Enhancements to Model-reduced Fluid Simulation

Dan Gerszewski, University of Utah
Ladislav Kavan, University of Pennsylvania
Peter-Pike Sloan, Activision
Adam W. Bargteil, University of Utah

Abstract

We present several enhancements to model-reduced fluid simulation that allow improved simulation bases and two-way solid-fluid coupling. Specifically, we present a basis enrichment scheme that allows us to combine data driven or artistically derived bases with more general analytic bases derived from Laplacian Eigenfunctions. We handle two-way solid-fluid coupling in a time-splitting fashion—we alternately timestep the fluid and rigid body simulators, while taking into account the effects of the fluid on the rigid bodies and vice versa. We employ the vortex panel method to handle solid-fluid coupling and use dynamic pressure to compute the effect of the fluid on rigid bodies.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation; I.6.8 [Simulation and Modeling]: Types of Simulation—Animation.

Keywords: Fluid simulation, model reduction, solid-fluid coupling

1 Introduction

One of the most significant drawbacks of physics-based animation is "the curse of dimensionality"—the quest for ever-higher fidelity leads to an explosion in the number of degrees of freedom. This problem naturally leads to the consideration of dimensionality reduction techniques. Dimensionality reduction was first applied to fluid simulation by Treuille and colleagues [2006], who described how each step of a fluid simulation can be performed in the reduced space. Since that work researchers have also developed modular techniques [Wicke et al. 2009], experimented with different bases [Gupta and Narasimhan 2007; Long and Reinhard 2009; De Witt et al. 2012], applied a cubature approach for non-linear functions [Kim and Delaney 2013], and even included inverse operators for solid-fluid coupling [Stanton et al. 2013].

In this short paper, we present several enhancements to the basic reduced fluid simulation pipeline. Specifically, we present a basis enrichment scheme for combining both data-driven and analytic or artistically authored bases and a new approach to two-way solid-fluid coupling that scales to a large number of rigid bodies. The analytic bases act somewhat like regularization allowing our approach to generalize outside the training data and requiring significantly less source data without the risk of over-fitting. We treat two-way solid-fluid coupling in a time-splitting fashion—we first compute the effect of the solid on the fluid and then compute the effect of the fluid on the solid. We employ a *vortex panel method* to compute obstacles' effects on the fluid and dynamic pressure to compute forces induced on the obstacle by the surrounding fluid. In precomputation, we must invert the dense "panel matrix," however, at runtime solid-fluid coupling reduces to matrix multiplications. We handle multiple obstacles by iteratively computing the coupling in a way similar to Schwarz alternating methods [Toselli and Widlund 2004]. Fluid-solid coupling is achieved using dynamic pressure to compute forces for a rigid body simulator from fluid velocities. Our results demonstrate that our enhancements are practical for two-way coupled reduced fluid simulation with rigid bodies.

2 Methods

In this section, we will first briefly review the mechanics of reduced fluid simulation, then introduce our basis enrichment scheme, and finally present our approach for two-way solid-fluid coupling.

2.1 Reduced Fluid Simulation

The basic mechanics for reduced fluid simulation were introduce by Treuille and colleagues [2006]. We begin with the incompressible Navier-Stokes equations which describe the motion of a viscous fluid,

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla)\mathbf{u} - \nu \nabla^2 \mathbf{u} + \nabla p + \mathbf{f}_e \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{2}$$

where $\mathbf{u}$ is the velocity, ν is the viscosity parameter, p is the pressure, and $\mathbf{f}_e$ are the external forces. The goal of reduced simulation is to reduce the dimensionality of $\mathbf{u}$ through Galerkin projection onto a low-dimensional basis,

$$\tilde{\mathbf{r}} = \mathbf{B}^T \mathbf{u} \tag{3}$$

where, $\tilde{\mathbf{r}} \in \mathbb{R}^r$ represents the reduced coefficients and $\mathbf{B}$ is the basis represented as matrix with r columns, each representing a basis function.

A typical fluid simulation in computer graphics employs *operator splitting* breaking the simulation into several individual steps: advection, applying external forces, applying viscosity, and projection onto a divergence-free field. To perform reduced fluid simulations, we must address each of these steps.

Fortunately, because we only include divergence free fields in our basis, we can only represent divergence free fields removing the need for the expensive projection step. External forces are easily handled by Galerkin projection onto the basis. Specifically, given external forces, $\mathbf{f}_e$, we compute reduced forces

$$\tilde{\mathbf{f}}_e = \mathbf{B}^T \mathbf{f}_e. \tag{4}$$

These are simply scaled and added to the reduced velocity coefficients,

$$\tilde{\mathbf{r}} := \tilde{\mathbf{r}} + s\tilde{\mathbf{f}}_e, \tag{5}$$

for some scaling factor s that accounts for density, grid-spacing, and timestep.

The diffusion term is also easily handled. Being a linear operator, the discretization of the diffusion operator $\nabla^2 \mathbf{u}$ can be represented

Contact email: {danger,adamb}@cs.utah.edu

as a matrix $\mathbf{D}$. Projecting into the subspace we get the reduced diffusion matrix

$$\tilde{\mathbf{D}} = \mathbf{B}^T \mathbf{D} \mathbf{B}, \tag{6}$$

which is precomputed.

The non-linear advection operator, $-(\mathbf{u} \cdot \nabla)\mathbf{u}$, is more complicated. The non-linearities preclude it from being written as a single reduced matrix. Instead, a reduced advection matrix for each basis function can be precomputed and then at runtime combined into the final reduced advection operator. The discretization of the advection operator for a given velocity field, $\mathbf{u}$ can be expressed as a matrix, $\mathbf{A_u}$. This matrix, when applied to a field, $\mathbf{v}$, (i.e $\mathbf{A_u v}$) has the effect of advecting $\mathbf{v}$ through $\mathbf{u}$.

Thus, we precompute, for each basis function or *mode*, $\mathbf{b}_i$, in the basis $\mathbf{B} = [\mathbf{b}_1 \dots \mathbf{b}_r]$ a matrix, $\mathbf{A}_{\mathbf{b}_i}$, that represents advection through the velocity field $\mathbf{b}_i$. Each of these matrices can be reduced

$$\tilde{\mathbf{A}}_{\mathbf{b}_i} = \mathbf{B}^T \mathbf{A}_{\mathbf{b}i} \mathbf{B}, \tag{7}$$

during precomputation. During simulation, the reduced advection matrix is computed by summing all mode advection matrices weighted by their corresponding reduced state coefficient

$$\tilde{\mathbf{A}} = \sum_i \tilde{\mathbf{A}}_{\mathbf{b}_i} r_i. \tag{8}$$

Viscosity and advection can be combined into a single update from time t to $t+1$ and can be written as:

$$\tilde{\mathbf{r}}^{t+1} = \left(e^{\Delta t (\nu \tilde{\mathbf{D}} + \tilde{\mathbf{A}})} \right) \tilde{\mathbf{r}}^t. \tag{9}$$

This matrix-vector product is computed efficiently using an iterative Taylor approximation [Wicke et al. 2009].

We note that while the reduced simulation can proceed without the notion of a *grid*, for collecting training data and visualization purposes a grid is useful. In our system, we explicitly use the grid for solid fluid coupling.

2.2 Basis Enrichment

The divergence free basis used in reduced fluid simulations have been constructed in either of two ways. The first method involves running a training simulation and then extracting a reduced basis using a Singular Value Decomposition (SVD). This process is accomplished by concatenating velocity-field snapshots of a high-resolution fluid simulation into a matrix, computing the SVD, and then selecting r singular vectors [Treuille et al. 2006]. However, a basis generated in this way can suffer from a number of problems. Arbitrary motion during runtime can be problematic as the basis may not generalize well to motion outside of the training simulation, e.g. a training simulation where an obstacle generates flow in one half of the domain but during runtime moves to the other half. To minimize problems from over fitting, a significant amount of simulation data has to be precomputed. Additionally, it can be difficult for artists to know what kinds of training simulations to run in order to generate a suitable basis, not to mention the large amount of pre computation space and time needed. However, motion similar to the training simulation can be represented well in the least squares sense.

The second method involves creating a basis consisting of eigenfunctions of the Laplacian operator. For a few simple domains, these can be computed analytically. In more general domains, the eigenfunctions of the discrete Laplacian operator are computed using an Eigendecomposition [De Witt et al. 2012]. In simple domains like a box, the advection operators can be computed analytically and are only loosely coupled, resulting in sparse matrices. These modes work well for gross flow and do not suffer from overfitting, but detailed flow can require an impractically large number of modes. An example of this would be a simple jet inside a box that can be turned on or off in the presence of dynamic obstacles.

To give the artist control over generating a basis, we provide a velocity drawing tool. After the velocity has been drawn, it is projected onto a divergence-free field and the artist can timestep the simulation to generate the desired velocity field. This allows the artist to create different flow effects such as vortices or laminar flow paths, with minimal training data. Alternatively, artists can simply interact with the simulation to generate training data. We will now describe how to combine both these types of bases, a similar idea has been used in the context of reduced bases for direct to indirect transfer [Loos et al. 2011].

To exploit any sparsity that might exist in the Laplacian Eigenfunctions, we would like to keep this basis intact when including the data driven, artist generated modes. Thus, given a Laplacian Eigenfunction basis, $\mathbf{E}$, and velocity fields generated by an artist, $\mathbf{D} = [\mathbf{d}_1 \dots \mathbf{d}_N]$, where each column is a user generated velocity field scaled to unit length, we would like to construct a combined basis that keeps the structure of $\mathbf{E}$ intact. First, the SVD of $\mathbf{D} = \mathbf{USV}^T$ is computed and the left singular vectors, $\mathbf{U}$, with corresponding singular values greater than zero are retained. $\mathbf{U}$ is then deflated against the basis,

$$\mathbf{U}_d = \mathbf{U} - \mathbf{E}\mathbf{E}^T \mathbf{U}, \tag{10}$$

where the columns of $\mathbf{U}_d$ now contain the parts of the velocity fields that could not be represented by the basis. The columns of matrix $\mathbf{U}_d$ are now orthogonal to the columns of $\mathbf{E}$ but may no longer be orthogonal to each other, i.e., $\mathbf{U}_d^T \mathbf{U}_d$ may not be identity. To generate a basis that spans the same subspace we simply compute the SVD of $\mathbf{U}_d$ and retain the singular vectors corresponding to non-zero singular values[1], resulting in an orthonormal basis $\mathbf{R}$. Concatenation of $\mathbf{E}$ and $\mathbf{R}$ forms an orthogonal basis, perfectly valid for reduced fluid simulation. From now on we therefore assume that $\mathbf{B}$ is the concatenated matrix $[\mathbf{E}|\mathbf{R}]$.

We would also like the ability to specifically activate the artist generated modes during runtime. If one wishes to directly excite an artist created mode during run time, the projection of those modes into $\mathbf{B}$ can be precomputed. At run time the resulting coefficients can be added to the reduced state. No projection is necessary during run time.

2.3 Two-way Solid-fluid Coupling

We use the reduced fluid simulation engine described in Section 2.1 and Box2D [2011] for rigid body simulation. To couple them we use a time splitting technique and alternately timestep each simulator while taking into account the effects of the fluid on the rigid bodies and vice versa.

2.3.1 Solid to Fluid Coupling

To account for the effect of rigid bodies on the fluid flow, we adopt a vortex panel method. This approach has two advantages over previous work. First, obstacles are not limited to a finite range of

[1] While $\mathbf{U}$ is full rank, if there is a large overlap between $\mathbf{U}$ and $\mathbf{E}$, deflation will result in a rank deficient matrix $\mathbf{U}_d$ (with zero singular values).

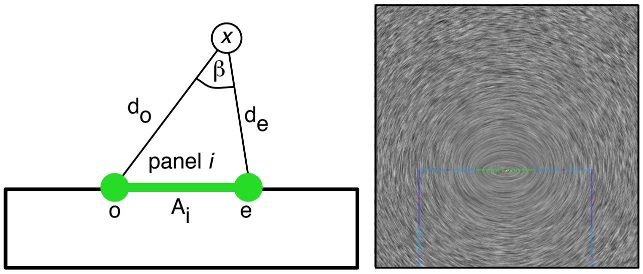

Figure 1: *Left: Panel coordinate system. Right: Velocity field induced by the panel.*

spatial influence. In fact, they have global influence, though the falloff is quite fast. Second, we avoid the substantial precomputation of sampling the object's effect at various positions and orientations in the domain. Our only precomputation involves inverting matrices. Finally, we note that our approach generalizes beyond reduced fluid simulation and could be used in other contexts, such as smoothed particle hydrodynamics, Eulerian, or semi-Lagrangian methods.

The vortex panel method was developed to study flow around airfoils [Cottet and Koumoutsakos 2000] and was introduced to graphics by Park and Kim [2005] to handle obstacles in a vortex particle method. More recent variations have been used to simulate smoke as a surface [Pfaff et al. 2012; Brochu et al. 2012].

In two dimensions, objects are discretized into M piecewise linear segments called panels. In our system, the panel lengths are chosen to be on the order of the fluid simulation's grid spacing. The panels are then used both as quadrature points and as vorticity sources that cancel flow normal to the obstacle.

The velocity generated by a panel at a point x in the local coordinate system of the panel coordinate is given by

$$u_x = \frac{\gamma\beta}{2\pi}, \quad u_y = \frac{\gamma}{2\pi} ln \frac{d_o + \epsilon}{d_e + \epsilon}, \tag{11}$$

where γ is the *panel strength*, β is the angle subtended by the panel from the point x, d_o , d_e are the distances from x to the origin and end of the panel respectively, and ϵ is a small constant to avoid division by zero (see Figure 1).

To cancel the flow normal to an object we must consider the interactions between all the panels of the object. To do so we compute a coupling matrix $\mathbf{P} \in \mathbb{R}^{M\times M}$ that encodes the influence of the strength of panel i on the velocity at panel j. Specifically, let $\bar{\mathbf{u}}_{ij}$ be the velocity induced at the mid-point of panel j by panel i when panel i has unit strength (i.e. $\gamma_i = 1$). Then the P_{ji} is given by

$$P_{ji} = -\bar{\mathbf{u}}_{ij} \cdot \mathbf{n}_j, \tag{12}$$

where $\mathbf{n}_j$ is the normal vector of the j-th panel.

Given $\mathbf{P}$ and a velocity field, $\mathbf{u}$, to cancel the flow normal to the obstacle we must solve the linear system,

$$\mathbf{P}\gamma = \mathbf{b} \tag{13}$$

where γ is the panel strength vector, and $\mathbf{b}$ is a vector encoding the violation of the boundary condition. Specifically,

$$b_i = A_i \left(\mathbf{u}_f - \mathbf{u}_o\right) \cdot \mathbf{n}_i \tag{14}$$

where b_i is the violation at panel i, A_i is the panel area, $\mathbf{u}_f$ is the fluid velocity evaluated at the midpoint of the panel, and $\mathbf{u}_o$ is the velocity of the object. This approach corresponds to a 1-point quadrature rule. Of course, higher order methods could be used.

As described, the $M \times M$ panel coupling matrix $\mathbf{P}$ is singular and an additional constraint must be added in order to obtain a unique solution. We add the constraint that there is zero circulation around the boundary, i.e.

$$\sum_i^M A_i\gamma_i = 0. \tag{15}$$

This constraint is encoded by adding an row to the panel matrix containing the panel lengths and a zero to the end of $\mathbf{b}$. The panel matrix is computed in object space, allowing for rigid body transformations without modification. $\mathbf{P}$ can be inverted during precomputation; at runtime panel strengths are computed with a single matrix-vector product.

Some distributions of panels are problematic when objects contain symmetries. For example, a square with two panels per side is unable to cancel the normal velocities induced from rigid body rotation. In such cases it suffices to use an odd number of panels per side.

Multiple Bodies Thus far we have described how to handle a single object. To handle multiple objects we must account for their interaction. Ideally, we would compute a single coupling matrix encoding the interactions of all panels in the system. However, this would require solving a new and much larger linear system every step, removing the ability to precompute an inverse [Brochu et al. 2012]. Instead, we employ a fixed point iteration approach that takes advantage of the precomputed inverse panel matrices. First, the panel strengths of each object are computed to satisfy the boundary conditions of the reduced velocity field, i.e. for all objects i we compute

$$\gamma_i = \mathbf{P}_i^{-1}\mathbf{b}_i. \tag{16}$$

We then iteratively solve for panel strengths that additionally satisfy object-object interactions.

Each iteration, for each object i in our simulation:

1. Compute $\mathbf{b}_i^{obj}$, which is the boundary violation induced by all other objects.

2. Store the previously computed panel strengths.

3. Solve for the new panel strengths,

$$\gamma_i = \mathbf{P}_i^{-1}(\mathbf{b}_i^{orig} + \mathbf{b}_i^{obj}). \tag{17}$$

4. Compute the norm of the difference in panel strengths.

Iterations are performed until the panel strengths converge, or a user specified tolerance or iteration limit is reached. This scheme, which falls into the class of Schwarz alternating methods [Toselli and Widlund 2004], is guaranteed to converge to a unique solution for second order PDE's. Golas et al. [2012] successfully demonstrate an alternating method to couple Eulerian grids with vortex particle methods.

This alternating scheme may fail due to the singularities that occur when evaluating the velocity very near a panel. Velocities evaluated too close to a panel should not be relied upon and instead another approach should be taken, such as interpolating from reliable positions [Hess and Smith 1962].

Feedback The resulting velocity field is a combination of the reduced fluid velocity, $\mathbf{u}$, and a summation over all the panel velocities and can be evaluated at any specific point in space. However, the panel strengths have no memory and are recomputed at the next timestep, thus we need to feedback their contribution into the reduced fluid simulation. In our implementation this step is accomplished by iterating over the panels and summing their contribution to the background grid. The resulting velocity field is projected into the reduced space and added to the reduced coefficients.

2.3.2 Fluid to Solid Coupling

We incorporate fluid to solid coupling by computing the dynamic pressure on the boundary of the rigid body. From the dynamic pressure we compute the force, which is then added to the rigid body simulation. The dynamic pressure, sometimes called the velocity pressure, is

$$q = \frac{1}{2}\rho\mathbf{u}^T\mathbf{u}, \tag{18}$$

where ρ is the density of the fluid, and $\mathbf{u}$ is the fluid velocity. For each panel we have already computed the difference in relative velocity between the obstacle and fluid when solving for the panel strengths. From that velocity, we compute the dynamic pressure q at panel centers and then multiply by the panel area to get forces [Saffman 1995], which are normal to the panels. Specifically, the force on panel i is

$$\mathbf{f_i} = A_i q \mathbf{n}_i \tag{19}$$

are then applied to the rigid body at the panel centers.

Buoyancy forces can optionally be included with

$$\mathbf{f}_i = -\rho A_i h_i g \mathbf{n}_i, \tag{20}$$

h_i is the depth of the panel center and where g is the scalar gravitational constant. The minus sign is to signify that the force is in the direction opposite the surface normal of the panel.

3 Results

In our first example, we have a single data driven mode with 63 eigenmodes. The artist input and pressure projected mode are shown in Figure 2. The eigenmodes poorly capture this "jet," but represent gross flow well, while our enhanced basis is able to capture the jet well, see Figure 3.

Our second example is of two pairs of falling objects, each pair has one object above the other. After being released, the objects above catch up to the objects below closing the gap between them. The objects that start out above, draft off of the objects below allowing them to fall faster through the fluid demonstrating the effects of solid-fluid coupling and object-object interaction, see Figure 4.

Finally, we have combined both our basis enhancement and two-way coupling into a simple 2D game, see Figure 5. The game uses 64 eigenmodes and there are 15 objects with a total of 147 panels. Timing results in Table 1 show that feedback from the panel velocities back to the reduced simulation dominates timing, taking $22ms$ in this example. This is an obvious area for future work. While we could precompute the coupling to the reduced space by aggressively sampling as others have done [Treuille et al. 2006], a more promising direction is to precompute a coupling to a hierarchical basis.

Description	Time (ms)
Advect	0.48
Diffuse	0.00385
Panel Solves	5.994
Panel Feedback	22.468

Table 1: *Timings in ms for game scene with 64 modes on a 65x65 staggered grid.*

4 Conclusion and Future Work

We have presented several enhancements to prior work on dimensionally reduced fluids simulations: An enrichment scheme to mix data-driven and analytic modes, and a new approach to two-way solid-fluid coupling through the use of vortex panel methods and dynamic pressure. Our enrichment scheme combines the generality of Eigenmodes with data-driven modes ability for art direction. The vortex panel method enables more robust coupling of dynamic objects to the gross flow, and requires no training data.

In future work, we will investigate acceleration of the coupling between the panel methods and the reduced fluid and extend the technique to 3D.

Acknowledgments

The authors wish to thank the anonymous reviewers for their helpful comments. This work was supported in part by a gift from Adobe Systems Incorporated and National Science Foundation Awards CNS-0855167, IIS-1249756, and IIS-1314896.

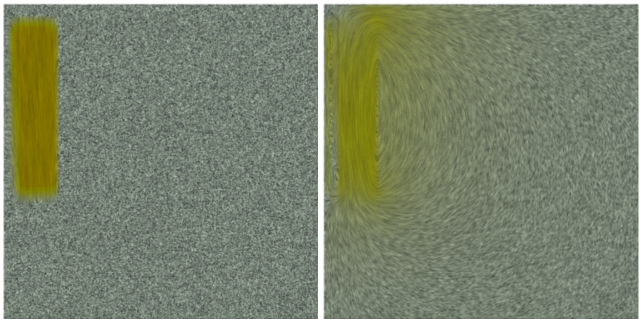

Figure 2: *Input on the left, pressure projected data driven mode on the right.*

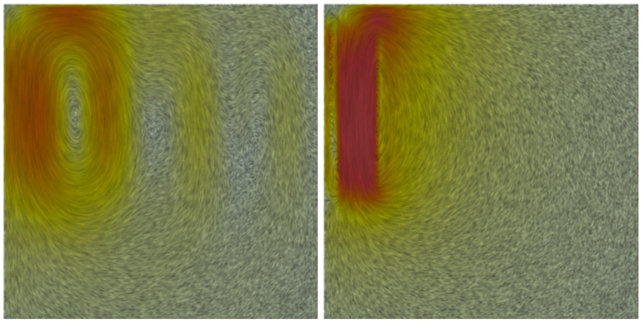

Figure 3: *Left: Only Eigenmodes. Right: a data driven mode with Eigenmodes. Exciting the jet with high intensity the induced flow is not well represented just using the Eigenmodes.*

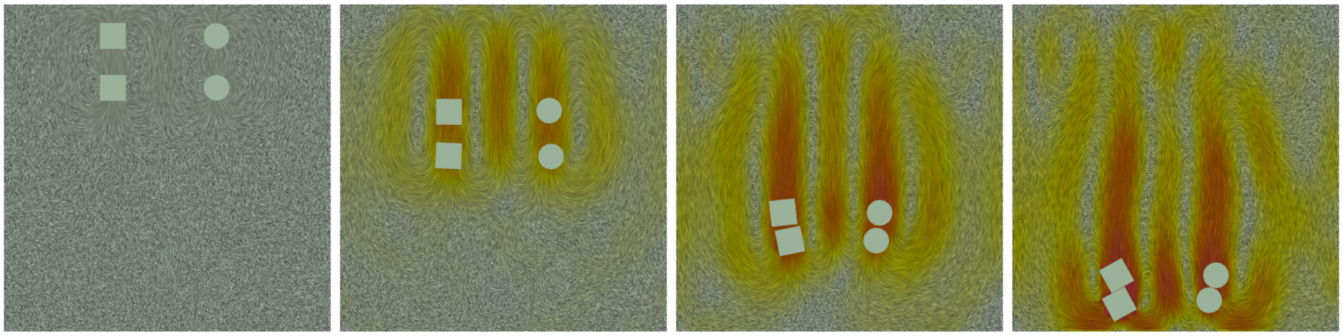

Figure 4: *Drafting example: solid-fluid coupling and object-object interactions.*

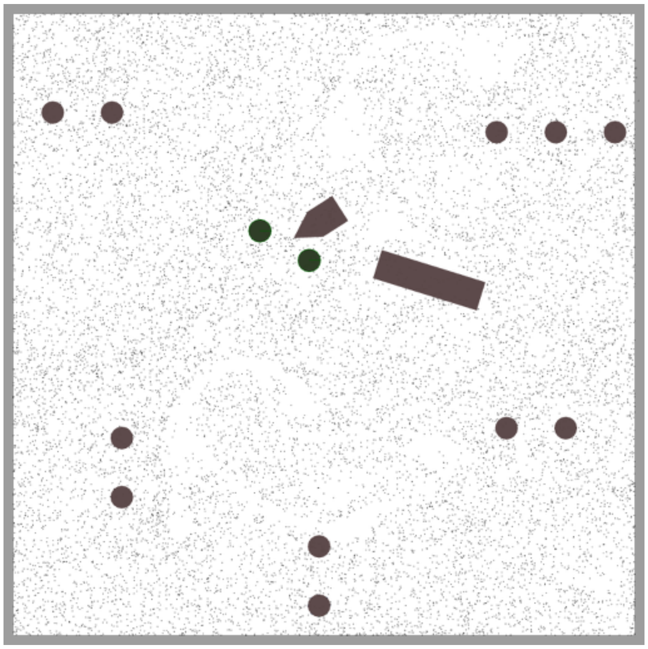

Figure 5: *Image from game using our system, see the video.*

References

BOX2D. 2011. Erin Catto.

BROCHU, T., KEELER, T., AND BRIDSON, R. 2012. Linear-time smoke animation with vortex sheet meshes. In *Proc. of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 87–95.

COTTET, G.-H., AND KOUMOUTSAKOS, P. 2000. *Vortex methods: Theory and practice*. Cambridge University Press, June.

DE WITT, T., LESSIG, C., AND FIUME, E. 2012. Fluid simulation using laplacian eigenfunctions. *ACM Trans. Graph. 31*, 1 (Feb.), 10:1–10:11.

GOLAS, A., NARAIN, R., SEWALL, J., KRAJCEVSKI, P., DUBEY, P., AND LIN, M. 2012. Large-scale fluid simulation using velocity-vorticity domain decomposition. *ACM Trans. Graph. 31*, 6 (Nov.), 148:1–148:9.

GUPTA, M., AND NARASIMHAN, S. G. 2007. Legendre fluids: a unified framework for analytic reduced space modeling and rendering of participating media. In *Proc. of the ACM SIGGRAPH/Eurographics symposium on Computer animation*, 17–25.

HESS, J., AND SMITH, A. 1962. Calculation of non-lifting potential flow about arbitrary three-dimensional bodies. Tech. Rep. E.S. 40622, Douglas Aircraft Division.

KIM, T., AND DELANEY, J. 2013. Subspace fluid re-simulation. *ACM Trans. Graph. 32*, 4 (July), 62:1–62:9.

LONG, B., AND REINHARD, E. 2009. Real-time fluid simulation using discrete sine/cosine transforms. In *Proc. of the symposium on Interactive 3D graphics and games*, 99–106.

LOOS, B. J., ANTANI, L., MITCHELL, K., NOWROUZEZAHRAI, D., JAROSZ, W., AND SLOAN, P.-P. 2011. Modular radiance transfer. *ACM Trans. Graph. 30*, 6 (Dec.), 178:1–178:10.

PARK, S. I., AND KIM, M. J. 2005. Vortex fluid for gaseous phenomena. In *Proc. of the ACM SIGGRAPH/Eurographics symposium on Computer animation*, 261–270.

PFAFF, T., THUEREY, N., AND GROSS, M. 2012. Lagrangian vortex sheets for animating fluids. *ACM Trans. Graph. 31*, 4 (July), 112:1–112:8.

SAFFMAN, P. 1995. *Vortex Dynamics*. Cambridge University Press.

STANTON, M., SHENG, Y., WICKE, M., PERAZZI, F., YUEN, A., NARASIMHAN, S., AND TREUILLE, A. 2013. Non-polynomial galerkin projection on deforming meshes. *ACM Trans. Graph. 32*, 4 (July), 86:1–86:14.

TOSELLI, A., AND WIDLUND, O. 2004. *Domain Decomposition Methods - Algorithms and Theory*. Springer.

TREUILLE, A., LEWIS, A., AND POPOVIĆ, Z. 2006. Model reduction for real-time fluids. *ACM Trans. Graph. 25*, 3 (July), 826–834.

WICKE, M., STANTON, M., AND TREUILLE, A. 2009. Modular bases for fluid dynamics. *ACM Trans. Graph. 28*, 3 (July), 39:1–39:8.