Modelling Effects of Wind Fields in Cloth Animations

M. Keckeisen, S. Kimmerle, B. Thomaszewski, and M. Wacker
WSI/GRIS
University of Tübingen
Sand 14,
D-72076 Tübingen, Germany
{keckeisen, kimmerle, wacker}@gris.uni-tuebingen.de
b.thomaszewski@t-online.de

ABSTRACT

In this paper we show how to incorporate effects of wind fields in cloth animations. We discuss two different approaches to model force fields describing air motion and show how these models can be augmented to exhibit interaction with deformable thin objects such as textiles. The first model is based on the Navier-Stokes equations, while the second method extends simple particle tracing methods by the effect of lee. In each case, we present a method for simulating the interaction of cloth movements with the wind field. Both methods have been integrated in an existing cloth simulation system, and we compare their respective advantages and disadvantages.

Keywords
Wind Simulation, Cloth Animation, Physically Based Modelling, Computer Graphics.

1. INTRODUCTION

While the simulation of wind is an area of vast interest in aerodynamic engineering, computational fluid dynamics (CFD), and animation/visualisation of fluids in computer graphics, it has been a rather abandoned subject in simulation of deformable objects such as cloth simulation. This is mostly due to the fact that conventional CFD applications require enormous computational power. However, aerodynamic effects are obviously capable of enhancing the realism of an animated scene and thus are an important part of a cloth simulation system. For example, air resistance is a vital component which cannot be neglected if realistic animation is desired [BTH+03].

In this work we discuss two different approaches to model force fields describing air motion and show how these forces can be applied to generate aerodynamic effects on textiles. The first method is based on modelling air flows with the Navier-Stokes equations. With this model, the effect of wind fields on smoke has already been investigated [FSJ01], and in this paper we extend this approach to wind effects on textiles.

However, the solution of the sophisticated Navier-Stokes equations is not desired and not necessary in all situations where wind effects shall be integrated into cloth simulations. For this case, we present a much simpler model based on tracing wind particles that move along a global force field. During the simulation, the wind field is evaluated for each wind particle at its current position. The so determined flow field is used to compute the wind force which is added to the forces that act on the textiles. By detecting collisions between the wind particles and objects in the scene, we are able to simulate the important effect of lee even with this straightforward method.

Of course, both approaches have different characteristics and aim at different applications for wind simulation. In this work, we compare the models' advantages and disadvantages and show practical results of the described methods.

The rest of this paper is organized as follows. After giving an overview of related work in section 2,

Journal of WSCG, Vol.12, No.1-3., ISSN 1213-6972
WSCG 2004, February 2-6, 2003, Plzen, Czech Republic.

we describe the aerodynamic force model applied in this work in section 3. In section 4 we discuss the two wind field models, namely the one based on the Navier-Stokes equations in 4.1, and the one based on tracing wind particles in 4.2. Finally, we show some practical results of the implemented algorithms in section 5, before we conclude the paper in section 6.

2. PREVIOUS WORK

Models for fluid dynamics can be essentially subdivided into two categories. Simple models which are commonly used in most computer graphics applications describe the wind flow by predefined flow functions. Here, global functions are defined to model the velocity of wind. Either special flow primitives can be combined [WH91; LDG96; Li00] or visually pleasing functions introducing random turbulence [SF92; SF93; Sta97] are taken into account to model even complex wind scenes. Many models use this method to move objects in the wind field through the scene [Ree83; Sim90; WH91; BLM95]. In addition, physically based fluid dynamics solving equations of motions with particle methods were presented recently [IK03].

However, fixed flow functions lack interaction with the user or objects in the scene. Hence, with increasing computer power, computer graphics concentrates on physically more accurate simulations. In many fields the Navier-Stokes equations are the standard mathematical formulation to model fluid dynamics. A vast literature exists on how to solve these equations numerically. CFD are applied in this field for engineering tasks with a high degree of quality requirements. Unfortunately, it is quite difficult to apply these algorithms in computer graphics due to enormous calculation times. Hence, faster fluid solvers were investigated for computer graphics applications. Kajiya et al. [KvH84], Yaeger et al. [YU86], and Gamito et al. [GLG95] worked on fluid dynamics solvers in two dimensions, and many improvements and variants followed [CdVLHM97; KM90]. Foster and Metaxas [FM96; FM97], and Griebel et al. [GDN98] presented a solver for the fully three dimensional Navier-Stokes equations. Due to explicit integration methods very small step sizes had to be used. To enable faster simulations, a solution with an unconditionally stable solver was introduced by Stam [Sta99] and further extended in [FSJ01; Sta01; Sta03]. Modelling interaction of fluids with solid objects has been investigated by Takahashi et al. [TFK+03] and Génevaux et al. [GHD03]. Recently, Wei et al. [WZF+03] presented an interesting approach to simulate lightweight objects like soap bubbles and feathers in a wind flow using a Lattice Boltzmann Model extended with a subgrid model.

For interaction of highly deformable objects and especially cloth-like objects only few models have been investigated. Simple models consist in the calculation of lift and drag forces from the surrounding velocity field [SF92; Pro95; KCC+00; KCCL01]. More complex interaction models calculate the wind force by a panel method [LDG96; Li00] introducing local vortices.

In this work, we show how recent results in fluid dynamics for computer graphics can be exploited to simulate interaction of wind flows with textiles. Moreover, we extend the more straightforward approach of global wind field functions by the effect of lee.

3. AERODYNAMICS

To incorporate wind effects in a physically based animation we have to apply additional external forces in the dynamical model of the deformable objects. Hence, given a wind flow represented by a velocity field in the scene we calculate the forces which are exerted on the simulated objects. In this section, we briefly describe the model we use to compute the effective aerodynamic forces such as wind force and air resistance, mainly following [SF92].

The wind force acting on objects in an air stream is decomposed into two components: the lift force $\mathbf{F}_L$ and the drag force $\mathbf{F}_D$ (see figure 1).

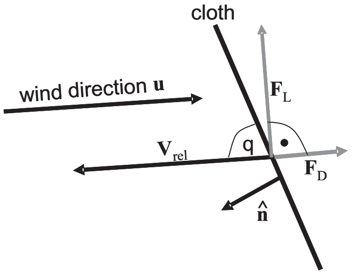


Figure 1: The decomposition of wind forces (side view).

The direction of the drag force $\mathbf{F}_D$ is diametral to the relative velocity $\mathbf{v}_{\mathrm{rel}} = \mathbf{v}_{\mathrm{object}} - \mathbf{u}$, where $\mathbf{v}_{\mathrm{object}}$ is the object's velocity and $\mathbf{u}$ is the velocity field of the wind. Note that in the case of a windless situation, i.e. $\mathbf{u} = 0$, we still have air resistance for moving objects. Since two-dimensional objects do not exhibit an inside and outside, the unit normal $\hat{\mathbf{n}}_i$ of the i-th face of the object mesh (cf. figure 1) is replaced by

$$\tilde{\mathbf{n}}_i = \begin{cases} \hat{\mathbf{n}}_i & \text{if } \mathbf{n}_i \cdot \mathbf{v}_{i,\mathrm{rel}} > 0 \,, \\ -\hat{\mathbf{n}}_i & \text{else} \,. \end{cases} \qquad (3.1)$$

The drag force per face is then given by

$$\mathbf{F}_{i,D} = \frac{1}{2} C_D \rho |\mathbf{v}_{i,\mathrm{rel}}|^2 A \cdot (\tilde{\mathbf{n}}_i \cdot \hat{\mathbf{v}}_{i,\mathrm{rel}}) \cdot (-\hat{\mathbf{v}}_{i,\mathrm{rel}}) \,,$$

where C_D is the specific air resistance coefficient, ρ the density of air, A is the area of the corresponding face, and $\hat{\mathbf{v}}_{i,\mathrm{rel}}$ the unit relative velocity vector of the face.

The direction of the lift force, which is perpendicular to $\mathbf{v}_{i,\mathrm{rel}}$ and lies in the plane spanned by $\mathbf{v}_{i,\mathrm{rel}}$ and $\tilde{\mathbf{n}}_i$, is given by

$$\hat{\mathbf{u}}_i = (\tilde{\mathbf{n}}_i \times \hat{\mathbf{v}}_{i,\mathrm{rel}}) \times \hat{\mathbf{v}}_{i,\mathrm{rel}} \; .$$

Then the lift force is calculated as

$$\mathbf{F}_{i,L} = \frac{1}{2} C_L \rho |\mathbf{v}_{i,\mathrm{rel}}|^2 A \cos\theta \cdot \hat{\mathbf{u}}_i \; ,$$

where C_L is the lift force coefficient, and θ is the angle between $\mathbf{v}_{i,\mathrm{rel}}$ and the actual face.

4. WIND FIELD MODELS

In this section we describe two different wind field models and show how they can be used to model wind effects on textiles. The first model is based on the work of Stam [Sta97] and calculates the numerical solution of the Navier-Stokes equation with a semi-Lagrangian approach. This model is extended to interaction of the wind flow with textiles. The second model employs precomputed wind flows and particle tracing methods. This approach is much easier to implement and can be added to existing simulation modules without additional computational cost. In section 4.2, we show how to produce realistic effects of wind on textiles including lee effects.

4.1 The Navier-Stokes equations

The Navier-Stokes equations describe a precise mathematical model for fluid flows. The numerical algorithms used in CFD to solve these equations are designed for physical accuracy for engineering applications and are expensive in computation. But in our case where this precision is not necessary simplifications can be made which greatly reduce the computation costs as described by Stam [Sta03]. Since the arising wind velocities are clearly below the speed of sound, compressibility effects are negligible, and the wind is modelled as an incompressible constant density fluid. This notably simplifies the numerical approximation, and the incompressible Navier-Stokes equations can be written in a compact vector notation as

$$\nabla \cdot \mathbf{u} = 0, \tag{4.1}$$

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla)\mathbf{u} - \frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}. \tag{4.2}$$

Here, $\mathbf{u}$ describes the (three-dimensional) velocity field, ν is the kinematic viscosity of the fluid, ρ its density, p the pressure in the wind field, and $\mathbf{f}$ accounts for external forces. The first equation states that the velocity field should be incompressible while the second one describes the evolution of a velocity field over time. The first term on the right hand side reflects the change of velocity due to advection, while the second expression accounts for any external force $\mathbf{f}$ and acceleration caused by the local pressure gradient ∇p and by viscous drag depending on ν.

4.1.1 The solution of the Navier-Stokes equations

In the following we briefly describe the numerical solution of the Navier-Stokes equations (4.1) and (4.2) by Stam [Sta97]. To solve these equations numerically they first have to be discretised. For this, the computational domain is diced up into equally sized cubes forming a grid as described in section 4.2.2, and sample values of velocity and pressure are defined at the cell centres. Foster and Metaxas [FM96] use a finite difference approximation for the discretisation of the operators in equation (4.2). Then they update the cell's velocities according to the divergence value computed in each direction, respectively, using an explicit integration scheme. Since time steps in explicit computations usually need to be very small, we follow Stam [Sta99] who proposes an implicit integration scheme, which allows stable simulations with large time steps. While the linear terms in equation (4.2) are straightforward to solve implicitly, the term $-(\mathbf{u}\cdot\nabla)\mathbf{u}$ is nonlinear and deserves special attention. Here, a different approach based on the method of characteristics is used to solve the advection equation. Equation (4.2) does not provide a divergent-free velocity field. Therefore, the divergence of each cell in the grid has to be projected to zero using the Helmholtz-Hodge decomposition [Sta03].

4.1.2 Wind effects on clothes

The major advantage of Navier-Stokes based approaches consists in the fact that the evolution of the wind flow over time is calculated. It enables us to model global effects like convection and diffusion on a physical basis. We present a model to exploit these wind models for calculating the interaction of deformable objects with the air flow by a boundary condition method.

As already stated by Stam [Sta03] “a velocity field of its own isn't really visually interesting until it starts moving objects [...]”. That means in particular that all objects in the scene interact with the fluid present in it, i.e. in our case clothes with the wind. On the one hand the wind deforms the objects which on the other hand change the wind flow. To describe the above situation by a physical model we require the Neumann boundary condition

$$\frac{\partial \mathbf{u}}{\partial \mathbf{n}} = 0$$

to be satisfied for the wind flow $\mathbf{u}$ at any boundary point of an object with normal $\mathbf{n}$. Rigid objects like walls will influence the fluid field but will not be affected by fluid forces themselves. Deformable objects like cloth are supposed to both experience fluid forces and itself influence the fluid flow. This in fact is a major difficulty. Consider a point $\mathbf{p}_b$ on the boundary of a deformable object in the scene. Let $\mathbf{u}(\mathbf{p}_b)$ be the corresponding wind velocity at that point and $\mathbf{n}$ be its normal. On the one hand, we want the Neumann boundary condition $\mathbf{u}(\mathbf{p}_b) \cdot \mathbf{n} = 0$ to be satisfied. On the other hand, the wind velocity orthogonal to the object's surface is just what causes the aerodynamic forces. Without further remedial action setting the boundary according to the Neumann condition would mean that the fluid will not exert forces on the objects.

Here we propose a method which meets both requirements. For every deformable object the velocity value of the surrounding wind field for every vertex of the representing mesh is tracked. In the solver's cycle, the boundary conditions are then set prior to any other operation: For every object in the scene each triangle of its representing mesh is registered in the fluid grid, which means that the cell of the wind field occupied by the object is marked as occupied. The wind velocity at the vertex positions of the object is recorded. Additionally, the normals of these vertices are stored. Then, the aerodynamic forces as described in section 3 are calculated. Finally, for every marked cell in the scene the previously stored normals are averaged in one space cell which are used to update the velocity at the cell to satisfy the Neumann boundary condition. Thus, the boundary conditions are met and yet aerodynamic forces are obtained.

A different issue is how to deal with the inside of (rigid) objects. The method to set boundary conditions as described above does not account for the interior of objects. Thus, a nonzero velocity could be mistakenly assigned to cells lying inside an object. To avoid this situation, the path of the wind flow is checked for object intersection, whereby the collision detection of the cloth simulation system provides a simple method to deal with this issue [MKE03].

4.2 Particle Tracing on Wind Fields

Here we combine the idea of creating wind fields by predefined flow primitives with particle tracing in given flow fields. To define a wind scene we first built up the air flow by simple primitives such as parallel directed wind fields, vortices, etc. We then use a particle tracing method in the defined wind field to determine the effect of lee by detecting windless areas. This method is very easy to implement and yields very plausible and nicely looking results.

4.2.1 Global Wind Fields

A simple approach to generate complex air flows is to define a wind field by mathematical functions which assign to each point in space a unique velocity value. As Wejchert et al. [WH91] have shown, this already enables an animator to design even complex wind fields: Assuming an irrotational ($\nabla \times \mathbf{u} = 0$), inviscid, and incompressible ($\nabla \cdot \mathbf{u} = 0$) fluid, the Navier-Stokes equations which describe the mechanics of any fluid (see section 4.1) can be simplified to give the Laplace equation

$$\nabla \cdot \mathbf{u} = \nabla \nabla \phi = \nabla^2 \phi = 0, \qquad (4.3)$$

where ϕ is the potential of the given wind field. Thus, the velocity field $\mathbf{u}$ is given by

$$\mathbf{u} = \nabla \phi \, .$$

The linearity of equation (4.3) enables an animator to combine basic flows which satisfy equation (4.3) as he likes and thus to obtain complex fluid flows. Some primitives common to fluid simulations are depicted in figure 2.

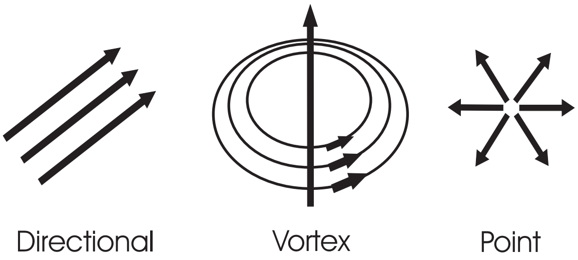


Figure 2: Flow primitives

One drawback of this model is that it cannot handle objects exhibiting complex boundaries. The approach to model solid objects in the scene taken by Wejchert et al. consists in placing a wind source using a mirror principle in order to extinguish the air flow at the boundary of the object. While this works for simple objects this approach is not feasible at all with deformable objects like textiles.

Another more serious drawback of this model for our application consists in the lack of interaction with objects. The wind flow defined by the primitives will not react on objects in the scene which means for example that tissues in the lee of other objects will be affected by the wind flow as well.

However, this method can be combined with the aerodynamic model described in section 3 to give nice and fast results as will be shown in section 5. To solve the described problems we propose a model which combines the simple global wind flow techniques with a particle tracing method. Here, particles are moved along the wind field to determine the effect of objects in the scene.

4.2.2 Particle Tracing

This model divides the scene into parallelepiped cells. There are two common approaches to discretising the continuous velocity field defined in space: one can either choose the midpoint of a cell [Sta99] or its six faces [FM96] to define the field. As usual, values between the defining points of the grid are interpolated using trilinear functions.

The basic idea of the particle tracing method is to trace wind particles through a field $\mathbf{w} = \sum_i \mathbf{w_i}$ defined by linear superposition of wind sources corresponding to flow primitives with respective velocity fields $\mathbf{w_i}$. The field $\mathbf{w}$ does not account for lee effects caused by objects in the flow. Therefore we compute the wind field $\mathbf{u}$ containing these effects as follows. In our model every wind source is also a particle source: These particles form an uncoupled particle system which can be considered as a wind gust. The wind particles are emitted into the velocity field $\mathbf{w_i}$ of the corresponding wind source which is defined on a grid. The specific emission intervals and amounts depend on the properties of the flow sources.

In every time step each particle in a wind gust moves along its velocity field $\mathbf{w_i}$ defined by the corresponding wind source. Notice that the movement of the particles in a wind gust is only affected by the wind source they belong to. The global superposition of all wind sources has no effect on these particles. To calculate the wind particles' positions we used the explicit Euler integration scheme. For a wind particle at position $\mathbf{p}_t$ and time t this results in a path $\mathbf{s}(\mathbf{p}_t, \mathbf{p}_{t+\Delta t})$, where $\mathbf{p}_{t+\Delta t}$ denotes the position after time step Δt according to

$$\mathbf{p}_{t+\Delta t} = \mathbf{p}_t + \mathbf{w_i}(\mathbf{p}_t, \Delta t) \; .$$

As a particle moves along its path in space, all grid cells colliding with the path are updated with the velocity of the associated wind source with respect to the position of the particle. The particle might cross several grid cells on its way during a single time step. If this is the case, the path of the particle has to be subdivided into parts not exceeding the size of a grid cell. This path is then tested for collisions with the objects in the scene. The velocity field $\mathbf{u}$ is then computed as

$$\mathbf{u} = \sum \mathbf{w_i}$$

for each grid cell separately, where $\mathbf{w_i}$ are all those wind sources whose particles have reached the cell.

If a collision is detected at position $\mathbf{p}_{col}$ the normal of the colliding object $\mathbf{n}_{obj}(\mathbf{p}_{col})$ is determined and the velocity of the particle is set to

$$\mathbf{w_i}(\mathbf{p}_{\mathrm{col}}, t{+}\Delta t) = \mathbf{w_i}(\mathbf{p}_{\mathrm{col}}, t) - (\mathbf{n}_{\mathrm{obj}} \cdot \mathbf{w_i}(\mathbf{p}_{\mathrm{col}}, t)) \cdot \mathbf{n}_{\mathrm{obj}} \; .$$

This assures that the velocity component of the resulting field $\mathbf{u}$ is orthogonal to the collision object's surface at $\mathbf{p}_{col}$ is zero, i.e.

$$\mathbf{u}(\mathbf{p}_{col}, t + \Delta t) \cdot \mathbf{n}_{obj} = 0 \; ,$$

and thus no flow propagates through the object.

The wind force effective on objects in the scene is then computed from the velocity field $\mathbf{u}$. Since $\mathbf{u}$ is determined using the wind particles, every point $\mathbf{p}$ that could not be reached by any wind particle will hold zero velocity even if $\mathbf{w}$ may hold a nonzero velocity. Thus, this method solves the problems described in section 4.2.1.

Note that the somewhat tempting simplification of tagging each cell to either have wind in it or not is not valid. Imagine the simple scene in which there are two directional wind sources with opposite wind directions. Let them further have equal velocity magnitude and no distance attenuation. If we now place a solid object in between these two sources a rather undesired effect would occur using this simplification: on both sides of the solid object all cells would be tagged as having wind. But evaluating the wind field at every cell we would obtain a zero velocity. This is due to the extinguishing effect of the superposition of the two wind sources. Therefore, it is crucial for the particles to have the associated velocity of their wind source and not just the velocity resulting from the global superposition of all wind sources.

4.3 Comparison

In this section we comment on the different models described in this paper. For physically accurate simulations based on the common method in fluid dynamics the model introduced by Stam produces realistic effects which global wind field models can never achieve. It produces nice swirls and vortices derived from dynamical characteristics of the fluid. However implementing the fluid solver is quite complex and using a high grid resolution is computationally expensive. Hence, the global wind field model is better suited for an easy to implement tool which is easy to adapt to specific situations. Particle systems are very common in the simulation engines and most functionality can be adapted to integrate the proposed wind model. Even with this straightforward approach, nice, realistic looking results can be achieved which is illustrated in the next section.

5. RESULTS

In this section we present some practical results of the previously described methods. We implemented the wind models described in sections 4.1 and 4.2 in a cloth animation system that employs a fast finite element method to simulate the drape of textiles with measured material properties [EKS03]. For collision detection (between deformable objects, rigid objects, and wind particles) we use k-DOP hierarchies as described in [MKE03].

Figure 3 shows a flag blowing in the wind. We used the particle tracing method described in section 4.2 to model the effects of a directional wind field on the flag.

In figures 4 and 5 we show the ability of both the Navier-Stokes equations model (section 4.1) and the particle tracing method (section 4.2) to model the effect of lee. Two flags are exposed to a wind field, but the wind is blocked by a wall, so one of the flags is not affected by the wind.

The images in figure 6 show a character wearing a shirt and standing in a wind stream coming from the front. Images (a) and (b) are snapshots from the beginning of the animation, images (c) and (d) show the result after the wind field has affected the clothes. To show the improved realism when simulating lee effects, we let the the wind act on all polygons of the shirt on the right (no lee effect). For the shirt on the left we used the Particle Tracing Method to simulate lee effects which, we think, gives more realistic results (see also the accompanying video).

6. CONCLUSIONS

We presented two models for including advanced wind effects into cloth simulation systems. The first concentrates on physically accurate computations using a semi-Lagrangian approach to solve the Navier-Stokes equations, the second model incorporates a particle tracing method for global wind fields. As illustrated in the previous section both methods produce realistic looking results which are capable of enhancing the realism of computer animations. While the first model has a wider range of applications, the second one provides an easy method which still delivers realistic effects such as air resistance and lee.

All methods described in this work should be easy to extend to three-dimensional deformable objects. All the methods apply the same except for simple changes. Since three-dimensional objects define an inner and outer part, the adaption of the face normals in equation (3.1) is not necessary. Moreover, in the wind field computation care has to be taken that no wind field is present in the object. Here, the same method as described for rigid objects (section 4.2.2) can be applied.

7. Acknowledgements

This work was partially supported by the bmb+f research project Virtual Try-On (www.VirtualTryOn.de).

8. References

[BLM95] B. Becker, D.A. Lane, and N. Max. Unsteady Flow Volumes. In *Proc. of Visualization*, 1995.

[BTH$^+$03] K. S. Bath, C. D. Twigg, J. K. Hodgins, P. K. Khosla, Z. Popovic, and S. M. Seitz. Estimating Cloth Simulation Paramters from Video. In *Proc. SIGGRAPH Symposium on Computer Animation*, 2003.

[CdVLHM97] J.X. Chen, N. da Vittorio Lobo, C.E. Hugues, and J.M. Moshell. Real-Time Fluid Simulation in a dynamic Virtual Environment. *IEEE Computer Graphics and Applications*, pages 52–61, 1997.

[EKS03] Olaf Etzmuß, Michael Keckeisen, and Wolfgang Straßer. A Fast Finite Element Solution for Cloth Modelling. *Proc. Pacific Graphics*, 2003.

[FM96] Nick Foster and Dimitri Metaxas. Realistic animation of liquids. *Graphical models and image processing: GMIP*, 58(5):471–483, 1996.

[FM97] Nick Foster and Dimitri Metaxas. Modeling the Motion of a Hot, Turbulent Gas. In *Proc. SIGGRAPH Symposium on Computer Animation*, pages 181–188, 1997.

[FSJ01] Ronald Fedkiw, Jos Stam, and Henrik Wann Jensen. Visual simulation of smoke. In *Computer Graphics (Proc. SIGGRAPH)*, pages 15–22, 2001.

[GDN98] M. Griebel, T. Dornseifer, and T. Neunhoeffer. *Numerical Simulation in Fluid Dynamics: A Practical Introduction*. SIAM, Philadelphia, 1998.

[GHD03] O. Génevaux, A. Habibi, and J.-M. Dischler. Simulating Fluid-Solid Interaction. In *Graphics Interface*, pages 31–38, 2003.

[GLG95] M.N. Gamito, P.F. Lopez, and M.R. Gomes. Two-dimensional Simulaton of Gaseous Phaenonemna Using Vortex Particles. In *Proc. of the 6th Eurographics Workshop on Computer Animation and Simulation*, pages 3–15, 1995.

[IK03] T. Ilmonen and J. Kontkanen. The Second Order Particle System. *Journal of WSCG*, 11(1), 2003.

[KCC$^+$00] Young-Min Kang, Jeong-Hyeon Choi, Hwan-Gue Cho, Do-Hoon Lee, and Chan-Jong Park. Real-time Animation Technique for Flexible and Thin Objects. In *Journal of WSCG*, pages 322–329, 2000.

[KCCL01] Y.-M. Kang, J.-H. Choi, H.-G. Cho, and D.-H. Lee. An efficient animation of wrinkled cloth with approximate implicit integration. *The Visual Computer*, 17(3), 2001.

[KM90] M. Kass and G. Miller. Rapid, Stable Fluid dynamics for Computer Graphics. In *Proc. SIGGRAPH Symposium on Computer Animation*, pages 49–57, 1990.

[KvH84] J.T. Kajiya and B.P. von Herzen. Ray Tracing Volume Densities. In *Proc. SIGGRAPH Symposium on Computer Animation*, pages 165–174, 1984.

[LDG96] L. Li, M. Damoran, and R. K. Gay. Aerodynamic force models for animating cloth motion in air flow. *The Visual Computer*, 12, 1996.

[Li00] L. Li. *Cloth Modeling and Animation*, chapter Aerodynamic Effects, pages 175–195. A.K. Peters, 2000.

[MKE03] Johannes Mezger, Stefan Kimmerle, and Olaf Etzmuß. Hierarchical Techniques in Collision Detection for Cloth Animation. *Journal of WSCG*, 11(2):322–329, 2003.

[Pro95] Xavier Provot. Deformation Constraints in a Mass-Spring Model to Describe Rigid Cloth Behavior. In *Graphics Interface '95*, pages 147–154, 1995.

[Ree83] W. T. Reeves. Particle Systems — A Technique for Modeling a Class of Fuzzy Objects. In *Computer Graphics (Proc. SIGGRAPH)*, pages 359–376, 1983.

[SF92] M. Shinya and A. Fournier. Stochastic Motion: Motion under the Influence of Wind. *Computer Graphics Forum*, 58(5):119–128, 1992.

[SF93] Jos Stam and Eugene Fiume. Turbulent Wind Fields for Gaseous Phenomena. *Computer Graphics (Proc. SIGGRAPH)*, pages 369–376, 1993.

[Sim90] Karl Sims. Particle animation and rendering using data parallel computation. In *Computer Graphics (Proc. SIGGRAPH)*, pages 405–413, 1990.

[Sta97] Jos Stam. A General Animation Framework for Gaseous Phenomena. ERCIM Research Report R047, 1997.

[Sta99] J. Stam. Stable fluids. *Computer Graphics (Proc. SIGGRAPH)*, pages 121–128, 1999.

[Sta01] Jos Stam. A simple fluid solver based on the FFT. *Journal of Graphics Tools: JGT*, 6(2):43–52, 2001.

[Sta03] J. Stam. Real-time fluid dynamics for games. *Proceedings of the Game Developer Conference*, March 2003.

[TFK$^+$03] T. Takahashi, H. Fuijii, A. Kunimatsu, K. Hiwada, T. Saito, K. Tanaka, and H. Ueki. Realistic Animation of Fluid with Splash and Foam. *Computer Graphics Forum*, 22(3):391–400, 2003.

[WH91] J. Wejchert and D. Haumann. Animation aerodynamics. *Computer Graphics (Proc. SIGGRAPH)*, 25(2):19–22, July 1991.

[WZF$^+$03] Xiaoming Wei, Ye Zhao, Zhe Fan, Wei Li, Suzanne Yoakum-Stover, and Arie Kaufman. Blowing in the wind. In *Proc. of SIGGRAPH Symposium on Computer Animation 2003*, pages 75–85. Eurographics Association, 2003.

[YU86] L. Yaeger and C. Upson. Combining Physical and Visual Simulation. Creation of the Planet Jupiter for the Film 2010. In *Proc. SIGGRAPH Symposium on Computer Animation*, pages 85–93, 1986.

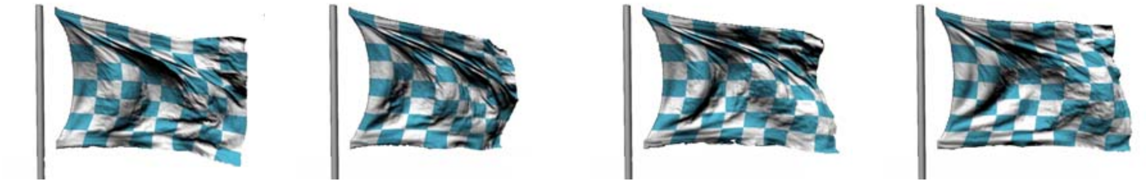


Figure 3: *Flag blowing in the wind (particle tracing method).*

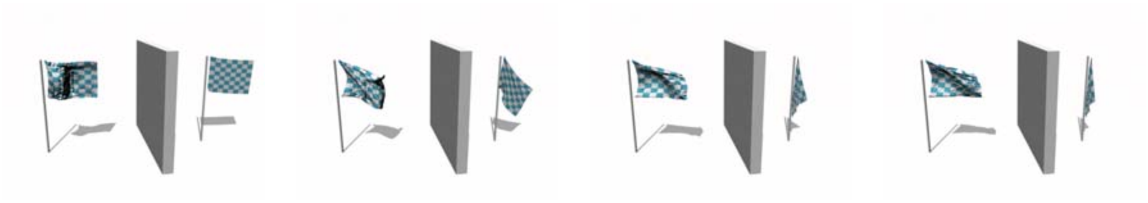


Figure 4: *Two flags blowing in the wind (Navier-Stokes equations). The wind is blocked by the wall, so the right flag is in the lee. Note that the simulation starts with both flags in an unfolded state.*

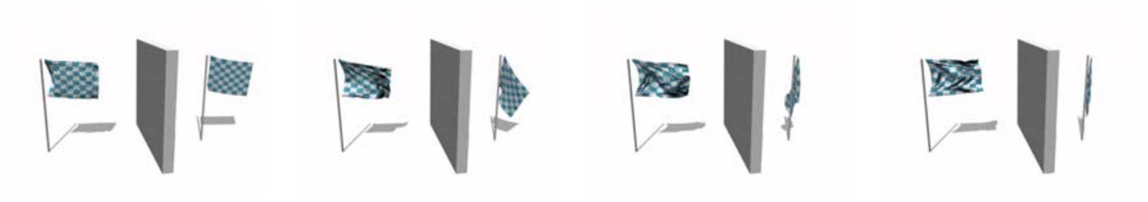
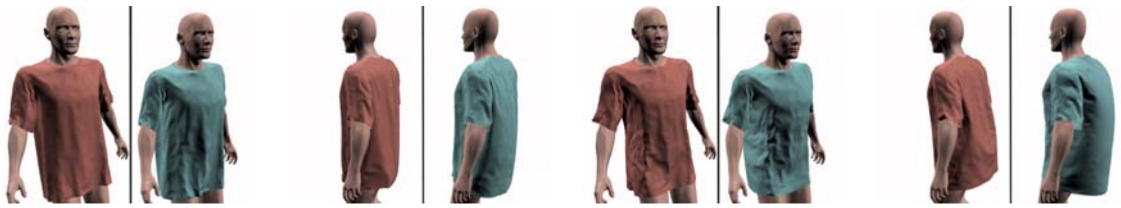


Figure 5: *Two flags blowing in the wind (particle tracing method). The wind is blocked by the wall, so the right flag is in the lee (compare figure 4).*



(a) (b) (c) (d)

Figure 6: *A character wearing a shirt and standing in a wind stream coming from the front. Images (a) and (b) are snapshots from the beginning of the animation, images (c) and (d) show the result after the wind field has affected the clothes. To show the improved realism when simulating lee effects, we let the the wind act on all polygons of the shirt on the right (no lee effect). For the shirt on the left we used the particle tracing method to simulate lee effects which gives more realistic results.*