1. INTRODUCTION

The modern Common Rail Injection Systems provide an effective spray atomisation and hence fuel mixture in the combustion chamber, leading to a continued reduction of engine cycle pollutants and fuel consumption. In this context the detailed spray physics at nozzle outlet is extremely difficult to model, and the classical DDM approach \cite{4} is hardly applicable to such a dense and turbulent region. For this reason an Eulerian spray model \cite{7}, used in the region at the nozzle orifice, can improve the diesel injection simulation. Furthermore, the influence of cavitating nozzle flow on spray formation can be treated very well within the Eulerian spray approach \cite{5,10}. A new code coupling module allows a coupling of Eulerian and Lagrangian spray approaches in engine calculations: this means that two different CFD codes run together exchanging source terms and boundary conditions during the numerical simulation. This method ensures the highest flexibility, because existing engine codes can be coupled with the Eulerian spray simulation, providing a better description of the spray formation in the region close to the injector.

2. MODEL

The requirement of permanent improvement of accuracy enforces the application of specialised numerical models for the simulation of different physical flow regimes. In the case of high-pressure injection of diesel, these are (see Figure 1) internal nozzle flow (A), spray injection and dense-spray flow (B), and gas- and thin-spray flow with combustion in the engine chamber (C).

The focus of this paper is on coupling a CFD-simulation adopting an Eulerian spray model in the dense spray region (B) with a CFD-simulation applying a DDM spray model in the thin spray/combustion region (C). A separate, fine mesh is used for (B) and arbitrarily overlaps the coarser, moving mesh for the entire chamber, used for (C). The boundary
conditions for (B) are taken from the field values in (C). In order to synchronise the flow fields, which are computed both in (B) and (C), momentum sources from the spray in (B), which are not computed in (C), are applied separately to (C). Eulerian spray leaving region (B) is converted to DDM spray in (C).

2.1 Spray Model

This section describes briefly the applied model equations of the Eulerian and the Lagrangian spray models.

**Eulerian spray model.** The basis of the Eulerian spray is the multi-phase approach, which is obtained through the ensemble averaging process of the conservation equations [3]. This approach treats the different phases, which are gas and liquid in the considered case, as interpenetrating continua represented by their phase volume fractions. Furthermore the liquid is divided into different droplet size classes; each represented by a separate droplet phase. Table 1 shows the phase specification of the model. The first phase is always the gaseous phase for the gas mixture. The vapour mass fraction is transported by a separate scalar transport equation within the gaseous phase. The next phases are the liquid droplet phases. A constant droplet class diameter is assigned to each of these phases. The last phase is the bulk liquid phase from the nozzle, which disintegrates into the liquid droplet phases. A constant droplet class diameter is assigned to each of these phases. The last phase is the bulk liquid phase from the nozzle, which disintegrates into the droplet phases due to primary break-up processes. For every phase the complete set of conservation equations is solved. The higher the number of phases, the higher the resolution in the droplet diameter space, but the higher is also the computational effort.

<table>
<thead>
<tr>
<th>Phase</th>
<th>1</th>
<th>2, ..., n-1</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content</td>
<td>Gas mixture</td>
<td>Droplets</td>
<td>Bulk liquid</td>
</tr>
</tbody>
</table>

The multi-phase conservation equations for mass, momentum and enthalpy are shown in Eqs. (1), (2) and (3) [3]. Additional transport equations for turbulence kinetic energy and turbulence dissipation rate according to the k-ε model and a transport equation for the vapour mass fraction in the gaseous phase are solved. These equations are not shown here. The right hand sides of the conservation equation contain the exchange terms between the phases, \( \Gamma_{kl} \), \( M_{kl} \) and \( H_{kl} \).

\[
\begin{align*}
\frac{\partial \alpha_k \rho_k}{\partial t} + \nabla \cdot \alpha_k \rho_k \mathbf{v}_k &= \sum_{i=1}^{n} \Gamma_{ik} \\
\frac{\partial \alpha_k \rho_k \mathbf{v}_k}{\partial t} + \nabla \cdot \alpha_k \rho_k \mathbf{v}_k \mathbf{v}_k &= -\alpha_k \nabla p + \sum_{i=1}^{n} \Gamma_{ik} \\
\nabla \cdot (\alpha_k (\mathbf{v}_k + \mathbf{g}) + \sum_{i=1}^{n} M_{ik} \mathbf{v}_i + \mathbf{v}_k \sum_{i=1}^{n} \Gamma_{ik} \\
\frac{\partial \alpha_k \rho_k \mathbf{h}_k}{\partial t} + \nabla \cdot \alpha_k \rho_k \mathbf{v}_k \mathbf{h}_k &= \nabla \cdot (\alpha_k (\mathbf{q}_k + \mathbf{q}_i) + \mathbf{h}_k \sum_{i=1}^{n} \Gamma_{ik} \\
\alpha_k \rho_k \mathbf{h}_k + \alpha_k \mathbf{v}_k \cdot \mathbf{v}_k + \alpha_k \rho_k \frac{dp}{dt} + \sum_{i=1}^{n} H_{ik} + h_k \sum_{i=1}^{n} \Gamma_{ik} \\
\end{align*}
\]

The exchange term for the interfacial mass exchange between phase \( k \) and phase \( l \), \( \Gamma_{kl} \) in Eq. (4), gets its contribution from droplet evaporation, primary break-up and secondary break-up. The term \( \dot{m}_{kl} \) determines the evaporated mass exchange rate of a single droplet, which is calculated according to the model of Abramzon & Sirignano [1]. Note that the number of droplets per unit volume, the droplet number density, can be expressed by \( N_k = 6 \alpha_k \pi D_k^3 \). The sign is positive, if the phase receives mass, and it is negative if the phase loses mass. The primary break-up model describes the disintegration of the bulk liquid phase \( n \) into the droplet liquid phases 2 to \( n-1 \). The mass exchange rate \( \dot{m}_{P,kl} \) is calculated with the model of Bianchi & Pelloni [2], which delivers a correlation for the radius change rate \( (dR/dt)_{P,kl} \) of the disintegrated blobs. The droplets produced from the primary break-up are subjected to further secondary break-up processes. These are modelled using the standard WAVE break-up model [11], which determines the radius change rate \( (dR/dt)_{S,kl} \). The break-up models are applied in each cell of the spray region. If the predicted size of break-up products is less than the parent drop diameter, mass is transferred according to the break-up rate into the corresponding droplet size class. The break-up model equations for both, primary and secondary break-up, as well as the translation of the equation into the Eulerian frame are described in more detail in [6,7].

The momentum exchange between the gaseous phase and the liquid phases, \( M_{kl} \), is determined by drag forces, \( M_{D,kl} \), and turbulent dispersion forces, \( M_{T,kl} \), as shown in Eq. (5). The drag coefficient \( c_D \) is a function of the droplet Reynolds number, the liquid volume fraction and the deformation. The turbulent dispersion force is modelled following the approach of [9], with a constant or modelled turbulent dispersion force coefficient \( c_T \).

The evaporation model determines the interfacial enthalpy exchange, \( H_{kl} \), between the gaseous phase and the liquid phases. The term \( \dot{Q}_{E,kl} \) in Eq. (6) represents the heat flow rate into a single droplet and is calculated according to the correlation of [1].

\[
\Gamma_{kl} = \Gamma_{E,kl} + \Gamma_{P,kl} + \Gamma_{S,kl} = 6 \alpha_k \pi D_k^3 \left\{ \pm \dot{m}_{E,kl} + \dot{m}_{P,kl} \pm \dot{m}_{S,kl} \right\} \\
\text{with } \dot{m}_{P,kl} = \rho_k \pi D_k^3 \left( \frac{dR}{dt} \right)_{P,kl} ; \dot{m}_{S,kl} = \rho_s \pi D_k^3 \left( \frac{dR}{dt} \right)_{S,kl} \\
M_{D,kl} = M_{D,kl} + M_{T,kl} = c_D \frac{6 \alpha_k \rho_k}{D_k} (v_i - v_k) (v_i - v_k) + c_T \rho_k h_k \nabla \alpha_k = -M_{kl} \\
H_{kl} = N_k \dot{Q}_{E,kl} = 6 \alpha_k \pi D_k^3 \dot{Q}_{E,kl} = -H_{kl} \\
\]

**Lagrangian spray model.** The Lagrangian DDM method [4] solves the equation of motion for the spray parcels, which consists of a certain number of droplet and particles of similar properties, by integrating the momentum equation, Eq. (7), over time. The parcels motion is determined by the forces acting on it: the drag force \( \mathbf{F}_d \), the pressure force \( \mathbf{F}_P \) and the body force \( \mathbf{F}_b \). The turbulent dispersion force is included in the drag force term \( \mathbf{F}_D \) providing a fluctuating
component in statistically random direction. The same evaporation and break-up models, as described above for the Eulerian spray model, are applied to influence the mass \( m_d \) and the energy of the parcel.

\[
m_d \frac{dN_d}{dt} = F_{id} + F_{ps} + F_B
\]  

(7)

Within AVL’s CFD code FIRE both approaches, the Eulerian spray and the Lagrangian DDM model are available and are used in the coupled simulation presented in this work.

2.2 Code Coupling Tool

For the considered coupled spray- and engine-simulation, boundary conditions for the spray-region need to be derived from the engine-simulation and momentum sources from the spray-region need to be applied to the engine-simulation, so as to synchronize the flow in both regions.

For such coupled simulations, the AVL Code Coupling Interface (ACCI), similar to MpCCI (see www.mpcci.org), has been developed. ACCI performs two basic tasks: the data exchange between an arbitrary number of coupled simulation processes, and the transformation of field-values between the different meshes used in the coupled processes to describe the same space. The exchanged field-values are called attributes, and the exchange-spaces are called interfaces. ACCI allows to configure the coupling at run-time. The configuration, defined in a text file, describes which attributes are exchanged at which interfaces between which processes.

The coupling interface has been implemented as server-client model. For each coupled simulation, there is one server program running. The clients, who are the coupled simulation processes, send/receive data only to/from this server. The communication is based on the TCP/IP-protocol, which allows running clients and server on different computers in the same network (Figure 2).

**Figure 2:** Client-server-architecture in the ACCI interface

The configuration file mainly contains specifications in the format \( client_id \) {set|gets} \( attr_id \) at \( itf_id \). \( client_id \), \( attr_id \) and \( itf_id \) stand for a client-identifier, attribute-identifier and interface-identifier, respectively. The keyword \( set \) indicates that a client will provide data, \( gets \) indicates that it requests data. A configuration file for a simplified version of the coupling described in this paper might look like that:

```
engine sets p at spray_boundary
engine sets v at spray_boundary
```

This tells ACCI that the client \( engine \), e.g. the engine simulation, provides fields for pressure \( p \) and velocity \( v \) at the interface \( spray\_boundary \), which is the space defined by the surface of the spray-domain, and that the client \( spray \) receives this data. It further tells that \( spray \) provides momentum sources from its volume and that \( engine \) receives these sources. Attributes may hence be scalars, as pressure, or vectors, as velocity.

The exchange is specified by separate \( set \) and \( get \) statements, instead of a combined \( sends-to \) command, because this allows for an \( n\)-to-\( m \)-relation of clients at an interface: an arbitrary number of clients may send data and a different number of clients may receive attributes at the same interface. This again implies that the interface can be distributed among an arbitrary number of simulation processes on either side – provider and receiver. This allows a convenient handling of clients running under MPI, in that ACCI handles each MPI-client-sub-process as separate client.

The interface identifiers are logical names for spaces. In general, the clients use meshes of different topology and resolution to represent the same spaces. ACCI transfers the attribute values between the different meshes. The meshes may even be of different topological dimensions. As an example, \( spray\_boundary \) is the set of all boundary-faces of the mesh in \( spray \). In \( engine \), this surface arbitrarily intersects some cells, and the values in these topologically three-dimensional cells have to be transferred to the topologically two-dimensional boundary faces.

Currently, ACCI supports values given at cell or face centers, as usual in CFD. The transfer of attribute values is done based on the intersection-volumes of cells (or intersection-areas of faces, or between cells and faces), so that the spatial integrals of the attribute values are conserved. Figure 3 shows the intersection of a prismatic receiver-cell \( j \) with volume \( S_j \) with a hexahedral provider-cell \( i \) with volume \( S_i \), together with the intersection polyhedron with volume \( S_{ij} \).

**Figure 3:** Intersection of cells between different meshes of two clients

\[
b_j = \sum_i w_{ij} a_i
\]  

(8)
An attribute value \( b_j \) for a receiver-cell \( j \) is obtained from a weighted sum of the values \( a_i \) over all provider-cells \( i \) according to Eq. (8). The weighting factor \( w_j \) is computed from the intersection volumes (or areas), depending on whether the attribute is extensive or not. The cell values of extensive attributes, e.g. of a mass source in [kg/s] or a force in [N], are proportional to the cell size, and the weighting factor is given by Eq. (9). While for non-extensive attributes, as pressure or velocity, the weighting factor is determined by Eq. (10).

ACCI assumes all cells to be convex when the intersection volumes are computed. Supported cell types are hexahedra, tetrahedra, pyramids, triangular prisms, triangular and quadrilateral faces and line-segments. To speed up the calculation of all the intersection volumes of all provider-cells with all receiver-cells, a background grid is used to sort out quickly potentially intersecting cells.

ACCI consists of the server program and a library, which has to be linked to the simulation code. The server processes requests from clients, buffers data and transfers data between different meshes. The library communicates with the server and provides an interface (API) to the simulation code. Figure 4 shows the interaction sequence of the simulation code, the ACCI library and the ACCI server, for a client with one set-event and one get-event defined. Only calls to the API-functions \( \text{init, exchange} \) and \( \text{exit} \) have to be inserted directly into the simulation code. \( \text{init} \) passes the client-id to the library, which in turn registers the client at the server. The server is identified by a server-host-name and a TCP/IP port number. The server, who has already read the coupling configuration file, returns the configuration information relevant for the client. This is stored by the library. Within its time-step-loop the simulation code has to call the exchange function, passing the current time \( t \) as argument. This triggers the data-exchange in the library. For all coupling-events specified in the configuration for the client, the library first retrieves the corresponding interface-data, i.e. vertex coords, cell types and vertices, from the simulation code by calling a function \( \text{get_interface(if_id)} \). This function has to be implemented in the simulation code. \( \text{if_id} \) is the interface-id as specified in the configuration file. The library then sends this information, with the time-stamp \( t \), to the server. For static meshes, this part of the sequence is done only at the first time step. For moving meshes this is done each step.

After that, the library processes all get- and set-events at all interfaces. For get-events, it calls a function \( \text{get_values(if_id, attr_id)} \), which has to be implemented in the simulation code and which must return the values of the attribute \( \text{attr_id} \) for all cells in the interface \( \text{if_id} \). Then it sends these data, again with the time-stamp \( t \), to the server. For set-events, the library requests data, identified by interface- and attribute-id, from the server. The server, if it did receive the data from any other client, transfers the data to the requesting client’s interface-mesh and sends back the values to the library, which calls a function \( \text{set_values(if_id, attr_id, value)} \). This function must be implemented in the simulation code and must set the sent values as boundary conditions or source-terms. If the server did not receive appropriate data yet, it returns not available and the library blocks until the data are available. This mechanism synchronizes the coupled clients.

After the time-step-loop, the simulation code calls \( \text{exit} \), which in turn un-registers the client at the server. When all coupled clients have un-registered, the server terminates.

The above description is simplified. ACCI in fact allows an arbitrary number of exchange points in the code, which can be addressed in the configuration file. In FIRE there are further exchange points at initialization and at the beginning of each solver-iteration. It further allows different time-steps in all clients. In this case data exchange is only done at specified coupling time-steps. At time steps between exchanges the ACCI library optionally sums-up or integrates any get-values, so that an integral or sum over the coupling time step is actually sent to the server.

### Figure 4: Interaction-sequence-diagram for a client in a coupled simulation

#### 2.3 Data Flow between the Codes

Figure 5 shows the basic data flow for the coupled spray-engine-simulation:

1. **Computational Initialisation**
The computational set-up of both clients has to be performed. Especially the fluid properties for both, liquid and gas species, have to be defined according to the species database.

2. **Initial conditions**

For all phases and all cells the thermodynamic conditions in the spray client have to be defined: pressure, turbulent viscosity, temperature, turbulence kinetic energy, turbulence dissipation rate, volume fraction, gas composition and scalar.

3. **Definition of time-step**

The time-step in the spray solver is usually smaller than in the engine one, as the cell dimensions and the dynamic conditions are different in both codes. The time-step of the engine client is a multiple of the spray client time-step. The engine time-step is used as coupling-time-step.

4. **Spray inlet**

The conditions at the spray inlet are either determined by an injection rate table or through a previous nozzle flow simulation, which provides the boundary conditions for each droplet and gas phase via a nozzle-flow data file.

5. **Source terms**

The interactions between the droplets and the gas are described by exchange terms, the sources. In the overlapping domain of the two computational grids the gas phase flow field is calculated on both clients (see Figure 6). Therefore the source terms of the spray client are transferred to the engine client, providing similar description of the phase interactions in both clients. The following sources are transferred via the code coupling server:

- **Momentum**: due to drag and turbulent dispersion interaction between liquid phases and the gaseous phase;
- **Mass**: due to evaporation of droplet phases;
- **Scalar**: source for the transport of vapour mass fraction;
- **Energy**: due to evaporation and heat exchange

The mapping procedure, using weighting factors as described in the previous section, ensures conservation of these extensive attributes. If a smaller time-step is used in the spray-simulation, the ACCI library automatically integrates the values between the coupling-time-steps.

6. **DDM parcels**

The computational droplets have to be introduced inside the engine grid according to criteria on parcel number, mass and distribution on the outlet boundary of the spray grid. If the mass leaving the spray domain (region II of Figure 6) through the boundary face \( A_f \) is greater than a user defined threshold \( m_{d,min} \), a new parcels with mass \( m_d \) according to Eq. (11) is initialised. Equation (12) determines the number of physical droplets of the spray parcel \( n_d \).

\[
m_d = \sum_{k} \left( \alpha_k \rho_k v_k \right), A_f \Delta t \geq m_{d,min}
\]

\[
n_d = \frac{m_d}{D_f \gamma P_t}
\]

7. **Updating physical properties**

During the simulation the physical properties of the gas and liquid phases have to be updated according to environment conditions on i.e. temperature and pressure. The values are extracted from the species database.

8. **Output timing**

The frequency of local or integral result output can be prescribed via user functions.

9. **Spray boundaries**

The engine client provides gas flow field boundary conditions for the spray client, as shown in region I and II of Figure 6. The transferred values are pressure or velocity, temperature, turbulence kinetic energy and turbulence dissipation rate.

**Figure 5**: The interfaces for the data exchange

**Figure 6**: Coupling regions for source terms, spray parcels and boundary conditions

3. **SIMULATION AND VALIDATION**

3.1 **Coupling Geometries and Calculation Set-up**

The adopted grids are shown in Figure 7. The Eulerian spray grid contains about 8400 cells and has a length of 15 mm. The nozzle inlet is geometrically fully resolved and has a diameter of 205 μm. The engine segment has about 27000 cells and covers a range of 80 mm. Figure 7 right shows the intersected cell volumes of the two grids, calculated by the ACCI server for determining the weighting factors.

The injection velocity profile is extracted from I-Level experimental data on a 1-hole injector (Bosch 200/1, DLLA0PV3185842): the spray injection velocity reaches a maximal value of 370 m/s resulting from 800 bar rail pressure and the injection runs for 2.5 ms.

The Eulerian spray simulation uses six phases, one gaseous phase, four droplet phases and one bulk liquid phase.
The droplet class diameters are 5, 10, 20 and 40 μm for the droplet phases, and a blob diameter of 205 μm is assigned to the bulk liquid phase. The activated models are evaporation, primary break-up, secondary break-up, drag and turbulent dispersion. The engine calculation is performed as single-phase simulation with standard species transport.

Two test cases are presented in this work: one in a cold and the other in a hot environment, in order to give a complete overview of the coupling concept. The simulations of both clients, the Eulerian spray and the Lagrangian engine simulation have been performed with AVL FIRE. However, the applied methodology is not restricted to coupling of simulations with FIRE, it also allows coupling with other CFD codes.

3.2 Cold Spray Case

The cold spray test case is easier to handle, because there is no evaporation and hence there is no exchange of mass, vapour species and energy sources. That means only the momentum sources between the liquid droplet phases and the gaseous phase are transferred from the spray client to the engine client. Furthermore the liquid mass leaving the flow domain of the spray client is transferred to the engine client to initialise the Lagrangian DDM parcels. On the other hand the flow field of the engine client acts as boundary condition of the spray client. The chamber pressure is 20 bar and the injection is simulated from 0 to 2.5 ms.

The gas velocity fields of both clients and the introduced DDM parcels at 0.6 ms are shown in Figure 8. The velocity vectors have uniform length and are plotted on a courser grid. One can see the gas flow field is well propagated from the engine to the spray simulation.

Figure 9 shows the cell values of the gas velocity fields of both clients at 0.5 and 1.5 ms. The velocity fields appear similar, and the integral momentum of the whole domain transferred via the coupling server is conserved. However, there must be differences in the gas velocity fields, especially in the region close to the injection due to the following reasons:

- Due to the presence of the liquid phase, the cell mass in the spray is bigger than in the engine simulation.
- The velocity inlet is fully resolved by the spray mesh, while the corresponding engine mesh has a wall at that position, which definitely affects the gas velocity field.

All in all the Eulerian spray simulations delivers a more accurate description of the gas velocity field in the dense spray region. The momentum exchange between gaseous and liquid phases is transferred to the engine simulation to provide there the benefits of the Eulerian spray.

Figure 7: The computational grids and the intersected mesh

Figure 8: Gas velocity fields for Eulerian spray simulation and engine code simulation at 0.6 ms

Figure 9: Gas velocity fields at engine and Eulerian spray simulation at 0.5 ms and 1.5 ms

Figure 10 shows a three-dimensional view of the cold spray, total liquid volume fraction in the Eulerian spray domain and the spray parcels in the engine domain, at 1.5 ms. The second plot of the velocity vectors of the gas flow field is shifted from its default position for better illustration. One can see that the maximum velocity there is about 140 m/s.
The simulation is performed in a hot environment of 900 K with chamber pressure of 54 bar, in order to achieve conditions similar to a real engine. The same inlet velocity table as for the cold spray case is applied. The simulation performed from 0 to 2.5 ms started at rest. Additional to the momentum source, the sources of mass, scalar and enthalpy in the intersected region of the computational meshes are transferred from the spray client to the engine client. Figure 11 shows the gas temperature field and the distribution of the vapour mass fraction of both clients at 0.5 and 1.5 ms. Temperature as well as vapour mass fraction fields look similar in both codes. Thus, energy, mass and scalar sources are correctly mapped to the engine client. Minor differences can be observed in the region close to the injector inlet. Due to the higher grid resolution, the coexistence of gaseous and liquid phases in the Eulerian spray simulation and the reasons exploited in section 3.2, the results of the spray client seems to be more reliable.

Figure 12 shows the total volume fraction of the Eulerian spray simulation and the spray parcels as well as the shifted plot of the vapour mass fraction of the engine simulation at 1.5 ms. Due to evaporation the spray parcels vanishes after a certain time. The diameter of the droplet phase with the biggest droplet size class diameter is 40 micron. One can observe spray parcels initialised with that droplet diameter, which appear in red colour. That means that the droplet break-up processes have not been finished within the Eulerian spray domain. Further secondary break-up of the spray parcels occurs in the engine simulation.

3.4 Simulation with Cross-Flow

Swirl supports the mixture formation of common diesel engines significantly. The proposed methodology is able to treat flows with given initial conditions, e.g. the spray simulation starts with the imposed flow field of the engine client. Therefore the pressure, velocity, density, turbulence and temperature field are transferred via the ACCI interface at an additional data exchange point, called initialisation. That means the gas flow field of the spray client is completely initialised by the gas flow field of the engine client. During the simulation the changes in the flow field are imposed via the boundary conditions.

This feature has been tested in a coupled simulation with cross-flow, which approximates the swirl of the real engine geometry. The flow field of the engine simulation has been initialised similar to the hot spray case of section 3.2. An additional cross-flow of 5 m/s in direction perpendicular to the spray axis has been applied. Figure 13 shows a comparison of the simulations without (left) and with cross-flow (right) at 1 ms after begin of injection. As is expected, the total liquid volume fraction and the spray vapour mass fraction are deflected in direction of the cross flow. Small droplets are stronger deflected than big droplets, as the ratio between drag forces and inertia forces is greater for small droplets. That leads to separation of the droplets and accelerates the evaporation.
4. CONCLUSION AND OUTLOOK

The coupling method allows a flexible integration of the Eulerian spray model in engine applications. The FIRE-FIRE tests show good results for the spray and the AVL ACCI coupling interface appears reliable, flexible and already consolidated, and allows an efficient data transfer between the coupled simulations. However, this method is not restricted to FIRE-FIRE coupling. With moderate effort, the coupling of FIRE with a third-party CFD code can be realised as well.

Next steps will treat a coupled simulation with combustion in the engine code. Therefore energy and vapour sources, effected through the combustion models, will be transferred from the engine to the spray client. This method can be extended for an on-line-coupling with the nozzle flow simulation. The computed two-phase flow field at the nozzle-outlet, typically asymmetric due to cavitation, is then directly coupled with the inlet boundary of the Eulerian spray simulation. Additionally, this allows to simulate the reaction of the engine flow on the nozzle flow, which is not possible with the current off-line coupling based on a nozzle flow data file.

5. ACKNOWLEDGEMENTS

This work has been supported by the European NICE project (New Integrated Combustion system for future passenger car Engines) under contract number TIP3-CT-2004-506201-NICE. Thanks to the partners of sub-project B.1, CFD Tools.

6. NOMENCLATURE

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>surface</td>
<td>m²</td>
</tr>
<tr>
<td>a,b</td>
<td>physical attribute</td>
<td>[-]</td>
</tr>
<tr>
<td>c</td>
<td>model coefficient</td>
<td>[-]</td>
</tr>
<tr>
<td>D</td>
<td>diameter</td>
<td>m</td>
</tr>
<tr>
<td>F</td>
<td>force</td>
<td>N</td>
</tr>
<tr>
<td>g</td>
<td>spec. body force</td>
<td>m/s²</td>
</tr>
<tr>
<td>H</td>
<td>enthalpy exchange</td>
<td>W/m²</td>
</tr>
<tr>
<td>h</td>
<td>spec. enthalpy</td>
<td>J/kg</td>
</tr>
<tr>
<td>k</td>
<td>turb. kin. energy</td>
<td>m²/s²</td>
</tr>
<tr>
<td>M</td>
<td>momentum exch.</td>
<td>N/m³</td>
</tr>
<tr>
<td>m</td>
<td>mass</td>
<td>kg</td>
</tr>
<tr>
<td>N</td>
<td>number density</td>
<td>1/m³</td>
</tr>
<tr>
<td>n</td>
<td>number of phases</td>
<td>[-]</td>
</tr>
<tr>
<td>nd</td>
<td>number of parcels</td>
<td>[-]</td>
</tr>
<tr>
<td>p</td>
<td>pressure</td>
<td>Pa</td>
</tr>
<tr>
<td>Q</td>
<td>heat flux</td>
<td>W/m²</td>
</tr>
<tr>
<td>S</td>
<td>intersec. volume</td>
<td>m³</td>
</tr>
<tr>
<td>t</td>
<td>time</td>
<td>s</td>
</tr>
<tr>
<td>V</td>
<td>volume</td>
<td>m³</td>
</tr>
<tr>
<td>v</td>
<td>velocity</td>
<td>m/s</td>
</tr>
<tr>
<td>w</td>
<td>weighting factor</td>
<td>[-]</td>
</tr>
<tr>
<td>α</td>
<td>volume fraction</td>
<td>[-]</td>
</tr>
<tr>
<td>Γ</td>
<td>mass exch. rate</td>
<td>kg/sm³</td>
</tr>
<tr>
<td>ρ</td>
<td>density</td>
<td>kg/m³</td>
</tr>
<tr>
<td>τ</td>
<td>stress tensor</td>
<td>N/m²</td>
</tr>
<tr>
<td>θ</td>
<td>heat source</td>
<td>W/kg</td>
</tr>
</tbody>
</table>

7. REFERENCES