Accelerating the simulation of equation-based models by replacing non-linear algebraic loops with error-controlled machine learning surrogates

Presentation · October 2023

DOI: 10.13140/RG.2.2.19830.16963

CITATIONS
0

READS
42

4 authors, including:

Andreas Heuermann
Hochschule Bielefeld
10 PUBLICATIONS **115** CITATIONS
SEE PROFILE

Philip Hannebohm
Hochschule Bielefeld (HSBI)
4 PUBLICATIONS **0** CITATIONS
SEE PROFILE

HSBI

Hochschule Bielefeld

University of Applied Sciences and Arts

Accelerating the simulation of equation-based models by replacing non-linear algebraic loops with error-controlled machine learning surrogates

Andreas Heuermann[1]

Philip Hannebohm[1]

Matthias Schäfer[2]

Bernhard Bachmann[1]

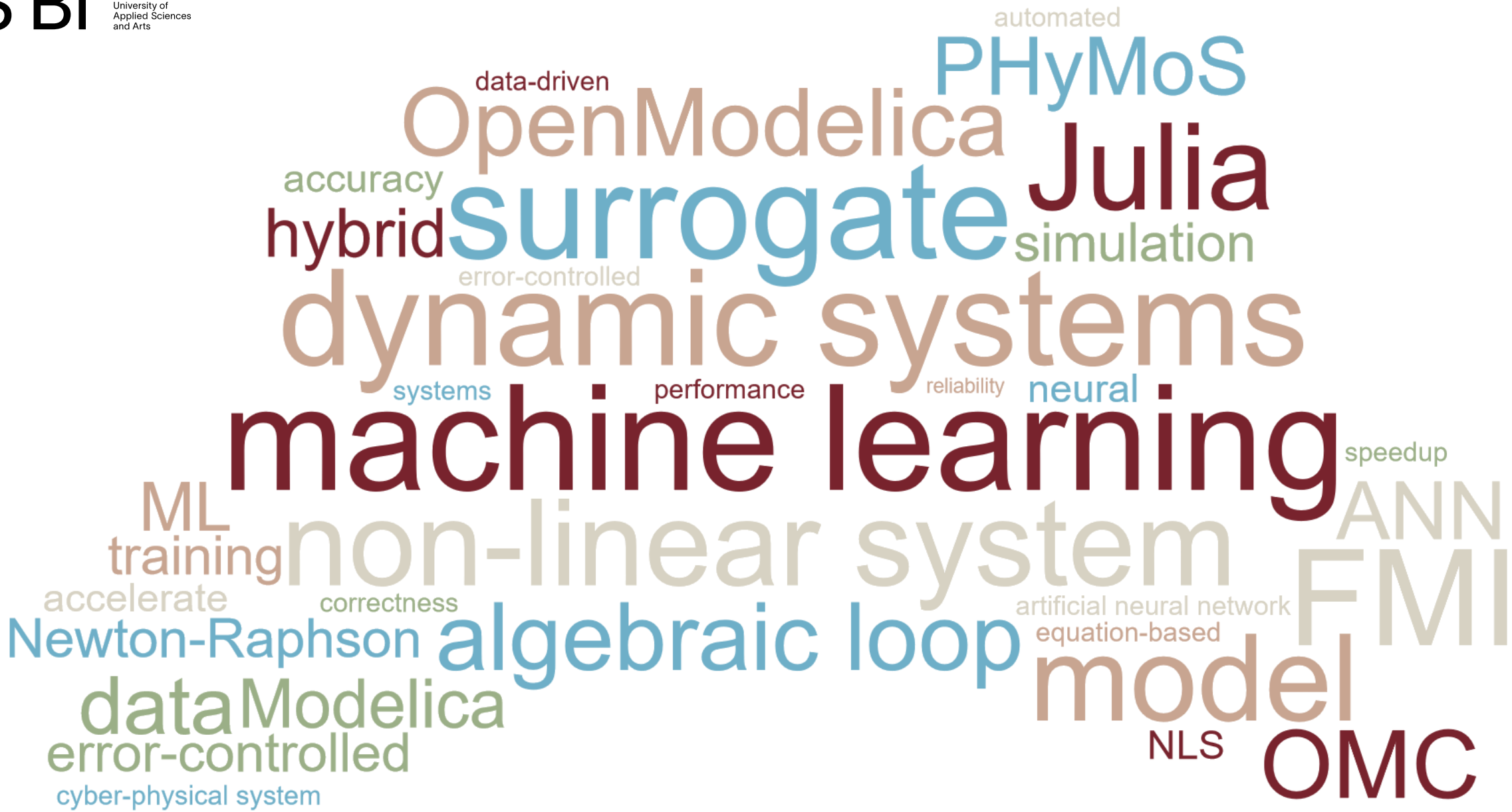
[1] Bielefeld University of Applied Sciences and Arts

[2] LTX Simulation GmbH

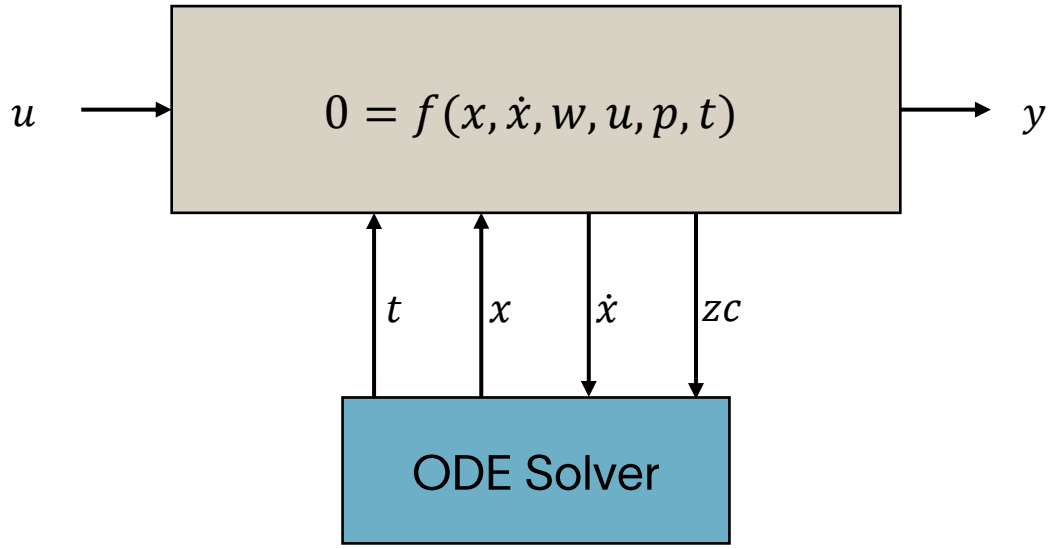
10th October 2023

iDaS

LTX


automated
PHyMoS
data-driven
OpenModelica
Julia
accuracy
surrogate
hybrid
simulation
error-controlled
dynamic systems
systems
performance
reliability
neural
machine learning
speedup
ML
ANN
non-linear system
training
FMI
accelerate
correctness
artificial neural network
Newton-Raphson
algebraic loop
equation-based
model
data
Modelica
NLS
OMC
error-controlled
cyber-physical system

REPLACING NON-LINEAR SYSTEMS



t	Time
p	Parameters
$u(t)$	Inputs
$x(t)$	Continuous states
$w(t)$	Local variables
$y(t)$	Outputs
$zc(t)$	Event indicators

REPLACING NON-LINEAR SYSTEMS

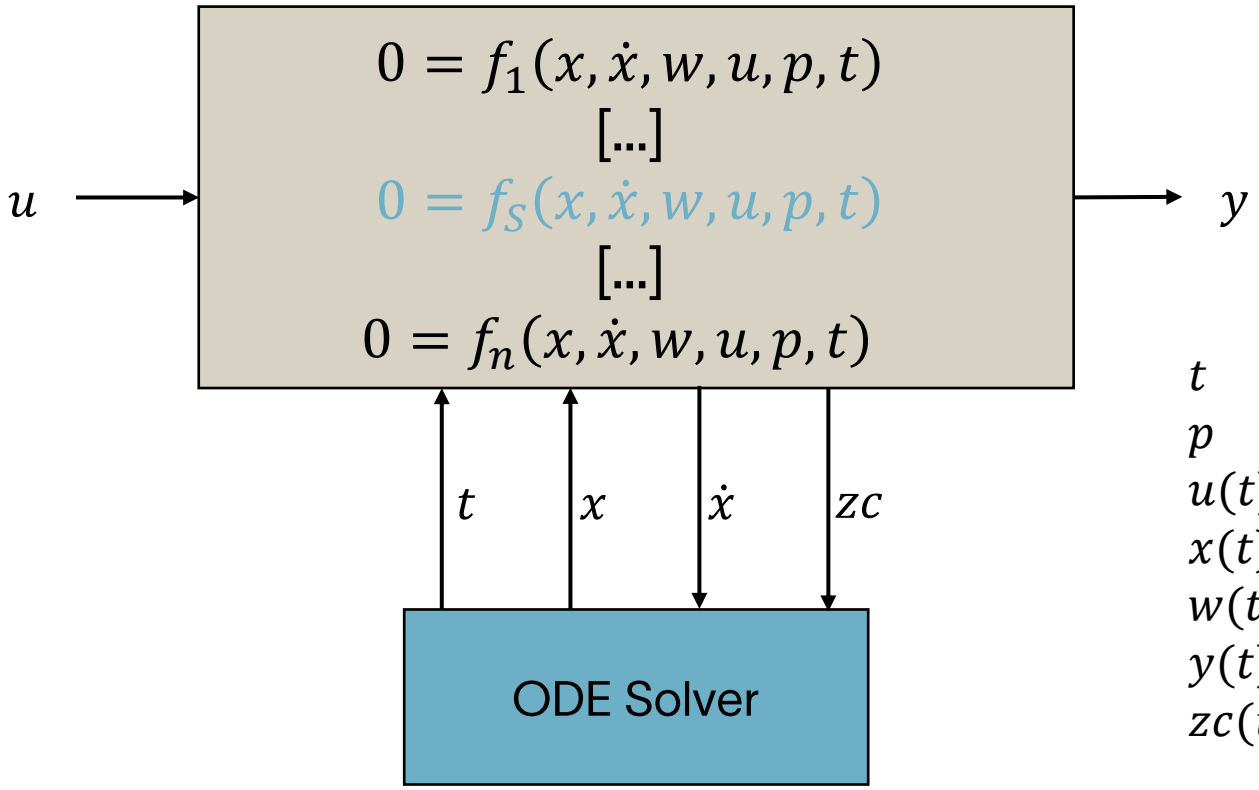
$$z(t) := \begin{pmatrix} \dot{x}(t) \\ w(t) \end{pmatrix} \text{ system unknowns}$$

$$\begin{aligned} f_1(z_3, z_4) &= 0 \\ f_2(z_2) &= 0 \\ f_3(z_2, z_3, z_5) &= 0 \\ f_4(z_1, z_2) &= 0 \\ f_5(z_1, z_3, z_5) &= 0 \end{aligned}$$

$$\begin{aligned} f_2(z_2) &= 0 \\ f_4(z_1, z_2) &= 0 \\ f_{NLS}(z_1, z_2, z_3, z_5) &= 0 \\ f_1(z_3, z_4) &= 0 \end{aligned} \xrightarrow{\text{Machine learning}\ \text{surrogate } f_S} \begin{aligned} f_2(z_2) &= 0 \\ f_4(z_1, z_2) &= 0 \\ f_S(z_1, z_2) &= z_3, z_5 \\ f_1(z_3, z_4) &= 0 \end{aligned}$$

$$\begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{array} \begin{array}{c} \begin{array}{ccccc} z_1 & z_2 & z_3 & z_4 & z_5 \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix} \end{array} \qquad \begin{array}{c} \\ f_2 \\ f_4 \\ f_3 \\ f_5 \\ f_1 \end{array} \begin{array}{c} \begin{array}{ccccc} z_2 & z_1 & z_3 & z_5 & z_4 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix} \end{array} \qquad \begin{array}{c} \\ f_2 \\ f_4 \\ f_S \\ f_1 \end{array} \begin{array}{c} \begin{array}{cccc} z_2 & z_1 & z_3, z_5 & z_4 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \end{array}$$

REPLACING NON-LINEAR SYSTEMS



t	Time
p	Parameters
$u(t)$	Inputs
$x(t)$	Continuous states
$w(t)$	Local variables
$y(t)$	Outputs
$zc(t)$	Event indicators

REPLACING NON-LINEAR SYSTEMS

Non-linear system (solve for x, y):

$$r^2 = x^2 + y^2$$
$$rs + b = x + y$$

Tearing

Translate to:

Inner equation: $x = rs + b - y$

Residual equation: $0 = y^2 + x^2 - r^2$

2 Unknowns:	x, y
1 Iteration variable:	y
Parameters:	b
2 Knowns:	r, s

```
model simpleLoop
  Real r(min = 0);
  Real s(min = -sqrt(2), max = sqrt(2));
  Real x(start=1.0), y(start=0.5);
  parameter Real b = -0.5;
equation
  r = 1+time;
  s = sqrt((2-time)*0.9);

  r^2 = x^2 + y^2;
  r*s + b = x + y;
end simpleLoop;
```

REPLACING NON-LINEAR SYSTEMS

Non-linear system (solve for x, y):

$$r^2 = x^2 + y^2$$
$$rs + b = x + y$$

Tearing

Translate to:

Inner equation: $x = rs + b - y$

Residual equation: $0 = y^2 + x^2 - r^2$

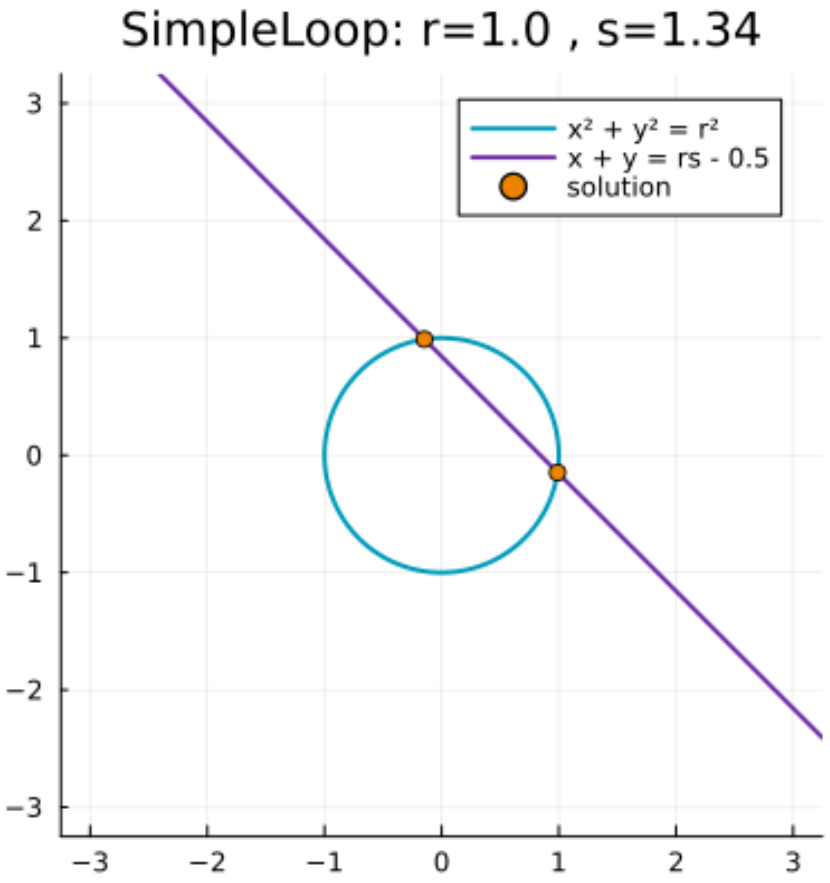
ML Surrogate:

$$y = f_S(r, s, b)$$
$$x = rs + b - y$$

Inputs: r, s, b

Outputs: y

Evaluate Inner Equations: x



WORKFLOW

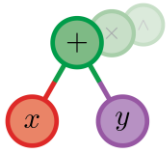
1. Identify equations to replace
2. Generate data
3. Train surrogate
4. Export hybrid model

OpenModelica

julia

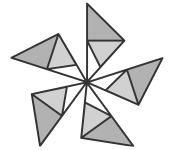
fmi.jl

flux



SymbolicRegression.jl

ONNX

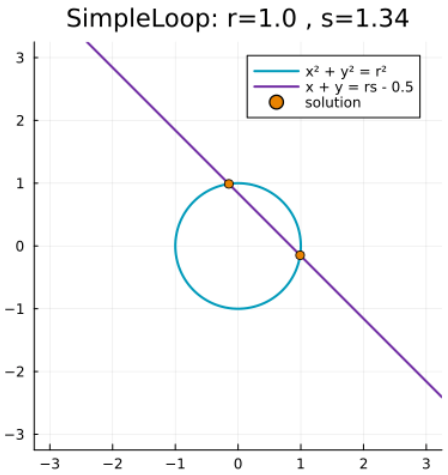


ONNX RUNTIME

Functional Mock-up Interface

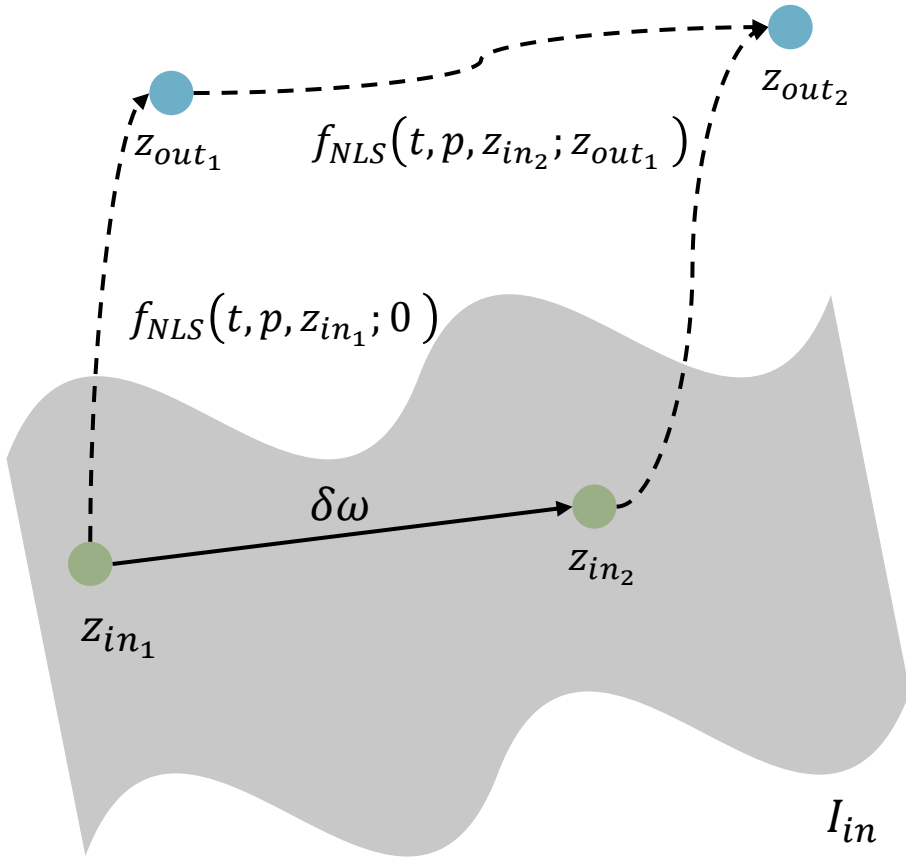
DATA GENERATION

- Fast: Evaluate NLS only
- Issues
  - Data post-processing
  - Ambiguous solutions
  - Start values



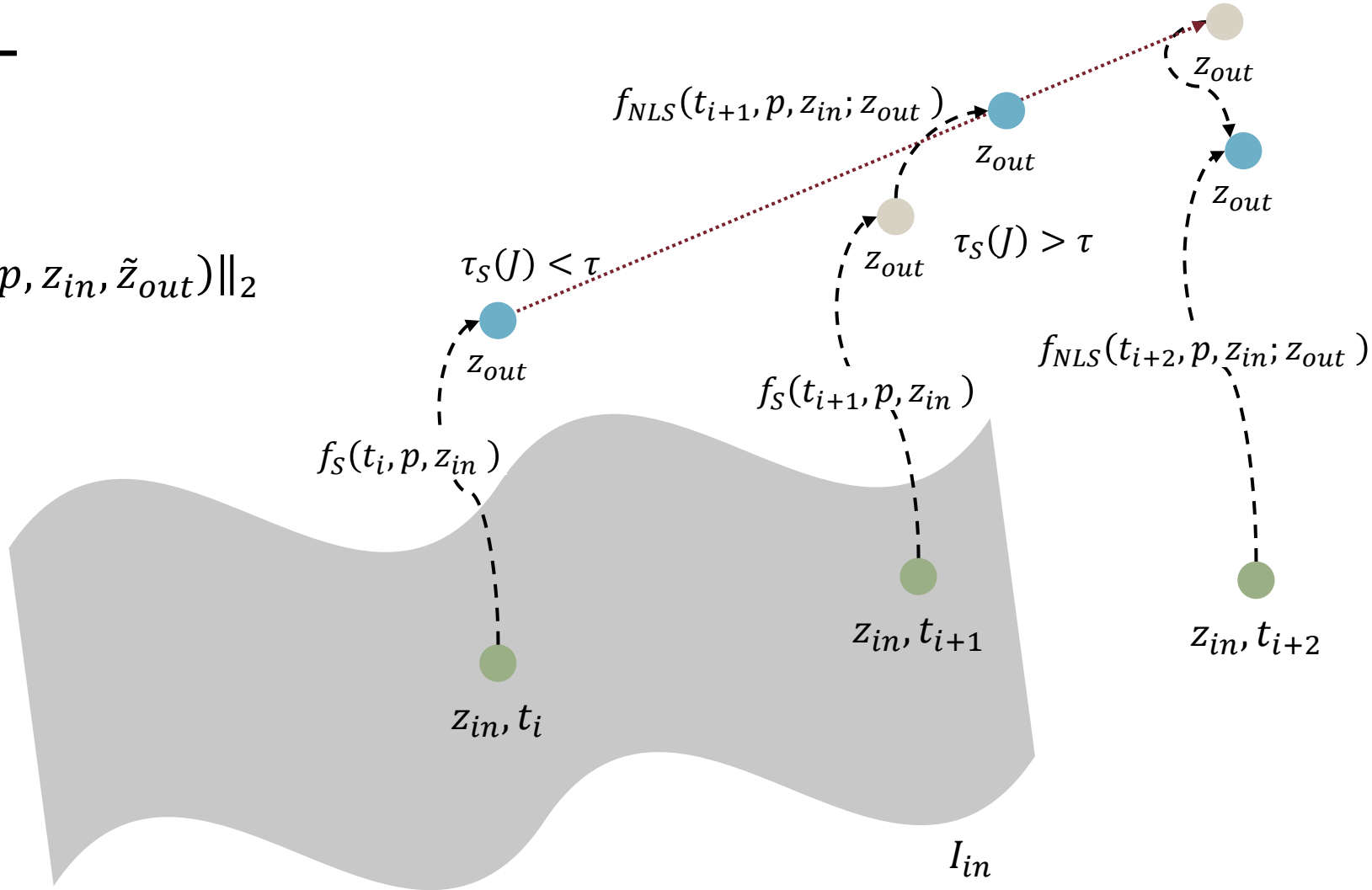
RANDOM WALK

1 **procedure** RandomWalk(δ, Δ_t)
2 $z_{out} \leftarrow 0$
3 $z_{in} \leftarrow$ random value from I_{in}
4 **for** $t = t_{start}, t_{start} + \Delta_t, \dots, t_{end}$ **do**
5 $z_{out} \leftarrow f_{NLS}(t, p, z_{in})$,
using previous z_{out} as start value
6 Save (z_{in}, z_{out})
7 $z_{in} \leftarrow z_{in} + \delta\omega$, where $\omega \in [-1,1]^{n_{in}}$ random
8 Ensure $z_{in} \in I_{in}$



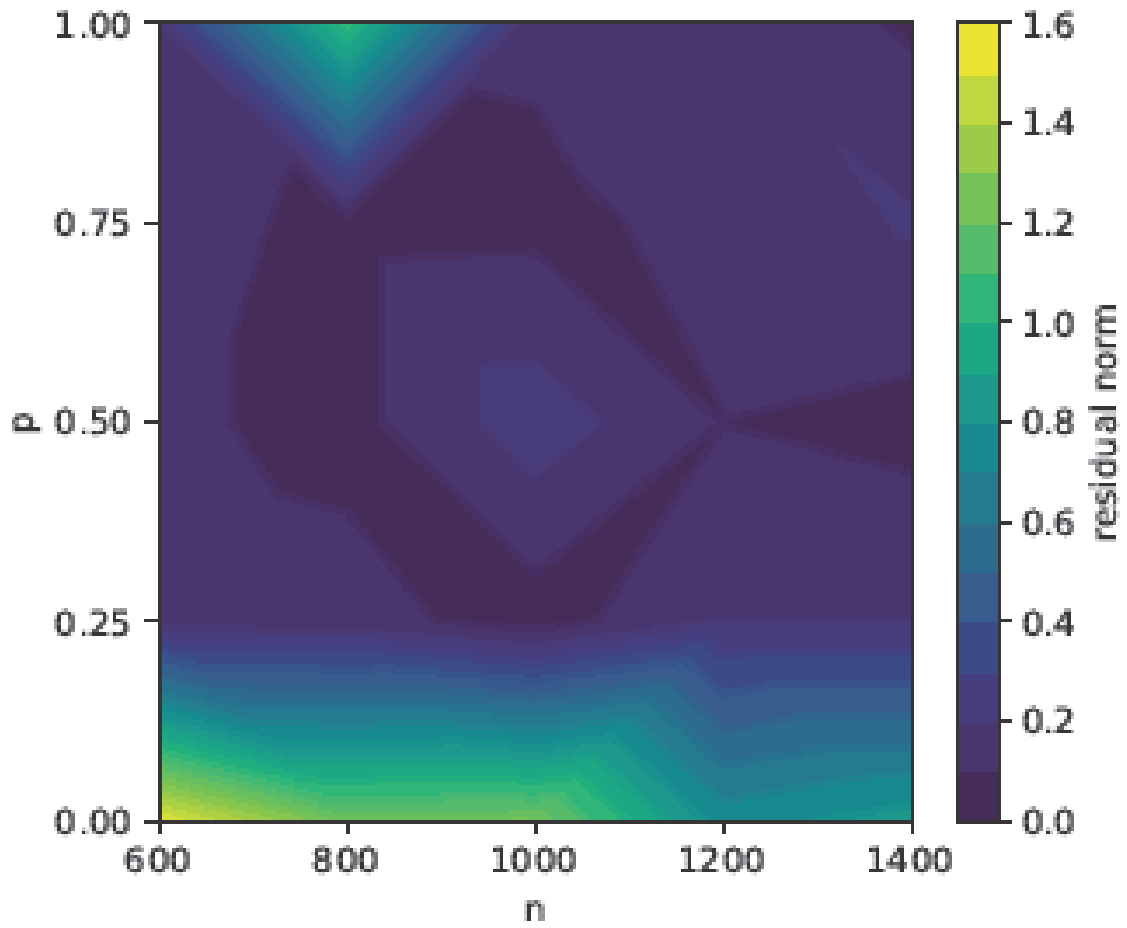
ERROR CONTROL

- Scaled residual error
  $$\tau_S(J) \coloneqq \|s(J) \circ f_{res}(t, p, z_{in}, \tilde{z}_{out})\|_2$$
- Error tolerance τ



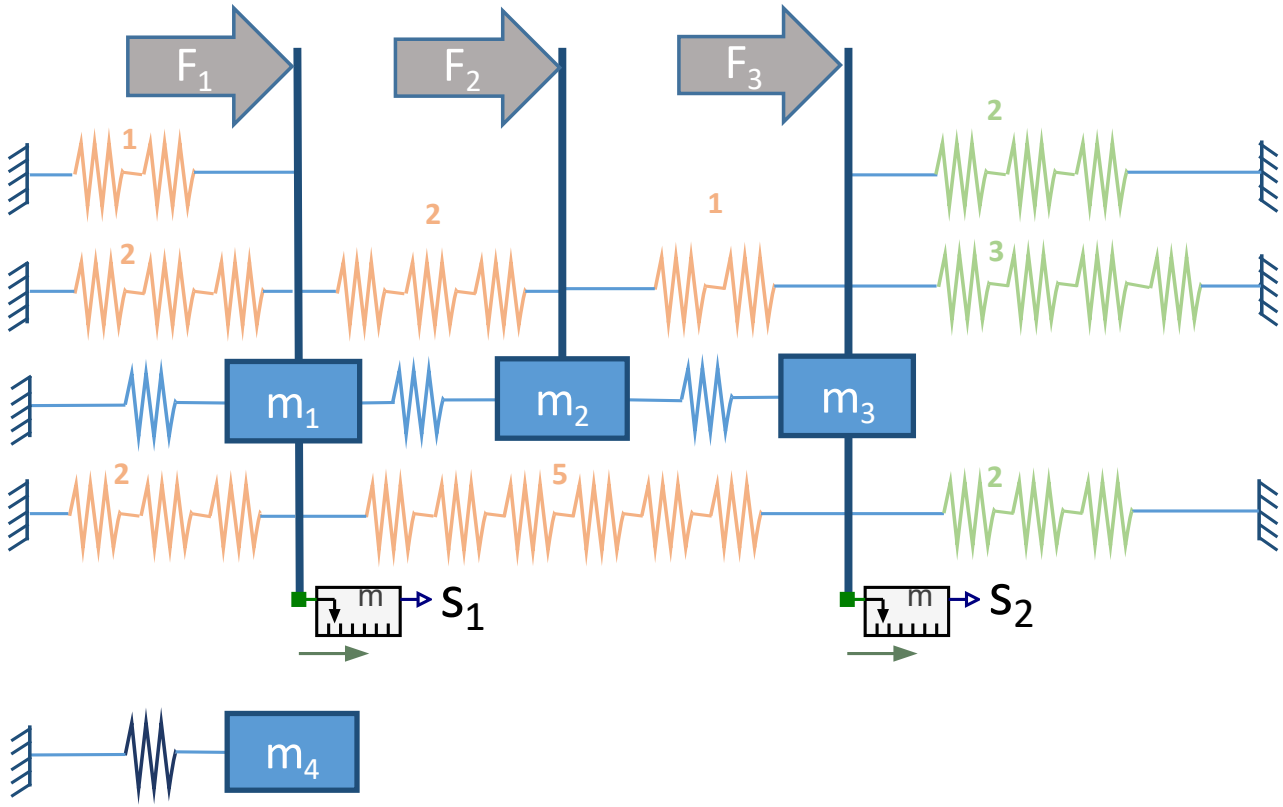
ACTIVE LEARNING

- Generate training data where ANN is weak
- Save training and data generation time
- Genetic bees algorithm to find area of with highest error

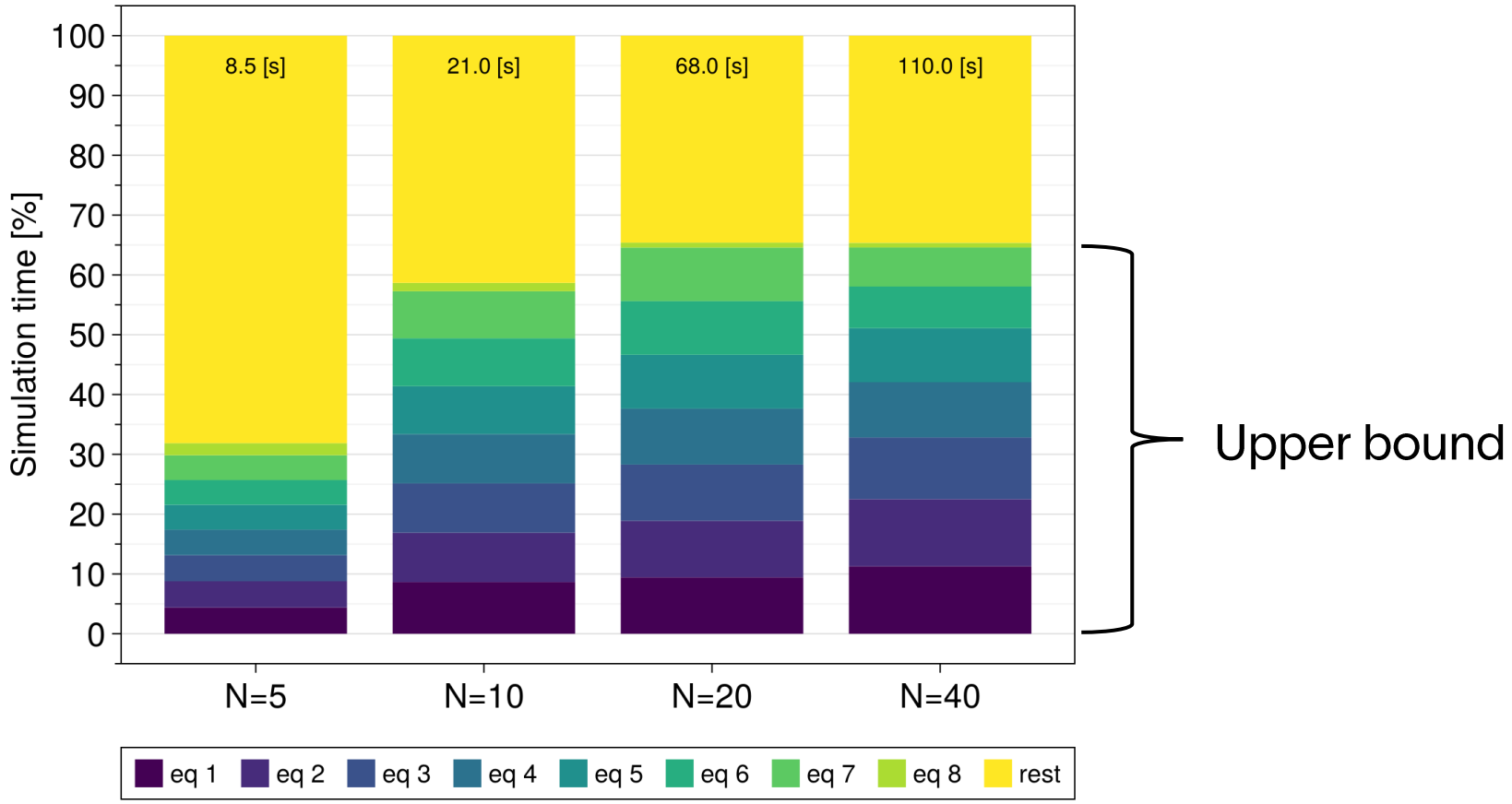


SCALABLE TRANSLATION STATISTICS LIBRARY

- Scalable Translation
  - # Algebraic Loops
  - # States
  - Numeric Jacobians
  - Stiffness
- Open-Source (BSD 3)



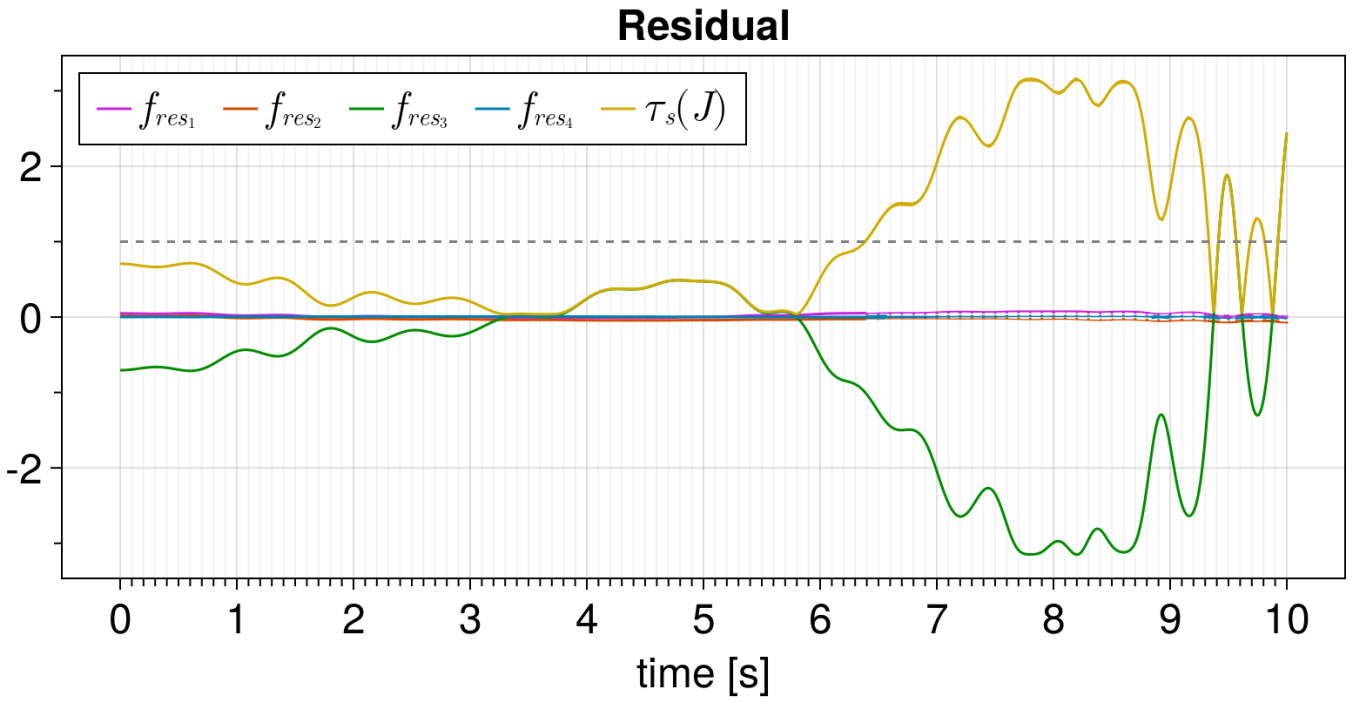
Download: ltx.de/download/ModelicaLibraries/ScalableTranslationStatistics

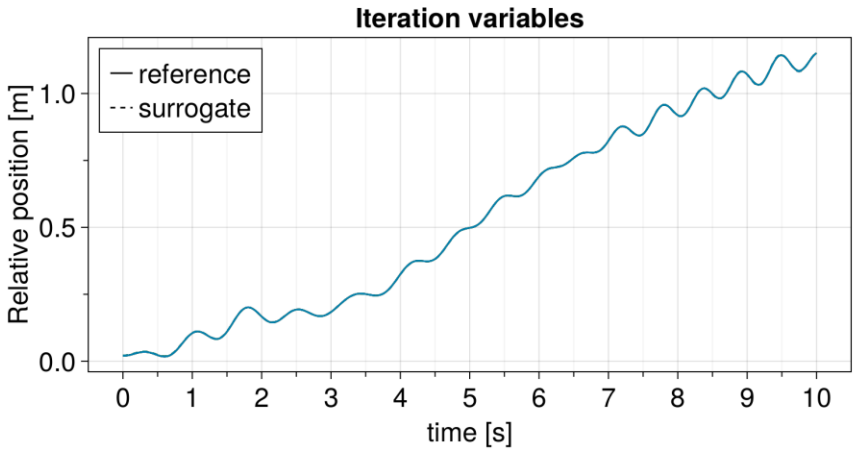


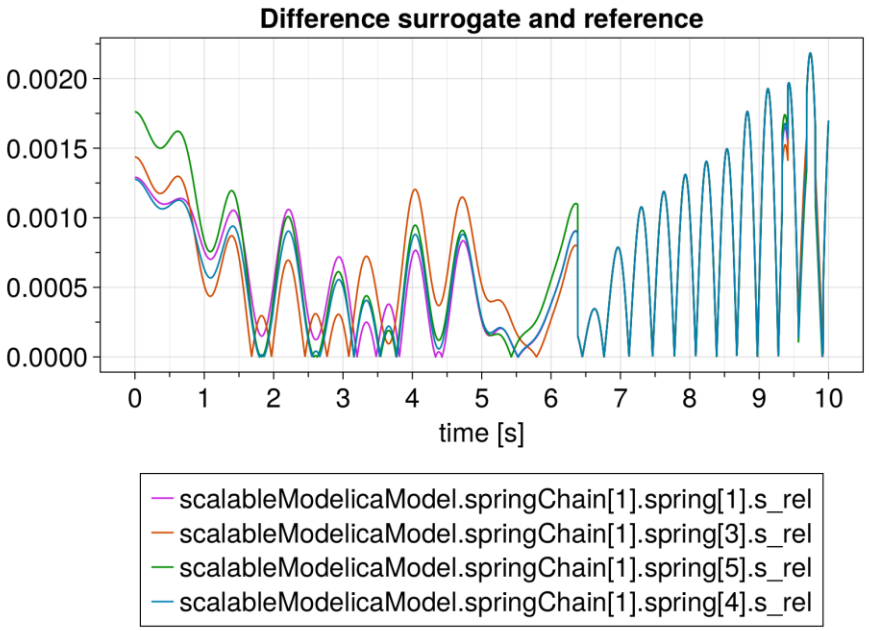
Relative simulation times for
ScalableTranslationStatistics.Examples.ScaledNLEquations.NLEquations_N

$N \approx$ number of iteration variables

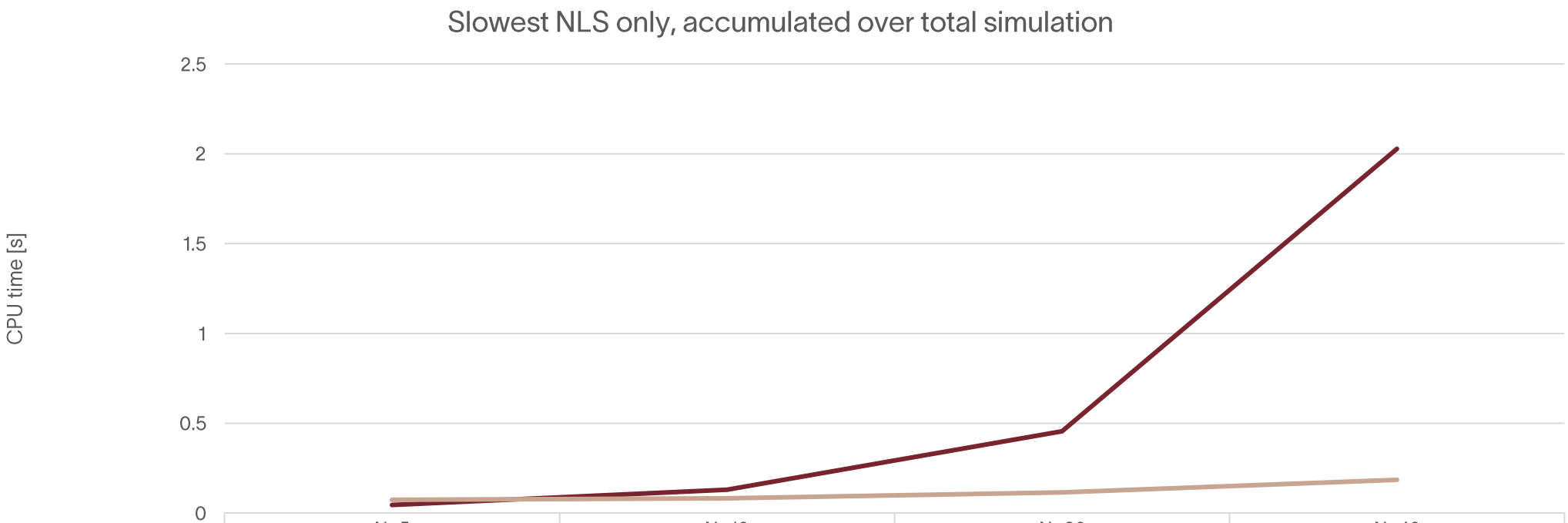
SURROGATE VS REFERENCE SOLUTION





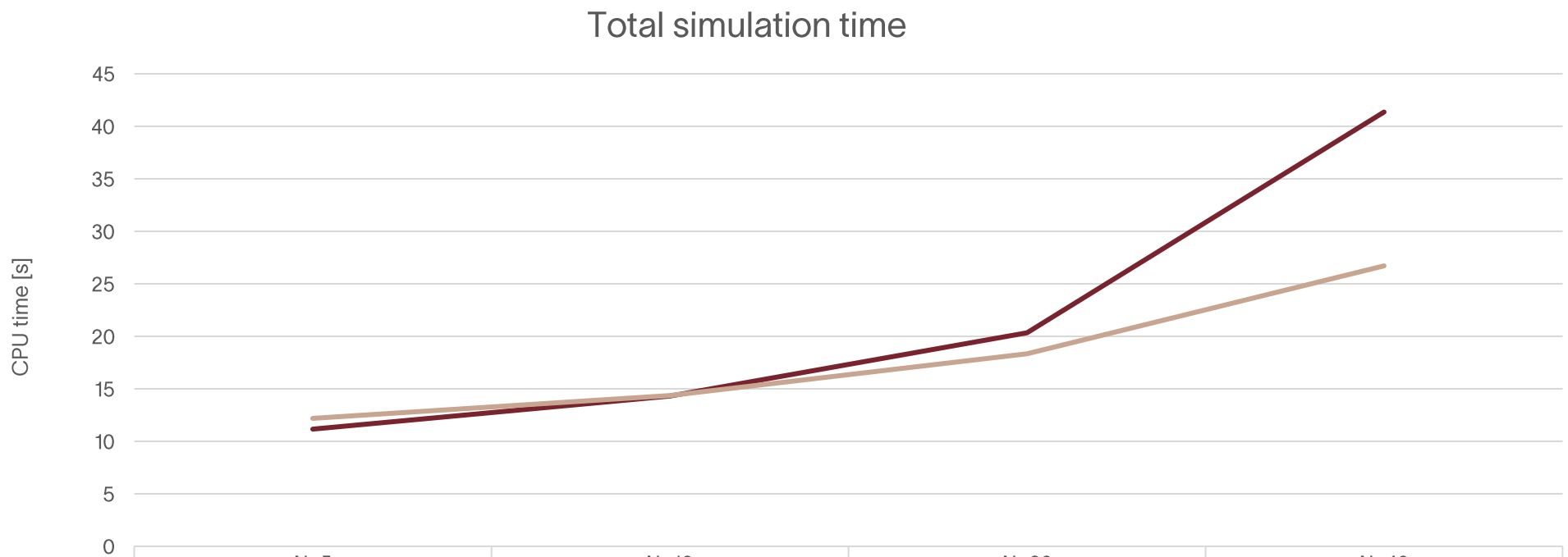


REFERENCE VS SURROGATE
SLOWEST NLS



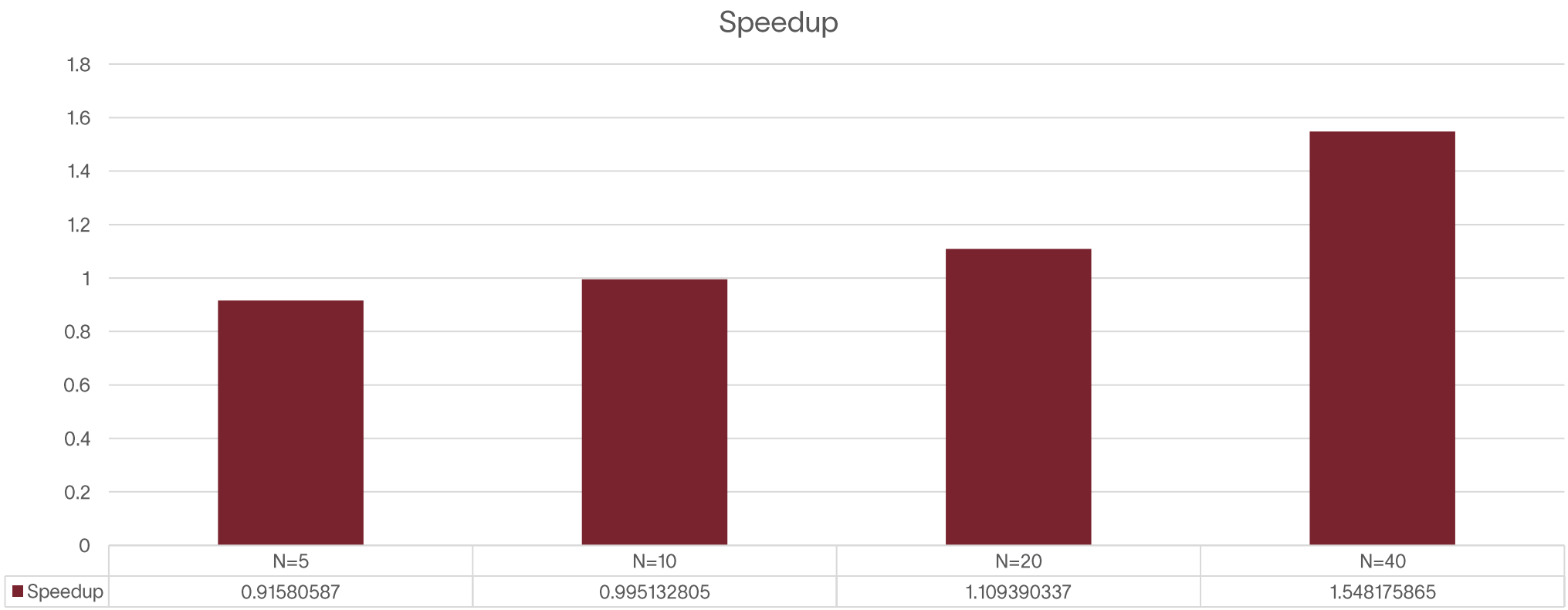
	N=5	N=10	N=20	N=40
Newton-Raphson	0.045782	0.129935	0.454989	2.028005
Surrogate, error ctrl	0.074091	0.081612	0.115263	0.185338

REFERENCE VS SURROGATE
TOTAL SIMULATION TIME



	N=5	N=10	N=20	N=40
Newton-Raphson	11.171	14.312	20.344	41.375
Surrogate, error ctrl	12.198	14.382	18.338	26.725

REFERENCE VS SURROGATE
TOTAL SPEEDUP



SUMMARY

- Speedup simulation time for large non-linear systems
- Upper bound on possible gain
- Simple ML methods lead to decent results
- Error control working

ONGOING WORK

- Error control of ODE solver not compatible with ANN surrogates
- Training surrogates on large and highly non-linear problems difficult
- Maybe loosing numeric stability of ODE

OUTLOOK

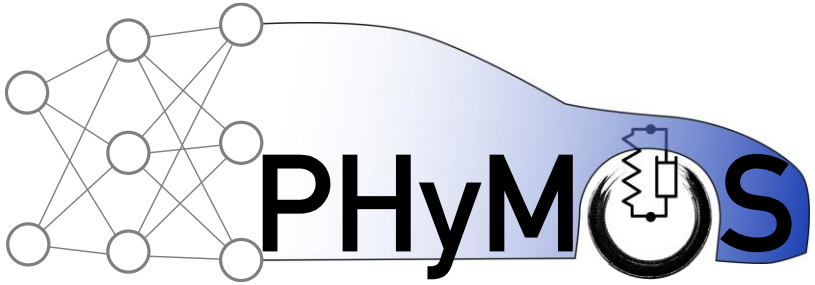
- Trying different ML methods to replace NLS
  - Symbolic regression promising
- Error control needs scaling → needs Jacobian
- Refine training data generation with sensitivity analysis
- Train parameters

HSBI

Hochschule Bielefeld

University of Applied Sciences and Arts

Questions



Funded by the European Union
NextGenerationEU

Supported by:
Federal Ministry for Economic Affairs and Climate Action
on the basis of a decision by the German Bundestag

iDaS

LTX