Keyframe Control of Smoke Simulations

SIGGRAPH 2003

Keyframe Control of Smoke Simulations

SIGGRAPH 2003

Authors: Adrien Treuille (University of Washington)
Antoine McNamara (University of Washington)
Zoran Popovic (University of Washington)
Jos Stam (Alias|Wavefront)

Presented by: Roland Angst (rangst@student.ethz.ch)

Outline

1. Motivation and General Idea
2. Introduction to Navier-Stokes Equations
3. Keyframe-Control Approach
4. Exact Derivatives
5. Control Parameters
6. Layered Multiple Shooting
7. Results
8. Problems
9. My Own Thoughts

1. Motivation

	Animation	vs. Physical based Simulation
Pro	Complete artistic freedom	Plausible scenes
Cons	For physical plausible scenes it becomes quickly rather tedious	- computational resources - limited artistic freedom

1.1 Physical Based Simulation

Simulation with considering physics	⟷	Defining an initial state q^0 and temporal integration of physical laws.

Influence capabilities of an artist:

► Manipulating the initial state of the simulation

► Leads to almost unpredictable simulation behaviour!

Goal: Combine the artistic freedom of animations with physical plausibility of simulations.

1.2 Keyframing

► **Key Frame:** „A key frame (●) is a frame in an animated sequence of frames that was drawn or otherwise constructed directly by the user. ... The computer fills in the gap (—). This is called tweening.“ [Wikipedia]

Key frame in simulation ⟷ Defining the system state q^T at a fixed time T

Keyframing for Smoke

What we have:

► physical description of the fluid dynamics through PDEs.

What we want:

► physical plausible interpolation or approximation of the key frames

Idea: Influence the dynamics by addition of parameterised, external control forces.

► automatic optimization process searches for suitable control force parameters to approximate the given key frames.

1.3 Two Main Contributions

- **Optimization approach:** Definition of a target function we have to minimize.
- **Minimization technique:** gradient based approach

Method for exact calculation of the derivatives of the fluid simulation states.

- Optimization with multiple key frames needs a lot of computation.

New multiple shooting approach for animations with several key frames.

2. Introduction to Navier-Stokes Equations

- Short introduction or refresh of the Navier-Stokes equations for fluid simulation
- This is not an actual part of the paper, but it is required for the comprehension.
- For further information see presentation of Jos Stam:

www.dgp.utoronto.ca/~stam/reality/Talks/FluidsTalk/FluidsTalkNotes.pdf

2.1 Navier-Stokes Equations

- ► Navier-Stokes equations completely describe dynamic behaviour of an **incompressible** fluid (gas or liquid)

- ► Navier-Stokes equations consist of a scalar and a vector valued PDE

- ► State q of a point in a fluid (gas or liquid) is described by:
  - ► velocity field v
  - ► density field ρ

- ► 3 DoF of v + 1 DoF of ρ = 4 DoF per point

Mathematical Description

1. Mass conservation in incompressible medium:

$$\rho_t + \nabla \cdot (\rho \vec{v}) = 0 \xrightarrow{incompressible} \nabla \cdot \vec{v} = 0$$

▶ v is divergence free

2. Momentum conservation (row wise to understand, i.e. 3 equations):

$$\vec{v}_t = -\underbrace{(\vec{v} \cdot \nabla)\vec{v}}_{Advection} + \underbrace{\mu \Delta \vec{v}}_{Diffusion} + \underbrace{\vec{f}_{extern}}_{external\ Forces} - \underbrace{\nabla p}_{pressure\ Gradient\ Field}$$

▶ v_t is a linear combination of 4 terms

2.2 Numerical Solution

- State $\mathbf{q}^T$ in point in time T: Grid of densities and velocities:

$$\vec{q}^T = \left(\rho^T, \vec{v}^T\right)$$

- Integration of the velocity field in time (ex. with Euler):

$$\vec{v}_t^{\,T} = -\underbrace{\left(\vec{v}^{\,T} \cdot \nabla\right)\vec{v}^{\,T}}_{Advection} + \underbrace{\mu \Delta \vec{v}^{\,T}}_{Diffusion} + \underbrace{\vec{f}_{extern}^{\,T}}_{external\ Forces} - \underbrace{\nabla p^T}_{pressure\ Gradient\ Field}$$

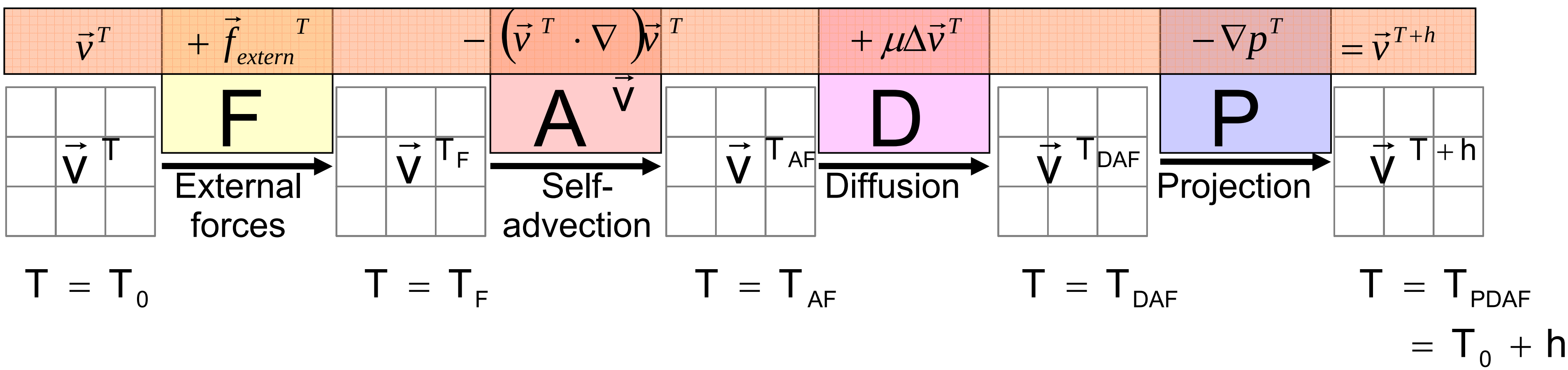
$$\vec{v}^{\,T+1} = \vec{v}^{\,T} + h\vec{v}_t^{\,T}$$

Notation: To prevent confusion with partial time derivatives the points in time are indicated by super- instead of subscripts (This is a difference to the notation used in the paper…)

Unconditional Stable Method

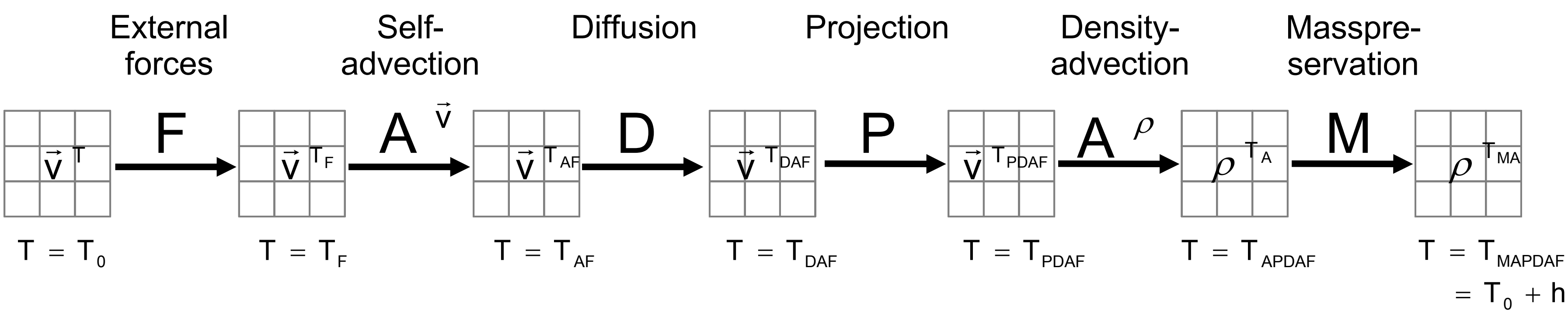
Splitting computation of v^{T+1} in four smaller steps:

1. Add external forces
2. Self-advect velocity field
3. Diffusion
4. Use remaining DoF of the density field ρ to ensure a divergence free velocity field (aka. projection step)



Whole Simulation Step

1. Calculate v^{T+1} by splitting it into four smaller steps
2. Advect the density field through this newly calculated velocity field
3. Compensate the dissipation (inherently in unconditional stable methods) by a mass conserving step.



3. Keyframe-Control Approach

- **Given:** keyframe state at time T: q^T_*
- **Goal:** add external control forces $f_{control}(u)$ to
  - Approximate the key frame states q^T_* by the simulation states q^T while
  - Minimizing the ‘artificial introduces’ external control forces $f_{control}(u)$

3.1 Optimization Approach

- Definition of a target function: $\varphi(q^0, u) = \varphi_s + \varphi_k$
- Parameterise the control forces by a parameter vector u → gradient based minimization technique usable:

$$\arg\min_{\vec{u}} \varphi\left(\vec{q}^{\,0}, \vec{u}\right) = \arg\min_{\vec{u}} \left(\varphi_k + \varphi_s\right)$$

φ_s : Penalty term for added external control forces $f_{control}(u)$:

$$\varphi_s = k_s \sum_{T \in Timesteps} \left\| \vec{f}_{control}^{\;T} \right\|^2$$

φ_k : Difference metric between keyframes $q_*^T = (\rho_*^T, v_*^T)$ and corresponding simulation states $q^T = (\rho^T, v^T)$:

$$\varphi_k = k_d \sum_{T \in Timesteps \;\; with\ density\ keyframes} \left\| B\left(\rho^T - \rho_*^T\right)\right\|^2 + k_v \sum_{T \in Timesteps \;\; with\ velocity\ keyframes} \left\| B\left(\vec{v}^{\,T} - \vec{v}_*^{\,T}\right)\right\|^2$$

3.2 Target Function Gradient

$$\varphi_s = k_s \sum_{T \in Timesteps} \left\| \vec{f}_{control}{}^T \right\|^2$$

differentiate

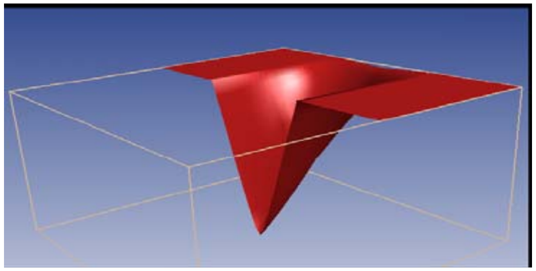
$$\frac{d\varphi_s}{du_i} = 2k_s \sum_{T \in Timesteps} \vec{f}_{control}{}^T \cdot \frac{d\vec{f}_{control}{}^T}{du_i}$$

$$\varphi_k = k_d \sum \left\| B\left(\rho^T - \rho_*^T\right) \right\|^2 + k_v \sum \left\| B\left(\vec{v}^T - \vec{v}_*^T\right) \right\|^2$$

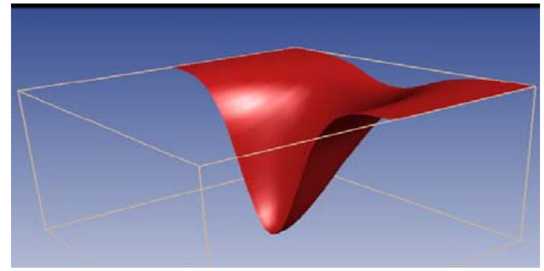
differentiate

$$\frac{d\varphi_k}{du_i} = 2k_d \sum B\left(\rho^T - \rho_*^T\right) B\left(\frac{d\rho^T}{du_i}\right) + 2k_v \sum B\left(\vec{v}^T - \vec{v}_*^T\right) B\left(\frac{d\vec{v}^T}{du_i}\right)$$

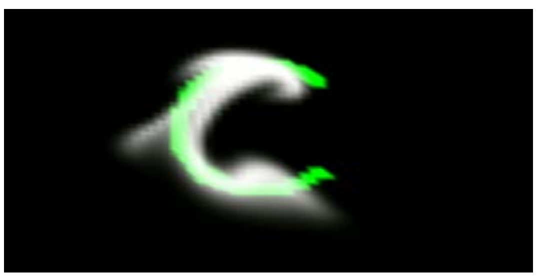
3.3 Blurring

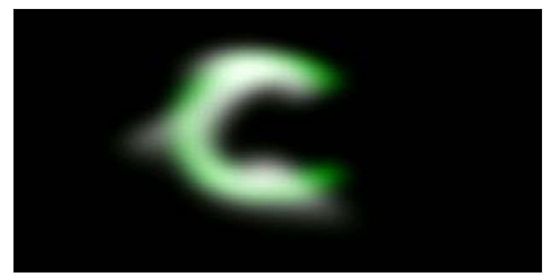


$$\sum_T \left\| \vec{q}^{\,T} - \vec{q}_*^{\,T} \right\|^2$$



$$\sum_T \left\| B\left(\vec{q}^{\,T} - \vec{q}_*^{\,T}\right) \right\|^2$$





Unblurred simulation state and keyframe → Blurred simulation state and blurred keyframe

4. Exact Derivatives

Needed terms to compute the gradient of the target function φ:

$$\frac{d\,\vec{f}_{control}^{\;T}}{d\,u_i} \text{ and } \frac{d\,\rho^T}{d\,u_i} \text{ and } \frac{d\,\vec{v}^{\,T}}{d\,u_i}$$

4.1 Three Solution Approaches

1. Analytic derivatives of the Navier-Stokes equations

Problem:

► no absolutely physically correct numerical solution
► Therefore analytic derivatives of Navier-Stokes equations need not agree with derivatives of numerical simulation!

2. Finite Difference Approximation

Problem:

Unsuitable because slow and very inaccurate!

4.1 Three Solution Approaches

3. New method:

Augment the state of the simulation $q^T = (v^T, \rho^T)$ with the needed derivatives:

$$\vec{q}^{\,T} = \left(\vec{v}^{\,T}, \rho^T, \frac{\partial \vec{v}^{\,T}}{\partial u_1}, \frac{\partial \rho^T}{\partial u_1}, \frac{\partial \vec{v}^{\,T}}{\partial u_2}, \frac{\partial \rho^T}{\partial u_2}, \ldots \right)$$

$$\vec{q}^{\,0} = \left(\vec{v}^{\,0}, \rho^0, 0, 0, 0, 0, \ldots \right)$$

Motivated by [Popovic 2000].

4.2 New Method

Remember: Needed terms to calculate the gradient of the target function φ:

- ▶ $\frac{d\ \vec{f}_{control}^{\,T}}{d\,u_i}$ Analytic term derivable since control forces are directly parameterised by **u**

- ▶ $\frac{d\,\rho^T}{d\,u_i}$ and $\frac{d\,\vec{v}^T}{d\,u_i}$ **Idea:** apply the simulation steps not only on q = (v, ρ) but also on the partial derivatives with respect to every control force parameter u_i:

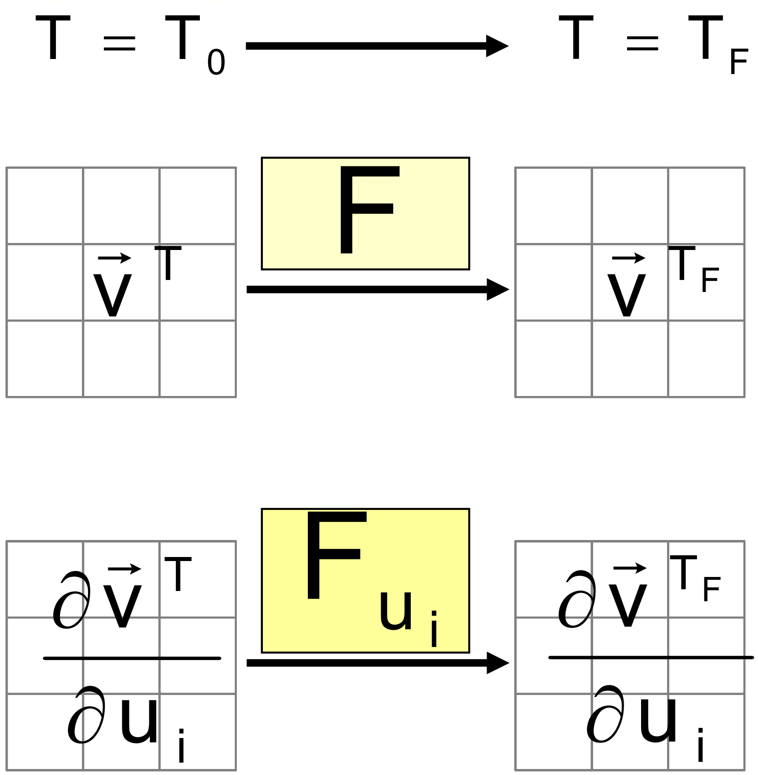
$$\frac{d\,\vec{q}}{d\,u_i} = \left(\frac{d\,\vec{v}}{d\,u_i}, \frac{d\,\rho}{d\,u_i} \right)$$

4.3 Partial Stepped Derivatives

Recall: One simulation step of Navier-Stokes equations by multiple small partial steps!

Carrying along the derivatives in time	⟷	Every partial simulation step has corresponding partial step for the derivatives

Partial External Force Step



$$\vec{v}^{\,T_F} = \vec{v}^{\,T} + f^{\,T}_{extern}$$

Differentiate

$$\frac{\partial \vec{v}^{\,T_F}}{\partial u_i} = \frac{\partial \vec{v}^{\,T}}{\partial u_i} + \frac{\partial f^{\,T}_{extern}}{\partial u_i}$$

Parallel Partial Steps

$T = T_0$, $T = T_F$, $T = T_{AF}$, $T = T_{DAF}$, $T = T_{PDAF}$, $T = T_{APDAF}$, $T = T_{MAPDAF} = T_0 + h$

$$\vec{v}^{\,T} \xrightarrow{F} \vec{v}^{\,T_F} \xrightarrow{A^{\vec{v}}} \vec{v}^{\,T_{AF}} \xrightarrow{D} \vec{v}^{\,T_{DAF}} \xrightarrow{P} \vec{v}^{\,T_{PDAF}} \xrightarrow{A^{\rho}} \rho^{T_A} \xrightarrow{M} \rho^{T_{MA}}$$

External forces | Self-advection | Diffusion | Projection | Density-advection | Masspreservation

$$\frac{\partial \vec{v}^{\,T}}{\partial u_i} \xrightarrow{F_{u_i}} \frac{\partial \vec{v}^{\,T_F}}{\partial u_i} \xrightarrow{A^{\vec{v}}_{u_i}} \frac{\partial \vec{v}^{\,T_{AF}}}{\partial u_i} \xrightarrow{D_{u_i}} \frac{\partial \vec{v}^{\,T_{DAF}}}{\partial u_i} \xrightarrow{P_{u_i}} \frac{\partial \vec{v}^{\,T_{PDAF}}}{\partial u_i} \xrightarrow{A^{\rho}_{u_i}} \frac{\partial \rho^{T_A}}{\partial u_i} \xrightarrow{M_{u_i}} \frac{\partial \rho^{T_{MA}}}{\partial u_i}$$

5. Control Parameters

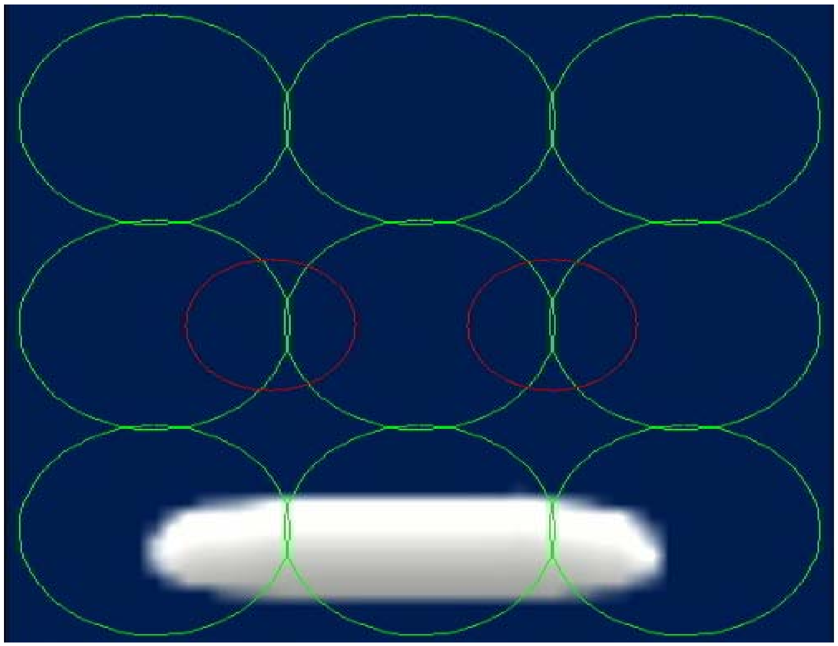
Wind Forces: a single vector scaled by a Gaussian falloff function

$$\vec{u} = \begin{pmatrix} wind \quad directrion \\ Gaussian \quad center \end{pmatrix}$$

Vortex Forces: a fixed rotation matrix scaled by a Gaussian falloff function and a parameter r

$$\vec{u} = \begin{pmatrix} vortex \quad center \\ r \end{pmatrix}$$

5. Control Parameters

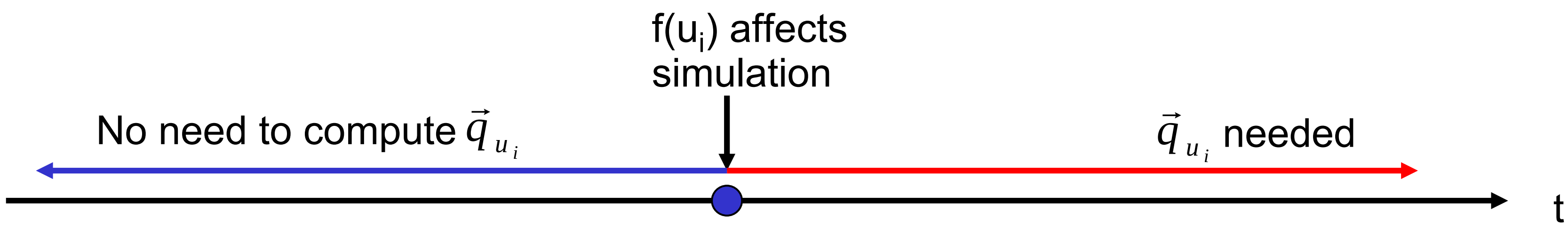


wind force

vortex force

6. Layered Multiple Shooting

- **1st Problem:**
  Computing $\vec{q}_{u_i}$ only from the timestep on when the control force belonging to u_i affected the simulation.



- **2nd Problem:**
  Local minima of cost function

6.1 Idea of Multiple Shooting

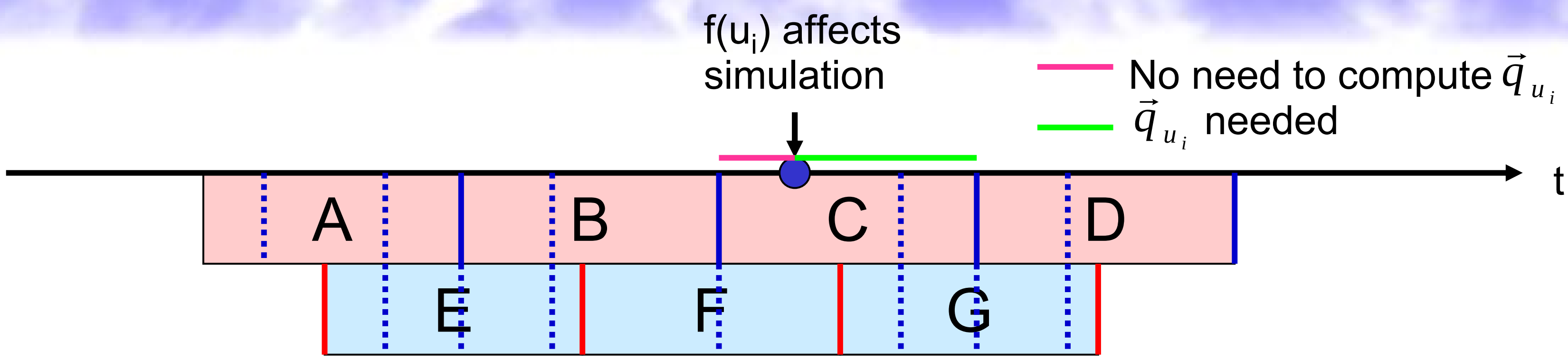
Multiple Shooting:

- ► Temporally break a complex problem into a set of subproblems.
- ► Use local solutions of these subproblems to propagate knowledge back and forth to get a global solution.

Problem:

- ► no physical meaningful interpolation to construct a global solution.

6.2 Layered Multiple Shooting



Boundary original keyframe

Non boundary original keyframe

Culled from intermediate states of the initial segments

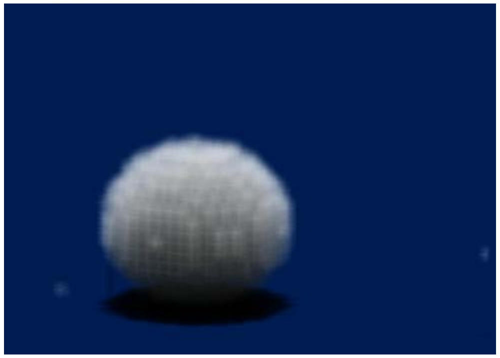
Initial schedule

Alternate schedule

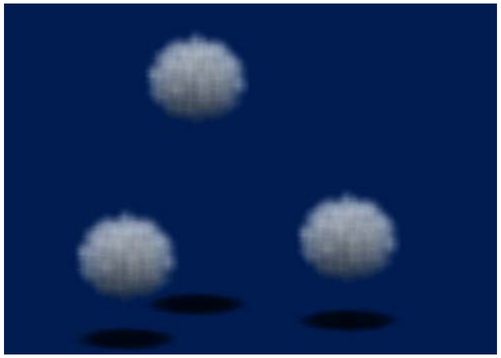
Parallel Processing

Sequential Processing

7. Results



Keyframe $\vec{q}_*^{\,0}$



Keyframe $\vec{q}_*^{\,T_{end}}$

Gridsize:	$30 \cdot 30 \cdot 30$
Nr. of control forces:	20
Computation time:	2h on P4 2GHz

7. Results

Parameters: 408
Keyframes: 2
Steps: 35
Time: ca. 24h

8. Problems

- Optimization process rather slow
  One single evaluation of the target function needs a run of the whole simulation with augmented states!!!

- Local Minima: method not fully automated (yet?)

Possible Solution: inserting additional keyframe to guide the optimization process

- Result "too controlled" and not "smoke-like"

9. My Own Thoughts

- 1st approach to combine physically based simulations with artistic creativity (in the domain of fluid simulation)
- Shown results look good

  But: how much fine tuning was needed to get them?
- Process is terribly slow!

  How does it scale for larger grid sizes and more control parameters?
- Is minimizing a cost function the right way to go?

Further Ideas

- Multiresolution force framework...
- Other cost function...
- Non-gradient based optimization technique...

End.

Thank you for your attention.