Time-optimal and energy-optimal path planning in two-dimensional potential flows

Rose Gebhardt

Thursday May 5, 2022

ENAE788V - Spring 2022

References

[1] Elston, J. and Frew, E. 2010. Unmanned Aircraft Guidance for Penetration of Pretornadic Storms. Journal of Guidance, Control, and Dynamics. American Institute of Aeronautics and Astronautics (AIAA).

[2] Kularatne, D. et al. Time and Energy Optimal Path Planning in General Flows. Robotics: Science and Systems XII. Robotics: Science and Systems Foundation.

[3] Lecture 4: Big-O Notation, Graph Search (time permitting: high-level concept survey), Slide 25

[4] Sethian, J.A. and Vladimirsky, A. 2003. Ordered Upwind Methods for Static Hamilton--Jacobi Equations: Theory and Algorithms. SIAM Journal on Numerical Analysis. Society for Industrial & Applied Mathematics (SIAM).

[5] Free, B.A. and Paley, D.A. 2018. Model-based observer and feedback control design for a rigid Joukowski foil in a Kármán vortex street. Bioinspiration & Biomimetics. IOP Publishing.

[6] Wolek, A. and Woolsey, C. 2015. Feasible Dubins Paths in Presence of Unknown, Unsteady Velocity Disturbances. Journal of Guidance, Control, and Dynamics. American Institute of Aeronautics and Astronautics (AIAA).

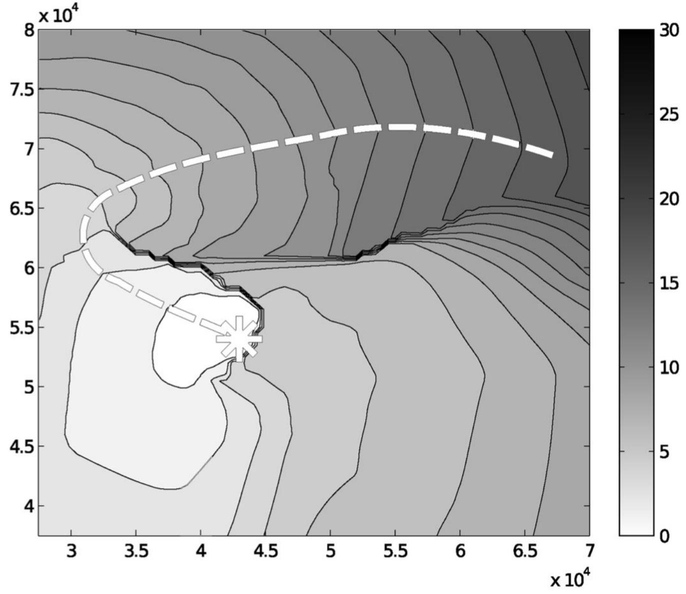
Motivation

Autonomous underwater vehicles (AUVs) will often have to navigate in strong ocean currents with velocities larger than the vehicle's speed which makes accounting for the external velocity field essential

Typical considerations for AUV navigation are speed, energy-efficiency, and safety

The external velocity field in many bio-inspired robotics applications can often be represented by a low-dimensional potential flow field

These methods can also be applicable for aircrafts operating in strong wind fields



Time cost level sets of an air vehicle in a simulated storm [1]

System Definition

Vehicle kinematics

External velocity definition:

$$\mathbf{v}(\mathbf{x}) = \nabla\phi(\mathbf{x})$$

where $\mathbf{x} = [x, y]^T \in \mathbb{R}^2$

$$\mathbf{v}(\mathbf{x}) = [v_x, v_y]^T \in \mathbb{R}^2$$

$$v_x(\mathbf{x}) = \frac{\partial\phi(\mathbf{x})}{\partial x}$$

$$v_y(\mathbf{x}) = \frac{\partial\phi(\mathbf{x})}{\partial y}$$

Kinematic model definition:

Define $\mathbf{X}(t) = [X, Y]^T$ as the position of the swimmer

$$\dot{X}(t) = V\cos(\theta) + v_x(\mathbf{X})$$

$$\dot{Y}(t) = V\sin(\theta) + v_y(\mathbf{X})$$

Where V is the velocity of the swimmer and θ is the orientation of the swimmer

Simplifying assumption: the swimmer can be modeled as a one-dimensional point so we can neglect the effects of dynamics and fluid-structure interactions

Formal Problem Definitions

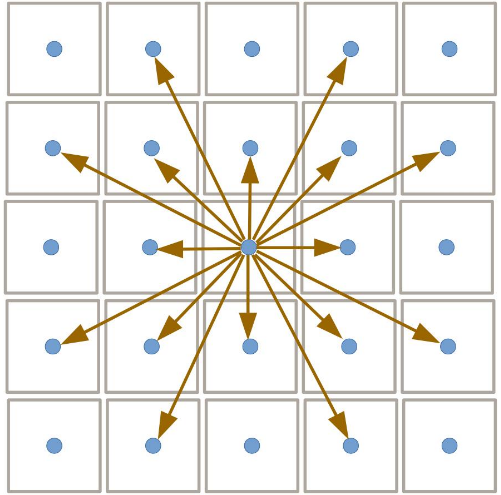
Suppose we are given a discretized workspace $\mathbb{W} \subset \mathbb{R}^2$ with associated external velocity field components $v_x(\mathbf{x})$ and $v_y(\mathbf{x})$, a start location $\mathbf{x}_{start} \in \mathbb{W}$, and a goal location $\mathbf{x}_{goal} \in \mathbb{W}$. Any path Γ in the workspace is subject to the constraints $\Gamma(t = 0) = \mathbf{x}_{start}$, $\Gamma(t = T) = \mathbf{x}_{goal}$, and $V \leq V_{max}$.

Problem Definition 1 [Time Optimization]: Given a differential time cost function dt, find the path $\mathbf{\Gamma}^*$ which minimizes the functional $\int_\Gamma dt$ subject to the constraints specified.

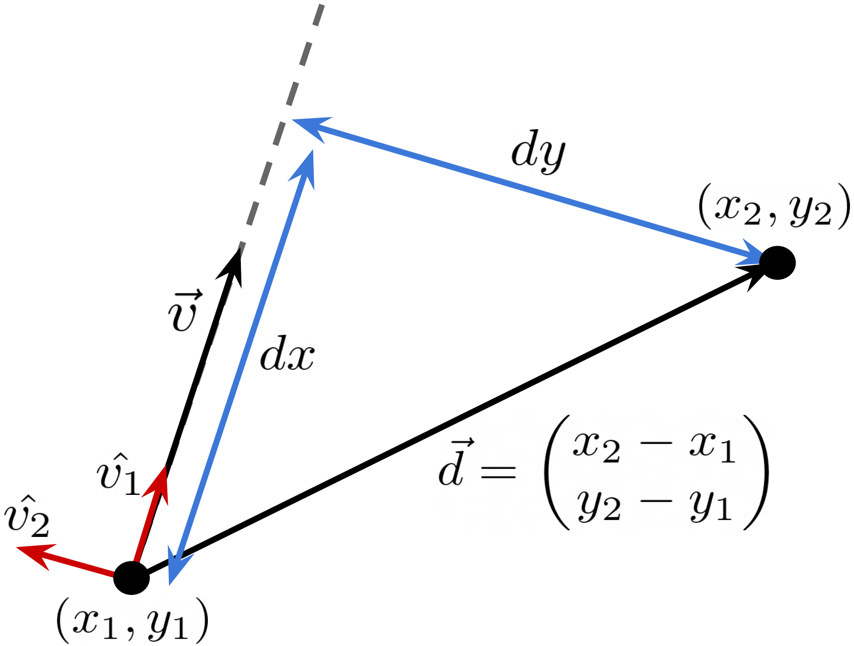
Problem Definition 2 [Energy Optimization]: Given a differential energy cost function dW, find the path $\mathbf{\Gamma}^*$ which minimizes the functional $\int_\Gamma dW$ subject to the constraints specified.

Optimization Algorithms

Workspace definition



Uniform rectangular grid over the workspace, each node has an associated position and external velocity vector, connected to 16 neighboring nodes [2]



Flow velocity coordinates: $\hat{v_1} = \frac{\vec{v}}{\|\vec{v}\|}$ $\hat{v_2} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \hat{v_1}$

In flow velocity coordinates, flow velocity and distance vectors are:

$$\vec{v} = \begin{pmatrix} v \\ 0 \end{pmatrix} \qquad \vec{d} = \begin{pmatrix} dx \\ dy \end{pmatrix}$$

Time cost function

If V_{max} is the maximum speed of the swimmer, travel time is minimized when

$$\underbrace{V_{max}^2}_{\text{movement caused by swimmer}} = \underbrace{\left(\frac{\mathrm{d}x}{\mathrm{d}t} - v\right)^2}_{\text{movement in } \hat{v}_1\text{-direction without flow field contribution}} + \underbrace{\left(\frac{\mathrm{d}y}{\mathrm{d}t}\right)^2}_{\text{movement in } \hat{v}_2\text{-direction}}$$

Resulting in the quadratic equation:

$$(v^2 - V_{max}^2)\mathrm{d}t^2 - (2v\mathrm{d}x)\mathrm{d}t + (\mathrm{d}x^2 + \mathrm{d}y^2) = 0$$

$$\mathrm{d}t = \frac{v\mathrm{d}x - \sqrt{V_{max}^2(\mathrm{d}x^2 + \mathrm{d}y^2) - v^2\mathrm{d}y^2}}{v^2 - V_{max}^2}$$

Note: If dt is negative or complex, the motion is not possible and the cost function is set to infinity

Energy cost function

Let dx_{still} and v_{still} be the displacement and velocity of the swimmer without an external flow field and let κ be the drag coefficient. The drag force and the displacement over which the force is applied are

$$\vec{F} = \kappa \vec{v}_{still}, \ \mathrm{d}x_{still} = \vec{v}_{still}\mathrm{d}t$$

Thus, the energy exerted to overcome the drag force is

$$\mathrm{d}W = \vec{F} \cdot \mathrm{d}x_{still} = \kappa \|\vec{v}_{still}\|^2 \mathrm{d}t = \kappa \left(\left(\frac{\mathrm{d}x}{\mathrm{d}t} - v \right)^2 + \left(\frac{\mathrm{d}y}{\mathrm{d}t} \right)^2 \right) \mathrm{d}t$$

Notice that dt is energy optimal when the movement is entirely due to the external flow field

$$\mathrm{d}t = \frac{\sqrt{\mathrm{d}x^2 + \mathrm{d}y^2}}{v}$$

Plugging in this value lets us write the cost function independent of time

$$\mathrm{d}W = 2\kappa v \left(\sqrt{\mathrm{d}x^2 + \mathrm{d}y^2} - \mathrm{d}x \right)$$

Dijkstra’s and A* Algorithm

Graph-based search methods which guarantees resolution-optimal path [3]

```
1. UNVISITED ←V\{start}
2. Q.INSERT(start)
3. while Q.TOP() ≠ Ø
        4. v ← Q.POP()
        5. while v.NextNeighbor() ≠ Ø
                6. u ← v.NextNeighbor ()
                7. if u ∈ UNVISITED or u.costToStart > v.costToStart + Cost(v,u)
                        8. UNVISITED ← UNVISITED\{u}
                        9. u.parent ← v
                        10. u.costToStart ← v.costToStart + Cost(v,u)
                        11. Q.UPDATE(u,u.costToStart+ h(u,goal))          // insert if u not in Q
        12. if v = goal
                13. return SUCCESS
14. return FAILURE
```

Time-optimization heuristic:

$$h(\vec{x}, \vec{x_{goal}}) = \frac{(x_{goal} - x)^2 + (y_{goal} - y)^2}{V_{max} + v_{max}}$$

Energy-optimization heuristic:

$$h(\vec{x}, \vec{x_{goal}}) = 0$$

Wavefront Expansion Algorithm

Produces a cost-to-go map over a discretized configuration space by propagating backward from the goal configuration

Ordered upwind method can be applied to general anisotropic cases

Produces a cost map which converges to the viscosity solution of Hamilton-Jacobi partial differential equation, solves the optimal control problem [1]

Hamilton-Jacobi Equations:

$$\begin{aligned} &\|\nabla u\| F\left(\boldsymbol{x}, \frac{\nabla u}{\|\nabla u\|}\right) = 1, \quad && \boldsymbol{x} \in \Omega, \\ &u(\boldsymbol{x}) = 0, && \boldsymbol{x} \in \partial\Omega. \end{aligned}$$

u = viscosity solution
F = velocity at position x in direction of gradient of u
Ω = workspace
∂**Ω** = goal region

Control-theoretic ordered upwind method [4]

1. Start with all the mesh points in Far.
2. Move the mesh points on the boundary ($\boldsymbol{y} \in \partial\Omega$) to *Accepted* ($U(\boldsymbol{y}) = q(\boldsymbol{y})$).
3. Move all the mesh points $\boldsymbol{x}$ adjacent to the boundary into $Considered$ and evaluate the tentative values

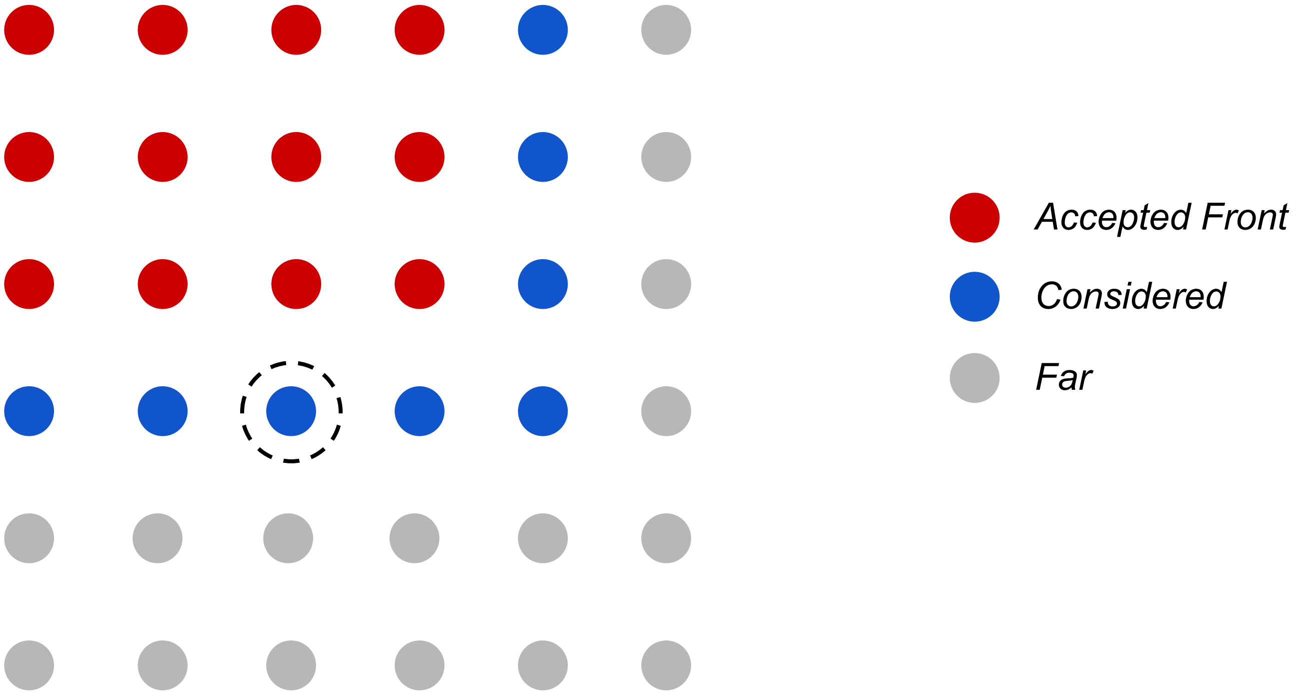
$$V(\boldsymbol{x}) := \min_{\boldsymbol{x}_j\boldsymbol{x}_k \in \mathrm{NF}(\boldsymbol{x})} V_{\boldsymbol{x}_j,\boldsymbol{x}_k}(\boldsymbol{x}). \tag{32}$$

4. Find the mesh point $\bar{\boldsymbol{x}}$ with the smallest value of V among all the $Considered$.
5. Move $\bar{\boldsymbol{x}}$ to *Accepted* ($U(\bar{\boldsymbol{x}}) = V(\bar{\boldsymbol{x}})$) and update the $AcceptedFront$.
6. Move the Far mesh points adjacent to $\bar{\boldsymbol{x}}$ into $Considered$ and compute their tentative values by (32).
7. Recompute the value for all the other $Considered$ $\boldsymbol{x}$ such that $\bar{\boldsymbol{x}} \in \mathrm{NF}(\boldsymbol{x})$

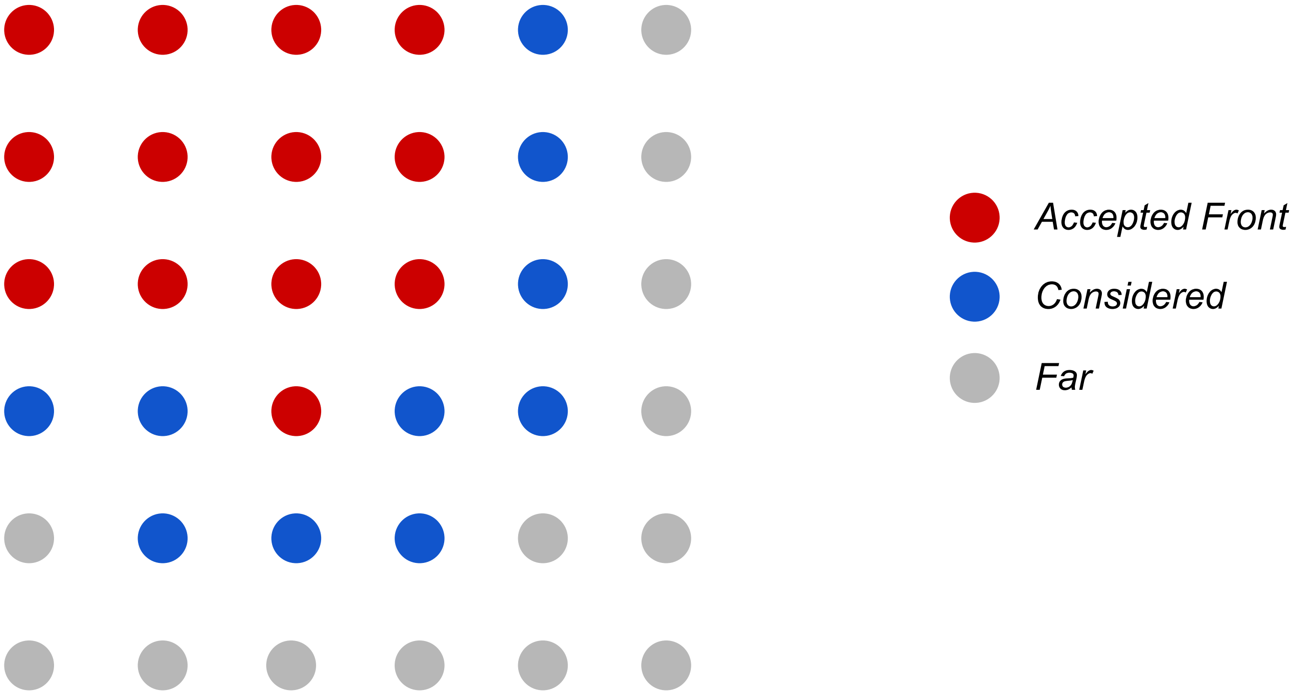
$$V(\boldsymbol{x}) := \min\left\{V(\boldsymbol{x}), \min_{\bar{\boldsymbol{x}}\boldsymbol{x}_i \in \mathrm{NF}(\boldsymbol{x})} V_{\bar{\boldsymbol{x}},\boldsymbol{x}_i}(\boldsymbol{x})\right\}. \tag{33}$$

8. If $Considered$ is not empty, then go to 4.

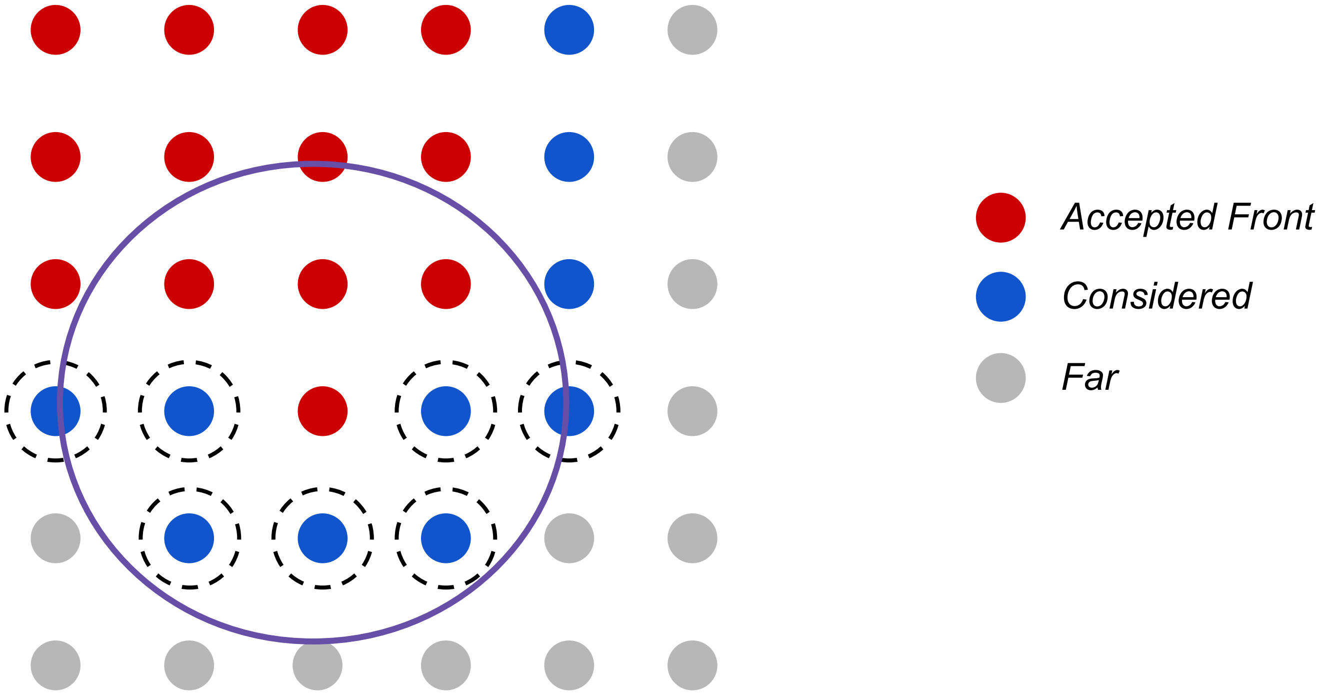
Control-theoretic ordered upwind method



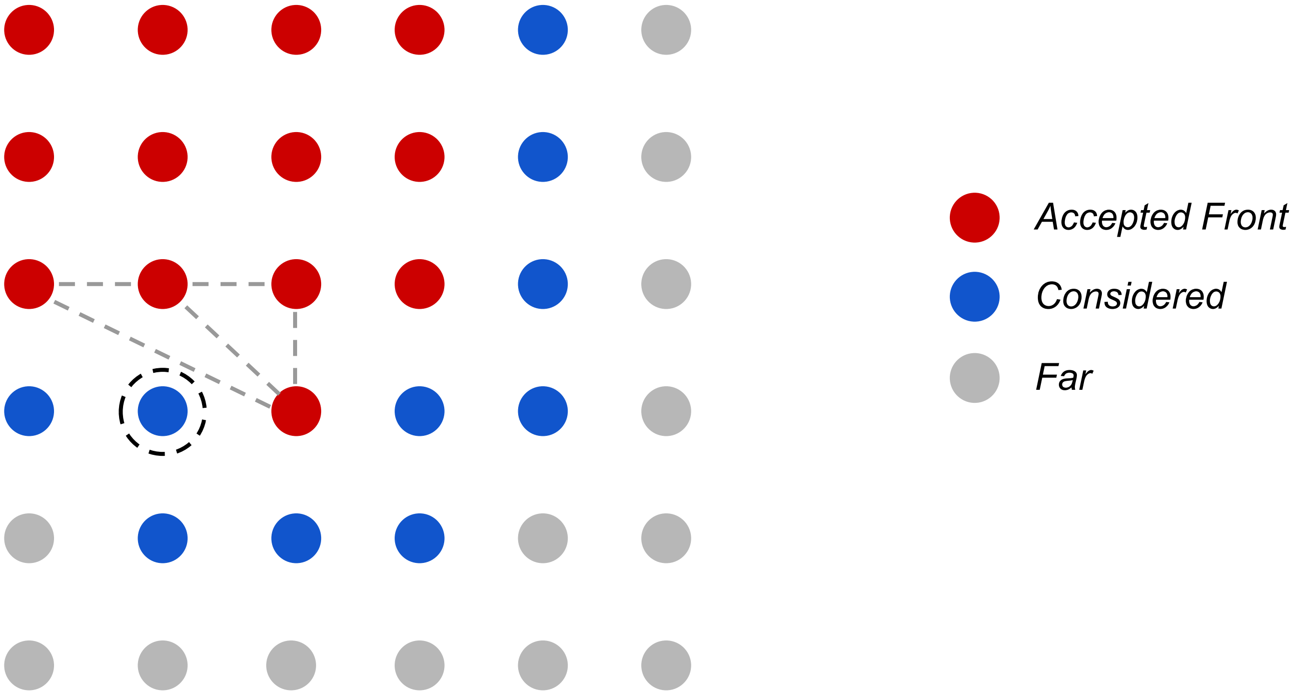
Control-theoretic ordered upwind method



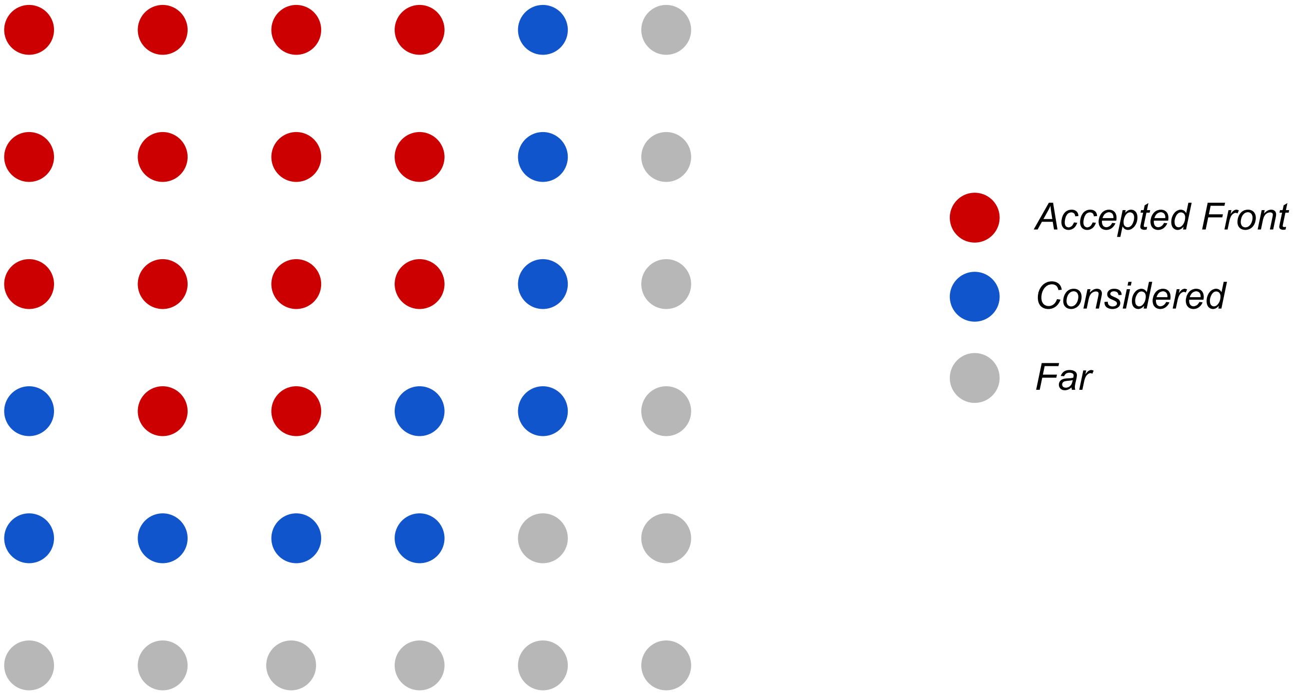
Control-theoretic ordered upwind method



Control-theoretic ordered upwind method



Control-theoretic ordered upwind method

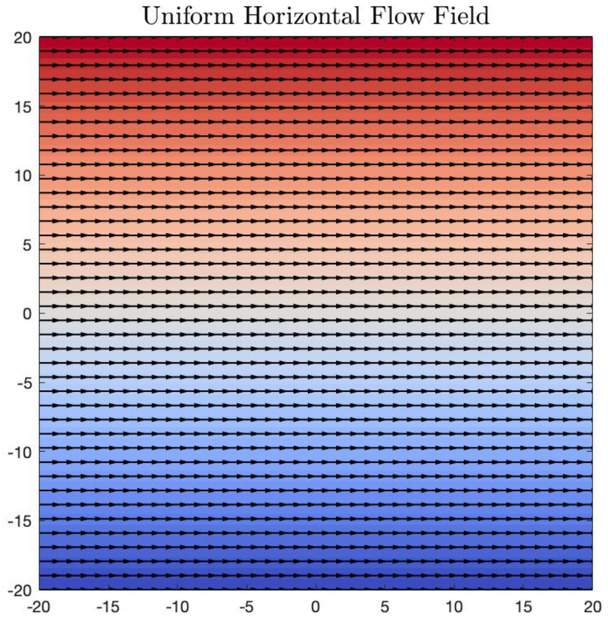


Simulations

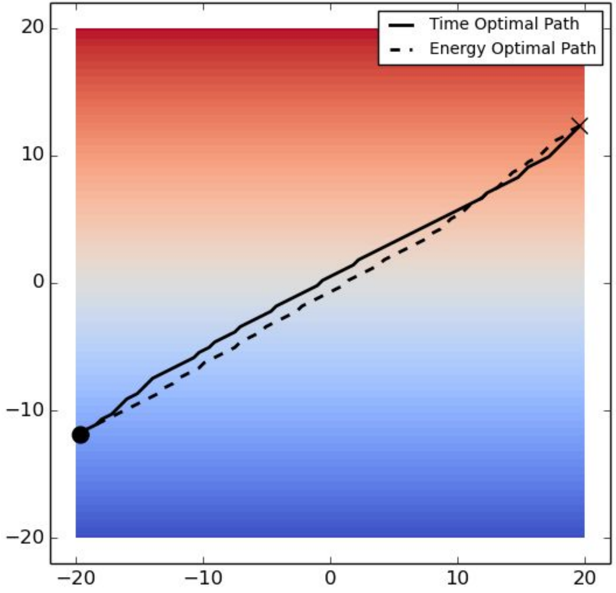
Uniform horizontal flow

Time-optimization vs. energy-optimization

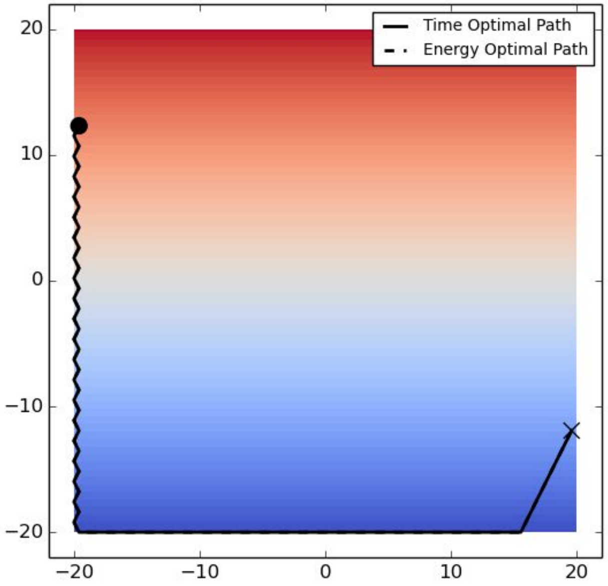
Few cases run, used mainly for debugging



Quiver plot of flow field



Moving in direction of flow, time-optimal and energy-optimal paths are roughly the same

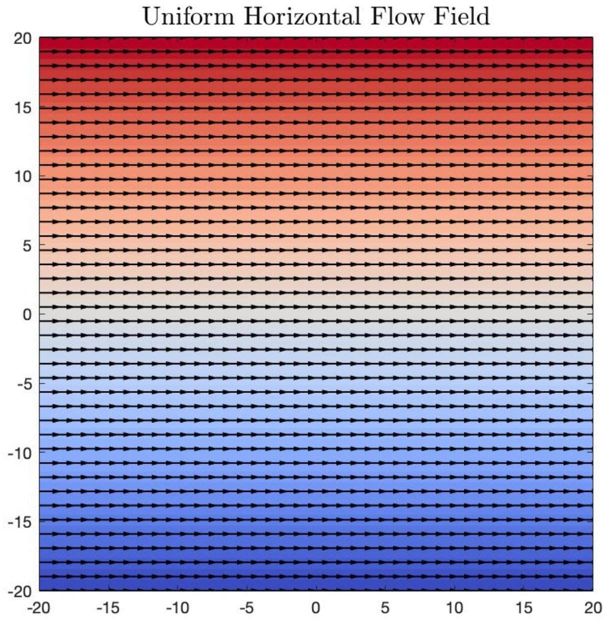


Moving against flow (u_0 = 1) with small maximum velocity (V_{max} = 0.25). Fails to find a path with finite cost, as expected.

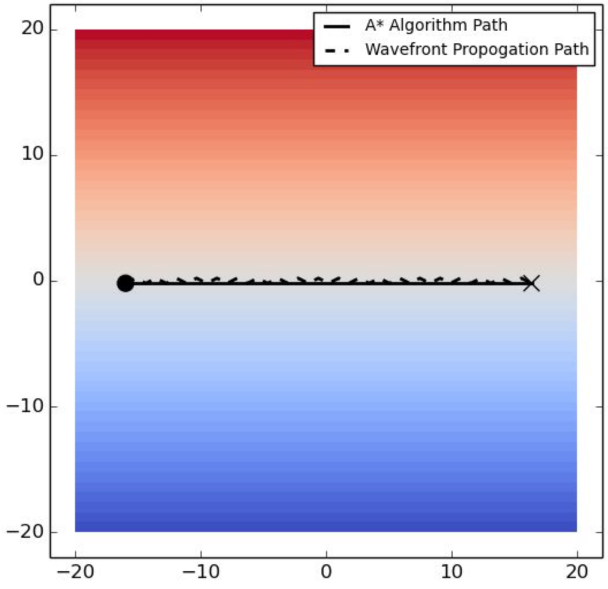
Uniform horizontal flow

A vs. wavefront propagation*

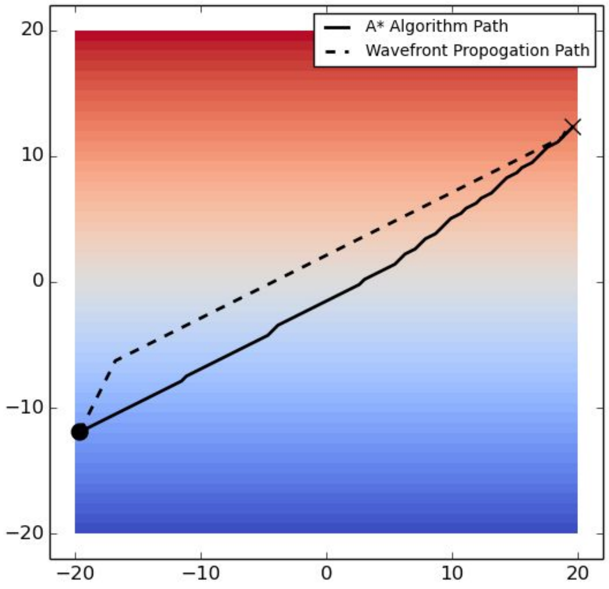
Few cases run, used mainly for debugging



Quiver plot of flow field

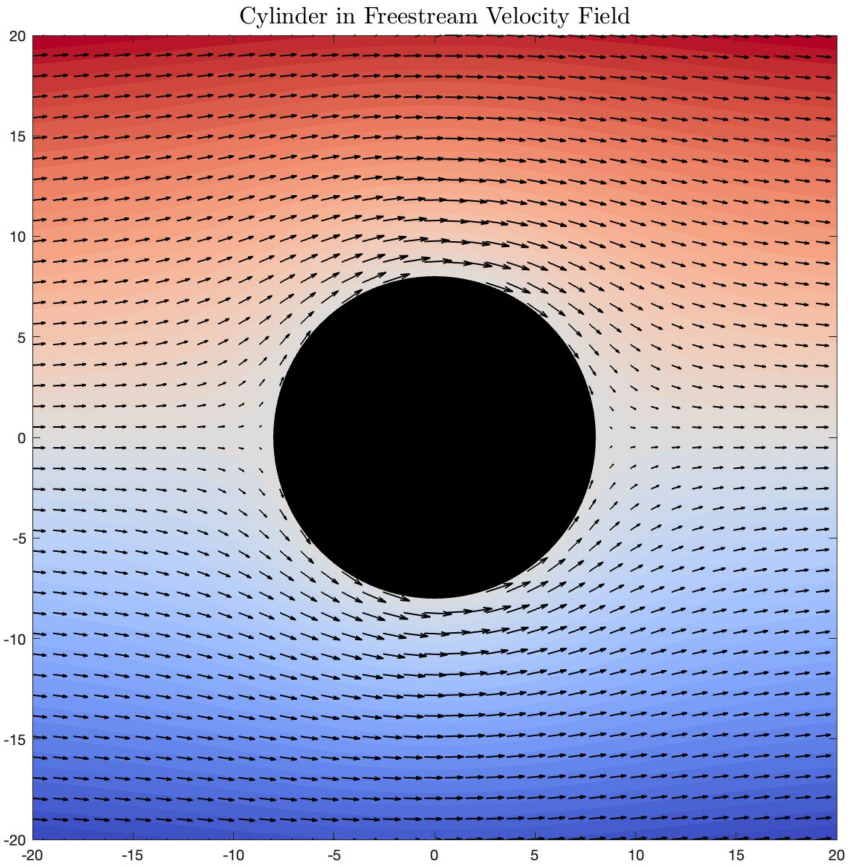


Simplest case: wavefront propagation algorithm shows some deviation from direct path



Similar behavior, A* takes a more direct path

Cylinder in freestream model



Complex flow potential:

$$F(z) = u_0 \left(z + \frac{r_0^2}{z} \right)$$

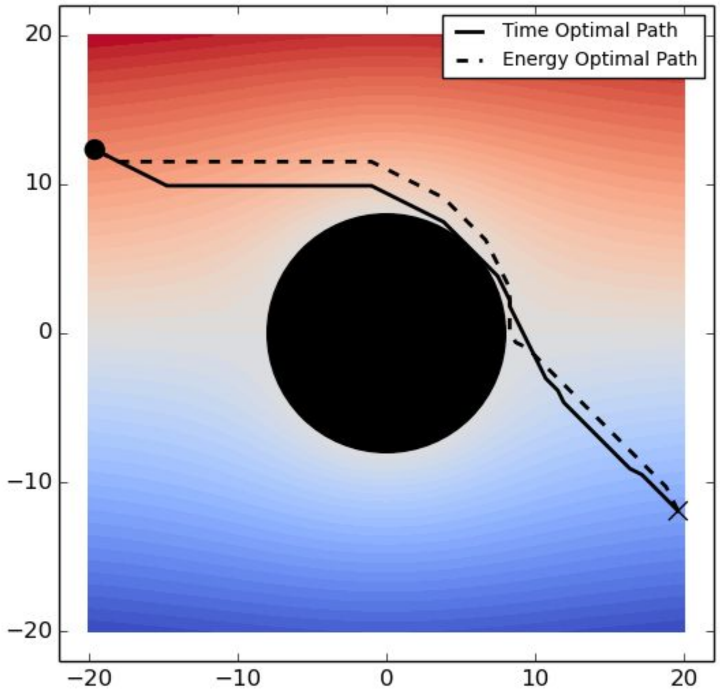
Complex conjugate velocity:

$$\frac{\mathrm{d}F(z)}{\mathrm{d}z} = u_0 \left(1 - \frac{r_0^2}{z^2} \right) = v_x - i v_y$$

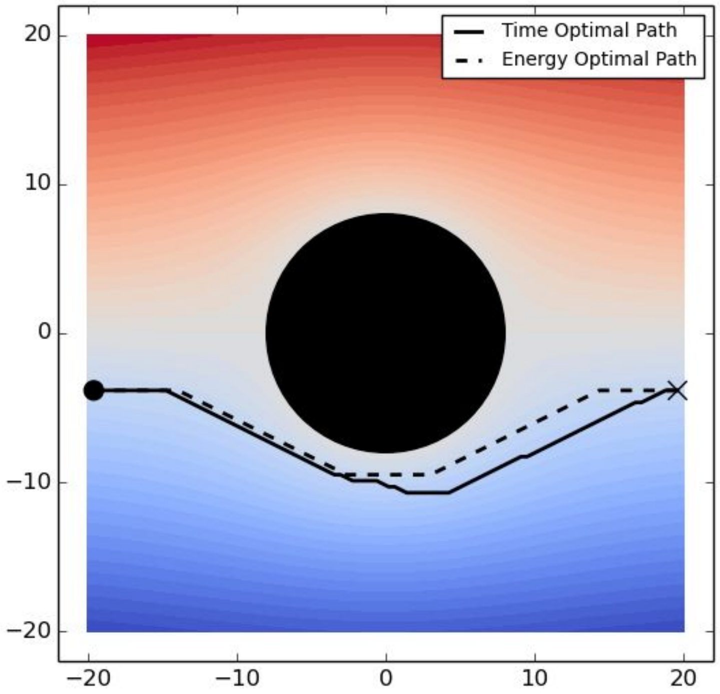
Parameters used: $u_0 = 1,\ r_0 = 8$

Cylinder in freestream: in direction of flow

Time-optimization vs. energy-optimization



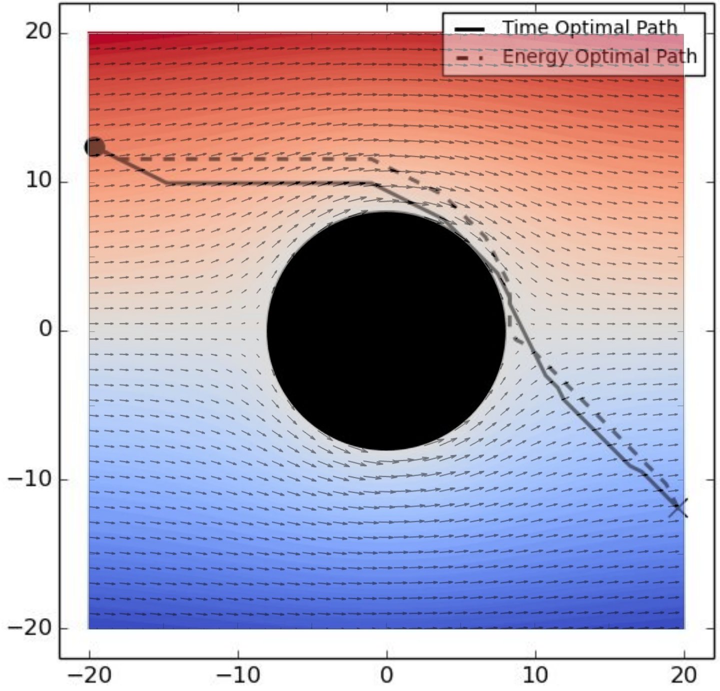
The time-optimal path is more direct than the energy-optimal path



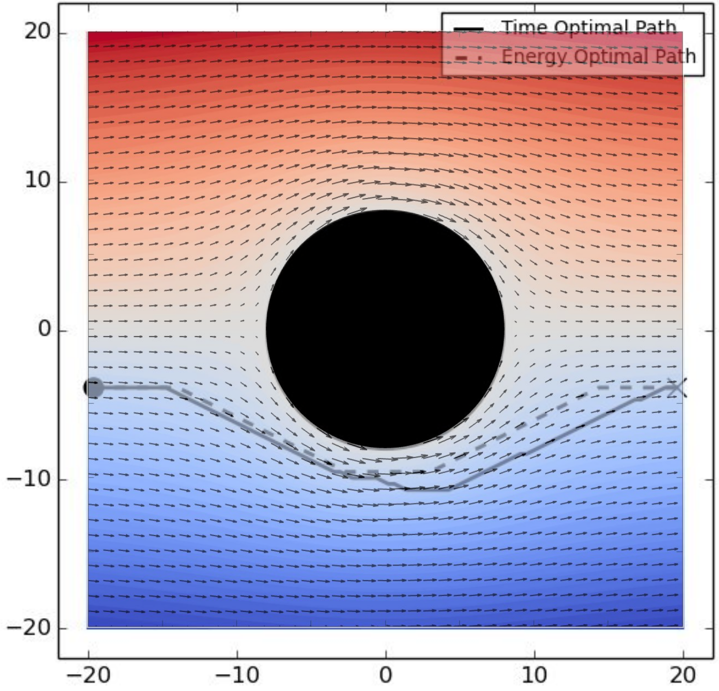
Time-optimal path is less direct than the energy-optimal path: moves to a region with a higher velocity

Cylinder in freestream: in direction of flow (velocity overlaid)

Time-optimization vs. energy-optimization



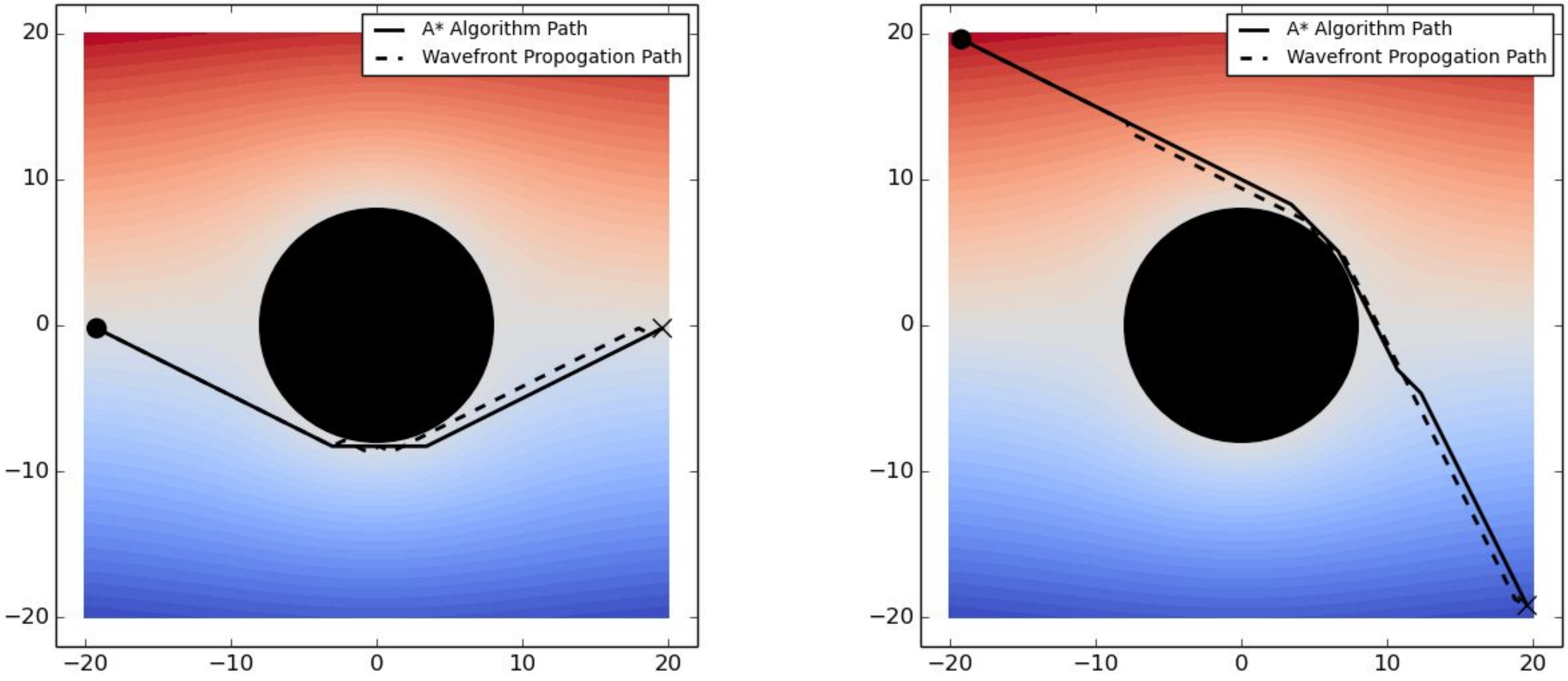
The time-optimal path is more direct than the energy-optimal path



Time-optimal path is less direct than the energy-optimal path: moves to a region with a higher velocity

Cylinder in freestream: in direction of flow

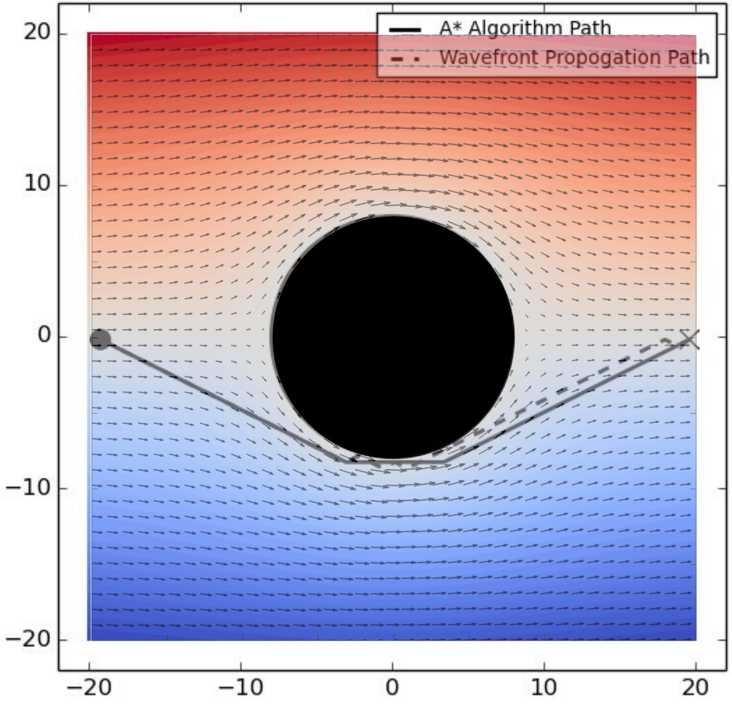
A vs. wavefront propagation*

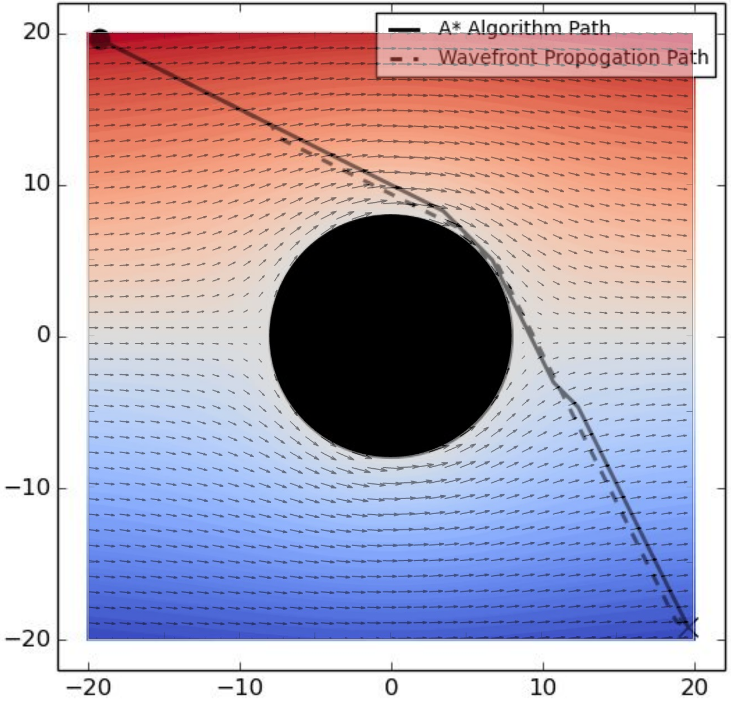


The paths found by the two optimization algorithms are in close agreement

Cylinder in freestream: in direction of flow

A vs. wavefront propagation*

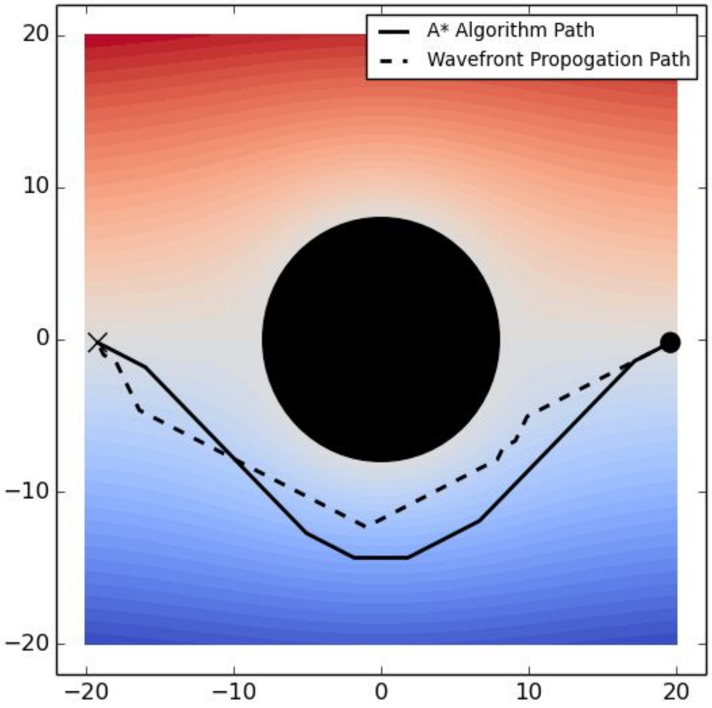


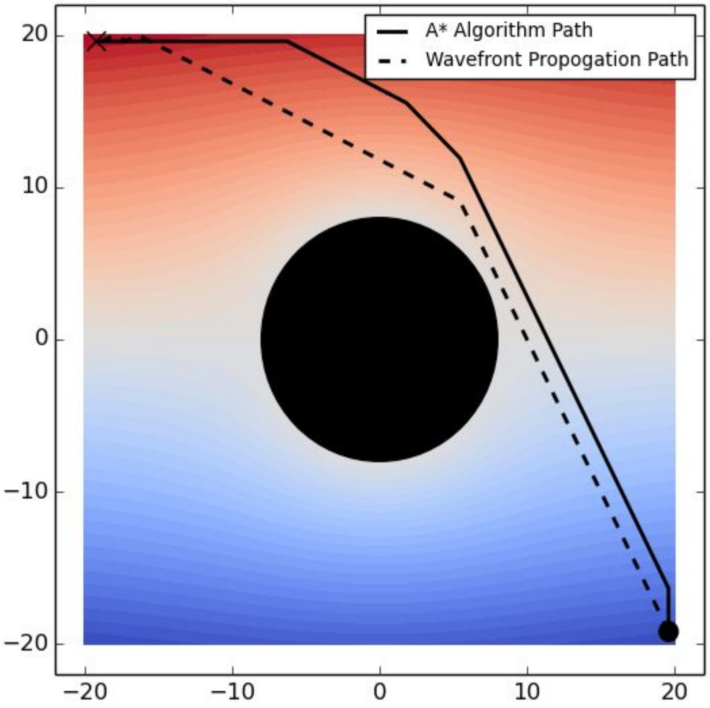


The paths found by the two optimization algorithms are in close agreement

Cylinder in freestream: against direction of flow

A vs. wavefront propagation*

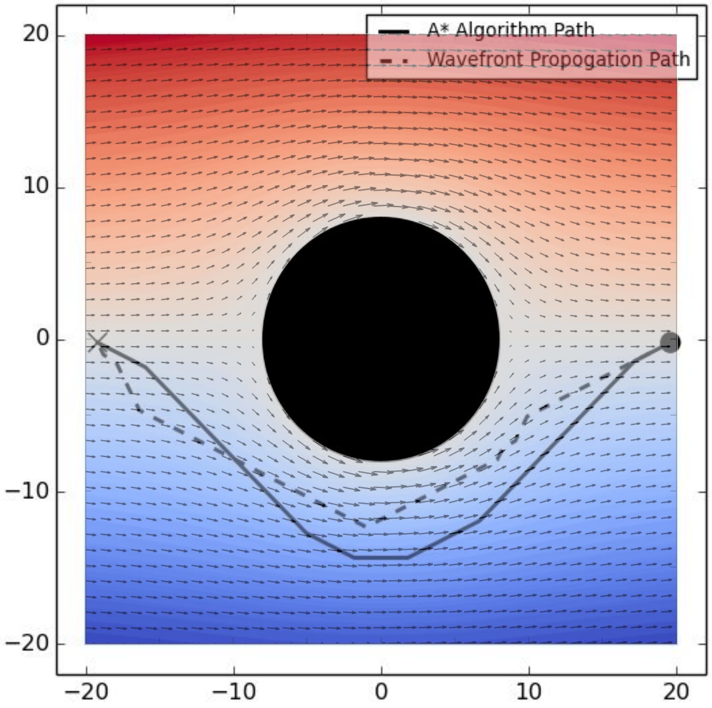


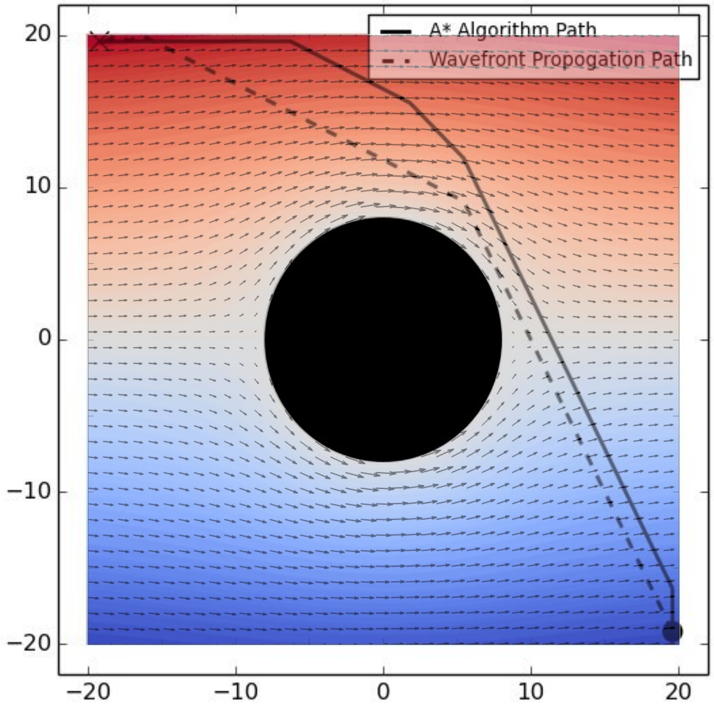


Similar general behavior, wavefront propagation methods seems to be finding a more direct path

Cylinder in freestream: against direction of flow

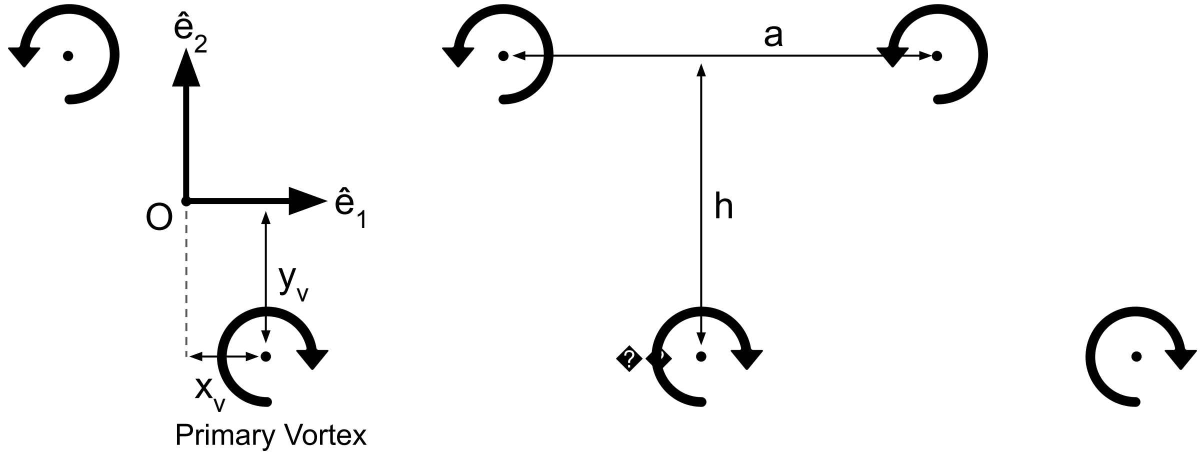
A vs. wavefront propagation*





Similar general behavior, wavefront propagation methods seems to be finding a more direct path

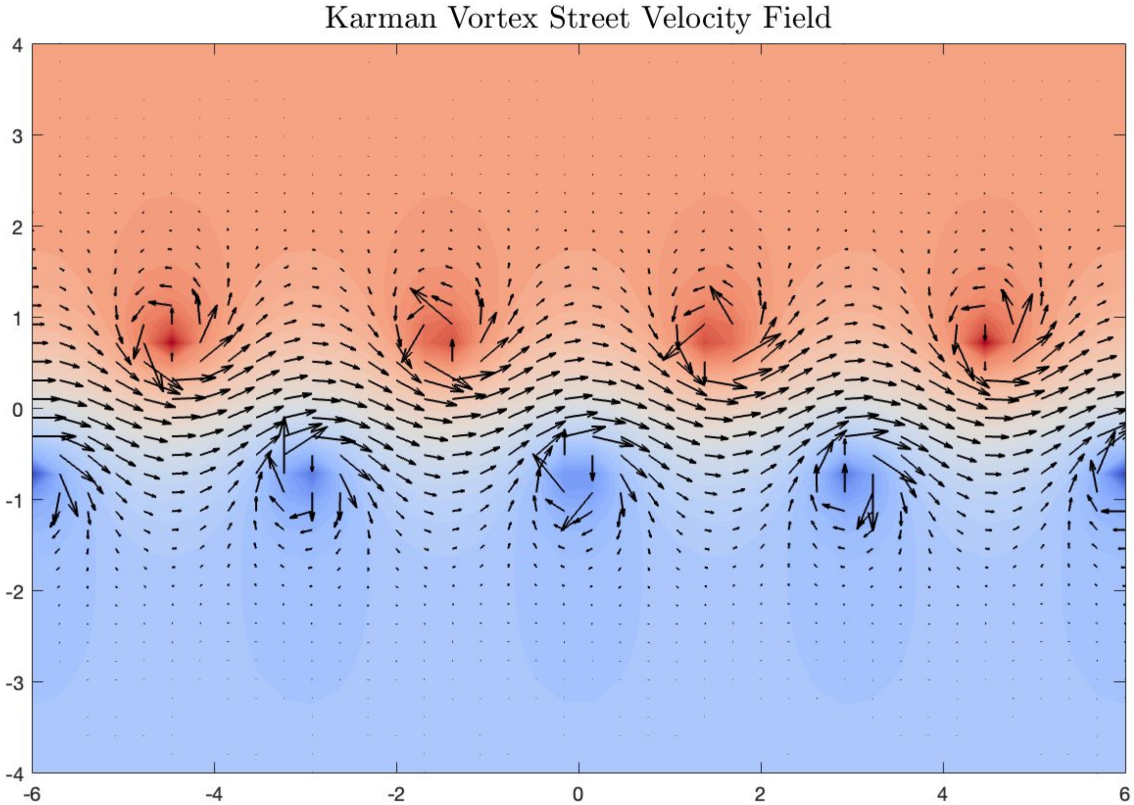
Kármán vortex street model [5]



$$F(z) = \frac{i\Gamma}{2\pi}\left[\log\left(\sin\left[\frac{\pi(z-z_v)}{a}\right]\right) - \log\left(\sin\left[\frac{\pi}{a}\left(z - z_v - \left[\frac{a}{2}+ih\right]\right)\right]\right)\right]$$

$$\frac{\mathrm{d}F(z)}{\mathrm{d}z} = \frac{i\Gamma}{2a}\left[\cot\left(\frac{\pi[z-z_v]}{a}\right) - \cot\left(\frac{\pi}{a}\left[z - z_v - \left(\frac{a}{2}+ih\right)\right]\right)\right] = v_x - iv_y$$

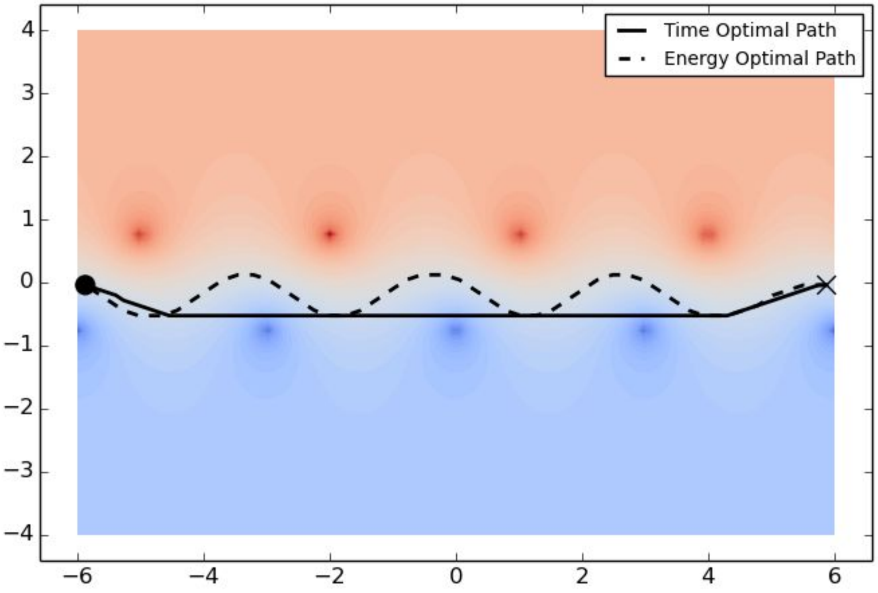
Kármán vortex street model



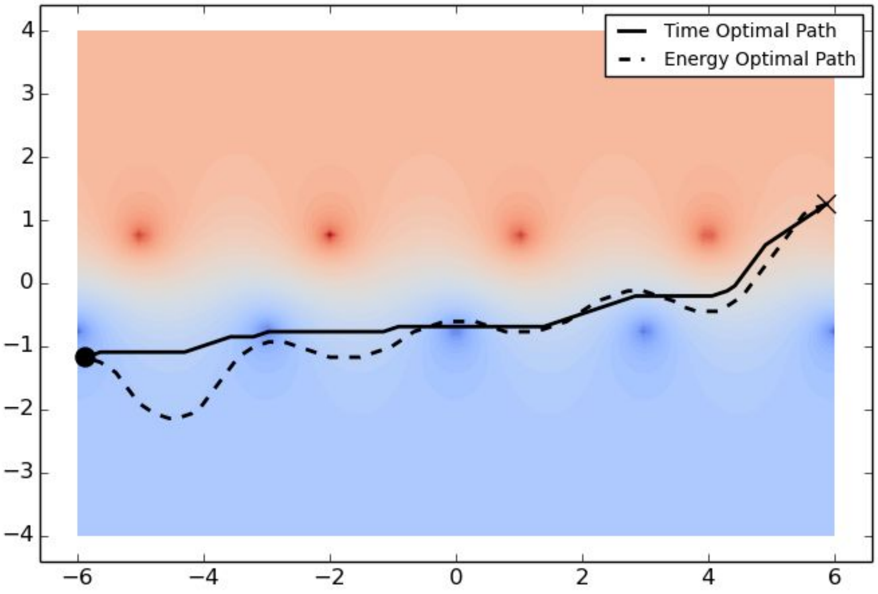
$$z_v = x_v + iy_v = 0 - 0.75i,\ a = 3,\ h = 1.5,\ \Gamma = 3$$

Kármán vortex street

Time-optimization vs. energy-optimization

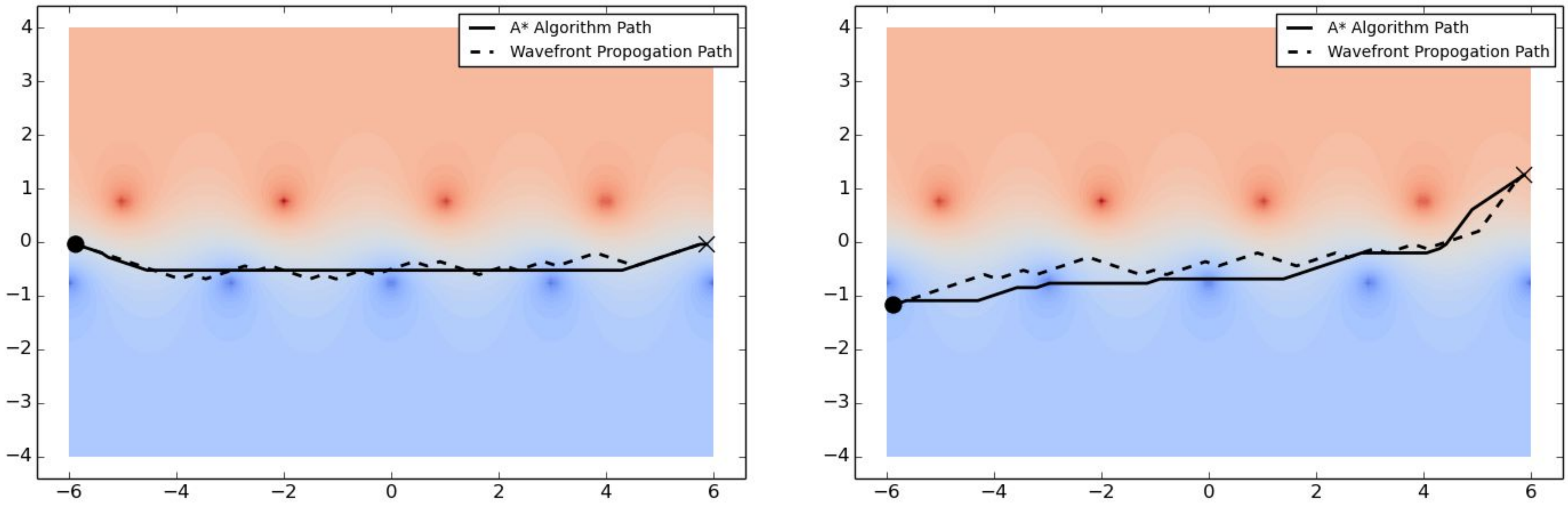


Moving through the stream: Time-optimal to take a direct path, energy-optimal Kármán gaiting pattern



Moving vertically through stream: time-optimal takes direct path, energy-optimal takes advantage of vortices

Kármán vortex street
A vs. wavefront propagation*



A* generates a more direct path, wavefront propagation shows more interaction with the vortices

Future Work

Nonholonomic Constraints [6]

The path planning does not explicitly take the kinematic model into account, but if the grid spacing is large with respect to the minimum turn radius of the swimmer, finding a feasible control law should not be an issue [1]

Standard Dubin's car equations of motion:

$$\begin{pmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{pmatrix} = \begin{pmatrix} V\cos\theta + v_x(t) \\ V\sin\theta + v_y(t) \\ u(t) \end{pmatrix}$$

Define v_{max} and ϵ as:

$$\max_{\mathbf{x}\in W} v(t) = v_{max}$$

$$\epsilon = \frac{v_{max}}{V} < 1$$

Relation between control input and minimum turn radius (without external velocity field):

$$|u(t)| \leq \frac{V}{R_0}$$

The minimum turn radius of the system with an external velocity field becomes:

$$R_0' = R_0(1+\epsilon)^2$$

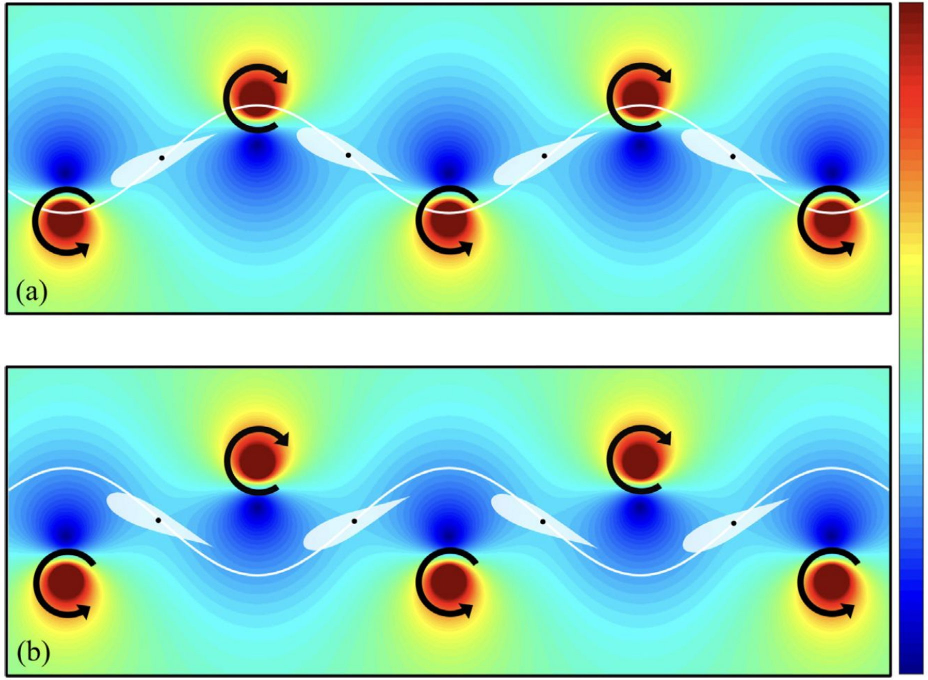
Optimally observable path planning [5]

Observability is the ability of a system to determine its internal state based on a set of outputs

Pressure sensors along sides of swimmer make up the artificial lateral line: used to estimate the state of the vortex street

A scalar measure of observability comes from the reciprocal of the minimum singular value of the observability Gramian

A path should take into account both the accuracy of the vortex street estimation and energy efficiency



(a) Optimally observable path
(b) Energy optimal path: typical slaloming behavior [5]

Time-optimal and energy-optimal path planning in two-dimensional potential flows

Rose Gebhardt

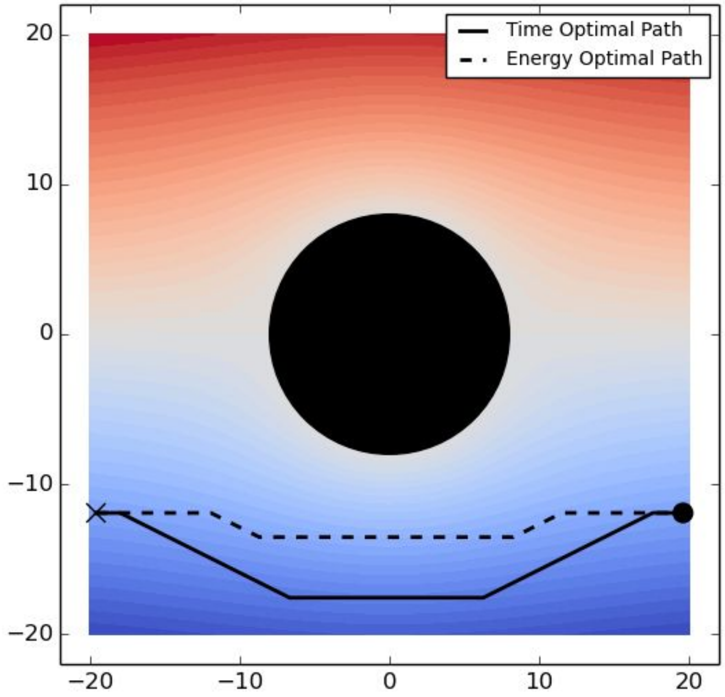
Thursday May 5, 2022

ENAE788V - Spring 2022

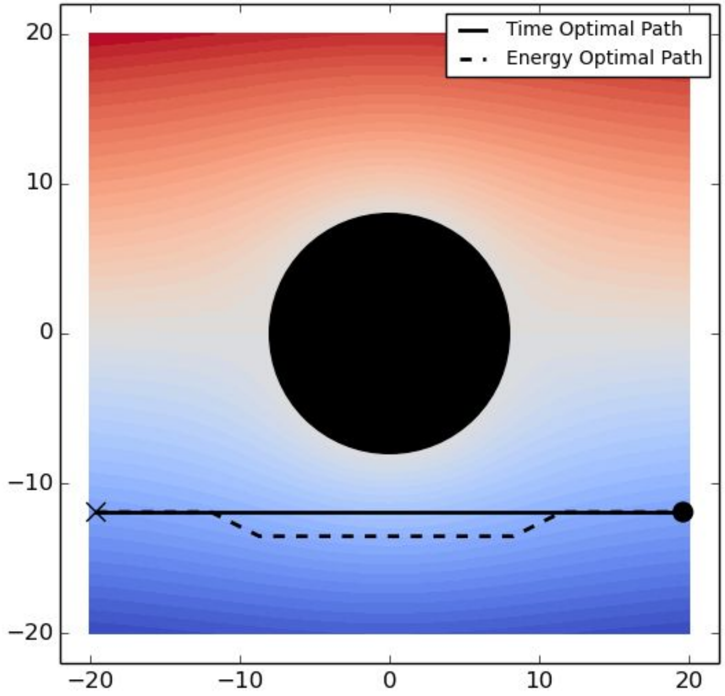
Appendices

Cylinder in freestream: against direction of flow

Time-optimization vs. energy-optimization



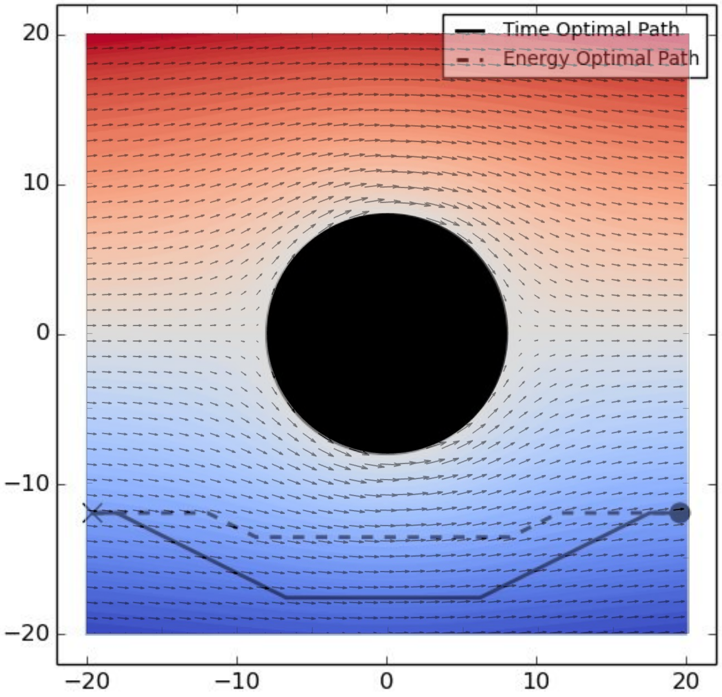
V_{max} = 2.0: It is time-optimal and energy-optimal to take a less direct path to avoid vertical velocity field



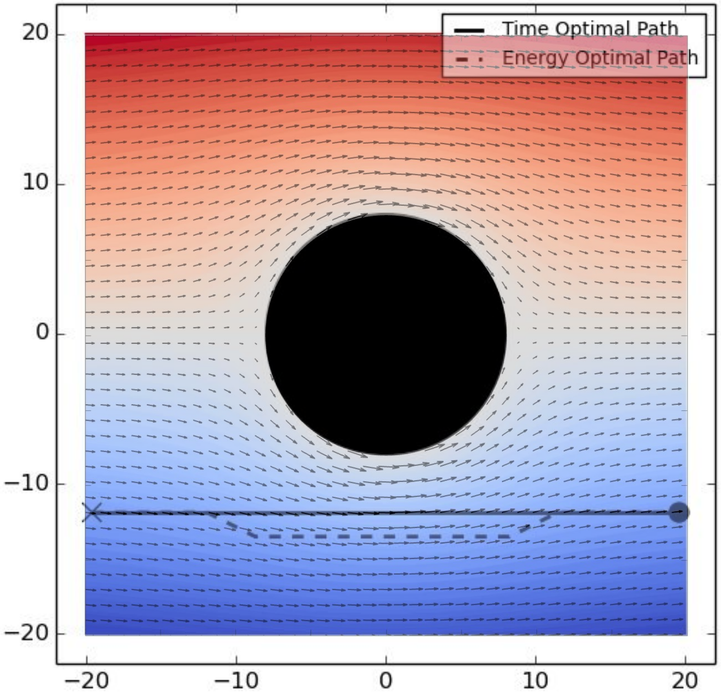
V_{max} = 4.0: It is time-optimal to take the most direct path, it is energy-optimal take a path with less vertical motion

Cylinder in freestream: against direction of flow (velocity overlaid)

Time-optimization vs. energy-optimization

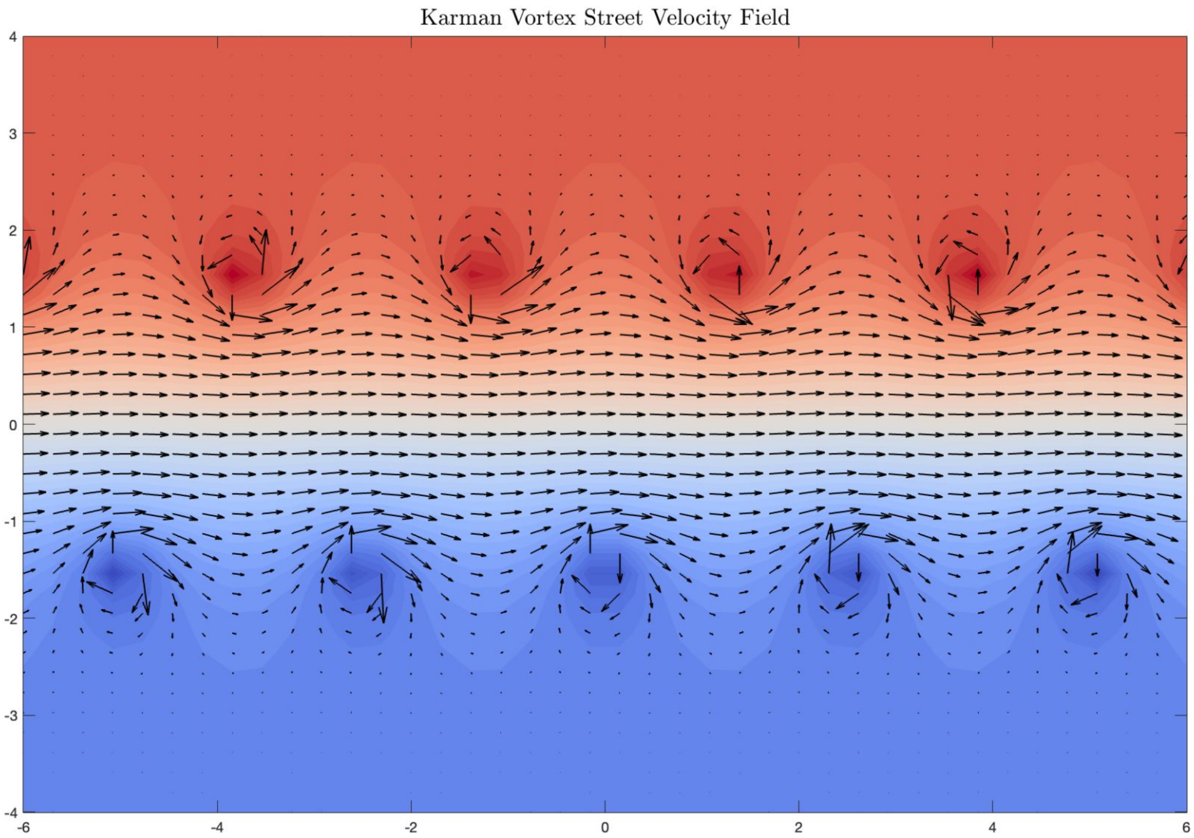


V_{max} = 2.0: It is time-optimal and energy-optimal to take a less direct path to avoid vertical velocity field



V_{max} = 4.0: It is time-optimal to take the most direct path, it is energy-optimal take a path with less vertical motion

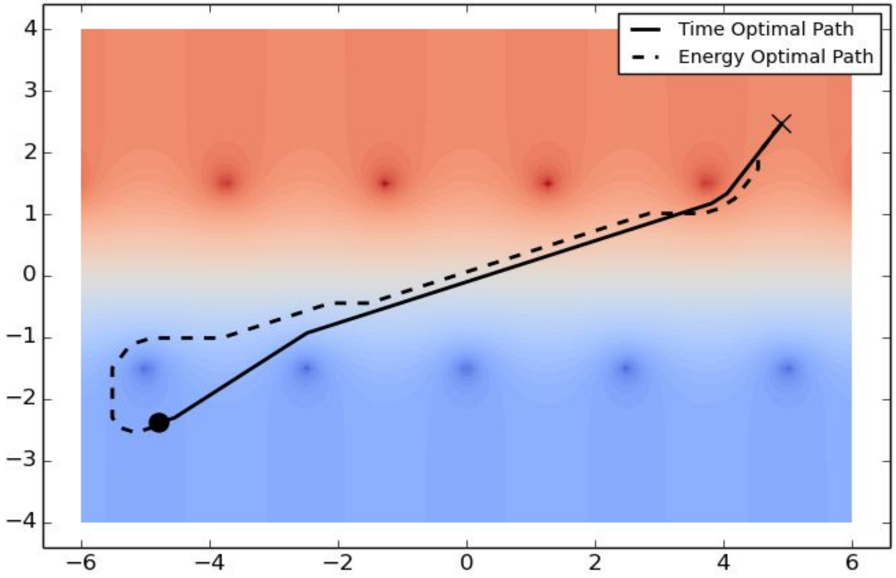
Kármán vortex street model



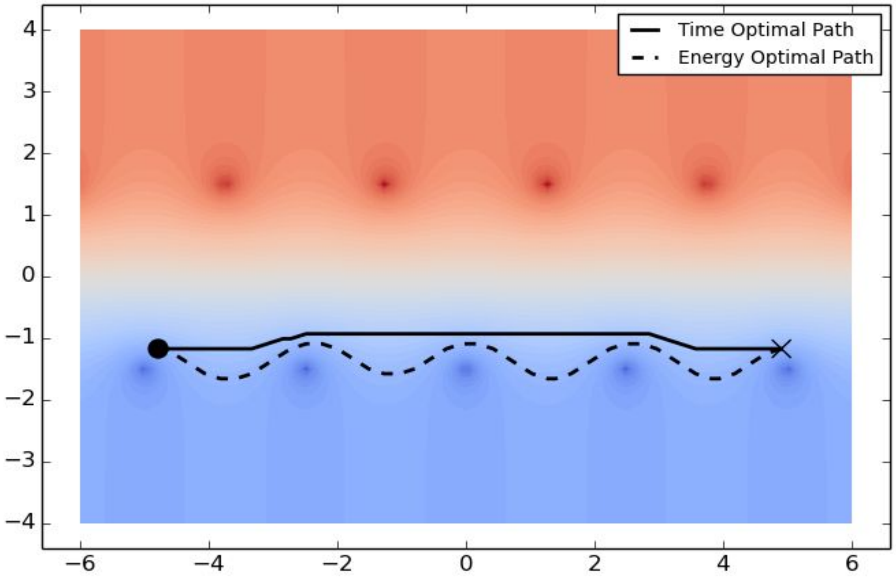
$$z_v = x_v + iy_v = 0 - 1.5i,\ a = 2.5,\ h = 3,\ \Gamma = 3$$

Kármán vortex street: in direction of flow

Time-optimization vs. energy-optimization



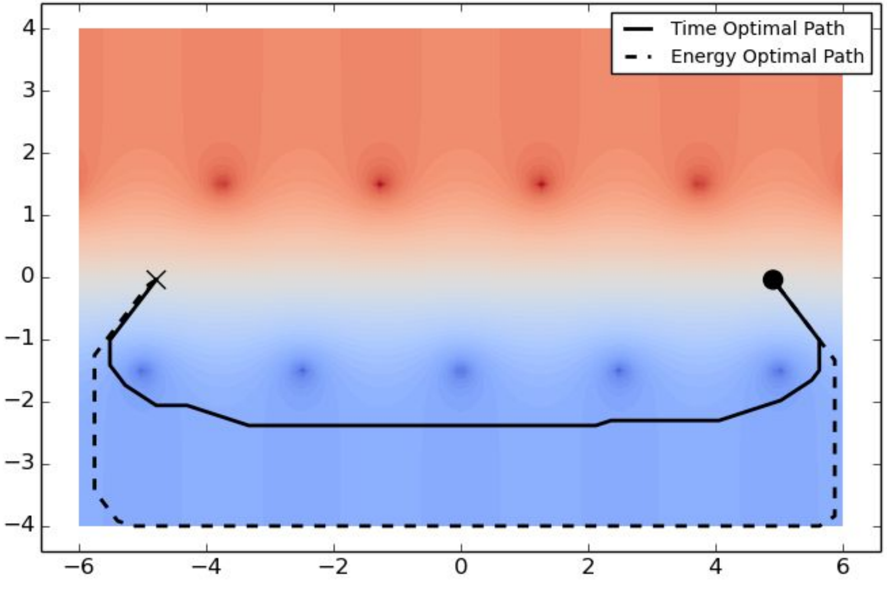
Moving through the stream: energy-optimal to follow streamlines, time-optimal to take a more direct path



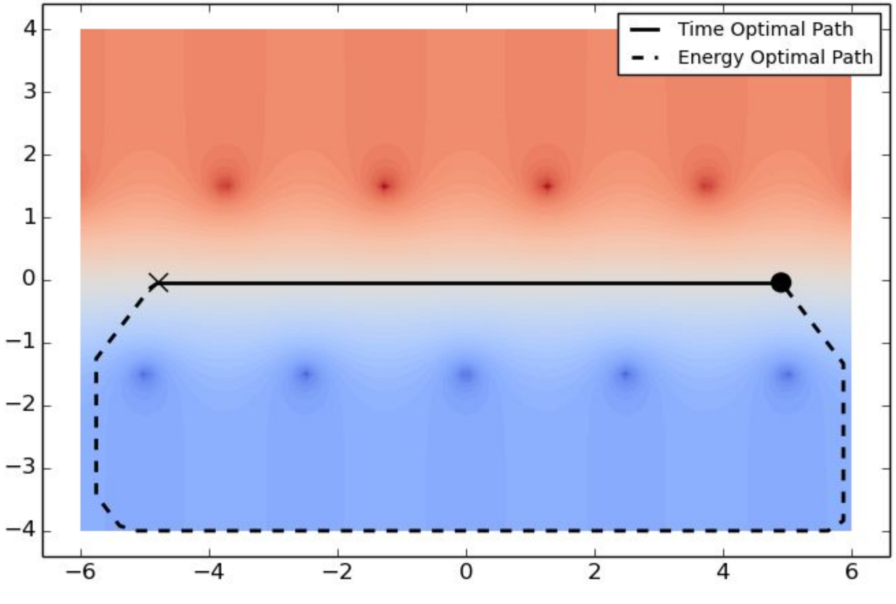
Moving below stream: energy-optimal to oscillate, time-optimal to take a more direct path

Kármán vortex street: against direction of flow

Time-optimization vs. energy-optimization



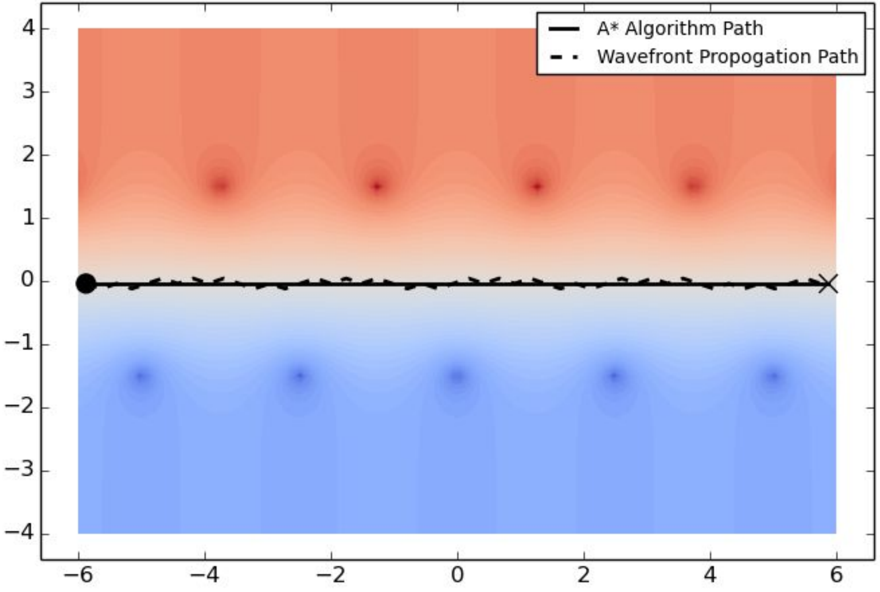
V_{max} = 1.0: It is time-optimal and energy-optimal to move out the stream created



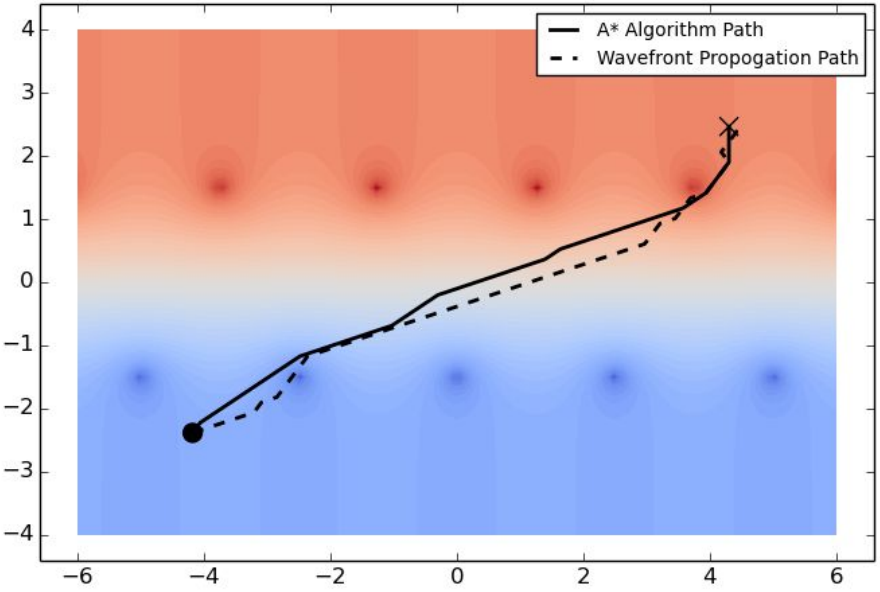
V_{max} = 8.0: It is time-optimal to move against the stream created but energy-optimal to move out of the stream

Kármán vortex street: in direction of flow

A vs. wavefront propagation*



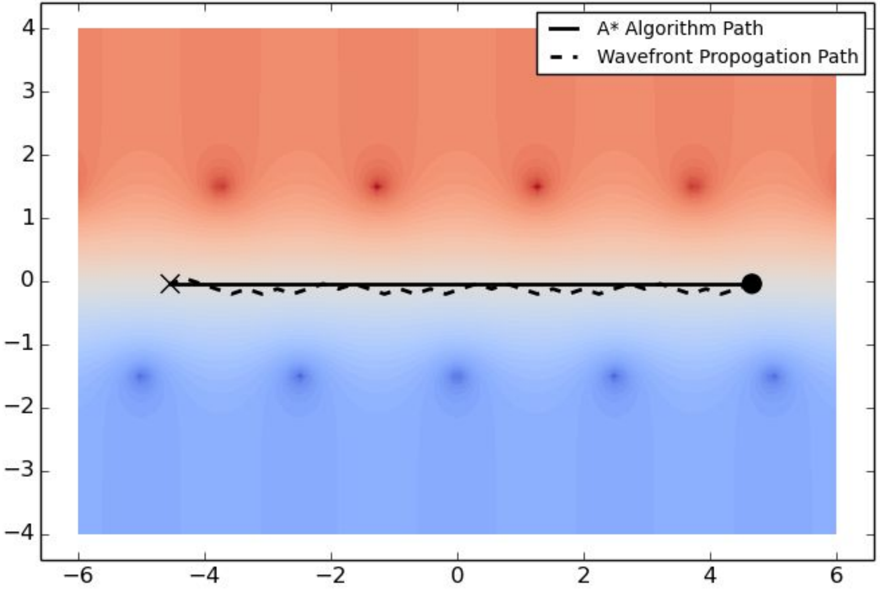
Similar behavior, but wavefront propagation shows deviation from direct path



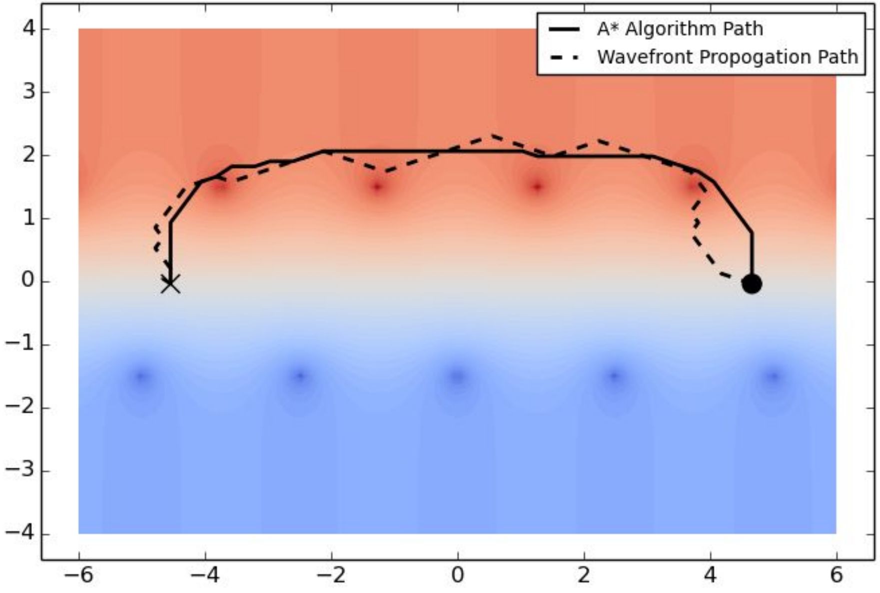
Similar behavior, wavefront propagation path gets closer to the vortices

Kármán vortex street: against direction of flow

A vs. wavefront propagation*

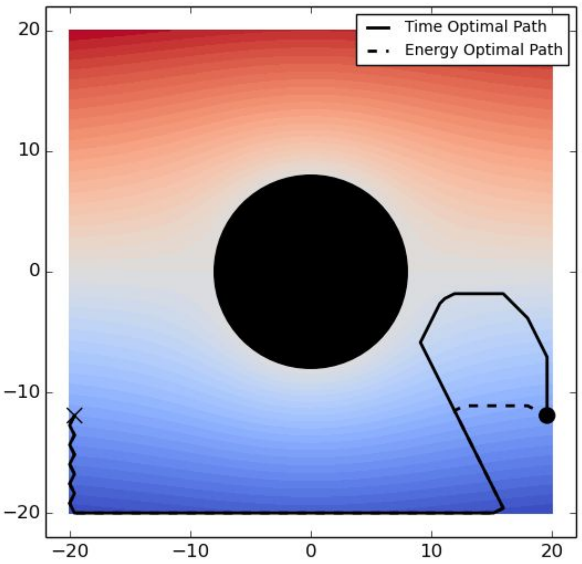


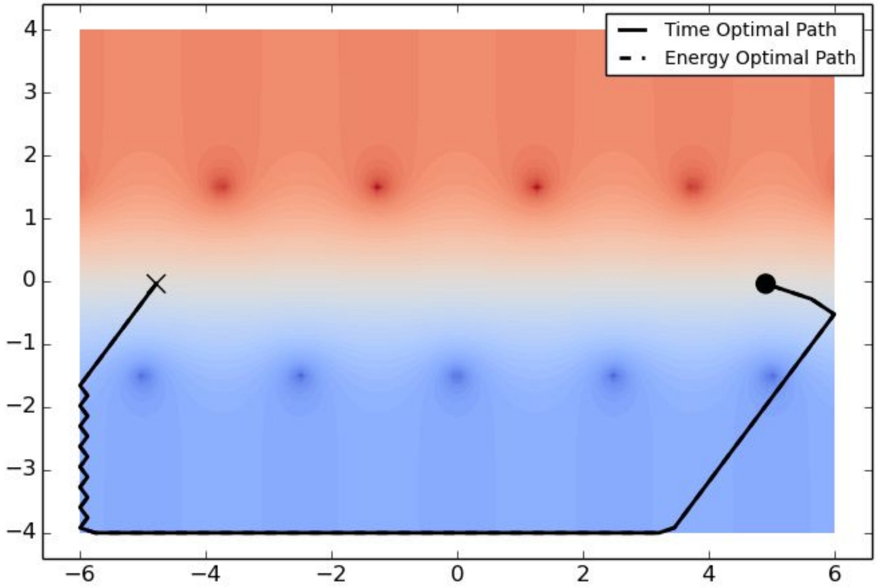
V_{max} = 8: Similar behavior, but wavefront propagation shows deviation from direct path



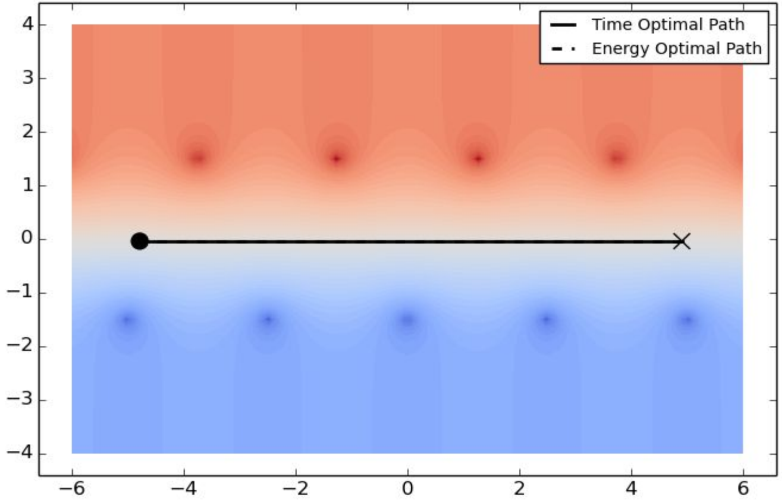
V_{max} = 2: Similar behavior, wavefront propagation path gets closer to the vortices

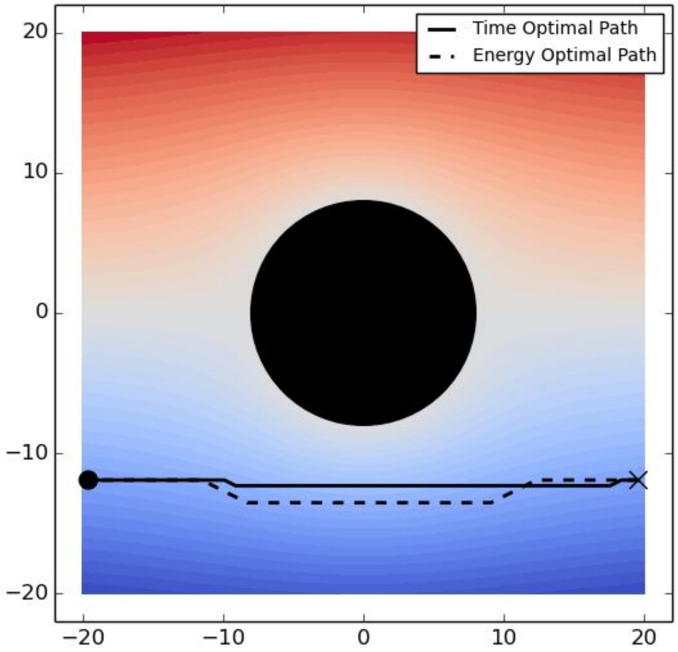
Failure Cases





Boring Cases





Formal Problem Definitions

Suppose we are given a discretized workspace $\mathbb{W} \subset \mathbb{R}^2$ with associated external velocity field components $v_x(\mathbf{x})$ and $v_y(\mathbf{x})$, a start location $\mathbf{x}_{start} \in \mathbb{W}$, and a goal location $\mathbf{x}_{goal} \in \mathbb{W}$. Any path Γ in the workspace is subject to the constraints $\Gamma(t = 0) = \mathbf{x}_{start}$, $\Gamma(t = T) = \mathbf{x}_{goal}$, and $V \leq V_{max}$.

Problem Definition 1 [Time Optimization]: Given a differential time cost function dt, find the path $\mathbf{\Gamma}$* which minimizes the functional $\int_\Gamma dt$ subject to the constraints specified.

Problem Definition 2 [Energy Optimization]: Given a differential energy cost function dW, find the path $\mathbf{\Gamma}$* which minimizes the functional $\int_\Gamma dW$ subject to the constraints specified.