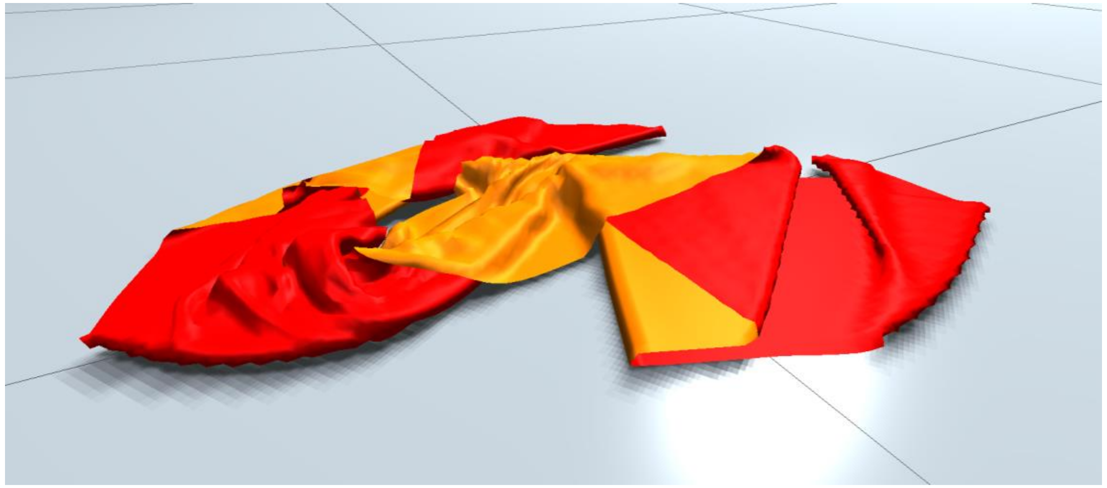
A Few Announcements

- twitter.com/tenMinPhysics
- tenMinPhysics@gmail.com
- Send me your own demos as htmls, I will publish them here

  www.matthiasmueller.info/tenMinutePhysics/contribs

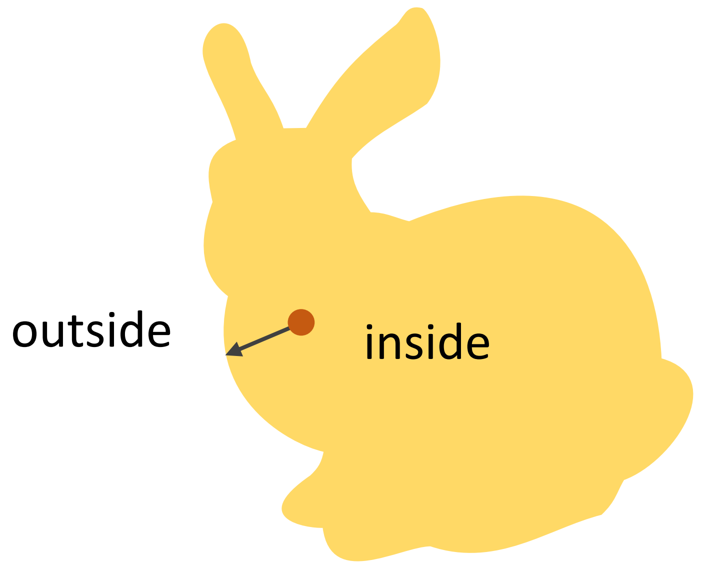


Cloth Self-Collision Handling

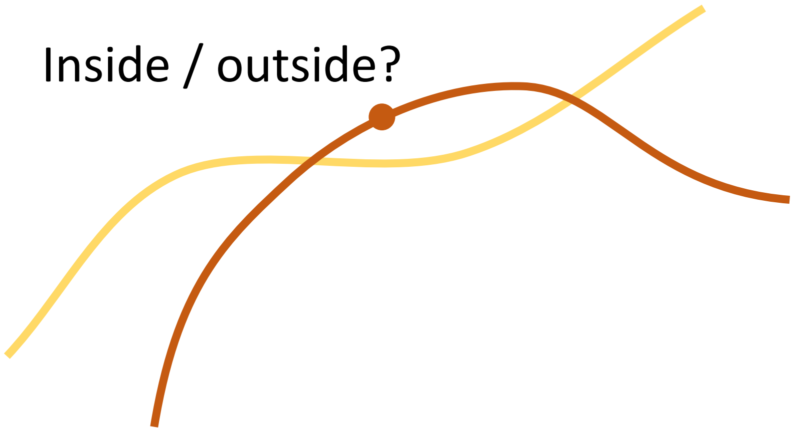
Matthias Müller, Ten Minute Physics

www.matthiasmueller.info/tenMinutePhysics

Tricky Problem!



body



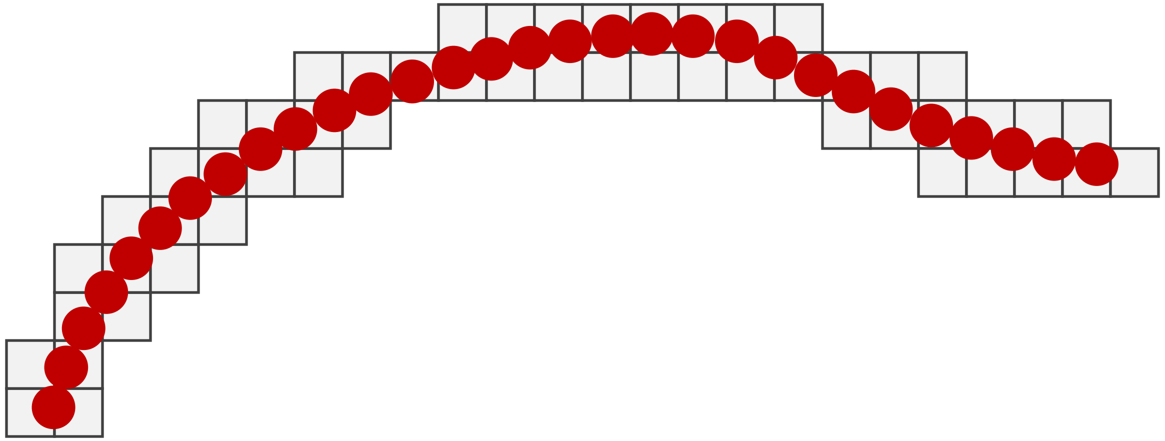
cloth

- Resolving collisions is a global problem, multiple possible solutions
- Start in valid state, make sure entanglement never happens (sometimes not avoidable)

Five Tricks

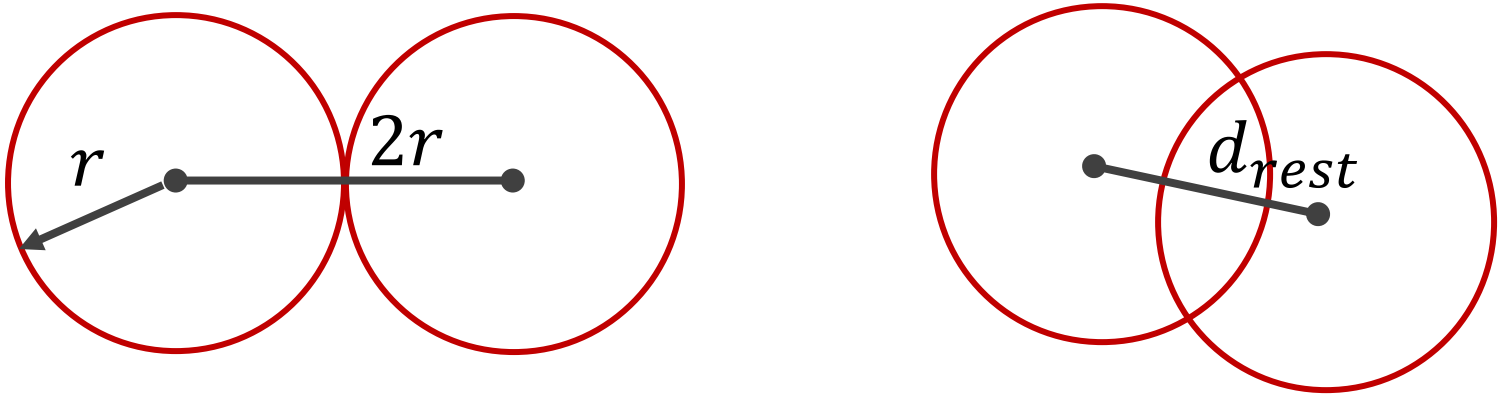
- Use particles and a particle hash
- Use rest distance to avoid jittering
- Use sub-stepping, not CCD
- Enforce maximal velocity
- Use unconditionally stable cloth-cloth friction

1. Use Particles and a Particle Hash



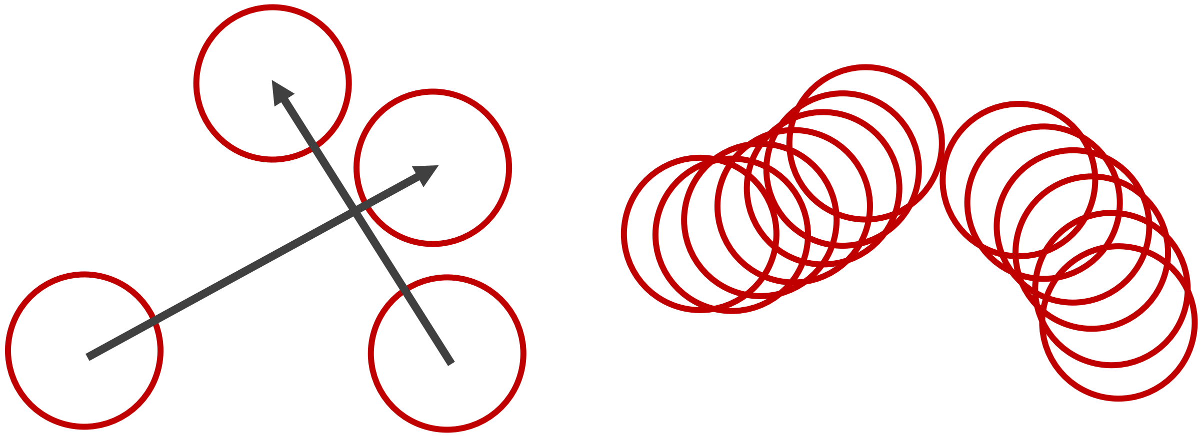
- In general: use many simple primitives instead of few complicated ones!
- Simpler to implement, more degrees of freedom, higher fidelity
- Can use a simple hash for uniform particles (see tutorial number 11)

2. Consider Rest Distance to avoid Jittering



- If $d_{rest} < 2r$ the distance constraints and the collision constraints fight each other
- Set $d_{coll} = \min(2r, d_{rest})$
- compute on the fly from rest positions of particles

3. Use Sub-Stepping, not CCD



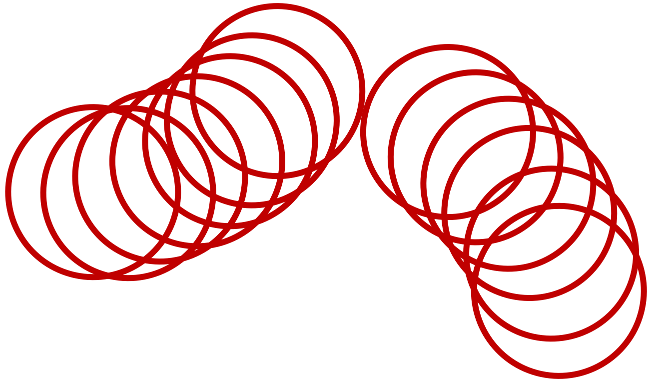
- Continuous collision detection (CCD)
- Overlap test of swept volumes
- Volume touched by objects that rotate and move in curved ways
- Rollback somehow

$\Delta t_s \leftarrow \Delta t / n$
while simulating
    createHash()
    for n sub-steps
        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, Δt_s)
        for all particles i
            $\mathbf{v}_i \leftarrow (\mathbf{x}_i - \mathbf{p}_i) / \Delta t_s$
    renderScene()

→Tutorial 9 (XPBD)

4. Enforce Maximal Velocity

$$v_{max} = \frac{r}{\Delta t_{substep}} = \frac{r}{\Delta t / n_{sub-steps}} = \frac{r\, n_{sub-steps}}{\Delta t}$$



- The larger the number of sub-steps, the larger the limiting velocity!
- Exampe: $r = 1cm,\ n_{substeps} = 20, \Delta t = 1/30s$

  $v_{max} = 6m/s = 20km/h = 13mph$
- Fast running speed, not a severe restriction!

5. Stable Cloth-Cloth Friction

$\mathbf{v}_1 \leftarrow (\mathbf{x}_1 - \mathbf{p}_1)/h$

$\mathbf{v}_2 \leftarrow (\mathbf{x}_2 - \mathbf{p}_2)/h$

$\mathbf{v}_{avg} \leftarrow (\boldsymbol{v}_1 + \mathbf{v}_2)/2$

$\mathbf{x}_1 \leftarrow \mathbf{x}_1 + d(\mathbf{v}_{avg} - \mathbf{v}_1) \cdot h$

$\mathbf{x}_2 \leftarrow \mathbf{x}_2 + d(\mathbf{v}_{avg} - \mathbf{v}_2) \cdot h$

$\mathbf{p}_i$ previous position of particle i

$\mathbf{x}_i$ current position of particle i

- Time step size h cancels, can be omitted
- Damping coefficient $d \in [0,1]$, unconditionally stable
- Make physical by choosing $d = clamp(h \cdot d_{physical}, 0,1)$

Let’s look into the code…