Fluid Structure Interaction analysis using Abaqus and FlowVision

A. Aksenov, A. Dyadkin, T. Luniewski, V. Pokhilko

Capvidia, Belgium

Abstract: The proposed fluid-structure interaction (FSI) approach is based on a two-way coupling between finite-element code Abaqus and finite-volume code FlowVision. The FSI simulation is possible due to a unique mesh generation method used in FlowVision. The method is called Sub-Grid Resolution Method (SGRM). The SGRM connects seamlessly FE and CFD meshes without introducing any intermediate structures into the FSI layer. It allows to link different mesh types with different discretization levels and ensures a fully conservative (no data loss) bi-directional data transfer between structure and fluid simulation domains. The approximation accuracy of the governing equations is maintained in both simulation domains and the FSI layer. Sloshing water in a flexible cylindrical tank is simulated to illustrate the described fluid structure interaction approach.

1. Introduction

The explosive development of numerical methods in the past decades opened new possibilities to conduct most sophisticated structural and fluid dynamics simulations. Complex scientific and industrial problems can be efficiently solved today, but structural and fluid simulations are still performed separately. It is no problem to calculate the aircraft wing bending under external load, it is no problem to model the airflow over the wing, but it is difficult to simulate both effects simultaneously.

A complex fluid-structure interaction approach is required to simulate dynamics of turbine blade excited by the passing flow, flutter analysis of aircraft wings, hydroplaning of automotive tires, aerodynamics of sails or airbags, etc. This category of problems requires a simultaneous solution of governing equations as the physical phenomena are coupled and influenced by each other.

Coupling fluid dynamics and structure dynamics codes is difficult due to different domain discretization (fluid grids vs. FE-meshes) and different numerical approach for solving the governing equations. Fluid dynamics codes generally use a finite-volume approach; structure dynamics codes use a finite-element approach. One solution to “conciliate” the codes is to organize the data exchange via intermediate structures which interpolate and transfer the information from one grid to another (MpCCI, http://www.scai.fraunhofer.de).

In this paper another approach to solve fluid structure interaction problems is presented. This approach is based on the Sub-Grid Resolution method (Aksenov et al, 1996, Aksenov et al, 1998). In this method the Cartesian grid with local adaptation is introduced in the fluid flow computational domain. The structure is described by a finite-element mesh. The external surface of the finite-element mesh is used as a boundary for the fluid flow. The cells in the fluid domain are intersected by the surface mesh describing the structure. The resulting cells are not simplified and stored in the form of complex polyhedrons. The cell sides that are part of the surface mesh have links to the original faces of the finite-element mesh. Solving fluid dynamics equations on such a grid provides a natural data transfer mechanism allowing transfer of data from fluid to structure computational domains without any intermediate interpolation.

The above described approach was tested using ABAQUS for the structural dynamic simulation and FlowVision for fluid dynamic simulation. FlowVision uses the finite-volume approach and the Sub-Grid Geometry Resolution Method to solve the governing fluid dynamics equations.

The numerical approach for solving fluid structure interaction problems using the Sub-Grid Geometry Resolution Method is described in this paper. To demonstrate a fluid-structure interaction problem, the simulation of sloshing water in a flexible tank is chosen.

2. Sub-Grid Geometry Resolution Method for fluid domain

A rectangular calculation grid is introduced in the fluid computational domain. To capture small geometry details and solution variations a local automatic grid adoption is used. The local grid adoption subdivides the initial cell into 8 different cells (1st level grid adoption).

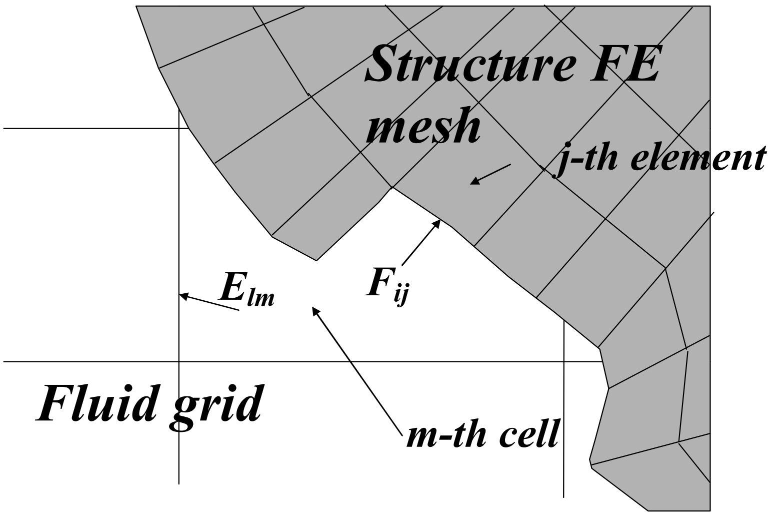

Figure 1. Cutting finite-volume cell with finite-element mesh.

The computational domain boundary is presented by a set of plane facets (surface mesh). This surface mesh is formed by a set of external faces F_{ij} of the finite-element mesh (Figure 1), where i is the face number of element j. If a boundary crosses the grid cell the initial cell is cut by the surface mesh transforming the original cell into a multifaceted polyhedron. A shape S_m of the m-th cell can be presented as a sum of cell faces E_{lm} (here l is number of face) and part of the j-th element i-th face dF_{ijm} that was cut by the cell itself

$$(1) \qquad S_m = \sum_{l=1,L} E_{lm} + \sum_{i=1,I} dF_{ijm} \ ,$$

where L is the number of existing cell faces and I is the number of faces. All this information is "stored" in the cell. The boundary cell "knows" the corresponding face dF_{ij} in the finite element mesh. Using this link data can be easily transmitted to the FE code and back. For example, the pressure of fluid P_{ij} on face F_{ij} is found by integrating pressure over all cells containing this face:

$$(2) \qquad P_{ij} = \frac{1}{s(F_{ij})} \sum_{m=1,M} P_m s(dF_{ijm}),$$

where $s(F)$ is the area of face F. The information from the finite-element code, for example, velocity of face F_{ij} , is easily transmitted to the CFD code.

3. FSI Numerical Method

3.1 Governing equations

The dynamic equilibrium (for the discrete model in finite-element system) is defined in terms of the external applied forces (**P**), the internal element forces (**I**), and the nodal accelerations:

$$(3) \qquad \mathbf{M}\frac{d^2\mathbf{u}}{dt^2} = \mathbf{P} - \mathbf{I}$$

where **M** is the mass matrix of the finite element system, **u** – displacement of the nodes. **P** is the sum of forces that act on the structure, including forces from the fluid.

Navier-Stokes equations are used for the description of the fluid flow (in this paper the incompressible fluid is considered).

$$(4) \qquad \frac{\partial \mathbf{V}}{\partial t} + \nabla(\mathbf{V} \otimes \mathbf{V}) = -\frac{\nabla P}{\rho} + \frac{1}{\rho}\nabla(\mu \nabla \mathbf{V}) + \mathbf{g}$$

$$(5) \qquad \nabla \mathbf{V} = 0$$

where **V** is the velocity vector, P is pressure, μ - effective viscosity (turbulent and molecular), ρ – density, **g** – gravity vector. The effective viscosity is calculated using a k-ε turbulent model (the turbulence model does not affect the fluid-structure interaction and is not further discussed in this paper)t.

3.2 Boundary conditions on fluid-structure interface

Except ordinary boundary conditions for simulating structure deformations, boundary conditions at fluid-structure interface are specified in terms of pressure loading from the fluid calculated in (2) for each element face that is in contact with the fluid.

The general boundary conditions are applied to inlet, outlet and walls and special boundary conditions are specified on the fluid-structure interface:

$$\mathbf{V}_{ij} = \sum_{n=1,N} w_{nij} \frac{d\mathbf{u}_{nij}}{dt} \tag{6}$$

where N is the number of nodes for element face F_{ij} , w_{nij} is weight coefficient for calculating face velocity depending on the face geometry.

3.3 Numerical scheme

Let’s assume we have a numerical solution for the displacement of structure nodes and velocities of the fluid at time step n at time moment $t=t^n$. To find the solution for the system of equations (3-5) at time step n+1 ($t^{n+1}=t^n+\Delta t$) FlowVision uses split algorithms solving Navier-Stokes equations (Belotserkovsky, 1994) based on an accurate approximation scheme for convective impulse transport (Aksenov et al, 1993). Navier-Stokes equations for cells with volumes changing in time have the following notation:

$$\frac{\mathbf{V}^{n+1} - \mathbf{V}^n}{\partial t} + \nabla(\mathbf{V}^{n+1} \otimes \mathbf{V}) = -\mathbf{V}^{n+1} \frac{v^{n+1} - v^n}{v^{n+1}} - \frac{\nabla P^{n+1}}{\rho} + \frac{1}{\rho} \nabla(\mu \nabla \mathbf{V}^{n+1}) + \mathbf{g} \tag{7}$$

$$\nabla \mathbf{V}^{n+1} = -\frac{v^{n+1} - v^n}{v^{n+1}} \tag{8}$$

here v^n and v^{n+1} are the cell volume at time step t^n and t^{n+1}(Figure 2).

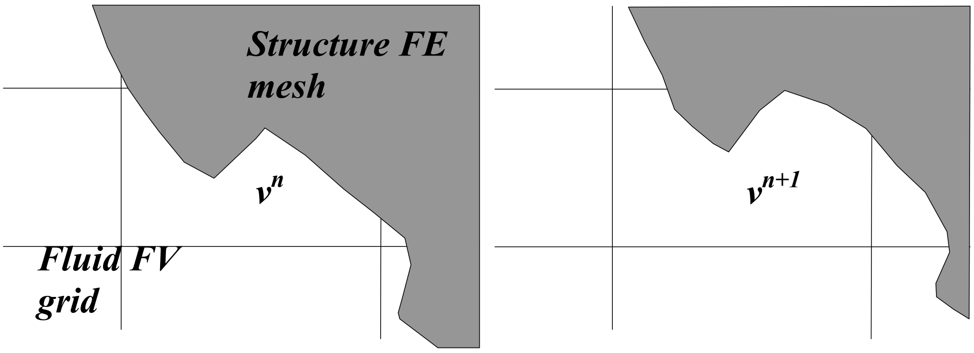

Figure 2. Changing volume of fluid cell from v^n to v^{n+1} in time step.

The first term in the right part of equation (7) is the reaction force from the structure on the fluid. The term in the right side of equation (8) describes the changing mass inside the cell. If the cell doesn't contain a moving boundary, the term $\frac{v^{n+1}-v^n}{v^{n+1}}$ equals zero and (7-8) corresponds to the approximation of ordinary Navier-Stokes equations.

An explicit method is used for solving structure dynamics equations.

$$\left.\frac{d^2\mathbf{u}}{dt^2}\right|^n = \mathbf{M}^{-1}(\mathbf{P}-\mathbf{I})\Big|^n \tag{9}$$

$$\left.\frac{d\mathbf{u}}{dt}\right|^{n+1/2} = \left.\frac{d\mathbf{u}}{dt}\right|^{n-1/2} + \left.\frac{(\Delta t^{n+1}+\Delta t^n)}{2}\frac{d^2\mathbf{u}}{dt^2}\right|^n \tag{10}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^{n-1} + \Delta t^{n+1}\left.\frac{d\mathbf{u}}{dt}\right|^{n+1/2} \tag{11}$$

FlowVision supplies the numerical solution for system (7,8), ABAQUS solves the system (9-11).

3.4 Selection of the time step Δt

The time step Δt is selected as minimum time steps of a stable solution in Abaqus and FlowVision.

The maximum time step is selected in Abaqus and equals

$$\Delta t_{structure} = \frac{L_{\min}}{c_d}, \tag{12}$$

where L_{min} is the smallest element dimension in the FEA mesh and c_d is the dilatational wave speed.

$$c_d = \sqrt{\frac{\lambda + 2\mu}{\rho}}$$

where λ and μ are Lame constants for structure material.

The maximum time step for fluid modeling is selected in FlowVision and equals

$$(13) \qquad \Delta t_{fluid} = \min(\frac{h_{mk}}{V_{mk}})\Big|_{fluid-structure\,\mathrm{int}\,erface}$$

where h_{mk} distance between m-th and k-th cells and V_{mk} is velocity of the fluid between those cells. Cells m and k are cells that lie near the fluid-structure interface.

One can see that both time steps are responsible for transferring the disturbance between the nodes. The time step for integration of the system (7-11) can be selected as minimum between $\Delta t_{structure}$ and Δt_{fluid} to get a stable solution

(14) $\Delta t = \min(\Delta t_{structure}, \Delta t_{fluid})$

On another hand, the solution of the system (7-11) can lead to quite different time steps for the "structure" part and the "fluid" part. The "limiting" system is the system requiring the solution with the smallest time step. To avoid solving the system of equation with very small time step, the time step is chosen as maximum ($\Delta t_{structure}$, Δt_{fluid}). In this case to satisfy the stability conditions, the "limiting" system is solved iteratively with smaller time increments. During these iterations, inside the time step, boundary conditions transferred from the "non-limiting" system stay constant until the end of each time step.

3.5 Implementation

Link with ABAQUS is made using the user DLOAD function (VDLOAD for ABAQUS explicit) for transfer of loads calculated in FlowVision. Node positions are obtained by FlowVision via ODB utilities.

The following algorithm for calculating n-th time step is used:

1. Preliminary increment dT of n-th time step is calculated.
   $t^n = t^{n-1}+dT$
2. ABAQUS calls DLOAD, pressure from n-1 time step transferred from FlowVision.
3. ABAQUS specifies its own stable time increment dt
4. ABAQUS makes increment, current time ABAQUS time t' increased by dt

5. ABAQUS writes data to ODB file.
6. FlowVision seek last write in ODB file to get ABAQUS time t'.
7. if $t' < t^n$, go to 2
8. FlowVision reads ODB file to get node positions on n-th time step and value of n-th time step t^n.
9. FlowVision makes simulations with time increment $t^n - t^{n-1}$
10. if it is not end of simulation, go to 1

4. Simulating sloshing water in a flexible tank

The described simulation approach of the fluid structure interaction problem is performed for calculating water motion inside a flexible tank (Figure 3, a). The bottom of the tank is oscillating with speed 0.2sin(1.8 t) m/sec, where t is time in sec. The walls of the tank have the following properties – density 3.000 kg/m^2, Young modulus 9x10^5 N/m^2, Poisson coefficient 0.3. Dimensions of the tank and finite element mesh (Figure 3, a). The tank is filled with water with height 2 meter (Figure 3, b)

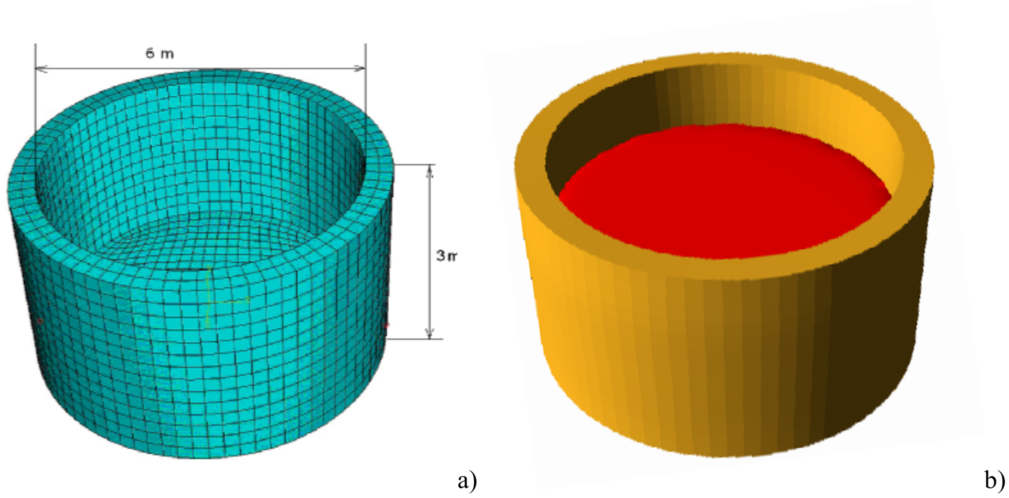

Figure 3. Flexible tank. a) finite-element mesh and dimension, b) initial water state

The problem is solved using ABAQUS-Explicit v6.4 and FlowVision 2003 solving Navier-Stokes equations for incompressible fluid with k-ε turbulence model. The water-air interface is tracked by VOF (Volume of Fluid) -like method (Hirt , Nichols, 1981).

The number of cells in the finite-volume grid is 16.000, the finite-element (tank) mesh includes 5.100 nodes and 3.300 elements. The time step is 0.03 sec.

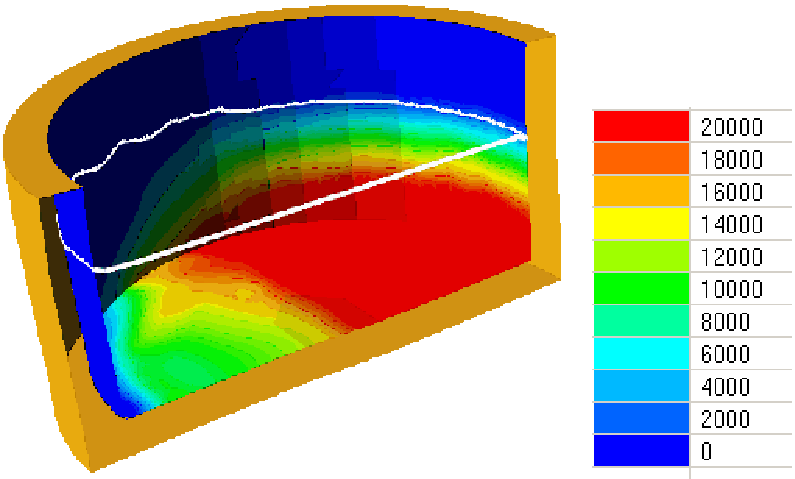

Figure 4. Pressure (Pa) and water surface distribution at time 3 sec.

The water pressure distribution is shown in Figure 4 by color map, time moment is 3 sec. The water surface is shown by contour. The computational domain is cut by plane to visualize details inside the tank.

5. Conclusion

An efficient realization of the fluid-structure interaction is possible due to the unique Sub-Grid Resolution Method providing natural link between CFD and FEM meshes. The simulation of sloshing water in a flexible tank with oscillating bottom is just one example. The described approach has been successfully used for other simulations involving complex geometry, multi-phase media, moving bodies and large structural deformations.

Known limitation of the approach: the fluid boundary is defined by the FEM mesh, which in some cases is not accurate enough to describe the boundary for the fluid flow calculation. An illustrative example is bending of a turbine blade caused by the passing gas flow. To describe accurately the flow separation on the blade surface the fluid grid has to be very fine. Thus, the finite element mesh must also be very fine. This is not practical as the resulting FEM models become huge and unmanageable. To overcome it we will use the original CAD geometry. The Sub-Grid Resolution Method constructs the fluid grid by intersecting the orthogonal fluid cells with the CAD surface exactly describing the fluid-structure border. The elements on the fluid-structure border can now be matched with the much coarser FEM mesh. This is ultimately solving all limitations of the described above fluid-structure interaction approach and will be implemented in the next FlowVision release.

6. References

1. Aksenov A, Dyadkin A, Pokhilko V. Overcoming of Barrier between CAD and CFD by Modified Finite Volume Method, Proc. 1998 ASME Pressure Vessels and Piping Division Conference, San Diego, ASME PVP-Vol. 377-1., 1998
2. Aksenov A., Dyadkin A., Gudzovsky A., 1996, “Numerical Simulation of Car Tire Aquaplaning”. Computational Fluid Dynamics ’96, J.-A. Desideri, C.Hirsch, P.Le Tallec, M.Pandolfi, J.Periaux edts, John Wiley&Sons, pp. 815-820.
3. Aksenov A., Gudzovsky A., Serebrov A., 1993, “Electrohydrodynamic Instability of Fluid Jet in Microgravity”, 19-24, in Proc. of 5th Int. Symposium on Computational Fluid Dynamics (ISCFD), Aug. 31 - Sept. 3, 1993, Sendai, Japan, Japan Society of Computational Fluid Dynamics, Vol.1, 1993.
4. Belotserkovsky, 1994 Numerical Methods in Continuum Mechanics, Moscow, Fizmatlit, 2nd edition, p.441.
5. Hirt C., Nichols B. 1981, “Volume of Fluid (VOF) method for the dynamics of free boundaries”, J.Comput. Phys. v. 39, pp 201-225.