SANDIA REPORT

SAND2014-17871
Unlimited Release
Printed September 2014

Implementation and Evaluation of the Virtual Fields Method: Determining Constitutive Model Parameters From Full-Field Deformation Data

Sharlotte L.B. Kramer, William M. Scherzinger

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone:	(865) 576-8401
Facsimile:	(865) 576-5728
E-Mail:	reports@adonis.osti.gov
Online ordering:	http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone:	(800) 553-6847
Facsimile:	(703) 605-6900
E-Mail:	orders@ntis.fedworld.gov
Online ordering:	http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2014-17871
Unlimited Release
Printed September 2014

Implementation and Evaluation of the Virtual Fields Method: Determining Constitutive Model Parameters From Full-Field Deformation Data

Sharlotte L.B. Kramer
Structural Mechanics Laboratory
Sandia National Laboratories
P.O. Box 5800
Mail Stop 0557
Albuquerque, NM 87185-0557
slkrame@sandia.gov

William M. Scherzinger
Solid Mechanics Department
Sandia National Laboratories
P.O. Box 5800
Mail Stop 0840
Albuquerque, NM 87185-0840
wmscher@sandia.gov

Abstract

The Virtual Fields Method (VFM) is an inverse method for constitutive model parameter identification that relies on full-field experimental measurements of displacements. VFM is an alternative to standard approaches that require several experiments of simple geometries to calibrate a constitutive model. VFM is one of several techniques that use full-field experimental data, including Finite Element Method Updating (FEMU) techniques, but VFM is computationally fast, not requiring iterative FEM analyses. This report describes the implementation and evaluation of VFM primarily for finite-deformation plasticity constitutive models. VFM was successfully implemented in MATLAB and evaluated using simulated FEM data that included representative experimental noise found in the Digital Image Correlation (DIC) optical technique that provides full-field displacement measurements. VFM was able to identify constitutive model parameters for the BCJ plasticity model even in the presence of simulated DIC noise, demonstrating VFM as a viable alternative inverse method. Further research is required before VFM can be adopted as a standard method for constitutive model parameter identification, but this study is a foundation for ongoing research at Sandia for improving constitutive model calibration.

Acknowledgment

This work was funded under LDRD Project Number 165667 and Title“Determining Constitutive Material Properties From Full-Field Experiments Using the Virtual Fields Method”.

We would like to acknowledge the contributions of the following collaborators who helped with the research during the course of the project: Phillip Reu for providing the experimental study of DIC noise; Miguel Aguilo for assisting with optimization algorithms in MATLAB; and Glen Hansen for providing insight on mapping numerical data between different meshes.

Contents

Appendix

List of Figures

List of Tables

Chapter 1

Introduction

Sandia requires reliable predictive capabilities for both safety and design of their systems for a variety of demanding environments. One area for improvement is more robust material modeling. Two ubiquitous problems that face scientists and engineers during any study that involves materials are (1) what material models are appropriate in this situation and (2) what are the material parameters in those models for the materials of interest. Selection and calibration of material models are key capabilities for predictive computational simulations. Materials scientists and solid mechanicians develop material models for new materials systems and for complex material deformation to be used in design and system characterization.

At first glance, these problems appear solved with current techniques, but these techniques have limitations that can result in poor predictive capabilities and poor understanding of material behavior. The standard practice is to extrapolate material properties from simple tests such as the uniaxial tension test. While quite straight forward for an isotropic material, a single uniaxial tension test is not sufficient to characterize shear or torsional behavior, anisotropy, spatially heterogeneous material properties, damage evolution, different fracture modes, combined loading, strain-rate dependent behavior, among many others. Despite these limitations, analysts often only have the time and funding to support simple material tests with mixed results.

When more fidelity is required, researchers and analysts often require multiple experiments to help with developing and calibrating an appropriate material model for their applications; but, multiple experiments take considerable time to complete and are rather complex, which is rather demanding for Sandia's experimental capabilities. One response to a need for more data from complex experiments is for the experimentalists to measure full-field quantities such as displacements and temperatures with advanced optical techniques and specialized equipment. Performing experiments with increased complexity where quantities of interest vary spatially in essence allows for multiple types of deformation in fewer experiments, hence the appeal of full-field measurements during these experiments. Unfortunately, interpretation of full-field experimental data or even global measurements from multiple experiments is not straightforward, i.e., the inverse problem to determine material properties from data from many experiments and/or full-field experiments is far more difficult than for a uniaxial tension test. The tools to solve these inverse problems are in their nascent form and require extensive research and development.

The experimental mechanics research community has developed several approaches for inverse problem solving for material properties identification using full-field experimental data over the past two decades. To clarify, the forward problem is the determination of forces and stresses over the problem domain given deformation / loading information, domain geometry, and calibrated material models. The inverse problem is the determination of the material model parameters that connect measured force and displacements. The inverse problem must first be solved before the forward problem can be solved for a different geometry and loading condition.

The basic types of inverse problem methods are described in detail in [1]. The first type is a finite element method updating (FEMU) technique, where either a measured boundary force or displacement field over the domain is used to drive a FEM simulation that iteratively adjusts material properties in a chosen model to minimize an objective function comparing the experimental and simulation displacements or forces. The second type is called the Constitutive Equation Gap Method (CEGM), where an FEM simulation and an overdetermined set of displacement measurements and force measurements are compared using an equilibrium equation that contains a material model, hence an iterative minimization of the equilibrium gap for material parameter identification. The third type is based on the Principle of Virtual Work, called the Virtual Fields Methods (VFM) [1–13], where full-field measurements like displacements or strains, and global resultant force measurements are used in the Principle of Virtual Work along with advantageously chosen, kinematically admissible, virtual fields, to directly solve for the material parameters without need for iterative FEM simulations.

The appeal of the FEMU and CEGM methods is that a complete set of full-field data over the entire domain is not required, but the iterative simulation requirements can be cost- and time-prohibitive. Additionally, these techniques require that the user selects a material model before the iterative search, so the effort invested in these methods would be multiplied by the number of material models that the user would like to try out, which is not promising for material model selection or development. The appeal of VFM is that it does not require iterative FEM simulations, thus identifying material parameters much more quickly, allowing for identification of properties for many different material models using the same experimental data. The main issue with VFM is that it does require full-field experimental data over the entire domain, which is difficult for some applications, particularly where through-thickness assumptions (i.e. plane stress) do not apply, requiring internal measurements that are hard to obtain.

Given that there are research opportunities to deal with this issue with VFM (i.e. kinematic assumptions and full-field volumetric displacement measurement techniques using computed tomography (CT) scans), this method has great promise to be an effective and fast tool for material parameter identification. This technique is powerful in that VFM (1) enables comparison of different material models, helping analysts choose the best models for their materials and giving them greater reliability in the material behavior in their simulations; (2) aides constitutive modelers and materials scientists in the development of new constitutive models; and, (3) extends to many different deformations and material models (including rates, temperatures, complex mixed-mode loading). Essentially, VFM is widely applicable to any problem wherever the Principle of Virtual Work can be applied in a configuration where full-field experimental measurements can be made.

This report describes recent efforts at Sandia to implement and evaluate VFM for constitutive parameter identification primarily for large deformation plasticity models, but also includes an example of how VFM may be utilized for small-strain theory. This study includes the implementation of VFM in MATLAB and evaluation of the VFM identification capability using FEM-simulated deformation data. In order to simulate experimental measurement uncertainties and their effect on the constitutive parameter identification process, this study also describes the effects of uncertainties associated with Digital Image Correlation (DIC) displacement measurements, which is currently the most practical experimental technique to gather full-field displacement data. The complete VFM inverse problem methodology is simulated using computationally derived displacements with superimposed DIC measurement uncertainties, which were quantified for representative DIC experimental setups, as inputs to the identification algorithm. The simulated VFM process that incorporates measurement errors allows for characterizing the impact of DIC uncertainties based on realistic experimental parameters without requiring cost-prohibitive iterative experimental investigations of the full VFM process. This study can help with specimen-geometry and DIC-setup designs so that the real experiments provide optimal measurements for lower uncertainties in finite deformation VFM material property identification. The examples described in this report consider the deformation of a ductile metal, 304L stainless steel, using nominal constitutive model parameters.

Chapter 2

The Virtual Fields Method

The Virtual Fields Method (VFM) is an approach to constitutive model parameter identification that utilizes full-field deformation measurements and the Principle of Virtual Work. VFM was first developed in the late 1980s and has been applied to static, vibrational, and high strain rate deformations for several materials including composites, metals, polymers, foam, and wood [2]. In general, VFM utilizes heterogeneous displacement field data from a mechanical test and global force data with well-defined boundary conditions for the area of interest. An analytical constitutive model is chosen for the material, and the virtual fields are carefully constructed depending on the type of model, allowing for determination of the constitutive parameters through numerically solving a system of equations that enforces the principle of virtual work, as previously has been addressed in literature [3–7]. Currently, there is limited work in VFM applied to metals for elasto-plasticity [8], heterogeneous elasto-plasticity [9], elasto-visco-plasticity [10], and 3D plasticity in the necking regime [11–13]. Depending on the application, VFM can be written in terms of small-strain theory or finite-deformation theory, which are both described in the following sections.

2.1 VFM for Small-Strain Theory

Given a volume V with internal stress σ_{ij}, internal strain ϵ_{ij}, applied tractions T_i on a surface S applied volumetric body forces b_i, density ρ, and acceleration a_i, the Principle of Virtual Work for a virtual displacement u_i^* and virtual strain ϵ_{ij}^*, in general, is

$$-\int\limits_V \sigma_{ij}\epsilon_{ij}^* \,\mathrm{d}V + \int\limits_S T_i u_i^* \,\mathrm{d}S + \int\limits_V b_i u_i^* \,\mathrm{d}V = \int\limits_V \rho a_i u_i^* \,\mathrm{d}V \quad \forall\ u_i^* \text{ KA}, \tag{2.1}$$

where KA means kinematically admissible,

$$W_{int}^* = -\int\limits_V \sigma_{ij}\epsilon_{ij}^* \,\mathrm{d}V \tag{2.2}$$

is the virtual work due to internal forces,

$$W_{ext}^* = \int\limits_S T_i u_i^* \,\mathrm{d}S + \int\limits_V b_i u_i^* \,\mathrm{d}V \tag{2.3}$$

is the virtual work due to external forces, and

$$W^*_{acc} = \int\limits_V \rho a_i u^*_i \, \mathrm{d}V \tag{2.4}$$

is the virtual work due to acceleration [2]. The relationship between the virtual displacement u^*_i and virtual strain ϵ^*_{ij} for *small deformations* is the same as that between the actual displacement and strain [2]:

$$\epsilon^*_{ij} = \frac{1}{2}\Big(u^*_{i,j} + u^*_{j,i}\Big). \tag{2.5}$$

Kinematically admissible u_i implies that the virtual displacement field is $\mathcal{C}(0)$, meaning it is continuous and differentiable, and that the virtual displacement field is consistent with the boundary conditions of the problem. The virtual strain field ϵ_{ij} can be discontinuous. The virtual fields do not necessarily have to represent actual displacements and strains, allowing for an infinite number of potential virtual fields. Some virtual fields are more advantageous for the purpose of determination of constitutive parameters, but the only actual restriction is the kinematic admissibility.

For plane stress conditions where $\sigma_{13} = \sigma_{23} = \sigma_{33} = 0$, the Principle of Virtual Work further reduces to the following, assuming u^*_1 and u^*_2 are independent of x_3:

$$-t\int\limits_S (\sigma_{11}\epsilon^*_{11} + \sigma_{22}\epsilon^*_{22} + 2\sigma_{12}\epsilon^*_{12})\,\mathrm{d}S + t\int\limits_{L_f} T_i u^*_i \,\mathrm{d}l + t\int\limits_S b_i u^*_i \,\mathrm{d}S = t\int\limits_S \rho a_i u^*_i \,\mathrm{d}S \quad \forall\ u^*_i \,\mathrm{KA}, \tag{2.6}$$

where t is the thickness of the specimen, and L_f is the line over which the traction force is applied [2]. For deformations without body forces and under quasi-static loading, the Principle of Virtual Work reduces to the following, after dividing by t and switching to Voigt notation:

$$\int\limits_S (\sigma_1\epsilon^*_1 + \sigma_2\epsilon^*_2 + \sigma_6\epsilon^*_6)\,\mathrm{d}S = \int\limits_{L_f} T_i u^*_i \,\mathrm{d}l \quad \forall\ u^*_i \,\mathrm{KA}. \tag{2.7}$$

A linear elastic isotropic material model in plane stress for elastic modulus E and Poisson's ratio ν is

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_6 \end{bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} & 0 \\ Q_{12} & Q_{11} & 0 \\ 0 & 0 & Q_{66} \end{bmatrix} \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_6 \end{bmatrix}, \tag{2.8}$$

where

$$Q_{11} = Q_{22} = \frac{E}{1-\nu^2},\ Q_{12} = Q_{21} = \frac{\nu E}{1-\nu^2},\ \text{and}\ Q_{66} = \frac{Q_{11} - Q_{12}}{2} = \frac{E}{2(1+\nu)}. \tag{2.9}$$

This can be written as

$$\begin{bmatrix} \epsilon_1 \\ \\ \epsilon_2 \\ \\ \epsilon_6 \end{bmatrix} = \begin{bmatrix} \dfrac{Q_{11}}{Q_{11}^2 - Q_{12}^2} & \dfrac{Q_{12}}{Q_{11}^2 - Q_{12}^2} & 0 \\ \dfrac{Q_{12}}{Q_{11}^2 - Q_{12}^2} & \dfrac{Q_{11}}{Q_{11}^2 - Q_{12}^2} & 0 \\ 0 & 0 & \dfrac{1}{Q_{66}} \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \\ \sigma_2 \\ \\ \sigma_6 \end{bmatrix}. \tag{2.10}$$

Inserting Eq. (2.8) into Eq. (2.7) gives the following:

$$\int\limits_S Q_{11}\epsilon_1\epsilon_1^*\mathrm{d}S + \int\limits_S Q_{11}\epsilon_2\epsilon_2^*\mathrm{d}S + \int\limits_S Q_{12}(\epsilon_1\epsilon_2^* + \epsilon_2\epsilon_1^*)\mathrm{d}S + \int\limits_S \frac{Q_{11} - Q_{12}}{2}\epsilon_6\epsilon_6^*\mathrm{d}S = \int\limits_{L_f} T_i u_i^*\mathrm{d}l \quad \forall\ u_i^*\mathrm{KA}. \tag{2.11}$$

Assuming that the material is homogeneous allows for the following simplification of the Principle of Virtual Work in Eq. (2.11) [2]:

$$Q_{11}\int\limits_S \left(\epsilon_1\epsilon_1^* + \epsilon_2\epsilon_2^* + \frac{1}{2}\epsilon_6\epsilon_6^*\right)\mathrm{d}S + Q_{12}\int\limits_S \left(\epsilon_1\epsilon_2^* + \epsilon_2\epsilon_1^* - \frac{1}{2}\epsilon_6\epsilon_6^*\right)\mathrm{d}S = \int\limits_{L_f} T_i u_i^*\mathrm{d}l \quad \forall\ u_i^*\mathrm{KA}. \tag{2.12}$$

2.2 VFM for Finite Deformation

VFM has been applied to finite-deformation plasticity [11, 12] using a nonlinear minimization process where the constitutive model with an unknown parameter set ξ, measured displacements $\boldsymbol{u}(\boldsymbol{x}_o, t)$ relative to the reference material coordinates $\boldsymbol{x}_o$, and user-defined virtual fields are inserted into the Principle of Virtual Power for finite deformations to satisfy equilibrium, resulting in the desired ξ.

Consider a volume V whose current position is given by $\boldsymbol{x}_t$. The motion / deformation of that volume is given by $\boldsymbol{\chi}$, and the displacement of the body relative to a reference configuration V_o with reference material position $\boldsymbol{x}_o$ is

$$\boldsymbol{u}(\boldsymbol{x}_o, t) = \boldsymbol{\chi}(\boldsymbol{x}_o, t) - \boldsymbol{x}_o. \tag{2.13}$$

The deformation gradient $\mathbf{F}$ is then

$$\mathbf{F}(\boldsymbol{x}_o, t) = \nabla\boldsymbol{\chi}(\boldsymbol{x}_o, t) = \nabla\boldsymbol{u}(\boldsymbol{x}_o, t) + \mathbf{I}. \tag{2.14}$$

The volume is subjected to tractions $\boldsymbol{t}$ and body forces $\boldsymbol{b}$ in general. For a kinematically admissible velocity vector field $\delta\mathbf{v}$, assuming quasi-static loading and no body forces, the Principle of Virtual Power in the current configuration is

$$\int\limits_{V_t} \boldsymbol{\sigma} \cdot \delta\mathbf{D}\ dV = \int\limits_{\partial V_t} (\boldsymbol{\sigma}\boldsymbol{n}_t) \cdot \delta\mathbf{v}\ dS \tag{2.15}$$

where $\boldsymbol{\sigma}$ is the Cauchy stress, $\boldsymbol{n}_t$ is the surface normal in the current configuration, and the virtual stretch rate $\delta\mathbf{D}$ is

$$\delta\mathbf{D} = \frac{1}{2}(\nabla\delta\mathbf{v} + \nabla^T\delta\mathbf{v}). \tag{2.16}$$

An equivalent form of the Principle of Virtual Power is in terms of the Lagrangian description:

$$\int\limits_{V_o} \mathbf{\Pi}^T : \delta\dot{\mathbf{F}}\, dV = \int\limits_{\partial V_o} (\mathbf{\Pi}^T\boldsymbol{n}_o)\cdot\delta\mathbf{v}\, dS \tag{2.17}$$

where $\mathbf{\Pi}$ is the first Piola-Kirchoff stress tensor, $\delta\dot{\mathbf{F}} = \nabla\delta\mathbf{v}(\boldsymbol{x}_o, t)$ is the virtual velocity gradient tensor, ∂V_o is the boundary of the reference volume, and $\boldsymbol{n}_o$ is the surface normal in the undeformed configuration.

The purpose of VFM is to identify a constitutive parameter set ξ from measured full-field displacements and applied loads for heterogeneous stress states during the course of a test. By choosing kinematically admissible virtual velocity fields and appropriately managing the nonlinear VFM problem for the case of a nonlinear constitutive model, the user can identify ξ by minimizing the following cost function $\Phi(\xi)$:

$$\Phi(\xi) = \sum_{i=1}^{N_{vf}}\sum_{j=1}^{N_{step}} \Big[\int\limits_{V_o} \mathbf{\Pi}_j^T : \delta\dot{\mathbf{F}}_i\, dV - \int\limits_{\partial V_o} (\mathbf{\Pi}_j^T\boldsymbol{n}_o)\cdot\delta\mathbf{v}_i\, dS\Big]^2 \tag{2.18}$$

where N_{vf} is the number of virtual fields, and N_{step} is the number of measurement steps. Eq. (2.18) is the balance of the internal virtual power in the volume and the external virtual power on the boundary. $\mathbf{\Pi}$ is dependent on the measured displacements and the constitutive parameter set ξ and can be written in terms of the Cauchy stress $\boldsymbol{\sigma}$ (i.e. $\mathbf{\Pi}^T = (\det\mathbf{F})\,\boldsymbol{\sigma}\cdot\mathbf{F}^{-\mathbf{T}}$, where $\mathbf{F}$ is the actual deformation gradient tensor). The Cauchy stress can be written in terms of the deviatoric stress tensor and the hystrostatic pressure such that $\boldsymbol{\sigma} = \mathbf{s} + \frac{1}{3}(\mathrm{tr}\boldsymbol{\sigma})\,\mathbf{I}$. The internal virtual power can be rewritten as

$$\begin{aligned}
\int\limits_{V_o} \mathbf{\Pi}^T : \delta\dot{\boldsymbol{F}}\, dV &= \int\limits_{V_o} ((\det\mathbf{F})\boldsymbol{\sigma}\cdot\mathbf{F}^{-\mathbf{T}}) : \delta\dot{\mathbf{F}}\, dV \\
&= \int\limits_{V_o} ((\det\mathbf{F})(\mathbf{s} + \frac{\mathbf{1}}{\mathbf{3}}(\mathrm{tr}\boldsymbol{\sigma})\mathbf{I})\cdot\mathbf{F}^{-\mathbf{T}}) : \delta\dot{\mathbf{F}}\, dV \\
&= \int\limits_{V_o} ((\det\mathbf{F})\mathbf{s}\cdot\mathbf{F}^{-\mathbf{T}}) : \delta\dot{\mathbf{F}}\, dV + \int\limits_{V_o} ((\det\mathbf{F})(\frac{\mathbf{1}}{\mathbf{3}}(\mathrm{tr}\boldsymbol{\sigma})\mathbf{I})\cdot\mathbf{F}^{-\mathbf{T}}) : \delta\dot{\mathbf{F}}\, dV.
\end{aligned} \tag{2.19}$$

Additionally, the distribution of an applied force on the boundary ∂V_o is generally unknown, and only the resultant load $\boldsymbol{f}$ is measured. Therefore, the virtual fields $\delta\mathbf{v}$ are chosen to be a constant $\overline{\delta\mathbf{v}}$ along the boundary where $\boldsymbol{f}$ is applied. The external virtual power becomes

$$\int\limits_{\partial V_o} (\mathbf{\Pi}^T\boldsymbol{n}_o)\cdot\delta\mathbf{v}\, dS = \Big(\int\limits_{\partial V_t} \boldsymbol{t}\, dS\Big)\cdot\overline{\delta\mathbf{v}} = \boldsymbol{f}\cdot\overline{\delta\mathbf{v}} \tag{2.20}$$

Thus, given virtual velocities and known resultant loads $\boldsymbol{f}$, the Principle of Virtual Power and the cost function for identification of constitutive parameter set Φ become, respectively,

$$\int\limits_{V_o} ((\det\mathbf{F})\boldsymbol{\sigma} \cdot \mathbf{F^{-T}}) : \delta\dot{\mathbf{F}}\ \mathbf{dV} = \boldsymbol{f} \cdot \overline{\delta\mathbf{v}} \tag{2.21}$$

and

$$\Phi(\xi) = \sum_{i=1}^{N_{vf}} \sum_{j=1}^{N_{step}} \Big[\int\limits_{V_o} ((\det\mathbf{F_i})\boldsymbol{\sigma}_\mathbf{j} \cdot \mathbf{F_i^{-T}}) : \delta\mathbf{F_i}\ \mathbf{dV} - \Big(\boldsymbol{f}_\mathbf{j} \cdot \overline{\delta\mathbf{v_i}}\Big)\Big]^\mathbf{2} \tag{2.22}$$

By using a plasticity model in the cost function $\Phi(\xi)$ with measured displacements and appropriate virtual velocities, the minimization process of $\Phi(\xi)$ returns values for the unknown parameters set ξ.

Chapter 3

VFM Implementation and Evaluation

This chapter describes the numerical implementation of the VFM theory presented in Chapter 2 and the evaluation of the VFM implementation based on simulated full-field experimental data based on nominal properties of 304L stainless steel. Section 3.1 gives an example of a VFM for small-strain theory in the case of a diametrically compressed disk. Section 3.2 provides the background for the numerical methods implemented for finite-deformation VFM. Section 3.3 describes the finite element analyses for the finite deformation examples of thin sheet 304L stainless steel. Section 3.4 explains how we simulated DIC experimental data based on an experimentally derived DIC noise analysis. Section 3.5 describes how the VFM objective function for the finite deformation VFM implementation was evaluated. Section 3.6 presents the finite-deformation examples of VFM identification based on the simulated DIC data and describes the viability of this method.

3.1 Linear Elasticity Example

3.1.1 Diametrically Compressed Disk Problem Setup

A diametrically compressed disk of thickness t and radius R has a load on the edge of the disk of magnitude P, as shown in Fig. 3.1. The analytical solution for the stress fields in Cartesian coordinates is the following [14, p. 243]:

$$\sigma_1 = -\frac{P}{\pi t}\left\{\frac{(R-x_2)x_1^2}{[x_1^2+(R-x_2)^2]^2}+\frac{(R+x_2)x_1^2}{[x_1^2+(R+x_2)^2]^2}-\frac{1}{2R}\right\} \tag{3.1}$$

$$\sigma_2 = -\frac{P}{\pi t}\left\{\frac{(R-x_2)^3}{[x_1^2+(R-x_2)^2]^2}+\frac{(R+x_2)^3}{[x_1^2+(R+x_2)^2]^2}-\frac{1}{2R}\right\} \tag{3.2}$$

$$\sigma_6 = \frac{P}{\pi t}\left\{\frac{(R-x_2)^2x_1}{[x_1^2+(R-x_2)^2]^2}-\frac{(R+x_2)^2x_1}{[x_1^2+(R+x_2)^2]^2}\right\}. \tag{3.3}$$

The analytical strain fields ϵ_{ij} for a linearly elastic isotropic material can be calculated by using Eq. (2.10) and Eq. (3.1)-(3.3).

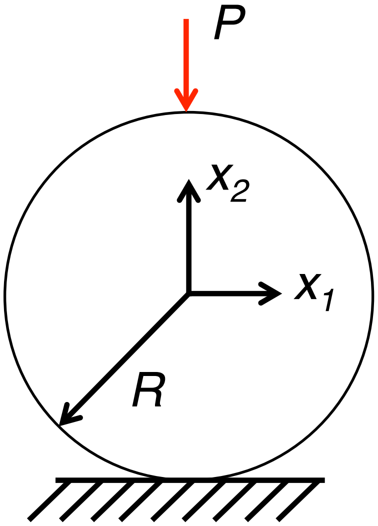


Figure 3.1: Schematic of a diametrically compressed disk

From the example given in [2, p. 61-65], the first manually chosen virtual field is

$$\begin{bmatrix} u_1^{*(1)} \\ u_2^{*(1)} \end{bmatrix} = \begin{bmatrix} 0 \\ -k_1(R+x_2) \end{bmatrix} \qquad \begin{bmatrix} \epsilon_1^{*(1)} \\ \epsilon_2^{*(1)} \\ \epsilon_6^{*(1)} \end{bmatrix} = \begin{bmatrix} 0 \\ -k_1 \\ 0 \end{bmatrix} \tag{3.4}$$

where k_1 is a nonzero constant. The virtual field $u_2^{*(1)}$ at $(x_1, x_2) = (0, -R)$ is zero, so $u_i^{*(1)}$ is KA. From Eq. (3.4), the internal virtual work $W_{int}^{*(1)}$ for this virtual field becomes

$$Q_{11} \int\limits_S \left(\epsilon_1 \epsilon_1^{*(1)} + \epsilon_2 \epsilon_2^{*(1)} + \frac{1}{2} \epsilon_6 \epsilon_6^{*(1)} \right) \mathrm{d}S + Q_{12} \int\limits_S \left(\epsilon_1 \epsilon_2^{*(1)} + \epsilon_2 \epsilon_1^{*(1)} - \frac{1}{2} \epsilon_6 \epsilon_6^{*(1)} \right) \mathrm{d}S$$

$$= -k_1 Q_{11} \int\limits_S \epsilon_2 \, \mathrm{d}S - k_1 Q_{12} \int\limits_S \epsilon_1 \, \mathrm{d}S. \tag{3.5}$$

The first manually chosen virtual field leads to a nonzero term in the $W_{ext}^{*(1)}$ term, utilizing the resultant load P applied along $(x_1, x_2) = (0, R)$ through thickness t, where $\mathbf{u_2^{*(1)}}(0, R) = -2Rk_1\hat{\mathbf{e}}_\mathbf{2}$ and the applied load is $\mathbf{F} = -P\hat{\mathbf{e}}_\mathbf{2}$:

$$\int\limits_{L_f} T_i u_i^{*(1)} \, \mathrm{d}l = \frac{1}{t} \mathbf{F} \cdot \mathbf{u_2^{*(1)}}(0, R) = \left(-\frac{P}{t} \right) (-2Rk_1) = \frac{2Pk_1}{t}. \tag{3.6}$$

Combining Eq. (3.9) and (3.6), factoring by $-k_1$, gives the following expression for the Principle of Virtual Work for the first virtual field:

$$Q_{11} \int\limits_S \epsilon_2 \, \mathrm{d}S \; + \; Q_{12} \int\limits_S \epsilon_1 \, \mathrm{d}S = \frac{-2P}{t}. \tag{3.7}$$

The second manually chosen virtual field for this problem is

$$\begin{bmatrix} u_1^{*(2)} \\ u_2^{*(2)} \end{bmatrix} = \begin{bmatrix} k_2 x_1 \\ 0 \end{bmatrix} \qquad \begin{bmatrix} \epsilon_1^{*(2)} \\ \epsilon_2^{*(2)} \\ \epsilon_6^{*(2)} \end{bmatrix} = \begin{bmatrix} k_2 \\ 0 \\ 0 \end{bmatrix} \tag{3.8}$$

where k_2 is a nonzero constant. The virtual field $u_1^{*(2)}$ at $(x_1, x_2) = (0, -R)$ is zero, so $u_i^{*(2)}$ is KA. From Eq. (3.8), the internal virtual work $W_{int}^{*(2)}$ for this virtual field becomes

$$Q_{11} \int\limits_S \left(\epsilon_1 \epsilon_1^{*(2)} + \epsilon_2 \epsilon_2^{*(2)} + \frac{1}{2}\epsilon_6 \epsilon_6^{*(2)}\right) \mathrm{d}S + Q_{12} \int\limits_S \left(\epsilon_1 \epsilon_2^{*(2)} + \epsilon_2 \epsilon_1^{*(2)} - \frac{1}{2}\epsilon_6 \epsilon_6^{*(2)}\right) \mathrm{d}S$$

$$= k_2 Q_{11} \int\limits_S \epsilon_1 \, \mathrm{d}S + k_2 Q_{12} \int\limits_S \epsilon_2 \, \mathrm{d}S. \tag{3.9}$$

The second virtual field leads to a null term for the $W_{ext}^{*(2)}$ term. Therefore, the Principle of Virtual Work for the second virtual field is

$$Q_{11} \int\limits_S \epsilon_1 \, \mathrm{d}S \; + \; Q_{12} \int\limits_S \epsilon_2 \, \mathrm{d}S = 0. \tag{3.10}$$

Combining Eq. (3.7) and (3.10) gives the following $\mathbf{AQ} = \mathbf{B}$ matrix equation:

$$\begin{bmatrix} \int\limits_S \epsilon_2 \, \mathrm{d}S & \int\limits_S \epsilon_1 \, \mathrm{d}S \\ \int\limits_S \epsilon_1 \, \mathrm{d}S & \int\limits_S \epsilon_2 \, \mathrm{d}S \end{bmatrix} \begin{bmatrix} Q_{11} \\ Q_{12} \end{bmatrix} = \begin{bmatrix} \dfrac{-2P}{t} \\ 0 \end{bmatrix} \tag{3.11}$$

where ϵ_i are the actual strain fields on the surface S_a. The strain fields are measured in full-field at N discrete points centered in small areas $S^{[q]}$, so the surface integrals in Eq. (3.11) can be approximated by the following summation:

$$\int\limits_S \epsilon_1 \, \mathrm{d}S \simeq \sum_q^N \epsilon_1^{[q]} S_a^{[q]} \quad \text{and} \quad \int\limits_S \epsilon_2 \, \mathrm{d}S \simeq \sum_q^N \epsilon_2^{[q]} S_a^{[q]} \tag{3.12}$$

where $\epsilon_i^{[q]}$ is the q-th strain measurement in the field at location $(x_1^{[q]}, x_2^{[q]})$ on S. If the small areas $S^{[q]}$ are regularly spaced and sized as S_a, then the small areas can be moved to the front of the summations. This implies that Eq. (3.11) can be written as the following, for regularly spaced strain measurements, using Eq. (3.12):

$$\begin{bmatrix} \sum\limits_q^N \epsilon_2^{[q]} & \sum\limits_q^N \epsilon_1^{[q]} \\ \sum\limits_q^N \epsilon_1^{[q]} & \sum\limits_q^N \epsilon_2^{[q]} \end{bmatrix} \begin{bmatrix} Q_{11} \\ Q_{12} \end{bmatrix} \simeq \begin{bmatrix} \dfrac{-2P}{tS_a} \\ 0 \end{bmatrix} \tag{3.13}$$

This derivation of the problem setup for VFM shows the simplicity of VFM for linear elasticity. Full-field strain measurements and load measurement at a single load step are sufficient to determine the elastic properties. Sections 3.1.2–3.1.3 describe how this the elastic problem is solved using simulated data without and with Gaussian noise.

3.1.2 Diametrically Compressed Disk Without Gaussian Noise

To simulate experimental data for this problem, a MATLAB script, `LinEDiskV02.m` was written using the analytical solution for the strain fields. The "camera CCD" imaged a nominal initial 2000 pixels × 2000 pixels, and the regular data step size was 10 pixels. The length to "camera" pixel ratio was the $2R/2000$. The disk radius was 50 mm, and the disk thickness was 2 mm. Nominal material properties for 304L stainless steel were used: E of 200 GPa and ν of 0.29. The input Q_{11} and Q_{12} are then 218 GPa and 63.3 GPa, respectively. The applied load was 7000 N, which led to average stresses ($\sim$ 12 - 44 MPa) that were below the nominal yield stress of the material, which is 170 MPa. For this analytical solution the stresses increase beyond yield near the load application points, such that the less than 2.2% of the area would have yielded. The effect of the yielded data points is ignored in this simulated data. The goal of this was to simulate a full-field strain field with strain values typical for linear elastic measurements (100 - 200 $\mu\epsilon$ on average).

Since the step size in the data is 10 pixels, then the small areas S_a had an area of 0.25 mm^2. Since this strain data is discrete and and the object is round, the discrete areas around the edge of the disk will not exactly represent the curved area of the disk, leading to error in the identification of Q_{11} and Q_{12}. The ratio of the total area used in the discrete data to the actual area of the Disk is 1.0004, so the discretized area is slightly larger than the actual disk. The MATLAB script reported values for Q_{11} and Q_{12} from the Virtual Fields Method identification process were 219 GPa and 63.2 GPa, which had errors of 0.29% and -0.21%. The identified values for E and ν were 200.8 GPa (0.38% error) and 0.289 (-0.5% error), respectively. The MATLAB script, `LinEDiskV02.m`, is given in the Appendix.

If the step size is changed, then the identified parameters Q_{11} and Q_{12}, the relative errors, and the ratio of the discretized versus actual area are given in Table 3.1. This example of simulated strain data without any noise demonstrates how VFM can simply identify linear elastic properties with small errors due to the discretization of the object.

Table 3.1: Error in identification of E and ν due to data discretization for $P = 7000$ N ($E_{ref} = 200$ GPa, and $\nu_{ref} = 0.29$)

Step Size (pixels)	Length / Pixel (mm/pixel)	S_a (mm^2)	Data Points	E (GPa)	E Error (%)	ν	ν Error (%)	Discrete / Actual Area
1	0.05	0.0025	3141676	200.1	0.04	0.2898	−0.05	1.00003
5	0.05	0.0625	125676	200.4	0.19	0.2893	−0.25	1.0001
10	0.05	0.25	31428	200.8	0.38	0.2886	−0.50	1.0004
15	0.05	0.5625	13966	200.0	0.08	0.2930	1.02	1.0002
20	0.05	1	7860	201.5	0.76	0.2872	−0.97	1.0008
25	0.05	1.5625	5024	201.9	0.95	0.2865	−1.20	0.9995
30	0.05	2.25	3493	199.1	−0.47	0.2830	−2.43	1.0007

3.1.3 Diametrically Compressed Disk With Gaussian Noise

A MATLAB script, `LinEDiskNoiseV02.m` was written to simulate Gaussian noise with an amplitude of strain similar to noise found in experimental data to see the effect on the identification of constitutive parameters. The same 10-pixel step size described in Section 3.1.2 is used here, in conjunction with several standard deviations of noise. The results are given in Table 3.2.

Table 3.2: Error in identification of E and ν due to Gaussian noise for $P = 7000$ N, for the 10-pixel step size case ($E_{ref} = 200$ GPa, and $\nu_{ref} = 0.29$)

Std. Dev. of Stain Noise ($\mu\epsilon$)	Std. Dev. Noise / ϵ_1^{ave} (%)	Std. Dev. Noise / ϵ_2^{ave} (%)	Std. Dev. Noise / ϵ_6^{ave} (%)	E (GPa)	E Error (%)	ν	ν Error (%)
0	0	0	0	200.8	0.38	0.2886	−0.50
1	1.00	0.45	0.54	200.8	0.38	0.2885	−0.50
10	9.86	4.48	5.44	200.7	0.34	0.2885	−0.51
50	49.28	22.39	27.19	201.0	0.50	0.2880	−0.068
100	98.57	44.78	54.38	200.2	0.08	0.2921	0.71
200	197.2	89.57	108.8	198.8	−0.61	0.2856	−1.52
500	492.9	224.0	271.9	203.28	1.64	0.2767	−4.57

The error in the identified parameters does not vary in a predictable manner: as the Gaussian noise increases for noise levels from 0 to 100%, the noise is about the same. The error becomes nmore significant when the Gaussian noise rise to five times greater than the average strain. The error due to the summation approximation of the integrals in the Principle of Virtual Work (Eq. (3.13)) has similar order of magnitude as the error due to Gaussian noise investigated here. This analysis implies the following:

- the manually chosen virtual fields selected here produce well conditioned matrix **A** that produce stable results, but there is not guarantee that other virtual fields will be so stable with measurement noise and data point density;
- the summation approximation to the integrals in Eq. (3.13) in addition to the uniform grid of data representing a curved volume can be large sources of error, so care must be taken in the experiment to maximize the data density and better approximate the volume shape; and
- the Poisson's ratio is more sensitive to both the summation approximation and Gaussian noise in the strain fields than the elastic modulus.

The portion of the MATLAB script, `LinEDiskNoiseV02.m`, that adds the Gaussian noise is given in the Appendix.

3.2 Finite Deformation Plasticity

The prior examples for small-strain theory utilized a uniform grid of data overlaying a non-rectangular body, which is common for experimental techniques like Digital Image Correlation (DIC) where a uniform-grid CCD camera collects the images from which the experimental data is derived. As seen in section 3.1, the identification process is quite sensitive to the summation approximation using a uniform grid. One approach to better address numerical approximation of deformation of non-rectangular volumes is to utilize FEM approaches to meshes and kinematics, which has been implemented here for finite-deformation VFM. Other advantages of FEM meshes and kinematics are efficient evaluation of the constitutive models and approaches for mapping uniform grid experimental data to a non-rectangular volumes. The following section describes this implementation of FEM-theory into the VFM identification process for finite deformations, including the plasticity material models considered for the example of 304L stainless steel, the FEM kinematics utilized in the VFM code, the material model evaluation using the FEM kinematics, and an outline of the VFM procedure.

3.2.1 Plasticity Models Under Consideration

Two constitutive models were chosen for parameter estimation. The first model is an isotropic, rate-independent, piecewise linear hardening model. The model is based on J_2 plasticity and has a hardening function of the form

$$\bar{\sigma} = \sigma_y + h\left(\bar{\varepsilon}^p\right) \tag{3.14}$$

where $\bar{\sigma}$ is the von Mises stress, σ_y is the initial yield stress, $\bar{\varepsilon}^p$ is the equivalent plastic strain, and h is the hardening function. The hardening function is represented as a piecewise linear function of the equivalent plastic strain.

The second constitutive model is the Bammann-Chiesa-Johnson (BCJ) model [15]. This model is a viscoplastic model with temperature-dependent hardening and recovery. For simplicity, the form of the model in this study has no rate or temperature dependence. In this case, the effective stress has the form

$$\bar{\sigma} = \sigma_y + \kappa \tag{3.15}$$

where σ_y is the initial yield stress, and κ is a measure of statistically stored dislocations. The evolution of κ is

$$\kappa(\bar{\varepsilon}^p) = \frac{H}{R_d}\left[1 - \exp(-R_d\bar{\varepsilon}^p)\right] \tag{3.16}$$

where H and R_d account for hardening and recovery respectively.

3.2.2 Outline of VFM Algorithm

Here are the basic steps involved with the VFM algorithm implemented in MATLAB from a displacement field that would typically come form DIC and external resultant load over the course of an "experiment" (note that the "VFM mesh" is the mesh used to perform the VFM identification and this mesh is required whether or not the input data is from an experiment or FEM simulation):

1. Input 3D nodal displacements for each time step, resultant force for each time step, nodal connectivity for all VFM-mesh elements, specimen geometry, reference nodal locations, and time steps.

   (a) If original input data comes from an FEM simulation, then the FEM solution must be mapped to the VFM mesh, either by directly sampling the FEM mesh at the isoparametric coordinates of the VFM mesh nodes, or by the following:

      i. Map FEM mesh data to uniform grid of DIC data locations by sampling the FEM mesh at the isoparametric coordinates of the DIC points, and

      ii. Add experimentally derived DIC noise to the DIC displacements, U_{DIC};

      iii. Map the uniform grid of DIC points, U_{DIC}, to the nodal locations of the VFM mesh, $u_{VFM-nodal}$, using the isoparametric coordinates of the DIC locations relative to the VFM mesh to get the relevant shape functions, Φ, solving the equation $\{U_{DIC}\} = [\Phi]\{u_{VFM-nodal}\}$ by using a least-squares minimization algorithm (see Kim et al., 2013 [16]).

   (b) If the original input data comes for an experiment with a uniform grid of DIC points, then map the uniform grid of DIC points, U_{DIC}, to the nodal locations of the VFM mesh, $u_{VFM-nodal}$, using a least-squares minimization algorithm that uses the isoparametric coordinates of the DIC locations relative to the VFM mesh to get the relevant shape functions, Φ, solving the equation $\{U_{DIC}\} = [\Phi]\{u_{VFM-nodal}\}$.

2. Define virtual velocity fields.

3. Calculate kinematics using either the uniform gradient element approach (central integration point for both the deviatoric stress and for the pressure term) or the Q1P0 element approach (8 Gauss integration points for the deviatoric stresses and central integration point for the pressure term) to be used in the calculation of Principle of Virtual Power:

   (a) Gradient operator definition;

   (b) Material gradient of displacement field at each time step;

   (c) Deformation gradient at each time step;

   (d) Inverse transpose of deformation gradient at each time step;

   (e) Rate of deformation (mid-point in time) at each time step;

   (f) Rotation matrix from polar decomposition of deformation gradient at each time step; and,

   (g) Material gradient of virtual velocity field.

4. Define material model in MATLAB for either uniform gradient or Q1P0 hexahedron elements.

5. Define bounds on identified constitutive parameters.

6. Perform VFM identification by solving a nonlinear minimization problem where the cost function that is minimized is based on the Principle of Virtual Power, using the constitutive model implemented in MATLAB with the given kinematics and the initial guess for constitutive parameters.

7. Compare identified parameters and stress history to FE-model parameters and stress history.

3.2.3 FEM Kinematics

Computing Rate of Deformation from the Deformation Gradient

Given the nodal displacement field discretized in time, $u_{iI}\left(t_n\right)$, then the increments in displacement are

$$\Delta u_{iI} = u_{iI}\left(t_{n+1}\right) - u_{iI}\left(t_n\right) \tag{3.17}$$

The deformation gradient at every time increment is

$$F_{ij}\left(t_n\right) = \delta_{ij} + \frac{1}{V} u_{iI} B_{jI} \tag{3.18}$$

We can also calculate the inverse deformation gradient at every time, $F_{ij}^{-1}\left(t_n\right)$. We want the rate of deformation, or strain increment, measured from the mid-step. We will need the deformation gradient at the midstep. This is

$$F_{ij}\left(t_{n+1/2}\right) = F_{ij}\left(t_n\right) + \frac{1}{2V} \Delta u_{iI} B_{jI} \tag{3.19}$$

The inverse is, $F_{ij}^{-1}\left(t_{n+1/2}\right)$. The midstep velocity gradient is

$$L_{ij}\left(t_{n+1/2}\right) = \frac{1}{V\Delta t} \Delta u_{iI} B_{kI} F_{kj}^{-1}\left(t_{n+1/2}\right) \tag{3.20}$$

The midstep rate of deformation is

$$D_{ij}\left(t_{n+1/2}\right) = \frac{1}{2}\left[L_{ij}\left(t_{n+1/2}\right) + L_{ji}\left(t_{n+1/2}\right)\right] \tag{3.21}$$

Objectivity

The Green-McInnis stress rate that we use in our codes is

$$\hat{\boldsymbol{\sigma}} = \dot{\boldsymbol{\sigma}} - \boldsymbol{\Omega} \cdot \boldsymbol{\sigma} + \boldsymbol{\sigma} \cdot \boldsymbol{\Omega} \tag{3.22}$$

where $\boldsymbol{\sigma}$ is the Cauchy stress tensor and $\boldsymbol{\Omega}$ is the rate of the rotation tensor from the polar decomposition. We numerically implement this stress rate by integrating the constitutive model in an "un-rotated" configuration. To do this we will need to keep *two* rotations around: the rotation at t_n and t_{n+1}. Since the polar decomposition is

$$\mathbf{F} = \mathbf{R} \cdot \mathbf{U} = \mathbf{V} \cdot \mathbf{R} \tag{3.23}$$

where $\mathbf{R}$ is the rotation, $\mathbf{U}$ is the right stretch and $\mathbf{V}$ is the left stretch. We have at t_n and t_{n+1}

$$\mathbf{F}_n = \mathbf{R}_n \cdot \mathbf{U}_n = \mathbf{V}_n \cdot \mathbf{R}_n \tag{3.24}$$

$$\mathbf{F}_{n+1} = \mathbf{R}_{n+1} \cdot \mathbf{U}_{n+1} = \mathbf{V}_{n+1} \cdot \mathbf{R}_{n+1} \tag{3.25}$$

The algorithm that the Sierra FEM codes use is the following:

First, we have the Cauchy stress, $\boldsymbol{\sigma}_n$, and the un-rotated Cauchy stress, $\mathbf{T}_n$, at time t_n

$$\mathbf{T}_n = \mathbf{R}_n^T \cdot \boldsymbol{\sigma}_n \cdot \mathbf{R}_n \tag{3.26}$$

Next, we calculate the rate of deformation, $\mathbf{D}$, and un-rotate it with the rotation at time t_{n+1} to give us $\mathbf{d}$

$$\mathbf{d} = \mathbf{R}_{n+1}^T \cdot \mathbf{D} \cdot \mathbf{R}_{n+1} \tag{3.27}$$

Send this un-rotated rate of deformation to the constitutive model and update the un-rotated stress to get $\mathbf{T}_{n+1}$

$$\mathbf{T}_{n+1} = \mathbf{T}_n + \Delta\mathbf{T}\left(\mathbf{d}\right) \tag{3.28}$$

This is essentially what the role of the constitutive model is. The stress is passed to the constitutive model at time t_n - really $\mathbf{T}_n$ - and the “strain rate” - really $\mathbf{d}$ - and it returns the stress at the end of the time step - $\mathbf{T}_{n+1}$. We could just as easily write the above equation as a function

$$\mathbf{T}_{n+1} = \mathbf{f}\left(\mathbf{T}_n, \mathbf{d}\right) \tag{3.29}$$

Of course we also have state variables, which we have ignored. If we have an array of state variables, $\boldsymbol{\xi}$, we can write

$$\left[\mathbf{T}_{n+1}, \boldsymbol{\xi}_{n+1}\right] = \mathbf{f}\left(\mathbf{T}_n, \boldsymbol{\xi}_n, \mathbf{d}\right) \tag{3.30}$$

For example, our model only has one state variable, the equivalent plastic strain. It is a scalar variable, so we can write the above as

$$\left[\mathbf{T}_{n+1}, \bar{\varepsilon}^p_{n+1}\right] = \mathbf{f}\left(\mathbf{T}_n, \bar{\varepsilon}^p_n, \mathbf{d}\right) \tag{3.31}$$

Finally, rotate the un-rotated stress at time t_{n+1} to the Cauchy stress at time t_{n+1} using the rotation at time t_{n+1}

$$\boldsymbol{\sigma}_{n+1} = \mathbf{R}_{n+1} \cdot \mathbf{T}_{n+1} \cdot \mathbf{R}_{n+1}^T \tag{3.32}$$

3.2.4 Uniform Gradient FEM Models

The uniform gradient (UG) element is based on the formulation developed by Flanagan and Belytschko [17]. The main assumption of the UG element is that the stress is assumed constant over the element. Using this in the internal virtual work results in the derivation of a gradient operator that is volume averaged over the element. The interested reader is directed to Flanagan and Belytschko [17].

The main result is the discrete gradient operator for the UG element. The discrete gradient operator, B_{iI}, is

$$\frac{\partial g}{\partial x_i} = \frac{1}{V}\sum_{I=1}^{8} B_{iI} g_I \tag{3.33}$$

where x_i is the coordinate, V is the volume of the element, g is some field (scalar, vector, or tensor) and, g_I is the field evaluated (or calculated) at node I. The expression for B_{iI} is

$$B_{iI} = \int\limits_V \frac{\partial \phi_I}{\partial x_i} dV = \begin{Bmatrix} y_J z_K \\ z_J x_K \\ x_J y_K \end{Bmatrix} C_{IJK} \tag{3.34}$$

where ϕ_I is the linear element shape function

$$\phi_I = \frac{1}{8}\Sigma_I + \frac{1}{4}\left(\xi_1\Lambda_{1I} + \xi_2\Lambda_{2I} + \xi_3\Lambda_{3I}\right) + \frac{1}{2}\left(\xi_2\xi_3\Gamma_{1I} + \xi_3\xi_1\Gamma_{2I} + \xi_1\xi_2\Gamma_{3I}\right) + \xi_1\xi_2\xi_3\Gamma_{4I} \tag{3.35}$$

and

$$C_{IJK} = \frac{1}{192}\varepsilon_{ijk}\left(3\Lambda_{iI}\Lambda_{jJ}\Lambda_{kK} + \Lambda_{iI}\Gamma_{kJ}\Gamma_{jK} + \Gamma_{kI}\Lambda_{jJ}\Gamma_{iK} + \Gamma_{jI}\Gamma_{iJ}\Lambda_{kK}\right) \tag{3.36}$$

where i, j, k range from 1 to 3. The Λ_{iI} and Γ_{iI} are the element base vectors given in Table 3.3.

Using this gradient operator, the volume average displacement gradient for the UG element is

Table 3.3: Element Base Vectors for Uniform Gradient Elements

Node	ξ_1	ξ_2	ξ_3	Σ_I	Λ_{1I}	Λ_{2I}	Λ_{3I}	Γ_{1I}	Γ_{2I}	Γ_{3I}	Γ_{4I}
1	-1/2	-1/2	-1/2	1	-1	-1	-1	1	1	1	-1
2	1/2	-1/2	-1/2	1	1	-1	-1	1	-1	-1	1
3	1/2	1/2	-1/2	1	1	1	-1	-1	-1	1	-1
4	-1/2	1/2	-1/2	1	-1	1	-1	-1	1	-1	1
5	-1/2	-1/2	1/2	1	-1	-1	1	-1	-1	1	1
6	1/2	-1/2	1/2	1	1	-1	1	-1	1	-1	-1
7	1/2	1/2	1/2	1	1	1	1	1	1	1	1
8	-1/2	1/2	1/2	1	-1	1	1	1	-1	-1	-1

$$\bar{u}_{i,j} = \frac{1}{V}\int\limits_V \frac{\partial u_i}{\partial x_j} dV = \frac{1}{V}\sum_{I=1}^{8} u_{iI} B_{jI} \tag{3.37}$$

The internal force at each node, f_{iI}, is calculated as

$$f_{iI}\delta u_{iI} = \int\limits_V \sigma_{ij}\delta u_{i,j} dV = V\bar{\sigma}_{ij}\delta\bar{u}_{i,j} \quad \rightarrow \quad f_{iI} = \bar{\sigma}_{ij}B_{jI} \tag{3.38}$$

The volume averaged stress, $\bar{\sigma}_{ij}$ is computed from the constitutive model using the strain rate calculated from the volume averaged incremental displacement gradient.

One major problem with this formulation is that there are zero-energy modes. These are modes of deformation where the volume average displacement gradient is zero, which gives no stress and no resistance to the deformation. Zero-energy modes, or hourglass modes, are controlled with some method of hourglass control. This method adds some artificial energy to the problem which has some ramifications for VFM, as will be discussed in section 3.6.1. For more detail on hourglass control see Flanagan and Belytschko [17].

3.2.5 Q1P0 FEM Models

The Q1P0 finite element models are based on a fully integrated hexahedron with $2 \times 2 \times 2$ Gauss quadrature [18–20]. The fully integrated element is known to lock since it is over-constrained. This constraint is relaxed by integrating the pressure with a single integration point and fully integrating the deviatoric response. An slight modification to the Q1P0 element is to volume average the pressure response over the entire element rather than evaluate it at the integration point at $\xi_i = 0$. This second approach is also common in finite element codes, but it is not what we chose to implement.

For the Q1P0 element the shape functions are the same - linear shape functions for a hexahedron. The internal force for a fully integrated element is calculated from the internal virtual work

$$
\begin{gathered}
f_{iI}\delta u_{iI} = \int\limits_V \sigma_{ij}\delta u_{i,j} dV = \int\limits_V \sigma_{ij}\frac{\partial \phi_I}{\partial x_j}\delta u_{iI} dV = \delta u_{iI}\int\limits_V \sigma_{ij}\frac{\partial \phi_I}{\partial x_j} dV \\
f_{iI} = \int\limits_V \sigma_{ij}\frac{\partial \phi_I}{\partial x_j} dV
\end{gathered}
\tag{3.39}
$$

As stated earlier, this element formulation is over-constrained. To ease the constrain the pressure is integrated at the center of the element. Decomposing the stress into a pressure and a deviatoric stress

$$
\sigma_{ij} = s_{ij} + p\delta_{ij} \quad ; \quad p = \frac{1}{3}\sigma_{kk} \tag{3.40}
$$

we can substitute this into the expression for the internal virtual work and get

$$
f_{iI} = \int\limits_V s_{ij}\frac{\partial \phi_I}{\partial x_j} dV + V p_0 \left(\frac{\partial \phi_I}{\partial x_j}\right)_0 \tag{3.41}
$$

where $(\cdot)_0$ denotes a quantity evaluated in the element at the isoparametric coordinates $\xi_i = 0$.

The first integral is evaluated in the standard way

$$
\int\limits_V s_{ij}\frac{\partial \phi_I}{\partial x_j} dV = V\sum_{k=1}^{n_{ip}} s_{ij}^{(k)}\frac{\partial \phi_I^{(k)}}{\partial x_j} w^{(k)} \tag{3.42}
$$

where n_{ip} is the number of integration points (8 in our case), k is the integration point number, and $w^{(k)}$ is the weight of the integration point.

3.2.6 Material Model Evaluation

Stress State

In plasticity theory, the plastic, or permanent, deformation does not depend on the hydrostatic pressure. Therefore, it is advantageous to formulate a plasticity model using the deviatoric stress. Given the Cauchy stress, $\boldsymbol{\sigma}$, the deviatoric stress is

$$\mathbf{s} = \boldsymbol{\sigma} - \frac{1}{3}\mathrm{tr}\left(\boldsymbol{\sigma}\right)\mathbf{I} \tag{3.43}$$

Some general statement about the stress state can then be made by looking at the magnitude of the deviatoric stress. This basically amounts to replacing the tensorial information contained in the deviatoric stress with a scalar. The magnitude of the deviatoric stress is

$$\|\mathbf{s}\| = \sqrt{\mathbf{s} : \mathbf{s}} \tag{3.44}$$

Rather than looking at the magnitude of the deviatoric stress, it is sometimes easier to use the von Mises stress. One big advantage of the von Mises stress is that it is equal to the stress in a uniaxial stress problem. For multi-axial stress states this allows us to equate, in some way, a complex stress state with an "equivalent" uniaxial stress state. Of course looking at this "equivalent" stress state may or may not be justified; we just don't know enough about the behavior of materials under multi-axial stress states.[1]

The von Mises stress is

$$\bar{\sigma} = \sqrt{\frac{3}{2}\mathbf{s} : \mathbf{s}} \tag{3.45}$$

so

$$\bar{\sigma} = \sqrt{\frac{3}{2}}\|\mathbf{s}\| \tag{3.46}$$

Yield Surface

Plasticity models define a region in stress space that is bounded by something called the *yield surface.* The idea of a yield surface is in many ways quite simple: it bounds a region of

[1] One should not confuse *knowing* how material models behave with *knowing* how materials behave.

allowable stress states. So in plasticity, the stress state is restricted to lie inside or on the yield surface. Furthermore, the response of the material inside the yield surface is elastic.

If a stress state is on or inside the yield surface, and an elastic load increment results in a stress state that is also inside the yield surface, then the response is elastic. If a stress state is on or inside the yield surface and an elastic load increment results in a stress state outside of the yield surface, then plastic deformation has occurred. The description of the yield surface is very important for plasticity models. For isotropic materials the yield surface is adequately described by its radius, R. The radius of the yield surface is related to the magnitude of the deviatoric stress. In particular, the stress state must satisfy

$$\|\mathbf{s}\| \leq R \tag{3.47}$$

Hardening Law

The hardening law for a plasticity model is, in general, written as

$$\bar{\sigma} = \sigma_y + h\left(\bar{\varepsilon}^p\right) \tag{3.48}$$

where $h\left(\bar{\varepsilon}^p\right)$ is the hardening function that is a function of the equivalent plastic strain, $\bar{\varepsilon}^p$. This function can be linear, nonlinear or piecewise linear. Equating this to the radius of the yield surface we have

$$R = \sqrt{\frac{2}{3}}\left[\sigma_y + h\left(\bar{\varepsilon}^p\right)\right] \tag{3.49}$$

We can take the time derivative of each side and generate something more useful in what follows

$$\dot{R} = \sqrt{\frac{2}{3}}\frac{d(h\left(\bar{\varepsilon}^p\right))}{d\bar{\varepsilon}^p}\dot{\bar{\varepsilon}}^p \tag{3.50}$$

Stress Update

For updating the stress given a strain increment $Delta\,\boldsymbol{\varepsilon}$, we start with a trial elastic stress state

$$\boldsymbol{\sigma}^{tr} = \boldsymbol{\sigma}_n + \mathbb{C}^e : \Delta\,\boldsymbol{\varepsilon} \tag{3.51}$$

The fourth order tensor $\mathbb{C}$ is the elasticity tensor

$$\mathbb{C} = \lambda\,\mathbf{I} \otimes \mathbf{I} + 2\mu\,\mathbb{I} \tag{3.52}$$

If the trial stress state is elastic, i.e. $\|\mathbf{s}^{tr}\| < R_n$, then $\Delta\,\boldsymbol{\varepsilon}^e = \Delta\,\boldsymbol{\varepsilon}$ and $\boldsymbol{\sigma}_{n+1} = \boldsymbol{\sigma}^{tr}$ and we are done. If the trial stress state is outside of the yield surface, i.e. $\|\mathbf{s}^{tr}\| > R_n$, then we need to solve for the plastic strain increment, $\Delta\,\boldsymbol{\varepsilon}^p$. Solving for this will give us the final stress state

$$\boldsymbol{\sigma}_{n+1} = \boldsymbol{\sigma}^{tr} - \mathbb{C}^e : \Delta\,\boldsymbol{\varepsilon}^p \tag{3.53}$$

Flow Rule

The plastic strain increment is found from the *flow rule.* The flow rule gives us a rule for how the plastic strain evolves. For metal plasticity, the material response is independent of hydrostatic pressure, and the flow rule reflects this. The plastic strain rate is in a *direction* normal to the yield surface.

$$\Delta\,\boldsymbol{\varepsilon}^p : \mathbf{I} = 0 \tag{3.54}$$

The above equation can be written in terms of the deviatoric stress

$$\mathbf{s}_{n+1} = \mathbf{s}^{tr} - 2\mu\Delta\,\boldsymbol{\varepsilon}^p \tag{3.55}$$

Returning radially to the yield surface we have

$$\Delta\,\boldsymbol{\varepsilon}^p = \Delta\gamma\mathbf{N} \quad ; \quad \mathbf{N} = \frac{\mathbf{s}^{tr}}{\|\mathbf{s}^{tr}\|} \tag{3.56}$$

where the expression for the normal comes from a backward Euler algorithm. This gives us

$$\mathbf{s}_{n+1} = \left(\|\mathbf{s}^{tr}\| - 2\mu\Delta\gamma\right)\mathbf{N} \tag{3.57}$$

The radius of the yield surface at the end of the step is

$$R_{n+1}^2 = \mathbf{s}_{n+1} : \mathbf{s}_{n+1} = \left(\|\mathbf{s}^{tr}\| - 2\mu\Delta\gamma\right)^2 \tag{3.58}$$

or

$$R_{n+1} = \|\mathbf{s}^{tr}\| - 2\mu\Delta\gamma \tag{3.59}$$

The radius of the yield surface at the end of the time step is also given by

$$\begin{aligned} R_{n+1} &= \sqrt{\frac{2}{3}}\left[\sigma_y + h\left(\bar{\varepsilon}^p_{n+1}\right)\right] \\ &= R_n + \sqrt{\frac{2}{3}}\left[h\left(\bar{\varepsilon}^p_n + \Delta\,\bar{\varepsilon}^p\right) - h\left(\bar{\varepsilon}^p_n\right)\right] \end{aligned} \tag{3.60}$$

For linear hardening this is simply

$$R_{n+1} = R_n + \sqrt{\frac{2}{3}} H \Delta\,\bar{\varepsilon}^p \tag{3.61}$$

For isotropic hardening the equivalent plastic strain is

$$\bar{\varepsilon}^p = \int\limits_0^t \sqrt{\frac{2}{3} \dot{\boldsymbol{\varepsilon}}^p : \dot{\boldsymbol{\varepsilon}}^p dt} \tag{3.62}$$

which can be simplified to

$$\bar{\varepsilon}^p = \sqrt{\frac{2}{3}} \int\limits_0^t \gamma dt \tag{3.63}$$

so

$$\dot{\bar{\varepsilon}}^p = \sqrt{\frac{2}{3}} \gamma \quad ; \quad \Delta\,\bar{\varepsilon}^p = \sqrt{\frac{2}{3}} \Delta\gamma \tag{3.64}$$

which gives us

$$R_{n+1} = R_n + \frac{2}{3} H \Delta\,\gamma \tag{3.65}$$

This finally gives us, for linear hardening

$$R_{n+1} = \|\mathbf{s}^{tr}\| - 2\mu\Delta\gamma = R_n + \frac{2}{3} H \Delta\,\gamma \tag{3.66}$$

or

$$\Delta\gamma = \frac{3}{2} \frac{\|\mathbf{s}^{tr}\| - R_n}{3\mu + H} \tag{3.67}$$

For nonlinear hardening, a Newton-Raphson algorithm is used. For this algorithm we start with

$$\|\mathbf{s}^{tr}\| - 2\mu\Delta\gamma = R_n + \sqrt{\frac{2}{3}} \left[h\left(\bar{\varepsilon}^p_n + \sqrt{\frac{2}{3}} \Delta\gamma \right) - h\left(\bar{\varepsilon}^p_n\right) \right] \tag{3.68}$$

More simply, by defining $\Delta\,h\,(\Delta\,\gamma)$

$$\Delta\,h\,(\Delta\,\gamma) = h\left(\bar{\varepsilon}^p_n + \sqrt{\frac{2}{3}} \Delta\gamma \right) - h\left(\bar{\varepsilon}^p_n\right) \tag{3.69}$$

we have

$$\|\mathbf{s}^{tr}\| - 2\mu\Delta\,\gamma = R_n + \sqrt{\frac{2}{3}}\Delta\, h\,(\Delta\,\gamma) \tag{3.70}$$

For the Newton-Raphson algorithm we define $\phi\,(\Delta\,\gamma)$ as

$$\phi\,(\Delta\,\gamma) = \|\mathbf{s}^{tr}\| - R_n - 2\mu\Delta\,\gamma - \sqrt{\frac{2}{3}}\Delta\, h\,(\Delta\,\gamma) \tag{3.71}$$

and solve $\phi\,(\Delta\,\gamma) = 0$. The derivative of ϕ is

$$\phi'\,(\Delta\,\gamma) = -2\mu - \frac{2}{3}\frac{dh}{d\bar{\varepsilon}^p}\,(\Delta\,\gamma) \tag{3.72}$$

The value for $\Delta\,\gamma$ in the algorithm is

$$\Delta\,\gamma^{(k+1)} = \Delta\,\gamma^{(k)} - \frac{\phi}{\phi'} \tag{3.73}$$

For nonlinear hardening, ϕ and ϕ' are functions of $\Delta\,\gamma$, and we must iterate for a solution. For linear hardening, as shown above, we have

$$\phi = \|\mathbf{s}^{tr}\| - R_n - \left(2\mu + \frac{2}{3}H\right)\Delta\,\gamma \quad ; \quad \phi' = -2\mu - \frac{2}{3}H \tag{3.74}$$

which gives us

$$\Delta\gamma = \frac{3}{2}\frac{\|\mathbf{s}^{tr}\| - R_n}{3\mu + H} \tag{3.75}$$

in one step. So what we need is an algorithm to solve for $\Delta\,\gamma$ and then use that value to update the equivalent plastic strain

$$\bar{\varepsilon}^p_{n+1} = \bar{\varepsilon}^p_n + \sqrt{\frac{2}{3}}\Delta\,\gamma \tag{3.76}$$

3.3 Model Experiments and FEM Simulations for Plasticity Examples

3.3.1 Uniform Gradient FEM Models

Two plate configurations were examined: a notched tensile plate and a T-shape. Each configuration had two different geometries. The analyses were run with uniform gradient elements. Three constitutive models are chosen for parameter estimation. The two models are an isotropic, rate-independent, linear hardening model and an isotropic, rate-independent, *piecewise* linear hardening model called MLEP. The models are based on J_2 plasticity and has a yield function of the form

$$\bar{\sigma} = \sigma_y + h\left(\bar{\varepsilon}^p\right) \tag{3.77}$$

where $\bar{\sigma}$ is the von Mises stress, σ_y is the initial yield stress, $\bar{\varepsilon}^p$ is the equivalent plastic strain, and $h\left(\bar{\varepsilon}^p\right)$ is the hardening function. The hardening function are represented as a linear or a piecewise linear function of the equivalent plastic strain, respectively. The finite element analyses for the MLEP model were the same as those of the linear hardening model, but we defined the hardening curve as four segments with the same $\Delta\sigma_N$ and $\Delta\epsilon_N$; the purpose of the searching for the MLEP segments is to determine how well VFM can find values for piecewise hardening model.

The third constitutive model is the Bammann-Chiesa-Johnson (BCJ) model [15]. This model is a viscoplastic model with temperature-dependent hardening and recovery. For simplicity, the form of the model in this study has no rate- or temperature-dependence. In this case, the effective stress has the form

$$\bar{\sigma} = \sigma_y + \kappa \tag{3.78}$$

where σ_y is the initial yield stress, and κ is a measure of statistically stored dislocations. The evolution of κ is

$$\kappa(\bar{\varepsilon}^p) = \frac{H}{R_d}\big[1 - \exp(-R_d\bar{\varepsilon}^p)\big] \tag{3.79}$$

where H and R_d account for hardening and recovery respectively. The only real difference between the two models is the recovery term, R, in the BCJ model. The material properties are in Table 3.4

Table 3.4: Material properties

$E = 200$ GPa	$H = 2.28$ GPa
$\nu = 0.249$	$R = 1.1$
$\sigma_y = 193$ MPa	

3.3.2 Notched Tensile Configuration

Two geometries were studied for the notched tensile configuration: one with a deep notch and one with a shallow notch. Figure 3.2 shows the notched tensile geometry and defined the length variables for the model. Geometry 1 was $\{W, H, h, d, d_c\} = \{6.5, 4.0, 0.5, 1.2, 0.0\}$ mm, and Geometry 2 was $\{W, H, h, d, d_c\} = \{6.5, 4.0, 0.5, 1.2, 0.6\}$ mm. The mesh for these models were 5-elements thick with around 25,000 elements.

The boundary conditions were prescribed displacement on the top ($+y$) and bottom ($-y$) surfaces. On the top surface the x and z displacements were fixed and a prescribed y-displacement of 0.2 mm was applied. On the bottom surface the x, y, and z displacements were all fixed. The analysis was run in 100 equal time steps. Each time step applied an incremental y displacement of 0.002 mm to the top surface.

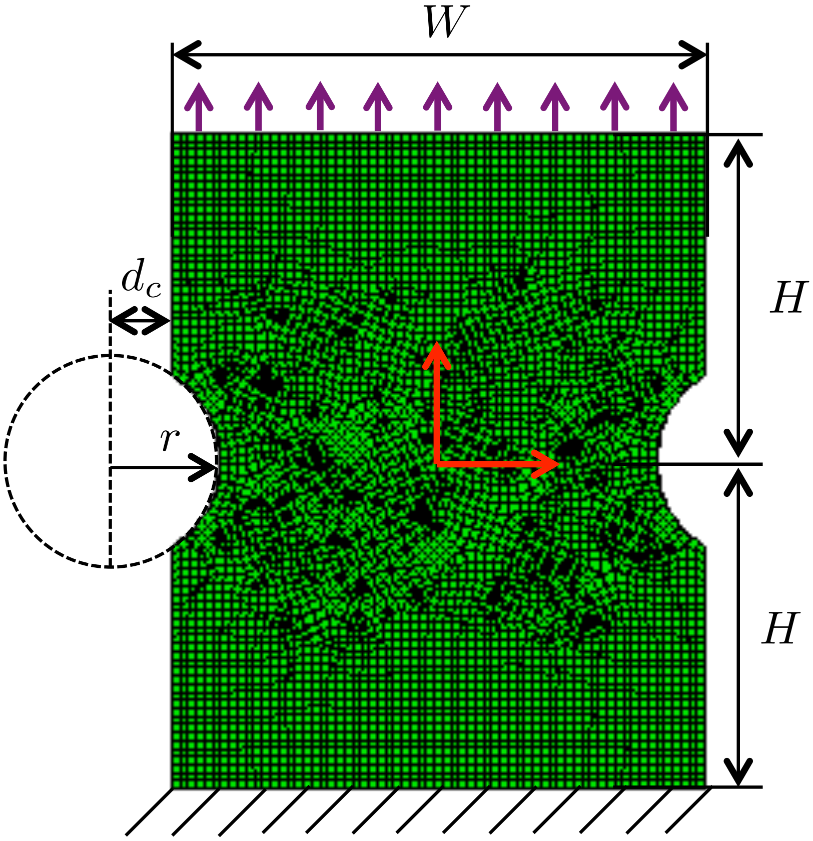


Figure 3.2: Notched tensile configuration for the uniform gradient element finite element analyses.

The maximum equivalent plastic strains, which can be used as an indicator of how much plastic deformation occurs in each problem, depended on the geometry and the constitutive models. For the shallow notched geometry with the linear hardening plasticity model, $\bar{\varepsilon}^p_{max} = 0.08373$, while for the BCJ model $\bar{\varepsilon}^p_{max} = 0.08485$. The BCJ model showed slightly more plastic deformation which is due to the recovery term in the model. For the deep notched geometry the plastic deformation was greater. For the linear hardening model $\bar{\varepsilon}^p_{max} = 0.09786$, while for the BCJ model $\bar{\varepsilon}^p_{max} = 0.09935$. Once again the BCJ model showed slightly more plastic deformation than the linear hardening model.

3.3.3 T-Shape Configuration

Two geometries were studied for the T-shape configuration, schematically shown in Figure 3.3: one with a slightly thicker section on the T-shape configuration and one with a much thicker configuration on the T-shape specimen. The length variables for the model defined in Figure 3.3 are as follows: $\{L, H, t, L_1, L_2, H_1, H_2\}$ is $\{1, 1, 0.05, 0.3, 0.4, 0.3, 0.2\}$ mm for Geometry 1 and $\{1, 1, 0.05, 0.3, 0.4, 0.6, 0.2\}$ mm for Geometry 2.

The boundary conditions apply a y displacement of 0.02 mm on the top surface with the

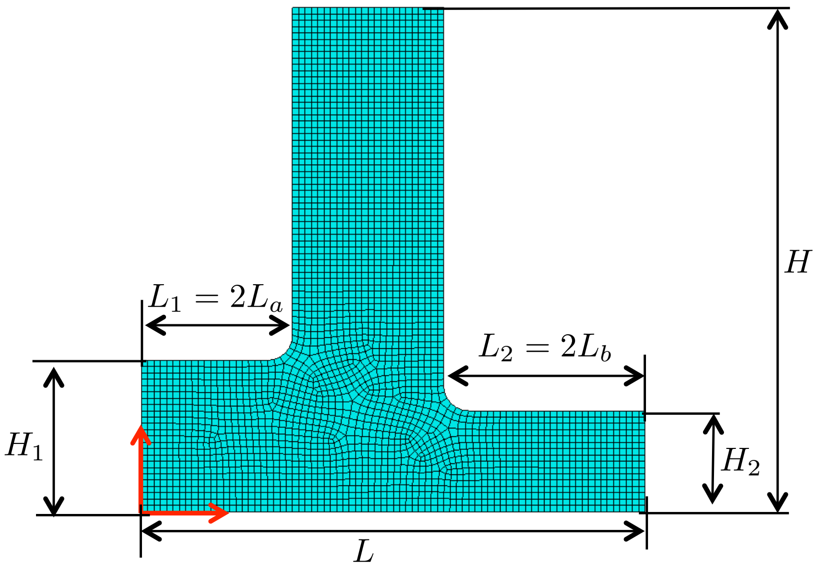


Figure 3.3: T-shape configuration with thickness t for the uniform gradient element finite element analyses.

two side surfaces fixed in the x, y, and z directions. The analysis was run in 100 equal time steps with each time step applying an incremental y displacement of 0.0002 mm to the top surface. The mesh had 5 elements through the thickness with around 12,000 elements.

As before, looking at the maximum equivalent plastic strains we see that the results depend on the geometry and the material model. For the first specimen with linear hardening $\bar{\varepsilon}^p_{max} = 0.07520$ while for the BCJ model $\bar{\varepsilon}^p_{max} = 0.07580$. Once again the BCJ model shows slightly more plastic deformation. For the first specimen with linear hardening $\bar{\varepsilon}^p_{max} = 0.04273$ while for the BCJ model $\bar{\varepsilon}^p_{max} = 0.04278$. Here the BCJ model shows slightly more plastic deformation, again, but both models show less plastic deformation than in the first geometry. This is because there is simply less deformation, specifically bending deformation, in the thicker specimen.

3.3.4 Q1P0 FEM Models

Given the uncertainties when trying to deal with the hourglass energy in the uniform gradient finite element analyses, as will be described in section 3.6.1, we decided to concentrate modeling on our most likely candidate for a geometry - the notched tensile specimen - with a better element formulation, the Q1P0 element. The generic notched tensile configuration is shown in Figure 3.4, allowing for offset notches on the left and right of the specimen. Initially ten geometries, labeled A though I, were examined, where Case A is similar to the geometry from the uniform gradient analyses in that the notches are symmetric, and where the other geometries led to asymmetry in the displacement fields. The values for the geometry variables are given in Table 3.5. Each geometry had $W = 66.96$ mm, $H = 80$ mm , and $t = 1.524$ mm, which were more realistic lengths for an actual physical experiment as compared to the analyses with the uniform gradient. From those ten geometries, we chose three geometries to concentrate our evaluation of the VFM-identification process, as will be described in section 3.6.2. All geometries were run for two different mesh densities, denoted coarse and fine. Both meshes were 1-element thick; the coarse meshes had an aspect ratio of 1.25 in-plane to 1 out-of-plane, and the fine meshes had an aspect ratio of 0.63 in-plane to 1 out-of-plane. These mesh sizes led to models with around 1400 elements for the coarse meshes and 5900 elements for the fine meshes. Figure 3.5 shows the coarse and fine meshes for Case I.

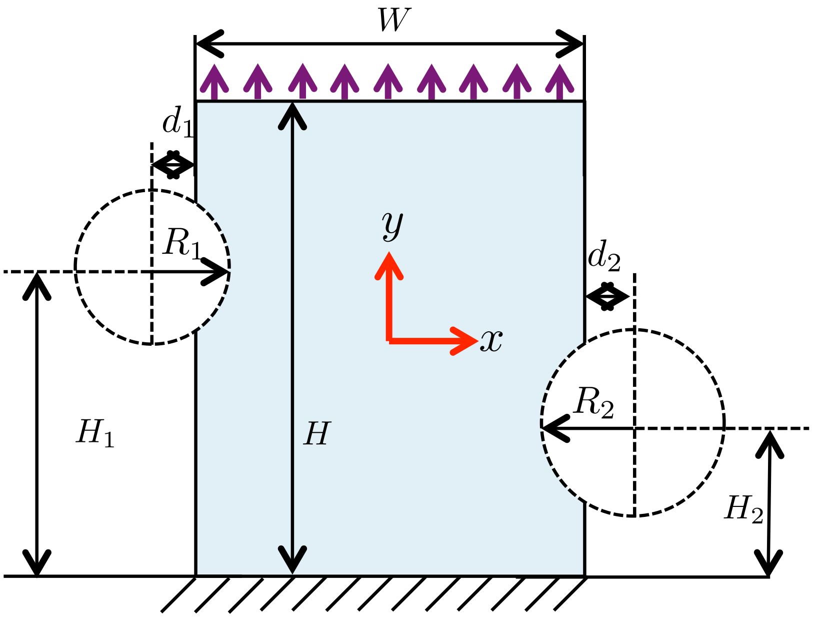
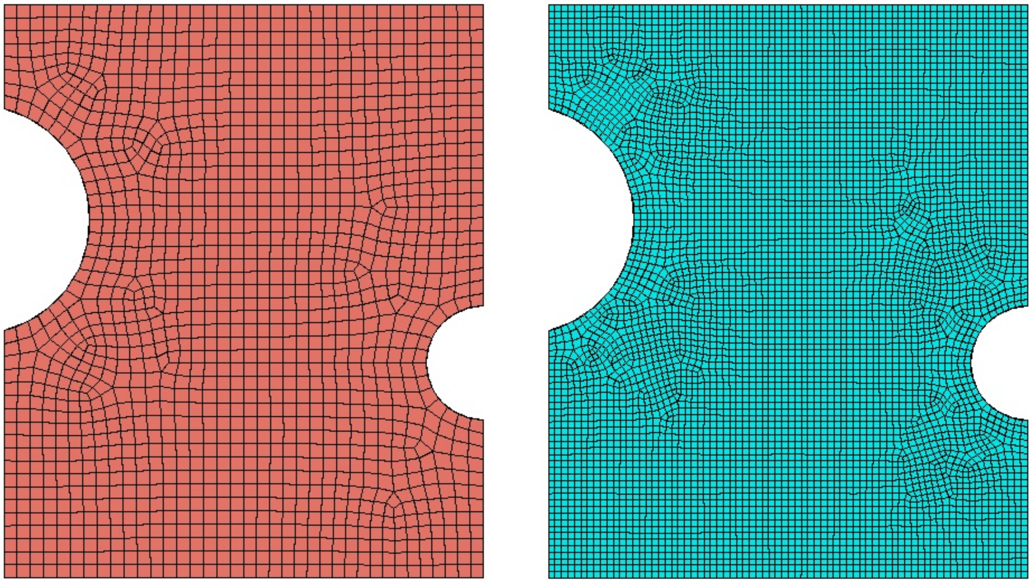


Figure 3.4: Q1P0 notched tensile specimen geometries.

Table 3.5: Geometry Variation for Q1P0 Notched Tension Specimens (All Units in mm)

Case	H_1	H_2	R_1	R_2	d_1	d_2
A	40	40	16	16	0	0
B	50	30	16	16	0	0
C	40	40	16	8	0	0
D	40	40	16	16	4	0
E	40	40	16	16	8	0
F	50	30	16	8	0	0
G	50	30	16	16	4	0
H	50	30	16	16	8	0
I	50	30	16	8	4	0
J	50	30	16	8	8	0

The BCJ model using nominal 304L stainless steel properties in Table 3.4 were used for all analyses with the Q1P0 elements. Each model was loaded in tension in the y direction with



(a) Coarse

(b) Fine

Figure 3.5: Meshes for Q1P0 notched tensile geometry Case I.

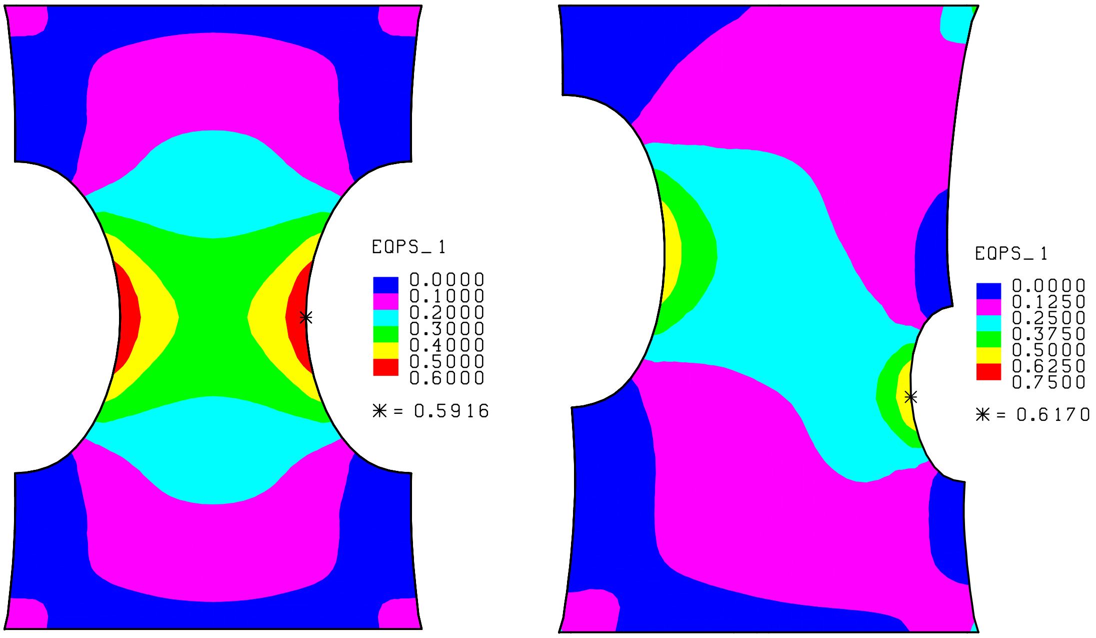
a prescribed displacement of 20 mm, while on the bottom surface the y displacement was fixed. This gives a"nominal engineering strain" of 0.223 for the entire specimen. The x and z directions on the boundaries were free to contract.

The three geometries that were chosen were based partly on the maximum equivalent plastic strain seen in the analyses since this is a measure of how much plastic deformation can be expected in each test. The maximum equivalent plastic strains for each integration point can be seen in Table 3.6. Figure 3.6 shows the equivalent plastic strain maps at the end of the simulation for Cases A and F. Case A has heterogeneous deformation with quarter symmetry, while Case F lacks symmetry in its heterogeneous deformation, though the largest deformation is still concentrated in the center of the model. It should be noted that the equivalent plastic strain, a constitutive model state variable, exists for each integration point in the Q1P0 element, including the center of the element. This results in nine equivalent plastic strains per element. Integration point 0 in Table 3.6 is the integration point in the center of the element, the other 8 integration points are the Gauss quadrature points.

In general, $\bar{\varepsilon}^p_{max}$ for each integration point does not always occur in the same element. In fact, only for case I in Table 3.6 is it the case that the maximum equivalent plastic strain for each integration point occurs in the same element. This fact, however, does not eliminate this state variable as a measure for plastic deformation in a model. It is also worth noting that the equivalent plastic strains were significantly higher in the notched specimens than the"nominal" applied strain of 0.223.

Table 3.6: Maximum equivalent plastic strains in the notched specimen runs with the Q1P0 element for each integration point.

integration point	A	F	I
0	0.5918	0.6209	0.6259
1	0.6034	0.6517	0.6569
2	0.6030	0.6502	0.6553
3	0.6043	0.6073	0.5961
4	0.6050	0.5967	0.6017
5	0.6034	0.6517	0.6569
6	0.6030	0.6502	0.6553
7	0.6043	0.6073	0.5961
8	0.6050	0.5967	0.6017



(a) Case A

(b) Case F

Figure 3.6: Equivalent plastic strain field at the end of the simulation for Cases A and F for the Q1P0 examples.

3.4 Simulated Data Without and With Experimentally Derived Noise

The simulated DIC data used in the VFM identification process is generated through a superposition of experimentally derived uncertainty of position and displacement measurements onto a FEM solution. The DIC errors come from stereo-DIC lab setups of a representative field-of-view (FOV), here approximately 100-mm along one side for a 2448×2048 pixel CCD cameras with 75-mm lenses. The uncertainty of the calibration, position, and displacements are calculated from a combination of experimental images, Correlated Solutions' VIC3D software, and Monte Carlo runs of the DIC analysis process as described in [21]. A large calibration set of images are used to quantify the calibration uncertainties, and the matching error between the two cameras is the error reported by VIC3D software in the calibration. The position error are the standard deviation of the X, Y, and Z locations at each subset of the image. Typically, the positional errors are smaller in the center of the FOV (around 0.0005-mm in X and Y and 0.002-mm in Z) and larger at the edges. The displacement errors are determined using a Monte Carlo approach of images between two positions of the speckle pattern, which was translated approximately 0.5-mm in X, Y, and Z. The displacement errors are the standard deviations of these Monte Carlo runs.

A uniform grid of potential DIC data points are overlaid on a FEM solution at a data density typical of DIC setups (for example, a step size of 12 pixels for a subset size of 25 pixels.) DIC points are located at the center of the subsets, and the entirety of acceptable subsets must lie inside the speckle pattern on the specimen. To simulate this, each potential DIC subset is tested to determine if it is all on the surface of the FEM mesh, and all unacceptable subsets are eliminated. Fig. 3.7 shows the front surface of a FEM mesh of an example notched tension specimen and a closeup of the acceptable DIC data points.The FEM displacement solution is sampled at the acceptable simulated DIC subset center points using standard linear shape functions of the FEM elements. Simulated error in u, v, and w at each DIC data point is taken as a random sample from a Gaussian distribution with standard deviation equal to the experimentally measured uncertainties σ_u, σ_v, and σ_w at the corresponding pixel location on the CCD camera. These error values are added to the displacements at each point.

The DIC data points are projected on the VFM mesh nodal locations for the VFM identification process. The scattered DIC points around each node inform each nodal displacement [16]. The uniform grid of DIC points, U_{DIC}, are mapped to the nodal locations of the VFM mesh, $u_{VFM-nodal}$, using the isoparametric coordinates of the DIC locations relative to the VFM mesh to get the relevant shape functions, Φ, solving the equation $\{U_{DIC}\} = [\Phi]\{u_{VFM-nodal}\}$ by using a least-squares minimization algorithm, as previously mentioned in section 3.2.2. The least-squares minimization algorithm to solve this equation is the built-in MATLAB function `mldivide` or \, which solves a system of linear equations of type $Ax = B$. In this system of equations, only the DIC points that are located in the elements connected to each node are used to inform the displacements of that node. In order to determine which DIC points are relevant to a node, first each node is related to a a set of elements, and then the DIC points related to the elements in that set can be related to the node. Another aspect of this algorithm is determining the isoparametric coordinates of the DIC points relative to a parent element of the VFM mesh to get the relevant shape functions, Φ. The method for determining the isoparametric coordinates of each DIC point in at least one VFM element is as follows:

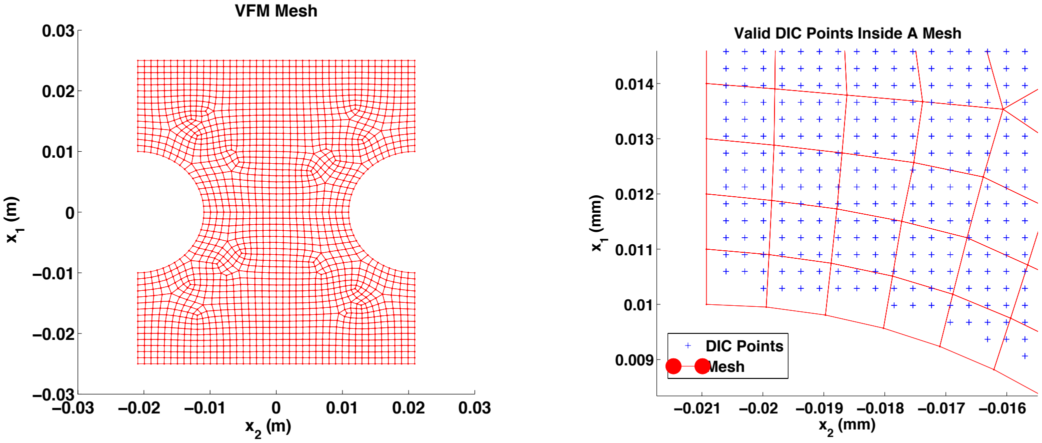
1. Determine if each DIC location is located within or on a bounding box of each element, and note the potential parent elements for each DIC location;

2. Calculate the isoparametric coordinates of a DIC location in each potential parent element; and

3. Associate a DIC location with a parent element only if all of the isoparametric coordinates are between -1 and 1.

This process of DIC-to-VFM mesh data projection requires that every element in the VFM mesh have at least one DIC point. This implies that the VFM cannot be too fine relative to the experimentally-achievable DIC data density since the DIC data cannot be collected near edges of bodies. This balance of VFM mesh and DIC data density must be considered when designing the experiment for the VFM identification, as will be described in section 3.6.2.

This process for simulating the use of DIC data, including data density and experimentally derived error values, allows for detailed characterization of how using DIC data affects the identification process with the VFM. Additionally, this process allows the user to simulate the experimental parameters such as DIC data density and specimen geometry prior to performing the actual experiments, so the user can optimize these parameters for optimal constitutive parameter identification with VFM.



(a) FEM / VFM mesh of a notched tensile specimen

(b) Closeup of the DIC data points relative to the FEM / VFM mesh

Figure 3.7: Example of a VFM mesh and overlay of DIC data locations.

3.5 Evaluation of VFM Objective Function

The VFM identification process for finite deformations requires a nonlinear optimization algorithm to minimize the value of the cost function, Eq. 2.22. The nonlinear optimization algorithm used in this implementation of VFM is the MATLAB SQP algorithm in the `fmincon` function, requiring the user to define an initial guess and lower and upper bounds for the parameters, ξ. The parameters of interest here are the constitutive model parameters that will minimize the cost function. From experience with this MATLAB algorithm, we determined that the algorithm performs faster when the parameters were of similar order of magnitude; thus we scaled all the parameters such that the values of interest would be $\mathcal{O}(0.1)$–$\mathcal{O}(1)$, though we did allow for larger bound than that range.

The SQP algorithm solves a user-defined MATLAB function that returns the value of the cost function, Eq. 2.22, requiring the following inputs:

- Quantities to calculate the Cauchy stress history with the constitutive model:
  - Current scaled ξ,
  - $\mathbf{d}(t_{n+1/2})$: Un-rotated midstep rate of deformation at each integration point for all the elements for all times, and
  - Time steps;
- Quantities to evaluate the internal and external power of the Principal of Virtual Power, Eq. 2.21, in addition to the Cauchy stress history:
  - det$\mathbf{F}$: Determinant of the deformation gradient at each integration point for all the elements for all times,
  - $\mathbf{F}^{-\mathbf{T}}$: Inverse transpose of the deformation gradient at each integration point for all the elements for all times,
  - det$\mathbf{J}$: Determinant of the Jacobian at each integration point for all the elements for all times,
  - $\mathbf{R}$: Rotation matrix at each integration point for all the elements for all times,
  - $\boldsymbol{f}$: Applied external load history for all times, and
  - $\delta\mathbf{v}$: Virtual velocity fields for all times.

The SQP algorithm iteratively updates the parameters, ξ, in order to minimize the cost function, requiring only updating the constitutive model to solve for the Cauchy stress history; this does not require recalculating the kinematics, which only has to occur *once* based on the input displacement data prior to using the SQP algorithm. Therefore the VFM technique is much faster than FEMU and CEGM techniques that require a full FEM run with each update of the constitutive model parameters ξ.

3.6 Results for Plasticity Examples

The plasticity examples to evaluate the VFM identification process were aimed to gradually increase the complexity of the simulated data to go from the ideal “perfect” FEM data to data derived from FEM data in a manner as close to reality as in an experiment using the DIC to measure surface displacements. Each level of complexity was intended to reveal different sources of error in the VFM identification process. As will be described, some error sources were minimized with changes in approach while others were not completely resolved or even addressed, requiring further investigation. This section will focus on those areas that could be well characterized and improved upon, with only brief descriptions of those areas that need further investigation.

The main criteria for whether or not a procedure or aspect of the simulated experiment led to significant error was the comparison of the VFM-identified constitutive model parameters and the FEM-input values. The following is a list of levels of complexity that were considered, some to a greater extent than others:

1. VFM kinematics and constitutive model evaluation code with “perfect” FEM displacement data evaluated on the same original FEM mesh (no mapping of data between different meshes);

2. Different geometries of specimens with “perfect” FEM displacement data evaluated on the same original FEM mesh;

3. Mapping FEM data to a uniform grid similar to that of a DIC measurement and then mapping the uniform grid data onto the VFM mesh;

4. Different selections of virtual fields;

5. Adding experimentally derived noise to the simulated uniform-grid DIC data, and propagating that error to the VFM mesh data;

6. Varying the density of the DIC data and the VFM mesh density; and

7. Geometry of the specimen to varying the heterogeneity of the displacement field in light of the simulated noise, data density, and VFM mesh density.

3.6.1 Uniform Gradient Examples Without Noise

Evaluating the VFM identification capability using the uniform gradient elements involved testing the VFM algorithm using “perfect” FEM displacement data evaluated on the same original FEM mesh (no mapping of data between different meshes) for three plasticity models and different specimen geometries as described in section 3.3.1. The selection of virtual fields is an open topic of research; thus far there has only been a few early studies on virtual field optimization, which focused on small-strain theory [7]. There has been no published research on selection of virtual fields for optimal VFM identification with nonlinear optimization algorithms that are required for finite-deformation VFM implementations; thus, we are left to select virtual fields manually at present. Some rules-of-thumb for virtual fields are to (1) select spatially slowly varying virtual fields as to reduce effects from noisy displacement data, particularly in regions of large gradients; (2) ensure that the virtual fields are kinematically admissible; and, (3) ensure that the boundaries of the virtual fields allow for simple evaluation of the surface integrals in the Principal of Virtual Work / Power. These guidelines tend to lead the user towards simple algebraic and periodic functions. Additionally, the user can select multiple sets of virtual fields in order to increase or decrease the relative contribution of certain areas of the deformed geometry, providing some tailoring ability in the VFM-identification process. Here, two sets of virtual fields were selected for the different geometries. For the notched tensile geometries, the manually selected virtual fields were

Virtual Field Set 1:

$$\delta v_1 = 0 \tag{3.80}$$

$$\delta v_2 = \frac{x_2}{H} \tag{3.81}$$

$$\delta v_3 = \frac{x_3}{h} \sin\left(\frac{\pi x_2}{H}\right) \tag{3.82}$$

$$\boldsymbol{f} \cdot \overline{\delta \boldsymbol{v}} = 2 f_{load} \tag{3.83}$$

and Virtual Field Set 2:

$$\delta v_1 = \frac{x_1(x_2 - H)(x_2 + H)}{WH^2} \tag{3.84}$$

$$\delta v_3 = \frac{x_3(x_2 - H)(x_2 + H)}{hH^2} \tag{3.85}$$

$$\delta v_3 = \frac{x_3(x_2 - H)^2}{hH^2} \tag{3.86}$$

$$\boldsymbol{f} \cdot \overline{\delta \boldsymbol{v}} = 0. \tag{3.87}$$

For the T-shaped geometries, the manually selected virtual fields were

Virtual Field Set 1:

$$\delta v_1 = 0 \tag{3.88}$$

$$\delta v_2 = \Big(\frac{(H-x_2)L}{2\pi H}\Big)\sin\Big(\frac{2\pi x_1}{L}\Big) + \Big(\frac{Hx_1}{2\pi L_a}\Big)\cos\Big(\frac{2\pi x_2}{H}\Big), \quad 0 \leq x_1 \leq L_a \tag{3.89}$$

$$\delta v_2 = \Big(\frac{(H-x_2)L}{2\pi H}\Big)\sin\Big(\frac{2\pi x_1}{L}\Big) + \Big(\frac{H}{2\pi}\Big)\cos\Big(\frac{2\pi x_2}{H}\Big), \quad L_a < x_1 < (L-L_b) \tag{3.90}$$

$$\delta v_2 = \Big(\frac{(H-x_2)L}{2\pi H}\Big)\sin\Big(\frac{2\pi x_1}{L}\Big) + \Big(\frac{H(L-x_1)}{2\pi L_b}\Big)\cos\Big(\frac{2\pi x_2}{H}\Big), (L-L_b) \leq x_1 \leq L \tag{3.91}$$

$$\delta v_3 = \frac{x_3}{h}\sin\Big(\frac{2\pi x_1}{L}\Big)\sin\Big(\frac{2\pi x_2}{H}\Big) \tag{3.92}$$

$$\boldsymbol{f}\cdot\overline{\delta\boldsymbol{v}} = \frac{H}{2\pi}f_{load} \tag{3.93}$$

and Virtual Field Set 2:

$$\delta v_1 = \frac{x_1(x_1-L)}{L}\sin\Big(\frac{2\pi x_2}{H}\Big) \tag{3.94}$$

$$\delta v_2 = \frac{x_1(x_1-L)(H-x_2)}{LH} + \frac{x_1 x_2}{L_a}, \quad 0 \leq x_1 \leq L_a \tag{3.95}$$

$$\delta v_2 = \frac{x_1(x_1-L)(H-x_2)}{LH} + x_2, \quad L_a < x_1 < (L-L_b) \tag{3.96}$$

$$\delta v_2 = \frac{x_1(x_1-L)(H-x_2)}{LH} + \frac{(L-x_1)x_2}{L_b}, \quad (L-L_b) \leq x_1 \leq L \tag{3.97}$$

$$\delta v_3 = -\frac{x_3}{h}\sin\Big(\frac{2\pi x_1}{L}\Big)\sin\Big(\frac{2\pi x_2}{H}\Big) \tag{3.98}$$

$$\boldsymbol{f}\cdot\overline{\delta\boldsymbol{v}} = Hf_{load}. \tag{3.99}$$

Tables 3.7 – 3.10 list the constitutive model parameters used in the FEM models, the initial guesses for the SQP optimization algorithm, and the upper and lower bounds for the SQP optimization algorithm. Note that the lower bound of R_d for the T-shape geometries was higher than the notched specimens, the reason for which will be described later.

Table 3.7: Linear Hardening Parameters and Associated SQP Algorithm Values For All Geometries

Model Parameters	Ref.	Initial Guess	Lower Bound	Upper Bound
E (GPa)	200	10	0.1	500
ν	0.249	0.1	0.01	0.499
σ_y (MPa)	193	100	1	10000
H (GPa)	2.280	1	0.01	1000

Table 3.8: MLEP Parameters and Associated SQP Algorithm Values For All Geometries With $\Delta\bar{\varepsilon}^p_N = 0.02$

Model Parameters	Ref.	Initial Guess	Lower Bound	Upper Bound
E (GPa)	200	10	0.1	500
ν	0.249	0.1	0.01	0.499
σ_y (MPa)	193	100	1	10000
$\Delta\sigma_1$ (MPa)	45.6	100	1	10000
$\Delta\sigma_2$ (MPa)	45.6	100	1	10000
$\Delta\sigma_3$ (MPa)	45.6	100	1	10000
$\Delta\sigma_4$ (MPa)	45.6	100	1	10000

Table 3.9: BCJ Parameters and Associated SQP Algorithm Values For the Notched Tensile Geometry

Model Parameters	Ref.	Initial Guess	Lower Bound	Upper Bound
E (GPa)	200	10	0.1	500
ν	0.249	0.1	0.01	0.499
σ_y (MPa)	193	100	1	10000
H (GPa)	2.280	1	0.01	1000
R_d	1.1	0.1	0.001	10

Table 3.10: BCJ Parameters and Associated SQP Algorithm Values For the T-shaped Geometry

Model Parameters	Ref.	Initial Guess	Lower Bound	Upper Bound
E (GPa)	200	10	0.1	500
ν	0.249	0.1	0.01	0.499
σ_y (MPa)	193	100	1	10000
H (GPa)	2.280	1	0.01	1000
R_d	1.1	2	1	10

The VFM identification process using the uniform gradient element gave mixed results and revealed deficiencies in the element formulation. Tables 3.11 – 3.13 list the VFM-identified parameters for each model and the percent error of these parameters relative to the FEM input reference values. Figure 3.8 includes the true σ - true ϵ plots for each material model based on the FEM input constitutive model parameter values and the VFM-identified constitutive model parameter values. All geometries allowed for reasonable VFM identification of the elastic properties and the initial yield stress for each constitutive model, but VFM had varied success in identifying the hardening properties for each constitutive model. The notched geometries performed better than the T-shape geometries for identifying H. Both geometries allowed for good identification of the first segment of the MLEP model, with increasingly worse identification for later segments.

For the BCJ model, the notched geometries had very poor identification of the recovery term R_d, with the value remaining close to the initial guess of 0.1, though very good identification of the other parameters. When we ran the FEM simulation to five times more y-displacement on the upper boundary of the notch tensile geometries providing more deformation to help with determining R_d, the VFM identification was almost identical for R_d and worse for all the other parameters. When using same the bounds and initial guess for the parameter set for the T-shape as for the notched tensile, the VFM-identified R_d value went to the lower bound of 0.001, which lead to a poor identification of H by an order of magnitude too small, even though the elastic parameters were identified well (similar to those found in the linear hardening case). When the bounds and initial guess were adjusted for the T-shape such that the R_d value could not be below 1, then the optimization algorithm still pushed the value to the lower limit; this essentially reduced the BCJ model to a linear hardening model, which is why the VFM-identified parameters for the T-shape geometry were the same for the linear hardening model and the BCJ model. These behaviors suggest that the objective function is not sensitive enough to the R_d parameter to allow for reasonable identification. This, in combination with the poor identification of the later segments of the MLEP model, suggests that there were not enough regions with large deformation in the geometry to help the algorithm identity the parameters of the plasticity models associated with the large deformation.

In comparing VFM identification for the two geometries, we observed that the T-shape geometry tended to have worse identification values for the plasticity parameters. After some investigation, we determined that a major error source was due to artificial hourglass control present in Sierra SM, which manifests itself as a hourglass energy that is not present in the kinematics and constitutive model evaluations in the VFM code. This leads to incorrect values of the constitutive parameters in order to best match the displacement fields produced from an FEM with artificial hourglass energy. The value for the cost function in the VFM optimization algorithm when using the FEM input values was usually larger than that found to be a minimum in the optimization algorithm by an order of magnitude, implying that the Principal of Virtual Power is not balancing when the FEM input values are used due to the hourglass energy that the VFM code cannot account for. A small error in the energy calculation, which is the area under the true σ - true ϵ curve, would lead to deviation in the true σ - true ϵ curve, and therefore a larger error in the plasticity parameters. Hourglass energy is an consequence of the hourglass control of the uniform gradient element that is a function of the FEM mesh density. Figure 3.9 shows how hourglass energy changes with mesh density in the T-shape geometry 1, where the coarse mesh has 1496 elements, the original mesh has $11,836$ elements, and the fine mesh has $97,200$ elements. This figure shows how much of the total energy of the simulation is hourglass energy as the a function of simulation time. The original mesh used in the VFM identification process has more than 2% hourglass energy by the end of the simulation, which is most likely the largest error source for this geometry. In the notched tensile case with five times more displacement on the y boundary, there would have been more hourglass energy accumulated for the larger displacement FEM analysis than for the shorter one; the fact that the identification overall became worse indicates that the hourglass energy has a significant effect on the quality of the VFM identification process with uniform gradient element.s

Given that the VFM code needs to best approximate the energy evolution of a real experiment, then the VFM code should not rely on the kinematics of the uniform gradient element that requires hourglass control to balance the Principal of Virtual Power. The VFM code would be better formulated if the kinematics of the element did not require any artificial energy source, but could balance the Principal of Virtual Power with just the experimental inputs of displacements and external force. Therefore, we implemented the Q1P0 element, which does not have hourglass energy issues, in the VFM code.

Table 3.11: VFM-Identified Parameters for the Linear Hardening Model Using Uniform Gradient Elements

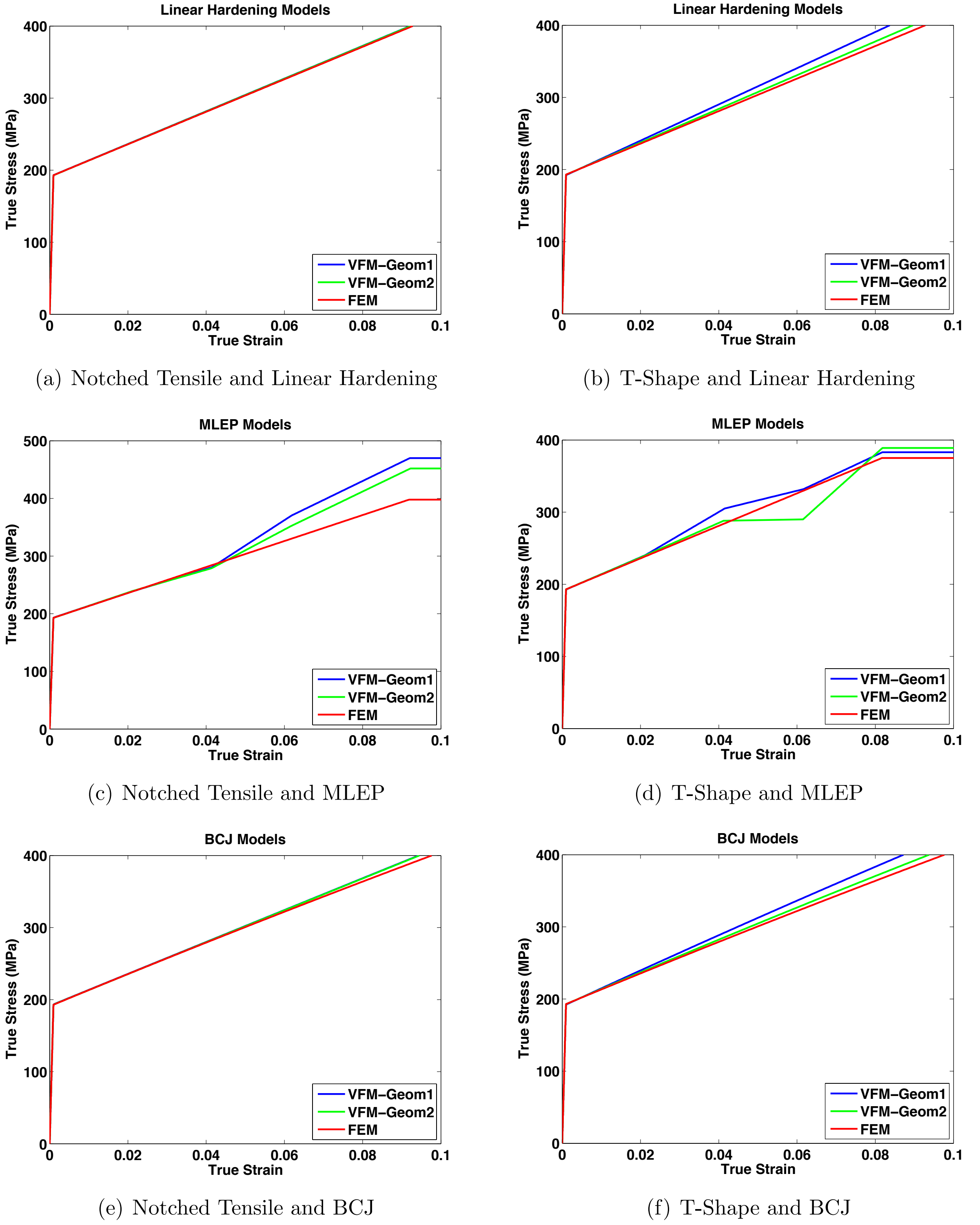
Spec. Geom.	Cost Function Value	E (GPa) Found, Error	ν Found, Error	σ_y (MPa) Found, Error	H (GPa) Found, Error
Notched Geom. 1	4.29 $\times 10^{-11}$	199.9 -0.04%	0.2493 0.12%	193.1 0.05%	2.296 0.70%
Notched Geom. 2	7.43 $\times 10^{-11}$	200.48 -0.24%	0.2484 -0.24%	192.9 -0.05%	2.295 0.66%
T-Shape Geom. 1	8.68 $\times 10^{-4}$	199.8 -0.11%	0.2481 -0.36%	192.2 -0.41%	2.543 11.54%
T-Shape Geom. 2	1.90 $\times 10^{-3}$	200.6 0.31%	0.2452 -1.53%	192.9 -0.05%	2.364 3.68%
Reference Values		200	0.249	193	2.280

Table 3.12: VFM-Identified Parameters for the MLEP Model Using Uniform Gradient Elements

		E		σ_y	$\Delta\sigma_1$	$\Delta\sigma_2$	$\Delta\sigma_3$	$\Delta\sigma_4$
Spec.	Cost	(GPa)	ν	(MPa)	(MPa)	(MPa)	(MPa)	(MPa)
Geom.	Function	Found,	Found,	Found,	Found,	Found,	Found,	Found,
	Value	Error	Error	Error	Error	Error	Error	Error
Notched Geom. 1	6.33 $\times 10^{-11}$	200.02 0.01%	0.2492 0.08%	193.3 0.16%	45.6 0.00%	41.5 −8.99%	90.0 97.37%	100.5 120.39%
Notched Geom. 2	6.66 $\times 10^{-11}$	200.33 0.17%	0.2487 −0.12%	192.9 −0.05%	46.5 1.97%	39.5 −13.38%	73.4 60.96%	99.8 118.86%
T-Shape Geom. 1	8.08 $\times 10^{-4}$	199.63 −0.19%	0.2483 −0.28%	192.7 −0.05%	47.6 4.39%	64.8 42.11%	26.8 −41.23%	51.4 12.72%
T-Shape Geom. 2	1.80 $\times 10^{-3}$	200.5 0.25%	0.2455 −1.41%	192.9 −0.05%	47.1 3.29%	48.6 6.58%	1 −97.81%	100 119.3%
Reference Values		200	0.249	193	45.6	45.6	45.6	45.6

Table 3.13: VFM-Identified Parameters for the BCJ Model Using Uniform Gradient Elements

		E		σ_y	H	
Spec.	Cost	(GPa)	ν	(MPa)	(GPa)	R_d
Geom.	Function	Found,	Found,	Found,	Found,	Found,
	Value	Error	Error	Error	Error	Error
Notched Geom. 1	1.90 $\times 10^{-10}$	199.5 −0.24%	0.2498 0.32%	193.5 0.26%	2.247 −1.45%	0.1005 −90.86%
Notched Geom. 2	5.96 $\times 10^{-11}$	200.0 0.00%	0.2490 0.00%	193.3 0.16%	2.426 −1.49%	0.1007 −90.85%
T-Shape Geom. 1	8.76 $\times 10^{-4}$	199.8 −0.11%	0.2481 −0.36%	192.2 −0.41%	2.545 11.62%	1.000 −9.09%
T-Shape Geom. 2	1.90 $\times 10^{-3}$	200.6 0.30%	0.2452 −1.53%	192.9 −0.05%	2.362 3.60%	1.000 −9.09%
Reference Values		200	0.249	193	2.280	1.10



(a) Notched Tensile and Linear Hardening

(b) T-Shape and Linear Hardening

(c) Notched Tensile and MLEP

(d) T-Shape and MLEP

(e) Notched Tensile and BCJ

(f) T-Shape and BCJ

Figure 3.8: True Stress - True Strain plots of each material model and geometry based on the FEM input values and the VFM-identified parameters (note: these plots do not capture the effect of ν).

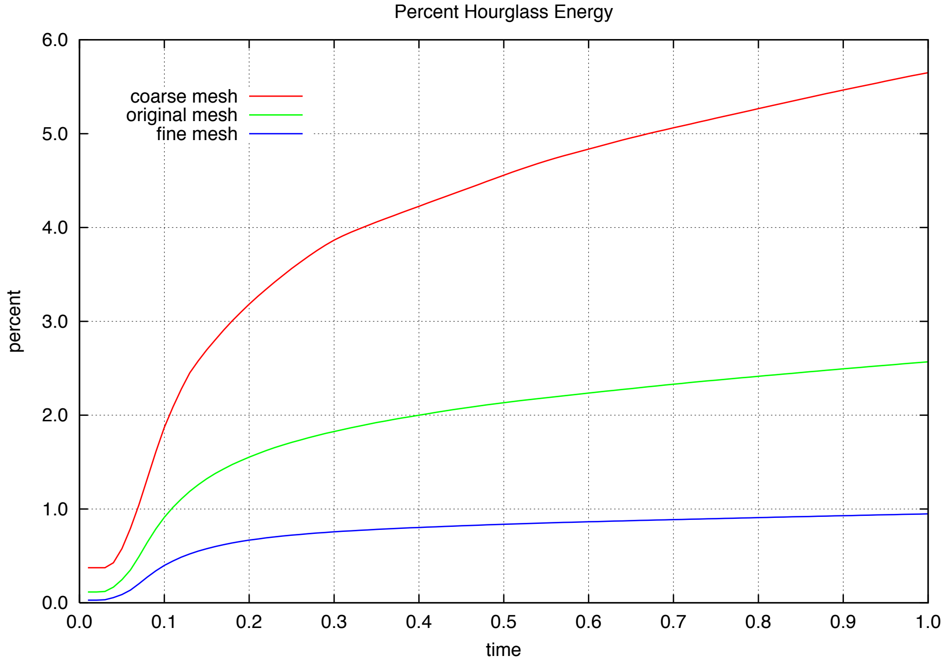


Figure 3.9: Evolution of hourglass energy through the T-shape geometry simulation depending on FEM mesh density.

3.6.2 Q1P0 Examples Without and With Noise

All prior examples focused on demonstrating how the VFM-identification process would reasonably find constitutive model parameters for simulated data with little consideration of aspects of real experimental data. Instead of iteratively demonstrating the finite deformation VFM-identification process with costly and time-consuming DIC experimental data, we first simulated experimental data for VFM so that we could isolate different factors of the experiment that would affect the VFM identification. The main factors considered were (1) transfer of uniform grid DIC displacement data to the VFM mesh; (2) experimental noise in the position and displacement data from DIC; (3) specimen geometry; and, (4) the interplay of VFM mesh density and DIC data density. We also briefly investigated the effect of different manually selected virtual velocity fields. The following examples also demonstrated that the Q1P0 element formulation of VFM does not suffer from the hourglass energy issues of the uniform gradient element.

The finite element analyses with the Q1P0 element used the notched tensile geometry, described in section 3.3.4, because this would be practically feasible in an experiment on 304L stainless steel thin sheets; the T-shape geometry may have a tendency to buckle for large deformations. The global geometry of the notched tensile specimens would be a reasonable size such that the region of interest would fit in a typical DIC camera setup of approximately 84 × 100 mm FOV for the 2048 × 2448-pixel CCD camera (see section 3.4); in other words, the region of interest would stay in the DIC field of view for the duration of an experiment. This size specimen would also be reasonable in terms of maximum load for a 400-kN load frame to pull.

The DIC setup simulated here has a subset size of 25 pixels with either 7 or 12-pixel step sizes, meaning the uniform grid of DIC has a physical step size of 0.286 or 0.490 mm. In this VFM-identification process, the FEM mesh and the VFM mesh are the same. The 7-pixel step size was used for the fine VFM mesh, and the 12-pixel step size was used for the coarse mesh. In these examples, there were two DIC setups to cover the front and back surfaces of the thin-sheet specimen. Even though the algorithm from DIC-to-VFM mesh project could have used the DIC data from both surfaces to enrich nodes on both surfaces of the VFM mesh given that these examples were 1-element thick, the DIC data from the front surface $(+z)$ was projected only on the front surface nodes, and the DIC data from the back surface $(-z)$ was projected only on the back surface nodes. With the DIC data density described above, each DIC setup contributed 1–9 DIC points per element , leading to 6-101 DIC points per node for the DIC-to-VFM mesh data projection. Figure 3.10 shows the coarse FEM / VFM mesh relative to the DIC data for Case A, where there are 18368 DIC data points per surface relative to the 1449 nodes per surface. The simulated DIC data had noise added to the z-position and u-, v-, and w- displacements, with average values of 0.0005 mm, 0.0001 mm, 0.0001 mm, and 0.0005 mm, respectively. This led to small errors in the VFM-mesh displacement fields relative to the original FEM displacement data, but these VFM-mesh displacement errors were larger where there were fewer DIC points at the edges of specimens, particularly around the notches, which happens to coincide with the largest displacements; thus the small DIC noise did have significant effect on the identification, as will be discussed below. Figure 3.11 shows an overlay of the end-of-simulation FEM and VFM-mesh deformed shapes for Case A; the error is indistinguishable on a global level.

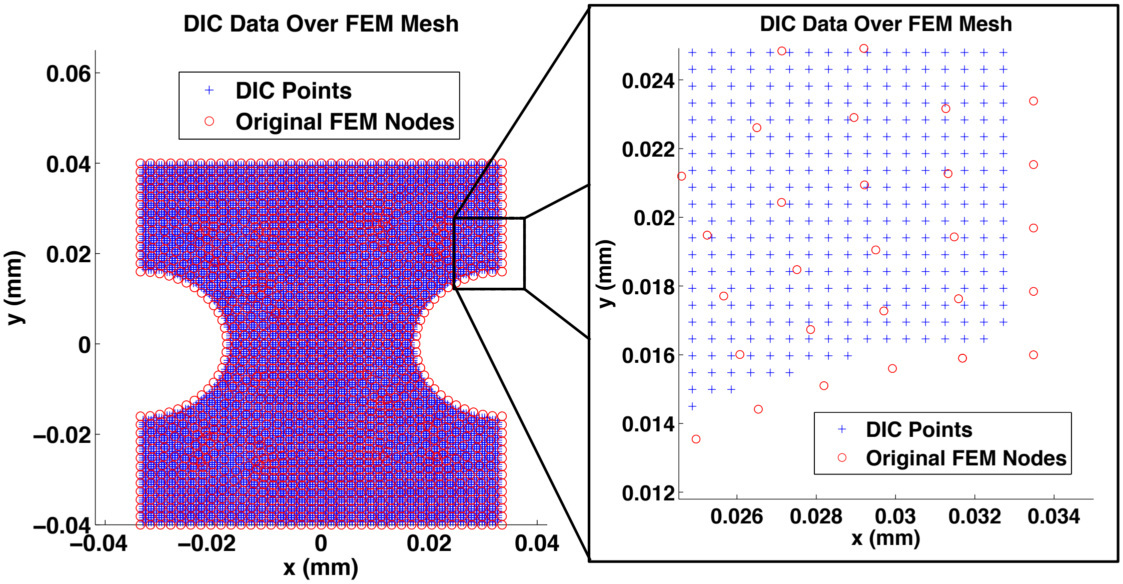


Figure 3.10: Example of a FEM / VFM mesh for Case A and overlay of DIC data locations.

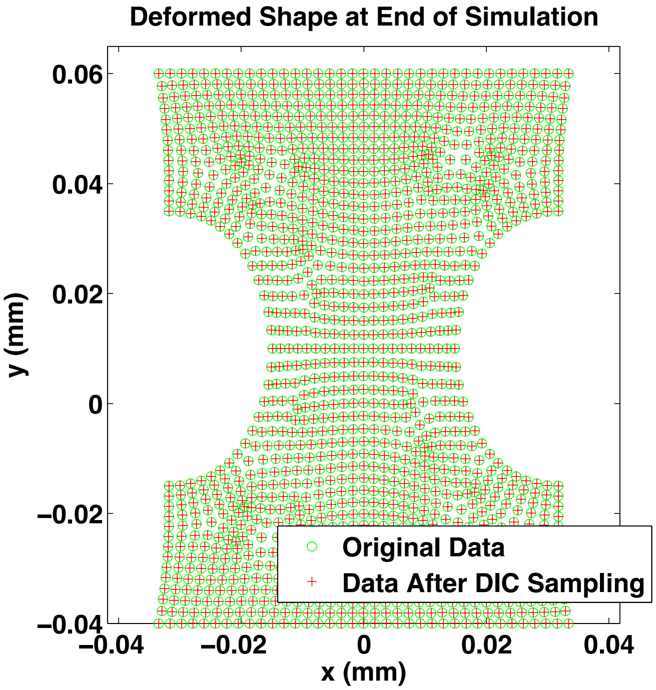


Figure 3.11: Original FEM deformed model with and overlay of the VFM-mesh deformed shaped at the end of the simulation for Case A.

For these VFM-identification runs, several different virtual velocity field sets were considered for Case A. We looked at the error in the cost function when we varied each of the five constitutive model parameters to see how sensitive the combination of the cost function and each virtual velocity set was to error in the parameters. From this, we selected two sets of virtual fields that were the most sensitive to error in the parameters. Figure 3.12 shows the change in in the cost function value with percent error of each of the constitutive model parameters for Case A geometry. The cost function and these virtual fields lack sensitivity to error in R_d as compared to their sensitivity to the other constitutive parameters, but these virtual fields were more sensitive to R_d than other virtual fields considered such as those used in Section 3.6.1. Based on these high-level considerations, we selected the following virtual fields for all of Cases A–I:

Virtual Field Set 1:

$$\delta v_1 = \Big(\frac{2x_2 - H}{H}\Big)\Big(\frac{2x_2 + H}{H}\Big)\cos\left[\frac{\pi x_1}{W}\right] \tag{3.100}$$

$$\delta v_2 = \Big(\frac{2x_2 + H}{H}\Big) \tag{3.101}$$

$$\delta v_3 = 0 \tag{3.102}$$

$$\boldsymbol{f} \cdot \overline{\delta \boldsymbol{v}} = 2 f_{load} \tag{3.103}$$

and Virtual Field Set 2:

$$\delta v_1 = 0 \tag{3.104}$$

$$\delta v_2 = 1 + \sin\left[\frac{\pi x_2}{H}\right] \tag{3.105}$$

$$\delta v_3 = 0 \tag{3.106}$$

$$\boldsymbol{f} \cdot \overline{\delta \boldsymbol{v}} = 2 f_{load}. \tag{3.107}$$

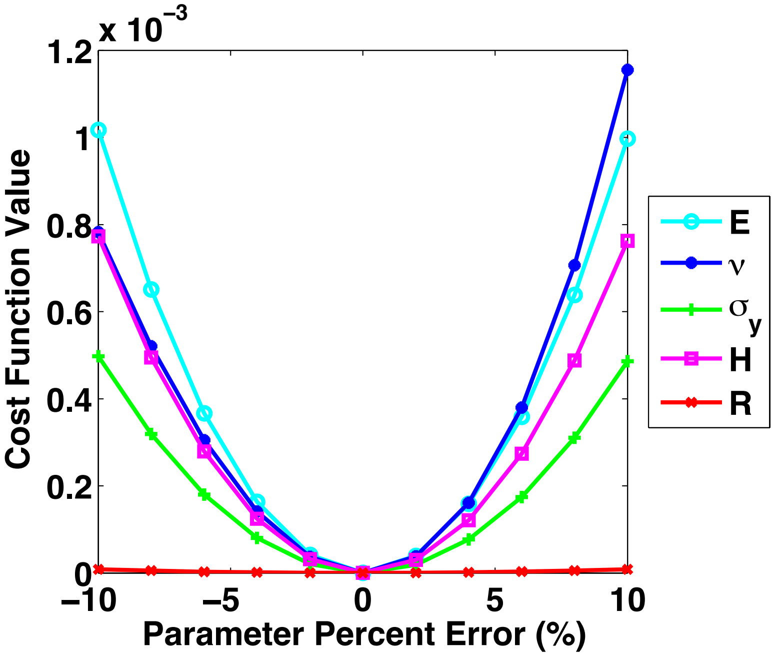


Figure 3.12: Sensitivity of the cost function to error in constitutive parameter value with the chosen virtual fields in Q1P0 geometry case A.

Though we performed the VFM identification for all cases, we will focus the discussion of the results on Cases A, F, and I, which represent the general trends in the results. Table 3.14 and Figure 3.13 summarize the VFM identification results for Cases A, F, and I for the coarse mesh without and with DIC noise, where the average absolute percent error is an average of the absolute values of error for each parameter. The cases without noise underwent the same transfer of data from the FEM data to the DIC uniform grid and then to the VFM mesh as with the case with noise added, so that the results capture error from the numerical method of the transfer of data from the DIC uniform grid to the VFM mesh.

Here, the DIC noise does not necessarily lead to poor identification when compared to the case without noise: the overall error is about the same for each geometry case, and for Cases F and I, the overall error is lower. The identifications for Case A for each parameter are quite different for the noise vs. no-noise cases, though the overall error is about the same. Cases F and I, whether with noise or not, all had better identification than Case A; this implies that the asymmetry in the geometry (for virtual fields that are relatively symmetric over the body) appears to improve the identification. In fact, Case I with the most asymmetry of the three geometries had the best identification of parameters.

The hardest parameters to identify appear to be ν and R_d. The difficulty with identifying R_d is likely due to the cost function not being sensitive to error inR_d for the amount of deformation in these finite element analyses (recall Figure 3.12). We are not certain why Poisson's ratio is difficult to identify given that the cost function is quite sensitive to that parameter; perhaps the ensemble multi-dimensional cost function surface (not investigated here) allows for a larger variation in ν when trying to identify all the parameters at once than when only varying a single parameter.

The presence of DIC noise increases the minimum cost function value in each geometry by two orders of magnitude but without marked difference in the quality of the identification, suggesting that the cost function surface increases in value without drastically changing its overall shape due to the DIC noise. This also implies that the identification is not very sensitive to DIC noise for these geometries with this coarse mesh and DIC data density. Surprisingly, the lower minimum cost function values do not necessarily correspond to better parameter identification. We had originally hypothesized that geometries with lower minimum cost function values would have better parameter identification; we were going to use the absolute cost function value as a criterion for specimen geometry down-selection, but these results tell us that that notion is not a viable criterion.

Figure 3.14 shows the true σ - true ϵ plots based on the FEM input constitutive model parameter values and the VFM-identified constitutive model parameter values for the coarse meshes of Cases A, F, and I with added DIC noise. The small variations in the BCJ model parameters found in these cases lead to similar true σ - true ϵ plots that are nearly indistinguishable from the original input to the FEM analyses, even in the presence of the DIC noise. Thus, the variation in parameters between cases appear to balance out to achieve good agreement with the model used in the FEM analyses.

Table 3.14: VFM-Identified BCJ Model Parameters for the Coarse Mesh Q1P0 Geometries With and Without DIC Noise

Case	Added DIC Noise	Cost Function Value	E (GPa) Found, Error	ν Found, Error	σ_y (MPa) Found, Error	H (MPa) Found, Error	R_d Found, Error	Absolute Average Error
A	No	5.40 $\times 10^{-8}$	196.7 -1.66%	0.260 4.42%	190.3 -1.4%	2.220 -2.64%	1.033 -6.08%	3.24%
A	Yes	1.95 $\times 10^{-6}$	204.9 2.46%	0.240 -3.61%	196.4 1.74%	2.298 0.81%	1.002 -8.93%	3.51%
F	No	3.95 $\times 10^{-8}$	198.7 -0.92%	0.251 0.97%	193.6 0.32%	2.265 -0.64%	1.052 -4.36%	1.44%
F	Yes	2.62 $\times 10^{-6}$	198.1 -0.96%	0.253 1.47%	193.1 0.03%	2.264 -0.69%	1.068 -2.90%	1.21%
I	No	2.45 $\times 10^{-8}$	198.2 -0.88%	0.251 0.97%	193.4 0.20%	2.270 -0.43%	1.069 -2.85%	1.06%
I	Yes	2.68 $\times 10^{-6}$	198.6 -0.72%	0.249 -0.01%	195.0 1.04%	2.281 -0.04%	1.081 -1.71%	0.70%
Reference Values			200	0.249	193	2.28	1.10	
Initial SQP Algo. Guess			205	0.25	190	2.0	1.0	
Lower SQP Algo. Bound			190	0.24	170	0.10	0.60	
Upper SQP Algo. Bound			210	0.26	200	10.00	1.50	

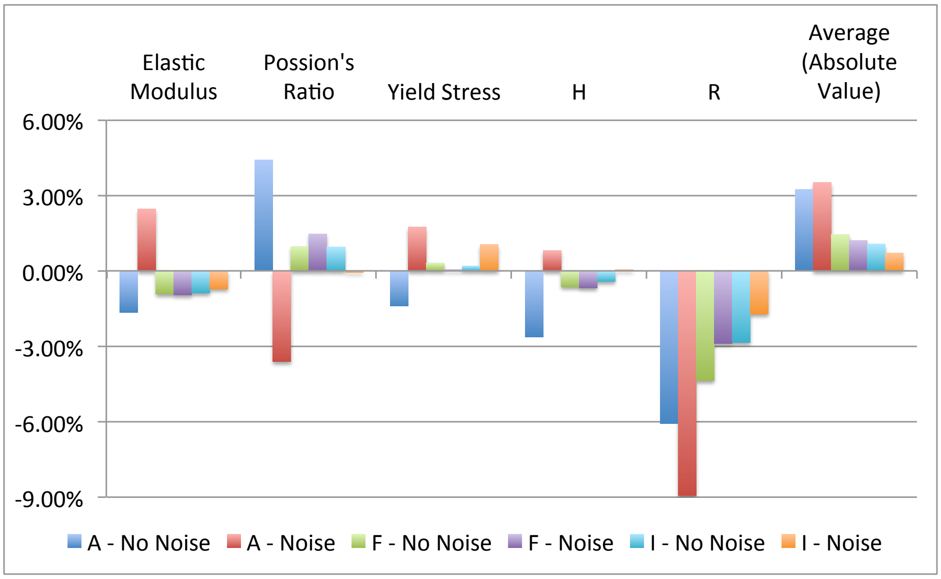


Figure 3.13: Percent errors in VFM identification of BCJ model parameters for coarse meshes with and without added DIC noise.

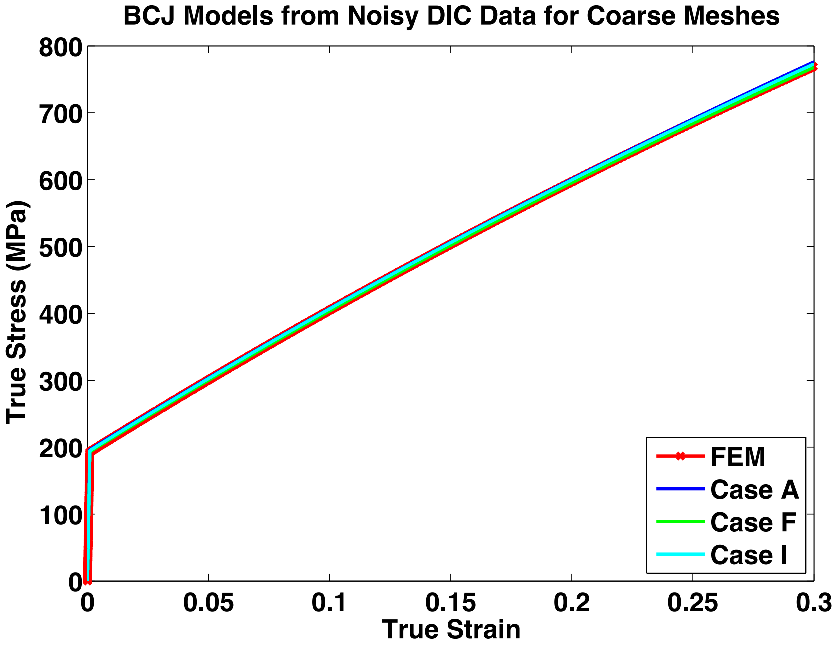


Figure 3.14: True stress - true strain plots based on the BCJ model parameters from the FEM analyses and the VFM identifications for the coarse meshes of Cases A, F and I with DIC noise.

Table 3.15 and Figure 3.15 summarize the VFM identification results for Cases A, F, and I for the coarse and fine meshes with DIC noise. The ratio of DIC points to nodes is approximately 12–13 to 1 for the coarse meshes and about 9–10 to 1 for the fine meshes, which are relatively similar, neutralizing the effect of that ratio when comparing those meshes. It is important to note that the fine mesh cases without noise tended to have similar identification behavior as the coarse mesh cases without noise, so the results for the fine mesh without noise cases are not presented here. The equivalence of the coarse and fine mesh cases without noise suggests that the numerical methods of transferring data do not lead to significant error, but rather that experimental noise and other issues regarding data and mesh density have more effect on error in the identification. We did not consider the less dense DIC data set for the fine mesh because not all of the nodes had DIC points close to them. We also did not consider the finer density DIC data set for the coarse mesh in this study, but that would be an interesting variable to consider more carefully in the future.

The comparison of coarse and fine meshes in the presence of simulated DIC noise shows that the finer mesh density leads to worse identification of nearly every parameter for all geometry cases than the coarse mesh density. Identification of parameters E, σ_y and R_d tend to be too high for the fine mesh cases, and the identification of ν and H tend to be too low for the fine mesh cases. We hypothesize that the coarse meshes filter the DIC noise better than the fine meshes, leading to better identification. Table 3.16 shows the average and maximum error (absolute value of the difference) of the position and displacement nodal values, comparing the VFM mesh noisy data and the original FEM mesh data. The average errors are slightly higher for the fine meshes, but the maximum displacement errors are much larger for the fine meshes. The fine meshes have more DIC error at the edge elements due to slightly fewer DIC points there.

Again, Case I had the best identification of parameters, for both the fine and coarse meshes in the presence of DIC noise, likely due to the greater asymmetry of the geometry, as seen in the comparison of the coarse meshes with and without DIC noise. Figure 3.16 shows the true σ - true ϵ plots based on the FEM input constitutive model parameter values and the VFM-identified constitutive model parameter values for the coarse and fine meshes of the best geometry, Case I, with added DIC noise. Here, the coarse mesh data had better identification, leading to a very slight visible difference in the true σ - true ϵ plots, though the difference is only visible in the large deformation regime.

These issues with the larger effect of noise in the case of the fine meshes on the VFM identification for these geometries, despite the fine meshes having only slightly fewer DIC points per node, leads us to believe that there is a delicate balance of VFM mesh size and DIC data density that must be struck to have optimal identification. Not knowing where that balance is tells us that these numerical studies are invaluable for not only determining the optimal geometry, but also in designing the VFM mesh and the DIC experimental setup.

Overall, the Q1P0-element examples described above demonstrate that the VFM identification process implemented in this study can successfully find constitutive model parameters given global displacement data similar to that generated from full-field DIC experiments. The simulated experiments allowed us to evaluate the effect on theVFM identification of transferring uniform grid DIC data to a VFM mesh, of experimental noise, of specimen geometry, the balance of VFM mesh density and DIC data density, and different virtual velocity fields. A purely experimental program of testing the VFM identification implementation would not have allowed to us to decouple any of these factors. There certainly are many other variables that can be considered to best optimize the VFM identification process, but we believe these were some of the key variables to demonstrate the viability of VFM for identifying finite-deformation constitutive model parameters from full-field DIC displacement data.

Table 3.15: VFM-Identified BCJ Model Parameters for the Coarse and Fine Meshes Q1P0 Geometries With DIC Noise

Case	Coarse	Cost	E (GPa)	ν	σ_y (MPa)	H (MPa)	R_d	Absolute
	or Fine	Function	Found,	Found,	Found,	Found,	Found,	Average
	Mesh	Value	Error	Error	Error	Error	Error	Error
A	Coarse	1.95	204.9	0.240	196.4	2.298	1.002	
		$\times 10^{-6}$	2.46%	−3.61%	1.74%	0.81%	−8.93%	3.51%
A	Fine	3.57	202.3	0.247	200.0	2.188	0.878	
		$\times 10^{-6}$	1.13%	−0.94%	3.63%	−4.03%	−20.21%	5.99%
F	Coarse	2.62	198.1	0.253	193.1	2.264	1.068	
		$\times 10^{-6}$	−0.96%	1.47%	0.03%	−0.69%	−2.90%	1.21%
F	Fine	3.33	208.4	0.240	198.1	2.189	0.879	
		$\times 10^{-6}$	4.20%	−3.61%	2.63%	−3.98%	−20.13%	6.91%
I	Coarse	2.68	198.6	0.249	195.0	2.281	1.081	
		$\times 10^{-6}$	−0.72%	−0.01%	1.04%	−0.04%	−1.71%	0.70%
I	Fine	5.06	209.3	0.240	196.3	2.211	0.966	
		$\times 10^{-6}$	4.67%	−3.61%	1.72%	−3.04%	−12.14%	5.04%

Reference Values	200	0.249	193	2.28	1.10	
Initial SQP Algo. Guess	205	0.25	190	2.0	1.0	
Lower SQP Algo. Bound	190	0.24	170	0.10	0.60	
Upper SQP Algo. Bound	210	0.26	200	10.00	1.50	

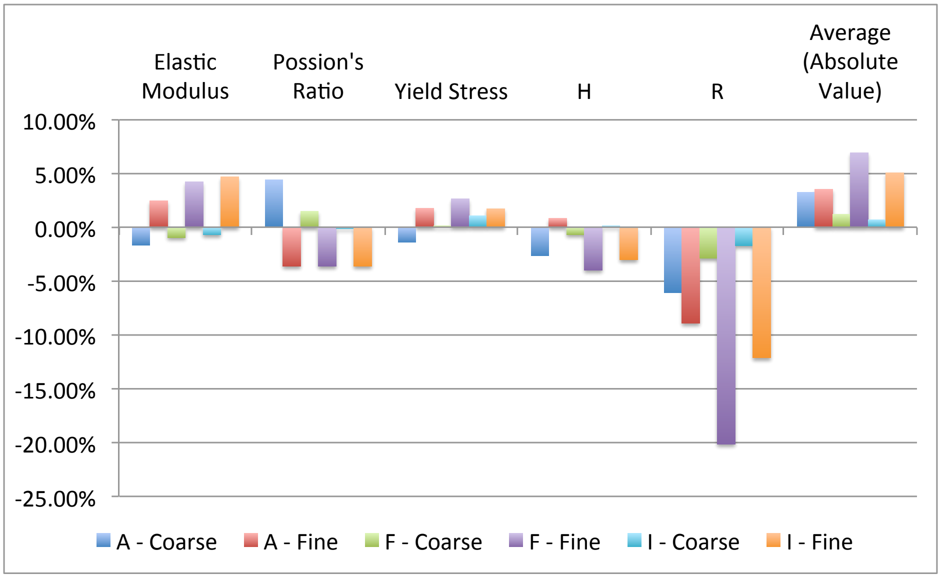


Figure 3.15: Percent errors in VFM identification of BCJ model parameters for coarse fine meshes meshes with added DIC noise.

Table 3.16: Error (μm) in Position and Displacement in VFM Mesh as Compared to the Original FEM Mesh Data Due to DIC Noise

Error	Coarse	Fine
(μm)	Case I	Case I
z (Ave.)	0.269	0.372
z (max.)	0.269	0.372
u (Ave.)	0.057	0.076
u (max.)	1.453	24.35
v (Ave.)	0.059	0.079
v (max.)	1.359	33.95
w (Ave.)	0.266	0.352
w (max.)	6.728	134.4

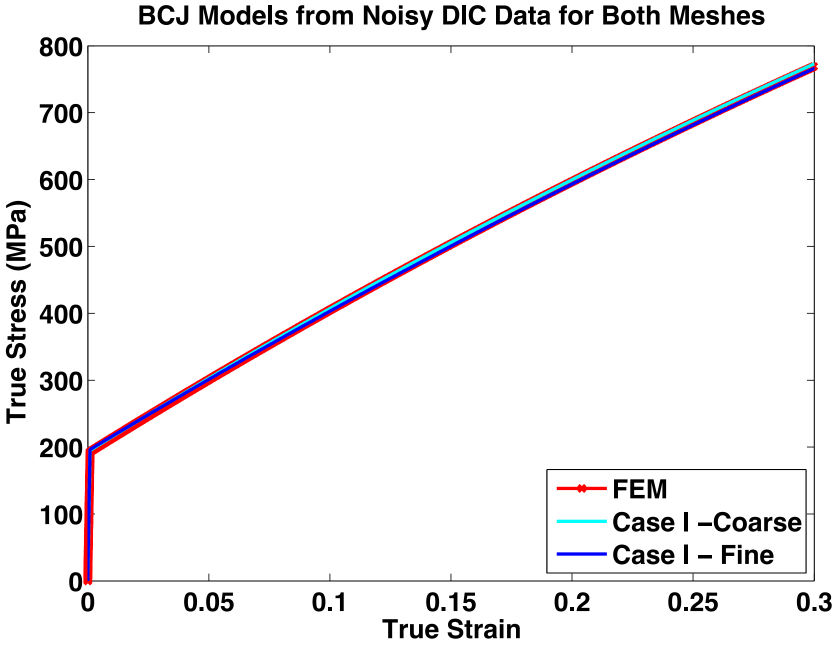


Figure 3.16: True stress - true strain plots based on the BCJ model parameters from the FEM analyses and the VFM identifications for the coarse and fine meshes of Case I with DIC noise.

Chapter 4

Future Work

The main purpose of this study was to implement and evaluate the use of VFM for identification of constitutive model parameters, which was achieved through simulated data. We focused the research on the key aspects of implementation and evaluation, but we did not address in detail several aspects of the simulated experiments, nor did we demonstrate the technique using actual experimental data. The following describes several unresolved research topics for VFM that should eventually be addressed before this technique can confidently be adopted as a reliable alternative to the standard methods.

First, the examples for finite deformation described in section 3.6 utilize the same meshes for the VFM identification as the FEM analyses. Some preliminary work in this area during the course of this study showed that using different meshes for the VFM mesh, such as perhaps a coarser VFM mesh than the FEM analysis, led to very poor identification, even with the Q1P0 elements. One might think that a very fine mesh FEM analysis would best simulate a real material, so then a slightly coarser VFM mesh that could reasonably be populated with DIC data would presumably show similar results as described in section 3.6.2. Unfortunately, that was not the case in the few examples that we tried. We do not have any explanation for this issue. Fortunately, this issue does not automatically disqualify the use of VFM in conjunction with real experimental data since we have seen in the literature many examples of VFM working well with experimental data, including plasticity models [13, 16]. Further research is required to fully understand the FEM-VFM mesh relationship for simulated data.

Second, there is no research in the published literature that address optimal selection of virtual fields for VFM when the identification requires nonlinear optimization algorithms. This is a particularly difficult topic of research since it may depend on the chosen nonlinear optimization algorithm. The choice of virtual field should technically only depend on it being kinematically admissible, but advantageous selection may help with experimental noise and with emphasizing the identification of parts of the constitutive model that are more important or particularly difficult to determine. For example, in the BCJ model used frequently in this study, the R_d parameter was particularly difficult to identify and the cost function we designed was not very sensitive to error in that identified parameter; if more optimal virtual fields were selected, perhaps this parameter would be easier to identify.

Third, the finite-deformation implementation in this study used only one cost function and only one type of nonlinear optimization algorithm. Other cost functions that satisfy the

Principle of Virtual Work / Power may have improved the identification. Other nonlinear optimization algorithms may have been faster computationally and performed better in the presence of DIC noise. These topics are certainly open research topics.

Fourth, the finite-deformation Q1P0 examples of the offset notched tensile geometry in section 3.6 could be tested experimentally and compared to standard inverse methods. In fact, specimens very similar to geometry Cases A, F, and I were fabricated and are ready for testing as of the date of this report, and the standard uniaxial tensile tests for the same lot of 304L stainless steel have previously been completed. The geometry Cases A, F, and I had to be slightly modified for practical experimental reasons; these geometries have been modeled with FEM analyses using the same nominal 304L stainless steel properties used in this report, awaiting VFM identification testing of the simulated data. Figure 4.1 is a schematic of the ungripped region of the offset notched tensile specimen, where the blue region is the region of interest of the DIC setup that can be tracked by the DIC setup for the course of the experiment, and where the red-brown portions (the 12.7-mm lengths above and below the blue region) are outside the region of interest of the DIC setup. Table 4.1 gives the notch geometries for the experimental specimens, where $W = H = 76.2$ mm and thickness $t = 1.5875$ mm. The actual specimens have additional 69.85-mm lengths above and below the ungripped region shown in Figure 4.1 so that the specimen could be gripped in a 400-kN uniaxial load frame in the Sandia Structural Mechanics Laboratory. From the investigation in section 3.6, we expect that Case I will provide the best identification, but those experiments must be completed before any conclusions can be made.

Table 4.1: Geometry for Experimental Specimens of 304L Stainless Steel Sheet.

Case	H_1 (mm)	H_2 (mm)	R_1 (mm)	R_2 (mm)	d_1 (mm)	d_2 (mm)
A	38.1	38.1	15.875	15.875	0	0
F	48.41875	27.78125	15.875	7.9375	0	0
I	48.41875	27.78125	15.875	7.9375	3.96875	0

Fifth, future research on the development of a new hybrid constitutive model parameter identification inverse method is planned for FY15–FY17; this new method will employ both VFM and FEMU techniques. The purpose of the hybrid method is to overcome the main limitation of VFM, which is the requirement of full-field displacement / strain data over the entire *volume* of the deforming body. This method would allow users to perform identification of parameters for a geometry where in the internal displacements / strains are unknown and would be estimated by an FEM analysis iteration. Also, this hybrid technique provides a better objective function, in the form of the Principal of Virtual Work / Power, than normally used in FEMU techniques that usually just consider misfit of the displacement field or force measurement between the current iteration of the FEM analysis and the experiment. The VFM objective function allows for more tailor-ability of the identification process: for example, the user could select virtual fields that more heavily weight the experimental data on the surface than the FEM-derived displacements / strains in the middle of the specimen. Two drawbacks to this technique are that it is computationally expensive to have iterative FEM analyses and that the user must select a single constitutive model for the identification

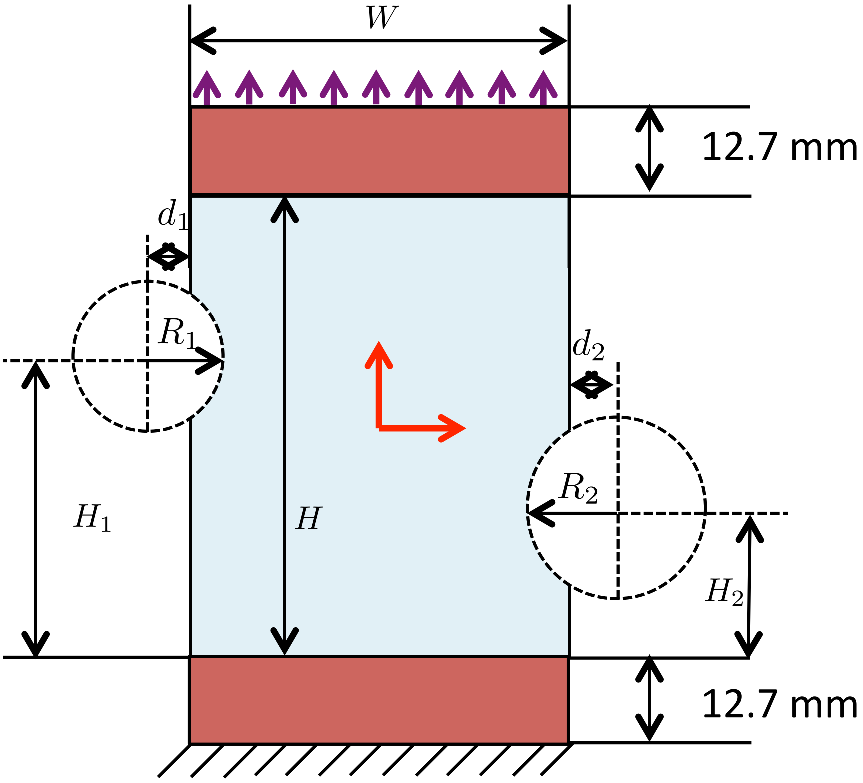


Figure 4.1: Schematic of the ungripped region of the notched tensile experimental 304L stainless steel specimens that await testing.

at the start of the process. These drawbacks reduce the advantage of VFM having relatively small computational effort that allows for investigation of many different constitutive models for the same experimental data set in a short period of time. We anticipate that the improved VFM objective function embedded in the FEMU technique will improve the convergence of the nonlinear optimization of the parameter identification, reducing the number of required FEM iterations.

Chapter 5

Conclusions

VFM is a promising inverse method for constitutive model parameter identification. In this study, we implemented VFM for small-strain linear elasticity and finite deformation plasticity in MATLAB and evaluated the quality of the VFM-identification process using simulated full-field deformation data. The VFM code utilizes the Principle of Virtual Work, full-field strain, and resultant external load for the case of small-strain linear elasticity, but uses the Principle of Virtual Power (integration over time of the Principle of Virtual Work), full-field displacements, and resultant external load for finite deformation plasticity. VFM allows the user to quickly identify constitutive model parameters without requiring iterative FEM analyses, which in turn allows the user to explore several constitutive models for the same set of experimental data in a short period of time. The main drawback of VFM is that the technique requires full-field measurements over the entire volume, which is only possible for thin-sheet applications or when 3-D measurement techniques such as Digital Volume Correlation (DVC) performed using a computed tomography (CT) or MRI are applicable for *in situ* testing. Nevertheless, this technique is very powerful for the regimes in which it is applicable, and can fundamentally change how we calibrate constitutive models.

In this study, we demonstrated that VFM performs very well in small-strain theory and can very simply be used with even a single load step of a specimen. The error sources for this regime are (1) summation approximation of the integrals in the Principle of Virtual Work given discrete strain data, (2) the technique for discretizing a non-rectangular volume with a uniform grid of data, (3) measurement noise from the experimental technique which may be relatively large compared to the absolute measurement for small strains, and, (4) the selection of virtual fields in the presence of measurement noise and discrete data. Even though these issues have been investigated in the literature for elasticity [2], we performed a short demonstration of the method for small-strain theory to show the simplest case of the technique, highlighting the ease of use of VFM.

We also implemented VFM in MATLAB using the Principle of Virtual Power, exploiting FEM kinematics and meshing methods, in conjunction with nonlinear optimization methods to perform the identification of finite-deformation plasticity constitutive models that have a nonlinear behavior over time. We used both uniform gradient elements and Q1P0 elements for the integration techniques. The first evaluation of these two element formulations was a direct use of the FEM-simulated data without any simulation of the experimental technique, DIC, in the data; this approach allowed us to see the best-case scenario where VFM-identified constitutive model parameters should match the input values to the original FEM analyses. We determined that the uniform gradient element formulation of VFM did not capture artificial hourglass energy that is accounted for in the FEM simulation, leading to significant errors in the identification of plasticity model parameters. On the other hand, the Q1P0 formulation, which does not suffer from the hourglass control issue, was able to return the FEM-analyses input values with little error.

To evaluate VFM identification of plasticity constitutive models in light of experimental data considerations, we simulated DIC experimental data, with experimentally derived noise, and implemented all the code necessary to utilize experimental data in our VFM MATLAB code; we then ran the VFM identification for various specimen geometries for a notched tensile specimen with nominal 304L stainless steel properties for the BCJ constitutive model, VFM mesh densities, and DIC data densities. We determined that the Q1P0 VFM implementation did return reasonable identification of the FEM-analyses input values, even in the presence of simulated DIC noise. This evaluation showed that

- Asymmetry in the geometry tends to improve the identification of parameters;
- DIC noise does not necessarily decrease the quality of the VFM identification, though it does increase the minimum value found for the cost function of the nonlinear optimization algorithm;
- Not all parameters are identified as well when all parameters are determined as an ensemble, though comparison of true σ - true ϵ plots based on the FEM-input values and; the VFM-identified values are qualitatively indistinguishable;
- A lower cost function value does not necessarily correspond to better identification of the parameters;
- The numerical method of transferring uniform grid data without noise to the VFM mesh for identification does appear to greatly affect the end result, where good results require sufficient data points around each node of the VFM mesh; and
- There is a delicate balance between VFM mesh density and DIC data density, particularly near the specimen edges where DIC noise tends to get amplified when there are few DIC data points, in order to get good identification.

A numerical study of the effect of DIC noise allowed us to isolate the different factors arising from using experimental data with VFM. An equivalent experimental program would not have allowed that isolation and would have been relatively cost-prohibitive. The numerical study also allowed us to select the best candidate specimen geometries to try experimentally. We identified several areas for future work including mesh sensitivity between the FEM and VFM mesh; optimal selection of virtual fields for plasticity model applications; improvements in the nonlinear optimization algorithm and associated cost function; experimental testing of the notched tensile specimens studied here for Q1P0 elements; and new hybrid inverse methods that extend VFM to geometries where the displacements cannot be measured over the entire volume. Overall, this study demonstrated that VFM is a viable technique for constitutive model parameter identification when using full-field DIC data.

References

[1] S. Avril, M. Bonnet, A. Bretelle, M. Grédiac, F. Hild, P. Ienny, F. Latourte, D. Lemosse, S. Pagano, E. Pagnacco, and F. Pierron. Overview of identification methods of mechanical parameters based on full-field measurements. *Exp. Mech.*, 48:381–402, 2008.

[2] F. Pierron and M. Grediac. *The Virtual Fields Method: Extracting constitutive mechanical parameters from full-field deformation measurements*. Springer, 2012.

[3] M. Grediac, E. Toussaint, and F. Pierron. Special virtual fields for the direct determination of material parameters with the virtual fields method. 1–principle and definition. *Int. J. Solid Struct.*, 39:2691–2705, 2002.

[4] M. Grediac, E. Toussaint, and F. Pierron. Special virtual fields for the direct determination of material parameters with the virtual fields method. 2–application to in-plane properties. *Int. J. Solid Struct.*, 39:2707–2730, 2002.

[5] M. Grediac and F. Pierron. Numerical issues in the virtual fields method. *Int. J. Numer. Meth. Eng*, 59:1287–1312, 2004.

[6] M. Grediac and F. Pierron. Apply the virtual fields method to the identificaiton of elasto-palstic constitutive paramenters. *Int. J. Plasticity*, 22:602–627, 2006.

[7] E. Toussaint, M. Grediac, and F. Pierron. The virtual fields method with piecewise virtual fields. *Int. J. Mech. Sci.*, 48:256–264, 2006.

[8] Y. Pannier, S. Avril, R. Rotinat, and F. Pierron. Identification of elasto-plastic consitutive parameters from statically undetermined tests using the virtual fields method. *Exp. Mech.*, 46:735–755, 2006.

[9] M.A. Sutton, J.H. Yan, S. Avril, F. Pierron, and S.M. Adeeb. Identification of heterogeneous constitutive parameters in a welded specimen: Uniform stress and virtual fields methods for material property estimation. *Exp. Mech.*, 48:451–464, 2008.

[10] S. Avril, F. Pierron, Sutton M.A., and J. Yan. Identification of elasto-visco-plastic parameters and characterization of lüders behavior using digital image correlation and the virtual fields method. *Mech. Mater.*, 40:729–742, 2008.

[11] M. Rossi and F. Pierron. Identification of plastic consitutive parameters at large deformations from three dimensional dispalcement fields. *Comput. Mech.*, 49:53–71, 2012.

[12] M. Rossi and F. Pierron. Identificaiton of the plastic behaviour in the post-necking regime using a three dimentional reconstruction technique. *Key Engr. Mater.*, 504-506: 703–708, 2012.

[13] D. Notta-Cuvier, B. Langrand, E. Markiewicz, and G. Portemont. Identification of

johnson-cook's viscoplastic model parameters using the virtual fields methods: application to titanium alloy ti6al4v. *Strain*, 49:22–45, 2013.

[14] R. G. Budynas. *Advanced Strength and Applied Stress Analysis*. WCB/McGraw-Hill, 1999.

[15] D.J. Bammann, M.L. Chiesa, and G.C. Johnson. Modeling large deformation and failure in manufacturing processes. In T. Tatsumi, E. Watanabe, and E. Kambe, editors, *Theoretical and Applied Mechanics*, pages 359–376. Elsevier Science, 1997.

[16] J.-H. Kim, A. Serpantié, F. Barlat, F. Pierron, and M.-G. Lee. Characterization of the post-necking strain hardening behavior using the virtual fields method. *Int. J. Solids and Struct.*, 50(24):3829–3842, 2013.

[17] D.P. Flanagan and T. Belytschko. A uniform strain hexahedron and quadrilateral with orthogonal hourglass control. *Int. J. Numer. Meth. Eng.*, 17:679–709, 1981.

[18] T. Belytschko, W.K. Lui, and B. Moran. *Nonlinear Finite Elements for Continua and Structures*. Wiley, New York, 2001.

[19] T. J. R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1987.

[20] Detlev Maurer. 8-node hexahedron elements applied to explicit finite element methods. Technical report, University of Applied Sciences Landshut, 2003. URL `https://people.fh-landshut.de/ maurer/hexa8n.pdf`.

[21] P.L. Reu. A study of the influence of calibration uncertainty on the global uncertainty for digital image correlation using a monte carlo approach. *Experimental Mechanics*, 53 (9):1661–1680, 2013.

Appendix A

Linear Elasticity Example Programs

Below is the MATLAB script, `LinEDiskV02.m`, for the linear elasticity example of the diametrically compressed disk without simulated noise in section 3.1.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%           VFM for Linear Elastic Isotropic Plane Stress
%           Diametrically Compressed Disk
%           Using Manually Chosen Virtual Fields
%           Written by Sharlotte Kramer, 13 March 2013
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Nominal Elastic Parameters for 304L SS
% Elastic Modulus
E = 200000; % MPa=N/mm^2
nu = 0.29;

% Consitutive Parameter Input Variable to be found, Q11 & Q12
InputQ11 = E/(1-(nu^2));
InputQ12 = nu*E/(1-(nu^2));

% Dimensions of Disk
% Radius
Rad = 50; % in mm
t = 2; % in mm

% CCD and 2D-DIC parameters
Step = 10; % in pixels
mmPerPix = 2*Rad/2000; % mm / pixel ratio
SmallArea = (Step*mmPerPix)^2; % in mm^2
```

```
% x and y coordinates in mm
x = (-Rad-(Step*mmPerPix/2):(Step*mmPerPix):Rad+(Step*mmPerPix/2));
y = (Rad+(Step*mmPerPix/2):(-Step*mmPerPix):-Rad-(Step*mmPerPix/2));
xMat = zeros(length(y),length(x)); % matrix of x coordinates
yMat = zeros(length(y),length(x)); % matrix of y coordinates
for i=1:length(y)
    xMat(i,:) = x;
end
for i=1:length(x)
    yMat(:,i) = y';
end

% Determining the polar coordinates from the Cartesian
theta = zeros(length(y),length(x));
r = zeros(length(y),length(x));
% using a for loop to calculate a column of n values for each y at every x
for i = 1:length(y)
   % Loop once for each y, for all x calculating one column of the n matrix
   % calculate r and theta for y(i) and all x in this loop
   [th, radius] = cart2pol(x,y(i));
   theta(i,:) = th;
   r(i,:) = radius;
end

figure;
imagesc(x,y,theta)
axis image
axis xy
colorbar

figure;
imagesc(x,y,r)
axis image
axis xy
colorbar

% Digital mask to outline the disk in the matrix
Mask = r<Rad; % in mm
NumDataPoints = sum(sum(Mask)); % number of data points

figure;
imagesc(x,y,r.*Mask)
axis image
axis xy
colorbar
```

```
% Applied Load
P = 7000; % in N

% Stresses in MPa
sigma1 = (-2*P/(pi*t)) * ((((Rad-yMat).*(xMat.^2))...
    ./(((xMat.^2)+((Rad-yMat).^2)).^2))...
    + (((Rad+yMat).*(xMat.^2))./(((xMat.^2)+((Rad+yMat).^2)).^2))...
    - (1/(2*Rad)));
sigma2 = (-2*P/(pi*t)) * ((((Rad-yMat).^3)...
    ./(((xMat.^2)+((Rad-yMat).^2)).^2))...
    + (((Rad+yMat).^3)./(((xMat.^2)+((Rad+yMat).^2)).^2))...
    - (1/(2*Rad)));
sigma6 = (2*P/(pi*t)) * (((((Rad-yMat).^2).*xMat)...
    ./(((xMat.^2)+((Rad-yMat).^2)).^2))...
    - ((((Rad+yMat).^2).*xMat)./(((xMat.^2)+((Rad+yMat).^2)).^2)));

% Strains
epsilon1 = ((InputQ11/((InputQ11^2)-(InputQ12^2)))*sigma1)...
    - ((InputQ12/((InputQ11^2)-(InputQ12^2)))*sigma2);
epsilon2 = -((InputQ12/((InputQ11^2)-(InputQ12^2)))*sigma1)...
    + ((InputQ11/((InputQ11^2)-(InputQ12^2)))*sigma2);
epsilon6 = (2/((InputQ11)-(InputQ12)))*sigma6;

%% VFM
% For Virtual Field #1
% U1=k2*x
% U2=k2*(-R-y)
% For Virtual Field #2
% U1=k2*x
% U2=0

% "A" matrix
A = zeros(2,2);
A(1,1) = sum(sum(epsilon2.*Mask));
A(2,2) = sum(sum(epsilon2.*Mask));
A(1,2) = sum(sum(epsilon1.*Mask));
A(2,1) = sum(sum(epsilon1.*Mask));

% "B" matrix
B = [(-2*P*Rad)/(t*SmallArea);0];

% Solving for the Q matrix
Q = A\B;
```

```
% Q = 1.0e+05 * [2.1900;0.6319]
% [InputQ11;InputQ12] = 1.0e+05 * [2.1836; 0.6333]
% Error for Q11 and Q12: 0.29% and -0.21%
QError = Q./[InputQ11;InputQ12];

EVFM = Q(1)*(1-((Q(2)/Q(1))^2));
% E from VFM: 2.0076e+05 (0.38% error)
Eerror = (EVFM-E)/E;

nuVFM = Q(2)/Q(1);
% nu from VFM: 0.2886 (-0.5% error)
nuError = (nuVFM-nu)/nu;

% Total area used / Actual disk area = 1.0004
AreaError = (sum(sum(Mask))*((Step*mmPerPix)^2)/(pi*(Rad^2));
```

Below is the MATLAB script, `LinEDiskNoiseV02.m`, for the linear elasticity example of the diametrically compressed disk *with* simulated noise in section 3.1.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%           VFM for Linear Elastic Isotropic Plane Stress
%           Diametrically Compressed Disk
%           Using Manually Chosen Virtual Fields
%           And Simulated Noise
%           Written by Sharlotte Kramer, 15 March 2013
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%
% Noisy Strains %
%%%%%%%%%%%%%%%%%
% amplitude to Gaussian noise, corresponding to standard deviation
NoiseAmp = 0.000001;
[NNX,NNY] = size(epsilon1);
epsilon1n = epsilon1 + (randn(NNX,NNY)*NoiseAmp);
epsilon2n = epsilon2 + (randn(NNX,NNY)*NoiseAmp);
epsilon6n = epsilon6 + (randn(NNX,NNY)*NoiseAmp);

%Percentage of noise as compared to average strain
NoiseRatio1 = NoiseAmp/(sum(sum(Mask.*abs(epsilon1)))/sum(sum(Mask)))
NoiseRatio2 = NoiseAmp/(sum(sum(Mask.*abs(epsilon2)))/sum(sum(Mask)))
NoiseRatio6 = NoiseAmp/(sum(sum(Mask.*abs(epsilon6)))/sum(sum(Mask)))
```

```
%% VFM
% For Virtual Field #1
% U1=k2*x
% U2=k2*(-R-y)
% For Virtual Field #2
% U1=k2*x
% U2=0

% "A" matrix
An = zeros(2,2);
An(1,1) = sum(sum(epsilon2n.*Mask));
An(2,2) = sum(sum(epsilon2n.*Mask));
An(1,2) = sum(sum(epsilon1n.*Mask));
An(2,1) = sum(sum(epsilon1n.*Mask));

% "B" matrix
Bn = [(-2*P*Rad)/(t*SmallArea);0];

% Solving for the Q matrix
Qn = An\Bn;
QnError = Qn./[InputQ11;InputQ12];

% Identified Parameters and associated errors
EnVFM = Qn(1)*(1-((Qn(2)/Qn(1))^2))
Enerror = (EnVFM-E)/E
nunVFM = Qn(2)/Qn(1)
nuError = (nunVFM-nu)/nu
```

DISTRIBUTION:

1	MS 0359	Donna Chavez, 7911 (electronic copy)
1	MS 0557	Darrick Jones, 1558 (electronic copy)
1	MS 0557	Thomas Bosiljevac, 1558 (electronic copy)
1	MS 0557	Jhana Gorman, 1558 (electronic copy)
1	MS 0840	Edmundo Corona, 1554 (electronic copy)
1	MS 0840	Benjamin Reedlunn, 1554 (electronic copy)
1	MS 0840	Kevin Long, 1554 (electronic copy)
1	MS 0845	Miguel Aguilo, 1542 (electronic copy)
1	MS 0845	Michael Tupek, 1542 (electronic copy)
1	MS 1139	Phillip Reu, 1535 (electronic copy)
1	MS 1318	Laura Swiler, 1441 (electronic copy)
1	MS 1321	Glen Hansen, 1443 (electronic copy)
		,
1	MS 0899	Technical Library, 9536 (electronic copy)

Sandia National Laboratories