

Portland State University

PDXScholar

REU Final Reports

Research Experiences for Undergraduates
(REU) on Computational Modeling Serving the
City

8-2019

Discretization of the Hellinger-Reissner Variational Form of Linear Elasticity Equations

Kevin A. Sweet

Portland State University

Follow this and additional works at: https://pdxscholar.library.pdx.edu/reu_reports



Part of the [Other Mathematics Commons](#)

Let us know how access to this document benefits you.

Citation Details

Sweet, Kevin A., "Discretization of the Hellinger-Reissner Variational Form of Linear Elasticity Equations" (2019). *REU Final Reports*. 12.

https://pdxscholar.library.pdx.edu/reu_reports/12

This Report is brought to you for free and open access. It has been accepted for inclusion in REU Final Reports by an authorized administrator of PDXScholar. Please contact us if we can make this document more accessible: pdxscholar@pdx.edu.

Discretization of the Hellinger-Reissner Variational Form of Linear Elasticity Equations

Kevin Sweet
sweetkev@pdx.edu

August 23, 2019

Abstract

This paper addresses the derivation of the Hellinger-Reissner Variational Form from the strong form of a system of linear elasticity equations that are used in relation to geological phenomena. The problem is discretized using finite element discretization as described in [1]. This allowed the creation of a program that was used to run tests on various domains. The resultant displacement vectors for tested domains are shown at the end of the paper.

1 Background

1.1 Motivation

Linear elasticity is used to describe small, reversible deformations to a structure. This particular set of linear elasticity equations has an application to geological phenomena, such as landslides, but I was unable to meet with the geologist associated with the project to learn the exact nature of the equations' application.

1.2 Previous Work

The stabilized mixed finite element method used to discretize the linear elasticity equations are introduced and described in [1]. The work that is done to get from the strong form of the equations to the discretized form of the Hellinger-Reissner variational form is also described in [2], which is a paper that was completed with this project last year. This paper will additionally include information about computing the linear system that is developed.

2 Methodology

2.1 Deriving the Hellinger-Reissner Variational Form

The work in the two following sections is described in [1] and was also done during the previous summer as a part of this project, detailed in [2].

2.1.1 Linear Elasticity Equations

Let $\Omega \subset \mathbb{R}^3$ be a domain with the boundary $\partial\Omega$. The strong form of the problem is to find a displacement vector $\vec{u} : \Omega \rightarrow \mathbb{R}^3$ and symmetric stress tensor $\sigma : \Omega \rightarrow \mathbb{R}_{\text{sym}}^{3 \times 3}$ such that:

$$\begin{cases} \mathcal{A}\sigma = \epsilon(\vec{u}) & \text{in } \Omega \\ -(\nabla \cdot \sigma) = f & \text{in } \Omega \end{cases} \quad (1)$$

where $\mathbb{R}_{\text{sym}}^{3 \times 3}$ are the symmetric 3×3 matrices, the function $f : \Omega \rightarrow \mathbb{R}^3$ is given, and the symmetric gradient $\epsilon(\vec{u}) : \Omega \rightarrow \mathbb{R}_{\text{sym}}^{3 \times 3}$ and compliance tensor $\mathcal{A}\sigma : \Omega \rightarrow \mathbb{R}_{\text{sym}}^{3 \times 3}$ are defined by

$$\epsilon(\vec{u}) = \frac{1}{2} (\nabla \vec{u} + (\nabla \vec{u})^T) \quad (2)$$

$$\mathcal{A}\sigma = \frac{1}{2\mu} \left(\sigma - \frac{\lambda}{3\lambda + 2\mu} \text{tr}(\sigma) I \right) \quad (3)$$

Next, the equations from (1) are multiplied by test functions $\vec{v} : \Omega \rightarrow \mathbb{R}^3$ and $\tau : \Omega \rightarrow \mathbb{R}_{\text{sym}}^{3 \times 3}$ and integrated over the domain, resulting in the following:

$$\begin{cases} \int_{\Omega} \mathcal{A}\sigma : \tau \, dx = \int_{\Omega} \epsilon(\vec{u}) : \tau \, dx \\ - \int_{\Omega} (\nabla \cdot \sigma) \cdot \vec{v} \, dx = \int_{\Omega} f \cdot \vec{v} \, dx \end{cases} \quad (4)$$

The *double dot product* of two matrices $A, B \in \mathbb{R}^{3 \times 3}$ is defined by

$$A : B = \sum_{i,j=1}^3 a_{ij} b_{ji} \quad (5)$$

whcih is equivalent to $\sum_{i=1}^3 a_i b_i$ with symmetric matrices, such as those in this problem.

2.1.2 Integration by Parts

Using integration by parts, the following equation is derived:

$$\int_{\partial\Omega} \vec{v} \cdot (\tau \vec{n}) \, ds = \int_{\Omega} \nabla \vec{v} : \tau \, dx + \int_{\Omega} \vec{v} \cdot (\nabla \cdot \tau) \, dx \quad (6)$$

where $\vec{v} : \Omega \rightarrow \mathbb{R}^3$ and $\tau : \Omega \rightarrow \mathbb{R}_{\text{sym}}^{3 \times 3}$. When the right hand side of the first equation in (4) is applied, it results in the equation,

$$\int_{\partial\Omega} \vec{u} \cdot (\tau \vec{n}) \, ds = \int_{\Omega} \epsilon(\vec{u}) : \tau \, dx + \int_{\Omega} \vec{u} \cdot (\nabla \cdot \tau) \, dx . \quad (7)$$

2.1.3 The Hellinger-Reissner Variational Form

Applying the equation above to (4) and rearranging terms gives, with the assumption $\vec{u} = 0$ on $\partial\Omega$,

$$\begin{aligned} \int_{\Omega} \mathcal{A}\sigma : \tau \, dx + \int_{\Omega} (\nabla \cdot \tau) \cdot \vec{u} \, dx &= 0 \\ - \int_{\Omega} (\nabla \cdot \sigma) \cdot \vec{v} \, dx &= \int_{\Omega} f \cdot \vec{v} \, dx . \end{aligned} \quad (8)$$

This is equivalent to,

$$\begin{aligned} a(\sigma, \tau) + b(\tau, \vec{u}) &= 0 \\ -b(\sigma, \vec{v}) &= \int_{\Omega} f \cdot \vec{v} \, dx \end{aligned}$$

where

$$a(\sigma, \tau) = \int_{\Omega} \mathcal{A}\sigma : \tau \, dx \quad , \quad b(\tau, \vec{v}) = \int_{\Omega} (\nabla \cdot \tau) \cdot \vec{v} \, dx .$$

The problem is now to find $\vec{u} \in L^2(\Omega; \mathbb{R}^3)$ and $\sigma \in H(\text{div}, \Omega; \mathbb{R}_{\text{sym}}^{3 \times 3})$ such that

$$\begin{cases} a(\sigma, \tau) + b(\tau, \vec{u}) = 0 \\ -b(\sigma, \vec{v}) = \int_{\Omega} f \cdot \vec{v} \, dx \end{cases} \quad (9)$$

for all $\vec{v} \in L^2(\Omega; \mathbb{R}^3)$ and $\tau \in H(\text{div}, \Omega; \mathbb{R}_{\text{sym}}^{3 \times 3})$, where

$$\begin{aligned} L^2(\Omega; \mathbb{R}^3) &= \left\{ \vec{v} : \Omega \rightarrow \mathbb{R}^3 : \int_{\Omega} \vec{v} \cdot \vec{v} \, dx < \infty \right\}, \\ H(\text{div}, \Omega; \mathbb{R}_{\text{sym}}^{3 \times 3}) &= \left\{ \tau : \Omega \rightarrow \mathbb{R}_{\text{sym}}^{3 \times 3} : \int_{\Omega} \tau : \tau \, dx < \infty \text{ and } \int_{\Omega} (\nabla \cdot \tau) \cdot (\nabla \cdot \tau) \, dx < \infty \right\}. \end{aligned}$$

2.2 Discretization of the Hellinger-Reissner Variational Form

2.2.1 Stabilized Mixed Finite Element Method

The method described in [1] will now be applied. Let $\Omega \subset \mathbb{R}^3$ be a polyhedron partitioned into a tetrahedral mesh $\mathcal{T} = \{K\}$, with $\mathcal{F} = \{F\}$ as the set of all triangular faces in the mesh. The goal is to find $(\sigma_h, \vec{u}_h) \in \Sigma_h \times V_h$ such that

$$\begin{cases} a(\sigma_h, \tau_h) + b(\tau_h, \vec{u}_h) = 0 \\ -b(\sigma_h, \vec{v}_h) + c(\vec{u}_h, \vec{v}_h) = \int_{\Omega} f \cdot \vec{v}_h \, dx \end{cases} \quad (10)$$

for all $\tau_h \in \Sigma_h$ and $\vec{v}_h \in V_h$. The finite dimensional spaces

$$\Sigma_h \subset H(\text{div}, \Omega; \mathbb{R}_{\text{sym}}^{3 \times 3}), \quad V_h \subset L^2(\Omega; \mathbb{R}^3)$$

are defined in the next subsection, and the jump stabilization term is defined as

$$c(\vec{u}_h, \vec{v}_h) = \sum_{F \in \mathcal{F}} h_F \int_F [[\vec{u}_h]] : [[\vec{v}_h]] \quad (11)$$

where h_F is the longest edge on F . For a face on the boundary of Ω ,

$$[[\vec{w}]] := \frac{1}{2} \left(\vec{w} \vec{n}^T + \vec{n} \vec{w}^T \right)$$

where \vec{n} is the outward normal vector to the face. For an edge that is on the interior of the mesh,

$$[[\vec{w}]] := \frac{1}{2} \left(\vec{w}_+ (\vec{n}_+)^T + \vec{n}_+ (\vec{w}_+)^T + \vec{w}_- (\vec{n}_-)^T + \vec{n}_- (\vec{w}_-)^T \right)$$

where K_+ and K_- are adjacent tetrahedra, and \vec{n}_+ and \vec{n}_- are their respective outward normal vectors on the face F .

2.2.2 Defining Σ_h and V_h

The finite dimensional spaces for \vec{u}_h and σ_h are defined by,

$$\Sigma_h = \{ \tau \in C(\Omega; \mathbb{R}_{\text{sym}}^{3 \times 3}) : \tau|_K \in \mathbb{P}_1(K; \mathbb{R}_{\text{sym}}^{3 \times 3}) \text{ for all } K \in \mathcal{T} \} \quad (12)$$

$$V_h = \{ \vec{v} : \Omega \rightarrow \mathbb{R}^3 : \vec{v}|_K \in \mathbb{R}^3 \text{ for all } K \in \mathcal{T} \}. \quad (13)$$

The basis used for V_h is

$$\{ \phi_{K,j} : K \in \mathcal{T} \text{ and } 1 \leq j \leq 3 \} \text{ where, } \phi_{K,j}(x) = \begin{cases} \mathbf{e}_j, & x \in K \\ \mathbf{0}, & x \notin K \end{cases},$$

which has the dimension $3NT$, where NT is the number of tetrahedra in the mesh. This basis will be numbered $\phi_i : 1 \leq i \leq 3NT$, where $i = 3(K-1) + j$. The two spaces from the definition of Σ_h are given by

$$\begin{aligned} C(\Omega; \mathbb{R}_{\text{sym}}^{3 \times 3}) &= \{\alpha : \Omega \rightarrow \mathbb{R}_{\text{sym}}^{3 \times 3} : \alpha_{ij} \in C(\Omega; \mathbb{R})\} , \\ \mathbb{P}_1(K; \mathbb{R}_{\text{sym}}^{3 \times 3}) &= \{\alpha : K \rightarrow \mathbb{R}_{\text{sym}}^{3 \times 3} : \alpha_{ij} \in \mathbb{P}_1(K; \mathbb{R})\} . \end{aligned}$$

A basis of the space $\mathbb{R}_{\text{sym}}^{3 \times 3}$ is given by

$$\begin{aligned} S_1 &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, S_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, S_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \\ S_4 &= \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, S_5 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, S_6 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}. \end{aligned}$$

Each $\alpha \in \Sigma_h$ is given by a linear function ℓ_i that is associated with a single vertex z_j , that is defined by its values at adjacent vertices. This is given by

$$\ell_i(z_j) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}, \quad 1 \leq i, j \leq 4.$$

A formula that gives ℓ_i is

$$\ell_i(x) = 1 - \frac{(x - z_i) \cdot \vec{n}_i}{h_i}$$

where h_i is the perpendicular distance from z_i to its opposite face and \vec{n}_i is the face's outward normal. From this, a basis for $\mathbb{P}_1(K; \mathbb{R}_{\text{sym}}^{3 \times 3})$ is

$$\ell_i S_j : 1 \leq i \leq 4, 1 \leq j \leq 6$$

with a dimension of 24. $\sigma_h(z)$ is given for each vertex z in the mesh, so $\dim(\Sigma_h) = 6NV$ where NV is the number of vertices in the mesh. This basis will be denoted $\psi_k : 1 \leq k \leq 6NT$ where $k = 6(z-1) + j$.

2.2.3 The Linear System

Now, a linear system can be developed to solve for the stress tensor σ_h and the displacement vector \vec{u} . The equations from (10) hold for all $\tau_h \in \Sigma_h$ and $\vec{v}_h \in V_h$ if and only if,

$$\begin{cases} a(\sigma_h, \psi_i) + b(\psi_i, \vec{u}_h) &= 0, \quad 1 \leq i \leq 6NV \\ -b(\sigma_h, \phi_k) + c(\vec{u}_h, \phi_k) &= \int_{\Omega} f \cdot \phi_k dx, \quad 1 \leq k \leq 3NT \end{cases} . \quad (14)$$

Because

$$\vec{u}_h = \sum_{j=1}^{3NT} y_j \phi_j \quad \sigma_h = \sum_{j=1}^{6NV} x_j \psi_j \quad (15)$$

for some coefficient vectors $\mathbf{x} \in \mathbb{R}^{3NT}$, $\mathbf{y} \in \mathbb{R}^{6NV}$, (14) is equivalent to: find $\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \in \mathbb{R}^{6NV+3NT}$ such that,

$$\begin{pmatrix} A & B^T \\ -B & C \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{f} \end{pmatrix} \quad (16)$$

where $A \in \mathbb{R}^{6NV \times 6NV}$, $B \in \mathbb{R}^{3NT \times 6NV}$, $C \in \mathbb{R}^{3NT \times 3NT}$ are given by

$$\begin{aligned} a_{ij} &= a(\psi_j, \psi_i), & 1 \leq i, j \leq 6NV \\ b_{ij} &= b(\psi_j, \phi_i), & 1 \leq i \leq 3NT \quad 1 \leq j \leq 6NV \\ c_{ij} &= c(\phi_j, \phi_i), & 1 \leq i, j \leq 3NT \end{aligned} \tag{17}$$

and $\mathbf{f} \in \mathbb{R}^{3NT}$ is given by $\mathbf{f}_k = \int_{\Omega} f \cdot \phi_k dx$, $1 \leq k \leq 3NT$.

2.3 Computing the Linear System

Here begins work presented in neither [1] nor [2]. The global matrix G , defined as $\begin{pmatrix} A & B^T \\ -B & C \end{pmatrix}$, is a sparse matrix. To compute the solution to the linear system, G is put into a compressed sparse row (CSR) format. The matrix is split into three arrays, the values array, which contains every non-zero entry in row-major order, the columns array, which contains the column number for each non-zero entry, and the row index, which gives the cumulative number of non-zero entries row by row. Their dimensions are as follows:

$$\begin{aligned} \dim(\text{values}) &= nnz \\ \dim(\text{columns}) &= nnz \\ \dim(\text{row index}) &= 6NT + 3NT + 1 \end{aligned}$$

where nnz is the number of non-zero entries in G . To allocate these arrays, matrices A, B , and C were analyzed separately. The sum of the number of non-zeroes for each matrix, which is dependent on NV and NT , gives nnz (i.e. $nnz = nnz_A + 2nnz_B + nnz_C$). To fill the global values array, local A_K and B_K matrices were computed over each tetrahedron K , whose values were then added to the global values matrix.

2.3.1 Matrix A

Each matrix A_K has four ℓ functions, one for each vertex, associated with it. Every vertex combines with each of six components of the $\mathbb{R}_{\text{sym}}^{3 \times 3}$ basis, S . Hence, A_K is a 24×24 (or $4(6) \times 4(6)$) matrix. This covers every non-zero value of the global matrix, because a function ψ_i can only have non-zero results when interacting with a function ψ_j associated with the same vertex or an adjacent vertex, which looping over each tetrahedron covers. For a single entry in A_K ,

$$a_{6(i-1)+m, 6(j-1)+n} = a(\ell_i S_m, \ell_j S_n) = \int_K \mathcal{A}(\ell_i S_m) : (\ell_j S_n) dx \quad 1 \leq i, j \leq 4 \quad 1 \leq m, n \leq 6.$$

The rightmost expression simplifies to

$$\frac{1}{2\mu} \left(S_m : S_n - \frac{\lambda}{3\lambda + 2\mu} \text{tr}(S_m) \mathcal{I} : S_n \right) \int_K \ell_i \ell_j dx$$

by using the equation from (3). The two parts of the of the expression result in the following values:

$$\begin{aligned} \frac{1}{2\mu} \left(S_m : S_n - \frac{\lambda}{3\lambda + 2\mu} \text{tr}(S_m) \mathcal{I} : S_n \right) &= \begin{cases} \frac{\lambda + \mu}{\mu(3\lambda + 2\mu)}, & 1 \leq m, n \leq 3 \text{ and } m = n \\ -\frac{\lambda}{2\mu(3\lambda + 2\mu)}, & 1 \leq m, n \leq 3 \text{ and } m \neq n \\ \frac{1}{\mu}, & 4 \leq m, n \leq 6 \text{ and } m = n \\ 0, & \text{else} \end{cases} \\ \int_K \ell_i \ell_j &= \begin{cases} \frac{|K|}{10}, & i = j \\ \frac{|K|}{20}, & i \neq j \end{cases} \end{aligned}$$

where $|K|$ is the volume of the tetrahedron. Using this, a Kronecker Product can be created between the two parts of the expression to give the local matrix A_K , as seen below:

$$A_K = \frac{|K|}{20} \begin{pmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 2 \end{pmatrix} \otimes \begin{pmatrix} \frac{\lambda+\mu}{\mu(3\lambda+2\mu)} & -\frac{\lambda}{2\mu(3\lambda+2\mu)} & -\frac{\lambda}{2\mu(3\lambda+2\mu)} & 0 & 0 & 0 \\ -\frac{\lambda}{2\mu(3\lambda+2\mu)} & \frac{\lambda+\mu}{\mu(3\lambda+2\mu)} & -\frac{\lambda}{2\mu(3\lambda+2\mu)} & 0 & 0 & 0 \\ -\frac{\lambda}{2\mu(3\lambda+2\mu)} & -\frac{\lambda}{2\mu(3\lambda+2\mu)} & \frac{\lambda+\mu}{\mu(3\lambda+2\mu)} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\mu} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\mu} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\mu} \end{pmatrix}$$

This gives the values to be input into matrix A from each tetrahedron. The structure of the matrix is also revealed from this endeavor, allowing the total number of non-zero entries to be calculated. For the first three basis functions of $\mathbb{R}_{\text{sym}}^{3 \times 3}$, S_{1-3} , associated with a vertex z on the mesh, the number of non-zero entries in A is equal to $3v$ and for S_{4-6} the number of non-zero entries is v , where v is the number of interacting vertices with z , including itself. Thus,

$$nnz_A = \sum_{z=1}^{NV} 12v_z .$$

2.3.2 Matrix B

For each matrix B_K , each vertex function combines with each $\mathbb{R}_{\text{sym}}^{3 \times 3}$ basis, and then interacts with the three ϕ functions over tetrahedra. Therefore, the matrix B_K is a 3×24 matrix. Again, looping over each tetrahedron returns all the values for the matrix B . For a single entry in B_K ,

$$b_{i,6(m-1)+j} = b(\ell_j S_m, \phi_i) = \int_{\Omega} (\nabla \cdot \ell_j S_m) \cdot \phi_i \, dx \quad 1 \leq i \leq 3, \quad 1 \leq j \leq 4, \quad 1 \leq m \leq 6 .$$

When simplified, the rightmost expression equals

$$|K|(S_m \nabla \ell_j) \cdot \phi_i .$$

This results in the following entries in B_K for a single ℓ_j :

$$|K| \begin{pmatrix} \frac{\partial \ell_j}{\partial x} & 0 & 0 & \frac{\partial \ell_j}{\partial y} & \frac{\partial \ell_j}{\partial z} & 0 \\ 0 & \frac{\partial \ell_j}{\partial y} & 0 & \frac{\partial \ell_j}{\partial x} & 0 & \frac{\partial \ell_j}{\partial z} \\ 0 & 0 & \frac{\partial \ell_j}{\partial z} & 0 & \frac{\partial \ell_j}{\partial x} & \frac{\partial \ell_j}{\partial y} \end{pmatrix} .$$

In addition to giving every value of the matrix B , the structure is also given. For each ϕ function, there are three non-zero entries with each vertex on the tetrahedron. Hence,

$$nnz_B = \sum_{K=1}^{NT} 3(3)(4) = \sum_{K=1}^{NT} 36 .$$

2.3.3 Matrix C

The matrix C is computed with values determined the the faces in the tetrahedral mesh. To develop the matrix, a loop over faces determined the values for each of the three ϕ functions over a face for a single

tetrahedron and, if applicable, its adjacent tetrahedron. For ϕ functions over K and its adjacent tetrahedron \widehat{K} ,

$$\begin{aligned} \llbracket \phi_{K,i} \rrbracket : \llbracket \phi_{K,j} \rrbracket &= \begin{cases} \frac{1}{2}(\delta_{ij} + (e_i^T n_+)(e_j^T n_+)), & i = j \\ \frac{1}{2}(e_i^T n_+)(e_j^T n_+), & i \neq j \end{cases} \\ \llbracket \phi_{K,i} \rrbracket : \llbracket \phi_{\widehat{K},j} \rrbracket &= \begin{cases} -\frac{1}{2}(\delta_{ij} + (e_i^T n_+)(e_j^T n_+)), & i = j \\ -\frac{1}{2}(e_i^T n_+)(e_j^T n_+), & i \neq j \end{cases} \end{aligned}$$

$$1 \leq K, \widehat{K} \leq NT \quad 1 \leq i, j \leq 3.$$

Because each ϕ function interacts with every ϕ function on its own tetrahedron or adjacent tetrahedron, there are only non-zero entries for interacting tetrahedra. Thus,

$$nnz_C = \sum_{K=1}^{NT} 9t_K$$

where t_K is the number tetrahedra that interact with tetrahedron K .

This, with nnz_A and nnz_B , gives the total number of non-zero entries in the global matrix, which gives the dimensions of the values and columns arrays. The work done to give the structure of the three pieces also gives the exact location of each non-zero value in the matrix, which allows the arrays to be filled appropriately.

3 Results

The code that was developed did not match the reductions in error as described in [1], however it did show similar behavior. More work would need to be done to identify why the code does not produce accurate results, but the current results over the Fichera Corner will be shown below. First, the two tables showing the error derived from calculations over the unit cube $\Omega = (0, 1)^3$ with $\lambda = .3$ and $\mu = .35$, and whose exact \vec{u} is

$$\vec{u}(x) = \vec{u}(x_1, x_2, x_3) = \begin{pmatrix} 2^4 \\ 2^5 \\ 2^6 \end{pmatrix} x_1(1-x_1)x_2(1-x_2)x_3(1-x_3)$$

from which σ and f can be calculated.

Table 1: Convergence of approximations of the stress tensor and displacement vector from [1].

h	$\ \sigma - \sigma_h\ _{H(\text{div}),\mathcal{A}}$	order	$\ \vec{u}_h\ _C$	order	$\ \vec{u} - \vec{u}_h\ _0$	order
2^{-1}	4.1723E+00	—	4.0747E-01	—	2.4720E-01	—
2^{-2}	2.3595E+00	0.82	3.5554E-01	0.20	1.7403E-01	0.51
2^{-3}	1.2849E+00	0.88	2.5527E-01	0.48	1.1168E-01	0.64
2^{-4}	6.8023E-01	0.92	1.5243E-01	0.74	6.3889E-02	0.81
2^{-5}	3.5167E-01	0.95	8.3310E-02	0.87	3.4309E-02	0.90

Table 2: Convergence of approximations of the stress tensor and displacement vector from our code.

h (arbitrary units)	$\ \sigma - \sigma_h\ _{H(\text{div}),\mathcal{A}}$	order	$\ \vec{u}_h\ _C$	order	$\ \vec{u} - \vec{u}_h\ _0$	order
1	5.93E+00	—	6.36E-01	—	2.99E-01	—
.5	4.71E+00	0.33	5.58E-01	0.19	2.44E-01	0.29
.25	3.42E+00	0.46	4.64E-01	0.27	1.56E-01	0.64
.125	2.66E+00	0.36	3.00E-01	0.62	8.74E-02	0.83

The code that we are currently using clearly does not match the convergence shown in [1]

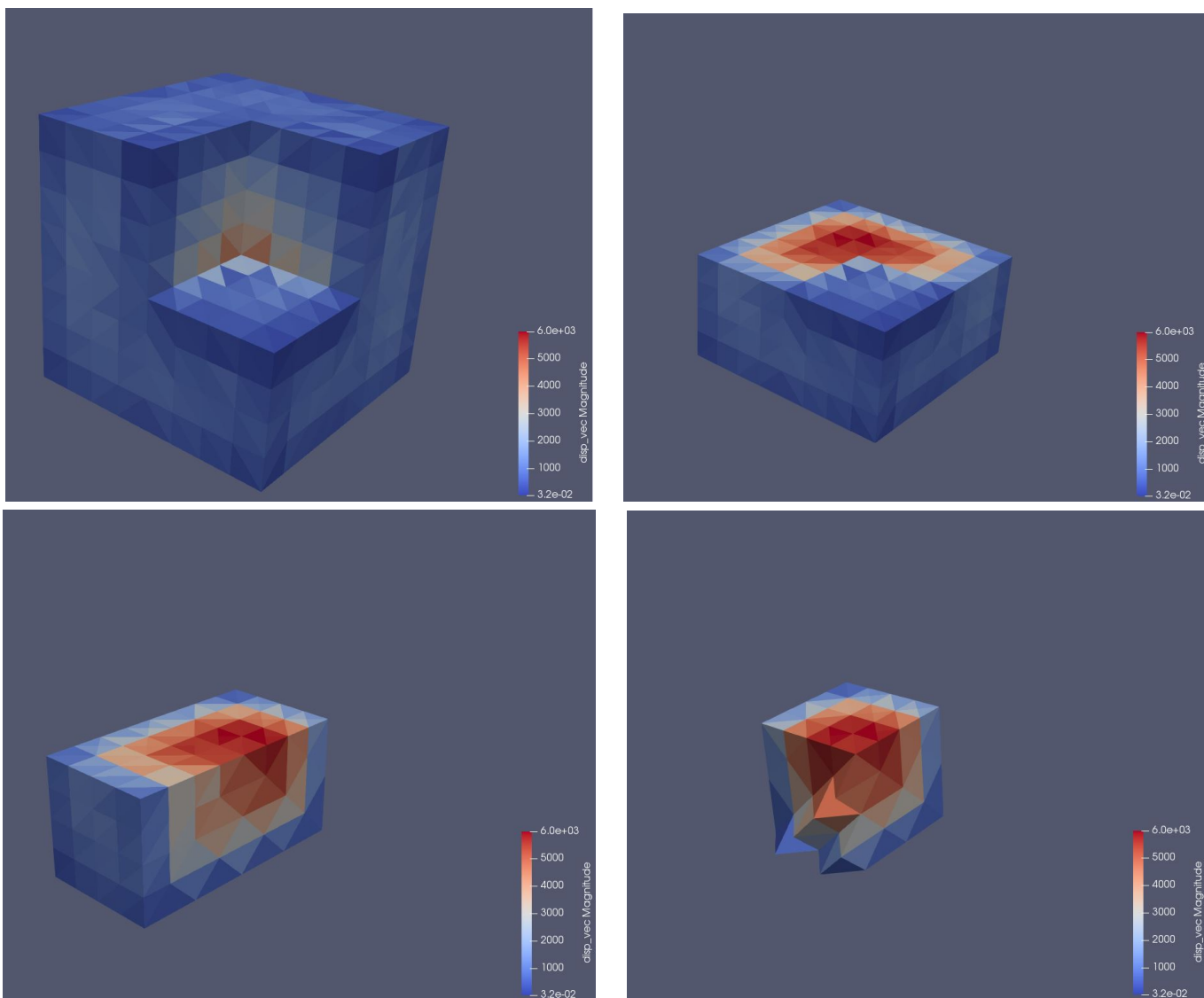


Figure 1: The Fichera Corner cut to view different magnitudes of \vec{u}_h , as calculated by our code.

Acknowledgments

- The REU Site is supported by the National Science Foundation under grant no. 1758006.
- My faculty mentor for this project was Dr. Jeff Oval

References

- [1] L. Chen, J. Hu, and X. Huang. Stabilized Mixed Finite Element Methods for Linear Elasticity on Simplicial Grids in \mathbb{R}^n . *Computational Methods in Applied Mathematics*, 17(1):17–31, January 2017.
- [2] B. Fouts. Derivation of the Hellinger-Reissner Variational Form of the Linear Elasticity Equations, and a Finite Element Discretization. August 2018.