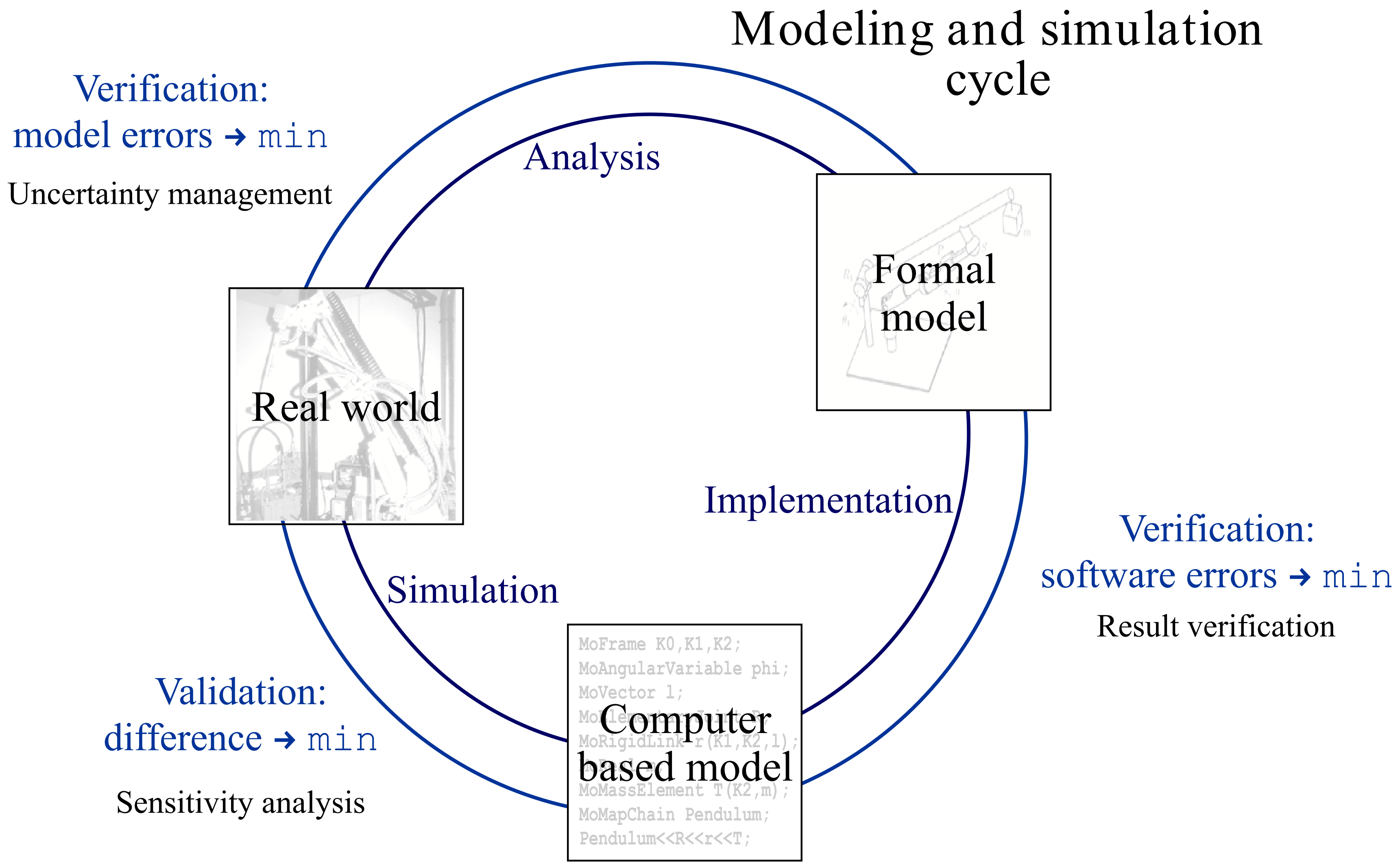
# Verification and Validation of Two Biomechanical Problems

**Ekaterina Auer, Wolfram Luther**

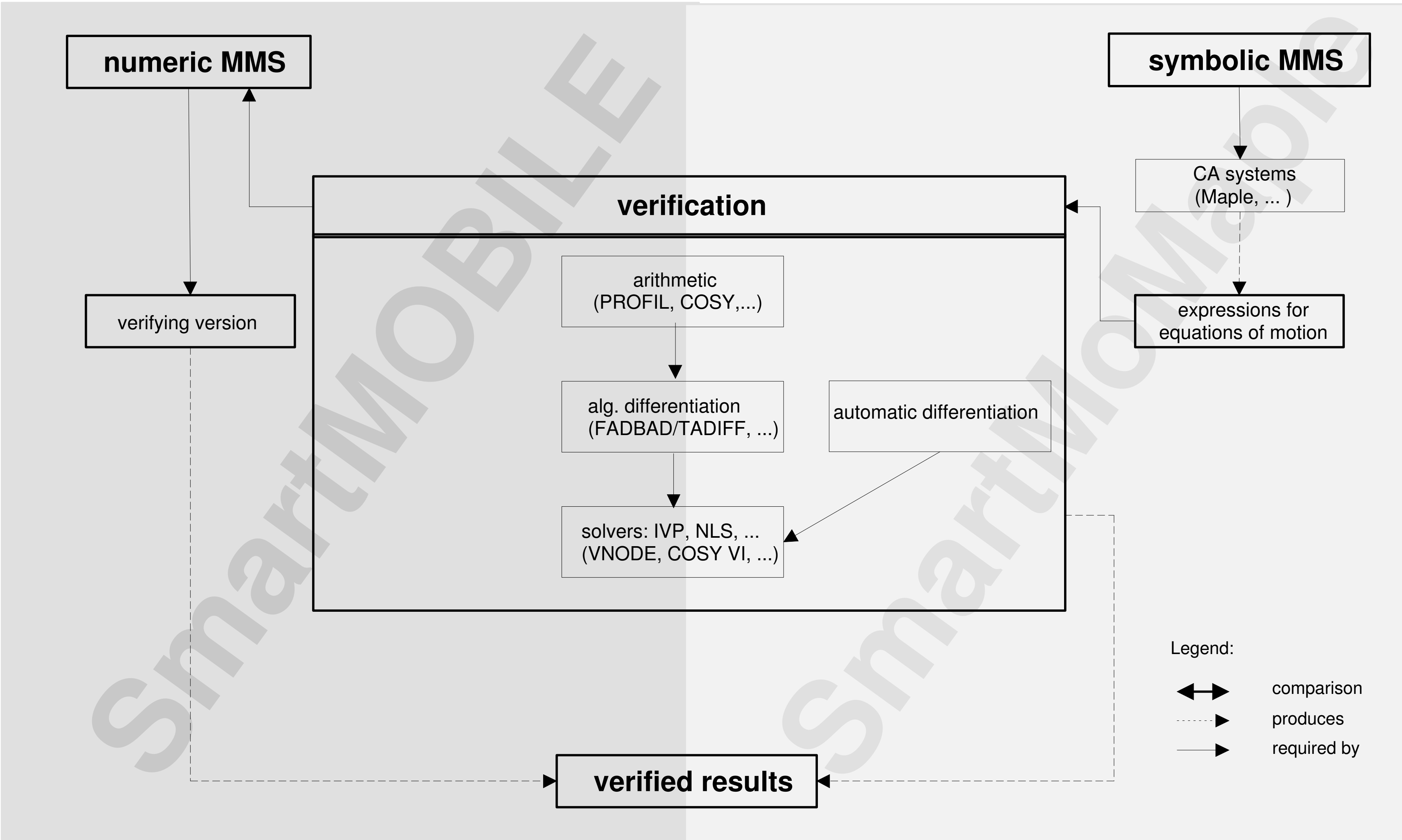
**University of Duisburg-Essen, Germany**

**SCAN, September-October 2008**

# 1 Motivation



# Approaches



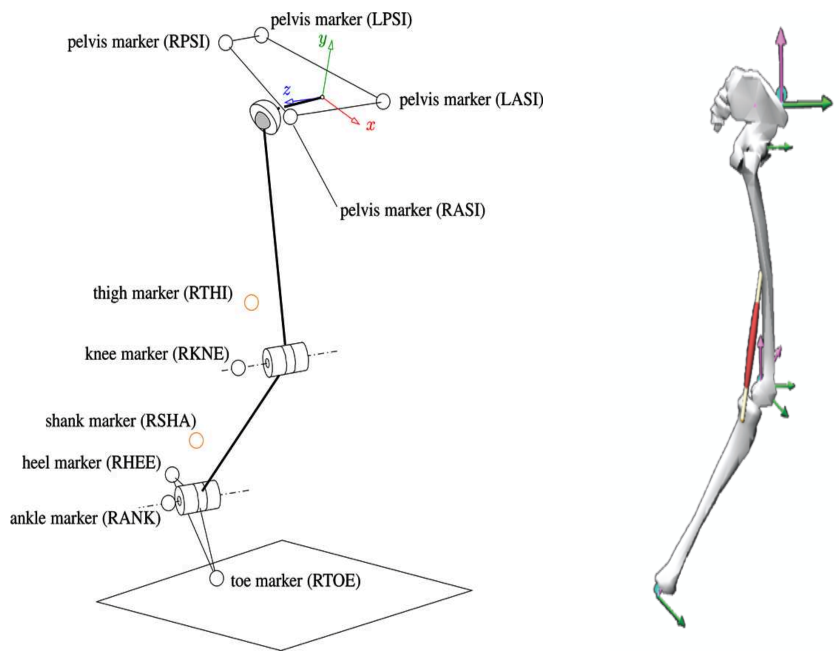

SmartMOBILE / SmartMoMaple —

**S**imulation and **M**odeling of dyn**A**mics: **R**eliable and **T**emplate-based/ au**T**omatized

# Applications in Biomechanics

Find out how changes in parameters influence the result

1. **Identification of body segment motion**
   - quantify the influence of marker displacements on motion




2. **Muscle activation model**
   - find the sensitivity of the model to its parameters
   - work with piecewise (differentiable) functions

3. **Many more...**

## Topics

- Theoretical aspects
- Main features of SmartMOBILE and SmartMoMaple
- Identification of body segment motion
- A simplified muscle activation model
- Conclusions

# 2 Theory: Important Concepts and Facts

**Arithmetics** Intervals, Taylor models

**Solvers**

Pure iteration: need Jacobians or rely on their arithmetic

Series based: need e.g. Taylor coefficients and their Jacobians

**AD** Code transformation, overloading

**Sensitivity**

Definition: $s = \dfrac{\partial[x]}{\partial[p]}$, where $[p]$ is an (uncertain) parameter

Meaning: a linear measure of uncertainty influence

Reference: $s = \dfrac{\partial x}{\partial p} \times \text{uncertainty}(p)$ with intervals

# 3 Software with Result Verification: Choices

| | |
|---|---|
| Intervals: | PROFIL/BIAS by O. Knüppel, 1998 |
| Taylor models: | – COSY INFINITY by M. Berz, 2004 |
| | – RiOT by I. Eble, 2006 |
| IVP solvers: | – VNODE by N. Nedialkov, 2001 |
| | – ValEncIA-IVP by A. Rauh, 2006 |
| | – RiOT by I. Eble, 2006 |
| | – VSPODE by M. Stadtherr, 2005 |
| AD: | – FADBAD++ by C. Bendsten, 2007 |
| | – CppAD by B. M. Bell, 2002 |

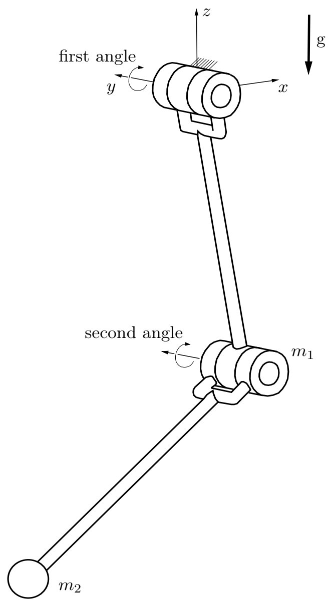
# 4 Main Features of SmartMOBILE

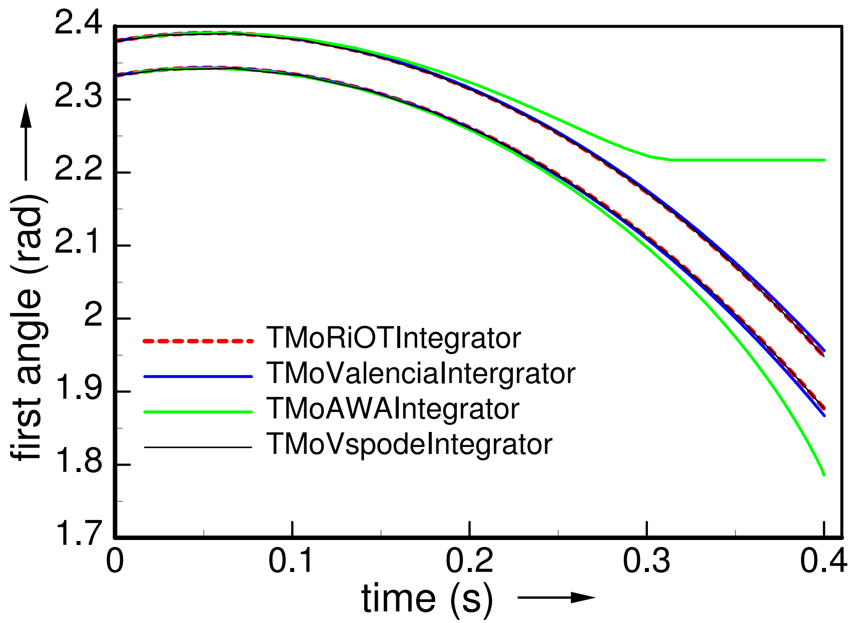
1. Based on MOBILE by A. Kecskeméthy
2. Verified kinematics/dynamics + uncertainty management
3. Free choice of the underlying arithmetic: templates + solvers

| Type | Integrator | Purpose |
|---|---|---|
| `MoReal` | `MoAdams,...` | nonverified dynamics |
| `TMoInterval` | `TMoAWA` | |
| `TMoFInterval` | `TMoValencia` | verified dynamics of |
| `TMoTaylorModel` | `TMoRiOT` | ODE based systems |
| `TMoTaylorModel` | `TMoVSPODE` | |
| `RDAInterval` | `---` | Taylor model based kinematics |
| `MoFInterval` | `MoIGradient` | verified equilibria<br>kinematics with constraints |
| `MoSInterval` | `TMoValenciaS` | verified sensitivity |

4. Converters MOBILE $\longrightarrow$ SmartMOBILE

# Example: Dynamics of a Double Pendulum with an Uncertain Initial Angle




```
# define TMoInterval t;
TMoFrame<t> K0, K1, K2, K3, K4;
TMoAngularVariable<t> psi1, psi2;
// transmission elements
TMoVector<t> l1(0,0,-1), l2(0,0,-1);
TMoElementaryJoint<t> R1(K0,K1,psi1,xAxis);
TMoElementaryJoint<t> R2(K2,K3,psi2,xAxis);
TMoRigidLink<t> rod1(K1,K2,l1),rod2(K3,K4,l2);
t m1(1),m2(1);
TMoMassElement<t> Tip1(K2,m1),Tip2(K4,m2);
// the complete system
TMoMapChain<t> Pend;
Pend << R1<<rod1<<Tip1<<R2<<rod2<<Tip2;
// dynamics
TMoVariableList<t> q; q << psi1<<psi2;
TMoMechanicalSystem<t> S(q,Pend,K0,zAxis);
TMoAWAIntegrator I(S,0.0001,ITS_QR,15);
I.doMotion();
```

| Strategy | TMoAWA (variable $h$) | TMoRiOT ($0.0002 \le h \le 0.2$) | TMoValencia ($h = 10^{-4}$) | TMoVSPODE (variable $h$) |
|---|---|---|---|---|
| Break–down | 0.420 | 0.801 | 0.531 | 0.656 |
| CPU Time* | 5 | 285 | 22 | 10 |

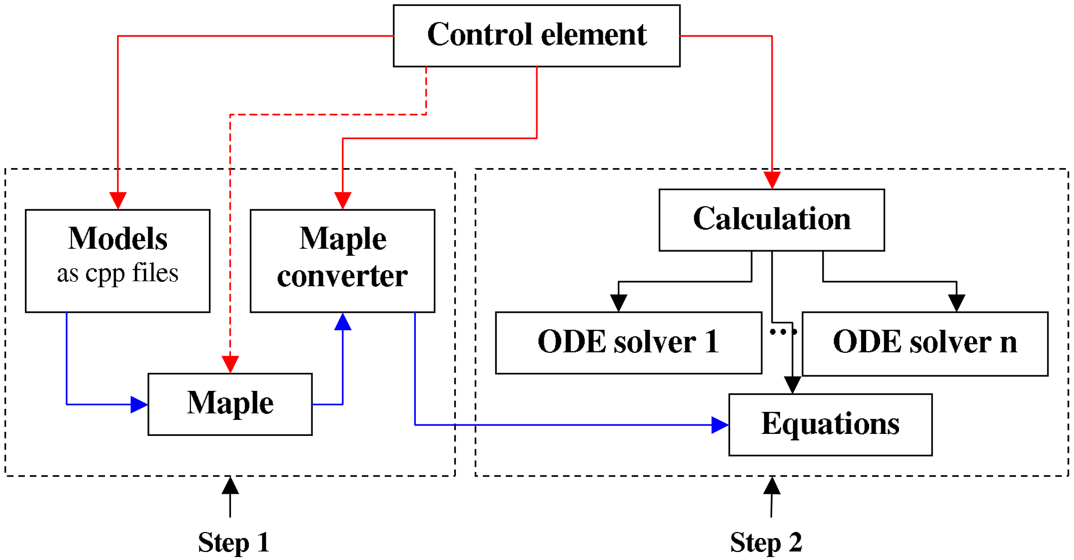
* computed on 8 $\times$ Intel Xeon CPU 2.00GHz under Linux 2.6.25.14-69.fc8

# 5 SmartMoMaple – Symbolic Counterpart to SmartMOBILE

MoMaple

- a package for MOBILE
- creates *symbolic* expressions via Maple
- no closed loop systems
- modeling only, simulation not automatized

SmartMoMaple*




- automatizes symbolic simulations
- semi-automatic integration of (verified) ODE solvers possible
- currently available: VNODE-LP, RiOT, ValEncIA-IVP

* Inna Reinke, Master thesis, October 2008

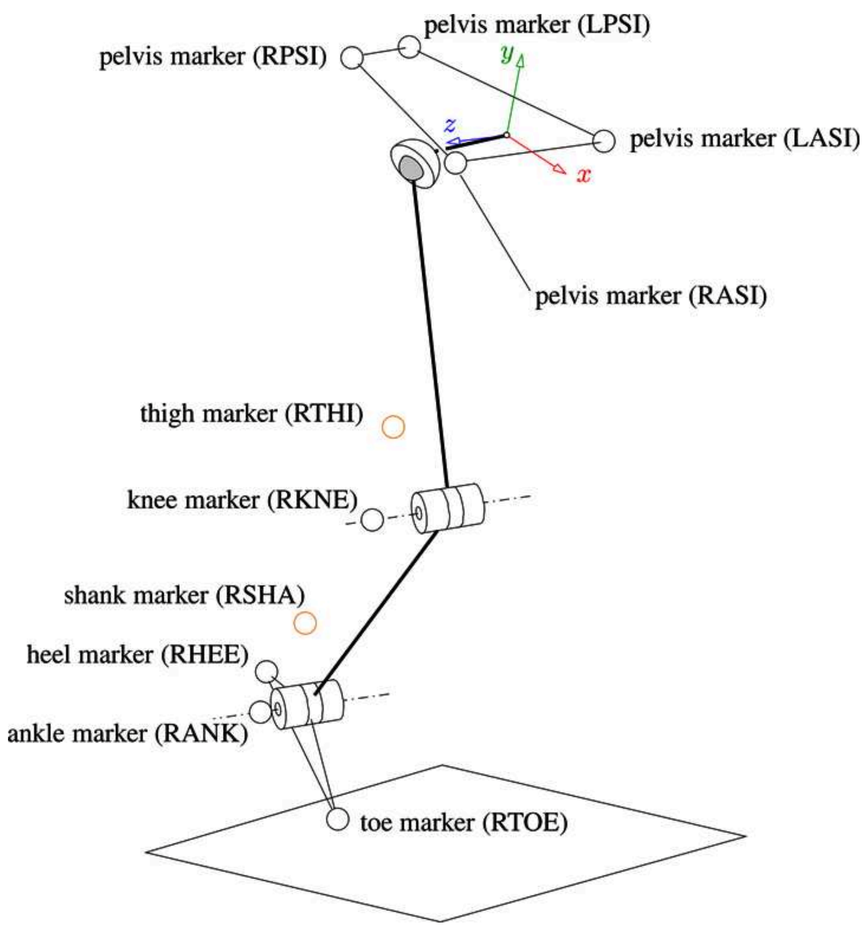
# Example: Dynamics of a Double Pendulum in SmartMoMaple

(numbers in this color show data from SmartMOBILE)

| Strategy | VNODE-LP (variable $h$) | | RiOT ($0.0002 \leq h \leq 0.2$) | | ValEncIA-IVP ($h = 10^{-4}$) | |
|---|---|---|---|---|---|---|
| Break–down | 0.520 | 0.420 | $> 2$ | 0.801 | 0.606 | 0.531 |
| CPU Time | 6 | 5 | 80min | 285 | 106 | 22 |

**SmartMoMaple:**
- **slower for this example**
- **later breakdown**

# 6 Identification of Body Segment Motion




## Parameters (mm):

| | |
|---|---|
| knee width | $120 \pm 10$ |
| ankle width | $80 \pm 10$ |
| displacements | tangential/soft tissue $\pm$ 10 |
| | normal $\pm$ 5 |

## Femur length (mm):

| | TMoRDA | INTERVAL |
|---|---|---|
| Knee, ankle | [377.6; 396.7] | $[0; \infty]$ |
| Skin displacement | [0.000; 621.4] | no answer |

## Sensitivity of femur wrt.

| Knee | Ankle | Tangential | Normal | Soft |
|---|---|---|---|---|
| 0.4 | -0.3 | -2 | 0.7 | 1.4 |

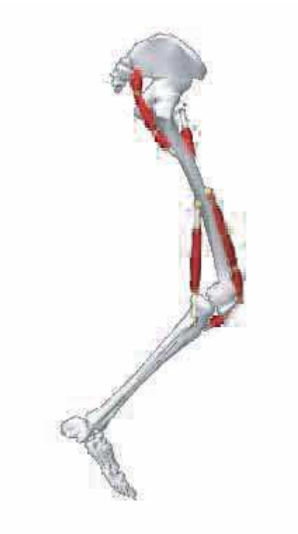
**Reference uncertainty:** Knee, Ankle: **$\pm$7mm**; Tangential, Normal, Soft: **$\pm$37.5mm**

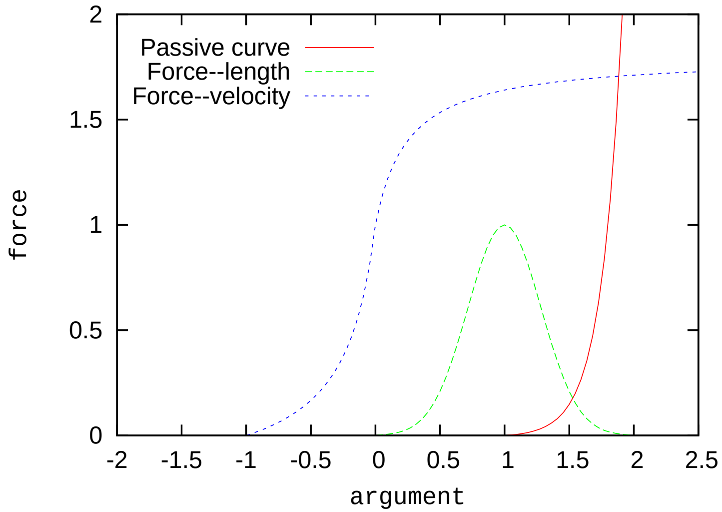
# 7 A Simplified Muscle Activation Model

## Model Description*

**The Hill type muscle model**:

$$F = F_{max} F_P(l) + F_{max} a(t) F_L(l) F_H(v)$$

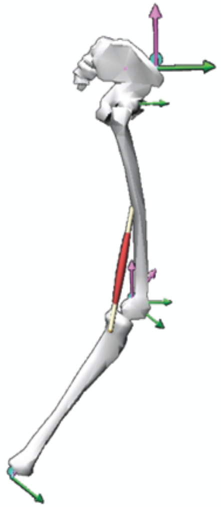





$a(t)$: linear equidistant piecewise and exponential camel-back

**Simplified for verification**:

PT muscle model

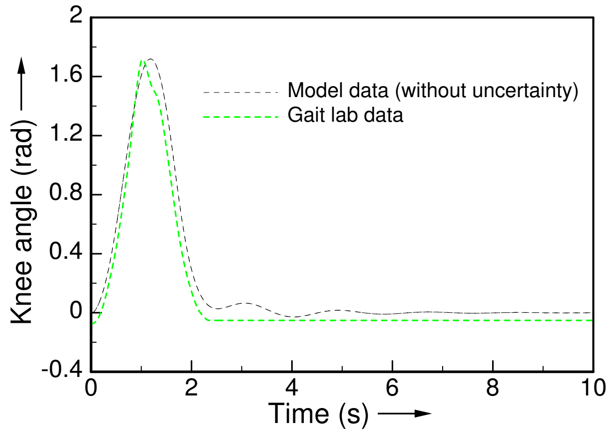


$$F(s) = \frac{P}{1 + Ts}$$

$a(t)$: exponential camel-back

* D. Strobach et al., *A simplified approach for rough identification of muscle activation profiles via optimization and smooth profile patches*, ECCOMAS Proceedings, 2005

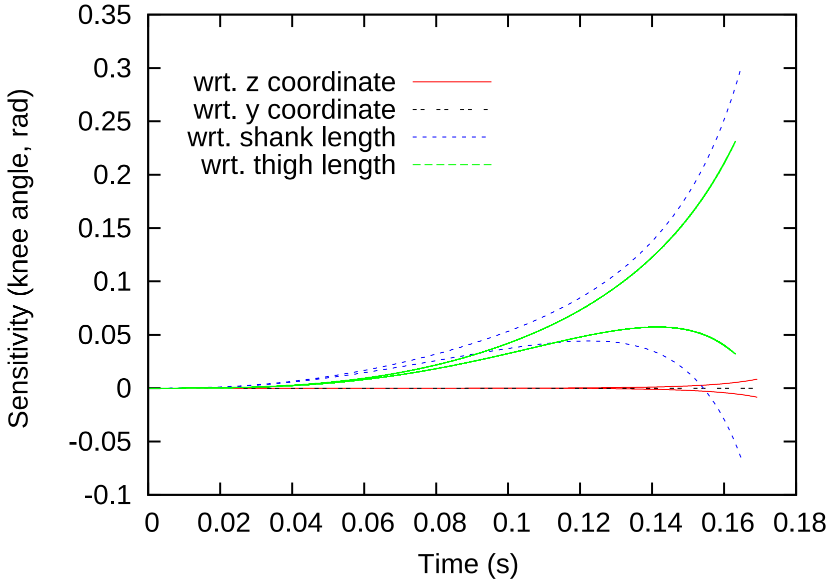
# Verified Simulation in SmartMOBILE (PT Muscle Model)




| Uncertainty (±0.1%) | Break down |
|---|---|
| Thigh length = 0.45 | 0.118 |
| Shank length = 0.49 | 0.125 |
| $z = -0.2281$ | 0.135 |
| $y = -0.0253$ | 0.389 |

## Sensitivities under ±0.1% uncertainty in

- thigh and shank lengths
- muscle insertion points
  - $z$ coordinate
  - $y$ coordinate



# 8 Towards Working With Piecewise Functions

**Necessary:** AD for `if` operator

**Provided by:** CppAD — template based AD library (`double`)

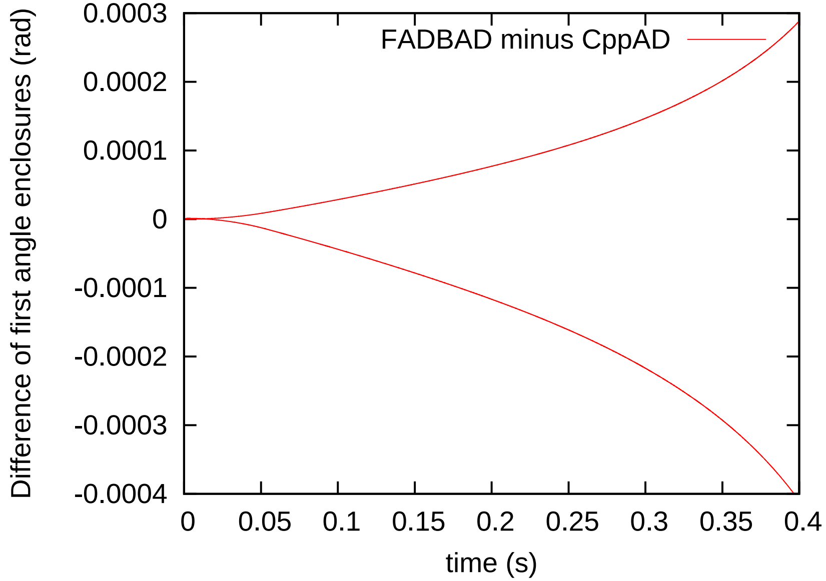
Comparison: The Double Pendulum with CppAD (1) and FADBAD (2)

**Kinematics:**

| | | $\partial y/\partial\beta_1$ | $\partial y/\partial\beta_2$ | CPU ($ms$) |
|---|---|---|---|---|
| `double` | (1) | 0.1011 | 0.8082 | 0.106 |
| | (2) | 0.1011 | 0.8082 | 0.118 |
| `INTERVAL` | (1) | [0.0705,0.1317] | [0.7513,0.8647] | 0.400 |
| | (2) | [0.0661,0.1360] | [0.7513,0.8647] | 0.300 |

$\beta_1$ for double = Mid($[\beta_1]$)

**Dynamics with ValEncIA:**




CPU: $36s$ for (1), $22s$ for (2)

# Kinematics with Piecewise Differentiable Functions

**Hill type muscle model:** Find sensitivity of the knee position to $\pm 1\%$ uncertainty in the thigh length $p = 0.45$m

| | $x$ | $z$ | $\partial x/\partial p$ | $\partial z/\partial p$ |
|---|---|---|---|---|
| `double` | $0.226$ | $0.514$ | $-0.033$ | $-0.9995$ |
| `INTERVAL` | $[0.222, 0.231]$ | $[0.509, 0.519]$ | $-[0.032, 0.033]$ | $-[0.9994, 0.9995]$ |

**Reference uncertainty:** $\pm$1mm in $x$, $\pm$2.3mm in $z$

**Vastus intermedius**
with $l_i = [0.809, 1.199]$,
$l_d = 1.001$

| | `double` | `INTERVAL` |
|---|---|---|
| passive curve | $4.938e-05$ | $[-0.004, 0.001]$ |
| force–length | $0.999992$ | $[0.782, 1.000]$ |
| force–velocity | not used | not used |

## Piecewise Functions For Intervals

| | |
|---|---|
| **Difficulty:** | Comparison operators for intervals |
| **Comparison:** | `if (a<b) x=x1 else x=x2` ,<br>where $a$ interval, $b$ point interval |
| CppAD **syntax:** | `x=CondExpLt(a,b,x1,x2)` |
| **a<b for** `double`**:** | yes or no |
| `CondExpLt` **for** `double`**:** | `double CondExpLt(a, b, x1, x2){`<br>`  if(a<b) return x1; else return x2; }` |
| **a<b for** `INTERVAL`**:** | inclusion or yes/no/possibly |
| `CondExpLt` **for** `INTERVAL`**:** | `INTERVAL CondExpLt(a, b, x1, x2){`<br>`  if(Sup(a)<Inf(b)) return x1;`<br>`    else if(Inf(a)>Sup(b)) return x2;`<br>`      else return x3=f(a,b,x1,x2)=?; }` |

## Choices for $f(a, b, x1, x2)$

**Possibility 1:** Externally through `x=CondExpLt(a,b,x1,x2,maybeCase)` , where `maybeCase`$:=f$ (not supported in CppAD)

**Possibility 2:** Construct from `x1` and `x2`

**But:**

- `x1`, `x2` intervals, not functions of $x$
- Unknown function values in $b$
- No knowledge about if `a`, `b` are function or argument values
- No general strategy possible without extra knowledge

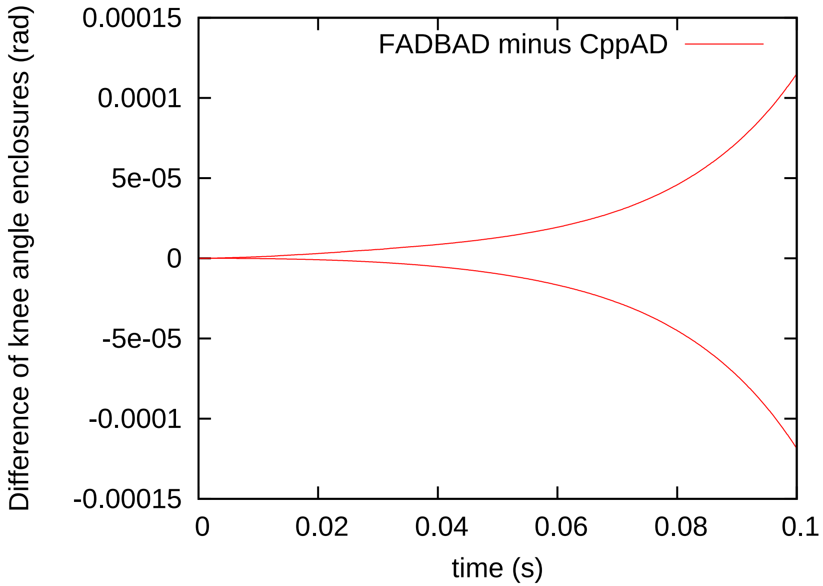
**Solution:** A special comparison operator, preferably without disjoint intervals?

# Dynamics With Simple Piecewise Functions (PT Muscle Model)

**Physical constraints for**

**activation:** $0 \leq a(t) \leq 1$

**force of the PT muscle:** $F \leq 0$




Comparison operator:

`f(a,b,x1,x2)=`

`Hull(Inf(x1), Sup(x2))`

Function values are compared!

**Unfortunately, no improvement in break down times for this model!**

# 9 Conclusions

**Segment motion:**
- overestimation reduced through Taylor models
- reference uncertainty is considerable too

**Muscle activation:**
- sensitivities of the smooth model obtained
- simple piecewise functions tested for dynamics
- kinematics for the Hill type muscle model

**Open question:** definition of `if` and `CondExp` for intervals