CS**5643**

13 Position-based Dynamics

Steve Marschner
Cornell University
Spring 2025

Constrained dynamics

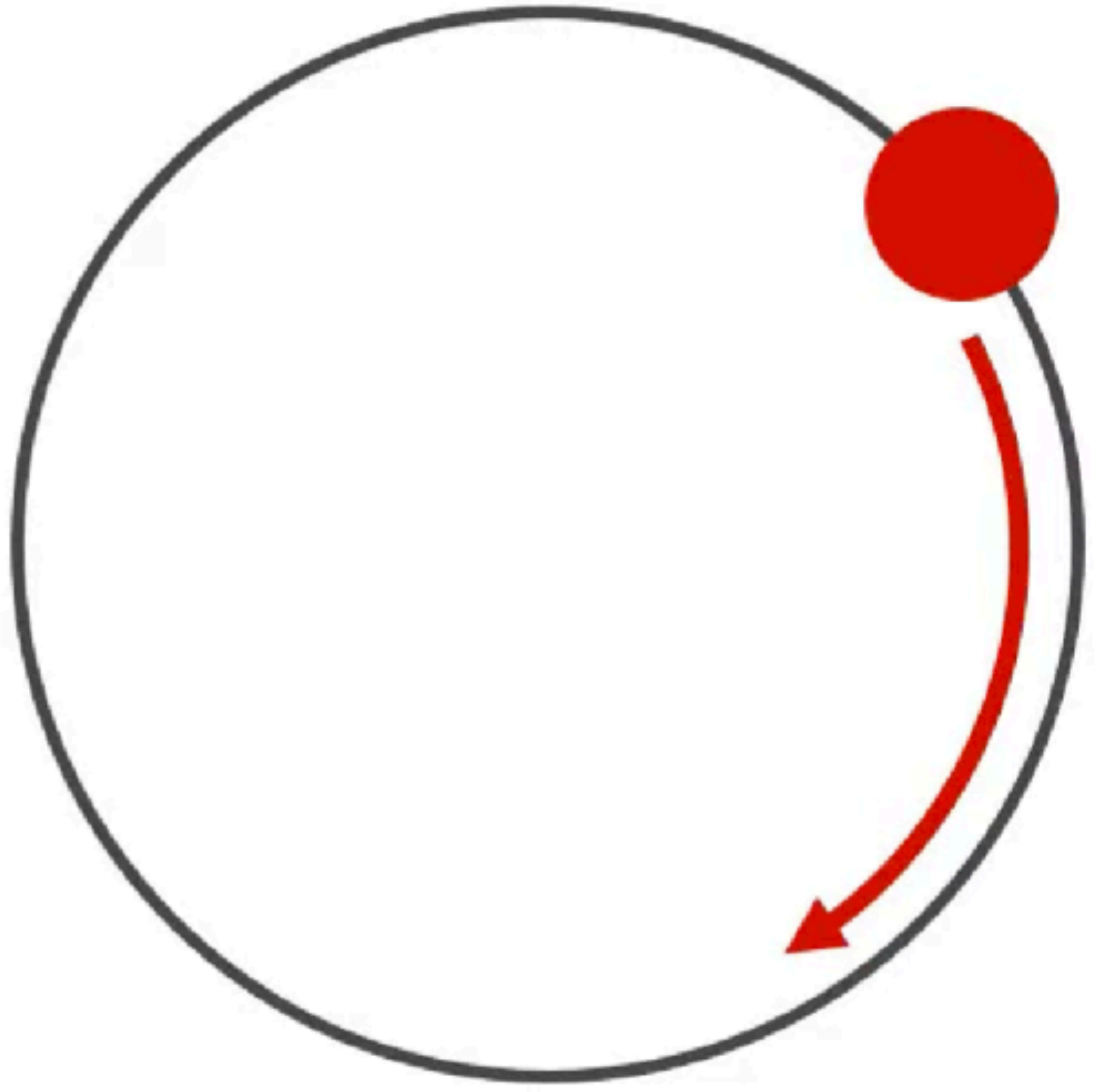
Objects affected by forces as always, but also constraints

$$M\ddot{\mathbf{x}} = f(\mathbf{x}, \dot{\mathbf{x}}) + \underbrace{\nabla C(\mathbf{x})^T \lambda}$$

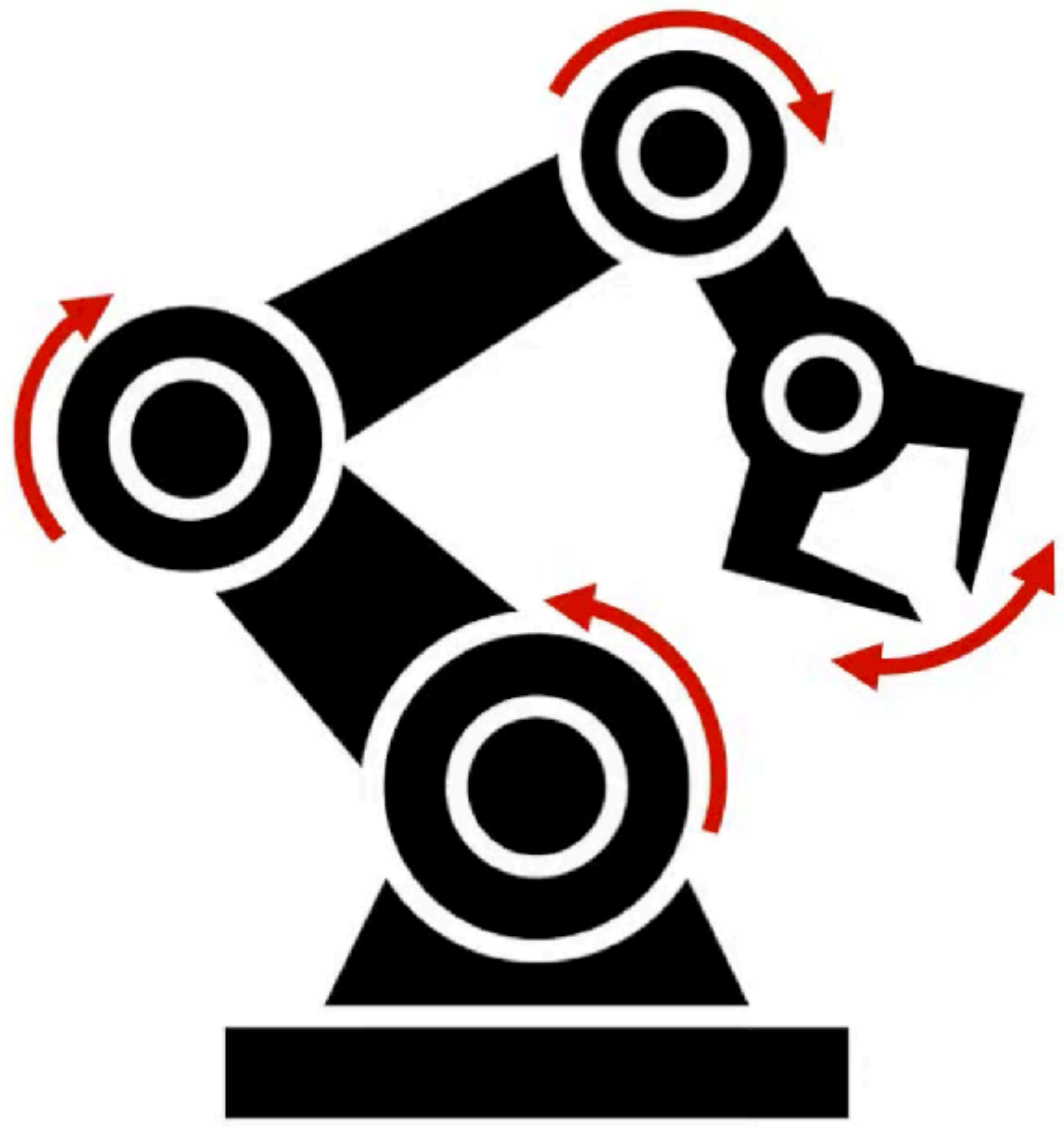
$$C(\mathbf{x}) \equiv \mathbf{0}$$

passive constraint forces (do no work)

Motion of objects is restricted



1 degree of freedom (dof)



4 dofs

Slide from Matthias Müller, Ten Minute Physics

Methods for solving constraints

Springs

- pull system state towards $C = \mathbf{0}$
- must be stiff to be effective, leads to numerical stiffness

Reduced coordinates

- parameterize the set of feasible states—great for special problems, hard to use in general

Constraint forces

- solve for forces that ensure $C = \dot{C} = \ddot{C} = \mathbf{0}$
- works great but requires solving global systems

Post-step projection

- first compute timestep ignoring constraints, then fix the constraints

Position-based dynamics

A post-step projection type method

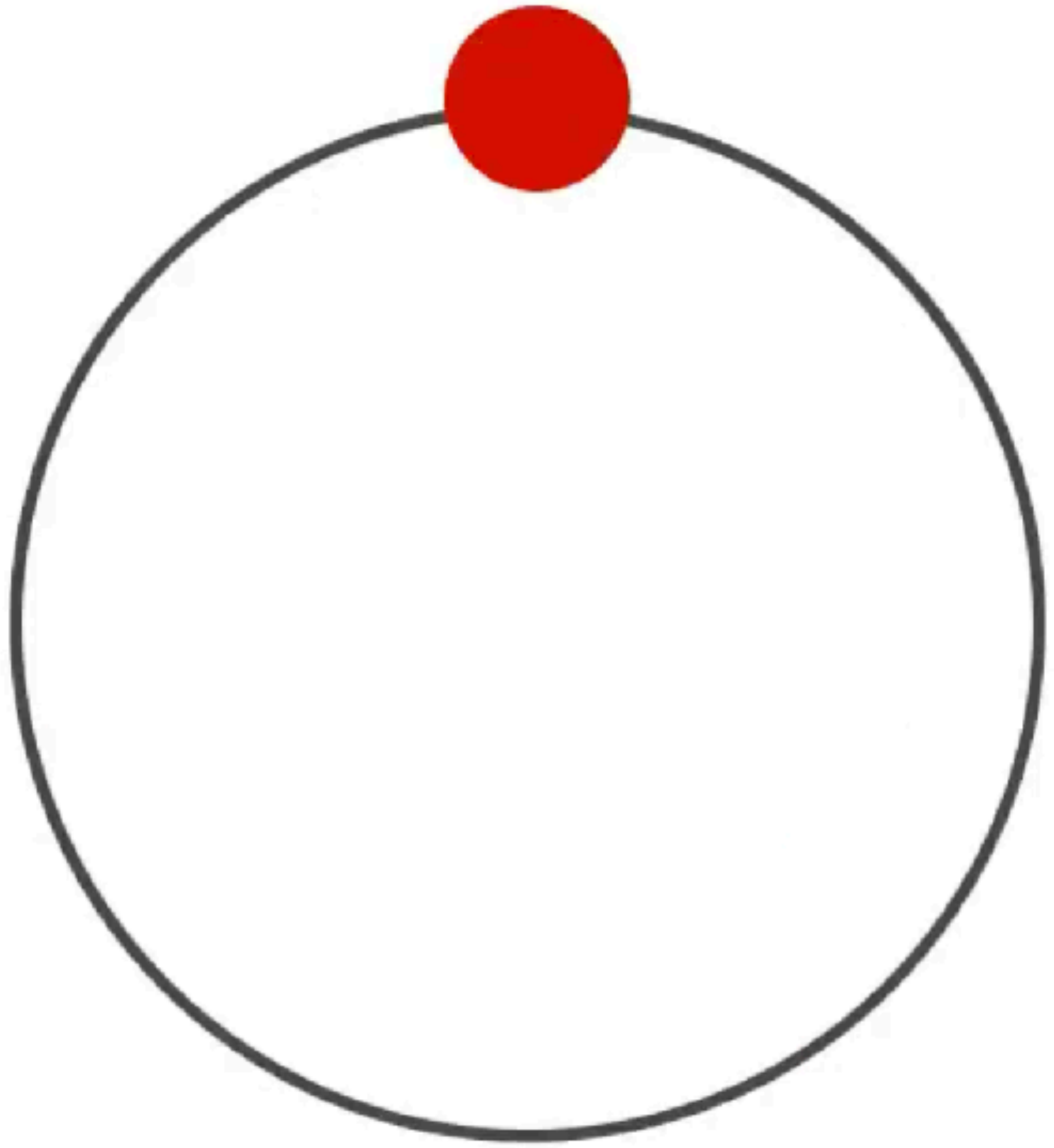
Carefully chooses methods for simple implementation

- projects end-step positions onto constraints then defines velocity to fit the projected positions
- solves for positions using Gauss-Seidel iteration
- computing a G-S iteration turns out to involve simple per-constraint steps
- degrades gracefully for infeasible constraints
- can simulate deformable objects by just not letting the iteration converge

PBD history

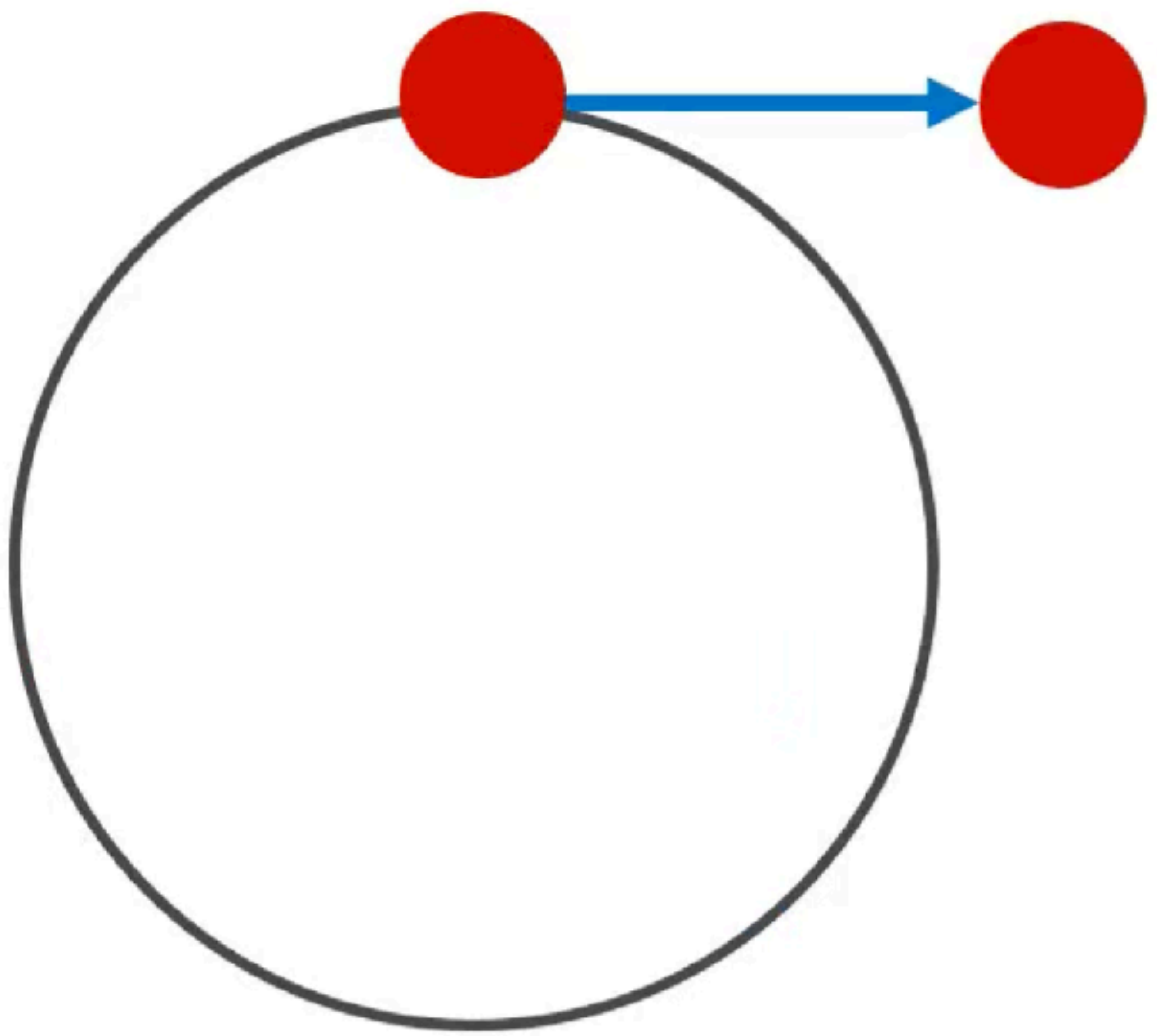
- Müller et al. 2006 “PBD” introduced basic mechanism
- Macklin et al. 2016 “XPBD” cleaned it up and improved handling of deformables
- Macklin et el. 2019 “Small Steps” further refined the solver iteration

The simplest way to fix the velocity



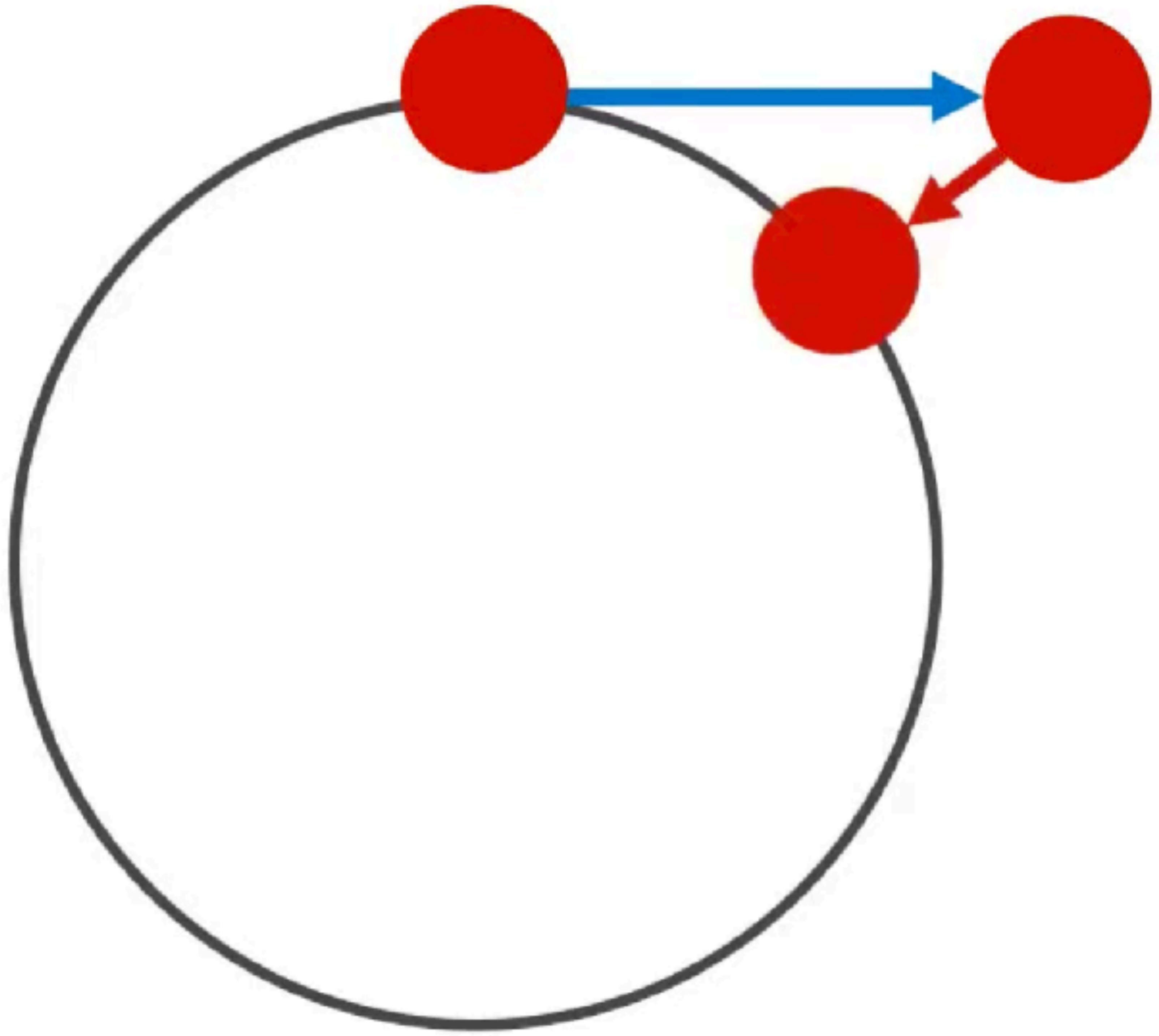
Slide from Matthias Müller, Ten Minute Physics

The simplest way to fix the velocity



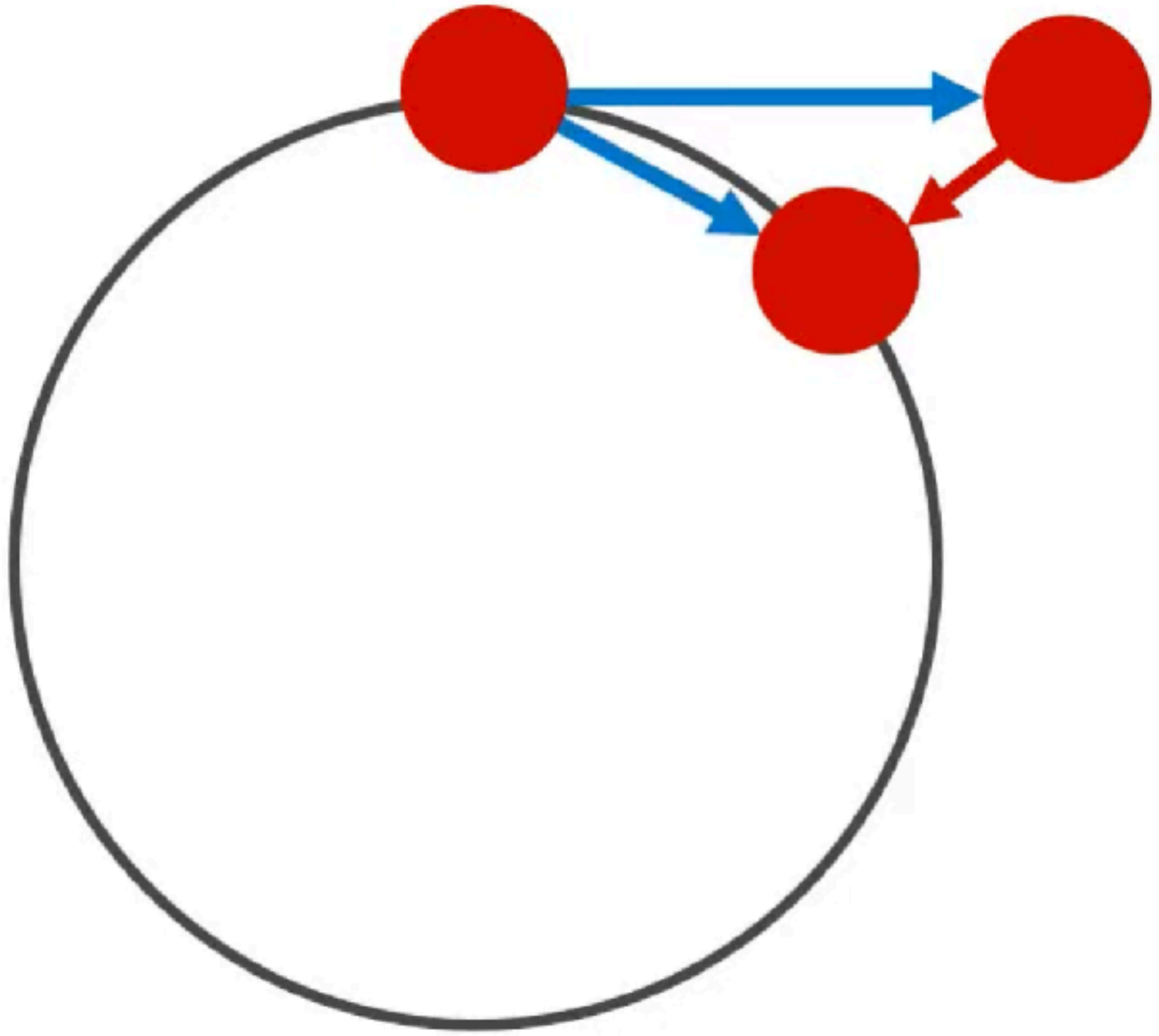
Slide from Matthias Müller, Ten Minute Physics

The simplest way to fix the velocity



Slide from Matthias Müller, Ten Minute Physics

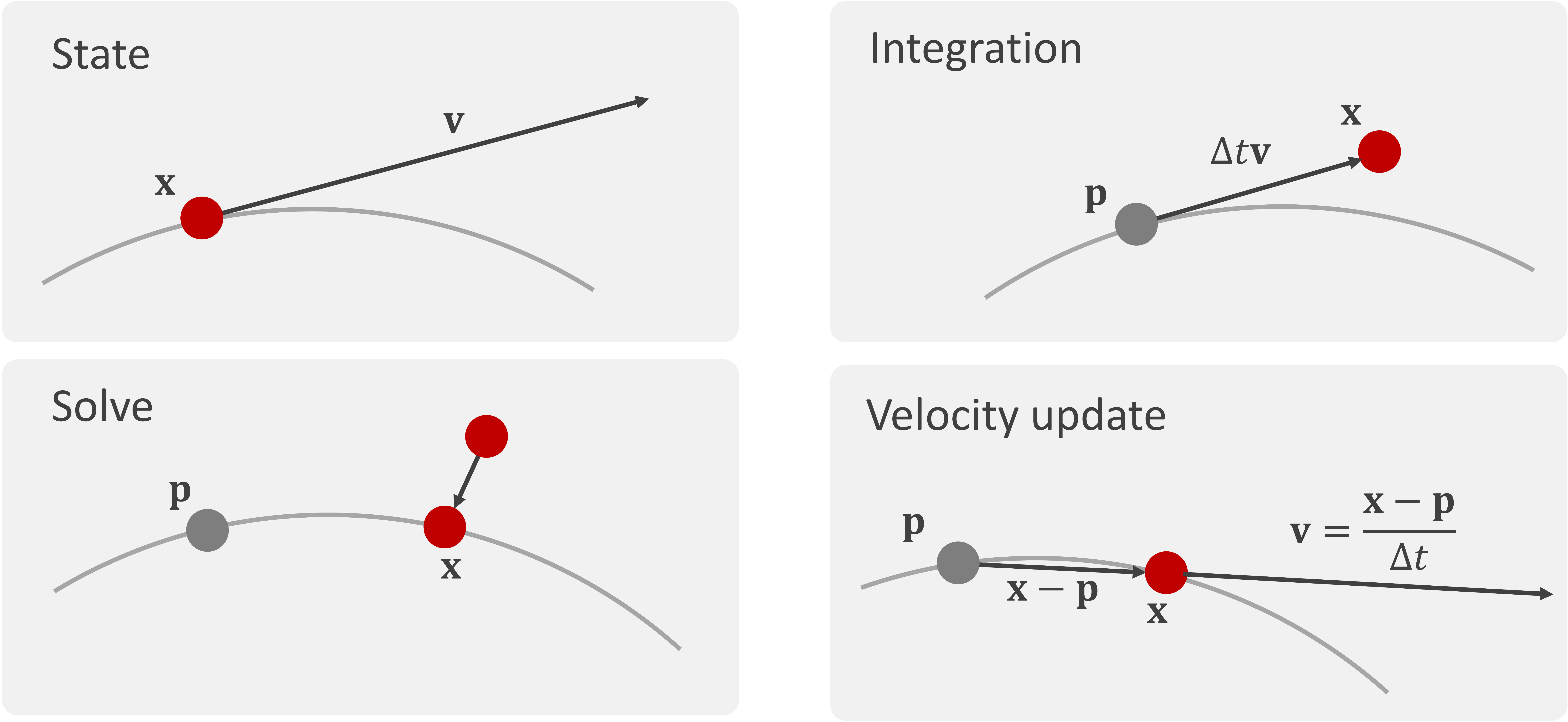
The simplest way to fix the velocity



velocity = (position at the end of the time step
- position at the beginning of the time step) / dt

Slide from Matthias Müller, Ten Minute Physics

Bead on a Wire



PBD = integrator and solver!

Slide from Matthias Müller, Ten Minute Physics

PBD Algorithm

while simulating
  for all particles i
    $\mathbf{v}_i \leftarrow \mathbf{v}_i + \Delta t\ \mathbf{g}$
    $\mathbf{p}_i \leftarrow \mathbf{x}_i$
    $\mathbf{x}_i \leftarrow \mathbf{x}_i + \Delta t\ \mathbf{v}_i$

  for all constraints C
    solve($C, \Delta t$)

  for all particles i
    $\mathbf{v}_i \leftarrow (\mathbf{x}_i - \mathbf{p}_i)/\Delta t$

solve($C, \Delta t$):

for all particles i of C
  compute $\Delta \mathbf{x}_i$
  $\mathbf{x}_i \leftarrow \mathbf{x}_i + \Delta \mathbf{x}_i$

Slide from Matthias Müller, Ten Minute Physics

Iterations vs. Sub-steps

while simulating
  for all particles i
    $\mathbf{v}_i \leftarrow \mathbf{v}_i + \Delta t\ \mathbf{g}$
    $\mathbf{p}_i \leftarrow \mathbf{x}_i$
    $\mathbf{x}_i \leftarrow \mathbf{x}_i + \Delta t\ \mathbf{v}_i$
  for n iterations
    for all constraints C
      solve$(C, \Delta t)$
  for all particles i
    $\mathbf{v}_i \leftarrow (\mathbf{x}_i - \mathbf{p}_i)/\Delta t$

$\Delta t_s \leftarrow \Delta t / n$
while simulating
  for n substeps
    for all particles i
      $\mathbf{v}_i \leftarrow \mathbf{v}_i + \Delta t_s \mathbf{g}$
      $\mathbf{p}_i \leftarrow \mathbf{x}_i$
      $\mathbf{x}_i \leftarrow \mathbf{x}_i + \Delta t_s \mathbf{v}_i$
    for all constraints C
      solve$(C, \Delta t_s)$
    for all particles i
      $\mathbf{v}_i \leftarrow (\mathbf{x}_i - \mathbf{p}_i)/\Delta t_s$

- Spending fixed time budget with sub-steps is much is more effective than with iterations!
- XPBD much simpler (no λ tracking)

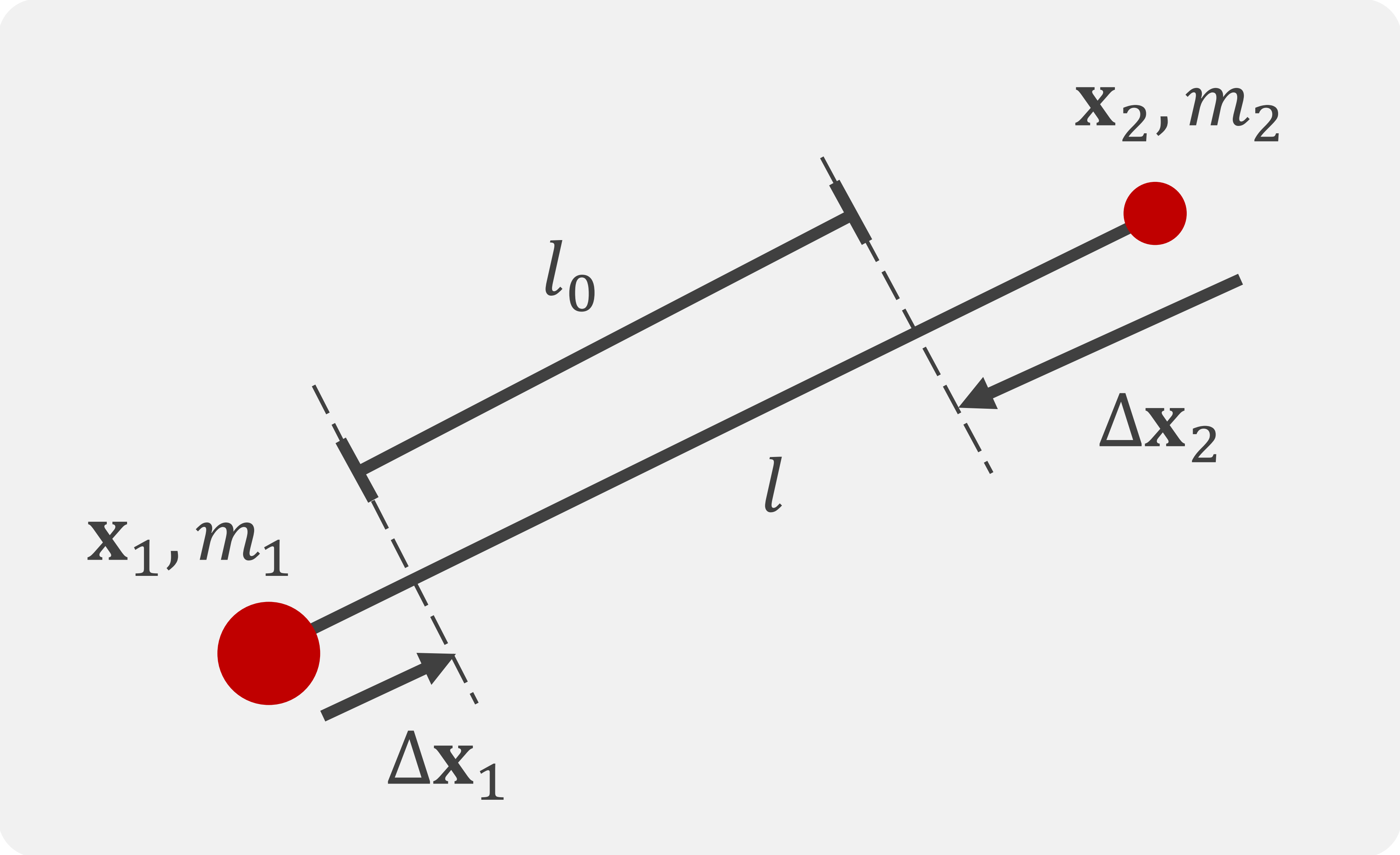
Slide from Matthias Müller, Ten Minute Physics

Distance Constraint

- Rest distance l_0
- Current distance l
- Masses m_i
- Inverse masses $w_i = 1/m_1$

$$\Delta \mathbf{x}_1 = \frac{w_1}{w_1 + w_2}(l - l_0)\frac{\mathbf{x}_2 - \mathbf{x}_1}{|\mathbf{x}_2 - \mathbf{x}_1|}$$

$$\Delta \mathbf{x}_2 = -\frac{w_2}{w_1 + w_2}(l - l_0)\frac{\mathbf{x}_2 - \mathbf{x}_1}{|\mathbf{x}_2 - \mathbf{x}_1|}$$



Slide from Matthias Müller, Ten Minute Physics

General Constraint

$\mathbf{x}_1$

$\mathbf{x}_2$

…

$\mathbf{x}_n$

particles participating in constraint

$C(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$

constraint function

C

constraint error (number)

For the distance constraint:

$$C_{\text{dist}}(\mathbf{x}_1, \mathbf{x}_2) = |\mathbf{x}_2 - \mathbf{x}_1| - l_0$$

Slide from Matthias Müller, Ten Minute Physics

Constraint Gradient ∇C

$\nabla C(\mathbf{x})$ is a vector

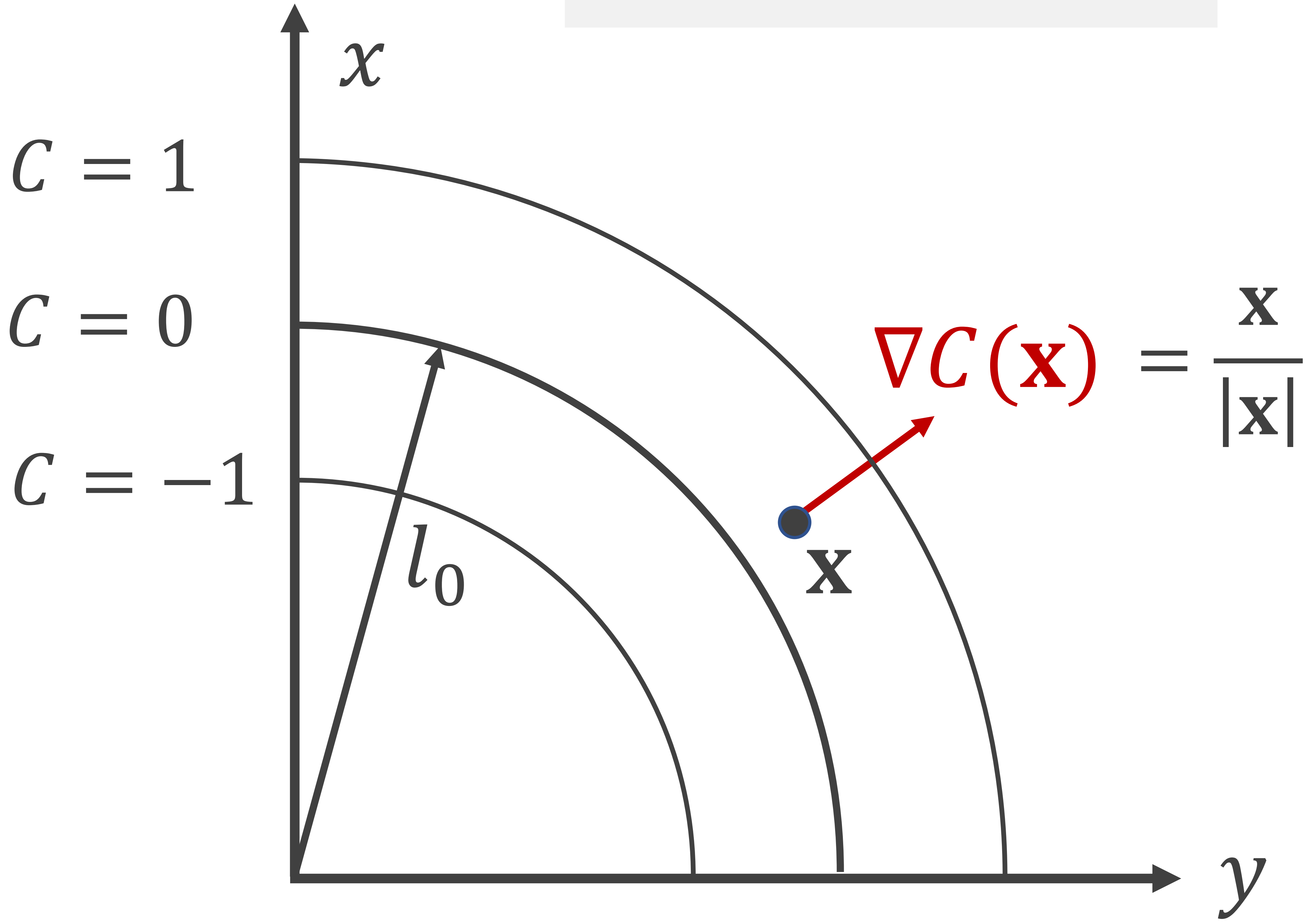
Vector direction:

- direction in which C increases the most

Vector length:

- how much C changes when moving $\mathbf{x}$ by one unit

$C_{\text{dist}} = |\mathbf{x}| - l_0$



Slide from Matthias Müller, Ten Minute Physics

Solving a General Constraint (PBD)

Compute the scalar value λ (same for all participating particles):

$$\lambda = \frac{-\mathrm{C}}{w_1|\nabla C_1|^2 + w_2|\nabla C_2|^2 + \cdots + w_n|\nabla C_n|^2}$$

∇C_i: How to move $\mathbf{x}_i$ for a maximal increase of C

$|\nabla C_i|^2$ the squared length of ∇C_i

Compute correction for point $\mathbf{x}_i$ as: $\Delta\mathbf{x}_i = \lambda w_i \nabla C_i$

Slide from Matthias Müller, Ten Minute Physics

Making the Constraint Soft

PBD:

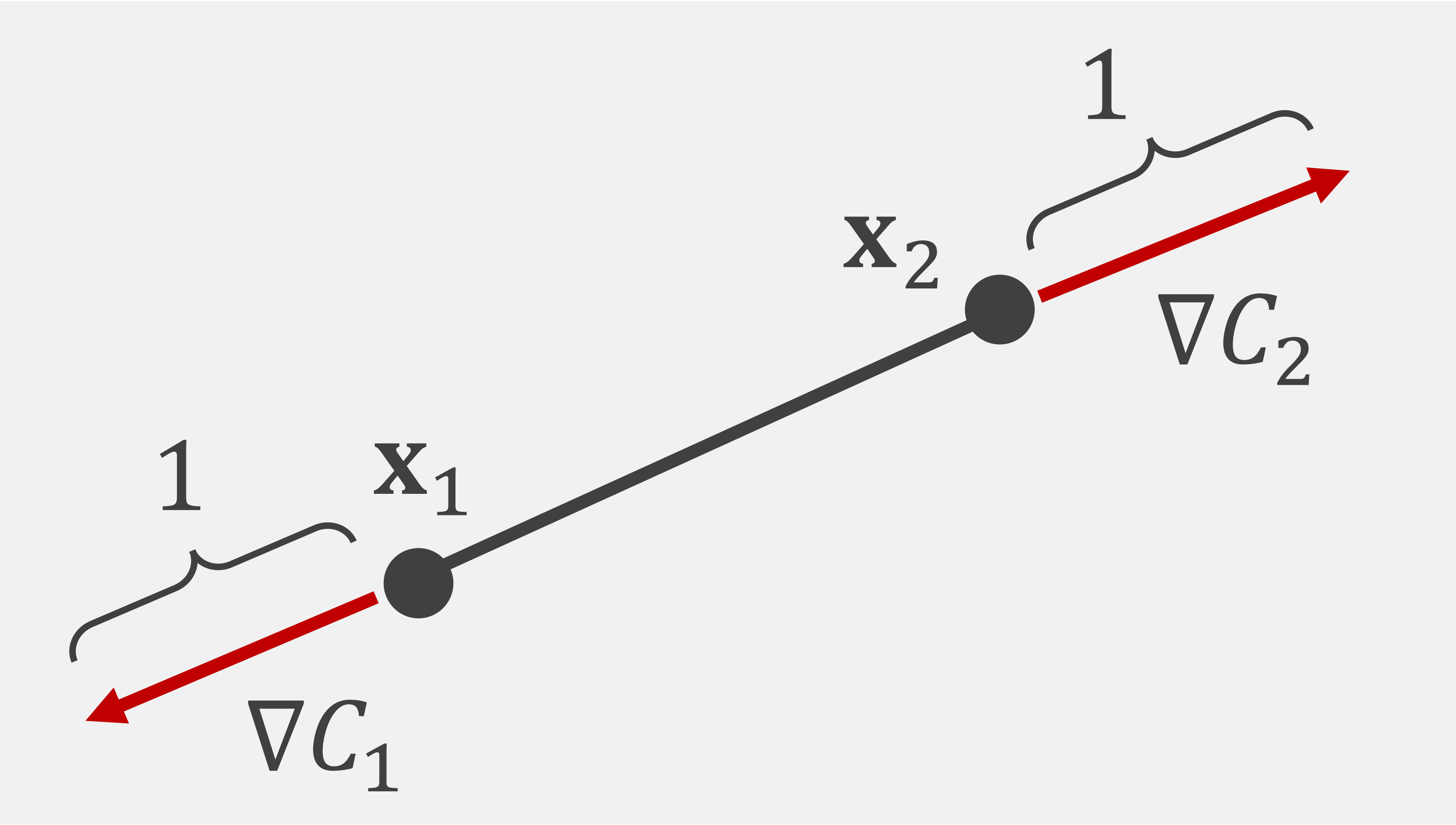
- Scale the correction as $\Delta \mathbf{x}_i = k\, \lambda w_i\, \nabla C_i$
- Stiffness $k \in [0,1]$
- Easy to tune!
- Dependent on time step (stiffer for smaller time steps)

XPBD:

$$\lambda = \frac{-C}{w_1|\nabla C_1|^2 + w_2|\nabla C_2|^2 + \cdots w_n|\nabla C_n|^2 + \frac{\alpha}{\Delta t^2}}$$

- Compliance α is the inverse of physical stiffness
- Infinitely stiff (hard) when $\alpha = 0$

Example: Distance Constraint



$$\nabla C_1 = \frac{\mathbf{x}_1 - \mathbf{x}_2}{|\mathbf{x}_1 - \mathbf{x}_2|}$$

$$\nabla C_2 = \frac{\mathbf{x}_2 - \mathbf{x}_1}{|\mathbf{x}_2 - \mathbf{x}_1|}$$

$$\lambda = \frac{-\mathrm{C}}{w_1|\nabla C_1|^2 + w_2|\nabla C_2|^2 + \cdots + w_n|\nabla C_n|^2} = \frac{-(l - l_0)}{w_1 \cdot 1 + w_2 \cdot 1}$$

$$\Delta\mathbf{x}_1 = \lambda\, w_1\, \nabla C_1 = -\frac{w_1}{w_1 + w_2}(l - l_0)\frac{\mathbf{x}_2 - \mathbf{x}_1}{|\mathbf{x}_2 - \mathbf{x}_1|}$$

Slide from Matthias Müller, Ten Minute Physics