Adjoint-based Calibration of Plasticity Model Parameters from Digital Image Correlation Data

D. Thomas Seidl and Brian N. Granzow

September 20, 2018

1 Abstract

Parameter estimation for mechanical models of plastic deformation utilized in nuclear weapons systems is a laborious process for both experimentalists and constitutive modelers and is critical to producing meaningful numerical predictions. In this work we derive an adjoint-based optimization approach for a stabilized, large-deformation J_2 plasticity model that is considerably more computationally efficient but no less accurate than current state of the art methods. Unlike most approaches to model calibration, we drive the inversion procedure with full-field deformation data that can be experimentally measured through established digital image or volume correlation techniques. We present numerical results for two and three dimensional model problems and comment on various directions of future research.

2 Introduction

This project focused on the calibration of parameters in a constitutive equation using full-field deformation data as provided by digital image correlation. The motivation for this work comes from numerical simulation of nuclear weapon systems. Accurate prediction of finite-strain, plastic mechanical behavior at the component and system level is important to assess performance in normal and abnormal environments. The mechanical properties of components are inputs to such models and must be appropriately characterized for the models to have credibility.

Traditional approaches to calibration involve mechanical tests that are designed to elicit specific, well-understood stress-strain states in the material of interest. The canonical example is the uniaxial tension test. In such a test, a specimen (typically a "dogbone" shape) is pulled apart at a specified rate. Two important quantities recorded are the extension of the sample from its undeformed length ΔL and the load applied to the sample F. From this information, along with geometrical information about the sample, the uniaxial stress-strain curve for the material can be obtained. Using this curve, it is possible to estimate Young's modulus E, the yield stress Y, and hardening moduli for certain hardening models.

The main advantage of the uniaxial test approach is its simplicity, where it is easy to interpret the data. However, its disadvantages also stem from this simplicity. For more complicated material models, such as anisotropic or viscoplastic materials, a single test is not enough to fully estimate the model parameters. Instead, multiple samples made from different orientations of material, pulled at different rates, etc., must be tested. Combinations of this data are then utilized to inform constitutive parameter identification. A key point is that, during such tests, only a small number of recorded "global" quantities (e.g. $\Delta L, F$) serve as the basis for characterization of the material's behavior.

Digital image correlation (DIC) is a metrology technique that provides full-field measurements of displacement with an accuracy of the order of microns for typically-sized lab specimens. The term "full-field" refers to the density of the measured field, which may result in the acquisition of displacement data at millions of points on the sample.

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 Energys National Nuclear Security Administration under contract DE-NA-0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

DIC is employed in uniaxial tests, but its full-field data is averaged to compute a single quantity like ΔL or axial strain.

We realized fairly early in the project that given our budget and time constraints the application of our computational techniques to experimental DIC data would not be feasible. Instead, we focused on applying our algorithms to synthetically-generated DIC data and designed them in such a way that experimental data could be accommodated at a later point.

The use of an information-rich measurement as provided by DIC to drive model calibration problems has been extensively explored by the experimental mechanics community, but is still very much in the research phase. The two leading techniques from this community are the virtual fields method (VFM) and finite element updating (FEMU). Speaking generally, FEMU is regarded as more robust, but more computationally demanding than the VFM.

In FEMU, a constrained optimization problem is formulated where the goal is to minimize the weighted difference between the DIC and FE model-predicted displacement data. This is most often achieved by coupling an off-the-shelf commercial FE solver to an optimization software package or using the solver to compute approximate sensitivity matrices for a Gauss-Newton algorithm. The computational expense of FEMU increases with the complexity of the material model of interest and the dimension of the parameter space. This disadvantage has limited its adoption.

Automatic differentiation (AD) is a useful technique to computationally evaluate analytic derivatives (to machine precision) of a given function. In the context of partial differential equation (PDE)-constrained optimization, AD provides the ability to compute various required sensitivities based on a single PDE residual implementation and a single objective function implementation, without requiring manual derivative calculations and implementations. These sensitivities include the derivative of the PDE residual with respect to the state variables, the derivative of the PDE residual with respect to the design parameters, and the derivative of the objective function with respect to the state variables.

The PDE-constrained optimization community has long considered various forms of full-field material model calibration problems, but have paid little attention to plastically-deforming materials. Researchers in the field of topology optimization have worked with a great variety of plasticity models, such as damage, viscoplastic, kinematic hardening, and ansiotropic variants. In contrast to model calibration, the optimization space for topology optimization is a heterogenous density field rather than a vector of model parameters. Additionally, practitioners of topology optimization who use plasticity models have not yet leveraged the capabilities of AD.

The use of AD in a plasticity code has the potential to drastically reduce the barrier of entry for material model calibration by eliminating tedious and error-prone sensitivity derivations, as well as dramatically reducing (compared to FEMU) time-to-solution for model calibration by enabling the use automatically computed adjoint sensitivities. Additionally, adjoint-based objective gradients obtained via AD are exact (to machine precision) as compared to the inexact gradients obtained with FEMU. Given these realizations, we pose the question: does the adjoint-based approach to PDE-constrained optimization using AD offer computational savings and yield more accurate solutions in the context of model calibration for plasticity models?

We have found that yes, using our AD adjoint-based approach, we obtain a speed-up of close to 5x the FEMU approach (which is considered state of the art), but the accuracy of the methods was comparable. We believe that by modifying our current formulation we could compute numerically-exact sensitivities, but it is unclear how much of a bearing this would have on the accuracy of the parameter estimation procedure.

The primary contributions of this work are an AD-based plasticity finite element code capable of computing forward and adjoint-based sensitivities for optimization from full-field data. In the following sections we describe the mechanical model of interest, the sensitivity analysis approaches to the gradient, present some results, and comment on them. We end this report with possible extensions and follow-on plans and conclusions.

3 Detailed Description of Method

3.1 Mechanical Model

We use a finite deformation plasticity model based on the J_2 yield criterion. This model is commonly employed for materials with large deformations and is a starting point for more advanced models.

We solve the momentum equation and neglect body forces in a total Lagrangian formulation:

$$\begin{cases} -\nabla \cdot \boldsymbol{P} = \boldsymbol{0}, & \text{in } \mathcal{B}, \\ \boldsymbol{u} = \boldsymbol{g}, & \text{on } \Gamma_g, \\ \boldsymbol{P} \cdot \boldsymbol{n} = \boldsymbol{h}, & \text{on } \Gamma_h, \end{cases} \tag{1}$$

where $\boldsymbol{u}$ is displacement, $\boldsymbol{n}$ is the outward unit normal, $\mathcal{B}$ denotes the configuration space for the body, Γ_g the portion of the boundary over which Dirichlet boundary conditions (BCs) are applied, Γ_h the portion for traction BCs, and $\boldsymbol{P}$ is the first Piola-Kirchhoff stress tensor

Details for the computational model can be found in Simo and Hughes text *Computational Plasticity*. We use a finite deformation model that relies on a multiplicative decomposition of the deformation gradient $\boldsymbol{F}$ in plastic and elastic components such that $\boldsymbol{F} = \boldsymbol{F}^p\boldsymbol{F}^e$. This formulation implies the existence of a stress-free intermediate configuration. Our implementation uses two internal variables stored at Gauss points to keep track of the history of plastic deformation. These are the equivalent plastic strain α and the inverse of the plastic right Cauchy-Green strain tensor $\boldsymbol{G}^p := \left((\boldsymbol{F}^p)^T\boldsymbol{F}^p\right)^{-1}$.

The algorithm computes an elastic trial step and checks if that brings the state of the material beyond the current radius of the yield surface. If it does, the classic radial return algorithm is used to compute the update to the Kirchhoff stress $\boldsymbol{\tau} := \det(\boldsymbol{F})\,\boldsymbol{\sigma}$, where $\boldsymbol{\sigma}$ is the Cauchy stress tensor, that lies on the yield surface and to update the internal variables appropriately. We assume a nonlinear isotropic hardening model characterized by a linear hardening modulus K, saturation modulus S, and hardening exponent D, such that the expression for the yield surface f is

$$f = \|\text{dev}\,(\boldsymbol{\tau})\,\| - \sqrt{\frac{2}{3}}\left\{Y + K\alpha + S\left[1 - \exp\left(-D\alpha\right)\right]\right\}. \tag{2}$$

The elastic part of the response is governed by a NeoHookean strain-energy density function. In totality, there are six parameters that characterize the model: Young's modulus E, Poisson's ratio ν, quasi-static yield stress Y, and hardening moduli K, S, and D which collect into a vector $\boldsymbol{\beta} \in \mathbb{R}^e$.

The isochroric nature of plastic deformation requires special treatment for a stable numerical model. We introduced a pressure degree of freedom (DOF) p with residual-based stabilization. Alternative approaches include enhanced strain methods, reduced selective integration, and hourglass control methods. We utilize a stabilized Petrov-Galerkin approach where the weak form of the discretized momentum equation is perturbed by a function $\boldsymbol{w} + \tau_e\boldsymbol{F}^{-T}\nabla q$. Here q is a weighting function for the pressure and $\tau_e := \frac{h_e^2}{2\mu}$ is a stabilization parameter that depends on a characteristic size of a mesh element $h_e := \text{meas}(\mathcal{B}_e)$. We use triangles and tetrahedral elements to discretize Ω and a backward Euler temporal discretization.

The Cauchy stress tensor is decomposed in hydrostatic and deviatoric components such that $\boldsymbol{\sigma} := \boldsymbol{\sigma}' + p\boldsymbol{I}$. The stabilized semi-discrete weak form aims to find a solution $\boldsymbol{U} := [\boldsymbol{u}, p] \in \mathcal{S} := \mathcal{V}_u \times \mathcal{V}_p$ such that $\forall \boldsymbol{W} := [\boldsymbol{w}, q] \in \mathcal{V} := V_w \times \mathcal{V}_p$ the sum of contributions from each load step n

$$\sum_{n=1}^{N} \boldsymbol{R}^n = \boldsymbol{0}, \tag{3}$$

where

$$\begin{aligned} \boldsymbol{R}^n := &\int_{\mathcal{B}} (J\boldsymbol{\sigma}'\boldsymbol{F}^{-T}) : \nabla\boldsymbol{w}\,\mathrm{d}V + \int_{\mathcal{B}} (Jp\boldsymbol{F}^{-T}) : \nabla\boldsymbol{w}\,\mathrm{dV} + \int_{\mathcal{B}} \left[\frac{p}{k} - \frac{1}{2J}(J^2-1)\right] q\,\mathrm{dV} + \\ &\sum_{e=1}^{n_{el}} \int_{\mathcal{B}_e} \tau_e (J\boldsymbol{F}^{-1}\boldsymbol{F}^{-T}) : (\nabla p \otimes \nabla q)\,\mathrm{d}V - \int_{\Gamma_h} \boldsymbol{h}\cdot\boldsymbol{w}\;\mathrm{d}A. \end{aligned} \tag{4}$$

Here, all load-step dependent variables are evaluated at load step n.

The system (3) is comprised of nonlinear algebraic equations that are temporally coupled through neighboring load steps. Newton's method is employed at each load step to solve these systems.

The MPI-parallelized software implementation of (3) is written in C++. It makes use of several packages in the Trilinos project: Sacado for AD, Tpetra for distributed linear algebra objects, MueLu for algebraic multigrid preconditioning, Belos for matrix-free iterative linear solvers, and the MiniTensor package for convenient storage and manipulation of tensor objects. We also utilize the software SCOREC for mesh data-structures, I/O, and discretization-related computations.

3.2 Optimization

The residual (3) defines an implicit relationship between the state variables $\boldsymbol{U}$ and parameters $\boldsymbol{\beta}$. We formulate the parameter identification problem as a PDE-constrained optimization problem with box constraints:

$$\begin{aligned} \underset{\boldsymbol{\beta}}{\text{minimize}} \quad & \mathcal{J}\left(\boldsymbol{U}(\boldsymbol{\beta})\right) \\ \text{subject to} \quad & \boldsymbol{\mathcal{R}}(\boldsymbol{U},\boldsymbol{\beta}) = \boldsymbol{0}, \\ \text{such that} \quad & \boldsymbol{\beta}_{lo} \leq \boldsymbol{\beta} \leq \boldsymbol{\beta}_{hi}, \end{aligned} \tag{5}$$

Each component of $\boldsymbol{\beta}$ is transformed to $[-1,1]$ to promote even scaling of the components of the gradient of the objective function $\frac{d\mathcal{J}}{d\tilde{\boldsymbol{\beta}}}$ according to the component-wise formulas

$$\begin{aligned} \boldsymbol{s} &= \frac{1}{2}\left(\boldsymbol{\beta}_{hi} - \boldsymbol{\beta}_{lo}\right), \\ \boldsymbol{m} &= \frac{1}{2}\left(\boldsymbol{\beta}_{hi} + \boldsymbol{\beta}_{lo}\right), \\ \tilde{\boldsymbol{\beta}} &= \frac{\boldsymbol{\beta} - \boldsymbol{m}}{\boldsymbol{s}}. \end{aligned} \tag{6}$$

We assume that full-field data $\boldsymbol{d}$ is available at N load steps that coincide with the pseudo-temporal discretization of the plasticity model. The objective function is

$$\mathcal{J}\left(\boldsymbol{U}(\boldsymbol{\beta})\right) = \sum_{n=1}^{N} \mathcal{J}^n = \sum_{n=1}^{N} \int_{\Omega} \left(\boldsymbol{u}^n(\boldsymbol{\beta}) - \boldsymbol{d}^n\right) \cdot \left(\boldsymbol{u}^n(\boldsymbol{\beta}) - \boldsymbol{d}^n\right) d\Omega. \tag{7}$$

The backward Euler temporal discretization in the plasticity model couples neighboring load steps through the internal variables α and $\boldsymbol{G}^p$, which we collect into a vector $\boldsymbol{\xi}$. We write the coupled residual as a sum over load steps

$$\boldsymbol{\mathcal{R}} = \sum_{n=1}^{N} \boldsymbol{R}^n\left(\boldsymbol{U}^n, \boldsymbol{\xi}^{n-1}(\boldsymbol{u}^{n-1}), \boldsymbol{\beta}\right), \tag{8}$$

where $\boldsymbol{R}^n$ denotes the discrete residual for load step n.

In this work we use the bound-constrained optimization algorithm L-BFGS-B with a line search globalization as implemented by the Rapid Optimization Library (ROL), an optimization software package in Sandia's Trilinos project. It requires evaluations of the objective function and its gradient. In the following subsections we describe three approaches to obtain the objective gradient in order of increasing computational implementation complexity.

3.2.1 Gradient with Finite Differences

The finite difference approach is the most straightforward route to the gradient. All it requires is an implementation of the plasticity model and the ability to evaluate the objective function. Algorithm 1 contains the details for its computation. The simplicity of this scheme has resulted in its widespread adoption. Indeed, it used to calculate the gradient in essentially all of the FEMU literature.

Unfortunately, there several drawbacks associated with this approach. First, the accuracy of the approximation depends on the value of ε. Large values result in a larger truncation error while smaller values suffer from cancellation error due to the finite precision of floating point numbers. A "good" choice for ε lies in between these regimes. ROL chooses $\varepsilon = \sqrt{\epsilon}\beta_i$ for $\beta_i > \epsilon$ and $\varepsilon = \epsilon$ otherwise, where ϵ is machine epsilon. It is important to note that even if ε is judiciously chosen, the gradient will contain error.

A second disadvantage is the computational cost. The residual must be solved for each component of the gradient, resulting in a total of $P+1$ nonlinear, pseudo-transient solves per gradient evaluation. Thus this method scales poorly with the dimension of the parameter space.

Algorithm 1 Finite Difference Gradient

1. Given $\boldsymbol{\beta}$, solve the residual for $\boldsymbol{U}_{\text{ref}}$
2. Evaluate the objective function $\mathcal{J}_{\text{ref}} = \mathcal{J}(\boldsymbol{U}_{\text{ref}})$
3. Given $\varepsilon > 0$, compute each component of the P-dimensional gradient vector through a forward finite difference approximation:

FOR $i = 1, \ldots, P$ DO

$\hat{\boldsymbol{\beta}}_i = \boldsymbol{\beta} + \varepsilon \boldsymbol{e}_i$

Given $\hat{\boldsymbol{\beta}}_i$, solve the residual for $\hat{\boldsymbol{U}}_i$

Evaluate the perturbed objective function $\hat{\mathcal{J}}_i = \mathcal{J}(\hat{\boldsymbol{U}})_i$

$\left[\frac{d\mathcal{J}}{d\boldsymbol{\beta}}\right]_i = \frac{1}{\varepsilon}\left(\hat{\mathcal{J}}_i - \mathcal{J}_{\text{ref}}\right)$

DONE

3.2.2 Gradient with Forward Sensitivities

The forward or direct sensitivity approach requires intrusive modification of the finite element solver, but less so than the adjoint method described in the next section. We derive it by differentiating the objective function and applying the chain rule:

$$\frac{d\mathcal{J}}{d\boldsymbol{\beta}} = \sum_{n=1}^{N} \frac{d\mathcal{J}^n}{d\boldsymbol{U}^n} \frac{d\boldsymbol{U}^n}{d\boldsymbol{\beta}}. \tag{9}$$

The vector $\frac{d\mathcal{J}^n}{d\boldsymbol{U}^n}$ can be obtained analytically in our case or more generally through AD. The quantity $\frac{d\boldsymbol{U}^n}{d\boldsymbol{\beta}}$ is known as a sensitivity matrix, and its computation is more difficult. We differentiate the residual at load step n to obtain an expression for its evolution:

$$\frac{\partial \boldsymbol{R}^n}{\partial \boldsymbol{U}^n} \frac{d\boldsymbol{U}^n}{d\boldsymbol{\beta}} = -\frac{\partial \boldsymbol{R}^n}{\partial \boldsymbol{\beta}} - \frac{\partial \boldsymbol{R}^n}{\partial \boldsymbol{\xi}^{n-1}} \frac{d\boldsymbol{\xi}^{n-1}}{d\boldsymbol{U}^{n-1}} \frac{d\boldsymbol{U}^{n-1}}{d\boldsymbol{\beta}} \tag{10}$$

Note that $\frac{d\boldsymbol{U}^0}{d\boldsymbol{\beta}}$ is zero, as the initial condition is specified and has no dependence on the parameters. Equation (10) is solved one column at a time. Thus, in this approach P *linear* systems are solved at each load step per gradient evaluation.

The Jacobian matrix $\frac{\partial \boldsymbol{R}^n}{\partial \boldsymbol{U}^n}$ is required for the Newton iterations used to solve for $\boldsymbol{U}^n$. The parameter Jacobian $\frac{\partial \boldsymbol{R}^n}{\partial \boldsymbol{\beta}}$ is typically not present in finite element codes, but we can calculate it using AD. The term on the right hand side that involves derivatives of the right hand side with respect to the state variables from the previous time step is also computed with AD. We store $\boldsymbol{\xi}^n(\boldsymbol{U}^n)$ with its derivative information at the end of each converged load step so that it can be used to calculate this term.

The forward sensitivity approach lends itself to a memory-efficient implementation. After the Newton solve for load step n has converged, we compute that step's contribution to the gradient by constructing the necessary linear algebra objects and solving (10). We also store the sensitivity matrix $\frac{d\boldsymbol{U}^n}{d\boldsymbol{\beta}}$ and internal variables vector $\boldsymbol{\xi}^n$ for use in the subsequent load step and discard information that is no longer needed.

Unfortunately, this method still scales with the dimension of the parameters space, but it requires the solution of P linear equations at each load step. Therefore it outperforms the finite difference gradient approach in terms of computational cost but requires the construction and storage of additional linear algebra objects.

3.2.3 Gradient with Adjoint Sensitivities

The final method for computing the gradient is known as the adjoint approach. In this technique, a linear pseudo-transient PDE is solved for an auxiliary "adjoint" variable we denote by $\boldsymbol{\lambda}$. We derive the method by defining a

Lagrangian and make it stationary with respect to each of its arguments.

$$\mathcal{L}\left(\boldsymbol{U}, \boldsymbol{\lambda}, \boldsymbol{\beta}\right) = \sum_{n=1}^{N} \left(\mathcal{J}^n(\boldsymbol{U}^n) + (\boldsymbol{\lambda}^n)^T \boldsymbol{R}^n(\boldsymbol{U}^n, \boldsymbol{U}^{n-1}, \boldsymbol{\beta}) \right). \tag{11}$$

The adjoint variable is a Lagrange multiplier that enforces the plasticity PDE constraint. We recover $\boldsymbol{R}^n$ by differentiating (11) with respect to $\boldsymbol{\lambda}^n$. Like before, for a given $\boldsymbol{\beta}$ the constraint gives us a method to evaluate the corresponding $\boldsymbol{U}$ vector.

Differentiating (11) with respect to $\boldsymbol{U}^n$ to obtain the stationary condition and taking the transpose of both sides yields the adjoint equation for load step n:

$$\left(\frac{\partial \boldsymbol{R}^n}{\partial \boldsymbol{U}^n}\right)^T \boldsymbol{\lambda}^n = -\left(\frac{d\mathcal{J}^n}{d\boldsymbol{U}^n}\right)^T - \left(\frac{\partial \boldsymbol{R}^{n+1}}{\partial \boldsymbol{U}^n}\right)^T \boldsymbol{\lambda}^{n+1} \tag{12}$$

The adjoint PDE is a linear *terminal* boundary value problem. We specify the final condition $\boldsymbol{\lambda}^{N+1}$ to be zero. It requires the solution and storage of the entire time history of $\boldsymbol{U}$.

Finally, we derive the expression for the gradient. The Lagrangian and objective function are identical on the surface where the residual is zero. Additionally, the solution of (12) implies that the only non-zero contribution to the total differential of the Lagrangian is its derivative with respect to $\boldsymbol{\beta}$, which is equal to $\frac{d\mathcal{J}}{d\boldsymbol{\beta}}$. Therefore the adjoint-based gradient evaluation formula is

$$\frac{d\mathcal{J}}{d\boldsymbol{\beta}} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{\beta}} = \sum_{n=1}^{N} (\boldsymbol{\lambda}^n)^T \frac{\partial \boldsymbol{R}^n}{\partial \boldsymbol{\beta}}. \tag{13}$$

The greatest advantage of the adjoint-based gradient is that it is independent of the dimension of the parameter space. It is the most memory-intensive option, as it requires the storage of the entire time history of $\boldsymbol{U}$. Checkpointing algorithms can reduce the amount of the storage, but further increase the computational effort. It does not require storing the entire time history of $\boldsymbol{\lambda}$. The gradient can be computed as the adjoint equation is being solved backwards in time, and the adjoint solution can be discarded after its contribution to the right hand side of (12) has been computed.

4 Results

In this section we present two problems that highlight the benefits of using adjoint or forward sensitivities (FS) approaches over FEMU with finite difference-based gradients. First, we present a plane strain example to gauge the performance and accuracy of each method. The second example shows the application of the adjoint-based approach to a 3D problem. We choose initial guesses that are relatively close to the exact solution as our focus in this project was the demonstration of the computational cost and accuracy of each approach rather than robustness of the inverse methods, which can be examined and improved upon in subsequent work.

We generated synthetic DIC data by solving a forward problem using a specified geometry, mesh, BCs, and parameter vector $\boldsymbol{\beta}$. Noise is then added to the displacement field to create a synthetic "measured" dataset. We note that the amount of noise added is higher than what is typically observed experimental DIC displacement fields, which appear considerably smoother visually than the noisy datasets shown in Figures 1 and 2.

The two-dimensional example is a plate with a circular hole cut out of the lower left corner. In this problem we use non-dimensionalized units for the material properties and geometric dimensions, and discretize the domain into approximately $1,500$ triangle elements. We set the parameters S and D to zero, resulting in a linear, isotropic hardening model. The displacement BCs were $u_x = 0$ on the left edge of the plate and u_y = 0 on the bottom edge. The top was pulled upwards with a vertical traction vector $\boldsymbol{h}(t) = 3t\boldsymbol{e}_y$. A pseudo-time discretization of ten load steps was used. Figure 1 displays the noiseless and noise-corrupted y component of the synthetic DIC measurement on the undeformed mesh and the equivalent plastic strain α on the deformed mesh at the final load step.

We tested the FEMU, forward sensitivities, and adjoint approaches using noiseless data and noise-corrupted data. The "measured" data $\boldsymbol{d}$ is stored at integration points and entered in through data-fields in the mesh.

The initial guess for the optimization procedure, estimated parameters, and inversion time for the noise-corrupted 2D problem are given in Table 1. We do not show the noiseless results because each approach was able to match the

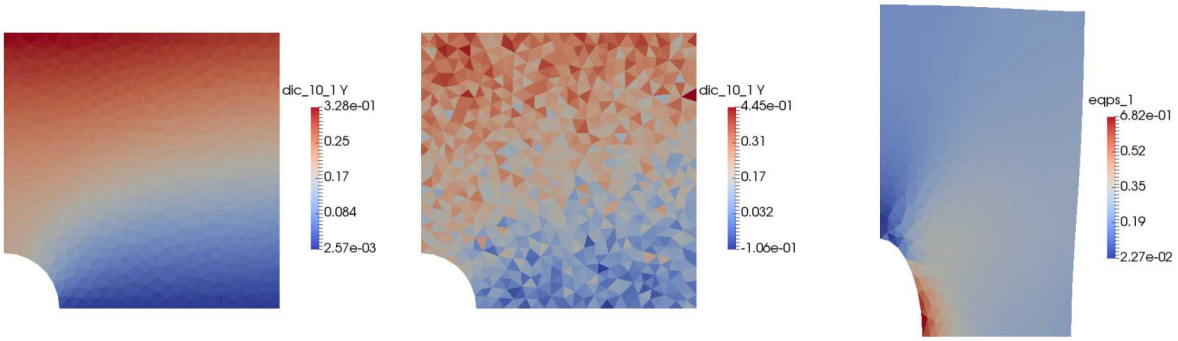

Figure 1: The 2D notch problem. All displayed fields are from the final load step. Left: Noiseless u_y. Middle: u_y with zero mean Gaussian noise with $\sigma = 0.05$. Right: Equivalent plastic strain α on the deformed mesh.

	E	ν	K	Y	Time
Truth	1000	0.25	10	100	n/a
Initial	1100	0.32	9	110	n/a
FEMU	919.4 (8.0%)	0.259 (3.6%)	10.32 (3.2%)	99.43 (0.57%)	9m 41s
FS	935.8 (6.4%)	0.255 (2.0%)	10.30 (3.0%)	99.26 (0.74%)	1m 59s
Adjoint	933.9 (6.6%)	0.255 (2.0%)	10.30 (3.0%)	99.27 (0.73%)	1m 51s

Table 1: Inverse problem solutions for the plane strain notch problem. % error relative to the true solutions are reported next to the parameter values.

true parameters to six or more digits. All of these problems were solved using a single core on the same machine to make timing comparisons fair. We note that except for E, the recovered parameters from each method agree to a few significant figures.

The second example is a 3D problem that was executed using four MPI ranks. This example demonstrates three features of our computational implementation: the high-performance and parallel nature of the code, the ability to calibrate a nonlinear hardening model, and the applicability of the formulation to 3D.

In this problem a specimen with two circular notches was studied. We fixed all components of the displacement field to zero at the bottom of the material, and applied a vertical traction $\boldsymbol{h}(t) = 0.5t\boldsymbol{e}_y$ over ten integer-valued load steps.

The non-dimensionalized model parameters were chosen so that there was a non-negligible amount of plastic strain in the material by the final load step and the hardening model was exponential (i.e. K was fixed to zero). The geometry and noise-corrupted displacement data is displayed for the middle and final load steps, as well as the equivalent plastic strain at the final step on the deformed configuration are shown in Figure 2. Noise with a standard deviation of 0.001 was used in this problem. This amount of noise closer to the noise floor of experimental DIC data than that of the plane stain example, but is still considerably larger.

The mesh in this problem contained around $30,000$ tetrahedral elements, resulting in a total of approximately $25,000$ degrees of freedom for the displacement and pressure state variables. Like in the previous problem, we assume that displacement data is available at the center of each element. This assumption doesn't make sense for DIC, but it would for its 3D generalization digital volume correlation (DVC). Like DIC, DVC is being actively researched and developed at Sandia, although applications of full-field inversion techniques to DVC has received considerably less attention.

The parameters used to generate the synthetic data, initial guess, and inversion results are presented in Table 2. The parameters are all recovered to within 2% of their true values. We expect that a similar simulation performed with FEMU would take nearly six hours due to the additional nonlinear solves required for each component of the finite difference gradient.

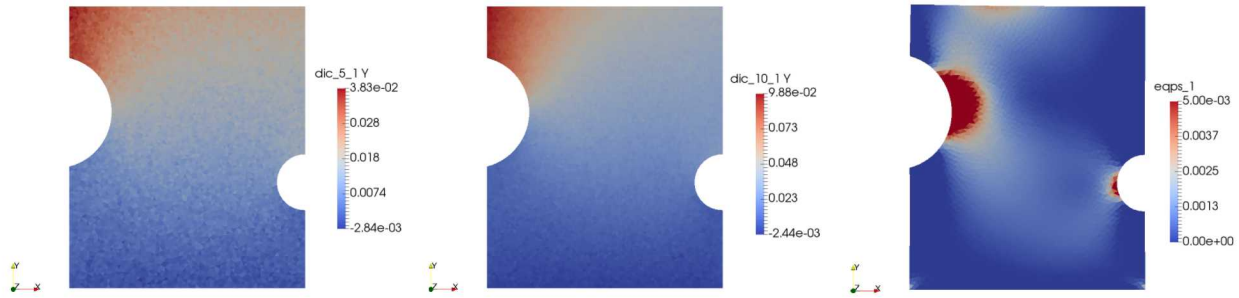

Figure 2: The 3D model problem. Left: Noiseless u_y at load step five. Middle: u_y with zero mean Gaussian noise with $\sigma = 0.001$. Right: Equivalent plastic strain α on the deformed mesh. This plot is scaled to highlight the variation in this field. The unscaled maximum value is 0.0274.

	E	ν	K	S	D	Time
Truth	1000	0.32	5	5	300	n/a
Initial	1050	0.34	6	5.5	315	n/a
Final	999.98 (0.0022%)	0.3199 (0.028%)	4.988 (0.24%)	4.9865 (0.27%)	304.8 (1.6%)	44m 24s

Table 2: Summary of the results for the 3D model problem. % error relative to the true parameter value is reported next to the estimated value.

5 Discussion

The results from the previous section conclusively show that both forward and adjoint-based sensitivity computations can dramatically accelerate the solution time of parameter identification problems. In this section we discuss various enhancements to our existing framework that would facilitate the use of experimental DIC or DVC data. We conclude by introducing a coupled formulation of the inverse problem that may have advantages over the one we implemented.

The code we produced requires several modifications before it could be used to analyze DIC data from real mechanical tests. These include the mapping of experimental data into the mesh to inform Dirichlet BCs and the objective function, the integration of load information into the inverse problem, and modifications to address the planar nature of DIC data.

DIC data is a point cloud, and its spatial positions have no relation to the FE mesh on which the inverse problem is solved, *unless the mesh is specially designed*. It is up to an analyst to either interpolate the data to the Gauss points used in the simulation or appropriately modify the objective function to explicitly match the FE displacement field at the native locations of the DIC data points. The density of DIC data is extremely high, so a scheme that could accommodate either option would be valuable in practice. Attempting to include every measurement could easily result in a mesh with millions of elements, such that an adjoint-based approach would be the only feasible option. Finally, all DIC codes produce an uncertainty metric that could be used as a weighting term in the objective function.

In our current model displacement BCs (DBCs) are assumed. With real experimental data it is much better to impose the BCs are measured by DIC as DBCs along all edges of the mesh. However, there is a technical issue when the forward problem is entirely driven by DBCs. Force information is required to determine the absolute units for Young's modulus, the yield stress, and some of the hardening moduli. Otherwise they can only be determined up to a multiplicative constant. In our example problems we circumvented this issue by prescribing traction BCs , but in practice the traction distribution is unknown. The only available load information is a resultant force measured by a load cell, a global time-varying quantity.

This information can be integrated into the inverse problem by adding a force-matching term to the objective function so that the modified objective is

$$\mathcal{J}_{\text{FM}}(\boldsymbol{\beta}) = \sum_{n=1}^{N} \left[\mathcal{J}^n + \frac{1}{2} \left(\boldsymbol{F}^n - \int_{\Gamma_t} \boldsymbol{P}^n(\boldsymbol{\beta}) \cdot \boldsymbol{N} \, d\Gamma \right)^2 \right], \tag{14}$$

where $\boldsymbol{F}$ is the resultant force. With this approach the DIC data can drive the forward problem through the DBCs

and uniquely calibrated parameters can be obtained. This extra term will alter both the adjoint and forward sensitivity approaches, but the use of AD makes computing these contributions straightforward.

We are currently limited to synchronizing load steps with DIC data frames. In practice, solvers for plastically-deforming finite elements adapt the time step to ensure robust performance of the nonlinear solver. With this modification, the pseudo-time discretization and that of the inverse problem data will no longer be aligned. An inversion code with adaptive time-stepping would need to ensure that there is always a simulation data step for each DIC data frame.

Finally, we used either a plane strain or full 3D model in this project. In practice, the plane stress assumption is a more appropriate approximation than plane strain, because many test samples in material characterization are thin. Furthermore, DIC data is only available on a surface. This issue can be addressed by either adjusting the objective function for the 3D problem so that the displacement mismatch term acts over the DIC surface rather than throughout the volume, or to use a plane stress model and extend the DIC deformation throughout the thickness dimension in the model. It is not trivial, however, to impose a plane stress approximation in plasticity FE models. It requires the imposition of an additional constraint term, which would modify the entire problem formulation.

It is standard in PDE-constrained optimization to assess the accuracy of forward sensitivity and adjoint-based gradients by comparing them to finite difference (FD) gradients obtained using a decreasing sequence of FD perturbations. It is typically the case that properly implemented gradients will match the FD gradient to six or seven digits when iterative linear solver tolerances are "tight."

We only observed one to four digits of accuracy in the gradients produced by our adjoint and forward sensitivity implementations. It is unlikely that there is an error the code used to perform the "FD check," and it is difficult to determine if and what is incorrect in our formulation because it performs essentially identically to FEMU on the model problems we examined.

After a considerable amount of debugging and contemplation, we determined that a reformulation of the inverse problem with a coupled set of nonlinear algebraic constraints could be a route to obtaining numerically-exact gradients. In this formulation we would utilize the following Lagrangian:

$$\hat{\mathcal{L}}\left(\boldsymbol{U},\boldsymbol{\xi},\boldsymbol{\eta},\boldsymbol{\phi},\boldsymbol{\beta}\right)=\sum_{n=1}^{N}\left(\mathcal{J}^{n}(\boldsymbol{U}^{n})+(\boldsymbol{\eta}^{n})^{T}\boldsymbol{R}^{n}(\boldsymbol{U}^{n},\boldsymbol{\xi}^{n},\boldsymbol{\beta})+(\boldsymbol{\phi}^{n})^{T}\boldsymbol{C}^{n}(\boldsymbol{U}^{n},\boldsymbol{\xi}^{n},\boldsymbol{\xi}^{n-1}\boldsymbol{\beta})\right). \tag{15}$$

In this problem the momentum equation residual and constitutive equation evolution are now explicity rather than implicity coupled. Thus we have to introduce *two* Lagrange multipliers (i.e. adjoint variables) $\boldsymbol{\eta}$ and $\boldsymbol{\phi}$.

The momentum residual $\boldsymbol{R}$ is satisfied at nodes, while $\boldsymbol{C}$ is enforced at Gauss points. We expand the vector $\boldsymbol{\xi}$ to include the Cauchy stress, the plastic multiplier, equivalent plastic strain, and $\boldsymbol{G}^{p}$. This approach is more complex, but lends itself to advanced sensitivity analyses and could lead to novel computationally efficient inverse problem formulations.

6 Anticipated Outcomes and Impacts

We believe that this work motivates a significant investment in the use of AD-based sensitivities for constitutive parameter identification. We have demonstrated in example problems that the use of this approach can reduce the running time of model calibration by a factor of 5x without sacrificing accuracy. Adjoint-based gradients have no dependence on the dimension of the parameter space, so the gap between their computational cost and that of FEMU will only widen as models with more parameters are considered.

In FEMU there is a reliance on finite differences to obtain parameter sensitivities. The use of AD increases both the computational speed of obtaining sensitivity information as well as making the development of more complicated plasticity models (e.g. viscoplastic, anisotropic, damage) considerably more straightforward than traditional approaches. It completely removes the to manually derive consistent tangent elastoplastic stiffness moduli.

At this point in time, full-field experimental data is being necessarily discarded due to the computational expense of current parameter identification techniques. Typical DIC experiments acquire images at a rate close to sixty frames per second. In solid mechanics, finite element models of plastic deformation are most often stated in a total Lagrangian formulation and thus with a Lagrangian mesh. The deformation imposed on mechanical test specimens for characterization purposes can be extremely large, and thus a fine mesh is used to avoid element distortion that leads to a loss of numerical accuracy. Given a fixed computational budget, a finer spatial and/or time discretization can be utilized in a numerical model with an adjoint-based approach.

One anticipated outcome of this project is a journal article that summarizes this effort that will be submitted to *Computer Methods in Applied Mechanics and Engineering*. We began a draft that covers the stabilized mechanical problem, its discretization, and our optimization approaches, but stalled on its submission when we discovered the discrepancy between our sensitivity analysis and adjoint-based gradients and the FD gradient. It is highly probable that a reviewer would ask if we performed this check and would demand an explanation as to why ours do not match.

Therefore we would like to recast the problem in terms of the coupled residual formulation mentioned in the previous section, as we feel it has a better chance of obtaining numerically-exact sensitives. We plan on mentioning the results of the formulation utilized in this work in the article to show that through a simpler formulation, results that are comparable to FEMU can be obtained with significantly less computational effort.

There has been criticism in the topology optimization community about the expense of using AD, even for the solution of the forward model. Sandia is one of the few institutions that utilizes AD in topology optimization, but most of their work involves forward sensitivity analyses. We demonstrated in this project that we could produce an adjoint-based HPC optimization code using AD as a foundation and plan to share our results with researchers in topology optimization to determine if our ideas could be useful in that arena.

The first author was a member of an LDRD project that ended this FY lead out of 1500 that explored novel full-field material inversion methods distinct from the examined in this project. Given the positive results of this exploratory LDRD, we plan to include enhancements of our current approach and its application to experimental datasets in any proposals that aim to extended the work of that LDRD.

An important next step for this method would be a validation study. Besides demonstrating its use on experimental data, we would also need to show that it provides parameter estimates that are consistent with traditional mechanical characterization tests. This study would lend the adjoint-based approach credibility among experimentalists and constitutive modelers and provide motivation for application to more complicated plasticity models that are difficult to calibrate using current techniques.

Finally, an overview of this project was presented in the material characterization (MC^2) symposium. This workshop involved approximately fifty researchers from Sandia's Albuquerque and California sites in the areas of constitutive modeling, NW systems analysts, and experimentalists. Participation in this workshop gave the authors an impression that parameter estimation for plasticity models is an important problem to Sandia's mission. We plan to communicate the results of this project to attendees of the workshop to fund further progress in advanced approaches to constitutive model parameter identification.

7 Conclusion

In this work we demonstrated a full-field inversion capability for an elastoplastic, finite deformation mechanical model. Our reliance on AD for the residual, objective function, and gradient calculations is a first for plasticity models. We were able to verify our results on a simple example and show the scalability of the implementation in a 3D problem. This approach could be modified to calibrate parameters in a wide range of plastic material models using experimental DIC or DVC displacement data.