Damage Detection and Quantification Through Differentiable Numerical Solvers

JAMES XU and MATTHEW DEJONG

ABSTRACT

Structural health monitoring strategies typically attempt to identify and quantify damage by utilizing sensor data. This is typically referred to as the inverse problem. While sensor data of the response of dynamical systems is usually abundant, data of damage location and quantity are typically difficult to acquire, even in lab settings. This lack of labels hinders traditional supervised learning, as the inverse problem is primarily concerned with changes in certain parameters of the dynamical system (e.g., stiffness degradation). This issue necessitates some form of parametric estimation, where the underlying structure of the physics model, parameterized by "damage-sensitive" parameters, is already known. This inverse problem setup naturally lends itself to Partial Differential Equation (PDE)-constrained optimization, where the objective function is the misfit between sensor data and the predicted sensor data of the physics model, and is constrained by a PDE residual equality constraint.

Several methods exist to solve this constrained optimization problem, some using non-gradient-based optimization, e.g., Bayesian optimization. In this work, we use a gradient-based optimization method, namely automatic differentiation, to determine the gradients necessary for optimization. This is possible due to the ability to represent numerical solvers as computational graphs, which can be differentiated. This method, a variant of the primal method, offers the additional benefit of incorporating physics as a hard constraint, unlike other methods, such as the penalty method. Case studies on synthetic problems related to wind turbines are presented and evaluated, and practical workarounds for memory constraints are discussed.

James Xu, Ph.D. Candidate, Email: james.xu@berkeley.edu. Structural Engineering, Mechanics and Materials; Department of Civil and Environmental Engineering, University of California, Berkeley; Berkeley, CA, USA

Matthew DeJong, Associate Professor, Email: dejong@berkeley.edu. Structural Engineering, Mechanics and Materials; Department of Civil and Environmental Engineering, University of California, Berkeley; Berkeley, CA, USA

INTRODUCTION

As wind energy becomes increasingly popular as a renewable source of energy, its respective maintenance also grows. Typically, operations and maintenance teams on wind farms manually inspect the wind turbines, which is a costly, time-consuming, and hazardous process, especially in settings such as offshore wind turbines. It is, therefore, desirable to understand the health of wind turbines through remote sensors. This is commonly referred to as *structural health monitoring*, where one of the primary goals is to detect and quantify damage in structures using sensor data. In a sense, this creates a digital twin of the structural system. Typically, degradation of material strength or stiffness can be inferred as damage to the material. However, measuring degradation of stiffness directly with sensors is challenging, thus posing a challenge in inferring damage due to the lack of labels.

Additionally, due to the digital twin setting and the specific task of inferring damage in real-time from sensor data, the dataset size will be limited in a certain sense. This leads to an issue where additional structure needs to be added. In this study, we incorporate the physics of the structural system to infer damage. To do so, we can pose this as a PDE-constrained optimization problem, where we aim to minimize a data loss between the sensor data and predicted data, which is subject to the physics of the structural system.

Several optimization problems, such as parameter estimation and optimal control, in engineering typically have physical laws incorporated into their constraints. However, physical laws, modeled as PDEs, usually do not have an analytical solution. Partial differential equations (PDEs) are typically solved using a numerical method, such as the Finite Difference Method (FDM), Finite Volume Method (FVM), or Finite Element Method (FEM). In structural mechanics problems, FEM can be shown to provide, in some sense, the most optimal solution. FEM discretizes the PDEs into a system of algebraic equations, which creates a set of linear constraints. In this study, we aim to solve the inverse problem by estimating parameters that minimize the misfit between observed and predicted states, employing a discrete-then-optimize strategy. Forward computations are performed using FEM, and gradients of the loss function with respect to parameters are computed using automatic differentiation, ensuring that the solution satisfies the governing physics.

BACKGROUND AND PREVIOUS WORK

In the structural health monitoring community, researchers typically employ classical Bayesian filtering methods to learn the system's dynamics. Extended or Unscented Kalman filters are typically used to perform state and parameter estimation [1, 2]. However, Bayesian filtering is computationally expensive and does not scale well with larger degrees of freedom systems. Recently, Gaussian processes and Bayesian updating have been employed to develop a surrogate model of the structural system, utilizing SCADA data as input and the structural response as output [3, 4]. In these works, they first train the Gaussian process with data. If the Gaussian process begins to produce results that deviate from the sensor data, they infer that deviation as damage. However, this method does not provide any information on the localization of damage, only that the structural system has changed in a manner such that the output would be different. This can be

due to a number of causes that may not actually be related to the structural health of the system.

Regarding solving more general inverse problems, researchers pose the inverse problem as a PDE-constrained optimization problem and use the finite volume method as the primary way to solve their PDE [5]. For structural systems, accurate numerical solutions of the governing PDEs are necessary. When modeling structural systems, researchers typically use the finite element method, which provides, in a sense, the best approximate solution [6].

METHODS

We aim to solve the inverse problem, namely, to detect damage to structural systems through sensor data. Due to the nature of the problem setting, we are constrained to a limited dataset. Thus, to provide an additional signal, we incorporate our physical understanding into the problem. We can do so by posing the inverse problem as a PDE-constrained optimization problem. The general form of a PDE-constrained optimization problem is given as

$$\begin{aligned} \mathbf{E}^* = \quad & \text{argmin}_\theta \quad \mathcal{L}(u_{target}, u_{pred}(\theta)) \\ & subject\ to \quad \mathcal{F}(u_{pred}, \theta) = 0 \end{aligned} \tag{1}$$

Here, $\mathcal{F}(u_{pred}, \theta) = 0$ is the PDE residual. For the linear static case, we define our loss function as the mean squared error (MSE) of the squared ℓ_2 norm of the predicted $u_{predicted}$ vector the true u_{target}, as shown in Equation 2

$$\mathcal{L}(u_{pred}(\theta)) = \|(u^{(i)}_{target} - u^{(i)}_{pred}(\theta)\|^2_2 \tag{2}$$

For the linear dynamic case, we define our loss function as the squared Frobenius norm of the misfit of the predicted $u_{predicted}$ vector the true u_{target}, averaged over the trajectory, as shown in Equation 3.

$$\mathcal{L}(u_{pred}(\theta)) = \frac{1}{n_{free} N_t} \|(u^{(i)}_{target} - u^{(i)}_{pred}(\theta)\|^2_F \tag{3}$$

Here, n_{free} is the number of free degrees of freedom of the discretized system, and N_t is the number of time steps of the trajectory. We aim to solve for the underlying parameter vector, $\mathbf{E}$, from which we can infer changes as damage. In the following methods, we impose the physics of the problem as a hard constraint to incorporate the known governing PDEs. We avoid using soft constraints to solve the optimization problem due to the potential failure modes introduced by imposing soft constraints via the penalty method [7, 8], which can be expressed as

$$\mathbf{E}^* = \quad \text{argmin}_\theta \quad \mathcal{L}(u_{target}, u_{pred}(\theta)) + \lambda \mathcal{F}(u_{pred}, \theta) \tag{4}$$

where λ is the standard penalty parameter.

LINEAR ELASTOSTATICS, ELASTODYNAMICS, AND THE FINITE ELEMENT METHOD

For linear elastosstatics, the PDE residual, $\mathcal{F}(u_{pred}, \theta) = 0$, is composed of governing equations for linear elastostics which can then be numerically solved using the FEM. We aim to solve the canonical linear elastostatic problem. The strong form and boundary conditions can be written as

$$\begin{aligned} \nabla \cdot \sigma + \mathbf{f} &= 0 \quad in\, \Omega \\ \sigma \mathbf{n} &= \mathbf{t} \quad on\, \Gamma_q \\ \mathbf{u} &= \overline{\mathbf{u}} \quad on\, \Gamma_u \end{aligned} \tag{5}$$

We can then employ the finite element method. We can then use the standard Galerkin method and obtain the weak or variational formulation of the strong form using the test function $\mathbf{w}$ as

$$\int_\Omega \nabla_s \mathbf{w} : \sigma d\Omega = \int_\Omega \mathbf{w} \cdot \mathbf{f} d\Omega + \int_{\Gamma_q} \mathbf{w} \cdot \bar{\mathbf{t}} d\Gamma \tag{6}$$

where the admissible spaces $\mathcal{U}$ and $\mathcal{W}$ defined as

$$\begin{aligned} \mathcal{U} &= \{\mathbf{u} \in H^1(\Omega) \mid \mathbf{u} = \overline{\mathbf{u}}\, on\, \Gamma_u\} \\ \mathcal{W} &= \{\mathbf{w} \in H^1(\Omega) \mid \mathbf{w} = \mathbf{0}\, on\, \Gamma_u\} \end{aligned} \tag{7}$$

where $H^1(\Omega)$ is the Sobolev space of order 1. We can then recover the linear system of equations for each element using the standard finite element procedure as

$$\mathbf{K}^e \mathbf{u}^e = \mathbf{F}^e \tag{8}$$

After assembly of the element matrices into the global matrix, we then have the linear system to solve, being

$$\mathbf{K}\mathbf{u} = \mathbf{F} \tag{9}$$

Similarly to static case, the strong form of the dynamic case can be written as

$$\begin{aligned} \nabla \cdot \sigma + \mathbf{f} &= \rho \mathbf{a} \quad in\, \Omega \times I \\ \sigma \mathbf{n} &= \mathbf{t} \quad on\, \Gamma_q \times I \\ \mathbf{u} &= \overline{\mathbf{u}} \quad on\, \Gamma_u \times I \\ \mathbf{u}(\mathbf{x}, 0) &= \mathbf{u}_0(\mathbf{x}) \quad in\, \Omega \\ \mathbf{v}(\mathbf{x}, 0) &= \mathbf{v}_0(\mathbf{x}) \quad in\, \Omega \end{aligned} \tag{10}$$

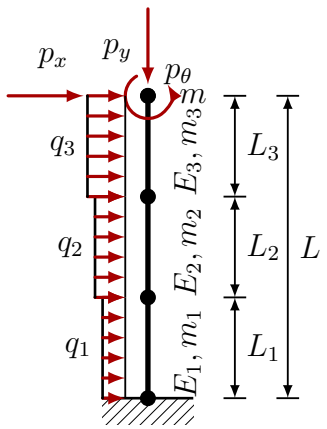

Figure 1. Idealization of the problem in a steady state condition (steady winds).

We then have the week form as

$$\int_\Omega \nabla_s \mathbf{w} \cdot \rho \mathbf{a} d\Omega + \int_\Omega \nabla_s \mathbf{w} : \sigma d\Omega = \int_\Omega \mathbf{w} \cdot \mathbf{f} d\Omega + \int_{\Gamma_q} \mathbf{w} \cdot \bar{\mathbf{t}} d\Gamma \tag{11}$$

where the admissible spaces $\mathcal{U}$ and $\mathcal{W}$ defined as

$$\begin{aligned} \mathcal{U} &= \{\mathbf{u} \in H^1(\Omega \times I) \mid \mathbf{u} = \bar{\mathbf{u}} \; on \; \Gamma_u \times I, \mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0, \dot{\mathbf{u}}(\mathbf{x}, 0) = \mathbf{v}_0\} \\ \mathcal{W} &= \{\mathbf{w} \in H^1(\Omega \times I) \mid \mathbf{w} = \mathbf{0} \; on \; \Gamma_u \times I, \mathbf{u}(\mathbf{x}, 0) = \mathbf{0}, \dot{\mathbf{u}}(\mathbf{x}, 0) = \mathbf{0}\} \end{aligned} \tag{12}$$

With the standard finite element procedure, we can then recover the global system of equations as

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{F} \tag{13}$$

To perform time-stepping, the standard Newmark-Beta method with $\beta = 0.25$ and $\gamma = 0.5$ was employed. During the process of converting the strong form into a set of linear equations, several reasonable assumptions can be made. One of these assumptions is the choice of element. Here, we use a linear elastic frame element, making the standard infinitesimal strain assumptions [9].

As shown in Figure 1, we then discretize our real-world system of a wind turbine as a three-element cantilever column. We made the discretization since turbines are typically composed of three elements that are then bolted together. We idealized the forces from the wind that impact the blades and nacelle as point loads, p_x, p_y, and p_θ, at the top node, and a uniformly distributed load, q, representing steady wind.

THE PRIMAL METHOD AND MEMORY CONSTRAINTS

Through solving the PDE using FEM, we obtain the solution $u_{pred} = \mathcal{G}(\mathbf{E})$, that satisfies $\mathcal{F}(\mathcal{G}(\mathbf{E}), \mathbf{E}) = 0$. Here, $\mathcal{G}(\mathbf{E})$ is the computational graph that was used to

solve for the PDE, which used the current estimate of $\mathbf{E}$. We have the PDE-constrained optimization problem as

$$\mathbf{E}^* = \quad \text{argmin}_{\mathbf{E}} \quad \mathcal{L}(u_{target}, \mathcal{G}(\mathbf{E})) \tag{14}$$

This now becomes an unconstrained optimization problem and off-the-shelf optimizers such as Adam and L-BFGS can be used. In this work, we created a custom differentiable finite element solver using JAX [10] to enable automatic differentiation through the numerical solver. We used vanilla gradient descent to directly update the parameter estimate $\mathbf{E}$. We also scale the gradients by the parameter value so that the gradients have the same units of the response, which can also be seen as tuning the learning parameter. This optimization process is summarized in Algorithm 1 for clarity.

Algorithm 1 Vanilla gradient descent for PDE-constrained optimization

Given: Target displacement u_{target}, initial guess $\mathbf{E}^{(0)}$
Set: Learning rate η, number of epochs N
for $k = 0, 1, \ldots, N-1$ **do**
    Solve FEM forward problem to get $u^{(k)} = \mathcal{G}(\mathbf{E}^{(k)})$ such that $\mathcal{F}(u^{(k)}, \mathbf{E}^{(k)}) = 0$
    Compute loss: $\mathcal{L}^{(k)} = \mathcal{L}(u_{\text{target}}, u^{(k)})$
    Compute gradient: $\nabla\mathcal{L}^{(k)} = \nabla_{\mathbf{E}}\mathcal{L}(u_{\text{target}}, \mathcal{G}(\mathbf{E}^{(k)}))$
    Update estimate: $\mathbf{E}^{(k+1)} = \mathbf{E}^{(k)} - \eta \cdot \nabla\mathcal{L}^{(k)}$
end for
Return: Optimized parameters $\mathbf{E}^* = \mathbf{E}^{(N)}$

The gradient computation has the same objective as classical sensitivity analysis, with some caveats. Typically, sensitivity analysis is a few-to-many problem and is thus efficient in computing with forward mode AD. Additionally, before the advent of AD, the analytical gradients were computed by hand for each step of the numerical solver. They required a great deal of engineering effort to implement successfully. In our problem, the problem setting is typically few-to-many or many-to-many, as there are generally many sensor points monitoring a structure.

For nonlinear PDEs, we can formulate a set of implicit equations using the FEM; however, we would then have to iterate to determine the solution. However, we could use the implicit function theorem to obtain the gradient as

$$\frac{\partial\mathcal{L}(\mathbf{E})}{\partial\mathbf{E}} = -\frac{\partial\mathcal{L}(u)}{\partial u}\left(\frac{\partial\mathcal{F}(\mathbf{E},u)}{\partial u}\bigg|_{u=\mathcal{G}(\mathbf{E})}\right)^{-1}\frac{\partial\mathcal{F}(\mathbf{E}.u)}{\partial\mathbf{E}}\bigg|_{u=\mathcal{G}(\mathbf{E})} \tag{15}$$

This would avoid the need to store the entire computational graph of the fixed-point iteration typically used to solve nonlinear systems of equations [5]. Another key note is that for trajectories in the dynamic case that run for N_t time steps, the memory requirement can become very large if N_t is large. Since memory constraints are often the bottleneck over compute time, selective gradient checkpointing can be used to reduce the memory constraint at the cost of increasing the compute required [11, 12].

Experiment	$\mathbf{E_1}$	$\mathbf{E_2}$	$\mathbf{E_3}$	Static $\mathbf{E}$ Errors	Dynamic $\mathbf{E}$ Errors
1	200e9	180e9	200e9	1.857e-14	1.415e-13
2	180e9	185e9	180e9	1.986e-14	2.186e-13
3	194e9	195e9	197e9	6.948e-15	1.789e-13

TABLE I. True $\mathbf{E}$ values and relative errors.

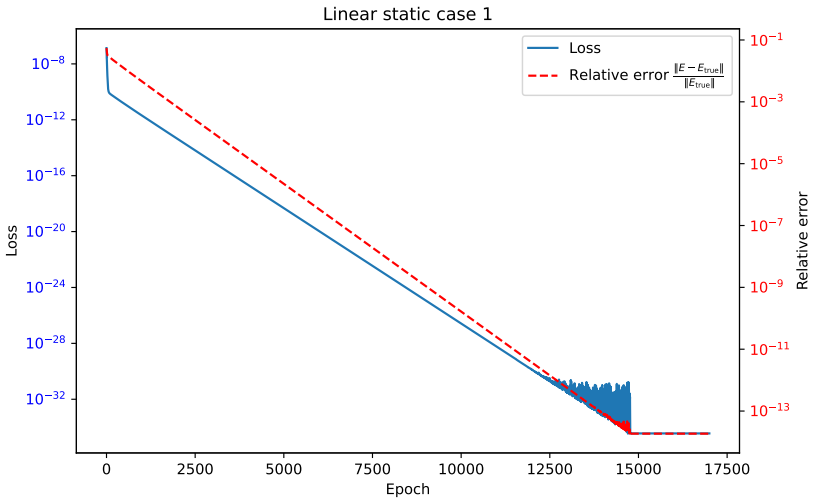

Figure 2. Loss and relative error of $\mathbf{E}$ for the linear static case 1.

EXPERIMENTS AND RESULTS

The target data was generated using OpenSees [13], using elastic frame elements. The elements were constructed using the properties of a 1.6 mW GE wind turbine. For the linear static case, loadings for p_x, p_y, p_θ, and q were subjected to the finite element model. For the linear dynamic case, a sinusoidal loading for p_x with some added noise, a sinusoidal loading for p_θ, and constant loadings for q and p_y, were subjected to the model. For all cases, $\mathbf{E}$ was initialized as $\mathbf{E} = \begin{bmatrix} E_1 = 200e9, & E_2 = 200e9, & E_3 = 200e9 \end{bmatrix}$ (in Pa) since this represents the undamaged state, which is well-known by the designers of the structural system. We note that the typical Young's Modulus of steel is $E = 200e9\ Pa$. Three cases of the true $\mathbf{E}$ values were used, which are described in Table I. These cases were run for both the linear and dynamic simulations. The static and dynamic $\mathbf{E}$ relative errors were computed using the optimized $\mathbf{E}_{converged}$ vector as Relative error $= \frac{\|\mathbf{E}_{converged} - \mathbf{E}_{true}\|_2}{\|\mathbf{E}_{true}\|_2}$.

Focusing on case 1, the true vector is given by $\mathbf{E}_{true} = \begin{bmatrix} 200e9, & 180e9, & 200e9 \end{bmatrix}$. This can be interpreted as the middle element (halfway up the tower) experienced damage and had a 10% reduction in its effective Young's Modulus.

We have the loss and relative error of $\mathbf{E}$ curves for the primal method shown in Figure 2 and 3. The loss curves for the other cases can be found in the Appendix. We note that during optimization, there was no access to $\mathbf{E}_{true}$ since it is not known in realistic scenarios and that it was only used to evaluate how close the estimate of $\mathbf{E}$ was to the true value during optimization.

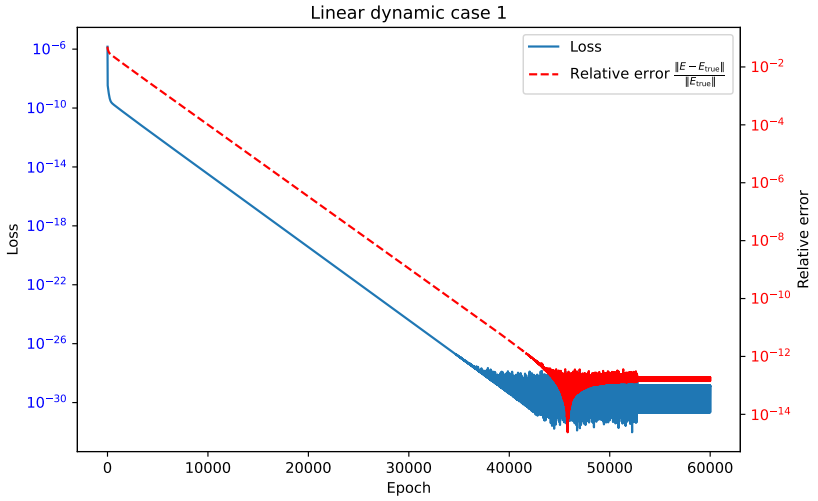

Figure 3. Loss and relative error of $\mathbf{E}$ for the linear dynamic case 1.

We note that the loss and relative error of $\mathbf{E}$ plateau for the static case around the 15000th epoch, as shown in Figure 2, and around the 45000th epoch for the dynamic case as shown in Figure 3. The number of epochs can be dramatically reduced by using more sophisticated optimization algorithms. The optimization curves plateau due to floating-point precision limitations, as the loss values become very small.

For all cases described in Table I, the losses were able to reach floating point error levels, and the relative errors of $\mathbf{E}_{converged}$ were of magnitudes around 10^{-13}.

CONCLUSIONS

In this work, we posed the inverse problem of damage detection and quantification as a PDE-constrained optimization problem. To solve the PDE-constrained optimization problem, the finite element method was used to solve the constraint. To create a differentiable simulator, automatic differentiation was utilized to compute gradients to optimize the estimation of the Young's modulus of the elements in the finite element model using vanilla gradient descent.

We presented a representative case of a wind turbine that underwent various damage scenarios and found that vanilla gradient descent consistently converged to the correct parameters. We observed that the loss evaluations converged monotonically until reaching a plateau, which was due to the limited floating-point precision. We acknowledge that a change in the material properties of an element would no longer result in a linear elastic static or dynamic simulation. However, this study sets the foundation for future work to include more sophisticated elements and material nonlinearities. Using gradient-based optimization enables the estimation of higher-degree-of-freedom systems that scale favorably, allowing for greater model complexity of the structure to match realistic structural systems. By leveraging advances in the optimization community, methods such as gradient checkpointing and the use of the implicit function theorem are presented to mitigate memory constraints.

REFERENCES

1. Song, M., B. Moaveni, H. Ebrahimian, E. Hines, and A. Bajric. 2023. "Joint parameter-input estimation for digital twinning of the Block Island wind turbine using output-only measurements," *Mechanical Systems and Signal Processing*, 198:110425.
2. Olivier, A. and A. W. Smyth. 2018. "A marginalized unscented Kalman filter for efficient parameter estimation with applications to finite element models," *Computer Methods in Applied Mechanics and Engineering*, 339:615–643.
3. Mehrjoo, A., E. M. Tronci, B. Moynihan, B. Moaveni, F. Rüdinger, R. McAdam, and E. Hines. 2025. "Recursive Bayesian estimation of wind load on a monopile-supported offshore wind turbine using output-only measurements," *Mechanical Systems and Signal Processing*, 224:112183.
4. Moynihan, B., E. M. Tronci, M. C. Hughes, B. Moaveni, and E. Hines. 2024. "Virtual sensing via Gaussian Process for bending moment response prediction of an offshore wind turbine using SCADA data," *Renewable Energy*, 227:120466.
5. Xu, K., A. M. Tartakovsky, J. Burghardt, and E. Darve. 2020. "Inverse modeling of viscoelasticity materials using physics constrained learning," *arXiv preprint arXiv:2005.04384*.
6. Papadopoulos, P. 2022, "Introduction to the Finite Element Method," .
7. Krishnapriyan, A., A. Gholami, S. Zhe, R. Kirby, and M. W. Mahoney. 2021. "Characterizing possible failure modes in physics-informed neural networks," *Advances in neural information processing systems*, 34:26548–26560.
8. Xu, K. and E. Darve. 2022. "Physics constrained learning for data-driven inverse modeling from sparse observations," *Journal of Computational Physics*, 453:110938.
9. Anand, L. and S. Govindjee. 2020. *Continuum mechanics of solids*, Oxford University Press.
10. Bradbury, J., R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang. 2018, "JAX: composable transformations of Python+NumPy programs," .
11. Chen, T., B. Xu, C. Zhang, and C. Guestrin. 2016. "Training deep nets with sublinear memory cost," *arXiv preprint arXiv:1604.06174*.
12. Maddison, J. R. 2024. "Step-based checkpointing with high-level algorithmic differentiation," *Journal of Computational Science*, 82:102405.
13. McKenna, F. 2011. "OpenSees: a framework for earthquake engineering simulation," *Computing in Science & Engineering*, 13(4):58–66.

APPENDIX

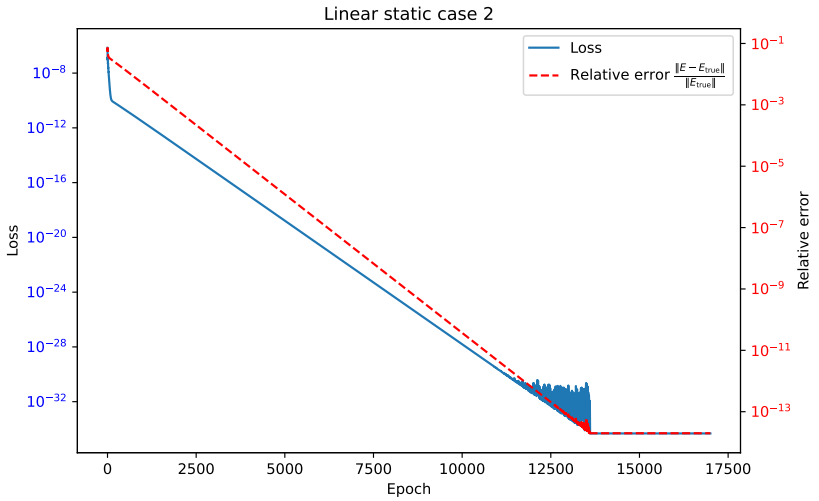

Figure 4. Loss and relative error of $\mathbf{E}$ for the linear static case 2.

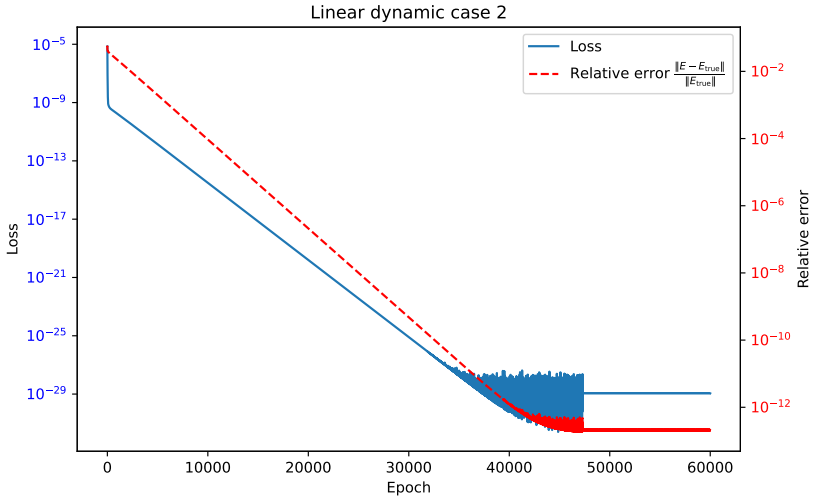

Figure 5. Loss and relative error of $\mathbf{E}$ for the linear dynamic case 2.

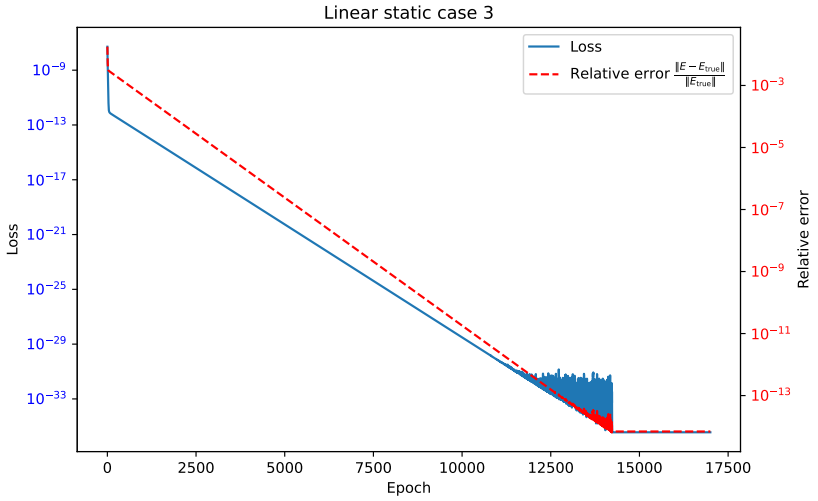

Figure 6. Loss and relative error of $\mathbf{E}$ for the linear static case 3.

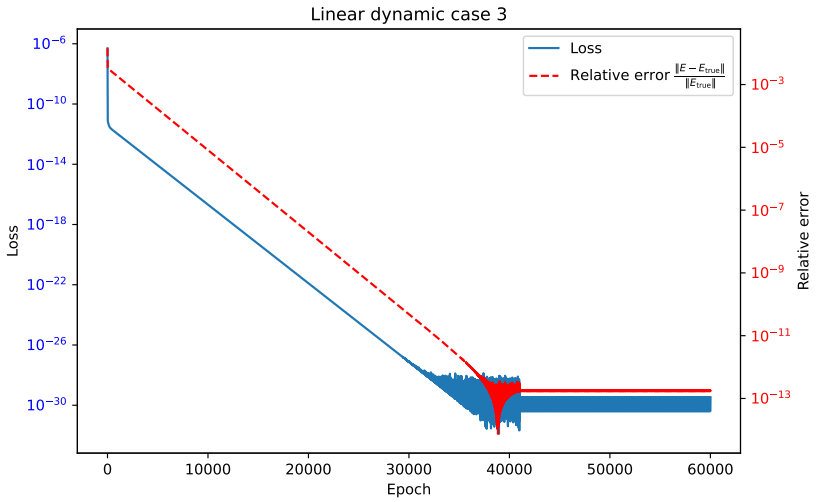

Figure 7. Loss and relative error of $\mathbf{E}$ for the linear dynamic case 3.