The 9th European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Congress 2024
3–7 June 2024, Lisboa, Portugal

LOCAL TIMESTEP SUPER-TIME-STEPPING INTEGRATION METHODS APPLIED TO HEAT AND MASS TRANSFER IN ANISOTROPIC POROUS MEDIA

DELLINGER N.[1], DUFOUR G.[2], LAMBOLEY X.[1], REBOUL L.[2] AND ROGIER F. [2]

[1] ONERA/DMPE, Université de Toulouse
e-mail: Nicolas.Dellinger@onera.fr, Xavier.Lamboley@onera.fr

[2] ONERA/DTIS, Université de Toulouse
email: Guillaume.Dufour@onera.fr, Louis.Reboul@onera.fr, Francois.Rogier@onera.fr

Key words: Diffusion, Temporal schemes, STS methods, Multi-step methods, Zonal timestep

Summary. In fire safety problems, the simulation of the thermal degradation of anisotropic porous materials is complex due to the large amplitudes of time and space scales. Degraded areas where temperature increases and gas transport occurs might be very localised, sadly reducing the overall stability of numerical solvers and increasing the computation time. This can be mitigated thanks to Super-Time-Stepping methods, which are based on the use of a multi-step explicit scheme. In this study, these methods are first presented and their advantage to handle diffusion-advection problems such as the thermal degradation of anisotropic porous materials are emphasized on a simple use case. Then, a possible acceleration of the computation by using local timestep Super-Time-Stepping methods, particularly adapted to the heterogeneous problems previously mentioned, is discussed.

1 INTRODUCTION

Super-Time-Stepping (STS) methods are a family of numerical integration methods based on the use of a multi-step explicit scheme, in order to lower the stability constraints on the Fourier and CFL numbers to integrate parabolic partial differential equations. These methods therefore enable a computation time reduction compared to more limited classical explicit methods, while guaranteeing a second-order approximation [1, 4, 6]. In this study, the general formulation of Super-Time-Stepping schemes is first presented and applied to the simulation of diffusion-advection problems in anisotropic reactive porous materials encountered in the field of fire safety. An acceleration factor up to 7.5 is obtained against state-of-the-art second-order Runge-Kutta schemes. In the second part of the study, focus is made on the development of a Local Super-Time-Stepping method improving the original integration schemes. The main idea is to locally reduce the timestep by adjusting the number of steps of the integration scheme depending on local features of the physical problem to be solved, minimising the number of evaluations of the right-hand-side of the system of equations. This improvement is of particular interest for diffusion-advection problems in anisotropic reactive porous materials, where high pressure gradients and gas velocities are generally concentrated in areas exposed to heat sources. The zonal timestep adjustment greatly improves the computation time in this configuration, where

the timestep is increased in most of the computation domain while keeping a second-order scheme. The third part of this study aims at validating the implemented schemes and evaluating the possible acceleration to be expected.

2 STS METHODS IN THE CONTEXT OF HEAT AND MASS TRANSFER APPLICATIONS

2.1 Time integration as a polynomial formalism

We are interested in solving equations of the form (1)

$$\left\{ \begin{array}{rcll} \partial_t u - Au & = & 0, & \text{in } R^d \\ u(0,.) & = & u_0 & \end{array} \right. \tag{1}$$

where $u(t.) \in H$, H being an Hilbert space and A is a self-adjoint negative-definite linear operator on H (typically a Laplace operator) whose discrete approximation A_h, in the form of a hermitian negative-definite matrix is given, h being the mesh size characteristic parameter. The problem then turns into solving the semi-discretized equation

$$\left\{ \begin{array}{rcl} \partial_t u - A_h u & = & 0, \\ u(0,.) & = & u_0^h \end{array} \right. \tag{2}$$

Considering an arbitrary timestep $\Delta t = t^{n+1} - t^n$, with obvious notations, the exact solution from (2) reads:

$$u\left(t^{n+1}\right) = \exp\left(\Delta t A_h\right) u\left(t^n\right) = \sum_{k=0}^{+\infty} \frac{\left(\Delta t A_h\right)^k}{k!} u\left(t^n\right), \tag{3}$$

so that any explicit numerical scheme of the form $u_{n+1} = P\left(\Delta t A_h\right) u_n$ with $P \in \mathbb{R}_d\left[X\right]$, $p \leq d$, is automatically of consistency order p provided that $\forall k \in \{0, \cdots, p\}$, $P^{(k)}\left(0\right) = 1$. Many classical explicit methods for ODEs can be put under this polynomial formalism:

$$\text{Euler} \qquad P\left(\Delta t A_h\right) = 1 + \Delta t A_h, \tag{4a}$$

$$\text{RK2} \qquad P\left(\Delta t A_h\right) = 1 + \Delta t A_h \left(1 + \frac{\Delta t}{2} A_h\right), \tag{4b}$$

$$\text{Heun} \qquad P\left(\Delta t A_h\right) = \frac{1}{2}\left(\left(1 + \Delta t A_h\right) + \left(1 + \Delta t A_h\right)^2\right), \tag{4c}$$

$$\text{Taylor Series of order } p \qquad P\left(\Delta t A_h\right) = \sum_{k=0}^{p} \frac{\left(\Delta t A_h\right)^k}{k!}. \tag{4d}$$

One can note that Euler is consistent at first order while Heun and RK2 are at second order. Regarding stability, per our assumptions on A_h, these polynomial schemes are stable in l^2-norm for all $\Delta t > 0$ such that $\left|P\left(-\Delta t \lambda\right)\right| \leq 1$ with $\lambda \in \left[0, \rho\left(A_h\right)\right]$ where $\rho\left(A_h\right)$ the spectral radius of A_h. It is for example known that the stability domain of the Euler scheme is limited to $\Delta t < \frac{2}{\rho(A_h)}$, which for a standard 1D finite difference discretization A_h of A implies $\Delta t = \mathcal{O}\left(h^2\right)$. The aim of Super-Time-Stepping schemes [1, 5, 6] is to replace the polynomial expansion of the form (4d), by another one of higher order $d > p$, still ensuring the consistency at order p, while using additional degrees of freedom when to expand the domain of stability.

2.2 STS method using generalized Jacobi polynomials

As in the previous work described in [1, 5, 6] we will focus on second order approximations. In order to expand the stability domain, they consider degree s polynomials of the form:

$$R_s(z) = a_s + b_s P(1 + w_s z), \tag{5}$$

where $P \in \mathbb{R}_s[X]$ and the parameters a_s, b_s and w_s are chosen in order to ensure the second-order and broaden the stability domain. As for the choice of P, Chebyshev or Legendre polynomials are classically used in order to develop STS schemes. In fact, it can be seen that both class of polynomials are subcategories of Jacobi polynomials [7]. A first contribution of this work is then to extend the STS method using the general form of Jacobi polynomials. Following [1, 5, 6], we choose to use a polynomial of the form:

$$R_s(w_s, z) = a_s + b_s J_s^{\alpha,\beta}(1 + w_s z), \tag{6}$$

where $J_s^{\alpha,\beta} = P_s^{\alpha,\beta}/P_s^{\alpha,\beta}(1)$, with $P_s^{\alpha,\beta}$ the Jacobi polynomials as defined in [7], of degree $s \geq 0$, and parameters $\alpha, \beta \geq -1/2$.

Proposition 1 *For $s \geq 2$ there exists a set of parameters a_s, b_s and w_s such that the method is consistent at second order. Moreover, if $\alpha \geq \beta \geq -1/2$, the numerical scheme is stable for each timestep such that $\Delta t w_s \rho(A_h) \leq 1$. Moreover, we have $w_s \sim s^{-2}$ as $s \to +\infty$.*

Proof. The coefficients a_s, b_s and w_s are determined by the second order condition applied to R_s:

$$a_s + b_s J_s^{\alpha,\beta}(1) = 1, \tag{7a}$$

$$b_s w_s J_s^{\alpha,\beta\prime}(1) = 1, \tag{7b}$$

$$b_s w_s^2 J_s^{\alpha,\beta\prime\prime}(1) = 1. \tag{7c}$$

By construction $J_s^{\alpha,\beta}(1) = 1$ and by injecting the square of (7b) into (7c) we obtain, via straightforward computations (see (25)):

$$a_s + b_s = 1 \tag{8a}$$

$$b_s = \frac{J_s^{\alpha,\beta\prime\prime}(1)}{\left(J_s^{\alpha,\beta\prime}(1)\right)^2} = \frac{s-1}{s}\frac{\alpha+1}{\alpha+2}\frac{s+\alpha+\beta+2}{s+\alpha+\beta+1} \tag{8b}$$

$$\frac{1}{w_s} = \frac{J_s^{\alpha,\beta\prime\prime}(1)}{J_s^{\alpha,\beta\prime}(1)} = \frac{s-1}{2}\frac{s+\alpha+\beta+2}{\alpha+2} \tag{8c}$$

As for stability, it has been demonstrated in [7] that $P_s^{\alpha,\beta}$ reaches its maximum value on $[-1,1]$ (in absolute value as well) at $x = 1$ provided that $\alpha \geq \beta \geq -1/2$. Consequently, by normalization of $J_s^{\alpha,\beta}$, $\left|J_s^{\alpha,\beta}(x)\right| \leq 1$ for all $x \in [-1,1]$. Additionally, for $s \geq 2$ (case for which achieving second order is possible) $w_s, b_s > 0$. Consequently if we prove that $b_s \leq 1$, stability properties on R_s directly follow from that of $J_s^{\alpha,\beta}$. But we have:

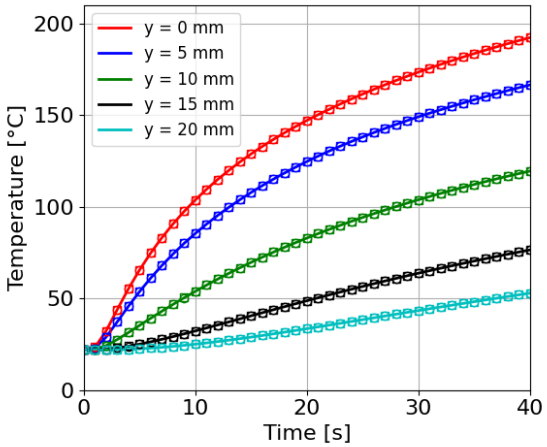
$$b_s = \left(1 - \frac{1}{s}\right)\left(1 - \frac{1}{\alpha+2}\right)\left(1 + \frac{1}{s+\alpha+\beta+1}\right). \tag{9}$$

Consequently, it is enough to show that $(1 - 1/s)(1 + 1/(s + \alpha + \beta + 1)) \leq 1$ to obtain $b_s \leq 1$. For all $0 < B < A < 1$, $(1 - A)(1 + B) \leq (1 - B)(1 + B) = 1 - B^2 < 1$. Since we have $1/s \geq 1/(s + \alpha + \beta + 1)$ as long as $\alpha, \beta \geq -1/2$, we do obtain $b_s \leq 1$.
By construction, we now inherit l^2-stability so long $\Delta t w_s \rho(A_h) \leq 1$. Moreover, we get directly from (8c) that, as $s \to +\infty$, $w_s \sim s^{-2}$.

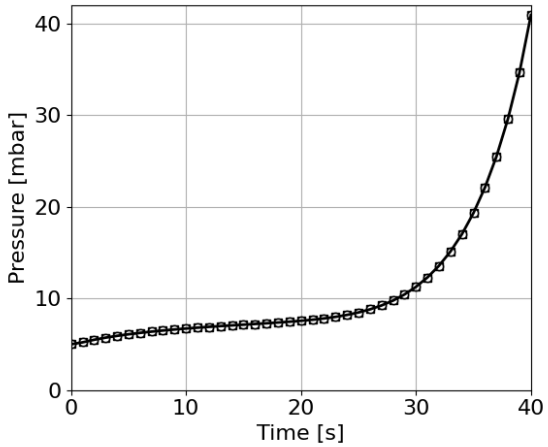
Remark. It follows from Proposition 1 that, for a given degree s, the STS algorithm requires an additional number of operations of order s while allowing the timestep to be s^2 times bigger, resulting in an overall gain in computational cost of a factor s.

2.3 Application on an advection-diffusion problem

To emphasize the advantage of STS methods, a second order Runge-Kutta-Chebyshev scheme is applied to an advection-diffusion problem to be solved by the finite-volume MoDeTheC solver [3, 2] and compared to classical first-order Euler and second-order Heun's schemes. MoDeTheC models heat and mass transfer within pyrolysable anisotropic porous materials, based on Fourier's and Darcy's laws, for which Super-Time-Stepping methods are expected to be very efficient.



(a) Temperature evolution at multiple locations on the unexposed surface of the composite sample along x = 0 cm. [The center of the unexposed surface is located at (0, 0).]



(b) Pressure evolution at the center of the composite sample.

Figure 1: Comparison of Euler, Heun's and Runge-Kutta-Chebyshev schemes for an advection-diffusion problem: lines (Euler [3]), square symbols (Heun), circle symbols (Runge-Kutta-Chebyshev).

A 80 x 80 mm orthotropic composite sample is exposed to a 76.2 kW m^{-2} Gaussian laser source as in [3]. The mesh, as well as the physical setup, initial and boundary conditions, are identical to the previous study, except that oxidation reactions are not taken into account for the sake of simplicity in this numerical study. The time-step is fixed to 1 s, and the maximum admissible Fourier and CFL are respectively set to 0.5 and 1 for Euler and Heun's scheme, 81 and 162 for

the 25 stages Runge-Kutta-Chebyshev scheme.
Figures 1a and 1b show the evolution of respectively the temperature on the unexposed surface of the composite sample and the pressure at the center of the sample for all schemes, with no discrepancies. All simulations are carried out on 48 threads (2 Intel Xeon Cascade Lake - 6240R cores), and cover 40 s. The STS method allows reducing the restitution time by a factor 7.5: 12.7 min for the Heun's scheme, 6.5 min for the Euler scheme, to 1.7 min for the Runge-Kutta-Chebyshev scheme.
This result shows the ability of our STS scheme to accurately simulate the diffusion dynamics while using timesteps equal to several times the standard CFL condition. However, in this setup, the number of steps s of the algorithm is entirely determined by the worst-case CFL in the domain and the desired timestep to reach. For diffusion problems with a space-dependant diffusion coefficient, the number of steps in the integration algorithm might then be overestimated with respect to the local CFL and the target timestep, thus resulting in unnecessary computational expenses. In order to tackle this drawback, we propose in the next section a method for adapting locally the degree s of the approximation.

3 TOWARD A LOCAL STS METHOD

The aim of this section is, for a uniform timestep target on the whole computational domain, to adapt the number of iterations of the STS method depending on the local CFL constraints in order to avoid unnecessary computations where the CFL constraints are reduced.

3.1 STS method as a Taylor expansion of the semi-group

To present in the simplest way the underlying ideas behind the local STS method, we assume that we have subdivided our initial domain Ω into two subdomains Ω_1 and Ω_2 and that we wish to apply on each of these domains an STS methods with respectively s_1 and s_2 stages, as described in the previous section. The difficulty lies in dealing with the interface between the two subdomains. Not only do both methods have different steps, but theses steps are mostly disconnected in the sense that there is no obvious way to interpolate the sub-steps of one method to obtain the values necessary to compute the other.
Indeed, if we follow the classic implementation of an STS method as decribed in Section 2.2, we use the recurrence formula, with obvious notations

$$\begin{aligned} R_s(w_s,x) &= \delta_s^{\alpha,\beta} R_{s-1}(w_s,x) + \eta_s^{\alpha,\beta} x R_{s-1}(w_s,x) + \gamma_s^{\alpha,\beta} R_{s-2}(w_s,x) + \theta_s^{\alpha,\beta} x + \varepsilon_s^{\alpha,\beta}, \\ R_0(x) = a_0 + b_0 = 1, \quad R_1(w_s,x) &= \underbrace{a_1 + b_1 J_1^{\alpha,\beta}(1)}_{=1} + b_1 \frac{\alpha+\beta+2}{2(\alpha+1)} w_s x, \end{aligned} \tag{10}$$

where we only require $a_0, b_0, a_1, b_1 \geq 0$ and $a_0 + b_0 = a_1 + b_1 = 1$. We derive the recurrence coefficients $\delta_s^{\alpha,\beta}, \eta_s^{\alpha,\beta}, \gamma_s^{\alpha,\beta}, \theta_s^{\alpha,\beta}, \varepsilon_s^{\alpha,\beta}$ from recurrence coefficients of Jacobi polynomials found

in [7]:

$$a_s^{\alpha,\beta} = \frac{(2s+\alpha+\beta-1)(2s+\alpha+\beta)}{2s(s+\alpha+\beta)}, \tag{11a}$$

$$b_s^{\alpha,\beta} = \frac{(\beta^2-\alpha^2)(2s+\alpha+\beta-1)}{2s(s+\alpha+\beta)(2s+\alpha+\beta-2)}, \tag{11b}$$

$$c_s^{\alpha,\beta} = \frac{(s+\alpha-1)(s+\beta-1)(2s+\alpha+\beta)}{s(s+\alpha+\beta)(2s+\alpha+\beta-2)}, \tag{11c}$$

later normalized:

$$d_s^{\alpha,\beta} = \frac{s}{\alpha+s}a_s^{\alpha,\beta}, \quad e_s^{\alpha,\beta} = \frac{s}{\alpha+s}b_s^{\alpha,\beta}, \quad f_s^{\alpha,\beta} = \frac{s}{\alpha+s}\frac{s-1}{\alpha+n}c_s^{\alpha,\beta}, \tag{12}$$

to obtain:

$$\begin{aligned}
&\delta_{k,s}^{\alpha,\beta} = \frac{b_k}{b_{k-1}}\left(d_k^{\alpha,\beta} - e_k^{\alpha,\beta}\right), \quad \eta_{k,s}^{\alpha,\beta} = \frac{b_k}{b_{k-1}}d_k^{\alpha,\beta}w_s, \quad \gamma_{k,s}^{\alpha,\beta} = -\frac{b_k}{b_{k-2}}f_k^{\alpha,\beta}, \quad \theta_{k,s}^{\alpha,\beta} = -\frac{b_k}{b_{k-1}}a_{k-1}d_k^{\alpha,\beta}w_s,\\
&\varepsilon_{k,s}^{\alpha,\beta} = a_k - \frac{b_k}{b_{k-1}}a_{k-1}\left(d_k^{\alpha,\beta} - e_k^{\alpha,\beta}\right) + \frac{b_k}{b_{k-2}}a_{k-2}f_k^{\alpha,\beta}.
\end{aligned} \tag{13}$$

In order to tackle the problem if intricate coefficients, we propose to reformulate the method in a slightly different manner. In fact, the STS algorithm can also be developed as

$$\begin{aligned}
R_s(\Delta t A_h) = &\underbrace{a_s + b_s J_s^{\alpha,\beta}(1)}_{=1} + \underbrace{b_s w_s J_s^{\alpha,\beta\prime}(1)}_{=1}\Delta t A_h + \underbrace{b_s w_s^2 J_s^{\alpha,\beta\prime\prime}(1)}_{=1}\frac{\Delta t^2}{2}A_h^2\\
&+ b_s w_s^3 J_s^{\alpha,\beta\prime\prime\prime}(1)\frac{\Delta t^3}{6}A_h^3 + \cdots + b_s w_s^s J_s^{\alpha,\beta(s)}(1)\frac{\Delta t^s}{s!}A_h^s
\end{aligned} \tag{14}$$

This form (14) is more suited for an additive implementation and has the advantage of explicitly using the consistency constraints, while the form (10) can be interpreted in an iterative manner, which is the reason why this latter formulation is usually the one retained in the literature [1, 5, 6]. Nonetheless, let us explicit the drawback of the iterative formulation justifying the need for changing the STS algorithm. In the case of the iterative form (10), the approximation u_{n+1} at t^{n+1} is computed from the approximation u_n at time t^n using the following algorithm:

$$\begin{aligned}
&u_n^0 = u_n\\
&u_n^1 = u_n + b_1\frac{\alpha+\beta+2}{2(\alpha+1)}w_s\Delta t A_h u_n\\
\forall k \in \{2,\cdots,s\}, \quad &u_n^k = \delta_s^{\alpha,\beta}u_n^{k-1} + \eta_s^{\alpha,\beta}\Delta t A_h u_n^{k-1} + \gamma_s^{\alpha,\beta}u_n^{k-2} + \theta_s^{\alpha,\beta}\Delta t A_h u_n + \varepsilon_s^{\alpha,\beta}u_n
\end{aligned} \tag{15}$$

It then becomes obvious that this manner of computation is efficient as long as the same STS method (or the same s) is applied to the whole domain, since to obtain $A_h u_n^{s,k}$ one needs to use the values of neighboring cells together. For this to work, the coefficients need to be the same for all cells at each sub-step. Thus, when different values of s are used on Ω_1 and Ω_2 it is no longer possible to have the same factorization on both subdomains, rendering this formulation unusable.

3.2 Reformulating the STS algorithm

Conversely to the iterative process, the additive formulation (14) allows to decouple some of the coefficient arising from the iterative process

$$\begin{aligned} u_n^0 &= u_n \\ u_n^{s,0} &= u_n \\ \forall k \in \{0, \cdots, s-1\}, \quad u_n^{k+1} &= \frac{\Delta t}{k+1} A_h u_n^k \\ u_n^{s,k+1} &= u_n^{s,k} + b_s w_s^{k+1} J_s^{\alpha,\beta(k+1)}(1) u_n^{k+1} \end{aligned} \tag{16}$$

This reinterpretation needs all different terms to be of the same order of magnitude in order to avoid the sum to be ill-conditioned. But the key point with this algorithm is that storing the sequence $\left(u_n^k\right)$ is independent of the choice of s, allowing for different number of steps for each subdomain.
It still has to be noted that some overlap area between both subdomains will be required for the area with the higher value of s, where it will be necessary for the sequence $\left(u_n^k\right)$ to go further than the sequence $\left(u_n^{s,k}\right)$. The extension of the algorithm to a local number of steps then comes at a price of a higher storage cost. The number of cells required for the overlap will depend on both the difference between s_1 and s_2 and the choice for the discretization A_h of the operator A. Since these parameters are known at the beginning of the simulation, this can entirely be defined in a preprocessing phase, mitigating the overhead of this formulation.

4 NUMERICAL SIMULATIONS

4.1 Validation on a 1D test-case

In order to assess the preservation of the order of approximation by our new local algorithm, we present a 1D test-case with an analytical solution. Let us assume that:

$$\partial_t u - \partial_x \left(D_x \partial_x u\right) = 0, \quad \text{with } D_x(x) = \begin{cases} D_{x_L}, & x_L \leq x < x_C, \\ D_{x_R}, & x_C \leq x \leq x_R. \end{cases} \tag{17}$$

We are looking for solutions of the form $u = u_L \mathbb{1}_{x<x_C} + u_R \mathbb{1}_{x>x_C}$, where $u_L, u_R \in \mathcal{C}^\infty([x_L, x_R])$. The classical Rankine-Hugoniot solution imposes that following relation holds:

$$D_{x_R} \partial_x u_R\left(x_C^+\right) - D_{x_L} \partial_x u_L\left(x_C^-\right) = 0. \tag{18}$$

To obtain an overall $\mathcal{C}^1$ function, we also require:

$$u_R\left(x_C^+\right) - u_L\left(x_C^-\right) = 0, \tag{19a}$$

$$\partial_x u_R\left(x_C^+\right) - \partial_x u_L\left(x_C^-\right) = 0. \tag{19b}$$

Assuming $D_{x_L} \neq D_{x_R}$, the combination of (18) and (19b) yields $\partial_x u_L\left(x_C^-\right) = \partial_x u_R\left(x_C^+\right) = 0$.

Considering homogeneous Neumann boundary conditions, a $\mathcal{C}^1$ solution of (17) may be obtained by choosing

$$u_L(t,x) = u_0 \exp\left(-\frac{t}{\tau_L}\right) \cos\left(2\pi k_L \frac{x - x_L}{x_C - x_L}\right), \tag{20a}$$

$$u_R(t,x) = u_0 \exp\left(-\frac{t}{\tau_R}\right) \cos\left(2\pi k_R \frac{x - x_C}{x_R - x_C}\right), \tag{20b}$$

with $k_L, k_R \in \mathbb{Z}$ and

$$\frac{1}{\tau_L} = D_{x_L}\left(\frac{2\pi k_L}{x_C - x_L}\right)^2, \qquad \frac{1}{\tau_R} = D_{x_R}\left(\frac{2\pi k_R}{x_R - x_C}\right)^2. \tag{21}$$

Conditions (18) and (19b) always hold for this form of solution, and yield a classical solution on $[x_L, x_C[$ and $]x_C, x_R]$. Only the continuity condition (19a) is satisfied if and only if $\tau_L = \tau_R$, or equivalently:

$$D_{x_L}\left(\frac{k_L}{x_C - x_L}\right)^2 = D_{x_R}\left(\frac{k_R}{x_R - x_C}\right)^2. \tag{22}$$

For instance, a possible choice of parameters is $x_C = \frac{x_L + x_R}{2}$, $k_R = 2k_L$, $D_{x_L} = 4D_{x_R}$.
The reference solution at initial and final time is displayed on Figure 2 for $t_f = 0.001$, $D_{xL} = 1$, $D_{xR} = 4D_{xL}$.

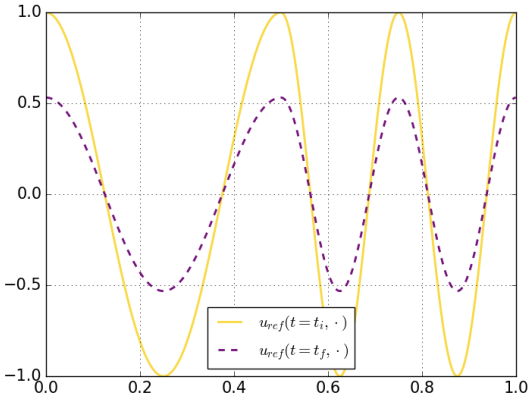
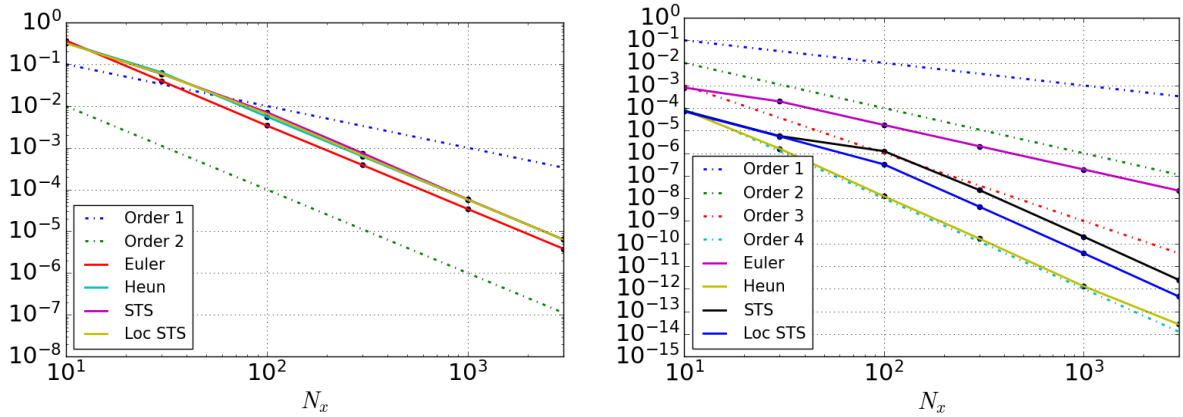


Figure 2: 1D exact solution.

In each case the timestep was calculated so that it satisfies a CFL = 0.95. We can verify on this test case that both the global and local STS enforce at least second-order order. A fourth-order is even reached in the case of a fourth-order diffusion operator, due to the regularity of the solution. It can also be seen (black curve on Figure 3(b)) that when s is too high relatively to N_x, there might be a loss of order due to the amount of additional steps involved.

4.2 Expansion to a 2D application case

Similarly a 2D test case where we have discontinuities for the diffusion coefficient in both x and y directions was set up to verify that the STS methods give satisfying convergence orders. The



(a) Using a 2nd order diffusion operator.

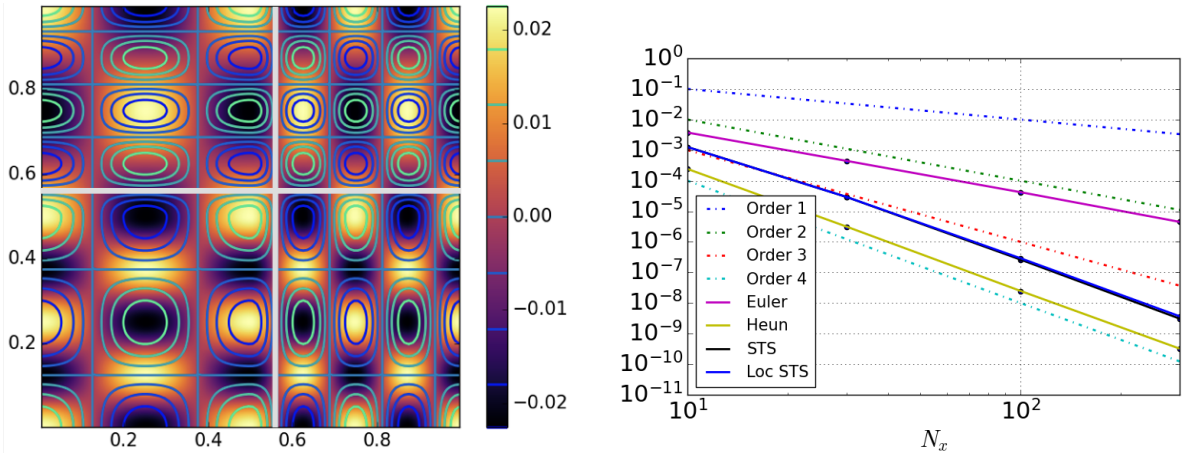
(b) Using a 4th order diffusion operator.

Figure 3: Convergence orders of global and local STS methods compared to classical Euler and Heun methods on a 1D test case.

discontinuities occur for $x = x_C$ and $y = y_C$ creating four different zones. We then consider a diffusion phenomenon with an anisotropic diffusion coefficient:

$$\begin{aligned} \partial_t u + \nabla.(\underline{D}\nabla u) &= 0 \\ D_{xL} = 4 \quad &; \quad D_{xR} = 1 \\ D_{yU} = 1 \quad &; \quad D_{yD} = 4 \end{aligned}$$

where L, R, U and D subscripts respectively stand for left, right, upper and down parts of the domain. In the same way as the 1D case, an analytical solution can also be determined. It will not be detailed here but is displayed on Figure 4a on the left with $x_C = y_C = 0.625$.



(a) Exact solution with a discontinuous diffusion coefficient.

(b) Convergence orders for each time integration method with a 4th-order diffusion operator.

Figure 4: 2D test case analysis.

On Figure 4b are displayed the l^2 error obtained with the different methods against the number of points. In this example the STS parameters were $\alpha = \beta = 0$ corresponding to the Runge-Kutta-Legendre method, with $s_R = 4$ in the right domains $s_L = 4s_R$ on the left domains for the local method, and $s = s_L$ with the global STS method. We have ensured here that at least a 2nd-order convergence rate is obtained with the STS method applied with the same order in the whole domain as well as with the local STS method. Like in the 1D case, higher orders are achieved due to the regularity of the solution but are not provided by the STS schemes.

4.3 Computational gains

In order to estimate the computational savings induced by our method, we determine the number of evaluations of the discretized operator required. For the sake of simplicity, this is done in a 1D context but can easily be extended to higher dimensions. To this extent, we recall that $\Delta x = L/N_x$ and set $D_{x_{\max}} = \max(D_{x_L}, D_{x_R})$. We first explicit in Table 1 the timestep required by each method:

	Euler	Heun	STS	STS Loc
$\Delta t = \text{CFL} \times$	$\frac{\Delta x^2}{2D_{x_{\max}}}$	$\frac{\Delta x^2}{2D_{x_{\max}}}$	$\frac{\Delta x^2}{2D_{x_{\max}} w_s}$	$\frac{\Delta x^2}{2} \min\left(\frac{1}{D_{x_L} w_{s_L}}, \frac{1}{D_{x_R} w_{s_R}}\right)$

Table 1: Time step value for all time integration methods.

We now set as reference $N_{t_{ref}} = \lfloor (t_f - t_i)/\Delta t \rfloor$ with Δt defined from Euler method and where we have set $T = t_f - t_i$ and $\lfloor x \rfloor$ denotes the integer part of x. We assume without loss of generality that $D_{x_L} \geq D_{x_R}$. To keep the maximum time steps in both areas aligned we need $D_{x_L} w_{s_L} \approx D_{x_R} w_{s_R}$ which is ensured in practice by imposing $s_L/s_R \approx \sqrt{D_{x_L}/D_{x_R}}$ via the formula $s_R = \lfloor s_L \sqrt{D_{x_R}/D_{x_L}} \rfloor$. We further assume that to maintain similar performances as the usual STS method we have $s_L = s$.
The total cost of the method is obtained by multiplying the number of steps N_t by the number of stages in one step as well as the number of cells. Regarding the STS method we have for asymptotically large s:

$$s \times \lfloor \frac{2TD_{x_{\max}} w_s}{\Delta x^2} \rfloor \times N_x \sim s w_s N_{t_{ref}} \times N_x \sim \frac{N_{t_{ref}} \times N_x}{s}. \tag{23}$$

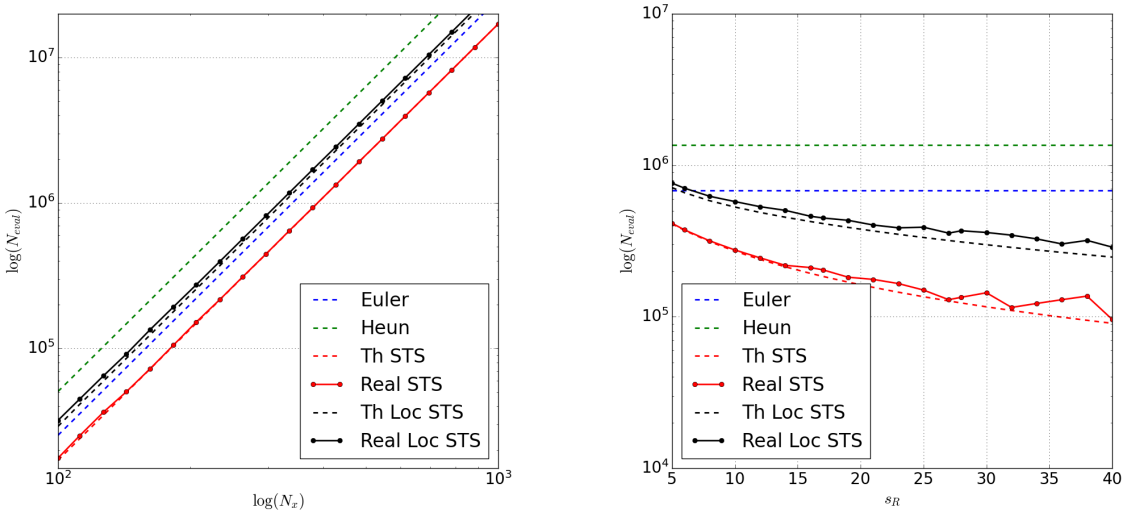
Finally, under our assumptions we have for the local STS method a cost of:

$$s_L \lfloor \frac{2TD_{x_L} w_{s_L}}{\Delta x^2} \rfloor \times N_{x_L} + s_R \lfloor \frac{2TD_{x_R} w_{s_R}}{\Delta x^2} \rfloor \times N_{x_R} \sim \frac{N_{t_{ref}}}{s} \left(N_{x_L} + \frac{s_R}{s_L} N_{x_R} \right). \tag{24}$$

This formula demonstrates that the local method is especially relevant when the diffusion coefficient is large only in a small area, i.e. $D_{x_L} \gg D_{x_R}$ and $N_{x_L} \ll N_{x_R}$.
This is summarized in the following table of total cost, where we recall $w_s \sim 1/s^2$ as $s \to \infty$:
On Figure 5 are represented the number of evaluations for each method depending on the number of points with a fixed number of steps of the STS methods (Fig. 5a, $s_R = 4$ and $s_L = 4s_R$ for the local method and $s = s_L$ for the global method) and depending on the order s of the STS

Euler	Heun	STS	STS Loc
$N_{t_{ref}} \times N_x$	$2 \times N_{t_{ref}} \times N_x$	$\sim \frac{N_{t_{ref}} \times N_x}{s}$	$\sim \frac{N_{t_{ref}}}{s} \times \left(N_{x_L} + \frac{s_R}{s_L} N_{x_R} \right)$

Table 2: Number of evaluations of A_h for all time integration methods.



(a) Against the number of points N_x with $s_R = 4$ and $s = s_L = 4s_R$.

(b) Against the method order on right side s_R and $s = s_L = 4s_R$.

Figure 5: Theoretical (dashed) and effective (plain) number of evaluations for all time integration methods.

method with a fixed number of points (Fig. 5b, $N_x = 300$). The same physical parameters as in Section 4.1 are chosen.

As expected the number of evaluations of the discretized operator decreases when we increase the method order, making both the global and local STS more efficient than classical time schemes as soon as the method order is high enough, while keeping a 2nd-order convergence order. The current implementation of both the global and local STS scheme reach the optimal number of evaluations, where the small differences between theoretical and real curves are due to floor effects when calculating the number of timesteps with the STS methods.

5 CONCLUSIONS

We have presented both global and local timestep Super-Time-Stepping methods. Global timestep STS methods were developed for solving parabolic equations while lowering stability constraints on the CFL number. These methods have proved to be very efficient for solving heat and mass transfer equations in anisotropic porous materials. First initiatives proposed in this study to extend STS methods to local timestep integration have proven to be of primary interest to further reduce the computational cost of time schemes. To that end, it is necessary

to utilize the physical problem topology to explicitly distinguish zones where the number of steps of the STS methods can be lowered relative to the local stability condition, which is the case for fire-safety applications that motivated the study. Further work will be dedicated to the application of the local timestep STS method to heat and mass transfer within anisotropic porous materials. Analysis of the conservativeness and appropriate corrections to the schemes should also be investigated.

A Computational aid

Assuming $J_s^{\alpha,\beta}(1) = 1$, one can derive from (3.94) and (3.96) in 3.2.1.1 of [7] that:

$$J_s^{\alpha,\beta\prime}(1) = \frac{s}{2}\frac{(s+\alpha+\beta+1)}{\alpha+1}, \qquad J_s^{\alpha,\beta\prime\prime}(1) = \frac{s\,(s-1)}{4}\frac{(s+\alpha+\beta+1)\,(s+\alpha+\beta+2)}{(\alpha+1)\,(\alpha+2)}, \tag{25a}$$

$$\frac{J_s^{\alpha,\beta\prime\prime}(1)}{J_s^{\alpha,\beta\prime}(1)} = \frac{s-1}{2}\frac{s+\alpha+\beta+2}{\alpha+2}, \quad \frac{J_s^{\alpha,\beta\prime\prime}(1)}{J_s^{\alpha,\beta\prime}(1)^2} = \frac{s-1}{s}\frac{\alpha+1}{\alpha+2}\frac{s+\alpha+\beta+2}{s+\alpha+\beta+1} \tag{25b}$$

REFERENCES

[1] Vasilios Alexiades, Geneviève Amiez, and Pierre Alain Gremaud. Super-time-stepping acceleration of explicit schemes for parabolic problems. *Communications in Numerical Methods in Engineering*, 12(1):31–42, 1996.

[2] Nicolas Dellinger, David Donjat, Emmanuel Laroche, and Philippe Reulet. Experimental and numerical modelling of the interaction between a turbulent premixed propane/air flame and a composite flat plate. *Fire Safety Journal*, 141:103959, 2023.

[3] Nicolas Dellinger, Gillian Leplat, Cédric Huchette, Valentin Biasi, and Frédéric Feyel. Numerical modeling and experimental validation of heat and mass transfer within decomposing carbon fibers/epoxy resin composite laminates. *International Journal of Thermal Sciences*, 201:109040, 2024.

[4] K. F. Gurski and S. O'Sullivan. An Explicit Super-Time-Stepping Scheme for Non-Symmetric Parabolic Problems. *AIP Conference Proceedings*, 1281(1):761–764, 09 2010.

[5] Chad D. Meyer, Dinshaw S. Balsara, and Tariq D. Aslam. A second-order accurate Super TimeStepping formulation for anisotropic thermal conduction. *Monthly Notices of the Royal Astronomical Society*, 422(3):2102–2115, 2012.

[6] Chad D. Meyer, Dinshaw S. Balsara, and Tariq D. Aslam. A stabilized Runge-Kutta-Legendre method for explicit super-time-stepping of parabolic and mixed equations. *Journal of Computational Physics*, 257(PA):594–626, 2014.

[7] Jie Shen, Tao Tang, and Li-Lian Wang. *Spectral Methods : Algorithms, Analysis and Applications*. Springer, 2011.