

# Semi implicit solver for high fidelity LES/DNS solutions of reacting flows

A.Surapaneni<sup>a</sup>, D.Mira<sup>a</sup>

<sup>a</sup>Barcelona Supercomputing Center (BSC), Barcelona, Spain.

---

## Abstract

A semi-implicit/point-implicit stiff solver (ODEPIM) for integrating chemistry in context of high fidelity LES/DNS simulations is presented. A detailed overview of the algorithm and its numerical formulation is discussed. The solver is then compared against a state-of-the-art multi-order implicit solver CVODE in terms of accuracy and costs. It was found that for typical LES/DNS timestep sizes ODEPIM was about one order faster than CVODE, which would make it a compelling alternative to pure implicit methods. ODEPIM, as mentioned in the literature depends on a fixed sub-timestep size to do the integration steps, this limits the speedup that can be achieved by the solver. A modification to the ODEPIM algorithm to determine the sub-timestep size dynamically is proposed enabling greater speedup. Solutions of a triple flame problem obtained using static and dynamic ODEPIM solvers are compared against the reference solutions obtained with CVODE. The dynamic ODEPIM solver was found to use the maximum permissible sub-timestep size, which on average was 8 to 4 times higher than the fixed sub-timestep size of the static ODEPIM solver. The size of the sub-timestep size directly correlates to the cpu cost, hence the dynamic ODEPIM solver is significantly faster than the static solver, this improvement however, comes at negligible loss in accuracy.

### Keywords:

Finite Rate Combustion (FRC), Strang Splitting, Stiff chemistry integration, Semi-implicit, ODEPIM, CVODE

---

## 1. Introduction

With the advancements in HPC technology and the onset of the upcoming Exa-scale computing era, detailed chemistry simulation can now be used in practical LES/DNS applications to understand complex thermo-chemical phenomena. A typical finite rate solver, in addition to the Navier-Stokes equations would involve solving an Advection-Diffusion-Reaction (ADR) equation for every species present in the chemical mechanism. Chemical timescales are often orders of magnitude less than flow timescales and in order to have an efficient solver splitting methods are to be applied[1]. These methods separate the stiff reaction term from the advection and the diffusion terms which can then be treated explicitly and the chemical source term can be treated with a stiff solver.

A stiff solver has to be implicit in nature because to have a stable explicit solution, integration has to be done in timestep sizes of the order of the chemistry. Critical chemical timestep sizes can be 3 to 5 orders

smaller than the flow timescales. Hence a pure explicit solution would be computationally unfeasible, at least for simulations in the low-Mach region, which is the focus of this study.

Multi-order implicit schemes, though stable at larger timesteps are computationally expensive scaling with the square of the number of species present in the mechanism[2]. Sizes of chemical mechanisms are continuously growing as more complex thermo-chemical phenomena are understood. Hence integration of chemical sources based on such large mechanisms using implicit methods is going to be a major bottleneck for finite rate chemistry simulations. A solution to this could be methods which combine the accuracy and stability of implicit methods and the computational speed of an explicit method.

Liang et.al [3] mentions that by separating the chemical sources into production and destruction terms and by treating the destruction term implicitly, the stability of an explicit method can be improved greatly. By this treatment fully explicit methods can now be used to integrate stiff chemical mechanisms, however by treating the destruction terms implicit there is an error introduced, the magnitude of the error is shown in Equa-

---

\*Corresponding author: Anurag Surapaneni

Email address: anurag.surapaneni@bsc.es (A.Surapaneni)

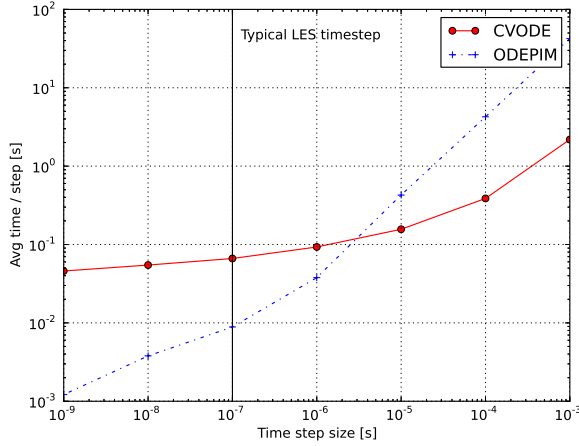


Figure 1: Cost per timestep

tion 2. For large destruction rates the error introduced would be substantial, to alleviate this Katta et.al [4] proposed to add a point-wise convergence loop making the method semi-implicit. The point implicit solver (ODEPIM) was used with success by Katta et.al to simulate canonical combustion problems with detailed chemistry. Accuracy of ODEPIM was found to be closer to implicit methods, while the computational cost was found to be closer to explicit methods.

Figure 1 shows the time taken for chemistry integration at various timestep sizes by CVODE and ODEPIM in a homogeneous auto-ignition problem of stoichiometric mixture of  $\text{C}_4\text{-Air}$  at 1500 K and 1 bar of pressure using the GRI 3.0 mechanism [5]. A typical timestep size for an explicit LES problem would be in the order of  $10^{-7}$  s and it is seen that for these timestep sizes time taken by ODEPIM is almost one order less than that of CVODE, however at larger timesteps ( $10^{-5}$ ) ODEPIM, due to its explicit formulation becomes more expensive than CVODE. This is due to increase in number of sub-iterations required in order to ensure stability. Since the current study focuses on the use of ODEPIM for high fidelity LES and DNS simulations, timestep sizes are always in favor of ODEPIM.

The original algorithm for ODEPIM relied on a fixed timestep size definition for the inner iterative loop, this limits the speedup that can be achieved by the method as the timestep sizes are independent of the chemical state. A dynamic implementation of ODEPIM is proposed wherein, the timestep is dynamically determined by the chemical state of the problem.

## 2. Algorithm

A typical species source term can be written as Equation 1, here  $I^+$  and  $I^-$  represent the production and destruction source terms respectively.

$$\frac{d\dot{Y}_i}{d\ell} = I_i^+ + I_i^- \quad (1)$$

By treating the destruction term implicit and rearranging the terms we obtain Equation 2.

$$\frac{\dot{Y}_i^{n+1} - \dot{Y}_i^n}{\Delta\ell} = I_i^+ + I_i^- \left( \frac{\dot{Y}_i^{n+1}}{\dot{Y}_i^n} \right) \quad (2)$$

Here the term in the brackets aims to reduce the stiffness of the problem to allow for the use of larger steps during integration, however this term also introduces and error and is hence to be minimised. Expressing Equation 2 in discrete form gives a point implicit formulation for  $\dot{Y}_i^{n+1}$  as seen in Equation 3.

$$\dot{Y}_i^{n+1} = \frac{\dot{Y}_i^n + I_i^+ \Delta\ell}{1 + \frac{I_i^- \Delta\ell}{\dot{Y}_i^n}} \quad (3)$$

Equation 3 represents a damped Euler forward integration scheme, however advancing explicitly in Equation 3 is unstable and inaccurate. Hence an iterative loop (<) is performed in order to minimise the error during the integration, mathematically shown in Equation 4. States at time  $\ell = \ell^n$  are set as the initial state for the sub-iteration at  $\ell = 0$ .

$$\dot{Y}_i^{n+1-<} = \frac{\dot{Y}_i^n + I_i^+ \Delta\ell}{1 + \frac{I_i^- \Delta\ell}{\dot{Y}_i^{n-<}}} \quad (4)$$

$$n = \left\langle \begin{matrix} 0G \\ 1 \leq \leq 8C4A \end{matrix} \left( 1 - \frac{>6_{10} \cdot \dot{Y}_i^{n-<}}{>6_{10} \cdot \dot{Y}_i^{n-<+1}} \right) \dot{Y}_i^{n-<} \right\rangle \quad (5)$$

Once the error ( $n$ ) satisfies Equation 5 then iterations stops and the solution advances from  $\ell^n$  to  $\ell^{n+1}$ . In order to accelerate convergence, species with concentrations less than ( $10^{-3}$  ppm), which tend to have large uncertainties are not considered during the evaluation of Equation 5. ODEPIM solver as presented in the literature used a fixed timestep of the order of (100 ns), though this is good estimate of the timestep which minimises the error, the static nature of the solver does not allow for maximum speed up. In this study critical timestep for chemistry is estimated using the chemical source of the fastest species, and this is used as the timestep

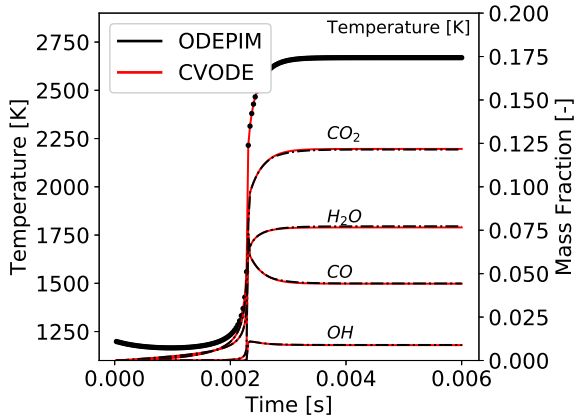


Figure 2: Homogeneous Auto-ignition,  $T_G = 1200.0$  [K]

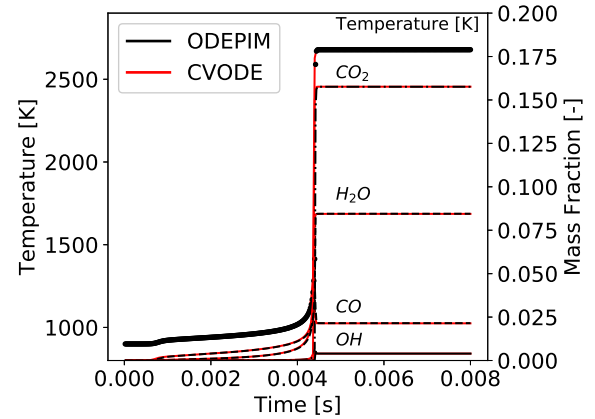


Figure 3: Homogeneous Auto-ignition,  $T_G = 900.0$  [K]

size for ODEPIM. The viability of this timestep size is determined by changes in key variables which is computed at the end of the integration. Key variables could be species of interest, fastest reacting species or thermodynamic variables. If changes in key variables are larger than the specified tolerance the timestep size is reduced by a factor, the previous solution is discarded and the entire timestep is run again using the modified timestep size. This iterative loop is done to limit the error accumulated during integration. The algorithm for the modified ODEPIM using dynamic timesteps is summarised in Algorithm 1

```

1: INPUTS = Temp, Press, #,  $\beta$ 
2: while  $\Delta A_j > \Delta A_{j, \text{max}}$  do
3:    $\beta_{BD1} = \beta_{BD1} / \alpha$ 
4:   Set_state_to_inputs
5:    $\beta_{DAA4} = 0$ 
6:   while  $\beta_{DAA4} < \beta_{DAA4, \text{max}}$  do
7:     while  $\Delta A_j > \Delta A_{j, \text{max}}$  do
8:       Solve Equation --  $j$  (4)
9:       Compute error --  $j$  (5)
10:    end while
11:     $\beta_{DAA4} = \beta_{DAA4} + \beta_{BD1}$ 
12:  end while
13: end while

```

**Algorithm 1:** Dynamic ODEPIM

### 3. Validation

Temporal evolution of two homogeneous auto-ignition problems of a stoichiometric mixture of N-heptane and air at 16 bar pressure and 1200 K and 900

K temperature, respectively are computed using ODEPIM and CVODE. Chemistry is computed using Lu and Law's [6] mechanism containing 188 species and 1719 reactions making the mechanism extremely stiff. A relatively large timestep size of  $10^{-5}$  s was chosen and the simulation was run till equilibrium. The stiff mechanism and the large timestep were meant to present the worst case scenario for a semi-implicit method like ODEPIM, though in this particular case ODEPIM was more expensive than CVODE, this test aimed to portray the stability and the accuracy of ODEPIM against pure implicit methods. As seen in Figure 2 and Figure 3, solutions from both CVODE and ODEPIM are identical in terms of temperature and species.

### 4. Dynamic ODEPIM

Critical timestep size for the integration is computed based on local chemical sources, the viability of this timestep size is then assured by limiting changes in the concentration of  $\beta$  and temperature, the idea is to limit changes in the thermo-chemical state so as to reduced the instantaneous and accumulated error.  $\beta$  species is one among the fastest reacting species and is known to directly correlate to the heat released during combustion, and hence was used as an indicator to capture changes in the thermo-chemical state. A typical iteration with static and dynamic ODEPIM is shown in Figure 4

ODEPIM with a fixed timestep size is compared against the proposed dynamic ODEPIM solver in a 2D detailed simulation of a triple flame. The motivation behind choosing the triple flame was that it was a well studied and understood problem and represents an ideal

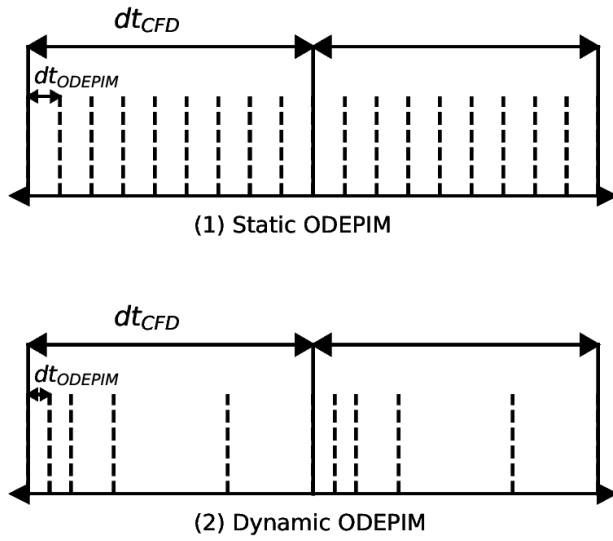


Figure 4: Timestep in static and Dynamic ODEPIM

test case for multi-regime complex chemistry combustion problem.

A typical triple flame contains of a propagating stoichiometric flame front which forms the leading edge with trailing rich and lean partially premixed flames developing towards the fuel and the oxidiser streams, respectively. A post-diffusion trailing flame is formed downstream of the partially premixed flame. There have been many configurations of the triple flame, the current study uses the setup as mentioned in [7], the velocity of the incoming stream is scaled according to the stoichiometric flame speed of the fuel.

The triple flame problem was set for a  $\phi = 1.4$  - Air configuration at 1 bar and chemistry was computed using the GRI 3.0 mechanism [5]. The problem was run with a constant fixed CFD timestep size of  $6 \cdot 10^{-6}$  s and an initial sub timestep size of  $2 \cdot 10^{-6}$  s is chosen for the dynamic ODEPIM solver. This sub-timestep size is 20 times larger than the value mentioned in the literature, making the dynamic method, ideally 20 times faster. The tolerance as defined in Equation 5 was fixed to  $10^{-5}$  with an iteration (#*IC4A*) limit of 20. For the dynamic ODEPIM solver changes in key variables -  $\phi$  concentration and Temperature was limited to 1 % and 5 % respectively. In case of changes larger than the limits the solver would reduce the timestep size by an order of 2 and the entire timestep would be integrated again with the modified timestep. The choice of limits on the changes in key variables and reduction factor for the timestep size was driven by experience gained

after running a wide range of simulations, these values can however be modified to achieve different dynamic behaviour.

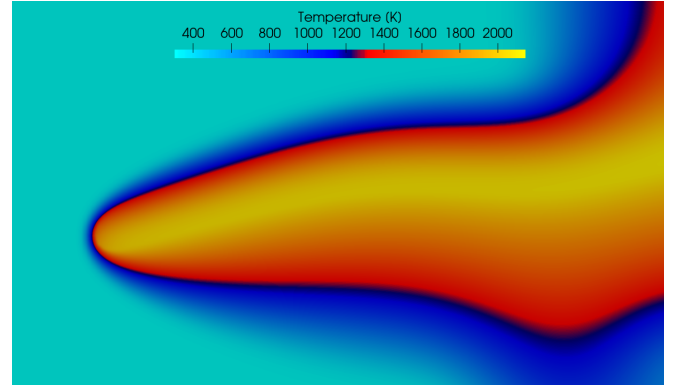


Figure 5: Triple flame - Temperature [K]

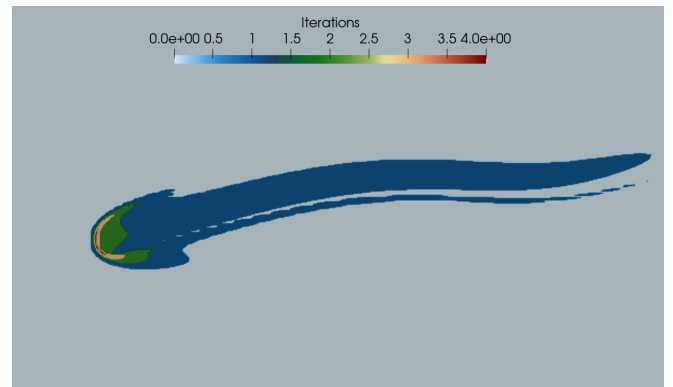


Figure 6: Triple flame - Dynamic ODEPIM iterations [-]

Simulations were carried out using an in-house multi-physics code - ALYA [8], Figure 5 shows the temperature profile obtained after running the simulation for 10 [ms]. Figure 6 shows the number of dynamic iterations the solver had to take in order to satisfy the criteria of tolerance and change in key variables, mentioned previously. It is observed that the solver takes more iterations ( $\approx 4$ ) in regions with high-reactivity i.e, the premixed flame front. In moderately reacting zones the solver takes between 2-3 steps and in the lower reacting zone the solver takes just one step. As the initial sub-timestep size for the dynamic ODEPIM method was 20 times larger than the reference, even in the most reacting parts of the flame where the dynamic solver needed 4 iterations it was faster than the reference as the reduced sub timestep size was still 1.25 times larger. The accuracy of the dynamic ODEPIM solver is verified by compar-

