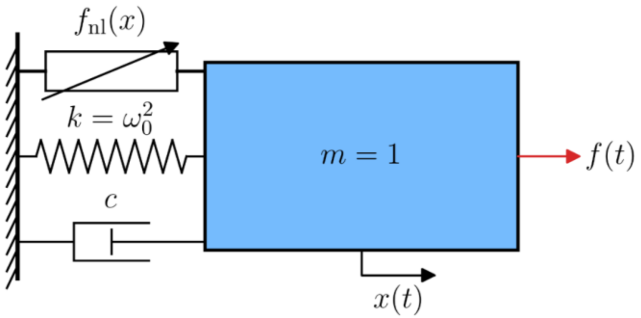
Symbolic implementation of the Method of Multiple Scales

An analytical analysis tool



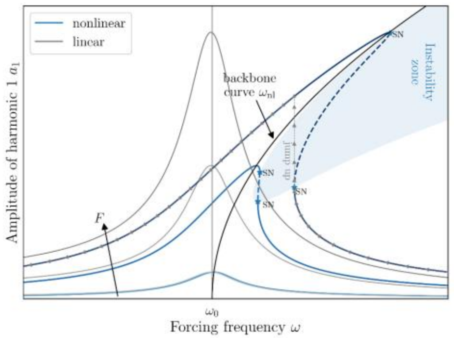
Vincent Mahé
vincent.mahe@ec-nantes.fr

EX-MODELI – 17/10/2025 – Lille

Motivation

linear oscillator | nonlinearities | forcing

Eq. $\ddot{x} + c\dot{x} + \omega_0^2 x + f_{\mathrm{nl}}(x, \dot{x}, \ddot{x}) = F\cos(\omega t)$



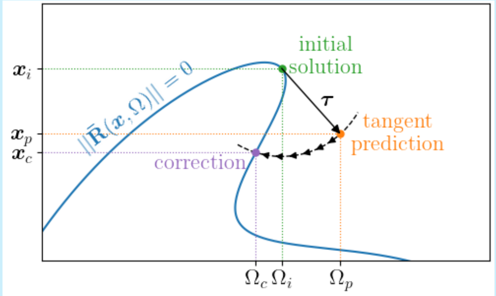
Numerical methods

Advantages

Accurate solutions

Complex **geometries**

Many **dof**



Drawbacks

Computational **time**

Long **parametric** studies

Less **insight**

Numerical **stability**

- Time integration
- Continuation methods
- …

Analytical methods

Advantages

Parametric investigation
➞ Deep **understanding**
➞ Design **rules**

Closed-form solution
➞ Computational **efficiency**

$$x(t) = x_0 + \epsilon x_1 + \cdots$$

Drawbacks

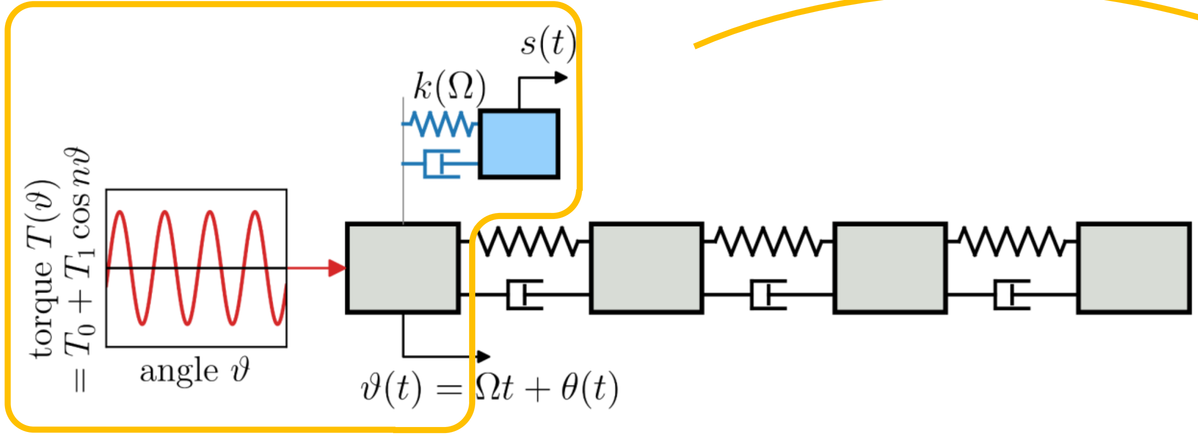
Difficult to carry-out

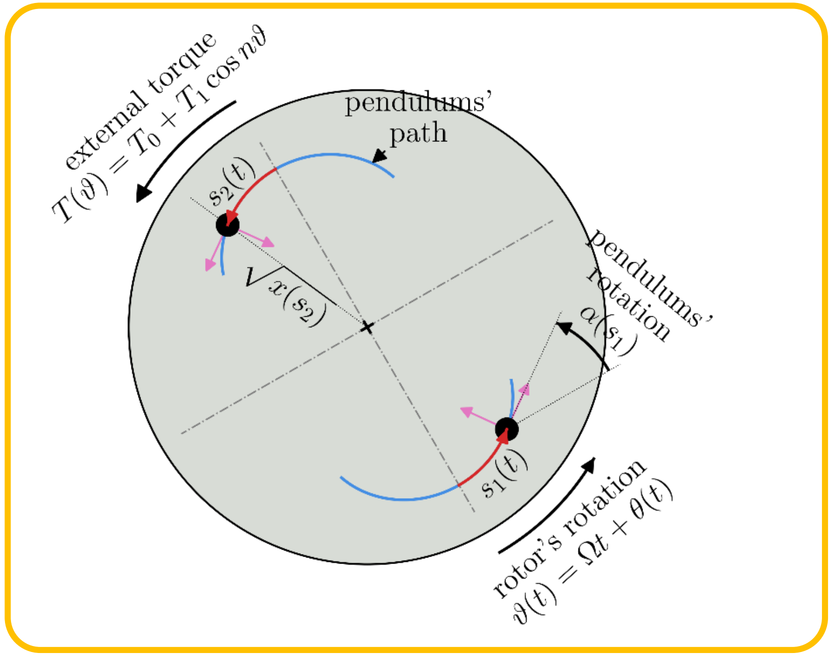
Approximate solutions

Limited to
➞ Simple geometries
➞ Few dof

- Lindstedt-Poincaré
- **Method of Multiple Scales (MMS)**
- …

Motivation





Objectives

1. **Improve** the absorption capacity

   → Find the **optimum choices** for
   - The pendulums' **path** $x(s_i)$
   - The pendulums' **rotation** $\alpha(s_i)$

2. **Understand** physical phenomena at stake

   → Determine the **dynamical effects** of
   - The pendulums' **path** $x(s_i)$
   - The pendulums' **rotation** $\alpha(s_i)$

Analytical approach

1. Provide **design guidelines**

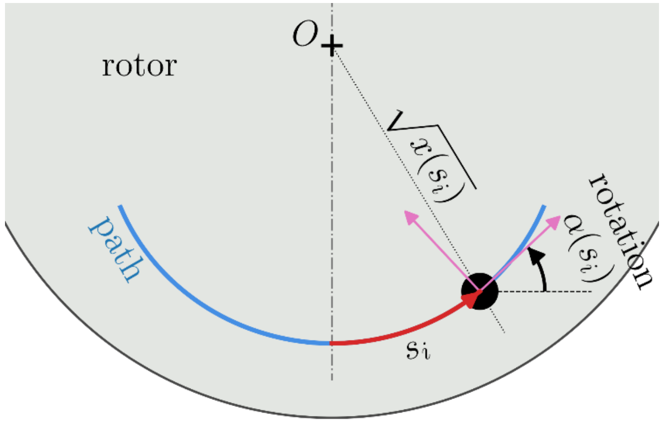
$$\begin{cases} x_{\mathrm{opt}}(s_i) = f_x(s_i; \boldsymbol{p}) \\ \alpha_{\mathrm{opt}}(s_i) = f_\alpha(s_i; \boldsymbol{p}) \end{cases}$$

sys. param.

2. **Asymptotic** solutions

$$s_i = f_i(\boldsymbol{p}), \qquad i = 1, 2$$

Reasonable hand application of the Method of Multiple Scales



	Linear effects		**Nonlinear** effects
Path	$x(s_i) = 1 - n_t^2 s_i^2$	+	$x_4 s_i^4$
Rotation	$\alpha(s_i) = \alpha_1 s_i$	+	$\alpha_3 s_i^3$

Problem to solve

$$\tilde{s}_i'' + n_p^2 \tilde{s}_i = -\frac{\epsilon}{\Lambda_m}\left[\underbrace{\frac{n_p^2 \tilde{\mu} \Lambda_c^2}{N} \sum_{j=1}^{N} \tilde{s}_j}_{\text{coupling between the pendulums}} + \underbrace{\tilde{b} \tilde{s}_i'}_{\text{damping}} - \underbrace{2\tilde{x}_{[4]} \tilde{s}_i^3}_{\text{path nonlinearity}} + \underbrace{6\eta\alpha_{[1]}\tilde{\alpha}_{[3]}(\tilde{s}_i^2 \tilde{s}_i'' + \tilde{s}_i \tilde{s}_i'^2)}_{\text{rotation nonlinearity}} + \underbrace{\Lambda_c \tilde{T}_1 \cos(n\tau)}_{\text{external forcing}} \right]$$

→ N linearly coupled nonlinear equations

Apply the Method of Multiple Scales

Design guidelines

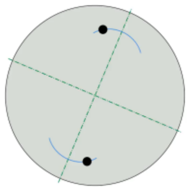
Linear tuning: $n_p(n_t, \alpha_1) = n$

Nonlinear tuning: $c_p(x_4, \alpha_3) = 0$

Choose to
- Avoid instabilities
- Lock the antiresonance

Prohibited hand application of the Method of Multiple Scales

Simplest nonlinear pendulum model



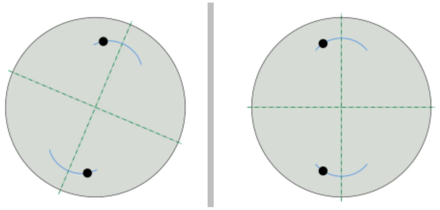
→ Direct unison response

$$\tilde{s}_i'' + n_p^2 \tilde{s}_i = -\frac{\epsilon}{\Lambda_m}\Bigg[\underbrace{\frac{n_p^2 \tilde{\mu} \Lambda_c^2}{N} \sum_{j=1}^{N} \tilde{s}_j}_{\text{coupling between the pendulums}} + \underbrace{\tilde{b}\tilde{s}_i'}_{\text{damping}} - \underbrace{2\tilde{x}_{[4]}\tilde{s}_i^3}_{\text{path nonlinearity}} + \underbrace{6\eta\alpha_{[1]}\tilde{\alpha}_{[3]}(\tilde{s}_i^2\tilde{s}_i'' + \tilde{s}_i\tilde{s}_i'^2)}_{\text{rotation nonlinearity}} + \underbrace{\Lambda_c \tilde{T}_1 \cos(n\tau)}_{\text{external forcing}} \Bigg].$$

Can be done by hand

Design guidelines $\begin{cases} n_p(n_t, \alpha_1) = n \\ c_p(x_4, \alpha_3) = 0 \end{cases}$

Advanced nonlinear pendulum model



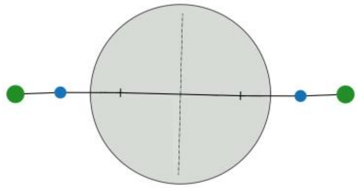
→ Direct unison response
→ Subharmonic responses

$$s_i'' + n_p^2 s_i = -\frac{\epsilon}{\Lambda_m}\Bigg\{ \underbrace{\frac{\Lambda_c^2 \tilde{\mu}}{N} \sum_{j=1}^{N} n_p^2 s_j}_{\text{linear coupling between the pendulums}} + \underbrace{\tilde{b} s_i'}_{\text{damping}} + \underbrace{\frac{\tilde{\mu} n_t^2 \Lambda_c}{N}\left[\sum_{j=1}^{N} s_j(2s_j' + s_i')\right]}_{\text{nonlinear (quadratic) coupling between the pendulums}}$$
$$+ \underbrace{\frac{\tilde{\mu} n_t^2}{N}\left[\sum_{j=1}^{N}(1+n_t^2)\Lambda_c\left(s_j {s_j'}^2 - \frac{n_p^2}{2}(s_j^3 + s_j s_i^2)\right) + 2\Lambda_m s_j s_j' s_i'\right]}_{\text{nonlinear (cubic) coupling between the pendulums}} - \underbrace{2\tilde{x}_{[4]} s_i^3}_{\text{path nonlinearity}}$$
$$+ \underbrace{6\eta\alpha_{[1]}\tilde{\alpha}_{[3]}(s_i {s_i'}^2 + s_i^2 s_i'')}_{\text{rotation nonlinearity}} + \underbrace{\left(\Lambda_c + \Lambda_m s_i' - \frac{n_t^2(1+n_t^2)}{2} s_i^2\right)\tilde{T}_1\cos(n\vartheta)}_{\text{external forcing (direct and parametric)}} \Bigg\}.$$

Automation tremendously helped

Design guidelines $\begin{cases} n_p(n_t, \alpha_1) = n \\ c_p(x_4, \alpha_3) = c_c \end{cases}$ | $\begin{cases} n_p(n_t, \alpha_1) = n/2 \\ c_p(x_4, \alpha_3) = 0^+ \end{cases}$

Double pendulums



$$\Lambda_{m1}\varphi_{1i}'' + \Lambda_{c12}\varphi_{2i}'' + k_1\varphi_{1i} + k_{12}\varphi_{2i} = -\epsilon f_{p1i}(\boldsymbol{\varphi}, \vartheta),$$
$$\Lambda_{c12}\varphi_{1i}'' + \Lambda_{m2}\varphi_{2i}'' + k_{12}\varphi_{1i} + k_2\varphi_{2i} = -\epsilon f_{p2i}(\boldsymbol{\varphi}, \vartheta).$$

1 page-long equations…

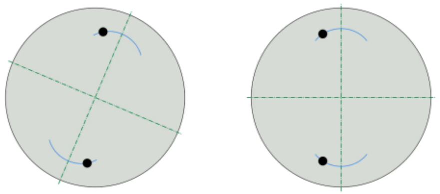
Cannot be done by hand

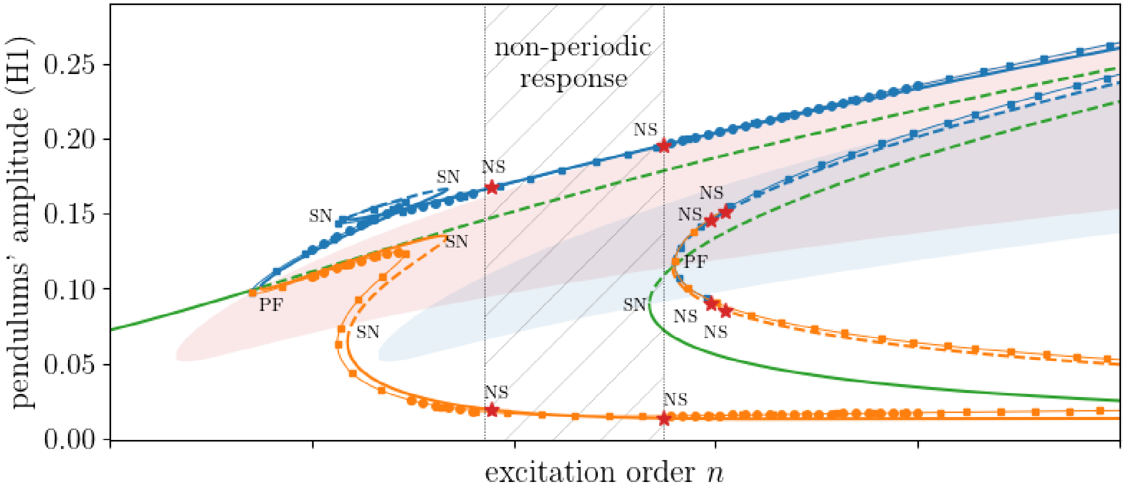
Automation required

Design guidelines $\begin{cases} n_{p_i} = n \\ c_{p_i} = c_{c_i} - c_{d_i} \end{cases}$

Analytical results are accurate enough

Numerical validation

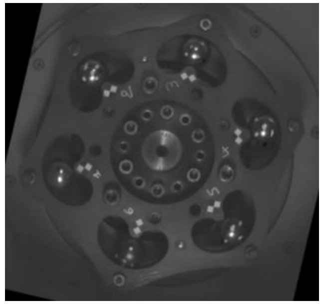


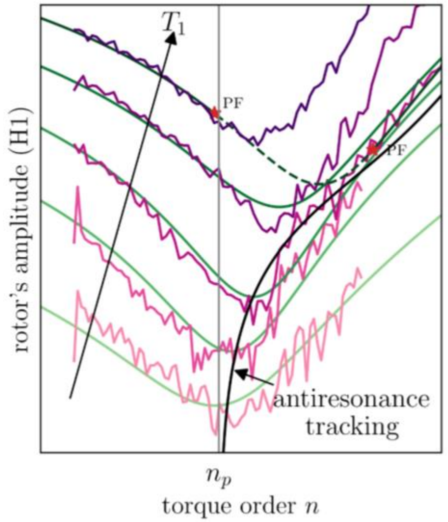


- ➜ Unison response
- ➜ Coupled-mode response
- ➜ Stability

Experimental validation

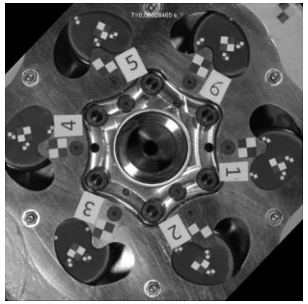
6 pendulums

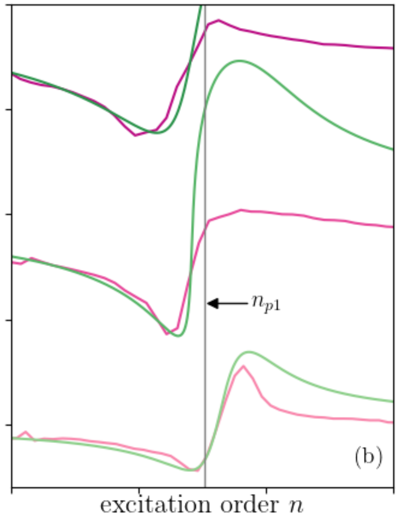




- ➜ Unison response
- ➜ Antiresonance tracking

6 double pendulums





- ➜ Unison response
- ➜ Antiresonance tracking

The Method of Multiple Scales

1. Scaling
small param. ϵ

2. Independent **time scales**
$t_0 = t, \quad t_1 = \epsilon t$

3. Asymptotic solution
$x(t) = x_0(t_0, t_1) + \epsilon x_1(t_0, t_1)$

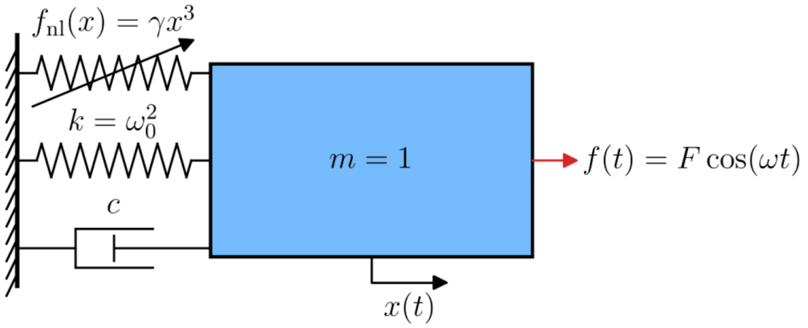
4. Frequency **range**
$\omega = r\omega_0 + \epsilon\sigma$

5. Solve x_n **and substitute in** f_{n+1} **successively**

6. Secular analysis

Nonlinear Eq.

$$\ddot{x} + c\dot{x} + \omega_0^2 x + f_{\mathrm{nl}}(x) = f(t)$$



Linear system

$$\begin{cases} \epsilon^0 \rightarrow \dfrac{\partial^2 x_0}{\partial t_0^2} + \omega_0^2 x_0 = 0 \\ \epsilon^1 \rightarrow \dfrac{\partial^2 x_1}{\partial t_0^2} + \omega_0^2 x_1 = f_1(x_0, t_0, t_1) \end{cases}$$

Solutions and slow evolution

$$x_0 = a_0(t_1)\cos\big(\omega_0 t_0 - \beta_0(t_1)\big)$$

$$\begin{cases} \dfrac{\partial a_0}{\partial t_1} = f_a^{(1)}(a_0, \beta_0) \\ a_0 \dfrac{\partial \beta_0}{\partial t_1} = f_\beta^{(1)}(a_0, \beta_0) \end{cases} \rightarrow$$

Amplitude and phase **evolution** with the **slow time**

The Method of Multiple Scales

Evolution equations

$$\begin{cases} \dfrac{\mathrm{d}a_0}{\mathrm{d}t} = f_a(a_0, \beta_0) \\ a_0 \dfrac{\mathrm{d}\beta_0}{\mathrm{d}t} = f_\beta(a_0, \beta_0) \end{cases}$$

→ **Time evolution** of the amplitude and phase

7. Steady-state

$\mathrm{d}a_0/\mathrm{d}t = \mathrm{d}\beta_0/\mathrm{d}t = 0$

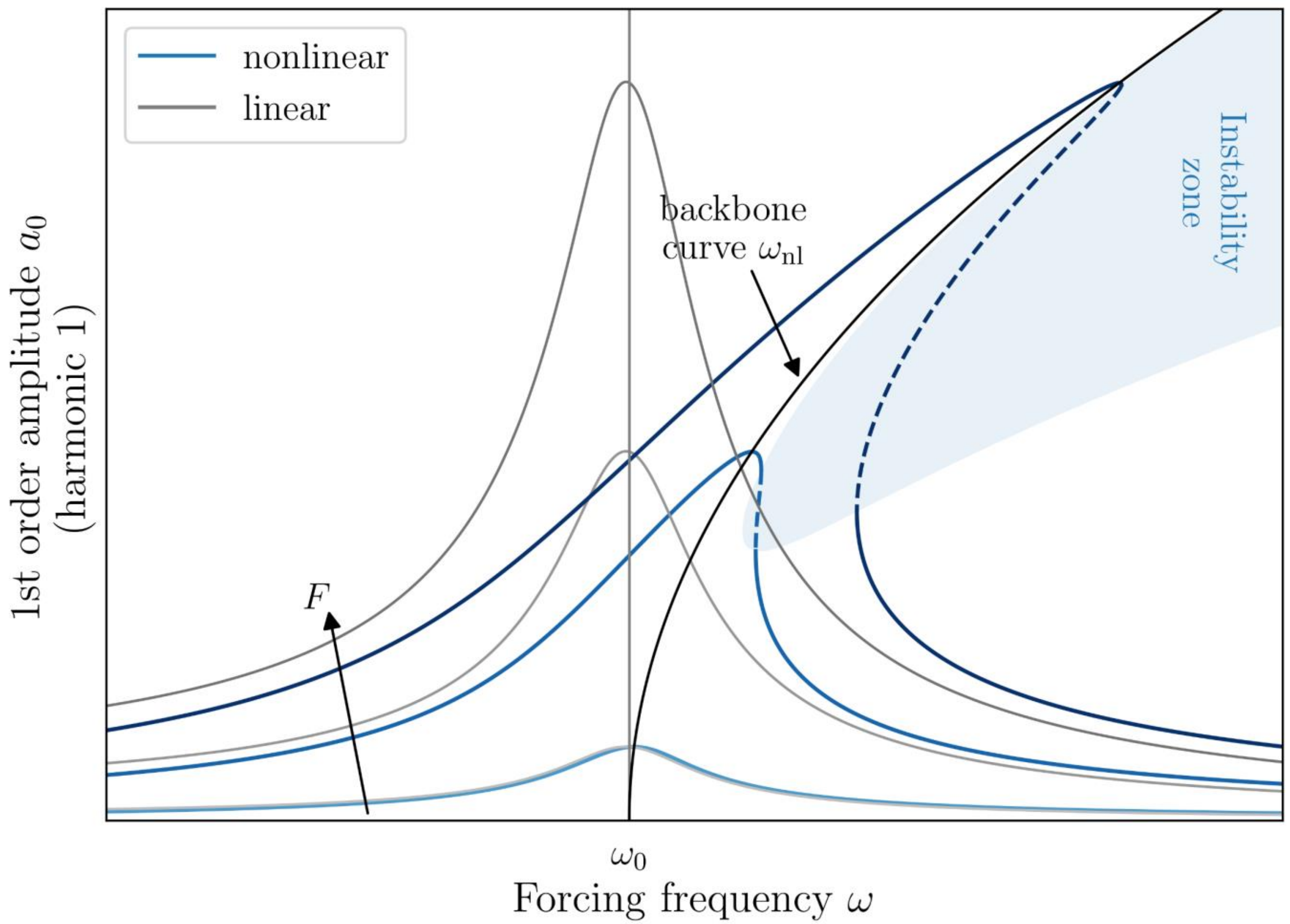
NL forced response

Solve $\begin{cases} f_a(a_0^*, \beta_0^*) = 0 \\ f_\beta(a_0^*, \beta_0^*) = 0 \end{cases}$ → Amplitude and phase **response**

8. Stability analysis

Stability information

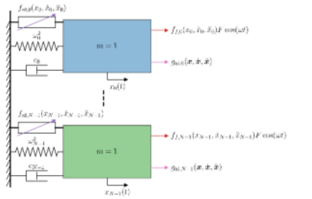
Jacobian $\boldsymbol{J}|_{(a_0^*, \beta_0^*)}$ eigenvalue analysis → Stability of (a_0^*, β_0^*)

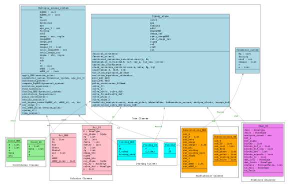


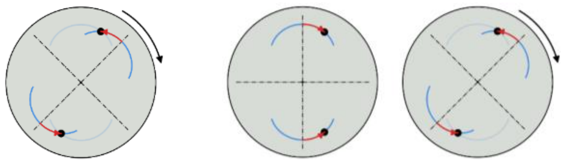
Implementation of the MMS

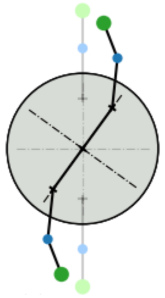
2019	2020 - 2021	2022	2023	Spring 2025	Summer 2025
Hand application	Symbolic-aided hand application	Full symbolic implementation	Broadened application cases	Class structure Code robustness	Open source Documentation Python package

SymPy SymPy





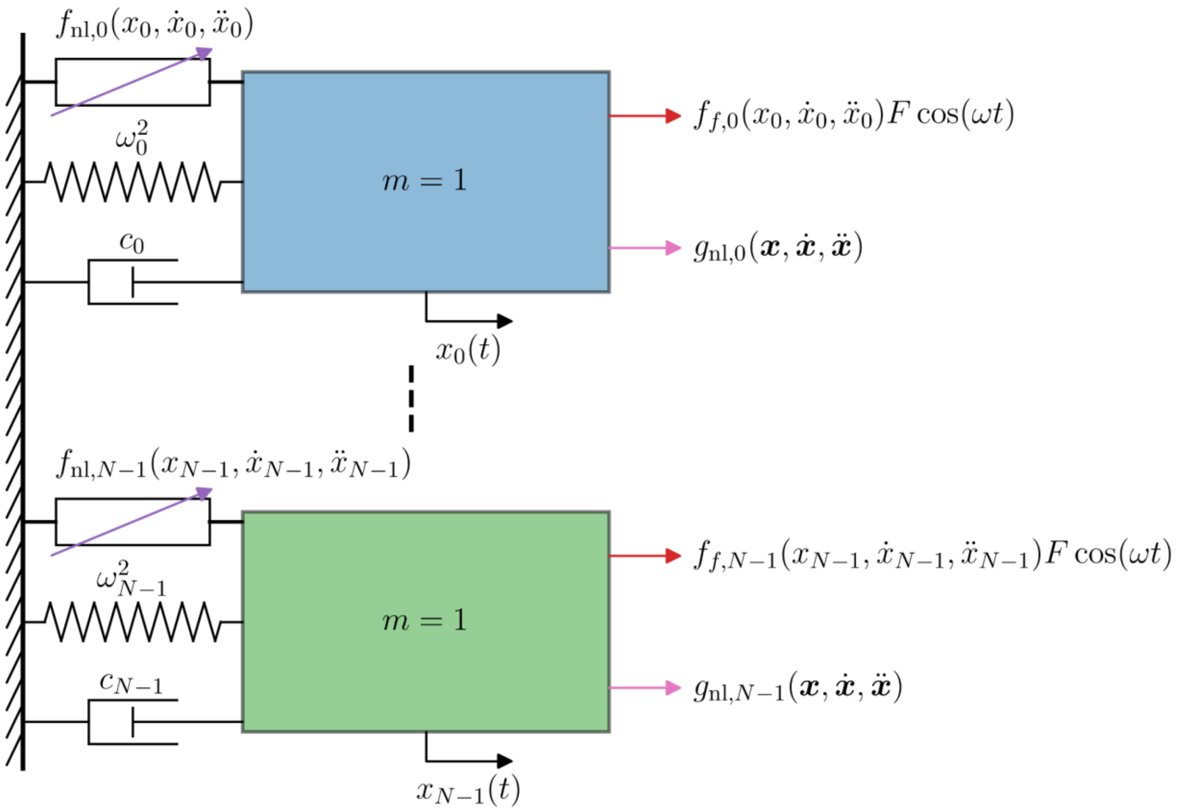




OSCILATE

oscilate

Nonlinear systems considered



Oscillators

- → $\boldsymbol{N}$ oscillators $x_0(t), \dots, x_{N-1}(t)$
- → Linear, conservative at **leading order**

Nonlinearities

- → **Weak**
- → **Polynomial** (otherwise a Taylor expansion is performed)
- → Either in **displacement**, **velocity** or **acceleration**

Coupling

- → **Weak**
- → **Linear** or **nonlinear**
- → Any **internal resonance** relation

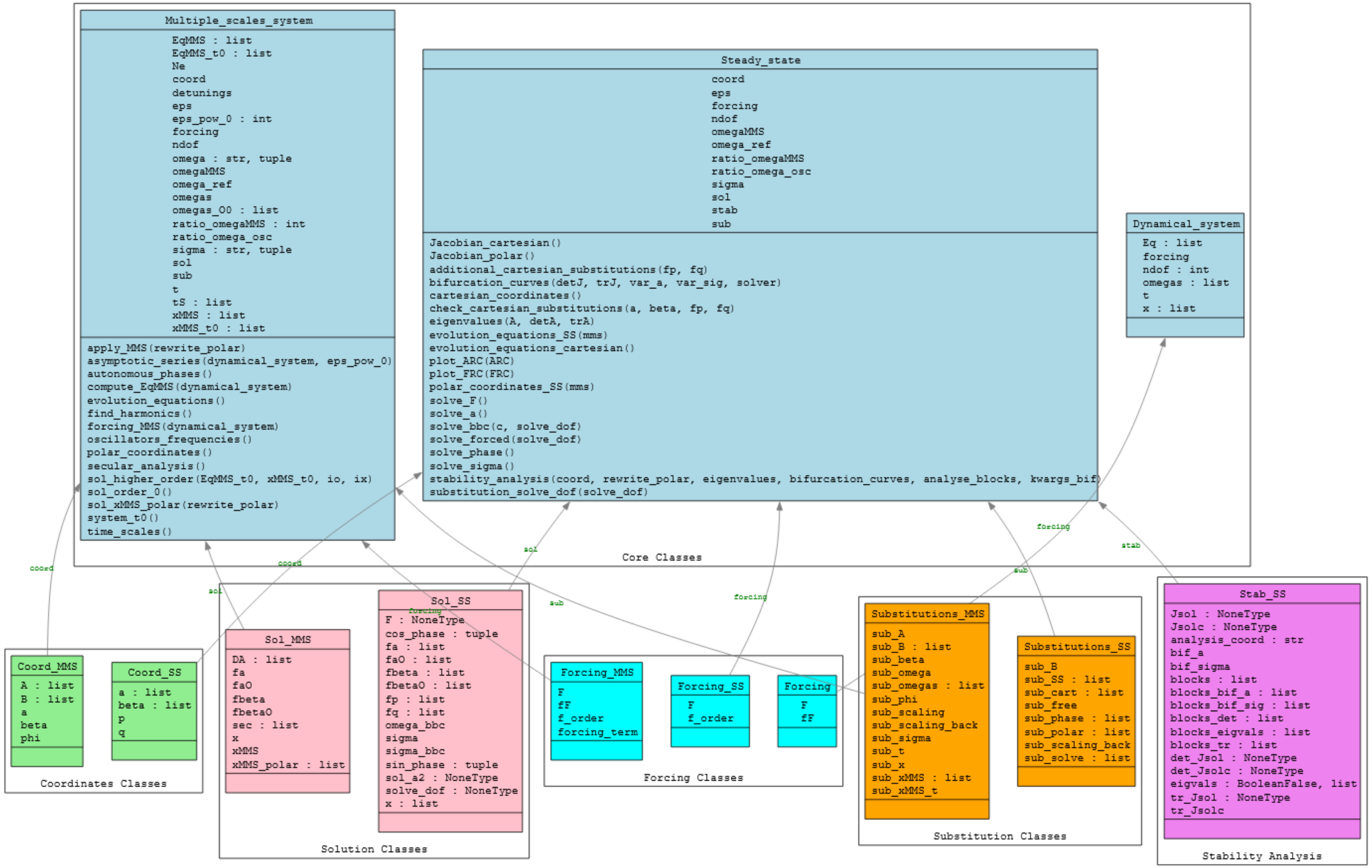
Forcing

- → **Weak** or **hard**
- → **Harmonic**
- → **Parametric**

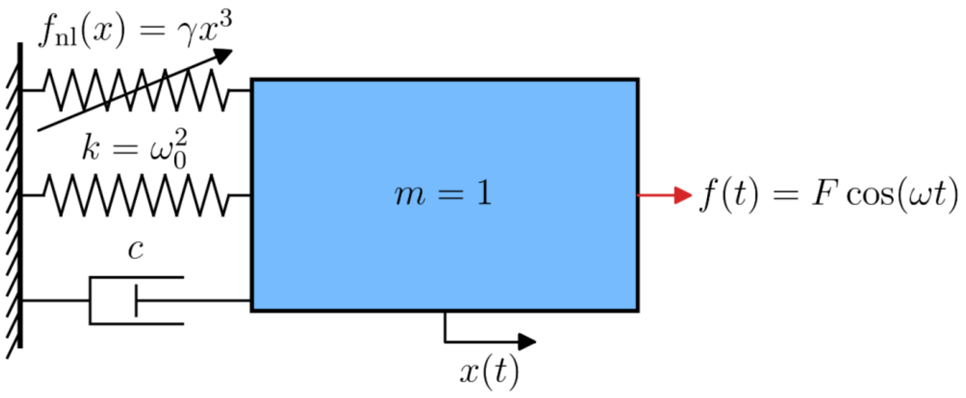
Code architecture

Main classes

- Dynamical_system
  - → Description of the dynamical system
- Multiple_scales_system
  - → Application of the MMS
- Steady_state
  - → Evaluation at steady state
  - → Computation of forced responses
  - → Computation of backbone curves
  - → Stability analysis
  - → Plotting tools



Code outputs



MMS at order 3

$$\begin{cases} t_0 = t, \qquad t_1 = \epsilon t, \qquad t_2 = \epsilon^2 t \\ x(t) = x_0(\boldsymbol{t}) + \epsilon x_1(\boldsymbol{t}) + \epsilon^2 x_2(\boldsymbol{t}) \end{cases}$$

$$\boldsymbol{t} = [t_0, t_1, t_2]$$

[2]
```
ss.sol.fa[0]
```
✓ 1.1s

$$-\frac{\epsilon^3 \tilde{c} a_0}{2} + \frac{\epsilon^3 \tilde{F} \sin(\beta_0)}{2\omega_0}$$

[3]
```
ss.sol.fbeta[0]
```
✓ 1.2s

$$\frac{\epsilon^3 \tilde{F} \cos(\beta_0)}{2\omega_0} - \frac{123\epsilon^3\tilde{\gamma}^3 a_0^7}{8192\omega_0^5} + \frac{15\epsilon^2\tilde{\gamma}^2 a_0^5}{256\omega_0^3} + \epsilon\sigma a_0 - \frac{3\epsilon\tilde{\gamma} a_0^3}{8\omega_0}$$

[4]
```
ss.sol.omega_bbc.expand()
```
✓ 1.1s

$$\frac{123\epsilon^3\tilde{\gamma}^3 a_0^6}{8192\omega_0^5} - \frac{15\epsilon^2\tilde{\gamma}^2 a_0^4}{256\omega_0^3} + \frac{3\epsilon\tilde{\gamma} a_0^2}{8\omega_0} + \omega_0$$

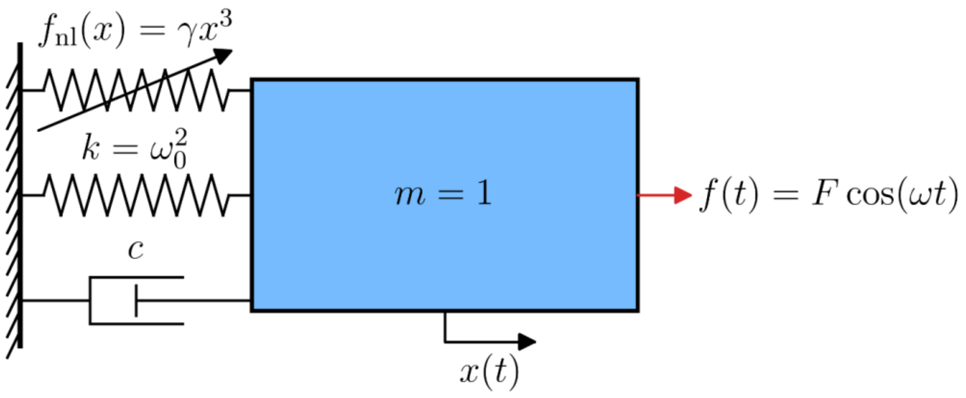
$$\begin{cases} \dfrac{\mathrm{d}a_0}{\mathrm{d}t} = f_a(a_0, \beta_0) \\ a_0\dfrac{\mathrm{d}\beta_0}{\mathrm{d}t} = f_\beta(a_0, \beta_0) \end{cases}$$

evolution equations

backbone curve

$$\omega_{\mathrm{bbc}} = \omega_0 + f_{\mathrm{bbc}}(a_0)$$

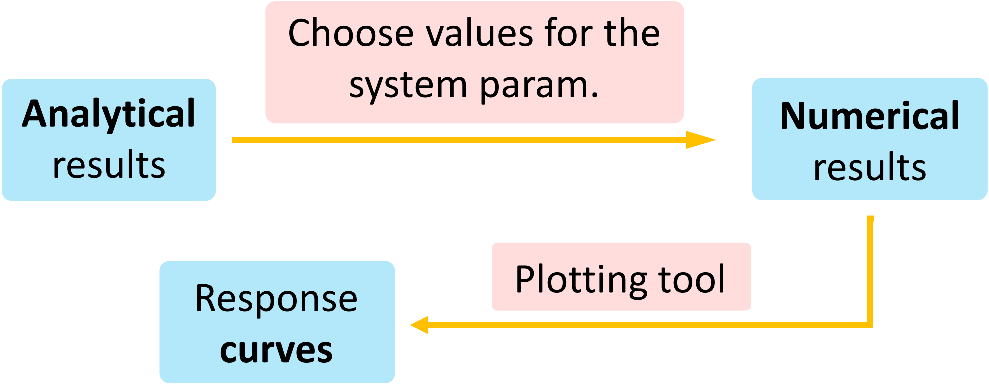
Code outputs

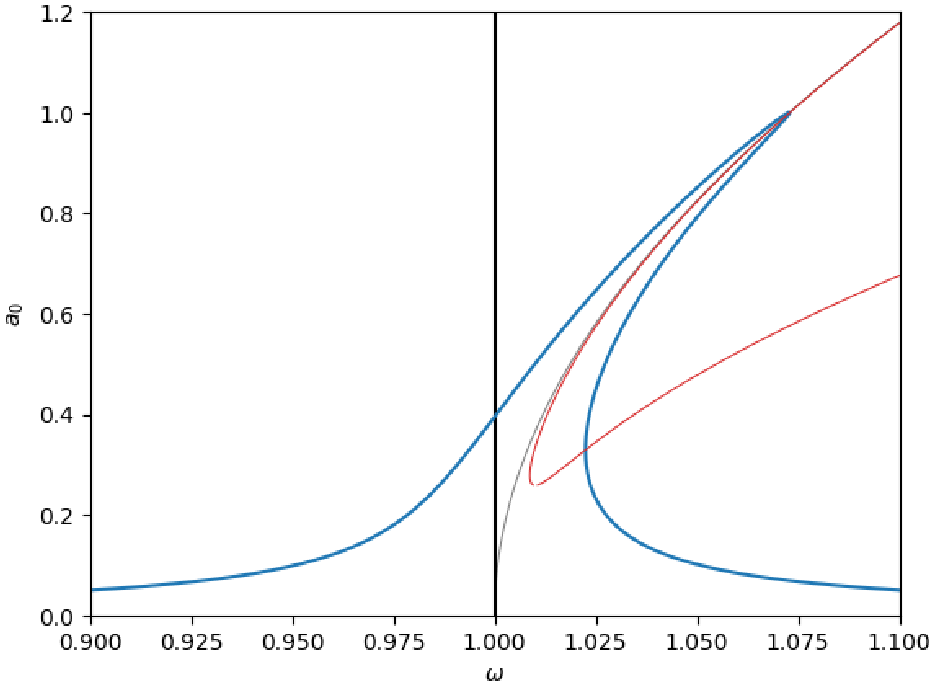


MMS at order 3

$$\begin{cases} t_0 = t, \quad t_1 = \epsilon t, \quad t_2 = \epsilon^2 t \\ x(t) = x_0(\boldsymbol{t}) + \epsilon x_1(\boldsymbol{t}) + \epsilon^2 x_2(\boldsymbol{t}) \end{cases}$$

$$\boldsymbol{t} = [t_0, t_1, t_2]$$





Open source project and python package

Open source codes on GitHub: https://github.com/VinceECN/OSCILATE

Project called **OSCILATE**

Archived on SoftwareHeritage

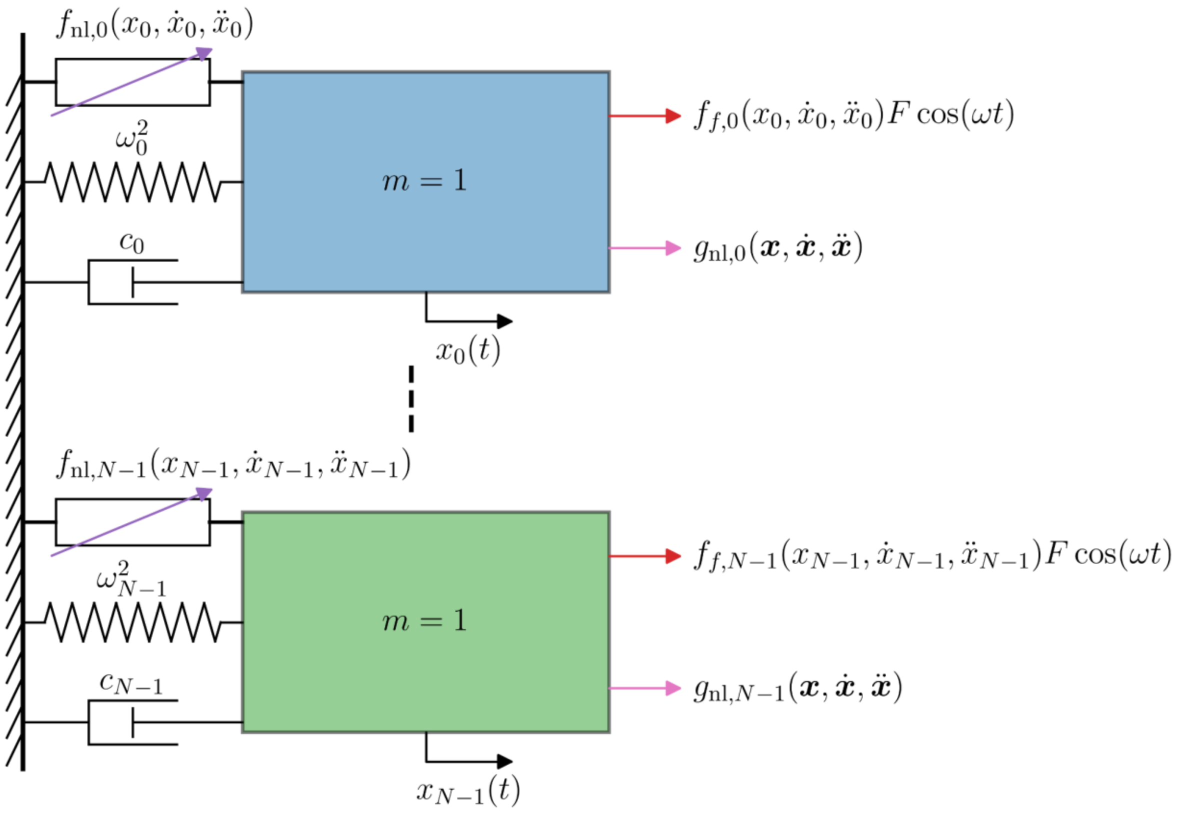
Python **package**: https://pypi.org/project/oscilate/

Package called **oscilate**

Easy install with `pip install oscilate`

Detailed **documentation:** https://vinceecn.github.io/OSCILATE/

Conclusion and perspectives



SymPy

python Package Index

DOCS

Pre-processing

- ➔ Equations of motion
- ➔ Weak coupling

More test cases

- ➔ Specific nonlinearities
- ➔ Complex internal resonances

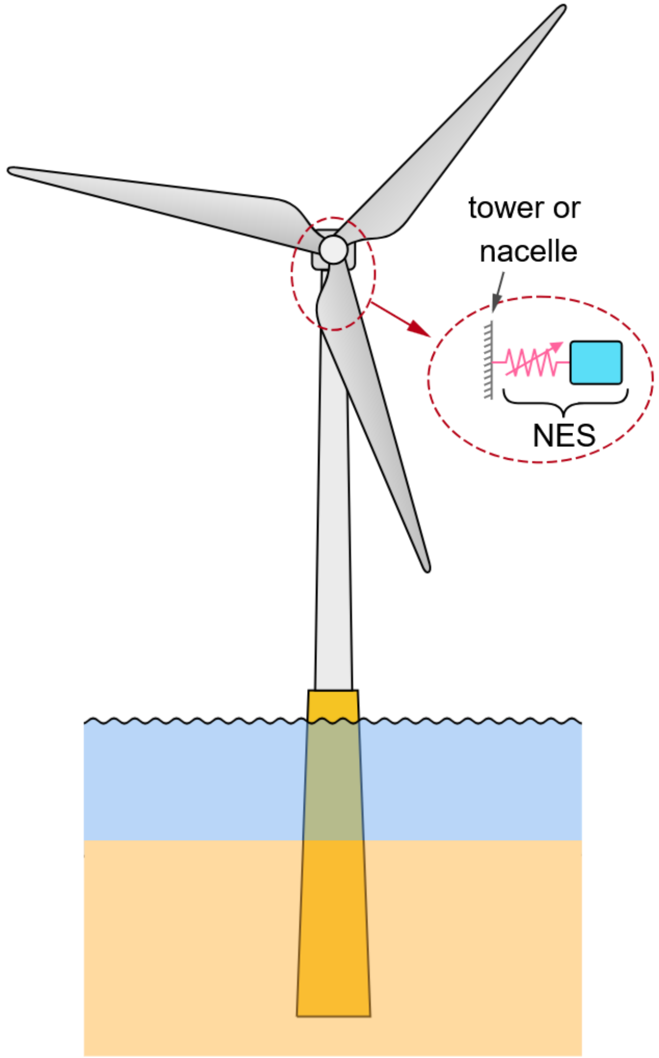
Minor additions

- ➔ Steady state solver
- ➔ Polar ↔ Cartesian
- ➔ Periodic forcing

Major additions

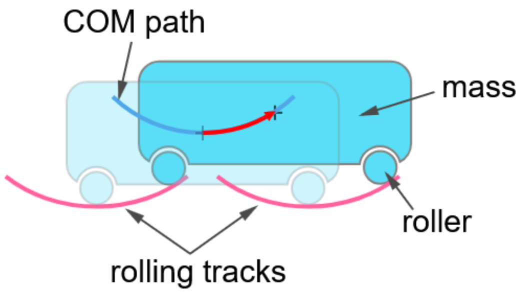
- ➔ MMS procedure: homogeneous solutions
- ➔ Nonlinear wave dispersion analysis

Postdoc offer



Objective

Improve the design of track NES



Supervision	V. Mahé
Collaboration	B. Chouvion CREA, Salon-de-Pce
Duration	1 year
Start	Between 01/26 and 09/26
Location	GeM, Nantes