

Comparison of a fully coupled and a decoupled solver for the simulation of Fluid Catalytic Cracking

Paper number 341c

Edward Baudrez,¹ Juray De Wilde,² Geraldine Heynderickx^{1,*} and Guy B. Marin¹

1. *Chemische Proceskunde en Technische Chemie, Universiteit Gent, Laboratorium voor Petrochemische Techniek, Krijgslaan 281, S5, 9000 Gent, Belgium*

2. *Materials and Process Engineering Department, Université Catholique de Louvain, Bâtiment Réaumur, Place Sainte Barbe 2, 1348 Louvain-la-Neuve, Belgium*

* *Corresponding author, Geraldine.Heynderickx@UGent.be*

Background

The ongoing research effort presented here aims at finding the most time-effective solver for reactive flow problems with strong volumetric expansion, and strong feedback effects of the reactions on the flow. This solver has to perform steady-state simulations of important chemical processes involving multiphase flow and detailed kinetics with minimal total CPU time requirements. This total CPU time requirement is the product of CPU time per iteration and the number of iterations required to obtain convergence.

The focus is on the Fluid Catalytic Cracking process (Das et al. 2003, Quintana-Solórzano et al. 2005), as a benchmark to assess the performance of the steady-state solvers that are being compared. Fluid Catalytic Cracking (or FCC) is a major conversion process in the petrochemical industry, used extensively to convert heavy crude oil fractions to lighter, commercially more valuable, transport fuels. It is a prime example of a multiphase process, involving three aggregation states. It is characterized by high feed concentrations and important density variations, a feature which sets it apart from combustion, where the heat effect, rather than the volumetric expansion effect of the reactions is of major concern for the numerical scheme.

The FCC process was selected for several reasons. It is a numerically challenging process. The density variations induced by the reactions are found to have an important impact on the robustness of the numerical scheme. Furthermore, the numerical behaviour when solving this process is sufficiently different from problems in combustion, an area of active research, to warrant separate attention.

Finding the least time-consuming way to solve the equations of reactive flow is the long-term target of many researchers. In this presentation, the results obtained using two different solvers will be discussed. Both solvers have been implemented in a computer program solving the one-dimensional equations of reactive gas–solid flow, for ease of development. However, the concept and the results are readily generalized to higher dimensions, and the use of one space dimension is not an intrinsic limitation of the solvers presented here. The first solver is a fully coupled solver in line with traditional Navier-Stokes solvers for compressible, nonreactive flow. The second solver is a generalization of the post-processing approach, frequently used for the *a posteriori* imposition of chemical reactions on a previously determined flow field.

Fully coupled solver

In the fully coupled solver, the equations for reactive flow are all treated alike. This means that the Navier-Stokes equations and the continuity equations are discretized and integrated

simultaneously. After discretization of the original partial differential equations, a set of coupled linear algebraic equations is obtained. Each of these algebraic equations expresses the changes in the state variables due to convection, diffusion, reaction and other source terms. Because the state variables are linked to one another, a matrix equation has to be solved to obtain the changes for the individual state variables. Obviously there are as many equations as there are state variables. Hence, the coefficient matrix has size $n \times n$, n representing the number of state variables. It is to be noted that the computational work for solving a matrix equation with an $n \times n$ matrix is *at least* of the order n^2 . This implies that the computational work is more than proportional to the number of equations. The latter fact will grow in importance as the number of reactive components increases.

Simulations of the FCC process with the fully coupled solver indicate that it is a stable and robust solver. By using an implicit treatment of the source terms, the solver copes well with the numerical stiffness caused by the chemical reactions. The number of iterations to reach convergence is nearly identical to that of a solver for nonreactive flow.

The main advantage of the fully coupled solver is that it is a logical extension of a Navier-Stokes equation solver as the additional continuity equations are discretized and integrated in exactly the same way as the Navier-Stokes equations. The solver copes well with the density variations. Its most serious disadvantage is that, even for a moderate number of reactive components, the coefficient matrix quickly becomes large. Since the computational work involved in solving the matrix equation is at least proportional to n^2 , it follows that any reduction in the size of this matrix is rewarded with a more-than-proportional decrease in execution time. This is the major motivation for the algebraically decoupled solver discussed next.

Algebraically decoupled solver

The algebraically decoupled solver differs from the fully coupled solver, in that the Navier-Stokes equations and the continuity equations are decoupled at the algebraic level. Consider the coefficient matrix of the fully coupled solver. If certain elements of the coefficient matrix are set to zero, the equations decouple algebraically. The matrix equation with the modified coefficient matrix can be split in two matrix equations, each with a smaller coefficient matrix. Evidently, the total number of algebraic equations remains the same. However, the computational *work* involved is reduced significantly. Please note that cancelling elements is only allowed inside the coefficient matrix governing the time derivative term, which vanishes at steady state. It is never applied to the other terms, to avoid changing the steady state itself. In other words, only the convergence behaviour may be modified, but the consistency of the differential equations and the numerical implementation should always be satisfied.

The algebraically decoupled solver is a generalization of the post-processing approach which is often used for the 'superposition' of chemical reactions on a previously determined flow field. Often, a flow field in a reactor is determined using a CFD package, excluding chemical reactions, producing density, pressure, temperature and velocity fields. The latter are then used as an input to another package that calculates the chemical reactions. Provided the reactions do not influence the flow field significantly, this chemical post-processing is the last stage of a reactive flow simulation. The generalization consists in transferring control from the flow part to the reaction part and vice versa, not once, but several times. The post-processing approach is a limiting case of this, consisting of one pass for the Navier-Stokes equations, followed by one pass for the continuity equations.

Simulations of the FCC process with the algebraically decoupled solver are not found

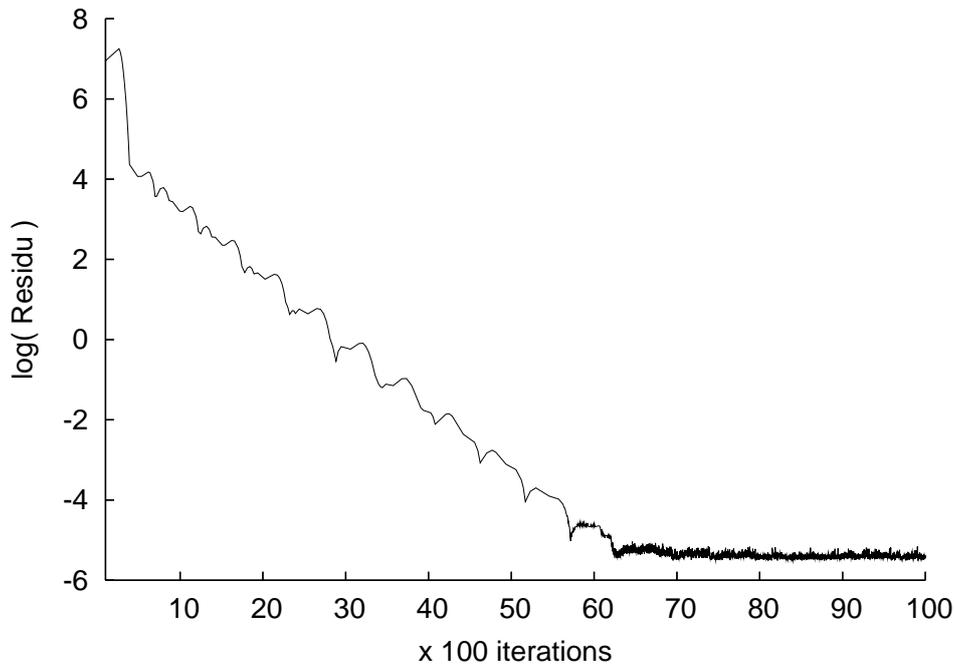


Figure 1: Convergence history of the coupled solver. Steady state reached for the overall flow+reaction problem.

to be successful. As discussed below, many more iterations are needed to reach convergence than the fully coupled solver needs. The experience gained with the fully coupled solver is now of value to understand the cause. Closer investigation of the elements of the coefficient matrix which are assumed to be zero in the algebraically decoupled solver, reveals that some of these elements in fact do not have negligible values. It is exactly this which causes the limited performance of the algebraically decoupled solver for the FCC process. However, the values of these elements can be quantified, and linked to operating conditions and/or physical properties. Hence, it is possible to derive an objective criterion which can be calculated *a priori*, to determine if the application of the algebraically decoupled solver can be successful.

The advantage of the algebraically decoupled solver is that it is relatively straightforward from a conceptual and technical point of view, compared to other methods of decoupling. Since the continuity equations are of the same nature as the Navier-Stokes equations, *viz.* of convection–diffusion–source type, the discretization and integration need not be changed. Furthermore, the same boundary conditions as in the fully coupled solver can be applied. A disadvantage is that the algebraically decoupled solver loses its competitive edge over the fully coupled solver if the number of reactive components is many times larger than the number of flow equations. Another disadvantage is that this decoupled solver is not unconditionally stable.

CPU time & convergence results

The final aim is the reduction of the total CPU time needed for a simulation. The total CPU time is determined almost exclusively by the computational work of the linear algebra routine. Since the computational load of the linear algebra routine is (at least) of the order n^2 , the theoretical gain in CPU time per iteration by solving two systems of half size instead of one full system, is therefore: $2 \times (n/2)^2 = n^2/2$, or 50%. In practice, when the coupled solver is run for n iterations, and the decoupled solver is run for n flow iterations and n reaction iterations, the decoupled solver is only 10% faster than the coupled solver. This rather low figure can

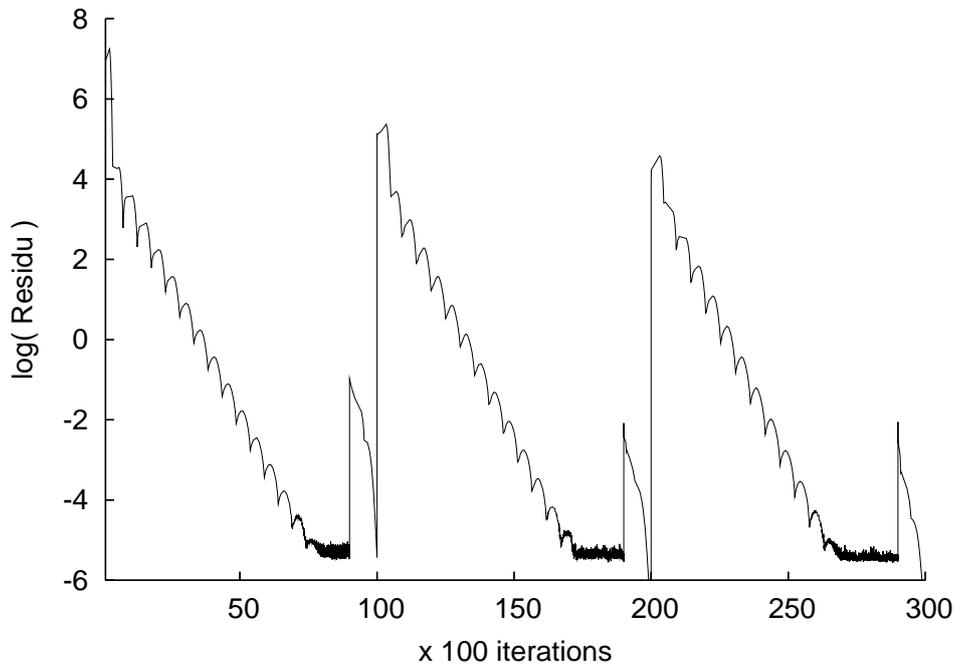


Figure 2: Convergence history of the algebraically decoupled solver, with six cycles: one flow cycle (long) alternating with one reaction cycle (short), repeated three times. Steady state reached in every of the six cycles. Note that steady state for the overall flow+reaction problem is not achieved, even after six cycles.

be explained partly by the increased overhead of the decoupled solver for managing the solution of two different equation blocks. More important though, is the observation that the reaction equations converge much faster than the flow equations; typically 10 times faster. Therefore the coupled solver should be compared with a decoupled solver running only $n + n/10$ iterations, resulting in a speed-up of more than 60% using the decoupled solver. The potential for speed-up is even higher considering that the reaction equations were solved with a CFL number of only .2, due to instabilities linked to changes in the density field. The flow equations were solved with a CFL number of 1. Note that the CFL numbers used are still orders of magnitude larger than the CFL number of a solver with only one single iteration per flow cycle and per reaction cycle (Baudrez et al., 2005).

However, the total CPU time is also influenced by the number of iterations needed for convergence. A plot of the residu as a function of the number of iterations is plotted for the coupled solver in Figure 1, and for the algebraically decoupled solver in Figure 2. The coupled solver has only one cycle, during which the combined flow–reaction problem is solved until a steady-state solution is obtained (which is detected by checking whether the norm of the residu has reached a sufficiently low value). The algebraically decoupled solver performs several alternating flow/reaction cycles. Only six cycles, alternating between flow and reaction, are presented. It is seen that the first cycle (flow equations without reaction equations) converges almost as fast as the coupled solver. The presence or absence of reaction source terms does not have a significant influence on the convergence rate. When a steady-state solution is reached for the flow equations, the reaction equations are solved, using the converged flow fields as an initial guess. It is seen that the reaction equations indeed converge very fast to a steady-state solution.

The important point to note is that, when the composition obtained from the steady-state solution of the reaction equations, is substituted back into the flow equations, the state

does no longer satisfy the flow equations. This is visible as a peak, i.e. a high initial residu of the flow equations in the third cycle. Overall, the maximum values of these peaks is seen to decrease, although only slightly over the six cycles presented in Figure 2.

The explanation is that the density of the mixture is very sensitive to the composition. During the flow cycle, the composition is assumed to be constant in time, and the flow field is determined using a fixed composition. This composition is only updated in the reaction equations. When control is returned to the flow cycle, density is updated with the new composition. However, the density is very important for the convective fluxes, and the flow field does not satisfy the conservation equations any more. Convergence needs to be re-established using this new density field. Conversely, when a steady-state solution is obtained for the flow field, and control is transfered to the reaction block, the fluxes change as well, and convergence needs to be re-established for the composition. This explains the high peaks in the convergence history, every time a switch from flow to reaction cycle, or reaction to flow cycle, takes place. The height of the peaks is more pronounced if the changes caused by the flow or the reaction block to the density field are larger. In FCC, physical properties and operating conditions are such that this is the case, and many more flow and reaction cycles than in the coupled solver are needed to obtain convergence for flow and reaction simultaneously.

Conclusions

In this work, a fully coupled and an algebraically decoupled solver for reactive two-phase flow with strong volumetric expansion have been presented. The fully coupled solver is stable and robust. Its convergence is not deteriorated by the volumetric expansion effect due to the cracking reactions. However, although the number of iterations is low, it is expensive in terms of the computational cost per iteration. The algebraically decoupled solver, although the computational cost per iteration is much lower, needs many more iterations. The major reason for this is the two-way coupling between flow and reactions through the density. This coupling can be 'visualized' and quantified, and linked to physical properties and/or operating conditions, quantities which are known a priori. The post-processing approach is a limiting case of this algebraically decoupled solver. Hence, this conclusion is more generally useful for assessing the feasibility of chemical post-processing on given flow fields, where the effect of composition on the density is important.

References

- Baudrez E., De Wilde J., Vierendeels J., Heynderickx G. and Marin G.B. (2005). Full versus partial coupling of flow and reaction in the simulation of catalytic cracking riser flow. In *Proceedings of the 7th World Congress of Chemical Engineering*. Institution of Chemical Engineers
- Das A.K., Baudrez E., Marin G.B. and Heynderickx G.J. (2003). Three-dimensional simulation of a fluid catalytic cracking riser reactor. *Industrial and Engineering Chemistry Research*, 42(12):2602–2617
- Quintana-Solórzano R., Thybaut J.W., Marin G.B., Lødeng R. and Holmen A. (2005). Single-Event MicroKinetics for coke formation in catalytic cracking. *Catalysis Today*, 107–108:619–629