DAE REDUX

Uri Ascher

Department of Computer Science
University of British Columbia

Outline

1. Numerical DAEs in the 1990's
2. Simulating deformable objects subject to friction and contact
3. Neural DAEs
4. Conclusions

DAE = Differential-Algebraic Equation

Numerical DAEs in the 1990's (selective view)

General form

$$\mathbf{F}(t, \mathbf{u}, \dot{\mathbf{u}}) = \mathbf{0}, \quad a \le t \le b$$

where $\dot{\mathbf{u}} := \frac{d\mathbf{u}}{dt}$ and the Jacobian $\frac{\partial \mathbf{F}}{\partial \dot{\mathbf{u}}}$ may be singular

Semi-explicit DAE $\quad \mathbf{u} = (\mathbf{x}, \mathbf{z})$

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \mathbf{z}) \tag{1a}$$

$$\mathbf{0} = \mathbf{c}(t, \mathbf{x}, \mathbf{z}) \tag{1b}$$

- Can differentiate constraints (1b) repeatedly, until by plugging in (1a) can eliminate algebraic variables $\mathbf{z}$ algebraically
  Index = number of required differentiations + 1

[Brenan, Campbell & Petzold '95; Hairer & Wanner '96]

Example: Equations of motion

- Masses times accelerations equal forces ($\mathbf{v} = \dot{\mathbf{q}}$ in time t)

$$M\dot{\mathbf{v}}(t) = \mathbf{f}_{\mathrm{els}}(\mathbf{q}) + \mathbf{f}_{\mathrm{dmp}}(\mathbf{q}, \mathbf{v}) + \mathbf{f}_{\mathrm{ext}} + \mathbf{f}_{\mathrm{con}}$$

- with the elastic and damping forces

$$\mathbf{f}_{\mathrm{els}}(\mathbf{q}) = -\frac{\partial}{\partial \mathbf{q}} W_{\mathrm{els}}(\mathbf{q}), \quad \mathbf{f}_{\mathrm{dmp}}(\mathbf{q}, \mathbf{v}) = -D\mathbf{v},$$

  where $W_{\mathrm{els}}(\mathbf{q})$ is the elastic potential of the corresponding model
- Given position constraints

$$\mathbf{0} \;=\; \mathbf{c}(\mathbf{q})$$

  we have constraint force $\mathbf{f}_{\mathrm{con}} = -G^T\mathbf{z}$ with $G = \frac{\partial \mathbf{c}}{\partial \mathbf{q}}$ and $\mathbf{z}$ Lagrange multiplier functions
- So to eliminate $\mathbf{z}$ need to differentiate constraints twice: this is an index-3 DAE

Constraint drift-off

Continue with semi-explicit DAE

- By eliminating algebraic variables $\mathbf{z}$ obtain a DE system with algebraic invariant
- However, simply ignoring the invariant manifold may lead to the numerical solution drifting off the constraints
- Various projection methods were proposed [Ascher & Petzold '98]
- Note also that for PDAEs constraint differentiation should be handled with extra care due to spatial boundary conditions [Sidilkover & Ascher '95]

Penalty methods

Continue with semi-explicit DAE

- Alternatively can use a penalty approach viewing the DAE as limit of a singularly perturbed DE

$$\varepsilon \dot{\mathbf{z}} = B\mathbf{c}(t, \mathbf{x}, \mathbf{z})$$

  with parameter $0 < \varepsilon \ll 1$ and matrix function $B = B(\mathbf{u})$ carefully chosen
- However, the singularly perturbed DE can be more complex and demanding than the given DAE

Outline

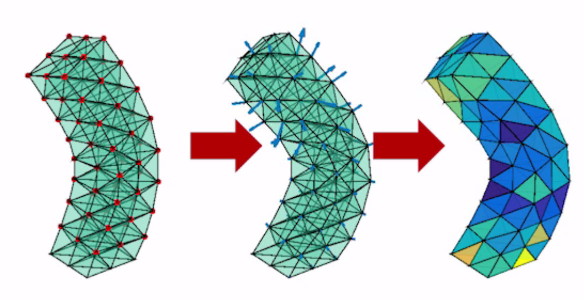
1. Numerical DAEs in the 1990's
2. **Simulating deformable objects subject to friction and contact**
3. Neural DAEs
4. Conclusions

[Larionov, Longva, Ascher, Bender & Pai '22; Larionov's PhD thesis '22]

Deformable object simulation

[Edwin Chen, Dinesh Pai; Danny Kaufman, Dave Levin; Bin Wang, Hui Huang]

- Ubiquitous in current computer graphics animation and robotics research
- High quality simulations can be very expensive to obtain
- For control and fabrication may require more accurate simulations



Deformable object simulation

For a given calibration (material properties):

- Semi-discretize elastodynamics equations in variational form using a finite element method (FEM) on a moving tetrahedral mesh
- Obtain a large ODE system (equations of motion) in time
- Define the tangent stiffness matrix at $\mathbf{q} = \mathbf{q}(t)$ as $K = -\frac{\partial}{\partial \mathbf{q}} \mathbf{f}_{\mathrm{els}}(\mathbf{q})$
- Write as $\dot{\mathbf{u}}(t) = \mathbf{g}(\mathbf{u}(t))$

$$
\begin{aligned}
\dot{\mathbf{u}}(t) \equiv \begin{pmatrix} \dot{\mathbf{q}}(t) \\ \dot{\mathbf{v}}(t) \end{pmatrix} &= \begin{pmatrix} \mathbf{v} \\ M^{-1} \mathbf{f}_{\mathrm{tot}}(\mathbf{q}, \mathbf{v}) \end{pmatrix} \\
&= \begin{pmatrix} 0 & I \\ -M^{-1}K & -M^{-1}D \end{pmatrix} \begin{pmatrix} \mathbf{q}(t) \\ \mathbf{v}(t) \end{pmatrix} + \begin{pmatrix} \mathbf{0} \\ \mathbf{r}(\mathbf{u}(t)) \end{pmatrix}
\end{aligned}
$$

Often there is highly oscillatory stiffness even though the observed motion is damped and is **not seen** to vibrate rapidly

Nonlinear forces

- Elastic forces: Neo-Hookean, StVK, Mooney-Rivlin, ARAP, artist-generated, etc.
- Often arise for large deformations [Ciarlet '88; Sifakis & Barbic '12]
- Stiffness matrix K depends on $\mathbf{q}(t)$ and might become indefinite
- Damping forces: the popular Rayleigh force $\mathbf{f}_{\mathrm{dmp}}(\mathbf{q}, \mathbf{v}) = (a_0 M + a_1 K)\,\mathbf{v}$, with parameters $a_0, a_1 \geq 0$, is good for theory
- Generally, modeling with nonlinear elastic forces as well as damping forces for visual purposes is **still an open problem**

Time discretizations for $\dot{\mathbf{u}} = \mathbf{g}(\mathbf{u})$

At time t we know $\mathbf{u}$ and seek $\mathbf{u}_+$ at $t_+ = t + h$

Can't use forward Euler or RK4 due to stiffness

- Highly damping methods e.g., BDF and Gauss-Radau
  Backward Euler (BE)

$$\mathbf{u}_+ = \mathbf{u} + h\mathbf{g}(\mathbf{u}_+)$$

- Conservative methods e.g., Trapezoidal (TR) and variants
  Implicit midpoint (IM)

$$\mathbf{u}_+ = \mathbf{u} + h\mathbf{g}((\mathbf{u}_+ + \mathbf{u})/2))$$

- In-between methods, still L-stable
  TR-BDF2

$$\begin{aligned} \mathbf{u}_* &= \mathbf{u} + \frac{h}{4}(\mathbf{g}(\mathbf{u}_*) + \mathbf{g}(\mathbf{u})) \\ \mathbf{u}_+ &= \mathbf{u}_* + \frac{1}{3}(\mathbf{u}_* - \mathbf{u} + h\mathbf{g}(\mathbf{u}_+)) \end{aligned}$$

SEMI-IMPLICIT METHODS FOR $\dot{\mathbf{u}} = \mathbf{g}(\mathbf{u})$

Must solve nonlinear algebraic system for $\mathbf{u}_+$ at each time step $[t, t+h]$ — tough for **large** time steps h

Fortunately, **absenting serious contact and friction effects** can often get away with semi-implicit versions

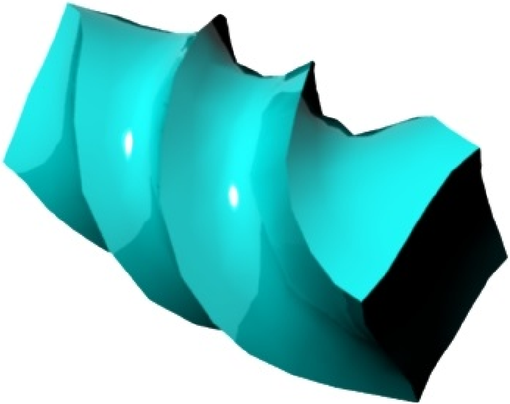
- SI: semi-implicit BE

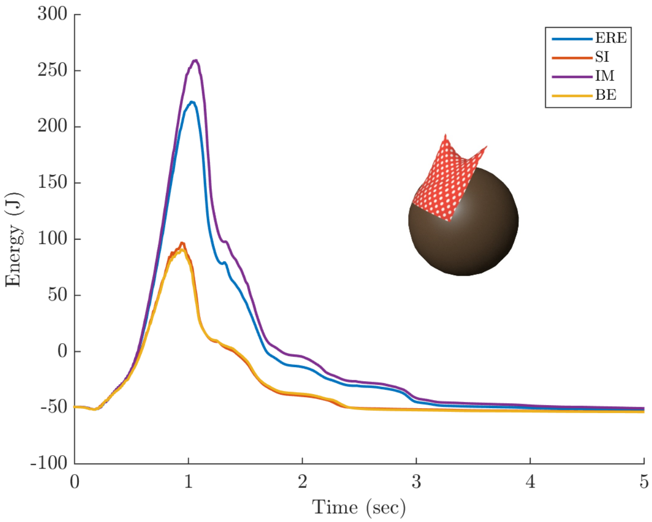
  One iteration of Newton's method for BE step using Jacobian matrix $J = \frac{\partial \mathbf{g}}{\partial \mathbf{u}}$

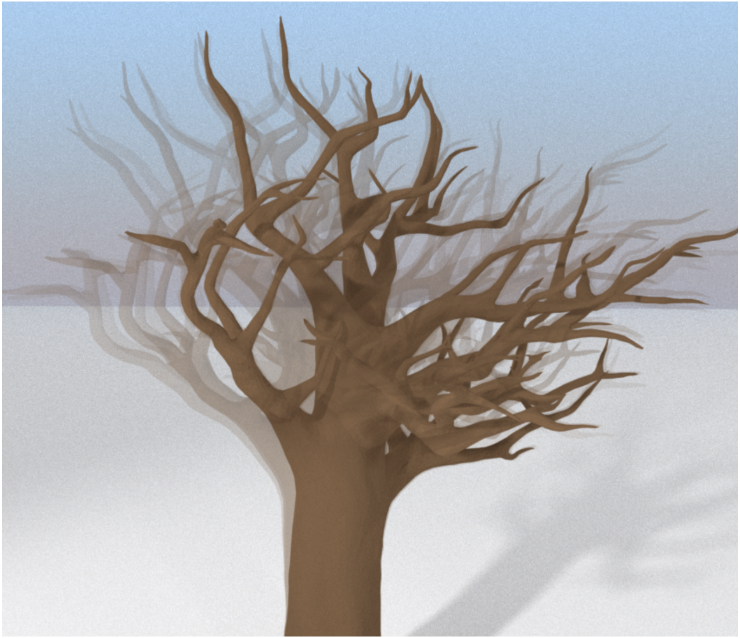
  $$\mathbf{u}_+ = \mathbf{u} + h[I - hJ(\mathbf{u})]^{-1}\mathbf{g}(\mathbf{u})$$

  Solve a <u>linear</u> system of equations for $\mathbf{u}_+$ at each time step
- Exponential methods ERE and SIERE

  [Chen, Ascher & Pai '18; Chen, Sheen, Ascher & Pai '20]




300
250
200
150
100
50
0
-50
-100
Energy (J)
0
1
2
3
4
5
Time (sec)
ERE
SI
IM
BE



Simple damping analysis

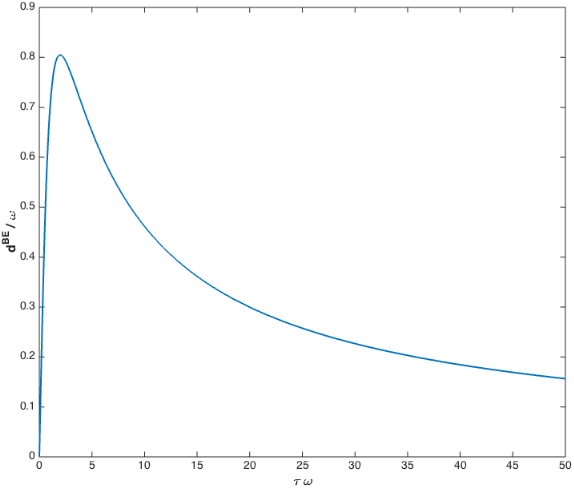
[Chen, Ascher & Pai '18; Ascher, Larionov, Sheen & Pai '21]

- Consider the scalar constant-coefficient ODE

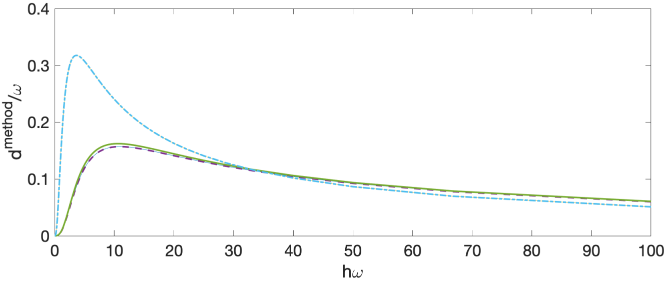
$$\ddot{q} + d\dot{q} + \omega^2 q = 0$$

  where $d \geq 0$ is a damping parameter, and $\omega > d/2$ is a real-valued frequency
- Setting $d = 0$, apply numerical discretization
- Associate resulting decay with artificial damping factor d^{method}

BE ≡ SI artificial damping curve

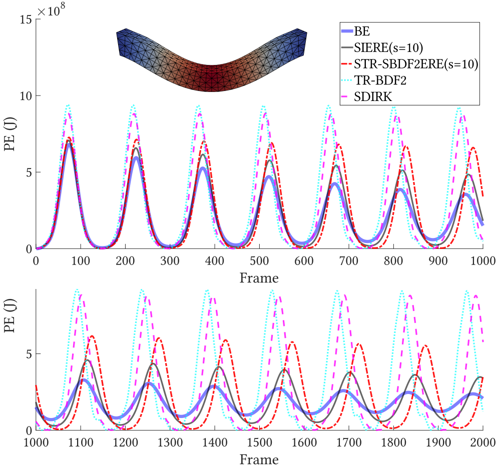


BDF2 and TR-BDF2 artificial damping curves



Damping curves for TR-BDF2 (dashed) and BDF2 (dash-dot)

Comparing methods' energies



Adding contact constraints

e.g. airplane landing, deformable objects colliding

- Must avoid mesh inter-penetration
- Handle inequality constraints that become equality upon contact
- Get DAE on subintervals with sensitive event locations, DAI otherwise
- In this setting penalty (and interior point + CCD) methods become attractive!
- Get a differentiable model

[Li, Ferguson, Langlois, Zorin, Panozzo, Jiang & Kaufmann '20]
[Geilinger, Hahn, Zehnder, Bacher, Thomaszewski & Coros '20]

Contact constraint force

- Penalty function

$$b(x;\delta,\kappa) = \kappa \begin{cases} -(x-\delta^3) & x<\delta \\ 0 & \text{otherwise} \end{cases}$$

- Apply to each contact point $d_i = d_i(\mathbf{q})$ giving contact energy

$$W_{\text{con}}(\mathbf{q}) = \sum_i b(d_i;\delta,\kappa)$$

- Obtain contact force

$$\mathbf{f}_{\text{con}} = -\frac{\partial W_{\text{con}}}{\partial \mathbf{q}}$$

Time-stepping via "optimization"

For constrained deformable objects must at each time step solve nonlinear algebraic system carefully

- Consider e.g. BE for elastic plus contact forces

$$\begin{aligned} M(\mathbf{v}_+ - \mathbf{v}) = h\mathbf{f}(\mathbf{q}_+, \mathbf{v}_+) &= h\left(\mathbf{f}_{\text{els}}(\mathbf{q}_+) + \mathbf{f}_{\text{con}}(\mathbf{q}_+)\right) \\ &= -h\left(\frac{\partial W_{\text{els}}}{\partial \mathbf{q}} + \frac{\partial W_{\text{con}}}{\partial \mathbf{q}}\right) \end{aligned}$$

- So $\mathbf{v}_+$ solves the optimization problem

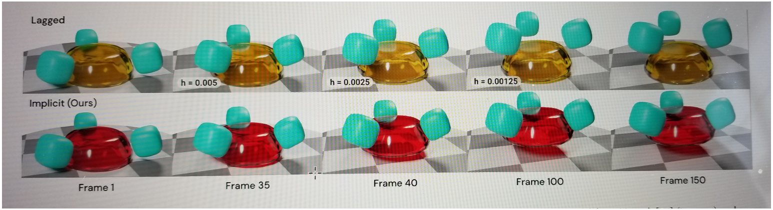
$$\min_{\mathbf{v}_+} \frac{1}{2}\|\mathbf{v}_+ - \mathbf{v}\|_M^2 + h\left(W_{\text{els}}(\mathbf{q}_+) + W_{\text{con}}(\mathbf{q}_+)\right)$$

  where $\mathbf{q}_+ = \mathbf{q} + h\mathbf{v}_+$

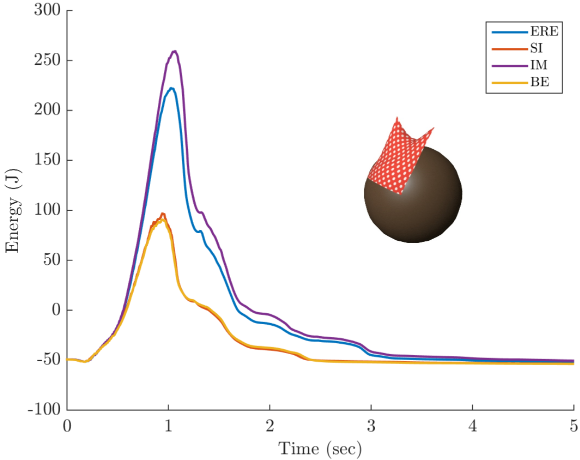
- Potentially neater than solving the nonlinear equations using more general controlled damped Newton

Friction constraint force

Friction constraint force

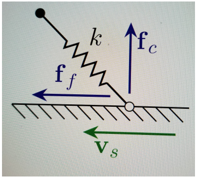


- Solution for Coulomb force is an inclusion which is difficult to handle and non-smooth
- Approximate it using a smoothed penalty model
- *Can't apply "optimization" approach*
- Instead of using a lagged frictional contact (which uses **q** for evaluating friction force) use damped Newton for the "fully implicit" equations



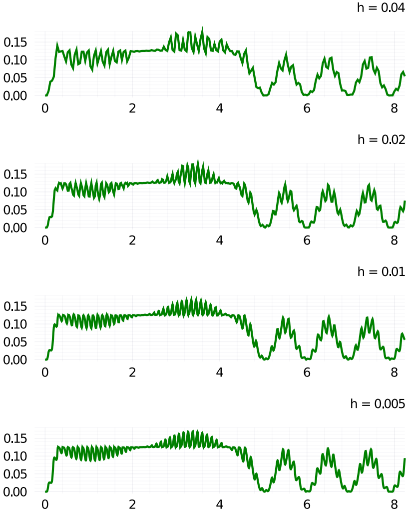
- With lagged friction might drop the bowl
- With lagged friction + TR might get instability
- Using inexact Newton
- Two real-world examples

Possible instabilities?

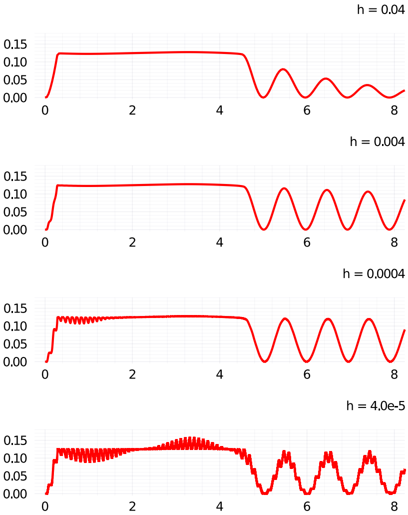


- Yes they can arise
- Fortunately they are local and overall stability not affected
- Less noticed for highly damping integrators, more for conservative ones
- Analysis in Larionov's thesis and our paper

Local instability using TR



Local instability using BE



TR-BDF2 is in between TR and BE

Outline

[Boesen, Haber & Ascher '22]

Neural ODE

- Statistical learning: given data pairs $(\mathbf{x}_i, \mathbf{y}_i)$, $i = 1, \ldots, n$ find a function $\mathbf{h}$ depending on parameters $\boldsymbol{\theta}$ s.t.

$$\mathbf{h}(\mathbf{x}, \boldsymbol{\theta}) = \mathbf{y} \qquad \mathbf{x} \in X, \ \mathbf{y} \in Y$$

Look at neural networks for $\mathbf{h}$

- A residual neural network may be considered as a simple discretization of a large initial-value ODE system

$$\begin{aligned} \mathbf{u}(0) &= K_o \mathbf{x} \\ \frac{d\mathbf{u}}{d\tau} &= \mathbf{f}(\mathbf{u}, \boldsymbol{\theta}) \quad 0 \le \tau \le 1 \\ \mathbf{y} &= K^o \mathbf{u}(1) \end{aligned}$$

Next, suppose we also know $\mathbf{c}$ such that $\mathbf{c}(\mathbf{y}) = \mathbf{c}(K^o \mathbf{u}(1)) = \mathbf{0}$

Neural ODE

- Statistical learning: given data pairs $(\mathbf{x}_i, \mathbf{y}_i), \quad i = 1, \ldots, n$ find a function $\mathbf{h}$ depending on parameters $\boldsymbol{\theta}$ s.t.

$$\mathbf{h}(\mathbf{x}, \boldsymbol{\theta}) = \mathbf{y} \qquad \mathbf{x} \in X, \; \mathbf{y} \in Y$$

Look at neural networks for $\mathbf{h}$

- A residual neural network may be considered as a simple discretization of a large initial-value ODE system

$$\begin{aligned} \mathbf{u}(0) &= K_o \mathbf{x} \\ \frac{d\mathbf{u}}{d\tau} &= \mathbf{f}(\mathbf{u}, \boldsymbol{\theta}) \quad 0 \le \tau \le 1 \\ \mathbf{y} &= K^o \mathbf{u}(1) \end{aligned}$$

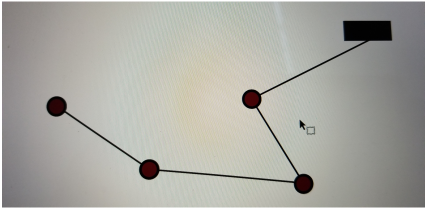
Next, suppose we also know $\mathbf{c}$ such that $\mathbf{c}(\mathbf{y}) = \mathbf{c}(K^o \mathbf{u}(1)) = \mathbf{0}$

Constrained neural ODE

- Suppose we also know $\mathbf{c}$ such that $\mathbf{c}(\mathbf{y}) = \mathbf{c}(K^o\mathbf{u}(1)) = \mathbf{0}$
- Can constrain $K^o\mathbf{u}(\tau)$ in a lower dimensional physical space so

$$\mathbf{c}(K^o\mathbf{u}(\tau)) = \mathbf{0} \qquad 0 \leq \tau \leq 1$$

- May want to constrain the residual neural network to better respect the physical constraint



RESULTS SAMPLE

	Constraints	$n_t = 100$		$n_t = 1000$		$n_t = 10000$	
		MAE	CV	MAE	CV	MAE	CV
$k = 100$	No constraints	17.7	10.3	7.39	5.06	1.54	1.15
	Auxiliary loss $\eta = 3$	17.7	7.41	7.77	4.35	1.51	0.94
	Penalty $\gamma = 3$	17.5	9.65	7.01	3.99	1.63	0.96
	End con $\gamma, \eta = 3$	14.0	0.00	5.70	0.00	1.29	0.00
	Smooth con $\gamma, \eta = 3$	**11.2**	0.00	**3.35**	0.00	**1.02**	0.00
$k = 200$	No constraints	46.1	25.2	26.1	13.4	3.26	2.30
	Auxiliary loss $\eta = 1$	46.1	20.5	25.9	10.4	3.32	2.03
	Penalty $\gamma = 1$	45.5	24.2	25.9	13.1	3.42	2.34
	End con $\gamma, \eta = 1$	42.6	6.91	27.8	2.66	3.04	0.00
	Smooth con $\gamma, \eta = 1$	**33.3**	0.00	**19.1**	0.00	**2.21**	0.00

TABLE: Mean absolute error (MAE) and constraint violation (CV), in cm, over a test set of 1000 samples on the multi-body pendulum problem. n_t is the number of training samples, while k is the number of steps predicted ahead. Each experiment was repeated three times and the average of those runs are shown.

Neural DAE

- A residual neural network may be considered as a simple discretization of a large initial-value ODE system
- Recall that a given DAE in the physical world can often be considered as an ODE with an algebraic invariant
- Ignoring the invariant we can construct a neural ODE also for a physical semi-explicit DAE
- Moreover *the traditional issue of drift off the constraint is possibly avoided because the NN is learned from solution examples independent of constraint differentiation*

Outline

1. Numerical DAEs in the 1990's
2. Simulating deformable objects subject to friction and contact
3. Neural DAEs
4. **Conclusions**

DAE = Differential-Algebraic Equation

Conclusions

- Old understanding of the nature of DAEs and of methods for handling them numerically has proved useful when facing new complex applications
- New perspectives had to be introduced to handle these new situations