TRW’s multidisciplinary engineering by using the coupling interface MpCCI: coupling between CFX-5 and MSC.Marc

Ch. Jung\(^1\) & M. Kuntz\(^2\)
\(^1\) TRW Airbag Systems GmbH, Aschau am Inn, Germany
\(^2\) AEA Technology GmbH, Otterfing, Germany

Abstract

Numerical simulation of physical phenomena is now a commonly used tool in industry and research during development of new products and services. Most of the phenomena are related to a single discipline, however, our world is multidisciplinary. The simulation of these complex phenomena requires software capable of dealing with it. One solution is the connection of two or more software codes by a coupling library. As a result the development of new codes for multidisciplinary problems can be avoided and the reuse of established and validated software is possible. This paper describes the general concept of the coupling between CFX-5 and MSC.Marc via MpCCI. Also, the application interface concept is explained and the two communication concepts provided by MpCCI are discussed. The capabilities of this technology will be demonstrated by real applications at TRW.

1 Introduction

One strategy used to simulate fluid structure interaction is to employ a single solution strategy by solving all governing equations with either a finite volume or a finite element method. The advantage of this procedure is a uniform data structure and a single solution method, which allows for a strong coupling of the problem and avoids complex data transfer [1]. A disadvantage of such approaches is that one has to reformulate and re-implement all the physics within a new code. Another promising simulation strategy is to solve each part with existing high level (commercial) software packages and to couple both processes by a data interface. The advantage of using existing software packages is that one can
benefit from their advanced features, in particular the governing equations can be solved efficiently using complex numerical sub-models.

The explicit coupling between two codes was intensively investigated at the Institute for Algorithms and Scientific Computing (SCAI) [2] in the projects CISPAR and GRISSLi. CISPAR and GRISSLi differed with regard to the application areas, concepts and the main foci [2]. The main topic in CISPAR was the development of the library COCOLIB for coupling commercial codes, whereas GRISSLi was focused on numerical topics (e.g. suitable numerical coupling strategies, appropriate interpolation of data between the coupled codes, design of a coupling interface). The EU-funded project CISPAR was finalised by SCAI in 1999 and the coupling interface GRISSLi (funded by the German Federal Ministry for Research and Education BMB+F) was finalised in 1998. Based upon the results of these two projects a new coupling library called MpCCI (Meshed based parallel Code Coupling Interface) was specified and developed by SCAI. This work was also funded by the EU in a project called DEBUT (Multi-Displinary Engineering By Using Coupling Technology). The main goal of DEBUT was to solve TRW applications by coupling CFX-5 and MSC.Marc and thereby demonstrate the coupling technology of MpCCI to the engineering market. DEBUT was finalised in 2002. This paper gives an overview on the corresponding results from the view of an industrial company.

2 MpCCI – basics and concepts

Figure 1: Architecture of a MpCCI-application. The codes can run in parallel modes: here code A uses two and code B uses three sub-processes.

The software library MpCCI allows the exchange of grid specific data among grid based data structures (structured and unstructured grids) whereas the interface enables the exchange of data among several simultaneously running calculation systems. MpCCI is based on the MPI-standard (Message-Passing-Interface). Figure 1 illustrates the architecture of an application using MpCCI. Two codes (A and B) are coupled, whereby each can run in serial or parallel mode. Additionally a coupling process handles the variable definition of the selected quantities at the interface. In general the interface grids of both code have a non-coincident nodes distribution and different topology. MpCCI provides all the functionality which is needed to couple the codes, i.e. communication handling via MPI, neighbourhood search and interpolation capabilities. The connection of each code process with the coupling library is called Application Interface (API). In general it includes
two parts, the set-up phase and the data exchange part. During the set-up the interface grids are defined by each code and the grid data is transferred to the coupling library. Based on this data the relation of both interface grids (connectivity between the nodes) is computed. During the data exchange phase variables are transferred between the codes. The definition of a variable is code specific (e.g. interface element nodes of element centres). For a code running parallelized by domain decomposition, the set-up phase and data exchange phase are carried out by each process being involved in the coupling interface. The coupling algorithm can be an alternating strategy (code A is doing a time step or inner iteration while code B is waiting; after receiving data from code A code B is doing a time step or inner iteration and code A is waiting) or a synchronous strategy (both code A and code B are doing an inner iteration or time step and then exchanging data).

![Diagram](image)

Figure 2: The coupling behaviour at a synchronisation point SyPt.

The main advantage of using APIs to couple codes for specific applications is that the original code and the coupling parts are strictly separated. Each modification of the coupling routines, the coupling algorithms or the coupling quantities does not affect the code itself. In the years 1998/99 TRW has successfully realised some projects using this method of programming the coupling via user-subroutines within MSC.Marc, CFX-4 and the library GRSISSLi [3]. However, there are some disadvantages to this method. The handling of user-subroutines is often troublesome, especially if the problem type or an application class changes. Furthermore, one is inflexible when a more flexible coupling strategy is required. In order to reach a more general coupling, the interaction between the numerical algorithm of the code and the coupling algorithm has to be more intensive. This goal was reached within DEBUT fulfilling the following requirements:

(a) Integration of MpCCI without knowledge of the other codes to couple to.
(b) To couple two codes with integrated MpCCI no further code changes should be necessary.
(c) The user should be able to define the quantities to be exchanged.
(d) The user should be able to define or to modify the coupling algorithm.

To fit these requirements, well defined communication ports were required. In both coupled codes so called “coupling ports” are defined. These ports are located at certain positions in the codes, where a data exchange is reasonable (e.g. at the
beginning or the end of a time step during a transient run). The coupling ports of both codes are connected by the synchronisation point concept of MpCCI, see Figure 2. Each synchronisation point defines a bi-directional communication, for which data can be sent or received by the codes. The user will then connect the synchronisation points of the codes to form the coupling algorithm and choose which quantities should be transferred. Of course this puts more responsibility into the hands of the user but also gives him more flexibility.

3 Coupling CFX-5 and MSC.Marc using MpCCI

3.1 CFX-5 and MSC.Marc

CFX-5 is a leading general purpose CFD code that is well known for its robust and accurate numerical methods [4]. The governing equations for turbulent flow are the Reynolds-Averaged Navier-Stokes (RANS) equations. These equations are discretised using a finite volume method, which is conservative and time-implicit [4]. The equations are evaluated by second order bounded high resolution advection scheme. The computational unstructured mesh can consist of different element types such as hexahedrals, prisms, wedges, or tetrahedrals. The mass flow is evaluated such that a pressure-velocity coupling is achieved by the Rhie and Chow algorithm [6]. The Reynolds stresses in the momentum equations are computed by using the SST two-equation turbulence model [5] and an automatic wall treatment. The discrete systems of equations are solved by a coupled algebraic multi-grid method [4]. The numerical effort of this solver scales linearly with the number of grid nodes. The CFX-5 solver runs in parallel by using the domain decomposition concept based on the Single Program Multiple Data (SPMD) algorithm [7]. Concerning MpCCI the solution variables are easily imported / exported via a data interface [8].

MSC.Marc is a general purpose non-linear finite element solver for structural (i.e. mechanical and heat transfer) and non-structural (i.e. electrostatic, magnetostatic and acoustical) steady state or transient analysis. MSC.Marc includes automated 3D contact, automated mesh enrichment, the parallel processing algorithm utilises the Domain Decomposition Method for performance scalability on multi processor computers. Over 150 element types are available to represent any geometry in 2D and 3D. The list of predefined material models includes metallic materials undergoing elastic and plastic deformations, composite and rubber materials. This can be extended by user-defined models. Large deformation behaviour can be represented using total and updated Lagrange formulations.

3.2 Implementation of coupling ports

Five synchronisation points have been implemented in CFX-5 and MSC.Marc in a similar way, see Figure 3. The first point SyPt(1) is reached at the start of the analysis. It can be used to exchange initial data, for example at a restart. The synchronisation points SyPt(2) and SyPt(3) coincide with the start and end of an increment (time step). Using these two points, a weak coupling algorithm can be set up in which the two codes communicate only once per time step. No convergence check on the data exchanged is performed in that case.
Synchronisation points SyPt(4) and SyPt(5) are located just before and after the non-linear iteration loop. They can be used to define a strong coupling per time step. If SyPt(5) is enabled, the iteration loop is repeated a number of times until the interface variable \( \phi \) that has been sent at SyPt(5) is converged. Convergence is achieved if the convergence ratio, \( \varepsilon = \frac{\text{RMS}(\phi^{\text{new}} - \phi^{\text{old}})}{\text{RMS}(\phi^{\text{new}})} \) with RMS= root mean square, is smaller than a given user defined tolerance. At the beginning of a new increment (time step), \( \phi^{\text{old}} \) is always set to zero. Hence, in the first iteration the convergence ratio is always one. Furthermore, the data to be exchanged via MpCCI at time step \( n \) can be under-relaxed with the interface values of the last coupling step \( n-1 \). This procedure increases the stability of the coupled computation. The relaxation factor \( R \) is defined per exchange quantity \( \phi \), \( \phi^{(n)} = R \phi^{(n)} + (1-R) \phi^{(n-1)} \). \( \varepsilon \) and \( R \) can be easily defined in the MpCCI input file. It is not necessary to transfer data at each time step.

![Diagram of analysis flow](image)

Figure 3: Left: implementation of MpCCI in CFX-5 and MSC.Marc. Right: quantities which can be sent (S) or received (R) in MSC.Marc | CFX-5.

The user is able to connect the synchronisation points of the codes in order to form the coupling algorithm and to choose which quantities should be transferred. The quantities that can be transferred depend on the implementation of MpCCI in the corresponding code. For the implementation in CFX-5 and MSC.Marc the right part of Figure 3 gives an overview of what can be sent / received at the synchronisation points.

Obviously, the exchange of data between different codes should be in a common set of units. Within the framework of the DEBUT project, data exchange was in SI units. In case where the MSC.Marc model is created using different units, then a scale factor for the geometry and a shift value for the temperature are available in the MpCCI input file to allow conversion to SI units.
Two or more (commercial) codes which support the described concept of coupling ports via MpCCI can be coupled (up to now: CFX-5, MSC.Marc, Star-CD, PERMAS, ANSYS; planned in 2003: MSC.Nastran and Abaqus). Independent of that it is naturally always possible to couple codes via a self written API for MpCCI as described in chapter 2. MpCCI is supported by Pallas GmbH [www.pallas.de]. For current and more detailed information about MpCCI see [www.mpcci.org]. There is also an annual user meeting.

3.3 Coupling performance

The performance was analysed for a thermally coupled system with approximately 900000 elements for both fluid and solid. Both parts were computed with CFX-5, coupled to MpCCI and exchanging temperature and heat flux density at the beginning of each time step [SyPt(2)]. The codes run simultaneously. The parallel speed-up factor was computed by comparing the wall clock time consumption for a coupled analysis on one processor with the wall clock time required using N processors for fluid \(N_{\text{fluid}}\) and solid part \(N_{\text{solid}}\). Figure 4 demonstrates, that for a given total number of processors different speed-up values are obtained depending on the assignment to fluid or solid solver. The optimal distribution of processors is strongly dependent on the relation of the computational effort of both coupled parts.

![Figure 4: Speed up as function of the number N of processors. The analysis is carried out on a hpcLine Linux-Cluster](image)

4 Applications

4.1 Heat loss in a tank (thermal coupling)

The aim of a tank test is to carry out an inflator performance check by measuring the tank pressure. The inflator is to be installed pressure tightly into a 60 litres tank so that the result from the burn-up will be collected. The heating up of the tank gas by the inflator deployment is simulated by CFX-5 and the heating up of the tank wall is calculated by MSC.Marc. We implemented the following coupling strategy (exchange at each time step \(dt=1e-5s\)):
Fluid Structure Interaction II

MSC.Marc at SyPt(2): receive heat flux
MSC.Marc at SyPt(3): send temperature
CFX-5 at SyPt(3): send heat flux, receive temperature

The synchronisation points SyPt(3) of CFX-5 and SyPt(3) of MSC.Marc as well as the points SyPt(3) of CFX-5 and SyPt(2) of MSC.Marc have to be matched resulting in a weak coupling.

The mesh in CFX-5 consisted of 35500 tetrahedral cells whereby the mesh using in MSC.Marc was a hexahedral mesh (12800). Because of the symmetry of the tank and the inflator only a sector of 15° needed to be considered. The output of the inflator was modelled with an inlet. The corresponding boundary conditions of this inlet were mass flow and total temperature. The deployment of the inflator took about 50ms with a maximal mass flow of 1.2 kg/s at 8ms and a nearly constant total temperature of 1100 K. For the tank walls the material specification of steel was used and no radiation was taken into account.

Figure 5(a) shows stream lines calculated by CFX-5 at 8ms. In Figure 5(b) the spatial temperature field on the tank walls calculated by MSC.Marc at 70ms has a maximum value of about 30°C (white colour). Due to the dependency of the heat transfer between gas and solid on the local gas velocity, the highest temperature values on the wall are near the highest gas velocities and the spatial temperature distribution is obviously linked to the stream lines. In Figure 6 the measured pressure in the tank volume is plotted in comparison to the calculated mean values with and without heat loss. The simulation with heat loss matches the experimental values much better.

Figure 5: (a) velocity stream lines at 8ms calculated by CFX-5 (dark colour = high velocity) (b) temperature of the tank walls calculated by MSC.Marc at 70ms. The highest temperature (white) is about 30°C, the lowest 23°C (dark).

Figure 6: Measured pressure values in comparison with the calculated values with and without heat loss, $P_{ref} = 0.1\text{MPa}$.
4.2 Hybrid inflator SHI-40/25

Two design approaches have been widely used for vehicle airbag inflators. One design approach involves the use of a pyrotechnic charge or propellant grain mounted in a combustion chamber. Upon deployment, an ignition device ignites the propellant, which causes a reaction that produces hot gases. The second design approach is commonly referred to as the hybrid design approach, which involves the combination of a pyrotechnic inflator and a pre-stored pressurised gas in a pressure vessel, compare Figure 7.

Figure 7: SHI-12/25: The hot gas of pyrotechnic melts the metal burst disk of the vessel. The cold gas intermingles with the hot gas and flows through the filter openings of the filter tube in order to fill a bag (white arrows).

TRW’s Side-Hybrid-Inflator (SHI) family is with about 16 million units per year the most accepted hybrid inflator with a wide range of applications (side and front impact, rollover and aviation airbags). To find an optimum in respect of these different applications concerning performance, stability and costs, it is essential among other things to get accurate information about the deformation behaviour. Up to now spatially uniform pressure distributions are applied for static structure calculations. But the computational results did not help to understand some observed behaviour during experimental investigations. For this purpose a spatio-temporal pressure input for structure calculations was necessary.

Figure 8 shows a 180° section of the geometry. The gas dynamic in the combustion chamber and in the vessel were excluded in order to reduce the computational effort. Their influence was taken into account by suitable inflow conditions. Within the structural analysis the combustion chamber as well as the vessel were rigid bodies and only the filter tube was considered deformable.

Figure 8: SHI-40/25 without burst disk and internal parts of combustion chamber.

A weak coupling was chosen as a suitable strategy for the mechanical interaction between the gas and the filter tube (exchange at each time step 1e-5s):

- MSC.Marc at SyPt(1): send mesh
- MSC.Marc at SyPt(2): receive pressure_rel
- MSC.Marc at SyPt(3): send mesh
- CFX-5 at SyPt(1): receive mesh
- CFX-5 at SyPt(3): send pressure, receive mesh
Figure 9: Relative mean pressure in filter tube calculated by CFX-5, $p_{\text{ref}} = 0.1 \text{ MPa}$ (upper part). The lower part gives the $z$-displacement $dz$ of the grid-node with the maximal $z$-displacement. It lies on the top near the exit holes of the filter tube.

Figure 9 gives in the upper part the calculated mean pressure in the filter tube. At about $1.3\text{ms}$ the burst disk ruptures and therefore the pressure strongly increases, which clearly has an impact on the displacements. As shown in the lower part of Figure 9 the time $t=15\text{ms}$ of the strongest $z$-displacement does not correlate with the time of the highest pressure. In contradiction with this behaviour a static simulation with spatially uniform pressure loads leads to the strongest displacements at the maximum of the pressure curve, $22 \text{ MPa}$. Even if the characteristic of the oval deformation (lower diameter of filter tube at the holes and higher diameter $90^\circ$ rotated) is similar in both cases, the ranges of the deformation as well as the plastic strain are different, compare Figure 10.

Figure 10: MISES-stress of the filter tube (dark=low) of a static (right, $p=22\text{MPa}$, plastic strain =0.06%) and a dynamic coupled analysis at $t=1.5\text{ms}$ (left, $p=17\text{MPa}$, plastic strain=0.22%). The displacements ($dx,dy,dz$) in ‘mm’ are $([-0.019; 0.048], [-0.09; 0.089], [-0.171; 0.009])$ for the static and $([-0.04;0.068],[0.056; 0.113],[-0.269; 0.015])$ for the dynamic analysis.

The comparison of absolute values of experiment and simulation is not reasonable due to the simplifications. However, the comparison of trends allows some instructive conclusions. For example dynamic and static burst pressure tests were performed with inflators. The goal of these experiments was to determine the failure pressure of some parts of an inflator. In static burst pressure tests water is pressed into the inflator up to the failure point. In dynamic tests an extremely aggressive propellant is ignited in the inflator and the maximal pressure is also measured. In dynamic and static tests it was found that with the SHI-40/25 filter tube the dynamic burst pressure is smaller then the static. This behaviour is predicted by the simulation: in the dynamic coupled simulation higher plastic strains at lower pressure in the filter tube are observed.
5 Conclusion

In 1998199 TRW has already successfully realised some coupled fluid-structure-simulations using the library GRISSLi [3]. Although the way via explicit programming of user sub-routines was time-consuming and fault-prone, the new insights were often helpful in order to understand some experimental observation or to improve some parts of an inflator. From there the coupled simulation has already found its place within the engineering process at TRW. With MpCCI a “closer” coupling between CFX-5 and MSC.Marc becomes possible with more flexibility concerning the exchange of variables and the coupling strategy. A set-up of a new calculation takes significantly less time and the complete coupling process is more efficient and more robust. Nevertheless the CPU requirements for coupled simulations are still high. Therefore it is necessary to use parallelized simulation codes to reduce the overall CPU time consumption. The coupling library MpCCI supports the direct coupling of codes running in parallel, but the benefit of performance in this case is strongly dependent on the relation of the computational effort of both fluid and structure analysis.

Acknowledgements

The authors are grateful for the funding of the project DEBUT by the European Commission. The authors also wish to acknowledge the valuable contribution in this work of the collaborators of DEBUT, namely Maarten Oudendijk (MSC.Software, Gouda, Netherlands), Michael Gabler (TRW Airbag Systems GmbH, Aschau, Germany), and the project coordinator of DEBUT, Ullrich Becker-Lemgau (Pallas GmbH, Berlin, Germany).

References