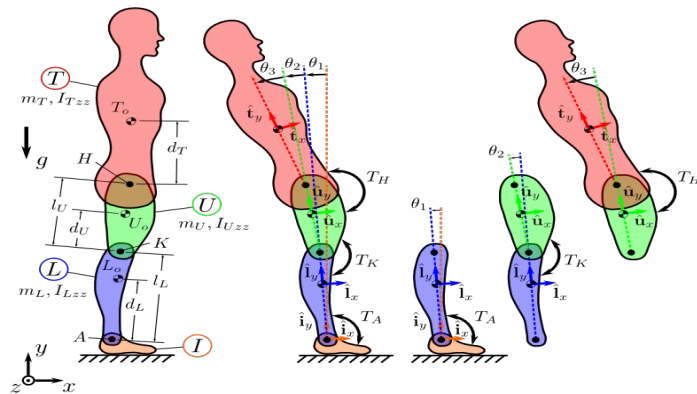


Pendule, inertie, mouvements de robot: une modélisation de la mécanique avec SymPy

K.I.A.Derouiche
kamel.derouiche@gmail.com

Algerian
IT Security Group

PyConFr



SymPy c'est quoi ?

- 100 % Python
- Objectif: devenir un **Computer Algebra System** !
- Manipulation d'expressions symboliques
- Utilisations : module (`from sympy import *`), IPython,

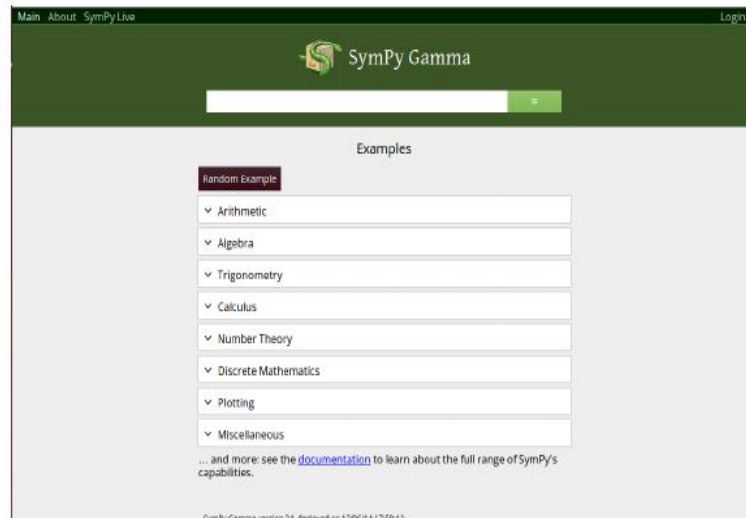
SymPyGamma, SymPyLive



SymPy on-web

<http://www.sympygamma.com>

- Interface web
- Pas d'installation
- Courbe d'apprentissage très douce
- Monter en puissance



Pourquoi SymPy ?

- Développement rapide (communauté de développeur Python) !
- Éviter d'apprendre différent langage (vs Mathematica, Maxima, Pari, Maple, Matlab)
- Graphique contrairement à (OpenAxiom, Pari,...)
- SymPy vs Sagemath:
 - Sympy est contenu dans Sagemath (exemple) !
 - La seule dépendance requise est le langage Python.
 - La taille !

Crée une expression

In [8]: `from sympy import symbols, init_printing`

In [9]: `from sympy import sin, cos`

In [10]: `A, b, c = symbols('A b c')`

In [11]: `expr = sin(A) + cos(b**2)*c`

In [12]: `expr`

Out[13]: `sin(A) + cos(b**2) *c`

Calcul numérique !

In [27]: Rational(1,2)

Out [28]: 1 / 2

In [29]: Rational(1,2).evalf()

Out [30]: 0.5000000000000000

In [31]: pi.evalf(100)

Out [32]:

3.1415926535897932384626433832795028841971693993751058209749445923
07816406286208998628034825342117068

Résoudre

```
In [50]: x = Symbol('x')
```

```
In [51]: lhs = (4*x**2 - 6*x + 9, x)
```

```
In [52]: expr = Eq(lhs, 0)
```

```
In [53]: expr1 = x**2 + x + 9
```

#Solution de l'expression

```
In [54]: solve(expr1, x)
```

```
Out[55]: [-1/2 - sqrt(35)*I/2, -1/2 + sqrt(35)*I/2]
```

Les modules SymPy

- `matrix` Matrice classe, orthogonalization
- `nththeory` Théorie des nombres
- Algèbre linéaire
- Combinatoire
- Géométrie Différentielle
- **Physique**
- Théorie des catégories

etc...

Les modules physiques de SymPy

In [1]: from sympy.physics.

sympy.physics.gaussopt sympy.physics.quantum

sympy.physics.hep sympy.physics.secondquant

sympy.physics.hydrogen sympy.physics.sho

sympy.physics.mechanics sympy.physics.units

sympy.physics.paulialgebra sympy.physics.vector

etc...

Mécanique classique

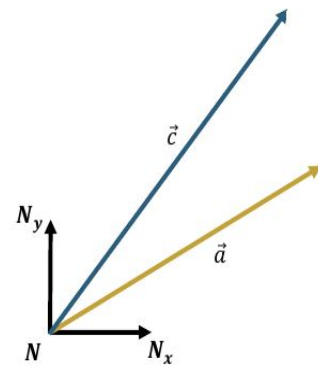
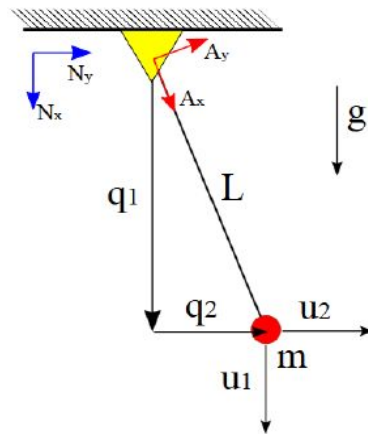
$$\ddot{\theta} + \omega^2 \sin \theta = 0$$

$$f_c(q_1, q_2, t) = q_1^2 + q_2^2 - L^2 = 0$$

```
def simple_pendulum(theta_thetadot, t):  
    theta, theta_dot = theta_thetadot  
    return [theta_dot, - np.sin(theta)]
```

```
t = np.linspace(0, 5 * np.pi, 1000)
```

```
odeint(simple_pendulum, (np.pi/3, 0), t)
```



Symbolique, numérique, ou les deux à la fois?

Pourquoi le symbolique? :

- Une meilleure compréhension des équations de mouvements
- Un code numérique plus rapide généré par un formalisme symbolique
- Une qualité d'impression et de rendu pour les publications et la communications scientifiques(`mechanics_printing`)

PyDy

Un outil open source pour la l'analyse, la simulation et la visualisation des systèmes multicorps.

- PyDy (Python Dynamics) :

Systèmes multicorps, englobe les trois aspects.

- SymPy Mechanics :

Le moteur symbolique de base pour équation du mouvement dérivation est construit au sommet d'sympy et est au cœur de PyDy.

Software Design

system.py: fournit une classe de système pour gérer la simulation de système mécanique.

models.py: fournit certains modèles en conserve des systèmes classiques.

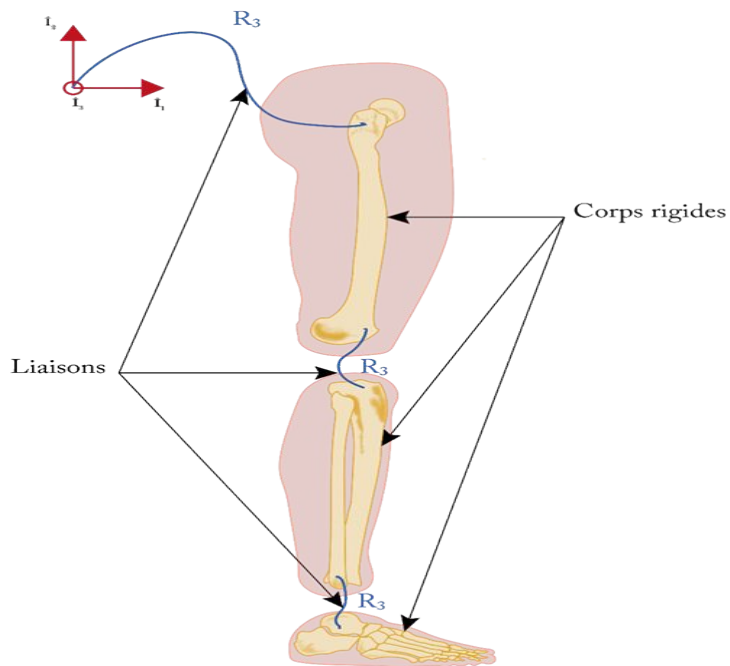
pydy.codegen: Contient divers outils pour générer du code numérique à partir d'expression des équations du mouvement symboliques.

pydy-viz: outil graphique basée sur un navigateur intégré pour générées les visualisations.

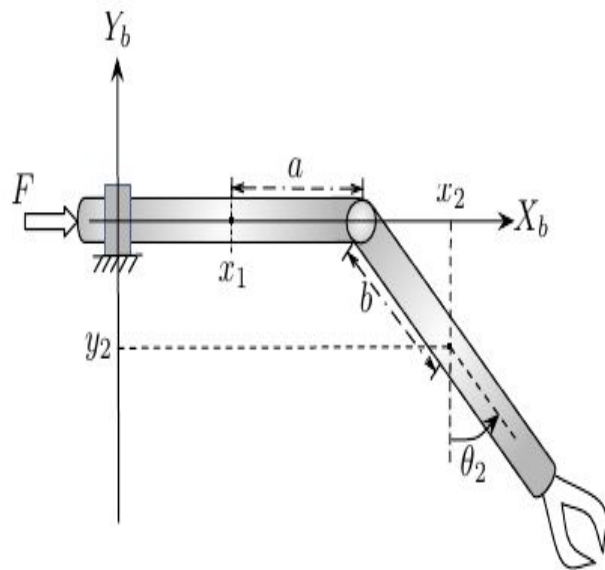
Application

- Robotique
- Biomécanique
- Véhicule dynamique
- Physique/engineering éducation
- Sport biomécanique

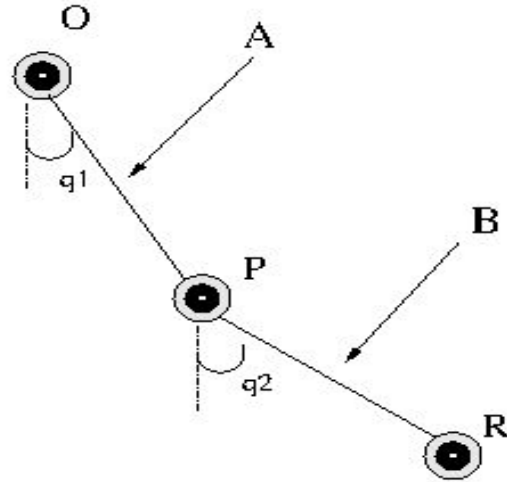
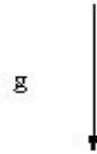
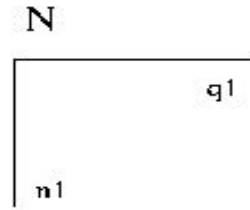
Mouvements d'un robot



Modèle multi-corps simplifié du membre inférieur



Mouvements d'un robot (cont.)



Documentation

- **SymPy** sympy.org
- **PyDy** pydy.org
- **Projets qui utilisent SymPy:**
 - <https://github.com/cdsousa/sympybotics>
 - <https://pypi.python.org/pypi/Hamilton>
 - <https://pypi.python.org/pypi/arboris>
 - <https://pypi.python.org/pypi/PyODE>
 - <https://pypi.python.org/pypi/odeViz>
 - <https://pypi.python.org/pypi/ARS>

Questions ???