Interlinked SPH Pressure Solvers for Strong Fluid-Rigid Coupling

Appendix

Christoph Gissler

Contents

B Diagonal element computation 1
C Rendering 4

B Diagonal element computation

The goal of this document is to explain the computation of the diagonal element $b_r$ in more detail. All references that do not start with "B" are references to formulas or sections in the main manuscript. Intuitively, $b_r$ is the coefficient which relates a pressure increase of particle $r$ to a predicted density change of the same particle.

Mathematically, the diagonal element $b_r$ can be derived by extracting all the coefficients of $p_r$ from Eq. (8):

\[ s_r = \rho_r \nabla \cdot \left( \Delta t \sum_{r_r} V_{rr} K_{rr} \nabla p_{rr} \right), \]

(8 revisited)

where $r_r$ are particles of the same rigid $R$ as $r$.

As shown in Subsection 3.4.2, we can compute the right-hand side of Eq. (8) using SPH as:

\[- \sum_{r_k} V_{rk} \rho_{rk} (\mathbf{v}_{rk}^{rr} - \mathbf{v}_{rr}^{rr}) \cdot \nabla W_{rrk},\]

where $r_k$ are neighboring rigid particles of $r$ of other rigid bodies $K$. 

1
By using the formula for \( v_{rr}^T \) and \( v_{r}^T \) based on Eq. (8), we get

\[
- \sum_{r_k} V_{r_k} \rho_{r_k} \left[ \left( \Delta t \sum_{r_{kr}} V_{r_{kr}} K_{r_{kr}} \nabla p_{r_{kr}} \right) - \left( \Delta t \sum_{r_r} V_r K_{rr} \nabla p_r \right) \right] \cdot \nabla W_{rrk},
\]

(B.1)

where \( r_k \) are neighboring particles of \( r \) belonging to other rigid bodies \( K \), \( r_{kr} \) are rigid particles of the same rigid body \( K \) as \( r_k \) and \( r_r \) are rigid particles of the same rigid body \( R \) as \( r \).

As described in Subsection 3.4.2, we calculate the pressure gradient as:

\[
\nabla p_r = \rho_r \sum_{r_k} V_{r_k} \rho_{r_k} \left( \frac{p_r}{\rho_r^2} + \frac{p_{r_k}}{\rho_{r_k}^2} \right) \nabla W_{rrk},
\]

(B.2)

where \( r_k \) are neighboring rigid particles of \( r \) belonging to other rigid bodies \( K \).

Using Eq. (B.2) in Eq. (B.1), we get:

\[
- \sum_{r_k} V_{r_k} \rho_{r_k} \left[ \left( \Delta t \sum_{r_{kr}} V_{r_{kr}} K_{r_{kr}} \rho_{r_{kr}} \left( \frac{p_{r_{kr}}}{\rho_{r_{kr}}^2} + \frac{p_{r_{kr}}}{\rho_{r_{kr}}^2} \right) \nabla W_{r_{kr},r_{kr}} \right) \right]
- \left( \Delta t \sum_{r_r} V_r K_{rr} \rho_{r_r} \left( \frac{p_{r_r}}{\rho_{r_r}^2} + \frac{p_{r_r}}{\rho_{r_r}^2} \right) \nabla W_{r_r,r_r} \right) \right] \cdot \nabla W_{rrk},
\]

(B.3)

In Eq. (B.3), \( r_{kr} \) are neighboring particles of \( r_k \), which belong to other rigid bodies than \( r_{kr} \).

To get the coefficients of \( p_r \), we can remove all the pressure terms where \( p_r \) cannot occur. These are the fractions including \( p_{r_k} \) (since these only include all particles of a rigid body \( K \) of neighboring particles \( r_k \) of particle \( r \)) and \( p_{r_{kr}} \) (since these are neighboring particles belonging to other rigid bodies \( K \)).

\[
- \sum_{r_k} V_{r_k} \rho_{r_k} \left[ \left( \Delta t \sum_{r_{kr}} V_{r_{kr}} K_{r_{kr}} \rho_{r_{kr}} \left( \frac{p_{r_{kr}}}{\rho_{r_{kr}}^2} \right) \nabla W_{r_{kr},r_{kr}} \right) \right]
- \left( \Delta t \sum_{r_r} V_r K_{rr} \rho_{r_r} \left( \frac{p_{r_r}}{\rho_{r_r}^2} \right) \nabla W_{r_r,r_r} \right) \right] \cdot \nabla W_{rrk},
\]

(B.4)

We replace the remaining pressure terms with \( p_r \) and also adapt the associated
Finally, we never want that a pressure increase of particle \( r \) neighboring rigid \( K \) in Algorithm B.1 could be skipped. Instead, in Line 8, the velocity of the not necessarily important to consider the changed velocities of the neighboring increase of the same particle for since this would mean that the solver computes in praxis, we observed that for the calculation of \( b_r \) of Eq. (8) can be computed by two loops over all rigid particles as described in Subsection 3.4.2, we can compute the diagonal for all rigid particles in a single loop over all particles. Algorithm B.1 shows an overview of the necessary steps.

\[
- \sum_{r_k} V_{r_k} \rho_{r_k} \left[ \left( \Delta t \sum_{r_{kr}} V_{r_{kr}} K_{r_{kr}} \left[ \rho_{r_{kr}} \sum_{r \in r_{kr}} V_r \left( \frac{p_r}{\rho_r} \right) \nabla W_{r_k, r} \right) \right) \right. \\
- \left. \left( \Delta t V_r K_{rr} \left[ \rho_r \sum_{r_k} V_{r_k} \rho_{r_k} \left( \frac{p_r}{\rho_r} \right) \nabla W_{rr_k} \right] \right) \right] \cdot \nabla W_{rr_k}. 
\]

(B.5)

Finally, we move \( p_r \) out of the term to get the diagonal \( b_r \):

\[
b_r = - \sum_{r_k} V_{r_k} \rho_{r_k} \left[ \left( \Delta t \sum_{r_{kr}} V_{r_{kr}} K_{r_{kr}} \left[ \rho_{r_{kr}} \sum_{r \in r_{kr}} V_r \left( \frac{1}{\rho_r} \right) \nabla W_{r_k, r} \right) \right) \right. \\
- \left. \left( \Delta t V_r K_{rr} \sum_{r_k} V_{r_k} \rho_{r_k} \left( \frac{1}{\rho_r} \right) \nabla W_{rr_k} \right) \right] \cdot \nabla W_{rr_k}. 
\]

(B.6)

On a first glance, it seems computationally expensive to compute \( b_r \) using Eq. (B.6) for each rigid particle. However, similar to how the right-hand side of Eq. (8) can be computed by two loops over all rigid particles as described in Algorithm B.1 shows an overview of the necessary steps.

1: foreach particle \( r \) of rigid body \( R \) do
2: Compute gradient \( \nabla \rho^b_r = \rho_r \sum_{r_k} V_{r_k} \rho_{r_k} \frac{1}{\rho_r} \nabla W_{rr_k} \)
3: Compute linear \( \mathbf{v}^b_r \) and angular velocity \( \omega^b_r \) of \( R \) using \( \nabla \rho^b_r \)
4: Compute particle velocity \( \mathbf{v}^b_r \) using \( \mathbf{v}^b_k \) and \( \omega^b_k \)
5: foreach neighboring rigid \( K \) do
6: Compute pairwise \( \nabla \rho^b_{r_{kr}} = \rho_{r_{kr}} V_{r_{kr}} \frac{1}{\rho_{r_{kr}}} \nabla W_{r_{kr}} \) for all neighbors \( r_k \) of \( r \)
7: Compute \( \mathbf{v}^b_k \) and \( \omega^b_k \) of \( K \) using \( \nabla \rho^b_{r_{kr}} \) of all neighbors \( r_k \) of \( r \)
8: Compute divergence \( \sum_{r_k \in K} V_{r_k} \rho_{r_k} (\mathbf{v}^b_{r_k} - \mathbf{v}^b_r) \cdot \nabla W_{rr_k} \) where \( \mathbf{v}^b_{r_k} \) is computed on-the-fly using \( \mathbf{v}^b_k \) and \( \omega^b_k \)
9: \( b_r = - \sum_{r_k \in K} V_{r_k} \rho_{r_k} (\mathbf{v}^b_{r_k} - \mathbf{v}^b_r) \cdot \nabla W_{rr_k} \) for all \( K \) using sub-sums from Line 8

Algorithm B.1: Computing the diagonal element of particle \( r \) of rigid body \( R \) with neighboring rigid bodies \( K \).

In praxis, we observed that for the calculation of \( b_r \) it is in most scenarios not necessarily important to consider the changed velocities of the neighboring rigid bodies based on the pressure increase of \( r \). Accordingly, Lines 6 and 7 in Algorithm B.1 could be skipped. Instead, in Line 8, the velocity of the neighboring rigid \( K \) is assumed to be unchanged by the pressure of \( r \). This saves one loop over the neighbors of particle \( r \) during the calculation of \( b_r \).

Finally, we never want that a pressure increase of particle \( r \) leads to a density increase of the same particle for since this would mean that the solver computes
attracting forces between rigid particles that have a current density that is below their rest density. To prevent this, we clamp the computed \( b_r \) to be at least 0.

C Rendering

We would like to acknowledge the software we use to render the scenes. The surface mesh generation is done using PreonLab by FIFTY2 Technology GmbH [2019]. PreonLab was also used to render the rising sphere and moored buoys scenes. The valley scene was rendered using Houdini by Side Effects Software [2019]. All the other scenes were rendered using the Cycles renderer in Blender by the Blender Online Community [2019].

References

