Siconos

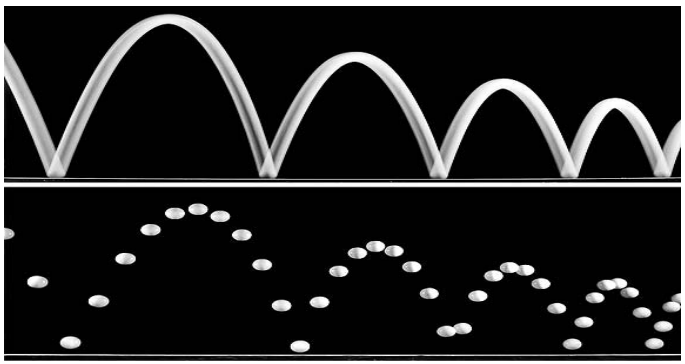
An open source library for nonsmooth mechanical systems involving contact, impact, friction, plasticity and fracture.

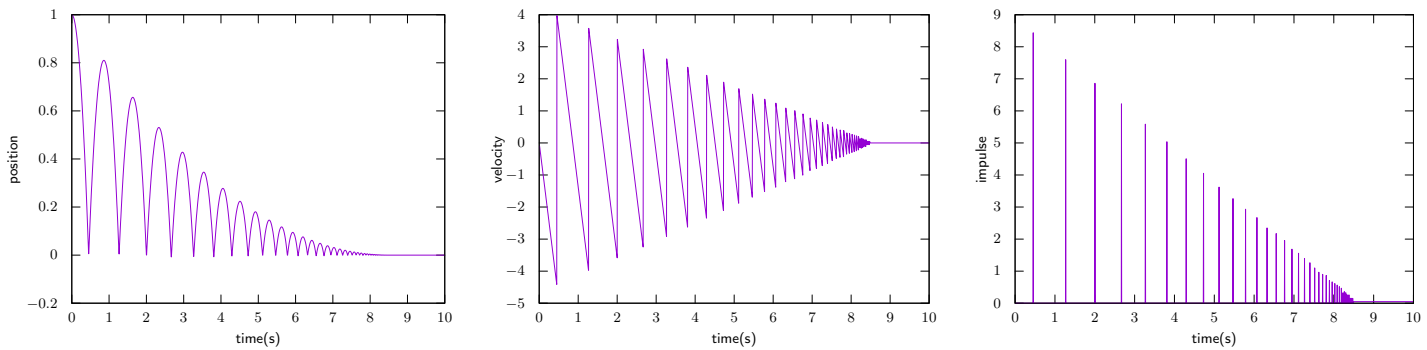
V. Acary & F. Pérignon
siconos-team@inria.fr
https://siconos.org

Inria - Centre de l'Université Grenoble Alpes - Laboratoire Jean Kuntzmann

Nonsmooth dynamical systems

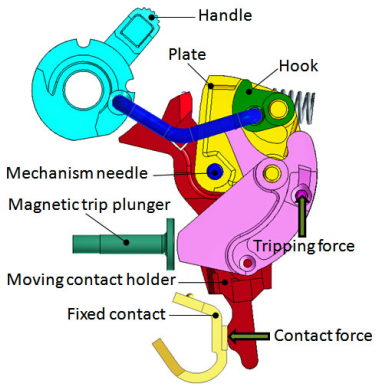
nonsmooth = lack of continuity/differentiability

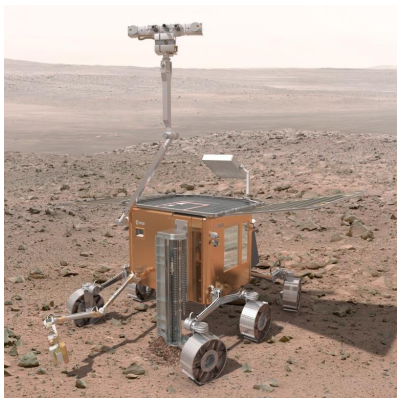




- nonsmooth solutions in time (jumps, kinks, measures, distributions)
- nonsmooth modeling and constitutive laws (set–valued mapping, inequality constraints, complementarity, impact laws)

Application fields.



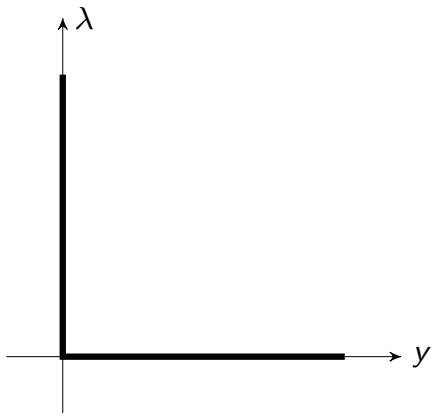


- Mechanical systems with unilateral contact, Coulomb friction and impacts : multi-body systems, robotic systems, frictional contact oscillators, granular materials, plasticity, fracture.
- Switched electrical circuits (diodes, transistors, switchs).
- Fluid Mechanics: cavitation, gas appearance multi-phasic fluid, permeability
- Hybrid and Cyber–physical systems
- Biology : gene regulatory networks
- Fluid transportation networks with queues.

Nonsmooth approach is crucial for a correct modeling and an efficient simulation

Nonsmooth dynamical systems

Difficulty: Standard tools of numerical analysis and simulation (in finite dimension) are no longer suitable due to the lack of regularity.



$$
\begin{array}{c}
0 \leqslant y \perp \lambda \geqslant 0 \\
\Updownarrow \\
-y \in N_{\mathbb{R}_+}(\lambda) \\
\Updownarrow \\
y^\top(\lambda' - \lambda) \geqslant 0, \forall \lambda' \in \mathbb{R}_+
\end{array}
$$

Specific tools

Differential measure theory. Convex, nonsmooth and variational Analysis (Clarke, Wets & Rockafellar). Complementarity theory. Maximal monotone operators.

An example with unilateral contact and plasticity

A single differential measure variational inequality.

$$
\left\{
\begin{array}{l}
M\mathrm{d}v + B^\top \sigma(t)\mathrm{d}t = f_{\mathrm{ext}}(t)\mathrm{d}t + H(u(t))\mathrm{d}i_{\mathrm{N}} \\
\dot{u}(t) = v(t) \\
S\dot{\sigma}(t) = Bv(t) - \dot{\varepsilon}^p(t) \\
\dot{a}(t) = -D\dot{\alpha}(t) \\
v_{\mathrm{N}}(t) = H^\top(u(t))v(t) \\
\\
\begin{pmatrix} \dot{\varepsilon}^p(t) \\ \dot{\alpha}(t) \\ -\mathrm{d}i_{\mathrm{N}} \end{pmatrix} \in \mathbb{N}_{C \times T_{\mathbb{R}^m_+}(g_{\mathrm{N}}(t))} \begin{pmatrix} \sigma(t) \\ a(t) \\ (v_{\mathrm{N}}(t) + ev_{\mathrm{N}}^-(t)) \end{pmatrix}.
\end{array}
\right. \tag{1}
$$

Siconos overview

Context and History

- The Siconos Platform was one of the main outcome of the Siconos EU project (2001–2005).
- There was no other general, common and open software suitable for the modeling and the simulation all of these NSDS

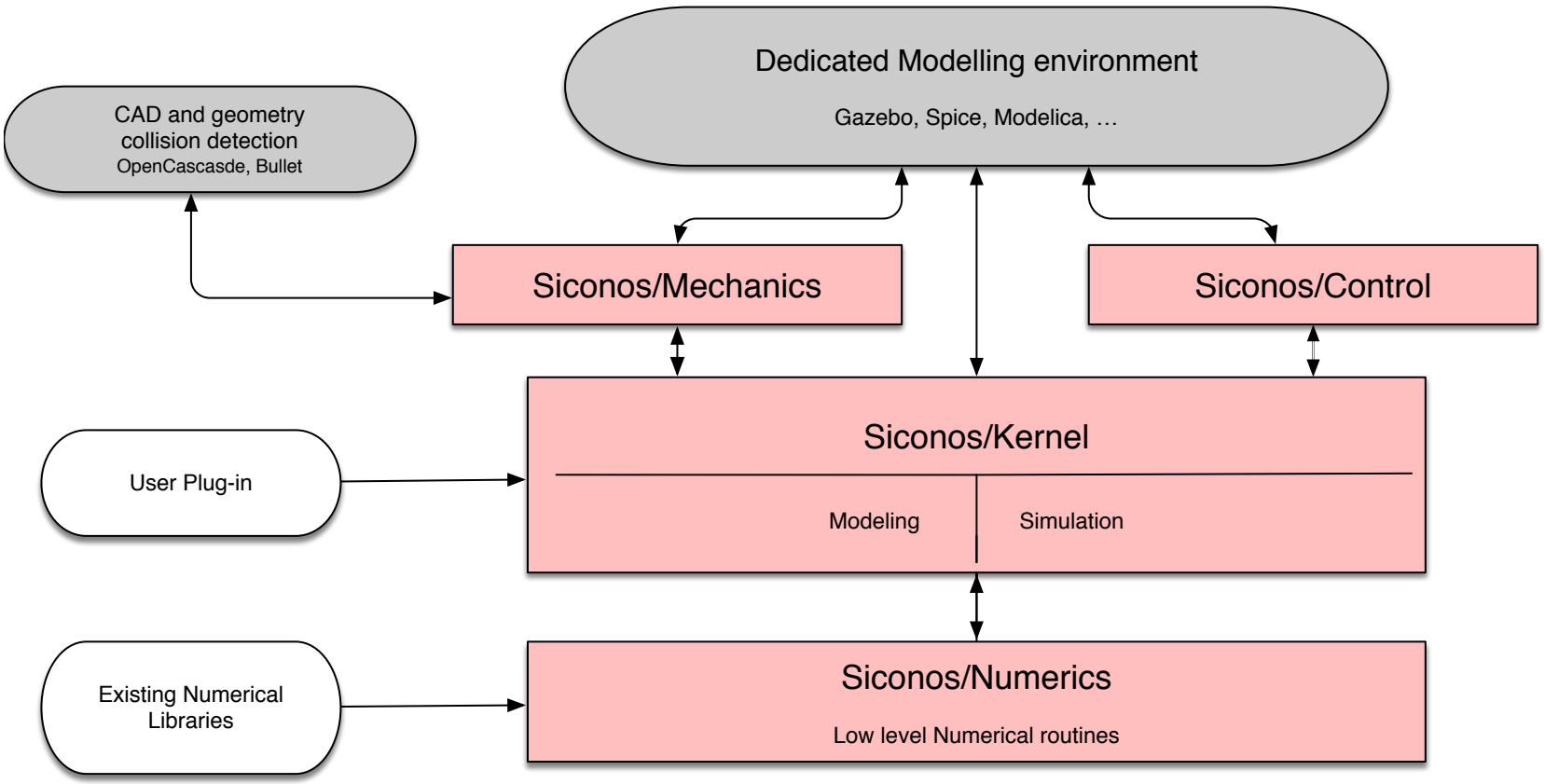
Main goal: provide with a general-purpose software.

Modeling, simulation, analysis and control of NSDS

- For research in nonsmooth dynamics:
  experiment new models, methods and test new algorithms and solvers
- For end-users: usable as a kernel toolkit (plug-in in dedicated modeling environment, dedicated extra toolboxes)
  - Computational mechanics: FEM Software for space discretisation (MFEM, Fenics, . . .)
  - Hybrid systems: Hybrid modeling Language (Modelica)
  - Electrical circuits (SPICE solver)

Siconos Overview

Siconos Modules



Siconos/numerics

Collection of C routines to solve problems for frictional contact:

- VI solvers: Fixed point, Extra-Gradient, Uzawa
- VI based projection/splitting algorithm: NSGS, PSOR
- Semismooth Newton methods
- Optimization based solvers. Panagiotopoulos, Tresca, SOCQP, ADMM
- Interior point methods, ...

Collection of routines for optimization and complementarity problems

- LCP solvers (iterative and pivoting (Lemke))
- Standard QP solvers (Projected Gradient (Calamai & Moré), Projected CG (Moré & Toraldo), active set technique)
- linear and nonlinear programming solvers.

ODE/DAE integrators

- Lsode suite with LSODAR (Hindmarsh, Alan C., (LLNL))
- HEM5 DAE solver (Hairer, Ernst, Université de Genève)

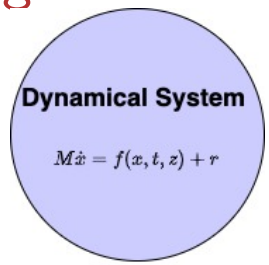
Siconos/Numerics

Implementation details

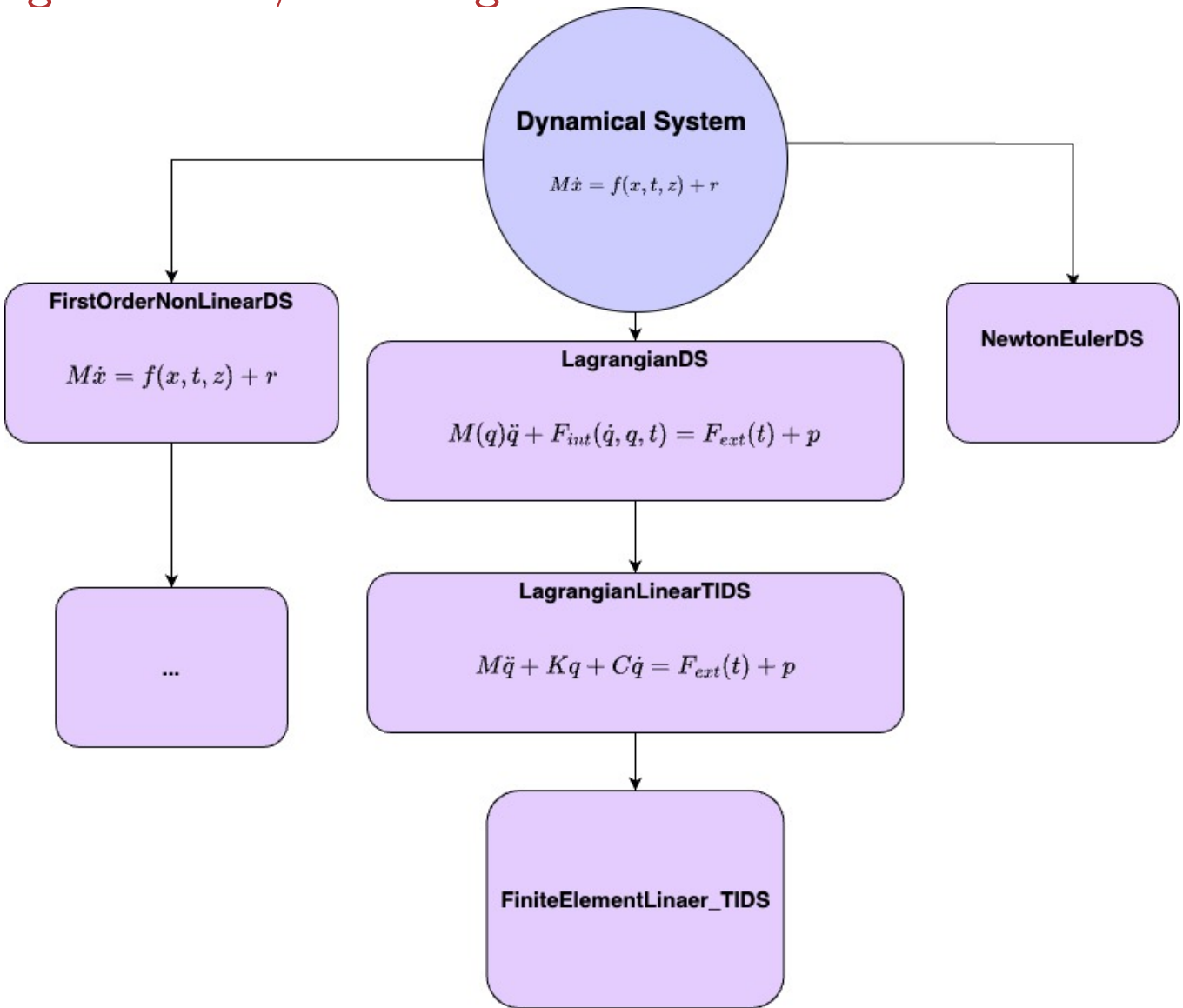
- Matrix format.
  - dense (column-major)
  - sparse matrices (triplet, CSR, CSC)
  - sparse block matrices
- Linear algebra libraries and solvers.
  - BLAS/LAPACK, MKL
  - MUMPS, SUPERLU, UMFPACK,
  - PETSc (in progress)
- Python interface (swig (pybind11 coming soon))
- Generic structure for problem, driver and options

```
int fc3d_driver(FrictionContactProblem* problem,
                double* reaction,
                double* velocity,
                SolverOptions* numerics_solver_options);
```

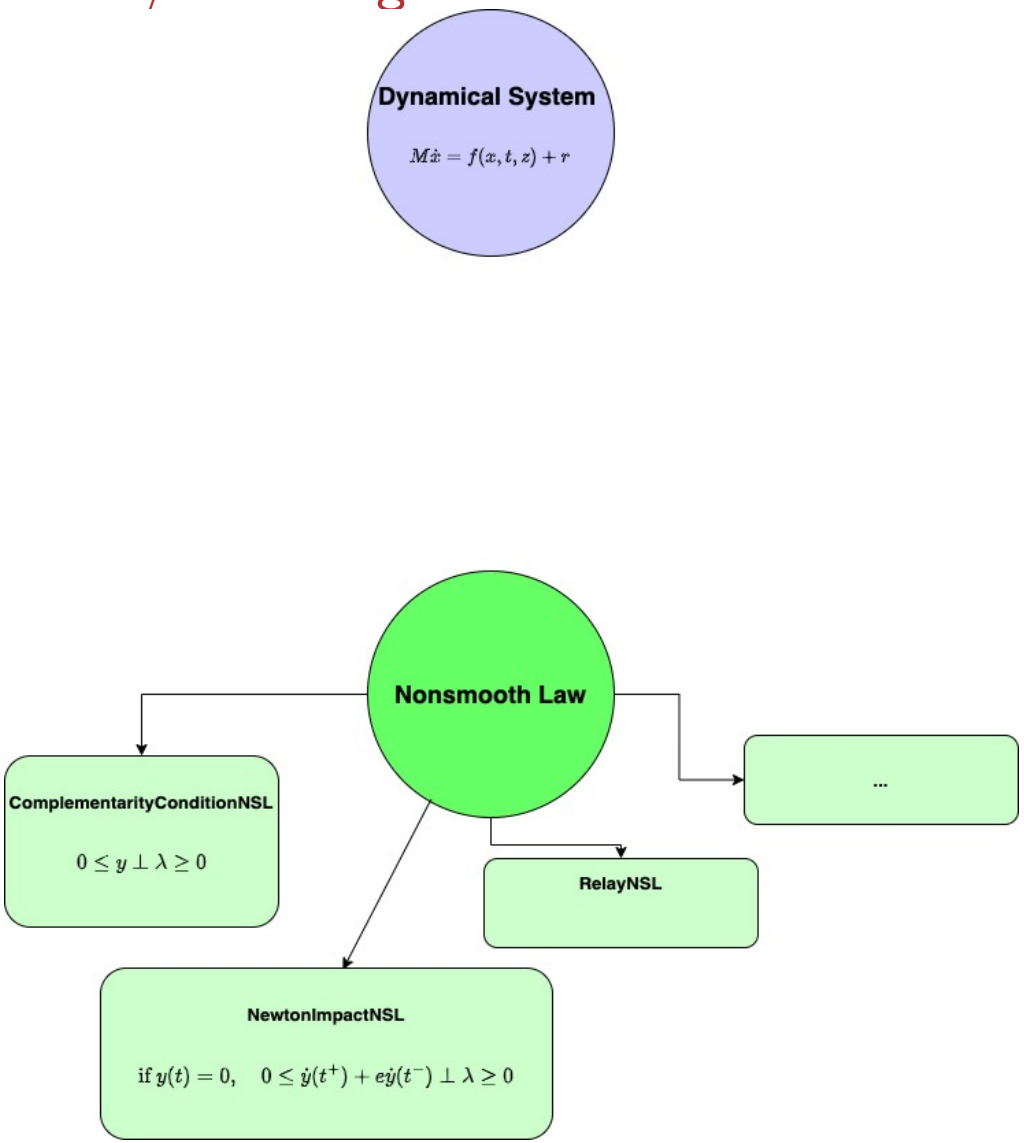
Siconos design - Kernel/Modeling



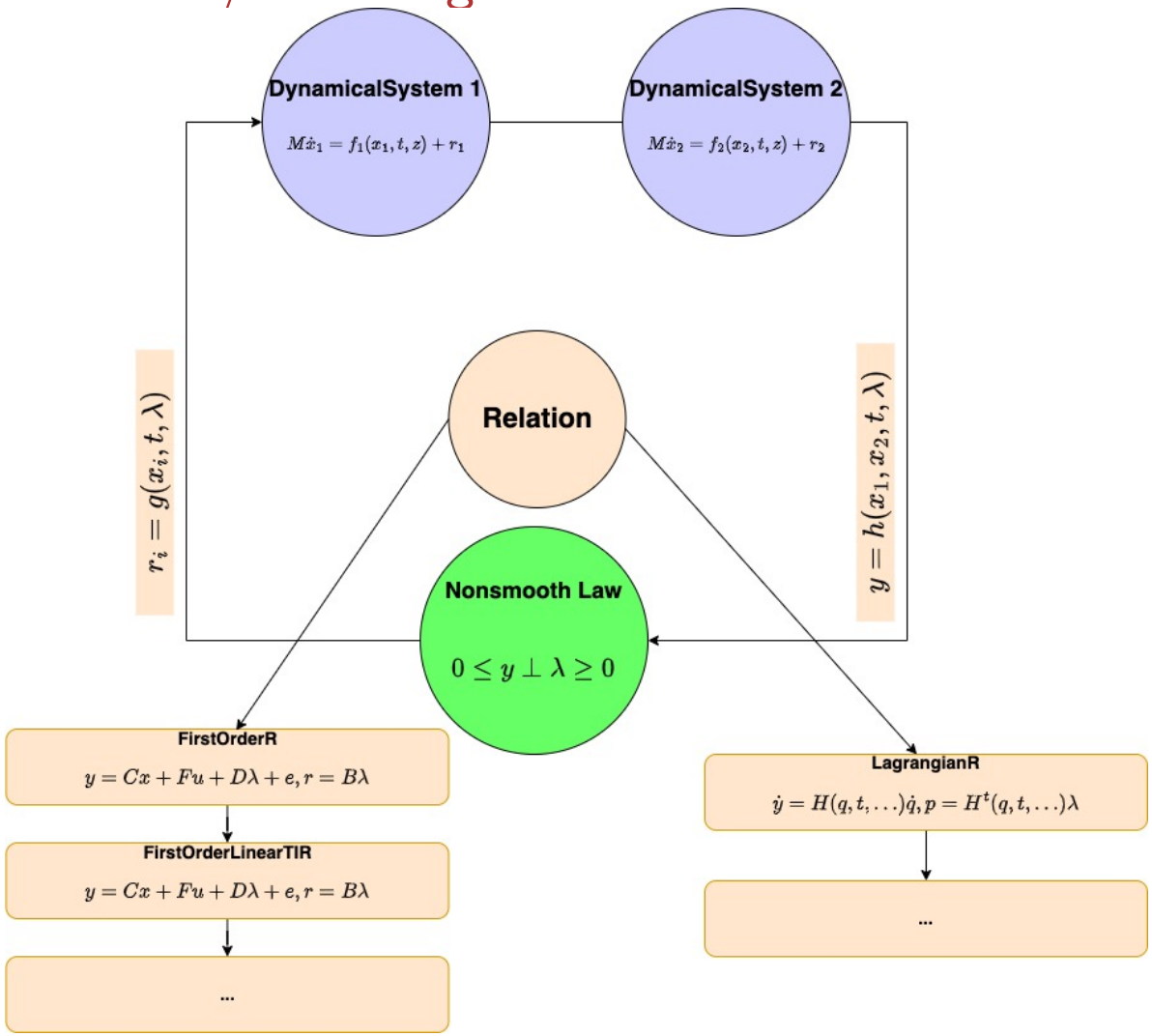
Siconos design - Kernel/Modeling



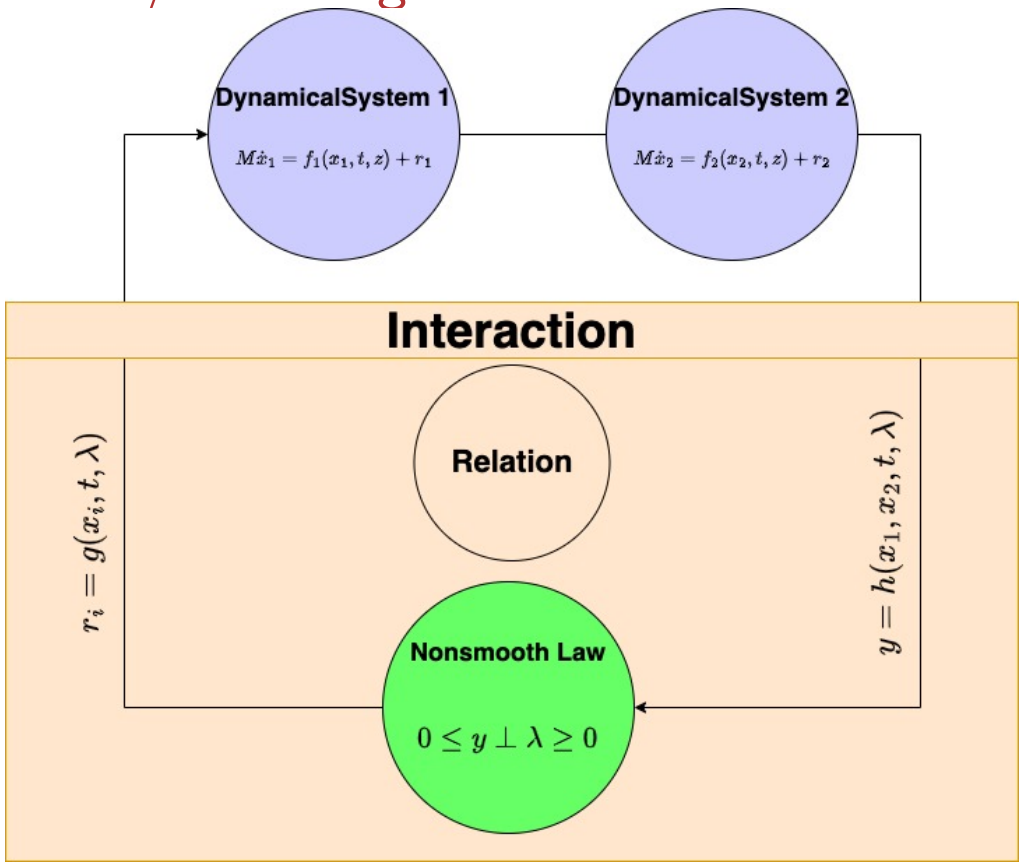
Siconos design - Kernel/Modeling



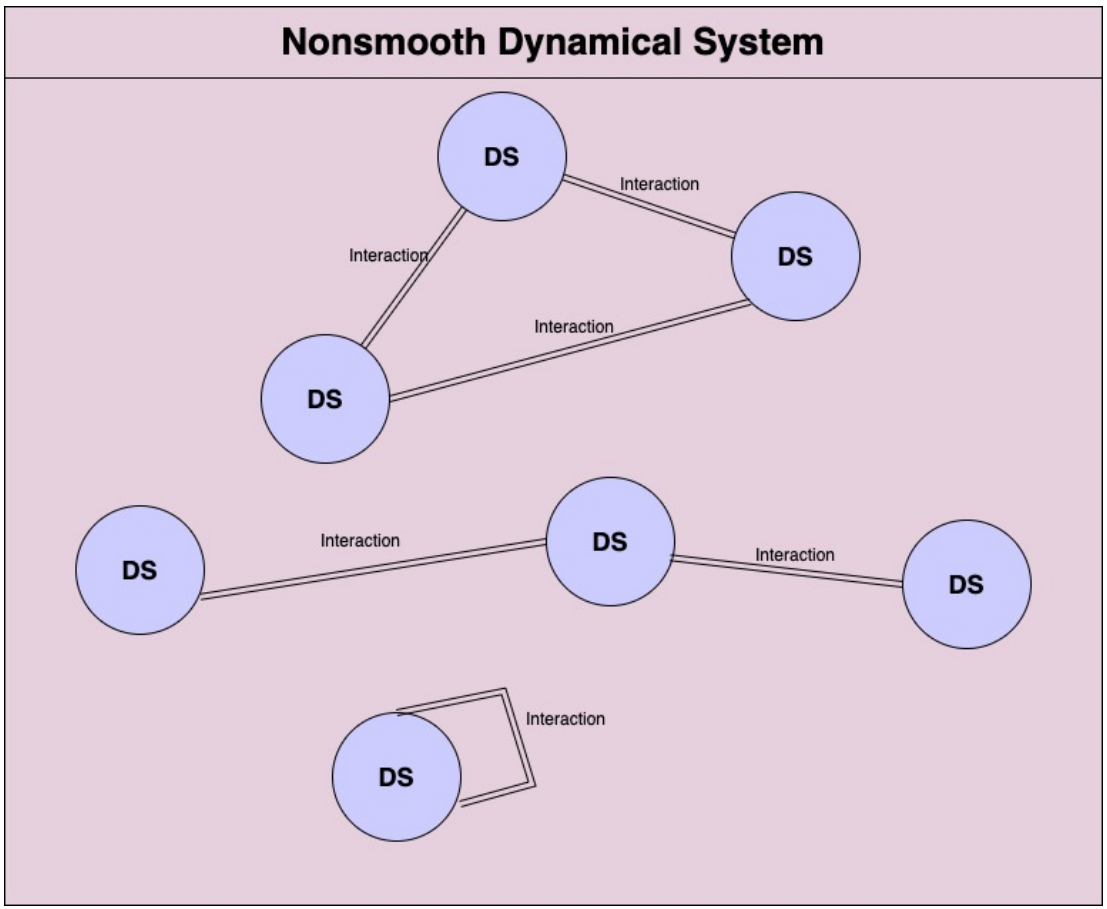
Siconos design - Kernel/Modeling



Siconos design - Kernel/Modeling



Siconos design - Kernel/Modeling



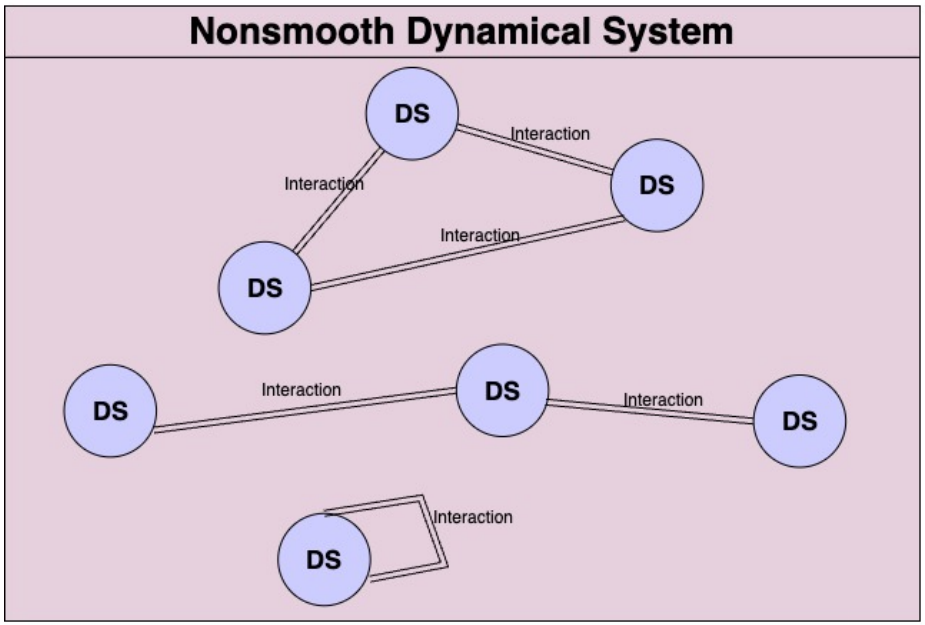
A nonsmooth dynamical system in Siconos as a directed graph.
Algorithms from graph theory (traversing, connected components, ...)

Python example of a nonsmooth dynamical system description

```
# definition and construction of a dynamical system
x = [1,0,0] # initial position
v = [0,0,0] # initial velocity
mass = eye(3)       # mass matrix
mass[2,2]=2./5 * r * r
ds = LagrangianLinearTIDS(x, v, mass)
# set external forces
weight = [-m * g, 0, 0]
ds.setFExtPtr(weight)
...
nonsmoothlaw = NewtonImpactNSL(0.9)
relation = LagrangianLinearTIR(H)
inter = Interaction(nslaw, relation)

nsds = NonSmoothDynamicalSystem(t0, T)
nsds.insertDynamicalSystem(ds)
# link the interaction and the dynamical system
nsds.link(inter, ds)
```

Siconos design - Kernel/Simulation

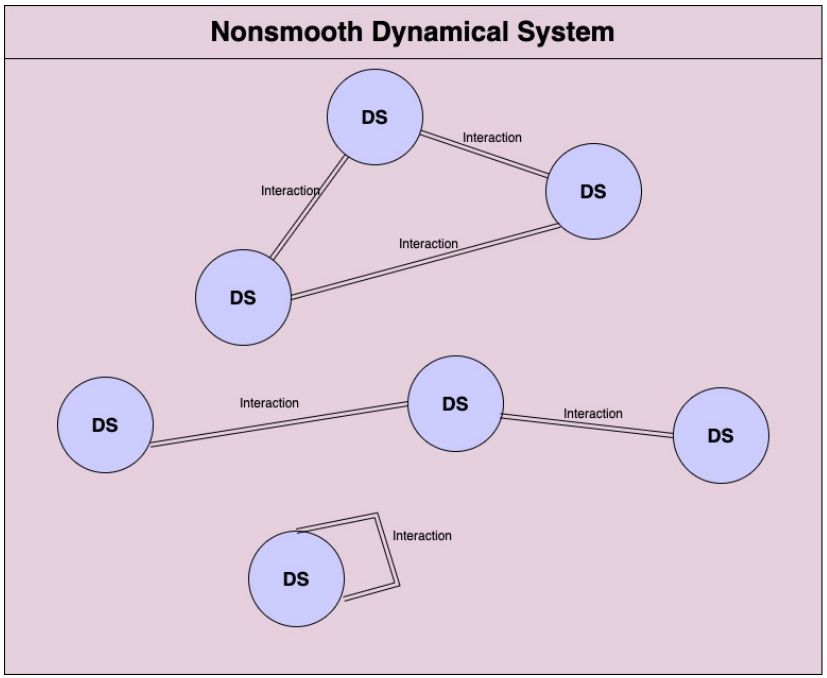


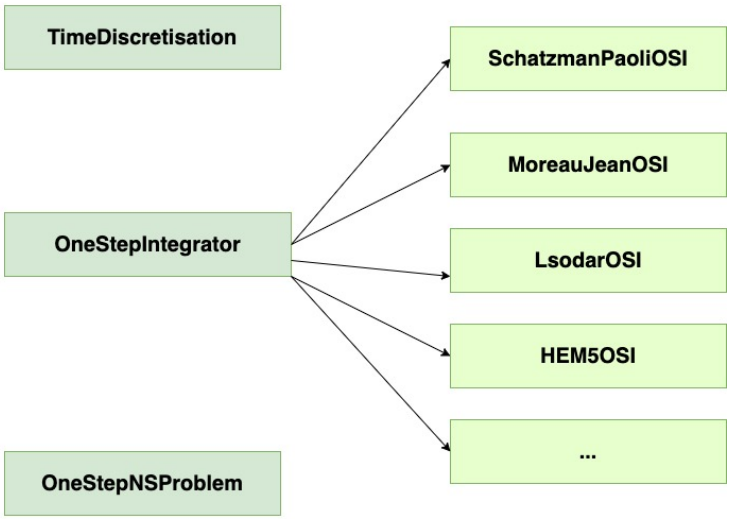
TimeDiscretisation

OneStepIntegrator

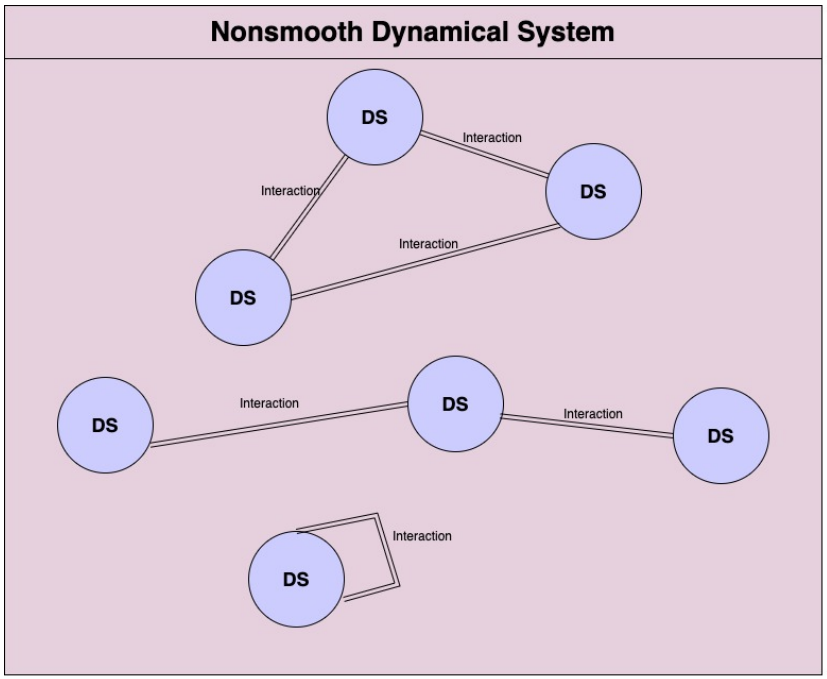
OneStepNSProblem

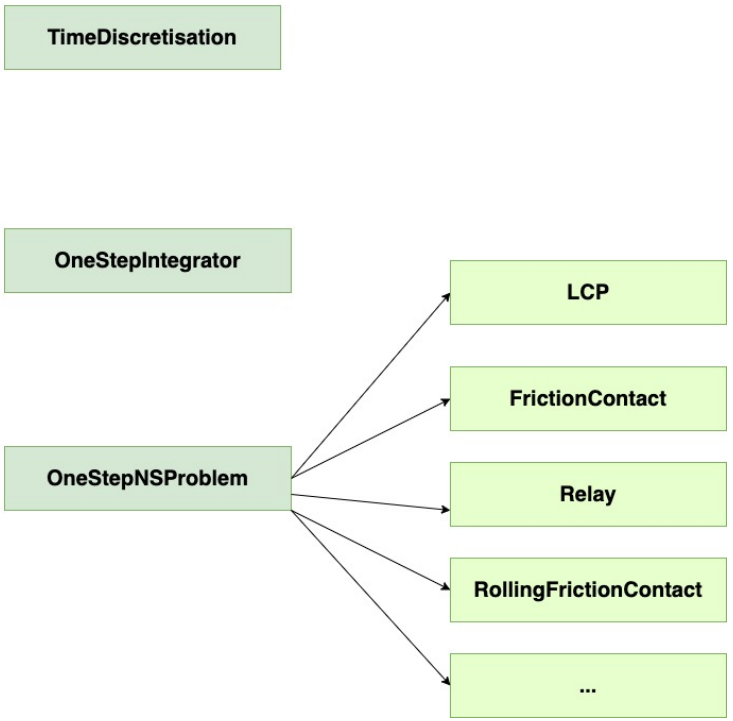
Siconos design - Kernel/Simulation



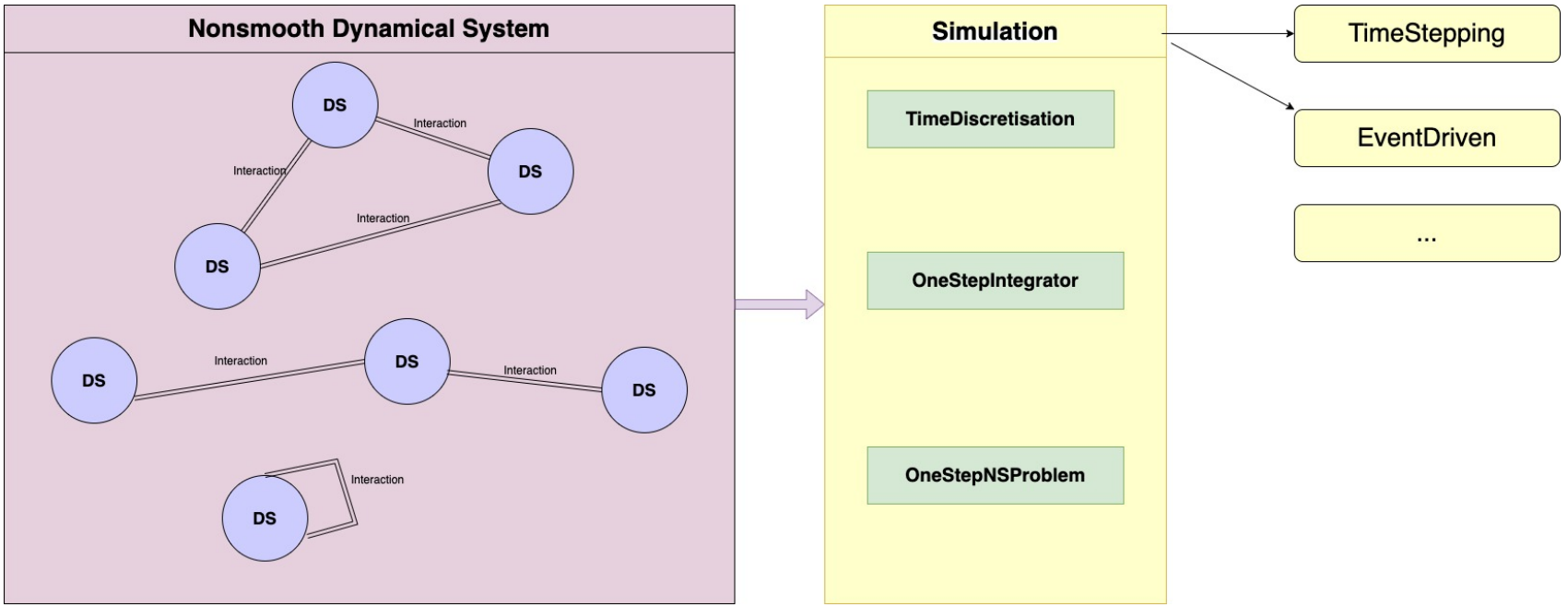


Siconos design - Kernel/Simulation





Siconos design - Kernel/Simulation



Siconos design - Kernel/Simulation

OneStepIntegrator:

- **MoreauJean**: Moreau–Jean Time-stepping integrator
- **SchatzmanPaoli**: Schatzman–Paoli Time-stepping integrator
- **D1MinusLinear**: Time–Discontinuous Galerkin method.
- **Lsodar**: Numerical integration scheme based on the Livermore Solver for Ordinary Differential Equations with root finding.
- **HEM5**: Half–explicit method of Brasey & Hairer for index-2 mechanical systems.

OnestepNSproblem: Numerical one step non smooth problem formulation and solver.

- **LCP** Linear Complementarity Problem

$$\begin{cases} w = Mz + q \\ 0 \leqslant w \perp z \geqslant 0 \end{cases}$$

- **FrictionContact** Two(three)-dimensional contact friction problem
- **QP** Quadratic programming problem

$$\begin{cases} \min \dfrac{1}{2} z^T Q z + z^T p \\ z \geqslant 0 \end{cases}$$

- **Relay**

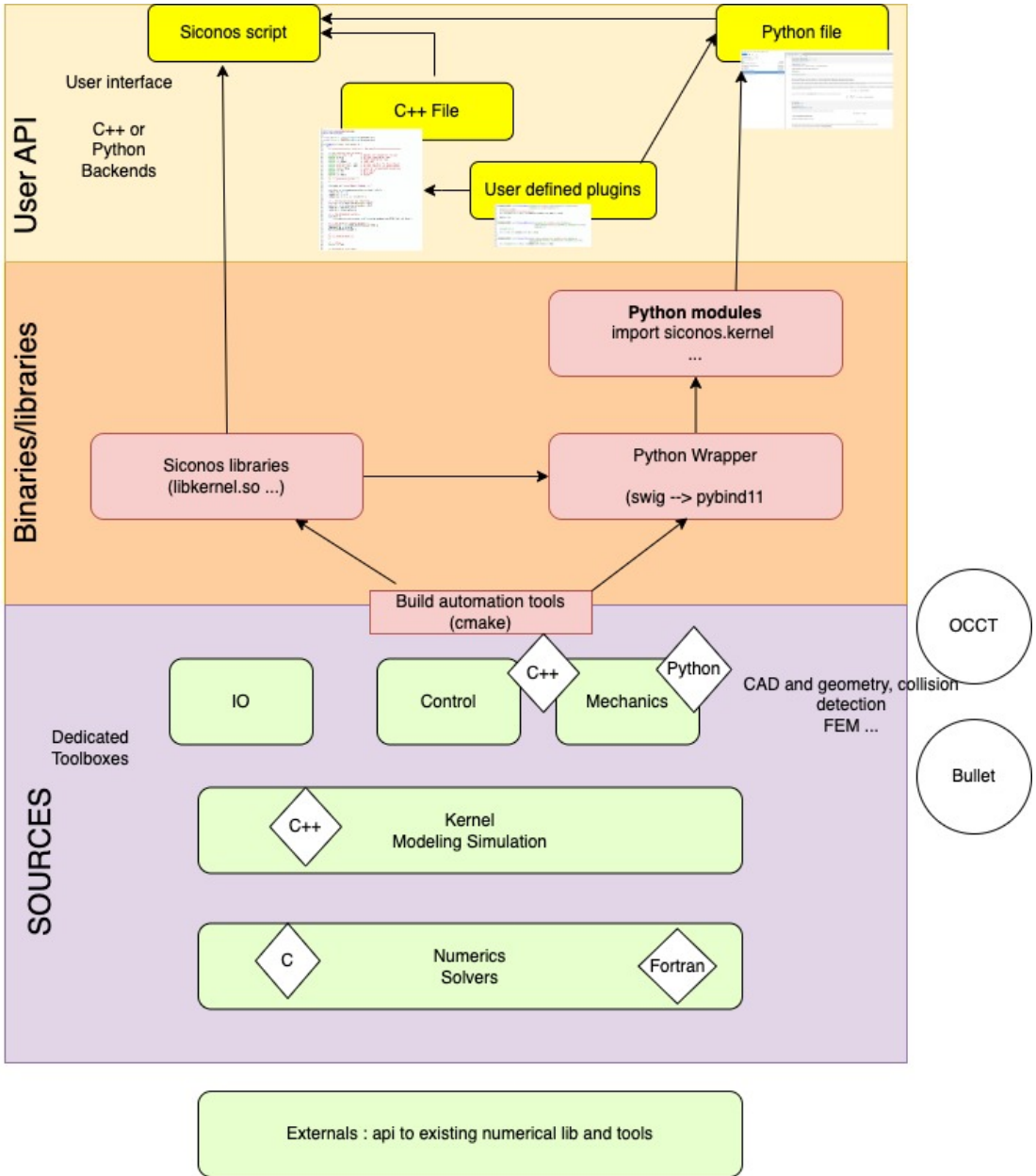
```
# (1) The time discretisation --
t = TimeDiscretisation(t0,h)

# (2) The integrator
OSI = MoreauJeanOSI(theta)

# (3) formulation of the one-step nonsmooth problem
osnspb = LCP()

# (4) ``Apply'' the simulation to the nsds, setup with (1) (2) (3)
simu = TimeStepping(nsds,t, OSI, osnspb)

# run the simulation
while simu.hasNextEvent():
    simu.computeOneStep()
    simu.nextStep()
```


Siconos script
Python file
User interface
C++ File
C++ or
Python
Backends
User defined plugins
User API
Python modules
import siconos.kernel
...
Binaries/libraries
Siconos libraries
(libkernel.so ...)
Python Wrapper
(swig --> pybind11
Build automation tools
(cmake)
OCCT
IO
Control
C++
Mechanics
Python
CAD and geometry, collision
detection
FEM ...
Dedicated
Toolboxes
Bullet
SOURCES
C++
Kernel
Modeling Simulation
C
Numerics
Solvers
Fortran
Externals : api to existing numerical lib and tools

Siconos today - Development process

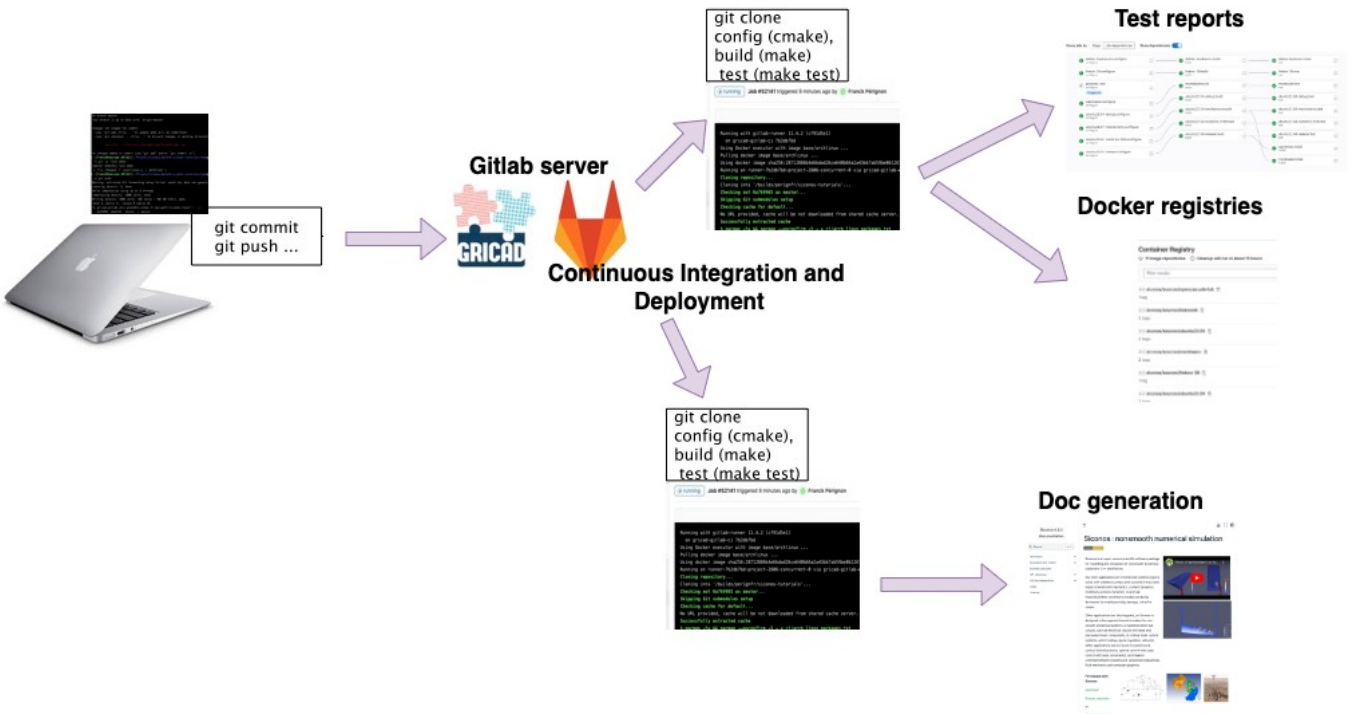
- Around 200000 lines of Open–source code in C++, C, Python, Fortran
- Code versioning since 2005 (cvs, svn, git), almost 10000 commits
- Use of build automation tools (cmake) to make compilation, build and deployment easier and automatic
- Extensive use of collaborative platforms
  - gitlab repository (a private gitlab repo for devel, a public gitlab and a mirror on github)
  - issues, merge/pull request, ...
- Strong efforts towards open science and reproducibility (non regression tests, Software Heritage archives)
- Modularity, portability concerns

Siconos today

- Continuous integration for tests and portability since 2008
- Continuous Deployment ("siconos-ready" docker images for users, binder, auto-generated documentation, ...)
- Documentation (API with doxygen/apidoc, install, user and developer manual with Sphinx, ...)

Siconos today

Automatic pipelines.



Siconos in the future

Towards a more efficient code

- Modern C++ refactoring (C++20)
- Siconos/numerics: extensive use of PETSc
- High level and portable parallelization in C++ (Kokkos, Raja, Thrust)
- Other development paradigms
  - Less Object Oriented Programming and more Functional Programming
  - Structures of Array (SoA) vs Array of Strucures (AoS)
  - Entity component system for memory management (MineCraft-like)

Help and Documentation

- Sphinx and Doxygen for automatic documentation
- Users and developers manuals
- siconos-tutorials : examples collection as templates (more than 250 examples).

Diffusion

- SICONOS is distributed under Apache 2.0 licence.
- colloborative developement framwork. (git, issues, pull-request)

  https://gricad-gitlab.univ-grenoble-alpes.fr/nonsmooth/siconos
- Visit the website for more info https://siconos.org

Use and Contribute !!

Thank you for your attention.