An introduction to Partial Differential Equations

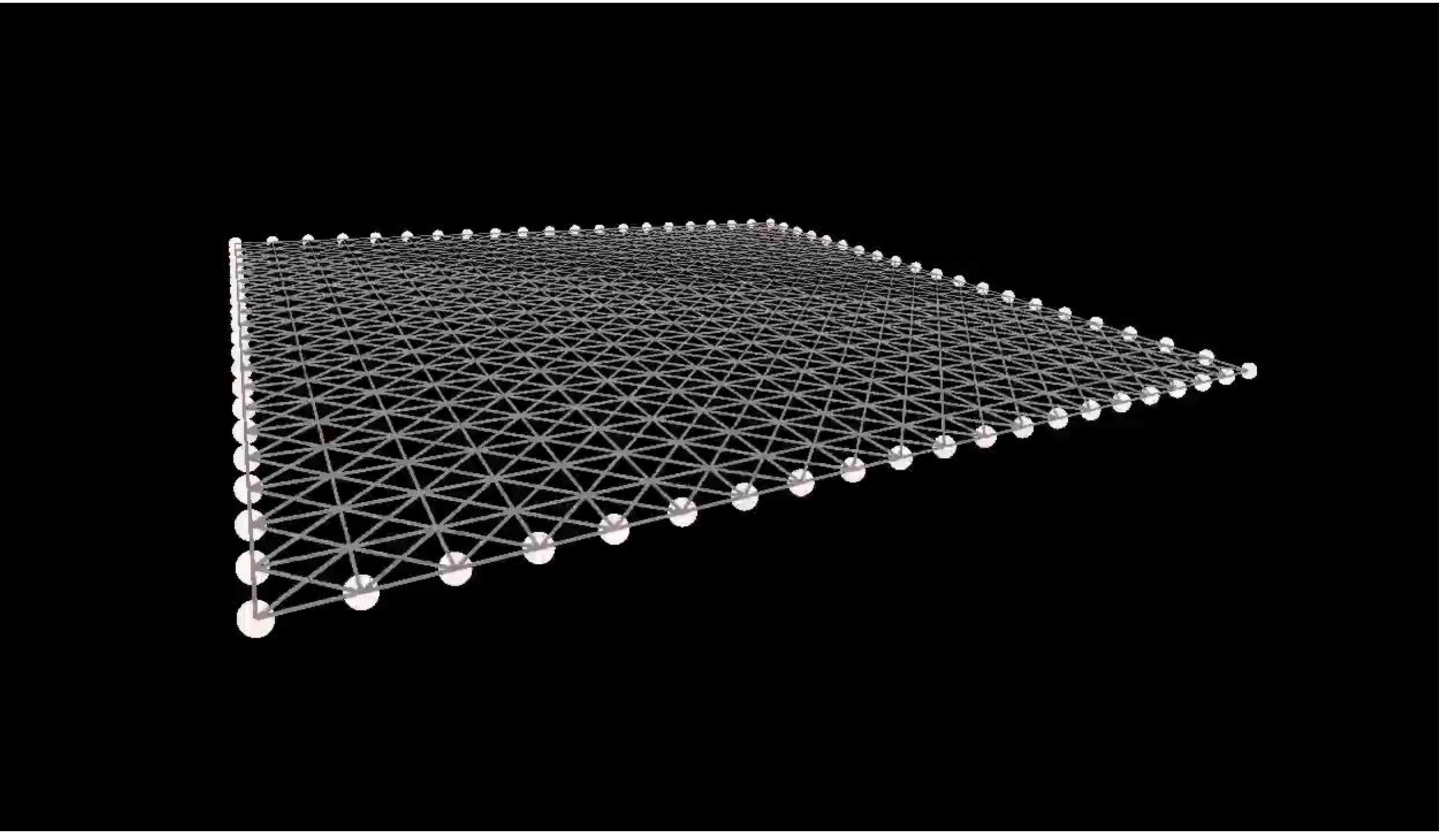
Computer Graphics
CMU 15-462/15-662, Fall 2016

Assignments/grading

- (5%) Warm-up Math (P)Review
  - Written exercises on basic linear algebra and vector calc
- (60%) Five programming assignments
  - Four programming assignments
  - One “go further” fnal assignment (in pairs)
- (7%) Take-home quizzes
- (3%) Class participation
  - In-class/website comments, other contributions to class
- ~~(25%) Midterm / Final~~
- (12.5%) Midterm

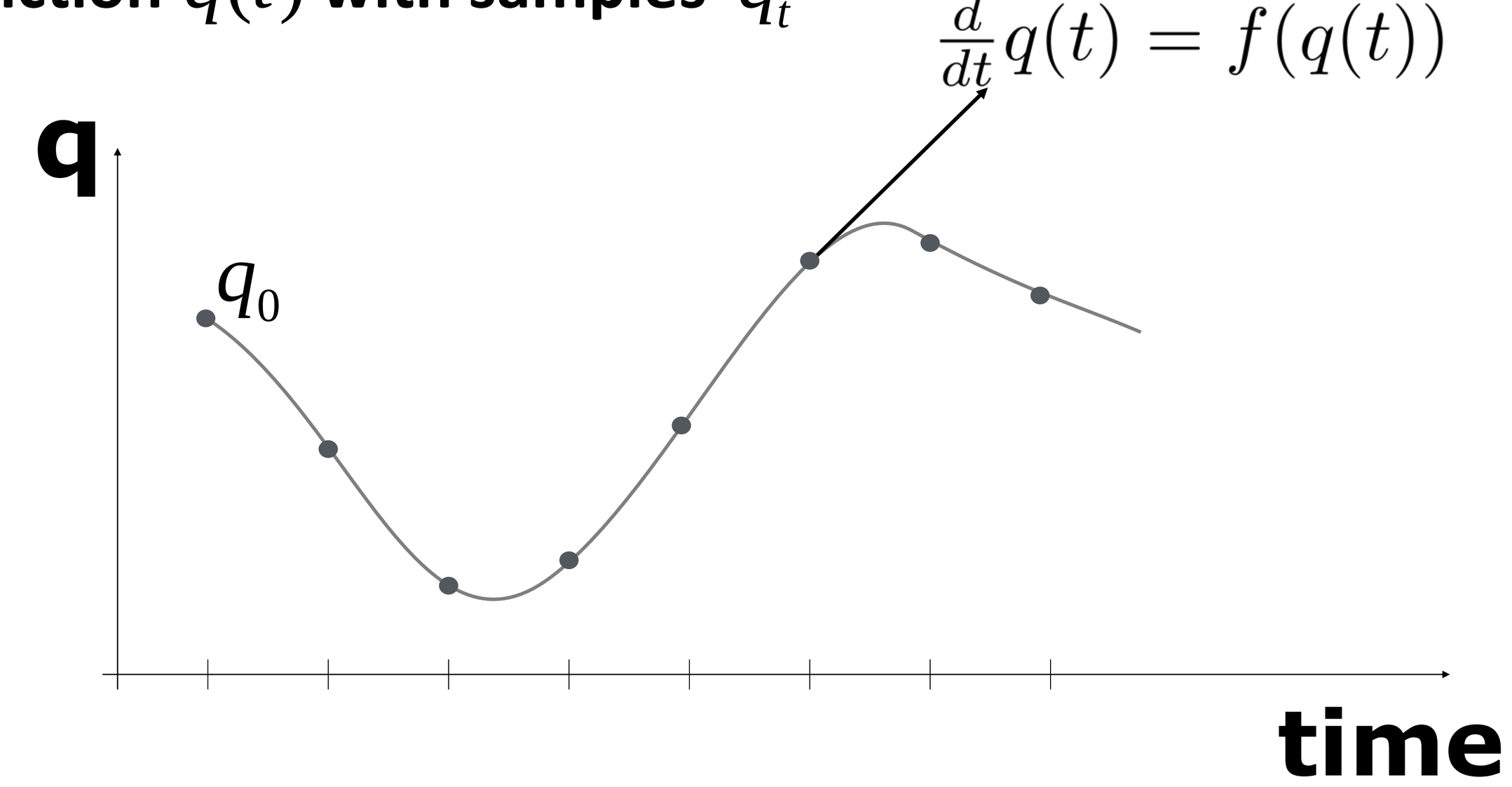
- Final grade : / 87.5

Example: Mass Spring System



Recap: ODEs

- **ODEs: implicitly define a function through its time derivative**
- **Numerical solve: approximate time-continuous function $q(t)$ with samples q_t**



Numerical Integration

- **How do you compute time-discretized samples?**
  - **replace *derivatives* with *differences***

$$\frac{d}{dt}q(t) = f(q(t))$$

$$\Downarrow$$

$$\frac{q_{k+1} - q_k}{\tau} = f(q)$$

new configuration (unknown—want to solve for this!)

current configuration (known)

“time step,” i.e., interval of time between q_k and q_{k+1}

Wait... where do we evaluate the velocity function? At the new or old configuration?

Forward Euler

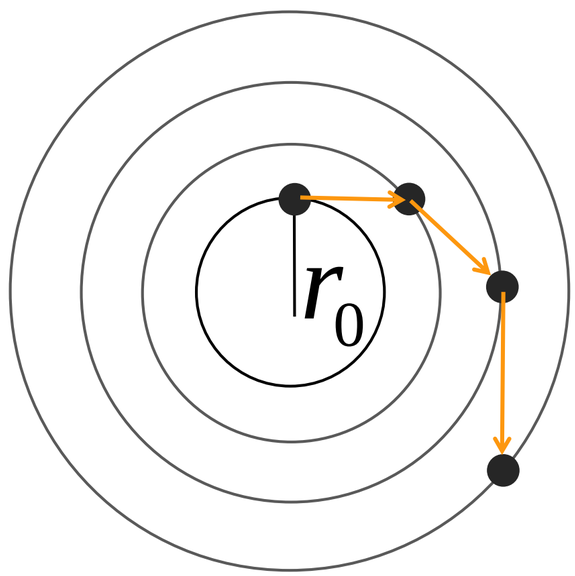
- Simplest scheme: evaluate velocity at current configuration
- New configuration can then be written *explicitly* in terms of known data:

new configuration — current configuration — velocity at current time

$$q_{k+1} = q_k + \tau f(q_k)$$

- Very intuitive: walk a tiny bit in the direction of the velocity
- Unfortunately, not very *stable* ☹
- Consider the following 2D, first order ODE (what does it do?):

$$\dot{q} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} q$$



Forward Euler - Stability Analysis

- Let's consider behavior of forward Euler for simple linear ODE (e.g. temperature of an object):

$$\dot{u} = -au, \quad a > 0$$

- Importantly: *u* should *decay* (exact solution is $u(t)=e^{-at}$)
- Forward Euler approximation is

$$\begin{aligned} u_{k+1} &= u_k - \tau a u_k \\ &= (1 - \tau a) u_k \end{aligned}$$

- Which means after n steps, we have

$$u_n = (1 - \tau a)^n u_0$$

- Decays only if $|1-\tau a| < 1$, or equivalently, if $\tau < 2/a$
- In practice: may need *very small* time steps ("stiff ODE")

Backward Euler

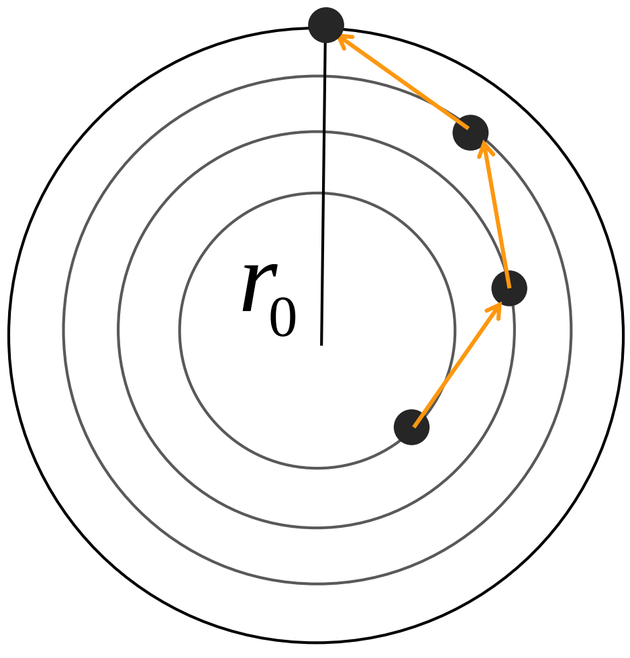
- Let's try something else: evaluate velocity at *next* configuration
- New configuration is then *implicit* - we must solve for it:

new configuration — current configuration — velocity at next time

$$q_{k+1} = q_k + \tau f(q_{k+1})$$

- Much harder to solve, since in general f can be very nonlinear!

$$\dot{q} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} q$$



Backward Euler - Stability Analysis

- **Again consider a simple linear ODE:**

$$\dot{u} = -au, \quad a > 0$$

- **Remember: *u* should *decay* (exact solution is $u(t)=e^{-at}$)**
- **Backward Euler approximation is**

$$(u_{k+1} - u_k)/\tau = -au_{k+1}$$

$$\iff \quad u_{k+1} = \frac{1}{1+\tau a} u_k$$

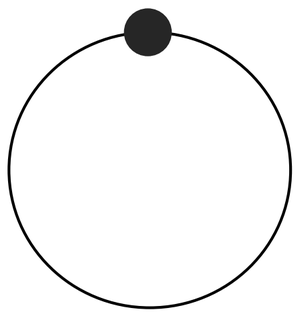
- **Which means after n steps, we have**

$$u_n = \left(\frac{1}{1+\tau a}\right)^n u_0$$

- **Decays if $|1+\tau a| > 1$, which is always true!**
- **⇒ Backward Euler is *unconditionally stable* for linear ODEs**

Symplectic Euler

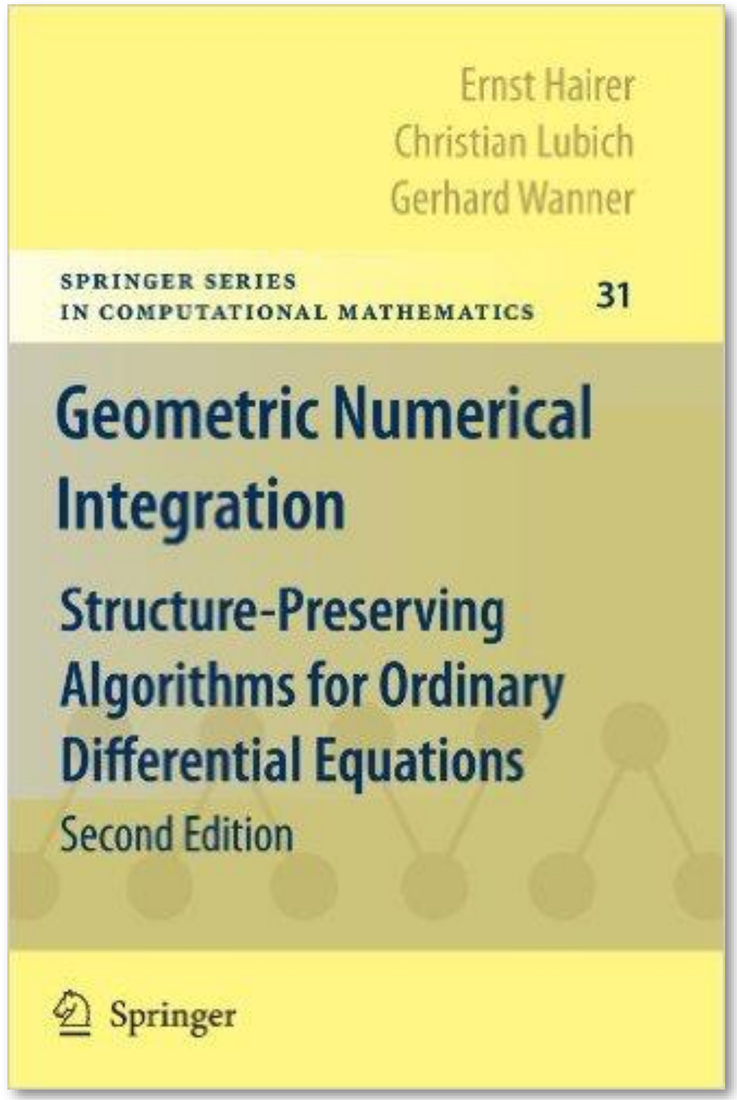
- Backward Euler was stable, but we also saw (empirically) that it exhibits *numerical damping* (damping not found in original eqn.)
- Nice alternative is symplectic Euler (for 2nd order ODEs)
  - update velocity using current configuration
  - update configuration using *new* velocity
- Easy to implement; used often in practice (or leapfrog, Verlet, ...)
- Energy is conserved *almost exactly*, forever. Is that desirable?



(Proof? The analysis is not so easy...)

Numerical Integrators

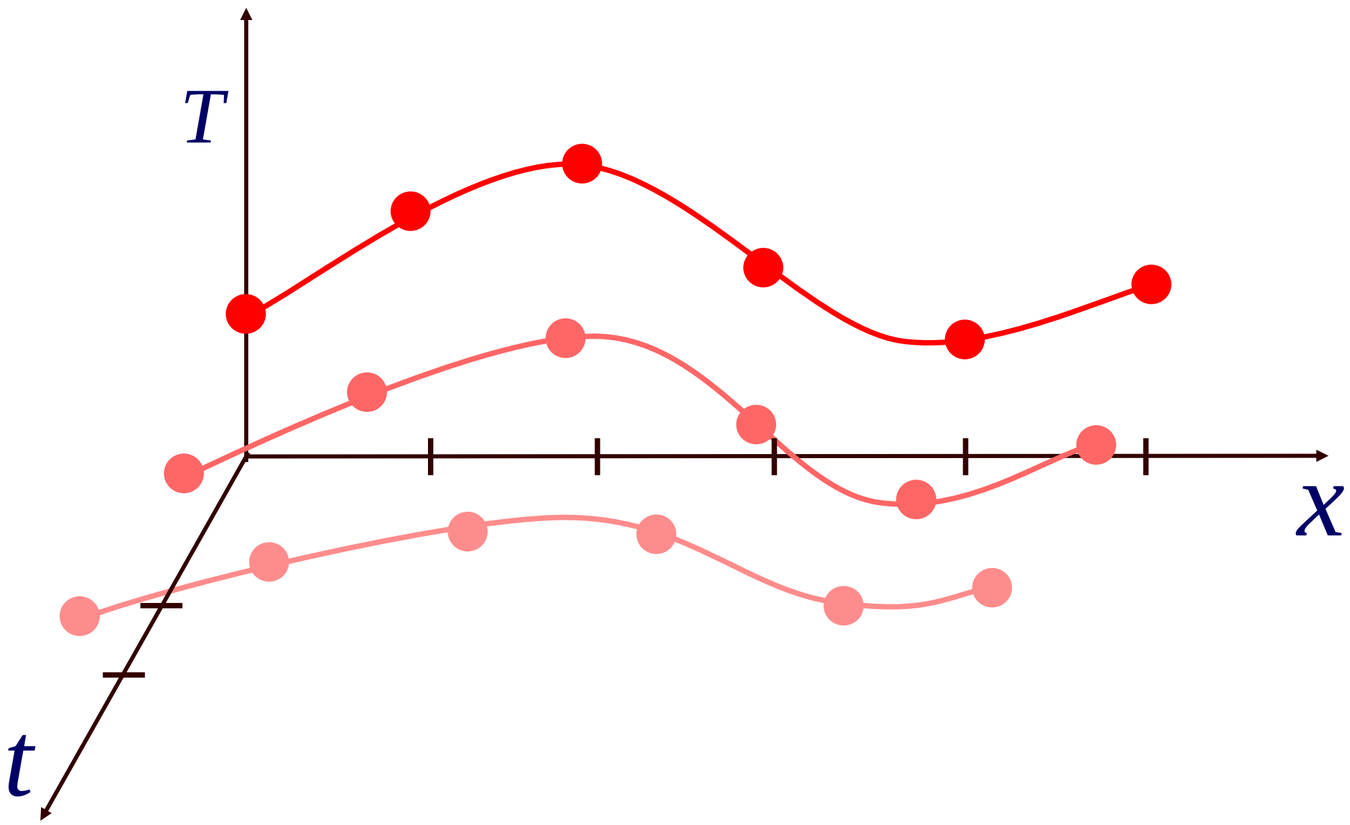
- Barely scratched the surface
- *Many* different integrators
- Why? Because many notions of "good":
  - stability
  - accuracy
  - convergence
  - conservation, symmetry, ...
  - computational efficiency (!)
- No one "best" integrator—*pick the right tool for the job!*
- Could do (at least) an entire course on time integration...
- Great book: Hairer, Lubich, Wanner



Let’s look at an example…

Partial Differential Equations (PDEs)

- ODE: Implicitly describe function in terms of its time derivatives
- Like any implicit description, have to solve for actual function
- PDE: Also include spatial derivatives in description
- An example: temperature of a particle
  - on a wire



$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2}$$

To make a long story short…

- **Solving ODE looks like *"add a little velocity each time"***

$$q_{k+1} = q_k + \tau f(q)$$

- **Solving a PDE looks like *"use neighbor information to get velocity (…and then add a little velocity each time)"***

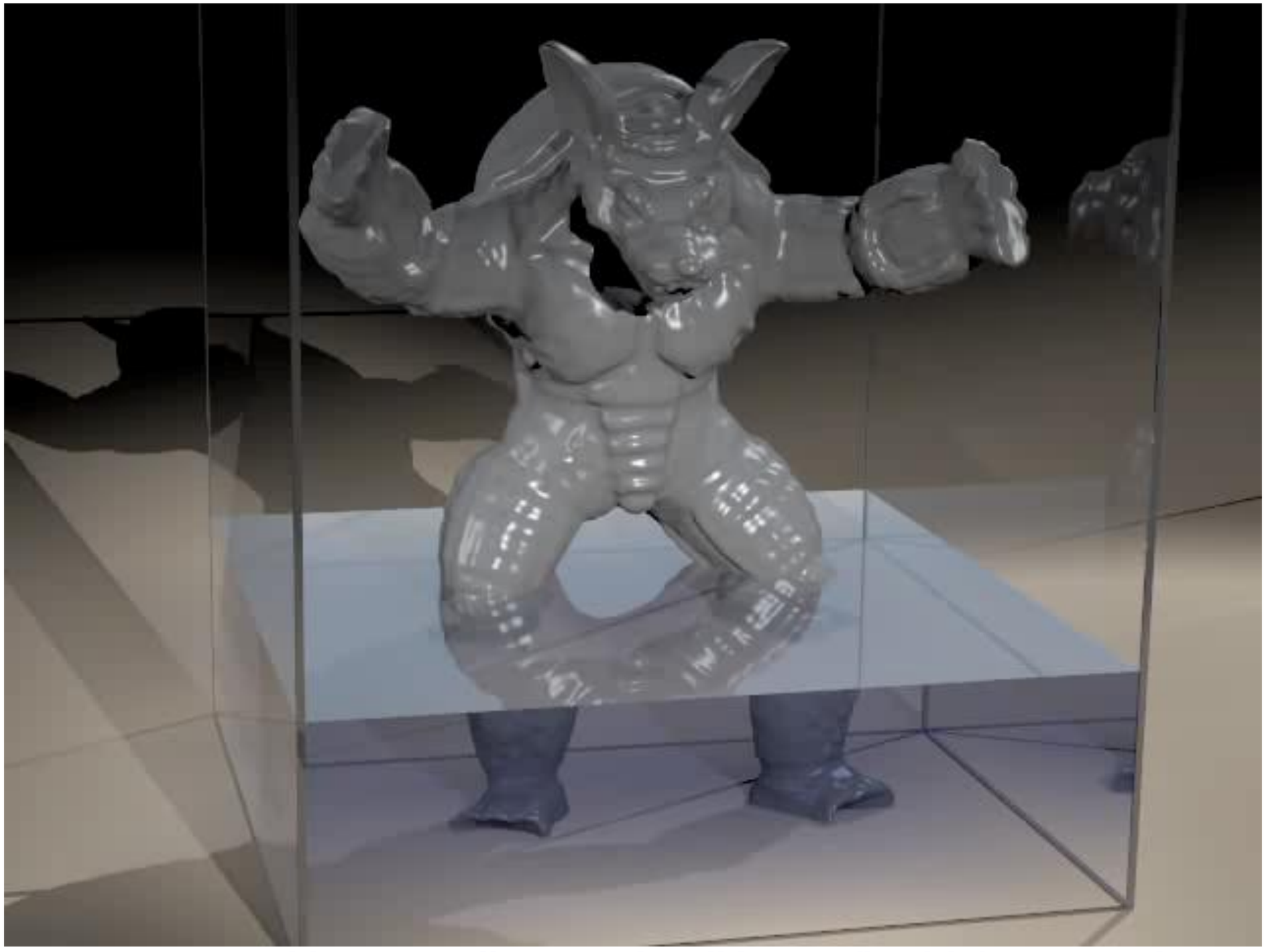
	1	
1	-4	1
	1	

$f(q)$

$$q_{k+1} = q_k + \tau f(q)$$

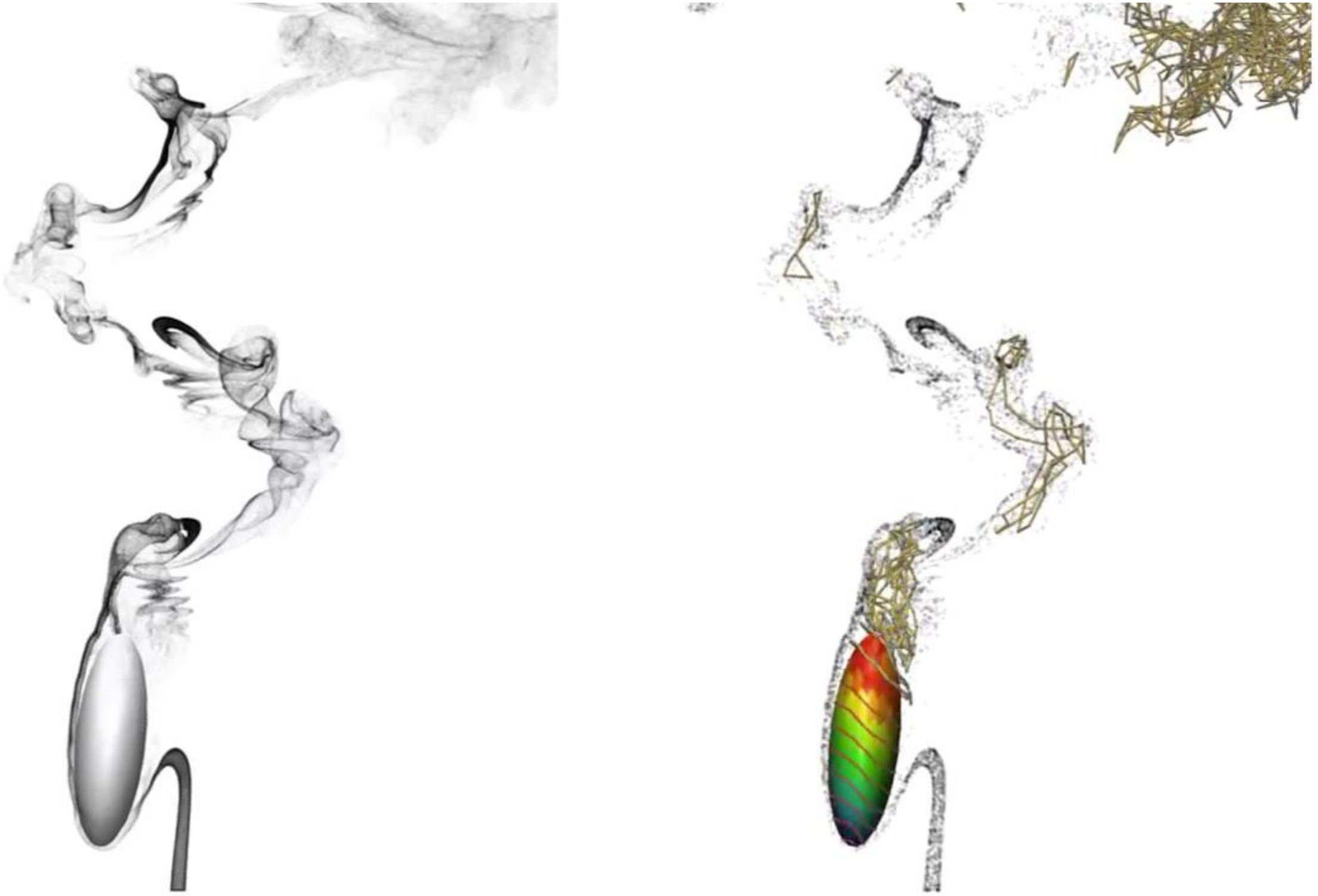
…obviously there is a *lot* more to say here!

Fluid Simulation in Graphics



Losasso, F., Shinar, T. Selle, A. and Fedkiw, R., *"Multiple Interacting Liquids"*

Smoke Simulation in Graphics

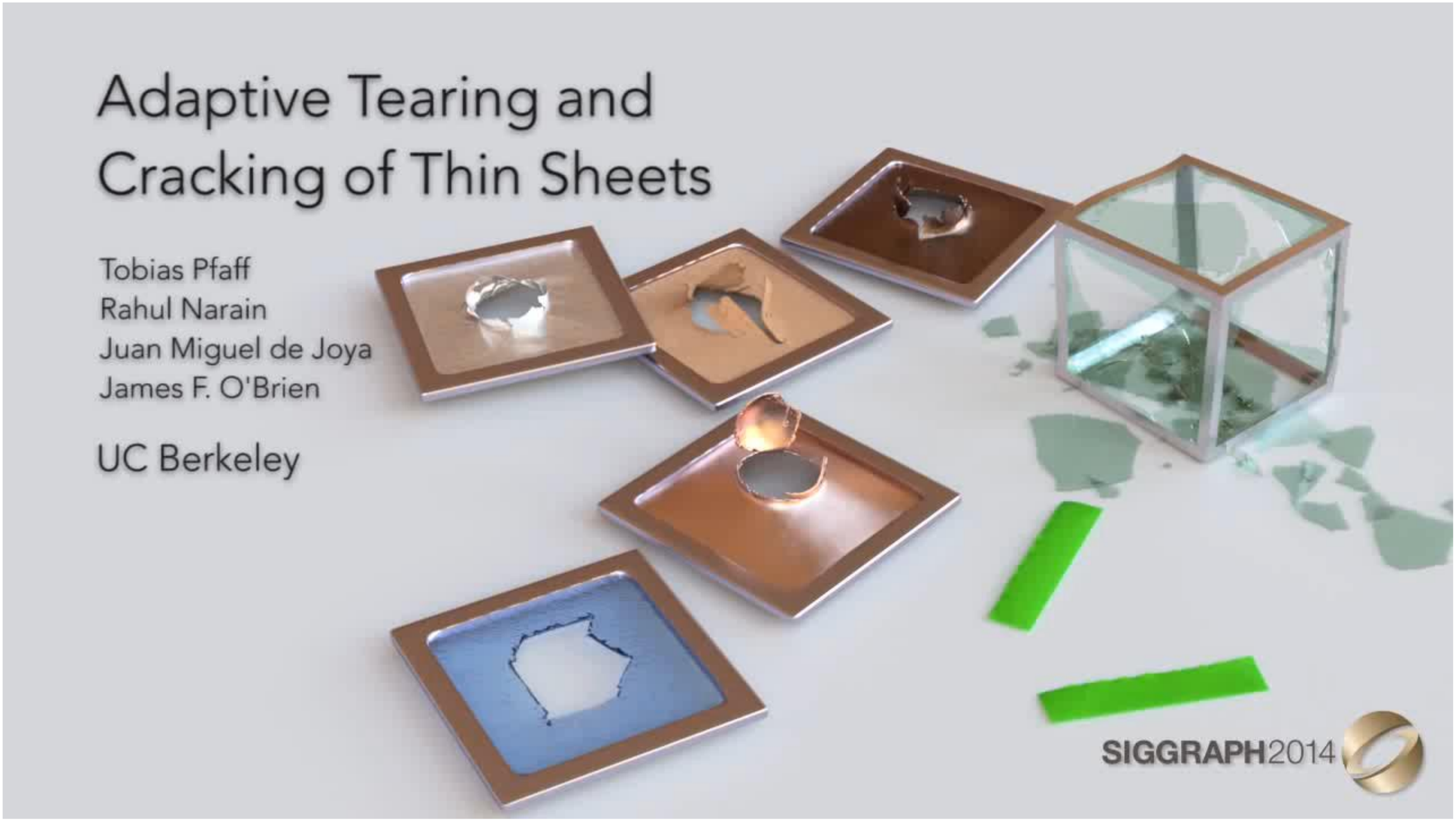


S. Weißmann, U. Pinkall. "*Filament-based smoke with vortex shedding and variational reconnection*"

Cloth Simulation in Graphics

Zhili Chen, Renguo Feng and Huamin Wang, *"Modeling friction and air effects between cloth and deformable bodies"*

Fracture in Graphics



Tobias Pfaff, Rahul Narain, Juan Miguel de Joya, James F. O'Brien,
"Adaptive Tearing and Cracking of Thin Sheets"

Hair Simulation in Graphics

Danny M. Kaufman, Rasmus Tamstorf, Breannan Smith, Jean-Marie Aubry, Eitan Grinspun, "*Adaptive Nonlinearity for Collisions in Complex Rod Assemblies*"

Definition of a PDE

- **Want to solve for a function of time *and* space**

$$u(t, x)$$

time space

- **Function given implicitly in terms of derivatives:**

$$\dot{u}, \ddot{u}, \frac{d}{dt^3}u, \frac{d}{dt^4}u, \ldots$$

any combination of time derivatives

$$\frac{\partial u}{\partial x_1}, \frac{\partial u}{\partial x_2}, \frac{\partial^2 u}{\partial x_1 \partial x_2}, \frac{\partial^m + nu}{\partial x_i^m \partial x_j^n}, \ldots$$

plus any combination of space derivatives

- **Example:**

$$\dot{u} + uu' = au''$$

(Burgers' equation)

Anatomy of a PDE

- **Linear vs. nonlinear: how are derivatives combined?**

nonlinear!

$$\dot{u} + u u' = a u''$$

(Burgers' equation)

$$\dot{u} = a u''$$

(diffusion equation)

- **Order: how many derivatives in space & time?**

1st order in time — 2nd order in space

$$\dot{u} + u u' = a u''$$

(Burgers' equation)

2nd order in time — 2nd order in space

$$\ddot{u} = a u''$$

(wave equation)

- **Nonlinear / higher order ⇒ HARDER TO SOLVE!**

Model Equations

- **Fundamental behavior of many important PDEs is well-captured by three model linear equations:**

Solve numerically?

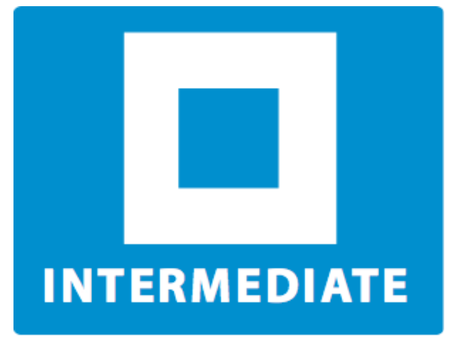
"Laplacian" (more later!)

LAPLACE EQUATION ("ELLIPTIC") $\Delta u = 0$

"what's the smoothest function interpolating the given boundary data"

HEAT EQUATION ("PARABOLIC") $\dot{u} = \Delta u$

"how does an initial distribution of heat spread out over time?"



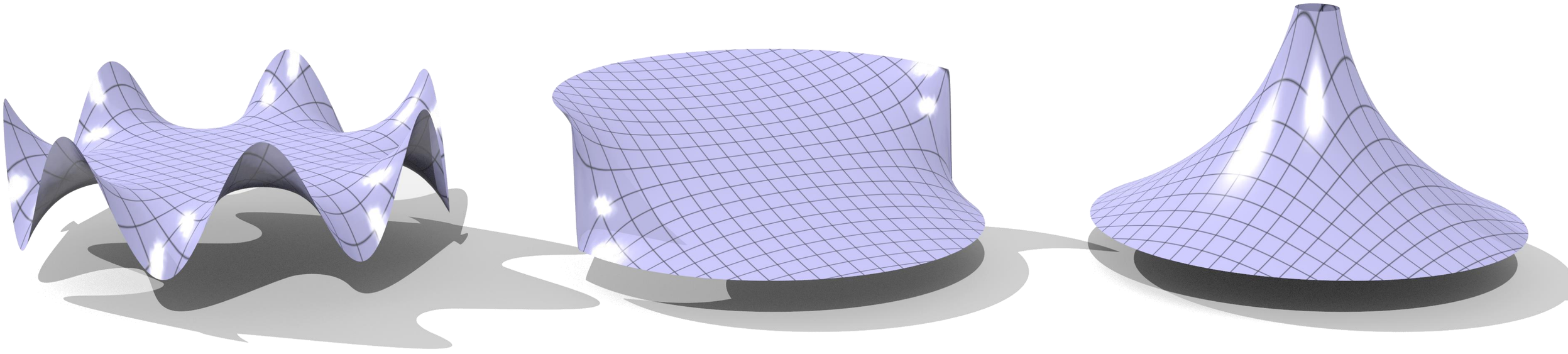
WAVE EQUATION ("HYPERBOLIC") $\ddot{u} = \Delta u$

"if you throw a rock into a pond, how does the wavefront evolve over time?"

[NONLINEAR + HYPERBOLIC + HIGH-ORDER]

Elliptic PDEs / Laplace Equation

- **“What’s the smoothest function interpolating the given boundary data?”**

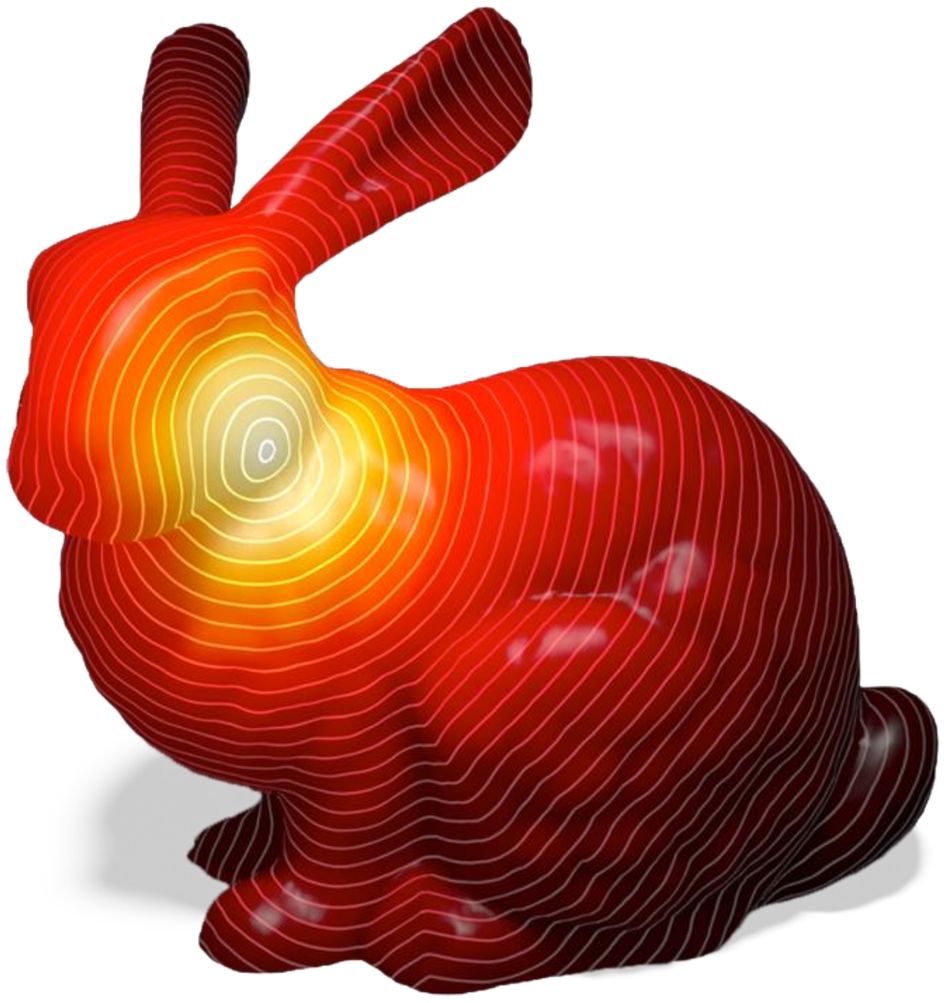


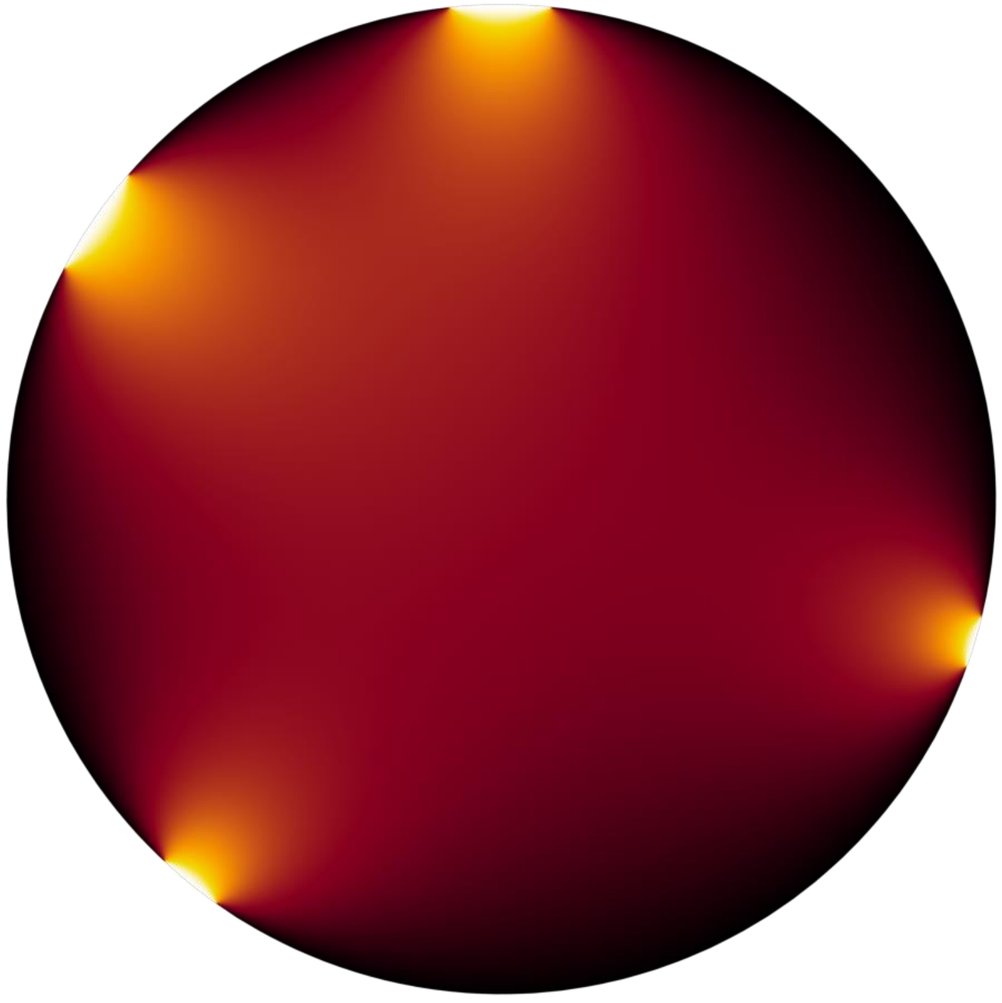
- **Conceptually: each value is at the average of its “neighbors”**
- **Roughly speaking, why is it easier to solve?**
- **Very robust to errors: just keep averaging with neighbors!**

Image from Solomon, Crane, Vouga, “Laplace-Beltrami: The Swiss Army Knife of Geometry Processing”

Parabolic PDEs / Heat Equation

- **"How does an initial distribution of heat spread out over time?"**





- **After a long time, solution is same as Laplace equation!**
- **Models damping / viscosity in many physical systems**

Hyperbolic PDEs / Wave Equation

- **“If you throw a rock into a pond, how does the wavefront evolve over time?”**

- **No steady state solution. Errors made at the beginning will persist for a long time! (hard)**

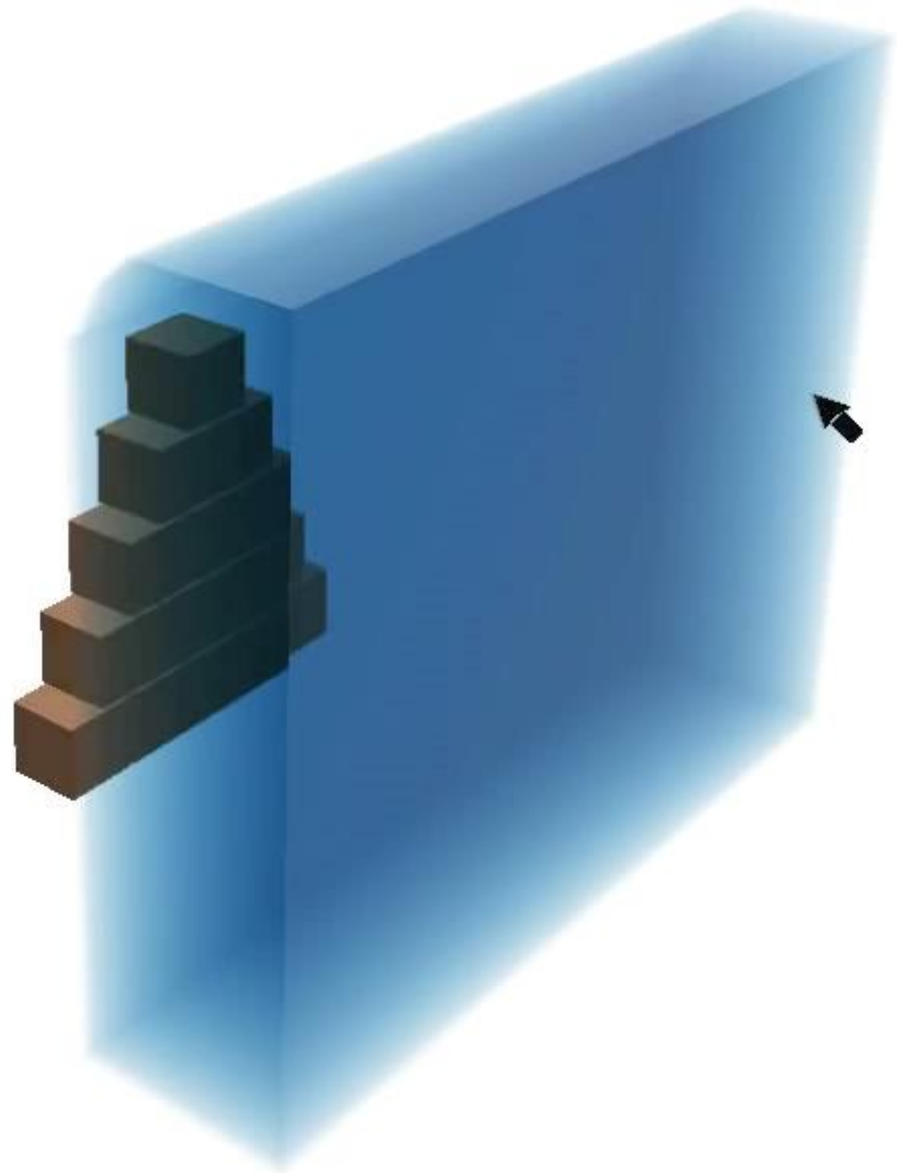
How can we do that?

Numerical Solution of PDEs—Overview

- Like ODEs, many interesting PDEs are difficult or impossible to solve analytically
- Must instead use numerical integration
- Basic strategy:
  - pick a spatial discretization (TODAY)
  - pick a time discretization (forward Euler, backward Euler...)
  - as with ODEs, run a time-stepping algorithm
- Historically, very expensive—only for “hero shots” in movies
- Computers are ever faster...
- More & more use of PDEs in games, interactive tools, ...

Real Time PDE-Based Simulation (Fire)

Real Time PDE-Based Simulation (Water)

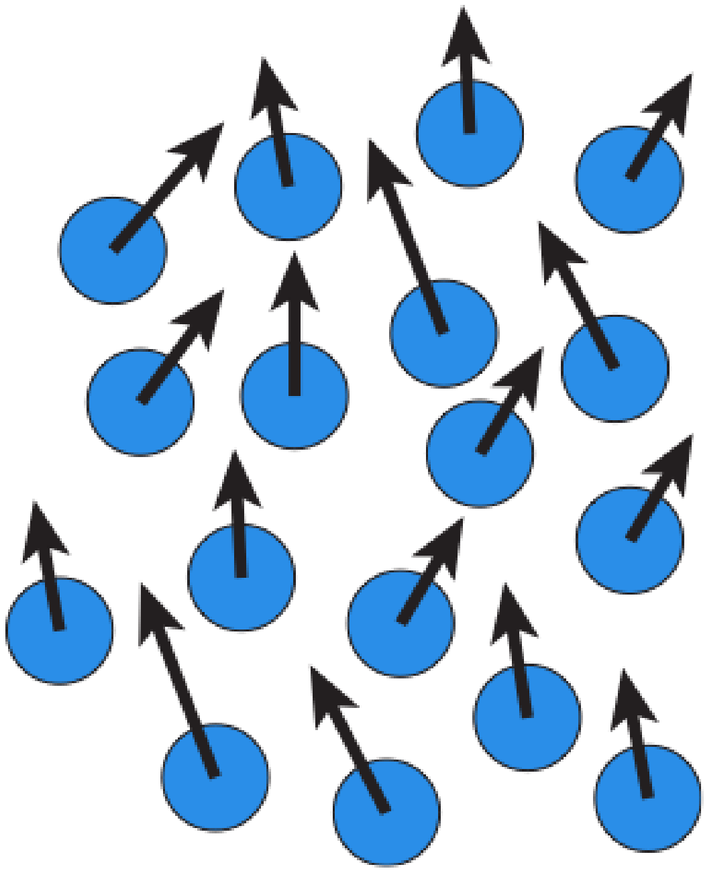


[Nuttapong Chentanez](), [Matthias Müller](), ***"Real-time Eulerian water simulation using a restricted tall cell grid"***

Lagrangian vs. Eulerian

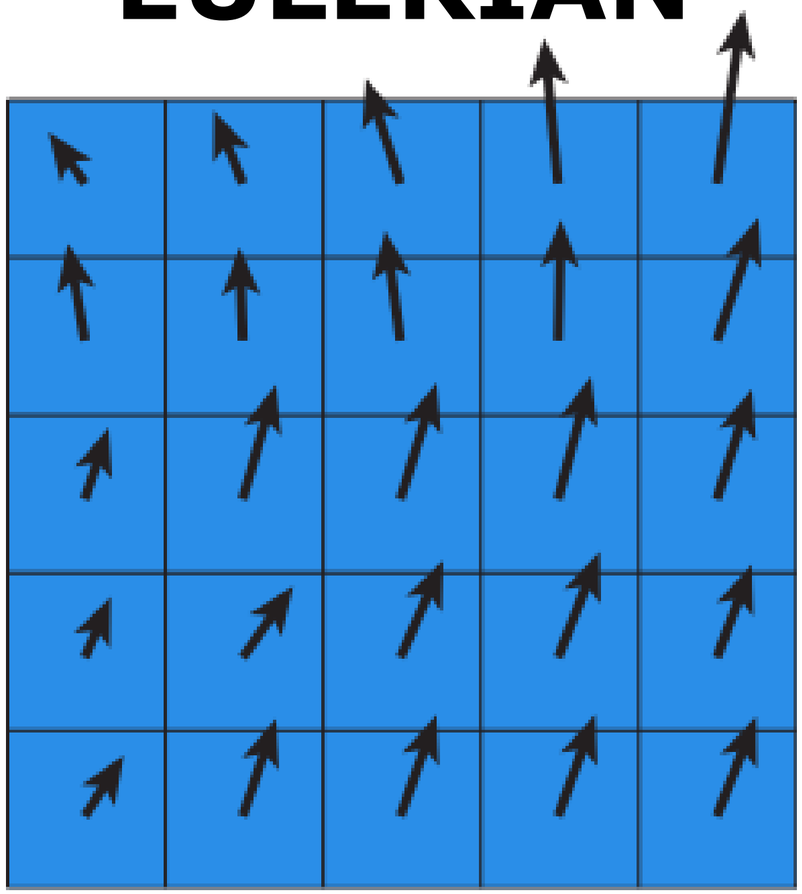
- Two basic ways to discretize space: Lagrangian & Eulerian (particle-based & grid-based)
- Suppose you want to keep track of the weather…

LAGRANGIAN



track moving particles and read what they are measuring

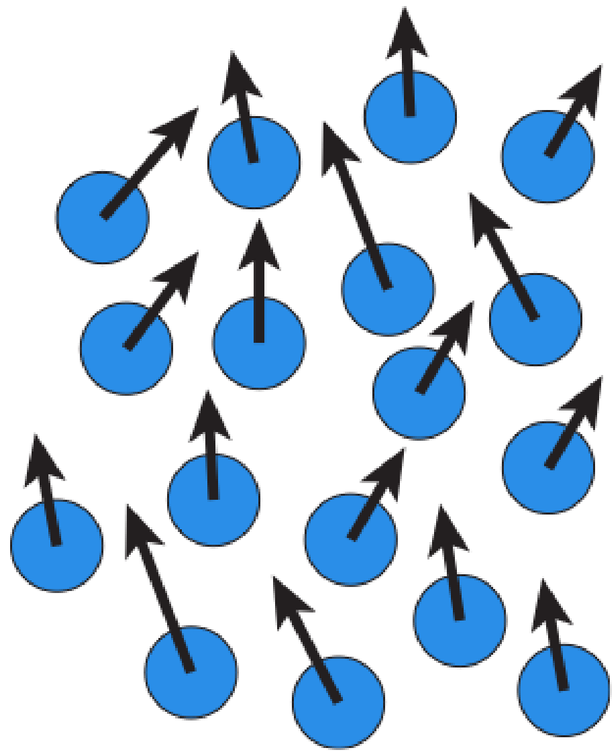
EULERIAN



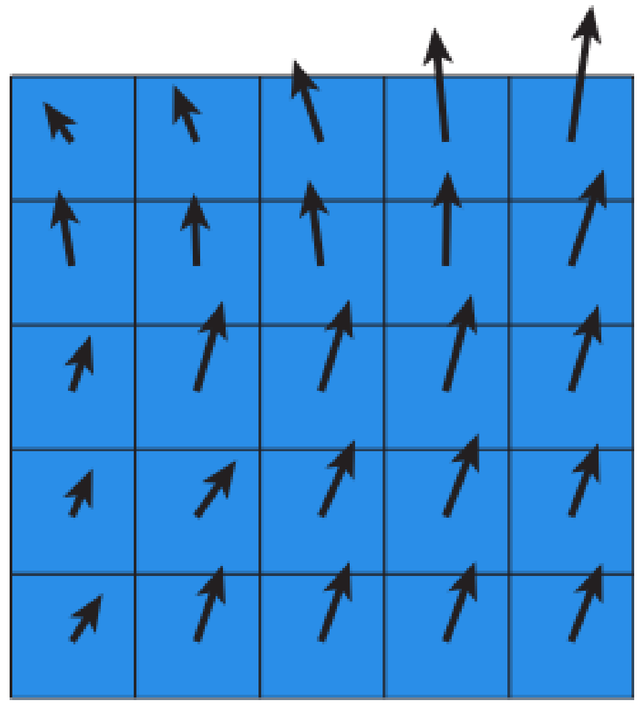
record temperature at fixed locations in space

Lagrangian vs. Eulerian—Trade-Offs

- Lagrangian
  - conceptually easy (like polygon soup!)
  - resolution/domain not limited by grid
  - good particle distribution can be tough
  - finding neighbors can be expensive



- Eulerian
  - fast, regular computation
  - easy to represent, e.g., smooth surfaces
  - simulation “trapped” in grid
  - grid causes “numerical diffusion” (blur)
  - need to understand PDEs



Mixing Lagrangian & Eulerian

- Of course, no reason you have to choose just one!
- Many modern methods mix Lagrangian & Eulerian:
  - PIC/FLIP, particle level sets, mesh-based surface tracking, Voronoi-based, arbitrary Lagrangian-Eulerian (ALE), ...
- *Pick the right tool for the job!*

Maya *Bifrost*

Ok, but we're getting *way* ahead of ourselves.

How do we solve *easy* PDEs?

Numerical PDEs—Basic Strategy

- Pick PDE that models phenomenon of interest
  - Which quantity do we want to solve for?
- Pick spatial discretization
  - How do we approximate derivatives in space?
- Pick time discretization
  - How do we approximate derivatives in time?
  - When do we evaluate forces?
  - Forward Euler, backward Euler, symplectic Euler, ...
- Finally, we have an *update rule*
- Repeatedly solve to generate an animation

The Laplace Operator

- All of our model equations used the Laplace operator
- Different conventions for symbol:

Δ ← same symbol used for "change"

∇^2 ← same symbol used for Hessian!

- Differential operators

Nabla operator

$$\nabla = \left(\frac{\partial}{\partial x_1}, \ldots, \frac{\partial}{\partial x_d} \right)^t \qquad \nabla u = \left(\frac{\partial u}{\partial x_1}, \ldots, \frac{\partial u}{\partial x_d} \right)^t$$

Laplace operator

$$\Delta = \nabla \cdot \nabla = \sum_{k=1}^{d} \frac{\partial^2}{\partial x_k^2} \qquad \Delta u = \nabla \cdot \nabla u$$

div grad

$$\Delta u = \frac{\partial u^2}{\partial x_1^2} + \cdots + \frac{\partial u^2}{\partial x_n^2}$$

The Laplace Operator

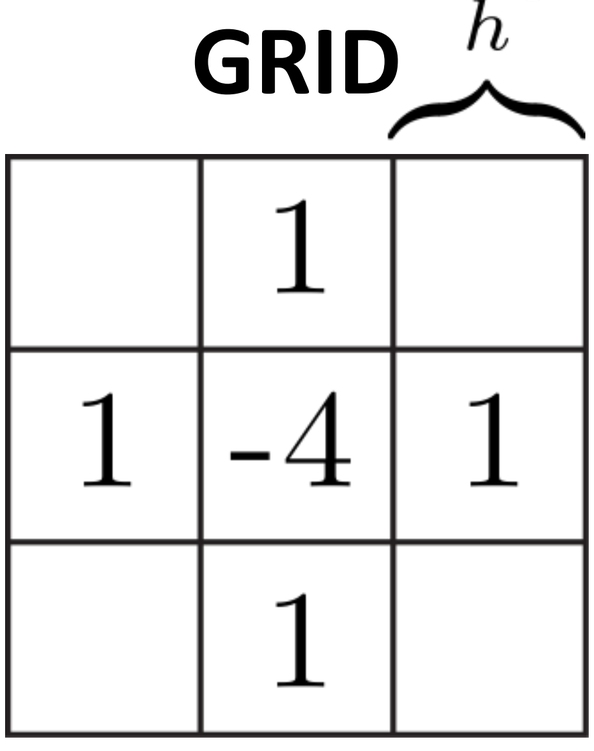
- **All of our model equations used the Laplace operator**
- ***Unbelievably* important object showing up everywhere across physics, geometry, signal processing, ...**
- **Ok, but what does it mean?**
- ***Differential operator:* eats a function, spits out its "2nd derivative"**
- **What does that mean for a function u: $R^n \to R$?**
  - **divergence of gradient**
  - **trace of hessian**
  - **sum of second derivatives**
  - **"average" curvature**

$$\Delta u = \frac{\partial u^2}{\partial x_1^2} + \cdots + \frac{\partial u^2}{\partial x_n^2}$$

Discretizing the Laplacian

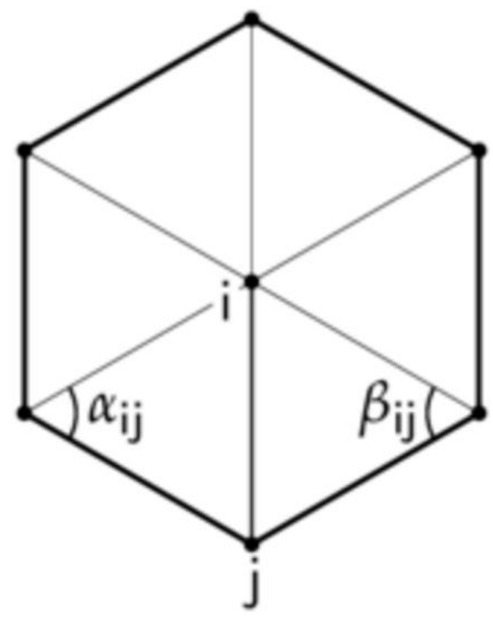
- How do we approximate the Laplacian?
- Depends on discretization (Eulerian, Lagrangian, grid, mesh, …)
- Two extremely common ways in graphics:

GRID



(actually, this becomes that)

TRIANGLE MESH



$$\frac{4u_{ij} - u_{i+1,j} - u_{i-1,j} - u_{i,j+1} - u_{i,j-1}}{h^2}$$

$$\frac{1}{2}\sum_j (\cot\alpha_{ij} + \cot\beta_{ij})(u_j - u_i)$$

- *u* could be displacement in normal direction, for example
- Also not too hard on point clouds, polygon meshes, …

Numerically Solving the Laplace Equation

- Want to solve $\Delta u = 0$
- Plug in one of our discretizations, e.g.,

	c	
d	a	b
	e	

$$\frac{4a - b - c - d - e}{h} = 0$$

$$\iff a = \frac{1}{4}(b + c + d + e)$$

- At solution that solves the Laplace Equation, each value is the average of neighboring values.
- How do we solve this?
- One idea: keep averaging with neighbors! ("Jacobi method")
- Correct, but *slow* convergence

Boundary Conditions for Discrete Laplace

- **What values do we use to compute averages near the boundary?**

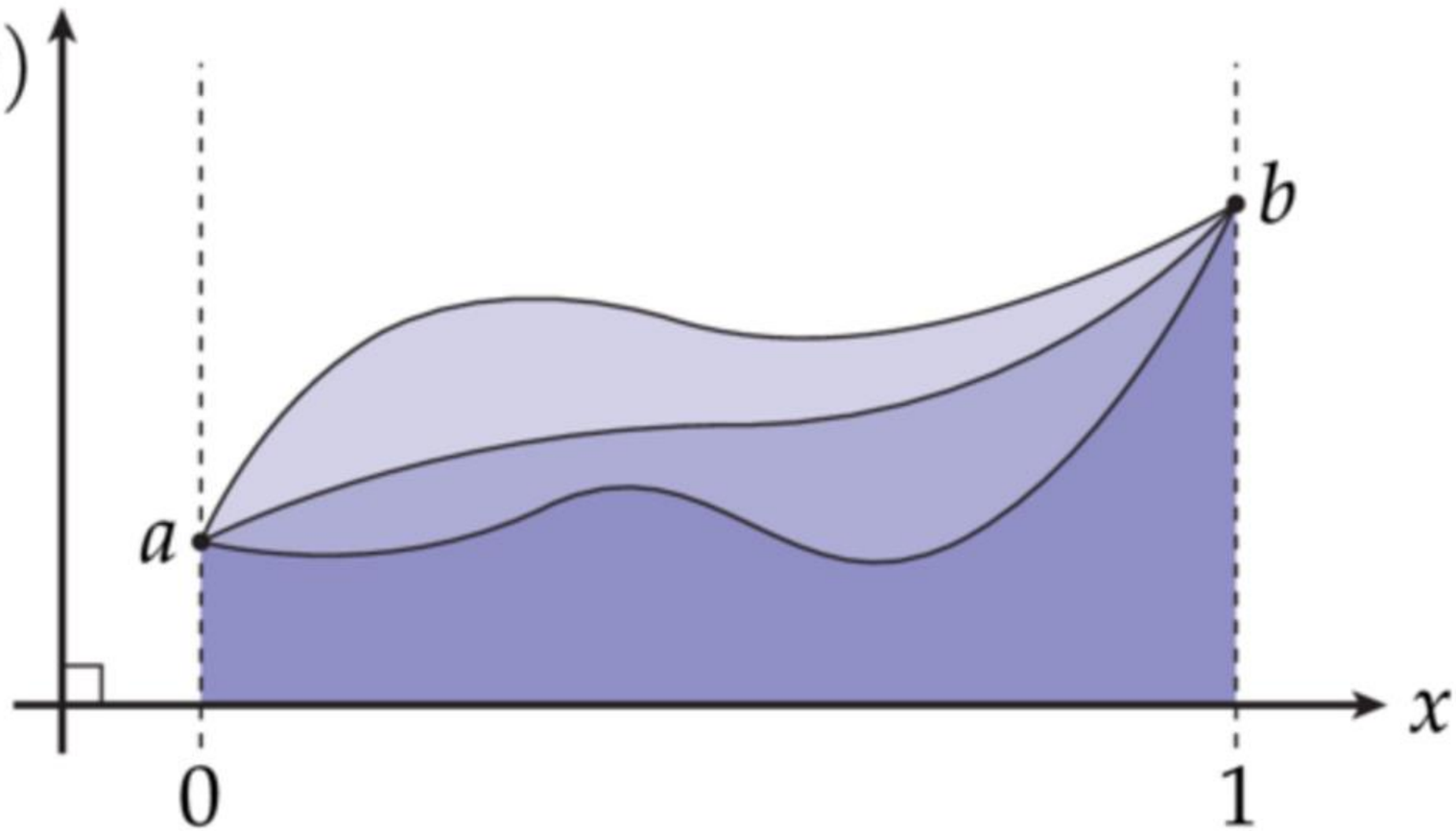
	c	
$?$	a	b
	e	

$$a = \frac{1}{4}(b + c + ? + e)$$

- **A: We get to choose—this is the data we want to interpolate!**
- **Two basic boundary conditions:**
  1. ***Dirichlet*—boundary data always set to fixed values**
  2. ***Neumann*—specify derivative (difference) across boundary**
- **Also mixed (*Robin*) boundary conditions (and more, in general)**

Dirichlet Boundary Conditions

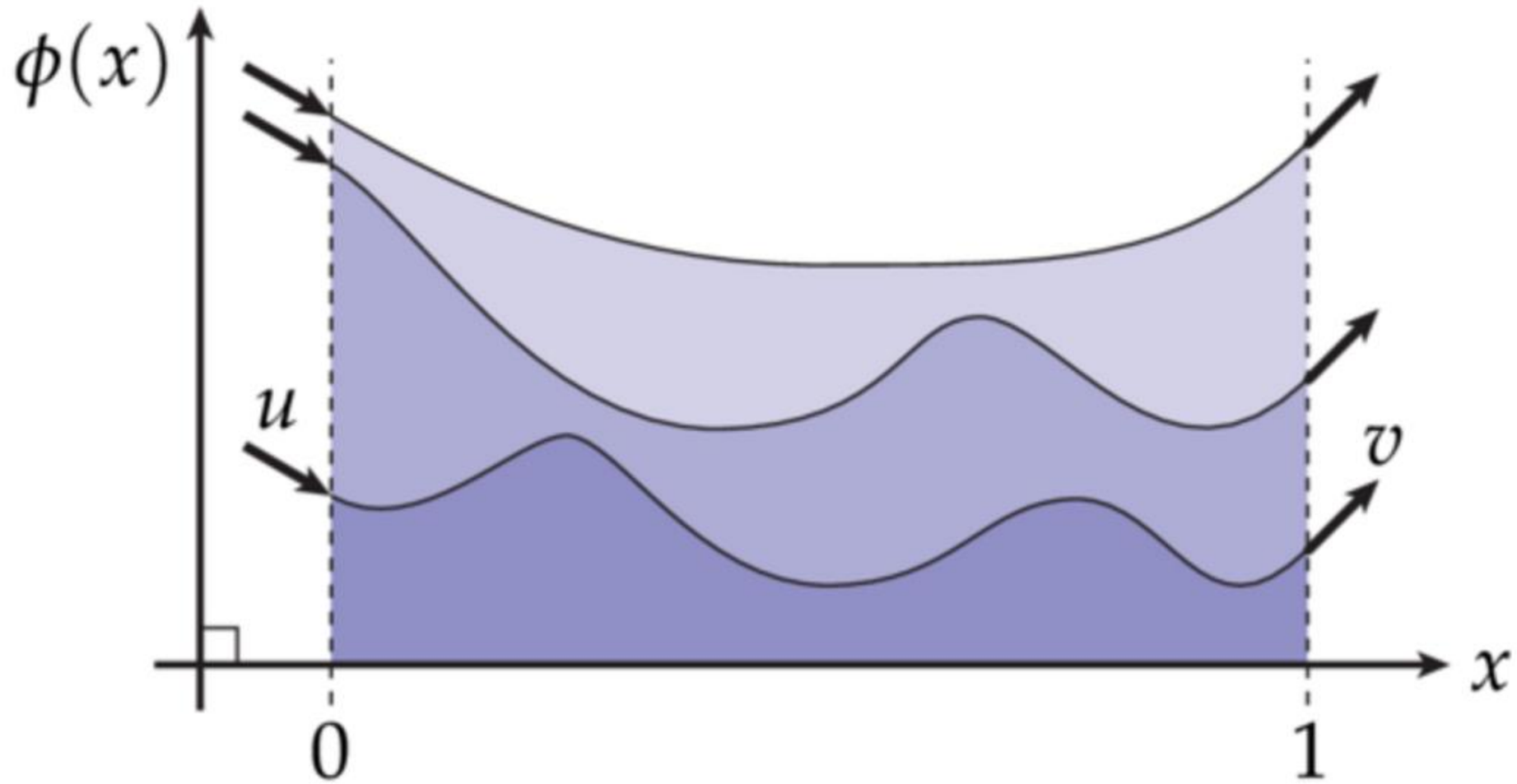
- Let's go back to smooth setting, function on real line
- *Dirichlet* means "prescribe values"
- E.g., Φ(0)=a, Φ(1) = b



- Many possible functions "in between"!

Neumann Boundary Conditions

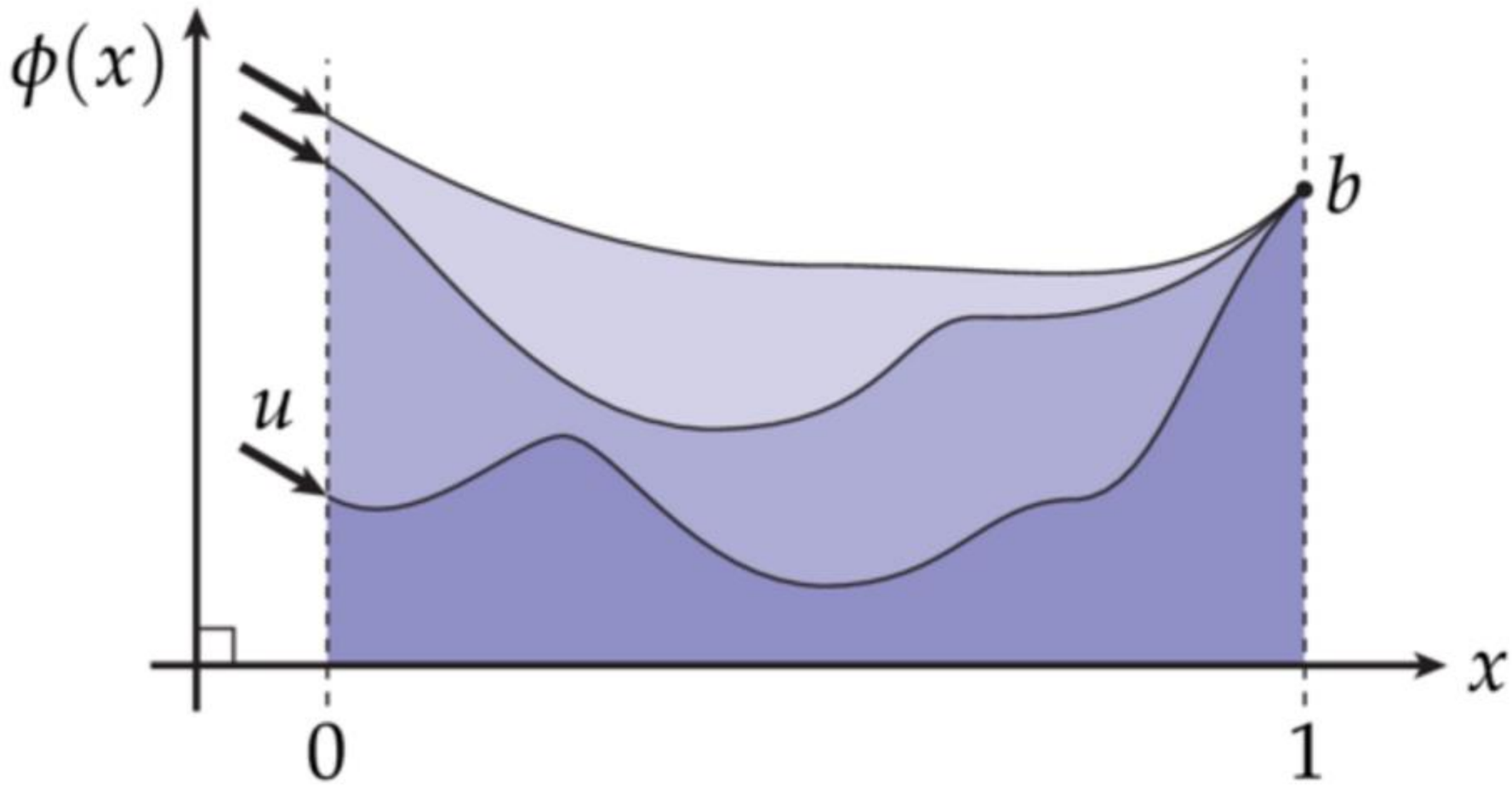
- *Neumann* means “prescribe derivatives”
- E.g., Φ’(0)=u, Φ’(1) = v



- Again, many possible functions!

Both Neumann & Dirichlet

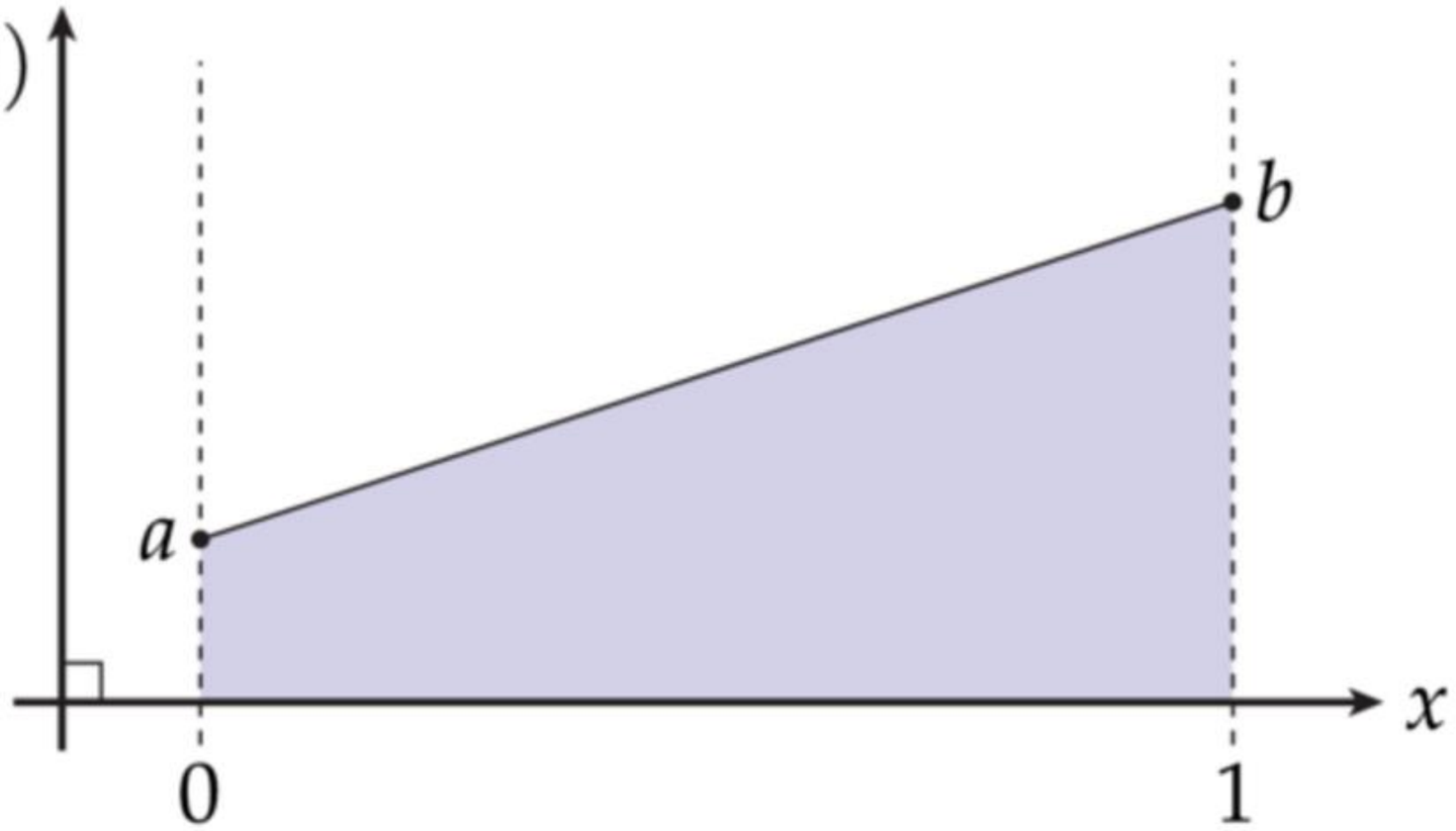
- Or: prescribe some values, some derivatives
- E.g., Φ’(0)=u, Φ(1) = b



- Q: What about Φ’(1)=v, Φ(1) = b? Does that work?
- Q: What about Φ’(0) + Φ(0) = p, Φ’(1) + Φ(1) = q? (*Robin*)

1D Laplace w/ Dirichlet BCs

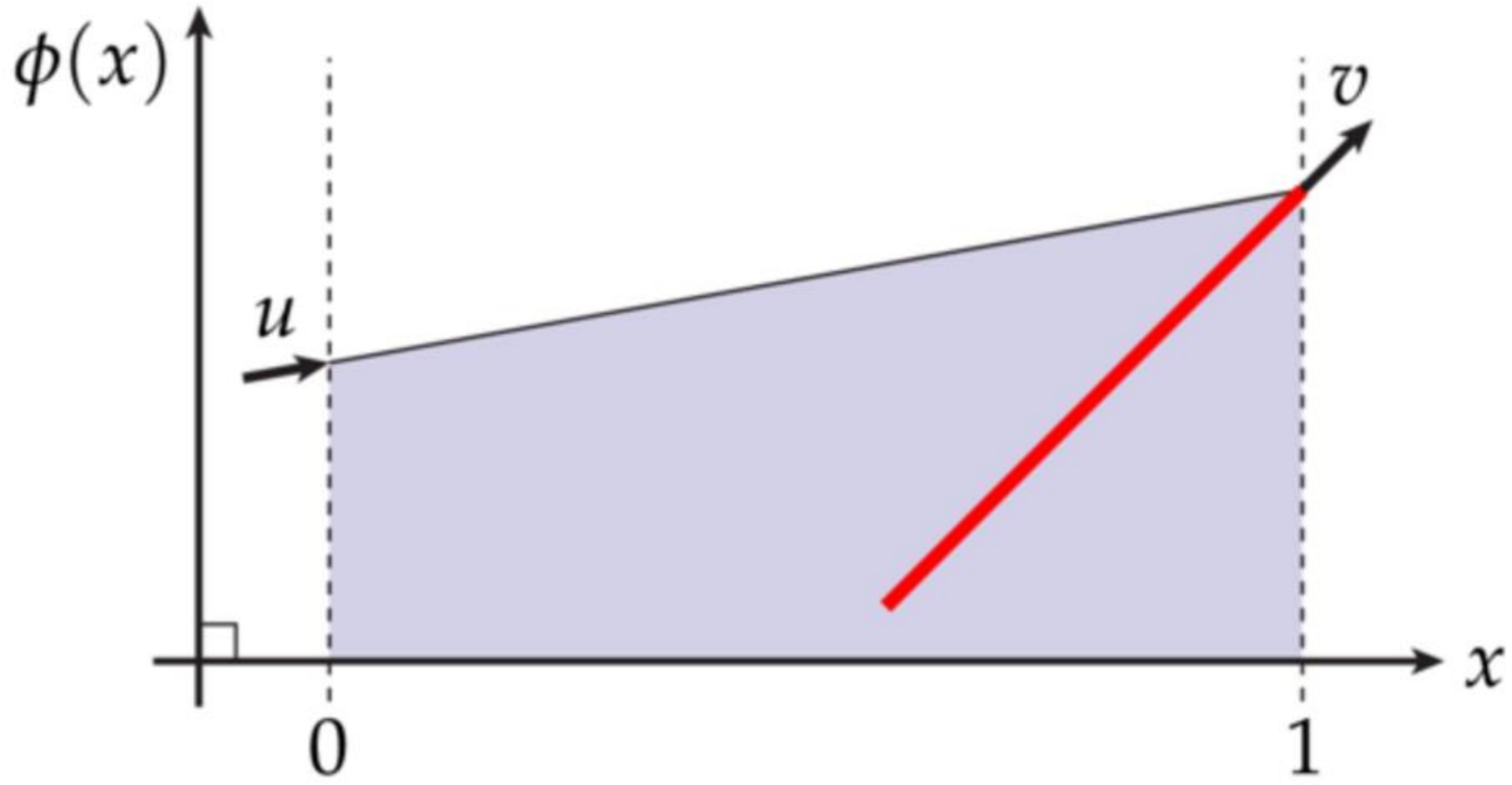
- 1D Laplace: $\partial^2\Phi/\partial x^2 = 0$
- Solutions: $\Phi(x) = cx + d$
- Q: Can we *always* satisfy given Dirichlet boundary conditions?



- Yes: a line can interpolate any two points.

1D Laplace w/ Neumann BCs

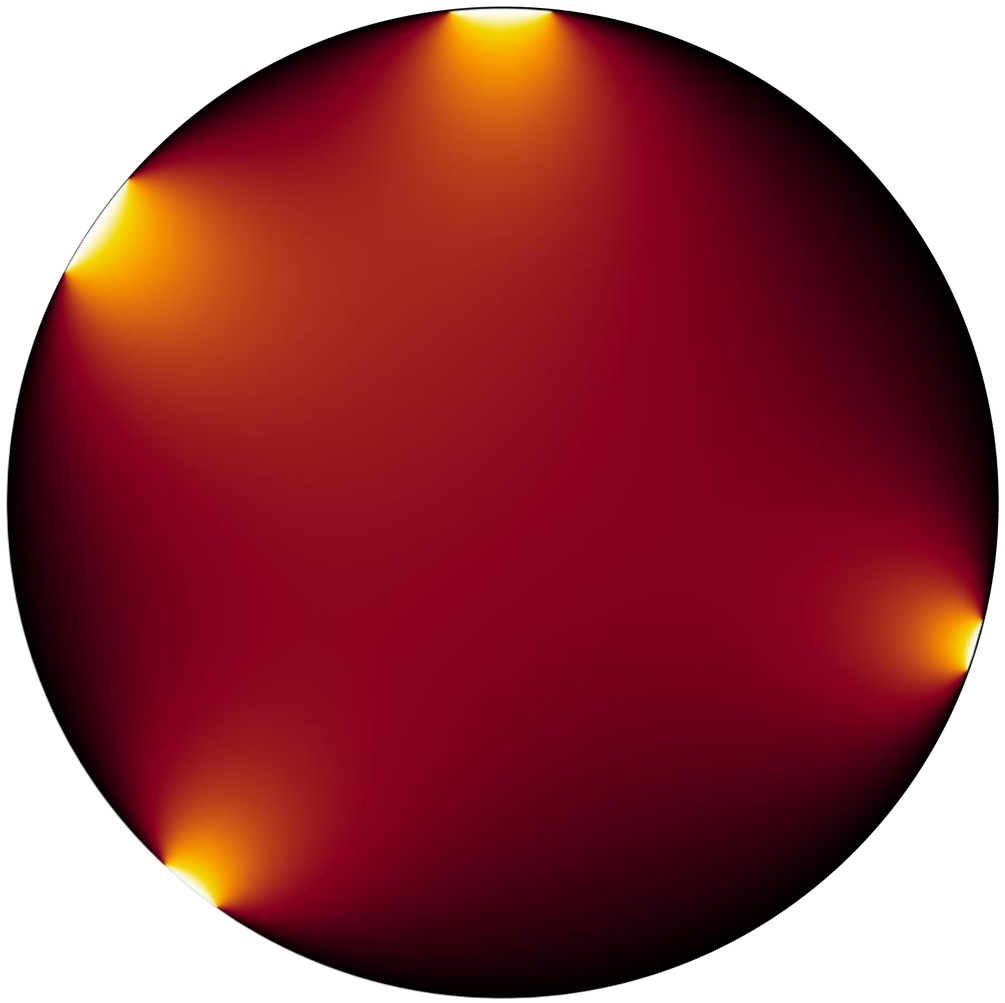
- What about Neumann BCs?
- Q: Can we prescribe the derivative at both ends?



- Only if u and v are equal! A straight line only has one slope.
- In general, solution to a PDE may not exist for given BCs.

2D Laplace w/ Dirichlet BCs

- 2D Laplace: $\Delta \Phi = 0$
- Q: Can satisfy any Dirichlet BCs? (given data along boundary)

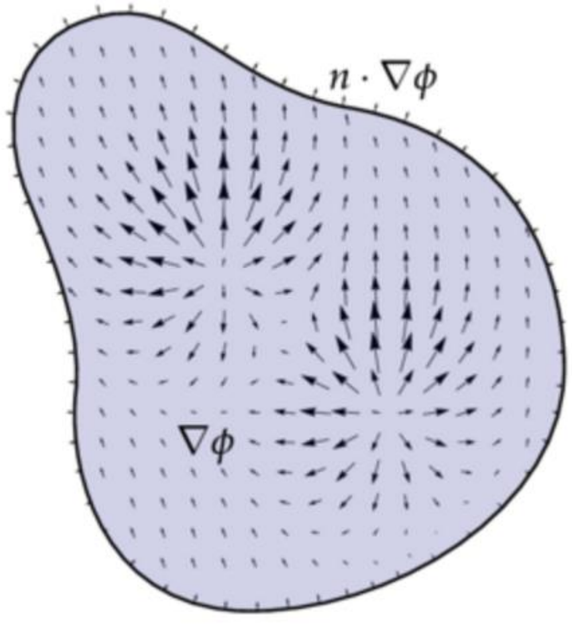


- Yes: Laplace is long-time solution to heat flow
- Data is “heat” at boundary. Then just let it flow...

2D Laplace w/ Neumann BCs

- What about Neumann BCs for $\Delta\Phi=0$?
- Neumann BCs prescribe derivative in normal direction: $n\cdot\nabla\Phi$
- Q: Can it always be done? (Wasn't possible in 1D...)
- In 2D, we have the *divergence theorem:*

$$\int_{\partial\Omega} n\cdot\nabla\phi = \int_{\Omega}\nabla\cdot\nabla\phi = \int_{\Omega}\Delta\phi \stackrel{!}{=} 0$$



- Should be called, "what goes in must come out theorem!"
- Can't have a solution unless the net flux through the boundary is zero.

- Numerical software will *not* always tell you if there's a problem! (Especially if you wrote it yourself...)

Solving the Heat Equation

- **Back to our three model equations, want to solve *heat diffusion equation***

$$\dot{u} = \Delta u$$

- **Just saw how to discretize Laplacian**
- **Also know how to do time (forward Euler, backward Euler, …)**
- **E.g., forward Euler:**

$$u^{k+1} = u^k + \Delta u^k$$

- **Q: On a grid, what's our overall update now at $u_{i,j}$?**

$$u_{i,j}^{k+1} = u^k + \frac{\tau}{h^2}(4u_{i,j}^k - u_{i+1,j}^k - u_{i-1,j}^k - u_{i,j+1}^k - u_{i,j-1}^k)$$

- ***Not* hard to implement! Loop over grid, add up some neighbors.**

Solving the Wave Equation

- Finally, wave equation:

$$\ddot{u} = \Delta u$$

- Not much different; now have 2nd derivative in time
- By now we've learned two different techniques:
  - Convert to two 1st order (in time) equations:

$$\dot{u} = v, \quad \dot{v} = \Delta u$$

  - Or, use centered difference (like Laplace) in time:

$$\frac{u^{k+1} - 2u^k + u^{k-1}}{\tau^2} = \Delta u^k$$

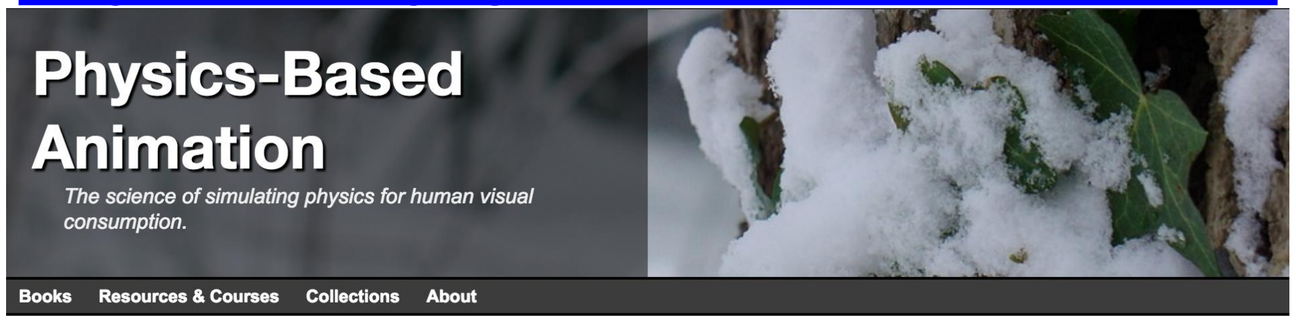
- Plus all our choices about how to discretize Laplacian.
- *So many choices!* And many, many *(many)* more we didn't discuss.

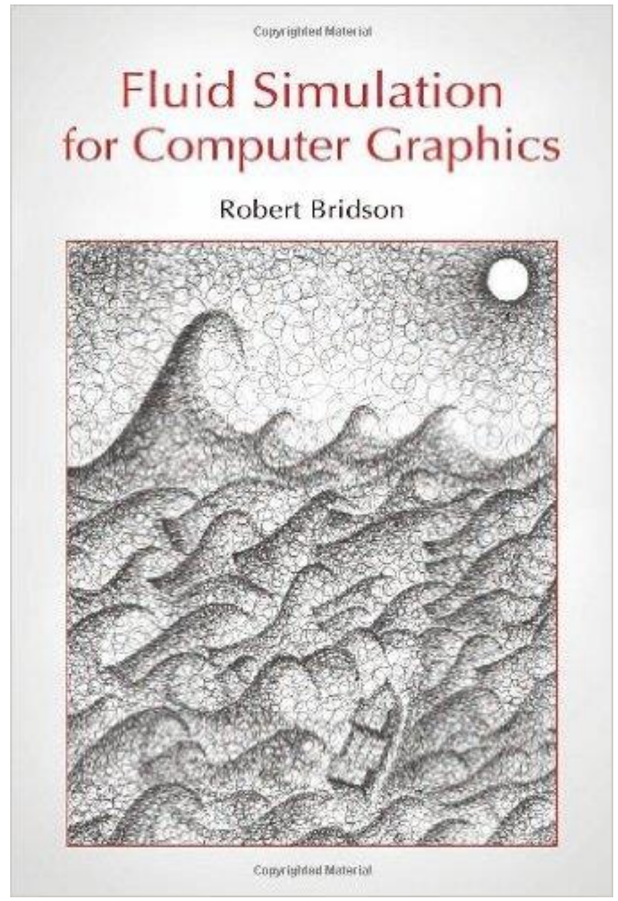
Wait, what about all those cool fluids and stuff?

Want to Know More?

- There are some good books:
- And papers:

http://www.physicsbasedanimation.com/





- Also, what did the folks who *wrote* these books & papers read?