Keyframe Control of Complex Particle Systems Using the Adjoint Method

Daniel Fader

Seminar Computer Animation

WS 06/07

Based on

- **Keyframe Control of Complex Particle Systems Using the Adjoint Method**

  Chris Wojtan*, Peter J. Mucha† and Greg Turk*

  ACM/SIGGRAPH Symposium on Computer Animation 2006

  http://www-static.cc.gatech.edu/~wojtan/

* Georgia Institute of Technology

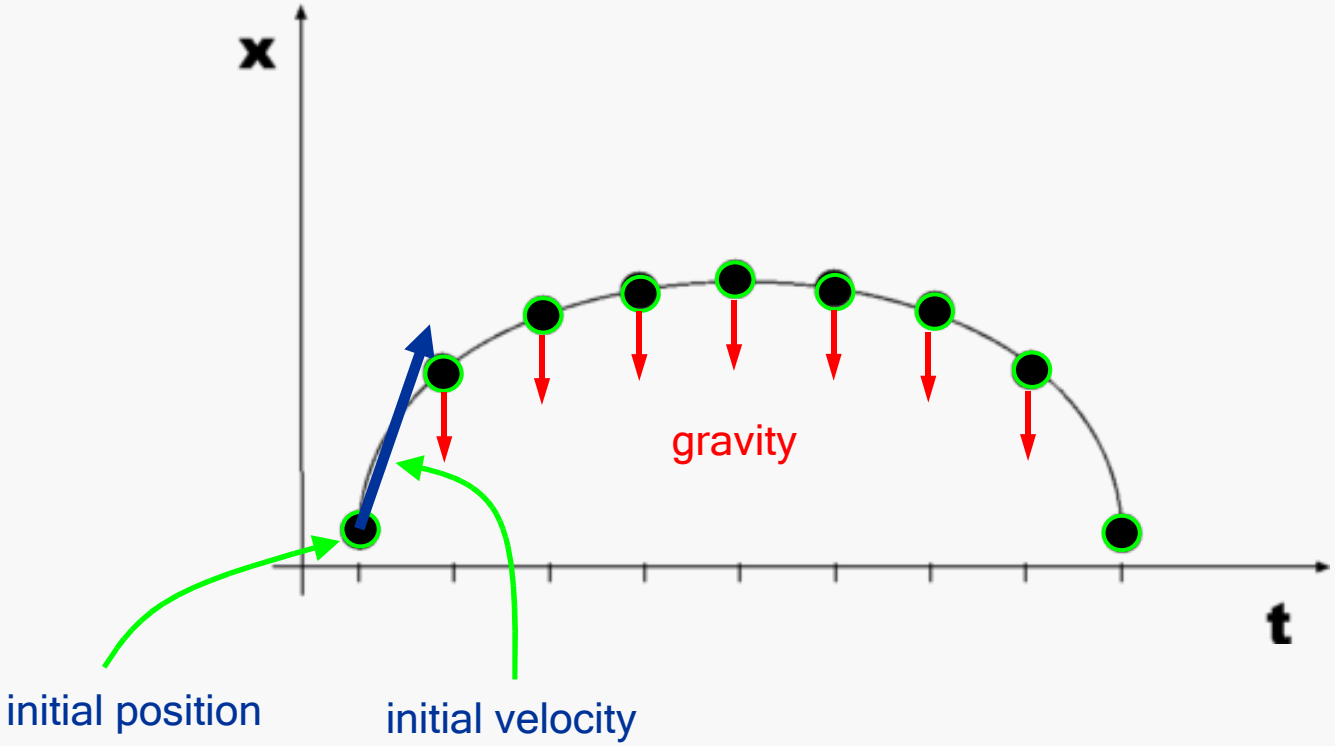
† University of North Carolina

Outline

- Motivation
- Introduction
  - Particle systems
  - Simulation of particle systems
  - Approach to controlled simulation
  - Objective function
- Gradient-based optimization
  - Gradient calculation
  - Linear duality
  - The adjoint method
- Flocking
- Cloth
- Summary

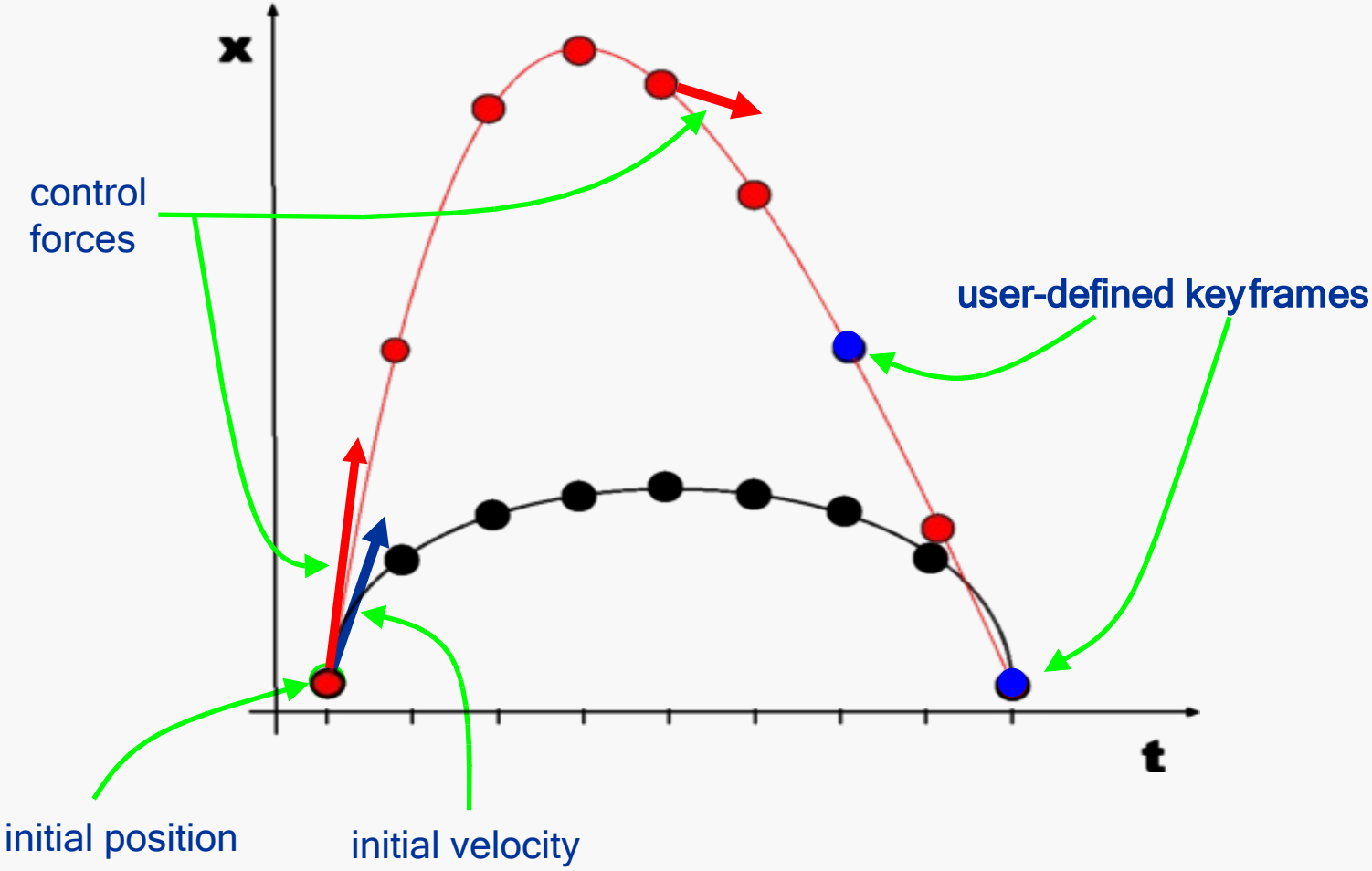
Motivation

- How does an **uncontrolled** simulation work?



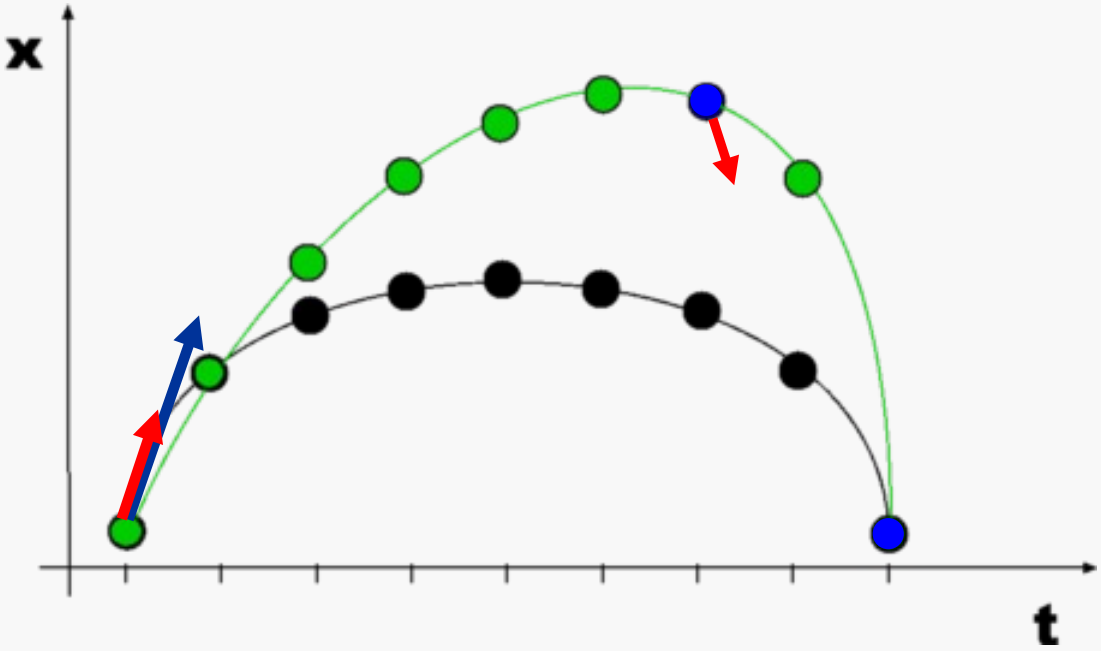
Motivation

- How does a **controlled** simulation work?



Motivation

- Controlled simulation: **Match** keyframes but **minimize** control forces!



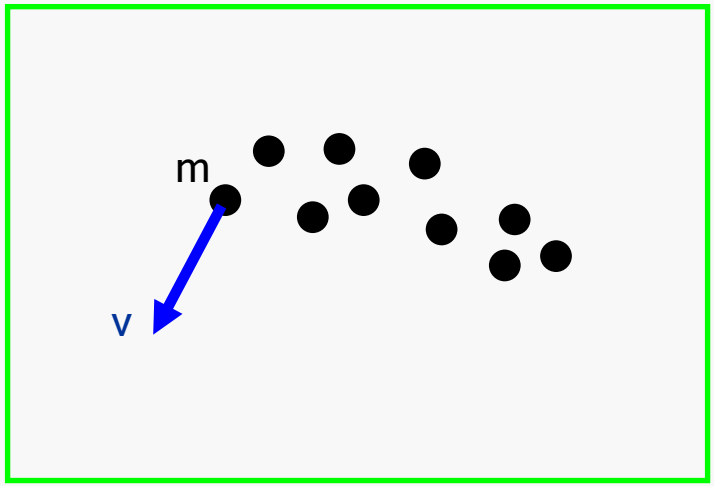
Outline

- Motivation
- Introduction
  - Particle systems
  - Simulation of particle systems
  - Approach to controlled simulation
  - Objective function
- Gradient-based optimization
- Flocking
- Cloth
- Summary

Particle systems

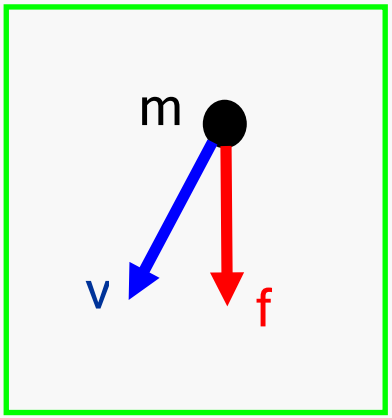
- Consist of many small particles
- Each particle has a
  - **mass**
  - **position**
  - **velocity**

  defined at each time step t_n
- Collect these quantities into
  - **position vector** $\mathbf{x}$
  - **velocity vector** $\mathbf{v}$ (both varying in time)
  - **mass matrix** $\mathbf{M}$ (constant)



Particle systems

- There are diverse model specific **forces** influencing the particle motion
- Particles interact with their environment
- In *coupled particle systems* particles also interact with each other



Particle systems usage

- Particle systems can be deployed to simulate
  - **Flocks**
  - **Cloth**
  - Fire, explosions
  - Snow
  - Fluids, waterfalls
  - Smoke, sand, dust, vortex
  - ...

[WIKI]

- They basically differ in the forces affecting the particles

Simulation of particle systems

- Fixed initial state $\mathbf{q}_0 = (\mathbf{x}_0, \mathbf{v}_0)$ given
- Evolve $\mathbf{q}_0$ into n subsequent states $\mathbf{q}_1, \ldots, \mathbf{q}_n$ according to the update rule

$$\mathbf{q}_{i+1} = \mathbf{f}_i(\mathbf{q}_i, \mathbf{u}) \text{ , } \mathbf{u}\text{: control forces}$$

- Aggregate these into one long vector

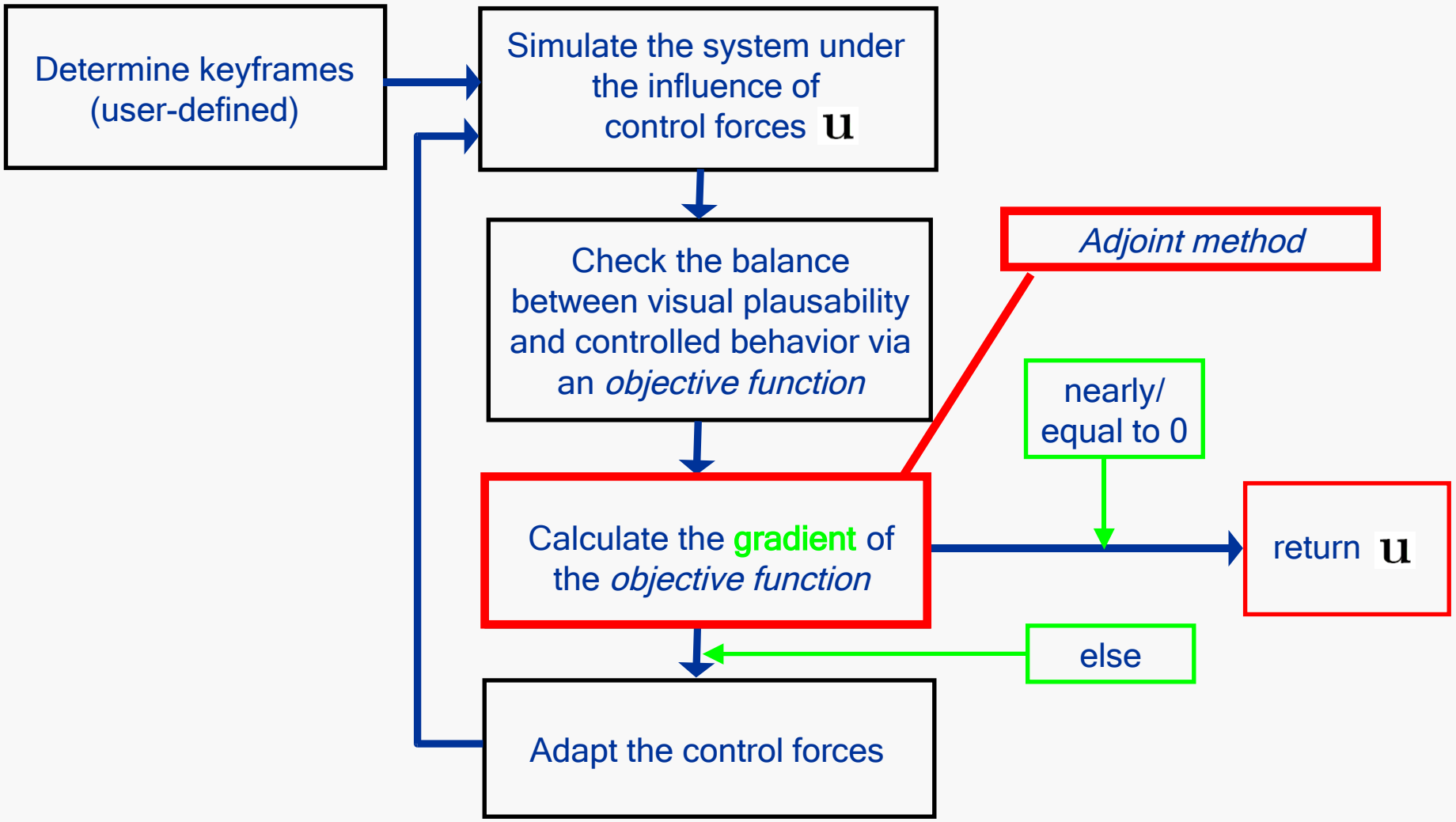
$$\mathbf{Q} = \left[\mathbf{q}_1^T, \ldots, \mathbf{q}_n^T\right]^T$$

- And one long vector function

$$\mathbf{F}(\mathbf{Q}, \mathbf{u}) = \left[\mathbf{f}_0(\mathbf{q}_0, \mathbf{u})^T, \ldots, \mathbf{f}_{n-1}(\mathbf{q}_{n-1}, \mathbf{u})^T\right]^T$$

$$\Longrightarrow \quad \mathbf{Q} = \mathbf{F}(\mathbf{Q}, \mathbf{u})$$

Approach to controlled simulation



Objective function

$$\phi(\mathbf{u}, \mathbf{Q}) = \frac{1}{2} \sum_{n=0}^{N} \left(\| W_n \left(\mathbf{q}_n - \mathbf{q}_n^* \right) \|^2 + \alpha \| C_n \mathbf{u} \|^2 \right)$$

- $\mathbf{q}_n = \langle \mathbf{x}_n, \mathbf{v}_n \rangle$ contains the state of each particle at t_n
- $\mathbf{q}_n^*$ is a keyframe at frame n
- W_n is a weight matrix; typically only a few frames have non-zero weights
- C_n maps the control forces $\mathbf{u}$ to the particles at each frame

Objective function

$$\phi(\mathbf{u}, \mathbf{Q}) = \frac{1}{2} \sum_{n=0}^{N} \left(\| W_n \left(\mathbf{q}_n - \mathbf{q}_n^* \right) \|^2 + \alpha \| C_n \mathbf{u} \|^2 \right)$$

Increases when the simulation does not match the keyframes

Increases when too much external control is used

Objective function

Goal: Find a set of controls, $\mathbf{u}$, that

- satisfies the animator‘s constraints
- minimizes non-physical behavior, i. e.

$$\underset{\mathbf{u}}{\operatorname{argmin}}\ \phi(\mathbf{u}, \mathbf{Q})$$

⟹ Use a gradient-based optimization routine

Outline

- Motivation
- Introduction
- Gradient-based optimization
  - Gradient calculation
  - Linear duality
  - The adjoint method
- Flocking
- Cloth
- Summary

Gradient calculation

- Calculate the gradient of the objective function:

$$\frac{d\phi}{d\mathbf{u}} = \frac{\partial\phi}{\partial\mathbf{Q}}\frac{d\mathbf{Q}}{d\mathbf{u}} + \frac{\partial\phi}{\partial\mathbf{u}}$$

(use the chain rule, because $\mathbf{Q}$ depends on $\mathbf{u}$)

- Remind the objective function:

$$\phi(\mathbf{u},\mathbf{Q}) = \frac{1}{2}\sum_{n=0}^{N}\left(\|W_n\left(\mathbf{q}_n - \mathbf{q}_n^*\right)\|^2 + \alpha\|C_n\mathbf{u}\|^2\right)$$

Gradient calculation

- The gradient of the objective function

$$\frac{d\phi}{d\mathbf{u}} = \frac{\partial\phi}{\partial\mathbf{Q}}\frac{d\mathbf{Q}}{d\mathbf{u}} + \frac{\partial\phi}{\partial\mathbf{u}}$$

contains the matrix $\frac{d\mathbf{Q}}{d\mathbf{u}}$ which requires the **computation of derivatives of all state variables with respect to all possible controls ($\rightarrow$ extremely costly)**

- **Goal:** Sidestep the computation of $\frac{\partial\phi}{\partial\mathbf{Q}}\frac{d\mathbf{Q}}{d\mathbf{u}}$ using the **adjoint method**!

Outline

- Motivation
- Introduction
- Gradient-based optimization
  - Gradient calculation
  - Linear duality
  - The adjoint method
- Flocking
- Cloth
- Summary

Linear duality

Known: Matrices A and C, vector $\mathbf{g}$

Unknown: Matrix B

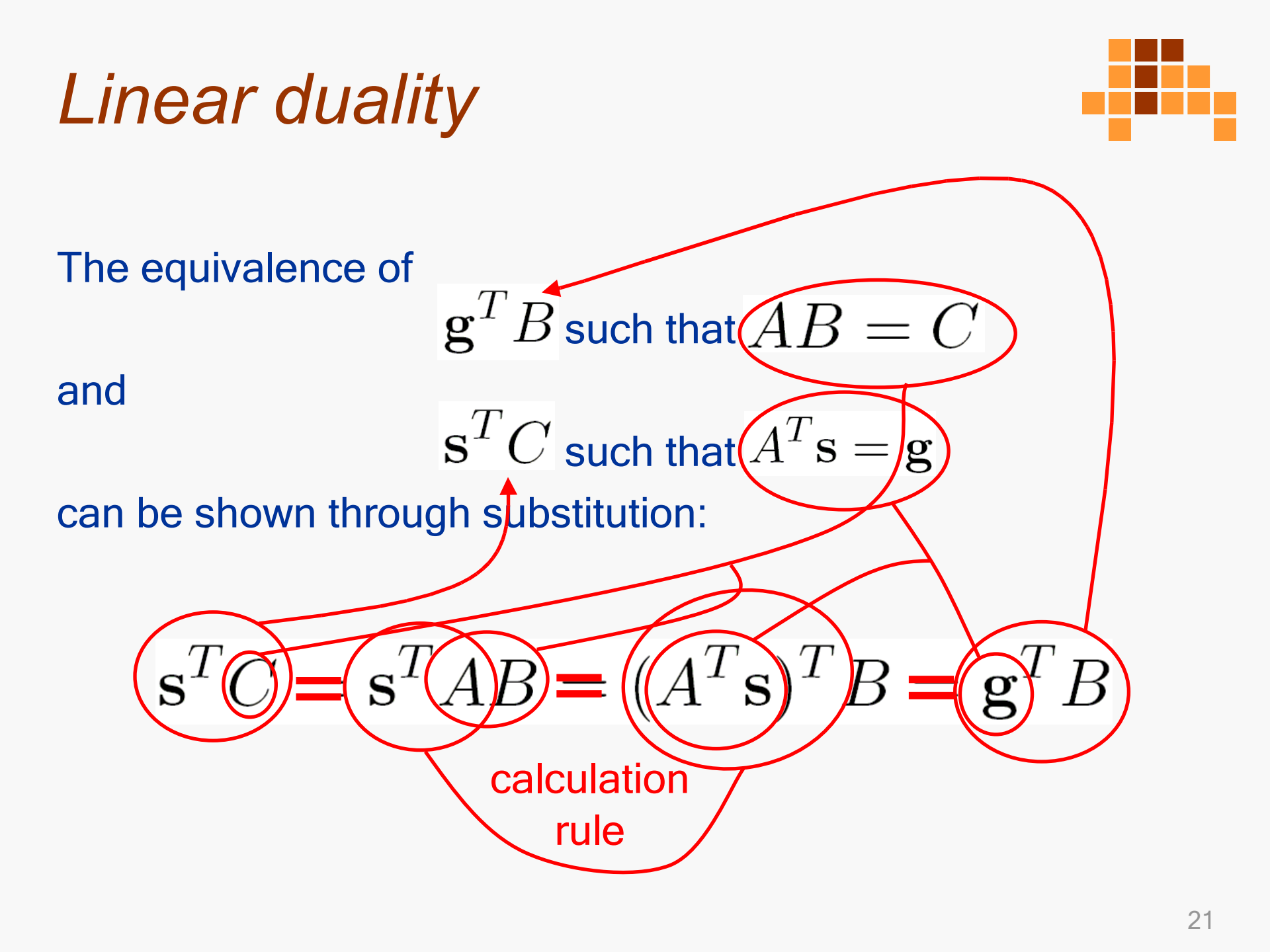
Calculate: $\mathbf{g}^T B$ such that $AB = C$

Naive approach: Solve for B and compute the product

Alternative approach: Introduce a vector $\mathbf{s}$ and compute $\mathbf{s}^T C$ such that $A^T \mathbf{s} = \mathbf{g}$

This is known as the **dual** of the problem

Linear duality



Summary – linear duality

The equivalence of

$\mathbf{g}^T B$ such that $AB = C$

and

$\mathbf{s}^T C$ such that $A^T \mathbf{s} = \mathbf{g}$

affords to save computation time without losing accuracy!

Outline

- Motivation
- Introduction
- Gradient-based optimization
  - Gradient calculation
  - Linear duality
  - The adjoint method
- Flocking
- Cloth
- Summary

The adjoint method

Goal: Sidestep the computation of $\frac{\partial \phi}{\partial \mathbf{Q}} \frac{d\mathbf{Q}}{d\mathbf{u}}$ using the **adjoint method**!

$$\frac{d\phi}{d\mathbf{u}} = \frac{\partial \phi}{\partial \mathbf{Q}} \frac{d\mathbf{Q}}{d\mathbf{u}} + \frac{\partial \phi}{\partial \mathbf{u}}$$

The adjoint method

Due to

$$\mathbf{Q} = \mathbf{F}(\mathbf{Q}, \mathbf{u})$$

you can write the gradient $\frac{d\mathbf{Q}}{d\mathbf{u}}$ as

$$\frac{d\mathbf{Q}}{d\mathbf{u}} = \frac{\partial \mathbf{F}}{\partial \mathbf{Q}} \frac{d\mathbf{Q}}{d\mathbf{u}} + \frac{\partial \mathbf{F}}{\partial \mathbf{u}}$$

or

$$\left(\mathbf{I} - \frac{\partial \mathbf{F}}{\partial \mathbf{Q}}\right) \frac{d\mathbf{Q}}{d\mathbf{u}} = \frac{\partial \mathbf{F}}{\partial \mathbf{u}}$$

$$- \frac{\partial \mathbf{F}}{\partial \mathbf{Q}} \frac{d\mathbf{Q}}{d\mathbf{u}}$$

and factor out

The adjoint method

Use the **dual** of the problem

$$\underbrace{\frac{\partial \phi}{\partial \mathbf{Q}}}_{\mathbf{g}^T}\underbrace{\frac{d\mathbf{Q}}{d\mathbf{u}}}_{B} \text{ such that } \underbrace{\left(\mathbf{I}-\frac{\partial \mathbf{F}}{\partial \mathbf{Q}}\right)}_{A}\underbrace{\frac{d\mathbf{Q}}{d\mathbf{u}}}_{B}=\underbrace{\frac{\partial \mathbf{F}}{\partial \mathbf{u}}}_{C}$$

which is

$$\underbrace{\hat{\mathbf{Q}}^T}_{\mathbf{s}^T}\underbrace{\frac{\partial \mathbf{F}}{\partial \mathbf{u}}}_{C} \text{ such that } \underbrace{\left(\mathbf{I}-\left(\frac{\partial \mathbf{F}}{\partial \mathbf{Q}}\right)^T\right)}_{A^T}\underbrace{\hat{\mathbf{Q}}}_{\mathbf{s}}=\underbrace{\left(\frac{\partial \phi}{\partial \mathbf{u}}\right)^T}_{\mathbf{g}}$$

to calculate the gradient of $\phi(\mathbf{u},\mathbf{Q})$ more quickly:

$$\frac{d\phi}{d\mathbf{u}}=\hat{\mathbf{Q}}^T\frac{\partial \mathbf{F}}{\partial \mathbf{u}}+\frac{\partial \phi}{\partial \mathbf{u}}$$

$\hat{\mathbf{Q}}$ is the *adjoint vector*. **How to compute it?**

The adjoint method

Rewrite

$$\left(\mathbf{I} - \left(\frac{\partial \mathbf{F}}{\partial \mathbf{Q}}\right)^T\right) \hat{\mathbf{Q}} = \left(\frac{\partial \phi}{\partial \mathbf{u}}\right)^T$$

as

$$\hat{\mathbf{Q}} = \left(\frac{\partial \mathbf{F}}{\partial \mathbf{Q}}\right)^T \hat{\mathbf{Q}} + \left(\frac{\partial \phi}{\partial \mathbf{Q}}\right)^T$$

which allows to write

$$\hat{\mathbf{q}}_i = \left(\frac{\partial \mathbf{f}_i}{\partial \mathbf{q}_i}\right)^T \hat{\mathbf{q}}_{i+1} + \left(\frac{\partial \phi}{\partial \mathbf{q}_i}\right)^T$$

The adjoint method

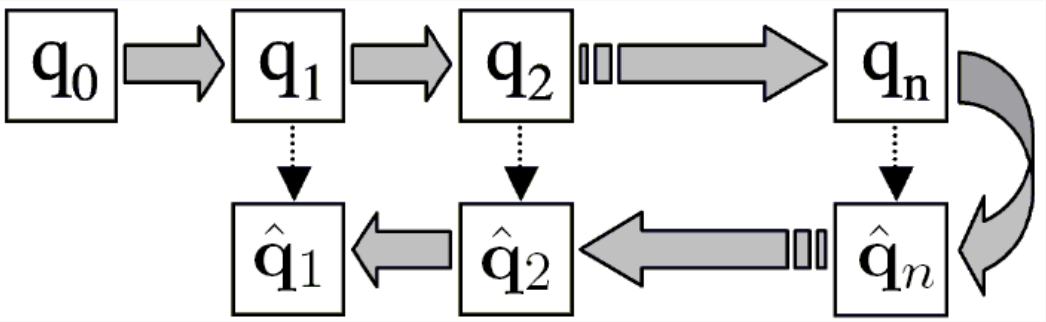
$$\hat{\mathbf{q}}_i = \left(\frac{\partial \mathbf{f}_i}{\partial \mathbf{q}_i}\right)^T \hat{\mathbf{q}}_{i+1} + \left(\frac{\partial \phi}{\partial \mathbf{q}_i}\right)^T$$

and

$$\hat{\mathbf{q}}_n = \left(\frac{\partial \phi}{\partial \mathbf{q}_n}\right)^T$$

define the adjoint vector $\hat{\mathbf{Q}}$

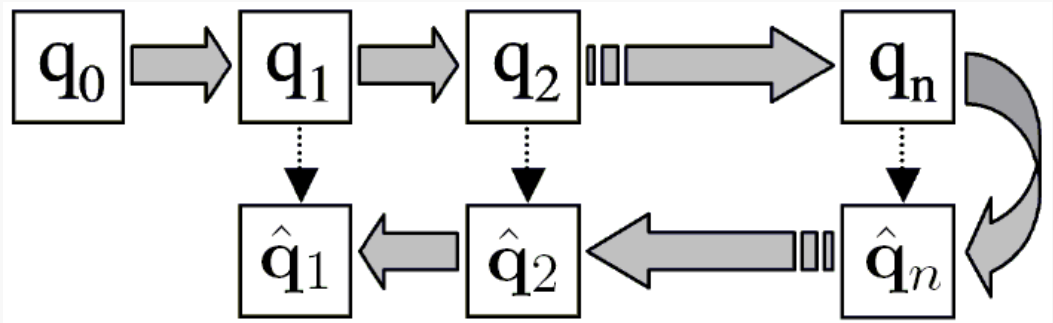
Each *adjoint state* $\hat{\mathbf{q}}_i$ depends on $\mathbf{q}_i$ and $\hat{\mathbf{q}}_{i+1}$



[MTPS04]

Summary - the adjoint method

- Uses a special case of **linear duality**
- A **substitution** of variables that allows to compute the **gradient** of a function quickly
- Calculate the forward states $\mathbf{q}_1, \ldots, \mathbf{q}_n$ first as in an ordinary simulation, but **store all simulation states**
- Then calculate the adjoint states $\hat{\mathbf{q}}_1, \ldots, \hat{\mathbf{q}}_n$ proceeding **backwards** through the sequence



[MTPS04]

Summary - the adjoint method

- When $\hat{\mathbf{q}}_1$ is finally calculated, the
  - objective function $\phi(\mathbf{u}, \mathbf{Q})$ and its
  - gradient $$\frac{d\phi}{d\mathbf{u}} = \hat{\mathbf{Q}}^T \frac{\partial \mathbf{F}}{\partial \mathbf{u}} + \frac{\partial \phi}{\partial \mathbf{u}}$$

  are known

- Iterate until the result has converged to the optimal solution

Outline

- Motivation
- Introduction
- Gradient-based optimization
- Flocking
  - Flocking simulation
  - Controlled flocking
- Cloth
- Summary

Flocking simulation

- Each agent in the flock is treated as a particle
- There are four forces affecting each agent:
  - **Collision Avoidance Force**
  - **Velocity Matching Force**
  - **Flock Centering Force**
  - **Wander Force**

Flocking simulation

- **Collision Avoidance**

- **Velocity Matching**

[WIKI]

Flocking simulation

- **Flock Centering**

[WIKI]

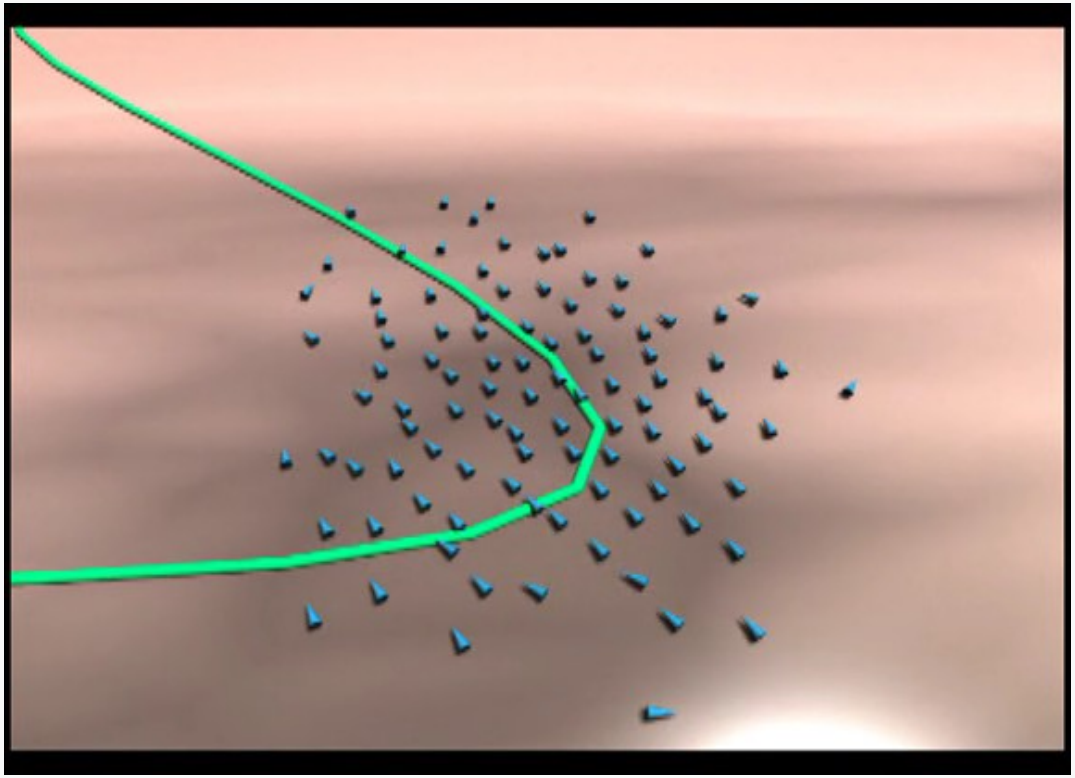
- **Wander Force** to avoid artificial-looking animation by applying a random force

Outline

- Motivation
- Introduction
- Gradient-based optimization
- Flocking
  - Flocking simulation
  - Controlled flocking
- Cloth
- Summary

Controlled Flocking

- A flock of 100 agents follows a pre-determined path



[WMT06]

by setting *center-of-mass* keyframes along the curve

Controlled Flocking

- To apply a center-of-mass constraint the distance term in the objective function would be changed from

$$\phi(\mathbf{u},\mathbf{Q}) = \frac{1}{2}\sum_{n=0}^{N}\left(\|W_n\,(\mathbf{q}_n - \mathbf{q}_n^*)\,\|^2 \;+\; \alpha\|C_n\mathbf{u}\|^2\right)$$

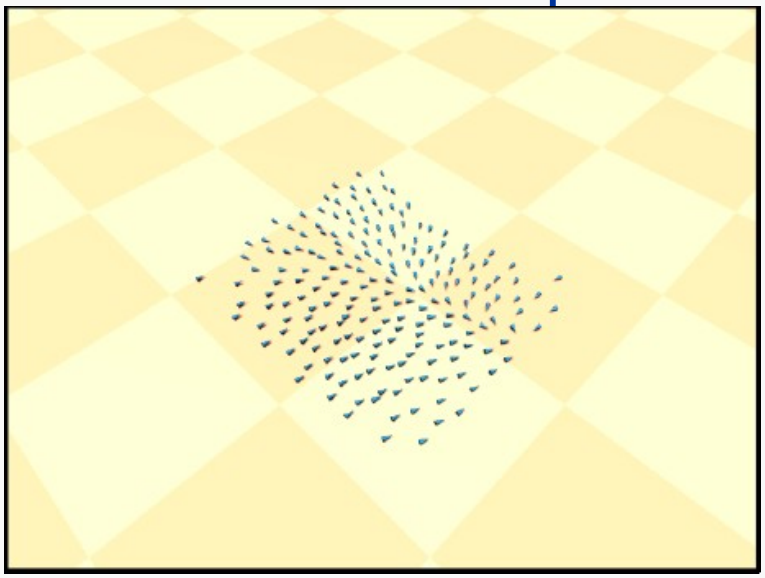
to

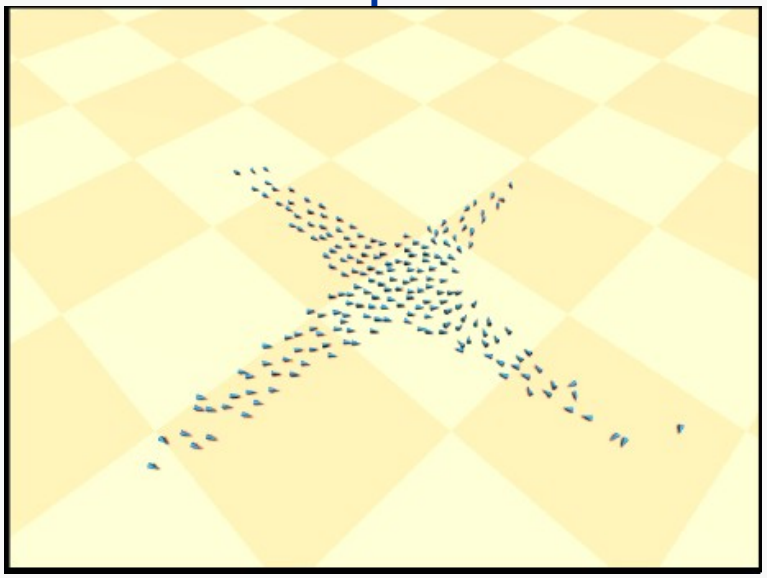
$$\|W_n\,(\mathbf{q}_n^{cm} - \mathbf{q}_n^{cm*})\,\|^2$$

where $\mathbf{q}_n^{cm}$ is computed and $\mathbf{q}_n^{cm*}$ the desired center-of-mass

Controlled Flocking

- A flock forms into a square and a cross shape





[WMT06]

by modifying the distance term in the objective function to sum all distances of agents outside the shape

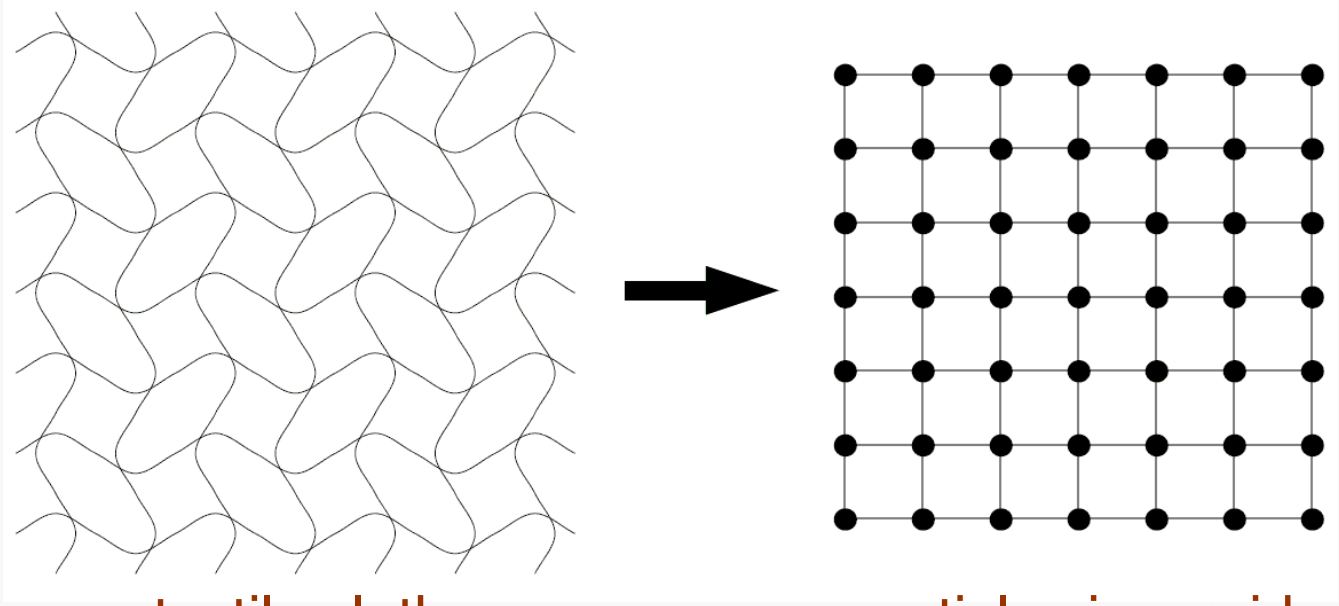
- Less than 250 iterations required

Outline

- Motivation
- Introduction
- Gradient-based optimization
- Flocking
- Cloth
  - Cloth simulation
  - Cloth control
- Summary

Cloth simulation

- Clothes can be modeled with particle systems



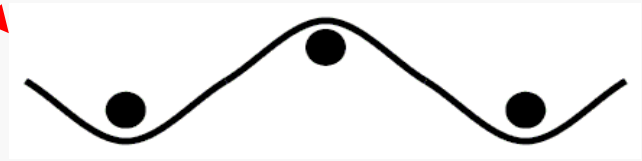
textile cloth

particles in a grid

[BHZ94]

Cloth simulation

- Forces in cloth simulation can be
  - Stretching
  - Bending [BHZ94]
  - Trellising [BHZ94]
  - Shear [CK02]
  - Flexural [CK02]
  - Compression [CK02]



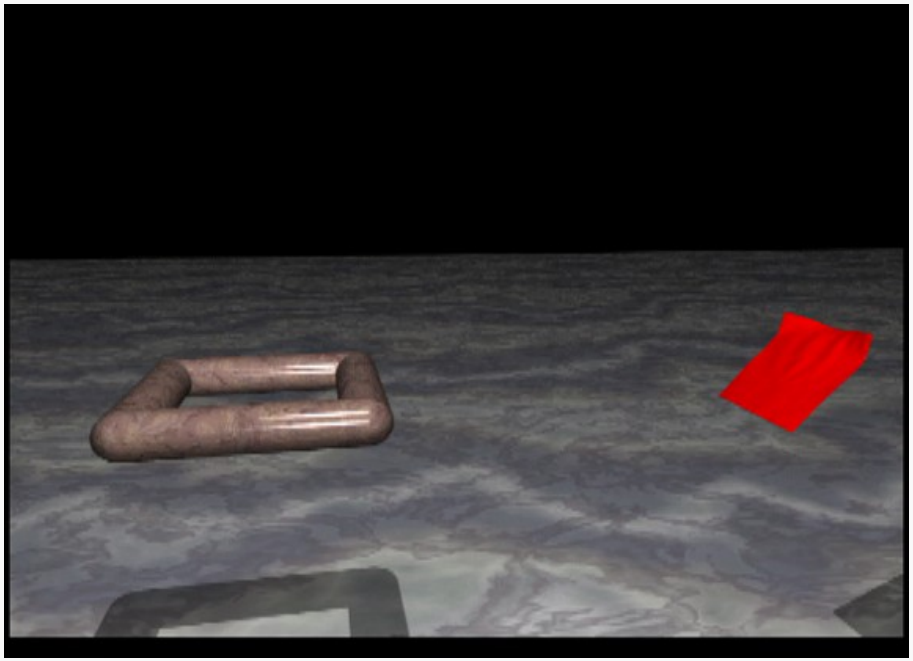
[BHZ94]

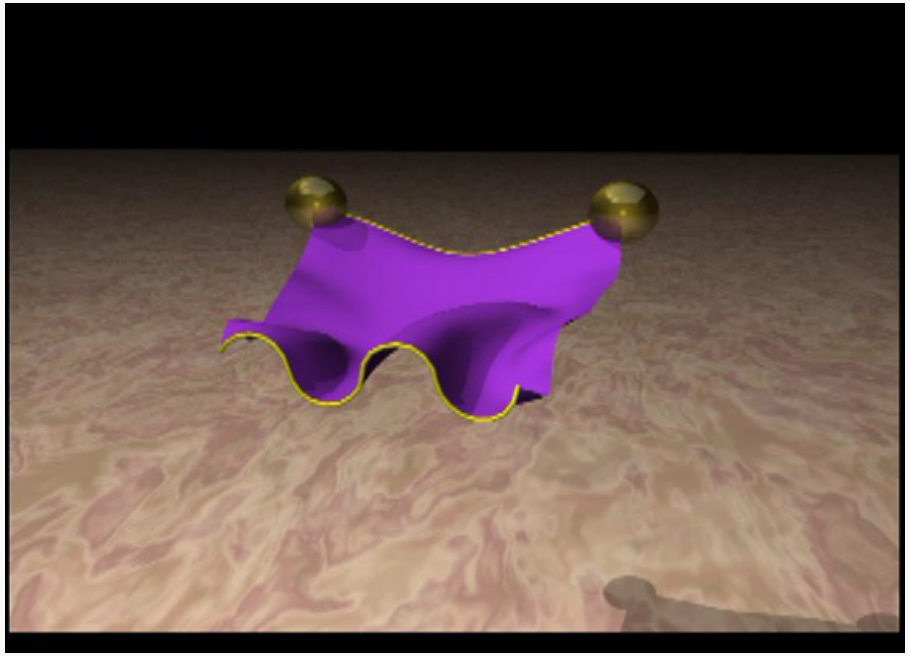
Outline

- Motivation
- Introduction
- Gradient-based optimization
- Flocking
- Cloth
  - Cloth simulation
  - Cloth control
- Summary

Cloth control

- A piece of cloth is forced to land in a ring and another one formes various shapes with his free edge





[WMT06]

using the introduced keyframe technique

Outline

- Motivation
- Introduction
- Gradient-based optimization
- Flocking
- Cloth
- Summary
  - References

Summary

- The simulation of particle systems can be easily controlled in principle
- The adjoint method exploits a powerful aspect of linear duality to accelerate the optimization routine
- The animator can control the simulation in various kinds
- The methodology is not restricted to a specific model

Drawbacks:

- Large discontinuities in forces (e. g. self-collision in cloth animation) can cause the gradient-based optimization to converge, if at all, very slowly
- Saving the simulation states required

References

- **[WMT06]** „Keyframe Control of Complex Particle Systems Using the Adjoint Method”,
  Chris Wojtan, Peter J. Mucha and Greg Turk
- **[MTPS04]** „Fluid Control Using the Adjoint Method“,
  Antoine McNamara, Adrien Treuille, Zoran Popović, Jos Stam
  Explains the adjoint method and linear duality
- **[BHZ94]** „A Particle-Based Model for Simulating the Draping Behavior of Woven Cloth”,
  David E. Breen, Donald H. House, Michael J. Wozny
  Introduces a particle-based cloth model and explains the forces
- **[CK02]** „Stable but Responsive Cloth“,
  Kwang-Jin Choi, Hyeong-Seok Ko
  Introduces a more stable cloth model

References

- „Simulation in Computer Graphics – Cloth Simulation“,
  Matthias Teschner, Matthias Mueller
  http://cg.informatik.uni-freiburg.de/course_notes/sim_03_cloth1.pdf
  Explains several forces in cloth simulation
- „Simulation in Computer Graphics – Introduction“,
  Matthias Teschner, Matthias Mueller
  http://cg.informatik.uni-freiburg.de/course_notes/sim_00_introduction.pdf
  A brief introduction to particle systems (slide 25-27)
- „LBFGS-B: Fortran subroutines for large-scale bound constrained optimization”,
  C. Zhu, R.H. Byrd, P. Lu and J. Nocedal
  An algorithm for solving large nonlinear optimization problems
- **[WIKI]** Wikimedia Commons,
  http://commons.wikimedia.org/
  Free pictures under the terms of GNU Free Documentation License
- Wikipedia – Flocking behavior
  http://en.wikipedia.org/wiki/Flocking_(behavior)
  Links to a couple of flocking simulators

The end

Thank you for your interest
and for your questions!

Appendum

Speeding up the optimization

- The adjoint method requires the repitition of many simulations
- Speeding up the particle simulation will speek up the adjoint method by the same amount
- Possible approaches:
  - Use adaptive time stepping
  - Use faster integration
  - Use faster collision-detection (cloth simulation)
  - Interpolate forces by using splines
  - Use low-resolution for the optimization, then high-resolution for the simulation

Appendum

Handling local minima

- Local minima are still an unsolved problem in gradient-based optimization
- They only really became a problem with severe discontinuities in the optimization-space (like self-collision in cloth simulation)
- Choosing a smart optimizer can avoid local minima
- The optimizer used in the paper is:
  ZHU C., BYRD R. H., LU P., NOCEDAL J.:
  LBFGS-B: Fortran subroutines for large-scale bound constrained optimization
  (q. v. references)