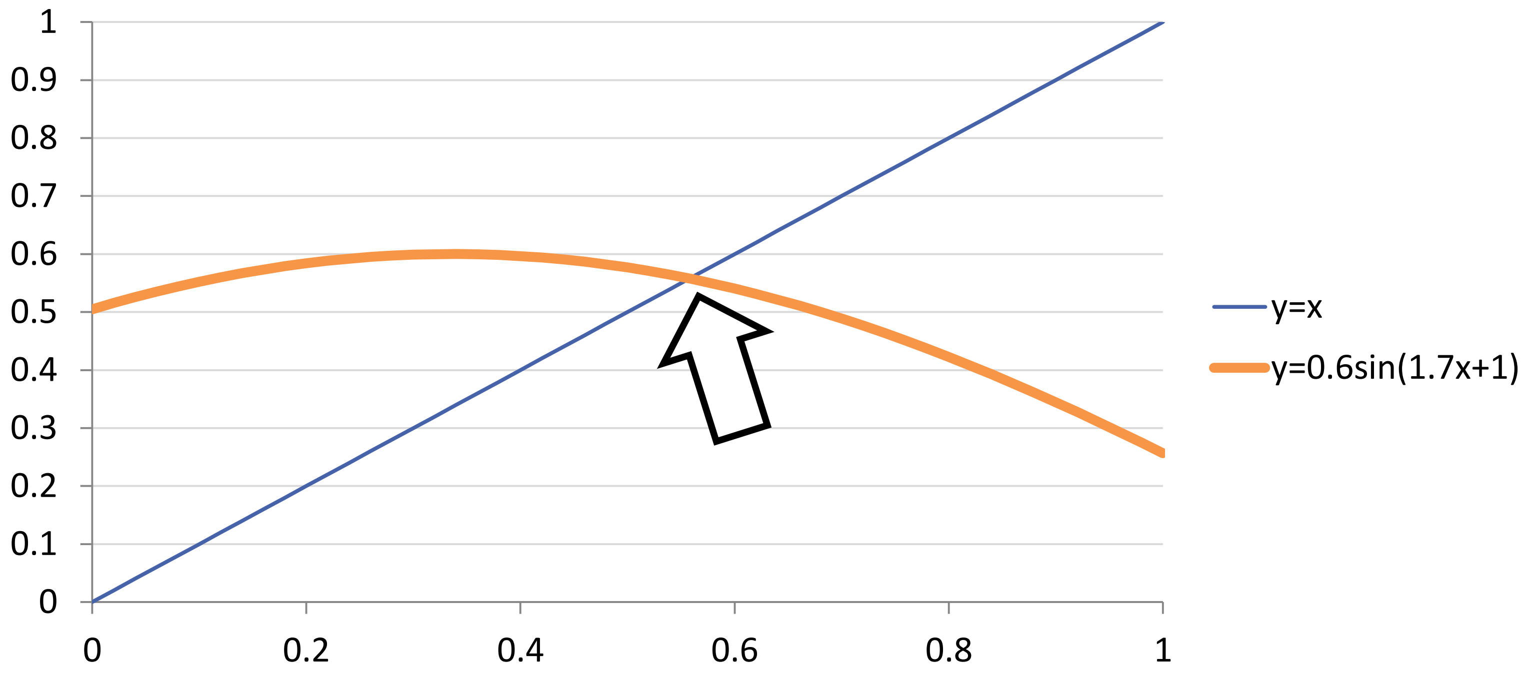
Fixed Point Iteration

a.k.a. SCIENCE!!!

Huw Bowles
Technical Lead
Studio Gobo

Pop Quiz



Pop Quiz

- Set equations equal
  - $x = 0.6\sin(1.7x + 1)$

Pop Quiz

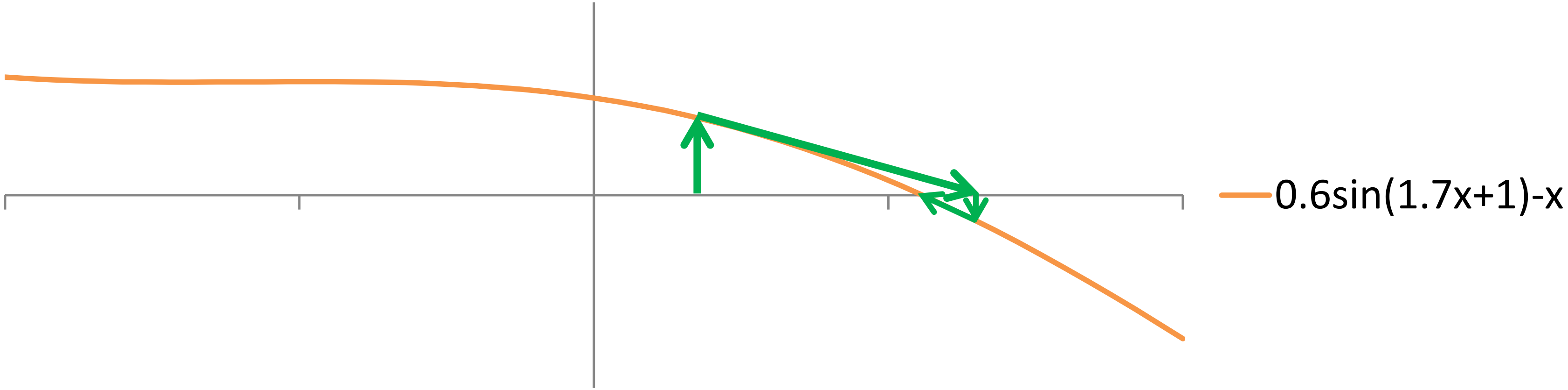
- Set equations equal
  - $\boldsymbol{x} = 0.6\sin(1.7\boldsymbol{x} + 1)$
- Solve for $\boldsymbol{x}$

Pop Quiz

- Set equations equal
  - $x = 0.6\sin(1.7x + 1)$
- Solve for x
- No algebraic solution!

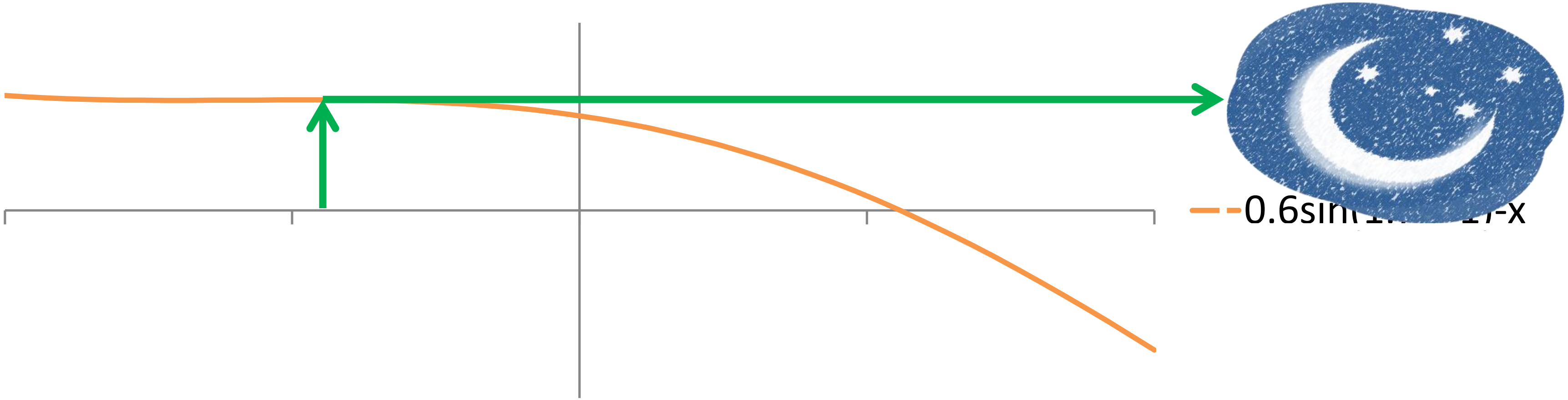
Numerical Methods

- Newton's method to the rescue



Numerical Methods

- Newton's method to the rescue



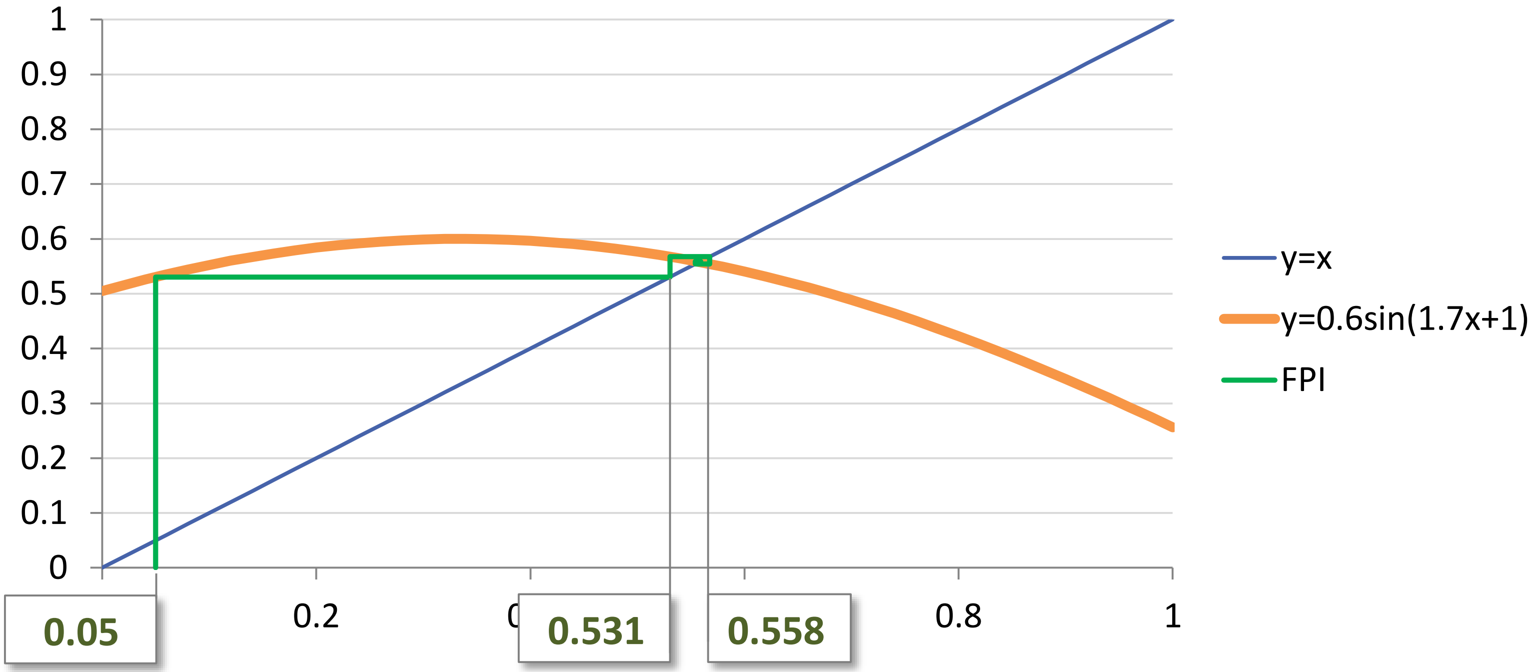
Fixed Point Iteration (FPI)

- Given equation of form:
  - $\boldsymbol{x} = F(\boldsymbol{x})$
- And initial guess $\boldsymbol{x_0}$
- Compute iterates $\boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3} \ldots$
  - $\boldsymbol{x_{i+1}} = F(\boldsymbol{x_i})$

Fixed Point Iteration (FPI)

- Equation
  - $x = 0.6\sin(1.7x + 1)$
- Let $F(x) = 0.6\sin(1.7x + 1)$
- Initial guess $x_0 = 0.05$
- Iterate:
  - $x_1 = F(x_0) = F(0.05) \cong 0.531$
  - $x_2 = F(x_1) = F(0.531) \cong 0.567$
  - …
  - $x_5 = F(x_4) = F(0.559) \cong 0.558$

Fixed Point Iteration (FPI)



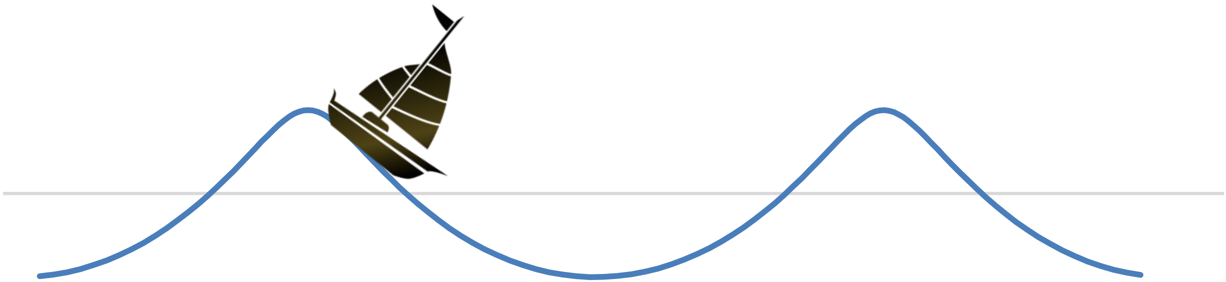
en.wikipedia.org/wiki/Cobweb_plot

Talk Outline

- Real world applications of FPI
- Convergence requirements
- Power-ups

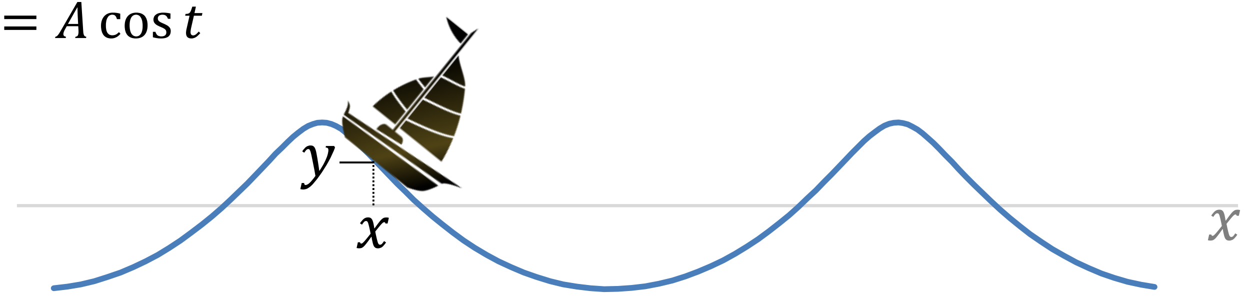
Case 1: Gerstner waves

- Deep water ocean wave shape [6]
  - Used in Uncharted 3, Disney Infinity: Pirates of the Caribbean [1,2]



Case 1: Gerstner waves

- Problem: Given x, compute y
- Parametric curve
  - $x = t - B \sin t$
  - $y = A \cos t$



Case 1: Gerstner waves

- Problem: Given x, compute y
- Parametric curve
  - $x = t - B \sin t$
  - $y = A \cos t$
- y is a function of t

Case 1: Gerstner waves

- Problem: Given x, compute y
- Parametric curve
  - $x = \boldsymbol{t} - B \sin \boldsymbol{t}$
  - $y = A \cos \boldsymbol{t}$
- y is a function of $\boldsymbol{t}$
- Need $\boldsymbol{t}$ from first equation
  - Same equation form as pop quiz!

Case 1: Gerstner waves

- Rearrange to get $F(t) = t$ form
  - $x = \boldsymbol{t} - B \sin \boldsymbol{t}$
  - $\boldsymbol{t} = x + B \sin \boldsymbol{t}$
  - Let $F(\boldsymbol{t}) = x + B \sin \boldsymbol{t}$
- Initial guess: $\boldsymbol{t_0} = x$
- Iterate: $\boldsymbol{t_1} = F(\boldsymbol{t_0})$, …
- Compute y using $\boldsymbol{t_n}$

Case 1: Gerstner waves

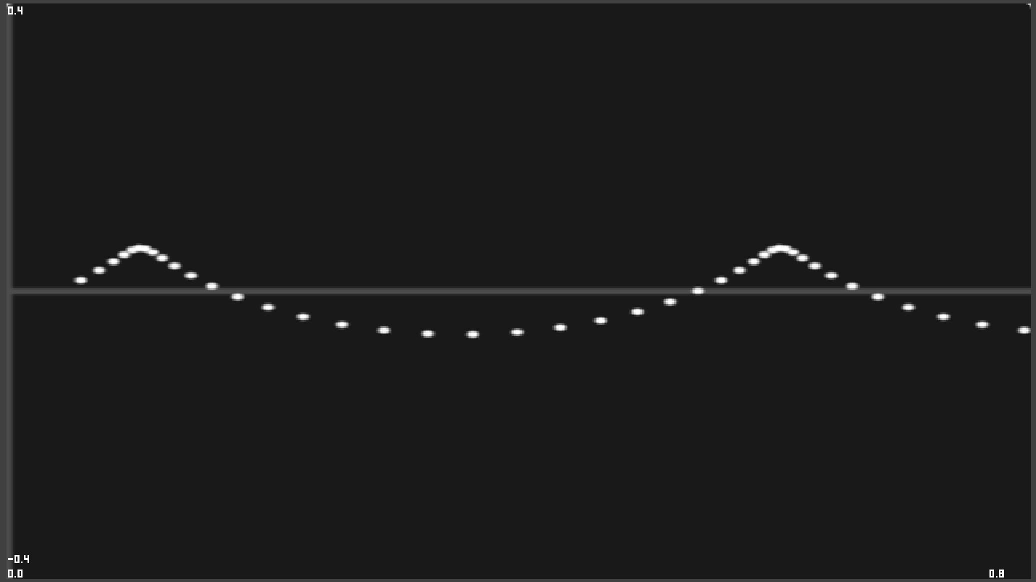


Diagram: shadertoy.com/view/MstSRN

Case 1: Gerstner waves

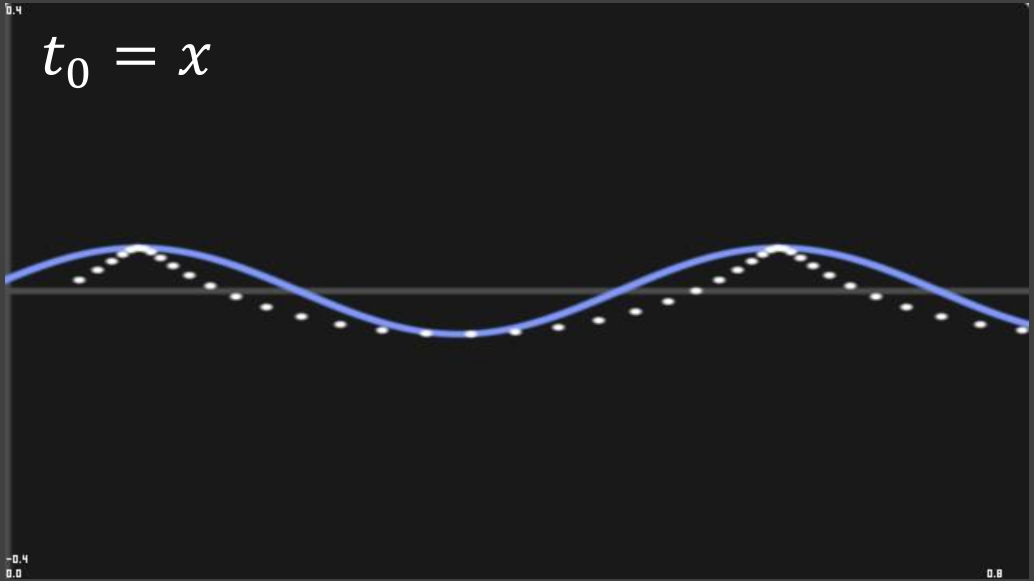


Diagram: shadertoy.com/view/MstSRN

Case 1: Gerstner waves

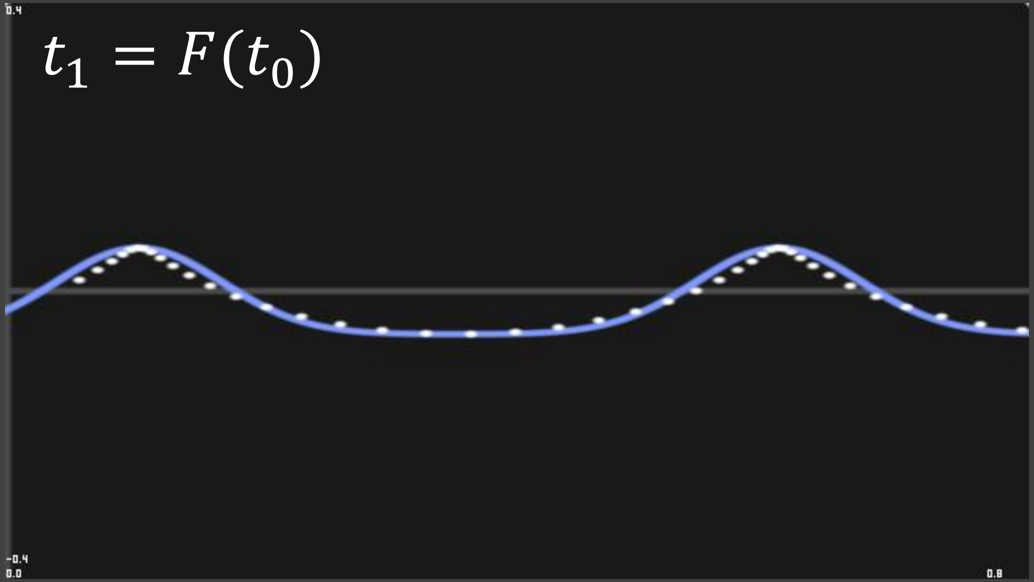


Diagram: shadertoy.com/view/MstSRN

Case 1: Gerstner waves

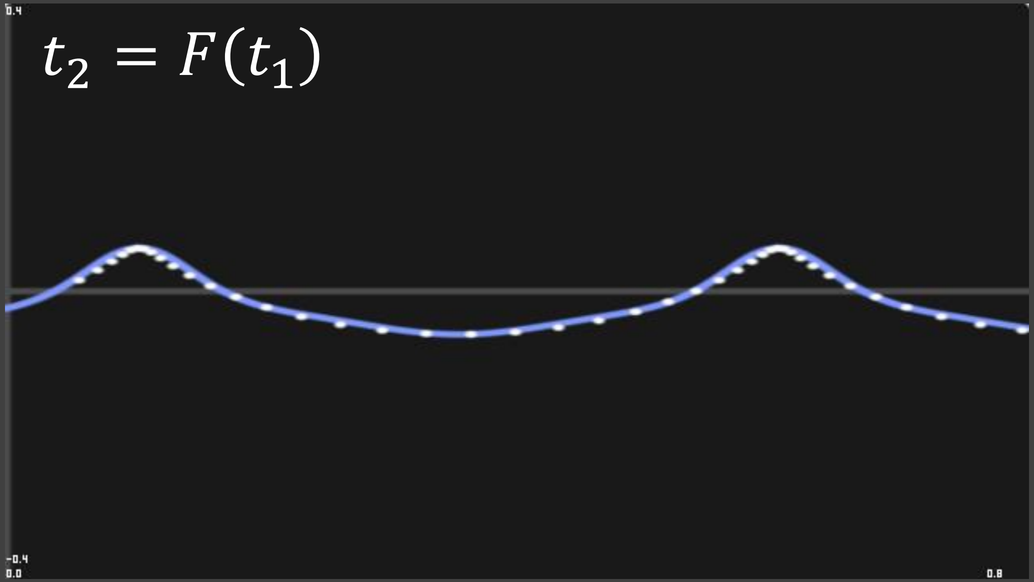


Diagram: shadertoy.com/view/MstSRN

Case 1: Gerstner waves

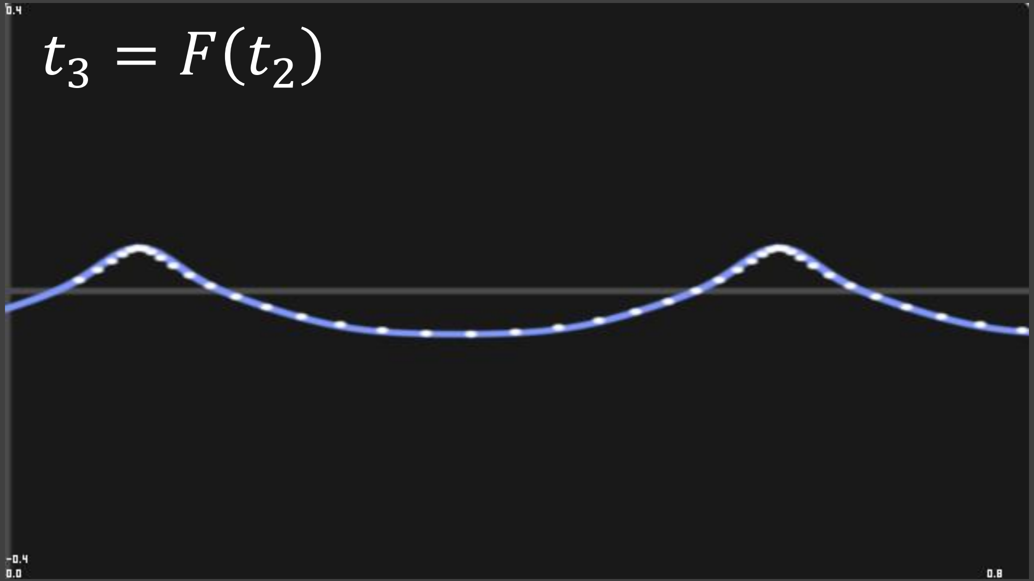


Diagram: shadertoy.com/view/MstSRN

Case 1: Gerstner waves

$$t_4 = F(t_3)$$

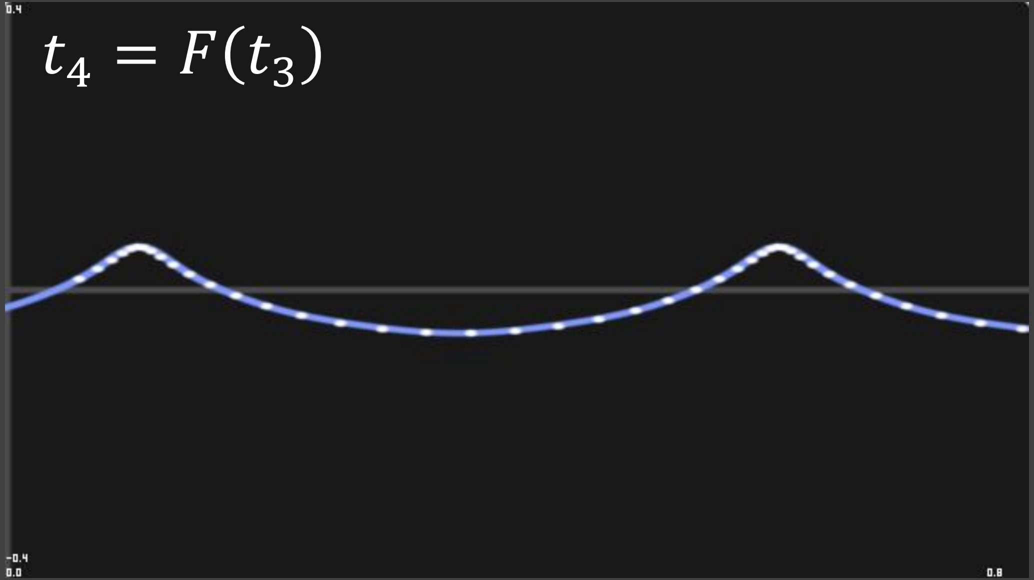


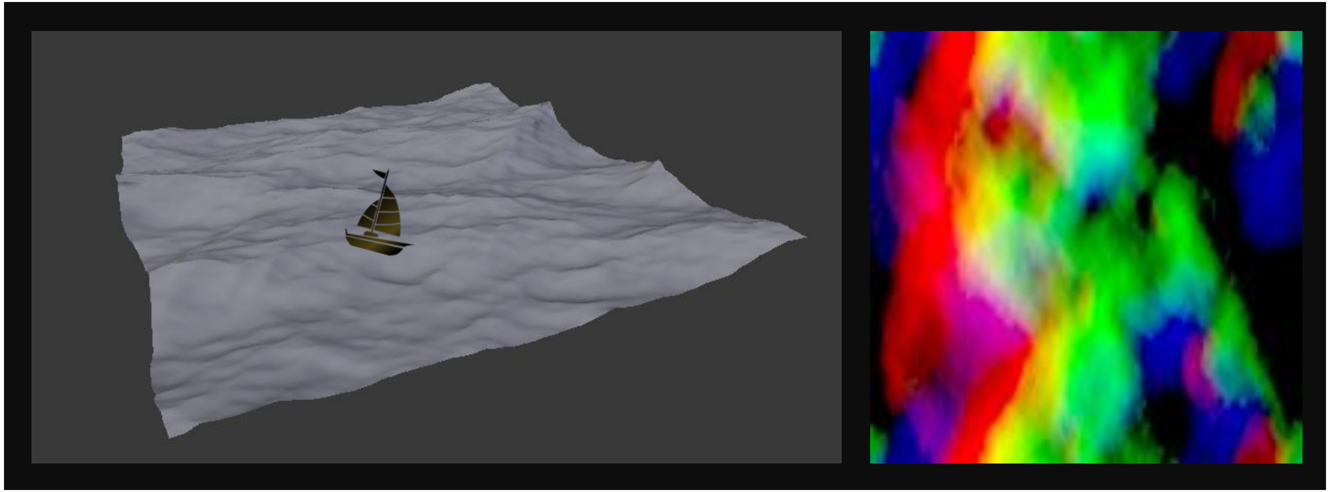
Diagram: shadertoy.com/view/MstSRN

Case 1 Summary

- FPI can solve hard problems
- Extremely general
  - $\boldsymbol{t} = F(\boldsymbol{t})$
  - More octaves? Sure!
- Stable – no gradient

Case 2: Vector Displacement Maps

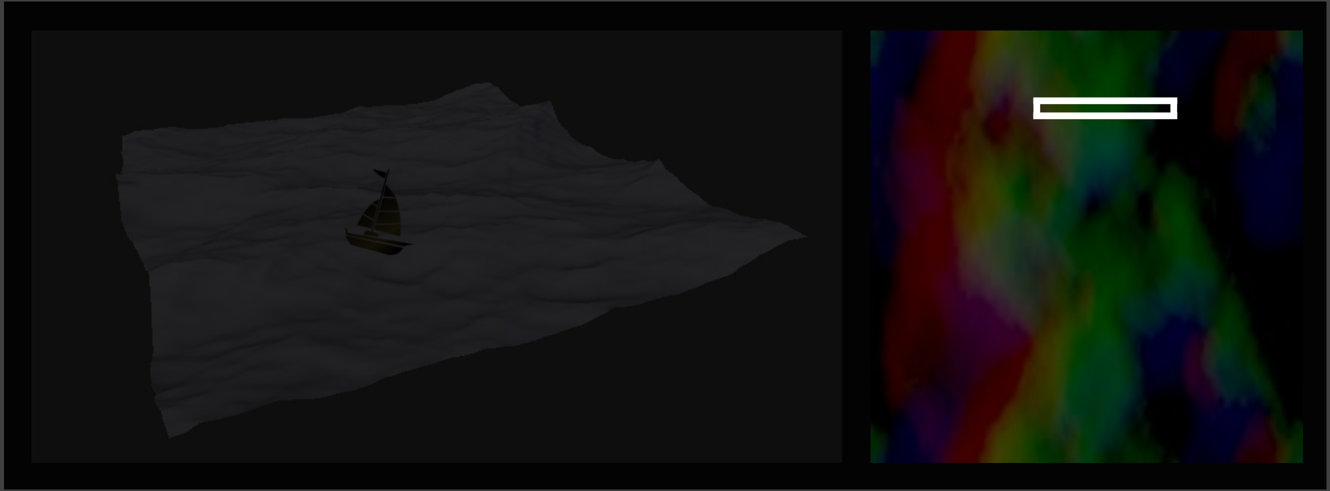
- Simulate the ocean offline instead [1,4]



[Source: Blender]

Case 2: Vector Displacement Maps

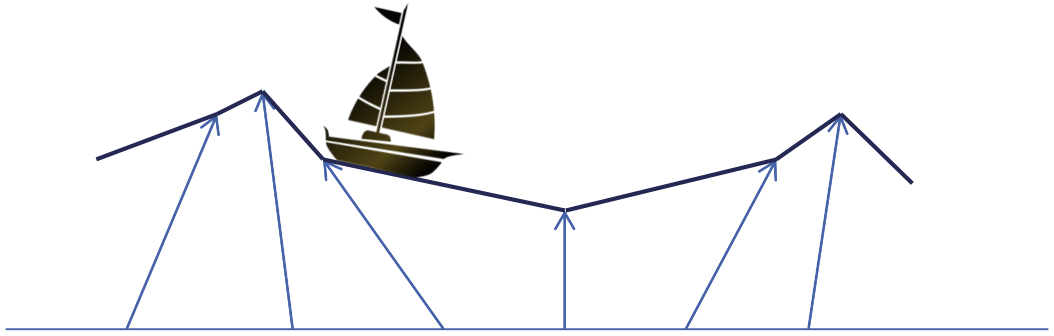
- Simulate the ocean offline instead



[Source: Blender]

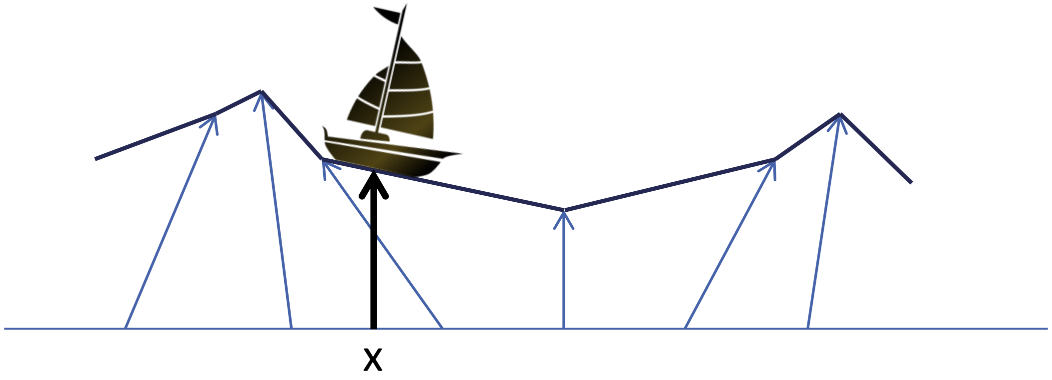
Case 2: Vector Displacement Maps

Case 2: Vector Displacement Maps



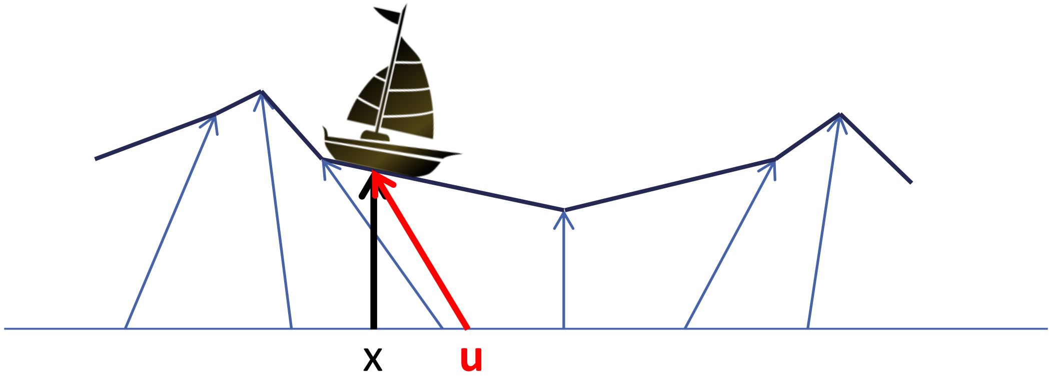
Case 2: Vector Displacement Maps

- Goal: compute height at x



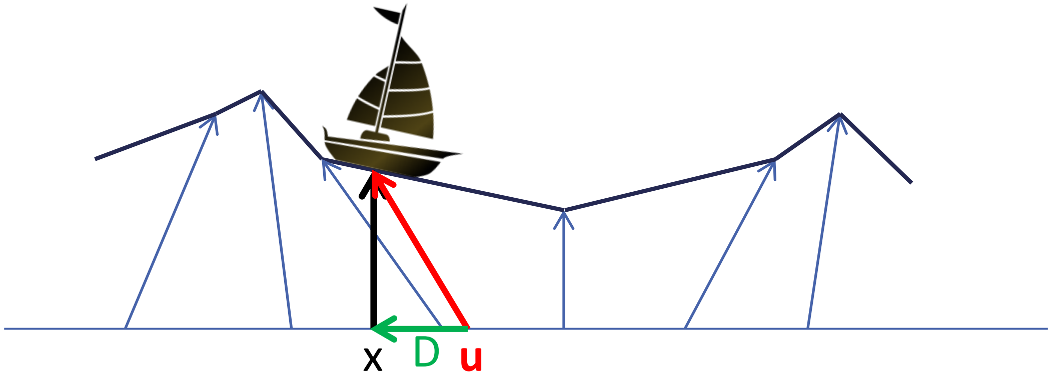
Case 2: Vector Displacement Maps

- Goal: compute height at x
- Height given at $\boldsymbol{u}$



Case 2: Vector Displacement Maps

- Need to find $\boldsymbol{u}$ which will be displaced to x
- Equation: $\boldsymbol{u} + D(\boldsymbol{u}) = x$

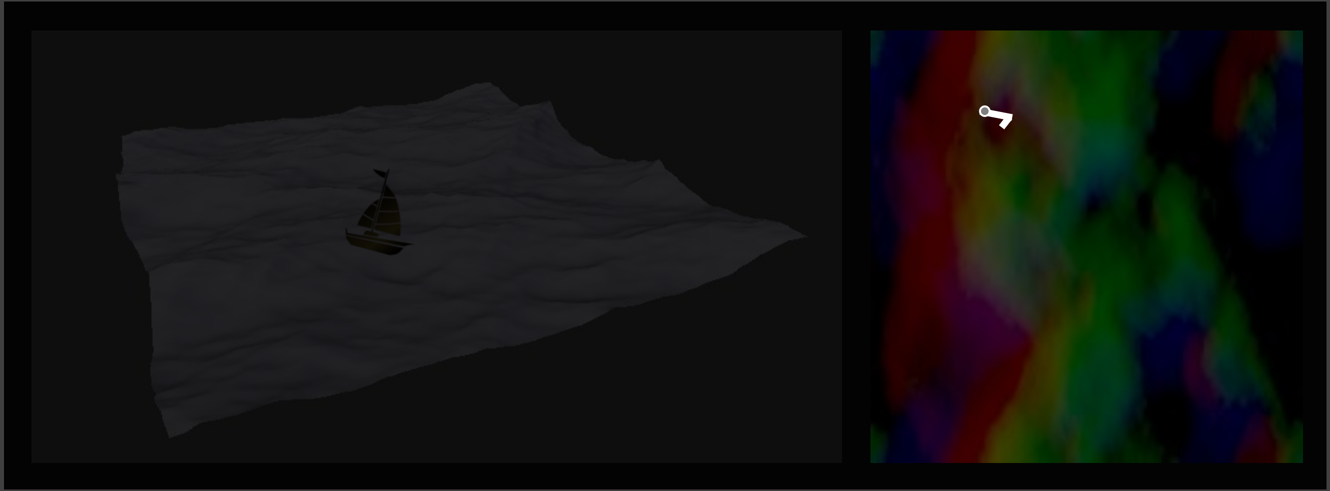


Case 2: Vector Displacement Maps

- FPI to the rescue!
- Equation: $\boldsymbol{u} + D(\boldsymbol{u}) = x$
- Rearrange: $\boldsymbol{u} = x - D(\boldsymbol{u})$
  - Let $\mathrm{F}(\boldsymbol{u}) = x - D(\boldsymbol{u})$
- Guess $\boldsymbol{u_0} = x$
- Iterate: $\boldsymbol{u_1} = x - D(\boldsymbol{u_0})$, …

Case 2: Vector Displacement Maps

- Simulate the ocean offline instead



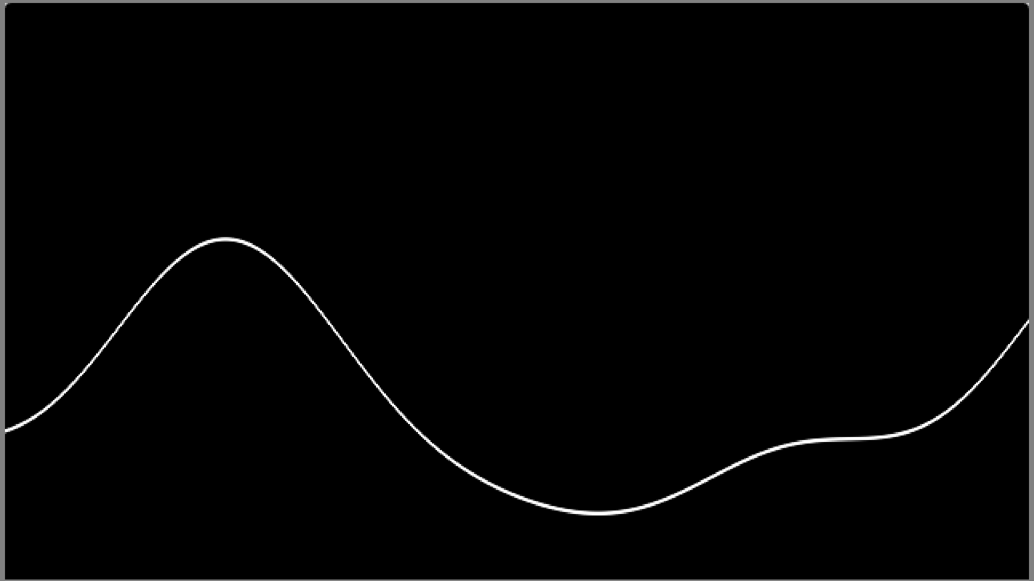
[Source: Blender]

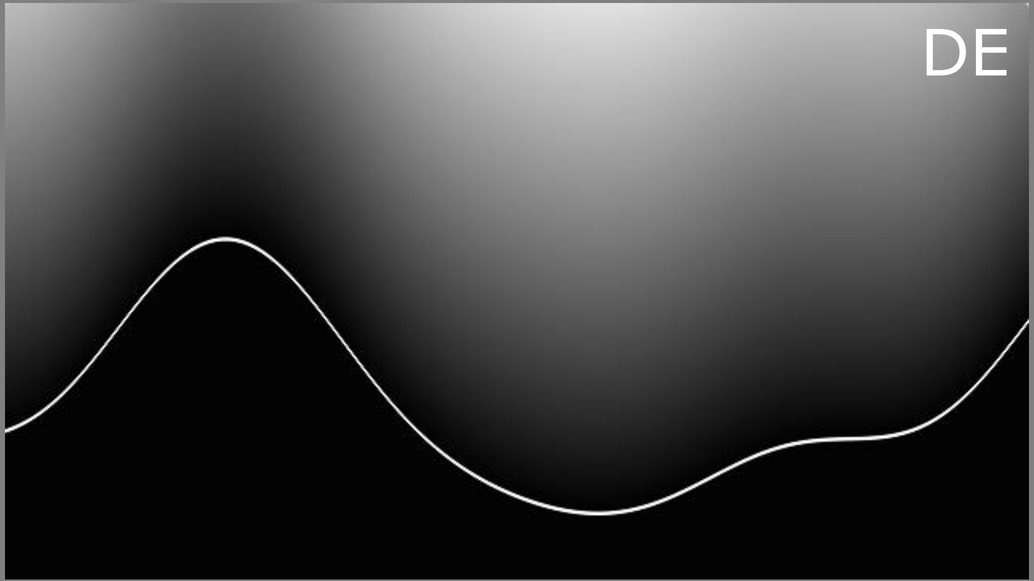
Case 2 Lessons

- FPI great for searching data
  - Simple to implement
  - Under-iterated result stable
- Generalises to higher dimensions

Case 3: Distance field terrain

- Render terrain represented by Distance Estimator (DE)



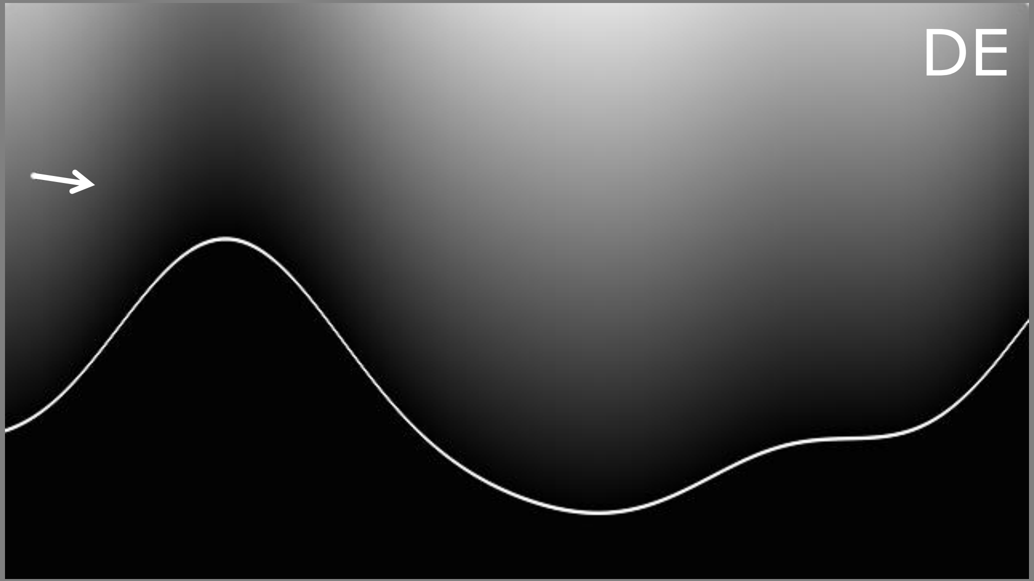

DE

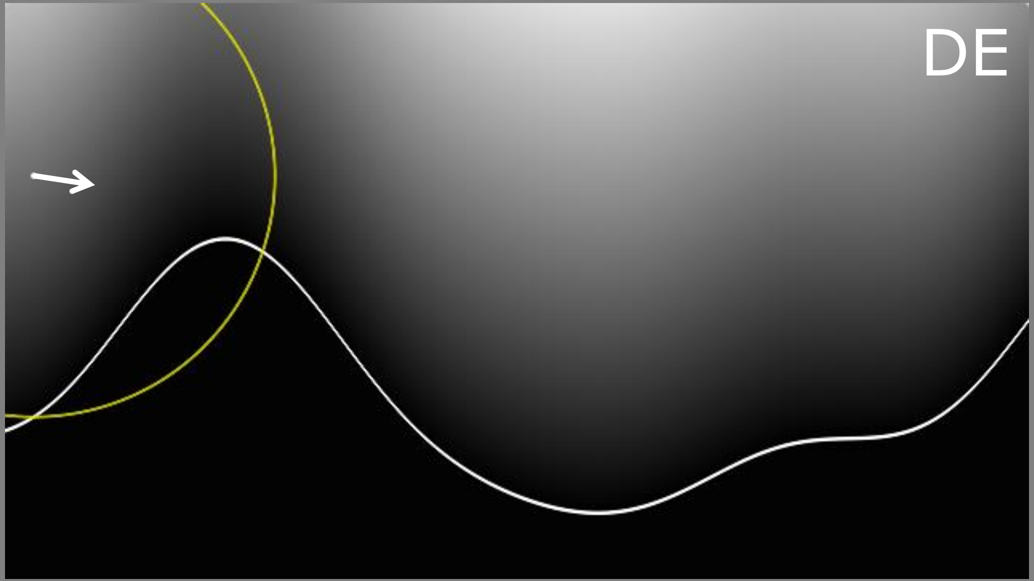
Case 3: Distance field terrain

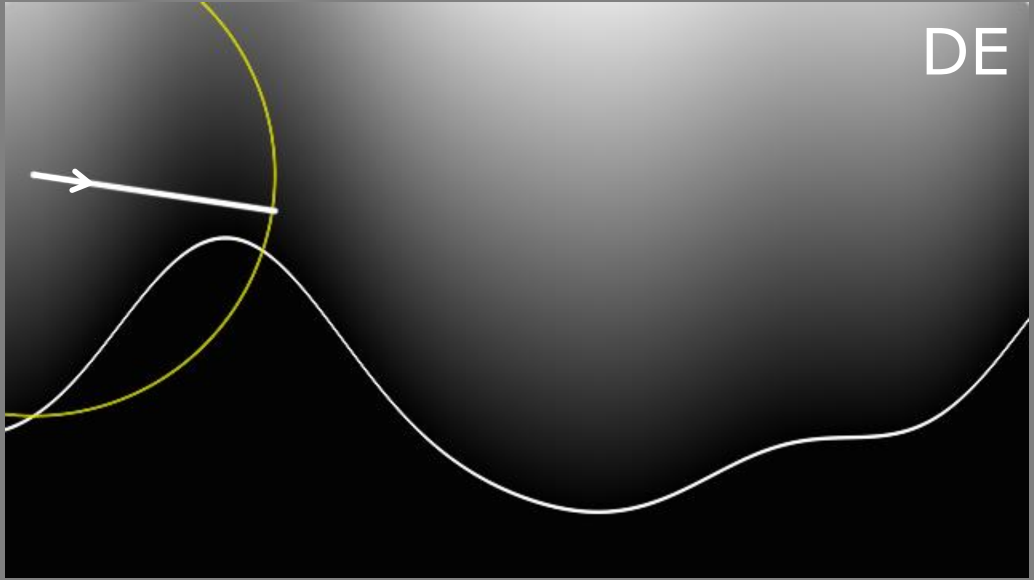
- Pixel ray
  - $\boldsymbol{p} = \boldsymbol{o} + t\boldsymbol{d}$
- Want intersection of ray with surface
  - $DE(\boldsymbol{x}) = 0$
- Plug in ray:
  - $DE(\boldsymbol{o} + t\boldsymbol{d}) = 0$
- Solve for t!
  - But this is not our usual form $t = F(t)$

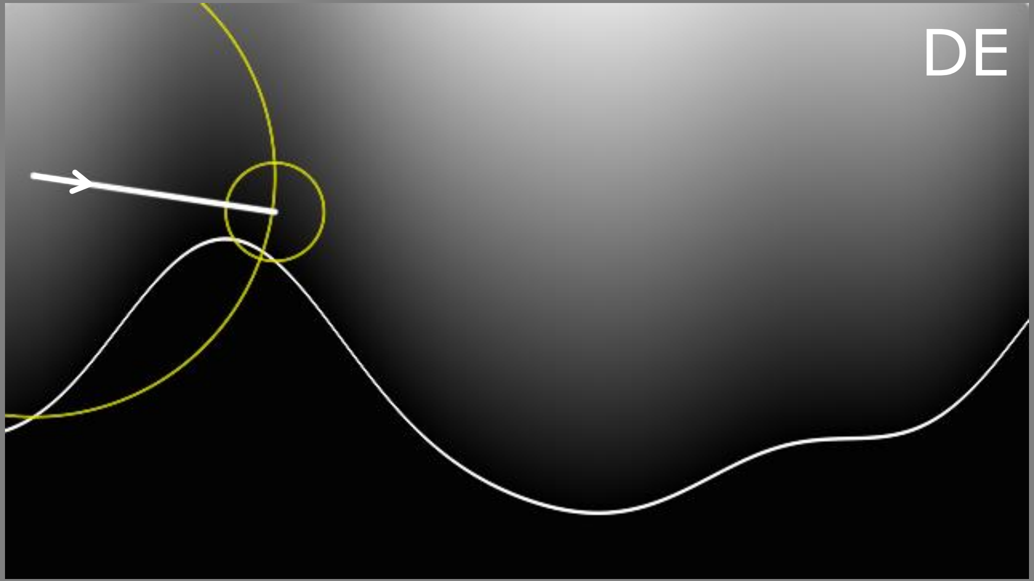
Case 3: Distance field terrain

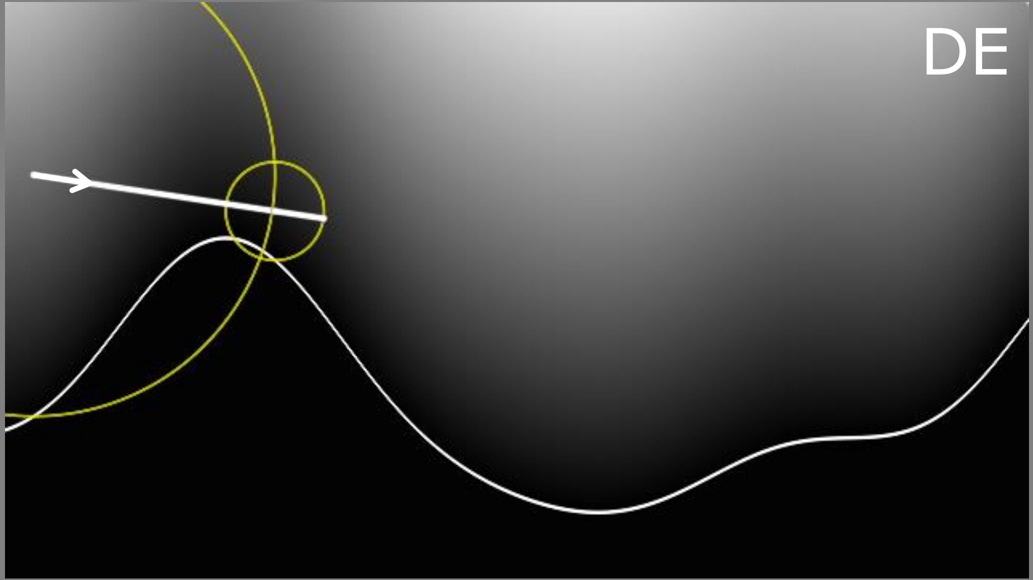
- Easy fix: add t to both sides
  - $0 = DE(\boldsymbol{o} + t\boldsymbol{d})$
  - $t = t + DE(\boldsymbol{o} + t\boldsymbol{d})$
- Let $F(t) = t + DE(\boldsymbol{o} + t\boldsymbol{d})$
- Start at camera: guess $t_0 = 0$
- Iterate: $t_{i+1} = t_i + DE(\boldsymbol{o} + t_i\boldsymbol{d})$
  - Standard raymarch algorithm!

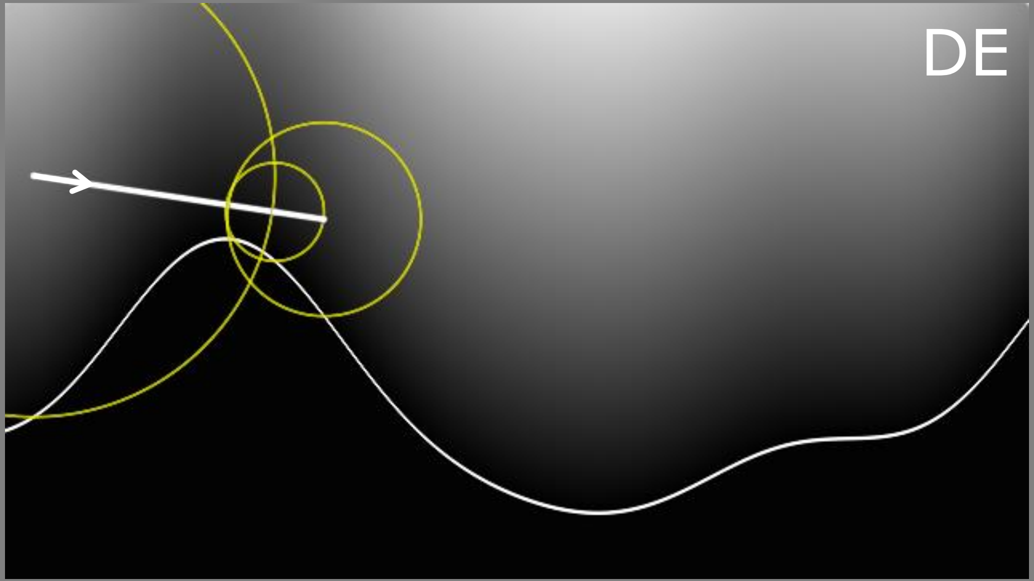

DE

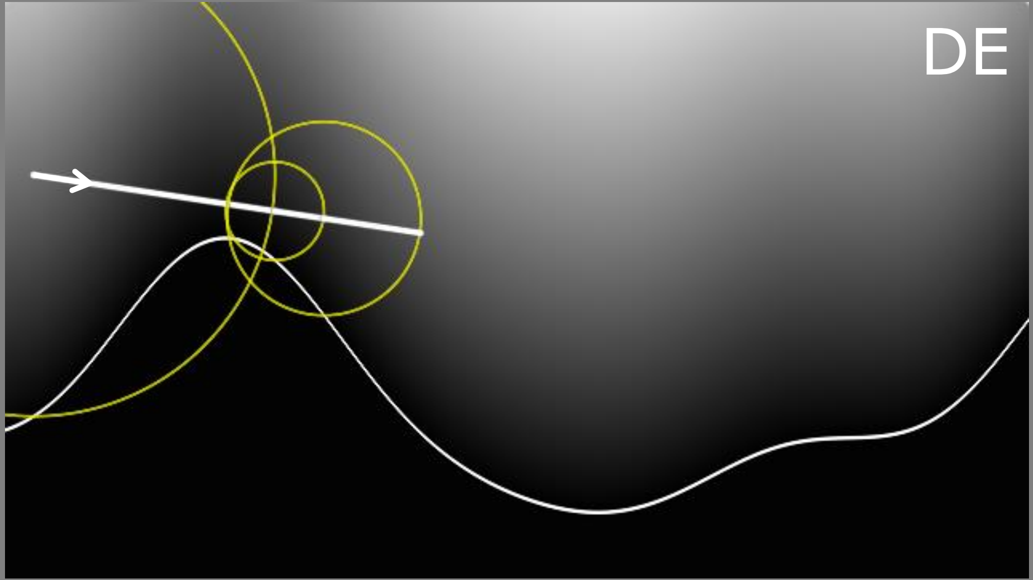

DE

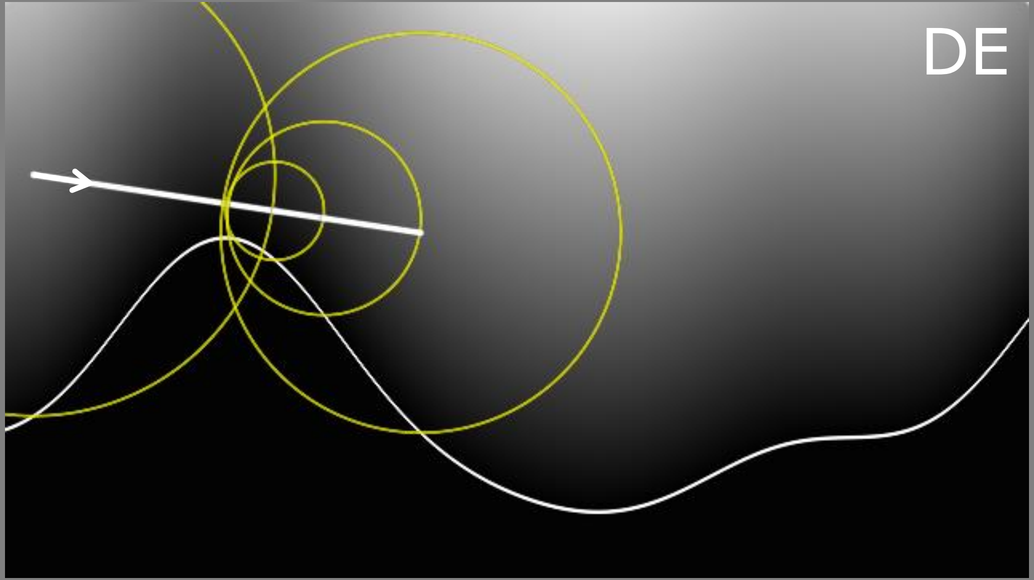

DE

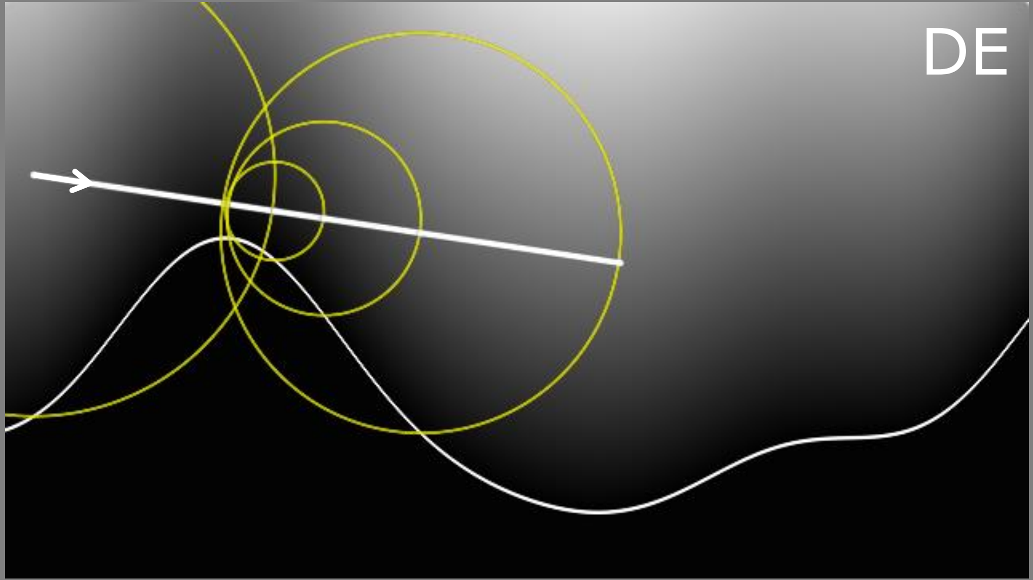

DE

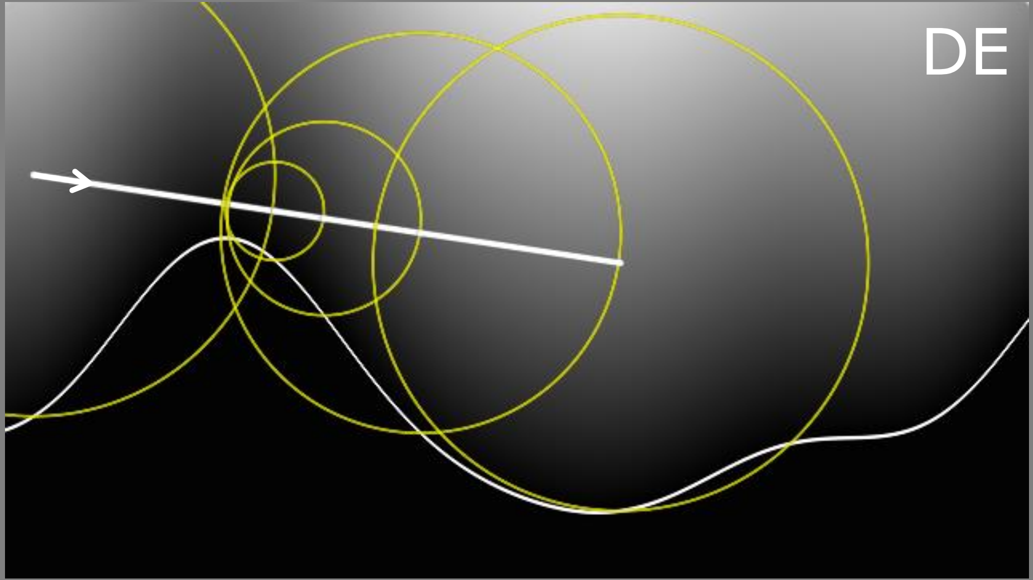

DE

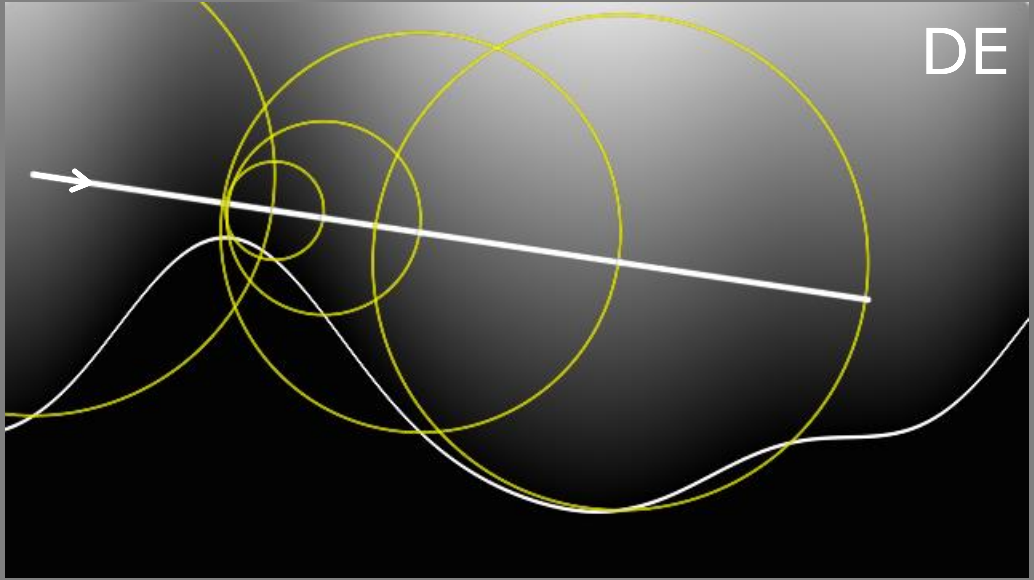

DE

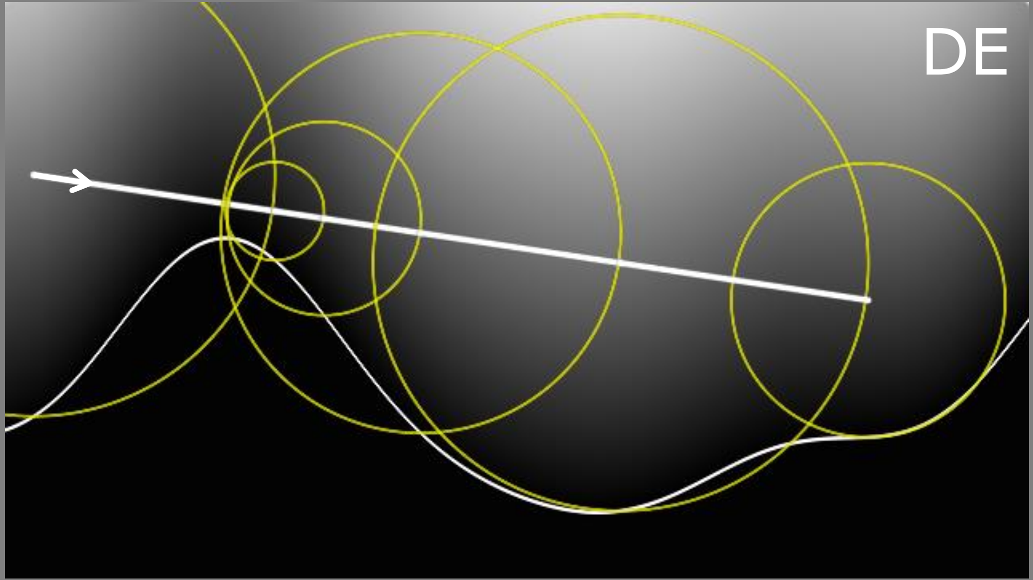

DE

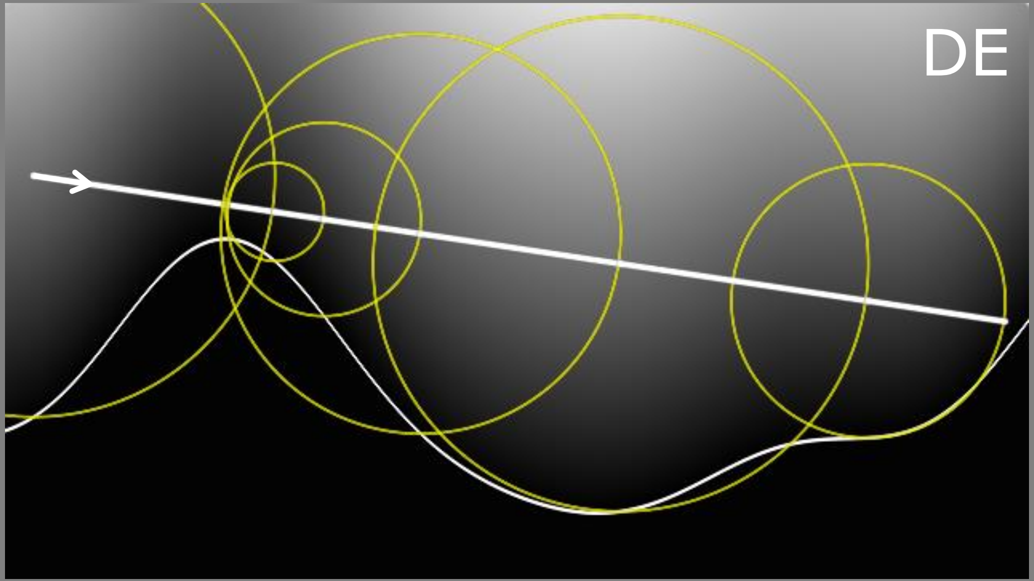

DE

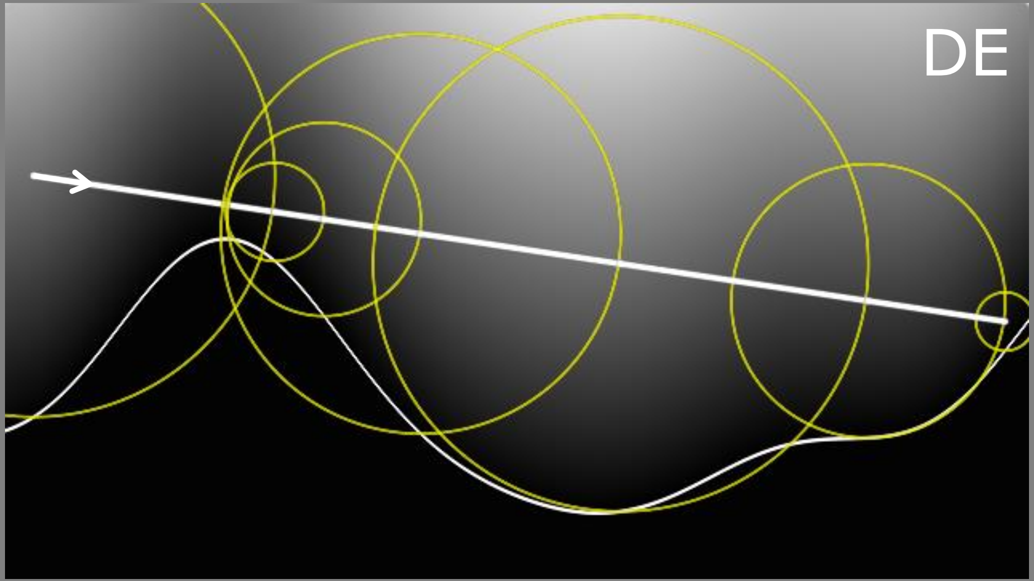

DE

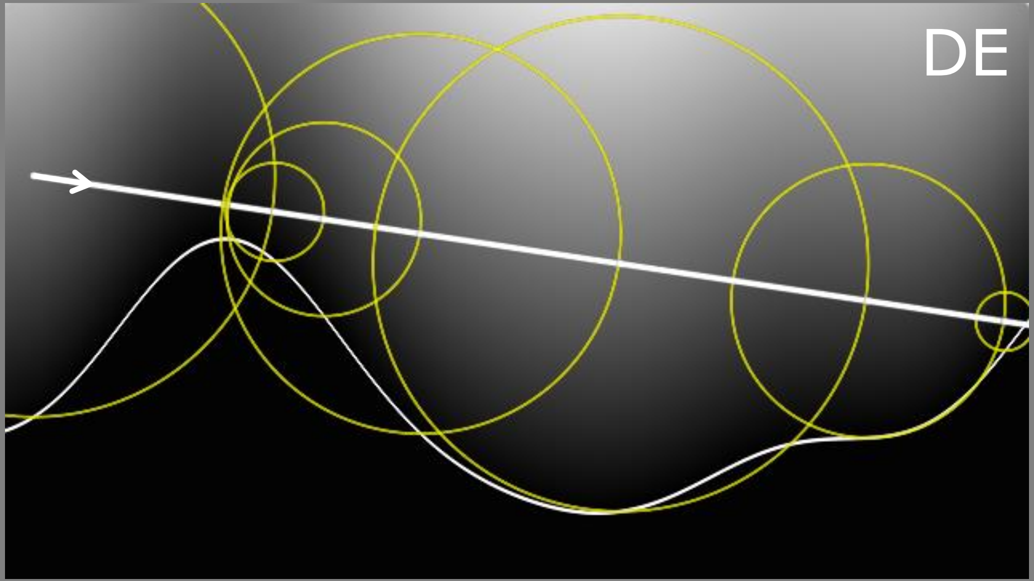

DE


DE


DE


DE


DE


DE

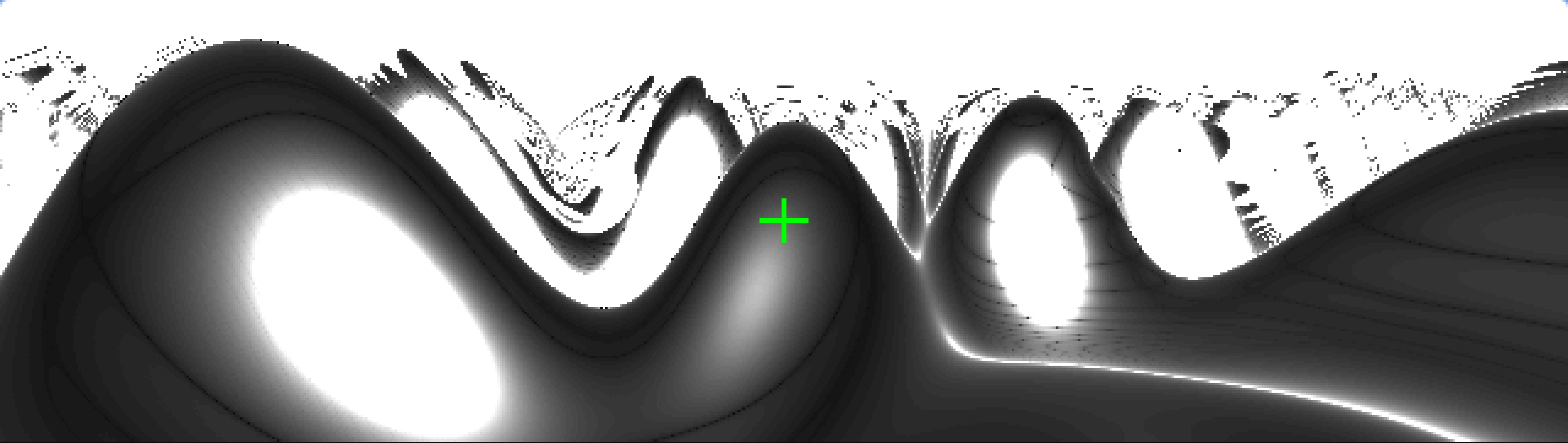
Case 3 Lessons

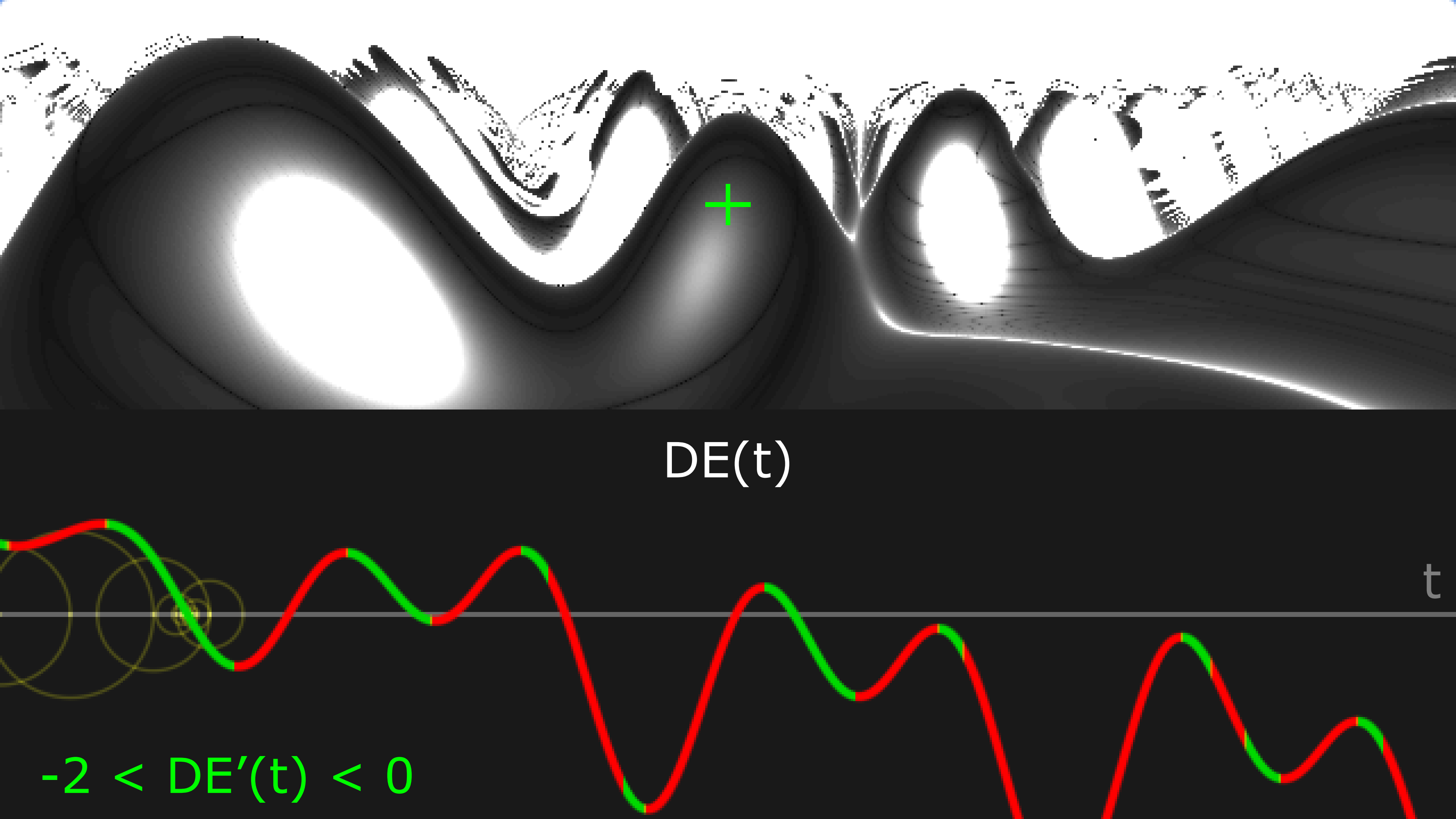
- Raymarching is equivalent to FPI
- Why?
  - FPI is fundamental
  - Any iterative update can be posed as FPI

Convergence Requirements

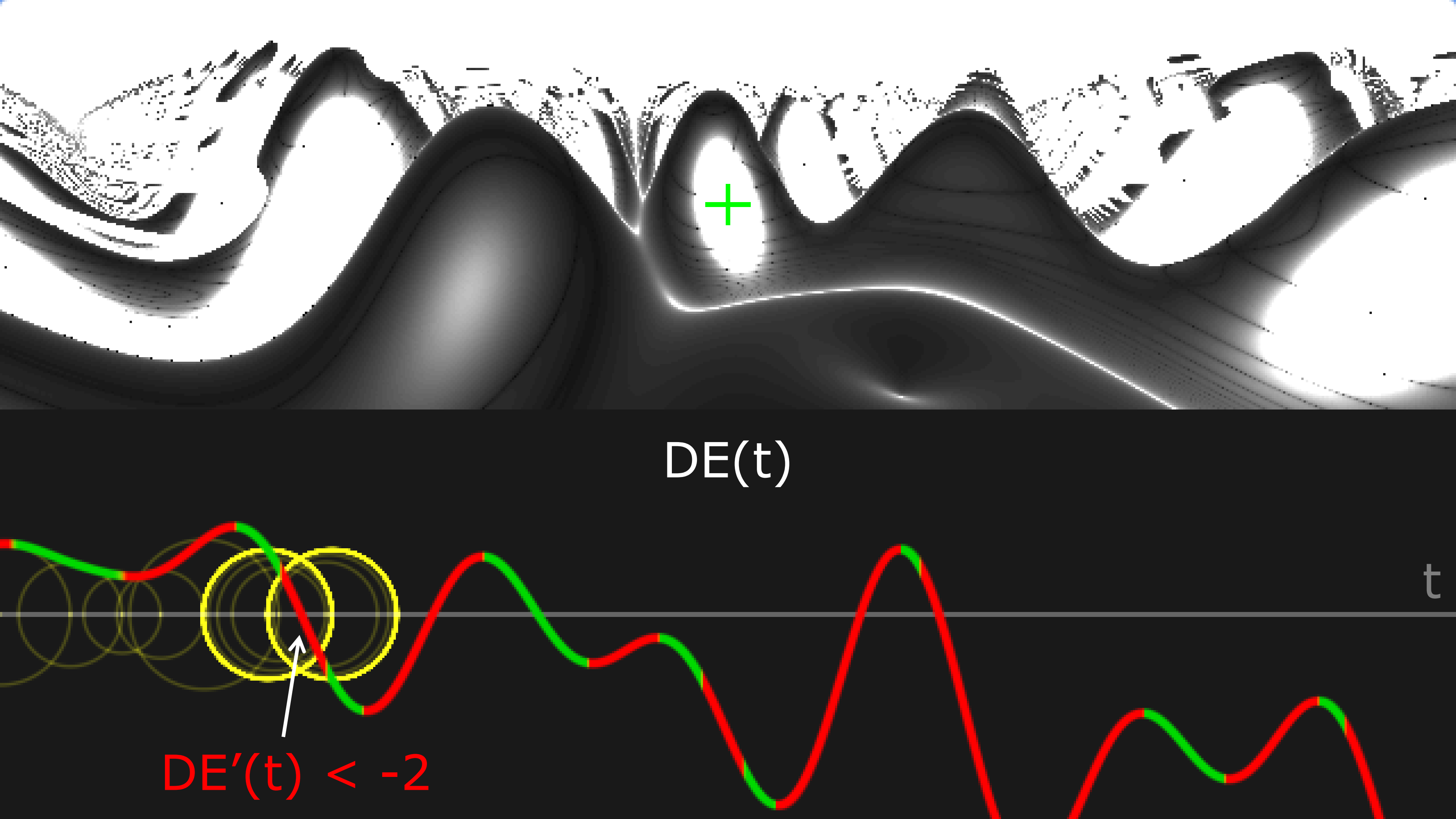
- Standard form: $x = F(x)$
  - Converges when $|F'(x)| < 1$ around solution

- Study raymarch case:
  - $t = t + DE(t)$
  - Converges when $|1 + DE'(t)| < 1$
  - Therefore $-2 < DE'(t) < 0$

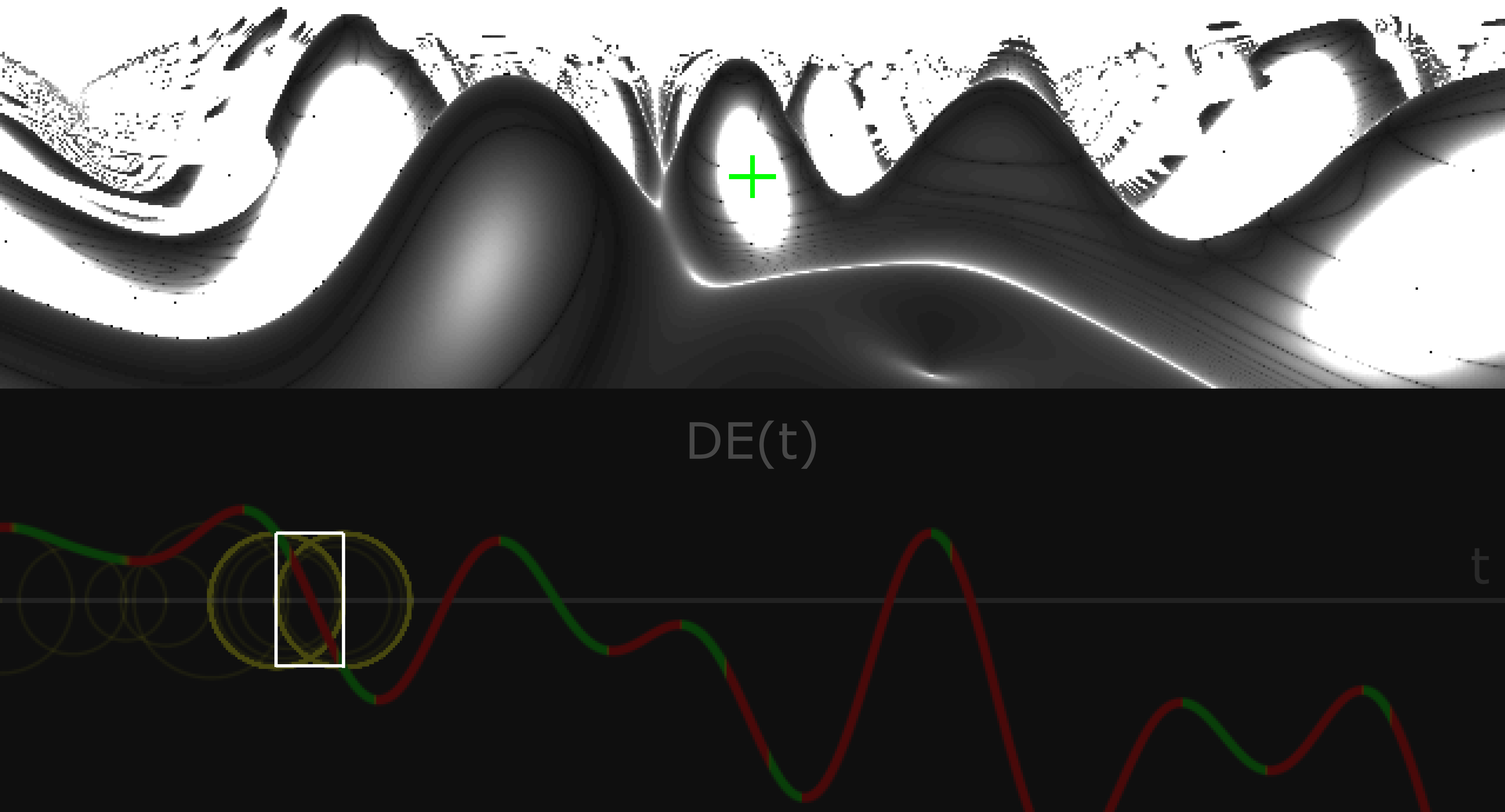


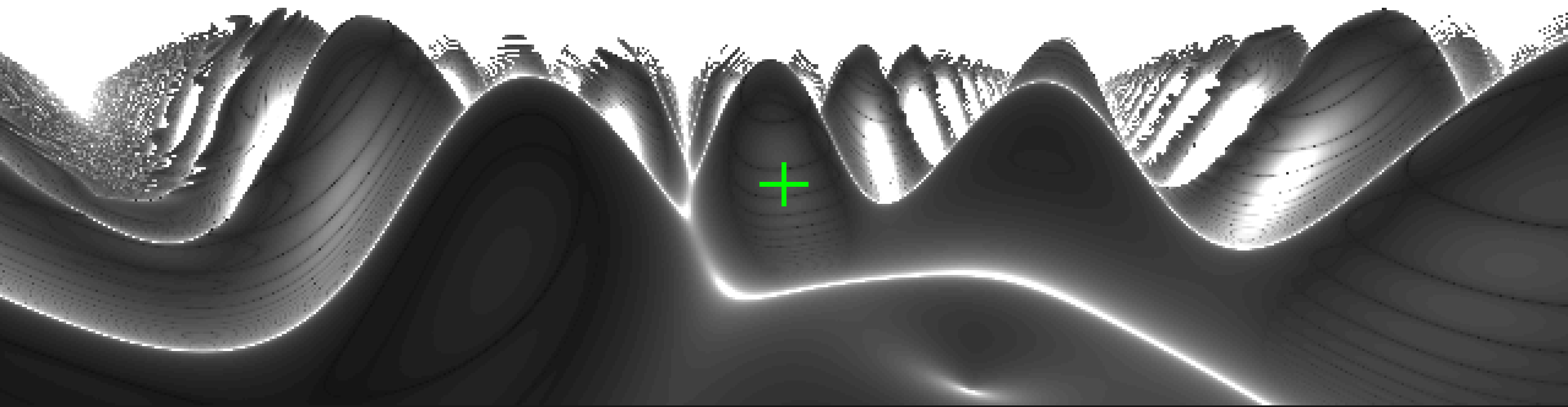


DE(t)
t
-2 < DE′(t) < 0

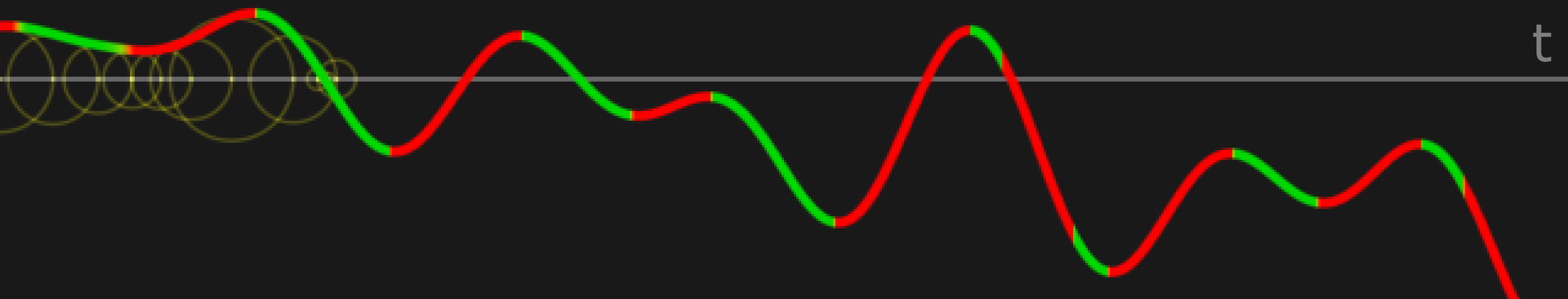


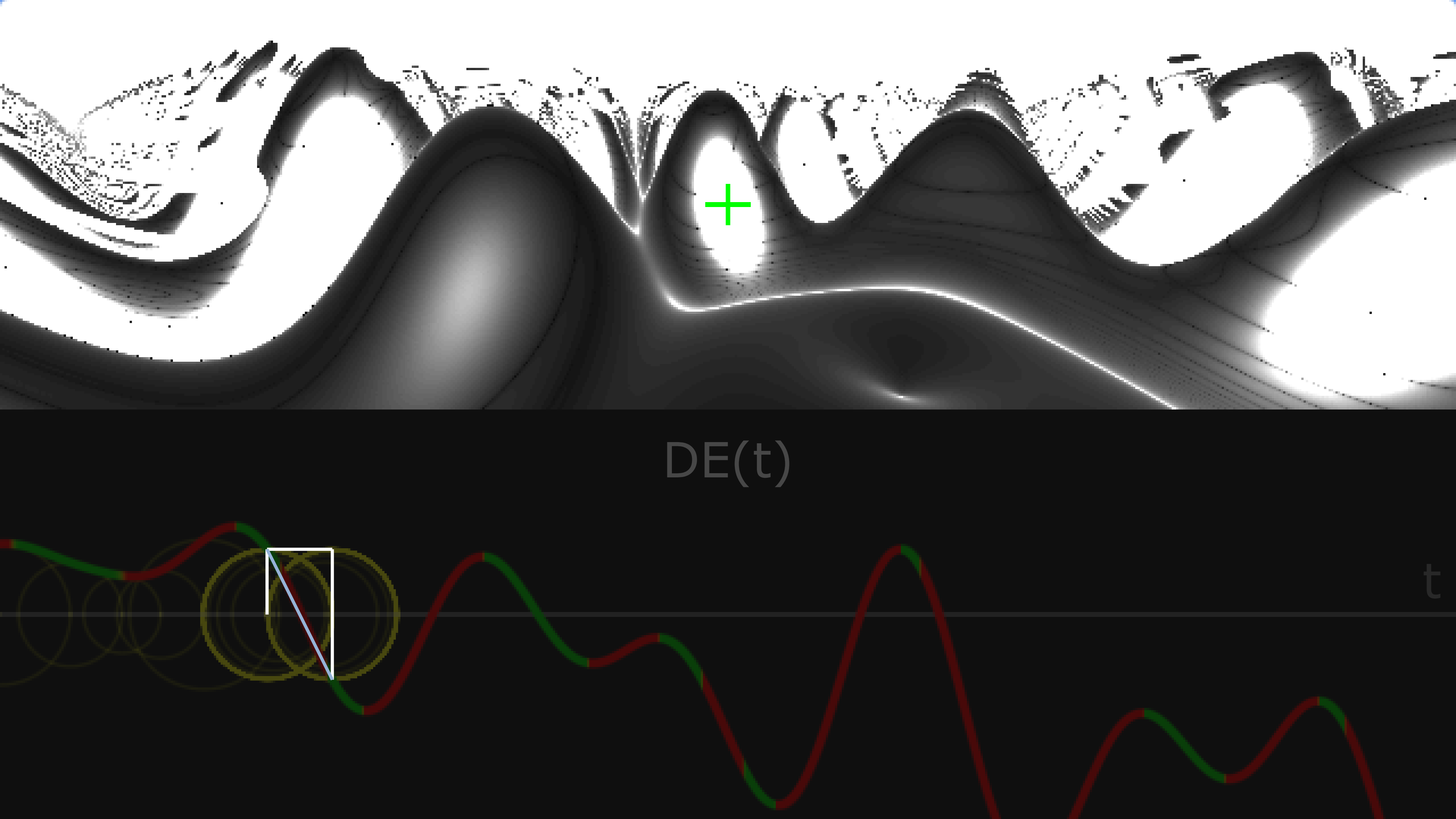
DE(t)
t
DE′(t) < -2

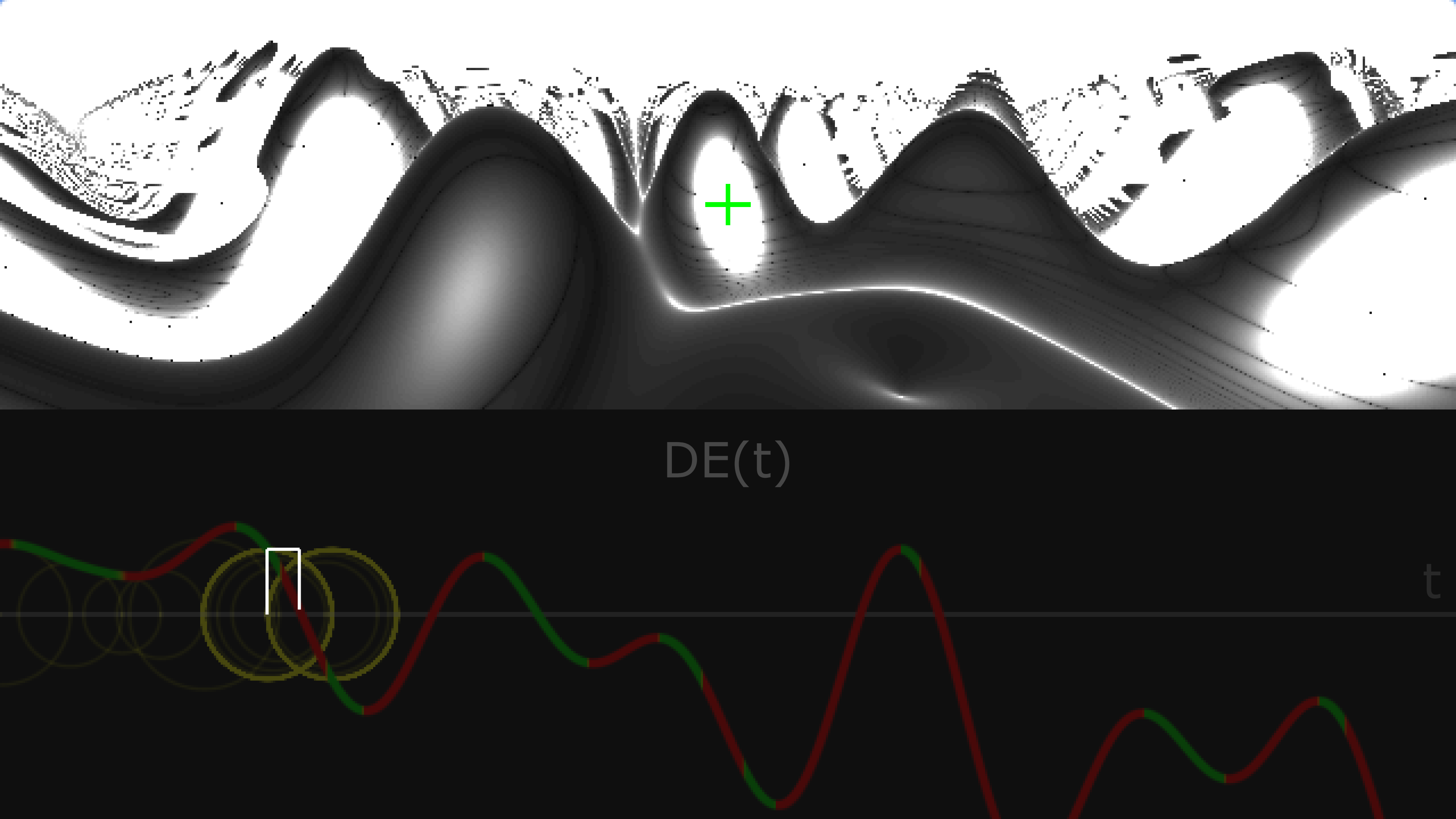

DE(t)
t

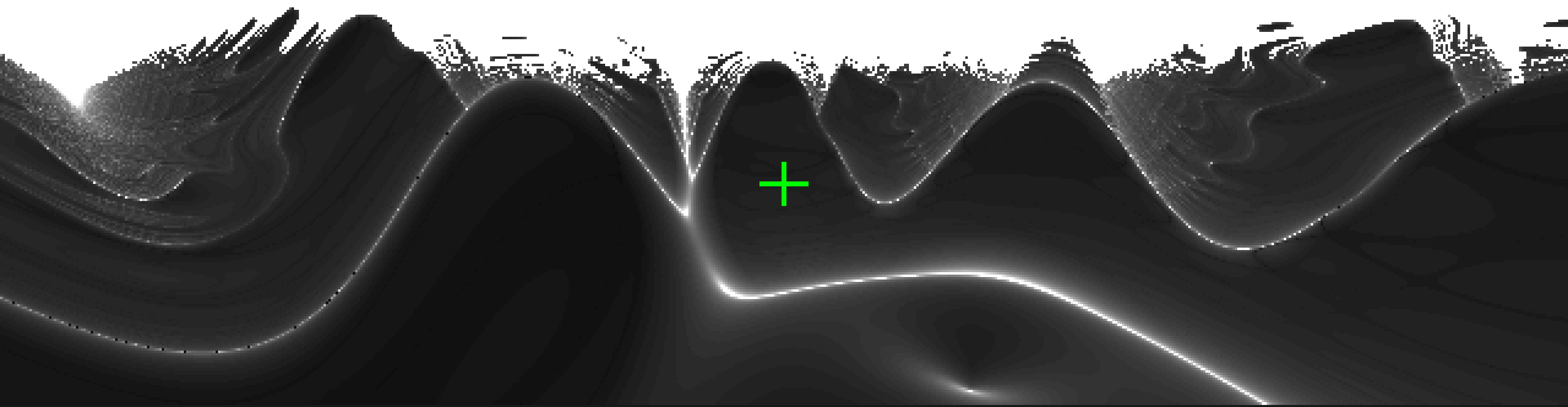


0.7DE(t)

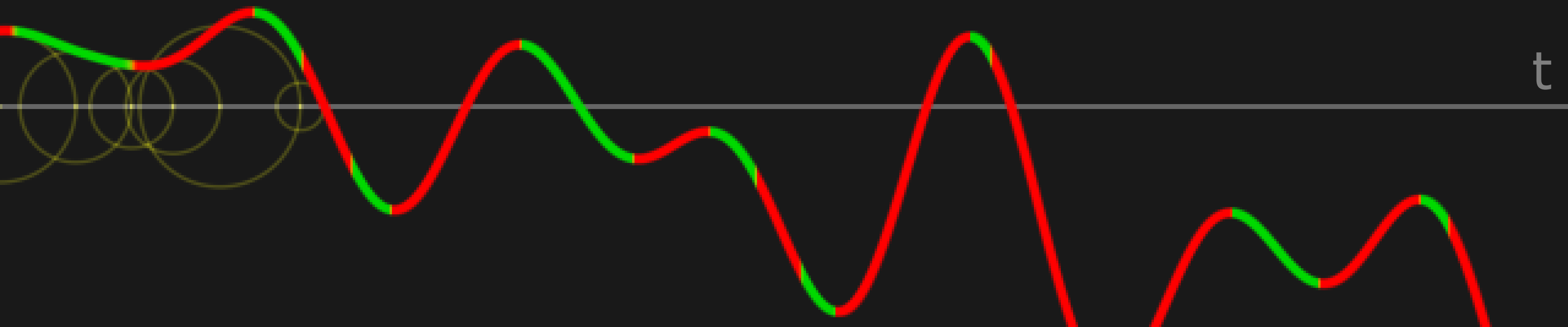



DE(t)
t


DE(t)
t



DE(t)



Milky: shadertoy.com/view/Msy3D1

Power-ups

- If flip flops around solution
  - Secant step
- Jumps over/misses solution? Scale down step…
  - Very common with distance estimators
- Slow to reach solution? Scale up steps…
  - Over-relaxation for raymarching [5]

The Pattern

- Write down equation with unknown
- Analytical solve? You're done.
- Get in form $x = F(x)$
- Iterate $x_{i+1} = F(x_i)$ to taste

Summary

- FPI arises naturally
- Simple and easy to try
- Coherent, stable, good for searching data
  - Unlike Newton!
- If diverges
  - Try to reduce step size
  - Try Secant Step

Summary - Higher Dimensions

- FPI not limited to 1D
- See [3] for application of FPI to image warping
  - Simple to implement, fast
  - Convergence condition for >1D given
  - Behaviour illustrated in the presence of many solutions
  - Technique described to catch all solutions

References

- [1] Bowles, H. and Zimmermann, D. *Oceans on a Shoestring: Shape Representation, Meshing and Shading*, Advances in Real-Time Rendering in Games course, SIGGRAPH 2013
- [2] Gonzales-Ochoa, C. *Water Technology of Uncharted*, GDC 2012
- [3] Bowles, H., Mitchell, K., Sumner, B., Moore, J. and Gross, M.. *Iterative Image Warping*, Eurographics 2012
- [4] Torres, G. and Ricour, H. *Technical Challenges of Assassin's Creed III: Real-Time Water Simulation and High-Quality Linear Gameplay Sequences*, SIGGRAPH Asia 2012
- [5] Keinert, B., Schaefer, H., Korndoerfer, J., Ganse, U. and Stamminger, M., *Enhanced Sphere Tracing*, STAG 2014
- [6] Tessendorf, J. *Simulating Ocean Water*, SIGGRAPH 2001

Thank you for listening!

- And thanks to..
  - Halldor Fannar, Daniel Zimmermann, Studio Gobo

- Misc notes: huwbowles.com/fpi-gdc-2016/

- Contact
  - Email: huw.bowles@gmail.com
  - LinkedIn: linkedin.com/in/huwbowles