Research directions for combination of AI/Simulation Tasks

All described methods below reducing effort, either temporally or spatially, to solve complicated mechanical systems. They are not essential for solving the proposed problems, but using them can enhance and release capacity from resources when dealing with non-linear and/or multiscale problems.

- Ease of calculation and calculations with high accuracy
- Reduction of computational efforts
- Obtaining new insight from a complicated domain (dynamic learning)

Methods based on AD (Automatic Differentiation)

We have talked about this point in the past conversations. Using AD is beneficial in increasing the accuracy of differentiation techniques as well as reducing the amount of code one has to write manually. This is because AD packages trace the computation automatically. These methods, however, could become memory intensive and often require multi-core solutions to make of for their lack of speed.

Possible applications of the method can be at the level of quad-points for calculation of consistent tangent, for example, where the constitutive laws are complicated and/or highly nonlinear.

I think in terms of applications we both have a good idea of how this can work. You have provided a lot of papers for this already, but I think it still has some novelty if we can employ the

Methods that treat a subset of a bigger problem, for example, stress update at mesoscale.

A NN (Neural Network) is simply a black box. With enough capacity (wide enough hidden layer, enough number of neurons) they can represent any function. This has to do "universal approximation property".

To learn the underlying dynamics of a system, NNs can be trained to learn the behavior of a system of the form $\dot{x} = F(x, t)$. The training is done via supervised learning, where we feed the network pairs of (x_t, x_{t+1}) at discrete time steps, for example:

- Input is (x_t, dt) and output is x_{t+1}
- Input is $(x_{t-N}, \ldots, x_t, dt)$, and output is x_{t+1}

More specifically, for the purpose of our conversation, one can imagine the update of internal variables in the rate form $\dot{\xi} = f(\mathbf{F}, \xi)$ to be done via this method, where stress if a function of deformation and internal variables, $\boldsymbol{\sigma} = g(\mathbf{F}, \xi)$.

Application of this can be thought of using these solvers to approximate the solution of constitutive equations at the level of quadrature points. The constitutive laws usually

have a rate form (they are linear or nonlinear ODEs) and they can be approximated using the described method above, i.e., devising a network that can learn the response of these equations. This is be beneficial for nonlinear equations with large degrees of freedom that require lots of iterations to solve. The trained method can replace the exact method with certain accuracy. I guess these methods (black-box solution of ODEs) will be more useful for approximate solution techniques, where the accuracy can be traded for gains in speed.

There are a couple of challenges for these methods when applied to engineering problems:

1. How to account for the change of parameters of the ODE. If we train the model of a set ODEs, how can we transfer these solutions to the ODE of same kind but with different parameters.
2. Related to above question, does the tuning for linear and nonlinear problems differ, and if so (the answer is most likely positive), how much larger of an effort the tuning of nonlinear parameters are?
3. Can these methods be applied to problems with adaptive time steps? By that I mean that we train the method for a time step Δt_0 and then we use it for a different time step Δt_1, where $\Delta t_1 \neq \Delta t_0$.

There are a lot work on this area done in MIT on these methods. Some literature survey is needed to see if any of these concerns are answered or not.

This work is related to the topic (using NNs as a black) but not exactly the same as what I discussed above. However, I put it here because it is still semi-relevant.

- STRESS FIELD PREDICTION IN CANTILEVERED STRUCTURES USING CONVOLUTIONAL NEURAL NETWORKS

Methods based on auto-encoders for capturing the dynamics of the domain

An auto-encoder (AE) is a NN architecture that when trained of a set of data, can recover the original data. In other words, an AE can learn the probability distribution of the underlying data and act as an identity function on the data point $x \xrightarrow{E} I \xrightarrow{D} x$. Here E and D are the encoder and decoder parts of the AE network architecture, respectively. The middle representation of the original data point, I (the bottleneck), has usually a lot less information in it, i.e., its dimension compared to the original data is smaller. AE are useful for dimensionality reduction, as well as exploiting the dynamics of underlying system under study. These two aspects, nonlinear dimensionality reduction and subspace simulation, are highlighted in the following publications below:

- Latent-space Dynamics for Reduced Deformable Simulation
- Deep learning approach based on dimensionality reduction for designing electromagnetic nanostructures

- Recovering missing CFD data for high-order discretizations using deep neural networks and dynamics learning

I still have to find a viable and promising simulation situation where we can use AE to our benefit. These are my two hunches at the moment:

1. Use the bottleneck space information to simplify the dynamics of nonlinear system in the original space. For example, imagine a nonlinear constitutive equation with more than hundred variables (crystal plasticity + transformation will give you that for start). Can we project these information and learn a nonlinear function with less degrees of freedom in the bottleneck space (between 5~10 variables). We solve constitutive law/ODE equations in this space and then project them back to the original space.
2. Can we use this for multiscale simulations where we use the information in the bottleneck layer for compression and simplification of underlying microstructure?

Learn a good preconditioned for a linear system

To be researched/discussed later.