Wind Energy Harvesting with a Kite

Antonin Bavoil[1], Jean-Baptiste Caillau[1], Lamberto Dell'Elce[2], Alain Nême[3] , Jean-Baptiste Leroux[3]

1: Université Côte d'Azur, CNRS, Inria, LJAD
2: Inria
3: ENSTA Bretagne, iRDL

Julia and Optimization Days 2024

Toulouse, 29 October 2024

Kite Electrical Energy Production (KEEP)

- KEEP is born as a follow up of Beyond the Sea®.
- Idea: generate (on land) electricity with a kite.
- 10x less material than a wind turbine, soft.
- Portable source of power in remote areas.

→ Islands, military.

https://www.dailymotion.com/video/x5fwyox

Kite Electrical Energy Production (KEEP)

- Approximately the same power as a wind turbine.
- Currently: 10% of the production used for control.
- Previous works:
  - U. Ahrens, M. Diehl, R. Schmehl. *Airborne Wind Energy.* Springer, 2013,
  - U. Fechner et al. *Dynamic Model of a Pumping Kite Power System*, Pergamon, 2015.
  - Startup Kitepower, by J. Peschel and R. Schmehl, since 2016

https://www.iksurfmag.com/reviews/kites/north-kiteboarding-evo-6m-2013/

→ *Can we generate a good amount of electricity while requiring close to no control?*

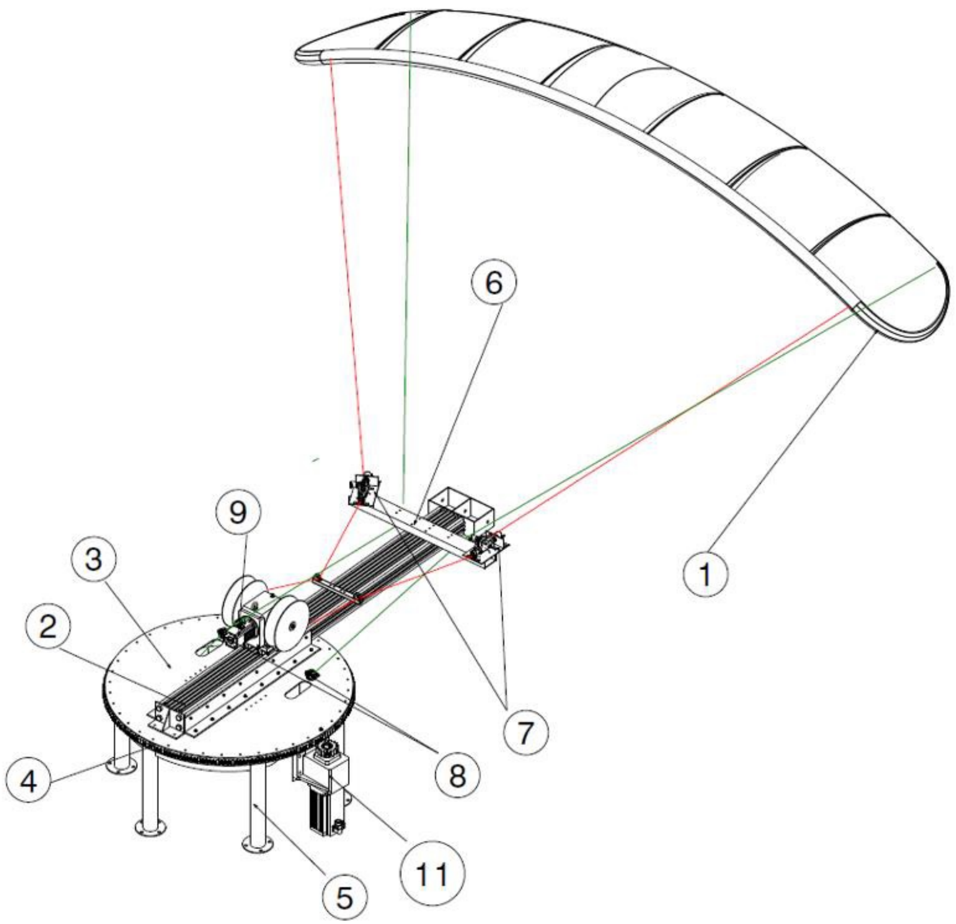
Wind Energy Harvesting with a Kite

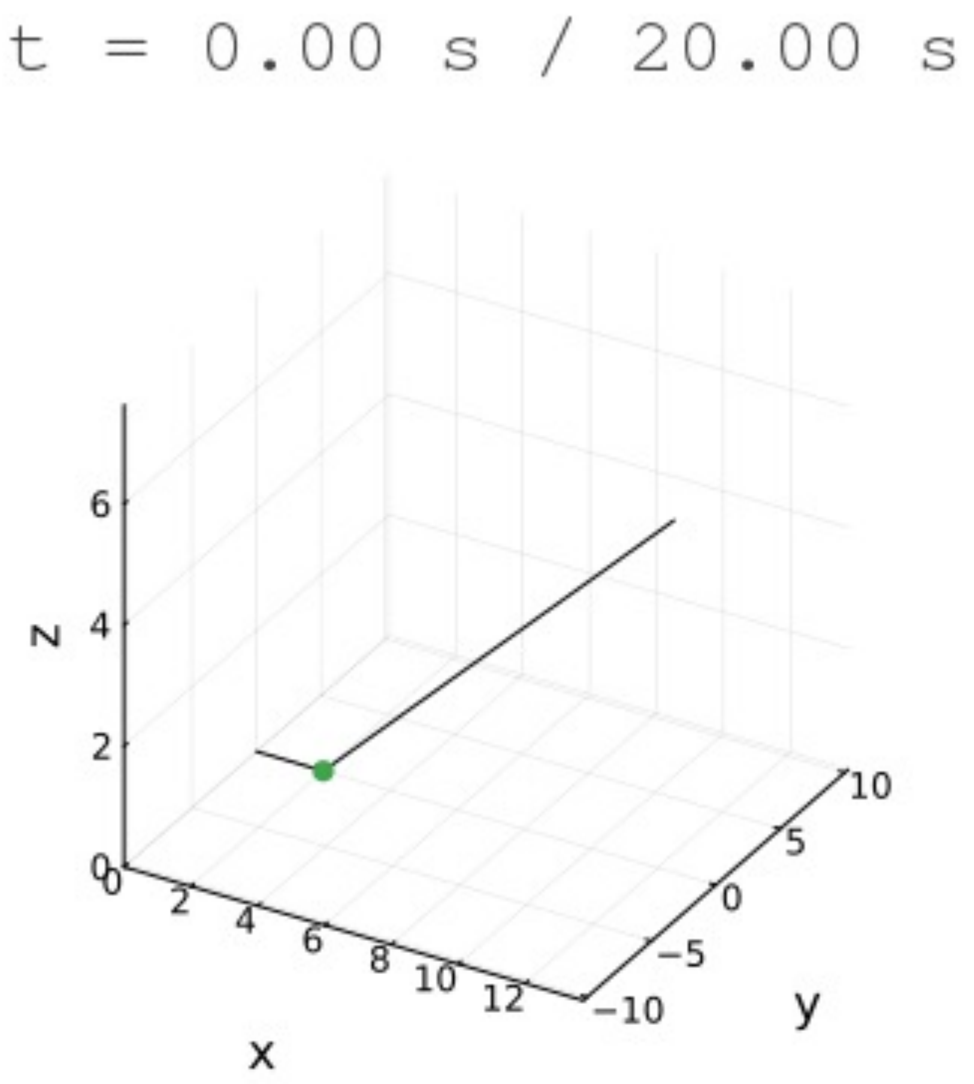
I. Modelling

II. Numerical Results

III. Optimization problem

Modelling





→ The kite does eights in the sky to swing the arm left to right as much as possible, without tangling the lines.

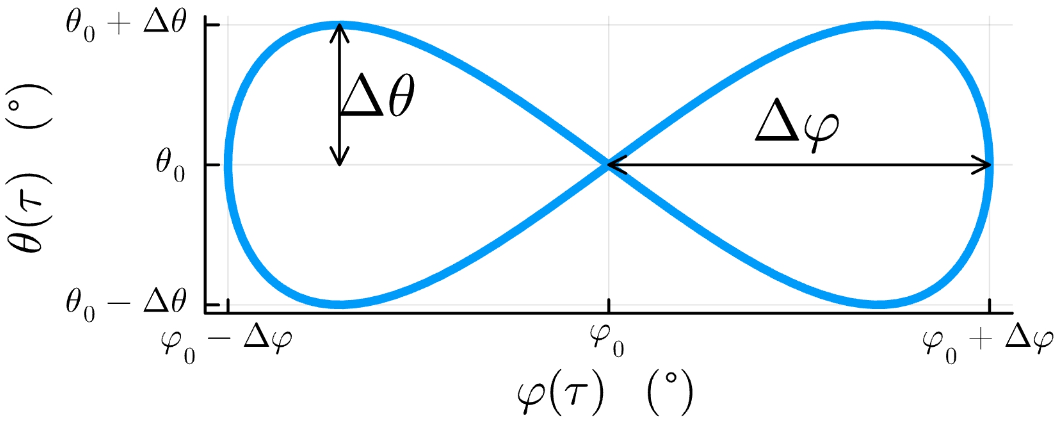
Modelling

Replacing the control with an algebraic constraint

- As a preliminary study, we replace the control with a geometric constraint.

→ The kite will stay on an 8-based cone centered on O.

Base of the cone



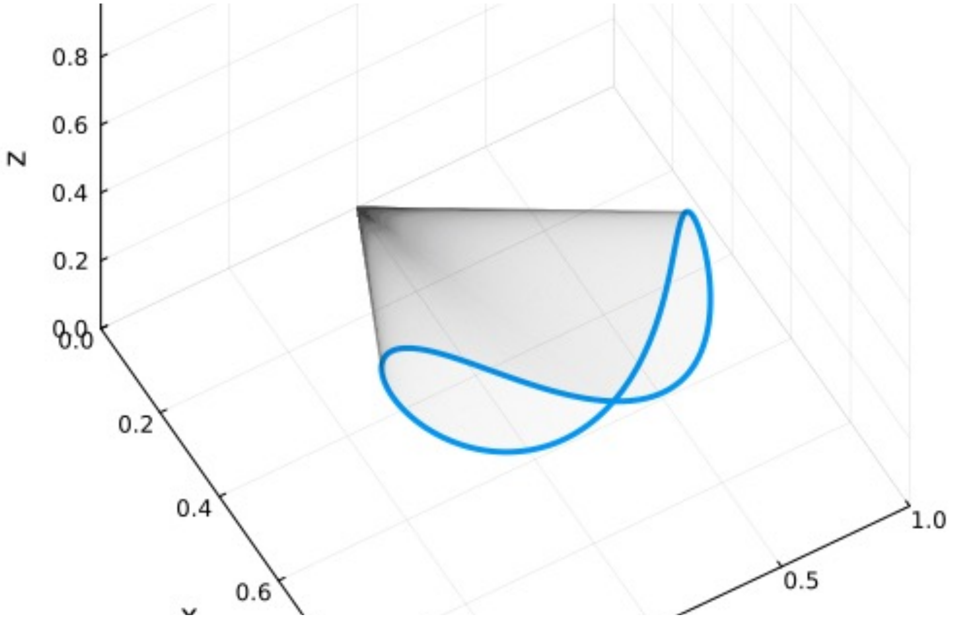
- Spherical coordinates in the inertial frame $Oxyz$:

Set of (R, θ, φ) s.t.

$$\theta = \theta_0 + \Delta\theta \sin(2\tau)$$

$$\varphi = \varphi_0 + \Delta\varphi \sin(\tau)$$

$$R = R(\alpha, \tau).$$

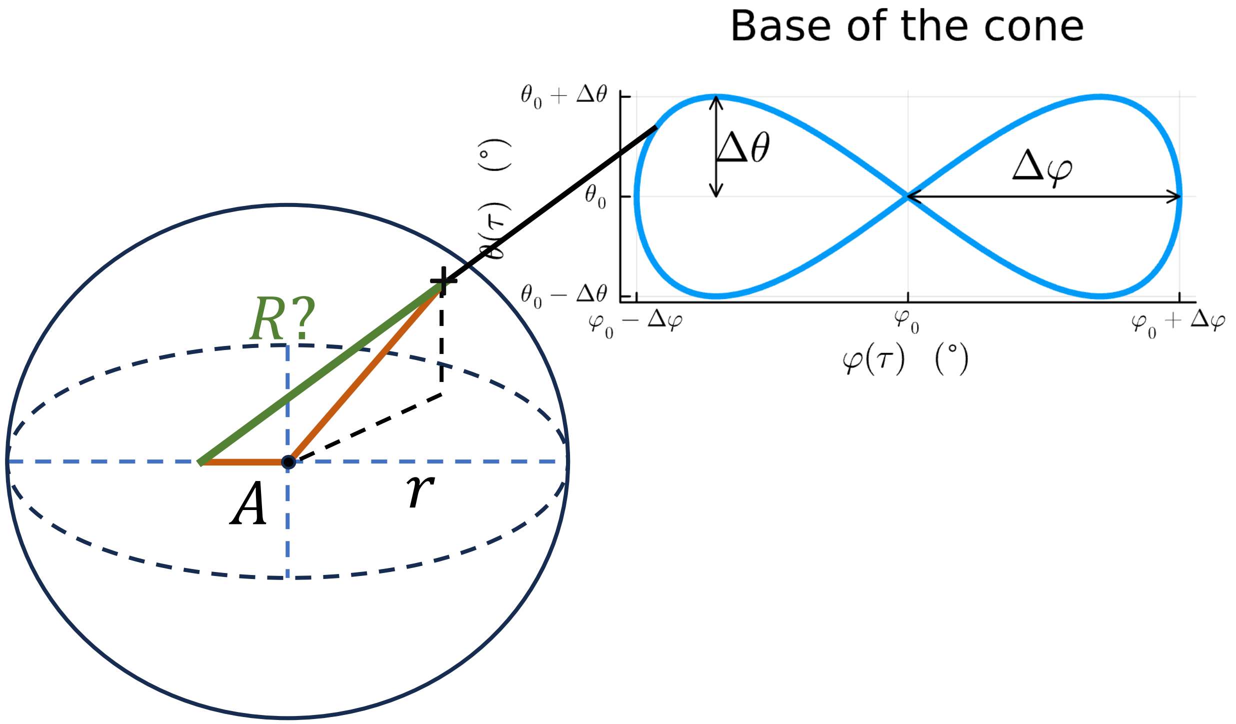


→ **Like a railway track guiding a train.**

Modelling

Coordinates

- State of the system:
  - α: 1D angle between the arm and the x-axis,
  - (R, τ): 2D position of the kite on the 8-based cone.



We can do better: get R through the intersection of a sphere (A, r) with a line:

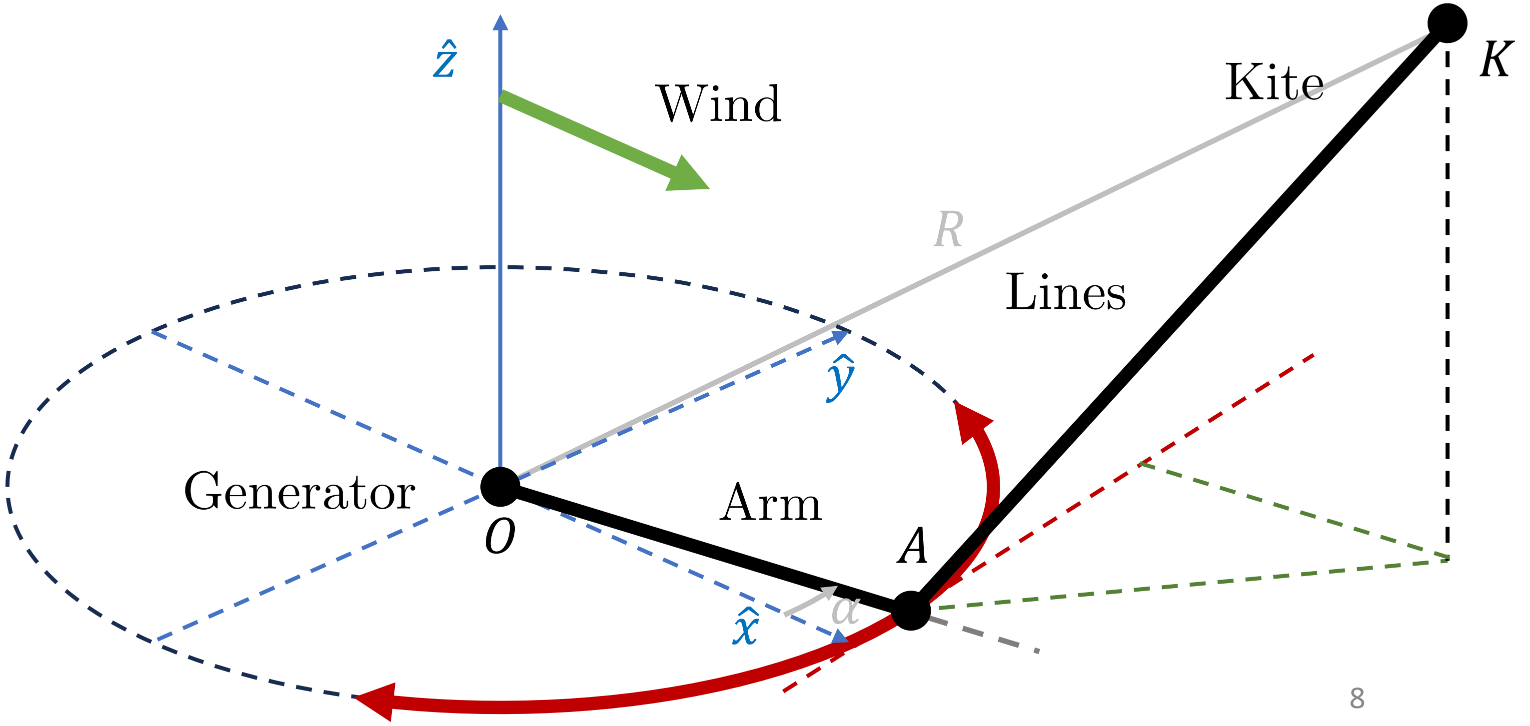
$$R(\alpha, \tau) = \hat{\tau} \cdot \overrightarrow{\mathrm{OA}} + \sqrt{\left(\hat{\tau} \cdot \overrightarrow{\mathrm{OA}}\right)^2 + r^2 - \mathrm{OA}^2}.$$

→ (α, τ) is enough to determine the state of the system.

Modelling

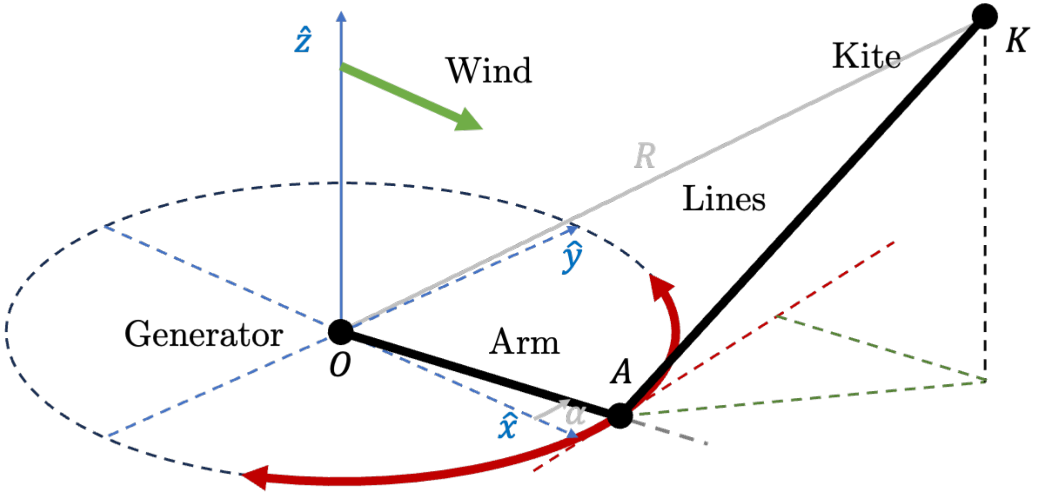
Compute the dynamics

- Dynamics: conservation of linear & angular momentum.
- All forces apply at point K (the kite):
  - Gravity,
  - Aerodynamical forces,
  - Line tension,
  - Fictitious cone force.



Modelling

Gravity

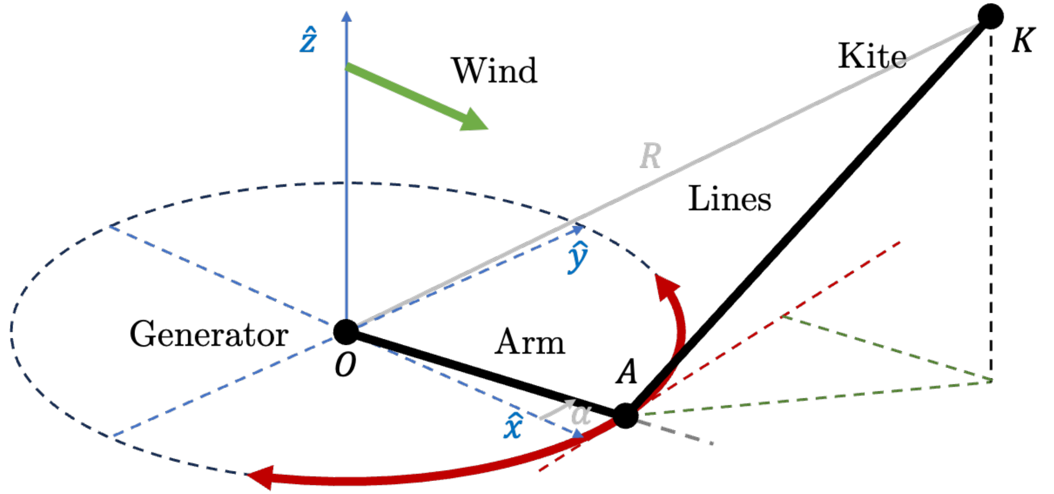


- Gravity force = weight of the kite + weight of the lines:

$$\overrightarrow{F_{\text{grav}}} = -g(m_{\text{kite}} + m_{\text{lines}})\hat{z}.$$

Modelling

Aerodynamical forces



- Apparent wind: $\overrightarrow{w_{\text{app}}} = \vec{v} - \vec{w}$ (velocity – wind).
- Aerodynamical force on the kite:

$$\overrightarrow{F_{\text{aero,kite}}} = -\frac{1}{2} S \rho_{\text{air}} \left\| \overrightarrow{w_{\text{app}}} \right\|^2 (C_L \hat{z}_w + C_D \hat{x}_w).$$

- Aerodynamical force on the lines, applied at point K:

$$\overrightarrow{F_{\text{aero,lines}}} = -\frac{1}{2} \frac{n b_l \, r \, d_l}{3} \rho_{\text{air}} \left\| \overrightarrow{w_{\text{app}}} - (\hat{r} \cdot \vec{v}) \hat{r} \right\|^2 C_{Dl} \, \hat{e}_w.$$

$$\overrightarrow{F_{\text{aero}}} = \overrightarrow{F_{\text{aero,kite}}} + \overrightarrow{F_{\text{aero,lines}}}.$$

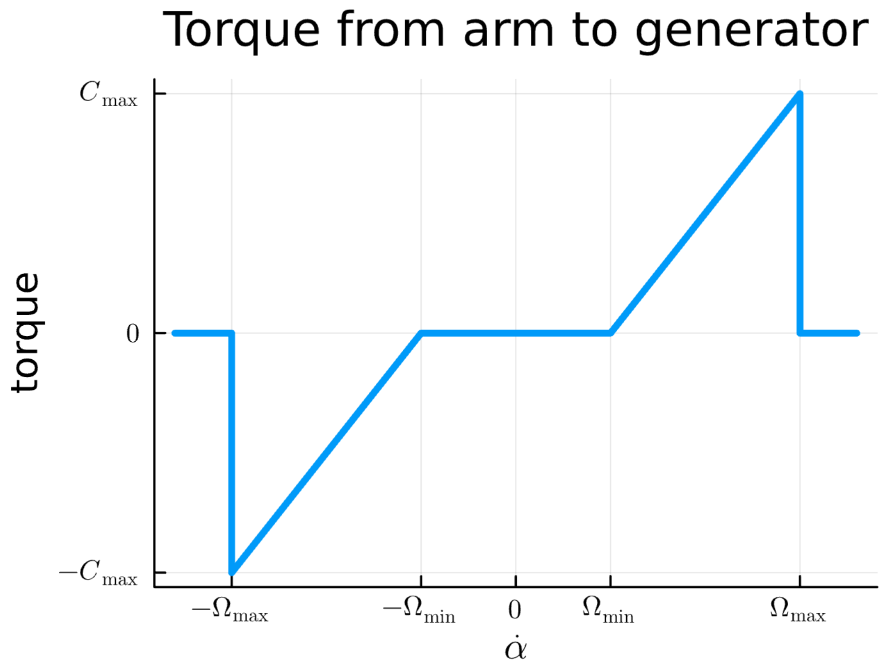
Modelling

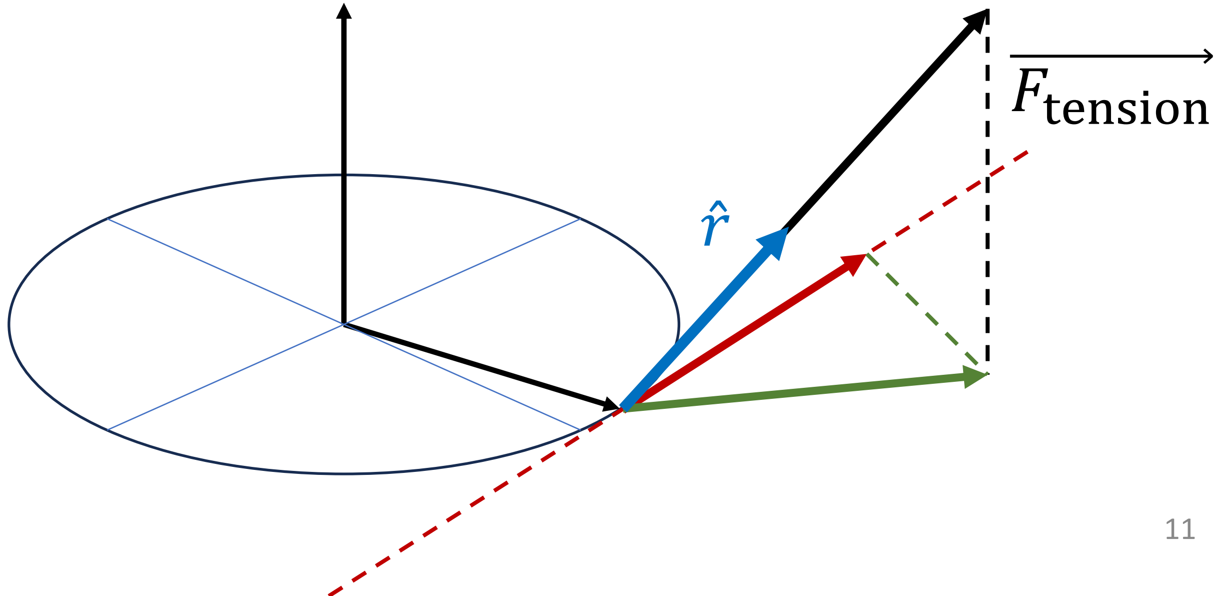
Line tension & conservation of angular momentum

- Conservation of angular momentum of the arm gives:

$$I_{\mathrm{eq}}\ddot{\alpha} = -\mathrm{torque}(\dot{\alpha}) + F_{\mathrm{tension}}\,\hat{r}\cdot\left(\hat{z}\times\overrightarrow{\mathrm{OA}}\right).$$

- Hence $\overrightarrow{F_{\mathrm{tension}}} = \frac{I_{\mathrm{eq}}\ddot{\alpha}-\mathrm{torque}(\dot{\alpha})}{\hat{r}\cdot(\hat{z}\times\overrightarrow{\mathrm{OA}})}\hat{r}$, where $\ddot{\alpha}$ is obtained by solving the dynamics cf. next slide.





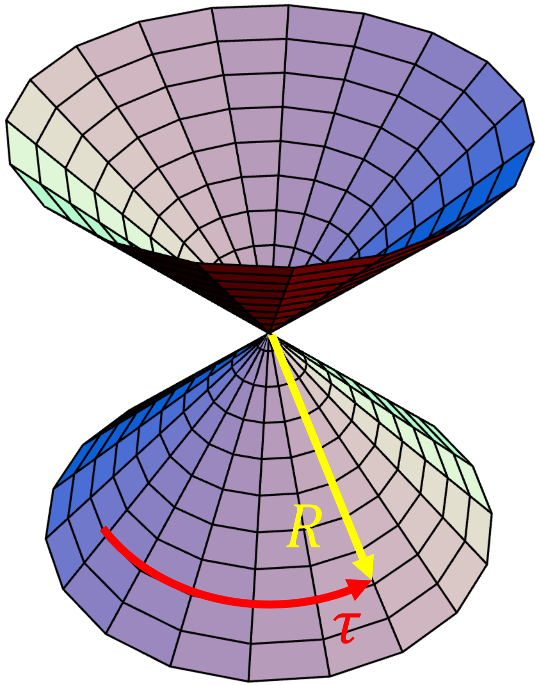
Modelling

Replacing the control with an algebraic constraint

- How? Add a fictitious force $\overrightarrow{F_{\mathrm{cone}}}$ that would result from a control.
- No friction with the cone:

$$\overrightarrow{F_{\mathrm{cone}}} \cdot \frac{\partial \overrightarrow{OK}}{\partial R} = 0 \text{ and } \overrightarrow{F_{\mathrm{cone}}} \cdot \frac{\partial \overrightarrow{OK}}{\partial \tau} = 0.$$

Circle-based cone



https://mathworld.wolfram.com/Cone.html

Modelling

Computing the explicit dynamics

- Recall that $\overrightarrow{F_{\text{cone}}} \cdot \frac{\partial \overrightarrow{OK}}{\partial R} = 0$ and $\overrightarrow{F_{\text{cone}}} \cdot \frac{\partial \overrightarrow{OK}}{\partial \tau} = 0$.

- Conservation of linear momentum gives: $m \underset{\boxed{(\ddot{\alpha}, \ddot{\tau})}}{\underset{\uparrow}{\frac{d\vec{v}}{dt}}} = \overrightarrow{F_{\text{cone}}} + \overrightarrow{F_{\text{grav}}} + \overrightarrow{F_{\text{aero}}} + \underset{\boxed{\ddot{\alpha}}}{\underset{\uparrow}{\overrightarrow{F_{\text{tension}}}}}$.

- Hence $\overrightarrow{F_{\text{cone}}} = m \frac{d\vec{v}}{dt} - \overrightarrow{F_{\text{grav}}} - \overrightarrow{F_{\text{aero}}} - \overrightarrow{F_{\text{tension}}}$. Linear in $\ddot{\mathbf{u}} = (\ddot{\alpha}, \ddot{\tau}) \in \mathbb{R}^2$.

- Solve $A(\mathbf{u})\, \ddot{\mathbf{u}} = b(\mathbf{u}, \dot{\mathbf{u}})$ for $\ddot{\mathbf{u}}$.

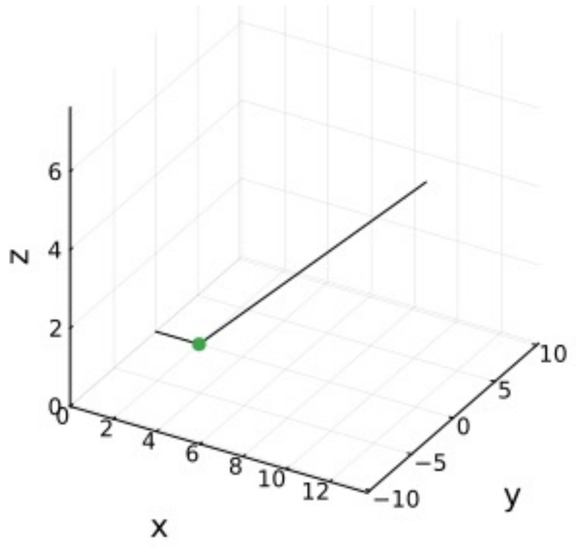
First Numerical Results

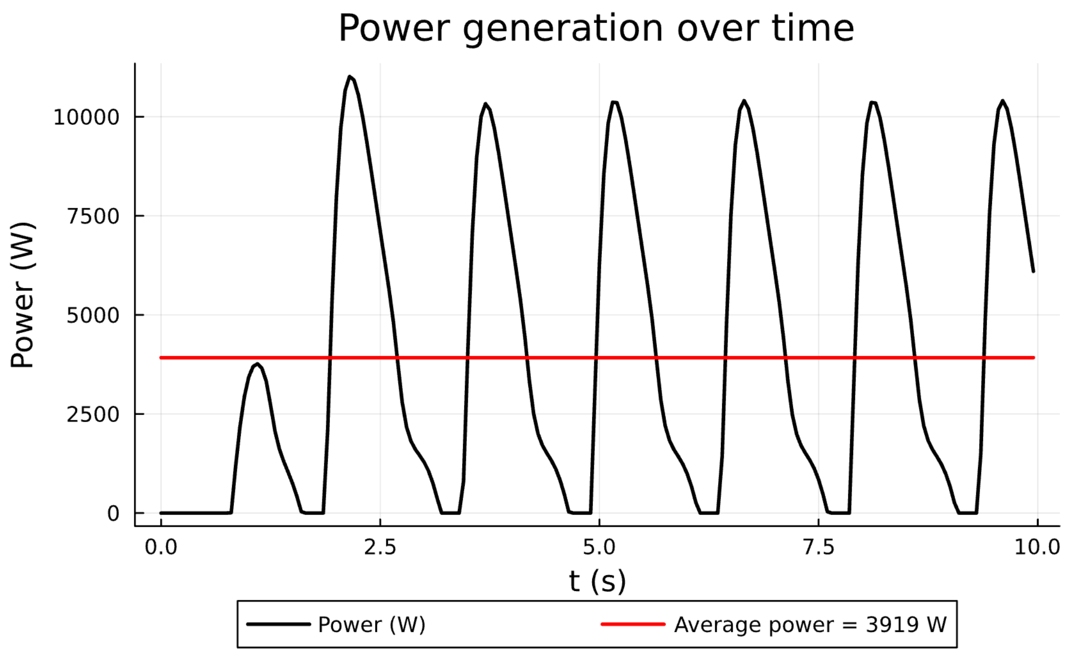
Solving the dynamics

- Assemble $A(\mathbf{u})$ and $b(\mathbf{u}, \dot{\mathbf{u}})$ using forward auto. diff. for exact, fast & maintainable derivatives:

```
julia> kite_speed = jvp(u -> OK(u, params), u, du)
```

- Runge-Kutta 5(4) (Tsitouras 2011):





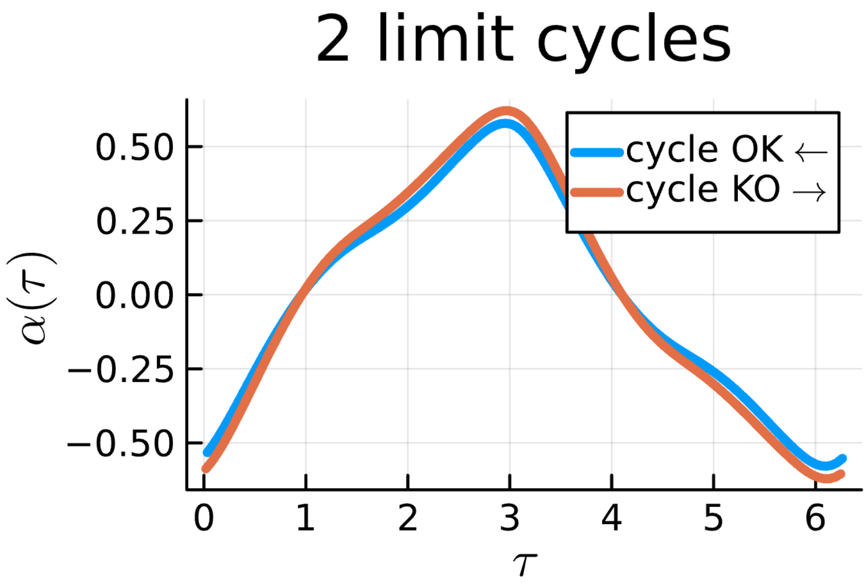
$\rightarrow$ Cyclic behavior: toward a limit cycle...

First Numerical Results

Limit Cycle

- Define a Poincaré section at $\tau \equiv 0\ [2\pi]$ where the cycle begins (arbitrary).
- Dense ODE output & continuous callback allow us to save the state every time it crosses one of these hyperplanes:

Iteration	$\Delta t = t(\tau = 2\pi) - t(\tau = 0)$ (seconds)
1	2.7857519406370397
2	3.0933710067285087
3	3.0930984474625856
…	…
7	3.0930980569211132
8	3.0930980569206064



→ 2 limit cycles, interested in the one with $\dot{\tau} < 0$ that does not tangle the lines.

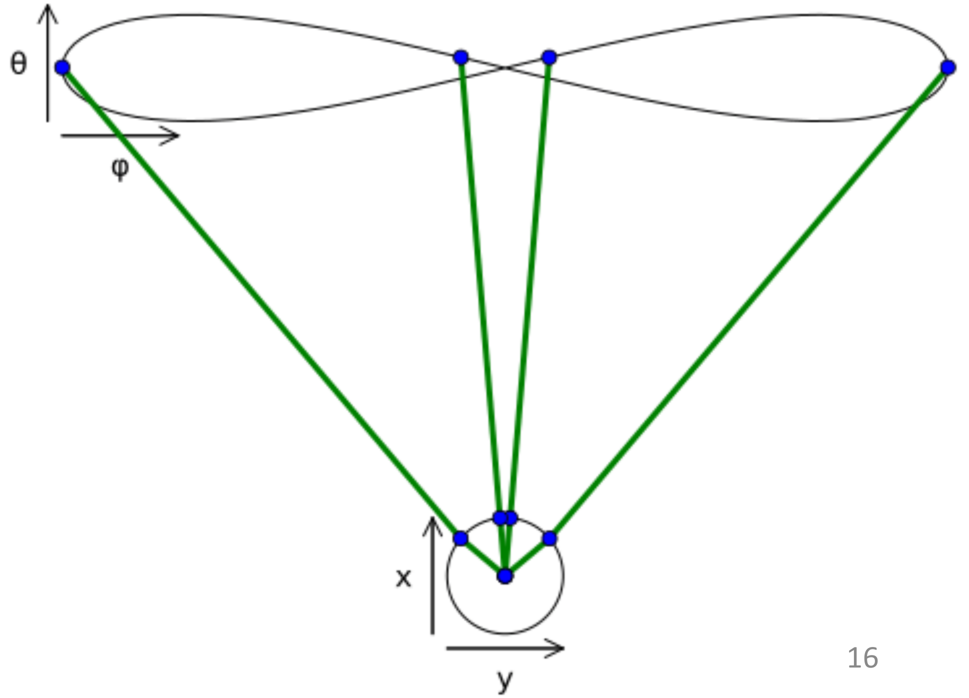
First Numerical Results

Equilibriums

Any initial condition could fall into either limit cycles...

- Set $\dot{\mathbf{u}} = 0$ and find $\mathbf{u}$ such that $\ddot{\mathbf{u}} = 0$,
- The 2 limit cycles, $\dot{\tau} > 0$ and $\dot{\tau} < 0$, are separated by equilibriums,

8 equilibriums



- From an equilibrium, set $\dot{\tau} < 0$ to fall in the good cycle,
- Integrate for 8 cycles → guaranted limit cycle.

Optimization

Problem: Maximize the Average Power over a limit cycle

- Augment the state with W, the total work of the generator :

$$\dot{W} = \dot{\alpha}\ \text{torque}(\dot{\alpha}) \quad (= P(t))$$

→ Compute the energy generated while solving the ODE instead of *a posteriori*

Optimization problem:

- Maximize $W(t_f)\ /\ tf$
- Over the design parameters: line length $r \in [r_{\min}, r_{\max}]$ and moment of inertia of the arm $I \in [I_{\min}, I_{\max}]$
- Such that $\alpha(t_f) = \alpha_0,\ \tau(0) = 0,\ \tau(t_f) = -2\pi,\ \dot{\alpha}(t_f) = \dot{\alpha}_0,\ \ \dot{\tau}(t_f) = \dot{\tau}_0$

Where t_f and the initial condition $(\alpha_0, 0, \dot{\alpha}_0, \dot{\tau}_0)$ are to be determined for each (r, I)

And $\alpha(t_f), \tau(t_f), \dot{\alpha}(t_f), \dot{\tau}(t_f)$ and $W(t_f)$ are obtained by solving the ODE.

Optimization

Problem: Maximize the Average Power over a limit cycle

Most implicit

```
optvar = (r, I)

function objective(r, I)
    determine tf, x0 such that
they produce a limit cycle
    integrate the ODE
    return W(tf)/tf
end
```

- A function gives the limit cycle for `(r, I)`
- An ODE solver gives `W(tf)`
- No nonlinear constraints

More explicit

```
optvar = (r, I, tf, α0, dα0,
dτ0, Wf)

function objective(optvar)
    return optvar.W/tf
End

function nlconstraints(optvar)
    α(tf) – α0
    τf – 2π
    dα(tf) – dα0
    dτ(tf) – dτ0
    W(tf) – Wf
end
```

- An ODE solver gives `_(tf)`
- 5 nonlinear constraints

Most explicit

```
optvar = (r, I, tf, α0, dα0, dτ0,
α[2:N+1], τ[2:N+1], dα[2:N+1],
dτ[2:N+1], W[2:N+1])

function objective(optvar)
    return W[N+1]/tf
end

function nlconstraints(optvar)
    x(i+1) - xi - h*f(x(i)), i=1:N
    τ[N+1] – 2π
    α[N+1] - α0
    dα[N+1] - dα0
    dτ[N+1] - dτ0
end
```

- Implement own **fixed-step** ODE solver
- N+5 nonlinear constraints

Optimization

Summary

- Define a dynamics `ForwardDiff.jl`, `ComponenArrays.jl`
- Solve the ODE problem and realize it converges toward a limit cycle `OrdinaryDiffEq.jl`, `Plots.jl`
- Find the equilibriums of the system to systematically compute the correct limit cycle `DiffEqCallbacks.jl`, `NonlinearSolve.jl`
- Define an optimization problem and initialize on a limit cycle, where constraints are satisfied `GCMAES.jl`, `ADNLPModels.jl` & `NLPModelsIpopt.jl`

- What's next ?
  - Sensibility analysis for guiding the engineering of a prototype
  - Moving forward to an optimal control model, without the cone constraint
  - Make the code publicly available

Annex 1: Basis vectors related to the local wind

- Apparent wind perpendicular to the lines:

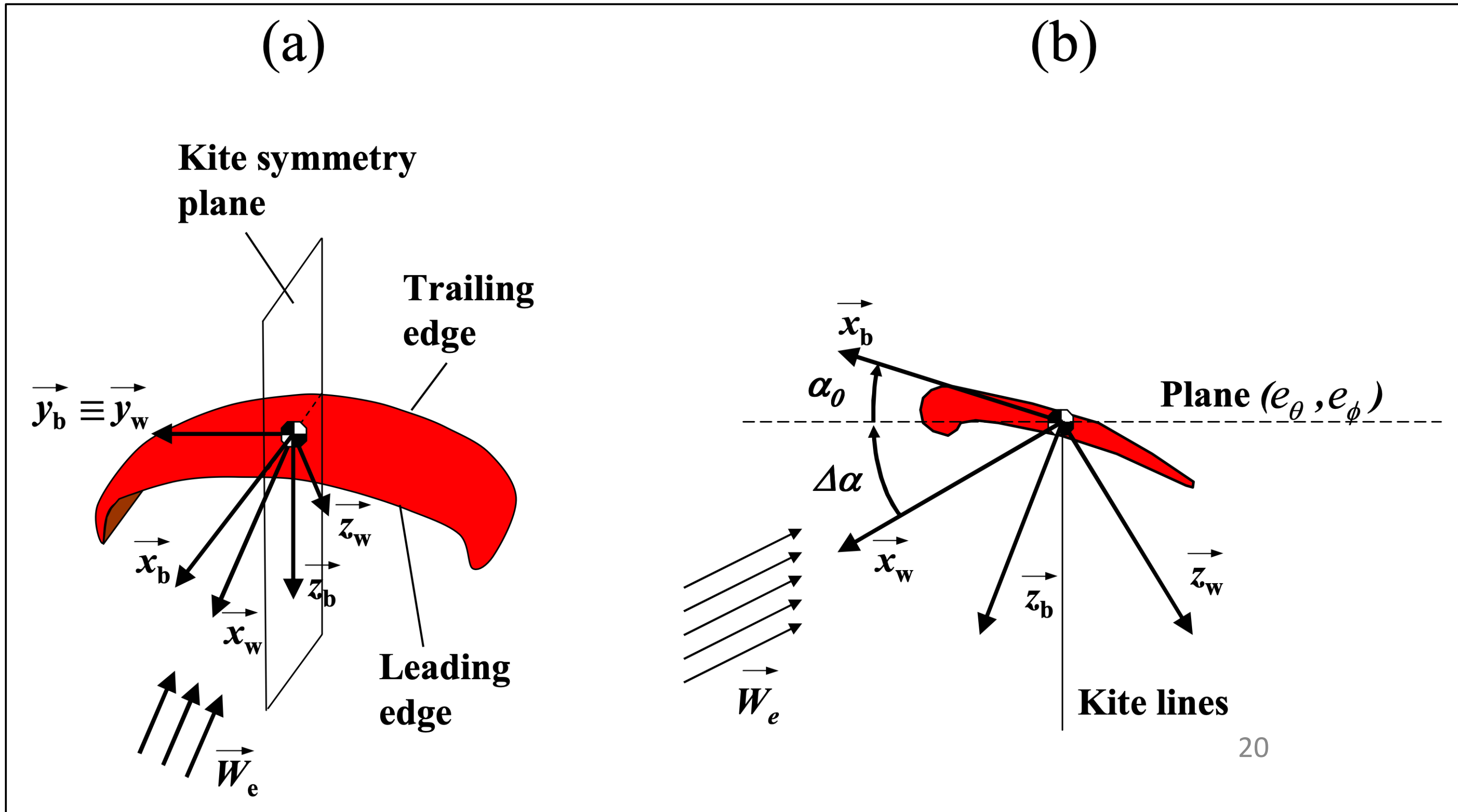
$$\widehat{e_w} = \frac{\overrightarrow{w_{app}} - (\hat{r} \cdot \vec{v})\hat{r}}{\left\|\overrightarrow{w_{app}} - (\hat{r} \cdot \vec{v})\hat{r}\right\|}$$

- Basis of the apparent wind:

$$\widehat{x_w} = -\frac{\overrightarrow{w_{app}}}{\left\|\overrightarrow{w_{app}}\right\|}$$

$$\widehat{y_w} = \hat{r} \times \widehat{e_w}$$

$$\widehat{z_w} = \widehat{x_w} \times \widehat{y_w}$$



Annex 2: Forward Automatic Differentiation

$$f(w_1, w_2) = \sin(w_1) + w_1 w_2$$

- Define ε st. $\varepsilon^2 = 0$
- A dual number is $a + b\varepsilon$
- Define the base operations:
  - $(a + b\varepsilon) + (c + d\varepsilon) = (a + c) + (b + d)\varepsilon$
  - $(a + b\varepsilon)(c + d\varepsilon) = ac + (ad + bc)\varepsilon$
  - $\dots$
- sin, cos, exp, etc. are defined using $+$, $\times$, $/$, etc.
- $f(x + \varepsilon) = f(x) + \varepsilon f'(x)$

$\rightarrow$ See ForwardDiff.jl