Nonsmooth dynamical systems. Numerical Time–integration schemes and the Siconos platform

Vincent Acary

vincent.acary@inrialpes.fr
INRIA Rhône–Alpes

séminaire LAMCOS, Insa Lyon, April 10, 2008
Mechanical systems with contact, impact and friction
Simulation of Circuit breakers (INRIA/Schneider Electric)
Applications

Mechanical systems with contact, impact and friction

Bipedal Robot INRIA BIPOP
Applications

Mechanical systems with contact, impact and friction

Stack of beads with perturbation
Applications

Mechanical systems with contact, impact and friction
FEM models with contact, friction cohesion, etc...

Joint work with Y. Monerie, IRSN.
Applications

Mechanical systems with contact, impact and friction

Divided Materials and Masonry
Introduction & Motivations

Lagrangian dynamical systems with unilateral constraints and friction

Numerical time–integration schemes

Adaptive time-step size time–stepping scheme

Time–stepping schemes of any order

Splitting based Time–stepping schemes

Conclusions & Perspectives

The Siconos Platform
The smooth multibody dynamics

Definition (Smooth multibody dynamics)

\[ \begin{align*}
M(q) \frac{dv}{dt} + F(t, q, v) &= 0, \\
v &= \dot{q}
\end{align*} \]  

(1)

where

\[ F(t, q, v) = N(q, v) + F_{\text{int}}(t, q, v) - F_{\text{ext}}(t) \]

Definition (Boundary conditions)

- Initial Value Problem (IVP):
  \[ t_0 \in \mathbb{R}, \quad q(t_0) = q_0 \in \mathbb{R}^n, \quad v(t_0) = v_0 \in \mathbb{R}^n, \]  
  (2)

- Boundary Value Problem (BVP):
  \[ (t_0, T) \in \mathbb{R} \times \mathbb{R}, \quad \Gamma(q(t_0), v(t_0), q(T), v(T)) = 0 \]  
  (3)
Perfect unilateral constraints

Unilateral constraints

- Finite set of \( \nu \) unilateral constraints on the generalized coordinates:

\[
g(q, t) = [g_\alpha(q, t) \geq 0, \quad \alpha \in \{1 \ldots \nu\}]^T. \tag{4}
\]

- Admissible set \( \mathcal{C}(t) \)

\[
\mathcal{C}(t) = \{q \in \mathcal{M}(t), g_\alpha(q, t) \geq 0, \alpha \in \{1 \ldots \nu\}\}. \tag{5}
\]

Normal cone to \( \mathcal{C}(t) \)

\[
N_{\mathcal{C}(t)}(q(t)) = \left\{ y \in \mathbb{R}^n, y = -\sum_\alpha \lambda_\alpha \nabla g_\alpha(q, t), \quad \lambda_\alpha \geq 0, \quad \lambda_\alpha g_\alpha(q, t) = 0 \right\}. \tag{6}
\]
Unilateral constraints as an inclusion

Definition (Perfect unilateral constraints on the smooth dynamics)

Introduction of the multipliers $\mu \in \mathbb{R}^m$

$$\begin{cases} M(q)\frac{dv}{dt} + F(t, q, v) = r = \nabla_q^T h(q, t) \lambda \\ -r \in N_{C(t)}(q(t)) \end{cases}$$

(7)

where $r = \nabla_q^T g(q, t) \lambda$ generalized forces or generalized reactions due to the constraints.

Remark

- The unilateral constraints are said to be perfect due to the normality condition.
- Notion of normal cones can be extended to more general sets. see (Clarke, 1975, 1983 ; ?)
- The right hand side is neither bounded (and then nor compact).
- The inclusion and the constraints concern the second order time derivative of $q$.

⇒ Standard Analysis of DI does no longer apply.
Non Smooth Lagrangian Dynamics

Fundamental assumptions.

- The velocity \( v = \dot{q} \) is of Bounded Variations (B.V).
  - The equation are written in terms of a right continuous B.V. (R.C.B.V.) function, \( v^+ \) such that
    \[
    v^+ = \dot{q}^+
    \]  
  (8)

- \( q \) is related to this velocity by
  \[
  q(t) = q(t_0) + \int_{t_0}^{t} v^+(t) \, dt
  \]  
  (9)

- The acceleration, ( \( \ddot{q} \) in the usual sense) is hence a differential measure \( dv \) associated with \( v \) such that
  \[
  dv([a, b]) = \int_{[a, b]} dv = v^+(b) - v^+(a)
  \]  
  (10)
Non Smooth Lagrangian Dynamics

Definition (Non Smooth Lagrangian Dynamics)

\[
\begin{cases}
M(q)dv + F(t, q, v^+)dt = dr \\
v^+ = \dot{q}^+
\end{cases}
\]

where \(dr\) is the reaction measure and \(dt\) is the Lebesgue measure.

Remarks

- The non smooth Dynamics contains the impact equations and the smooth evolution in a single equation.
- The formulation allows one to take into account very complex behaviors, especially, finite accumulation (Zeno-state).
- This formulation is sound from a mathematical Analysis point of view.

References

Non Smooth Lagrangian Dynamics

Decomposition of measure

\[
\begin{align*}
\text{dv} &= \gamma \, dt + (v^+ - v^-) \, d\nu + \text{dv}_S \\
\text{dr} &= f \, dt + p \, d\nu + \text{dr}_S
\end{align*}
\]  

(12)

where

- \( \gamma = \ddot{q} \) is the acceleration defined in the usual sense.
- \( f \) is the Lebesgue measurable force,
- \( v^+ - v^- \) is the difference between the right continuous and the left continuous functions associated with the B.V. function \( v = \dot{q} \)
- \( d\nu \) is a purely atomic measure concentrated at the time \( t_i \) of discontinuities of \( v \), i.e. where \( (v^+ - v^-) \neq 0 \), i.e. \( d\nu = \sum_i \delta_{t_i} \)
- \( p \) is the purely atomic impact percussions such that \( pd\nu = \sum_i p_i \delta_{t_i} \)
- \( \text{dv}_S \) and \( \text{dr}_S \) are singular measures with the respect to \( dt + d\eta \).
Impact equations and Smooth Lagrangian dynamics

Substituting the decomposition of measures into the non smooth Lagrangian Dynamics, one obtains

**Definition (Impact equations)**

\[ M(q)(v^+ - v^-)d\nu = pd\nu, \quad (13) \]

or

\[ M(q(t_i))(v^+(t_i) - v^-(t_i)) = p_i, \quad (14) \]

**Definition (Smooth Dynamics between impacts)**

\[ M(q)\gamma dt + F(t, q, v)dt = fdt \quad (15) \]

or

\[ M(q)\gamma^+ + F(t, q, v^+) = f^+ [dt - a.e.] \quad (16) \]
The Moreau’s sweeping process of second order

Definition (Moreau (1983, 1988))

A key stone of this formulation is the inclusion in terms of velocity. Indeed, the inclusion (7) is “replaced” by the following inclusion

\[
\begin{align*}
M(q)dv + F(t, q, v^+) dt &= dr \\
v^+ &= \dot{q}^+ \\
-dr &\in N_{T_C}(q)(v^+) 
\end{align*}
\] (17)

Comments

This formulation provides a common framework for the non smooth dynamics containing inelastic impacts without decomposition.

⇒ Foundation for the time-stepping approaches.
The Moreau’s sweeping process of second order

Comments

- The inclusion concerns measures. Therefore, it is necessary to define what is the inclusion of a measure into a cone.
- The inclusion in terms of velocity $\nu^+$ rather than of the coordinates $q$.

Interpretation

- Inclusion of measure, $-dr \in K$
  - Case $dr = r' dt = f dt$. 
    
    \[-f \in K\]  
    \[\text{(18)}\]
  - Case $dr = p_i \delta_i$.
    
    \[-p_i \in K\]  
    \[\text{(19)}\]
- Inclusion in terms of the velocity. Viability Lemma
  If $q(t_0) \in C(t_0)$, then
  \[\nu^+ \in T_C(q), \ t \geq t_0 \Rightarrow q(t) \in C(t), \ t \geq t_0\]

$\Rightarrow$ The unilateral constraints on $q$ are satisfied. The equivalence needs at least an impact inelastic rule.
The Moreau’s sweeping process of second order

The Newton-Moreau impact rule

\[- dr \in N_{T_C(q(t))}(v^+(t) + ev^-(t))\]  

(20)

where \( e \) is a coefficient of restitution.
The Moreau’s sweeping process of second order

The case of $C$ is finitely represented

$$C = \{ q \in M(t), g_\alpha(q) \geq 0, \alpha \in \{1 \ldots \nu\} \}. \quad (21)$$

Decomposition of $dr$ and $v^+$ onto the tangent and the normal cone.

$$dr = \sum_\alpha \nabla_q^T g_\alpha(q) \ d\lambda_\alpha \quad (22)$$

$$U^+_\alpha = \nabla_q g_\alpha(q) \ v^+, \alpha \in \{1 \ldots \nu\} \quad (23)$$

Complementarity formulation (under constraints qualification condition)

$$-d\lambda_\alpha \in N_{\mathbb{R}^+}(g_\alpha)(U^+_\alpha) \iff \text{if } g_\alpha(q) \leq 0, \text{ then } 0 \leq U^+_\alpha \perp d\lambda_\alpha \geq 0 \quad (24)$$

The case of $C$ is $\mathbb{R}^+$

$$-dr \in N_C(q) \iff 0 \leq q \perp dr \geq 0 \quad (25)$$

is replaced by

$$-dr \in N_T_C(q)(v^+) \iff \text{if } q \leq 0, \text{ then } 0 \leq v^+ \perp dr \geq 0 \quad (26)$$
Coulomb’s friction

\[ \vec{r}_n \parallel \vec{t}_s \parallel \vec{r}_T \]

**Figure:** Coulomb’s friction. The sliding case.

**Definition (Coulomb’s friction)**

Coulomb’s friction says the following. If \( g(q) = 0 \) then:

\[
\begin{cases}
\text{If } U_T(t) = 0 & \text{then } R \in \mathbf{C} \\
\text{If } U_T(t) \neq 0 & \text{then } ||R_T(t)|| = \mu |R_N| \text{ and there exists a scalar } a \geq 0 \\
& \text{such that } R_T(t) = -aU_T(t)
\end{cases}
\]

where \( \mathbf{C} = \{ R, ||R_T(t)|| \leq \mu |R_N| \} \) is the Coulomb friction cone

\[(27)\]
Coulomb’s friction

Definition (Coulomb’s friction as an inclusion into a disk)
Let us introduce the following inclusion (Moreau, 1988), using the indicator function \( \psi_D(\cdot) \):

\[-U_T \in \partial \psi_D(R_T)\]  \hspace{1cm} (28)

where \( D = \{ R_T, ||R_T(t)|| \leq \mu |R_N| \} \) is the Coulomb friction disk.

Definition (Coulomb’s friction as a variational inequality (VI))
Then (28) appears to be equivalent to

\[
\begin{cases}
R_T \in D \\
\langle U_T, z - R_T \rangle \geq 0 \text{ for all } z \in D
\end{cases}
\]  \hspace{1cm} (29)

and to

\[R_T = \text{proj}_D[R_T - \rho U_T], \text{ for all } \rho > 0\]  \hspace{1cm} (30)
Definition (Coulomb’s Friction as a Second–Order Cone Complementarity Problem)

Let us introduce the modified velocity \( \hat{U} \) defined by

\[
\hat{U} = [U_N + \mu ||U_T||, U_T]^T. \tag{31}
\]

This notation provides us with a synthetic form of the Coulomb friction as

\[
-\hat{U} \in \partial \psi_C(R), \tag{32}
\]

or

\[
C^* \ni \hat{U} \perp R \in C. \tag{33}
\]

where \( C^* = \{ v \in \mathbb{R}^n \mid r^Tv \geq 0, \forall r \in C \} \) is the dual cone.
Coulomb’s friction

Figure: Coulomb’s friction and the modified velocity \( \hat{U} \). The sliding case.
Introduction & Motivations

Lagrangian dynamical systems with unilateral constraints and friction

Numerical time–integration schemes

Adaptive time-step size time–stepping scheme

Time–stepping schemes of any order

Splitting based Time–stepping schemes

Conclusions & Perspectives

The Siconos Platform
State-of-the-art

Numerical time-integration methods for Nonsmooth Multibody systems (NSMBS):

Nonsmooth event capturing methods (Time-stepping methods)

⊕ robust, stable and proof of convergence
⊕ low kinematic level for the constraints
⊕ able to deal with finite accumulation
⊕ very low order of accuracy even in free flight motions

Nonsmooth event tracking methods (Event-driven methods)

⊕ high level integration of free flight motions
⊕ no proof of convergence
⊕ sensibility to numerical thresholds
⊕ reformulation of constraints at higher kinematic levels.
⊕ unable to deal with finite accumulation
Objectives & means

Objectives

Design nonsmooth event capturing methods with

- same properties as standard methods (robustness, accumulation, ...)
- Higher resolution (ratio error/computational cost)
- Higher order of accuracy

Means

1. Adaptive time-step size and order strategies for standard methods
2. Mixed integrators with rough pre-detection of events
3. Splitting strategies
4. Ad hoc detection of discontinuity and order of discontinuity methods.
NonSmooth Multibody Systems (NSMBS)

General definition

\[
\begin{aligned}
M(q)\dot{v} &= F(t, q, v) + G(t, q)\lambda \\
\dot{q} &= v \\
w &= g(t, q, v) \\
0 &\in S(w, \lambda, t) + T(w, \lambda, t) \\
v^+ &= F(v^-, q, t)
\end{aligned}
\]  \tag{34}

\begin{itemize}
  \item S : \mathbb{R}^{m \times m} \times \mathbb{R} \mapsto \mathbb{R}^{m \times m} \text{ continuously differentiable mapping}
  \item T : \mathbb{R}^{m \times m} \times \mathbb{R} \rightsquigarrow \mathbb{R}^{m \times m} \text{ multivalued mapping with a closed graph.}
\end{itemize}

With scleronomous holonomic perfect unilateral constraints

\[
\begin{aligned}
M(q)\dot{v} &= F(t, q, v) + G(q)\lambda \\
\dot{q} &= v \\
0 &\leq y = g(q) \perp \lambda \geq 0 \\
v^+ &= F(v^-, q, t)
\end{aligned}
\]  \tag{35}

where \( G(q) = \nabla g(q) \)
NonSmooth Multibody Systems (NSMBS)

Academic examples I

(a) Bouncing ball example
(b) Linear Oscillator example

Figure: Academic test examples with analytical solutions
NonSmooth Multibody Systems (NSMBS)

Figure: Analytical solutions. Bouncing ball example]
NonSmooth Multibody Systems (NSMBS)

Figure: Analytical solutions. Linear Oscillator
NonSmooth Multibody Systems (NSMBS)

Academic examples II

(a) N Bouncing balls example

Figure: Academic test examples
Moreau’s Time stepping scheme

Principle

\[
\begin{aligned}
M(q_{k+\theta})(v_{k+1} - v_k) - h\ddot{F}_{k+\theta} &= G(q_{k+\theta})P_{k+1}, \\
q_{k+1} &= q_k + hv_{k+\theta}, \\
U_{k+1} &= G^T(q_{k+\theta})v_{k+1} \\
-P_{k+1} &\in \partial\psi_{T_{\mathbb{R}^m}^+}(\tilde{y}_{k+\gamma})(U_{k+1} + eU_k), \\
\tilde{y}_{k+\gamma} &= y_k + h\gamma U_k, \quad \gamma \in [0, 1].
\end{aligned}
\]

with \( \theta \in [0, 1], \gamma \geq 0 \) and \( x_{k+\alpha} = (1 - \alpha)x_{k+1} + \alpha x_k \) and \( \tilde{y}_{k+\gamma} \) is a prediction of the constraints.

Properties

- Convergence results for one constraints
- Convergence results for multiple constraints problems with acute kinetic angles
- No theoretical proofs of order
Schatzman–Paoli’s Time stepping scheme

Principle

\[
\begin{align*}
\mathbf{M}(q_k+1)(q_{k+1} - 2q_k + q_{k-1}) - h^2 F(t_{k+\theta}, q_{k+\theta}, \nu_{k+\theta}) &= p_{k+1} (37a) \\
v_{k+1} &= \frac{q_{k+1} - q_{k-1}}{2h}, \quad (37b) \\
-p_{k+1} &\in N_K \left( \frac{q_{k+1} + eq_{k-1}}{1 + e} \right), \quad (37c)
\end{align*}
\]

where \( N_K \) defined the normal cone to \( K \).

For \( K = \{ q \in \mathbb{R}^n, y = g(q) \geq 0 \} \)

\[
0 \leq g \left( \frac{q_{k+1} + eq_{k-1}}{1 + e} \right) \perp \nabla g \left( \frac{q_{k+1} + eq_{k-1}}{1 + e} \right) P_{k+1} \geq 0 \quad (38)
\]

Properties

- Convergence results for one constraints
- Convergence results for multiple constraints problems with acute kinetic angles
- No theoretical proof of order
Measuring error and convergence

Convergence in the sense of filled-in graph (Moreau (1978))

\[ \text{gr}^*(f) = \{(t, x) \in [0, T] \times \mathbb{R}^n, 0 \leq t \leq T \text{ and } x \in [f(t^-), f(t^+)]\} \] (39)

Such graphs are closed bounded subsets of \([0, T] \times \mathbb{R}^n\), hence, we can use the Hausdorff distance between two such sets with a suitable metric:

\[ d((t, x), (s, y)) = \max\{|t - s|, \|x - y\|\}. \] (40)

Defining the excess of separation between two graphs by

\[ e(\text{gr}^*(f), \text{gr}^*(g)) = \sup_{(t, x) \in \text{gr}^*(f)} \inf_{(s, y) \in \text{gr}^*(g)} d((t, x), (s, y)), \] (41)

the Hausdorff distance between two filled-in graphs \(h^*\) is defined by

\[ h^*(\text{gr}^*(f), \text{gr}^*(g)) = \max\{e(\text{gr}^*(f), \text{gr}^*(g)), e(\text{gr}^*(g), \text{gr}^*(f))\}. \] (42)
Measuring error and convergence

An equivalent grid-function norm to the function norm in $L_1$

$$\|e\|_1 = h \sum_{i=0}^{N} |f_i - f(t_i)| \quad (43)$$

In the same way, the $p$ - norm can be defined by

$$\|e\|_p = \left( h \sum_{i=0}^{N} |f_i - f(t_i)|^p \right)^{1/p} \quad (44)$$

The computation of this two last norm is easier to implement for piecewise continuous analytical function than the Hausdorff distance.

**Global order of convergence.**

**Definition**

A one-step time–integration scheme is of order $q$ for a given norm $\| \cdot \|$ if there exists a constant $C$ such that

$$\|e\| = Ch^q + O(h^{q+1}) \quad (45)$$
Empirical order of convergence. Moreau’s time-stepping scheme

Figure: Empirical order of convergence of the Moreau’s time-stepping scheme.
Empirical order of convergence. Moreau’s time-stepping scheme

Figure: Empirical order of convergence of the Moreau’s time-stepping scheme.
Empirical order of convergence. Schatzman–Paoli’s time–stepping scheme

Figure: Empirical order of convergence of the Schatzman-Paoli’s time-stepping scheme.
Empirical order of convergence. Schatzman–Paoli’s time–stepping scheme

Figure: Empirical order of convergence of the Schatzman-Paoli’s time-stepping scheme.
Introduction & Motivations

Lagrangian dynamical systems with unilateral constraints and friction

Numerical time–integration schemes

Adaptive time-step size time–stepping scheme

Time–stepping schemes of any order

Splitting based Time–stepping schemes

Conclusions & Perspectives

The Siconos Platform
Local error estimates for the Moreau’s time-stepping

\[ e = x(t_k + h) - x_{k+1} = \begin{bmatrix} e_v \\ e_q \end{bmatrix} = \begin{bmatrix} \nu^+(t_k + h) - \nu_{k+1} \\ q(t_k + h) - q_{k+1} \end{bmatrix} \] (46)
One impact at time $t_\ast \in (t_k, t_{k+1}]$

Assumption

$$di = p \delta_{t_\ast}, \text{ or equivalently } dl = P \delta_{t_\ast}, \text{ with } P = G(t_\ast)p.$$  \hfill (47)

Notation

$$\mathcal{I} = \{\alpha, \alpha \in \{1..m}\}$$  \hfill (48)

$$\mathcal{I}_\ast = \{\alpha \in \mathcal{I}, P^\alpha \geq 0, U^{\alpha,+}(t_\ast) - U^{\alpha,-}(t_\ast) = -(1 + e)U^{\alpha,-}(t_\ast)\}$$  \hfill (49)

$$\mathcal{I}_p = \{\alpha \in \mathcal{I}, P^\alpha_{k+1} \geq 0, U^{\alpha}_{k+1} - U^{\alpha}_k = -(1 + e)U^{\alpha}_k\}$$  \hfill (50)

Lemma

Let us assume that we have only one elastic impact at time $t_\ast \in (t_k, t_{k+1}]$ without persistent contact, i.e., $di = p \delta_{t_\ast}$.

1. If $\mathcal{I}_\ast = \mathcal{I}_p$, then the local order of consistency of the scheme is given by

$$e_v = K_v h + O(h^2)$$

$$e_q = K_q h + O(h^2)$$  \hfill (51)

2. If $\mathcal{I}_\ast \neq \mathcal{I}_p$, then the local order of consistency of the scheme is given by

$$e_v = K_v + O(h)$$

$$e_q = K_q h + O(h^2)$$  \hfill (52)
Smooth Lagrange multiplier in persistent contact without impact in \((t_k, t_{k+1}]\)

Assumption

\[ di = \lambda(t) dt, \quad (53) \]

or equivalently

\[ dl = \Lambda(t) dt, \text{ with } \Lambda(t) = G(t) \lambda(t). \quad (54) \]

Notation

\[ \mathcal{I}_\Lambda(t) = \{ \alpha \in \mathcal{I}, \Lambda^\alpha(t) \geq 0, U^\alpha,+(t) = U^\alpha,-(t) = 0 \} \quad (55) \]

\[ \mathcal{I}_\Lambda,k+1 = \{ \alpha \in \mathcal{I}, \Lambda^\alpha_{k+1} \geq 0, U^\alpha_{k+1} = U^\alpha_k = 0 \} \quad (56) \]

Lemma

Assuming that \( \mathcal{I}_\Lambda(t) = \mathcal{I}_\Lambda,k+1 \) for all \( t \in (t_k, t_{k+1}] \). The local order of consistency of the scheme is one that is

\[ e_v = Kh^2 + \mathcal{O}(h^3) \]

\[ e_q = K_q h^2 + \mathcal{O}(h^3) \quad (57) \]
Local error estimates for the Moreau’s time–stepping

Other cases

- **One impact and smooth Lagrange multiplier** The same result holds as in the first Lemma.

- **Losing contact event (take–off) without impact** The order of the time–integration scheme depends on the regularity of the contact forces (at least continuous).

- **Finite accumulation** The order of the time–integration should be at least 0. Idea of the proof: use the fact that the velocity vanishes and is of bounded variations.
Practical error estimates for the Moreau’s time–stepping

Order “0” case
Standard error estimates do not apply for Order 0. We propose to extend it to the order 0 of consistency by assuming that the constant can be evaluated by

\[ C = \frac{2(e_1 - e_{1/2})}{h} \quad (58) \]

and the local error estimate by

\[ e_{1/2} = 2(x_{1/2} - x_1) + O(h^2) \quad (59) \]

The adaptive time–step control exposed for smooth ODE is then apply directly.
Order “0” time–step adjustment for the Moreau’s time–stepping

Figure: Precision Work diagram for the Moreau’s time-stepping scheme. Order 0
Order “0” time–step adjustment for the Moreau’s time–stepping

**Figure**: Precision Work diagram for the Moreau’s time-stepping scheme. Order 0
Sizing the error in the violation of constraints

The violation of constraints is sized by the following rule:

$$e_{\text{violation}} = \| \min(0, g(q)) \circ invtol \|_{\infty}$$  \hspace{1cm} (60)

Assuming that the scheme is of order 1 almost everywhere in smooth phase and may be controlled by $e_{\text{violation}}$ when an nonsmooth vent occurs, the step size adjustment is implemented by the means of the following error estimation

$$\text{error} = \max(e_{\text{violation}}, \| e_k \circ invtol \|_{\infty})$$  \hspace{1cm} (61)
Results on two academic test examples

![MoreauTS Precision-Work Diagram. Bouncing Ball Example](image)

**Figure:** Precision Work diagram for the Moreau’s time-stepping scheme. Order 0 + violation error.
Results on two academic test examples

![MoreauTS Precision-Work Diagram. Linear Oscillator Example](image)

**Figure:** Precision Work diagram for the Moreau’s time-stepping scheme. Order 0 + violation error
Results on two academic test examples

![MoreauTS Precision-Work Diagram. Bouncing Ball Example](image)

Figure: Precision Work diagram for the Moreau's time-stepping scheme. Order 1 + violation error
Results on two academic test examples

Figure: Precision Work diagram for the Moreau’s time-stepping scheme. Order 1 + violation error
Introduction & Motivations

Lagrangian dynamical systems with unilateral constraints and friction

Numerical time–integration schemes

Adaptive time-step size time–stepping scheme

Time–stepping schemes of any order

Splitting based Time–stepping schemes

Conclusions & Perspectives

The Siconos Platform
Higher Order Time-stepping schemes

Background
Work of Mannshardt (1978) on time-integration schemes of any order for ODEs with discontinuities (with tranversality assumption)

Principle

- Let us assume only one event per time-step at instants $t_\ast$.
- Choose any ODE solvers of order $p$
- Perform a rough location of the event inside the time step of length $h$
  Find an interval $[t_a, t_b]$ such that
  \[
  t_\ast \in [t_a, t_b] \text{ and } |t_b - t_a| = Ch^{p+1} + O(h^{p+2})
  \] (62)
  Dichotomy, Newton, Local Interpolants, Dense output, ...
- Perform an integration on $[t_k, t_a]$ with the ODE solver of order $p$
- Perform an integration on $[t_a, t_b]$ with Moreau’s time-stepping scheme
- Perform an integration on $[t_b, t_{k+1}]$ with the ODE solver of order $p$
Results on the linear oscillator

Figure: Precision Work diagram for the Moreau's time-stepping scheme.
Higher Order Time-stepping schemes

Finite accumulation

- Repeat the whole process on the remaining part of the interval $[t_b, t_k]$
- By induction, repeat this process up to the end of the original time step.
Results on the Bouncing Ball

Figure: Precision Work diagram for the Moreau’s time-stepping scheme.
Nonsmooth dynamical systems. Numerical Time–integration schemes and the Siconos platform

Vincent Acary

vincent.acary@inrialpes.fr

INRIA Rhône–Alpes

Introduction & Motivations

Lagrangian dynamical systems with unilateral constraints and friction

Numerical time–integration schemes

Adaptive time-step size time–stepping scheme

Time–stepping schemes of any order

Splitting based Time–stepping schemes

Conclusions & Perspectives

The Siconos Platform

References
Splitting–based methods.

**Principle for smooth ODEs**

Let us consider a smooth ODE which can be written as

\[
\dot{x}(t) = f(x, t) + g(x, t)
\]  

A example of splitting–based method is given by the following procedure

1. Perform the integration of \(f\) on \([t_k, t_{k+1}]\) to obtain \(\tilde{x}(t_{k+1})\) that is

\[
\tilde{x}(t_{k+1}) = x(t_k) + \int_{t_k}^{t_{k+1}} f(x, t) \, dt
\]  

2. Perform the integration of \(g\) on \([t_k, t_{k+1}]\) with initial value \(\tilde{x}(t_{k+1})\) to obtain \(\hat{x}(t_{k+1})\) that is

\[
\hat{x}(t_{k+1}) = \tilde{x}(t_{k+1}) + \int_{t_k}^{t_{k+1}} g(x, t) \, dt
\]

**Properties**

- \(x(t_k + 1) \neq \hat{x}(t_{k+1})\) is the general case. (except special linear case, constant dynamics, . . . )
- \(\hat{x}(t_{k+1}) \to x(t_{k+1})\) when \(t_{k+1} \to t_k\)
Splitting–based methods.

Splitting–based for Moreau scheme without continuous contact forces

- The first part is

\[
\begin{align*}
M(q)\dot{v} &= F(t, q, v), \\
\dot{q} &= v, \\
q(t_k) &= q_k, \quad v(t_k) &= v_k
\end{align*}
\]  

(66)

yielding to the approximations \(q_1 = q(t_{k+1})\) and \(v_1 = v(t_{k+1})\) which can integrated by any smooth ODE solvers.

- The second one is given by

\[
\begin{align*}
M(q)\dot{v} &= G(q)\lambda, \\
\dot{q} &= 0, \\
y &= g(q) \\
-\lambda &\in \partial\psi_{T_{\text{IR}^+}}(y)(\dot{y}(t^+) + e\dot{y}(t^-)) \\
q(t_k) &= q_1; v(t_k) &= v_1;
\end{align*}
\]  

(67)

and leads to the approximation \(q_{k+1} = q(t_{k+1})\) and \(q_{k+1} = q(t_{k+1})\).

Figure: Empirical order of convergence of the Splitting RKF45 time-stepping scheme

Figure: Empirical order of convergence of the Splitting RKF45 time-stepping scheme
Splitting–based methods with adaptive time–step.

**Figure**: Empirical order of convergence of the Splitting RKF45 time-stepping scheme
Splitting–based methods.

Splitting–based for Moreau scheme with continuous contact forces

- The first part is

\[
\begin{aligned}
M(q)\dot{v} &= F(t, q, v) + r(t), \\
\dot{q} &= v, \\
y &= g(q) \\
-r(t) &\in \partial\psi_{T_{IR^+}}(y)(\dot{y}(t)) \\
q(t_k) &= q_k, \quad v(t_k) = v_k
\end{aligned}
\]  

(68)

yielding to the approximations \( q_1 = q(t_{k+1}) \) and \( v_1 = v(t_{k+1}) \) which can integrated by any smooth ODE solvers.

- The second one is given by

\[
\begin{aligned}
M(q)\dot{v} &= G(q)\lambda, \\
\dot{q} &= 0, \\
y &= g(q) \\
-\lambda &\in \partial\psi_{T_{IR^+}}(y)(\dot{y}(t^+) + e\dot{y}(t^-)) \\
q(t_k) &= q_1; \quad v(t_k) = v_1;
\end{aligned}
\]  

(69)

and leads to the approximation \( q_{k+1} = q(t_{k+1}) \) and \( q_{k+1} = q(t_{k+1}) \).
Introduction & Motivations

Lagrangian dynamical systems with unilateral constraints and friction

Numerical time–integration schemes

Adaptive time-step size time–stepping scheme

Time–stepping schemes of any order

Splitting based Time–stepping schemes

Conclusions & Perspectives

The Siconos Platform
Conclusions

Adaptive time–step strategies

- Higher resolution schemes
- Work with finite accumulation of events

Higher order schemes

- Schemes of any orders
- Work with finite accumulation of events

Splitting based methods

- Higher resolution schemes
- Work with finite accumulation of events
 Perspectives

- Theoretical works on orders and practical error estimations
- Adaptive time–step strategies on the higher order time–stepping schemes.
- Improve the pre–detection process of the event and the order of discontinuity
- Test on nonsmooth and nonlinear mechanical systems.
- Adapt the schemes with a step without external forces when the Moreau's scheme is used
- Other types of time–stepping schemes . . .
Overview of the Siconos Platform

Context
The Siconos Platform is one of the main outcome of the Siconos EU project.

Functionalities
Modeling, simulation, (analysis and control) of Non Smooth Dynamical Systems.

Constraints and Requirements

- various applications fields (Mechanics, Electronics . . . ) and corresponding modeling habits and formulations
- various mathematical and numerical tools
- various skills in computer science (from the high performance computing to the Matlab users)
- links and interfaces with existing softwares:
  - low-level numerical libraries (BLAS, LAPACK, ODEPACK, . . . )
  - Matlab or Scilab dedicated user toolbox
  - simulation tools for an application field: Scicos, Simulink, FEM and DEM Sofware (LMGC90, . . . ), Hybrid Modeling Language (Modelica, . . . )
Siconos components diagram
**SICONOS components diagram**

- **SICONOS/Numerics API C**:
  shared dynamic library that provides low-level solvers and algorithms in C and fortran.
  Sources: NSSpack (LCP, Friction ...), odepack (Lsodar ...).

- **SICONOS/Kernel**: API C++: compiled command files with high level methods (C++ Constructors and/or XML file data loading.)
  \( \Rightarrow \) from simulation \( \rightarrow \) run() to DynamicalSystem \( \rightarrow \) computeFext(t)

- **SICONOS/Front-End**: “user-friendly” interface providing a more interactive way of using the platform.
  - API C++ with interactive environment Python scripting (Swig wrapper).
  - API C: Scilab and Matlab interfaces.
Modeling Principle:

Non Smooth Dynamical system

\[ \dot{x} = f(x, t) + r \]
Modeling Principle:

Non Smooth Dynamical system

Dynamical system
\[ \dot{x} = f(x, t) + r \]

Non Smooth Law
\[ 0 \leq y \perp \lambda \geq 0 \]
Modeling Principle:

Non Smooth Dynamical system

Dynamical system
\[ \dot{x} = f(x, t) + r \]

Input/Output Relation
\[ r = g(\lambda) \]

Non Smooth Law
\[ 0 \leq y \perp \lambda \geq 0 \]
Modeling Principle:

Non Smooth Dynamical system

\[ \dot{x} = f(x, t) + r \]

Non Smooth Interaction

Input/Output Relation

\[ r = g(\lambda) \]

Non Smooth Law

\[ 0 \leq y \perp \lambda \geq 0 \]
Kernel Modeling Part

Siconos Non Smooth Dynamical System:

- **Dynamical System**: a set of ODEs
- **Interaction**: a set of relations (ie constraints) and a non-smooth law
Kernel Modeling Part

Siconos Non Smooth Dynamical System:

- **Dynamical System**: a set of ODEs
- **Interaction**: a set of relations (i.e., constraints) and a non-smooth law
- **Topology**: link with the simulation, handles relative degrees, index sets...

Simplified Modeling Tools class diagram:
Dynamical Systems in Siconos/Kernel

- Parent Class **DynamicalSystem**

\[ \dot{x} = f(x, \dot{x}, t) + T(x)u(x, t) + r \]
Parent Class **DynamicalSystem**

\[ \dot{x} = f(x, \dot{x}, t) + T(x)u(x, t) + r \]

Derived Classes

- **LinearDS** Linear Dynamical Systems
  \[ \dot{x} = A(t)x + Tu(t) + b(t) + r \]

- **LagrangianDS** Lagrangian Dynamical Systems
  \[ M(q)\ddot{q} + NNL(q, \dot{q}) + F_{\text{int}}(\dot{q}, q, t) = F_{\text{ext}}(t) + T(q)u(q, t) + p \]

- **LagrangianLinearTIDS** Lagrangian Linear Time Invariant Systems
  \[ M\ddot{q} + C\dot{q} + Kq = F_{\text{ext}}(t) + Tu(t) + p \]

**Note:** all operators (\( f(x, t) \), \( M(q) \), ...) can be set either as matrices (when constant) or with a user-defined external function (plug-in).
Relations

- Parent Class **Relation**

\[ y = h(x, t, \ldots) \quad , \quad r = g(\lambda, t, \ldots) \]
Relations

- Parent Class **Relation**

  \[ y = h(x, t, ...) \quad , \quad r = g(\lambda, t, ...) \]

- Derived Classes:
  - **LinearTIR** Linear Time Invariant Relation
    \[ y = Cx + Fu + D\lambda + e, \quad r = B\lambda \]
  - **LagrangianR** Lagrangian Relation
    \[ \dot{y} = H(q, t, \ldots)\dot{q}, \quad p = H^t(q, t, \ldots)\lambda \]
  - **LagrangianLinearR** Lagrangian Linear Relation
    \[ \dot{y} = H\dot{q} + b, \quad p = H^t\lambda \]
Non Smooth laws

- **Parent Class** NonSmoothLaw
- **Derived Classes**
  - **ComplementarityConditionNSL** Complementarity condition or unilateral contact
    \[ 0 \leq y \perp \lambda \geq 0 \]
  - **Relay** condition.
    \[
    \begin{cases}
    \dot{y} = 0, |\lambda| \leq 1 \\
    \dot{y} \neq 0, \lambda = \text{sign}(y)
    \end{cases}
    \]
  - **NewtonImpactLawNSL** Newton impact Law.
    \[
    \text{if } y(t) = 0, \quad 0 \leq \dot{y}(t^+) + e\dot{y}(t^-) \perp \lambda \geq 0
    \]
  - **NewtonImpactFrictionNSL** Newton impact and Friction (Coulomb) Law.
Dynamical Systems definition:

DynamicalSystem * DS1 = new LagrangianLinearTIDS(nDof,q0,v0,Mass);
DS1→setComputeFExtFunction("BallPlugin.so", "ballFExt");
C++ description of a Model

- **Dynamical Systems definition:**

  DynamicalSystem * DS1 = new LagrangianLinearTIDS(nDof,q0,v0,Mass);
  DS1->setComputeFExtFunction("BallPlugin.so", "ballFExt");

- **Interactions definition:** non smooth law and relation:

  NonSmoothLaw * nslaw = new NewtonImpactNSL(e);
  Relation * relation = new LagrangianLinearR(H,b);
  Interaction * inter = new Interaction(name, listOfDS,dim, nslaw, relation);
C++ description of a Model

- **Dynamical Systems definition:**
  
  ```cpp
  DynamicalSystem * DS1 = new LagrangianLinearTIDS(nDof,q0,v0,Mass);
  DS1->setComputeFExtFunction("BallPlugin.so", "ballFExt");
  ```

- **Interactions definition:** non smooth law and relation:
  
  ```cpp
  NonSmoothLaw * nslaw = new NewtonImpactNSL(e);
  Relation * relation = new LagrangianLinearR(H,b);
  Interaction * inter = new Interaction(name, listOfDS,dim, nslaw, relation);
  ```

- **Non Smooth Dynamical System and Model**
  
  ```cpp
  NonSmoothDynamicalSystem * nsds = new NonSmoothDynamicalSystem(allIDS, allInteractions);
  Model * theModel = new Model(t0,T);
  theModel->setNonSmoothDynamicalSystemPtr(nsds);
  ```
Simulation tools in Siconos/Kernell

- Simulation
- TimeDiscretisation
- OneStepIntegrator
- OneStepNSProblems
- Solver
Simulation tools in Siconos/Kernel

Simulation description in C++ input file:

Simulation* s = new TimeStepping(theModel);
TimeDiscretisation* t = new TimeDiscretisation(timeStep,s);
OneStepIntegrator* OSI = new Moreau(listOfDS,theta,s);
OneStepNSProblem* osnspb = new LCP(s,"LCP",Lemke,parameters);
Simulation tools in Siconos/Kernel

Simulation description in C++ input file:

Simulation* s = new TimeStepping(theModel);
TimeDiscretisation * t = new TimeDiscretisation(timeStep,s);
OneStepIntegrator * OSI = new Moreau(listOfDS,theta,s);
OneStepNSProblem * osnspb = new LCP(s, "LCP",Lemke,parameters);

Unitary Relation and Index Sets

UR: \( y^i = h(q, \ldots) \).
Index Sets: set of Unitary Relations (UR).

- \( I_0 = \{ UR_\alpha \} \) all unilateral constraints in the system, ie all the potential interactions/relations of the systems.
- \( I_i = \{ UR_\alpha, \alpha \in I_{i-1}, y^{(i-1)} = 0 \} \subset I_{i-1} \)
Simulation tools in Siconos/Kernel
Simulation tools in Siconos/Kernel

- TimeStepping
- Simulation
- TimeDiscretisation
  - OneStepIntegrator
  - OneStepNSProblems
  - Solver
    - Moreau
    - FrictionContact
    - LCP

- EventDriven
  - Simulation
  - TimeDiscretisation
    - OneStepIntegrator
    - OneStepNSProblems
    - Solver
      - Event
      - Lsodar
      - LCP
**OneStepIntegrator:**
- **Moreau**: Moreau’s Time-stepping integrator
- **Lsodar**: Numerical integration scheme based on the Livermore Solver for Ordinary Differential Equations with root finding.

**OnestepNSproblem**: Numerical one step non smooth problem formulation and solver.
- **LCP** Linear Complementarity Problem
  \[
  \begin{align*}
  w &= Mz + q \\
  0 &\leq w \perp z \geq 0
  \end{align*}
  \]
- **FrictionContact2D(3D)** Two(three)-dimensional contact friction problem
- **QP** Quadratic programming problem
  \[
  \begin{align*}
  \min & \quad \frac{1}{2} z^T Q z + z^T p \\
  & \quad z \geq 0
  \end{align*}
  \]
- **Relay**
**Model:** Lagrangian Linear Time Invariant Dynamical Systems with Lagrangian Linear Relations, Newton Impact Law.

**Simulation:** Moreau’s Time Stepping or Event Driven.

**Bouncing Ball**

**Beads column**
A 4 diodes bridge wave rectifier.

**Model:** Linear Dynamical System with Linear Relations, Complementarity Condition Non Smooth Law.

**Simulation:** Moreau’s Time Stepping

Comparison between the SICONOS Platform (Non Smooth LCS model) and SPICE simulator (Smooth Diode model).
Woodpecker toy (sample from Michael Moeller (CR10))

Model: Lagrangian Linear Dynamical System, Lagrangian Linear Relations, Newton impact-friction law.
Simulation: Moreau’s Time Stepping
A Robotic Arm (Pa10)

Model: Lagrangian Non Linear Dynamical System with Lagrangian Non Linear Relations, Newton impact.
Simulation: Moreau’s Time Stepping
Help and Documentation

- Doxygen tools for automatic documentation in Numerics and Kernel
- Users, developers and theoretical manuals (in progress ...)
- Web pages, Bug tracker, forum ... on Gforge.
- Samples library as templates.

Diffusion

- The SICONOS platform is distributed under GPL licence.
- Visit the Gforge Web site for
  - Documentations
  - Mailing lists
  - Downloads
  - Bug tracker
  - Contributing, ...

http://gforge.inria.fr/projects/siconos/
Introduction & Motivations

Lagrangian dynamical systems with unilateral constraints and friction

Numerical time–integration schemes

Adaptive time-step size time–stepping scheme

Time–stepping schemes of any order

Splitting based Time–stepping schemes

Conclusions & Perspectives

The Siconos Platform
Introduction & Motivations

Lagrangian dynamical systems with unilateral constraints and friction

Numerical time–integration schemes

Adaptive time-step size time–stepping scheme

Time–stepping schemes of any order

Splitting based Time–stepping schemes

Conclusions & Perspectives

The Siconos Platform
Thank you for your attention.
Nonsmooth dynamical systems. Numerical Time–integration schemes and the Siconos platform
Vincent Acary
vincent.acary@inrialpes.fr
INRIA Rhône–Alpes

Introduction & Motivations
Outline
Lagrangian dynamical systems with unilateral constraints and friction
Numerical time–integration schemes
Adaptive schemes
Higher Order Schemes
Splitting based Schemes
Conclusions & Perspectives
The Siconos Platform
References


