SAND23XX-XXXXR
LDRD PROJECT NUMBER: 231002
LDRD PROJECT TITLE: MACHINE-LEARNED LINEAR STRUCTURAL DYNAMICS
PROJECT TEAM MEMBERS: PAYTON LINDSAY (PI), TIMOTHY SHELTON, KENDALL PIERSON (PM), ROBERT KUETHER, DAVID NAJERA-FLORES, JUSTIN WILBANKS, ERIC PARISH

ABSTRACT:

The tension between accuracy and computational cost is a common thread throughout computational simulation. One such example arises in the modeling of mechanical joints. Joints are typically confined to a physically small domain and yet are computationally expensive to model with a high-resolution finite element representation. A common approach is to substitute reduced-order models that can capture important aspects of the joint response and enable the use of more computationally efficient techniques overall. Unfortunately, such reduced-order models are often difficult to use, error prone, and have a narrow range of application. In contrast, we propose a new type of reduced-order model, leveraging machine learning, that would be both user-friendly and extensible to a wide range of applications.

INTRODUCTION AND EXECUTIVE SUMMARY OF RESULTS:

There are two common types of reduced-order models commonly used in linear structural dynamics. The first is general linear reduced-order models, with Craig-Bampton reduction[1] and component-mode synthesis[2] being two well-known examples. These reduced-order models are well-suited for representing a general domain, as well as being computationally efficient and accurate. However, they can only capture linear behavior and thus are a poor choice for more complex models involving mild nonlinearities such as the joint models as mentioned above.

Nonlinear reduced-order models have been developed for linear structural dynamics,[3,4] but they are typically focused on a narrow use case. Nonlinear reduced-order models frequently suffer from poor usability, require extensive "tuning" to fit expected behavior, and are not extensible to domains beyond what they were developed for. The hypothesis of this exploratory LDRD is that machine-learned reduced-order models can offer an attractive third option to analysts. Namely: a reduced-order model that offers the same usability and generality benefits as linear reduced-order models, but also offers the potential for additional extensibility beyond what linear reduced-order models currently support.

Our core development team has prior experience developing machine-learned reduced-order models for fully nonlinear quasi-statics under the Advanced Simulation and Computing Advanced Machine Learning initiative. That work was also foundational to several subsequent research efforts.[5,6] However, those previous efforts have all focused on predicting static or quasi-static response via learning effective stiffness properties. As mentioned above, a key departure of this work from the previous state of the art is in the prediction of dynamic response

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.

via learning the mass properties of a system. Specifically, we are trying to learn the *dynamic* equation of motion,

$$Ma + Kd = f$$

where M is the mass matrix of the system, $K$is the system stiffness matrix, f represents the external forces on the system, d is the system displacement, and $a = \ddot{d}$ is the acceleration. Previous efforts had focused on solving the steady state or *static* equation of motion,

$$Kd = f$$

which ignores any effects of the system mass or acceleration.

The finite element simulations for training the models were obtained using the Sierra/SD finite element code,[7] which is a part of the Sierra simulation code suite developed at Sandia National Laboratories. Sierra/SD provides a massively parallel implementation of structural dynamics finite element analysis and is frequently used for high fidelity, validated models used in modal, vibration, static, and shock analysis of weapons systems. While the trained models could be exercised in isolation, one of the main goals of this project was to provide an interface for running machine-learned submodels embedded into larger structural dynamics simulations. For this goal, we again focused our efforts towards Sierra/SD. Firstly, we use a lightweight interface layer that allows TensorFlow,[8] Keras,[9] and PyTorch[10] models to be ran natively in C++ application codes. This interface layer was built on top of the pocket-tensor library[11] with extensions added for our use case. Additionally, we developed a suite of end-user focused workflow tools for training the models, which can be challenging to new users. Finally, this work was developed using agile software best practices, and we have also developed an extensive suite of tests of the capability that are run nightly. The culmination of this effort is the development of a robust and user-friendly embedded machine-learning submodeling capability in Sierra/SD. This capability provides users with an attractive alternative to the existing Craig-Bampton reduced-order modeling capability and is extensible to incorporate a variety of training data as well as capturing more complex phenomena in a way that the existing offerings cannot easily replicate.

DETAILED DESCRIPTION OF RESEARCH AND DEVELOPMENT AND METHODOLOGY:

As mentioned above, our research departed from previous endeavors by learning *dynamic* system matrices. Recalling again the finite element equation of motion, we seek to learn an effective representation of the following system of equations:

$$M_{ML}a + K_{ML}d = f$$

where M_{ML} is a machine-learned approximation of the system mass matrix and K_{ML} is the machine-learned stiffness matrix approximation. To learn both an effective mass and stiffness, we use a two-stage learning procedure where we first learn the stiffness and then the mass.

Solving for the stiffness matrix proceeds as follows. A series of *static* finite element simulations are conducted, solving the following linear system of equations:

$$K_{FEM}d = f$$

where K_{FEM} is simply the standard finite-element stiffness matrix, which is derived from the element formulation and material properties of the system.

Subsequently, the system displacement d and total force f (comprised of the sum of reaction and external forces on the system) are used an inputs and outputs for training a machine-learned model, with the goal of learning an effective system matrix $K_{ML}d \approx f$.

After the stiffness matrix has been learned, we proceed to learning a mass matrix. To do so, we again conduct a series of finite element simulations. However, our training data now utilizes the results of a *dynamic* simulation as opposed to the static simulations that were used to learn the stiffness matrix,

$$M_{FEM}a + K_{FEM}d = f$$

where M_{FEM} is again the standard finite-element mass matrix.

Once again, we use the outputs from these dynamic simulations as inputs and outputs for training a machine-learned model. However, as opposed to learning the stiffness matrix, the input is now *acceleration* a and output $f - K_{ML}d$. Note the change from K_{FEM} to K_{ML}, meaning that we are feeding the results of the machine-learned stiffness prediction model into the training set of the second, mass machine-learned model. This approach eliminates the need for (potentially application intrusive) knowledge of the system matrices when training their machine-learned surrogates. This approach also enables the seamless fusion of data from disparate sources such as nonlinear simulations or physical experiments, which is something our team would like to investigate further in follow-on work.

When training our machine-learned surrogate models, we found that a stochastic gradient descent optimizer was a good choice of optimizer and significantly outperformed other popular choices such as Adam. We also found an adaptive learning rate to significantly improve the speed and robustness of the fit. Finally, we utilized PyTorch as our machine-learning framework due to ease of customization and straightforward application of constraints on the learning approach.

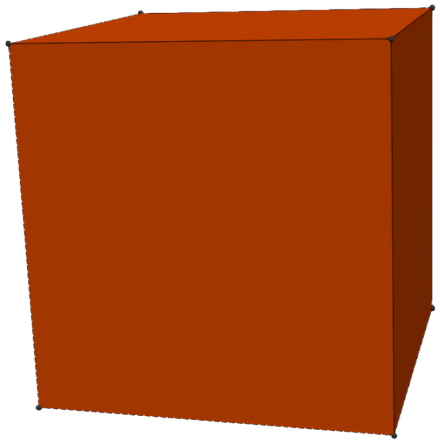
In this initial work, we chose to focus on a single dense layer as our neural network architecture. As expected, this choice proved more than adequate for predicting linear response. We also discovered that using double precision for our machine-learned models was essential to obtaining good accuracy and solution robustness, especially for eigen-analysis simulations.

Another significant departure from our previous work was in the decision to incorporate physical properties of the system directly into our training procedure (e.g., a *physics-informed neural network*). The desired properties were enforced via a custom loss function in our machine-learned model. Recall that the model is predicting a *matrix* quantity. When training the stiffness surrogate, we enforce through a penalty approach that the stiffness prediction should be symmetric, the diagonal entries should be positive, and there should be six rigid-body (zero-energy) eigenvalue modes. These virtues are all properties of well-formed stiffness matrices that we would like our machine-learned surrogate to have as well. Likewise, for the mass matrix surrogate model, we enforce predications to be symmetric, have positive diagonal entries, and the overall sum (a multiple of the total mass) should be greater than zero.

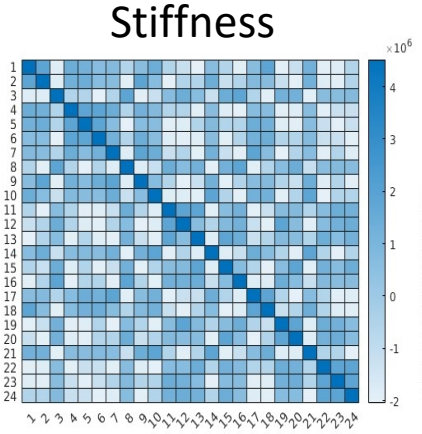
We have found including physical properties in this way to lead to more accurate results, as well as reduced training iterations. Additionally, encapsulation of these properties in the machine-learned model itself enables a less intrusive implementation when compared to enforcing them posteriorly in the finite element application code. Finally, this approach is easily modified to include more (or less) constraints based on experimental results, physical quantities, element formulation, etc.

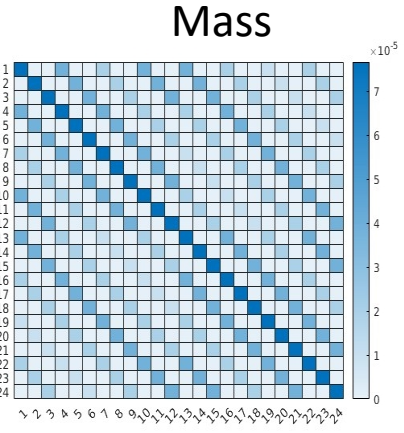
RESULTS AND DISCUSSION:

We first consider the problem of learning the mass and stiffness properties of a single eight-noded linear hexahedral element (shown below). The dimension of the element is 2”x2”x2”, and an isotropic material model was used with aluminum material properties (Young’s modulus 1e7 psi, density 0.1 lb/in^3, Poissons Ratio 0.35). In this case, the exact finite element mass and stiffness matrices are known, so we can compare them directly to their machine-learned surrogates.



Single hex element

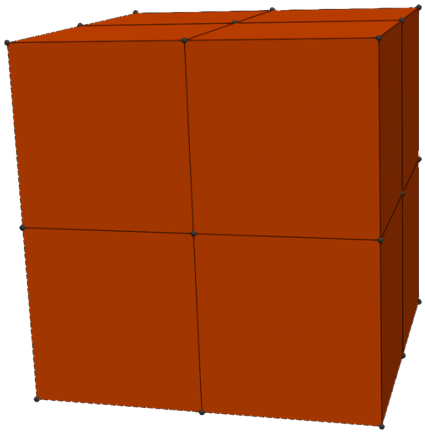




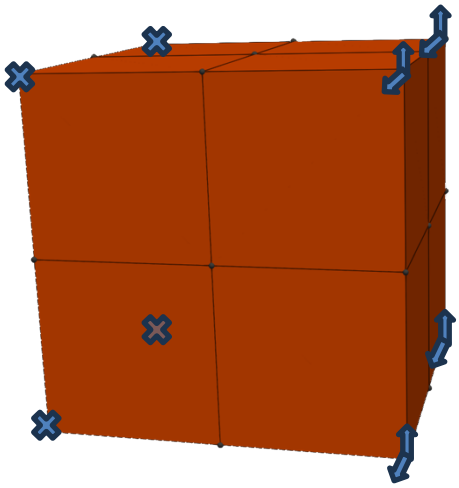
The training procedure described above is used here: a series of static and transient simulations were conducted to obtain displacements, accelerations, and forces that were in turn used to train machine-learned models to approximate the mass and stiffness matrix of the element. In this case, a training procedure was used by which a sine-wave forcing function with a maximum amplitude of 1e9 was applied to a single degree of freedom at a time (i.e., a single node, in a single x/y/z direction) over a series of 64 time steps of size 0.01, while the remaining degrees of freedom were held fixed. In the case of the static simulations, the density of the material model was set to zero, leading to a sequence of *quasistatic* simulations.

The relative error in the stiffness matrix approximation was approximately 1e-15, which is close to machine precision. The relative error in the mass matrix is 1e-12. Recall that the mass matrix is trained with results from the stiffness surrogate model and thus incurs a slight accumulation of errors, so a slightly higher error is expected. Even so, the matrix is still extremely accurate. From these results, we can see that a machine-learned model as outlined above is able to predict mass and stiffness properties to a high degree of accuracy. From these results, we can see that the machine learning procedure outlined above is indeed capable of learning the mass and stiffness properties of a system.

Secondly, we examine a collection of eight linear hexahedral elements arranged into a cube. The same training procedure as the single cube example is also used here. However, when generating the training set for the machine-learned models, the center node is intentionally left out. The result of this change is that in contrast to the previous example, in this case the training data is not complete, and the model must learn a reduced representation of the full system matrices. A similar training procedure as the single element example above was used. However, a linear sweep of force from −1e9 to 1e9 was used instead of a sine wave, and prescribed displacements were used at the free degree-of-freedom instead of prescribed force, with a range of −1 to 1.

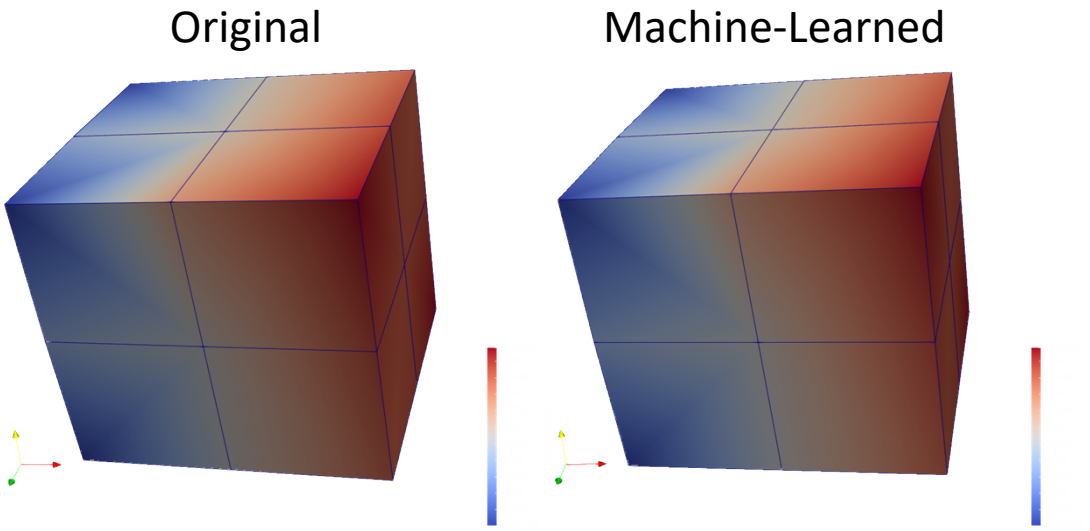


Eight hex elements



Boundary conditions

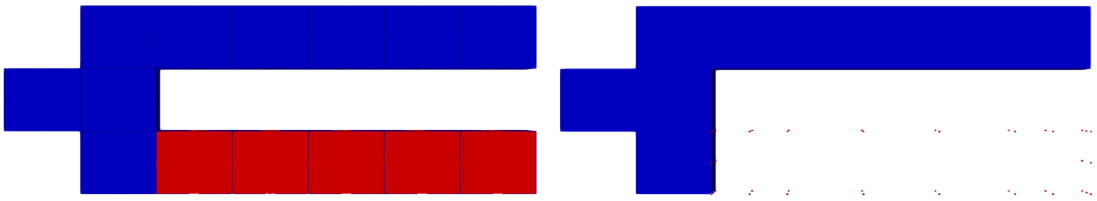
To exercise the machine-learned model, we run a transient analysis of both the original model and the machine-learned model that was trained with the missing data. The vertex nodes of the back face of the cube were fixed, and a bi-directional shearing load of 1e7 was applied to the vertex nodes of the front face (see *boundary conditions* figure above). A short transient analysis was conducted, with a simulation duration of 0.4 s and a timestep of 0.1s. The relative error over all time steps was less than 0.1%. A side-by-side comparison of the displacements for both the original and machine-learned surrogate models are plotted at the final time step below, where good agreement can be seen between the two approaches.



From this example, we can see that the machine-learned mass and stiffness approach is able to approximate the properties of multiple elements simultaneously, but also that the data set need not be complete to obtain an accurate approximation.

Finally, we consider a tuning fork model (shown below). This model is both geometrically more complex than the previous examples, and also utilizes a 20-noded quadratic hexahedral element formulation. The lower tine (shaded in red) is replaced with a machine-learned surrogate model at a subset of the nodes. As opposed to the previous two examples, however, the subregion to be replaced was simulated independently to obtain the training data. Another added complexity over the previous examples is in the amount of data: while the previous example was missing data from 1 node out of 27, we will be training this problem using only 40 out of a total of 68 nodes in the surrogate region. A similar training procedure was used to the previous example. However, *two* degrees-of-freedom are now free at a time, and less data (20 steps) was provided for each loading case. Additionally, 500 gaussian-random displacement & force distributions (for static & transient simulations respectively) were included in the training set at all degrees-of-

freedom simultaneously, with amplitudes matching the linear sweep (1 for displacement and 1e9 for force).



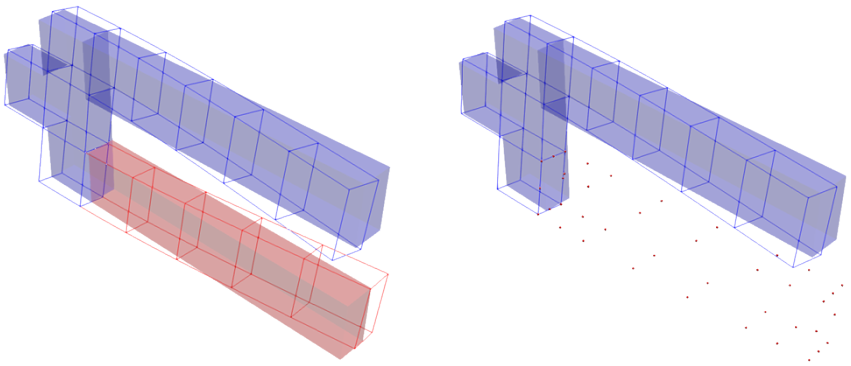
Original model

Residual model + ML surrogate

After the machine-learned models were trained, they were coupled back to the residual structure as seen in the image above. Subsequently, an eigen-analysis simulation was conducted for both the original and coupled surrogate model. ten modes were simulated: six rigid body (zero-energy) modes, and four elastic. As seen in the following table, the percent error in the 1st four elastic modes is all below 5%. Additionally, a comparison of the mode shapes shows good agreement, even in the surrogate region.

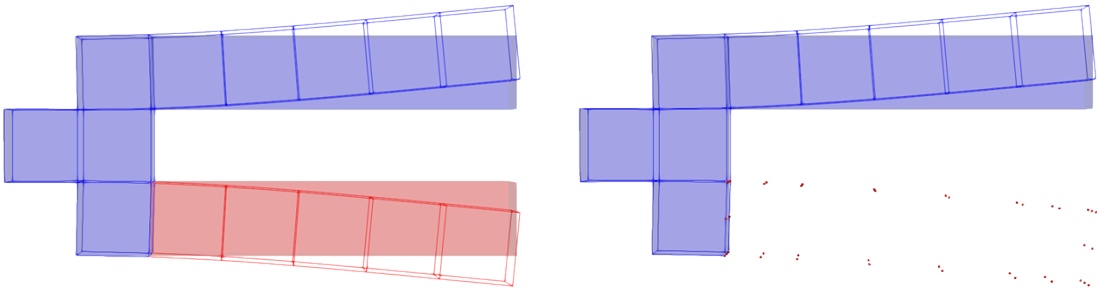
	1st elastic	2nd elastic	3rd elastic	4th elastic
% error	1.7	0.7	3.2	4.5

1st elastic:

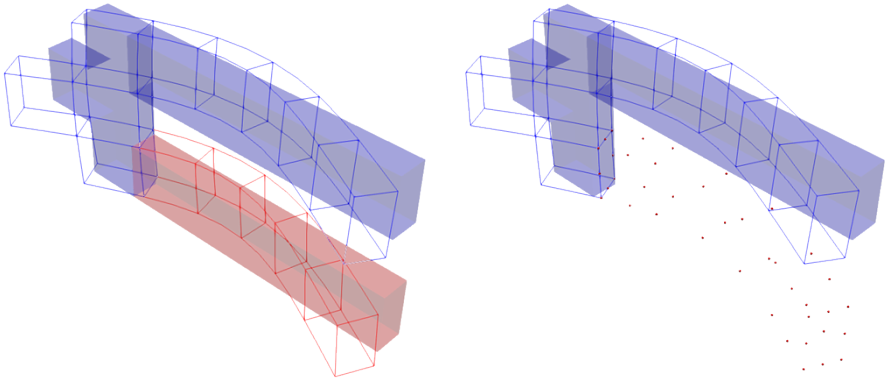


LABORATORY DIRECTED
RESEARCH & DEVELOPMENT
WHERE INNOVATION BEGINS

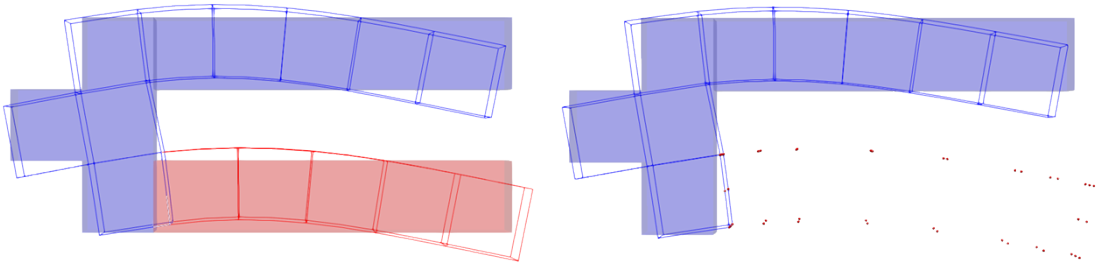
2nd elastic:



3rd elastic:



4th elastic:



This example demonstrates the applicability of the machine-learned approach to more complex models and element formulations. Additionally, the example demonstrates that a moderate amount of data can be missing from your training data set while still obtaining reasonable accurate results. Finally, this example demonstrates that this approach is well-suited for the typical reduced-order modeling workflow of off-line learning followed by embedded simulation.

ANTICIPATED OUTCOMES AND IMPACTS:

Based on our discussions with interested analysts, we anticipate this work having an immediate impact on both system and component modeling across many current and future weapons programs. This approach offers structural dynamics analysts a user-friendly alternative to the current state of the art found in existing reduced-order modeling approaches like Craig-Bampton reduction and opens the door to many exciting possibilities in the future.

One way that this machine-learned approach offers an immediate impact to Sandia analyses is through the incorporation of experimental results into the reduced-order model formulation. For example, there will always be model-form error in any numerical simulation, where the underlying assumptions of the chosen model do not match the physical reality. In such situations, we envision this machine-learned modeling approach being useful to account for such discrepancies. This accounting could take two forms. First, experimental data could be used to augment the simulation training data, thereby giving the machine-learned model a "richer" data set from which to learn. Alternatively (or additionally), experimental data could also be used to influence the machine-learning approach directly, similarly to the loss-penalty approach we utilized to embed key physical properties of system matrices into our machine-learned models.

This research was conducted using agile best practices, and all code development utilized test-driven development to ensure a clean and easily maintainable software implementation. Robust testing is vital to the longevity of a software product and was a key focus of our work here as well. We have well over 150 version-controlled tests, as well as two dozen unit tests. Nearly all the tests are run nightly on a wide variety of computing platforms and compilers, are easily accessible to all users of the Sierra software suite, and are illustrative of the basic workflow and use cases of this capability. In fact, all three examples presented in this document are part of our nightly testing suite.

We have also developed a robust suite of workflow tools that support every step from initial training setup to model generation and subsequent (re)use. This development represents a significant shift away from user-maintained one-off workflows and scripts, which are prone to being brittle, to a generable, maintained, and well-tested approach. Finally, this work opens the door to many exciting possibilities and serves as the foundation for several follow-on areas of research that we hope to explore.

One natural extension of this work would be in the modeling of system damping properties, which frequently require extensive tuning of the finite element model to obtain accurate predictive simulations. The equations of motion for a damped system are shown below:

$$Ma + Cv + Kd = f$$

where $v = \dot{d}$ represents the system velocity and C is the damping matrix, which often requires calibration against experimental data. In contrast, one could envision a three-step approach, starting with the existing simulation-driven two-step mass-and-stiffness learning procedure outlined here and then feeding forward into a third training leveraging experimental results to learn the damping properties.

Another area of interest is the development of a general *nonlinear* reduced-order modeling approach, where subsystems with mild nonlinearities could be incorporated into a linear dynamics application space. Indeed, previous work from our team in the area of reduced-order modeling of nonlinear structural mechanics systems has shown great promise. We would like to explore marrying the two application spaces, and we envision a use case where machine-learned (linearized) system properties would be coupled with a nonlinear machine-learned forcing function. Recalling again to the equation of motion, the system of equations we would be solving is the following:

$$M_{ML}a + K_{ML}d = f_{ML}(d, a) + f_{FEM}$$

where the machine-learned force correction vector f_{ML} that would correct for any nonlinearity that the mass and stiffness matrix models would not be able to capture. Such an approach would represent a revolutionary improvement in usability and generality vs. the domain-specific nonlinear ROMs commonly used today.

CONCLUSION:

In this work, we have demonstrated the viability of a machine-learned approach for approximating structural dynamic system matrices. The results presented here demonstrate that this approach has the ability to accurately reproduce mass and stiffness properties of a variety of element formulations, even in the presence of missing data. An interface is also provided to embed machine-learned mass and stiffness models in the widely used structural dynamics code, Sierra/SD.

REFERENCES:

[1] Craig, R. R. (1981). *Structural Dynamics: An Introduction to Computer Methods.* John Wiley & Sons.
[2] Kubomura, K. (1982). A Theory of Substructure Modal Synthesis. *ASME. J. Appl. Mech.,* 49(4): 903–909.
[3] Segalman, D. J. (2005). A four-parameter Iwan model for lap-type joints. *ASME. J. Appl. Mech.*, 72(5): 752–760.
[4] Mathis, A. T., Balaji, N. N., Kuether, R. J., Brink, A. R., Brake, M. R., & Quinn, D. D. (2020). A review of damping models for structures with mechanical joints. *Applied Mechanics Reviews*, 72(4).
[5] Mersch, J, Parish, E, Lindsay, P, & Shelton, T. (2022, October 16-19). *Machine-Learned Surrogate Models for Threaded Fastener Geometries Subjected to Multiaxial Loadings* [Conference presentation]. SES Annual Technical Meeting, College Station, TX, United States.
[6] de Oca Zapiain, DM, Bergel, GL, Lim, H, & Romero, V. (2022, Oct. 2-7). *Estimating Uncertainty of Neural Network Predictions for Inelastic Mechanical Deformation using Coupled FEM-NN Approach* [Conference presentation]. MMM10 Conference, Baltimore, MD, United States.
[7] Crane, N, Day, D, Dohrmann, C, Stevens, B, Lindsay, P, Plews, J, Vo, J, Bunting, G, Walsh, T, & Joshi, S. *Sierra/SD - User's Manual - 5.10*. United States.
[8] Tensorflow. https://www.tensorflow.org/
[9] Keras. https://keras.io/
[10] PyTorch. https://pytorch.org/
[11] G. Valiente, pocket-tensor. *https://github.com/GValiente/pocket-tensor*, 2018.

LABORATORY DIRECTED RESEARCH & DEVELOPMENT

WHERE INNOVATION BEGINS

ADDENDUM:

Machine-Learned Linear Structural Dynamics, LDRD 23-1116

Payton Lindsay (1542, PI), Timothy Shelton (1542), Kendall Pierson (1542, PM)
Consulting: Robert Kuether (1556), David Najera-Flores (1556), Justin Wilbanks (1553), Eric Parish (8739)

Purpose, Approach, and Goal

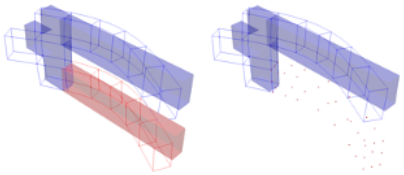
Motivation: nonlinear reduced-order modeling (e.g. joint modeling) in linear structural dynamics is challenging

Hypothesis: ML-ROMs can provide the generality of linear ROMs, but be readily extensible to capture nonlinear behavior

R&D approach: provide a production-ready linear implementation that can easily be extended to model nonlinear response

One key goal: demonstrate viability of ML-ROMs in linear structural dynamics

Representative Figure



Mode shape comparison: full model vs. ML surrogate

Key R&D Results and Significance

Summary of R&D

- Modeling dynamic response (mass properties)
  - 2-step training procedure

$$Ma + Kd = f \Longrightarrow K_{ML}d \approx f \Longrightarrow M_{ML}a \approx f - K_{ML}d$$

- Physics-constrained mass/stiffness models
  - K_{ML}: symmetric, positive diagonal, 6 rigid-body modes
  - M_{ML}: symmetric, positive diagonal, positive total mass
- Production-ready implementation (accepts TensorFlow/PyTorch)
  - Maintainable interface; extensive nightly testing
  - Extensible suite of workflow tools

The linear implementation (**one key goal**) was successful, and no issues were observed that would preclude a nonlinear extension

Modeling localized nonlinearities with ML-ROMs would be a **revolutionary improvement** vs. the current domain-specific ROMs

- We would like to explore this in the coming FY, through LDRD funding or direct application-development support

Lessons learned: consider using SGD optimization and PyTorch for constrained learning problems

LABORATORY DIRECTED RESEARCH & DEVELOPMENT

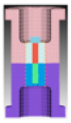
WHERE INNOVATION BEGINS

Summary

For 30 years, nonlinear submodeling in linear structural dynamics has relied on difficult, error-prone, and physically spurious modeling strategies.

- Key Question: Can we leverage machine learning to address current limitations?
  - Current modeling approaches are **narrow in scope** and **difficult to use**
- Mission alignment: Joint modeling is a well-known pain point for linear structural dynamics
  - Many Sandia applications could see **immediate impact** by this work, including system and component modeling
- Success metric: replicate current state of the art linear reduced-order modeling with a machine-learned representation
  - If successful, this work will result in the creation of a **general and extensible** approach with **broad application** to many use cases
- LDRD applicability: this work is foundational for several follow-on areas of research
  - Modeling localized nonlinearities (common in joint modeling, a high-impact capability) is a natural extension of this work
  - Shared advancements can **advance the state of the art** in fully nonlinear applications as well

My Plan



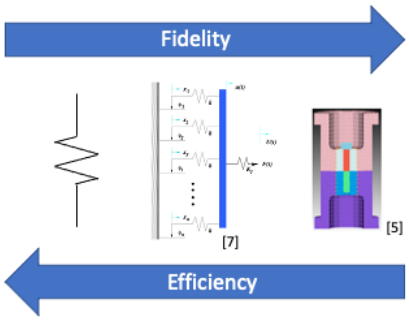
[5]

My budget

Motivation: Joint Modeling

Modeling joints in large finite element models is challenging:

- Nonlinear analysis is typically needed to accurately capture joint behavior, but can be **prohibitively expensive**
- Linear analysis is much more efficient, but cannot accurately capture nonlinearities



- Joints typically represent a (physically) **small subregion** of the overall model
  - Solution: embed a small nonlinear submodel into a larger (linear) model

Reduced-order models offer a **compromise between fidelity and efficiency**, and are well-suited to embed into a larger model

LABORATORY DIRECTED RESEARCH & DEVELOPMENT

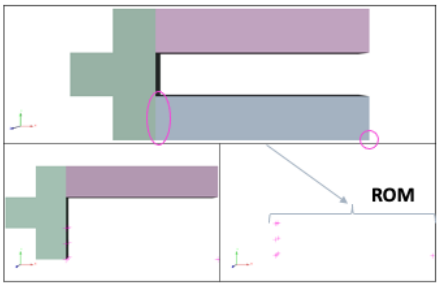
WHERE INNOVATION BEGINS

Existing Reduced-Order Modeling Approaches

The current state of the art for reduced-order models (ROMs) in linear structural dynamics falls into 2 main categories:

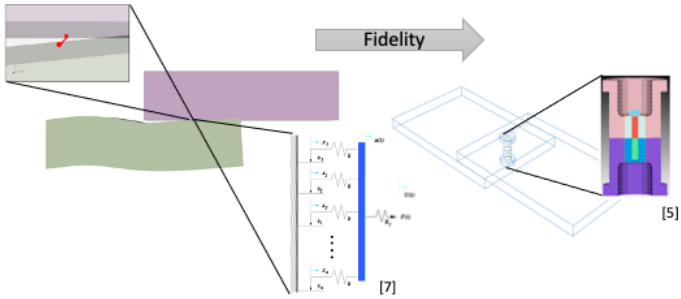
General linear ROMs

- Craig-Bampton reduction[1], component-mode synthesis[2], etc.
- Can represent a **general domain**
- Only represents **linear** behavior



Domain-specific ROMs[3,4]

- Can capture **nonlinear** behavior
- Focused on a **targeted/narrow domain**
- Requires extensive "tuning"
- Not applicable outside the intended domain



Model Selection

- **Physics-constrained learning**

$$\mathbf{M}a + \mathbf{K}\mathrm{d} = f$$

Constraints on M:
- Symmetric
- Positive diagonal
- Positive total mass

Constraints on K:
- Symmetric
- Positive diagonal
- 6 rigid-body modes

- **Dense neural network** architecture is well-suited to predicting linear system matrices
  - Only a single layer is needed
- **Stochastic gradient descent** (SGD) optimizer is a good choice of optimizer
  - Significantly outperformed other popular choices such as Adam in our testing
- **Adaptive learning rate** significantly improved the speed and robustness of the fit
- **PyTorch framework** enables easy customization and straightforward application of constraints
- **2-step learning procedure** for learning mass and stiffness (more later)

LABORATORY DIRECTED RESEARCH & DEVELOPMENT

WHERE INNOVATION BEGINS

User Experience

- Application-embedded interface to run trained models from the community-standard libraries **TensorFlow and PyTorch**
  - Handoff highly-vetted ROMs to (non-ML-expert) application end-users
- Clean and **easily maintainable software** implementation
  - Followed software best practices such as test-driven development
- **Extensive nightly testing** suite
  - Over 150 version-controlled tests; 2 dozen unit tests
  - Wide variety of computing platforms & compilers
  - Easily accessible to all users of the Sierra software suite
- Robust, well-tested suite of workflow tools
  - Initial training simulation setup
  - Model generation & sub-model embedding
  - Easily modified / extended
  - Significant shift from user-maintained one-off workflows and scripts

Learning Single Element Properties

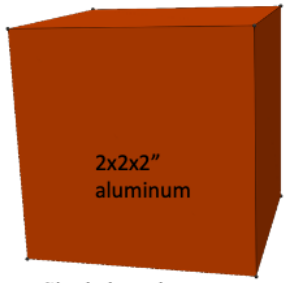
Training procedure:
- Force sweep one node+direction (dof)
- Fix all other dofs
- Repeat for one dof at a time

2-step training procedure

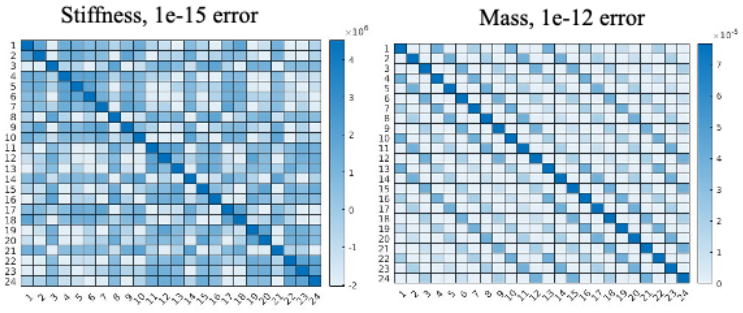
$Ma + Kd = f$

Static $K_{FEM}d = f$ ⇒ learn K_{ML} s.t. $K_{ML}d \approx f$

Dynamic $M_{FEM}a + K_{FEM}d = f$ ⇒ $M_{ML}a \approx f - K_{ML}d$



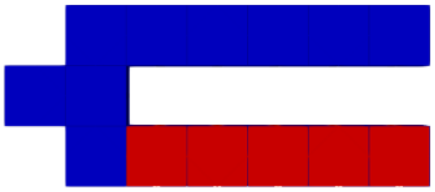
Single hex element



LABORATORY DIRECTED
RESEARCH & DEVELOPMENT
WHERE INNOVATION BEGINS

Embedded ML Submodel

20-noded quadratic hexahedrons



Original model

Original + ML submodel

40/68 nodes

Training procedure:

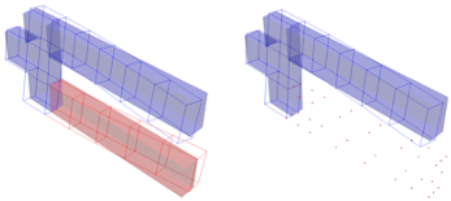
- Static: displacement sweep **two** dofs at a time
- Dynamic: force sweep **two** dofs at a time
- Fix all other dofs
- 500 additional gaussian-random samples

Elastic Mode	% Error
1	1.7
2	0.7
3	3.2
4	4.5

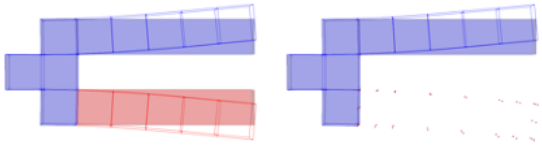
< 5% error

Elastic Mode Comparison

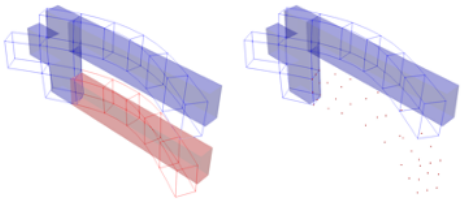
1st elastic (1.7%)



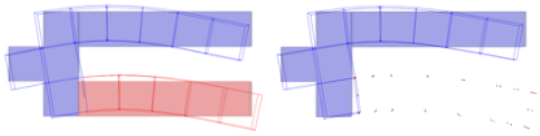
2nd elastic (0.7%)



3rd elastic (3.2%)



4th elastic (4.5%)



LABORATORY DIRECTED RESEARCH & DEVELOPMENT

WHERE INNOVATION BEGINS

Mission Impact

- This work promises a **significant improvement** over existing approaches for nonlinear submodeling in linear structural dynamics
  - We anticipate current (linear) machine-learned ROMs having an **immediate impact** on both system and component modeling at Sandia
  - The planned extension to support localized nonlinearities would offer a revolutionary improvement in usability and generality vs. the domain-specific nonlinear ROMs commonly used today

Comments from Analysts:

"Modeling joints in complex structural models is a well-known challenge that requires analysts to make various model simplifications. A ML-ROM representation of the joint, especially with the potential ability to handle nonlinear physics, presents an interesting methodology that would enable Sandians to quickly deliver accurate results to our customers."

"Providing the capability of a ML-ROM representation of a joint would greatly benefit the ability of structural dynamics (SD) analysts to quickly analyze complex systems and lay the foundations needed to help bridge the gap between our linear SD models and fully nonlinear representations."

"Having a joint model that is easy to integrate into workflows is important for broad adoptability and modeling agility. A majority of the upfront cost in model development in SD is due to identifying an appropriate joint representation along with the associated setup times for existing modeling approaches for full system models."

Capability Summary

Where can this be used?

Symbol	Meaning
☑	Implemented & tested
?	Testing needed
✗	Development needed

- Reduced-Order Modeling
  - Learn linear surrogate models ☑
  - Reduce/obfuscate geometric complexity ☑
  - Embedded simulation ☑
- Data Fusion
  - Incorporate experimental data into training/learning to address model-form error ?
  - Experimentally-driven damping properties ✗
- Capturing Nonlinear Response
  - Learn a linearized surrogate of a nonlinear model ?
  - Model mild nonlinearities natively in a linear application code ✗

$$M_{ML}a + K_{ML}d = f_{ML}(d, a) + f_{FEM}$$

[8]

References

[1] Craig, R. R. (1981). *Structural Dynamics: An Introduction to Computer Methods.* John Wiley & Sons.
[2] Kubomura, K. (1982). A Theory of Substructure Modal Synthesis. *ASME. J. Appl. Mech.,* 49(4): 903–909.
[3] Segalman, D. J. (2005). A four-parameter Iwan model for lap-type joints. *ASME. J. Appl. Mech.*, 72(5): 752–760.
[4] Mathis, A. T., Balaji, N. N., Kuether, R. J., Brink, A. R., Brake, M. R., & Quinn, D. D. (2020). A review of damping models for structures with mechanical joints. *Applied Mechanics Reviews*, 72(4).
[5] Mersch, J, Parish, E, Lindsay, P, & Shelton, T. (2022, October 16-19). *Machine-Learned Surrogate Models for Threaded Fastener Geometries Subjected to Multiaxial Loadings* [Conference presentation]. SES Annual Technical Meeting, College Station, TX, United States.
[6] de Oca Zapiain, DM, Bergel, GL, Lim, H, & Romero, V. (2022, Oct. 2-7). *Estimating Uncertainty of Neural Network Predictions for Inelastic Mechanical Deformation using Coupled FEM-NN Approach* [Conference presentation]. MMM10 Conference, Baltimore, MD, United States.
[7] Crane, N, Day, D, Dohrmann, C, Stevens, B, Lindsay, P, Plews, J, Vo, J, Bunting, G, Walsh, T, & Joshi, S. *Sierra/SD - User's Manual - 5.10*. United States.
[8] https://theworld.org/stories/2018-01-18/kirk-enterprise-piece-star-trek-your-pocket