Comparison between two fluidodyinamic conditions in a tubular reactor

Elisa Giovanna Faggioli, [a] Marcello Ferraro [b]

[a] *10628955, elisagiovanna.faggioli@mail.polimi.it*

[b] *10602732, marcello.ferraro@mail.polimi.it*

Abstract

The aim of this project is to investigate two different fluidodyinamic conditions inside of a triple-entrance reactor in which multiple reactions occur, both in series and in parallel. The final analysis of the configurations and a comparison between the two will be carried out by comparing the amount of species E produced at steady-state conditions. To solve the problem, the whole domain has been split into five sub-domains solved separately, the projection algorithm has been used as well as the successive over-relaxation (SOR) method for the Poisson equation for pressure. For the mass balance equations for species, the operator splitting method has been applied together with an explicit Euler solution, that provided the optimal balance between speed and precision. To better manage the variations of velocities in the reaction environment, the hybrid discretization scheme has also been implemented, with a using a Péclet number of 2 as a divide. The results show that the velocity profile in the whole reactor is regular and smooth, with numbers coherent with the given configurations, and the pressure field is almost constant, with small variations due to microscopic pressure changes. The concentrations of the species have regular profiles and in particular species E presents an hourly production of 18.64 and 14.68 mol/m^2 for the two considered schemes.

1 Introduction: domain description and operating conditions

To correctly understand the simulation hat has been carried out in all of its parts, the operating conditions need to be addressed first. The reactor that has been used is a two-dimensional tubular reactor with length of $1m$ and total height of $0.1m$ (Fig. 1).

1.1 Computational domain

The computational domain can then be split into five sub-domains: three inlet sections (the top and bottom ones with height of $0.01m$ and a central one with height of $0.05m$) with length equal to $0.2m$, a mixing chamber of dimensions $0.2m$ in length and $0.1m$ in height, and an outlet pipe with height of $0.05m$ that takes up the remaining length of the reactor ($0.6m$). These five subdomains will be used in the numerical solution and each will be solved separately.

The five species and the elementary reactions taken into consideration are:

$$A \to B$$

$$B \to C$$

$$B + D \to 2E$$

and they are ruled respectively by the following reaction rates:

$$r_1 = k_1 C_A \tag{1}$$

$$r_2 = k_2 C_B \tag{2}$$

$$r_3 = k_3 C_B C_D \tag{3}$$

These three reactions occur in isobaric conditions at constant temperature and in the gaseous phase (this means constant density using the ideal-gas-law approximation (4)), with transport properties independent from the composition (hence constant as well). Note that R is the universal gas constant equal to $8.314 J/mol/K$, and all other quantities are expressed accordingly.

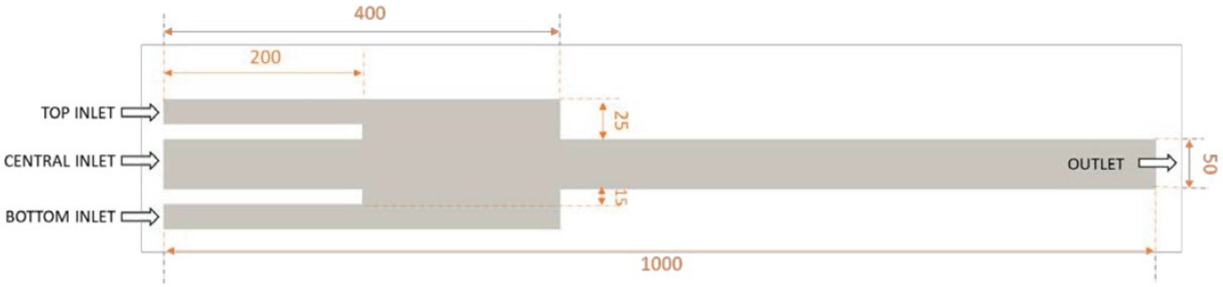

Fig. 1. Computational domain

$$pV = nRT \tag{4}$$

Below, the fluidodyinamic and kinetic data are provided.

1.2 Data

The molecular weight of species A and D is $MW = 46kg/kmol$, and it is the same for all other species as well, since the reactions are either isomerisations or reactions between species with the same molecular weight. For this reason, the molecular weight of the whole mixture is constant in both space and time.

The dynamic viscosity is $\mu = 10^{-4}\frac{kg}{ms}$, and the diffusion coefficient of all species is $\Gamma = 4 \times 10^{-5}\frac{m^2}{s}$, both constant in space and time.

The kinetic constants for equations (1), (2) and (3) are respectively:

$$k_1 = 3.5\frac{1}{s}$$

$$k_2 = 1.20\frac{1}{s}$$

$$k_3 = 95 \times 10^{-6}\frac{m^3}{mols}$$

Starting from these data, two fluidodynamic conditions have been taken into consideration as they differ for inlet velocities and species distribution, and they will be better presented below.

1.2.1 Case 1

In the first case, the top and bottom pipes are fed with pure A with velocity $u = 1m/s$ [1], while the central pipe is fed with pure D with velocity $u = 0.5m/s$. According to these initial data, a first deduction can be made that reactions will occur in the side pipes only, since species D will react when in contact with species C, and this happens from the mixing chamber onward.

1.2.2 Case 2

In the second case, the top and bottom pipes are fed with pure D with velocity $u = 0.5m/s$, and the central pipe is fed with pure A with velocity $u = 1m/s$. This clearly means that the reactions will occur in the opposite places with respect to case one, and this will result in a different profile in the mixing chamber as well.

2 Governing equations and boundary conditions

Now that the reactor has been described and the fundamental data has been provided, the governing equations and the initial and boundary conditions for solving the problem need to be addressed.

2.1 Governing equations

The equations that have been taken into account as a starting point when solving the problem are the continuity equation ((5), here in its two-dimensional formulation), the Navier-Stokes equation (6) and the mass balance equations for all species (generically speaking, in the form of equation (7)), all in differential form.

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{5}$$

$$\rho\frac{D\vec{v}}{Dt} = \mu\nabla^2\vec{v} - \nabla\vec{p} + \nabla\rho\vec{g} \tag{6}$$

$$\frac{DC_j}{Dt} = \Gamma\nabla^2 C_j + S(\vec{C}, t) \tag{7}$$

Some important considerations must be made upon these equations:

[1] u is to be considered the horizontal velocity.

1. the velocity vector $\vec{v}$ stands for the vectorial velocity in all three dimensions. This gets then reduced to a two-dimensional vector when analysing this particular case. It will so contain the two istantaneous velocities u and v [2] , in the form $\vec{v} = u\hat{i} + v\hat{j}$;
2. the term D/Dt that appears in both (6) and (7) is called **substantial derivative** (or Lagragian derivative) of a variable, and it can be expanded as in equation (8). This expanded format of the derivative will then be used in its discretised form;
3. as will be later discussed, neither the Navier-Stokes equation nor the mass balance equations will be solved directly in the shape they are presented. Instead, they will both be split into more easily-solvable parts by making use of the **Poisson equation** (9) and the **Advection-Diffusion equation** (10), which for simplicity have already been written in their 2D formulation but they can be easily extended to a 3D problem;
4. since the density ρ is considered constant and $\vec{g}$ is the gravitational constant, the last term of (6) gets simplified. If ρ wasn't constant by hypothesis, an additional equation describing the evolution of density should be considered (van der Waals or other) but this shouldn't be costly in terms of computational time;
5. equation (11) will also be used together with (10) in the solution of the mass balance equations for species, and it is namely the mass balance equation for a batch reactor. In this case, the reaction term R_j will correspond to the source term $S(\vec{C}, t)$ present in equation (7);

$$\frac{D\phi}{Dt} = \frac{\partial \phi}{\partial t} + \vec{v} \cdot \nabla \phi \tag{8}$$

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = S(x, y) \tag{9}$$

$$\frac{\partial \phi}{\partial t} + u\frac{\partial \phi}{\partial x} + v\frac{\partial \phi}{\partial y} - \Gamma(\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2}) = 0 \tag{10}$$

$$\frac{\partial C_j}{\partial t} = R_j \tag{11}$$

Note that R_j is the reaction rate of species j and it is defined as $R_j = \sum_{i=1}^{NR}(\nu_{i,j} r_i)$

2.2 Initial and boundary conditions

Briefly, regarding initial conditions, it should be said that the velocities and concentrations profiles are obtained at steady-state conditions, which are not influenced by the initial conditions. For this reason, the volume at $t = 0$ has been set filled with an **inert species**. This approach is also useful for testing if the numberical solution of the fluidodynamics (so with no reaction involved) is correct. This was made possible by the operator splitting method that has been applied, because velocities and mass balances are decoupled in their solution.

On the contrary, since all the variables are not calculated at a macroscopic level but they are determined on a cell-to-cell basis, boundary conditions are a necessary addition for the solution of the problem. In this multi-entrance geometry, all walls and inlet sections should be accurately addressed when writing boundary conditions, as well as outlet sections. In particular, **Neumann boundary conditions** (12) at the oulet section of the reactor have been used for continuity of fluxes [3] , while **Dirichlet boundary conditions** (13) are applied at every wall [4] . Moreover, the inlet sections also make use of the latter BCs (13) because the velocities there are known and set to the values described in (1.2.1) and (1.2.2). These boundary conditions have been then discretised and used for the finite-volume method, and their discretised forms are equations (14) and (15) [5] , better described in section 3.2.1.

Another important remark regarding the particular solution that will be later presented is that the total volume of the reactor has been split into five subdomains, already described in (1.1). This solution does not cause additional relevant errors because of the method used for linking the subdomains: the outlet values of each section (determined using (12)) will be stored in a separate vector and then given as a Dirichlet boundary condition to the following subsection, using again equation (13). This method, while it adds a relevant number of boundary conditions, is effective in not introducing additional numerical errors and it can also be applied to solve a more complex problem in parallel with multiple CPUs.

$$\frac{\partial \phi}{\partial x} = 0 \tag{12}$$

[2] non-vectorial v is the y component of the velocity field.

[3] the derivative of the function in the considered point is equal to zero.

[4] the parallel component of the velocity field is set to zero.

[5] both discretised along the x-axis, they follow the same discretisation along y.

$$\phi_{i,j} = \alpha \tag{13}$$

3 Discretisation and solution

As mentioned in (2.1), the equations cannot be solved numerically as they are because this would require the solution of partial differential equations which in general is difficult or impossible. Hence, they need to be transformed into a discretized, pointwise version of themselves for them to be able to give a field of values along both axis and in time as well.

In particular, the fields we are interested in are the two velocity fields, the pressure field and the C_j fields. In principle, the physical domain could be described with one computational grid only, but in practice **staggered grids** are required because it is the only way to correctly describe the pressure field and not fall in contradictions. This does not change the discretisation principles illustrated here, but it does require caution when applying boundary conditions and a field reconstruction in the post-processing phase is needed. The particular choices made in this simulation will be better presented in section 3.2, but first, the numerical techniques adopted need to be addressed.

3.1 Numerical techniques

Below are in short form the numerical techniques implemented for the solution of the Navier-Stokes equations in two dimensions and for the solution of the mass balance equations for species. They are related to the made discretisation choices but apart from that, they are not specific to this particular problem and they can be applied in general. For more in-depth descirption of the specific choices made in this case, refer to paragraph 3.2.

3.1.1 Projection algorithm

There are various ways to solve equation (6), but the adopted one was the **Projection algorithm**. This specific choice was made for a variety of reasons, the main ones being speed and stability, in addition to the discretisation method used. The algorithm works with a series of steps repeated at each time level for each cell of each grid. After assigning the initial and boundary conditions, the steps are the following:

1. the discretised version of (6) is split into two terms, that have the shape of (10) and (9). In particular, the first one contains the viscous ($\mathbf{A^n_{i,j}}$, cfr. equation (22)) and diffusive ($\mathbf{D^n_{i,j}}$, cfr. equation (23)) terms, while the second one is related to pressure;
2. a temporary velocity field is introduced ($\vec{u}'_{i,j}$) in order to solve separately the two terms. The advection-diffusion term is solved explicitelty like $\vec{u}'_{i,j} = \vec{u}^n_{i,j} + \Delta t(-\mathbf{A^n_{i,j}} + \mathbf{D^n_{i,j}})$[6]. This is possible because at time n, all quantities in the grid are known;
3. the pressure term cannot be determined explicitely, but instead it is calculated exploiting equation (5) with an iterative method (in particular, the Successive Over-Relaxation algorithm (4.3)). The Poisson equation for pressure is then $\nabla^2_h p_{i,j} = \frac{1}{\Delta t}\nabla_h \vec{u}'_{i,j}$, and its discretised form is found in section 3.2.4. Note that, since this is an iterative method, it requires the calculation of a residual error at each time level, and this choice is determined by the user. Tuning the error also refines results but increases computational time;
4. after finding the pressure gradient, the real velocity at the next time level is found by correcting the temporary velocity in the following way: $\vec{u}^{n+1}_{i,j} = \vec{u}'_{i,j} - \Delta t \nabla_h p_{i,j}$;
5. the time is advanced of a step Δt, and its dimension is chosen according to the stability analysis of the specific problem (cfr.4.1);

These steps are carried out for each point in the grid, and while the staggered grids used do not technically have the same indeces, by correctly initialising the fields it is possible to generalise the use of i and j for the center of each cell in every grid. The projection algorithm is useful to obtain results regarding the fluid-dynamic conditions inside of the reactor. In particular, these results can be interesting for the horizontal velocity u, because it represents well the symmetrical geometry ((Fig. 2) and (Fig. 3)). These fields are also coherent with the different inlet velocities and they show that the center pipe dominates the velocity field (the lower the velocity there, the lower it will be at the outlet).

After considering the solution of the Navier-Stokes equations, if the problem includes reactions, a specific solution needs to be described. In this specific case, the operator-splitting method has been chosen.

3.1.2 Operator-splitting method

When solving problems in which mass balances include a reaction term as well, technically the most precise method should be the one that couples reaction and fluid-dynamics. In most cases though, the fully-coupled method is not convenient (or completely unusable) because it requires too much computational memory. Hence, another algorithm needs to

[6] index i refers to the cells in the x direction, index j to y direction, while index n the current time step.

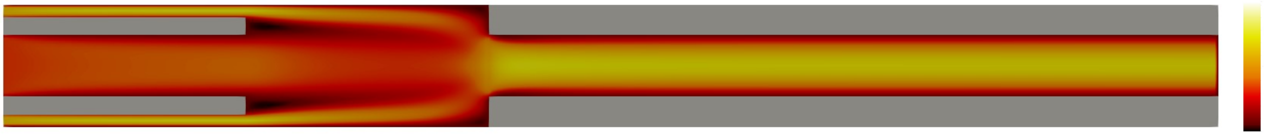

Fig. 2. $\vec{u}$ field in case 1

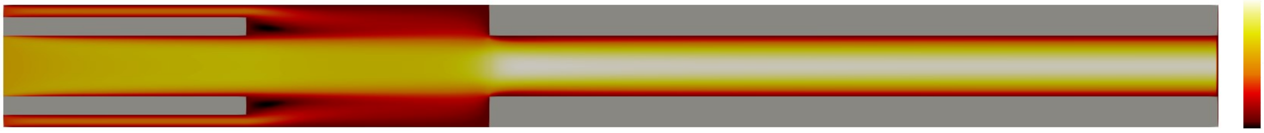

Fig. 3. $\vec{u}$ field in case 2

be applied, and in this case the **operator-splitting** method has been chosen.

This method consists in splitting the mass balance into two terms: the advection-diffusion one (10), and the reaction one (11). This can be done exploting a similar procedure as in 3.1.1, better described below:

1. the advection-diffusion equation is solved via a forward-euler method for time derivatives (17) and via the centered approximation for space derivatives ((18) and (20)). In its discretised form, it has the shape described in equation (3.2.3), and this explicit solution gives a field of temporary concentrations marked as C_j^*. This step is independent from the choices regarding the solution of the mass-balance term, as it is related intrinsically to the operator-splitting algorithm;
2. the rection step is also solved explicitely, starting from the temporary concentrations and updating them. This step is very simple numerically because the reaction term is a **local term**, meaning that its only dependance is from the concentrations in point (i, j) and not its neighbors. The choice of this solution is the most convenient one, because it is the one that ensures enough precision together with speed, as it will be better explained in section 3.2;
3. this procedure is iterated for each point in the grid at each time level. It is important to notice that the time step regarding the reaction term needs to be determined separately from the fluidodynamic one, because it can result in a significantly smaller number (cfr. 4.1).

This method does introduce some simplifications because the sole fact of splitting an equation into two is significant, but the results obtained can be considered sufficiently precise for a first understanding of the problem.

3.2 Chosen discretisation schemes

Strictly related to the numerical techniques adopted are the specific choices made for the discretisation of the equations illustrated in 2.1, because the numerical implementation of the solution and its speed directly depend on these. In this case, speed has been preferred over precision, because even if more simple, the choices made do not invalidate the results obtained.

3.2.1 Discretised boundary conditions

First of all, let us consider the boundary conditions in their discretised form, crucial for the correct solution of the problem. They have been written with respect to the x direction, considering the following definition of derivative:

$$\frac{df}{dx} = \frac{f(x+h) - f(x)}{h}$$

Considerig that the cells are h_x away from each other, equation (12) can be discretised as follows:

$$\frac{(f_{i+1,j} - f_{i,j})}{h_x} = 0 \tag{14}$$

Regarding equation (13), it needs to be said that it is applied at every wall to find the velocities in the ghost cells. Ghost cells are a concept used when solving the equations using the finite-volume method, and are better introduced in appendix A. In general, the continuous equation can be discretised in this way:

$$\frac{(f_{i+1,j} + f_{i-1,j})}{2} = \alpha \tag{15}$$

but the use made in this specific problem (in particular, $\alpha = u_{wall} = 0$) leads to an even simpler formulation that is also called **reflection technique**. In (16), the south wall is used as an example.

$$u_{i,1} = -u_{i,2} \tag{16}$$

The choice of using the finite-volume approach leads, in the case of the solution of Navier-Stokes equations, to choosing the **staggered grids** method as well. Here, the three grids used are for u, v and p, and, taking the pressure cells as the center, they are shifted respectively to the right and above of half a cell. This creates non-integer indeces that while might be more intuitive, are not supported when solving numerically. Hence, the indeces (i, j) have been chosen to mark the center of the current cell in the corresponding grid (for example, $u_{i+1/2,j}$ becomes $u_{i,j}$).

3.2.2 Discretised time and space derivatives

The time derivative has been discretised using the Forward Euler method, that is a simple explicit method but sufficiently precise. The time derivatives appearing in (6), (7) and (10) become then

$$\frac{\partial f}{\partial t} = \frac{f_{i,j}^{n+1} - f_{i,j}^{n}}{dt} \tag{17}$$

For space derivatives the time index has been omitted because they are all determined at time level n, and the centered approximation scheme has been used. Using the u velocity as an example, its derivatives give

$$\frac{\partial u}{\partial x} = \frac{u_{i+1/2,j} - u_{i-1/2,j}}{2h_x} \tag{18}$$

$$\frac{\partial u}{\partial y} = \frac{u_{i+1/2,j+1} - u_{i+1/2,j}}{2h_y} \tag{19}$$

In these equations, the fractional indeces are used because it is important to note that the values of $u_{i+1/2}$ and of $u_{i-1/2}$ are unknown, because the known values lay in the center of the pressure cell. These quantities are then interpolated using the known values and taking the arithmetic average. Second derivatives are generally discretised as in (20) and (21), using the centered scheme.

$$\frac{\partial^2 f}{\partial x^2} = \frac{f_{i+1,j} - 2f_{i,j} + f_{i-1,j}}{h_x^2} \tag{20}$$

$$\frac{\partial^2 f}{\partial y^2} = \frac{f_{i,j+1} - 2f_{i,j} + f_{i,j-1}}{h_y^2} \tag{21}$$

3.2.3 Discretised advection-diffusion

Now that the base components of the Advection-Diffusion equation have been discretised, it is easy to derive the whole equation in its discretised form as well. It is trivial how to go from the differential formulation to the integral formulation used in the finite-volume method (integral form of Navier-Stokes equation can be found in A, and by then exploiting the Gauss-Green theorem, equations (22) and (23) are obtained. The convective term of the equation (example using u and non-fractional indeces) using the centered approximation for first-oder derivatives is

$$\begin{aligned} A = \frac{1}{4}\frac{(u_{i+1,j} + u_{i,j})^2}{h_x} - \frac{1}{4}\frac{(u_{i,j} + u_{i-1,j})^2}{h_x} \\ + \frac{1}{4}\frac{(u_{i+1,j} + u_{i,j})(v_{i+1} + v_{i,j})}{h_y} \\ - \frac{1}{4}\frac{(u_{i,j} + u_{i,j-1})(v_{i+1,j-1} + v_{i,j-1})}{h_y} \end{aligned} \tag{22}$$

Since this problem does introduce relatively high horizontal velocities in the domain, one should consider applying an **upwind** discretisation scheme, which considers the outlet velocity of the cell to be equal to the one in the center ($u_{i+1,j} = u_{i,j}$). This method requires the calculation of fluxes at the borders of the cells, and it strongly depends on the Péclet number. For the horizontal velocity, it has the following shape:

$$Pe_u = \frac{|u_{i,j}|h_x}{\Gamma}$$

In this specific case, an **hybrid scheme** has been used, and the reasons behind this choice are better explained in the corresponding section (4.2).

The diffusion term of (10) is always discretised using the centered approximation scheme because it does not depend on velocities but on the fluid properties alone. Always using u as the reference velocity (noticing that the formulation for v is of identical nature and shape), (23) is obtained. [7] . These discretisations apply to the advection-diffusion equation for mass balance as well, where ϕ in (10) becomes the concentration field itself. In the mass balances though, only the centered scheme has been used because the variables to be determined are not the velocities themselves but concentrations, and so, using already-known u and v profiles, the changes in those have been already considered.

[7] at constant density, $\nu = \frac{\mu}{\rho}$

$$D = \frac{\nu}{h_x^2}(u_{i+1,j} - 2u_{i,j} + u_{i-1,j}) + \frac{\nu}{h_y^2}(u_{i,j+1} - 2u_{i,j} + u_{i-1,j}) \quad (23)$$

After determining these two terms separately, the first steps showed in 3.1.1 and in 3.1.2 are applied to produce temporary fields of velocities or concentrations.

3.2.4 Discretised Poisson

For the solution of the fluidodyinamic conditions, the last steps in the projection algorithm (Par. 3.1.1) require the solution of the Poisson equation. This equation needs to be solved iteratively, and it is done via the SOR method (cfr.4.3).

One may use this method because splitting the Navier-Stokes equations into two terms does not solve the issue of not having an explicit solution for the pressure term. In fact, the Poisson equation that will be later solved is derived from the temporary velocities exploiting the continuity equation (5). The equation to be solved is then (24), and of course it needs to be discretised as done for the previous ones.

$$\nabla^2 p_{i,j} = \frac{1}{\Delta t}\nabla \mathbf{\tilde{u}'_{i,j}} \quad (24)$$

The discretisation of these terms is very simple and it is obtained by following the same principles explained before. As it happened with the discretised Advection-Diffusion equation, it is convenient to derive separately the ∇p term and the $\vec{u}'$ term. The two give respectively (25) and (26), which are then combined to give an expression for $p_{i,j}$ in (27). [8]

$$\Delta = h_y^2(p_{i+1,j} + p_{i-1,j}) + h_x^2(p_{i,j+1} + p_{i,j-1}) \quad (25)$$

$$S = -\frac{h_x h_y^2}{\Delta t}(u'_{i,j} - u'_{i-1,j}) - \frac{h_x^2 h_y}{\Delta t}(v'_{i,j} - v'_{i,j-1}) \quad (26)$$

$$p_{i,j} = \frac{\beta}{\gamma_y h_y^2 + \gamma_x h_x^2}(\Delta + S) + (1-\beta)p_{i,j} \quad (27)$$

Notice that the coefficients γ_y and γ_x can be either equal to 1 or 2, and they are determined by the geometry of the problem: wherever the velocity is known (the inlet sections of every sub-domain and the walls) the coefficient corresponding to the known quantity is equal to 1. Everywhere else, it is equal to 2. This also means that there will be places where both the coefficients will be 1, for example at the corners of the domain, and places where they will be both equal to 2.

It is also important to note that the above expression is derived specifically for the Successive Over-Relaxation algorithm, and it needs to be solved by taking the error ε defined in (28), normalising it dividing by the total number of cells ($\varepsilon = \varepsilon/(n_x n_y)$) and iterating the procedure until ε becomes smaller than a set threshold (more details on the chosen maximum error in section 4). This procedure needs to be iterated for every cell in the domain before advancing in time.

$$\varepsilon = \varepsilon + |p_{i,j} - \frac{\beta}{\gamma_y h_y^2 + \gamma_x h_x^2}(\Delta + S)| \quad (28)$$

3.2.5 Discretised reaction term

When the fluidodyinamic conditions of the reactor are solved, the steps for the first part of the solution of the mass balances are exactly the same with a change of variables. After solving the Advection-Diffusion equation though, the source term treated with the Operator-Splitting algorithm is very easy. In particular, for this simulation it was decided to use an explicit method. After determining the reaction time step Δt_r (cfr.4.1), the equations (1), (2) and (3) have been used to solve the mass balances. Defining $\nu_{k,m}$ the stoichiometric coefficient of species k in reaction m, equation (29) is derived for every cell. This step is significantly quicker than the ones solved before because by using the C^* derived inside of the reactions, it is just an update of the variable.

$$C^k_{i,j} = C^{k*}_{i,j} + \Delta t_r(\sum_{m=1}^{NR} \nu_{k,m} r^m_{i,j}) \quad (29)$$

4 Details

Details and parameter-setting are a crucial part of the simulation because the wrong choices could bring to unphysical results or completely missing solutions.

[8] the parameter β is the over-relaxation parameter (cfr.4.3) and it has values $1 < \beta < 2$

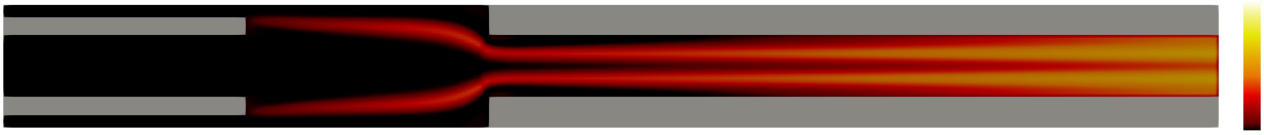

Fig. 4. field at steady-state of species E in case 1

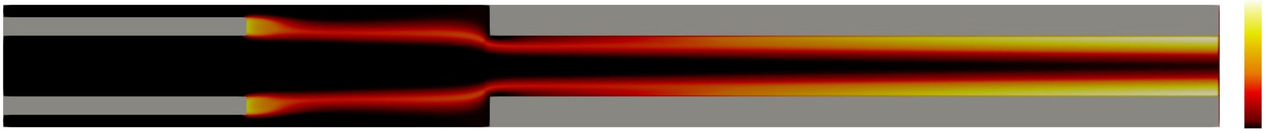

Fig. 5. field at steady-state of species E in case 2

4.1 Stability analysis and grid sensitivity

The first parameters to be checked and defined are the ones related to the time step. Since the equations need to be solved in a sufficiently precise way, the Δt and Δt_r need to be very small. How small is not only a matter of precision, but also of stability of the algorithms themselves. In principle, one could choose to adapt the time step at every iteration, but in this specific case a sufficiently small one was chosen, so that it could work well with all the conditions in the reactor.

The Δt used for the solution of the whole fluid-dynamic equations is a function of two adimensional numbers: the Courant number and the Diffusion number. These two are defined respectively as

$$Co = \frac{\vec{v}^2 \Delta t}{\Gamma}$$

$$Di = \frac{\Gamma \Delta t}{\Delta l^2}$$

where Δl is the characteristic distance (h_x or h_y). These two numbers need to be $Co \leq 4$ and $Di \leq 1/4$ and by exploiting these conditions, a correct Δt can be found. The threshold values change depending on the dimension of the problem (1D, 2D or 3D), but here only the 2D ones have been provided.

Since the inlet section of the reactor is characterised by some of the highest velocities (in the side pipes or in the central one depending on the case), it was chosen to take the minimum value for Δt between the ones calculated at the entrance. Because these velocities are not actually the highest in the whole configuration, a safety factor $\sigma = 0.4$ was introduced to make sure the time step was small enough.

From this, the number of steps necessary to carry out the simulation where set to be $n_{steps} = \tau/\Delta t$. The simulation time τ is in the order of magnitude of seconds, and in this case was set to approximately $4s$. By looking at the dimensions of the reactor and the velocities involved, it is clear that it is a long enough time to get to steady-state conditions. A larger simulation time would be completely unnecessary in terms of results and it would simply cause the processing to take longer. Clearly, by changing the inlet variables, it could be important to check if the provided value is still enough for reaching steady-state.

After performing this quick stability analysis, one should take care of the grid dimensions. In this case, we chose to perform the simulation with $n_x = 600$ and $n_y = 60$ points along the two directions. The results are obtained quickly and with precision, it was also checked that the equations were not sensitive to the grid dimensions (by using the hybrid discretisation scheme). A finer grid could favor precision but it would take longer to process, whereas a less fine grid would definitely give unstable and approximated results. It was found that these dimensions give the best ratio between quickness and precision, also because dividing the domain into five smaller ones meant sacrificing some speed (functions were called five times each and boundary conditions multiplied). For this same reason, some constraints were set on the number of cells in every direction, so to have an integer number of cells in each sub-domain. It important to note that a finer grid might also require a smaller safety factor when determining the correct Δt for the algorithm to be stable (possibly $\sigma = 0.3$ or even smaller). This, in addition to the higher number of cells, could bring to too long of a computational time.

4.2 Hybrid discretisation scheme

This discretisation scheme was introduced because the high velocities in the outlet pipes caused significant problems in the solution of the Advection-Diffusion equations for velocities and species. By using a barrier of $Pe = 2$, it was possible to solve the equation differently depending on the speed of the fluid in each specific cell, both horizontally and vertically. If $Pe \geq 2$, an upwind discretisation scheme has been chosen, where velocities in the next cell are set to be equal to the ones in the previous cell. If not, a standard centered differencing scheme was applied, allowing for better and more stable solution of the equation. This hybrid configuration was particularly useful in the pipes where horizontal velocities had a narrow parabolic shape, changing greatly along the y direction.

4.3 SOR parameters

The Successive Over-Relaxation algorithm used for solving the Poisson equation (3.2.4) includes two important parameters to be chosen with caution: β and the maximum error ε_{max}. In particular, the smaller the β coefficient is, the smaller the impact of the over-relaxation will be on the solution (and so the slower the algorithm will go). The same considerations apply for the maximum allowed error, that ensures higher accuracy but slower processing. In Table 1 are the specific parameters for each case.

Table 1
SOR parameters

	β	ε_{max}
case 1	1.7	1e-6
case 2	1.9	1e-6

This algorithm for solving the Poisson equation for pressure is very effective in accelerating the computation, because it takes the residue ε as in equation 28 and uses inside of it the previously calculated values of pressure at the current time step. By applying this concept inside of the residue and of the calculation of $p_{i,j}$ (cfr.(27)), convergence is reached quicker. Moreover, the addition of the β coefficient speeds up the calculations even more, making this method particularly useful for such an intensive use.

5 Results and comments

The aim of these simulations was to compare two different models for feeding in the same reactor. Because of how the reactions occur (cfr. 1.1), the production of species E starts in the mixing chamber (Fig. 4 and Fig. 5). The two simulations were carried out in series and the profiles obtained for each species in both cases can be found in appendix (B), whereas below the results will be commented.

5.1 Profiles of desired species E

The profiles obtained for species E differ in a significant way depending on the specific case, as seen from figures (6) and (7). These two profiles strongly depend on the amount of species B and D in the reactor and on the velocities involved. The results obtained do not differ by a significant amount but, since the outlet concentrations of species E are already low, a small change in them could result in a much different productivity over a long period of time.

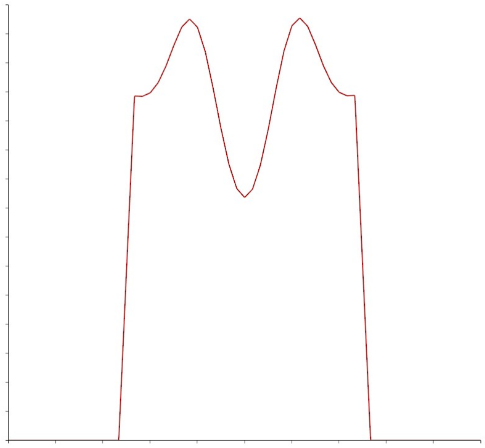

Fig. 6. y-axis outlet profile of species E, case 1

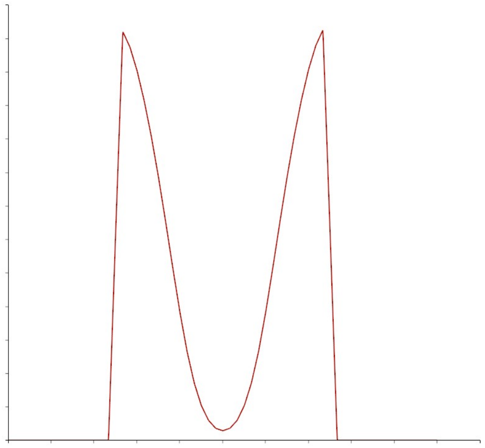

Fig. 7. y-axis oultet profile of species E, case 2

Table 2
Results for species E

	C_{max}	C_{mean}	$\dot{F}_E$	n_E
case 1	0.0151	0.00623	0.00516	18.64
case 2	0.0183	0.00488	0.00408	14.68

The first case presents a lower maximum value for the concentration of the species compared to the peak in the second case, but while this may indicate an higher production in case 2, case 1 maintains a greater mean value for the concentration[9]. Thus, it is important to refer to fluxes of species E rather than comparing concentrations only, because those could be misleading. Since the geometry of the reactor is the same

[9] all in S.I. units $[mol/m^3]$

in both cases, the analysed fluxes will be expressed in $mol/m^2/s$ by multiplying the outlet concentration vector $\vec{C}_E$ by the horizontal velocity field $\vec{u}$ at the outlet section as well. To have a decent approximation of the quantities involved, Table 2 presents the mean value of the flux at the outlet section, and the quantity of species E after $1h$ of operations is displayed as well (in mol/m^2), to better give a snapshot of the difference between the two operating conditions.

The results are obtained applying the operating conditions described in 4.1, and changing the parameters could give errors in the order of magnitude of 10^{-4}. It is important to note that on a numerical level, from simulation to simulation the results could give slight variations in the graphs results as well, but the ones provided in the table are a good first-approximation of the numerical values.

The correctness of the procedures illustrated so far can be confirmed with two checks one can easily perform: firstly, all the graphs obtained (Appendix B) are symmetrical with respect to the x axis, and as Fig.1 suggests this is coherent with the reactor shape. Moreover, since all the reactions occur without global change of moles (expansion factor $\varepsilon_f = 0$), the total concentration inside the reactor should be constant, and this happens as well in both cases. In Fig.B.1, case 1 has been used as an example because the two are identical in this sense.

5.2 Conclusion

Concluding, the simulations highlight that the difference in produced moles after $1h$ is sufficiently significant, and this, depending on whether the objective is to reduce or increase the productivity of species E, justifies leaning towards one or the other configuration. If one would like to maximise the amount of E, configuration 1 should be chosen. On the contrary, if E is undesired, case 2 should be preferred.

Appendix A Finite-volume discretisation

The choice of using the finite-volume method comes from the fact that is similar to the finite-difference one (in the sense that is a simple method) but it provides higher accuracy when solving the problem and it also allowed the use of the staggered-grid method and consequently the Projection algorithm for the Navier-Stokes equations.

About this method of considering the physical volume, it should be said that now the interesting quantities lay in the middle of a cell that has four faces surrounding it (in a two-dimensional problem): the east and west faces, and the north and south faces. This means that the first point in the grid lays half-a-cell inside, and its value cannot be determined by simply using a boundary condition. Instead, the concept of **ghost cells** should be introduced.

Ghost cells are additional cells put at the end of the domain, making the final grid composed of $n_x + 2$ and $n_y + 2$ cells. For how staggered grids are applied then, the pressure and concentrations grids have these dimensions, while the u grid has $n_x + 1$ and $n_y + 2$ cells, and the v grid has $n_x + 2$ and $n_y + 1$ cells. The boundary conditions (cfr.15, 14) are then used to derive the values of quantities in these cells, which are later used in the solution of the internal cells.

Also note that the use of this method requires writing the fundamental equations in their volume-integral form, and then applying the Gauss-Green theorem to transform them into surface integrals. As an example, below is the conservation of momentum along the x direction in its integral formulation (after applying the Gauss-Green theorem):

$$\begin{aligned}\frac{\partial}{\partial t}\int_V u dV = &-\oint_S (u\vec{u}\cdot\vec{n})dS \\ &-\frac{1}{\rho}\oint_S (pn_x)dS + \nu\oint_S(\nabla u\cdot\vec{n})dS\end{aligned}$$

This is a very brief introduction to the method, but these concepts are the most important ones for understanding the all the algorithms described in the report.

Appendix B Figures

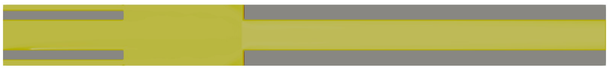

Fig. B.1. total concentration profile

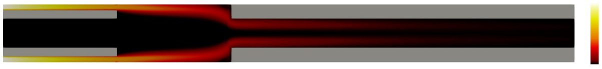

Fig. B.2. field at steady-state of species A in case 1

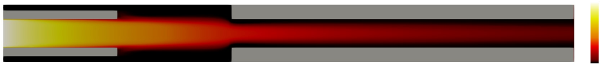

Fig. B.3. field at steady-state of species A in case 2

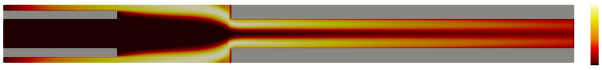

Fig. B.4. field at steady-state of species B in case 1

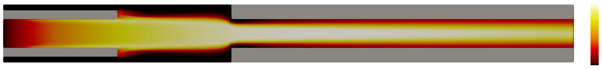

Fig. B.5. field at steady-state of species B in case 2

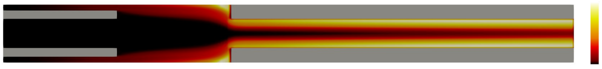

Fig. B.6. field at steady-state of species C in case 1

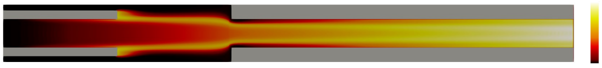

Fig. B.7. field at steady-state of species C in case 2

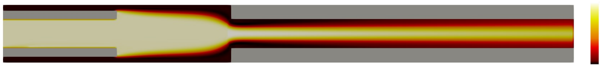

Fig. B.8. field at steady-state of species D in case 1

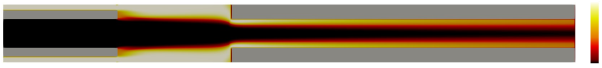

Fig. B.9. field at steady-state of species D in case 2