a thickness close to  $10^{-4}$ m. The generated meshes are extended by adding cells with gradually increasing sizes so that the abscissa of the outer cells are ten times greater than this thickness in order to minimise the influence of the Neumann boundary conditions. It has been verified that extending the mesh further does not change the result. The semi-analytical solution is taken as the initial state and advanced forward in time, until it stabilises on the steady-state solution corresponding to the discrete system relying on the finite-volume method of Section 2.3. This is done efficiently by taking large time steps with implicit Euler, as the initial state is very close to the CFD steady-state solution.

### 5.1.3. Results

We show in Figure 4 the convergence of the CFD result towards the semi-analytical solution for the reference case, as a function of the number of cells. Second-order accuracy in space is attained, and the relative errors reach  $10^{-8}$  on  $T_s$  ( $10^{-7}$  on  $r$  and similar results are obtained on temperature profiles) at around 4000 adapted mesh cells. Achieving a relative error lower than  $10^{-8}$  on  $T_s$  is difficult as this level of error is very close to the tolerance on the Newton step (relative precision of the CFD solution obtained by the Newton solver) and to the precision at which the Jacobian used by the Newton method is approximated. Overall, the error is sufficiently small so that we can consider that the CFD solution is converged in terms of spatial mesh and Newton iterations. The resolution of the steady-state is coherent between the two approaches, thus verifying the spatial schemes used in the finite volume approach.



**Figure 4** Convergence of the steady-state CFD solution towards the semi-analytical solution [47]



**Figure 5** Bode diagram of the response function  $R_{mp}$  to pressure fluctuations

## 5.2. Verification in unsteady regime

We now wish to verify the time integration algorithm implemented in the CFD code. No analytical solution is available in the general unsteady regime, however linear frequency responses to pressure oscillations have been available since decades [9]. They constitute a very interesting benchmark, and will be used here as a point of comparison for the response function evaluated numerically with our CFD code.

### 5.2.1. Process

The linearised response can be constructed analytically using coefficients that represent the sensitivity of the steady-state solution to certain parameters. These sensitivities can be easily generated with finite differences, using

the semi-analytical model or the CFD code. This provides a reference linear response function. The one-dimensional CFD code is then initialised with the steady-state profile and a sinusoidal pressure oscillation  $P(t) = \bar{p} + p' \sin(2\pi ft)$  is forced in the gas phase, with  $f$  a given frequency. The mean pressure  $\bar{P}$  is set to  $= 50 \times 10^5 \text{ Pa}$ , and the amplitude is set to  $p' = 0.001\bar{P}$ . This pressure oscillation leads to fluctuations in the pyrolysis mass flow rate  $m(t) = \bar{m} + m'(t)$ . After a few periods, these oscillations stabilise and we can determine the response function  $R_{mp} = (m'/\bar{m})/(p'/\bar{p})$  at the corresponding frequency. If we assume that the gas phase is quasi-steady, a linearisation of the heat equation in the solid and of the pyrolysis law yields an expression for the response function:

$$R_{mp}(f) = \frac{nAB}{s + \frac{A}{s} - (1 + A) + AB}$$

with  $s = \frac{1}{2} \left( 1 + \sqrt{\frac{1}{2}(y+1)} + i\sqrt{\frac{1}{2}(y-1)} \right)$  and  $y = \sqrt{1 + 16\Omega^2}$ , where  $\Omega = 2\pi f D_c / \bar{r}^2$ ,  $D_c$  is the solid phase thermal diffusivity, and  $\bar{r}$  is the steady-state regression rate. The coefficients  $A$  and  $B$  are defined as:

$$A = (\bar{T}_s - T_0) \left( \frac{\partial \ln(\bar{m})}{\partial \bar{T}_s} \right)_{\bar{P}}, \quad B = \frac{1}{(\bar{T}_s - T_0)\sigma_p}, \quad \sigma_p = \left( \frac{\partial \ln(\bar{m})}{\partial T_0} \right)_{\bar{P}}$$

### 5.2.2. Results

Figure 5 shows the comparison of the linearised and numerical frequency responses. We see that the agreement between both methods is excellent up to approximately 500 Hz, where the gas phase no longer has a quasi-steady behaviour, thus introducing a larger error in the linear response function. This serves as a global verification of our unsteady model. The secondary peak at high frequencies in the response can be obtained analytically if the unsteady gas phase equations are also linearised, as in [55], however this is a much more involved process.

An additional verification of the orders of convergence in time for unsteady simulations is presented in Appendix C, thus completing the verification process in terms of both spatial and temporal discretisations. In particular, algebraic constraints do not hinder the high-order convergence. We now wish to tackle three much more challenging test-cases and investigate the behaviour of the proposed strategy in terms of accuracy and computational efficiency.

## 6. Simulation of ignition transients

To simulate the ignition of a solid propellant, we add a laser heat flux of  $1 \text{ MW.m}^{-2}$ , which is partially absorbed at the surface, as an additional heat flux in equation (14), and partially absorbed in-depth inside the solid as an additional source term in equation (1).

### 6.1. Setup

We use the same simplified model as previously described in Section 5.1.1. The initial solution is a uniform temperature field at 300 K, with only combustion products in the gas phase, as a simpler alternative to adding nitrogen as initial gas, without much effect on the ignition process itself. The mesh has 99 cells in the solid phase and 291 in the gas phase. The cells are distributed such that the steady-state temperature profile is well resolved.

name	number of stages	L-stable	order
ESDIRK-54A	7 (6)	yes	5
ESDIRK-43B	5 (4)	yes	4
ESDIRK-32A	4 (3)	yes	3
CKN (Crank-Nicolson)	2 (1)	no	2
IE (Implicit Euler)	1	yes	1

**Table 1** Selected Runge-Kutta methods. The numbers in brackets correspond to the number of stages actually solved.

We compute the ignition transient with the methods listed in Table 1. The three ESDIRK methods are taken from [53], and they use the time adaptation strategy presented in Section 4.4. We also test the classic schemes implicit Euler (IE) and Crank-Nicolson (CKN), with a time adaptation based on the requirement that the solution has a relative variation that is below a certain value between two consecutive time steps. A discussion on a CFL-based time adaptation is presented at the end of this section. We use the abbreviation “tol” to refer to the relative integration error tolerance for ESDIRK methods, and to the allowed relative variation of the solution between consecutive time steps for IE and CKN. The maximum time step allowed is 0.1 s.

### 6.2. Physical interpretation of the ignition

Figure 6a shows the evolution of the surface temperature during ignition as computed by ESDIRK-54 with  $\text{tol} = 10^{-6}$ . The first phase is the inert heating of the solid propellant. The constant laser heat flux with partial in-depth absorption results in an evolution of  $T_s$  which is very close to being proportional to  $\sqrt{t}$ , that is coherent with the analytical solution of the surface temperature for a solid under a constant surface heat flux.

When  $T_s$  is sufficiently high, the pyrolysis mass flow rate given by (12) increases rapidly, causing the release of gaseous pyrolysis products in the gas phase, which chemically react and form a flame that heats up the solid even more. Typically at this point, more thermal energy is stored in the solid as compared to steady-state. This results in a momentarily higher regression rate at ignition, seen here near  $t \approx 0.355$  s, which evacuates this excess of solid phase thermal energy. The temperature profile then converges to the augmented steady-state solution, with “augmented” indicating that the laser flux slightly increases the burning rate as compared to the steady-state solution without laser.

### 6.3. Result

We compare mainly the evolution of  $T_s$ , the computational time, and the physical time  $t_{ign}$  at which the surface temperature first exceeds 1000 K. Although not shown here, the curves of  $T_s$  for each method are very similar, except for IE simulations with large tolerances that deviate slightly during the inert heating and ignite a few milliseconds earlier. Figure 6b shows the evolution of the time step for some of the simulations. We observe that, for ESDIRK embedded methods, increasing the order of the method allows for higher time steps to be used throughout the integration while maintaining the same accuracy. For example, the fifth-order method ESDIRK-54A is capable of taking steps 5 times bigger in average than the third-order method ESDIRK-32A. Finally, it is clear that IE needs many more steps to achieve a similar result as the ESDIRK methods.

To more quantitatively assess the efficiency of each method in precisely determining the ignition time, multiple simulations were run with each scheme. For the ESDIRK methods, the relative integration error tolerance was

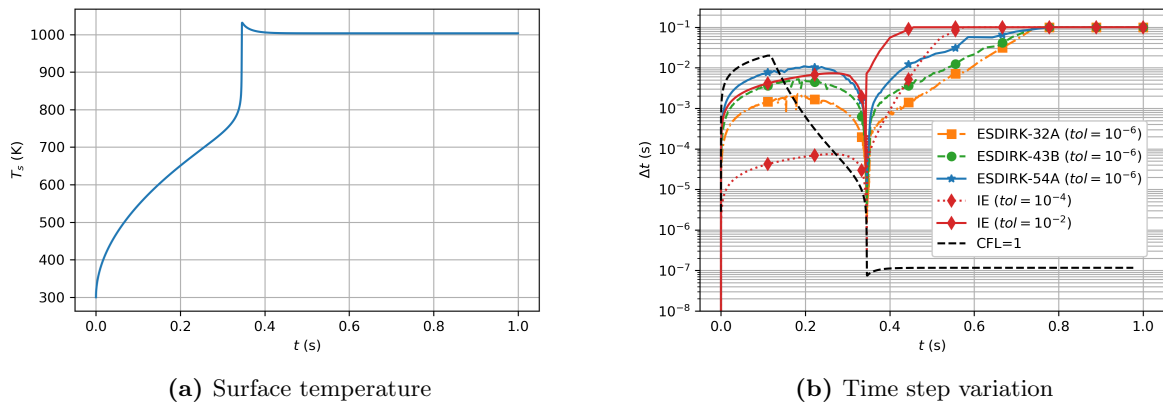


Figure 6 Ignition transient computed with various methods.

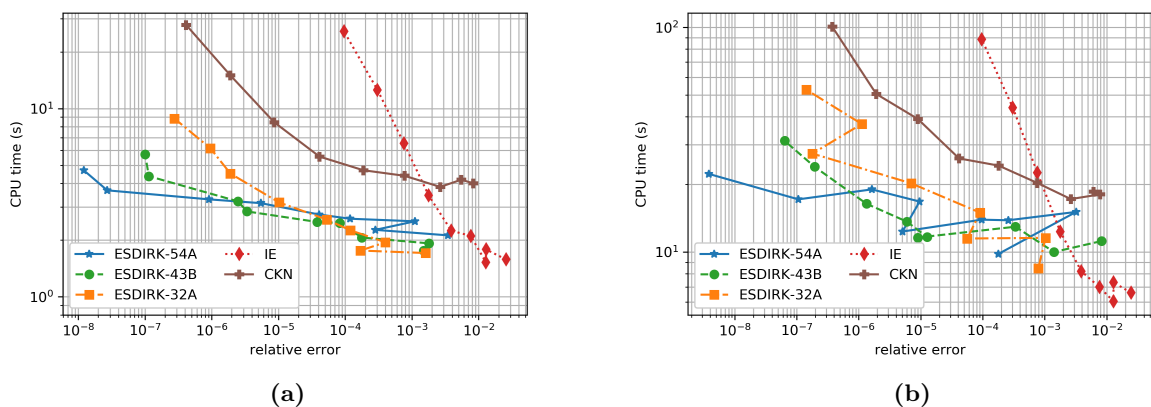


Figure 7 Work-precision diagram for the determination of  $t_{ign}$ : (a) coarse mesh (400 cells), (b) fine mesh (2049 cells).

varied from  $10^{-1}$  to  $10^{-6}$ . For IE and CKN, the relative solution variation allowed between successive time steps was varied from  $10^{-1}$  to  $10^{-4}$ , without any CFL-limitation. For each simulation the value of  $t_{ign}$  is evaluated and a relative error on this value can be inferred by comparing it to the ignition time obtained with ESDIRK-54A and the tightest tolerance. Figure 7a shows the computational time required by each method to achieve a given level of relative error on the initial mesh. Figure 7b shows the computational time required by each method on a refined mesh with 734 cells in the solid phase and 1315 cells in the gas phase. Overall in both cases, if a relatively large error on the ignition time, on the order of 1% is deemed sufficient, IE is a relevant choice. If however greater precision is required, ESDIRK schemes with adaptive time stepping as described in Section 4.4 are much more efficient. CKN does not perform well for the simulation of an ignition transient.

The black dashed curve in Figure 6b shows the time step evolution corresponding to  $CFL = 1$  for the coarse mesh. We clearly see that the CFL-limitation is irrelevant during the inert heating phase, as the mass flow rate is very low. If, after the inert heating, the CFL is to be limited to low values (1 to 100) as is usually done, many more time steps would be performed. ESDIRK methods are able to give very accurate results without any such limitation. We have determined that, if a simulation was to be performed with a maximum CFL of 10 with ESDIRK-54A and  $rtol = 10^{-6}$  on the physical time interval  $[0, 0.35]$  s, i.e. only up to  $T_s \approx 820$  K before the ignition, the simulation

would take 4 times more steps than required without CFL-limitation. If the final physical time is increased to 0.4 s to include most of the transient, the ratio of the number of steps would be 160. Knowing that the simulation without CFL-limitation already achieves an error smaller than  $10^{-6}$  on the ignition time and on the rest of the evolution, this clearly shows that a CFL-constraint is not a good choice in terms of computational efficiency.

The great advantage of embedded methods is that only a relative integration tolerance needs to be specified. No tuning of a CFL-criterion, maximum relative variation or fixed time step is required, hence such methods speed up the engineer task of simulating different scenarios, while still ensuring a controlled error. One interesting observation we made is that the time step values taken by ESDIRK methods were almost identical with both meshes. In all our testing, no correlation was found between the time step evolution of the embedded methods and the mesh refinement. This would not be the case if a CFL-criterion was used.

## 7. Investigation of limit cycles

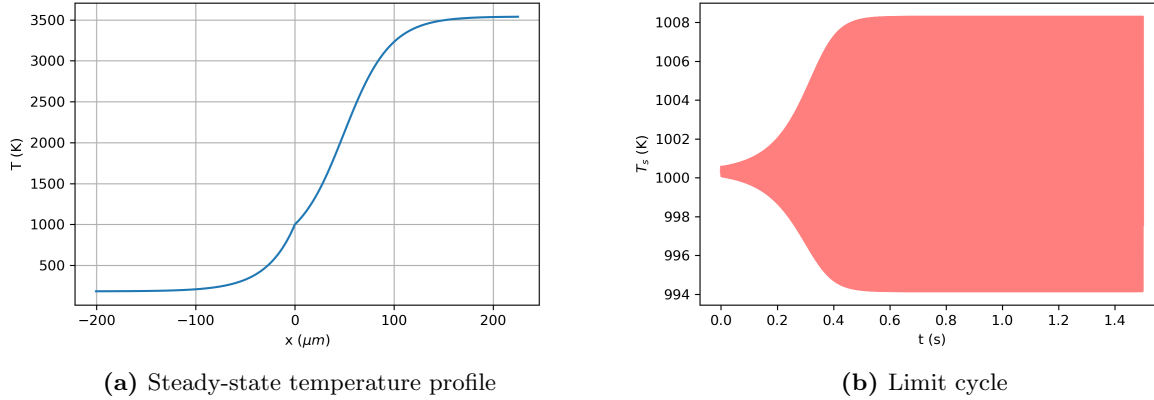
The effort made in terms of time integration strategy can be used to accurately study the non-linear behaviour of the propellant combustion, and in particular potential departures from an unstable steady-state travelling wave solution. Indeed, the time step is dynamically refined so as to closely follow the behaviour of the system, avoiding the excessive numerical damping encountered in low-order schemes, e.g. implicit Euler.

We use an optimisation algorithm in order to generate unstable variants of the simplified combustion model used in Section 6. The process is detailed in Appendix A. Starting from the initial model, physical parameters are varied until a Hopf bifurcation is reached, i.e. a point at which the steady-state travelling wave solution becomes unstable. Around the locus of this type of bifurcation, dynamical systems usually exhibit a limit cycle. We have iteratively determined a configuration that, while retaining realistic characteristics, produces such a limit cycle. The corresponding physical parameters are listed in Appendix A, and the unstable steady-state temperature profile is plotted in Figure 8a. Using this profile as initial condition and enforcing a small pressure perturbation, an unsteady simulation has been performed with time adaptation, with a relative integration error tolerance  $10^{-6}$ . The resulting history of surface temperature is plotted in Figure 8b clearly exhibiting a limit cycle, the discrete Fourier Transform of which is presented in Figure 8c. The limit cycle oscillations are a sum of sinusoidal harmonic oscillations, with a fundamental frequency close to 452 Hz, which is close to the analytical propellant natural frequencies defined in [9] and [45], at 518 Hz and 348 Hz respectively.

As reported in the literature for other applications [35], high-order methods are often needed to be able to numerically reproduce such a dynamical and nonlinear behaviour. It is therefore instructive to compare the methods from Table 1 already used for the ignition transient in Section 6, to see how each of these affects the unsteady result. Namely, it is expected that the integration methods will, depending on their order and the time step used, dampen the oscillating nature of the system and potentially cause a non-physical stabilisation of the solution.

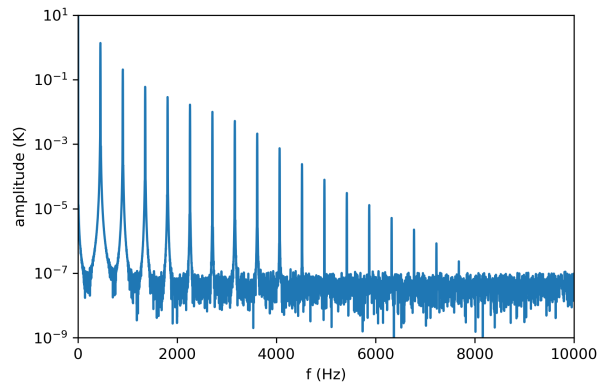
First, constant time steps simulations are performed with the ESDIRK methods and other classical schemes. Comparisons are made based on the ability to reproduce the initial amplification of the oscillations, the fundamental frequency of the limit cycle and its amplitude. Second, simulations with adaptive ESDIRK methods are presented. The quality of the results and the computational efficiency are assessed, both with fixed or variable time steps.





(a) Steady-state temperature profile

(b) Limit cycle



(c) Discrete Fourier Transform of the limit cycle

**Figure 8** Main features of the studied configuration

## 7.1. Method

### 7.1.1. Simulation setup

The integrators used are the stiffly-accurate Runge-Kutta methods described in Table 1. Included are implicit Euler and Crank-Nicolson. The latter has been historically praised for its damping properties. Crank-Nicolson’s absolute stability domain exactly contains all physically stable configurations and this scheme does not damp purely oscillating linear systems. This method actually is part of the ESDIRK class, as its first stage is explicit. Its main drawback is the lack of L-stability, a property which is very advantageous for index 1 DAE problems.

As described in Appendix A, the simulations are performed on a non-uniform mesh based on the temperature profile of the steady-state solution with 55 cells for the solid phase and 146 for the gas phase. It has been verified that additional refinement would not affect the solution dynamics.

### 7.1.2. Comparison process

We focus on several aspects. First we analyse visually the envelope curve of the surface temperature time history. For each simulation, this curve is constructed as the junction of the successive maxima of  $T_s$ . Unfortunately, for large values of the time step  $dt$ , the sampling frequency  $1/dt$  can be insufficiently high compared to the fundamental frequency of the limit cycle, causing artefacts to appear in the form of an oscillation of the envelope. This can be

improved by using a cubic interpolant of  $T_s$  to determine the successive maxima with greater precision, however envelope oscillations are still present, for example in Figure 9e.

Second, we analyse the growth of the linear instability near  $t = 0$ . For each simulation, the best exponential fit for the envelope of the evolution of  $T_s$  is determined, i.e. the curve joining the successive maxima of  $T_s$ . Such a fit is of the form  $T_{fit}(t) - T_s(0) = A \exp(bt)$ , with  $b$  the fitted amplification factor.

Third, the established limit cycle is considered, on the time window  $1 \leq t \leq 1.5$  s. A discrete Fourier transform of the surface temperature signal is computed via an FFT algorithm, as shown for the reference simulation in Figure 8c. Interpolation of the solution on a uniform time grid is performed if the simulation was not conducted with a constant time step. This FFT helps determine the approximate frequencies of the different harmonics with a precision of approximately 1-10 Hz. For each of these peaks in the spectrum, the peak frequency is then precisely computed by maximising the correlation between  $T_s(t)$  and  $\exp(2i\pi ft)$ , from which we can also determine the precise amplitude of the corresponding peak. These values offer a trustworthy and precise indication of how well the limit cycle is captured.

Finally, work-precision diagrams are given which represent the computational time required to achieve a specific level of relative error on the quantitative results, i.e. fitted amplification factor, fundamental frequency and amplitude. Relative errors are computed relative to the values obtained with the most refined solution.

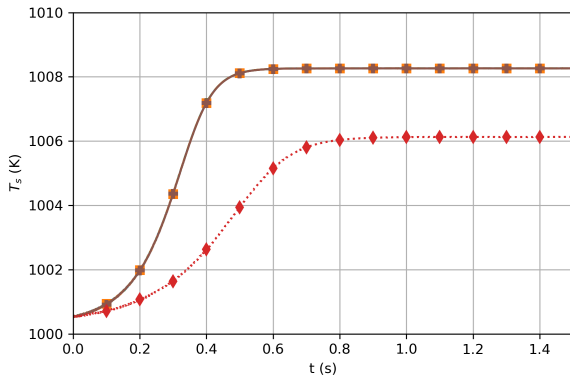
## 7.2. Analysis of schemes efficiency for a constant time step

### 7.2.1. Envelope of the surface temperature history

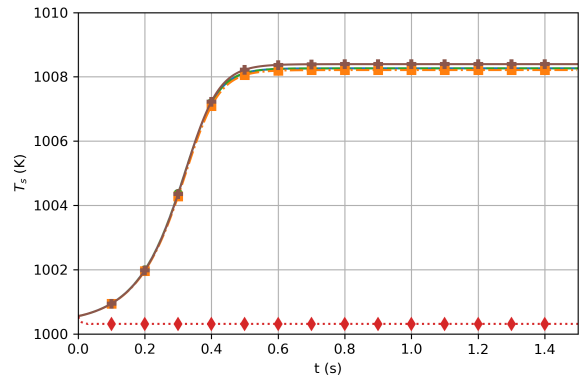
We first observe the envelope curves of  $T_s$  for various time step values in Figure 9. We see that all methods dampen out the oscillations when the time step is too big, except CKN which stabilises at a small oscillating amplitude. However, if we gradually decrease the time step, each method eventually produces a limit cycle. We see that ESDIRK-54A has the best behaviour in terms of reproducing the actual reference limit cycle. It generally seem that the higher the order of the method, the larger the time step can be while still resolving the limit cycle. An interesting behaviour is observed for the CKN method: though it is second-order accurate, it finds a relatively correct initial amplification with larger time steps than required by the fourth- and third-order methods. However it has the drawback that the solution diverges unless the time step is further reduced. A non-diverging solution with CKN is obtained with time steps for which all higher-order methods already provide better results. It seems that the lack of L-stability favours the initial destabilisation, but leads to the divergence of the solution. On the opposite, all L-stable methods dampen the oscillations when the time step is too big, and none of them diverges. Finally, we see that the first-order IE is not able to correctly reproduce the limit cycle, even with the smallest time step of  $10^{-6}$  s. Using such a time step is already prohibitive, therefore lowering it further cannot be considered a viable solution to achieve an accurate result. Convergence results presented hereafter are obtained on the range  $dt \in [10^{-5}, 10^{-1}]$  s, where IE never produces an initial amplification and established limit cycle, therefore its results are omitted for the sake of readability.

### 7.2.2. Initial growth

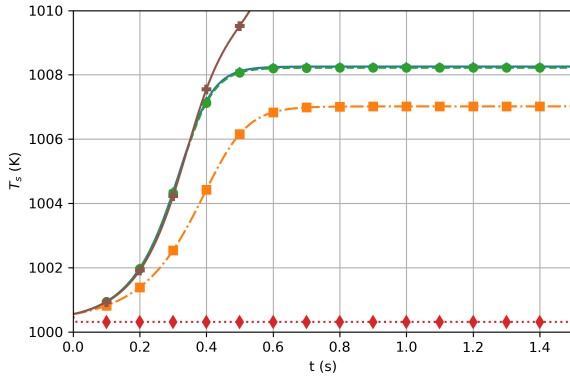
Figure 10a shows the evolution of the fitted exponential amplification factor  $b$  as the time step is lowered. We see that all methods converge to the same value, however ESDIRK-54A and CKN are the first methods that manage



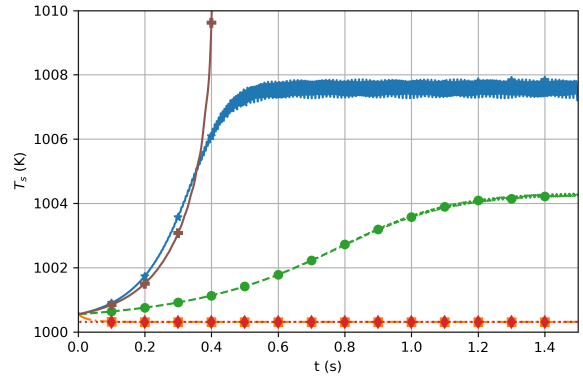
(a)  $dt = 1 \times 10^{-6}$  s



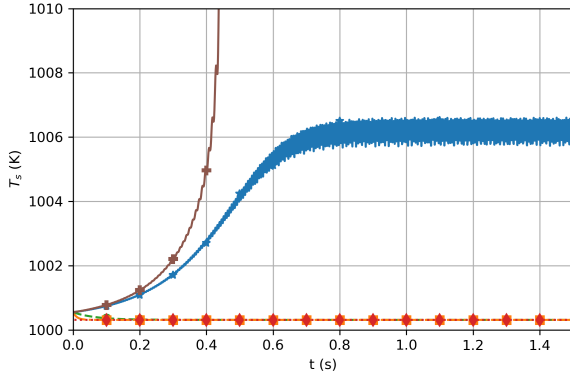
(b)  $dt = 3.79 \times 10^{-5}$  s



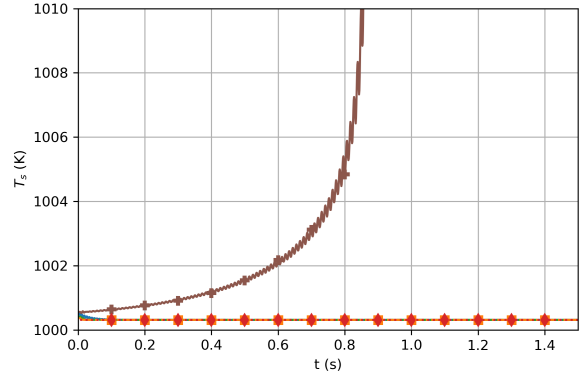
(c)  $dt = 1.13 \times 10^{-4}$  s



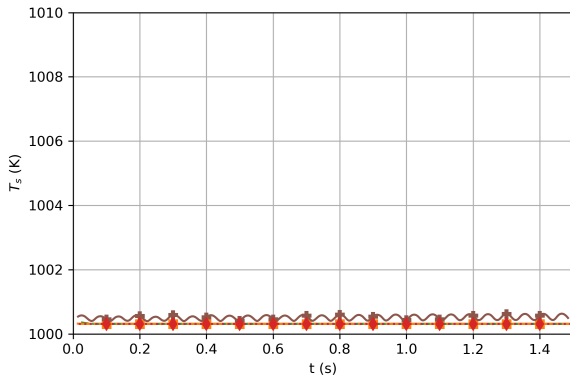
(d)  $dt = 3.36 \times 10^{-4}$  s



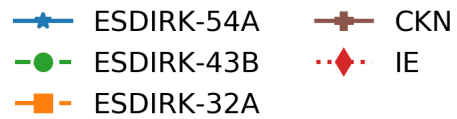
(e)  $dt = 4.83 \times 10^{-4}$  s



(f)  $dt = 1 \times 10^{-3}$  s



(g)  $dt = 5 \times 10^{-3}$  s



**Figure 9** Envelopes of the surface temperature histories computed for different time steps and integration methods

to capture a growth (crossing the line  $b = 0$ ), and also the quickest to converge to the correct value.

Figure 10b shows the relative error of the amplification factor with respect to the reference solution. We see once again the same ranking in terms of ability to find the correct factor. At any time step  $\Delta t \leq 5 \times 10^{-4}$  s, ESDIRK-54A yields the best accuracy. Moreover we can observe that each method has an asymptotic convergence region where the order of convergence is close to the order of the method. In particular CKN, which initially performs well for moderate time steps, is quickly overtaken by the other methods that possess a higher convergence rate.

For low time step values, the flattening of the convergence curves can be simply explained. The amplification factor is defined as a coefficient from an exponential fit, however this fit is only an approximation, as the nonlinear behaviour will let the unsteady evolution slightly deviate from the theoretical exponential initial growth. Also, the fit is based only on the successive maxima, not on the complete oscillating curve, which induces additional errors, e.g. imprecision in the abscissas of the maxima. Therefore there is a point at which the precision achieved with an exponential approximation cannot be improved further.

### 7.2.3. Limit cycle

The fundamental frequency of the established limit cycle is  $f \approx 452$  Hz. Figure 11a shows how the relative error on this frequency evolves with the time step, and Figure 11b shows the convergence of the amplitude of the fundamental frequency. The brown crossed curve for CKN is interrupted in the intermediate range of time step values, as the solution diverges, thus not allowing for an established limit cycle to be analysed. We can observe that ESDIRK-43B and ESDIRK-54A yield the most precise solutions at any given time step. In particular, ESDIRK-43B is able to capture a non-zero oscillation amplitude with larger time steps than required by the other methods. The frequency-finding process is not able to achieve unlimited accuracy in the determination of the fundamental frequency, hence the flattening of the convergence curves when the relative error reaches  $10^{-6}$ .

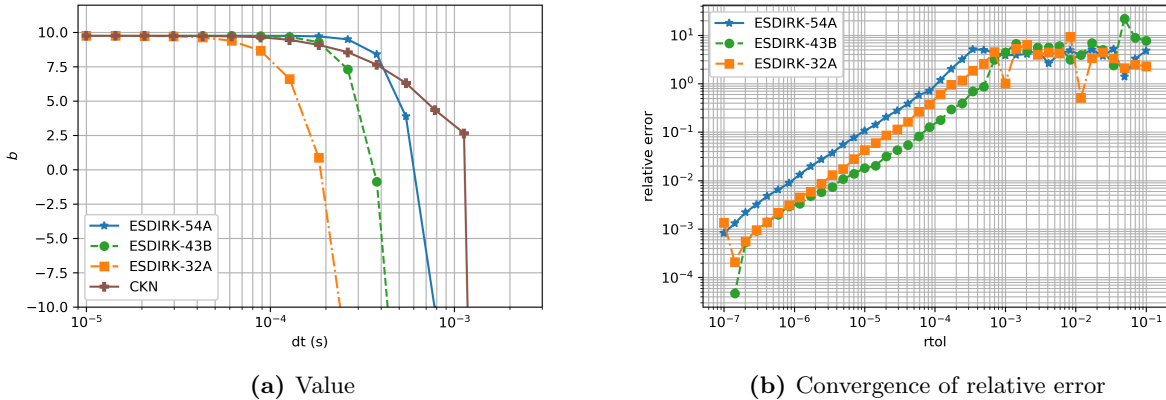
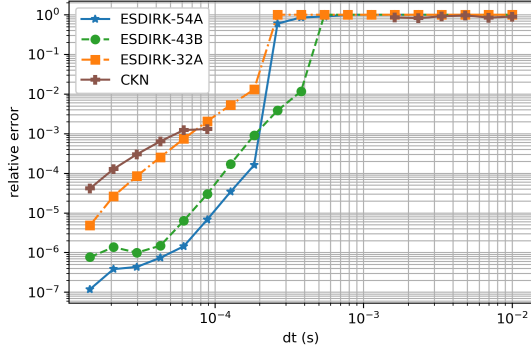


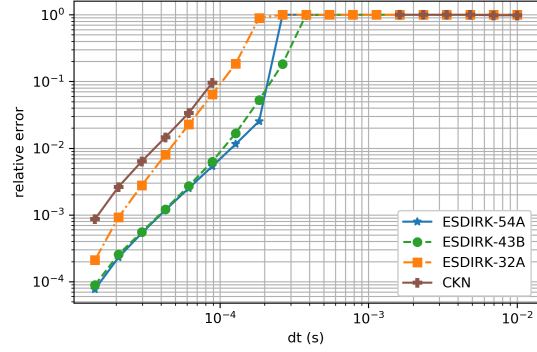
Figure 10 Fitted amplification factor

### 7.2.4. Computational cost

Based on the previous analysis, the high-order methods ESDIRK-54A and ESDIRK-43B seem particularly promising, as they require fewer time steps to achieve good results. However, due to the fact that these methods have more stages than the low-order methods, this does not mean that the actual computational times will be

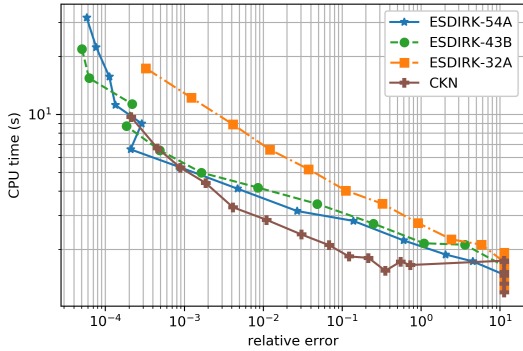


(a) Fundamental frequency

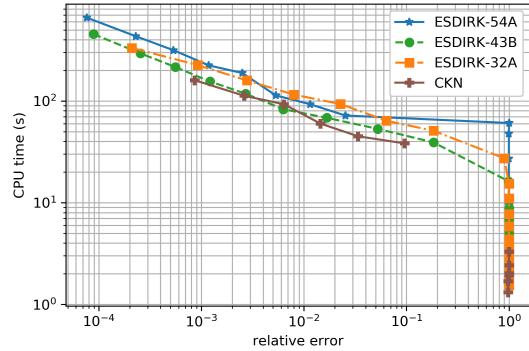


(b) Amplitude of the fundamental

Figure 11 Convergence of the limit cycle properties



(a) Amplification factor (transient-only)



(b) Amplitude of the fundamental of the established limit cycle

Figure 12 Work-precision diagrams for fixed time step simulations

advantageous. Figure 12a shows the computational times versus the achieved relative error on the amplification factor. Note that these simulations were run only on the physical time interval  $t \in [0, 0.1]$  s, so that computational times are truly representative. We see that CKN is the fastest method for a relative error higher than  $5 \times 10^{-3}$ , however this roughly corresponds to the zone where the solution diverges in finite time. ESDIRK-32A is not a very good performer, whereas ESDIRK-43B and ESDIRK-54A are performing well and have similar error levels.

Figure 12b shows the computational time required for a given level of relative error on the fundamental amplitude in the established limit cycle. Computational times are those of simulations run on the physical time interval  $t \in [0, 1.5]$  s. ESDIRK-43B is only marginally better than the other methods, no clear winner is to be picked.

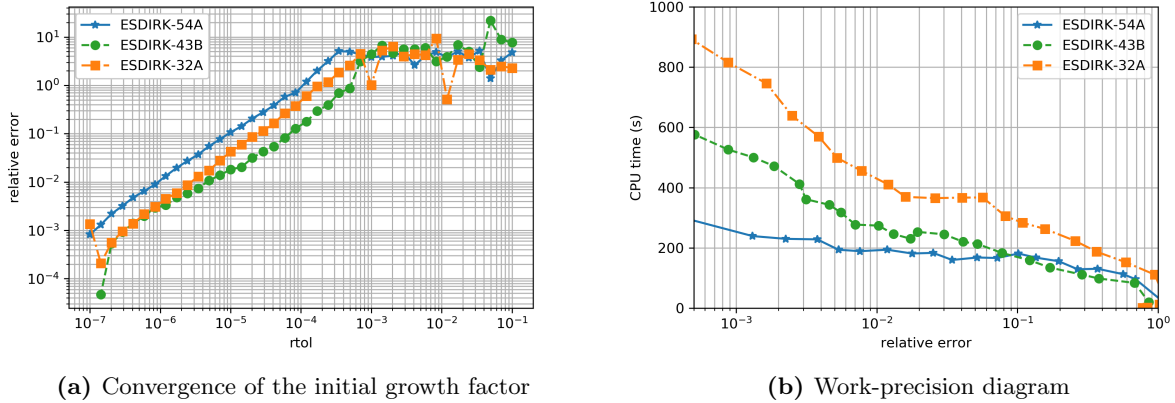
Overall, when using constant steps, the high-order methods ESDIRK-54A and ESDIRK-43B are almost identical and offer overall a very good performance. The Crank-Nicolson method is slightly misleading: its lack of stability when applied to DAEs leads to an easier destabilisation of the initial solution. However the method diverges quickly, unless the time step is very small, thus falsely leading to the conclusion of an unstable physical configuration.

### 7.3. Numerical experiment with time adaptation

The previous study with constant steps has shown that high-order methods are interesting for the simulation of a limit cycle. We now compare the embedded ESDIRK schemes with time adaptation enabled, to see if additional computational gains can be obtained. Following the methodology exposed in Section 4.4, the time step is controlled by the relative integration error tolerance  $rtol$ , which is varied between  $10^{-1}$  and  $10^{-7}$ . We first focus on the relative error achieved on the quantitative criteria used in the previous section. Finally, a comparison of the computational times and relative errors is presented, considering both fixed time step and adaptive simulations.

#### 7.3.1. Initial growth

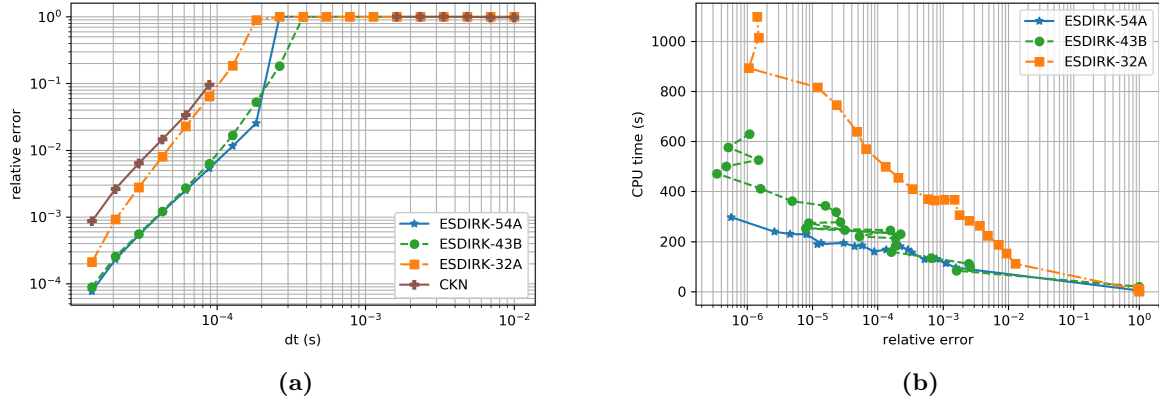
Figure 13a shows the convergence of the fitted amplification factor  $b$  when  $rtol$  is decreased. We see that with fine tolerances, all three methods resolve the transient quite well. However, each method has a different onset of the convergence, for example ESDIRK-43B starts to properly resolve the transient amplification with  $rtol \approx 3.5 \times 10^{-4}$ , whereas as ESDIRK-54A needs at least  $rtol \approx 8 \times 10^{-5}$ . In Figure 13b, we plot the accuracy achieved on the amplification factor (which is not equal to  $rtol$ ) versus the computational time. We clearly see that, when the tolerance is sufficiently low, the computational cost decreases as the order of the method increases. For example, if we require a relative error of  $10^{-3}$ , ESDIRK-54A is twice as fast as ESDIRK-43B, and three times as fast as ESDIRK-32A. Only for high levels of error ( $> 10^{-1}$ ) is ESDIRK-43B slightly more efficient than ESDIRK-54A.



**Figure 13** Convergence and computation cost for the amplification factor

#### 7.3.2. Limit cycle

We now compare the computational cost of each method when considering the resolution on the time interval  $t \in [0, 1.5]$  s. We have observed that the frequency of the fundamental and its amplitude converge equally well, therefore we only focus on the amplitude. Figure 14a shows how the relative error on the fundamental amplitude evolves with  $rtol$ . ESDIRK-43B has, for a given  $rtol$ , the lowest error, however there is an unexplained oscillation of the relative error. Figure 14b is the corresponding work-precision diagram. ESDIRK-32A is the worst performer by far, whereas ESDIRK-54A is the most efficient method.



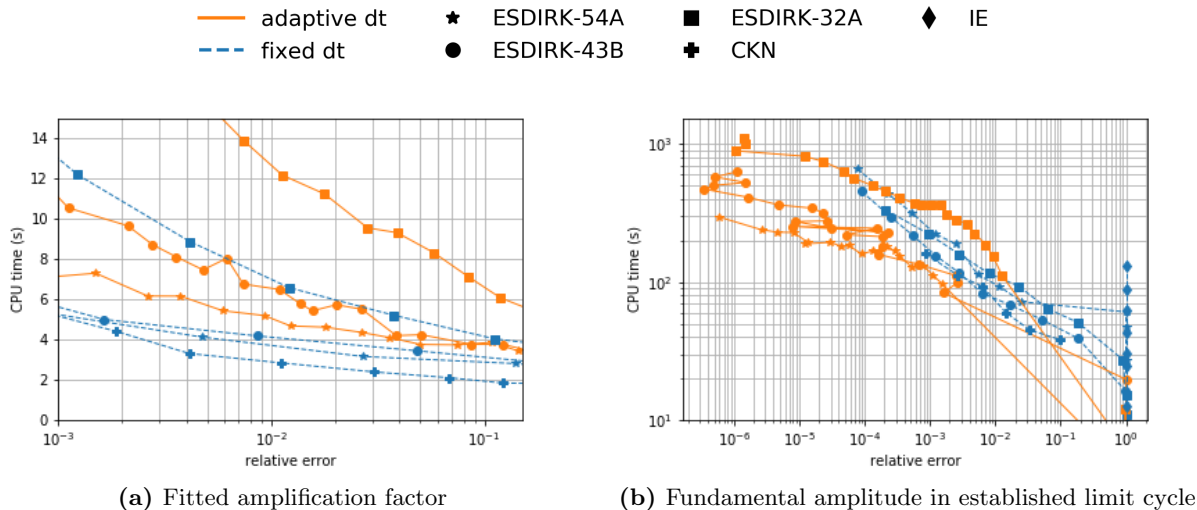
**Figure 14** Limit cycle with adaptive time stepping: (a) convergence of the fundamental amplitude, (b) work-precision diagram

### 7.3.3. Comparison with fixed time step results

We now wish to assess the performance gain achieved with time adaptation. To this end, we compare the computational times between fixed time step simulations and adaptive simulations for a given level of relative error. Figure 15a shows how the relative error on the amplification factor  $b$  during the initial growth evolves with computational time. The simulations were run for only 0.1 s of physical time, i.e. only for the initial growth. We observe that fixed time step simulations are always more efficient in this context. In particular, fixed time step implementations of the ESDIRK methods are faster than their adaptive counterparts. It has been assessed that this was due to an advantage in terms of number of Jacobian evaluations for the Newton method. Fixed time steps solutions only required up to 3 evaluations of this matrix, while adaptive solutions required up to 100 evaluations due to the repeated changes in time step.

The performance for the computation of the full limit cycle (initial growth and established cycle) is assessed in Figure 15b. The criteria is the relative error on the amplitude of the fundamental in the established limit cycle. One may think that the lightweight second-order Crank-Nicolson method could outperform the other methods for the established limit cycle, as this method is known to have good damping properties for oscillating systems, while only requiring one stage to be computed per time step. Indeed the method is the fastest among the fixed time step ones, and the fastest overall for relative error levels around  $10^{-2}$ . For lower error levels however, the adaptive high-order methods ESDIRK-54A and ESDIRK-43B are the most efficient methods. ESDIRK-32A in adaptive mode is generally slower than in fixed time step mode, unless very low errors are sought. Although not shown here, adaptive simulations require more Jacobian evaluations due to the successive changes in time step, but they require many fewer steps and Newton iterations overall, which, for the complete simulation, far outweighs the drawback of evaluating the Jacobian more often. This is supported by the observation that the slopes of CPU time versus relative error are smaller for adaptive methods compared to fixed time step implementations. Any additional cost, e.g. Jacobian evaluation, is overcome by the ability to take fewer steps. Still, adaptive methods are at a slight disadvantage in this test case, as the solution oscillates smoothly: the characteristic time scale of the system stays roughly constant throughout the simulation, therefore fixed time step simulations, with  $dt$  sufficiently low compared to this time scale, will be favoured by this consistency.

An interesting observation can be made: the adaptive methods always capture the correct limit cycle, unless  $rtol$  is too high, leading to a stabilisation of the solution. This is seen in Figure 15b, as all adaptive methods have a jump from important errors ( $\approx 1$ ) to much lower ones as  $rtol$  is lowered. On the opposite, fixed time step implementations do not have such a jump in error and are more likely to capture a non-accurate limit cycle (typically with an error higher than  $10^{-2}$ ) for an intermediate range of time step values. Overall, ESDIRK-54A and ESDIRK-43B seem to be the most reliable methods in this comparison.



**Figure 15** Comparison of the computational cost for a given level of relative error

Another practical consideration is that the time step corresponding to  $CFL = 1$  lies around  $10^{-6}$  s, which is approximately 100 times smaller than the time step necessary to obtain a very precise simulation with the fifth-order method ESDIRK-54A (see Figures 10b and 11b). The CFL-controlled time step is based on a stability analysis of the convection operator with an explicit time integration, which is not relevant for implicit integration and does not guarantee any level of error on the solution. As already discussed for the ignition transient in Section 6, use of a CFL-limitation would result in an important increase in computational time, without any valuable improvement on the solution accuracy. Finally, the constant time step simulations that yield accurate results more efficiently than with adaptive methods correspond to  $CFL \approx 100$ , which would usually not be expected to produce accurate unsteady results. One would rather safely choose a time step such that  $CFL = 1$ . This highlights one practical benefit of the time adaptation: precision is ensured based on a reliable mathematical criterion, and time step values can be used such that  $CFL \gg 1$ , while still ensuring a precise solution.

## 8. Application to unsteady combustion with detailed chemistry

We have now verified that the high-order adaptive methods perform well on relatively simple test problems. The next step is to test how they perform for the simulation of a propellant combustion with detailed chemistry. Additional stiffness is usually observed when a complex kinetic mechanism is used, thus it is useful to check the behaviour of the proposed numerical strategy in this context. The test case is the unsteady combustion of the AP monopropellant in a one-dimensional approach. The gas-phase kinetics is based on the AP-HTPB mechanism developed successively by Jeppson [56] and Tanner [57], initially for steady-state combustion. All reactions involving



carbonated species were removed to account for the absence of HTPB, resulting in a pure AP combustion mechanism involving 25 species and 80 reactions. Gas-phase molecular transport is treated in a simplified manner by using mixture-averaged approximations. Species diffusion fluxes are expressed by a Fickian formula with effective diffusion coefficients and are corrected to ensure they sum up to zero. Thermodynamic and transport properties are computed before-hand by CHEMKIN routines [58] and stored as lookup tables.

The solid phase and the surface are handled as in [45]: the solid is assumed inert, and all decomposition and gasification reactions occur at the surface. There are two global surface reactions: a direct dissociative sublimation, and a quasi-equilibrium decomposition. The regression speed is defined by a pyrolysis law taken from [45], and the proportions of gaseous products generated by the surface reactions are adjusted to obtain the experimentally measured regression rates at 20 atm, following the approach of Meynet [13]. These modelling choices allow for the use of detailed combustion kinetics while remaining within the simplified framework of solid and surface representation used in the present paper.

The computational mesh has 49 cells for the solid phase, and 126 cells for the gas phase, distributed in a non-uniform manner so that steady-state gradients are well resolved. Starting from a steady-state solution at  $P = 20.265 \times 10^5$  Pa, we study the transient occurring after a pressure step to  $P = 20 \times 10^5$  Pa.

The goal in this Section is not to demonstrate a physically realistic model, but rather to test the proposed numerical strategy on a case that is representative of complex solid-propellant simulations.

### 8.1. Order of convergence with fixed time steps

The orders of convergence of the methods from Table 1 have already been verified for the simple test case of Section 5.1 and are presented in Appendix C. We now want to verify that the orders are not affected by the additional complexity and stiffness induced by detailed kinetics. We simulate the unsteady evolution for  $t \in [0, 0.2]$  s and perform multiple integrations with various time steps.

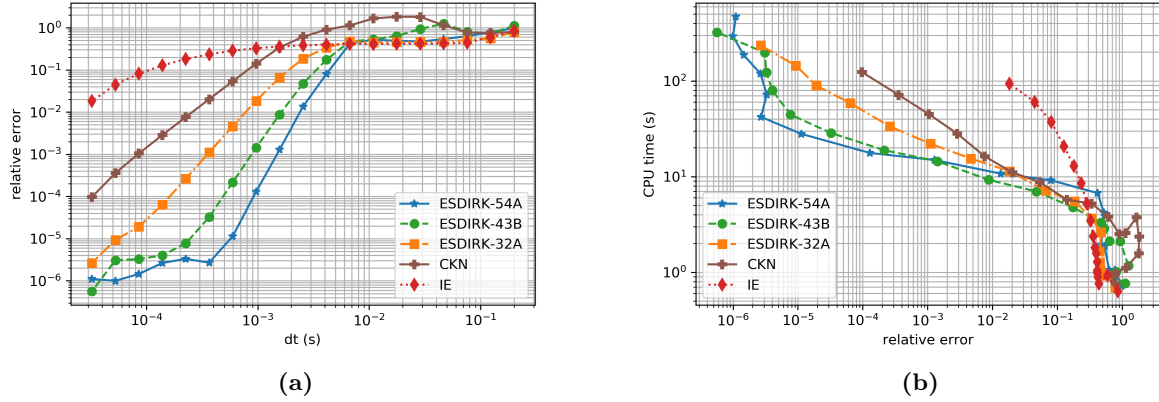
To quantify the accuracy of the overall time integration, we define the following error:  $\epsilon_{T_s} = \frac{1}{t_f} \int_0^{t_f} |T_s - T_{s,ref}| dt$ , with *ref* designating the reference simulation and  $t_f$  the final physical time. Cubic interpolation is used to compare both solutions on the same time grid.

Figure 16a shows the evolution of this error when the time step is varied. We see that each method attains its theoretical order of convergence. Similar results have been obtained for the other variables, both differential or algebraic, e.g. discrete mass flow rate field, cell temperatures. No order reduction is observed due to the stiffness of the chemical reactions with detailed kinetics. The ESDIRK methods perform well in this more complex scenario.

Figure 16b shows for each method the computational time required to achieve a given integration error  $\epsilon_{T_s}$ . By analysing the different simulations, it was determined that the curve of  $T_s$  is visually converged when  $\epsilon_{T_s} \leq 10^{-2}$ . ESDIRK-43B and ESDIRK-54A are the most efficient methods on this error range, and the speedup achieved by these high-order methods increases with the precision achieved. They are about 1.5 times faster than ESDIRK-32A and CKN for  $\epsilon_{T_s} = 10^{-2}$ , and approximately 5 times faster than CKN for  $\epsilon_{T_s} = 10^{-4}$ .

### 8.2. Computational performance with time step adaptation

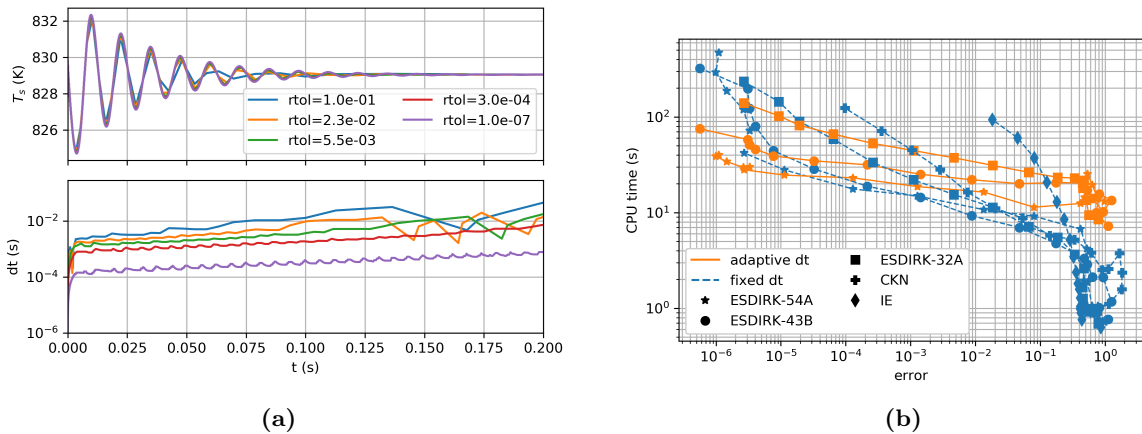
Now that we have verified that the convergence of the methods is not affected by the stiffness induced by complex kinetics, we use the ESDIRK methods with time step adaptation to see how they compare in terms of



**Figure 16** Accuracy of the integration with fixed time steps: (a) Convergence of  $\epsilon_{T_s}$ , (b) Work-precision diagram for  $\epsilon_{T_s}$

results. Different values of the relative integration tolerance  $rtol$  are used between  $10^{-1}$  and  $10^{-7}$ .

Figure 17a shows the complete transient for the surface temperature and the time step evolution for various values of  $rtol$ . We see that the change in time step is smooth, except for low tolerances when the time step becomes large, causing convergence issues. The temporal evolution of the surface temperature is well resolved even with relatively large values of  $rtol$ . Figure 17b shows the comparison of the computational time required to achieve a given level of error  $\epsilon_{T_s}$ , both with fixed time steps (blue lines) and adaptive time stepping (orange lines). Here, adaptive schemes do not seem to improve the performance globally. ESDIRK-54A is the best performing adaptive method, however it only becomes the fastest method overall for a very low level of error  $\epsilon_{T_s} \leq 10^{-5}$ . Its computational time is relatively close to the one of its fixed time step implementation. We observe that, for a given increase in accuracy, adaptive methods have a lower increase in computational time compared to their fixed time step counterparts.



**Figure 17** Integration with adaptive time stepping: (a) time step evolution for ESDIRK-43B, (b) work-precision diagram for the time integral error  $\epsilon_{T_s}$

This comparison is slightly unfair, because we had no a priori knowledge of the time step needed to properly resolve the transient. Without dynamic time adaptation, the time step would have typically been limited so that the CFL number is reasonably low, e.g. 1 to 10. The time step corresponding to  $CFL = 1$  is around  $7 \times 10^{-8}$  s, which is much lower than the time step required to achieve a good accuracy with most methods. A simulation

has been carried out only on the first 0.03 s of the transient, with CKN and a time step set such that  $CFL = 10$ . The computational time was 660 s. An equivalently well resolved transient can be obtained with ESDIRK-54A and  $rtol = 10^{-5}$  in only 11 s. As we can see in Figure 17a, even a less stringent tolerance would also be sufficient. From an engineering point of view, this represents a 60 times speedup, due to the fact that the time adaptation will automatically choose the relevant time step values. This adaptation ensures the problem is well resolved, while reaching CFL numbers that one would usually never trust to yield accurate unsteady results. Finally, time adaptation based on embedded methods automatically detects a slowdown of the dynamics as the solution stabilises and is able to increase the time step accordingly, whereas the CFL number stays roughly constant and cannot be an efficient time-step controlling criterion in that situation. This gain in engineering time is not quantifiable precisely, however it is definitely important.

## 9. Conclusion

This contribution presents the development of a high-fidelity one-dimensional model for the combustion of a solid propellant. The emphasis is on the numerical strategy choice to integrate in time the set of semi-discretised equations. It has been shown that the system is differential-algebraic in nature. Multiple test cases show that stiffly accurate singly-diagonally implicit Runge-Kutta methods are highly efficient for the time integration of such a system, in particular the embedded ESDIRK methods presented in [53]. Applications have been presented for ignition transients and limit cycle development with a simplified modelling, and appreciable computational gains have been observed. High-order methods can reliably capture dynamics which are practically impossible to reproduce with traditional low-order methods. It has also been verified that the proposed numerical strategy is robust and performs well when the modelling is much more complex, e.g. detailed kinetics in the gas phase.

Dynamic time step variation based on objective mathematical criteria ensures proper resolution of the unsteady phenomena. From an engineering point of view, the single parameter that controls the time step is the relative integration error tolerance  $rtol$ . In all our test cases, we have observed that  $rtol = 10^{-5}$  is sufficient to accurately resolve all unsteady phenomena. Using this value as standard tolerance liberates from the need of iterating over other practical criteria such as CFL limitation or maximum relative variation. The authors believe that a high-order adaptive method like ESDIRK-54A therefore allows for perceivable gains both in computational time, trustworthiness of the results, and the engineering time spent parametrising the time integration for a simulation.

Future work includes applying the presented framework to one-dimensional models involving a multiphase foam layer at the surface, and also the coupling of the one-dimensional code with a 2D or 3D CFD tool to more accurately describe surface phenomena in detailed heterogeneous solid propellant simulation codes, e.g. COMPAS from ONERA [20], or as an accurate dynamic boundary condition for combustion chamber simulations with the 3D multiphysics CFD tool CEDRE from ONERA [24], avoiding costly mesh refinement near the propellant surface.

## Acknowledgments

The present research was conducted thanks to a Ph.D grant co-funded by DGA, Ministry of Defence (E. Faucher, Technical Advisor), and ONERA.

## References

- [1] L. D. Luca, Theory of Nonsteady Burning and Combustion Stability of Solid Propellants by Flame Models, American Institute of Aeronautics and Astronautics, 1992, pp. 519–600.
- [2] B. Novozhilov, Theory of Nonsteady Burning and Combustion Stability of Solid Propellants by the Zeldovich-Novozhilov Method, American Institute of Aeronautics and Astronautics, 1992, pp. 601–641.
- [3] G. Lengelle, J. Duterque, J. Trubert, Physico-chemical mechanisms of solid propellant combustion, Solid Propellant Chemistry, Combustion, and Motor Interior Ballistics 185 (2000-01) 287–334.
- [4] F. Williams, M. Barrere, N. Huang, Fundamental aspects of solid propellant rockets, Advisory Group for Aerospace Research and Development of NATO, Technivision Services; London, 1969.
- [5] M. Brewster, Combustion mechanisms and simplified-kinetics modeling of homogeneous energetic solids, Energetic Materials: Part 2. Detonation, Combustion 13 (2003) 225–294.
- [6] M. Denison, E. Baum, A simplified model of unstable burning in solid propellants, ARS Journal 31 (1961) 1112–1122.
- [7] M. Beckstead, R. Derr, C. Price, A model of composite solid-propellant combustion based on multiple flames, AIAA Journal 8 (1970) 2200–2207.
- [8] M. Ward, S. Son, M. Brewster, Steady deflagration of HMX with simple kinetics: A gas phase chain reaction model, Combustion and Flame 114 (1998) 556–568.
- [9] F. Culick, A review of calculations for unsteady burning of a solid propellant., AIAA J. 6 (1968) 2241–2255.
- [10] F. Culick, Combustion instabilities in solid propellant rocket motors, Internal aerodynamics in solid rocket propulsion (2004). , RTO-EN-023 RTO/NATO.
- [11] E. Price, H. Bradley, G. Dehority, M. Ibricu, Theory of ignition of solid propellants, AIAA J. 4 (1966) 1153–1181.
- [12] V. Viliunov, V. Zarko, Ignition of solids, Studies in physical and theoretical chemistry, Elsevier, 1989.
- [13] V. Giovangigli, N. Meynet, M. Smooke, Application of continuation techniques to ammonium perchlorate plane flames, Combustion Theory and Modelling 10 (2006) 771–798.
- [14] D. Smith, Modeling Solid Propellant Ignition Events, Ph.D. thesis, Brigham Young University, 2011.
- [15] M. Beckstead, K. Puduppakkam, P. Thakre, V. Yang, Modeling of combustion and ignition of solid-propellant ingredients, Progress in Energy and Combustion Science 33 (2007) 497 – 551.
- [16] K. Meredith, M. Gross, M. Beckstead, Laser-induced ignition modeling of HMX, Comb. and Flame 162 (2014).
- [17] L. Massa, T. Jackson, J. Buckmaster, New kinetics for a model of heterogeneous propellant combustion, Journal of Propulsion and Power 21 (2005) 914–924.

- [18] T. Jackson, J. Buckmaster, Heterogeneous propellant combustion, *AIAA Journal* 40 (2002) 1122–1130.
- [19] S. Gallier, A. Ferrand, M. Plaud, Three-dimensional simulations of ignition of composite solid propellants, *Combustion and Flame* 173 (2016) 2–15.
- [20] D. Davidenko, Y. Fabignon, Some aspects of detailed modeling of solid rocket composite propellants, in: 6th European Conference for Aeronautics and Space Sciences (EUCASS), 2015.
- [21] M. Gross, K. Meredith, M. Beckstead, Fast cook-off modeling of HMX, *Comb. and Flame* 162 (2015) 3307 – 3315.
- [22] Q. Li, P. Liu, G. He, Fluid–solid coupled simulation of the ignition transient of solid rocket motor, *Acta Astronautica* 110 (2015) 180 – 190. Dynamics and Control of Space Systems.
- [23] W. A. Johnston, Solid rocket motor internal flow during ignition, *Journal of Propulsion and Power* 11 (1995) 489–496.
- [24] A. Refloch, B. Courbet, A. Murrone, P. Villedieu, C. Laurent, P. Gilbank, J. Troyes, L. Tessé, G. Chaineray, J.-B. Dargaud, E. Quémerais, F. Vuillot, Cedre software, *AerospaceLab Journal* (2011) 1–10.
- [25] A. Bizot, Ignition and unsteady combustion of AP-based composite propellants in subscale solid rocket motors, *International Journal of Energetic Materials and Chemical Propulsion* 4 (1997) 1046–1061.
- [26] W. Erikson, M. Beckstead, Modeling unsteady monopropellant combustion with full chemical kinetics, in: 36th AIAA Aerospace Sciences Meeting and Exhibit, 1998.
- [27] F. Harlow, A. Amsden, A numerical fluid dynamics calculation method for all flow speeds, *Journal of Computational Physics* 8 (1971) 197 – 213.
- [28] P. N. Brown, G. Byrne, A. Hindmarsh, VODE: A variable-coefficient ODE solver, *SIAM J. Sci. Stat. Comput.* 10 (1989) 1038–1051.
- [29] W. Erikson, M. Beckstead, Modeling pressure and heat flux responses of nitramine monopropellants with detailed chemistry, 1999.
- [30] J. Grcar, The Twopnt Program for Boundary Value Problems, Technical Report, Sandia National Labs., Livermore, CA, 1992.
- [31] K. E. Brenan, S. L. Campbell, L. R. Petzold, Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations, Society for Industrial and Applied Mathematics, 1995.
- [32] R. Kee, J. Grcar, M. Smooke, J. Miller, E. Meeks, Premix: a fortran program for modeling steady laminar one-dimensional premixed flames, Sandia Rep 143 (1985).
- [33] Y. Liau, V. Yang, Analysis of RDX monopropellant combustion with two-phase subsurface reactions, *Journal of Propulsion and Power* 11 (1995) 729–739.

- [34] S. Balay et al., PETSc Web page, 2019. URL: <https://www.mcs.anl.gov/petsc>.
- [35] T. Sayadi, V. Le Chenadec, P. J. Schmid, F. Richecoeur, M. Massot, Thermoacoustic instability – a dynamical system and time domain analysis, *Journal of Fluid Mechanics* 753 (2014) 448–471.
- [36] M. Smooke, V. Giovangigli, Numerical modeling of axisymmetric laminar diffusion flames, *IMPACT of Computing in Science and Engineering* 4 (1992) 46–79.
- [37] N. Darabiha, Transient behaviour of laminar counterflow hydrogen-air diffusion flames with complex chemistry, *Comb. Sci. and Tech* 86 (1992) 163–181.
- [38] M. Massot, M. Kumar, A. Gomez, M. D. Smooke, Counterflow spray diffusion flames of heptane: computations and experiments, in: *Proceedings of the 27th Symp. on Comb.*, The Comb. Institute, 1998, pp. 1975–1983.
- [39] S. Descombes, M. Duarte, T. Dumont, F. Laurent, V. Louvet, M. Massot, Analysis of operator splitting in the nonasymptotic regime for nonlinear reaction-diffusion equations. Application to the dynamics of premixed flames, *SIAM J. Numer. Anal.* 52 (2014) 1311–1334.
- [40] H. Im, L. L. Raja, R. J. Kee, L. R. Petzold, A numerical study of transient ignition in a counterflow nonpremixed methane-air flame using adaptive time integration, *Combustion Science and Technology* 158 (2000) 341–363.
- [41] M.-A. N’Guessan, M. Massot, L. Séries, C. Tenaud, High order time integration and mesh adaptation with error control for incompressible navier–stokes and scalar transport resolution on dual grids, *Journal of Computational and Applied Mathematics* (2019) 112542.
- [42] E. Hairer, G. Wanner, *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems*, volume 14 of *Springer Series in Comput. Math.*, 2nd ed., Springer-Verlag Berlin Heidelberg, 1996.
- [43] E. Washburn, T. Parr, D. Hanson-Parr, Micro videographic analysis of the melt layer of self-deflagrating HMX and RDX, *Propellants, Explosives, Pyrotechnics* 35 (2010) 46–52.
- [44] J. Davidson, M. Beckstead, Improvements to steady-state combustion modeling of cyclotrimethylenetrinitramine, *Journal of Propulsion and Power* 13 (1997) 375–383.
- [45] S. Rahman, Modélisation et simulation numérique de flammes planes instationnaires de perchlorate d’ammonium, Ph.D. thesis, Université Pierre et Marie Curie, 2012.
- [46] N. Meynet, Simulation numérique de la combustion d’un propergol solide, Ph.D. thesis, Université Pierre et Marie Curie, 2005.
- [47] L. François, J. Dupays, D. Davidenko, M. Massot, Travelling wave mathematical analysis and efficient numerical resolution for a one-dimensional model of solid propellant combustion, *Comb. Theory and Modelling* (2020) 1–35. , In Press.
- [48] D. Anderson, J. Tannehill, R. Pletcher, *Computational fluid mechanics and heat transfer*, Third edition, Boca Raton, CRC Press, 2013.

- [49] J. Crank, P. Nicolson, A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type, *Mathematical Proc. of the Cambridge Philosophical Society* 43 (1947) 50–67.
- [50] R. Alexander, Diagonally implicit Runge-Kutta methods for stiff O.D.E.s, *SIAM Journal on Numerical Analysis* 14 (1977) 1006–1021.
- [51] Y.-C. Liao, E. Kim, V. Yang, A comprehensive analysis of laser-induced ignition of RDX monopropellant, *Combustion and Flame* 126 (2001) 1680 – 1698.
- [52] E. Hairer, S. Nørsett, G. Wanner, *Solving Ordinary Differential Equations I Nonstiff problems*, second ed., Springer, Berlin, 2000.
- [53] A. Kværnø, Singly diagonally implicit Runge–Kutta methods with an explicit first stage, *BIT* 44 (2004) 489–502.
- [54] U. M. Ascher, S. J. Ruuth, B. T. R. Wetton, Implicit-explicit methods for time-dependent partial differential equations, *SIAM Journal on Numerical Analysis* 32 (1995) 797–823.
- [55] P. Clavin, D. Lazimi, Theoretical analysis of oscillatory burning of homogeneous solid propellant including non-steady gas phase effects, *Combustion Science and Technology* 83 (1992) 1–32.
- [56] M. Jeppson, M. Beckstead, Q. Jing, A kinetic model for the premixed combustion of a fine AP/HTPB composite propellant, volume 44, 1998.
- [57] M. Tanner, *Multidimensional Modeling of Solid Propellant Burning Rates and Aluminum Agglomeration and One-Dimensional Modeling of RDX/GAP and AP/HTPB*, Ph.D. thesis, Brigham Young University, 2008.
- [58] R. Kee, F. Rupley, J. Miller, *Chemkin-II: A fortran chemical kinetics package for the analysis of gas-phase chemical kinetics* (1989).
- [59] P. Virtanen et al., SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, *Nature Methods* 17 (2020) 261–272.
- [60] O. Knio, H. Najm, P. Wyckoff, A semi-implicit numerical scheme for reacting flow. II. Stiff, operator-split formulation, *J. Comput. Phys.* 154 (1999) 482–467.
- [61] H. Najm, O. Knio, Modeling Low Mach number reacting flow with detailed chemistry and transport, *J. Scientific Computing* 25 (2005) 263–287.
- [62] E. Motheau, J. Abraham, A high-order numerical algorithm for DNS of low-Mach-number reactive flows with detailed chemistry and quasi-spectral accuracy, *Journal of Computational Physics* 313 (2016) 430 – 454.

## Appendix A. Generating configurations with various degrees of instability

In order to highlight the benefit of the high-order adaptive time integration, we search for configurations which are linearly unstable around the corresponding steady-state solution. We generate such configurations with the

stable simplified model from Section 5.1.1 as baseline, by varying its parameters. We use existing theoretical tools to approximately evaluate the stability of the steady-state solution.

#### *Appendix A.1. Theoretical indicator of intrinsic instability*

The Zeldovich-Novozhilov (ZN) framework is a useful tool to study the stability of a steady-state solid propellant combustion. This theory was originally developed to only use sensitivity parameters which can be experimentally determined from a batch of steady-state experiments, without any modelling of the gas-phase phenomena. Considering a surface pyrolysis and an inert material, the solid phase is modelled with equation (1) and the boundary conditions  $T(-\infty) = T_0$  and  $T(0) = T_s$ . The pyrolysis mass flow rate  $m$  and the surface temperature  $T_s$  are linked in steady-state via laws of the form:

$$m = m(T_0, P), \quad T_s = T_s(T_0, P) \quad (\text{A.1})$$

Such relations can be obtained via theoretical modelling or numerical investigations for solid propellant models, or via experiments for real propellants.

The ZN approach consists in assuming that the steady-state relations (A.1) remain valid under unsteady conditions. However,  $T_0$  is generally a constant and, if  $P$  is also constant, these relations would result in a constant surface temperature and regression speed. To circumvent this issue, let us consider the steady-state profile temperature profile:  $T(x) = T_0 + (T_s - T_0) \exp(xmc_c/\lambda_c)$ . The temperature gradient just below the surface is  $\phi = \partial_x T(0^-) = mc_c(T_s - T_0)/\lambda_c$ . We can then replace  $T_0$  in equation (A.1) by  $T_0 = T_s - \phi\lambda_c/mc_c$ . In the unsteady regime, this value is different than the actual value used to compute the steady-state profile. It is usually called the ‘‘apparent’’ or ‘‘effective’’ initial temperature, e.g. the initial temperature that would correspond to an hypothetical steady-state for the given values of  $T_s$  and  $\phi$ . We can now formulate the previous steady-state laws (A.1) as:

$$m = m(\phi, P), \quad T_s = T_s(\phi, P) \quad (\text{A.2})$$

These laws are supposed to be valid in the unsteady regime. This is usually accepted, as long as the apparent initial temperature remains within acceptable bounds. It is also required that data for this initial temperature be available, or at least reasonably extrapolated.

The next core step of the ZN approach is to linearise the heat equation in the solid and the laws (A.2) around the steady-state solution. A mathematical study of the amplification of small compact perturbation leads to the definition of a stability criteria, which depends on two steady-state sensitivity coefficients:  $r = (\partial_{T_0} \bar{T}_s)_P$ ,  $k = (\bar{T}_s - T_0) (\partial_{T_0} \ln(\bar{m}))_P$ , with  $\bar{\cdot}$  denoting steady-state values. Steady-state combustion is always stable if  $k < 1$ . If  $k > 1$ , the steady-state is stable only if  $r > (k - 1)^2/1 + k$ . The line  $r = (k - 1)^2/1 + k$  is the locus of a Hopf bifurcation, where the steady-state solution becomes linearly unstable in a oscillating manner, with the possibility of stabilising on a limit cycle. If  $r > (\sqrt{k} - 1)^2$ , the instability grows purely exponentially. The associated stability diagram is shown in Figure A.18a: the leftmost parabola is the first stability limit, the second one is the onset of purely exponential instability. It has been shown that unsteady gas-phase phenomena tend to widen the stability area, however this first simplified analysis remains a good indicator of the stability bounds. We refer the reader



to [2] for extensive details on the ZN analysis and its extensions. This stability is called “intrinsic” because it is a property of the solid propellant as an isolated system, as opposed to other types of instabilities, for instance those that might appear when coupling the solid propellant combustion with a chamber [10], whose pressure varies in time based on the propellant regression rate.

### Appendix A.2. Optimisation problem

We use these ZN stability criteria to generate configurations whose coefficients  $(r, k)$  are at various positions in the stability diagram. To do so, we setup an optimisation problem. Let us denote as  $X$  the vector containing the parameters of the simple combustion model that we have chosen as free variables. For a given value of  $X$ , we can find the corresponding value of  $(r, k)$  by performing three steady-state simulations with the semi-analytical tool: one baseline simulation, one simulation with a perturbed initial temperature  $T_0$ , and one simulation with a perturbed pressure  $P$ . Then, by means of finite differences,  $r$  and  $k$  may be evaluated. This process can be summarised as the function  $f_{rk} : X \rightarrow (r, k)$ . The optimisation problem is then formulated as:

$$\min_x f_{obj}(x) \tag{A.3a}$$

$$\text{subject to } g(X) \leq 0 \tag{A.3b}$$

$$h(X) = 0 \tag{A.3c}$$

where the objective function  $f$  is defined as  $f_{obj} : X \rightarrow \|f_{rk}(X) - (r, k)_{target}\|_2^2$  with  $(\cdot, \cdot)$  denoting a vector formulation. The inequality constraints are gathered in the vector function  $g$ , and equality constraints are gathered in  $h$ . This problem is simply the constrained minimisation of the distance to the target  $(r, k)$  coefficients. Inequality constraints  $g$  are used to ensure the different physical parameters remain within realistic bounds. They can be supplemented with equality constraints  $h$  to enforce certain properties of the steady-state solution, e.g. surface temperature, regression speed...

Regarding practical implementation, the Sequential Quadratic Programming algorithm SLSQP of the Python library Scipy [59] is used. The Jacobians of the objective function and constraints are obtained via finite differences. The Hessian of  $f_{obj}$  is built iteratively through a BFGS update. Note that we use the semi-analytical tool to perform the simulation, as it is fast, very precise, does not need any specific mesh generation, and is ensured to converge. For more complex combustion models, the simulations can be carried out with the one-dimensional CFD code, however it is important that the steady-state solutions are converged with sufficient accuracy, so as not to introduce important errors in the Jacobian estimation.

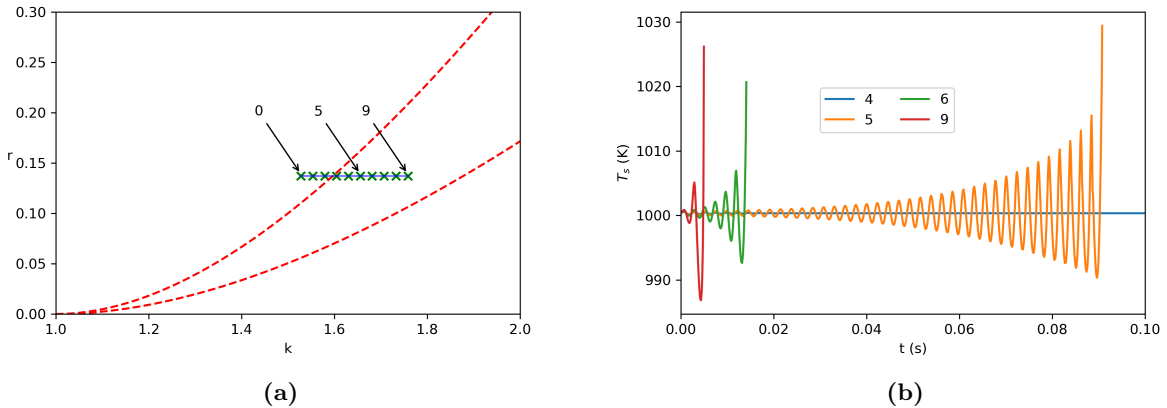
### Appendix A.3. Numerical assessment of intrinsic stability

We use the previous optimisation problem to generate configurations which have their sensitivity coefficients  $r$  and  $k$  distributed regularly on a segment defined as  $r = 0.137$  (baseline value) and  $k \in [1.5, 1.75]$ , thus crossing the ZN stability limit. The optimisation is constrained as to preserve physically sound characteristics (surface temperature at 1000 K, regression speed of 1 cm/s at 50 atm, 3540 K final flame temperature). The segment and the corresponding targeted points are displayed in Figure A.18a.

For each point, we numerically assess the stability of the corresponding steady-state combustion. Based on the stationary temperature profile provided by the semi-analytical tool, we generate a non-uniform mesh, such that the increase in temperature between each cell center is a chosen constant  $\Delta T$  (e.g. 10 K). Cells are then added to push the “infinity” boundaries further away, so as to minimise the impact of the Neumann boundary conditions. We typically use  $\Delta T = 20\text{K}$ , as we found that the steady-state surface temperature would then match up to less than 1% between the semi-analytical tool and the one-dimensional CFD code. After the mesh has been generated, a slight constant pressure perturbation (typically 0.1% of the prescribed pressure) is applied and the one-dimensional tool is run with implicit Euler and large time steps so as to converge to a perturbed initial steady-state. The pressure is then set back to its original value and an unsteady simulation is performed with ESDIRK-54A and a relative tolerance of  $10^{-6}$  on the integration error. The stability of the combustion can then be assessed numerically by analysing whether the perturbation is damped out or not.

The unsteady simulations for a few points are shown in Figure A.18b. We see that, up to the 4-th point, the system is stable. Starting from the 5-th point, the system diverges. Overall we observe that the numerical stability limit is slightly further to the right than predicted by the ZN method with quasi-steady gas phase, as already discussed. Refining the search between points 4 and 5 allows us to find a configuration that can exhibit a limit cycle. The corresponding model parameters are the same as in Section 5.1.1, except for the following changes:  $T_{ap} = 14668\text{ K}$ ,  $c_p = 692.8\text{ J/kg/K}$ ,  $c_c = 1253\text{ J/kg/K}$ ,  $T_0 = 182.4\text{K}$ ,  $\lambda_c = 0.65\text{ W/m/K}$ ,  $\lambda = 0.362\text{ W/m/K}$ ,  $\mathcal{M} = 57.9\text{ g/mol}$ ,  $\Delta h_f^0(G_1) = -2.28 \times 10^5\text{ J/kg}$ ,  $\Delta h_f^0(G_2) = -2.22 \times 10^6\text{ J/kg}$ ,  $A = 340.4$ .

The corresponding steady-state temperature profile is shown in Figure 8a, as computed by the semi-analytical tool. If we slightly perturb it and compute the unsteady evolution of the system, we obtain the surface temperature evolution plotted in Figure 8b. The linear instability causes the system to diverge near  $t = 0$ , however nonlinear effects allow for stabilisation on a limit cycle after  $t \approx 0.5\text{ s}$ , where the oscillation amplitude remains constant.



**Figure A.18** Generation of unstable configurations: (a) Segment travelled in the  $(r, k)$  stability diagram, (b) Unsteady simulations of some configurations (ESDIRK-54A with  $rtol = 10^{-6}$ )

## Appendix B. Further clarification regarding the continuity equation

Here, we show how the Runge-Kutta formulation can be used in order to provide an original treatment of the mass flow rate constraint in the gas phase. Let us recall that in our one-dimensional low-Mach approach, the density

$\rho$  cannot be considered a true variable of our problem, as it is uniquely determined from the temperature, mass fractions and pressure. Consequently, the continuity equation (16) should not be considered as an ODE on  $\rho$ , but rather as a constraint on the mass flow rate field coming from its PDE counterpart:  $\partial_x m = -\partial_t \rho$ . This source term is itself a function of  $m$  and can be obtained by differentiating the logarithm of the equation of state (11) with respect to time:

$$\partial_x m = -\partial_t \rho = -\rho \left( \partial_t \log P - \partial_t \log T - \partial_t \log \left( \sum_{\mathfrak{t}=1}^{n_e} \frac{Y_{\mathfrak{t}}}{\mathcal{M}_{\mathfrak{t}}} \right) \right)$$

as classically done in the combustion community (see for example [60, 61, 39, 62]). In the right hand-side,  $\partial_t P$  is an input (e.g. constant pressure, or evolution based on a combustion chamber model). Using the continuous equations (4) and (5), the terms involving  $\partial_t T$  and  $\partial_t Y_{\mathfrak{t}}$  can be gathered as a single function  $f$  of  $T$ ,  $Y$ ,  $P$  and  $m$ :

$$\partial_x m = -\rho(P, T, Y_1, \dots, Y_{n_e}) \left( \frac{\partial_t P}{P} + f(P, T, Y_1, \dots, Y_{n_e}, m) \right)$$

We could semi-discretise this equation in space to obtain a discrete constraint on the mass flow rate [62]. In our one-dimensional approach, it would replace the continuity equation (3). However this would require evaluating  $f$ , which, albeit entirely possible, would bring additional complexity to the code.

An alternative and original approach of this paper is to apply the Runge-Kutta scheme to the semi-discrete continuity equation (16) directly. Let  $\rho_n^i$  be the density in the  $i$ -th cell at time step  $n$ , and  $\rho_{n,i}^i$  the same density at the  $i$ -th stage of time step  $n$ . For the  $i$ -th stage of any DIRK method, we obtain:

$$\rho_{n,i}^i = \rho_n^i + \Delta t \sum_{j=1}^s a_{ij} (d_t \rho^i)_{n,j} \quad (\text{B.1})$$

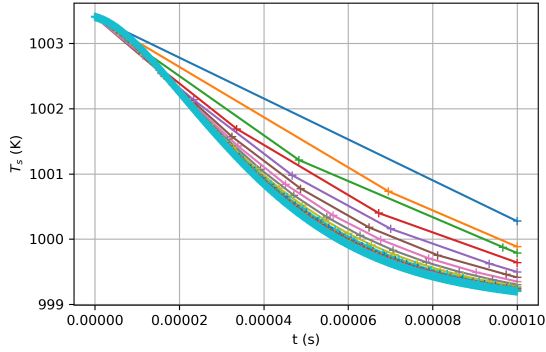
The term  $(d_t \rho^i)_{n,j}$  is the time derivative of  $\rho^i$  at time  $t = t_n + c_j \Delta t$  (i.e. at the  $j$ -th stage). The semi-discrete mass conservation equation (16) allows to reinterpret it as the numerical approximation of the mass flow rate spatial gradient at this stage. This equation is a constraint on  $m$ , as can be seen if we rearrange the terms:

$$-\frac{m_{n,i}^i - m_{n,i}^{i-1}}{x^i - x^{i-1}} = \frac{\rho_{n,i}^i - \rho_n^i}{a_{ii} \Delta t} - \sum_{j=1, j \neq i}^s \frac{a_{ij}}{a_{ii}} (d_t \rho^i)_{n,j} \quad (\text{B.2})$$

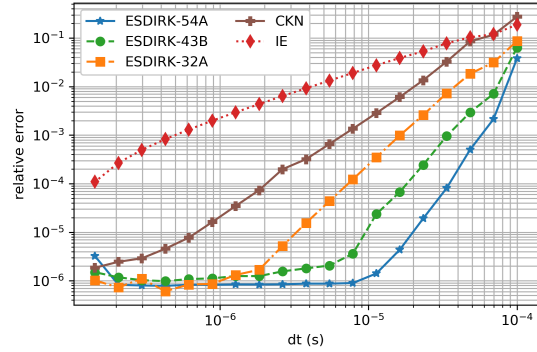
The right hand-side is the approximation of the source term  $(d_t \rho^i)_{n,i}$  as given by the Runge-Kutta method, which can be entirely expressed in terms of the mass flow rates at various stages. As a consequence, we only express the mass constraint in terms of  $\rho_{n,i}^i$ , which are functions of the other variables as explained above, as well as in terms of the mass flow rates at various stages. This approximation is constructed automatically by the Runge-Kutta scheme and relies on the fact that the time derivative of the density is involved as a source term. We have seen that it enables a high-order temporal resolution for the mass flow rate and other variables, without having to insert in the mass conservation equation the time evolution of the various variables as classically done.

## Appendix C. Verification of the order of convergence

To verify that a high order of convergence in time can indeed be attained on all variables, a simple test case is set up, using the simplified model presented in Section 5. A steady-state is computed at pressure  $P = 5.5 \times 10^6$  Pa and is given as initial solution for an unsteady simulation with a lower pressure at  $P = 5 \times 10^6$  Pa. The simulation is then run for a physical time of  $10^{-4}$  s with fixed time steps<sup>2</sup>. The curves of  $T_s$  obtained for various time step values are plotted in Figure C.19. The cyan curve represent the most refined solution.



**Figure C.19** Surface temperature histories obtained with IE when gradually lowering the time step



**Figure C.20** Global error  $\epsilon_{\rho u}$  on the mass flow rate field

The convergence of the algebraic variables is analysed for the surface temperature  $T_s$  as the difference between the value obtained at the final physical time for a given time step and the one for the smallest time step. A relative error is obtained by normalising by the latter. Additionally, an average error is considered for the mass flow rate field at final time ( $N$  global number of cells and  $ref$  for reference simulation):  $\epsilon_{\rho u} = \sqrt{\sum_0^N \frac{1}{N} (m_i(t_f; dt) - m_i(t_f; dt_{ref}))^2}$ . The corresponding relative errors are plotted in Figure C.20. The theoretical order of convergence is attained as long as we are not limited by the precision of the Newton algorithm and, although not reported here, similar convergence rates were observed for the differential variables.

<sup>2</sup>We start the simulation by performing a single step with IE and  $dt = 10^{-12}$  s, allowing all constraints to be satisfied.